



JBoss Enterprise Application Platform 6.4

Guide d'administration et de configuration

À utiliser dans Red Hat JBoss Enterprise Application Platform 6

JBoss Enterprise Application Platform 6.4 Guide d'administration et de configuration

À utiliser dans Red Hat JBoss Enterprise Application Platform 6

Notice légale

Copyright © 2015 Red Hat, Inc..

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Résumé

Cet ouvrage est un guide d'administration et de configuration de Red Hat JBoss Enterprise Application Platform 6, qui inclut des correctifs publiés.

Table des matières

CHAPITRE 1. INTRODUCTION	15
1.1. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 6	15
1.2. LES FONCTIONNALITÉS DE JBOSS EAP 6	15
1.3. JBOSS EAP 6 OPERATING MODES	16
1.4. LES SERVEURS AUTONOMES	16
1.5. LES DOMAINES GÉRÉS	17
1.6. CONTRÔLEUR DE DOMAINE	18
1.7. DOMAIN CONTROLLER DISCOVERY ET FAILOVER	19
1.8. CONTRÔLEUR HÔTE	20
1.9. LES GROUPES DE SERVEURS	21
1.10. PROFILS JBOSS EAP 6	22
CHAPITRE 2. GESTION DE SERVEURS D'APPLICATIONS	23
2.1. CONVENTIONS POUR LA DOCUMENTATION JBOSS EAP	23
2.2. DÉMARRER ET STOPPER JBOSS EAP 6	23
2.2.1. Démarrer JBoss EAP 6	23
2.2.2. Démarrez JBoss EAP 6 comme un serveur autonome	23
2.2.3. Démarrez JBoss EAP 6 comme domaine géré	24
2.2.4. Configuration d'un nom d'hôte dans un domaine géré	24
2.2.5. Créer un domaine géré sur deux machines	26
2.2.6. Démarrer JBoss EAP 6 avec une configuration différente	27
2.2.7. Stopper le serveur JBoss EAP 6	29
2.2.8. Référence aux variables et arguments à passer à l'exécution du serveur	31
2.3. DÉMARRER ET ARRÊTER LES SERVEURS	34
2.3.1. Démarrer et arrêter les serveurs par l'interface CLI	34
2.3.2. Démarrer un serveur par la console de gestion	36
2.3.3. Stopper un serveur qui utilise une console de gestion	36
2.4. CHEMINS D'ACCÈS AUX SYSTÈMES DE FICHIERS	37
2.4.1. Chemins d'accès aux systèmes de fichiers	37
2.5. FICHIERS DE CONFIGURATION	39
2.5.1. Fichiers de configuration de JBoss EAP 6	39
2.5.2. Remplacement de propriété basée descripteur	41
2.5.3. Activer/Désactiver un remplacement de propriété basé descripteur	43
2.5.4. Expressions imbriquées	45
2.5.5. Historique du fichier de configuration	46
2.5.6. Démarrer le serveur par une ancienne configuration	46
2.5.7. Sauvegarder un snapshot de configuration par l'interface CLI	47
2.5.8. Charger un snapshot de configuration par l'interface CLI.	47
2.5.9. Supprimer un snapshot de configuration par l'interface CLI	48
2.5.10. Lister tous les snapshots de configuration par l'interface CLI	49
CHAPITRE 3. INTERFACES DE GESTION	51
3.1. GESTION DU SERVEUR D'APPLICATIONS	51
3.2. LES API (DE L'ANGLAIS APPLICATION PROGRAMMING INTERFACES) DE GESTION	51
3.3. CONSOLE DE GESTION ET INTERFACE CLI	52
3.4. LA CONSOLE DE GESTION	53
3.4.1. Console de management	53
3.4.2. Se connecter à la console de gestion	53
3.4.3. Changer la langue de la console de management	54
3.4.4. Données analytiques dans la console EAP	55
3.4.5. Activer/désactiver Google Analytics dans la console EAP	55
3.4.6. Configurer un serveur par la console de management	57

3.4.7. Ajouter un déploiement dans une console de management	58
3.4.8. Créer un nouveau serveur dans la console de management	58
3.4.9. Modifier les niveaux de journalisation par défaut en utilisant la console de management	59
3.4.10. Créer un nouveau groupe de service dans la console de gestion	59
3.4.11. Visualisation des journaux dans la console de gestion	60
3.4.12. Intégration du Portail clients dans la Console de gestion	61
Search Customer Portal	61
Open Case	61
Modifier un cas	62
3.5. L'INTERFACE CLI	62
3.5.1. Gestion par interface en ligne de commande (CLI)	62
3.5.2. Lancement de l'interface CLI	62
3.5.3. Quitter l'interface CLI	63
3.5.4. Se connecter à une instance de serveur géré par l'interface CLI	63
3.5.5. Comment obtenir de l'aide par l'interface CLI	63
3.5.6. Utiliser l'interface CLI en mode par lot	64
3.5.7. Commandes CLI Mode Lot	65
3.5.8. Utiliser les opérations et les commandes de l'interface CLI	66
3.5.9. Options de configuration de l'interface CLI	69
3.5.10. Références de commandes d'interface CLI	71
3.5.11. Référence aux opérations d'interface CLI	74
3.5.12. Substitution dans l'interface de commandes CLI	76
3.6. OPÉRATIONS DE L'INTERFACE CLI	77
3.6.1. Afficher les attributs d'une ressource par l'interface CLI	77
3.6.2. Afficher l'utilisateur qui est actif dans l'interface CLI	79
3.6.3. Affiche les informations système et serveur dans l'interface CLI	80
3.6.4. Affiche une description d'opération en utilisant l'interface CLI	80
3.6.5. Afficher les noms de l'opération en utilisant l'interface CLI	82
3.6.6. Afficher les ressources disponibles en utilisant l'interface CLI	83
3.6.7. Afficher les descriptions de ressources disponibles en utilisant l'interface CLI	88
3.6.8. Charger à nouveau le serveur d'applications à l'aide du CLI	89
3.6.9. Fermer le serveur d'applications par l'interface CLI	89
3.6.10. Configurer un attribut à l'aide du CLI	90
3.6.11. Configurer les propriétés système par l'interface CLI	92
3.7. HISTORIQUE DES COMMANDES DE L'INTERFACE CLI	97
3.7.1. Utiliser l'historique de commandes à l'aide de l'interface CLI.	97
3.7.2. Afficher l'historique de commandes par interface CLI.	97
3.7.3. Effacer l'historique de commandes par interface CLI.	97
3.7.4. Désactiver l'historique de commandes par l'interface CLI.	98
3.7.5. Activer l'historique des commandes de l'interface CLI	98
3.8. LA JOURNALISATION D'AUDITING DE L'INTERFACE DE GESTION	99
3.8.1. La journalisation d'auditing de l'interface de gestion	99
3.8.2. Activer la Journalisation d'auditing de l'interface de gestion par l'intermédiaire d'un fichier.	99
3.8.3. Activer la journalisation d'auditing de l'interface de gestion par le serveur syslog.	100
3.8.4. Lire une Journalisation d'auditing de l'interface de gestion	101
CHAPITRE 4. GESTION DES UTILISATEURS	103
4.1. CRÉATION D'UTILISATEUR	103
4.1.1. Ajouter un utilisateur pour les interfaces de gestion	103
4.1.2. Passer des arguments au script add-user de la gestion utilisateur	104
4.1.3. Arguments pour la commande Add-user	105
4.1.4. Spécifier des fichiers de propriétés alternatifs pour les informations de gestion des utilisateurs	106
4.1.5. Exemples de lignes de commande de script Add-user	107

CHAPITRE 5. RÉSEAU ET CONFIGURATION DE PORT	110
5.1. INTERFACES	110
5.1.1. Les interfaces	110
5.1.2. Configurer les interfaces	111
5.2. GROUPE DE LIAISONS DE SOCKETS	115
5.2.1. Groupes de liaisons de sockets	115
5.2.2. Configurer les liaisons de sockets	118
5.2.3. Ports de réseau utilisés par JBoss EAP 6	120
5.2.4. Valeurs de décalage des ports pour les groupes de liaisons de sockets	123
5.2.5. Configurer Port Offset (valeurs de décalage de ports)	123
5.2.6. Configuration de la taille d'un message à distance	124
5.3. IPV6	125
5.3.1. Configurer les préférences de JVM Stack d'IPv6 Networking	125
5.3.2. Configurer les déclarations d'interface du réseautage IPv6	126
5.3.3. Configurer les Préférences JVM Stacks des adresses IPv6	126
CHAPITRE 6. GESTION DES SOURCES DE DONNÉES	128
6.1. INTRODUCTION	128
6.1.1. JDBC	128
6.1.2. Bases de données prises en charge par JBoss EAP 6	128
6.1.3. Types de sources de données	128
6.1.4. L'exemple de source de données	128
6.1.5. Déploiement des fichiers -ds.xml	129
6.2. PILOTES JDBC	129
6.2.1. Installer un pilote JDBC avec la console de gestion	129
6.2.2. Installer un pilote JDBC comme core module	130
6.2.3. Adresses des téléchargements de pilotes JDBC	132
6.2.4. Accès aux classes spécifiques à un fournisseur	133
6.3. SOURCES DE DONNÉES NON-XA	134
6.3.1. Créer une source de données non-XA avec les interfaces de gestion	134
6.3.2. Modifier une source de données non-XA par les interfaces de gestion	136
6.3.3. Supprimer une source de données non-XA par les interfaces de gestion	137
6.4. SOURCES DE DONNÉES XA	138
6.4.1. Créer une source de données XA par les interfaces de gestion	138
6.4.2. Modifier une Source de données XA par les Interfaces de gestion	140
6.4.3. Supprimer une Source de données XA par les Interfaces de gestion	142
6.4.4. XA Recovery	143
6.4.4.1. Les modules de recouvrement XA	143
6.4.4.2. Configurer les modules de recouvrement	143
6.5. SÉCURITÉ DES BASES DE DONNÉES	145
6.5.1. Sécurité des bases de données	145
6.6. VALIDATION DE LA CONNEXION À LA BASE DE DONNÉES	146
6.6.1. Configuration des paramètres de validation de connexion de la base de données	146
6.7. CONFIGURATION DES SOURCES DE DONNÉES	148
6.7.1. Paramètres de source de données	148
6.7.2. Les URL de connexion de sources de données	155
6.7.3. Extensions de sources de données	156
6.7.4. Voir les statistiques de sources de données	158
6.7.5. Statistiques de bases de données	158
6.8. EXEMPLES DE SOURCES DE DONNÉES	160
6.8.1. L'exemple de source de données PostgreSQL	160
6.8.2. Exemple de source de données PostgreSQL XA	161
6.8.3. Exemple de source de données MySQL	162

6.8.4. Exemple de source de données MySQL XA	163
6.8.5. L'exemple de source de données Oracle	164
6.8.6. Exemple de source de données d'Oracle XA	165
6.8.7. Exemple de source de données Microsoft SQLServer	167
6.8.8. Exemple de source de données Microsoft SQLServer XA	168
6.8.9. Exemple de source de données IBM DB2	169
6.8.10. Exemple de source de données IBM DB2 XA	170
6.8.11. L'exemple de source de données Sybase	172
6.8.12. L'exemple de source de données Sybase XA	173
CHAPITRE 7. CONFIGURATION DES MODULES	175
7.1. INTRODUCTION	175
7.1.1. Modules	175
7.1.2. Modules globaux	176
7.1.3. Les dépendances de modules	176
7.1.4. Isolement du chargeur de classes d'un sous-déploiement	177
7.2. DÉSACTIVER L'ISOLEMENT DE MODULE DE SOUS-DÉPLOIEMENT POUR TOUS LES DÉPLOIEMENTS	177
7.3. AJOUTER UN MODULE À TOUS LES DÉPLOIEMENTS	178
7.4. CRÉER UN MODULE PERSONNALISÉ	179
7.5. DÉFINIR UN RÉPERTOIRE DE MODULES JBOSS EXTERNE	181
7.6. RÉFÉRENCE	182
7.6.1. Les modules inclus	182
7.6.2. Nommage de modules dynamiques	182
CHAPITRE 8. JSVC	184
8.1. INTRODUCTION	184
8.1.1. Jsvc	184
8.1.2. Démarrer et arrêter JBoss EAP par Jsvc	184
CHAPITRE 9. VALVES GLOBALES	189
9.1. VALVES	189
9.2. VALVES GLOBALES	189
9.3. LES VALVES D'AUTHENTIFICATION	189
9.4. INSTALLATION D'UNE VALVE GLOBALE	189
9.5. CONFIGURATION D'UNE VALVE GLOBALE	190
CHAPITRE 10. DÉPLOIEMENT D'APPLICATIONS	192
10.1. LES DÉPLOIEMENTS D'APPLICATIONS	192
10.2. DÉPLOYER AVEC LA CONSOLE DE GESTION	193
10.2.1. Gérer le déploiement d'application à l'aide de la console de gestion	193
10.2.2. Activer une application déployée à l'aide de la console de gestion	193
10.2.3. Désactiver une application déployée à l'aide de la console de gestion	194
10.2.4. Retirer le déploiement d'une application à l'aide de la Console de gestion	195
10.3. DÉPLOYER PART L'INTERFACE DE COMMANDES CLI	196
10.3.1. Gérer le déploiement d'une application à l'aide de l'interface CLI	196
10.3.2. Déployer une application dans un serveur autonome à l'aide de l'interface CLI	196
10.3.3. Supprimer le déploiement d'une application dans un serveur autonome à l'aide de l'interface CLI	197
10.3.4. Déployer une application dans un domaine géré à l'aide de l'interface CLI	198
10.3.5. Supprimer le déploiement d'une application dans un domaine géré à l'aide de l'interface CLI	198
10.4. DÉPLOYER PAR L'API HTTP	199
10.4.1. Déployer une application par l'API HTTP	199
10.5. DÉPLOYER AVEC LE SCANNEUR DE DÉPLOIEMENT	202
10.5.1. Gérer le déploiement d'applications dans le scanneur de déploiement	202
10.5.2. Déployer une application dans une instance de serveur autonome par un scanneur de déploiement	

	202
10.5.3. Supprimer le déploiement d'une application dans une instance de serveur autonome par un scanner de déploiement	203
10.5.4. Redéploiement d'une application dans une instance de serveur autonome par le scanner de déploiement	205
10.5.5. Référence pour les fichiers de marquage de scanner de déploiement	206
10.5.6. Référence pour attributs de scanner de déploiement	207
10.5.7. Configurer le scanner de déploiement	208
10.5.8. Configurer le scanner de déploiement avec l'interface CLI	208
10.6. DÉPLOYER AVEC MAVEN	211
10.6.1. Gestion du déploiement d'applications dans Maven	212
10.6.2. Déployer une application dans Maven	212
10.6.3. Supprimer le déploiement d'une application dans Maven	213
10.7. CONTRÔLER L'ORDRE DES APPLICATIONS DÉPLOYÉES DANS JBOSS EAP 6	215
10.8. REMPLACEMENT DU DESCRIPTEUR DE DÉPLOIEMENT	215
CHAPITRE 11. CONFIGURATION DE SOUS-SYSTÈMES	217
11.1. APERÇU DE LA CONFIGURATION DU SOUS-SYSTÈME	217
CHAPITRE 12. LE SOUS-SYSTÈME DE JOURNALISATION	218
12.1. INTRODUCTION	218
12.1.1. Logging (Journalisation)	218
12.1.2. Frameworks de journalisations d'applications pris en charge par JBoss LogManager	218
12.1.3. Journalisation Bootup	218
12.1.4. Voir les erreurs de démarrage initial (bootup)	219
12.1.5. Journalisation de Garbage collection	221
12.1.6. Dépendances d'API de journalisation implicites	221
12.1.7. Emplacements de fichiers de journalisation par défaut	221
12.1.8. Filtre les expressions de journalisation	222
12.1.9. Niveaux de journalisation	225
12.1.10. Niveaux de journalisation pris en charge	225
12.1.11. Catégories de journalisation	226
12.1.12. Root Logger	226
12.1.13. Gestionnaires de journaux	227
12.1.14. Types de gestionnaires de journalisation	227
12.1.15. Log Formatters	228
12.1.16. Syntaxe de Formateur de journaux	228
12.2. CONFIGURER LA JOURNALISATION PAR LA CONSOLE DE GESTION	229
12.3. CONFIGURATION DE LOGGING DANS LE CLI	230
12.3.1. Configurer le root logger par le CLI	230
12.3.2. Configurer une Catégorie dans l'interface CLI	232
12.3.3. Configurer un log handler de console dans le CLI	235
12.3.4. Configurer un log handler de fichiers dans le CLI	238
12.3.5. Configurer un log handler périodique dans le CLI	242
12.3.6. Configurer un log handler de taille dans le CLI	247
12.3.7. Configurer un gestionnaire de journal de taille de fichiers périodique en rotation dans l'interface CLI	253
12.3.8. Configurer un log handler async dans le CLI	259
12.3.9. Configurer un gestionnaire personnalisé dans le CLI	263
12.3.10. Configurer un gestionnaire Syslog dans le CLI	265
12.3.11. Configurer un log handler personnalisé dans le CLI	265
12.4. LA JOURNALISATION PAR DÉPLOIEMENT	267
12.4.1. La journalisation par déploiement	267
12.4.2. Désactivation de la journalisation par déploiement	267

12.5. PROFILS DE JOURNALISATION	267
12.5.1. Profils de journalisation	267
12.5.2. Créer un nouveau profil de journalisation par le CLI	268
12.5.3. Créer un profil de journalisation par le CLI	268
12.5.4. Spécifier un profil de journalisation dans une application	269
12.5.5. Exemple de configuration de profil de journalisation	270
12.6. PROPRIÉTÉS DE LA CONFIGURATION DE JOURNALISATION	272
12.6.1. Propriétés root logger	272
12.6.2. Propriétés de catégorie de journalisation	272
12.6.3. Propriétés de log handlers de console	272
12.6.4. Propriétés de log handlers de fichiers	273
12.6.5. Propriétés de log handlers périodiques	274
12.6.6. Propriétés de log handlers de taille	276
12.6.7. Propriétés de gestionnaires de journaux périodiques	278
12.6.8. Propriétés de log handlers async	280
12.7. EXEMPLE DE CONFIGURATION XML DE LOGGING	280
12.7.1. Échantillon de Configuration XML pour root logger	281
12.7.2. Échantillon de Configuration XML pour une catégorie de journalisation	281
12.7.3. Échantillon de configuration XML pour un log handler de console	281
12.7.4. Échantillon de configuration XML pour un gestionnaire de journalisation de fichiers	281
12.7.5. Échantillon de configuration XML pour un log handler périodique	281
12.7.6. Échantillon de configuration XML pour un log handler de taille	282
12.7.7. Échantillon de Configuration XML pour un gestionnaire de journal de taille de fichiers périodique en rotation	282
12.7.8. Échantillon de Configuration XML pour un Log Handler Async	282
CHAPITRE 13. INFINISPAN	283
13.1. INFINISPAN	283
13.2. MODES DE CLUSTERING	283
13.3. CONTENEURS DE CACHE	284
13.4. STATISTIQUES INFINISPAN	286
13.5. ACTIVER LA COLLECTE DES STATISTIQUES D'INFINISPAN	286
13.5.1. Activer la collecte des statistiques d'Infinispan dans un fichier de configuration de démarrage	286
13.5.2. Active la collecte des statistiques d'Infinispan à partir de l'interface CLI	287
13.5.3. Vérifier que la collecte des statistiques d'Infinispan soit activée	288
13.6. JGROUPS	288
13.6.1. JGroups	288
13.7. RÉOLUTION DE PROBLÈMES JGROUPS	288
13.7.1. Les nœuds ne forment pas un cluster	288
13.7.2. Les causes	289
CHAPITRE 14. JVM	290
14.1. JVM	290
14.1.1. Paramètres de configuration de JVM	290
14.1.2. Afficher le statut JVM dans la console de gestion	292
14.1.3. Configuration d'une JVM	292
CHAPITRE 15. SOUS-SYSTÈME WEB	295
15.1. CONFIGURER LE SOUS-SYSTÈME WEB	295
15.2. CONFIGURER LE TIMEOUT DE SESSION HTTP	295
15.3. CONFIGURER LE SOUS-SYSTÈME DU SERVLET/HTTP	296
15.4. REMPLACER L'APPLICATION WEB WELCOME PAR DÉFAUT	299
15.5. PROPRIÉTÉS SYSTÈME	300
15.6. LES COOKIES DE GESTION DE SESSIONS HTTP-ONLY	306

CHAPITRE 16. SOUS-SYSTÈME DE SERVICES WEB	308
16.1. CONFIGURER LES OPTIONS DE SERVICES WEB	308
CHAPITRE 17. HTTP CLUSTERING ET ÉQUILIBRAGE DES CHARGES	310
17.1. CONVENTIONS DE NOM DE SERVEUR HTTP	310
17.2. INTRODUCTION	311
17.2.1. Clusters haute disponibilité (HA) et clusters d'équilibrage des charges	311
17.2.2. Composants pouvant bénéficier de la haute disponibilité (HA)	311
17.2.3. Connecteurs HTTP - Aperçu général	312
17.2.4. Types de noeuds	314
17.3. CONFIGURATION DE CONNECTEUR	315
17.3.1. Définir les pools de thread pour le connecteur HTTP dans JBoss EAP 6	315
17.4. CONFIGURATION DU SERVEUR WEB	318
17.4.1. Le serveur Apache HTTP Autonome	318
17.4.2. Installer le serveur Apache HTTP inclus dans JBoss EAP 6	318
17.4.3. Installer le serveur Apache HTTP dans Red Hat Enterprise Linux (RHEL) 5, 6, et 7 (RPM)	321
17.4.4. Configuration mod_cluster sur httpd	322
17.4.5. Utiliser un serveur web externe comme Web frontal pour les applications JBoss EAP 6.	327
17.4.6. Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes	328
17.5. CLUSTERING	330
17.5.1. Utiliser la communication TCP dans le sous-système de clusterisation	330
17.5.2. Configurer le sous-système JGroup pour une utilisation TCP	330
17.5.3. Désactiver les annonces dans le sous-système mod_cluster.	332
17.5.4. Passez d'UDP à TCP dans HornetQ Clustering	334
17.6. WEB, CONNECTEURS HTTP, ET HTTP CLUSTERING	336
17.6.1. Le connecteur HTTP mod_cluster	336
17.6.2. Configurer le sous-système mod_cluster	336
17.6.3. Installer le module mod cluster dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server (Zip)	349
17.6.4. Installer le module mod cluster dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server (RPM)	352
17.6.5. Configurer les propriétés de Server Advertisement de votre serveur web activé par votre mod_cluster	353
17.6.6. Configurer un nœud de worker de mod_cluster	354
17.6.7. Migration du trafic entre les clusters	360
17.6.8. Configurer fail_on_status pour mod_cluster	361
17.7. APACHE MOD_JK	361
17.7.1. Le connecteur Apache mod_jk HTTP	362
17.7.2. Configurer JBoss EAP 6 pour qu'il communique avec Apache Mod_jk	362
17.7.3. Installer le module jk_mod dans un serveur Apache HTTP (ZIP)	363
17.7.4. Installer le Module_jk_mod dans Apache HTTPD Server (RPM)	367
17.7.5. Référence de configuration pour les Apache Mod_jk Workers	370
17.8. APACHE MOD_PROXY	373
17.8.1. Le connecteur Apache mod_proxy HTTP	373
17.8.2. Installer un connecteur Mod_proxy HTTP sur le serveur Apache HTTPD	373
17.9. MICROSOFT ISAPI	376
17.9.1. Internet Server API (ISAPI)	376
17.9.2. Téléchargement et extraction de Webserver Connector Natives dans Microsoft IIS	376
17.9.3. Configurer Microsoft IIS pour qu'il puisse utiliser ISAPI	377
17.9.4. Configurer le re-directionneur ISAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP 6	378
17.9.5. Configurer le re-directionneur ISAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6	381
17.10. ORACLE NSAPI	384

17.10.1. Netscape Server API (NSAPI)	384
17.10.2. Configurer NSAPI dans Oracle Solaris	384
17.10.3. Configurer le re-directionneur NSAPI Redirector pour qu'il envoie des requêtes de clients à la plate- forme JBoss EAP 6	386
17.10.4. Configurer le re-directionneur NSAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6	387
CHAPITRE 18. MESSAGERIE	390
18.1. INTRODUCTION	390
18.1.1. HornetQ	390
18.1.2. Java Messaging Service (JMS)	390
18.1.3. Styles de messagerie pris en compte	390
18.2. CONFIGURATION DES TRANSPORTS	391
18.2.1. Accepteurs et connecteurs	391
18.2.2. Configuration de Netty TCP	392
18.2.3. Configuration de Netty Secure Sockets Layer (SSL)	394
18.2.4. Configuration de Netty HTTP	396
18.2.5. Configuration de Netty Servlet	397
18.3. JNDI (JAVA NAMING AND DIRECTORY INTERFACE)	399
18.4. DÉTECTION DE CONNEXION MORTE	399
18.4.1. Fermer les ressources de connexions mortes	399
18.4.2. Détection des échecs côté-client	401
18.5. TRAVAILLER AVEC DES MESSAGES VOLUMINEUX	402
18.5.1. Travailler avec des messages volumineux	402
18.5.2. Configurer des messages volumineux d'HornetQ	402
18.5.3. Configurer les paramètres	402
18.6. PAGINATION	403
18.6.1. La pagination	403
18.6.2. Les fichiers de pagination	404
18.6.3. Configuration d'un dossier de pagination	404
18.6.4. Mode de pagination	404
18.7. DIVERTS	406
18.7.1. Exclusive Divert	407
18.7.2. Non-exclusive Divert	407
18.8. CONFIGURATION	408
18.8.1. Configurer le serveur JMS	408
18.8.2. Configuration des paramètres de l'adresse JMS	413
18.8.3. Configurer la messagerie dans HornetQ	418
18.8.4. Activer la journalisation dans HornetQ	418
18.8.5. Configurer HornetQ Core Bridge	419
18.8.6. Configurer un pontage JMS	420
18.8.7. Configurer la re-livraison différée	422
18.8.8. Configurer les adresses de lettres mortes	423
18.8.9. Configurer les adresses d'expiration de messages	423
18.8.10. Référence pour les attributs de configuration d'HornetQ	424
18.8.11. Définir l'expiration des messages	432
18.9. GROUPEMENT DES MESSAGES	433
18.9.1. Groupement des messages	433
18.9.2. Utilisation d'un API Hornet Cor côté client	434
18.9.3. Configurer le serveur pour les clients JMS (Java Messaging Service)	434
18.9.4. Groupement clusterisé	435
18.9.5. Meilleures pratiques avec les groupements clusterisés	436
18.10. LA DÉTECTION DE MESSAGES DUPLIQUÉS	436

18.10.1. Détection de messages dupliqués	436
18.10.2. Utiliser la détection des messages en double pour l'envoi des messages	437
18.10.3. Configurer un cache d'ID dupliqué	438
18.10.4. Utilisation de la détection dupliquée par pontages et par connexions de cluster	438
18.11. PONTAGES JMS	438
18.11.1. Les ponts	438
18.11.2. Créer un pontage JMS	439
18.12. PERSISTANCE	441
18.12.1. Persistance dans HornetQ	441
18.13. HORNETQ CLUSTERING	442
18.13.1. Server Discovery	443
18.13.2. Broadcast Groups	443
18.13.2.1. Groupe de diffusion UDP (User Datagram Protocol)	444
18.13.2.2. Groupe de diffusion JGroups	445
18.13.3. Les groupes discovery	446
18.13.3.1. Configurer un groupe de diffusion UDP (User Datagram Protocol) sur le serveur	446
18.13.3.2. Configurer un groupe discovery JGroups sur le serveur	447
18.13.3.3. Configurer les groupes discovery pour les clients JMS (Java Messaging Service)	448
18.13.3.4. Configuration de discovery pour l'API principal	449
18.13.4. Équilibrage des charges côté serveur	449
18.13.4.1. Configuration des connexions du cluster	450
18.14. HAUTE DISPONIBILITÉ	453
18.14.1. Introduction à la haute disponibilité	453
18.14.2. HornetQ Shared Stores	454
18.14.3. Configurations de stockage d'HornetQ	455
18.14.4. Types de journaux HornetQ	455
18.14.5. Configurer HornetQ avec une topologie dédiée et un store partagé	456
18.14.6. La réplication de messages HornetQ	457
18.14.7. Configurer les serveurs HornetQ pour la réplication	458
18.14.8. High-availability (HA) Failover	459
18.14.9. Déploiements sur les serveurs de sauvegarde HornetQ	460
18.14.10. Modes de basculements HornetQ	460
18.14.11. Le basculement client automatique	460
18.14.12. Le basculement niveau application	461
CHAPITRE 19. SOUS-SYSTÈME DE TRANSACTION	462
19.1. CONFIGURATION DE SOUS-SYSTÈME DE TRANSACTIONS	462
19.1.1. Configuration des transactions	462
19.1.2. Configurer le Transaction Manager (TM) (ou gestionnaire de transactions)	462
19.1.3. Configurez votre source de données pour utiliser l'API de transaction JTA	467
19.1.4. Configuration d'une source de données XA	468
19.1.5. Messages de journalisation de transactions	469
19.1.6. Configurer la journalisation des sous-systèmes de transactions	470
19.2. ADMINISTRATION DES TRANSACTIONS	471
19.2.1. Naviguer et gérer les transactions	471
19.3. RÉFÉRENCES DE TRANSACTIONS	476
19.3.1. Erreurs et exceptions pour les transactions JBoss	476
19.3.2. Limitations sur les transactions JTA	476
19.4. CONFIGURATION ORB	477
19.4.1. CORBA (Common Object Request Broker Architecture)	477
19.4.2. Configurer l'ORB pour les transactions JTS	477
19.5. JDBC OBJECT STORE SUPPORT	478
19.5.1. JDBC Store de transactions	478

CHAPITRE 20. SOUS-SYSTÈME DE MESSAGERIE	481
20.1. UTILISER DES TRANSPORTS PERSONNALISÉS DANS LES SOUS-SYSTÈMES DE MESSAGERIE	481
CHAPITRE 21. ENTERPRISE JAVABEANS	484
21.1. INTRODUCTION	484
21.1.1. Entreprise JavaBeans	484
21.1.2. Entreprise JavaBeans pour Administrateurs	484
21.1.3. Beans Enterprise	484
21.1.4. Session Beans	485
21.1.5. Message-Driven Beans	485
21.2. CONFIGURER LES BEAN POOLS	485
21.2.1. Bean Pools	485
21.2.2. Créer un bean pool	485
21.2.3. Supprimer un bean pool	487
21.2.4. Modifier un bean pool	488
21.2.5. Assigner des beans pools aux beans de session et aux beans basés messages	489
21.3. CONFIGURER LES EJB THREAD POOLS	490
21.3.1. Entreprise Bean Thread Pools	490
21.3.2. Créer un thread pool	491
21.3.3. Supprimer un thread pool	492
21.3.4. Modifier un thread pool	493
21.4. CONFIGURER LES SESSION BEANS	494
21.4.1. Timeout d'accès au session bean	494
21.4.2. Définir les valeurs de timeout d'accès aux beans de session par défaut	495
21.4.3. Timeout de transaction de Session Bean	496
21.5. CONFIGURER LES MESSAGE-DRIVEN BEANS	496
21.5.1. Définir l'adaptateur de ressources par défaut des beans basés-message	497
21.6. CONFIGURER LE SERVICE EJB3 TIMER	498
21.6.1. Service de minuterie EJB3	498
21.6.2. Configurer le service de minuterie EJB3	498
21.7. CONFIGURER LE SERVICE D'INVOCATION ASYNCHRONE EJB	501
21.7.1. Service d'invocations asynchrones EJB3	501
21.7.2. Configurer le thread pool du service d'invocations asynchrones EJB3	501
21.8. CONFIGURER EJB3 REMOTE INVOCATION SERVICE	501
21.8.1. EJB3 Remote Service	501
21.8.2. Configurer EJB3 Remote Service	502
21.9. CONFIGURER LES EJB 2.X ENTITY BEANS	502
21.9.1. EJB Entity Beans	502
21.9.2. Container-Managed Persistence	502
21.9.3. Activer EJB 2.x Container-Managed Persistence	502
21.9.4. Configurer EJB 2.x Container-Managed Persistence	503
21.9.5. Les propriétés de sous-système CMP pour les générateurs de clés HiLo	505
CHAPITRE 22. JAVA CONNECTOR ARCHITECTURE (JCA)	506
22.1. INTRODUCTION	506
22.1.1. Java EE Connector API (JCA)	506
22.1.2. Java Connector Architecture (JCA)	506
22.1.3. Adaptateurs de ressources	507
22.2. CONFIGURATION DU SOUS-SYSTÈME JAVA CONNECTOR ARCHITECTURE (JCA)	507
22.3. DÉPLOYER UN ADAPTATEUR DE RESSOURCES	513
22.4. CONFIGURATION D'UN ADAPTATEUR DE RESSOURCES DÉPLOYÉES	514
22.5. RÉFÉRENCE DE DESCRIPTION D'ADAPTATEUR DE RESSOURCES	520
22.6. AFFICHAGES DES STATISTIQUES DE CONNEXION	525

22.7. STATISTIQUES D'ADAPTATEUR DE RESSOURCES	525
22.8. DÉPLOYER L'ADAPTATEUR DE RESSOURCES WEBSHERE MQ	526
22.9. INSTALLER L'ADAPTATEUR DE RESSOURCES DE JBOSS ACTIVE MQ	532
22.10. CONFIGURER UN ADAPTATEUR DE RESSOURCES JMS STANDARD À UTILISER AVEC UN FOURNISSEUR JMS DE TIERCE PARTIE	532
CHAPITRE 23. HIBERNATE SEARCH	537
23.1. INTRODUCTION À HIBERNATE SEARCH	537
23.1.1. Hibernate Search	537
23.1.2. Aperçu	537
23.1.3. Index Manager	537
23.1.4. Fournisseur de répertoires	538
23.1.5. Worker	538
23.1.6. Installation et opérations de Back End	538
23.1.6.1. Backend	538
23.1.6.2. Lucene	539
23.1.6.3. JMS	539
23.1.7. Stratégies Reader	540
23.1.7.1. Stratégie de partage	540
23.1.7.2. Stratégie de non-partage	541
23.1.7.3. Stratégies de lecture personnalisées	541
23.1.7.4. Configuration de stratégie de lecture	541
23.2. CONFIGURATION	541
23.2.1. Configuration minimum	541
23.2.2. Configurer l'IndexManager	542
23.2.2.1. Basée répertoire	542
23.2.2.2. Near Real Time (NRT)	542
23.2.2.3. Personnalisé	542
23.2.3. Configuration de DirectoryProvider	543
23.2.4. Partitionnement d'indexes	546
23.2.5. Configuration d'un worker	548
23.2.5.1. JMS Master/Esclave Backend	551
23.2.5.2. Noeuds esclaves	552
23.2.5.3. Noeud Master	553
23.2.6. Réglage de l'indexation de Lucene	554
23.2.6.1. Optimiser la performance d'indexation de Lucene	554
23.2.6.2. L'IndexWriter de Lucene	559
23.2.6.3. Configuration des options de performance	559
23.2.6.4. Ajuster la vitesse d'indexation	563
23.2.6.5. Contrôle de la taille de segment	564
23.2.7. Configuration LockFactory	564
23.2.8. Configuration pour la gestion des exceptions	566
23.2.9. Compatibilité de format d'index	566
23.2.10. Désactiver Hibernate Search	567
23.3. MONITORING	567
23.3.1. Monitoring	568
CHAPITRE 24. DÉPLOYER JBOSS EAP 6 DANS AMAZON EC2	569
24.1. INTRODUCTION	569
24.1.1. Amazon EC2	569
24.1.2. Amazon Machine Instances (AMIs)	569
24.1.3. JBoss Cloud Access	569
24.1.4. Fonctionnalités de JBoss Cloud Access	569

24.1.5. Types d'instances Amazon EC2 prises en charge	570
24.1.6. Les AMI Red Hat prises en charge	570
24.2. DÉPLOYER JBOSS EAP 6 DANS AMAZON EC2	571
24.2.1. Aperçu du déploiement de JBoss EAP 6 sur Amazon EC2	571
24.2.2. JBoss EAP 6 non clusterisée	571
24.2.2.1. Instances non-clusterisées	571
24.2.2.2. Instances non clusterisées	571
24.2.2.2.1. Lancer l'instance de JBoss EAP 6 non clusterisée	572
24.2.2.2.2. Déployer une application sur une instance de JBoss EAP 6 non clusterisée	573
24.2.2.2.3. Lancer l'instance de JBoss EAP 6 non clusterisée	574
24.2.2.3. Domaines gérés non clusterisés	575
24.2.2.3.1. Lancer une instance pour qu'elle serve de contrôleur de domaine	575
24.2.2.3.2. Lancer une ou plusieurs instances pour qu'elles servent de contrôleurs hôtes	577
24.2.2.3.3. Tester le domaine géré de JBoss EAP 6 non clusterisée	579
24.2.2.3.4. Configurer Domain Controller Discovery et Failover dans Amazon EC2	580
24.2.3. JBoss EAP 6 clusterisé	582
24.2.3.1. Instances clusterisées	582
24.2.3.2. Clouds privés virtuels	582
24.2.3.3. Créer un VPC (Virtual Private Cloud)	582
24.2.3.4. Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC	583
24.2.3.5. Configurer le routage par défaut du sous-système privé VPC	585
24.2.3.6. IAM (Identity and Access Management)	586
24.2.3.7. Configurer l'installation IAM	586
24.2.3.8. S3 Bucket	587
24.2.3.9. Configurer l'installation S3 Bucket	587
24.2.3.10. Instances clusterisées	588
24.2.3.10.1. Lancer les AMI de JBoss EAP 6 clusterisée	588
24.2.3.10.2. Lancer l'instance de JBoss EAP 6 clusterisée	592
24.2.3.11. Domaines gérés clusterisés	593
24.2.3.11.1. Lancer une instance pour qu'elle serve de contrôleur de domaine de cluster	593
24.2.3.11.2. Lancer une ou plusieurs instances pour qu'elles servent en tant que contrôleurs hôtes de cluster	595
24.2.3.11.3. Tester le domaine géré de JBoss EAP 6 clusterisée	597
24.3. METTRE EN PLACE LE MONITORING DANS JBOSS OPERATIONS NETWORK (JON)	598
24.3.1. AMI Monitoring	599
24.3.2. Prérequis de connectivité	599
24.3.3. Network Address Translation (NAT)	599
24.3.4. Amazon EC2 et DNS	600
24.3.5. Le routage dans EC2	600
24.3.6. Quitter ou Re-démarrer JON	601
24.3.7. Configurer une instance pour vous enregistrer dans le JBoss Operations Network	601
24.4. CONFIGURATION DU SCRIPT UTILISATEUR	601
24.4.1. Paramètres de configuration permanente	602
24.4.2. Paramètres de scripts personnalisés	605
24.5. RÉOLUTION DE PROBLÈMES	606
24.5.1. Résolution de problèmes dans Amazon EC2	606
24.5.2. Information de diagnostic	606
ANNEXE A. RÉFÉRENCES SUPPLÉMENTAIRES	608
A.1. TÉLÉCHARGER LES FICHIERS DU PORTAIL CLIENT DE RED HAT	608
A.2. CONFIGURER LE JDK PAR DÉFAUT DANS RED HAT ENTERPRISE LINUX	608
A.3. RÉFÉRENCE DE JOURNALISATION D'AUDITING DE L'INTERFACE DE GESTION	610

ANNEXE B. HISTORIQUE DES RÉVISIONS **614**

CHAPITRE 1. INTRODUCTION

1.1. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 6

Red Hat JBoss Enterprise Application Platform 6 (JBoss EAP 6) est une plate-forme middleware générée sur la base de standards ouverts et compatibles avec Java Enterprise Edition 6. Elle intègre JBoss Application Server 7 avec un clustering de haute disponibilité, une messagerie, une mise en cache distribuée et autres technologies.

JBoss EAP 6 comprend une nouvelle structure modulaire qui permet aux services d'être activés seulement si nécessaire, améliorant ainsi la vitesse de démarrage.

La console de gestion et l'interface en ligne de commandes (CLI) rendent la modification des fichiers de configuration XML inutile et rajoutent la capacité d'encoder et d'automatiser des tâches.

En plus, JBoss EAP 6 comprend des frameworks de développement et des API pour développer rapidement des applications de Java EE sécurisées et évolutives.

[Rapporter un bogue](#)

1.2. LES FONCTIONNALITÉS DE JBOSS EAP 6

Tableau 1.1. Fonctionnalités de JBoss EAP 6

Fonctionnalité	Description
Certification Java	JBoss Enterprise Application Platform 6 Full Profil et Web Profile certifiés.
Domaine géré	<ul style="list-style-type: none"> • Un domaine géré procure une gestion centralisée d'instances de serveurs multiples et d'hôtes physiques, tandis qu'un serveur autonome autorise une instance de serveur unique. • Gestion de groupe de configuration par-serveur, déploiement, liaisons de socket, modules, extensions et propriétés système. • Gestion centralisée et simplifiée de la sécurité des applications (y compris les domaines de sécurité).
Console de gestion et interface CLI	Interfaces de gestion de serveur autonome ou nouveaux domaines. L'édition des fichiers de configuration XML n'est plus nécessaire. L'interface CLI comprend également un mode batch qui peut encoder et automatiser les tâches de gestion.

Fonctionnalité	Description
La disposition du répertoire est simplifiée	Le répertoire modules contient maintenant les modules du serveur d'applications. Les répertoires communs et spécifiques au serveur lib sont obsolètes. Les répertoires domain et standalone contiennent les artefacts et les fichiers de configuration pour les déploiements autonomes et de domaine respectivement.
Mécanisme de chargement de classes modulaire	Les modules sont chargés et déchargés à la demande. Cela améliore la performance et la sécurité, et permet des démarrages et redémarrages plus rapides.
Gestion des sources de données simplifiée	Les pilotes de base de données peuvent être déployés comme tout autre service. En plus, les sources de données sont créées et gérées directement dans la console de gestion ou l'interface CLI.
Utilisation réduite et plus efficace des ressources	JBoss EAP 6 utilise moins de ressources système et les utilise plus efficacement que dans les versions précédentes. Entre autres avantages, JBoss EAP 6 démarre et s'arrête plus rapidement que JBoss EAP 5.

[Rapporter un bogue](#)

1.3. JBOSS EAP 6 OPERATING MODES

JBoss EAP 6 fournit deux modes d'opération pour les instances de JBoss EAP 6 : serveur autonome ou domaine géré.

Les deux modes diffèrent dans la façon dont les serveurs sont gérés, pas dans leur capacité à traiter les demandes de l'utilisateur final. Il est important de noter que la fonctionnalité de cluster de haute disponibilité (HA) est disponible avec les deux modes de fonctionnement. Un groupe de serveurs autonomes peut être configuré pour former un cluster HA.

[Rapporter un bogue](#)

1.4. LES SERVEURS AUTONOMES

Un mode de serveur autonome est un processus indépendant qui ressemble au mode d'exécution unique des anciennes versions de JBoss EAP.

L'instance de JBoss EAP 6 qui exécute en tant que serveur autonome est une instance unique, qui peut exécuter optionnellement dans une configuration clusterisée.

[Rapporter un bogue](#)

1.5. LES DOMAINES GÉRÉS

Le mode d'opération d'un domaine géré permet la gestion de multiples instances de JBoss EAP 6 à partir d'un seul point de contrôle.

Les collections de serveur JBoss EAP 6 centralement gérées sont connues comme membres d'un domaine. Toutes les instances JBoss EAP 6 d'un domaine partagent une stratégie de gestion en commun.

Un domaine consiste en un contrôleur de domaine, un ou plusieurs contrôleur(s) hôte(s), et zéro ou plusieurs groupes de serveurs par hôte.

Un contrôleur de domaine est un point central à partir duquel le domaine est contrôlé. Il s'assure que chaque serveur est configuré suivant la stratégie de gestion du domaine. Le contrôleur du domaine est également un contrôleur hôte.

Un contrôleur hôte est un hôte physique ou virtuel sur lequel le script `domain.sh` ou `domain.bat` exécute. Les contrôleurs hôtes sont configurés pour déléguer les tâches de gestion de domaine au contrôleur de domaine.

Le contrôleur hôte de chaque hôte interagit avec le contrôleur de domaine pour contrôler le cycle de vie des instances de serveur de l'application exécutant sur son hôte et pour aider le contrôleur de domaine à les gérer. Chaque hôte peut contenir plusieurs groupes de serveurs.

Un groupe de serveurs est un ensemble d'instances de serveurs avec JBoss EAP 6 installé dessus, et qui sont gérées et configurées comme une entité unique. Le contrôleur de domaine gère la configuration et les applications déployées sur les groupes de serveurs. Ainsi, chaque serveur dans un groupe de serveurs partage les mêmes configurations et déploiements.

Il est possible qu'un contrôleur de domaine, un contrôleur hôte unique et plusieurs serveurs s'exécutent dans la même instance de JBoss EAP 6, sur le même système physique.

Les contrôleurs hôtes sont liés à des hôtes physiques (ou virtuels) spécifiques. Vous pouvez exécuter plusieurs contrôleurs hôtes sur le même matériel si vous utilisez différentes configurations, afin d'éviter que les ports et autres ressources n'entrent en conflit.

1.7. DOMAIN CONTROLLER DISCOVERY ET FAILOVER

Lorsque vous configurez un domaine géré, chaque contrôleur hôte doit être configuré avec les informations nécessaires pour communiquer avec le contrôleur de domaine. Dans JBoss EAP 6, chaque contrôleur hôte peut être configuré avec de multiples options pour trouver le contrôleur de domaine. Les contrôleurs hôtes peuvent parcourir la liste des options jusqu'à ce qu'une d'entre elle réussisse.

Cela permet aux contrôleurs hôtes d'être pré configurés avec des informations de contact d'un contrôleur de domaine secondaire. Un contrôleur hôte de sauvegarde peut être promu pour maîtriser s'il y a un problème avec le contrôleur de domaine principal, permettant aux contrôleurs hôtes de basculer automatiquement vers le nouveau master une fois qu'il a été promu.

Ce qui suit est un exemple sur la façon de configurer un contrôleur hôte avec des options multiples pour trouver le contrôleur de domaine.

Exemple 1.1. Contrôleur d'hôte configuré avec de nombreuses options de contrôleur de domaine

```
<domain-controller>
  <remote security-realm="ManagementRealm">
    <discovery-options>
      <static-discovery name="primary" host="172.16.81.100"
port="9999"/>
      <static-discovery name="backup" host="172.16.81.101"
port="9999"/>
    </discovery-options>
  </remote>
</domain-controller>
```

Une option discovery statique inclut les attributs obligatoires suivants :

name

Le nom de cette option discovery de contrôleur de domaine

host

Le nom d'hôte du contrôleur de domaine distant.

Important

Le port du contrôleur de domaine distant.

Dans l'exemple suivant, la première option discovery est celle avec laquelle on attend un résultat positif. La seconde peut être utilisée pour les situations d'échec.

Si un problème survient avec le contrôleur principal de domaine, un contrôleur hôte qui a été démarré avec l'option `--backup` pourra être promu pour agir comme contrôleur de domaine.



NOTE

À partir d'un contrôleur hôte avec l'option `--backup` qui entraînera ce contrôleur à conserver une copie locale de la configuration du domaine. Cette configuration servira si le contrôleur hôte est reconfiguré pour pouvoir agir comme contrôleur de domaine.

Procédure 1.1. Promouvoir un contrôleur hôte comme contrôleur de domaine

1. Assurez-vous que le contrôleur de domaine d'origine a, ou est, arrêté.
2. Utiliser l'interface CLI pour vous connecter au contrôleur hôte qui deviendra le nouveau contrôleur de domaine.
3. Exécutez la commande suivante pour configurer le contrôleur hôte pour qu'il agisse comme nouveau contrôleur de domaine.

```
/host=HOST_NAME:write-local-domain-controller
```

4. Exécutez la commande suivante pour rechercher le contrôleur hôte.

```
reload --host=HOST_NAME
```

Le contrôleur hôte choisi à l'étape 2 agira maintenant en tant que contrôleur de domaine.

[Rapporter un bogue](#)

1.8. CONTRÔLEUR HÔTE

Un contrôleur hôte est lancé quand le script `domain.sh` ou `domain.bat` exécute. sur un hôte.

La principale responsabilité d'un contrôleur hôte est la gestion de serveurs. Il délègue les tâches de gestion de domaines et est chargé de démarrer ou stopper les processus de serveurs d'application individuels qui exécutent sur son hôte.

Il entre en interaction avec le contrôleur de domaines pour gérer la communication entre les serveurs et le contrôleur de domaines. Plusieurs contrôleurs hôtes d'un domaine peuvent interagir avec un contrôleur de domaine unique. Par conséquent, tous les contrôleurs hôtes et les instances de serveurs exécutant en mode de domaine unique ont un contrôleur de domaine unique et doivent appartenir au même domaine.

Chaque contrôleur hôte lit par défaut sa configuration à partir du fichier `domain/configuration/host.xml` situé dans le fichier d'installation de JBoss EAP 6 décompressé sur le système de fichiers de son hôte. Le fichier `host.xml` contient les informations de configuration suivantes spécifiques à l'hôte particulier :

- Les noms des instances de JBoss EAP 6 censées être exécutées à partir de l'installation.
- Une des configurations suivantes :
 - La façon dont le contrôleur contacte le contrôleur de domaines pour s'enregistrer lui-même et pour accéder à la configuration de domaine.
 - La façon de rechercher et contacter un contrôleur de domaines éloigné.
 - Comment le contrôleur hôtes doit se persuader lui-même d'agir en tant que contrôleur de domaines
- Les configuration spécifiques à l'installation physique locale. Ainsi, les définitions d'interfaces nommées déclarées dans `domain.xml` peuvent être mappées vers une adresse IP particulière appartenant à une machine dans `host.xml`. Les noms de chemins d'accès abstraits de

domain.xml peuvent être mappés vers les chemins d'accès du système de fichiers dans host.xml.

[Rapporter un bogue](#)

1.9. LES GROUPES DE SERVEURS

Un groupe de serveurs est un regroupement d'instances des serveurs qui sont gérés et configurés en un. Dans un domaine géré, chaque instance de serveur d'application appartient à un groupe de serveurs, même s'il en est le seul membre. Les instances de serveur d'un groupe partagent la même configuration de profil et le même contenu déployé.

Un contrôleur de domaines et un contrôleur hôte font appliquer la configuration standard sur toutes les instances de serveur de chaque groupe de serveurs sur son domaine.

Un domaine peut se composer de plusieurs groupes de serveurs. Différents groupes de serveurs peuvent être configurés avec des déploiements et des profils différents. Un domaine peut être configuré avec des niveaux de serveurs différents offrant des services différents.

Différents groupes de serveurs peuvent également avoir les mêmes profils et déploiements. Cela permet, par exemple, le cumul des mises à niveau de l'application quand l'application est mise à jour sur un groupe de serveurs, puis mise à jour sur un deuxième groupe de serveurs, évitant ainsi une interruption complète du service.

Voici un exemple de définition de groupe de serveurs :

Exemple 1.2. Définition de groupe de serveurs

```
<server-group name="main-server-group" profile="default">
  <socket-binding-group ref="standard-sockets"/>
  <deployments>
    <deployment name="foo.war_v1" runtime-name="foo.war"/>
    <deployment name="bar.ear" runtime-name="bar.ear"/>
  </deployments>
</server-group>
```

Un groupe de serveurs inclut les attributs obligatoires suivants :

- **nom** : le nom du groupe de serveurs
- **profil** : le nom du profil du groupe de serveurs
- **socket-binding-group** : le nom du groupe de liaisons de sockets par défaut à utiliser pour les serveurs dans le groupe. Ce nom peut être remplacé sur la base d'un serveur dans host.xml. Cependant, c'est un élément obligatoire pour chaque groupe de serveurs et le domaine ne peut pas démarrer s'il n'est pas présent.

Un groupe de serveurs inclut les attributs optionnels suivants :

- **deployments** : le contenu de déploiement à déployer sur les serveurs du groupe.
- **system-properties** : les propriétés système à définir sur les serveurs du groupe

- `jvm` : les paramètres de configuration JMV par défaut de tous les serveurs du groupe. Le contrôleur hôte fait fusionner ces paramètres dans n'importe quelle configuration fournie par `host.xml` pour établir les paramètres utilisés dans la JVM du serveur.
- `socket-binding-port-offset`: le décalage par défaut à ajouter aux valeurs de port données par le groupe de liaisons de sockets.
- `management-subsystem-endpoint`: défini à `true` pour que les serveurs qui appartiennent au groupe de serveurs puissent se connecter à nouveau au contrôleur de l'hôte en utilisant le point de terminaison du sous-système distant (le sous-système distant doit être présent pour que cela fonctionne).

[Rapporter un bogue](#)

1.10. PROFILS JBOSS EAP 6

Le concept des profils qui ont été utilisés dans les versions précédentes de JBoss EAP n'est plus utilisé. JBoss EAP 6 utilise maintenant un petit nombre de fichiers de configuration simples pour contenir toutes les informations de configuration.

Les modules et les pilotes sont chargés en fonction des besoins, donc le concept du profil par défaut utilisé dans les anciennes versions de JBoss EAP 6 où les profils étaient utilisés pour rendre le démarrage du serveur plus performant n'est pas très utile.

Au moment du déploiement, les dépendances du module sont définies, ordonnancées, et résolues par le serveur ou le contrôleur du domaine, et chargées dans le bon ordre. Les modules sont retirés du chargement quand ils ne sont plus utiles à aucun déploiement.

Il est possible de désactiver les modules ou de décharger les pilotes ou autres services manuellement en retirant les sous-systèmes de la configuration. Cependant, dans la plupart des cas, cela n'est pas utile. Si aucune de vos applications utilisent un module, il ne sera pas chargé.

[Rapporter un bogue](#)

CHAPITRE 2. GESTION DE SERVEURS D'APPLICATIONS

2.1. CONVENTIONS POUR LA DOCUMENTATION JBOSS EAP

Toutes les instances d' *EAP_HOME* de ce guide se rapportent au répertoire d'installation root de JBoss EAP, suivant la méthode que vous utilisez.

Méthode d'installation en téléchargement compressé

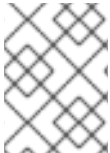
EAP_HOME est le répertoire à partir duquel le fichier JBoss EAP Zip est extrait.

Méthode d'installation par interface de commandes CLI ou par le GUI

EAP_HOME est le répertoire dans lequel vous décidez d'installer JBoss EAP.

Méthode d'installation RPM

EAP_HOME se rapporte au répertoire `/usr/share/jbossas`.



NOTE

L'annotation *EWS_HOME* est utilisée en référence aux emplacements d'installation de JBoss EWS, en suivant les mêmes conventions définies ci-dessus pour JBoss EAP.

[Rapporter un bogue](#)

2.2. DÉMARRER ET STOPPER JBOSS EAP 6

2.2.1. Démarrer JBoss EAP 6

Démarrer JBoss EAP 6 d'une des manières suivantes :

- [Section 2.2.2, « Démarrez JBoss EAP 6 comme un serveur autonome »](#)
- [Section 2.2.3, « Démarrez JBoss EAP 6 comme domaine géré »](#)

[Rapporter un bogue](#)

2.2.2. Démarrez JBoss EAP 6 comme un serveur autonome

Résumé

Cette rubrique couvre toutes les étapes à couvrir pour démarrer JBoss EAP 6 en tant que serveur autonome.

Procédure 2.1. Démarrer le service de plate-forme comme serveur autonome.

1. Dans Red Hat Enterprise Linux.
Exécuter la commande suivante : `EAP_HOME/bin/standalone.sh`
2. Dans Microsoft Windows Server
Exécuter la commande suivante : `EAP_HOME\bin\standalone.bat`

3. Option : indiquer les paramètres supplémentaires.

Pour imprimer une liste de paramètres supplémentaires à passer aux scripts de démarrage, utiliser le paramètre `-h`.

Résultat

L'instance de serveur autonome JBoss EAP 6 démarre.

[Rapporter un bogue](#)

2.2.3. Démarrez JBoss EAP 6 comme domaine géré

Ordre des opérations

Le contrôleur de domaines doit être démarré avant qu'un serveur esclave ne démarre dans des groupes de serveurs du domaine. Utiliser cette procédure sur le contrôleur de domaine pour commencer, puis, sur chaque contrôleur hôte associé et sur chaque hôte associé.

Procédure 2.2. Démarrer le service de plate-forme comme serveur géré

1. Dans Red Hat Enterprise Linux.

Exécutez la commande : `EAP_HOME/bin/domain.sh`

2. Dans Microsoft Windows Server

Exécutez la commande : `EAP_HOME\bin\domain.bat`

3. En option : passez des paramètres supplémentaires au script de démarrage.

Pour obtenir une liste de paramètres que vous pourrez passer au script de démarrage, utilisez le paramètre `-h`.

Résultat

L'instance de domaine géré de JBoss EAP 6 démarre.

[Rapporter un bogue](#)

2.2.4. Configuration d'un nom d'hôte dans un domaine géré

Résumé

Chaque hôte exécutant dans un domaine géré doit avoir un nom d'hôte unique. Pour faciliter l'administration et permettre l'utilisation de mêmes fichiers de configuration hôte sur plusieurs hôtes, le serveur utilise la priorité suivante pour déterminer le nom d'hôte.

1. Si défini, l'attribut de nom de l'élément hôte qui se trouve dans le fichier de configuration `host.xml`.
2. La valeur de la propriété système `jboss.host.name`.
3. La valeur qui suit le caractère (".") dans la propriété système `jboss.qualified.host.name`, ou toute la valeur s'il n'y a pas de point final (".").
4. La valeur qui suit le caractère (".") dans la variable d'environnement `HOSTNAME` pour les systèmes d'exploitation basés POSIX, la variable d'environnement `COMPUTERNAME` dans Microsoft Windows, ou toute la valeur s'il n'y a pas de point final (".").

Pour obtenir des informations sur la façon de définir les variables d'environnement, voir la documentation de votre système d'exploitation. Pour plus d'informations sur la façon de définir les propriétés système, voir [Section 3.6.11, « Configurer les propriétés système par l'interface CLI »](#).

Cette section décrit comment fixer le nom de l'hôte dans le fichier de configuration, à l'aide d'une propriété système ou d'un nom codé en dur.

Procédure 2.3. Configuration d'un nom d'hôte avec une propriété système

1. Ouvrir le fichier de configuration de l'hôte `host.xml` pour le modifier.
2. Chercher l'élément `host` dans le fichier, comme par exemple :

```
<host name="master" xmlns="urn:jboss:domain:1.6">
```

3. S'il est présent, retirer la déclaration d'attribut `name="HOST_NAME"`. L'élément `host` devra ressembler à l'exemple suivant :

```
<host xmlns="urn:jboss:domain:1.6">
```

4. Démarrer le serveur en saisissant `-Djboss.host.name` comme argument de ligne de commande, comme par exemple :

```
-Djboss.host.name=HOST_NAME
```

Procédure 2.4. Configuration d'un nom d'hôte avec un nom spécifique

1. Démarrer l'hôte esclave JBoss EAP à l'aide de la syntaxe suivante :

```
bin/domain.sh --host-config=HOST_FILE_NAME
```

Par exemple :

```
bin/domain.sh --host-config=host-slave01.xml
```

2. Lancer l'interface CLI.
3. Utiliser la syntaxe suivante pour remplacer le nom d'hôte :

```
/host=EXISTING_HOST_NAME:write-attribute(name="name",value=UNIQUE_HOST_NAME)
```

Par exemple :

```
/host=master:write-attribute(name="name",value="host-slave01")
```

Vous devriez voir apparaître le résultat suivant.

```
"outcome" => "success"
```

Cela modifie l'attribut `name` de l'hôte dans le fichier `host-slave01.xml` comme suit :

```
<host name="host-slave01" xmlns="urn:jboss:domain:1.6">
```

4. Vous devez charger à nouveau la configuration du serveur avec l'ancien nom d'hôte pour terminer le processus.

```
reload --host=EXISTING_HOST_NAME
```

Par exemple :

```
reload --host=master
```

[Rapporter un bogue](#)

2.2.5. Créer un domaine géré sur deux machines



NOTE

Vous devrez sans doute configurer votre pare-feu pour qu'il puisse exécuter cet exemple.

Vous pouvez créer un domaine géré sur deux machines, avec une machine en tant que contrôleur de domaine, et l'autre en tant qu'hôte. Pour plus d'informations, voir [Section 1.6, « Contrôleur de domaine »](#).

- IP1 = adresse IP du contrôleur de domaine (Machine 1)
- IP2 = adresse IP de l'hôte (Machine 2)

Procédure 2.5. Créer un domaine géré sur deux machines

1. Sur la machine 1

- a. Utiliser le script `add-user.sh` pour ajouter l'utilisateur de management. Par exemple, `slave01`, pour que l'hôte puisse authentifier le contrôleur de domaines. Notez la valeur `SECRET_VALUE` de la sortie `add-user`.
- b. Démarrer le domaine par le fichier de configuration `host-master.xml`, qui est préconfiguré pour un contrôleur de domaines exclusif.
- c. Utiliser `-bmanagement=$IP1` pour rendre le contrôleur de domaine visible auprès des autres machines.

```
[$JBOSS_HOME/bin]$ ./domain.sh --host-config=host-master.xml -
bmanagement=$IP1
```

2. Sur la machine 2

- a. Mettre à jour le fichier `$JBOSS_HOME/domain/configuration/host-slave.xml` avec les identifiants.

```
<?xml version='1.0' encoding='UTF-8'?>
  <host xmlns="urn:jboss:domain:1.6" name="slave01">
    <!-- add user name here -->
```

```

    <management>
      <security-realms>
        <security-realm name="ManagementRealm">
          <server-identities>
            <secret value="$SECRET_VALUE" />
            <!-- use secret value from add-user.sh
output-->
          </server-identities>
          ...

```

b. Démarrer l'hôte.

```

[$JBOSS_HOME/bin]$ ./domain.sh --host-config=host-slave.xml -
Djboss.domain.master.address=$IP1 -b=$IP2

```

3. Nous pouvons maintenant gérer le domaine.

via le CLI :

```

[$JBOSS_HOME/bin]$ ./jboss-cli.sh -c --controller=$IP1

```

via la console web :

```

http://$IP1:9990

```

Accéder à la page d'index du serveur :

```

http://$IP2:8080/
http://$IP2:8230/

```

[Rapporter un bogue](#)

2.2.6. Démarrer JBoss EAP 6 avec une configuration différente

Si vous n'indiquez pas de fichier de configuration, le serveur démarrera avec le fichier par défaut. Cependant, quand vous démarrez le serveur, vous pouvez spécifier Configuration manuelle. Le processus varie légèrement, suivant que vous utilisez un Domaine géré ou un Serveur autonome, et suivant le système d'exploitation que vous utilisez.

Conditions préalables

- Avant d'utiliser un fichier de configuration alternatif, préparez-le à l'aide de la configuration par défaut comme modèle. Pour un domaine géré, le fichier de configuration doit être placé dans *EAP_HOME/domain/configuration/*. Pour les serveurs autonomes, le fichier de configuration devra être mis dans le répertoire *EAP_HOME/standalone/configuration/*.



NOTE

Plusieurs exemples de configurations sont inclus dans les répertoires de configuration **EAP_HOME/docs/examples/configs/**. Utiliser ces exemples pour activer des fonctionnalités supplémentaires, comme clustering ou l'API XTS de Transactions.

Certains des exemples de configurations doivent être modifiés avant d'être utilisés. Les fichiers de configuration suivants produisent des erreurs s'ils sont utilisés sans être modifiés : **standalone-picketlink.xml**, **standalone-genericjms.xml** et **standalone-hornetq-colocated.xml**.

Procédure 2.6. Démarrage de l'instance par une configuration différente

1. Serveur autonome

Pour un domaine autonome, fournir le nom du fichier de configuration comme option du paramètre **--server-config**. Le fichier de configuration doit se trouver dans le répertoire **EAP_HOME/standalone/configuration/**, et vous devez indiquer le chemin d'accès du fichier de ce répertoire.

Exemple 2.1. Utiliser un fichier de configuration alternatif pour un serveur autonome Red Hat Enterprise Linux.

```
[user@host bin]$ ./standalone.sh --server-config=standalone-alternate.xml
```

Cet exemple utilise le fichier de configuration **EAP_HOME/standalone/configuration/standalone-alternate.xml**.

Exemple 2.2. Utiliser un fichier de configuration alternatif pour un serveur autonome Microsoft Windows.

```
C:\EAP_HOME\bin> standalone.bat --server-config=standalone-alternate.xml
```

Cet exemple utilise le fichier de configuration **EAP_HOME/standalone/configuration/standalone-alternate.xml**.

2. Domaine géré

Pour un domaine géré, fournir le nom du fichier de configuration comme option du paramètre **--domain-config**. Le fichier de configuration se trouve dans le répertoire **EAP_HOME/domain/configuration/**, et vous devez indiquer le chemin d'accès de ce répertoire.

Exemple 2.3. Utilisation d'un fichier de configuration alternatif pour un domaine géré dans Red Hat Enterprise Linux

```
[user@host bin]$ ./domain.sh --domain-config=domain-alternate.xml
```

Cet exemple utilise le fichier de configuration **EAP_HOME/domain/configuration/domain-alternate.xml**.

Exemple 2.4. Utilisation d'un fichier de configuration alternatif pour un domaine géré dans un serveur Microsoft Windows

```
C:\EAP_HOME\bin> domain.bat --domain-config=domain-alternate.xml
```

Cet exemple utilise le fichier de configuration `EAP_HOME\domain\configuration\domain-alternate.xml`.

Résultat

La plateforme JBoss Enterprise Application Platform est maintenant en cours d'exécution, avec votre fichier de configuration alternatif.

[Rapporter un bogue](#)

2.2.7. Stopper le serveur JBoss EAP 6

La façon dont vous arrêtez la plate-forme JBoss EAP 6 dépend de la façon dont elle a été lancée. Cette tâche couvre l'arrêt d'une instance qui a démarré de manière interactive, comment faire cesser une instance qui a été démarrée par un service et comment faire cesser une instance qui a été mise en arrière-plan par un script.



NOTE

Pour obtenir des informations sur la façon de stopper un serveur ou un groupe de serveurs dans un domaine géré, voir [Section 2.3.3, « Stopper un serveur qui utilise une console de gestion »](#). Pour obtenir des informations sur la façon de stopper un serveur par le CLI, voir [Section 2.3.1, « Démarrer et arrêter les serveurs par l'interface CLI »](#).

- **Procédure 2.7. Stopper une instance de JBoss EAP 6**
 - **Stopper une instance qui a été démarrée de façon interactive à partir d'une invite de commande.**
Appuyez sur `Ctrl-C` dans le terminal où JBoss EAP 6 exécute.

- **Procédure 2.8. Stopper une instance qui a démarré en tant que service de système d'exploitation.**

Suivant votre système d'exploitation, utiliser une des procédures suivantes :

- ■ **Red Hat Enterprise Linux**
Dans Red Hat Enterprise Linux, si vous avez écrit un script de service, utiliser sa fonction `stop`. Cela devra être inscrit dans le script. Ensuite, vous pourrez utiliser `service scriptname stop`, avec *scriptname* comme nom de script.
- **Microsoft Windows Server**
Dans Microsoft Windows, utiliser la commande `net service`, ou bien faites cesser le service à partir de l'applet `Services` qui se trouve dans le panneau de contrôle.
- **Procédure 2.9. Stopper une instance qui exécute en arrière-plan (Red Hat Enterprise Linux)**

1. Chercher l'ID de processus (PID) du processus :

- **Si une seule instance est en cours d'exécution (mode autonome)**

N'importe laquelle des commandes suivantes renverront le PID d'une simple instance de JBoss EAP 6 :

- `pidof java`

- `jps`

(La commande `jps` retournera un ID des deux processus ; un pour `jboss-modules.jar` et un pour `jps` lui-même. Utiliser l'ID de `jboss-modules.jar` pour stopper l'instance EAP)

- **Si plusieurs instances EAP sont en cours d'exécution (mode de domaine)**

Identifier le process qui convient pour y mettre un terme si plus d'une instance d'EAP en cours d'exécution nécessitent l'utilisation de commandes plus élaborées.

- La commande `jps` peut être utilisée en mode détaillé (verbose) pour qu'elle puisse fournir davantage d'informations sur les processus java qu'elle trouve.

Vous trouverez ci-dessous sous une sortie abrégée d'une commande détaillée `jps` qui identifie les différents processus d'EAP en cours par PID et rôle :

```
$ jps -v
12155 jboss-modules.jar -D[Server:server-one] -
XX:PermSize=256m -XX:MaxPermSize=256m -Xms1303m
...

12196 jboss-modules.jar -D[Server:server-two] -
XX:PermSize=256m -XX:MaxPermSize=256m -Xms1303m
...

12096 jboss-modules.jar -D[Host Controller] -Xms64m -
Xmx512m -XX:MaxPermSize=256m
...

11872 Main -Xms128m -Xmx750m -XX:MaxPermSize=350m -
XX:ReservedCodeCacheSize=96m -XX:+UseCodeCacheFlushing
...

11248 jboss-modules.jar -D[Standalone] -
XX:+UseCompressedOops -verbose:gc
...

12892 Jps
...

12080 jboss-modules.jar -D[Process Controller] -Xms64m -
Xmx512m -XX:MaxPermSize=256m
...
```

- La commande `ps aux` peut également être utilisée pour renvoyer des informations sur les instances multiples EAP.

Vous trouverez ci-dessous sous une sortie abrégée d'une commande détaillée `ps`

aux qui identifie les différents processus d'EAP en cours par PID et rôle :

```
$ ps aux | grep java
username 12080  0.1  0.9 3606588 36772 pts/0  Sl+  10:09
0:01 /path/to/java -D[Process Controller] -server -Xms128m
-Xmx128m -XX:MaxPermSize=256m
...

username 12096  1.0  4.1 3741304 158452 pts/0  Sl+  10:09
0:13 /path/to/java -D[Host Controller] -Xms128m -Xmx128m -
XX:MaxPermSize=256m
...

username 12155  1.7  8.9 4741800 344224 pts/0  Sl+  10:09
0:22 /path/to/java -D[Server:server-one] -XX:PermSize=256m
-XX:MaxPermSize=256m -Xms1000m -Xmx1000m -server -
...

username 12196  1.8  9.4 4739612 364436 pts/0  Sl+  10:09
0:22 /path/to/java -D[Server:server-two] -XX:PermSize=256m
-XX:MaxPermSize=256m -Xms1000m -Xmx1000m -server
...
```

Dans les exemples ci-dessus, les processus **Process Controller** sont des processus à stopper pour stopper tout le domaine.

L'utilitaire **grep** peut être utilisé avec une de ces commandes pour identifier le **Process Controller** :

```
jps -v | grep "Process Controller"
```

```
ps aux | grep "Process Controller"
```

2. Envoyer le signal **TERM** au processus en exécutant **kill PID**, quand **PID** est l'ID de processus identifié par une des commandes ci-dessus.

Résultat

Chacune de ces solutions ferme la plate-forme JBoss EAP 6 nettement, ce qui fait qu'aucune donnée n'est perdue.

[Rapporter un bogue](#)

2.2.8. Référence aux variables et arguments à passer à l'exécution du serveur

Le script de démarrage du serveur d'applications accepte l'ajout d'arguments et de variables en cours d'exécution. L'utilisation de ces paramètres permet au serveur d'être démarré sous d'autres configurations que celles qui sont définies dans les fichiers de configuration **standalone.xml**, **domain.xml** et **host.xml**. Cela peut comprendre le démarrage du serveur par un ensemble de liaisons de sockets différentes ou une configuration secondaire. Vous pourrez accéder à une liste des paramètres disponibles en passant la variable d'assistance au démarrage.

L'exemple suivant ressemble au démarrage de serveur expliqué dans [Section 2.2.2, « Démarrez JBoss EAP 6 comme un serveur autonome »](#) et [Section 2.2.3, « Démarrez JBoss EAP 6 comme domaine géré »](#), avec en plus les variables **-h** ou **--help**>. Les résultats de cette variable d'assistance sont

expliqués dans le tableau ci-dessous.

Exemple 2.5. Variable d'assistance « help »

Mode autonome :

```
[localhost bin]$ standalone.sh -h
```

Mode de domaine :

```
[localhost bin]$ domain.sh -h
```

Tableau 2.1. Tableau des arguments et variables du temps d'exécution

Argument ou Option	Mode	Description
--admin-only	Autonome	Définir le type d'exécution du serveur à ADMIN_ONLY . Cela le fera ouvrir les interfaces administratives et il pourra ainsi accepter les ordres de gestion, mais il ne pourra pas démarrer d'autres services de runtime ou accepter les demandes de l'utilisateur final.
--admin-only	Domaine	Définir le type d'exécution du contrôleur hôte à ADMIN_ONLY , ce qui le fera ouvrir les interfaces administratives et il pourra ainsi accepter les ordres de gestion, mais il ne pourra pas démarrer d'autres serveurs ou, si ce contrôleur hôte est le master du domaine, il pourra accepter les demandes des contrôleurs hôte esclaves.
-b <value>, -b=<value>	Autonome, Serveur	Définir la propriété système jboss.bind.address à la valeur donnée.
-b<interface>=<value>	Autonome, Serveur	Définir la propriété système jboss.bind.address.<interface> à la valeur donnée.
--backup	Domaine	Conserver une copie de la configuration de domaine persistante même si cet hôte n'est pas le contrôleur de domaines.
-c <config>, -c=<config>	Autonome	Nommer le fichier de configuration du serveur à utiliser. La valeur par défaut est standalone.xml .
-c <config>, -c=<config>	Domaine	Nom du fichier de configuration de serveur à utiliser. La valeur par défaut est domain.xml .

Argument ou Option	Mode	Description
<code>--cached-dc</code>	Domaine	Si l'hôte n'est pas le contrôleur de domaine et ne peut pas contacter le contrôleur de domaine au démarrage, puisque le processus de démarrage (booting) utilise une copie de la configuration de domaine mise en cache localement.
<code>--debug [<port>]</code>	Autonome	Active le mode de débogage par un argument en option qui indique le port. Ne fonctionne que si le script de lancement le supporte.
<code>-D<name>[=<value>]</code>	Autonome, Serveur	Définit une propriété système.
<code>--domain-config=<config></code>	Domaine	Nom du fichier de configuration de serveur à utiliser. La valeur par défaut est domain.xml .
<code>-h, --help</code>	Autonome, Serveur	Affiche le message d'assistance et sortir.
<code>--host-config=<config></code>	Domaine	Nom du fichier de configuration hôte à utiliser. La valeur par défaut est host.xml .
<code>--interprocess-hc-address=<address></code>	Domaine	Adresse à laquelle le contrôleur hôte doit écouter la communication en provenance du contrôleur de processus.
<code>--interprocess-hc-port=<port></code>	Domaine	Port sur lequel le contrôleur hôte doit écouter la communication en provenance du contrôleur de processus.
<code>--master-address=<address></code>	Domaine	Définit la propriété système jboss.domain.master.address à la valeur donnée. Dans une configuration de contrôleur hôte esclave par défaut, c'est utilisé pour configurer l'adresse du contrôleur hôte maître.
<code>--master-port=<port></code>	Domaine	Définit la propriété système jboss.domain.master.port à la valeur donnée. Dans une configuration de contrôleur hôte esclave par défaut, c'est utilisé pour configurer le port utilisé pour la communication de gestion native du contrôleur hôte maître.
<code>--read-only-server-config=<config></code>	Autonome	Nom du fichier de configuration du serveur à utiliser. Cela diffère de <code>--server-config</code> et <code>-c</code> en ce que le fichier d'origine n'est jamais écrasé.

Argument ou Option	Mode	Description
<code>--read-only-domain-config=<config></code>	Domaine	Nom du fichier de configuration du domaine à utiliser. Cela diffère de <code>--domain-config</code> et de <code>-c</code> en ce que le fichier de départ n'est jamais écrasé.
<code>--read-only-host-config=<config></code>	Domaine	Nom du fichier de configuration de l'hôte à utiliser. Cela diffère de <code>--host-config</code> en ce que le fichier de départ n'est jamais écrasé.
<code>-P <url>, -P=<url>, --properties=<url></code>	Autonome, Serveur	Télécharge les propriétés système de l'URL donné.
<code>--pc-address=<address></code>	Domaine	Adresse à laquelle le contrôleur de processus doit écouter les communications en provenance des processus qu'il contrôle.
<code>--pc-port=<port></code>	Domaine	Port sur lequel le contrôleur de processus doit écouter les communications en provenance des processus qu'il contrôle.
<code>-S<name>[=<value>]</code>	Autonome	Définit une propriété de sécurité.
<code>--server-config=<config></code>	Autonome	Nommer le fichier de configuration du serveur à utiliser. La valeur par défaut est <code>standalone.xml</code> .
<code>-u <value>, -u=<value></code>	Autonome, Serveur	Définit la propriété système <code>jboss.default.multicast.address</code> à la valeur donnée.
<code>-v, -V, --version</code>	Autonome, Serveur	Affiche la version du serveur d'applications et sortie.

[Rapporter un bogue](#)

2.3. DÉMARRER ET ARRÊTER LES SERVEURS

2.3.1. Démarrer et arrêter les serveurs par l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Vous pouvez démarrer et arrêter les serveurs avec l'interface CLI (en mode autonome) ou par la console de gestion. En mode de domaine, vous ne pouvez que démarrer les instances de serveur. Les deux outils de gestion vous permettent de contrôler une seule instance de serveur autonome, ou de gérer sélectivement un ou plusieurs serveurs dans un déploiement de domaine géré. Si vous utilisez la

console de gestion en mode de domaine, veuillez consulter [Section 2.3.2, « Démarrer un serveur par la console de gestion »](#) pour obtenir des instructions. Si vous utilisez l'interface CLI, le processus varie entre des instances de serveur autonome et de domaine géré.

Démarrer et arrêter un serveur autonome par l'interface CLI

Vous pouvez démarrer une instance de serveur autonome par les scripts de ligne de commande, et le fermer par l'interface CLI par la commande `shutdown`. Si vous avez besoin de l'instance par la suite, exécuter le processus de démarrage à nouveau selon les instructions dans [Section 2.2.2, « Démarrez JBoss EAP 6 comme un serveur autonome »](#).

Exemple 2.6. Démarrer et arrêter une instance de serveur autonome par l'interface CLI

```
[standalone@localhost:9999 /] shutdown
```

Démarrer et arrêter un domaine géré par l'interface CLI

Si vous exécutez un domaine géré, la console de gestion va vous permettre de démarrer ou de stopper sélectivement des serveurs spécifiques pour ce domaine. Cela inclut les groupes de serveurs dans tout le domaine, ainsi que les instances de serveur spécifiques sur un hôte.

Exemple 2.7. Stopper un hôte de serveur dans un domaine géré par l'interface CLI

Semblable à une instance de serveur autonome, la commande `shutdown` est utilisée pour fermer un hôte de domaine géré déclaré. Cet exemple stoppe un hôte de serveur nommé `master` en déclarant le nom d'instance avant d'appeler l'opération de fermeture. Utiliser la touche `tab` pour aider à la complétion de chaînes et pour exposer des variables visibles comme les valeurs hôtes disponibles.

```
[domain@localhost:9999 /] /host=master:shutdown
```

Exemple 2.8. Stopper un hôte de serveur dans un domaine géré par l'interface CLI

Cet exemple montre comment démarrer un groupe de serveurs par défaut nommé `main-server-group` en déclarant le groupe avant les opérations `start` et `stop`. Utilisez la touche `tab` pour aider à la complétion de chaînes et pour exposer des variables visibles comme les valeurs de nom de groupes de serveurs disponibles.

```
[domain@localhost:9999 /] /server-group=main-server-group:start-servers
```

```
[domain@localhost:9999 /] /server-group=main-server-group:stop-servers
```

Exemple 2.9. Démarrer et stopper une instance de serveur dans un domaine géré par l'interface CLI

Cet exemple montre comment démarrer et stopper une instance de serveur nommée `server-one` sur l'hôte `master` en déclarant la configuration de l'hôte et du serveur avant d'invoquer les opérations `start` et `stop`. Utilisez la touche `tab` pour aider à la complétion de chaînes et pour exposer des variables visibles comme les valeurs de configuration de l'hôte et du serveur.

```
[domain@localhost:9999 /] /host=master/server-config=server-one:start
```

```
[domain@localhost:9999 /] /host=master/server-config=server-one:stop
```

[Rapporter un bogue](#)

2.3.2. Démarrer un serveur par la console de gestion

Conditions préalables

- [Section 2.2.3, « Démarrez JBoss EAP 6 comme domaine géré »](#)
- [Section 3.4.2, « Se connecter à la console de gestion »](#)

Procédure 2.10. Démarrer le serveur d'un domaine géré

1. Sélectionner l'onglet **Runtime** qui se trouve en haut de la console. Étendre le menu **Server** et sélectionner **Overview**.
2. À partir de la liste **Server Instances**, sélectionner le serveur que vous souhaitez démarrer. Les serveurs qui sont en cours d'exécution sont indiqués.

Placez le curseur sur une instance de cette liste pour afficher les options en bleu dans le texte sous les informations sur le serveur.

3. Pour démarrer une instance, cliquer sur **Start Server** quand vous verrez ce texte. Une boîte de dialogue de confirmation va s'ouvrir. Cliquer sur le bouton **Confirm** pour démarrer le serveur.

Résultat

Le serveur sélectionné démarre et exécute.

[Rapporter un bogue](#)

2.3.3. Stopper un serveur qui utilise une console de gestion

Conditions préalables

- [Section 2.2.3, « Démarrez JBoss EAP 6 comme domaine géré »](#)
- [Section 3.4.2, « Se connecter à la console de gestion »](#)

Procédure 2.11. Stopper un serveur qui utilise une console de gestion dans un domaine géré

1. Sélectionner l'onglet **Runtime** qui se trouve en haut de la console. Étendre le menu **Domain** et sélectionner **Overview**.
2. Une liste d'instances **Server Instances** s'affiche dans le tableau **Hosts, groups and server instances**. Les serveurs en cours sont cochés.
3. Placer le curseur sur le serveur choisi. Cliquer sur le bouton **Stop Server** qui apparaît. Une fenêtre de dialogue de confirmation s'affichera.

4. Cliquer sur le bouton **Confirm** pour arrêter le serveur.

Résultat

Le serveur sélectionné est stoppé.

[Rapporter un bogue](#)

2.4. CHEMINS D'ACCÈS AUX SYSTÈMES DE FICHIERS

2.4.1. Chemins d'accès aux systèmes de fichiers

JBoss EAP 6 utilise des noms logiques pour les chemins de systèmes de fichiers. Les fichiers de configuration `domain.xml`, `host.xml` et `standalone.xml` incluent tous une section où les chemins d'accès peuvent être déclarés. D'autres sections de la configuration peuvent ensuite référencer ces chemins par leur nom logique, évitant la déclaration du chemin d'accès absolu pour chaque instance. Cela bénéficie aux efforts de configuration et d'administration car cela permet à des configurations hôtes spécifiques de résoudre des noms logiques universels.

Par exemple, la configuration du sous-système de logging comprend une référence au chemin `jboss.server.log.dir` qui pointe vers le répertoire `log` du serveur.

Exemple 2.10. Exemple de chemin d'accès relatif du répertoire de logging

```
<file relative-to="jboss.server.log.dir" path="server.log"/>
```

JBoss EAP 6 fournit un nombre de chemins d'accès standards automatiquement sans que l'utilisateur n'ait besoin de les configurer dans un fichier de configuration.

Tableau 2.2. Chemins d'accès standard

Valeur	Description
<code>java.ext.dirs</code>	Chemins de répertoires d'extension de JDK.
<code>jboss.home.dir</code>	Le répertoire root de la distribution JBoss EAP 6.
<code>user.home</code>	Le répertoire d'accueil de l'utilisateur.
<code>user.dir</code>	Le répertoire de travail actuel de l'utilisateur
<code>java.home</code>	Le répertoire d'installation de Java
<code>jboss.server.base.dir</code>	Le répertoire root d'une instance de serveur individuel.
<code>jboss.server.data.dir</code>	Le répertoire que le serveur va utiliser pour le stockage de fichiers de données persistantes.

Valeur	Description
<code>jboss.server.config.dir</code>	Le répertoire qui contient la configuration du serveur.
<code>jboss.server.log.dir</code>	Le répertoire que le serveur va utiliser pour le stockage de fichier journal.
<code>jboss.server.tmp.dir</code>	Le répertoire que le serveur va utiliser pour le stockage de fichiers temporaires.
<code>jboss.server.deploy.dir</code>	Le répertoire que le serveur utilisera pour stocker le contenu déployé.
<code>jboss.controller.temp.dir</code>	Le répertoire que le contrôleur hôte va utiliser pour le stockage de fichiers temporaires.
<code>jboss.domain.base.dir</code>	Le répertoire de base de contenu de domaine.
<code>jboss.domain.config.dir</code>	Le répertoire qui contient la configuration de domaine.
<code>jboss.domain.data.dir</code>	Le répertoire que le domaine utilisera pour le stockage de fichiers de données persistantes.
<code>jboss.domain.log.dir</code>	Le répertoire que le domaine va utiliser pour le stockage de fichier de journalisation persistant.
<code>jboss.domain.tmp.dir</code>	Le répertoire que le domaine va utiliser pour le stockage de fichier temporaire.
<code>jboss.domain.deployment.dir</code>	Le répertoire que le serveur utilisera pour stocker le contenu déployé.
<code>jboss.domain.servers.dir</code>	Le répertoire que le domaine va utiliser pour stocker les sorties d'instances de domaines gérés.

Substituer un chemin par un autre

Si vous exécutez en serveur autonome, vous pourrez substituer les chemins `jboss.server.base.dir`, `jboss.server.log.dir`, ou `jboss.server.config.dir` d'une ou de deux manières.

1. Vous pouvez passer des arguments par ligne de commande quand vous démarrez le serveur. Ainsi :

```
bin/standalone.sh -Djboss.server.log.dir=/var/log
```

2. Vous pouvez modifier la variable `JAVA_OPTS` dans le fichier de configuration du serveur. Ouvrir le fichier `EAP_HOME/bin/standalone.conf` et ajouter la ligne suivante à la fin du fichier :

```
JAVA_OPTS="$JAVA_OPTS Djboss.server.log.dir=/var/log"
```

La substitution de fichiers n'est pas prise en charge pour les serveurs exécutant dans un domaine géré.

Ajouter un chemin personnalisé

Vous pouvez également créer votre propre chemin personnalisé. Par exemple, vous pouvez définir un chemin relatif à utiliser pour la journalisation. Vous pouvez alors changer le gestionnaire de journal pour utiliser `my.relative.path`,

Exemple 2.11. Un chemin de journalisation personnalisé

```
my.relative.path=/var/log
```

[Rapporter un bogue](#)

2.5. FICHIERS DE CONFIGURATION

2.5.1. Fichiers de configuration de JBoss EAP 6

La configuration de JBoss EAP 6 a considérablement changé depuis les dernières versions. Une des différences principale est l'utilisation d'une structure de fichier de configuration simplifiée, qui comprend un ou plusieurs des fichiers répertoriés ci-dessous :

Tableau 2.3. Emplacements des fichiers de configuration

Mode du serveur	Emplacement	But
domain.xml	<code>EAP_HOME/domain/configuration/domain.xml</code>	Il s'agit du fichier de configuration principal d'un domaine géré. Seul le master du domaine lit ce fichier. Il peut être supprimé pour les autres membres du domaine.
host.xml	<code>EAP_HOME/domain/configuration/host.xml</code>	Ce fichier contient les détails de configuration spécifiques à un hôte physique dans un domaine géré, tels que les interfaces réseau, les liaisons de sockets, le nom de l'hôte et d'autres détails spécifiques à l'hôte. Le fichier <code>host.xml</code> contient toutes les fonctionnalités de <code>hôte-master.xml</code> et <code>hôte-slave.xml</code> qui sont décrites ci-dessous. Ce fichier n'est pas présent pour les serveurs autonomes.

Mode du serveur	Emplacement	But
host-master.xml	<i>EAP_HOME/domain/configuration/host-master.xml</i>	Ce fichier inclut tous les détails de configuration nécessaires pour exécuter un serveur en tant que serveur maître de domaine géré. Ce fichier n'est pas présent dans les serveurs autonomes.
host-slave.xml	<i>EAP_HOME/domain/configuration/host-slave.xml</i>	Ce fichier inclut tous les détails de configuration nécessaires pour exécuter un serveur en tant que serveur esclave de domaine géré. Ce fichier n'est pas présent dans les serveurs autonomes.
standalone.xml	<i>EAP_HOME/standalone/configuration/standalone.xml</i>	C'est le fichier de configuration par défaut pour un serveur autonome. Il contient toutes les informations sur le serveur autonome, y compris les sous-systèmes, réseautage, déploiements, les liaisons de sockets et autres détails configurables. Cette configuration est utilisée automatiquement lorsque vous démarrez votre serveur autonome.

Mode du serveur	Emplacement	But
standalone-full.xml	<i>EAP_HOME/standalone/configuration/standalone-full.xml</i>	Il s'agit d'un exemple de configuration pour un serveur autonome. Il prend en charge chaque sous-système possible à l'exception de ceux destinés à la haute disponibilité. Pour l'utiliser, arrêtez votre serveur et redémarrez à l'aide de la commande suivante : <i>EAP_HOME/bin/standalone.sh -c standalone-full.xml</i>
standalone-ha.xml	<i>EAP_HOME/standalone/configuration/standalone-ha.xml</i>	Cet exemple de fichier de configuration active tous les sous-systèmes par défaut et ajoute les mod_cluster et les sous-systèmes JGroups pour un serveur autonome, afin qu'il puisse participer à un cluster de haute disponibilité ou d'équilibrage de la charge. Ce fichier n'est pas applicable à un domaine géré. Pour utiliser cette configuration, arrêtez votre serveur et redémarrez le à l'aide de la commande suivante : <i>EAP_HOME/bin/standalone.sh -c standalone-ha.xml</i>
standalone-full-ha.xml	<i>EAP_HOME/standalone/configuration/standalone-full-ha.xml</i>	Il s'agit d'un exemple de configuration pour un serveur autonome. Il prend en charge chaque sous-système possible incluant ceux destinés à la haute disponibilité. Pour l'utiliser, arrêtez votre serveur et redémarrez à l'aide de la commande suivante : <i>EAP_HOME/bin/standalone.sh -c standalone-full-ha.xml</i>

Ce sont les emplacements par défaut uniquement. Vous pouvez indiquer un fichier de configuration différent en cours d'exécution.

[Rapporter un bogue](#)

2.5.2. Remplacement de propriété basée descripteur

La configuration d'application - par exemple, les paramètres de connexion de la source de données - varient normalement entre le développement, les tests, et les déploiements de production. Cette variation est parfois accommodée par les build system scripts, car la spécification Java EE ne contient pas de méthode pour externaliser ces configurations. Dans JBoss EAP 6, vous pouvez utiliser *le remplacement de propriété basée descripteur* pour gérer la configuration en externe.

Le remplacement de propriété basée descripteur remplace les propriétés basées sur des descripteurs, ce qui vous permet de supprimer les hypothèses concernant l'environnement de l'application et la chaîne de construction. Les configurations spécifiques à l'environnement peuvent être spécifiées dans les descripteurs de déploiement à la place des annotations ou des build system scripts. Vous pouvez fournir la configuration dans les fichiers ou en tant que paramètres en ligne de commande.

Le remplacement de propriétés basées descripteur est activé globalement par `standalone.xml` ou `domain.xml` :

Exemple 2.12. Remplacement de propriété basée descripteur

```
<subsystem xmlns="urn:jboss:domain:ee:1.1">
  <spec-descriptor-property-replacement>
    true
  </spec-descriptor-property-replacement>
  <jboss-descriptor-property-replacement>
    true
  </jboss-descriptor-property-replacement>
</subsystem>
```

Le remplacement de descripteur spécifique à Java EE est *désactivé* par défaut. Si activés, les descripteurs peuvent être remplacés dans les fichiers de configuration suivants : `ejb-jar.xml` et `persistence.xml`.

Le remplacement de descripteur spécifique à Java est *activé* par défaut. Si activés, les descripteurs peuvent être remplacés dans les fichiers de configuration suivants :

- `jboss-ejb3.xml`
- `jboss-app.xml`
- `jboss-web.xml`
- `*-jms.xml`
- `*-ds.xml`

Par exemple, avec un bean ayant l'annotation suivante :

Exemple 2.13. Exemple d'annotation

```
@ActivationConfigProperty(propertyName = "connectionParameters",
  propertyValue = "host=192.168.1.1;port=5445")
```

Avec le remplacement de propriété basée descripteur, `connectionParameters` peut être spécifié en ligne de commande par :

```
./standalone.sh -DconnectionParameters='host=10.10.64.1;port=5445'
```

Pour accomplir cela par les propriétés système, vous pouvez utiliser une *expression* à la place de la valeur littérale. Les expressions prennent le format `${parameter:default}`. Quand une expression est utilisée dans une configuration, la valeur de ce paramètre prend sa place. Si le paramètre n'existe pas, alors la valeur par défaut indiquée sera utilisée à la place.

Exemple 2.14. Utiliser une expression dans un descripteur

```
<activation-config>
  <activation-config-property>
    <activation-config-property-name>
      connectionParameters
    </activation-config-property-name>
    <activation-config-property-value>
      ${jms.connection.parameters:'host=10.10.64.1;port=5445'}
    </activation-config-property-value>
  </activation-config-property>
</activation-config>
```

L'expression `${jms.connection.parameters:'host=10.10.64.1;port=5445'}` permet aux paramètres de connexion d'être remplacés par un paramètre fourni en ligne de commande, tout en donnant une valeur par défaut.

[Rapporter un bogue](#)

2.5.3. Activer/Désactiver un remplacement de propriété basé descripteur

Résumé

Le contrôle précis du remplacement de propriété de descripteur a été introduit dans `jboss-as-ee_1_1.xsd`. Cette tâche couvre les étapes requises pour configurer le remplacement de propriété basé descripteur.

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#)
- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Les indicateurs de remplacement de propriété basé descripteur ont les valeurs booléennes suivantes :

- Si défini à `true`, les remplacements de propriété sont activés.
- Si défini à `false`, les remplacements de propriété sont désactivés.

Procédure 2.12. `jboss-descriptor-property-replacement`

`jboss-descriptor-property-replacement` est utilisé pour activer ou désactiver le remplacement de propriété dans les descripteurs suivants :

- `jboss-ejb3.xml`

- `jboss-app.xml`
- `jboss-web.xml`
- `*-jms.xml`
- `*-ds.xml`

La valeur par défaut de `jboss-descriptor-property-replacement` est `true`.

1. À l'aide de l'interface CLI, exécuter la commande suivante pour déterminer la valeur de `jboss-descriptor-property-replacement`:

```
/subsystem=ee:read-attribute(name="jboss-descriptor-property-replacement")
```

2. Exécuter la commande suivante pour configurer le comportement :

```
/subsystem=ee:write-attribute(name="jboss-descriptor-property-replacement",value=VALUE)
```

Procédure 2.13. `spec-descriptor-property-replacement`

`spec-descriptor-property-replacement` est utilisé pour activer ou désactiver le remplacement de propriété dans les descripteurs suivants :

- `ejb-jar.xml`
- `persistence.xml`
- `application.xml`
- `web.xml`

La valeur par défaut de `spec-descriptor-property-replacement` est `false`.

1. À l'aide de l'interface CLI, exécuter la commande suivante pour confirmer la valeur de `spec-descriptor-property-replacement`:

```
/subsystem=ee:read-attribute(name="spec-descriptor-property-replacement")
```

2. Exécuter la commande suivante pour configurer le comportement :

```
/subsystem=ee:write-attribute(name="spec-descriptor-property-replacement",value=VALUE)
```

Résultat

Les indicateurs de remplacement de propriété basé descripteur ont bien été configurés.

[Rapporter un bogue](#)

2.5.4. Expressions imbriquées

Les expressions peuvent être imbriquées, ce qui permet une utilisation plus avancée des expressions au lieu de valeurs fixes. Le format d'une expression imbriquée ressemble à celui d'une expression normale, mais une expression est imbriquée dans l'autre, comme par exemple :

Exemple 2.15. Transactions imbriquées

```
${system_value_1${system_value_2}}
```

Les expressions imbriquées sont évaluées de façon récursive, donc l'expression *inner* est d'abord évaluée, puis *outer*. Les expressions imbriquées sont autorisées partout où elles sont autorisées, à l'exception des commandes CLI de gestion.

En ce qui concerne les expressions normales, les sources prises en charge pour résoudre les expressions imbriquées sont : propriétés système, variables d'environnement et de l'archivage sécurisé. Pour les déploiements uniquement, la source peut correspondre à des propriétés qui sont listées dans un fichier `META-INF/jboss.properties` qui se trouve dans l'archive de déploiement. Dans un EAR ou dans un autre type de déploiement qui prenne en charge les sous-déploiements, la résolution est scoped à tous les sous-déploiements si `META-INF/jboss.properties` se trouve dans le déploiement externe (par ex. EAR) et est scoped à un sous-déploiement si `META-INF/jboss.properties` est dans l'archive d'un sous-déploiement (par ex. un WAR à l'intérieur d'une EAR).

Exemple 2.16. Utiliser une expression imbriquée dans un fichier de configuration

Une application real-life d'une expression imbriquée se trouve dans une définition de source de données. Si le mot de passe utilisé dans une définition de source de données est masqué, voici la ligne de texte qui en résulte dans la définition de la source de données :

```
<password>${VAULT::ds_ExampleDS::password::1}</password>
```

Pour une expression imbriquée, la valeur de `ds_ExampleDS` peut être remplacée par une propriété système. Si une propriété système `datasource_name` reçoit comme valeur `ds_ExampleDS`, la ligne de texte de la définition de source de données pourra ressembler à ceci :

```
<password>${VAULT::${datasource_name}::password::1}</password>
```

JBoss EAP doit tout d'abord évaluer l'expression `${datasource_name}`, puis l'ajouter à une plus grande expression et évaluer l'expression qui en résulte. L'avantage de cette configuration est que le nom de la source de données se soustrait de la configuration fixe

Les expressions peuvent également être récursives quand une expression est résolue en expression qui est résolue à son tour. Les expressions imbriquées et récursives représentent une forme d'indirection. Notez que les expressions récursives ne sont pas permises dans les commandes d'interface CLI.

Exemple 2.17. Expression récursive

Pour l'exemple ci-dessus, vous pouvez utiliser l'expression `${foo}` qui résolue en expression `${VAULT::ds_ExampleDS::password::1}`, elle-même résolue par une valeur contenue dans l'archivage sécurisé : `secret`.

[Rapporter un bogue](#)

2.5.5. Historique du fichier de configuration

Les fichiers de configuration du serveur d'applications incluent `standalone.xml`, ainsi que les fichiers `domain.xml` et `host.xml`. Alors que ces fichiers sont modifiables directement, la méthode recommandée consiste à configurer le modèle de serveur d'applications par les opérations de gestion disponibles, y compris l'interface CLI et la console de gestion.

Pour aider à la maintenance et à la gestion de l'instance de serveur, le serveur d'applications crée une version horodatée du fichier de configuration original au moment du démarrage. Toutes les modifications de configuration supplémentaires suite aux opérations de gestion résultent à la sauvegarde automatique du fichier d'origine, et une copie de travail de l'instance est alors conservée en tant que référence et pour la restauration. Cette fonctionnalité d'archivage s'étend à l'enregistrement, au chargement et à la suppression des snapshots de configuration du serveur pour autoriser les scénarios de rappel et de restauration.

- [Section 2.5.6, « Démarrer le serveur par une ancienne configuration »](#)
- [Section 2.5.7, « Sauvegarder un snapshot de configuration par l'interface CLI »](#)
- [Section 2.5.8, « Charger un snapshot de configuration par l'interface CLI. »](#)
- [Section 2.5.9, « Supprimer un snapshot de configuration par l'interface CLI »](#)
- [Section 2.5.10, « Lister tous les snapshots de configuration par l'interface CLI »](#)

[Rapporter un bogue](#)

2.5.6. Démarrer le serveur par une ancienne configuration

L'exemple suivant vous montre comment démarrer le serveur d'applications par une ancienne configuration dans un serveur autonome par `standalone.xml`. Le même concept s'applique à un domaine géré par `domain.xml` et `host.xml` respectivement.

Cet exemple nous remémore une ancienne configuration sauvegardée automatiquement par le serveur d'applications tandis que les opérations de gestion modifient le modèle de serveur.

Exemple 2.18. Démarrer le serveur avec une configuration sauvegardée

1. Identifier la version de sauvegarde que vous souhaitez démarrer. Cet exemple va rappeler l'instance de modèle de serveur qui précédait la première modification qui a lieu suite à un démarrage réussi.

```
EAP_HOME/standalone/configuration/standalone_xml_history/current/standalone.v1.xml
```

2. Démarrer le serveur par cette configuration du modèle de sauvegarde en passant le nom de fichier relatif sous `jboss.server.config.dir`.

```
EAP_HOME/bin/standalone.sh --server-config=standalone_xml_history/current/standalone.v1.xml
```

Résultat

Le serveur d'applications démarre par la configuration sélectionnée.

**NOTE**

L'historique de la configuration du domaine se trouve dans `EAP_HOME/domain/configuration/domain_xml_history/current/domain.v1.xml`

Démarrer le serveur par cette configuration du modèle de sauvegarde en passant le nom de fichier relatif sous `jboss.domain.config.dir`.

Pour démarrer le domaine par cette configuration :

```
EAP_HOME/bin/domain.sh --domain-
config=domain_xml_history/current/domain.v1.xml
```

[Rapporter un bogue](#)

2.5.7. Sauvegarder un snapshot de configuration par l'interface CLI**Résumé**

Les snapshots représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur.

L'exemple suivant utilise le fichier de configuration `standalone.xml`, mais le même processus s'applique aux fichiers de configuration `domain.xml` et `host.xml`.

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 2.14. Télécharger un snapshot de configuration et sauvegardez-le

- **Sauvegarde d'un snapshot**
Exécuter l'opération `take-snapshot` pour acquérir une copie de la configuration du serveur.

```
[standalone@localhost:9999 /] :take-snapshot
{
  "outcome" => "success",
  "result" =>
"/home/User/EAP_HOME/standalone/configuration/standalone_xml_history
/snapshot/20110630-172258657standalone.xml"
```

Résultat

Un snapshot de la configuration du serveur en cours a été sauvegardé.

[Rapporter un bogue](#)

2.5.8. Charger un snapshot de configuration par l'interface CLI.

Les snapshots de configuration représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur. Le processus de chargement des snapshots ressemble à celui de la méthode pour [Section 2.5.6, « Démarrer le serveur par une ancienne configuration »](#), à partir de la ligne de commande et non pas l'interface CLI utilisée pour créer, lister et supprimer les snapshots.

L'exemple suivant utilise le fichier `standalone.xml`, mais le même processus s'applique aux fichiers `domain.xml` et `host.xml`.

Procédure 2.15. Télécharger un snapshot de configuration

1. Identifier le snapshot à télécharger. Cet exemple va rappeler le fichier suivant du répertoire de snapshots. Le chemin par défaut des fichiers de snapshot est le suivant.

```
EAP_HOME/standalone/configuration/standalone_xml_history/snapshot/20110812-191301472standalone.xml
```

Les snapshots sont exprimés par leurs chemins relatifs, selon lesquels l'exemple ci-dessus peut être écrit ainsi.

```
jboss.server.config.dir/standalone_xml_history/snapshot/20110812-191301472standalone.xml
```

2. Démarrer le serveur par le snapshot de configuration sélectionné en passant le nom du fichier.

```
EAP_HOME/bin/standalone.sh --server-config=standalone_xml_history/snapshot/20110913-164449522standalone.xml
```

Résultat

Le serveur démarre à nouveau avec la configuration sélectionnée dans le snapshot téléchargé.

[Rapporter un bogue](#)

2.5.9. Supprimer un snapshot de configuration par l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Les snapshots représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur.

Les exemples suivants utilisent le fichier `standalone.xml`, mais le même processus s'applique aux fichiers `domain.xml` et `host.xml`.

Procédure 2.16. Supprimer un snapshot particulier

1. Identifier le snapshot à effacer. Cet exemple va effacer le fichier suivant du répertoire de snapshots.

```
EAP_HOME/standalone/configuration/standalone_xml_history/snapshot/20110630-165714239standalone.xml
```

- 2. Exécuter la commande `:delete-snapshot` pour supprimer un snapshot particulier, en spécifiant le nom du snapshot comme dans l'exemple ci-dessous.

```
[standalone@localhost:9999 /] :delete-snapshot(name="20110630-165714239standalone.xml")
{"outcome" => "success"}
```

Résultat

Le snapshot a été supprimé.

Procédure 2.17. Supprimer tous les snapshots

- Exécuter la commande `:delete-snapshot(name="all")` pour supprimer tous les snapshots comme dans l'exemple ci-dessous.

```
[standalone@localhost:9999 /] :delete-snapshot(name="all")
{"outcome" => "success"}
```

Résultat

Tous les snapshots ont été supprimés.

[Rapporter un bogue](#)

2.5.10. Lister tous les snapshots de configuration par l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Les snapshots représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur.

L'exemple suivant utilise le fichier `standalone.xml`, mais le même processus s'applique aux fichiers `domain.xml` et `host.xml`.

Procédure 2.18. Lister tous les snapshots de configuration

- **Lister tous les snapshots**
Lister tous les snapshots sauvegardés en exécutant la commande `:list-snapshots`.

```
[standalone@localhost:9999 /] :list-snapshots
{
  "outcome" => "success",
  "result" => {
    "directory" =>
"/home/hostname/EAP_HOME/standalone/configuration/standalone_xml_history/snapshot",
    "names" => [
      "20110818-133719699standalone.xml",
      "20110809-141225039standalone.xml",
      "20110802-152010683standalone.xml",
```

```
    ]  
  }  
}
```

```
"20110808-161118457standalone.xml",  
"20110912-151949212standalone.xml",  
"20110804-162951670standalone.xml"
```

Résultat

Les snapshots sont listés.

[Rapporter un bogue](#)

CHAPITRE 3. INTERFACES DE GESTION

3.1. GESTION DU SERVEUR D'APPLICATIONS

JBoss EAP 6 vous propose des outils de gestion multiples pour configurer et administrer votre implémentation suivant les besoins. Ces outils comprennent la nouvelle console de gestion et l'interface CLI, comme exemples d'API de gestion pour permettre aux utilisateurs experts de développer leurs propres outils s'ils le désirent.

[Rapporter un bogue](#)

3.2. LES API (DE L'ANGLAIS APPLICATION PROGRAMMING INTERFACES) DE GESTION

Clients de gestion

JBoss EAP 6 offre trois approches différentes pour configurer et gérer des serveurs, une interface web, un client de ligne de commande et un ensemble de fichiers de configuration XML. Tandis que les méthodes recommandées pour la modification du fichier de configuration incluent la console de gestion et l'interface CLI, les modifications de configuration de la part des trois sont toujours synchronisées à travers les différentes vues et sont conservées dans les fichiers XML. Notez que les modifications apportées aux fichiers de configuration XML pendant l'exécution d'une instance de serveur seront remplacées par le modèle de serveur.

HTTP API

Le point de terminaison de l'API HTTP est le point d'entrée pour les clients de gestion, comme par exemple la Console de gestion, qui dépend du protocole HTTP pour s'intégrer à la couche de gestion. Ce point de terminaison utilise un protocole JSON encodé et un API de style RPC de-typed, pour décrire et exécuter des opérations de gestion dans un domaine géré ou un serveur autonome. Il est utilisé par la console web, mais offre aussi des possibilités d'intégration pour un large éventail d'autres clients.

Le point de terminaison HTTP API est situé près du contrôleur de domaine ou d'une instance de serveur autonome. Il sert deux contextes différents; un pour l'exécution des opérations de gestion et de l'autre pour accéder à l'interface web. Par défaut, il s'exécute sur le port 9990.

Exemple 3.1. Exemple de fichier de configuration HTTP API

```
<management-interfaces>
  [...]
  <http-interface security-realm="ManagementRealm">
    <socket-binding http="management-http"/>
  </http-interface>
</management-interfaces>
```

La console web est desservie par le même port que l'API de gestion HTTP. Il est important de distinguer la façon dont on accède à la Console de gestion ou à l'API HTTP : la Console de gestion comme localhost par défaut, la Console de gestion à distance par un hôte spécifique et une combinaison de ports, et l'API HTTP exposé du domaine.

Tableau 3.1. Les URL d'accès à la Console de gestion ou à l'API HTTP exposé

URL	Description
<code>http://localhost:9990/console</code>	La console de gestion à laquelle accède l'hôte local, et qui contrôle la configuration du domaine géré.
<code>http://hostname:9990/console</code>	La console de gestion accédée à distance, qui nomme l'hôte et qui contrôle la configuration du domaine géré.
<code>http://hostname:9990/management</code>	L'API de gestion HTTP exécutée sur le même port que la console de gestion, affiche les mêmes valeurs et attributs bruts exposés à l'API.

API Natif

Le CLI de gestion est un exemple d'outil d'API Natif. Cet outil de gestion est disponible à une instance de serveur autonome ou à un domaine, permettant ainsi à un utilisateur de se connecter à une instance du serveur autonome ou au contrôleur du domaine, et d'exécuter des opérations de gestion rendues disponibles par le modèle de gestion de-types.

Le point de terminaison de l'API natif est le point d'entrée pour les clients de gestion qui s'appuient sur le protocole natif pour intégrer la couche de gestion. Il utilise un protocole binaire ouvert et une API style-RPC basée sur un très petit nombre de types Java pour décrire et exécuter des opérations de gestion. Il est utilisé par l'outil de gestion Interface CLI, mais offre des capacités d'intégration pour un large éventail d'autres clients également.

Le point de terminaison d'API natif est co-localisé avec un contrôleur hôte ou un serveur autonome. Il doit être activé pour utiliser l'interface CLI. Par défaut, il s'exécute sur le port 9999.

Exemple 3.2. Exemple de fichier de configuration d'API natif

```
<management-interfaces>
  <native-interface security-realm="ManagementRealm">
    <socket-binding native="management-native"/>
  </native-interface>
  [...]
</management-interfaces>
```

[Rapporter un bogue](#)

3.3. CONSOLE DE GESTION ET INTERFACE CLI

Dans JBoss EAP 6, toutes les instances de serveurs et toutes les configurations sont gérées par les interfaces de gestion, et non pas par modification de fichiers XML. Malgré que les fichiers de configuration XML puissent toujours être édités, la gestion par les interfaces de gestion fournit une validation supplémentaire et des fonctionnalités avancées pour la gestion persistante des instances de serveurs. Les modifications apportées aux fichiers de configuration XML, tandis que l'instance de serveur est en cours d'exécution, seront remplacées par le modèle de serveur, et des commentaires XML ajoutés disparaîtront ainsi. Seules les interfaces de gestion doivent être utilisées pour modifier les fichiers de configuration pendant l'exécution d'une instance de serveur

Pour gérer les serveurs par une interface utilisateur graphique d'un navigateur web, utiliser la console de gestion.

Pour gérer les serveurs par l'interface de ligne de commande, utiliser l'interface CLI.

[Rapporter un bogue](#)

3.4. LA CONSOLE DE GESTION

3.4.1. Console de management

La console de management est un outil administratif basé web pour la plateforme JBoss EAP 6.

Utilisez la console de management pour démarrer et arrêter des serveurs, déployer et annuler le déploiement des applications, régler les paramètres du système et apporter des modifications persistantes à la configuration du serveur. La console de management a également la capacité d'effectuer des tâches administratives, avec des notifications directes lorsque les modifications exigent que l'instance du serveur soit redémarrée ou rechargée.

Dans un domaine géré, les instances de serveur et les groupes de serveurs d'un même domaine peuvent être gérés de façon centralisée à partir de la console de management du contrôleur de domaine.

[Rapporter un bogue](#)

3.4.2. Se connecter à la console de gestion

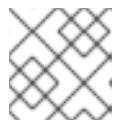
Conditions préalables

- Vous devez créer un utilisateur administratif comme indiqué ici : [Section 4.1.1, « Ajouter un utilisateur pour les interfaces de gestion »](#).

JBoss EAP doit être en cours d'exécution.

1. Naviguer vers la page de démarrage de la console de gestion

Lancez votre navigateur web et accédez à la Console de gestion dans votre navigateur web dans <http://localhost:9990/console/App.html>



NOTE

Port 9990 est prédéfini en tant que liaison de socket de Console de gestion.

2. Saisir le nom d'utilisateur et le mot de passe du compte que vous avez déjà créé pour vous connecter à l'écran de connexion de la console de gestion.

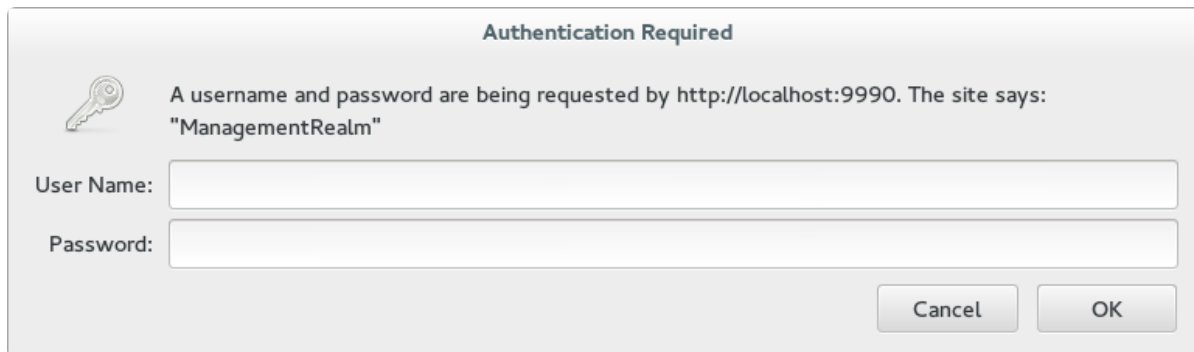


Figure 3.1. Écran de connexion de la console de gestion

Résultat

Une fois connecté, vous serez redirigé à l'adresse suivante et la page d'accueil de la Console de gestion apparaîtra : <http://localhost:9990/console/App.html#home>

[Rapporter un bogue](#)

3.4.3. Changer la langue de la console de management

Les paramètres de configuration de la console de management basée web utilisent l'anglais par défaut. Vous pouvez décider d'utiliser une des langues suivantes à la place.

Langues prises en charge

- Allemand (de)
- Chinois simplifié (zn-Hans)
- Portugais brésilien (pt-BR)
- Français (fr)
- Espagnol (es)
- Japonais (ja)

Procédure 3.1. Changer la langue de la console de management basée-web

1. Connectez-vous à la console de gestion.

Connectez-vous à la console de management basée web.

2. Ouvrir le dialogue de configuration.

Dans le coin gauche de l'écran, il y a une étiquette **Settings** de configuration. Cliquer sur cette étiquette pour ouvrir les paramètres de configuration de la console de management.

3. Sélectionner la langue désirée.

Sélectionner la langue désirée à partir de la case **Locale**. Puis sélectionner **Save**. Une autre case explicative vous demande de charger à nouveau l'application. Cliquer sur **Confirm**. Le système réactualise votre navigateur automatiquement pour pouvoir utiliser les nouveaux paramètres régionaux (Locale).

[Rapporter un bogue](#)

3.4.4. Données analytiques dans la console EAP

Google Analytics

Google Analytics est un service analytique web gratuit qui fournit des statistiques complètes sur un site Web. Il fournit des données essentielles relatives aux visiteurs d'un site comme leurs visites, pages vues, pages par visite et moyenne de temps passé sur le site. Google Analytics fournit une meilleure visibilité sur la présence d'un site Web et ses utilisateurs.

Google Analytics dans la console d'administration EAP

JBoss EAP 6 fournit aux utilisateurs l'option d'activer/de désactiver Google Analytics dans la console de gestion. La fonctionnalité Google Analytics vise à aider les équipes de Red Hat EAP à comprendre comment les clients utilisent la console et quelles parties de la console est particulièrement d'importance aux clients. Cette information aidera l'équipe à adapter l'apparence de la console, ses fonctionnalités et le contenu aux besoins immédiats des clients.

[Rapporter un bogue](#)

3.4.5. Activer/désactiver Google Analytics dans la console EAP

Pour activer Google Analytics dans la console d'administration EAP :

- Connectez-vous à la console d'administration
- Cliquer sur le bouton **Settings** en bas et à droite de la console

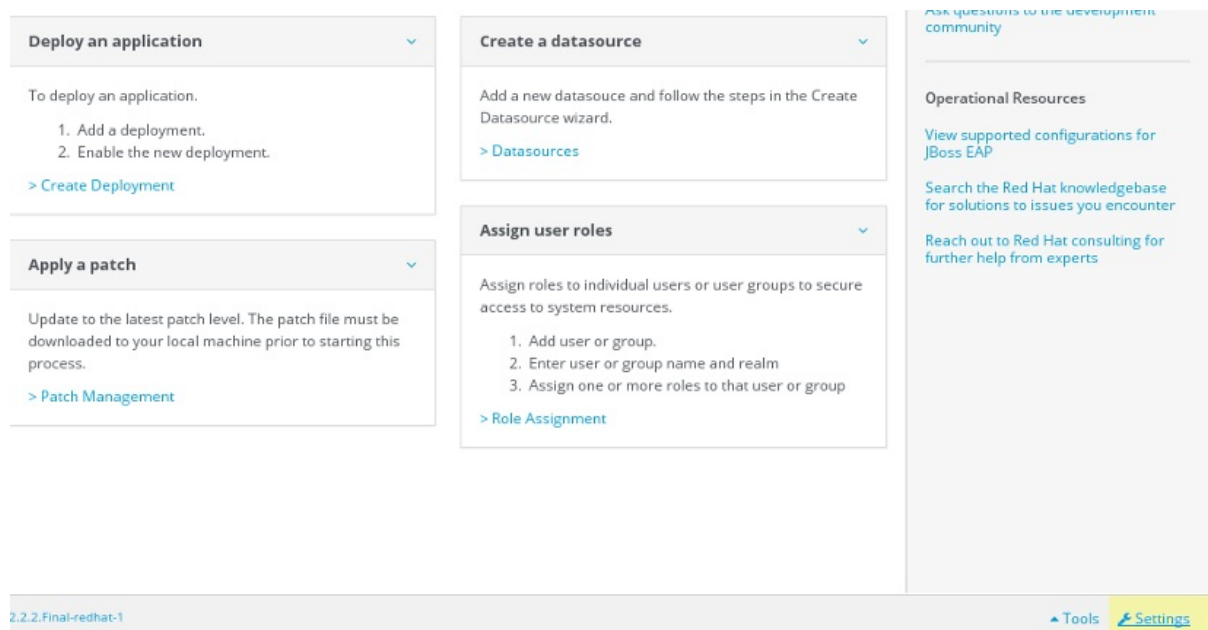


Figure 3.2. Connectez-vous à l'écran de console d'administration

- Sélectionner la case **Enable Usage Data Collection** sur la fenêtre **Settings** et cliquer sur le bouton **Save**. Confirmer le rechargement de l'application pour activer les nouveaux paramètres de configuration.

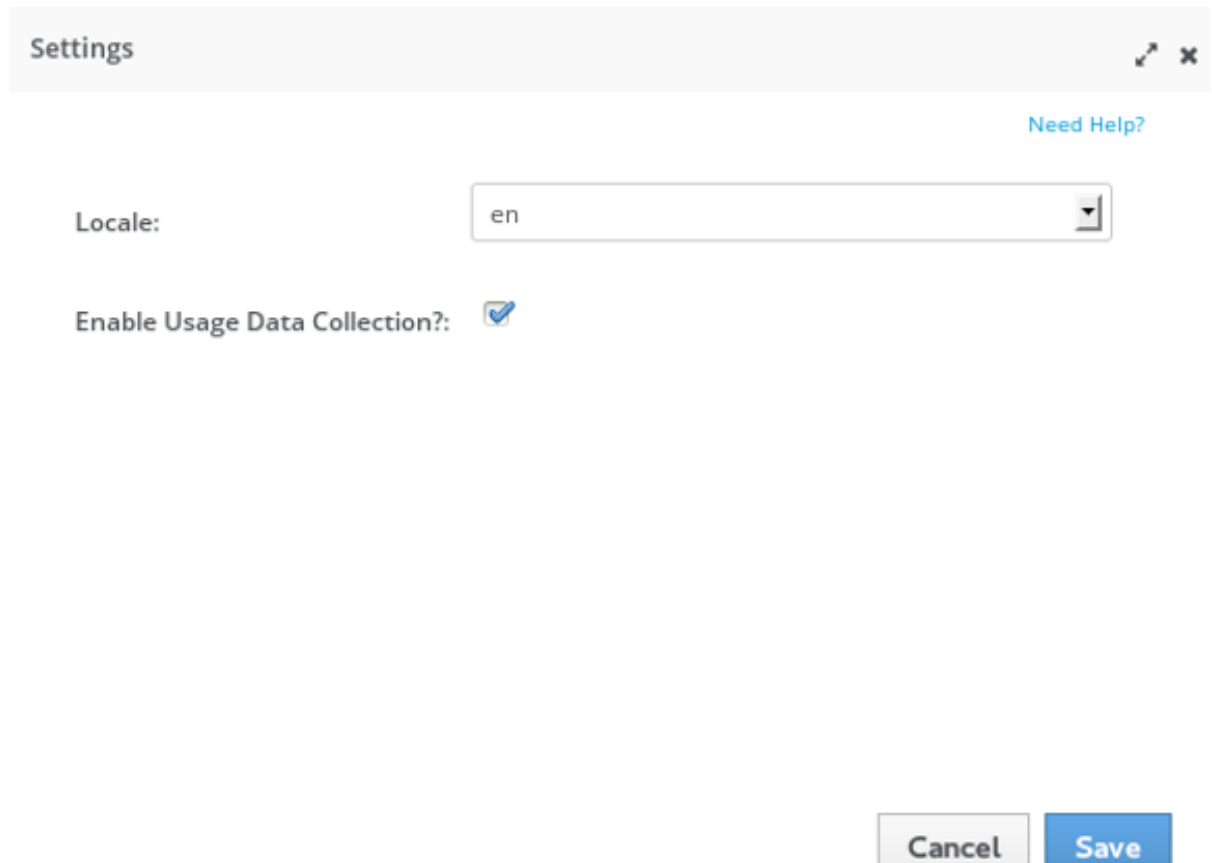


Figure 3.3. Configurer Window (Activer Usage Data Collection)

Pour désactiver Google Analytics dans la console d'administration après l'avoir activé, cliquer sur **Enable Usage Data Collection** dans la fenêtre de paramétrage **Settings** pour supprimer la sélection, puis cliquer sur le bouton **Save**. Confirmer le rechargement de l'application pour activer les nouveaux paramètres de configuration.

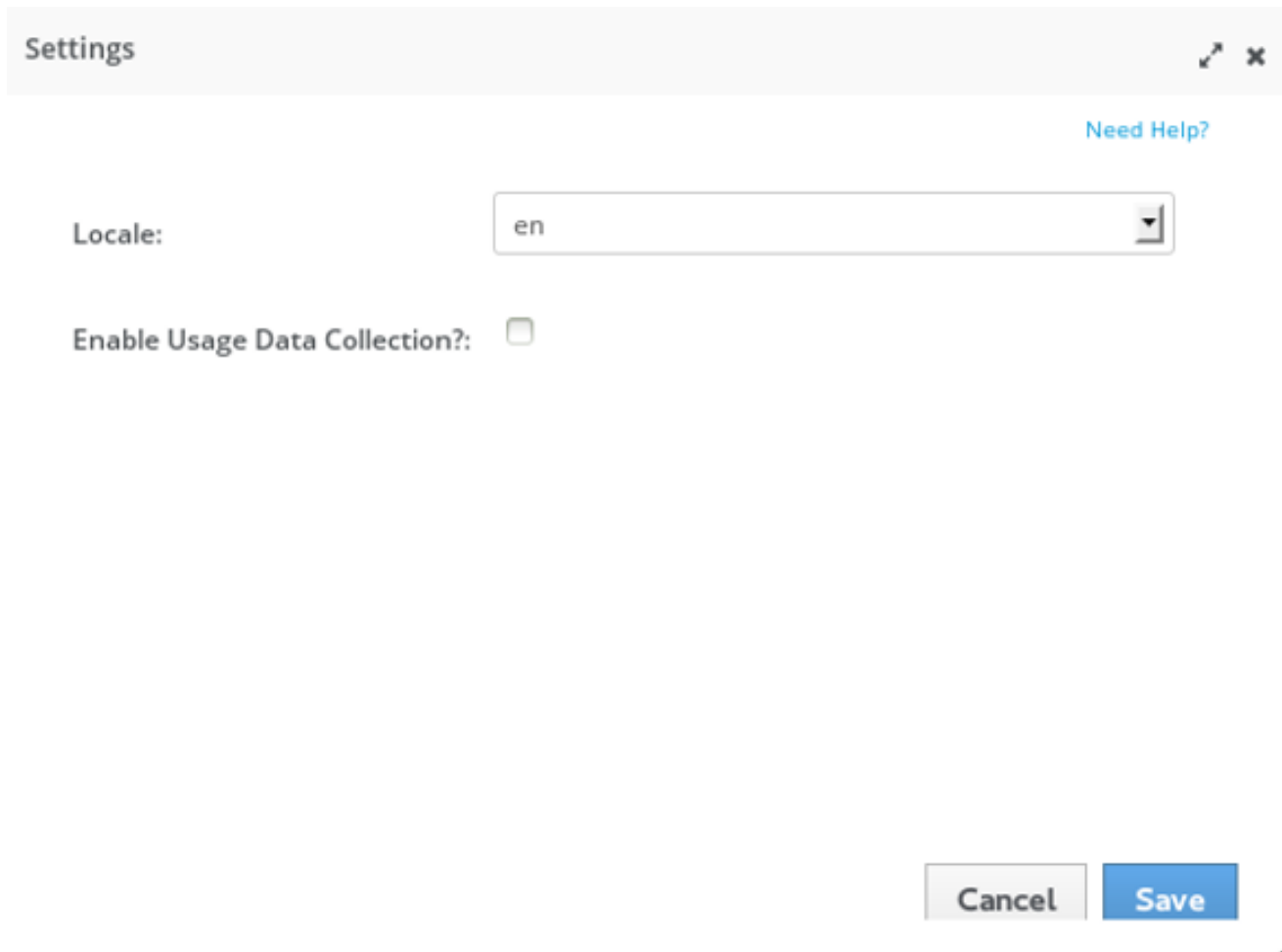


Figure 3.4. Configurer Window (Désactiver Usage Data Collection)



NOTE

Google Analytics est désactivé par défaut dans la console d'administration EAP 6 et son utilisation est optionnelle

[Rapporter un bogue](#)

3.4.6. Configurer un serveur par la console de management

Conditions préalables

- [Section 2.2.3, « Démarrez JBoss EAP 6 comme domaine géré »](#)
- [Section 3.4.2, « Se connecter à la console de gestion »](#)

Procédure 3.2. Configurer le serveur

1. Sélectionner l'onglet **Domain** en haut de la console. Les instances de serveur disponibles s'afficheront sur un tableau.
2. Sélectionner l'instance de serveur à partir du tableau **Available Server Configurations**.
3. Cliquer sur **Edit** au dessus des informations sur le serveur choisi.

4. Procédez avec les changements des attributs de configuration.
5. Cliquer sur le bouton **Save** pour terminer.

Résultat

La configuration du serveur a changé, et prendra effet la prochaine fois que le serveur démarrera.

[Rapporter un bogue](#)

3.4.7. Ajouter un déploiement dans une console de management

Conditions préalables

- [Section 3.4.2, « Se connecter à la console de gestion »](#)
1. Sélectionner l'onglet **Runtime** en haut de la console.
 2. Pour un serveur autonome, il vous faudra étendre l'élément de menu **Server** et sélectionner **Manage Deployments**. Pour un domaine géré, étendre l'élément de menu **Domain** et sélectionner **Manage Deployments**. Le panneau **Manage Deployments** apparaîtra.
 3. Sélectionner **Add** dans l'onglet **Content Repository**. Une boîte de dialogue **Create Deployment** apparaîtra.
 4. Dans la boîte de dialogue, cliquer sur **Browse**. Cherchez le fichier que vous souhaitez déployer, et sélectionnez-le en vue de son chargement. Cliquer sur **Next** pour continuer.
 5. Vérifier le nom du déploiement et le nom du runtime qui apparaît dans la boîte de dialogue **Create Deployments**. Sélectionner **Save** pour charger le fichier une fois que les noms auront été vérifiés.

Résultat

Le contenu sélectionné est téléchargé dans le serveur et est maintenant prêt à être déployé.

[Rapporter un bogue](#)

3.4.8. Créer un nouveau serveur dans la console de management

Conditions préalables

- [Section 2.2.3, « Démarrez JBoss EAP 6 comme domaine géré »](#)
- [Section 3.4.2, « Se connecter à la console de gestion »](#)

Procédure 3.3. Créer une nouvelle configuration de serveur

1. Naviguer sur la page **Server Configuration** qui se trouve dans la console de management
Sélectionner l'onglet **Domain** en haut de la console.
2. **Créer une nouvelle configuration**
 - a. Sélectionner le bouton **Add** qui se trouve en haut du tableau **Available Server Configuration**.

- b. Saisir les paramètres de base du serveur dans la boîte de dialogue **Create Server Configuration**.
- c. Cliquer sur le bouton **Save** pour enregistrer la nouvelle configuration de votre serveur.

Résultat

Le nouveau serveur créé est listé dans **Server Configurations**.

[Rapporter un bogue](#)

3.4.9. Modifier les niveaux de journalisation par défaut en utilisant la console de management

Procédure 3.4. Modifier les niveaux de journalisation

1. **Naviguer dans le panneau Logging de la console de gestion.**
 - a. Si vous travaillez dans un domaine géré, sélectionner l'onglet **Configuration** en haut de la console, puis, sélectionner le profil qui convient dans la liste déroulante à gauche de la console.
 - b. Pour un domaine géré ou un serveur autonome, étendre le menu **Core** qui se trouve à gauche de la console et cliquer sur **Logging**.
 - c. Cliquer sur l'onglet **Log Categories** en haut de la console.
2. **Modifier les informations du créateur du journal**
Modifier les informations des entrées du tableau **Log Categories**.
 - a. Sélectionner une entrée dans le tableau **Log Categories**, puis cliquer sur **Edit** dans la section **Details** ci-dessous.
 - b. Définir le niveau de journalisation de la catégorie dans la zone déroulante **Log Level**. Cliquer sur le bouton **Save** quand c'est fait.

Résultat

Les niveaux de journalisation des catégories qui conviennent sont maintenant mis à jour.

[Rapporter un bogue](#)

3.4.10. Créer un nouveau groupe de service dans la console de gestion

Conditions préalables

- [Section 3.4.2, « Se connecter à la console de gestion »](#)

Procédure 3.5. Configurer et ajouter un groupe de serveurs

1. **Se rendre dans la vue Server Groups**
Sélectionner l'onglet **Domain** au haut de la console.
2. Étendre l'étiquette **Server** dans le menu qui se trouve en haut de la colonne. Sélectionner **Server Groups**.

3. Ajouter un groupe de serveurs

Cliquer sur le bouton **Add** pour ajouter un nouveau groupe de serveurs.

4. Configurer le groupe de serveurs

- a. Saisir un nom pour le groupe de serveurs
- b. Sélectionner le profil d'un groupe de serveurs.
- c. Sélectionner la liaison de socket du groupe de serveurs.
- d. Cliquer sur le bouton **Save** pour sauvegarder le nouveau groupe.

Résultat

Le nouveau groupe de serveurs est visible dans la console de gestion.

[Rapporter un bogue](#)

3.4.11. Visualisation des journaux dans la console de gestion

Vous pouvez afficher des journaux de serveur et d'application dans la console de gestion JBoss EAP 6 pour aider à diagnostiquer des erreurs, des problèmes de performance et d'autres questions. Pour être visible dans la visionneuse du journal de console de gestion d'un journal, elle doit se situer dans le répertoire du serveur `jboss.server.log.dir`. La visionneuse du journal JBoss EAP 6 respecte également les attributions de rôles d'utilisateur RBAC, donc un utilisateur connecté à la console de gestion ne peut voir que des journaux pour lesquels il a l'autorisation d'accès.

Conditions préalables

- [Section 3.4.2, « Se connecter à la console de gestion »](#)

Procédure 3.6. Visualiser les journaux de JBoss EAP 6 dans la console de gestion

1. Sélectionner l'onglet **Runtime** en haut de la console de gestion.
 - a. Si vous utilisez un domaine géré, utilisez le bouton **Change Server** dans le menu de gauche pour sélectionner le serveur JBoss EAP 6 dont vous souhaitez afficher les journaux.
2. Étendre le menu **Platform** sur la gauche, et sélectionner le **Log Viewer**.
3. Sélectionner le fichier de journalisation de la liste, et cliquer sur le bouton **View**.

Vous pouvez également cliquer sur le bouton **Download** pour télécharger le fichier de journalisation de la machine locale.

**NOTE**

Le Log Viewer de la console de gestion affiche une confirmation si vous tentez d'ouvrir un fichier de journalisation plus grand que 15 Mo.

Le Log Viewer de la console de gestion n'est pas destiné à être un éditeur de texte de remplacement pour l'affichage des fichiers de journalisation très volumineux (> 100Mo). L'ouverture de fichiers journaux très volumineux dans le Log Viewer de la console de gestion peut bloquer votre navigateur web, donc il faut toujours mieux télécharger les gros fichiers de journalisation séparément et les ouvrir dans un éditeur de texte.

4. Le journal sélectionné ouvrira un nouvel onglet dans la console de gestion, Vous pouvez ouvrir plusieurs fichiers de journalisation dans d'autres onglets en retournant à l'onglet **LOG FILES** et en répétant l'étape précédente.

[Rapporter un bogue](#)

3.4.12. Intégration du Portail clients dans la Console de gestion

Vous pouvez utiliser l'interface access.redhat.com pour naviguer dans les sections du Portail clients de Red Hat sans quitter la console de gestion de votre installation JBoss EAP.

La barre de navigation supérieure de la console de gestion contient un menu déroulant : **Red Hat Access**. En cliquant sur ce menu, vous apercevrez trois liens de tâches spécifiques au portail client :

- **Search Customer Portal**
- **Open Case**
- **Modify Case**

Les fonctionnalités de ces liens seront discutées en détails ci-dessous.

**NOTE**

Si vous n'êtes pas déjà connecté au portail clients lorsque vous cliquez sur un de ces liens, une boîte de dialogue s'affichera, vous invitant à vous connecter. Vous devez être connecté au portail clients dans la session du navigateur que vous utilisez pour accéder à la console de gestion. Si vous êtes connecté au portail client dans un navigateur, mais utilisez un autre navigateur pour accéder à la console de gestion, on vous demandera d'ouvrir une session.

Search Customer Portal

Cliquer sur **Search Customer Portal** vous mène à une page contenant une case de recherche. Vous pourrez y saisir des termes ou des phrases pour chercher des articles de base de connaissance.

Une fois que vous aurez effectué votre recherche, vous pourrez sélectionner un élément d'une liste de résultats et apercevoir l'article complet dans un panneau séparé.

Open Case

La page **Open Case** vous permet d'ouvrir un nouveau dossier de prise en charge.

On vous présentera un formulaire à remplir afin d'ouvrir un nouveau dossier de prise en charge. Une liste d'articles de Base de connaissances recommandés est fournie à côté du formulaire. Cette liste reflète les actualisations sur la base des détails fournis dans le cas de la prise en charge.

Modifier un cas

La page **Modify Case** vous permet de visualiser et de modifier des dossiers de prise en charge existants.

Vous pouvez raffiner les résultats en limitant votre recherche aux cas groupés ou non-groupés, en par statut de cas (ouvert, fermé, ou autre).

Après avoir sélectionné un cas de prise en charge spécifique, vous pourrez voir ou mettre à jour les détails du cas de prise en charge, ainsi qu'ajouter des commentaires.

[Rapporter un bogue](#)

3.5. L'INTERFACE CLI

3.5.1. Gestion par interface en ligne de commande (CLI)

La gestion par interface en ligne de commande (CLI) est un outil d'administration en ligne de commande pour JBoss EAP 6.

Utiliser l'interface CLI pour démarrer et stopper les serveurs, déployer et retirer les déploiements d'applications, configurer les paramètres du système, ou encore, effectuer d'autre tâches administratives. Les opérations peuvent être effectuées par mode de lots, ce qui permet à plusieurs tâches d'être exécutées en groupe.

[Rapporter un bogue](#)

3.5.2. Lancement de l'interface CLI

Conditions préalables :

- [Section 2.2.2, « Démarrez JBoss EAP 6 comme un serveur autonome »](#)
- [Section 2.2.3, « Démarrez JBoss EAP 6 comme domaine géré »](#)

Procédure 3.7. Lancement du CLI dans un serveur Linux ou Windows

- **Lancement du CLI dans Linux**
Exécutez le fichier `EAP_HOME/bin/jboss-cli.sh` en saisissant ce qui suit dans la ligne de commande :

```
$ EAP_HOME/bin/jboss-cli.sh
```

- **Lancement du CLI dans un serveur Microsoft Windows**
Exécutez le fichier `EAP_HOME/bin/jboss-cli.bat` en cliquant deux fois, ou en saisissant ce qui suit dans la ligne de commande :

```
C:\>EAP_HOME\bin\jboss-cli.bat
```

[Rapporter un bogue](#)

3.5.3. Quitter l'interface CLI

Avec le Management CLI, saisir la commande `quit` :

```
[domain@localhost:9999 /] quit
```

[Rapporter un bogue](#)

3.5.4. Se connecter à une instance de serveur géré par l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 3.8. Se connecter à une instance de serveur gérée

- Exécutez la commande `connect`

À l'aide de l'interface CLI, saisir la commande `connect` :

```
[disconnected /] connect
Connected to domain controller at localhost:9999
```

- Sinon, pour vous connecter à un serveur géré quand vous démarrez une interface CLI sur un système Linux, utiliser le paramètre `--connect` :

```
$ EAP_HOME/bin/jboss-cli.sh --connect
```

- Le paramètre `--connect` peut être utilisé pour indiquer l'hôte et le port du serveur. Pour connecter l'adresse `192.168.0.1` à la valeur du port `9999` ce qui suit s'applique :

```
$ EAP_HOME/bin/jboss-cli.sh --connect --
controller=192.168.0.1:9999
```

[Rapporter un bogue](#)

3.5.5. Comment obtenir de l'aide par l'interface CLI

Résumé

Vous aurez parfois besoin d'aide pour mémoriser une commande d'interface ou pour vous rassurer que vous ne vous trompez pas. L'interface de gestion CLI dispose d'une boîte de dialogue d'assistance avec des options générales et des options sensibles au contexte. (notez que les commandes d'assistance dépendant du contexte de l'opération nécessitent une connexion à un contrôleur de domaine ou à un serveur autonome. Ces commandes ne seront pas affichées dans la liste, sauf si la connexion a été établie.)

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

1. Obtenir de l'aide générale

Avec le CLI, saisir la commande `help` :

-

```
[standalone@localhost:9999 /] help
```

2. Obtenir de l'aide par rapport au contexte

Avec le CLI, saisir la commande étendue `help --commands` :

```
[standalone@localhost:9999 /] help --commands
```

3. Pour plus d'informations sur une commande particulière, exécuter la commande suivie de `--help`.

```
[standalone@localhost:9999 /] deploy --help
```

Résultat

L'information d'assistance du CLI s'affichera.

[Rapporter un bogue](#)

3.5.6. Utiliser l'interface CLI en mode par lot

Résumé

Le traitement par lot permet à un certain nombre de requêtes d'être groupées par séquences et exécutées ensemble par unité. Si une des demandes opérationnelles d'une séquence échoue, tout le groupe d'opérations sera annulé.



NOTE

Le mode en lot ne prend pas en charge les énoncés conditionnels.

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)
- [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#)

Procédure 3.9. Commandes et opérations en mode par lot

1. Saisir un mode par lot

Saisir le mode par lot par la commande `batch`.

```
[standalone@localhost:9999 /] batch  
[standalone@localhost:9999 / #]
```

Le mode par lot est indiqué par le signe (`#`) dans l'invite.

2. Ajouter les demandes opérationnelles au lot

Une fois que vous serez en mode par lot, saisir les demandes opérationnelles comme d'habitude. Les demandes opérationnelles sont ajoutées au lot dans l'ordre de saisie.

Voir [Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI »](#) pour obtenir des informations sur la façon de formater les demandes opérationnelles.

3. Exécuter le lot

Une fois que toute la séquence de demandes opérationnelles est saisie, exécuter le lot avec la commande **run-batch**.

```
[standalone@localhost:9999 / #] run-batch
The batch executed successfully.
```

Voir [Section 3.5.7, « Commandes CLI Mode Lot »](#) pour obtenir une liste complète des commandes disponibles pour pouvoir travailler avec des lots.

4. Les commandes de lots sont stockées dans des fichiers externes

Les commandes fréquemment exécutées peuvent être stockées dans un fichier texte externe et être chargées en passant le chemin d'accès complet au fichier comme argument à la commande **batch** ou exécutées directement en étant un argument à la commande **run-batch**.

Vous pouvez créer un fichier de commande lot avec l'éditeur de texte. Chaque commande doit figurer sur une ligne par elle-même et l'interface CLI doit être en mesure d'y accéder.

La commande suivante chargera un fichier **myscript.txt** en mode lot. Toutes les commandes de ce fichier seront alors être prêtes à la modification ou à la suppression. De nouvelles commandes pourront être ajoutées. Les modifications effectuées au cours de cette session de lot ne persistera pas dans le fichier **myscript.txt**.

```
[standalone@localhost:9999 /] batch --file=myscript.txt
```

Ce qui suit exécutera instantanément les commandes de lot stockées dans le fichier **myscript.txt**

```
[standalone@localhost:9999 /] run-batch --file=myscript.txt
```

Résultat

La séquence de demandes opérationnelles saisies est effectuée sous forme de lot.

[Rapporter un bogue](#)

3.5.7. Commandes CLI Mode Lot

Ce tableau vous fournit une liste de commandes en mode lot valides pouvant être utilisées dans JBoss EAP 6 CLI. Ces commandes ne peuvent être utilisées que pour travailler en lots.

Tableau 3.2. Commandes CLI Mode Lot

Command Name	Description
list-batch	Liste des commandes et des opérations du lot en cours.
edit-batch-line line-number edited-command	Modifier une ligne du lot en cours en donnant un numéro de ligne à modifier et la commande éditée. Exemple: edit-batch-line 2 data-source disable --name=ExampleDS .

Command Name	Description
move-batch-line fromline toline	Réorganiser les lignes dans le lot en spécifiant le numéro de ligne à déplacer comme premier argument et sa nouvelle position comme deuxième argument. Exemple : move-batch-line 3 1 .
remove-batch-line linenumber	Supprimer la commande de lot à la ligne indiquée. Exemple: remove-batch-line 3 .
holdback-batch [batchname]	<p>Vous pouvez reporter à plus tard ou stocker un lot en cours à l'aide de cette commande. Utiliser cette option si vous voulez soudainement exécuter quelque chose dans la CLI en dehors du lot. Pour revenir à ce lot en attente, tapez simplement batch à nouveau à la ligne de commande CLI.</p> <p>Si vous fournissez un nom de lot en utilisant la commande holdback-batch, le lot sera stocké sous ce nom. Pour retourner au lot nommé, utilisez la commande batch batchname. L'appel de la commande batch sans un nom de lot va commencer un nouveau lot (sans nom). Il peut y avoir qu'un seul lot suspendu sans nom.</p> <p>Pour voir une liste de tous les lots suspendus, utiliser la commande batch -l.</p>
discard-batch	Rejète le lot actif en cours.

Rapporter un bogue

3.5.8. Utiliser les opérations et les commandes de l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)
- [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#)

Procédure 3.10. Créer, configurer et exécuter les requêtes

1. Construire la demande opérationnelle

Les demandes opérationnelles facilitent une interaction de bas niveau dans le modèle de gestion. Il s'agit d'une façon contrôlée de modifier les configurations du serveur. Une demande opérationnelle se présente en trois parties :

- une *adresse*, avec une barre oblique devant (/).
- un *nom d'opération*, avec deux points (:).
- un groupe optionnel de *paramètres*, entre parenthèses (()).

a. Déterminer l'adresse

La configuration est présentée sous forme de ressources auxquelles s'adresser et de façon hiérarchique. Chaque noeud de ressources procure un groupe différent d'opérations. Une adresse utilise la syntaxe suivante :

```
/node-type=node-name
```

- *node-type* correspond au type de noeud de ressource. Cela correspond à un nom d'élément dans la configuration XML.
- *node-name* correspond au nom du nœud. Cela correspond à l'attribut `nom` de l'élément dans la configuration XML.
- Séparer chaque niveau de l'arborescence de ressources par une barre oblique (/).

Voir les fichiers de configuration XML pour déterminer l'adresse qui convient. Le fichier `EAP_HOME/standalone/configuration/standalone.xml` contient la configuration d'un serveur autonome et les fichiers `EAP_HOME/domain/configuration/domain.xml` et `EAP_HOME/domain/configuration/host.xml` contiennent la configuration d'un domaine géré.



NOTE

Exécuter les commandes CLI en mode de domaine requiert que l'hôte et le serveur soient indiqués. Par exemple, `/host=master/server=server-one/subsystem=logging`

Exemple 3.3. Exemple d'adresses d'opérations

Pour procéder à une opération dans le sous-système de journalisation, utiliser l'adresse suivante dans la demande opérationnelle :

```
/subsystem=logging
```

Pour effectuer une opération sur la source de données Java, utiliser l'adresse suivante dans la demande opérationnelle :

```
/subsystem=datasources/data-source=java
```

b. Déterminer l'opération

Les opérations diffèrent avec chaque type de nœud de ressource. Une opération utilise la syntaxe suivante :

```
:operation-name
```

- *operation-name* correspond au nom de l'opération à demander.

Utiliser l'opération `read-operation-names` sur une adresse de ressources d'un serveur autonome pour lister les opérations disponibles.

Exemple 3.4. Opérations disponibles

Pour énumérer toutes les opérations disponibles du sous-système de journalisation, saisir la requête suivante dans un serveur autonome :

```
[standalone@localhost:9999 /] /subsystem=logging:read-
operation-names
{
  "outcome" => "success",
  "result" => [
    "add",
    "read-attribute",
    "read-children-names",
    "read-children-resources",
    "read-children-types",
    "read-operation-description",
    "read-operation-names",
    "read-resource",
    "read-resource-description",
    "remove",
    "undefine-attribute",
    "whoami",
    "write-attribute"
  ]
}
```

c. Déterminer un paramètre

Chaque opération a sans doute besoin de paramètres différents.

Les paramètres utilisent la syntaxe suivante :

```
(parameter-name=parameter-value)
```

- *parameter-name* correspond au nom du paramètre.
- *parameter-value* correspond à la valeur du paramètre.
- Les différents paramètres sont séparés par des virgules (,).

Afin de déterminer les paramètres qui conviennent, exécutez la commande **read-operation-description** sur un nœud de ressource, en faisant passer le nom de l'opération en tant que paramètre. Voir [Exemple 3.5, « Déterminer les paramètres des opérations »](#) pour plus de détails.

Exemple 3.5. Déterminer les paramètres des opérations

Afin de déterminer les paramètres qui conviennent pour l'opération **read-children-types** sur le sous-système de journalisation, saisir la commande **read-operation-description** comme suit :

```
[standalone@localhost:9999 /] /subsystem=logging:read-
operation-description(name=read-children-types)
{
  "outcome" => "success",
  "result" => {
```



```

        "operation-name" => "read-children-types",
        "description" => "Gets the type names of all the
children under the selected resource",
        "reply-properties" => {
            "type" => LIST,
            "description" => "The children types",
            "value-type" => STRING
        },
        "read-only" => true
    }
}

```

2. Saisir toute la demande opérationnelle

Une fois que l'adresse, l'opération et tous les paramètres auront été sélectionnés, saisir la demande opérationnelle complète.

Exemple 3.6. Exemple de demande opérationnelle

```

[standalone@localhost:9999 /] /subsystem=web/connector=http:read-
resource(recursive=true)

```

Résultat

L'interface de gestion effectue la demande opérationnelle sur la configuration du serveur.

[Rapporter un bogue](#)

3.5.9. Options de configuration de l'interface CLI

Le fichier de configuration de l'interface CLI - `jboss-cli.xml` - est chargé à chaque fois que le CLI démarre. Il doit se situer dans le répertoire `$EAP_HOME/bin` ou dans un système spécifié dans la propriété système `jboss.cli.config`.

default-controller

Configuration du contrôleur auquel il faut vous connecter si la commande `connect` est exécutée sans paramètre.

Paramètres de contrôleur par défaut

host

Nom d'hôte du contrôleur. La valeur par défaut est : `localhost`.

port

Numéro de port sur lequel il faut se connecter au contrôleur. La valeur par défaut est `9999`.

validate-operation-requests

Indique si la liste des paramètres des demandes d'opération doit être validée avant que les demandes ne soient envoyées au contrôleur pour être exécutées. Type : Booléen. Valeur par défaut : `true`.

history

Cet élément contient la configuration pour les commandes et pour le journal de l'historique des opérations.

Paramètres history

enabled

Indique si `history` est activé ou non. Type : booléen. Valeur par défaut : `true`.

file-name

Nom du fichier dans lequel l'historique doit être stocké. La valeur par défaut est `.jboss-cli-history`.

file-dir

Répertoire dans lequel l'historique doit être stocké. La valeur par défaut est `=$USER_HOME`

max-size

Taille maximum du fichier d'historique. La valeur par défaut : 500.

resolve-parameter-values

Indique si l'on doit résoudre les propriétés système qui sont spécifiées comme argument de commande (ou paramètres d'opérations) avant d'envoyer la demande d'opération au contrôleur ou laisser la résolution avoir lieu côté serveur. Type : Booléen. Valeur par défaut : `true`.

connection-timeout

Durée autorisée pour établir une connexion avec le contrôleur. Type : entier relatif. Valeur par défaut : 5000 secondes.

ssl

Cet élément contient la configuration pour les stores Trust et Key utilisés par SSL.



AVERTISSEMENT

Red Hat recommande que vous désactiviez SSL explicitement en faveur de TLSv1.1 ou TLSv1.2 dans tous les packages affectés.

Paramètres ssl

archivage sécurisé

Type : `vaultType`

key-store

Type : `string`.

key-store-password

Type : string.

alias

Type : string

key-password

Type : string

trust-store

Type : string.

trust-store-password

Type : string.

modify-trust-store

Si défini à `true`, le CLI invitera l'utilisateur quand des certificats non reconnus auront été reçus et les autorisera à les stocker dans le truststore. Type : Booléen. Valeur par défaut : `true`.

vaultType

Quand ni `code` ou `module` ne sont précisés, l'implémentation par défaut sera utilisée. Si le `code` est spécifié, mais pas le `module` il cherchera la classe spécifique dans le module de Picketbox. Si le `module` et le `code` sont spécifiés, il cherchera la classe spécifiée par le `code` dans le module spécifié par le 'module'.

code

Type : String.

module

Type : string

silent

Indique si les messages d'erreur et d'information doivent sortir dans le terminal. Même si la valeur `false` est indiquée, les messages continueront d'être enregistrés par le logger si sa configuration le permet et/ou si la cible de sortie est spécifiée dans la ligne de commande par `>`. La valeur par défaut est : `False`.

[Rapporter un bogue](#)

3.5.10. Références de commandes d'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Résumé

La section [Section 3.5.5, « Comment obtenir de l'aide par l'interface CLI »](#) décrit comment accéder aux fonctionnalités d'assistance de l'interface CLI. L'interface de gestion dispose d'une boîte de dialogue d'assistance avec des options générales et des options sensibles au contexte. Les commandes d'assistance dépendant du contexte de l'opération nécessitent une connexion à un contrôleur de domaine ou à un serveur autonome. Ces commandes ne seront pas affichées dans la liste, sauf si la connexion a été établie.

Tableau 3.3.

Commande	Description
batch	Démarrer le mode par lot en créant un nouveau lot ou, selon les lots existants retenus, réactiver l'un d'entre eux. Si il n'y a pas de lot existant retenu, cette commande, quand elle est invoquée sans argument, va commencer un nouveau lot . S'il y a un lot sans nom existant retenu, cette commande le réactivera. S'il y a des lots existants retenus, mais avec nom, il peuvent être activées en exécutant cette commande avec ce nom comme argument.
cd	Change le chemin du nœud en cours à l'argument. Le chemin du nœud en cours est utilisé comme adresse pour les requêtes opérationnelles qui ne contiennent pas la partie adresse. Si une opération n'inclut pas d'adresse, l'adresse incluse sera considérée comme relative au chemin du nœud en cours. Le chemin du nœud en cours peut finir en type de nœud. Dans ce cas, exécuter une opération en spécifiant un nom de nœud est suffisant, comme logging:read-resource.
clear	Efface l'écran
command	Vous permet d'ajouter, de supprimer ou de lister des commandes existantes de type standard. Une commande de type standard est une commande qui est assignée à un type de nœud spécifique et qui vous permet d'effectuer une opération disponible à une instance de ce type. Elle vous permet de modifier n'importe quelle propriété exposée par le type de n'importe quelle instance existante.
connect	Connecte le contrôleur à l'hôte et au port spécifiés.
connection-factory	Définit une usine de connexions
data-source	Gère les configurations de sources de données JDBC dans le sous-système de la source de données.
deploy	Déploie l'application désignée par le chemin d'accès au fichier ou bien, active une application qui est pré-existante, mais désactivée dans le référentiel. Si elle est exécutée sans argument, cette commande énumérera tous les déploiements existants.

Commande	Description
echo	<p>Disponible à partir de JBoss EAP 6.4, la commande echo produit le texte spécifié dans la console. Le texte est une sortie verbatim donc l'utilisation de variables n'est pas possible.</p> <p>Exemple :</p> <pre>echo Phase one complete</pre>
help	Affiche le message d'assistance. Peut être utilisé avec l'argument --commands pour fournir aux commandes données des résultats sensibles au contexte.
history	Affiche l'historique en mémoire de la commande CLI et affiche un statut pour savoir si l'expansion de l'historique est activée ou non. Peut être utilisé avec des arguments pour effacer, désactiver, ou activer l'expansion de l'historique selon les besoins.
jms-queue	Définit une file d'attente JMS dans le sous-système de messagerie.
jms-topic	Définit un topic dans le sous-système de messagerie.
ls	Lister les contenus du chemin d'accès au nœud. Par défaut, le résultat est imprimé dans des colonnes qui utilisent toute la largeur du terminal. Utiliser -l affichera les résultats sur la base d'un nom par ligne.
pwd	Affiche le chemin d'accès du nœud pour le nœud en cours
quit	Termine l'interface de ligne de commande.
read-attribute	Affiche la valeur et, suivant les arguments, la description de l'attribut d'une ressource gérée.
read-operation	Affiche la description d'une opération particulière, ou bien liste toutes les opérations si aucune n'est spécifiée.
undeploy	Annule une demande lorsque celle-ci est exécutée avec le nom de l'application prévue. Peut être exécuté avec des arguments pour supprimer également l'application du référentiel. Imprime la liste de tous les déploiements existants si exécutée sans application spécifiée.
version	Affiche la version de serveur d'applications et les informations d'environnement.
xa-data-source	Gère la configuration de la source de données JDBC XA du sous-système de la source de données.

3.5.11. Référence aux opérations d'interface CLI

Exposer les opérations d'interface CLI

Les opérations d'interface CLI peuvent être exposées par l'opération `read-operation-names` décrite dans la rubrique [Section 3.6.5, « Afficher les noms de l'opération en utilisant l'interface CLI »](#). Les descriptions des opérations peuvent être exposées par l'opération `read-operation-descriptions` décrite dans la rubrique [Section 3.6.4, « Affiche une description d'opération en utilisant l'interface CLI »](#).

Tableau 3.4. Les opérations d'interface CLI

Nom de l'opération	Description
<code>add-namespace</code>	Ajoute un mappage de préfixe d'espace-nom à la mappe d'attribut d'espace-nom.
<code>add-schema-location</code>	Ajoute un schéma de mappage d'emplacement à la mappe d'attribut schema-locations.
<code>delete-snapshot</code>	Efface un snapshot de la configuration serveur dans le répertoire de snapshots.
<code>full-replace-deployment</code>	Ajoute le contenu précédemment téléchargé de déploiement à la liste de contenu disponible, remplace le contenu existant du même nom dans le runtime et supprime le contenu remplacé dans la liste de contenu disponible. Voir lien pour plus de renseignements.
<code>list-snapshots</code>	Liste les snapshots de la configuration du serveur sauvegardée dans le répertoire des snapshots.
<code>read-attribute</code>	Affiche la valeur d'un attribut d'une ressource sélectionnée.
<code>read-children-names</code>	Affiche le nom de tous les enfants d'une ressource donnée ayant le type donné.
<code>read-children-resources</code>	Affiche des informations sur tous les enfants d'une ressource d'un type donné.
<code>read-children-types</code>	Affiche les noms de types de tous les enfants pour la ressource sélectionnée.
<code>read-config-as-xml</code>	Lit la configuration actuelle et l'affiche en format XML.
<code>read-operation-description</code>	Affiche les détails d'une opération de la ressource donnée.
<code>read-operation-names</code>	Affiche les noms de toutes les opérations de la ressource donnée.

Nom de l'opération	Description
read-resource	Affiche les valeurs des attributs d'un modèle de ressource avec des informations complètes ou de base sur n'importe quelle ressource enfant.
read-resource-description	Indique la description des attributs d'une ressource, les types de dépendants et les opérations.
reload	Charge le serveur à nouveau en fermant tous les services et en redémarrant.
remove-namespace	Supprime un mappage de préfixe d'espace-nom à la mappe d'attribut d'espace-nom.
remove-schema-location	Supprime un schéma de mappage d'emplacement à la mappe d'attribut schema-locations.
replace-deployment	Remplace le contenu existant du runtime par un contenu nouveau. Le nouveau contenu doit avoir été chargé auparavant dans le référentiel du contenu de déploiement.
resolve-expression	Opération qui accepte une expression comme entrée ou un string pouvant être compris comme une expression, et résolu en fonction du système local de propriétés et des variables d'environnement.
resolve-internet-address	Prend un ensemble de critères de résolution d'interface et trouve une adresse IP sur une machine locale qui correspond au critère, ou échoue si aucune adresse IP correspondante n'est trouvée.
server-set-restart-required	Met le serveur en mode «restart-required»
shutdown	Ferme le serveur via un appel à System.exit(0) .
start-servers	Démarre tous les serveurs configurés dans un domaine géré qui n'est pas actuellement en cours d'exécution.
stop-servers	Arrête tous les serveurs actuellement en cours d'exécution dans un domaine géré.
take-snapshot	Prend un snapshot de la configuration du serveur et la sauvegarde dans le répertoire des snapshots.
upload-deployment-bytes	Indique si le contenu de déploiement du tableau d'octets inclus doit être ajouté au référentiel du contenu de déploiement. Notez que cette opération n'indique pas que le contenu doive être déployé au runtime.

Nom de l'opération	Description
<code>upload-deployment-stream</code>	Indique si le contenu de déploiement disponible dans l'index des flux entrants doit être ajouté au référentiel du contenu de déploiement. Notez que cette opération n'indique pas que le contenu doive être déployé au runtime.
<code>upload-deployment-url</code>	Indique si le contenu de déploiement disponible dans l'URL doit être ajouté au référentiel du contenu de déploiement. Notez que cette opération n'indique pas que le contenu doive être déployé au runtime.
<code>validate-address</code>	Valide l'adresse de l'opération
<code>write-attribute</code>	Indique la valeur d'un attribut d'une ressource sélectionnée.

Rapporter un bogue

3.5.12. Substitution dans l'interface de commandes CLI

JBoss EAP 6 prend en charge l'utilisation des expressions de propriétés et d'éléments prédéfinis dans l'interface de commandes CLI. Ces expressions seront résolues à leurs valeurs définies lors de l'exécution de la commande.

Les propriétés suivantes peuvent être substituées par des expressions :

- la partie adresse d'opération de la demande d'opération (types de nodes et/ou les noms) ;
- nom d'opération ;
- noms de paramètres d'opérations ;
- noms d'en-têtes et valeurs ;
- noms de commandes ;
- noms d'arguments de commandes.

Par défaut, l'interface CLI effectue la substitution de propriétés à chaque ligne sauf pour les valeurs de paramètres ou d'arguments. Les valeurs de paramètres ou d'arguments sont résolues dans le serveur en cours d'exécution. Si vous avez besoin d'une substitution de propriété pour des valeurs de paramètre ou d'argument dans l'interface CLI client, et de lui faire envoyer les valeurs résolues au serveur, compléter la procédure suivante :

Procédure 3.11. Activer la substitution de propriété dans l'interface CLI

1. Ouvrir le fichier `EAP_HOME/bin/jboss-cli.xml`.
2. Trouver l'emplacement du paramètre `resolve-parameter-values` et changez-en la valeur à `true` (la valeur par défaut est `false`).

```
<!-- whether to resolve system properties specified as command
argument or operation parameter values in the Management CLI VM
before sending the operation requests to the controller -->
```



```
<resolve-parameter-values>true</resolve-parameter-values>
```

Cet élément n'affecte que les valeurs de paramètres de la demande d'opération et les valeurs d'arguments de commandes. Il n'a aucun effet sur le reste de la ligne de commande. Ce signifie que les propriétés système présentes dans la ligne de commande seront résolues lors du traitement de la ligne quelle que soit la valeur de l'élément *resolve-parameter-values*, sauf s'il si celui-ci correspond à une valeur de paramètre/argument.

Voir [Section 3.5.9, « Options de configuration de l'interface CLI »](#) pour obtenir d'autres options de configuration d'interface CLI.

Sachez que les valeurs de système utilisées dans les commandes d'interface CLI doivent être déjà définies. Vous devez inclure l'argument `--properties=/path/to/file.properties` ou bien, un ou plusieurs paramètres `-Dkey=VALUE`, quand vous commencez votre instance d'interface CLI. Le fichier de propriétés utilise une syntaxe standard `key=value`.

Les clés de propriétés son dénotées dans vos commandes d'interface CLI en utilisant la syntaxe `${MY_VAR}`.

Exemple 3.7. Exemple : utilisation de propriétés dans les commandes d'interface CLI

```
/subsystem=datasources/data-source=${datasourcename}:add(connection-  
url=jdbc:oracle:thin:@server:1521:ora1, jndi-name=java:/jboss/${name},  
driver-name=${drivername})
```

[Rapporter un bogue](#)

3.6. OPÉRATIONS DE L'INTERFACE CLI

3.6.1. Afficher les attributs d'une ressource par l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Résumé

L'opération `read-attribute` est une opération globale utilisée pour lire la valeur d'exécution d'un attribut sélectionné. Peut être utilisée pour exposer uniquement les valeurs qui ont été définies par l'utilisateur, en ignorant toute valeur par défaut ou non définie. Les propriétés de la requête incluent les paramètres suivants.

Propriétés de requêtes

`name`

Le nom de l'attribut pour obtenir la valeur sous la ressource sélectionnée.

`include-defaults`

Un paramètre booléen qui peut être défini à `false` pour limiter les résultats de l'opération aux attributs qui ont été définis par l'utilisateur uniquement, et ignorer les valeurs par défaut.

Procédure 3.12. Affiche la valeur de runtime en cours pour un attribut sélectionné.

- **Exécuter l'opération `read-attribute`**

À l'aide de l'interface CLI, utiliser l'opération `read-attribute` pour afficher la valeur d'un attribut de ressource. Pour plus d'informations sur les requêtes d'informations, voir le sujet [Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI »](#).

```
[standalone@localhost:9999 /]:read-attribute(name=name-of-attribute)
```

Un avantage de l'opération `read-attribute` est la possibilité d'exposer la valeur d'exécution actuelle d'un attribut spécifique. Des résultats similaires peuvent être obtenus avec l'opération `read-attribute`, mais seulement avec l'addition de la propriété de requête `include-runtime`, et uniquement dans le cadre d'une liste de toutes les ressources disponibles pour ce nœud. L'opération `read-attribute` est destinée aux requêtes d'attribut précises, comme le montre l'exemple suivant.

Exemple 3.8. Exécuter l'opération `read-attribute` pour exposer l'IP d'interface publique.

Si vous connaissez le nom de l'attribut que vous souhaitez exposer, vous pouvez utiliser la commande `read-attribute` pour renvoyer la valeur exacte dans le runtime en cours.

```
[standalone@localhost:9999 /] /interface=public:read-attribute(name=resolved-address)
{
  "outcome" => "success",
  "result" => "127.0.0.1"
}
```

L'attribut `resolved-address` est une valeur de runtime, donc ne s'affiche pas dans les résultats de l'opération `read-resource` standard.

```
[standalone@localhost:9999 /] /interface=public:read-resource
{
  "outcome" => "success",
  "result" => {
    "any" => undefined,
    "any-address" => undefined,
    "any-ipv4-address" => undefined,
    "any-ipv6-address" => undefined,
    "inet-address" => expression "${jboss.bind.address:127.0.0.1}",
    "link-local-address" => undefined,
    "loopback" => undefined,
    "loopback-address" => undefined,
    "multicast" => undefined,
    "name" => "public",
    "nic" => undefined,
    "nic-match" => undefined,
    "not" => undefined,
    "point-to-point" => undefined,
    "public-address" => undefined,
    "site-local-address" => undefined,
    "subnet-match" => undefined,
    "up" => undefined,
  }
}
```

```

    "virtual" => undefined
  }
}

```

Pour afficher **resolved-address** ou des autres valeurs de runtime, vous devrez utiliser la propriété de requête **include-runtime**.

```

[standalone@localhost:9999 /] /interface=public:read-resource(include-
runtime=true)
{
  "outcome" => "success",
  "result" => {
    "any" => undefined,
    "any-address" => undefined,
    "any-ipv4-address" => undefined,
    "any-ipv6-address" => undefined,
    "inet-address" => expression "${jboss.bind.address:127.0.0.1}",
    "link-local-address" => undefined,
    "loopback" => undefined,
    "loopback-address" => undefined,
    "multicast" => undefined,
    "name" => "public",
    "nic" => undefined,
    "nic-match" => undefined,
    "not" => undefined,
    "point-to-point" => undefined,
    "public-address" => undefined,
    "resolved-address" => "127.0.0.1",
    "site-local-address" => undefined,
    "subnet-match" => undefined,
    "up" => undefined,
    "virtual" => undefined
  }
}

```

Résultat

La valeur de l'attribut du runtime en cours est affichée.

[Rapporter un bogue](#)

3.6.2. Afficher l'utilisateur qui est actif dans l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Résumé

L'opération `whoami` est une opération globale utilisée pour identifier les attributs de l'utilisateur actif. L'opération expose l'identité de nom d'utilisateur et le domaine qui leur est attribué. L'opération `whoami` est utile pour les administrateurs qui gèrent plusieurs comptes d'utilisateurs dans plusieurs domaines, ou pour aider à assurer le suivi des utilisateurs actifs à travers les instances de domaine avec plusieurs sessions de terminal et les comptes utilisateurs.

Procédure 3.13. Affiche l'utilisateur qui est actif dans l'interface CLI par l'opération `whoami`

- **Exécuter l'opération `whoami`**

À partir de l'interface CLI, utiliser l'opération `whoami` pour afficher le compte utilisateur actif.

```
[standalone@localhost:9999 /] :whoami
```

L'exemple suivant utilise la commande `whoami` dans une instance de serveur autonome pour montrer que l'utilisateur actif est le `username`, et que l'utilisateur est assigné au domaine `ManagementRealm`.

Exemple 3.9. Utiliser la commande `whoami` dans une instance autonome

```
[standalone@localhost:9999 /]:whoami
{
  "outcome" => "success",
  "result" => {"identity" => {
    "username" => "username",
    "realm" => "ManagementRealm"
  }}
}
```

Résultat

Votre compte d'utilisateur actif en cours est affiché

[Rapporter un bogue](#)

3.6.3. Affiche les informations système et serveur dans l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 3.14. Affiche les informations système et serveur dans l'interface CLI

- **Exécuter la commande `version`**

À partir de l'interface CLI, saisir la commande `version` :

```
[domain@localhost:9999 /] version
```

Résultat

Les informations sur la version de votre serveur d'applications et sur votre environnement s'afficheront.

[Rapporter un bogue](#)

3.6.4. Affiche une description d'opération en utilisant l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 3.15. Exécuter la commande CLI suivante

- **Exécuter l'opération `read-operation-description`**

À partir de l'interface CLI, utiliser `read-operation-description` pour afficher des informations sur l'opération. L'opération requiert des paramètres supplémentaires dans le format d'une paire clé-valeur pour indiquer quelle opération afficher. Pour plus d'informations sur les requêtes d'informations, voir le sujet [Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI »](#).

```
[standalone@localhost:9999 /]:read-operation-description(name=name-of-operation)
```

Exemple 3.10. Afficher la description de l'opération «list-snapshots»

L'exemple suivant vous montre la méthode utilisée pour décrire l'opération `list-snapshots`.

```
[standalone@localhost:9999 /] :read-operation-description(name=list-snapshots)
{
  "outcome" => "success",
  "result" => {
    "operation-name" => "list-snapshots",
    "description" => "Lists the snapshots",
    "request-properties" => {},
    "reply-properties" => {
      "type" => OBJECT,
      "value-type" => {
        "directory" => {
          "type" => STRING,
          "description" => "The directory where the snapshots
are stored",
          "expressions-allowed" => false,
          "required" => true,
          "nillable" => false,
          "min-length" => 1L,
          "max-length" => 2147483647L
        },
        "names" => {
          "type" => LIST,
          "description" => "The names of the snapshots within
the snapshots directory",
          "expressions-allowed" => false,
          "required" => true,
          "nillable" => false,
          "value-type" => STRING
        }
      }
    },
    "access-constraints" => {"sensitive" => {"snapshots" => {"type"
=> "core"}}},
    "read-only" => false
  }
}
```

Résultat

La description est affichée pour l'opération choisie.

[Rapporter un bogue](#)

3.6.5. Afficher les noms de l'opération en utilisant l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 3.16. Exécuter la commande CLI suivante

- **Exécuter l'opération `read-operation-names`**
À partir de l'interface CLI, utiliser l'opération `read-operation-names` pour afficher les noms des opérations disponibles. Pour plus d'informations sur les requêtes d'informations, voir le sujet [Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI »](#).

```
[standalone@localhost:9999 /]:read-operation-names
```

Exemple 3.11. Afficher les noms de l'opération en utilisant l'interface CLI

L'exemple suivant vous montre la méthode utilisée pour décrire l'opération `read-operation-names`.

```
[standalone@localhost:9999 /]:read-operation-names
{
  "outcome" => "success",
  "result" => [
    "add-namespace",
    "add-schema-location",
    "delete-snapshot",
    "full-replace-deployment",
    "list-snapshots",
    "read-attribute",
    "read-children-names",
    "read-children-resources",
    "read-children-types",
    "read-config-as-xml",
    "read-operation-description",
    "read-operation-names",
    "read-resource",
    "read-resource-description",
    "reload",
    "remove-namespace",
    "remove-schema-location",
    "replace-deployment",
    "resolve-expression",
    "resolve-internet-address",
    "server-set-restart-required",
    "shutdown",
```

```

    "take-snapshot",
    "undefine-attribute",
    "upload-deployment-bytes",
    "upload-deployment-stream",
    "upload-deployment-url",
    "validate-address",
    "validate-operation",
    "whoami",
    "write-attribute"
  ]
}

```

Résultat

Les noms d'opérations disponibles sont affichés.

[Rapporter un bogue](#)

3.6.6. Afficher les ressources disponibles en utilisant l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Résumé

L'opération `read-resource` est une opération globale utilisée pour lire la valeur des ressources. Peut être utilisée pour exposer des informations complètes ou de base sur les ressources des nœuds en cours ou des nœuds enfants, ainsi qu'un ensemble de propriétés de requêtes qui peuvent étendre ou limiter l'étendue des résultats de l'opération. Les propriétés de la requête incluent les paramètres suivants.

Propriétés de requêtes

`recursive`

Pour savoir si on doit inclure récursivement des informations complètes sur les ressources enfant.

`recursive-depth`

La précision des informations de ressources enfant incluses

`proxies`

Si on doit inclure des ressources éloignées pour une recherche récursive. Par exemple, si on doit inclure les ressources niveau hôte à partir des contrôleurs hôtes esclave pour une demande de contrôleur de domaines.

`include-runtime`

Si on doit inclure des attributs de runtime dans la réponse, comme des valeurs d'attributs qui ne proviennent pas de la configuration persistante. Cette propriété de requête est définie sur `false` par défaut.

`include-defaults`

Une propriété de demande booléenne qui sert à activer ou à désactiver la lecture des attributs par

défaut. Si définie sur `false`, seuls les attributs définis par l'utilisateur seront renvoyés, ignorant ainsi ceux qui sont non définis.

Procédure 3.17. Exécuter la commande CLI suivante

1. Exécuter l'opération `read-resource`

Avec l'interface CLI, faites l'opération `read-resource` pour afficher les ressources disponibles.

```
[standalone@localhost:9999 /]:read-resource
```

L'exemple suivant vous montre comment il est possible d'utiliser l'opération `read-resource` dans une instance de serveur autonome pour exposer les informations de ressources générales. Les résultats ressemblent au fichier de configuration `standalone.xml`, qui affiche les ressources de système, les extensions, les interfaces et les sous-systèmes installés ou configurés pour l'instance du serveur. Ces résultats peuvent être interrogés directement.

Exemple 3.12. Exécuter l'opération `read-resource` niveau racine

```
[standalone@localhost:9999 /]:read-resource
{
  "outcome" => "success",
  "result" => {
    "management-major-version" => 1,
    "management-micro-version" => 0,
    "management-minor-version" => 7,
    "name" => "localhost",
    "namespaces" => [],
    "product-name" => "EAP",
    "product-version" => "6.4.0.GA",
    "profile-name" => undefined,
    "release-codename" => "Janus",
    "release-version" => "7.5.0.Final-redhat-17",
    "schema-locations" => [],
    "core-service" => {
      "service-container" => undefined,
      "server-environment" => undefined,
      "module-loading" => undefined,
      "platform-mbean" => undefined,
      "management" => undefined,
      "patching" => undefined
    },
    "deployment" => undefined,
    "deployment-overlay" => undefined,
    "extension" => {
      "org.jboss.as.clustering.infinispan" => undefined,
      "org.jboss.as.connector" => undefined,
      "org.jboss.as.deployment-scanner" => undefined,
      "org.jboss.as.ee" => undefined,
      "org.jboss.as.ejb3" => undefined,
      "org.jboss.as.jaxrs" => undefined,
      "org.jboss.as.jdr" => undefined,
      "org.jboss.as.jmx" => undefined,
      "org.jboss.as.jpas" => undefined,
```



```

    "org.jboss.as.jsf" => undefined,
    "org.jboss.as.logging" => undefined,
    "org.jboss.as.mail" => undefined,
    "org.jboss.as.naming" => undefined,
    "org.jboss.as.pojo" => undefined,
    "org.jboss.as.remoting" => undefined,
    "org.jboss.as.sar" => undefined,
    "org.jboss.as.security" => undefined,
    "org.jboss.as.threads" => undefined,
    "org.jboss.as.transactions" => undefined,
    "org.jboss.as.web" => undefined,
    "org.jboss.as.webservices" => undefined,
    "org.jboss.as.weld" => undefined
  },
  "interface" => {
    "management" => undefined,
    "public" => undefined,
    "unsecure" => undefined
  },
  "path" => {
    "jboss.server.temp.dir" => undefined,
    "user.home" => undefined,
    "jboss.server.base.dir" => undefined,
    "java.home" => undefined,
    "user.dir" => undefined,
    "jboss.server.data.dir" => undefined,
    "jboss.home.dir" => undefined,
    "jboss.server.log.dir" => undefined,
    "jboss.server.config.dir" => undefined,
    "jboss.controller.temp.dir" => undefined
  },
  "socket-binding-group" => {"standard-sockets" =>
undefined},
  "subsystem" => {
    "jaxrs" => undefined,
    "jsf" => undefined,
    "jca" => undefined,
    "jmx" => undefined,
    "threads" => undefined,
    "webservices" => undefined,
    "sar" => undefined,
    "remoting" => undefined,
    "infinispan" => undefined,
    "weld" => undefined,
    "ejb3" => undefined,
    "transactions" => undefined,
    "datasources" => undefined,
    "deployment-scanner" => undefined,
    "logging" => undefined,
    "jdr" => undefined,
    "pojo" => undefined,
    "jpa" => undefined,
    "naming" => undefined,
    "ee" => undefined,
    "mail" => undefined,
    "web" => undefined,

```

```

        "resource-adapters" => undefined,
        "security" => undefined
    },
    "system-property" => undefined
}
}

```

2. Exécuter l'opération `read-resource` pour un nœud enfant

L'opération `read-resource` peut être exécutée pour chercher les nœuds enfants à partir de la racine. La structure de l'opération commence par définir le nœud à exposer, puis s'ajoute à l'opération pour exécuter à ses côtés.

```
[standalone@localhost:9999 /]/subsystem=web/connector=http:read-resource
```

Dans l'exemple suivant, on peut exposer des informations de ressources spécifiques sur un composant de sous-système en redirigeant l'opération `read-resource` vers un nœud de sous-système web particulier.

Exemple 3.13. Exposer les ressources de nœud enfant à partir d'un nœud racine

```

[standalone@localhost:9999 /] /subsystem=web/connector=http:read-resource
{
    "outcome" => "success",
    "result" => {
        "configuration" => undefined,
        "enable-lookups" => false,
        "enabled" => true,
        "executor" => undefined,
        "max-connections" => undefined,
        "max-post-size" => 2097152,
        "max-save-post-size" => 4096,
        "name" => "http",
        "protocol" => "HTTP/1.1",
        "proxy-name" => undefined,
        "proxy-port" => undefined,
        "redirect-port" => 443,
        "scheme" => "http",
        "secure" => false,
        "socket-binding" => "http",
        "ssl" => undefined,
        "virtual-server" => undefined
    }
}

```

Les mêmes résultats sont possibles en utilisant la commande `cd` pour naviguer dans les nœuds enfants et en exécutant l'opération `read-resource` directement.

Exemple 3.14. Exposer les ressources de nœud enfant en changeant de répertoire

```
[standalone@localhost:9999 /] cd subsystem=web
```

```
[standalone@localhost:9999 subsystem=web] cd connector=http

[standalone@localhost:9999 connector=http] :read-resource
{
  "outcome" => "success",
  "result" => {
    "configuration" => undefined,
    "enable-lookups" => false,
    "enabled" => true,
    "executor" => undefined,
    "max-connections" => undefined,
    "max-post-size" => 2097152,
    "max-save-post-size" => 4096,
    "name" => "http",
    "protocol" => "HTTP/1.1",
    "proxy-name" => undefined,
    "proxy-port" => undefined,
    "redirect-port" => 443,
    "scheme" => "http",
    "secure" => false,
    "socket-binding" => "http",
    "ssl" => undefined,
    "virtual-server" => undefined
  }
}
```

3. Utiliser le paramètre récursif pour inclure des valeurs actives dans les résultats

Le paramètre récursif peut être utilisé pour exposer les valeurs de tous les attributs, y compris les valeurs non persistantes, celles qui sont passées au démarrage, ou les autres attributs normalement actifs du modèle d'exécution.

```
[standalone@localhost:9999 /]/interface=public:read-
resource(include-runtime=true)
```

Par rapport à l'exemple précédent, l'inclusion de la propriété de requête `include-runtime` expose des attributs actifs supplémentaires, comme des octets envoyés ou des octets reçus par le connecteur HTTP.

Exemple 3.15. Exposer des valeurs actives et supplémentaires par le paramètre `include-runtime`

```
[standalone@localhost:9999 /] /subsystem=web/connector=http:read-
resource(include-runtime=true)
{
  "outcome" => "success",
  "result" => {
    "any" => undefined,
    "any-address" => undefined,
    "any-ipv4-address" => undefined,
    "any-ipv6-address" => undefined,
    "inet-address" => expression
    "${jboss.bind.address:127.0.0.1}",
  }
}
```

```

"link-local-address" => undefined,
"loopback" => undefined,
"loopback-address" => undefined,
"multicast" => undefined,
"name" => "public",
"nic" => undefined,
"nic-match" => undefined,
"not" => undefined,
"point-to-point" => undefined,
"public-address" => undefined,
"resolved-address" => "127.0.0.1",
"site-local-address" => undefined,
"subnet-match" => undefined,
"up" => undefined,
"virtual" => undefined
}
}

```

[Rapporter un bogue](#)

3.6.7. Afficher les descriptions de ressources disponibles en utilisant l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 3.18. Exécuter la commande CLI suivante

1. Exécuter l'opération `read-resource-description`

À partir du CLI, utiliser l'opération `read-resource-description` pour lire et afficher les noms des ressources disponibles. Pour plus d'informations sur les requêtes d'opérations, voir le sujet [Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI »](#).

```
[standalone@localhost:9999 /]:read-resource-description
```

2. Utiliser les paramètres en option

L'opération `read-resource-description` autorise l'utilisation de paramètres supplémentaires.

- Utiliser le paramètre `operations` pour inclure les descriptions des opérations de la ressource.

```
[standalone@localhost:9999 /]:read-resource-
description(operations=true)
```

- Utiliser le paramètre `inherited` pour inclure ou pour exclure des descriptions des opérations héritées de ressource. L'état par défaut est `true`.

```
[standalone@localhost:9999 /]:read-resource-
description(inherited=false)
```

- Utiliser le paramètre `recursive` pour inclure les descriptions récursives des ressources

dépendantes.

```
[standalone@localhost:9999 /]:read-resource-
description(recursive=true)
```

- d. Utiliser le paramètre `locale` pour obtenir la description des ressources. Si « null », la locale régionale par défaut sera utilisée.

```
[standalone@localhost:9999 /]:read-resource-
description(locale=true)
```

Résultat

Les descriptions des ressources disponibles sont affichées.

[Rapporter un bogue](#)

3.6.8. Charger à nouveau le serveur d'applications à l'aide du CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Avec l'interface CLI, utiliser l'opération `reload` pour fermer tous les services et démarrer le runtime à nouveau. Une fois que la commande `reload` sera complétée, l'interface CLI se reconnectera automatiquement.

Pour obtenir plus d'informations sur les demandes d'opérations, voir [Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI »](#).

Exemple 3.16. Charger à nouveau le serveur d'applications

```
[standalone@localhost:9999 /]:reload
{"outcome" => "success"}
```

[Rapporter un bogue](#)

3.6.9. Fermer le serveur d'applications par l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 3.19. Fermer le serveur d'applications

- Exécuter l'opération `shutdown`
 - À partir du CLI, utiliser l'opération `shutdown` pour fermer le serveur via l'appel de système `System.exit(0)`. Pour plus d'informations sur les requêtes d'opérations, voir le sujet [Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI »](#).
 - En mode autonome, utiliser la commande suivante :

```
[standalone@localhost:9999 /]:shutdown
```

- En mode domaine, utiliser la commande suivante avec le nom d'hôte qui convient :

```
[domain@localhost:9999 /]/host=master:shutdown
```

- Pour vous connecter à une instance CLI détachée et pour fermer le serveur, exécuter la commande suivante :

```
jboss-cli.sh --connect command=:shutdown
```

- Pour vous connecter à une instance CLI éloignée et pour fermer le serveur, exécuter la commande suivante :

```
[disconnected /] connect IP_ADDRESS
[standalone@IP_ADDRESS:9999 /] :shutdown
```

Remplacer l'*IP_ADDRESS* par l'adresse IP de votre instance.



NOTE

Ajouter l'argument (**restart=true**) à la commande **:shutdown** (comme indiqué ci-dessous) entraînera le redémarrage du serveur.

```
[standalone@localhost:9999 /]:shutdown(restart=true)
```

Résultat

Le serveur d'applications se ferme. Le CLI sera déconnecté car le runtime n'est pas disponible.

[Rapporter un bogue](#)

3.6.10. Configurer un attribut à l'aide du CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Résumé

L'opération **write-attribute** est une opération globale utilisée pour écrire ou modifier un attribut de la ressource sélectionnée. Vous pouvez utiliser l'opération pour rendre les modifications persistantes ou pour modifier les paramètres de configuration de vos instances de serveur géré. Les propriétés de la requête incluent les paramètres suivants.

Propriétés de requêtes

name

Le nom de l'attribut pour définir la valeur sous la ressource sélectionnée.

value

La valeur désirée de l'attribut sous la ressource sélectionnée. Peut être « null » si le modèle sous-jacent supporte des valeurs « null ».

Procédure 3.20. Configurer un attribut de ressource par l'interface CLI

- **Exécuter l'opération `write-attribute`**

À partir de l'interface CLI, utiliser l'opération `write-attribute` pour modifier la valeur d'un attribut de ressource. L'opération peut être exécutée dans le nœud dépendant de la ressource, ou bien dans le nœud `root` du CLI où le chemin de ressource complet est spécifié.

Exemple 3.17. Désactiver le scanner de déploiement par l'opération `write-attribute`

L'exemple suivant utilise l'opération `write-attribute` pour désactiver le scanner de déploiement. L'opération est exécutée à partir d'un nœud `root`, en utilisant l'onglet de complétion de tâche pour pouvoir peupler le chemin de ressources qui convient.

```
[standalone@localhost:9999 /] /subsystem=deployment-
scanner/scanner=default:write-attribute(name=scan-enabled,value=false)
{"outcome" => "success"}
```

Le résultat de l'opération peut être confirmé directement par l'opération `read-attribute`.

```
[standalone@localhost:9999 /] /subsystem=deployment-
scanner/scanner=default:read-attribute(name=scan-enabled)
{
  "outcome" => "success",
  "result" => false
}
```

Les résultats peuvent également être confirmés en listant tous les attributs de ressources du nœud disponibles par l'opération `read-resource`. Dans l'exemple suivant, cette configuration particulière vous montre que l'attribut `scan-enabled` est maintenant défini à `false`.

```
[standalone@localhost:9999 /] /subsystem=deployment-
scanner/scanner=default:read-resource
{
  "outcome" => "success",
  "result" => {
    "auto-deploy-exploded" => false,
    "auto-deploy-xml" => true,
    "auto-deploy-zipped" => true,
    "deployment-timeout" => 600,
    "path" => "deployments",
    "relative-to" => "jboss.server.base.dir",
    "scan-enabled" => false,
    "scan-interval" => 5000
  }
}
```

Résultat

L'attribut de ressource est mis à jour.

3.6.11. Configurer les propriétés système par l'interface CLI

Procédure 3.21. Configurer les propriétés système par l'interface CLI

1. Démarrer le serveur JBoss EAP.
2. Lancer l'interface CLI par la commande pour votre système d'exploitation.

Dans Linux :

```
EAP_HOME/bin/jboss-cli.sh --connect
```

Dans Windows :

```
EAP_HOME\bin\jboss-cli.bat --connect
```

3. Ajouter une propriété système.

La commande que vous utilisez dépend de savoir si vous utilisez un serveur autonome ou un domaine géré. Si vous utilisez un domaine géré, vous pouvez ajouter des propriétés système à un ou à plusieurs serveurs exécutant sur ce domaine.

- o Ajouter une propriété système sur un serveur autonome en utilisant la syntaxe suivante :

```
/system-property=PROPERTY_NAME:add(value=PROPERTY_VALUE)
```

Exemple 3.18. Ajouter une propriété système sur un serveur autonome

```
[standalone@localhost:9999 /] /system-
property=property.mybean.queue:add(value=java:/queue/MyBeanQueu
e)
{"outcome" => "success"}
```

- o Ajouter une propriété système sur tous les hôtes et serveurs d'un domaine géré en utilisant la syntaxe suivante :

```
/system-property=PROPERTY_NAME:add(value=PROPERTY_VALUE)
```

Exemple 3.19. Ajouter une propriété système à tous les serveurs d'un domaine géré

```
[domain@localhost:9999 /] /system-
property=property.mybean.queue:add(value=java:/queue/MyBeanQueu
e)
{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => {"main-server-group" => {"host" =>
{"master" => {
  "server-one" => {"response" => {"outcome" =>
"success"}}},
```



```

        "server-two" => {"response" => {"outcome" =>
"success"}}}
    }}}}
}

```

- o Ajouter une propriété système à un hôte et à ses instances de serveur d'un domaine géré en utilisant la syntaxe suivante :

```

/system-property=PROPERTY_NAME:add(value=PROPERTY_VALUE)

```

Exemple 3.20. Ajouter une propriété système à un hôte et à ses serveurs dans un domaine

```

[domain@localhost:9999 /] /host=master/system-
property=property.mybean.queue:add(value=java:/queue/MyBeanQueu
e)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" =>
{"master" => {
    "server-one" => {"response" => {"outcome" =>
"success"}},
    "server-two" => {"response" => {"outcome" =>
"success"}}
    }}}}
}

```

- o Ajouter une propriété système à une instance de serveur d'un domaine géré en utilisant la syntaxe suivante :

```

/system-property=PROPERTY_NAME:add(value=PROPERTY_VALUE)

```

Exemple 3.21. Ajouter une propriété système à une instance de serveur d'un domaine géré

```

[domain@localhost:9999 /] /host=master/server-config=server-
one/system-
property=property.mybean.queue:add(value=java:/queue/MyBeanQueu
e)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" =>
{"master" => {"server-one" => {"response" => {"outcome" =>
"success"}}}}}}
}

```

4. Lire une propriété système.

La commande que vous utilisez dépend de savoir si vous exécutez sur un serveur autonome ou un domaine géré.

- o Lire une propriété système sur un serveur autonome en utilisant la syntaxe suivante :

```
/system-property=PROPERTY_NAME:read-resource
```

Exemple 3.22. Lire une propriété système sur un serveur autonome

```
[standalone@localhost:9999 /] /system-
property=property.mybean.queue:read-resource
{
    "outcome" => "success",
    "result" => {"value" => "java:/queue/MyBeanQueue"}
}
```

- o Lire une propriété système sur tous les hôtes et serveurs d'un domaine géré en utilisant la syntaxe suivante :

```
/system-property=PROPERTY_NAME:read-resource
```

Exemple 3.23. Lire une propriété système de tous les serveurs dans un domaine géré

```
[domain@localhost:9999 /] /system-
property=property.mybean.queue:read-resource
{
    "outcome" => "success",
    "result" => {
        "boot-time" => true,
        "value" => "java:/queue/MyBeanQueue"
    }
}
```

- o Lire une propriété système d'un hôte et de ses instances de serveur d'un domaine géré en utilisant la syntaxe suivante :

```
/host=master/system-property=PROPERTY_NAME:read-resource
```

Exemple 3.24. Lire une propriété système d'un hôte et de ses serveurs dans un domaine

```
[domain@localhost:9999 /] /host=master/system-
property=property.mybean.queue:read-resource
{
    "outcome" => "success",
    "result" => {
        "boot-time" => true,

```

```

        "value" => "java:/queue/MyBeanQueue"
    }
}

```

- o Lire une propriété système d'une instance de serveur d'un domaine géré en utilisant la syntaxe suivante :

```

/host=master/server-config=server-one/system-
property=PROPERTY_NAME:read-resource

```

Exemple 3.25. Lire une propriété système d'une instance de serveur dans un domaine géré

```

[domain@localhost:9999 /] /host=master/server-config=server-
one/system-property=property.mybean.queue:read-resource
{
    "outcome" => "success",
    "result" => {
        "boot-time" => true,
        "value" => "java:/queue/MyBeanQueue"
    }
}

```

5. Supprimer une propriété système.

La commande que vous utilisez dépend de savoir si vous exécutez sur un serveur autonome ou un domaine géré.

- o Supprimer une propriété système sur un serveur autonome en utilisant la syntaxe suivante :

```

/system-property=PROPERTY_NAME:remove

```

Exemple 3.26. Supprimer une propriété système sur un serveur autonome

```

[standalone@localhost:9999 /] /system-
property=property.mybean.queue:remove
{"outcome" => "success"}

```

- o Supprimer une propriété système de tous les hôtes et serveurs d'un domaine géré en utilisant la syntaxe suivante :

```

/system-property=PROPERTY_NAME:remove

```

Exemple 3.27. Supprimer une propriété système de tous les hôtes et de ses serveurs dans un domaine.

```

[domain@localhost:9999 /] /system-
property=property.mybean.queue:remove

```

```

{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => {"main-server-group" => {"host" =>
{"master" => {
  "server-one" => {"response" => {"outcome" =>
"success"}}},
  "server-two" => {"response" => {"outcome" =>
"success"}}
}}}}
}

```

- Supprimer une propriété système d'un hôte et de ses instances de serveur dans un domaine géré en utilisant la syntaxe suivante :

```
/host=master/system-property=PROPERTY_NAME:read-resource
```

Exemple 3.28. Supprimer une propriété système d'un hôte et de ses serveurs dans un domaine

```

[domain@localhost:9999 /] /host=master/system-
property=property.mybean.queue:remove
{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => {"main-server-group" => {"host" =>
{"master" => {
  "server-one" => {"response" => {"outcome" =>
"success"}}},
  "server-two" => {"response" => {"outcome" =>
"success"}}
}}}}
}

```

- Supprimer une propriété système d'une instance de serveur d'un domaine géré en utilisant la syntaxe suivante :

```
/host=master/server-config=server-one/system-
property=PROPERTY_NAME:remove
```

Exemple 3.29. Supprimer une propriété système d'un serveur dans un domaine géré

```

[domain@localhost:9999 /] /host=master/server-config=server-
one/system-property=property.mybean.queue:remove
{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => {"main-server-group" => {"host" =>
{"master" => {"server-one" => {"response" => {"outcome" =>
"success"}}}}}}
}

```

[Rapporter un bogue](#)

3.7. HISTORIQUE DES COMMANDES DE L'INTERFACE CLI

3.7.1. Utiliser l'historique de commandes à l'aide de l'interface CLI.

L'interface CLI contient une fonctionnalité d'historique de commandes qui est activée par défaut dans l'installation du serveur d'applications. L'historique est conservé en tant qu'archive dans la mémoire volatile de la session CLI active, et est ajouté le fichier journal qui sauvegarde automatiquement dans le répertoire d'accueil de l'utilisateur sous le nom `.jboss-cli-history`. Le fichier de l'historique est configuré par défaut pour enregistrer un maximum de 500 commandes CLI.

La commande `history` elle-même renverra l'historique de la session en cours, ou si accompagnée d'arguments, elle pourra désactiver, activer ou supprimer l'historique de la mémoire de session. L'interface CLI vous donne également la possibilité d'utiliser les flèches de votre clavier pour naviguer dans l'historique des commandes et des opérations.

Fonctions de l'historique de l'interface CLI

- [Section 3.7.2, « Afficher l'historique de commandes par interface CLI. »](#)
- [Section 3.7.3, « Effacer l'historique de commandes par interface CLI. »](#)
- [Section 3.7.4, « Désactiver l'historique de commandes par l'interface CLI. »](#)
- [Section 3.7.5, « Activer l'historique des commandes de l'interface CLI »](#)

[Rapporter un bogue](#)

3.7.2. Afficher l'historique de commandes par interface CLI.

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 3.22. Afficher l'historique de commandes par l'interface CLI.

- **Exécuter la commande `history`**
À partir de l'interface CLI, saisir la commande `history` :

```
[standalone@localhost:9999 /] history
```

Résultat

L'historique de la commande CLI stocké en mémoire depuis le démarrage du CLI ou la commande de suppression de l'historique est affiché.

[Rapporter un bogue](#)

3.7.3. Effacer l'historique de commandes par interface CLI.

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 3.23. Effacer l'historique de commandes par interface CLI.

- Exécuter la commande `history --clear`
À partir de l'interface CLI, saisir la commande `history --clear` :

```
[standalone@localhost:9999 /] history --clear
```

Résultat

L'historique des commandes enregistré depuis le démarrage du CLI est supprimé de la mémoire de session. L'historique de la commande est toujours présent dans le fichier `.jboss-cli-history` qui est sauvegardé dans le répertoire d'accueil de l'utilisateur.

[Rapporter un bogue](#)

3.7.4. Désactiver l'historique de commandes par l'interface CLI.

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 3.24. Désactiver l'historique de commandes par l'interface CLI.

- Exécuter la commande `history --disable`
À partir de l'interface CLI, saisir la commande `history --disable` :

```
[standalone@localhost:9999 /] history --disable
```

Résultat

Les commandes passées dans le CLI ne seront pas enregistrées dans la mémoire ou dans un fichier `.jboss-cli-history` sauvegardé dans le répertoire d'accueil de l'utilisateur.

[Rapporter un bogue](#)

3.7.5. Activer l'historique des commandes de l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Procédure 3.25. Activer l'historique des commandes de l'interface CLI

- Exécuter la commande `history --enable`
À partir de l'interface CLI, saisir la commande `history --enable` :

```
[standalone@localhost:9999 /] history --enable
```

Résultat

Les commandes passées dans le CLI seront enregistrées dans la mémoire ou dans un fichier `.jboss-cli-history` sauvegardé dans le répertoire d'accueil de l'utilisateur.

[Rapporter un bogue](#)

3.8. LA JOURNALISATION D'AUDITING DE L'INTERFACE DE GESTION

3.8.1. La journalisation d'auditing de l'interface de gestion

Quand la journalisation de l'auditing est activée, toutes les opérations effectuées par la console de gestion, l'interface CLI, ou l'application de gestion personnalisée manuellement sont assujetties à la journalisation d'auditing.

Les entrées de journalisation d'auditing sont stockées en format JSON et, basées sur votre configuration, elles peuvent être stockées dans des fichiers, elles peuvent être envoyées dans un serveur syslog ou les deux à la fois. Les journalisation d'auditing ne peut être configurée qu'à l'aide de l'interface CLI et est désactivée par défaut.

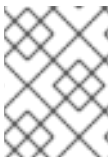
Les événements de connexion et de déconnexion ne peuvent pas être audités car il n'y a pas de session d'authentification dans EAP. Au lieu de cela, les messages d'auditing sont journalisés quand l'utilisateur reçoit une opération.



NOTE

Par défaut, la journalisation d'auditing n'est pas active. La journalisation d'auditing ne peut être configurée que par l'interface CLI.

Pour lister toutes les options disponibles de configuration de journalisation d'auditing de l'interface de gestion et leurs valeurs courantes, saisir la commande de ligne suivante :



NOTE

Ajouter le préfixe `/host=HOST_NAME` à la commande si vous devez appliquer les changements à un domaine géré.

```
[... /] /core-service=management/access=audit:read-resource(recursive=true)
```

[Rapporter un bogue](#)

3.8.2. Activer la Journalisation d'auditing de l'interface de gestion par l'intermédiaire d'un fichier.

Pour activer les sorties de journalisation dans un fichier, saisir la commande d'interface CLI suivante.



NOTE

Si la modification s'applique à un domaine géré, ajouter le préfixe `/host=HOST_NAME` à la commande suivante.

```
/core-service=management/access=audit/logger=audit-log:write-attribute(name=enabled,value=true)
```

Les opérations de gestion sont maintenant journalisées dans un fichier :

- En mode autonome : `EAP_HOME/standalone/data/audit-log.log`
- En mode de domaine : `EAP_HOME/domain/data/audit-log.log`

Pour obtenir des informations sur les attributs du gestionnaire de tous les fichiers, consulter [Section A.3, « Référence de Journalisation d'auditing de l'interface de gestion »](#).

[Rapporter un bogue](#)

3.8.3. Activer la journalisation d'auditing de l'interface de gestion par le serveur syslog.

Par défaut, la journalisation d'auditing est préconfigurée pour pouvoir être déversée dans un fichier une fois activée. Cette procédure configure la sortie dans un serveur syslog et active la journalisation d'auditing dans un fichier. Pour obtenir plus d'informations sur les attributs de gestionnaire syslog, consulter [Section A.3, « Référence de Journalisation d'auditing de l'interface de gestion »](#).



NOTE

Si la modification s'applique à un domaine géré, ajouter le préfixe `/host=HOST_NAME` à la commande `/core-service`.

Procédure 3.26. Activer la journalisation d'auditing sur un serveur syslog

1. Activer la journalisation d'auditing

Exécutez la commande suivante :

```
[.. /]/core-service=management/access=audit/logger=audit-log:write-attribute(name=enabled,value=true)
```

2. Créer un gestionnaire syslog

Dans cet exemple, le serveur `syslog` exécute sur le même serveur en tant qu'instance JBoss EAP, sur le port 514. Remplacer les valeurs de l'attribut `host` par des valeurs appropriées à votre environnement.

Exemple 3.30. Exemple de gestionnaire syslog

```
[.. /]batch
[.. / #]/core-service=management/access=audit/syslog-handler=mysyslog:add(formatter=json-formatter)
[.. / #]/core-service=management/access=audit/syslog-handler=mysyslog/protocol=udp:add(host=localhost,port=514)
[.. /]run-batch
```

3. Ajouter une référence au gestionnaire syslog

Exécuter ce qui suit :


```
[.. /]/core-service=management/access=audit/logger=audit-  
log/handler=mysyslog:add
```

Résultat

Les entrées de journalisation de l'auditing de l'interface de gestion sont alors enregistrées sur le serveur `syslog`.

[Rapporter un bogue](#)

3.8.4. Lire une Journalisation d'auditing de l'interface de gestion

Les sorties de journalisation d'auditing en fichier(s) peuvent être lues grâce à *visionneur*, tandis que les mêmes sorties dans un serveur `syslog` devront être visionnées à l'aide d'une application de visionneur de `syslog`.



NOTE

Utiliser un *éditeur* de texte pour visionner les fichiers de journalisation n'est pas conseillé car il est possible que cela empêche la journalisation ultérieure.

Les journaux d'auditing de l'interface de gestion sont présentés sous un format JSON. Chaque entrée de journalisation commence par une date de dernière modification suivie des champs listés dans le tableau *Management Interface Audit Log Fields*

Tableau 3.5. Champs de journalisation d'auditing de l'interface de gestion

Nom de champ	Description
<code>type</code>	Peut avoir pour valeur core , indiquant ainsi qu'il s'agit d'une opération de gestion, ou jmx indiquant une provenance de sous-système JMX (voir le sous-système JMX pour la configuration de la journalisation de l'auditing du sous-système JMX).
<code>r/o</code>	Sur true si l'opération ne modifie pas le modèle de gestion, sinon false .
<code>booting</code>	Sur true si l'opération a été exécutée à l'amorçage, false si elle a été exécutée une fois que le serveur était en route.
<code>version</code>	Le numéro de version de l'instance JBoss EAP.
<code>user</code>	Le nom d'utilisateur de l'utilisateur authentifié. Si l'opération a été journalisée par l'interface CLI sur le même ordinateur que le serveur en cours d'exécution, l'utilisateur spécial \$local sera utilisé.
<code>domainUUID</code>	Un identifiant pour relier toutes les opérations tandis qu'elles sont propagées du contrôleur de domaines vers ses serveurs, ses contrôleurs hôtes, et ses serveurs de contrôleurs hôtes esclaves.

Nom de champ	Description
access	Peut avoir une des valeurs suivantes : <ul style="list-style-type: none">• NATIVE - L'opération est venue par le biais de l'interface de gestion native, comme l'interface CLI.• HTTP - L'opération est venue par le biais de l'interface HTTP de domaine, comme la console de gestion.• JMX - L'opération est venue par le biais du sous-système JMX. Voir JMX pour voir la façon de configurer la journalisation d'auditing de JMX.
remote-address	L'adresse du client qui exécute l'opération.
success	Sur true si l'opération a réussi, false si non.
ops	Les opérations en cours. Liste des opérations sérialisées dans JSON. Au démarrage, cela correspond à toutes les opérations résultant du traitement XML. Une fois démarré, la liste contient normalement une seule entrée.

[Rapporter un bogue](#)

CHAPITRE 4. GESTION DES UTILISATEURS

4.1. CRÉATION D'UTILISATEUR

4.1.1. Ajouter un utilisateur pour les interfaces de gestion

Aperçu

Les instances de gestion de JBoss EAP 6 sont sécurisés par défaut car il n'y a aucun compte d'utilisateur disponible au départ, (sauf si vous avez installé la plateforme à l'aide de l'installateur graphique.) Il s'agit d'une mesure de précaution visant à éviter les failles de sécurité qui peuvent découler de simples erreurs de configuration.

La communication HTTP avec la plate-forme JBoss EAP 6 est considérée « communication à distance », même si le trafic prend sa source sur l'hôte local. Par conséquent, vous devez créer au moins un utilisateur afin de pouvoir utiliser la console de gestion. Si vous essayiez d'accéder à la console de gestion avant d'ajouter un utilisateur, vous recevez une erreur parce qu'il n'y a pas de déploiement tant que l'utilisateur n'a pas été créé.

Suivez ces étapes pour créer l'utilisateur d'administration d'origine, qui peut utiliser la console de gestion basée web et les instances éloignées de l'interface CLI pour configurer et administrer la plateforme JBoss EAP 6 à partir de systèmes distants.

Procédure 4.1. Créer l'utilisateur administratif d'origine pour les interfaces de gestion distantes

1. **Exécuter le script `add-user.sh` ou `add-user.bat`.**

Passez au répertoire `EAP_HOME/bin/`. Invoquer le script qui convient à votre système d'exploitation.

Red Hat Enterprise Linux

```
[user@host bin]$ ./add-user.sh
```

Microsoft Windows Server

```
C:\bin> add-user.bat
```

2. **Choisissez d'utiliser un utilisateur Management.**

Appuyer sur **ENTER** pour sélectionner l'option **a** pour ajouter un utilisateur Management.

Cet utilisateur sera ajouté au domaine `ManagementRealm` et il sera autorisé à effectuer des opérations de gestion par la Management Console basée web ou par l'interface CLI basé sur la ligne de commande. Autre alternative, **b**, ajoutera l'utilisateur au domaine `ApplicationRealm`, et ne fournit aucune permission particulière. Ce domaine est fourni pour être utilisé avec des applications.

3. **Saisir le nom d'utilisateur et le mot de passe que vous souhaitez.**

Quand on vous y invite, saisir le nom d'utilisateur et le nom de passe. On vous demandera de saisir le mot de passe une seconde fois pour confirmer.

4. **Saisissez les informations sur votre groupe.**

Ajouter le groupe ou les groupes auxquels l'utilisateur appartient. Si l'utilisateur appartient à plusieurs groupes, saisir une liste séparée par des virgules. Laisser vide si vous ne souhaitez pas que l'utilisateur appartienne à un groupe.

5. Vérifier les informations et confirmer.

On vous invitera à confirmer les informations. Quand vous serez satisfait, saisir **yes**.

6. Décidez si l'utilisateur représente une instance de serveur de JBoss EAP 6 à distance.

En plus des administrateurs, un autre type d'utilisateur qui a parfois besoin d'être ajouté à JBoss EAP 6 dans le domaine `ManagementRealm` est un utilisateur qui représente une autre instance de JBoss EAP 6, et qui a besoin d'être authentifié pour rejoindre un groupement en tant que membre. L'invite suivante vous permet de désigner votre utilisateur supplémentaire dans ce but. Si vous sélectionnez **yes**, on vous donnera une valeur `secret` de hachage, qui représentera le mot de passe de l'utilisateur, que vous aurez besoin d'ajouter dans un fichier de configuration différent. Dans le but de cette tâche, répondre **no** à cette question.

7. Saisir des utilisateurs supplémentaires.

Vous pouvez saisir des utilisateurs supplémentaires si vous le souhaitez, en répétant la procédure. Vous pouvez également les ajouter à tout moment sur un système en cours d'exécution. Au lieu de choisir le domaine de sécurité par défaut, vous pouvez ajouter des utilisateurs d'autres domaines afin d'ajuster leurs autorisations.

8. Créer des utilisateurs en mode non interactif.

Vous pouvez créer des utilisateurs en mode non interactif, en l'indiquant dans chaque paramètre de ligne de commande. Cette approche n'est pas recommandée sur les systèmes partagés, parce que les mots de passe seront visibles dans les fichiers de journalisation (log) et dans les fichiers d'historique. La syntaxe de la commande, pour le domaine de gestion, est la suivante :

```
[user@host bin]$ ./add-user.sh username password
```

Pour utiliser le domaine d'application, utiliser le paramètre `-a`.

```
[user@host bin]$ ./add-user.sh -a username password
```

9. Vous pouvez supprimer la sortie normale du script d'ajout d'utilisateur en passant le paramètre `--silent`. Cela s'applique uniquement si un minimum de paramètres, **nom d'utilisateur** et **mot de passe**, ont été indiqués. Le message d'erreur apparaîtra toujours.

Résultat

Tout utilisateur que vous ajoutez est activé dans les domaines de sécurité que vous avez indiqués. Les utilisateurs actifs dans le domaine `ManagementRealm` sont en mesure de gérer la plateforme JBoss EAP 6 à partir de systèmes éloignés.

[Rapporter un bogue](#)

4.1.2. Passer des arguments au script `add-user` de la gestion utilisateur

Vous pouvez exécuter la commande `add-user .sh` ou `add-user .bat` interactivement ou vous pouvez passer des arguments par la ligne de commande. Cette section décrit les options qui se présentent pour passer des arguments en ligne de commande au script `add-user`.

Pour obtenir une liste d'arguments en ligne de commandes disponibles pour `add-user .sh` ou `add-user .bat`, consulter [Section 4.1.3, « Arguments pour la commande Add-user »](#) .

Pour plus d'informations sur la façon d'indiquer un autre fichier de propriétés et son emplacement, consulter [Section 4.1.4, « Spécifier des fichiers de propriétés alternatifs pour les informations de gestion des utilisateurs »](#).

Pour obtenir des exemples qui montrent comment passer des arguments sur la commande `add-user . sh` ou `add-user . bat`, consulter [Section 4.1.5, « Exemples de lignes de commande de script Add-user »](#).

[Rapporter un bogue](#)

4.1.3. Arguments pour la commande Add-user

Le tableau suivant décrit les arguments disponibles pour la commande `add-user . sh` ou `add-user . bat`.

Tableau 4.1. Arguments pour la commande Add-user

Argument de ligne de commande	Valeur d'argument	Description
-a	S/O	Cet argument demande de créer un utilisateur dans le domaine de l'application. S'il est omis, un utilisateur sera créé par défaut dans le domaine de gestion.
-dc	<i>DOMAIN_CONFIGURATION_DIRECTORY</i>	Cet argument spécifie le répertoire de configuration de domaine qui contient les fichiers de propriétés. S'il est omis, le répertoire par défaut sera <i>EAP_HOME/domain/configuration/</i> .
-sc	<i>SERVER_CONFIGURATION_DIRECTORY</i>	Cet argument spécifie un répertoire de configuration de serveur autonome différent qui contient les fichiers de propriétés. S'il est omis, le répertoire par défaut sera <i>EAP_HOME/standalone/configuration/</i> .
-up --user-properties	<i>USER_PROPERTIES_FILE</i>	Cet argument spécifie le nom d'un autre fichier de propriétés utilisateur. Il peut correspondre à un chemin absolu ou il peut correspondre à un nom de fichier utilisé en conjonction avec l'argument <code>-sc</code> ou <code>-dc</code> qui spécifie le répertoire de configuration alternatif.
-g --group	<i>GROUP_LIST</i>	Une liste séparée par des virgules de groupes à assigner à cet utilisateur.
-gp --group-properties	<i>GROUP_PROPERTIES_FILE</i>	Cet argument spécifie le nom d'un autre fichier de propriétés de groupe. Il peut correspondre à un chemin absolu ou il peut correspondre à un nom de fichier utilisé en conjonction avec l'argument <code>-sc</code> ou <code>-dc</code> qui spécifie le répertoire de configuration alternatif.

Argument de ligne de commande	Valeur d'argument	Description
-p --password	<i>PASSWORD</i>	Le mot de passe utilisateur. Le mot de passe doit remplir les critères suivants : <ul style="list-style-type: none"> • Il doit contenir 8 caractères au moins. • Il doit contenir au moins un caractère de l'alphabet. • Il doit contenir un chiffre au moins. • Il doit contenir au moins un symbole non alphanumérique
-u --user	<i>USER_NAME</i>	Le nom de l'utilisateur. Seuls les caractères alphanumériques et les symboles suivants sont valides : ,./=@\.
-r --realm	<i>REALM_NAME</i>	Le nom du domaine utilisé pour sécuriser les interfaces de gestion. S'il est omis, la valeur par défaut sera ManagementRealm .
-s --silent	S/O	Exécuter le script add-user sans sortie vers la console.
-h --help	S/O	Afficher les informations d'utilisation du script add-user.

[Rapporter un bogue](#)

4.1.4. Spécifier des fichiers de propriétés alternatifs pour les informations de gestion des utilisateurs

Aperçu

Par défaut, les informations utilisateurs et rôles créés à l'aide du script `add-user .sh` ou `Add-user .bat` sont stockées dans des fichiers de propriétés situés dans le répertoire de configuration de serveur. Les informations de configuration du serveur sont stockées dans le répertoire `EAP_HOME/standalone/configuration/` et les informations de configuration de domaine sont stockées dans le répertoire `EAP_HOME/domaine/configuration/`. Cette rubrique décrit comment substituer les noms de fichier et emplacements par défaut.

Procédure 4.2. Spécifier des fichiers de propriétés alternatifs

- ○ Pour spécifier un autre répertoire pour la configuration du serveur, utilisez l'argument `-sc`. Cet argument spécifie un autre répertoire qui contiendra les fichiers de propriétés de configuration de serveur.

- Pour spécifier un répertoire alternatif de configuration de domaine, utiliser l'argument `-dc`. Cet argument spécifie un répertoire alternatif qui contient les fichiers de propriétés de configuration de domaines.
- Pour spécifier un fichier de configuration utilisateur différent, utiliser l'argument `-up` ou `-user-properties`. Peut correspondre à un chemin complet ou à un nom de fichier utilisé en conjonction avec `-sc` ou `-dc` spécifiant le répertoire de configuration alternatif.
- Pour spécifier un fichier de configuration de groupe différent, utiliser l'argument `-up` ou `-group-properties`. Peut correspondre à un chemin complet ou à un nom de fichier utilisé en conjonction avec `-sc` ou `-dc` indiquant le répertoire de configuration alternatif.



NOTE

La commande `add-user` a pour but d'opérer sur des fichiers de propriétés existants. Tout fichier de propriété alternatif spécifié dans un argument de ligne de commande devra sortir là où vous verrez l'erreur suivante :

```
JBAS015234: No appusers.properties files found
```

Pour obtenir plus d'informations sur les arguments de commande, consulter [Section 4.1.3, « Arguments pour la commande Add-user »](#).

Pour obtenir plus d'exemples sur les commandes `add-user`, consulter [Section 4.1.5, « Exemples de lignes de commande de script Add-user »](#).

[Rapporter un bogue](#)

4.1.5. Exemples de lignes de commande de script Add-user

Les exemples suivants montrent comment passer des arguments à la commande `add-user .sh` ou `add-user .bat`. À moins que cela soit notifié, ces commandes supposent la configuration d'un serveur autonome.

Exemple 4.1. Créer un utilisateur qui appartienne à un groupe unique en utilisant les fichiers de propriétés par défaut.

```
EAP_HOME/bin/add-user.sh -a -u 'appuser1' -p 'password1!' -g 'guest'
```

La commande ci-dessus produit les résultats suivants.

- L'utilisateur `appuser1` est ajouté aux fichiers de propriétés suivants par défaut qui contiennent les informations utilisateur.

```
EAP_HOME/standalone/configuration/application-users.properties
```

```
EAP_HOME/domain/configuration/application-users.properties
```

- L'utilisateur `appuser1` ayant pour groupe `guest` est ajouté aux fichiers de propriétés suivants par défaut qui contiennent les informations utilisateur.

```
EAP_HOME/standalone/configuration/application-roles.properties
```

```
EAP_HOME/domain/configuration/application-roles.properties
```

Exemple 4.2. Créer un utilisateur qui appartienne à plusieurs groupes en utilisant les fichiers de propriétés par défaut.

```
EAP_HOME/bin/add-user.sh -a -u 'appuser1' -p 'password1!' -g  
'guest, app1group, app2group'
```

La commande ci-dessus produit les résultats suivants.

- L'utilisateur **appuser1** est ajouté aux fichiers de propriétés suivants par défaut qui contiennent les informations utilisateur.

```
EAP_HOME/standalone/configuration/application-users.properties
```

```
EAP_HOME/domain/configuration/application-users.properties
```

- L'utilisateur **appuser1** ayant pour groupes **guest**, **app1group**, et **app2group** est ajouté aux fichiers de propriétés suivantes par défaut qui contiennent les informations utilisateur.

```
EAP_HOME/standalone/configuration/application-roles.properties
```

```
EAP_HOME/domain/configuration/application-roles.properties
```

Exemple 4.3. Créer un utilisateur ayant des privilèges admin dans le domaine par défaut en utilisant les fichiers de propriétés par défaut.

```
EAP_HOME/bin/add-user.sh -u 'adminuser1' -p 'password1!' -g 'admin'
```

La commande ci-dessus produit les résultats suivants.

- L'utilisateur **adminuser1** est ajouté aux fichiers de propriétés suivantes par défaut qui contiennent les informations utilisateur.

```
EAP_HOME/standalone/configuration/mgmt-users.properties
```

```
EAP_HOME/domain/configuration/mgmt-users.properties
```

- L'utilisateur **adminuser1** ayant pour groupe **admin** est ajouté aux fichiers de propriétés suivants par défaut qui contiennent les informations utilisateur.

```
EAP_HOME/standalone/configuration/mgmt-groups.properties
```

```
EAP_HOME/domain/configuration/mgmt-groups.properties
```

Exemple 4.4. Créer un utilisateur qui appartienne à un groupe unique en utilisant des fichiers de propriétés alternatifs pour stocker des informations.

```
EAP_HOME/bin/add-user.sh -a -u appuser1 -p password1! -g app1group -sc  
/home/someusername/userconfigs/ -up appusers.properties -gp  
appgroups.properties
```


La commande ci-dessus produit les résultats suivants.

- L'utilisateur **appuser1** est ajouté aux fichiers de propriétés suivants et ce fichier est maintenant le fichier par défaut qui contient les informations utilisateur.

/home/someusername/userconfigs/appusers.properties

- L'utilisateur **appuser1** ayant pour groupe **app1group** est ajouté aux fichiers de propriétés suivants et ce fichier est maintenant le fichier par défaut qui contient les informations utilisateur.

/home/someusername/userconfigs/appgroups.properties

[Rapporter un bogue](#)

CHAPITRE 5. RÉSEAU ET CONFIGURATION DE PORT

5.1. INTERFACES

5.1.1. Les interfaces

Le serveur d'applications utilise des références d'interfaces nommées dans la configuration. Cela permet à la configuration de référencer les déclarations d'interfaces individuelles avec des noms logiques, au lieu de référencer toutes les informations sur l'interface à chaque utilisation. L'utilisation de noms logiques permet également de conserver une homogénéité des références de groupe pour les interfaces nommées, quand les instances de serveur d'un domaine géré peuvent contenir des informations d'interface diverses à travers les machines. En utilisant cette méthodologie, chaque instance de serveur peut correspondre à un groupe de nom logique, ce qui facilite l'administration du groupe d'interfaces dans son ensemble.

Une interface réseau est déclarée en spécifiant un nom logique et un critère de sélection pour l'interface physique. Le serveur d'applications est fourni avec une configuration par défaut pour une gestion et un nom d'interface publique. Dans cette configuration, le groupe de l'interface publique est destiné à être utilisé par toute communication de réseau liée à l'application comme Web ou Messaging. Le groupe interface de gestion est destiné à tous les composants et les services requis par la couche de gestion, y compris HTTP Management Endpoint. Les noms d'interface sont fournis en tant que suggestions uniquement, alors que n'importe quel nom de groupe peut être substitué ou créé selon les besoins.

Les fichiers de configuration `domain.xml`, `host.xml` et `standalone.xml` comprennent tous des déclarations d'interface. Les critères de déclaration peuvent référencer une adresse générique ou spécifier un ensemble de caractéristiques qu'une interface ou une adresse doit avoir pour pouvoir établir une correspondance valide. Les exemples suivants illustrent plusieurs configurations possibles de déclarations d'interface, généralement définies soit dans le fichier de configuration `standalone.xml` ou `host.xml`. Cela permet à des groupes d'hôtes distants de maintenir leurs propres attributs d'interfaces spécifiques, tout en permettant une référence aux groupes d'interfaces dans le fichier de configuration `domain.xml` du contrôleur de domaine.

Le premier exemple montre une valeur d'`inet-address` indiquée pour le groupe de noms relatifs `management` et `public`.

Exemple 5.1. Un groupe d'interfaces créé avec une adresse inet

```
<interfaces>
  <interface name="management">
    <inet-address value="127.0.0.1"/>
  </interface>
  <interface name="public">
    <inet-address value="127.0.0.1"/>
  </interface>
</interfaces>
```

Dans l'exemple suivant, un groupe global interface utilise l'élément `any-address` pour déclarer une adresse générique.

Exemple 5.2. Un groupe global créé avec une déclaration générique

```
<interface name="global">
  <!-- Use the wild-card address -->
  <any-address/>
</interface>
```

L'exemple suivant déclare une carte d'interface de réseau sous un groupe relatif avec un nom externe.

Exemple 5.3. Un groupe externe créé avec un NIC

```
<interface name="external">
  <nic name="eth0"/>
</interface>
```

Dans l'exemple suivant, une déclaration est créée comme groupe par défaut pour un besoin spécifique. Dans ce cas, les caractéristiques des éléments supplémentaires définissent les conditions pour que l'interface soit une correspondance valide. Cela permet la création de groupes de déclaration d'interface très spécifique, avec la possibilité de les référencer de manière prédéfinie, réduisant ainsi le temps de configuration et d'administration sur plusieurs instances de serveur.

Exemple 5.4. Un groupe par défaut créé avec des valeurs conditionnelles spécifiques

```
<interface name="default">
  <!-- Match any interface/address on the right subnet if it's
        up, supports multicast, and isn't point-to-point -->
  <subnet-match value="192.168.0.0/16"/>
  <up/>
  <multicast/>
  <not>
    <point-to-point/>
  </not>
</interface>
```

Alors que les déclarations d'interface peuvent être faites et modifiées dans les fichiers de configuration des sources, l'interface CLI et la console de gestion fournissent un environnement sécurisé, contrôlé et persistant pour les modifications de configuration.

[Rapporter un bogue](#)

5.1.2. Configurer les interfaces

Les configurations de l'interface par défaut des fichiers de configuration `standalone.xml` et `host.xml` offrent trois interfaces nommées avec jetons d'interfaces relatives pour chacune. Vous pouvez utiliser la console de gestion ou l'interface CLI pour configurer des attributs et valeurs supplémentaires indiquées dans le tableau ci-dessous. Vous pouvez également remplacer les liaisons

d'interfaces relatives par des valeurs spécifiques selon les besoins, mais notez que si vous le faites, vous serez incapable de passer une valeur d'interface en cours d'exécution de serveur, car **-b>** peut seulement substituer une valeur relative.

Exemple 5.5. Configuration d'interface par défaut

```
<interfaces>
  <interface name="management">
    <inet-address value="{jboss.bind.address.management:127.0.0.1}"/>
  </interface>
  <interface name="public">
    <inet-address value="{jboss.bind.address:127.0.0.1}"/>
  </interface>
  <interface name="unsecure">
    <inet-address value="{jboss.bind.address.unsecure:127.0.0.1}"/>
  </interface>
</interfaces>
```

Tableau 5.1. Attributs et valeurs d'interface

Élément d'interface	Description
any	Élément vide de type exclusion d'adresse, utilisé pour forcer le critère de sélection.
any-address	Élément vide indiquant que les sockets qui utilisent cette interface doivent être liés à une adresse générique. L'adresse générique IPv6 (::) sera utilisée à moins que la propriété système java.net.preferIPv4Stack soit définie sur true, dans lequel cas, l'adresse générique (0.0.0.0) IPv4 sera utilisée. Si un socket est lié à une adresse anylocal IPv6 sur une machine dual-stack, il pourra accepter le trafic IPv6 et IPv4 ; si lié à l'adresse IPv4 anylocal (mappées IPv4), il ne peut accepter que le trafic IPv4.
any-ipv4-address	Élément vide indiquant que les sockets qui utilisent cette interface doivent être liés à une adresse générique (0.0.0.0) IPv4.
any-ipv6-address	Élément vide indiquant que les sockets qui utilisent cette interface doivent être liés à une adresse générique (::). IPv6.
inet-address	Soit une adresse IP en notation à points IPV6 ou IPV4, ou un nom d'hôte pouvant être résolu.
link-local-address	Élément vide indiquant qu'une partie du critère de sélection d'une interface devrait consister à savoir si oui ou non il a une adresse associée local-link.
loopback	Élément vide indiquant qu'une partie du critère de sélection d'une interface est de savoir s'il s'agit oui ou non d'une interface de loopback.

Élément d'interface	Description
loopback-address	Une adresse de loopback qui ne peut pas réellement être configurée sur l'interface de loopback de la machine. Diffère d'inet-addressType car la valeur donnée sera utilisée même si aucune carte réseau possédant l'adresse IP associée ne peut être trouvée.
multicast	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être si oui ou non il y a un support multi-diffusion.
nic	Le nom d'une interface de réseau (e.g. eth0, eth1, lo).
nic-match	Une expression standard à laquelle faire correspondre les noms des interfaces de réseau disponibles sur la machine pour trouver une interface qui convienne.
not	Élément vide de type exclusion d'adresse, utilisé pour forcer le critère de sélection.
point-to-point	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être de savoir si elle a oui ou non une interface d'un point à un autre.
public-address	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être de savoir si elle a oui ou non une adresse publiquement routable.
site-local-address	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être ou non une adresse associée à son site-local
subnet-match	Une adresse IP réseau et le nombre de bits dans le préfixe de réseau de l'adresse, sous la forme « / » ; par exemple "192.168.0.0/16".
up	Élément vide indiquant qu'une partie du critère de sélection d'une interface est active ou non.
virtual	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être ou non une interface virtuelle.

- **Configuration des attributs d'une interface**

Vous pouvez utiliser la fonction de saisie semi-automatique pour saisir la commande au fur et à mesure que vous saisissez, et afin d'exposer les attributs disponibles.

- **Configurer les attributs d'interface par l'interface CLI**

Utiliser l'interface CLI pour ajouter de nouvelles interfaces et pour écrire des nouvelles valeurs dans les attributs de l'interface.

- a. **Ajouter une nouvelle interface**

L'opération **add** (ajouter) crée de nouvelles interfaces selon les besoins. Vous pouvez exécuter cette commande **add** à partir de la racine de la session CLI, qui, dans

l'exemple suivant, crée un nouveau titre de nom d'interface *interfacename*, avec une **inet -address** déclarée *12.0.0.2*.

```
/interface=interfacename/:add(inet-address=12.0.0.2)
```

b. Modifier les attributs d'une interface

L'opération **write-attribute** écrit de nouvelles valeurs sur un attribut. L'exemple suivant met à jour la valeur de **inet -address** à *12.0.0.8*.

```
/interface=interfacename/:write-attribute(name=inet-address,  
value=12.0.0.8)
```

c. Vérifier les attributs d'interface.

Confirmer que les valeurs d'attribut ont changé en exécutant l'opération **read-resource** accompagnée du paramètre **include-runtime=true** pour exposer toutes les valeurs actives en cours du modèle de serveur. Par exemple :

```
[standalone@localhost:9999 interface=public] :read-  
resource(include-runtime=true)
```

o Configurer les attributs d'interface par la console de gestion

a. Connectez-vous à la console de management.

Connectez-vous à la console de gestion de votre instance de serveur autonome ou de domaine géré.

b. Naviguer dans l'écran d'interfaces

i. Naviguer dans l'onglet de configuration.

Sélectionner **Configuration** qui se trouve en haut de l'écran.

ii. Mode domaine uniquement

Sélectionner un profil à partir du menu déroulant **Profile** qui se situe en haut et à gauche de l'écran.

c. Sélectionner Interfaces à partir du menu de navigation.

Étendre le menu **General Configuration**. Sélectionner l'élément de menu **Interfaces** à partir du menu de navigation.

d. Ajouter une nouvelle interface

i. Cliquer sur Ajouter.

ii. Saisir les valeurs qui conviennent pour les Name (Nom), Inet Address (Adresse Inet) et Address Wildcard (Adresse générique).

iii. Cliquer sur le bouton Save.

e. Modifier les attributs d'une interface

i. Sélectionner l'interface que vous souhaitez modifier dans la liste Available Interfaces et cliquer sur le bouton Edit.

- ii. Saisir les valeurs qui conviennent pour les **Name (Nom)**, **Inet Address (Adresse Inet)** et **Address Wildcard (Adresse générique)**.
- iii. Cliquer sur le bouton **Save**.

[Rapporter un bogue](#)

5.2. GROUPES DE LIAISONS DE SOCKETS

5.2.1. Groupes de liaisons de sockets

Les liaisons de sockets et les groupes de liaisons de sockets vous permettent de définir les ports de réseau et leur relation aux interfaces de réseau requises pour votre configuration JBoss EAP 6.

Une liaison de socket est une configuration nommée pour un socket. Les déclarations pour ces configurations nommées se trouvent dans les deux fichiers de configuration `domain.xml` et `standalone.xml`. D'autres sections de la configuration peuvent alors faire référence à ces sockets par leur nom logique, plutôt que d'avoir à inclure tous les détails de la configuration de socket. Cela permet de référencer des configurations de sockets relatives qui peuvent varier sur des machines différentes.

Les liaisons de sockets sont rassemblées sous un groupe de liaisons de sockets. Un groupe de liaisons de sockets est une collection de déclarations de liaisons de sockets qui sont regroupées sous un nom logique. Le groupe peut ensuite être référencé dans l'ensemble de la configuration. Un serveur autonome contient uniquement un de ces groupes, alors qu'une instance managée de domaine peut contenir plusieurs groupes. Vous pouvez créer un groupe de liaison de socket pour chaque groupe de serveurs dans le domaine géré, ou partager un groupe de liaisons de sockets entre plusieurs groupes de serveurs.

Les groupes de noms permettent à des références simplifiées d'être utilisées pour des groupes particuliers de liaisons de sockets lors de la configuration des groupes de serveurs, dans le cas d'un domaine géré. Une autre utilisation courante est pour la configuration et la gestion de plusieurs instances du serveur autonome sur un seul système. Les exemples suivants montrent les groupes de liaison de sockets par défaut dans les fichiers de configuration des instances du domaine et du serveur autonome.

Exemple 5.6. Liaisons de sockets par défaut pour la configuration du serveur autonome.

Les groupes de liaisons de sockets par défaut dans le fichier de configuration `standalone.xml` sont groupés sous `standard-sockets`. Ce groupe est aussi référencé dans l'interface `public`, en utilisant la même méthodologie de référencement logique.

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="{jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="{jboss.management.native.port:9999}" />
  <socket-binding name="management-http" interface="management"
port="{jboss.management.http.port:9990}" />
  <socket-binding name="management-https" interface="management"
port="{jboss.management.https.port:9443}" />
  <socket-binding name="ajp" port="8009" />
  <socket-binding name="http" port="8080" />
  <socket-binding name="https" port="8443" />
  <socket-binding name="remoting" port="4447" />
```

```

    <socket-binding name="txn-recovery-environment" port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
    <outbound-socket-binding name="mail-smtp">
      <remote-destination host="localhost" port="25"/>
    </outbound-socket-binding>
  </socket-binding-group>

```

Exemple 5.7. Liaisons de sockets par défaut pour la configuration du serveur autonome.

Les groupes de liaisons de sockets par défaut dans le fichier de configuration `domain.xml` contiennent quatre groupes : **standard-sockets**, **ha-sockets**, **full-sockets** et **full-ha-sockets**. Ces groupes sont également référencés dans une interface appelée **public**.

```

<socket-binding-groups>
  <socket-binding-group name="standard-sockets" default-
interface="public">
    <!-- Needed for server groups using the 'default' profile
-->
    <socket-binding name="ajp" port="8009"/>
    <socket-binding name="http" port="8080"/>
    <socket-binding name="https" port="8443"/>
    <socket-binding name="remoting" port="4447"/>
    <socket-binding name="txn-recovery-environment"
port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
    <outbound-socket-binding name="mail-smtp">
      <remote-destination host="localhost" port="25"/>
    </outbound-socket-binding>
  </socket-binding-group>
  <socket-binding-group name="ha-sockets" default-
interface="public">
    <!-- Needed for server groups using the 'ha' profile -->
    <socket-binding name="ajp" port="8009"/>
    <socket-binding name="http" port="8080"/>
    <socket-binding name="https" port="8443"/>
    <socket-binding name="jgroups-mping" port="0" multicast-
address="${jboss      .default.multicast.address:230.0.0.4}"
multicast-port="45700"/>
    <socket-binding name="jgroups-tcp" port="7600"/>
    <socket-binding name="jgroups-tcp-fd" port="57600"/>
    <socket-binding name="jgroups-udp" port="55200" multicast-
address="${jboss      .default.multicast.address:230.0.0.4}"
multicast-port="45688"/>
    <socket-binding name="jgroups-udp-fd" port="54200"/>
    <socket-binding name="modcluster" port="0" multicast-
address="224.0.1.105"      multicast-port="23364"/>
    <socket-binding name="remoting" port="4447"/>
    <socket-binding name="txn-recovery-environment"
port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
    <outbound-socket-binding name="mail-smtp">
      <remote-destination host="localhost" port="25"/>
    </outbound-socket-binding>
  </socket-binding-group>

```



```

    <socket-binding-group name="full-sockets" default-
interface="public">
    <!-- Needed for server groups using the 'full' profile -->
    <socket-binding name="ajp" port="8009"/>
    <socket-binding name="http" port="8080"/>
    <socket-binding name="https" port="8443"/>
    <socket-binding name="jacob" interface="unsecure"
port="3528"/>
    <socket-binding name="jacob-ssl" interface="unsecure"
port="3529"/>
    <socket-binding name="messaging" port="5445"/>
    <socket-binding name="messaging-group" port="0" multicast-
address="${jboss      .messaging.group.address:231.7.7.7}" multicast-
port="${jboss.messaging.group.port:9876}"/>
    <socket-binding name="messaging-throughput" port="5455"/>
    <socket-binding name="remoting" port="4447"/>
    <socket-binding name="txn-recovery-environment"
port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
    <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="25"/>
    </outbound-socket-binding>
    </socket-binding-group>
    <socket-binding-group name="full-ha-sockets" default-
interface="public">
    <!-- Needed for server groups using the 'full-ha' profile
-->
    <socket-binding name="ajp" port="8009"/>
    <socket-binding name="http" port="8080"/>
    <socket-binding name="https" port="8443"/>
    <socket-binding name="jacob" interface="unsecure"
port="3528"/>
    <socket-binding name="jacob-ssl" interface="unsecure"
port="3529"/>
    <socket-binding name="jgroups-mping" port="0" multicast-
address="${jboss      .default.multicast.address:230.0.0.4}"
multicast-port="45700"/>
    <socket-binding name="jgroups-tcp" port="7600"/>
    <socket-binding name="jgroups-tcp-fd" port="57600"/>
    <socket-binding name="jgroups-udp" port="55200" multicast-
address="${jboss      .default.multicast.address:230.0.0.4}"
multicast-port="45688"/>
    <socket-binding name="jgroups-udp-fd" port="54200"/>
    <socket-binding name="messaging" port="5445"/>
    <socket-binding name="messaging-group" port="0" multicast-
address="${jboss      .messaging.group.address:231.7.7.7}" multicast-
port="${jboss.messaging.group.port:9876}"/>
    <socket-binding name="messaging-throughput" port="5455"/>
    <socket-binding name="modcluster" port="0" multicast-
address="224.0.1.105"      multicast-port="23364"/>
    <socket-binding name="remoting" port="4447"/>
    <socket-binding name="txn-recovery-environment"
port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
    <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="25"/>

```

```

        </outbound-socket-binding>
    </socket-binding-group>
</socket-binding-groups>

```

Les instances de liaisons de sockets peuvent être créées et modifiées dans les fichiers source `standalone.xml` et `domain.xml` du répertoire de l'application. La méthode recommandée pour gérer les liaisons consiste à utiliser la console de gestion ou l'interface CLI. Les avantages à utiliser la console de gestion est l'interface utilisateur graphique avec un écran de groupes de liaisons de sockets dédié dans la section **Configuration générale**. L'interface CLI propose un API et un flux de travail basés ligne de commande qui permettent le traitement par lots et l'utilisation de scripts aux niveaux supérieurs et inférieurs de la configuration de serveur d'applications. Les deux interfaces permettent la persistance des modifications ou bien leur enregistrement dans la configuration du serveur.

[Rapporter un bogue](#)

5.2.2. Configurer les liaisons de sockets

Les liaisons de sockets peuvent être définies dans des groupes de liaisons sockets uniques. Le serveur autonome contient un de ces groupes, le groupe `standard-sockets`, et ne peut pas créer de groupes supplémentaires. À la place, vous pouvez créer des fichiers de configuration de serveur autonome alternatif. Pour le domaine géré cependant, vous pouvez créer des groupes de liaisons de sockets et configurer les liaisons de sockets qu'ils contiennent selon vos besoins. Le tableau suivant montre les attributs disponibles pour chaque liaison de sockets.

Tableau 5.2. Attributs de liaisons de sockets

Attribut	Description	Rôle
name	Nom logique de la configuration de socket qui doit être utilisée ailleurs dans la configuration.	Requis
port	Port de base auquel un socket basé sur cette configuration doit être lié. Notez que les serveurs peuvent être configurés pour substituer cette valeur de base en appliquant une incrémentation ou décrémentation à toutes les valeurs de port.	Requis
interface	Nom logique de l'interface à laquelle un socket basé sur cette configuration doit être lié. Si non défini, la valeur de l'attribut default-interface du groupe de liaison de sockets enveloppant servira.	Option

Attribut	Description	Rôle
multicast-address	Si le socket doit être utilisé en multi-diffusion, c'est l'adresse multi-diffusion qu'il vous faut.	Option
multicast-port	Si le socket doit être utilisé en multi-diffusion, c'est le port multi-diffusion qu'il vous faut.	Option
fixed-port	Si défini sur true , déclare que la valeur de port doit toujours être utilisée par le socket et ne doit pas être remplacée par l'ajout d'un incrément ou d'un décrétement.	Option

- **Configurer des liaisons de sockets dans des groupes de liaisons de sockets**

Sélectionner le CLI ou la console de gestion pour configurer vos liaisons de sockets selon les besoins.

- **Configurer les liaisons de sockets par l'interface CLI**

Utiliser l'interface CLI pour configurer les liaisons de sockets.

- a. **Ajouter une nouvelle liaison de sockets**

Utiliser l'opération **add** (ajouter) pour créer une nouvelle configuration d'adresse si nécessaire. Vous pouvez exécuter cette commande à partir de la racine de la session de CLI, qui, dans l'exemple suivant, crée une nouvelle liaison de sockets intitulée *newsocket*, avec un attribut **port** déclaré comme *1234*. Les exemples s'appliquent à la fois pour la modification des serveurs autonomes et des serveurs gérés sur la liaison de sockets **standard-sockets** comme montré ci-dessous.

```
[domain@localhost:9999 /] /socket-binding-group=standard-sockets/socket-binding=newsocket/:add(port=1234)
```

- b. **Modifier les attributs de modèle**

Utiliser l'opération **write-attribute** pour écrire une nouvelle valeur dans un attribut. Vous pouvez utiliser l'onglet de complétion pour aider à compléter la chaîne de commande que vous saisissez, et pour exposer les attributs disponibles. L'exemple suivant met à jour la valeur **port** à *2020*

```
[domain@localhost:9999 /] /socket-binding-group=standard-sockets/socket-binding=newsocket/:write-attribute(name=port,value=2020)
```

- c. **Confirmer les attributs de modèle**

Confirmer que les valeurs ont changé en exécutant **read-resource** accompagné du paramètre **include-runtime=true** pour exposer toutes les valeurs actives en cours du modèle de serveur.

```
[domain@localhost:9999 /] /socket-binding-group=standard-sockets/socket-binding=newsocket/:read-resource
```

o **Configurer les liaisons de sockets par la console de gestion**

Utiliser la console de gestion pour configurer les liaisons de sockets.

a. **Connectez-vous à la console de management.**

Connectez-vous à la console de gestion de votre domaine géré ou de votre serveur autonome.

b. **Naviguer dans Configuration.**

Sélectionner **Configuration** qui se trouve en haut de l'écran.

c. **Sélectionner l'élément Socket Binding à partir du menu de navigation.**

Étendre le menu **General Configuration**. Sélectionner **Socket Binding**. Si vous utilisez un domaine géré, sélectionner le groupe désiré dans la liste **Socket Binding Groups**.

d. **Ajouter une nouvelle liaison de sockets**

i. Cliquer sur le bouton **Add** (ajouter).

ii. Saisir les valeurs qui conviennent pour les **Name** (Nom), **Port** (Port) et **Binding Group** (Groupe de liaisons).

iii. Cliquer sur le bouton **Save** pour terminer.

e. **Modifier la liaison de socket**

i. Sélectionner la liaison de sockets de la liste et cliquer sur le bouton **Edit**.

ii. Saisir les valeurs qui conviennent pour les **Name** (Nom), **Interface** (Interface) ou **Port** (Port).

iii. Cliquer sur le bouton **Save** pour terminer.

[Rapporter un bogue](#)

5.2.3. Ports de réseau utilisés par JBoss EAP 6

Les ports utilisés par la configuration JBoss EAP 6 par défaut sont liés à plusieurs facteurs :

- Le fait que vos groupes de serveurs utilisent le groupe de liaisons de sockets par défaut, ou un groupe de liaisons de sockets personnalisé.
- Les exigences de vos déploiements individuels.



NOTE

Un décalage de port numérique peut être configuré pour atténuer les conflits de ports lorsque vous exécutez plusieurs serveurs sur un même serveur physique. Si votre serveur utilise un décalage de port numérique, ajoutez la valeur de décalage au numéro de port par défaut pour le groupe de liaisons de sockets de son groupe de serveurs. Par exemple, si le port HTTP du groupe de liaisons de socket est **8080** et si votre serveur utilise un décalage de port de **100**, son port HTTP sera **8180**.

À moins d'instruction particulière, les ports utilisent le protocole TCP.

groupes de liaisons de sockets par défaut

- **full-ha-sockets**
- **full-sockets**
- **ha-sockets**
- **standard-sockets**

Ces groupes de liaisons de sockets sont disponibles uniquement dans `domain.xml`. Les profils de serveur autonomes contiennent uniquement le groupe de liaisons de sockets standards. Ce groupe correspond à `des-sockets standards` dans `standalone.xml`, `ha-sockets` pour `standalone-ha.xml`, `full-sockets` pour `standalone-full.xml`, et `full-ha-sockets` pour `standalone-full-ha.xml`. Les profils autonomes contiennent certaines liaisons de sockets supplémentaires, comme par exemple, `management-{native,http,https}`.

Tableau 5.3. Référence aux groupes de liaisons de sockets par défaut

Nom	Port	Port multi-diffusion	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
ajp	8009		Protocole Apache JServ. Utilisé pour le clustering HTTP et pour l'équilibrage des charges.	Oui	Oui	Oui	Oui
http	8080		Le port par défaut des applications déployées.	Oui	Oui	Oui	Oui
https	8443		Connexion cryptée-SSL entre les applications déployées et les clients.	Oui	Oui	Oui	Oui
jacorb	3528		Services CORBA pour les transactions JTS et autres services dépendants-ORB.	Oui	Oui	Non	Non
jacorb-ssl	3529		Services CORBA cryptés-SSL.	Oui	Oui	Non	Non

Nom	Port	Port multi-diffusion	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
jgroups-diagnostics		7500	Multidiffusion. Utilisée pour découvrir des homologues dans les groupements HA. Non configurable par les interfaces de gestion.	Oui	Non	Oui	Non
jgroups-mping		45700	Multidiffusion. Utilisée pour découvrir l'appartenance de groupe d'origine dans un cluster HA.	Oui	Non	Oui	Non
jgroups-tcp	7600		Découverte d'homologues unicastes dans les groupements HA avec TCP.	Oui	Non	Oui	Non
jgroups-tcp-fd	57600		Utilisé pour la détection des échecs en TCP.	Oui	Non	Oui	Non
jgroups-udp	55200	45688	Découverte de pairs multicast dans les groupements HA avec UDP.	Oui	Non	Oui	Non
jgroups-udp-fd	54200		Utilisé pour la détection des échecs par UDP.	Oui	Non	Oui	Non
messaging	5445		Service JMS.	Oui	Oui	Non	Non
messaging-group			Référencé par la diffusion HornetQ JMS et les groupes Discovery	Oui	Oui	Non	Non
messaging-throughput	5455		Utilisé par JMS Remoting.	Oui	Oui	Non	Non

Nom	Port	Port multi-diffusion	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
mod_cluster		23364	Port de multidiffusion pour la communication entre JBoss EAP 6 et l'équilibreur de charges HTTP.	Oui	Non	Oui	Non
remoting	4447		Utilisé pour l'invocation EJB.	Oui	Oui	Oui	Oui
txn-recovery-environment	4712		Gestionnaire de recouvrement des transactions JTA.	Oui	Oui	Oui	Oui
txn-status-manager	4713		Gestionnaire des transactions JTA / JTS.	Oui	Oui	Oui	Oui

Ports de gestion

En plus des groupes de liaisons de sockets, chaque contrôleur hôte ouvre deux ports supplémentaires pour la gestion :

- **9990** - Le port de console de gestion web
- **9999** - Le port utilisé par la console de gestion et par l'API de gestion.

De plus, si HTTPS est activé pour la console de gestion, 9443 sera également ouvert en tant que valeur de port par défaut.

[Rapporter un bogue](#)

5.2.4. Valeurs de décalage des ports pour les groupes de liaisons de sockets

Les valeurs de décalage des ports (Port offsets) est un décalage chiffré qui vient s'ajouter aux valeurs de port données par le groupe de liaisons de sockets pour ce serveur. Cela permet à un seul serveur d'hériter les liaisons de sockets du groupe de serveurs auquel il appartient, avec un décalage pour veiller à ce qu'il n'entre pas en conflit avec les autres serveurs du groupe. Par exemple, si le port HTTP du groupe de liaisons de sockets est 8080 et si votre serveur utilise un port offset de 100, son port HTTP sera 8180.

[Rapporter un bogue](#)

5.2.5. Configurer Port Offset (valeurs de décalage de ports)

- **Configurer Port Offset (valeurs de décalage de ports)**

Sélectionner l'interface CLI ou la console de gestion pour configurer vos valeurs de décalage de ports.

- **Configurer Port Offsets par l'interface CLI**

Sélectionner l'interface CLI pour configurer vos ports offsets.

- a. **Modifier les Ports Offsets**

Utiliser l'opération `write-attribute` pour écrire une nouvelle valeur référence de port. L'exemple suivant met à jour la valeur du `socket-binding-port-offset` de `server-two` à `250`. Ce serveur est membre du groupe hôte local par défaut. Un redémarrage est nécessaire pour que les modifications puissent prendre effet.

```
[domain@localhost:9999 /] /host=master/server-config=server-two/:write-attribute(name=socket-binding-port-offset,value=250)
```

- b. **Confirmer les attributs d'un Port Offset**

Confirmer que les valeurs ont changé en exécutant `read-resource` accompagné du paramètre `include-runtime=true` pour exposer toutes les valeurs actives en cours du modèle de serveur.

```
[domain@localhost:9999 /] /host=master/server-config=server-two/:read-resource(include-runtime=true)
```

- **Configurer les valeurs de décalage de ports par la console de gestion**

Sélectionner la console de gestion pour configurer vos valeurs de décalage de ports.

- a. **Connectez-vous à la console de gestion.**

Connectez-vous à la console de gestion de votre domaine géré.

- b. **Sélectionner l'onglet Domain**

Sélectionner l'onglet **Domain** en haut de l'écran.

- c. **Modifier les attributs de Port Offset**

- i. Sélectionner le serveur dans la liste **Available Server Configurations** et cliquer sur **Edit** en haut de la liste des attributs en dessous.

- ii. Saisir les valeurs que vous désirez dans le champ **Port Offset**.

- iii. Cliquer sur le bouton **Save** pour terminer.

[Rapporter un bogue](#)

5.2.6. Configuration de la taille d'un message à distance

Le sous-système de communication à distance (Remoting) offre la possibilité de limiter la taille des messages pour des protocoles de communication à distance. Vous pouvez définir la taille maximale des messages entrants (`MAX_INBOUND_MESSAGE_SIZE`) et la taille maximale des messages sortants (`MAX_OUTBOUND_MESSAGE_SIZE`) pour s'assurer que les messages sont reçus et envoyés dans la limite des tailles appropriées.

La configuration de la taille des messages dans des protocoles de communication à distance contribue à une utilisation efficace de la mémoire système et l'empêche d'atteindre un état « out of memory » pendant l'exécution des opérations importantes.

Si l'expéditeur envoie un message qui dépasse la limite maximale admissible (`MAX_OUTBOUND_MESSAGE_SIZE`), le serveur lèvera une exception et annulera la transmission des données. Toutefois, la connexion restera ouverte et l'expéditeur pourra choisir de fermer le message si nécessaire.

Si un message reçoit la limite permise maximum (`MAX_INBOUND_MESSAGE_SIZE`), le message sera fermé de façon asynchrone avec la connexion restée ouverte.

[Rapporter un bogue](#)

5.3. IPV6

5.3.1. Configurer les préférences de JVM Stack d'IPv6 Networking

Résumé

Cette section couvre l'activation du réseau IPv6 de l'installation JBoss EAP 6

Procédure 5.1. Désactiver la propriété IPv4 Stack Java

1. Ouvrir le fichier qui convient à l'installation :
 - o **Pour le serveur autonome :**
Ouvrir `EAP_HOME/bin/standalone.conf`.
 - o **Pour un domaine géré :**
Ouvrir `EAP_HOME/bin/domain.conf`.

2. Modifier la propriété IPv4 Stack Java à false :

```
-Djava.net.preferIPv4Stack=false
```

Par exemple :

Exemple 5.8. Options JVM

```
# Specify options to pass to the Java VM.
#
if [ "x$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="-Xms64m -Xmx512m -XX:MaxPermSize=256m -
Djava.net.preferIPv4Stack=false
-Dorg.jboss.resolver.warning=true -
Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -
Djava.net.preferIPv6Addresses=true"
fi
```

[Rapporter un bogue](#)

5.3.2. Configurer les déclarations d'interface du réseautage IPv6

Résumé

Suivre ces étapes de configuration de l'adresse inet d'interface dans IPv6 par défaut:

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#)
- [Section 3.4.2, « Se connecter à la console de gestion »](#)

Procédure 5.2. Configurer l'interface du réseautage IPv6

1. Sélectionner **Configuration** qui se trouve en haut de l'écran.
2. Étendre le menu **General Configuration** et sélectionnez **Interfaces**.
3. Sélectionner l'interface dans la liste **Available Interfaces**.
4. Cliquer sur **Edit** dans la liste d'informations.
5. Définir l'adresse inet à :


```
| ${jboss.bind.address.management:[ADDRESS]}
```
6. Cliquer sur le bouton **Save** pour terminer.
7. Démarrer le serveur à nouveau pour implémenter les changements.

[Rapporter un bogue](#)

5.3.3. Configurer les Préférences JVM Stacks des adresses IPv6

Résumé

Cette rubrique couvre la configuration de l'installation de JBoss EAP 6 pour qu'elle préfère les adresses IPv6 dans les fichiers de configuration.

Procédure 5.3. Configuration de l'installation JBoss EAP 6 pour qu'elle préfère les adresses IPv6

1. Ouvrir le fichier qui convient à l'installation :
 - **Pour le serveur autonome :**
Ouvrir `EAP_HOME/bin/standalone.conf`.
 - **Pour un domaine géré :**
Ouvrir `EAP_HOME/bin/domain.conf`.
2. Ajouter la propriété Java suivante aux options de la Java VM

```
| -Djava.net.preferIPv6Addresses=true
```

Par exemple :

-

Exemple 5.9. Options JVM

```
# Specify options to pass to the Java VM.
#
if [ "x$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="-Xms64m -Xmx512m -XX:MaxPermSize=256m -
Djava.net.preferIPv4Stack=false
-Dorg.jboss.resolver.warning=true -
Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -
Djava.net.preferIPv6Addresses=true"
fi
```

[Rapporter un bogue](#)

CHAPITRE 6. GESTION DES SOURCES DE DONNÉES

6.1. INTRODUCTION

6.1.1. JDBC

L'API JDBC est la norme qui définit comment les bases de données sont accessibles par les applications Java. Une application configure une source de données qui référence un pilote JDBC. Le code de l'application peut alors s'inscrire au pilote, à la place de la base de données. Le pilote convertit le code dans le langage de base de données. Cela signifie que si le pilote qui convient est installé, une application pourra être utilisée sans base de données supportée.

La norme JDBC 4.0 est définie ici : <http://jcp.org/en/jsr/detail?id=221>.

Pour débuter dans JDBC et avec les sources de données, voir la section JDBC Driver du Guide d'administration et de configuration de JBoss EAP 6.

[Rapporter un bogue](#)

6.1.2. Bases de données prises en charge par JBoss EAP 6

Liste des bases de données compatibles JDBC prises en charge par JBoss EAP 6 : <https://access.redhat.com/site/articles/111663>.

[Rapporter un bogue](#)

6.1.3. Types de sources de données

Les deux grands types de ressources sont nommés **les sources de données** et **les sources de données XA**.

Les sources de données non-XA sont utilisées pour les applications qui n'utilisent pas de transactions, ou les applications qui utilisent des transactions avec une base de données simple.

Les sources de données XA sont utilisées par les applications dont les transactions sont réparties à travers plusieurs bases de données. Les sources de données XA rajoutent un niveau supplémentaire.

Vous n'avez qu'à indiquer le type de source de données quand vous la créez dans la console de gestion ou l'interface CLI.

[Rapporter un bogue](#)

6.1.4. L'exemple de source de données

JBoss EAP 6 inclut une base de données H2. Il s'agit d'un système de gestion de bases de données relationnelles léger, qui donne aux développeurs la possibilité de construire des applications rapidement, et qui représente la source de données de référence pour la plate forme.



AVERTISSEMENT

Toutefois, elle ne devrait pas être utilisée dans un environnement de production. C'est une source de données très petite, autonome, qui prend en charge toutes les normes nécessaires pour le test et la création d'applications, mais qui n'est pas fiable ou suffisamment évolutive pour une utilisation en production.

Pour obtenir une liste de sources de données certifiées ou prises en charges, consulter [Section 6.1.2, « Bases de données prises en charge par JBoss EAP 6 »](#).

[Rapporter un bogue](#)

6.1.5. Déploiement des fichiers `-ds.xml`

Dans JBoss EAP 6, les sources de données sont définies comme une ressource du sous-système de serveur. Dans les versions précédentes, un fichier de configuration de source de données `*-ds.xml` était requis dans le répertoire de déploiement de la configuration du serveur. Les fichiers `*-ds.xml` peuvent toujours être déployés dans JBoss EAP 6, suivant le schéma de sources de données 1.1 disponible sous *Schemas* : <http://www.ironjacamar.org/documentation.html>.



AVERTISSEMENT

Cette fonctionnalité doit être utilisée pour le développement uniquement. Elle n'est pas conseillée en production car non supportée par les outils de gestion et d'admin de JBoss. Cette fonctionnalité est dépréciée dans JBoss EAP 6.4 et ne sera pas prise en charge dans la prochaine version du produit.



IMPORTANT

Il est obligatoire d'utiliser une référence à une entrée de `<driver>` (pilote) déjà déployé / défini quand on déploie les fichiers `*-ds.xml`.

[Rapporter un bogue](#)

6.2. PILOTES JDBC

6.2.1. Installer un pilote JDBC avec la console de gestion

Résumé

Avant que votre application puisse se connecter à une source de données JDBC, vos pilotes JDBC du fournisseur de votre source de données doivent être installés dans un endroit où JBoss EAP 6 puisse les utiliser. JBoss EAP 6 vous permettra de déployer ces pilotes comme tout autre déploiement. Cela

signifie que vous pouvez les déployer sur plusieurs serveurs dans un groupe de serveurs, si vous utilisez un domaine géré.

Conditions préalables

Vous devrez remplir les conditions suivantes avant de pouvoir effectuer cette tâche :

- Télécharger le pilote JDBC de votre fournisseur de base de données.



NOTE

Tout pilote conforme à JDBC 4 est automatiquement reconnu et installé dans le système avec son nom et de sa version. Un JAR JDBC utilisant le mécanisme de fournisseur de service Java est identifié. Ces JAR contiennent un fichier-texte nommé META-INF/services/java.sql.Driver qui indique les nom(s) de classes de pilote de ce JAR.

Procédure 6.1. Modifier le JAR du pilote JDBC

Si le JAR de pilote JDBC n'est pas conforme à JDBC 4, il peut être rendu déployable à l'aide de la commande suivante.

1. Modifier ou créer un répertoire temporaire vide.
2. Créer un sous-répertoire META-INF.
3. Créer un sous-répertoire META-INF/services.
4. Créer un fichier META-INF/services/java.sql.Driver qui contienne une ligne indiquant le nom de classe complet du pilote JDBC.
5. Utiliser l'outil de ligne de commande du JAR pour mettre à jour le JAR ainsi :

```
jar -uf jdbc-driver.jar META-INF/services/java.sql.Driver
```

6. [Section 3.4.2, « Se connecter à la console de gestion »](#)
7. Si vous utilisez un domaine géré, déployez le fichier JAR dans un groupe de serveurs. Sinon, déployez-le sur votre serveur. Voir [Section 10.2.2, « Activer une application déployée à l'aide de la console de gestion »](#).

Résultat :

Le pilote JDBC est déployé, et est disponible pour vos applications.

[Rapporter un bogue](#)

6.2.2. Installer un pilote JDBC comme core module

Conditions préalables

Vous devrez remplir les conditions suivantes avant de pouvoir effectuer cette tâche :

- Télécharger le pilote JDBC de votre base de données fournisseur. Voici les adresses de téléchargement pour le pilote JDBC : [Section 6.2.3, « Adresses des téléchargements de pilotes JDBC »](#).

- En extraire l'archive

Procédure 6.2. Installer un pilote JDBC comme core module

1. Créer une structure de chemin d'accès de fichier sous le répertoire *EAP_HOME/modules/*. Ainsi, pour un pilote MySQL JDBC, créer une structure de répertoire comme suit : *EAP_HOME/modules/com/mysql/main/*.
2. Copier le pilote JDBC dans le sous-répertoire *main/*.
3. Dans le sous-répertoire *main/*, créer un fichier *module.xml* ressemblant à l'exemple qui se trouve [Section 7.1.1, « Modules »](#). Le module XSD est défini dans le fichier *EAP_HOME/docs/schema/module-1_2.xsd*.
4. Démarrer le serveur.
5. Démarrer l'interface CLI.
6. Exécuter la commande CLI pour ajouter le module de pilote JDBC à la configuration du serveur.

La commande que vous avez choisie dépend du nombre de classe listées dans le fichier */META-INF/services/java.sql.Driver* qui se situe dans le JAR du pilote JDBC. Par exemple, le fichier */META-INF/services/java.sql.Driver* du JAR JDBC MySQL 5.1.20 liste deux classes :

- `com.mysql.jdbc.Driver`
- `com.mysql.fabric.jdbc.FabricMySQLDriver`

Quand il n'y a plus d'une entrée, vous devez également spécifier le nom de la classe de pilote. Sinon, vous provoquerez une erreur semblable à ce qui suit :

Exemple 6.1. Erreur de classe de pilote

```
JBAS014749: Operation handler failed: Service jboss.jdbc-driver.mysql is already registered
```

- Exécuter la commande CLI pour les JAR JDBC qui contiennent une entrée de classe.

```
/subsystem=datasources/jdbc-driver=DRIVER_NAME:add(driver-name=DRIVER_NAME, driver-module-name=MODULE_NAME, driver-xa-datasource-class-name=XA_DATASOURCE_CLASS_NAME)
```

Exemple 6.2. Commande CLI en mode autonome pour les JAR JDBC ayant une classe de pilote.

```
/subsystem=datasources/jdbc-driver=mysql:add(driver-name=mysql, driver-module-name=com.mysql, driver-xa-datasource-class-name=com.mysql.jdbc.jdbc2.optional.MySQLXADataSource)
```

Exemple 6.3. Commande CLI en mode de domaine pour les JAR JDBC ayant une classe de pilote.

```
/profile=ha/subsystem=datasources/jdbc-driver=mysql:add(driver-
name=mysql,driver-module-name=com.mysql,driver-xa-datasource-
class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource)
```

- o Exécuter la commande CLI pour les JAR JDBC qui contiennent plusieurs entrées de classe.

```
/subsystem=datasources/jdbc-driver=DRIVER_NAME:add(driver-
name=DRIVER_NAME,driver-module-name=MODULE_NAME,driver-xa-
datasource-class-name=XA_DATASOURCE_CLASS_NAME,driver-class-
name=DRIVER_CLASS_NAME)
```

Exemple 6.4. Commande CLI en mode autonome pour les JAR JDBC ayant plusieurs classes de pilote.

```
/subsystem=datasources/jdbc-driver=mysql:add(driver-
name=mysql,driver-module-name=com.mysql,driver-xa-datasource-
class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource,
driver-class-name=com.mysql.jdbc.Driver)
```

Exemple 6.5. Commande CLI en mode de domaine pour les JAR JDBC ayant plusieurs classes de pilote.

```
/profile=ha/subsystem=datasources/jdbc-driver=mysql:add(driver-
name=mysql,driver-module-name=com.mysql,driver-xa-datasource-
class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource,
driver-class-name=com.mysql.jdbc.Driver)
```

Résultat

Le pilote JDBC est maintenant installé et configuré comme core module. Il est maintenant prêt à être référencé par les sources de données d'application.

[Rapporter un bogue](#)

6.2.3. Adresses des téléchargements de pilotes JDBC

Le tableau suivant donne les adresses de téléchargement standard pour les pilotes JDBC de bases de données utilisées avec JBoss EAP 6. Ces liens pointent vers des sites tiers qui ne sont pas contrôlés ou surveillés activement par Red Hat. Pour les pilotes les plus à jour de votre base de données, consultez le site Web et la documentation de votre fournisseur de base de données.

Tableau 6.1. Adresses des téléchargements du pilote JDBC

Fournisseur	Adresse de téléchargement
MySQL	http://www.mysql.com/products/connector/
PostgreSQL	http://jdbc.postgresql.org/

Fournisseur	Adresse de téléchargement
Oracle	http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html
IBM	http://www-306.ibm.com/software/data/db2/java/
Sybase	http://www.sybase.com/products/allproductsaz/softwaredeveloperkit/jconnect
Microsoft	http://msdn.microsoft.com/data/jdbc/

[Rapporter un bogue](#)

6.2.4. Accès aux classes spécifiques à un fournisseur

Résumé

Cette section couvre les étapes à suivre pour utiliser les classes spécifiques JDBC. Cela est utile quand une application a besoin d'utiliser une fonctionnalité spécifique à un fournisseur ne faisant pas partie de l'API JDBC.



AVERTISSEMENT

Ceci est une procédure d'utilisation avancée. Seules les applications qui ont besoin d'une fonctionnalité que l'on ne peut pas trouver dans l'API JDBC doivent implémenter cette procédure.



IMPORTANT

Ce processus est requis pour le mécanisme de ré-authentification et pour accéder aux classes spécifiques au fournisseur.



IMPORTANT

Suivre les directives de l'API spécifiques au fournisseur de près, car la connexion est contrôlée par le conteneur IronJacamar.

Conditions préalables

- [Section 6.2.2, « Installer un pilote JDBC comme core module »](#).

Procédure 6.3. Ajouter une dépendance à l'application

Vous pouvez ajouter une dépendance à une application en utilisant la méthode que vous préférez.

- ○ Configurer le fichier `MANIFEST.MF`

- a. Ouvrir le fichier **META-INF/MANIFEST.MF** de l'application dans un éditeur de texte.
- b. Ajouter une dépendance au module JDBC et sauvegarder le fichier.

```
Dépendences : MODULE_NAME
```

Exemple 6.6. Fichier MANIFEST.MF avec com.mysql déclaré comme dépendance.

```
Dépendences : com.mysql
```

- o a. **Créer un fichier jboss-deployment-structure.xml**
Créer un fichier **jboss-deployment-structure.xml** dans le dossier **META-INF/** ou **WEB-INF** de l'application.

Exemple 6.7. Fichier jboss-deployment-structure.xml avec com.mysql déclaré comme dépendance.

```
<jboss-deployment-structure>
  <deployment>
    <dependencies>
      <module name="com.mysql" />
    </dependencies>
  </deployment>
</jboss-deployment-structure>
```

Exemple 6.8. Accéder à l'API spécifique du fournisseur

L'exemple ci-dessous accède à l'API MySQL.

```
import java.sql.Connection;
import org.jboss.jca.adapters.jdbc.WrappedConnection;

Connection c = ds.getConnection();
WrappedConnection wc = (WrappedConnection)c;
com.mysql.jdbc.Connection mc = wc.getUnderlyingConnection();
```

[Rapporter un bogue](#)

6.3. SOURCES DE DONNÉES NON-XA

6.3.1. Créer une source de données non-XA avec les interfaces de gestion

Résumé

Cette section explique les étapes à suivre pour créer une source de données non-XA, en utilisant la console de gestion ou l'interface CLI.

Conditions préalables

- Le serveur JBoss EAP 6 doit être en cours d'exécution.



NOTE

Avant la version 10.2 de la source de données Oracle, le paramètre `<no-tx-separate-pools/>` était requis, car le mélange de connexions transactionnelles et non-transactionnelles aurait créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

Procédure 6.4. Créer une source de données en utilisant l'interface CLI ou la console de gestion

- ○ **Interface CLI**
 - a. Lancer le CLI et connectez-vous à votre serveur.
 - b. Exécuter la commande d'interface de gestion CLI suivante pour créer une source de données non-XA, et configurer les variables comme il se doit :



NOTE

La valeur de `DRIVER_NAME` (nom de pilote) dépend du nombre de classes répertoriées dans le fichier `/META-INF/services/java.sql.Driver` situé dans le JAR du pilote JDBC. S'il n'y a qu'une seule classe, la valeur correspondra au nom du JAR. S'il y a plusieurs classes, la valeur correspondra au nom du JAR + `driverClassName` + «`_`» + `majorVersion` + «`_`» + `minorVersion`. Toute erreur provoquera l'erreur suivante dans le journal :

```
JBAS014775: New missing/unsatisfied dependencies
```

Par exemple, la valeur de `DRIVER_NAME` qu'il nous faut pour le pilote MySQL 5.1.31 est `mysql-connector-java-5.1.31-bin.jarcom.mysql.jdbc.Driver_5_1`.

```
data-source add --name=DATASOURCE_NAME --jndi-name=JNDI_NAME -
-driver-name=DRIVER_NAME --connection-url=CONNECTION_URL
```

- c. Activer la source de données :

```
data-source enable --name=DATASOURCE_NAME
```

- **Console de gestion**

- a. Connectez-vous à la console de gestion.
- b. **Naviguer dans le panneau Datasources qui se trouve dans la console de gestion**
 - i. Sélectionner **Configuration** qui se trouve en haut de la console.
 - ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.

- iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector**.
 - iv. Sélectionner **Datasources** à partir du menu à gauche de la console.
- c. **Créer une nouvelle source de données**
- i. Cliquer sur **Add** qui se trouve en haut du panneau **Datasources**.
 - ii. Saisir les attributs de la nouvelle source de données de l'assistant **Create Datasource** et appuyez sur **Next**.
 - iii. Saisir les informations sur le pilote JDBC dans l'assistant **Create Datasource** et cliquer sur **Next** pour continuer.
 - iv. Saisir les paramètres de connexion dans l'assistant **Create Datasource**.
 - v. Cliquer sur le bouton **Test Connection** pour tester la connexion à la ressource de données et vérifier que les paramètres de configuration sont corrects.
 - vi. Cliquer sur **Done** pour terminer.

Résultat

La source de données non-XA a été ajoutée au serveur. Elle est maintenant visible dans le fichier `standalone.xml` ou le fichier `domain.xml`, ainsi que dans les interfaces de gestion.

[Rapporter un bogue](#)

6.3.2. Modifier une source de données non-XA par les interfaces de gestion

Résumé

Cette section explique les étapes à suivre pour modifier une source de données non-XA, en utilisant la console de gestion ou l'interface CLI.

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#).



NOTE

Les sources de données non-XA peuvent être intégrées dans les transactions JTA. Pour intégrer la source de données dans JTA, veillez à ce que le paramètre `jta` soit défini à `true`.

Procédure 6.5. Modifier une source de données non-XA

- ○ **Interface CLI**
 - a. [Section 3.5.2, « Lancement de l'interface CLI »](#).
 - b. Utiliser la commande `write-attribute` pour configurer un attribut de source de données :

```
/subsystem=datasources/data-source=DATASOURCE_NAME:write-attribute(name=ATTRIBUTE_NAME,value=ATTRIBUTE_VALUE)
```

c. Charger à nouveau le serveur pour confirmer les changements :

```
:reload
```

o **Console de gestion**

a. [Section 3.4.2, « Se connecter à la console de gestion »](#) .

b. **Naviguer dans le panneau Datasources qui se trouve dans la console de gestion**

- i. Sélectionner **Configuration** qui se trouve en haut de la console.
- ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.
- iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector**.
- iv. Sélectionner **Datasources** à partir du menu déroulant.

c. **Modifier la source de données**

- i. Sélectionner une source de données dans la liste **Available Datasources**. Les attributs de source de données sont affichés ci-dessous.
- ii. Cliquer sur **Edit** pour modifier les attributs de la source de données.
- iii. Cliquer sur le bouton **Save** pour terminer.

Résultat

La source de données non-Xa a été configurée. Elle est maintenant visible dans le fichier `standalone.xml` ou le fichier `domain.xml`, ainsi que dans les interfaces de gestion.

- Pour créer une nouvelle source de données. voir : [Section 6.3.1, « Créer une source de données non-XA avec les interfaces de gestion »](#).
- Pour supprimer la source de données, voir : [Section 6.3.3, « Supprimer une source de données non-XA par les interfaces de gestion »](#).

Rapporter un bogue

6.3.3. Supprimer une source de données non-XA par les interfaces de gestion

Résumé

Ce sujet couvre les étapes requises pour supprimer une source de données non-XA de JBoss EAP 6, en utilisant la console de gestion ou l'interface CLI.

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#).

Procédure 6.3.3. Supprimer une source de données non-XA

Procédure 6.6. Supprimer une source de données non-XA

- ○ **Interface CLI**
 - a. [Section 3.5.2, « Lancement de l'interface CLI »](#).
 - b. Exécuter la commande suivante pour supprimer une source de données non-XA :

```
data-source remove --name=DATASOURCE_NAME
```
- **Console de gestion**
 - a. [Section 3.4.2, « Se connecter à la console de gestion »](#).
 - b. **Naviguer dans le panneau Datasources qui se trouve dans la console de gestion**
 - i. Sélectionner **Configuration** qui se trouve en haut de la console.
 - ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.
 - iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector**.
 - iv. Sélectionner **Datasources**.
 - c. Sélectionner la source de données enregistrée à supprimer, et cliquer sur **Remove**.

Résultat

La source de données non-XA a été supprimée dans le serveur.

- Pour créer une nouvelle source de données, voir : [Section 6.3.1, « Créer une source de données non-XA avec les interfaces de gestion »](#).

[Rapporter un bogue](#)

6.4. SOURCES DE DONNÉES XA

6.4.1. Créer une source de données XA par les interfaces de gestion

Conditions préalables :

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#)

Résumé

Cette section explique les étapes à suivre pour créer une source de données XA, en utilisant la console de gestion ou l'interface CLI.



NOTE

Avant la version 10.2 de la source de données Oracle, le paramètre `<no-tx-separate-pools/>` était requis, car le mélange de connexions transactionnelles et non-transactionnelles aurait créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

Procédure 6.7. Créer une source de données XA en utilisant l'interface CLI ou la console de gestion

- ○ **Interface CLI**

- a. [Section 3.5.2, « Lancement de l'interface CLI »](#).
- b. Exécuter la commande d'interface de gestion CLI suivante pour créer une source de données XA, et configurer les variables comme il se doit :

**NOTE**

La valeur de *DRIVER_NAME* (nom de pilote) dépend du nombre de classes répertoriées dans le fichier */META-INF/services/java.sql.Driver* situé dans le JAR du pilote JDBC. S'il n'y a qu'une seule classe, la valeur correspondra au nom du JAR. S'il y a plusieurs classes, la valeur correspondra au nom du JAR + *driverClassName* + «» + *majorVersion* + «» + *minorVersion*. Toute erreur provoquera l'erreur suivante dans le journal :

```
JBAS014775: New missing/unsatisfied dependencies
```

Par exemple, la valeur de *DRIVER_NAME* qu'il nous faut pour le pilote MySQL 5.1.31 est `mysql-connector-java-5.1.31-bin.jarcom.mysql.jdbc.Driver_5_1`.

```
xa-data-source add --name=XA_DATASOURCE_NAME --jndi-name=JNDI_NAME --driver-name=DRIVER_NAME --xa-datasource-class=XA_DATASOURCE_CLASS
```

- c. **Configurer les propriétés de la source de données XA**

- i. **Définir le nom du serveur**

Exécuter la commande suivante pour configurer le nom du serveur de l'hôte :

```
/subsystem=datasources/xa-data-source=XA_DATASOURCE_NAME/xa-datasource-properties=ServerName:add(value=HOSTNAME)
```

- ii. **Définir le nom de la base de données**

Exécuter la commande suivante pour configurer le nom de la base de données :

```
/subsystem=datasources/xa-data-source=XA_DATASOURCE_NAME/xa-datasource-properties=DatabaseName:add(value=DATABASE_NAME)
```

- d. **Activer la source de données :**

```
xa-data-source enable --name=XA_DATASOURCE_NAME
```

- **Console de gestion**

- a. [Section 3.4.2, « Se connecter à la console de gestion »](#).

- b. Naviguer dans le panneau **Datasources** qui se trouve dans la console de gestion
 - i. Sélectionner **Configuration** qui se trouve en haut de la console.
 - ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.
 - iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector**.
 - iv. Sélectionner **Datasources**.
- c. Sélectionner l'onglet **XA Datasource**.
- d. **Créer une nouvelle source de données XA**
 - i. Cliquer sur **Add**.
 - ii. Saisir les attributs de la nouvelle source de données XA de l'assistant **Create XA Datasource** et cliquer sur **Next**.
 - iii. Saisir les informations sur le pilote JDBC dans l'assistant **Create XA Datasource** et cliquer sur **Next**.
 - iv. Saisir les propriétés XA et cliquer sur **Next**.
 - v. Saisir les paramètres de connexion dans l'assistant **Create XA Datasource**.
 - vi. Cliquer sur le bouton **Test Connection** pour tester la connexion à la ressource de données XA et vérifier que les paramètres de configuration soient corrects.
 - vii. Cliquer sur **Done** pour terminer.

Résultat

La source de données XA a été ajoutée au serveur. Elle est maintenant visible dans le fichier `standalone.xml` ou le fichier `domain.xml`, ainsi que dans les interfaces de gestion.

Voir également :

- [Section 6.4.2, « Modifier une Source de données XA par les Interfaces de gestion »](#)
- [Section 6.4.3, « Supprimer une Source de données XA par les Interfaces de gestion »](#)

Rapporter un bogue

6.4.2. Modifier une Source de données XA par les Interfaces de gestion

Résumé

Cette section explique les étapes à suivre pour modifier une source de données XA, en utilisant la console de gestion ou l'interface CLI.

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#).

Procédure 6.8. Modifier une source de données XA en utilisant l'interface CLI ou la console de gestion

- ○ L'interface CLI

- a. [Section 3.5.2, « Lancement de l'interface CLI »](#).

- b. **Configurer les attributs de source de données XA**

Utiliser la commande `write-attribute` pour configurer un attribut de source de données :

```
/subsystem=datasources/xa-data-
source=XA_DATASOURCE_NAME:write-
attribute(name=ATTRIBUTE_NAME,value=ATTRIBUTE_VALUE)
```

- c. **Configurer les propriétés de la source de données XA**

Exécuter la commande suivante pour configurer une sous-ressource de source de données XA :

```
/subsystem=datasources/xa-data-source=DATASOURCE_NAME/xa-
datasource-properties=PROPERTY_NAME:add(value=PROPERTY_VALUE)
```

- d. Charger à nouveau le serveur pour confirmer les changements :

```
:reload
```

- Console de gestion

- a. [Section 3.4.2, « Se connecter à la console de gestion »](#) .

- b. **Naviguer dans le panneau Datasources qui se trouve dans la console de gestion**

- i. Sélectionner **Configuration** qui se trouve en haut de la console.

- ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.

- iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector** .

- iv. Sélectionner **Datasources**.

- c. Sélectionner l'onglet **XA Datasource**.

- d. **Modifier la source de données**

- i. Sélectionner la source de données XA qui convient à partir de la liste **Available XA Datasources**. Les attributs de la source de données XA sont affichés dans le panneau **Attributes** ci-dessous.

- ii. Sélectionner le bouton **Edit** pour modifier les attributs de la source de données.

- iii. Modifier les attributs de la source de données XA et sélectionner le bouton **Save** quand c'est fait.

Résultat

La source de données XA a été configurée. Les changements sont maintenant visibles dans le fichier `standalone.xml` ou le fichier `domain.xml`, ainsi que dans les interfaces de gestion.

- Pour créer une nouvelle source de données, voir : [Section 6.4.1, « Créer une source de données XA par les interfaces de gestion »](#).
- Pour supprimer la source de données, voir : [Section 6.4.3, « Supprimer une Source de données XA par les Interfaces de gestion »](#).

[Rapporter un bogue](#)

6.4.3. Supprimer une Source de données XA par les Interfaces de gestion

Résumé

Ce sujet couvre les étapes requises pour supprimer une source de données XA de JBoss EAP 6, en utilisant la console de gestion ou l'interface CLI.

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#).

Procédure 6.9. Supprimer une source de données XA en utilisant l'interface CLI ou la console de gestion

- ○ **Management CLI**
 - a. [Section 3.5.2, « Lancement de l'interface CLI »](#).
 - b. Exécuter la commande suivante pour supprimer une source de données :

```
xa-data-source remove --name=XA_DATASOURCE_NAME
```
- **Console de gestion**
 - a. [Section 3.4.2, « Se connecter à la console de gestion »](#).
 - b. **Naviguer dans le panneau Datasources qui se trouve dans la console de gestion**
 - i. Sélectionner **Configuration** qui se trouve en haut de la console.
 - ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.
 - iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector**.
 - iv. Sélectionner **Datasources**.
 - c. Sélectionner l'onglet **XA Datasource**.
 - d. Sélectionner la source de données XA enregistrée à supprimer, et cliquer sur le bouton **Remove** pour supprimer la source de données XA de façon permanente.

Résultat

La source de données XA a été supprimée dans le serveur.

- Pour ajouter une nouvelle source de données, voir : [Section 6.4.1, « Créer une source de données XA par les interfaces de gestion »](#).

[Rapporter un bogue](#)

6.4.4. XA Recovery

6.4.4.1. Les modules de recouvrement XA

Chaque ressource XA a besoin d'un module de recouvrement associé avec sa configuration. Le module de recouvrement doit étendre la classe `com.arjuna.ats.jta.recovery.XAResourceRecovery`.

JBoss EAP 6 fournit des modules de recouvrement pour les ressources JDBC et JMS XA. Pour ces types de ressources, les modules de recouvrement sont automatiquement enregistrés. Si vous avez besoin d'utiliser un module personnalisé, vous pouvez l'enregistrer dans votre source de données.

[Rapporter un bogue](#)

6.4.4.2. Configurer les modules de recouvrement

Pour la plupart des ressources JDBC et JMS, le module de recouvrement est automatiquement associé à la ressource. Dans de tels cas, vous aurez uniquement besoin de configurer les options qui permettent au module de recouvrement de se connecter à vos ressources afin de procéder au recouvrement.

Pour les ressources personnalisées qui ne sont ni JDBC, ni JMS, contacter Red Hat Global Services pour obtenir des informations sur les configurations qui sont prises en charge.

Chacun de ces attributs de configuration peut être défini lors de la création de la source de données, ou par la suite. Vous pouvez les définir en utilisant la console de gestion basée web ou l'interface CLI par ligne de commande. Se référer à [Section 6.4.1, « Créer une source de données XA par les interfaces de gestion »](#) and [Section 6.4.2, « Modifier une Source de données XA par les Interfaces de gestion »](#) pour obtenir des informations sur la façon de configurer des sources de données XA.

Voir les tableaux suivants pour les attributs de configuration de sources de données, et pour obtenir des informations sur la configuration spécifique à certains fournisseurs de bases de données.

Tableau 6.2. Attributs de configuration générale

Attribut	Description
recovery-username	Le nom d'utilisateur qui doit être utilisé par le module de recouvrement pour se connecter à la ressource de recouvrement.
recovery-password	Le mot de passe qui doit être utilisé par le module de recouvrement pour se connecter à la ressource de recouvrement.
recovery-security-domain	Le domaine de sécurité qui doit être utilisé par le module de recouvrement pour se connecter à la ressource de recouvrement.
recovery-plugin-class-name	Si vous devez utiliser un module de recouvrement personnalisé, définissez cet attribut au nom complet de classe du module. Le module doit étendre la classe <code>com.arjuna.ats.jta.recovery.XAResourceRecovery</code> .

Attribut	Description
recovery-plugin-properties	Si vous utilisez un module de récupération personnalisée qui requiert des propriétés à définir, définissez cet attribut à la liste de paires <i>key=value</i> séparée par des virgules pour les propriétés.

Informations de configuration spécifiques au fournisseur

Oracle

Si la source de données Oracle n'est pas configurée correctement, vous apercevrez sans doute les erreurs suivantes dans votre sortie de journalisation :

Exemple 6.9. Erreur de la configuration incorrecte

```
WARN [com.arjuna.ats.jta.logging.loggerI18N]
[com.arjuna.ats.internal.jta.recovery.xarecovery1] Local
XARecoveryModule.xaRecovery got XA exception
javax.transaction.xa.XAException, XAException.XAER_RMERR
```

Pour résoudre cette erreur, veillez à ce que l'utilisateur Oracle configuré dans *recovery-username* ait bien accès aux tableaux utiles au recouvrement. L'énoncé SQL suivant affiche les attributions d'instances correctes Oracle 11g ou Oracle 10g R2 corrigées pour le bogue 5945463 d'Oracle.

Exemple 6.10. Octroi de privilège pour la configuration

```
GRANT SELECT ON sys.dba_pending_transactions TO recovery-username;
GRANT SELECT ON sys.pending_trans$ TO recovery-username;
GRANT SELECT ON sys.dba_2pc_pending TO recovery-username;
GRANT EXECUTE ON sys.dbms_xa TO recovery-username;
```

Si vous utilisez une version Oracle 11g antérieure à 11g, modifier l'énoncé final **EXECUTE** ainsi:

```
GRANT EXECUTE ON sys.dbms_system TO recovery-username;
```

PostgreSQL

Voir la documentation PostgreSQL pour obtenir des instructions sur la façon d'activer des transactions (ex. XA) préparées. La version 8.4-701 du pilote JDBC de PostgreSQL a un bogue dans `org.postgresql.xa.PGXACConnection` qui empêche le recouvrement dans certaines situations. Cela a été résolu dans les nouvelles versions.

MySQL

Basé sur <http://bugs.mysql.com/bug.php?id=12161>, le recouvrement de transactions XA ne fonctionnait pas dans certaines versions de MySQL 5. Cela a été corrigé dans MySQL 6.1. Voir l'URL du bogue ou la documentation MySQL pour obtenir davantage d'informations.

IBM DB2

IBM DB2 s'attend à ce que la méthode `XAResource.recover` soit appelée uniquement pendant la phase de resynchronisation désignée qui se produit lorsque le serveur d'applications est redémarré après un accident ou une panne. Il s'agit d'une décision de conception pour l'implémentation de DB2, qui est en dehors du dessein de cette documentation.

Sybase

Sybase s'attend à ce que les transactions XA soient activées sur la base de données. Sans configuration correcte de la base de données, les transactions XA ne fonctionneront pas. `enable xact coordination` active ou désactive les services de coordination de transaction Adaptive Server. Lorsque ce paramètre est activé, Adaptive Server garantit que les mises à jour de données Adaptive Server distantes soient validées ou annulées avec la transaction d'origine. Pour permettre la coordination de transaction, utilisez :

```
sp_configure 'enable xact coordination', 1
```

Rapporter un bogue

6.5. SÉCURITÉ DES BASES DE DONNÉES

6.5.1. Sécurité des bases de données

La notion de sécurité des sources de données se réfère à l'encodage ou au déguisement des mots de passe pour les connexions aux sources de données. Les mots de passe peuvent être stockés au format texte brut dans les fichiers de configuration, mais cela représente un risque de sécurité.

La meilleure solution de sécurité de source de données est soit l'utilisation des domaines de sécurité, soit les mots de passe. Vous trouverez un exemple de chacun ci-dessous. Pour plus d'informations, consulter *Security Architecture* ou les autres documentations de sécurité JBoss.

Exemple 6.11. Exemple de domaine de sécurité

```
<security-domain name="DsRealm" cache-type="default">
  <authentication>
    <login-module code="ConfiguredIdentity" flag="required">
      <module-option name="userName" value="sa"/>
      <module-option name="principal" value="sa"/>
      <module-option name="password" value="sa"/>
    </login-module>
  </authentication>
</security-domain>
```

Le domaine `DsRealm` est référencé par une source de données comme :

```
<datasources>
  <datasource jndi-name="java:jboss/datasources/securityDs"
    pool-name="securityDs">
    <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1</connection-url>
    <driver>h2</driver>
    <new-connection-sql>select current_user()</new-connection-sql>
    <security>
```

```

        <security-domain>DsRealm</security-domain>
    </security>
</datasource>
</datasources>

```

Exemple 6.12. Exemple de mots de passe

```

<security>
  <user-name>admin</user-name>

  <password>${VAULT::ds_ExampleDS::password::N2NhZDYzOTMtNWE0OS00ZGQ0LWE4M
mEtMWNlMDMyNDdmNmI2TElORV9CUkVBS3ZhdWx0}</password>
</security>

```

[Rapporter un bogue](#)

6.6. VALIDATION DE LA CONNEXION À LA BASE DE DONNÉES

6.6.1. Configuration des paramètres de validation de connexion de la base de données

Aperçu

En raison de problèmes de maintenance de base de données, de problèmes de réseau ou autres événements peuvent amener JBoss EAP 6 à perdre la connexion à la base de données. Vous activez la validation de la connexion à la base de données par l'élément `< validation >` dans la section `< datasource >` du fichier de configuration du serveur. Suivez les étapes ci-dessous pour configurer les paramètres de source de données pour activer la validation de connexion de la base de données dans JBoss EAP 6.

Procédure 6.10. Configuration des paramètres de validation de connexion de la base de données

1. Choisissez une méthode de validation

Veuillez sélectionner une des méthodes de validation suivantes.

- o `<validate-on-match>true</validate-on-match>`

Lorsque l'option `< validate-on-match >` a comme valeur `true`, la connexion à la base de données est validée à chaque retrait du pool de connexions en utilisant le mécanisme de validation spécifié à l'étape suivante.

Si une connexion n'est pas valide, un avertissement sera inscrit dans le journal et la prochaine connexion sera extraite du pool. Ce processus se poursuit jusqu'à ce qu'une connexion valide soit enfin trouvée. Si vous préférez ne pas faire défiler chaque connexion du pool, vous pouvez utiliser l'option `<use-fast-fail>`. Si on ne trouve pas de connexion valide dans le pool, une nouvelle connexion sera créée. Si la création de la connexion échoue, une exception sera retournée à l'application qui en a fait la demande.

Ce paramètre entraîne une récupération plus rapide, mais crée une charge plus élevée sur la base de données. Cependant, c'est la sélection la plus sûre si une baisse de performance minimale n'est pas un sujet de préoccupation.

- **<background-validation>true</background-validation>**
Lorsque l'option `< background-validation` a comme valeur `true`, il est utilisé en combinaison à la valeur `<background-validation-millis>` pour déterminer la fréquence d'exécution de l'information en d'arrière-plan. La valeur par défaut du paramètre `<background-validation-millis>` est de 0 millisecondes, ce qui signifie que c'est désactivé par défaut. Cette valeur ne doit pas être à la même valeur que celle du paramètre `< idle-timeout-minutes >`.

Il est délicat de déterminer la valeur optimale de `<background-validation-millis>` pour un système particulier. Plus la valeur est faible, le plus fréquemment le pool sera validé et le plus rapidement les connexions plus tôt non valides sont supprimées de la piscine. Cependant, des valeurs les plus faibles prennent davantage de ressources de base de données. En outre, les valeurs élevées ont des contrôles de validation de connexion plus fréquents et utilisent moins de ressources de base de données, mais les liens morts demeurent indétectables pendant de plus longues périodes.



NOTE

Si l'option `<validate-on-match>` est définie à `true`, l'option `<background-validation>` devra être définie à `false`. Le contraire est vrai également. Si l'option `<background-validation>` est définie à `true`, l'option `<validate-on-match>` devra être définie à `false`.

2. Choisir un mécanisme de validation

Veuillez sélectionner un des mécanismes de validation suivants.

- **Spécifier un Nom de classe `<valid-connection-checker>`**
Il s'agit du mécanisme préféré car il est optimisé pour un RBMS particulier. JBoss EAP 6 fournit les vérificateurs de connexions suivants :
 - `org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker`
 - `org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker`
 - `org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLReplicationValidConnectionChecker`
 - `org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker`
 - `org.jboss.jca.adapters.jdbc.extensions.novendor.JDBC4ValidConnectionChecker`
 - `org.jboss.jca.adapters.jdbc.extensions.novendor.NullValidConnectionChecker`
 - `org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker`
 - `org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker`
 - `org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectionChecker`
- **Indiqué l'énoncé SQL pour `<check-valid-connection-sql>`**
Vous fournissez l'énoncé SQL utilisé pour valider la connexion.

Ce qui suit est un exemple qui vous montre comment spécifier un énoncé SQL pour valider une connexion dans Oracle :

```
<check-valid-connection-sql>select 1 from dual</check-valid-connection-sql>
```

Dans MySQL ou PostgreSQL, vous pouvez spécifier l'énoncé SQL suivant :

```
<check-valid-connection-sql>select 1</check-valid-connection-sql>
```

3. Définir le Nom de classe <exception-sorter>

Lorsqu'une exception est marquée comme étant fatale, la connexion est fermée immédiatement, même si la connexion participe à une transaction. Utilisez l'option de classe de triage d'exception trieuse pour détecter correctement et ensuite nettoyer les exceptions de connexion fatales. JBoss EAP 6 fournit les trieurs d'exception suivants :

- o org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter
- o org.jboss.jca.adapters.jdbc.extensions.informix.InformixExceptionSorter
- o org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter
- o org.jboss.jca.adapters.jdbc.extensions.novendor.NullExceptionSorter
- o org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter
- o org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter
- o org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorter

[Rapporter un bogue](#)

6.7. CONFIGURATION DES SOURCES DE DONNÉES

6.7.1. Paramètres de source de données

Tableau 6.3. Les paramètres de source de données communs aux sources XA ou non-XA

Paramètre	Description
jndi-name	Le nom JNDI unique pour la source de données.
pool-name	Le nom du pool de gestion de la source de données.
enabled	Indique si la source de données est activée.
use-java-context	Indique si on doit relier la source de données au JNDI global.
spy	Activer la fonctionnalité spy sur la couche JDBC. Cela journalisera tout le trafic JDBC dans la source de données. Notez que la catégorie de journalisation jboss.jdbc.spy doit également être définie au niveau DEBUG dans le sous-système de journalisation.

Paramètre	Description
use-ccm	Activer le gestionnaire de connexion cache.
new-connection-sql	Un énoncé SQL qui exécute quand la connexion est ajoutée au pool de connexion.
transaction-isolation	Un parmi : <ul style="list-style-type: none"> • TRANSACTION_READ_UNCOMMITTED • TRANSACTION_READ_COMMITTED • TRANSACTION_REPEATABLE_READ • TRANSACTION_SERIALIZABLE • TRANSACTION_NONE
url-selector-strategy-class-name	Une classe qui implémente l'interface <code>org.jboss.jca.adapters.jdbc.URLSelectorStrategy</code> .
sécurité	Contient des éléments dépendants en tant que paramètres de sécurité. Voir Tableau 6.8 , « Paramètres de sécurité ».
validation	Contient des éléments dépendants en tant que paramètres de validation. Voir Tableau 6.9 , « Paramètres de validation ».
timeout	Contient des éléments dépendants en tant que paramètres de timeout. Voir Tableau 6.10 , « Paramètres de timeout ».
énoncé	Contient des éléments dépendants en tant que paramètres d'énoncés. Voir Tableau 6.11 , « Paramètres d'instruction ».

Tableau 6.4. Paramètres de source de données non-xa

Paramètre	Description
jta	Active l'intégration JTA pour les sources de données non-XA. Ne s'applique pas aux sources de données XA.
connection-url	L'URL de connexion du pilote JDBC.
driver-class	Le nom complet de la classe de pilote JDBC.

Paramètre	Description
connection-property	Propriétés de connexions arbitraires passées à la méthode <code>Driver.connect(url, props)</code> . Chaque connection-property indique une paire name/value. Le nom de la propriété provient du nom, et la valeur provient du contenu de l'élément.
pool	Contient des éléments dépendants en tant que paramètres de pooling. Voir Tableau 6.6, « Les paramètres de pool communs aux sources XA ou non-XA » .
url-delimiter	Le délimiteur d'URLs d'une connexion url pour les bases de données clusterisées HA (Haute disponibilité).

Tableau 6.5. Paramètres de source de données XA

Paramètre	Description
xa-datasource-property	Une propriété pour assigner la classe d'implémentation <code>XDataSource</code> . Spécifiée par <code>name=value</code> . Si une méthode setter existe, dans le format <code>setName</code> , la propriété sera définie en appelant une méthode setter sous le format <code>setName(value)</code> .
xa-datasource-class	Le nom complet de la classe d'implémentation de <code>javax.sql.XDataSource</code> .
pilote	Unique référence au module de chargeur de classe qui contient le pilote JDBC. Le format accepté est <code>driverName#majorVersion.minorVersion</code> .
xa-pool	Contient des éléments dépendants en tant que paramètres de pooling. Voir Tableau 6.6, « Les paramètres de pool communs aux sources XA ou non-XA » et Tableau 6.7, « Paramètres du pool XA » .
recouvrement	Contient des éléments dépendants en tant que paramètres de recouvrement. Voir Tableau 6.12, « Paramètres de recouvrement » .

Tableau 6.6. Les paramètres de pool communs aux sources XA ou non-XA

Paramètre	Description
min-pool-size	Le nombre minimum de connexions contenues par un pool.

Paramètre	Description
max-pool-size	Le nombre maximum de connexions qu'un pool peut contenir
Pré-remplissage	Indique si l'on doit essayer de pré-remplir un pool de connexion. Un élément vide indique une valeur true . La valeur par défaut est false .
use-strict-min	Indique si la taille du pool est stricte. false par défaut.
flush-strategy	Indique si le pool est vidé en cas d'erreur. Les valeurs acceptées sont : <ul style="list-style-type: none"> • FailingConnectionOnly • IdleConnections • EntirePool La valeur par défaut est FailingConnectionOnly .
allow-multiple-users	Indique si plusieurs utilisateurs pourront avoir accès à la source de données par la méthode <code>getConnection(user, password)</code> , et si les types de pools internes ont une influence sur ce comportement.

Tableau 6.7. Paramètres du pool XA

Paramètre	Description
is-same-rm-override	Indique si la classe <code>javax.transaction.xa.XAResource.isSameRM(XAResource)</code> retourne true ou false .
entrelacement	Indique si on doit activer l'entrelacement pour les fabriques de connexion XA.
no-tx-separate-pools	Indique si on doit créer des sous-répertoires distincts pour chaque contexte. Cela est nécessaire pour les sources de données Oracle, qui ne permettent pas aux connexions XA d'être utilisées à la fois à l'intérieur et à l'extérieur d'une transaction de JTA Utiliser cette option entraînera la multiplication par deux de la taille du pool max-pool-size , car en fait, deux pools seront créés.
pad-xid	Indique si on doit remplir le Xid.

Paramètre	Description
wrap-xa-resource	Indique si on doit inclure XAResource dans une instance <code>org.jboss.tm.XAResourceWrapper</code> .

Tableau 6.8. Paramètres de sécurité

Paramètre	Description
user-name	Le nom d'utilisation pour créer une nouvelle connexion.
password	Le mot de passe à utiliser pour créer une nouvelle connexion
security-domain	Contient le nom d'un gestionnaire de sécurité JAAS, qui gère l'authentification. Ce nom correspond à l'attribut <code>application-policy/name</code> de la configuration de connexion JAAS.
reauth-plugin	Définit un plugin d'authentification à nouveau pour la réauthentification de connexions physiques.

Tableau 6.9. Paramètres de validation

Paramètre	Description
valid-connection-checker	Une mise en œuvre d'interface <code>org.jboss.jca.adapters.jdbc.ValidConnectionChecker</code> qui fournit une méthode <code>SQLException.isValidConnection(Connection e)</code> pour valider une connexion. Une exception signifie que la connexion est détruite. Cela remplace le paramètre <code>check-valid-connection-sql</code> s'il est présent.
check-valid-connection-sql	Un énoncé SQL pour vérifier la validité d'un pool de connexion. Peut être appelé quand une connexion gérée est tirée d'un pool.
validate-on-match	Indique si la validation de niveau de connexion est exécutée lorsqu'une fabrique de connexions essaie de correspondre à une connexion gérée pour un ensemble donné. Indiquer "true" pour <code>validate-on-match</code> n'est pas normalement fait en conjonction avec "true" pour <code>background-validation</code> . <code>Validate-on-match</code> est utile quand un client doit avoir une connexion validée avant utilisation. Ce paramètre est à false par défaut.

Paramètre	Description
background-validation	Indique que les connexions sont validées sur un thread d'arrière-plan. La validation de l'arrière-plan (Background validation) est une optimisation de performance lorsque non utilisé avec validate-on-match . Si Validate-on-match est sur true , l'utilisation de background-validation pourrait entraîner des contrôles redondants. La validation de l'arrière-plan pourrait provoquer une mauvaise connexion (une connexion qui irait mal entre le moment de l'analyse de validation et avant d'être donnée au client), l'application cliente doit par conséquent tenir compte de cette possibilité.
background-validation-millis	La durée, en millisecondes, pendant laquelle la validation d'arrière-plan exécute.
use-fast-fail	Si défini sur true , échoue une allocation de connexion lors de la première tentative, si la connexion est non valide. La valeur par défaut est false .
stale-connection-checker	Une instance de org.jboss.jca.adapters.jdbc.StaleConnectionChecker qui produit une méthode booléenne isStaleConnection(SQLException e) . Si cette méthode renvoie un true , l'exception sera contenue dans org.jboss.jca.adapters.jdbc.StaleConnectionException , qui correspond à une sous-classe de SQLException .
exception-sorter	Une instance de org.jboss.jca.adapters.jdbc.ExceptionSorter qui fournit une méthode booléenne isExceptionFatal(SQLException e) . Cette méthode validera si une exception est envoyée à toutes les instances d'un javax.resource.spi.ConnectionEventListener en tant que message connectionErrorOccurred .

Tableau 6.10. Paramètres de timeout

Paramètre	Description
-----------	-------------

Paramètre	Description
use-try-lock	Utiliser <code>tryLock()</code> au lieu de <code>lock()</code> . Vous essayerez ainsi d'obtenir un verrou pour le nombre de secondes configurées, avant le timeout, au lieu d'échouer immédiatement quand le verrou n'est pas disponible. La valeur par défaut est de 60 secondes. Par exemple, pour définir un timeout de 5 minutes, définir <code><use-try-lock>300</use-try-lock></code> .
blocking-timeout-millis	La durée maximale, en millisecondes, de blocage lorsque vous attendez une connexion. Après que ce délai soit dépassé, une exception sera levée. Cela bloque uniquement pendant que vous attendez un permis de connexion et ne lève pas d'exception si la création d'une nouvelle connexion prend beaucoup de temps. Par défaut, 30000, qui correspond à 30 secondes.
idle-timeout-minutes	La durée maximale, en minutes, avant qu'une connexion inactive soit fermée. La durée maximale réelle dépend de la durée d'analyse de <code>idleRemover</code> , qui correspond à la moitié du plus petit <code>idle-timeout-minutes</code> de n'importe quel pool.
set-tx-query-timeout	Indique si on doit définir le timeout d'interrogation par rapport au temps qui reste avant le timeout de transaction. Si aucune transaction n'existe, on utilisera le timeout de recherche qui a été configuré. La valeur par défaut est false .
query-timeout	Timeout pour les recherches, en secondes. La valeur par défaut est « no timeout ».
allocation-retry	Le nombre de tentatives de connexions avant d'envoyer une connexion. La valeur par défaut est 0 , pour qu'une exception puisse être envoyée à la première défaillance.
allocation-retry-wait-millis	Le temps, en millisecondes, qu'il faut attendre avant de retenter d'allouer une connexion. La valeur par défaut est 5 000, soit 5 secondes.
xa-resource-timeout	Si la valeur est non nulle, elle passe à la méthode <code>XAResource.setTransactionTimeout</code> .

Tableau 6.11. Paramètres d'instruction

Paramètre	Description
-----------	-------------

Paramètre	Description
track-statements	<p>Indique si l'on doit vérifier les instructions non fermées lorsqu'une connexion est renvoyée à un pool ou qu'une instruction est retournée dans le cache d'instruction préparée. Si false, les instructions ne seront pas suivies.</p> <p>Valeurs valides</p> <ul style="list-style-type: none"> • true : les instructions et les ensembles de résultats sont suivis, et un avertissement sera émis s'ils ne sont pas fermés. • false : ni les instructions, ni les ensembles de résultats ne seront suivis. • nowarn : les instructions sont suivies, mais il n'y a aucun avertissement. Valeur par défaut.
prepared-statement-cache-size	Le nombre d'instructions préparées par connexion, dans le cache LRU (le moins souvent utilisé récemment).
share-prepared-statements	Indique si le fait de demander la même instruction deux fois sans la fermer utilise la même instruction préparée sous-jacente. La valeur par défaut est false .

Tableau 6.12. Paramètres de recouvrement


Paramètre	Description
recover-credential	Une paire nom d'utilisateur/mot de passe ou domaine de sécurité pour le recouvrement.
recover-plugin	Une mise en œuvre de la classe <code>org.jboss.jca.core.spi.recoveryRecoveryPlugin</code> à utiliser pour le recouvrement.

[Rapporter un bogue](#)

6.7.2. Les URL de connexion de sources de données

Tableau 6.13. Les URL de connexion de sources de données

Source de données	URL de connexion
-------------------	------------------

Source de données	URL de connexion
PostgreSQL	<code>jdbc:postgresql://SERVER_NAME:PORT/DATABASE_NAME</code>
MySQL	<code>jdbc:mysql://SERVER_NAME:PORT/DATABASE_NAME</code>
Oracle	<code>jdbc:oracle:thin:@ORACLE_HOST:PORT:ORACLE_SID</code>
IBM DB2	<code>jdbc:db2://SERVER_NAME:PORT/DATABASE_NAME</code>
Microsoft SQLServer	<code>jdbc:sqlserver://SERVER_NAME:PORT;DatabaseName=DATABASE_NAME</code>  <p>NOTE</p> <p>Le modèle <code>jdbc:microsoft:sqlserver://SERVER_NAME:PORT;DatabaseName=DATABASE_NAME</code> ne fonctionne pas avec la nouvelle base de données.</p>

[Rapporter un bogue](#)

6.7.3. Extensions de sources de données

Les déploiements de sources de données peuvent utiliser plusieurs extensions de l'adaptateur de ressources JDBC pour améliorer la validation de la connexion, et vérifier si l'exception doit rétablir la connexion. Ces extensions sont les suivantes :

Tableau 6.14. Extensions de sources de données

Extension de sources de données	Paramètre de configuration	Description
<code>org.jboss.jca.adapters.jdbc.spi.ExceptionSorter</code>	<code><exception-sorter></code>	Vérifie si <code>SQLException</code> est fatale pour la connexion sur laquelle l'exception a été lancée
<code>org.jboss.jca.adapters.jdbc.spi.StaleConnection</code>	<code><stale-connection-checker></code>	Wraps stale <code>SQLExceptions</code> in a <code>org.jboss.jca.adapters.jdbc.StaleConnectionException</code>
<code>org.jboss.jca.adapters.jdbc.spi.ValidConnection</code>	<code><valid-connection-checker></code>	Vérifie si une connexion est valide pour être utilisée par l'application

JBoss EAP 6 comprend également des implémentations de ces extensions pour plusieurs bases de données prises en charge.

Implémentations des extensions

Générique

- `org.jboss.jca.adapters.jdbc.extensions.novendor.NullExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.novendor.NullStaleConnectionChecker`
- `org.jboss.jca.adapters.jdbc.extensions.novendor.NullValidConnectionChecker`
- `org.jboss.jca.adapters.jdbc.extensions.novendor.JDBC4ValidConnectionChecker`

PostgreSQL

- `org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker`

MySQL

- `org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLReplicationValidConnectionChecker`
- `org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker`

IBM DB2

- `org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionChecker`
- `org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker`

Sybase

- `org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectionChecker`

Microsoft SQLServer

- `org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker`

Oracle

- `org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker`
- `org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker`

[Rapporter un bogue](#)

6.7.4. Voir les statistiques de sources de données

Vous pourrez voir des statistiques de sources de données définies dans `jdbc` et `pool` qui utilisent des versions modifiées des commandes ci-dessous :

Exemple 6.13. Obtention des statistiques de sources de données

```
/subsystem=datasources/data-source=ExampleDS/statistics=jdbc:read-resource(include-runtime=true)
```

ou

```
/subsystem=datasources/data-source=ExampleDS/statistics=pool:read-resource(include-runtime=true)
```



NOTE

Veillez à spécifier l'argument *include-runtime=true* car tous les statistiques sont en runtime uniquement et la valeur par défaut est *false*.

[Rapporter un bogue](#)

6.7.5. Statistiques de bases de données

Statistiques principaux

Le tableau suivant montre une liste des statistiques principaux de sources de données pris en charge :

Tableau 6.15. Statistiques principaux

Nom	Description
ActiveCount	Le nombre de connexions actives. Chacune de ces connexions est soit utilisée par une application, ou disponible via pool
AvailableCount	Le nombre de connexions disponibles dans le pool
AverageBlockingTime	Le durée moyenne passée à bloquer l'obtention d'un verrou exclusif sur le pool. La valeur est en millisecondes.
AverageCreationTime	Le durée moyenne passée à créer une connexion. La valeur est en millisecondes.
CreatedCount	Le nombre de connexions créées.
DestroyedCount	Le nombre de connexions détruites.
InUseCount	Le nombre de connexions actuellement utilisées.

Nom	Description
MaxCreationTime	La durée maximum pour créer une connexion. La valeur est en millisecondes.
MaxUsedCount	Le nombre maximum de connexions utilisées
MaxWaitCount	Le nombre maximum de requêtes attendant une connexion en même temps.
MaxWaitTime	Le durée maximum à attendre un verrou exclusif sur le pool.
TimedOut	Le nombre de connexions expirées
TotalBlockingTime	Le durée à attendre un verrou exclusif sur le pool. La valeur est en millisecondes.
TotalCreationTime	La durée passée à créer des connexions. La valeur est en millisecondes.
WaitCount	Le nombre de requêtes en attente de connexion.

Statistiques JDBC

Le tableau suivant montre une liste des statistiques JDBC de sources de données pris en charge :

Tableau 6.16. Statistiques JDBC

Nom	Description
PreparedStatementCacheAccessCount	Le nombre de fois qu'un cache d'énoncé a été accédé.
PreparedStatementCacheAddCount	Le nombre d'énoncés ajoutés au cache de l'énoncé.
PreparedStatementCacheCurrentSize	Le nombre d'énoncés préparés et que l'on peut appeler, actuellement mis en cache dans un cache d'énoncé.
PreparedStatementCacheDeleteCount	Le nombre d'énoncés rejetés du cache.
PreparedStatementCacheHitCount	Le nombre de fois que des énoncés de cache ont été utilisés.
PreparedStatementCacheMissCount	Le nombre de fois qu'une requête d'énoncé a pu être réglée par un énoncé d'un cache.

Vous pouvez activer les statistiques **Core** et **JDBC** en utilisant des versions appropriément modifiées des commandes suivantes :

- `/subsystem=datasources/data-source=ExampleDS/statistics=pool:write-attribute(name=statistics-enabled,value=true)`
- `/subsystem=datasources/data-source=ExampleDS/statistics=jdbc:write-attribute(name=statistics-enabled,value=true)`

[Rapporter un bogue](#)

6.8. EXEMPLES DE SOURCES DE DONNÉES

6.8.1. L'exemple de source de données PostgreSQL

L'exemple ci-dessous est une configuration de source de données PostgreSQL. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.14. Configuration des sources de données PostSQL

```
<datasources>
  <datasource jndi-name="java:jboss/PostgresDS" pool-name="PostgresDS">
    <connection-
url>jdbc:postgresql://localhost:5432/postgresdb</connection-url>
    <driver>postgresql</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <background-validation-millis>60000</background-validation-
millis>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidCon
nectionChecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptio
nSorter"></exception-sorter>
    </validation>
  </datasource>
</drivers>
  <driver name="postgresql" module="org.postgresql">
    <xa-datasource-class>org.postgresql.xa.PGXADatasource</xa-
datasource-class>
  </driver>
</drivers>
</datasources>
```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données PostgreSQL ci-dessus.

Exemple 6.15. `module.xml`

```

<module xmlns="urn:jboss:module:1.1" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-9.1-902.jdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>

```

[Rapporter un bogue](#)

6.8.2. Exemple de source de données PostgreSQL XA

L'exemple ci-dessous est une configuration de source de données PostgreSQL XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.16. Source de données PostgreSQL XA

```

<datasources>
  <xa-datasource jndi-name="java:jboss/PostgresXADS" pool-
name="PostgresXADS">
    <driver>postgresql</driver>
    <xa-datasource-property name="ServerName">localhost</xa-datasource-
property>
    <xa-datasource-property name="PortNumber">5432</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">postgresdb</xa-
datasource-property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <background-validation-millis>60000</background-validation-
millis>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidCon
nectionChecker">
      </valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptio
nSorter">
      </exception-sorter>
    </validation>
  </xa-datasource>
  <drivers>
    <driver name="postgresql" module="org.postgresql">
      <xa-datasource-class>org.postgresql.xa.PGXADDataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>

```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données PostgreSQL XA ci-dessus.

Exemple 6.17. module.xml

```
<module xmlns="urn:jboss:module:1.1" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-9.1-902.jdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

[Rapporter un bogue](#)

6.8.3. Exemple de source de données MySQL

L'exemple ci-dessous est une configuration de source de données MySQL. La source de données a été activée, un utilisateur a été ajouté, et des options de validation ont été définies.

Exemple 6.18. Configuration de source de données MySQL

```
<datasources>
  <datasource jndi-name="java:jboss/MySqlDS" pool-name="MySqlDS">
    <connection-url>jdbc:mysql://mysql-localhost:3306/jbossdb</connection-url>
    <driver>mysql</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <background-validation-millis>60000</background-validation-millis>
      <valid-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker"></valid-connection-checker>
      <exception-sorter class-name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter"></exception-sorter>
    </validation>
  </datasource>
  <drivers>
    <driver name="mysql" module="com.mysql">
      <xa-datasource-class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-class>
    </driver>
  </drivers>
</datasources>
```

```

    </driver>
  </drivers>
</datasources>

```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données MySQL ci-dessus.

Exemple 6.19. `module.xml`

```

<module xmlns="urn:jboss:module:1.1" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.0.8-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>

```

[Rapporter un bogue](#)

6.8.4. Exemple de source de données MySQL XA

L'exemple ci-dessous est une configuration de source de données MySQL XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.20. Source de données MySQL XA

```

<datasources>
  <xa-datasource jndi-name="java:jboss/MysqlXADS" pool-name="MysqlXADS">
    <driver>mysql</driver>
    <xa-datasource-property name="ServerName">localhost</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">mysqldb</xa-datasource-
property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <background-validation-millis>60000</background-validation-
millis>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionC
hecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter"
></exception-sorter>
    </validation>
  </xa-datasource>
</drivers>
  <driver name="mysql" module="com.mysql">

```

```

    <xa-datasource-
class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-
class>
    </driver>
</drivers>
</datasources>

```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données MySQL XA ci-dessus.

Exemple 6.21. `module.xml`

```

<module xmlns="urn:jboss:module:1.1" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.0.8-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>

```

[Rapporter un bogue](#)

6.8.5. L'exemple de source de données Oracle



NOTE

Avant la version 10.2 de la source de données Oracle, le paramètre `<no-tx-separate-pools/>` était requis, car le mélange de connexions transactionnelles et non-transactionnelles aurait créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

L'exemple ci-dessous est une configuration de source de données Oracle. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.22. Configuration de source de données Oracle

```

<datasources>
  <datasource jndi-name="java:/OracleDS" pool-name="OracleDS">
    <connection-url>jdbc:oracle:thin:@localhost:1521:XE</connection-url>
    <driver>oracle</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <background-validation-millis>60000</background-validation-
millis>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectio

```



```

nChecker"></valid-connection-checker>
  <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectio
nChecker"></stale-connection-checker>
  <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorte
r"></exception-sorter>
  </validation>
</datasource>
<drivers>
  <driver name="oracle" module="com.oracle">
    <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-
datasource-class>
  </driver>
</drivers>
</datasources>

```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données Oracle ci-dessus.

Exemple 6.23. `module.xml`

```

<module xmlns="urn:jboss:module:1.1" name="com.oracle">
  <resources>
    <resource-root path="ojdbc6.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>

```

[Rapporter un bogue](#)

6.8.6. Exemple de source de données d'Oracle XA



NOTE

Avant la version 10.2 de la source de données Oracle, le paramètre `<no-tx-separate-pools/>` était requis, car le mélange de connexions transactionnelles et non-transactionnelles aurait créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

IMPORTANT

Les paramètres de configuration doivent être appliqués pour un utilisateur qui accède à une source de données Oracle XA pour que le recouvrement XA fonctionne correctement. La valeur user est une valeur définie par l'utilisateur pour pouvoir se connecter à partir de JBoss à Oracle :

- GRANT SELECT ON sys.dba_pending_transactions TO user;
- GRANT SELECT ON sys.pending_trans\$ TO user;
- GRANT SELECT ON sys.dba_2pc_pending TO user;
- GRANT EXECUTE ON sys.dbms_xa TO user; (If using Oracle 10g R2 (patched) or Oracle 11g)

OU

GRANT EXECUTE ON sys.dbms_system TO user; (If using an unpatched Oracle version prior to 11g)

L'exemple ci-dessous est une configuration de source de données Oracle XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.24. Source de données Oracle XA

```
<datasources>
  <xa-datasource jndi-name="java:/XAOracleDS" pool-name="XAOracleDS">
    <driver>oracle</driver>
    <xa-datasource-property name="URL">jdbc:oracle:oci8:@tc</xa-
datasource-property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <xa-pool>
      <is-same-rm-override>>false</is-same-rm-override>
      <no-tx-separate-pools />
    </xa-pool>
    <validation>
      <background-validation>>true</background-validation>
      <background-validation-millis>60000</background-validation-
millis>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectio
nChecker"></valid-connection-checker>
      <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectio
nChecker"></stale-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorte
r"></exception-sorter>
    </validation>
  </xa-datasource>
</drivers>
<driver name="oracle" module="com.oracle">
```

```

        <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-
datasource-class>
    </driver>
</drivers>
</datasources>

```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données Oracle XA ci-dessus.

Exemple 6.25. `module.xml`

```

<module xmlns="urn:jboss:module:1.1" name="com.oracle">
  <resources>
    <resource-root path="ojdbc6.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>

```

[Rapporter un bogue](#)

6.8.7. Exemple de source de données Microsoft SQLServer

L'exemple ci-dessous est une configuration de source de données Microsoft SQLServer. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.26. Configuration de source de données de SQLserver

```

<datasources>
  <datasource jndi-name="java:/MSSQLDS" pool-name="MSSQLDS">
    <connection-
url>jdbc:sqlserver://localhost:1433;DatabaseName=MyDatabase</connection-
url>
    <driver>sqlserver</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <background-validation-millis>60000</background-validation-
millis>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mssql.MSQLValidConnectionC
hecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.mssql.MSQLExceptionSorter"
></exception-sorter>
    </validation>
  </datasource>
</drivers>

```

```

    <driver name="sqlserver" module="com.microsoft">
      <xa-datasource-
class>com.microsoft.sqlserver.jdbc.SQLServerXADataSource</xa-datasource-
class>
    </driver>
  </datasources>

```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données Microsoft SQLServer ci-dessus.

Exemple 6.27. module.xml

```

<module xmlns="urn:jboss:module:1.1" name="com.microsoft">
  <resources>
    <resource-root path="sqljdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>

```

[Rapporter un bogue](#)

6.8.8. Exemple de source de données Microsoft SQLServer XA

L'exemple ci-dessous est une configuration de source de données Microsoft SQLServer XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.28. Source de données SQLserver XA

```

<datasources>
  <xa-datasource jndi-name="java:/MSSQLXADS" pool-name="MSSQLXADS">
    <driver>sqlserver</driver>
    <xa-datasource-property name="ServerName">localhost</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">mssqldb</xa-datasource-
property>
    <xa-datasource-property name="SelectMethod">cursor</xa-datasource-
property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
    </xa-pool>
    <validation>
      <background-validation>true</background-validation>
      <background-validation-millis>60000</background-validation-
millis>
      <valid-connection-checker class-

```

```

name="org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionC
hecker"></valid-connection-checker>
  </validation>
</xa-datasource>
<drivers>
  <driver name="sqlserver" module="com.microsoft">
    <xa-datasource-
class>com.microsoft.sqlserver.jdbc.SQLServerXADataSource</xa-datasource-
class>
    </driver>
  </drivers>
</datasources>

```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données Microsoft SQLServer XA ci-dessus.

Exemple 6.29. module.xml

```

<module xmlns="urn:jboss:module:1.1" name="com.microsoft">
  <resources>
    <resource-root path="sqljdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>

```

[Rapporter un bogue](#)

6.8.9. Exemple de source de données IBM DB2

L'exemple ci-dessous est une configuration de source de données IBM DB2. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.30. Configuration de source de données IBM DB2

```

<datasources>
  <datasource jndi-name="java:/DB2DS" pool-name="DB2DS">
    <connection-url>jdbc:db2:ibmdb2db</connection-url>
    <driver>ibmdb2</driver>
    <pool>
      <min-pool-size>0</min-pool-size>
      <max-pool-size>50</max-pool-size>
    </pool>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <background-validation-millis>60000</background-validation-

```

```

millis>
    <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionCheck
er"></valid-connection-checker>
    <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionCheck
er"></stale-connection-checker>
    <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter">
</exception-sorter>
    </validation>
</datasource>
<drivers>
    <driver name="ibmdb2" module="com.ibm">
    <xa-datasource-class>com.ibm.db2.jdbc.DB2XADataSource</xa-
datasource-class>
    </driver>
</drivers>
</datasources>

```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données IBM DB2 ci-dessus.

Exemple 6.31. `module.xml`

```

<module xmlns="urn:jboss:module:1.1" name="com.ibm">
  <resources>
    <resource-root path="db2jcc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>

```

[Rapporter un bogue](#)

6.8.10. Exemple de source de données IBM DB2 XA

L'exemple ci-dessous est une configuration de source de données IBM DB2 XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.32. Configuration de source de données IBM DB2 XA

```

<datasources>
  <xa-datasource jndi-name="java:/DB2XADS" pool-name="DB2XADS">
    <driver>ibmdb2</driver>
    <xa-datasource-property name="DatabaseName">ibmdb2db</xa-datasource-
property>
    <xa-datasource-property name="ServerName">hostname</xa-datasource-
property>
    <xa-datasource-property name="PortNumber">port</xa-datasource-
property>

```

```

<security>
  <user-name>admin</user-name>
  <password>admin</password>
</security>
<xa-pool>
  <is-same-rm-override>>false</is-same-rm-override>
</xa-pool>
<validation>
  <background-validation>>true</background-validation>
  <background-validation-millis>60000</background-validation-
millis>
  <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionCheck
er"></valid-connection-checker>
  <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionCheck
er"></stale-connection-checker>
  <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter">
</exception-sorter>
</validation>
<recovery>
  <recover-plugin class-
name="org.jboss.jca.core.recovery.ConfigurableRecoveryPlugin">
  <config-property name="EnableIsValid">>false</config-property>
  <config-property name="IsValidOverride">>false</config-property>
  <config-property name="EnableClose">>false</config-property>
  </recover-plugin>
</recovery>
</xa-datasource>
<drivers>
  <driver name="ibmdb2" module="com.ibm">
  <xa-datasource-class>com.ibm.db2.jcc.DB2XADataSource</xa-
datasource-class>
  </driver>
</drivers>
</datasources>

```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données IBM DB2 XA ci-dessus.

Exemple 6.33. `module.xml`

```

<module xmlns="urn:jboss:module:1.1" name="com.ibm">
  <resources>
    <resource-root path="db2jcc4.jar"/>
    <resource-root path="db2jcc_license_cisuz.jar"/>
    <resource-root path="db2jcc_license_cu.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>

```

[Rapporter un bogue](#)

6.8.11. L'exemple de source de données Sybase

L'exemple ci-dessous est une configuration de source de données Sybase. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.34. Configuration de la base de données Sybase

```
<datasources>
  <datasource jndi-name="java:jboss/SybaseDB" pool-name="SybaseDB"
  enabled="true">
    <connection-url>jdbc:sybase:Tds:localhost:5000/DATABASE?
  JCONNECT_VERSION=6</connection-url>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <background-validation-millis>60000</background-validation-
  millis>
      <valid-connection-checker class-
  name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectio
  nChecker"></valid-connection-checker>
      <exception-sorter class-
  name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorte
  r"></exception-sorter>
    </validation>
  </datasource>
  <drivers>
    <driver name="sybase" module="com.sybase">
      <datasource-class>com.sybase.jdbc4.jdbc.SybDataSource</datasource-
  class>
      <xa-datasource-class>com.sybase.jdbc4.jdbc.SybXADataSource</xa-
  datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données Sybase ci-dessus.

Exemple 6.35. module.xml

```
<module xmlns="urn:jboss:module:1.1" name="com.sybase">
  <resources>
    <resource-root path="jconn2.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```


[Rapporter un bogue](#)

6.8.12. L'exemple de source de données Sybase XA

L'exemple ci-dessous est une configuration de source de données Sybase XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

Exemple 6.36. Configuration de la base de données Sybase XA

```
<datasources>
  <xa-datasource jndi-name="java:jboss/SybaseXADS" pool-
name="SybaseXADS" enabled="true">
    <xa-datasource-property name="NetworkProtocol">Tds</xa-datasource-
property>
    <xa-datasource-property name="ServerName">myserver</xa-datasource-
property>
    <xa-datasource-property name="PortNumber">4100</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">mydatabase</xa-
datasource-property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <xa-pool>
      <is-same-rm-override>>false</is-same-rm-override>
    </xa-pool>
    <validation>
      <background-validation>>true</background-validation>
      <background-validation-millis>60000</background-validation-
millis>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectio
nChecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorte
r"></exception-sorter>
    </validation>
  </xa-datasource>
</drivers>
  <driver name="sybase" module="com.sybase">
    <datasource-class>com.sybase.jdbc4.jdbc.SybDataSource</datasource-
class>
    <xa-datasource-class>com.sybase.jdbc4.jdbc.SybXADataSource</xa-
datasource-class>
  </driver>
</drivers>
</datasources>
```

L'exemple ci-dessous est un fichier `module.xml` pour la source de données Sybase XA ci-dessus.

Exemple 6.37. `module.xml`

```
<module xmlns="urn:jboss:module:1.1" name="com.sybase">
  <resources>
    <resource-root path="jconn2.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

[Rapporter un bogue](#)

CHAPITRE 7. CONFIGURATION DES MODULES

7.1. INTRODUCTION

7.1.1. Modules

Un module est un regroupement logique des classes utilisées pour le chargement de classes et pour la gestion de dépendances. JBoss EAP 6 identifie deux types de modules, parfois appelés modules statiques et dynamiques. Cependant, la seule différence entre les deux est la façon dont ils sont emballés.

Modules statiques

Les modules statiques sont prédéfinis dans le répertoire *EAP_HOME/modules/* du serveur d'applications. Chaque sous-répertoire représente un module et contient un sous-fichier de configuration *main/* qui contient un fichier de configuration (`module.xml`) et tous les fichiers JAR requis. Le nom du module est défini dans le fichier `module.xml`. Toutes les API fournies par le serveur de l'application sont des modules statiques, y compris les API Java EE, et les autres API comme JBoss Logging.

Exemple 7.1. Exemple de fichier `module.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.1.15.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Le nom du module, `com.mysql`, doit correspondre à la structure du répertoire du module, à l'exception du nom de sous-répertoire `main/`.

Les modules fournis dans les distributions JBoss EAP se trouvent dans un répertoire `system` se trouvant lui-même dans le répertoire *JBOSS_HOME/modules*. Cela les rend séparés de tout module fourni par une tierce partie.

Tout produit mis en couche de Red Hat, se superposant sur JBoss EAP 6.1 ou version supérieure installera également leurs modules dans le répertoire `system`.

La création de modules statiques personnalisés peut être utile si plusieurs applications sont déployées sur un même serveur utilisant les mêmes bibliothèques de tierce partie. Au lieu d'un regroupement de ces bibliothèques pour chaque application, un module contenant ces bibliothèques peut être créé et installé par l'administrateur JBoss. Les applications peuvent ensuite déclarer une dépendance explicite sur les modules statiques personnalisés.

Les utilisateurs doivent s'assurer que les modules personnalisés soient installés dans le répertoire *JBOSS_HOME/modules*, en utilisant un répertoire par couche de modules. Cela garantit que les versions personnalisées de modules qui existent déjà dans le répertoire `system` soient bien

chargées à la place des versions fournies. Ainsi, les modules utilisateur auront la priorité sur les modules fournis par le système.

Si vous utilisez la variable d'environnement `JBOSS_MODULE_PATH` pour changer les emplacements où JBoss EAP cherche les modules, le produit ira chercher dans une structure de sous-répertoire `system` dans un des emplacements spécifiés. Une structure de sous-répertoire `system` doit exister quelquepart dans les emplacements spécifiés dans `JBOSS_MODULEPATH`.

Modules dynamiques

Les modules dynamiques sont créés et chargés par le serveur d'applications pour chaque déploiement JAR ou WAR (ou sous-déploiement d'un EAR). Le nom d'un module dynamique est dérivé du nom de l'archive déployée. Comme les déploiements sont chargés sous forme de modules, ils peuvent configurer des dépendances et peuvent être utilisés comme dépendances par d'autres déploiements.

Les modules ne sont chargés qu'en fonction des besoins. Cela a généralement lieu quand une application est déployée avec des dépendances implicites ou explicites.

[Rapporter un bogue](#)

7.1.2. Modules globaux

Un module global est un module que JBoss EAP 6 fournit comme dépendance pour chaque application. Chaque module peut être composé en l'ajoutant à la liste du serveur d'applications des modules globaux. Il n'est nul besoin de faire des changements au module.

[Rapporter un bogue](#)

7.1.3. Les dépendances de modules

Une dépendance de module est une déclaration qui indique qu'un module a besoin des classes d'un autre module pour pouvoir fonctionner. Les modules peuvent déclarer leurs dépendances sur un certain nombre d'autres modules. Quand le serveur d'applications charge un module, le chargeur de classes de module traite les dépendances de ce module et ajoute les classes de chaque dépendance à son chemin de classe. Si une dépendance particulière est introuvable, le module ne pourra pas charger.

Les applications déployées (JAR et WAR) sont chargées sous forme de modules dynamiques et utilisent des dépendances pour accéder aux API fournies par JBoss EAP 6.

Il y a deux types de dépendances : explicite et implicite.

Les dépendances explicites sont déclarées dans la configuration par le développeur. Les modules statiques peuvent déclarer des dépendances dans le fichier `modules.xml`. Les modules dynamiques peuvent avoir des dépendances déclarées dans les descripteurs de déploiement `MANIFEST.MF` ou `jboss-deployment-structure.xml`.

Les dépendances explicites peuvent être spécifiées comme étant optionnelles. Une erreur de chargement de dépendance optionnelle n'entraînera pas l'annulation d'un chargement de module. Cependant, si la dépendance est rendue disponible par la suite, elle ne sera PAS ajoutée au chemin de classe du module. Les dépendances doivent être rendues disponibles quand le module est chargé.

Des dépendances implicites sont ajoutées automatiquement par le serveur d'applications quand certaines conditions ou meta-données se trouvent dans un déploiement. Les API Java EE 6 fournies avec JBoss EAP 6 sont des exemples de modules ajoutés par détection de dépendances implicites

dans les déploiements.

Les déploiements peuvent également être configurés de façon à exclure des dépendances implicites particulières. Il vous faut pour cela le fichier de déploiement `jboss-deployment-structure.xml`. C'est normalement le cas quand une application empaquète une version spécifique de bibliothèque que le serveur d'applications tentera d'ajouter comme dépendance implicite.

Un chemin de classe de module ne contient que ses propres classes et celles de ses dépendances immédiates. Un module n'est pas en mesure d'accéder aux classes des dépendances. Cependant, un module peut indiquer quand une dépendance explicite est exportée. Une dépendance exportée est fournie à tout module qui dépend du module qui l'exporte.

Exemple 7.2. Les dépendances de module

Le Module A dépend du Module B et le Module B dépend du Module C. Le Module A peut accéder aux classes du Module B, et le Module B peut accéder aux classes du Module C. Le Module C ne peut pas accéder aux classes du Module C à moins que :

- Le Module A déclare une dépendance explicite sur le Module C, ou bien
- Le Module B exporte ses dépendances sur le Module C.

[Rapporter un bogue](#)

7.1.4. Isolement du chargeur de classes d'un sous-déploiement

Chaque sous-déploiement d'EAR (Archive Enterprise) est un module dynamique possédant son propre chargeur de classe. Par défaut, un sous-déploiement peut accéder aux ressources d'autres sous-déploiements.

Si un sous-déploiement ne doit pas accéder aux ressources d'autres sous-déploiements (une isolation de sous-déploiement est alors requise), alors cela pourra être activé.

[Rapporter un bogue](#)

7.2. DÉSACTIVER L'ISOLEMENT DE MODULE DE SOUS-DÉPLOIEMENT POUR TOUS LES DÉPLOIEMENTS

Cette tâche montre aux administrateurs du serveur comment désactiver l'isolement du module de sous-déploiement dans le serveur d'applications. Cela affecte tous les déploiements.



AVERTISSEMENT

Cette tâche requiert que vous éditiez les fichiers de configuration XML du serveur. Le serveur doit être arrêté avant cela. Ce n'est que temporaire car les outils administratifs de version finale supporteront ce type de configuration.

1. Arrêter le serveur

Mettre en halte le serveur JBoss EAP 6.

2. Ouvrir le fichier de configuration du serveur

Ouvrir le fichier de configuration du serveur dans un éditeur de texte

Ce fichier sera différent pour un domaine géré ou un serveur autonome. De plus, des emplacements et des noms de fichiers non-défauts peuvent être utilisés. Les fichiers de configuration par défaut sont `domain/configuration/domain.xml` et `standalone/configuration/standalone.xml` pour les domaines gérés et les serveurs autonomes respectivement.

3. Chercher la configuration de sous-système EE

Chercher la configuration de sous-système EE dans le fichier de configuration. L'élément `<profile>` du fichier de configuration contient plusieurs éléments du sous-système. L'élément du sous-système EE a comme espace-nom `urn:jboss:domain:ee:1.2`.

```
<profile>
  ...
  <subsystem xmlns="urn:jboss:domain:ee:1.2" />
  ...
```

La configuration par défaut a une balise en fermeture automatique unique mais une configuration personnalisée peut avoir des balises d'ouverture ou de fermeture distinctes (éventuellement avec d'autres éléments à l'intérieur) comme ceci :

```
<subsystem xmlns="urn:jboss:domain:ee:1.2" ></subsystem>
```

4. Remplacer les balises en fermeture automatique si nécessaire

Si l'élément de sous-système EE est une balise en fermeture automatique unique, remplacez-la par les balises d'ouverture ou de fermeture qui conviennent ainsi :

```
<subsystem xmlns="urn:jboss:domain:ee:1.2" ></subsystem>
```

5. Ajouter l'élément `ear-deployments-isolated`

Ajouter l'élément `ear-subdeployments-isolated` comme dépendant de l'élément du sous-système EE et ajouter le contenu de `false` comme suit :

```
<subsystem xmlns="urn:jboss:domain:ee:1.2" ><ear-subdeployments-
isolated>false</ear-subdeployments-isolated></subsystem>
```

6. Démarrer le serveur

Lancer à nouveau le serveur JBoss EAP 6 pour qu'il commence à exécuter avec la nouvelle configuration.

Résultat :

Le serveur va maintenant exécuter avec l'isolement de module de sous-déploiement désactivé pour tous les déploiements.

[Rapporter un bogue](#)

7.3. AJOUTER UN MODULE À TOUS LES DÉPLOIEMENTS

Cette tâche montre comment les administrateurs JBoss peuvent définir une liste de modules globaux.

Conditions préalables

1. Vous devez connaître le nom des modules qui ont été ajoutés comme modules globaux. Voir [Section 7.6.1, « Les modules inclus »](#) pour obtenir la liste des modules statiques inclus dans JBoss EAP 6. Si le module est dans un autre déploiement, voir [Section 7.6.2, « Nommage de modules dynamiques »](#) pour déterminer le nom du module.

Procédure 7.1. Ajouter un module à la liste des modules globaux

1. Connectez-vous à la console de gestion. [Section 3.4.2, « Se connecter à la console de gestion »](#)
2. Naviguer dans le panneau **EE Subsystem**.
 - a. Sélectionner **Configuration** qui se trouve en haut de la console.
 - b. **Mode Domaine uniquement**
 - i. Sélectionner le profil qui convient à partir du menu déroulant en haut à gauche.
 - c. Étendre le menu **Subsystems** qui se trouve à gauche de la console.
 - d. Sélectionner **Container** → **EE** à partir du menu à gauche de la console.
3. Cliquer sur **Add** dans la section **Subsystem Defaults**. La boîte de dialogue **Create Module** apparaîtra.
4. Saisir alors le nom du module et le slot de module, en option.
5. Cliquer sur **Enregistrer** pour ajouter le nouveau module global, ou bien cliquer sur **Annuler** pour annuler.
 - o Si vous cliquer sur **Enregistrer**, la boîte de dialogue va se fermer et le module spécifié sera ajouté à la liste des modules globaux.
 - o Si vous cliquer sur le bouton **Cancel**, la boîte de dialogue se fermera et il n'y aura aucun changement.

Résultat

Les modules ajoutés à la liste des modules globaux seront ajoutés en tant que dépendances à chaque déploiement.

[Rapporter un bogue](#)

7.4. CRÉER UN MODULE PERSONNALISÉ

La procédure suivante décrit comment créer un module personnalisé pour rendre les fichiers de propriétés et les autres ressources disponibles à toutes les applications qui exécutent sur le serveur JBoss EAP.

Procédure 7.2. Créer un module personnalisé

1. Créer et compléter la structure de répertoire `module/`.

- a. Créer une structure de répertoire sous le répertoire **EAP_HOME/module** qui contienne les fichiers et les JAR.

```
$ cd EAP_HOME/modules/
$ mkdir -p myorg-conf/main/properties
```

- b. Déplacer les fichiers de propriétés dans le répertoire **EAP_HOME/modules/myorg-conf/main/properties/** que vous avez créé à l'étape précédente.
- c. Créer un fichier **module.xml** dans le répertoire **EAP_HOME/modules/myorg-conf/main/** qui contienne l'XML suivant :

```
<module xmlns="urn:jboss:module:1.1" name="myorg-conf">
  <resources>
    <resource-root path="properties"/>
  </resources>
</module>
```

2. Modifier le sous-système **ee** du fichier de configuration du serveur. Vous pouvez utiliser l'interface en ligne de commande JBoss CLI, ou bien, vous pouvez éditer le fichier manuellement.

- o Suivez ces étapes pour modifier le fichier de configuration du serveur par l'interface en ligne de commande JBoss CLI.

- a. Démarrez le serveur et connectez-vous à l'interface en ligne de commande.

- Dans Linux, saisir ce qui suit au niveau de la ligne de commande :

```
EAP_HOME/bin/jboss-cli.sh --connect
```

- Dans Windows, saisir ce qui suit au niveau de la ligne de commande :

```
C:\>EAP_HOME\bin\jboss-cli.bat --connect
```

Vous devriez voir apparaître le résultat suivant :

```
Connected to standalone controller at localhost:9999
```

- b. Pour créer l'élément **<global-modules> myorg-conf** dans le sous-système **ee**, saisir ce qui suit dans la ligne de commande :

```
/subsystem=ee:write-attribute(name=global-modules, value=
[{"name"=>"myorg-conf", "slot"=>"main"}])
```

Vous devriez voir apparaître le résultat suivant :

```
{"outcome" => "success"}
```

- o Suivre ces étapes si vous préférez éditer manuellement le fichier de configuration du serveur.

- a. Arrêter le serveur et ouvrir le fichier de configuration du serveur dans un éditeur de

texte. Si vous exécutez sur un serveur autonome, il s'agira du fichier `EAP_HOME/standalone/configuration/standalone.xml` ou du fichier `EAP_HOME/domain/configuration/domain.xml` si vous exécutez sur un domaine géré.

- b. Chercher le sous-système `ee` et ajouter le module global de `myorg-conf`. Ce qui suit est un exemple d'élément du sous-système `ee` modifié pour inclure l'élément `myorg-conf` :

Exemple 7.3. élément `myorg-conf`

```
<subsystem xmlns="urn:jboss:domain:ee:1.0" >
  <global-modules>
    <module name="myorg-conf" slot="main" />
  </global-modules>
</subsystem>
```

3. Si vous copiez un fichier nommé `my.properties` dans l'emplacement de module qui convient, vous pourrez alors charger les fichiers de propriétés en utilisant un code qui ressemble à ceci :

Exemple 7.4. Télécharger le fichier de propriétés

```
Thread.currentThread().getContextClassLoader().getResource("my.properties");
```

[Rapporter un bogue](#)

7.5. DÉFINIR UN RÉPERTOIRE DE MODULES JBOSS EXTERNE

Résumé

Par défaut, JBoss EAP recherche les modules dans le répertoire `EAP_HOME/modules/`. Vous pouvez demander à JBoss EAP de regarder dans un ou plusieurs modules répertoires externes en définissant une variable d'environnement `JBOSS_MODULEPATH` ou en définissant la variable dans le fichier de configuration de démarrage. Cette section décrit les deux méthodes.

Procédure 7.3. Définissez la variable d'environnement `JBOSS_MODULEPATH`

- Pour spécifier un ou plusieurs répertoires de module externes, définir la variable d'environnement `JBOSS_MODULEPATH`.

Dans Linux, utiliser les deux-points pour délimiter une liste de répertoires. Par exemple :

Exemple 7.5. variable d'environnement `JBOSS_MODULEPATH`

```
export
JBOSS_MODULEPATH=EAP_HOME/modules:/home/username/external/modules
/directory/
```

Dans Linux, utiliser un point-virgule pour délimiter une liste de répertoires. Par exemple :

Exemple 7.6. variable d'environnement `JBOSS_MODULEPATH`

```
SET JBOSS_MODULEPATH=EAP_HOME\modules\;D:\JBoss-Modules\
```

Procédure 7.4. Définissez la variable `JBOSS_MODULEPATH` dans le fichier de configuration de démarrage.

- Si vous choisissez de ne pas définir une variable d'environnement globale, vous pouvez définir la variable `JBOSS_MODULEPATH` dans le fichier de configuration de démarrage de JBoss EAP. Si vous exécutez dans un serveur autonome, il s'agira du fichier `EAP_HOME/bin/standalone.conf`. Si le serveur exécute dans un domaine géré, il s'agira du fichier `EAP_HOME/bin/domain.conf`.

Vous trouverez ci-dessous un exemple de la commande qui définit la variable `JBOSS_MODULEPATH` dans le fichier `standalone.conf` :

Exemple 7.7. entrée `standalone.conf`

```
JBOSS_MODULEPATH="EAP_HOME/modules/:/home/username/external/modules/directory/"
```

[Rapporter un bogue](#)

7.6. RÉFÉRENCE

7.6.1. Les modules inclus

Un tableau énumérant les modules JBoss EAP 6 inclus et indiquant s'ils sont pris en charge se trouve dans le portail clients à <https://access.redhat.com/articles/1122333>.

[Rapporter un bogue](#)

7.6.2. Nommage de modules dynamiques

Tous les déploiements sont chargés en tant que modules par JBoss EAP 6 et sont nommés en fonction des conventions suivantes :

1. Les déploiements des fichiers WAR et JAR sont nommés selon le format suivant :

```
deployment.DEPLOYMENT_NAME
```

Par exemple, `inventory.war` et `store.jar` auront les mêmes noms de module que `deployment.inventory.war` et `deployment.store.jar` respectivement.

2. Les sous-déploiements des archives Enterprise sont nommés selon le format suivant :

```
deployment.EAR_NAME.SUBDEPLOYMENT_NAME
```

Ainsi, le sous-déploiement `reports.war`, qui se trouve dans l'archive Enterprise `accounts.ear`, aura le nom de module du `deployment.accounts.ear.reports.war`.

[Rapporter un bogue](#)

CHAPITRE 8. JSVC

8.1. INTRODUCTION

8.1.1. Jsvc

Jsvc est un ensemble de bibliothèques et d'applications qui permettent aux applications Java exécutées sur UNIX ou sur des plateformes style UNIX en arrière-plan. Jsvc permet à une application d'effectuer des opérations en tant qu'utilisateur privilégié, puis de changer d'identité en tant qu'utilisateur non privilégié.

Jsvc utilise trois processus : un processus de lanceur, un processus de contrôleur et un processus contrôlé. Le processus contrôlé est également le principal thread Java. Si la JVM plante le processus du contrôleur, il redémarrera dans les 60 secondes. Jsvc est un processus de démon et pour JBoss EAP 6, il doit être lancé par un utilisateur privilégié.



NOTE

Jsvc est peut être utilisé sur Red Hat Enterprise Linux, Solaris et HP-UX. Pour une fonctionnalité similaire dans Microsoft Windows, voir `prunsrv.exe` dans **Native Utilities for Windows Server** disponibles depuis le portail clients de Red Hat.

[Rapporter un bogue](#)

8.1.2. Démarrer et arrêter JBoss EAP par Jsvc

Les instructions de démarrage et d'arrêt de JBoss EAP par Jsvc varient, selon le mode opérationnel : autonome ou domaine. Sachez que si JBoss EAP est exécuté en mode de domaine, Jsvc ne gèrera que le processus de contrôleur de domaine. Quelle que soit la commande que vous utilisez pour démarrer JBoss EAP par Jsvc, elle doit être gérée par un utilisateur privilégié.

Conditions préalables

- Si JBoss EAP était installé par la méthode Zip :
 - Installez le package *Native Utilities* de votre système d'exploitation, disponible à partir du portail clients de Red Hat *Install Native Components and Native Utilities (Zip, Installer)* dans *Installation Guide*.
 - Créez le compte d'utilisateur sous lequel s'exécute l'instance de JBoss EAP 6. Le compte utilisé pour démarrer et arrêter le serveur doit posséder un accès lecture et écriture dans le répertoire où JBoss EAP a été installé.
- Si JBoss EAP était installé par la méthode RPM, installez le package *apache-commons-daemon-jsvc-eap6*. Voir *Install Native Components and Native Utilities (RPM Installation)* dans le guide *Installation Guide*.

Les commandes suivantes sont utilisées pour démarrer et arrêter JBoss EAP en mode autonome ou domaine. Notez que les emplacements de fichiers sont différents selon la méthode utilisée pour installer Jsvc dans JBoss EAP 6. Utilisez les tableaux ci-dessous pour déterminer les fichiers à utiliser pour résoudre les variables des commandes.

Mode autonome

Les instructions suivantes sont utilisées pour démarrer ou pour stopper JBoss EAP en mode autonome.

Tableau 8.1. Emplacements des fichiers Jsvc pour les installations zip - Mode autonome

Référence Fichier en Instructions	Emplacement fichier
EAP-HOME	<code>\${eap-installation-location}/jboss-eap-\${version}</code>
JSVC-BIN	<code>EAP_HOME/modules/system/layers/base/native/sbin/jsvc</code>
JSVC-JAR	<code>EAP_HOME/modules/system/layers/base/native/sbin/commons-daemon.jar</code>
CONF-DIR	<code>EAP_HOME/standalone/configuration</code>
LOG-DIR	<code>EAP_HOME/standalone/log</code>

Tableau 8.2. Emplacements des fichiers Jsvc pour les installations RPM - Mode autonome

Référence Fichier en Instructions	Emplacement fichier
EAP-HOME	<code>/usr/share/jbossas</code>
JSVC-BIN	<code>/usr/bin/jsvc-eap6/jsvc</code>
JSVC-JAR	<code>EAP_HOME/modules/system/layers/base/native/sbin/commons-daemon.jar</code>
CONF-DIR	<code>/etc/jbossas/standalone</code>
LOG-DIR	<code>/var/log/jbossas/standalone</code>

Démarrez JBoss en serveur autonome

- ```

JSVC_BIN \
-outfile LOG_DIR/jsvc.out.log \
-errfile LOG_DIR/jsvc.err.log \
-pidfile LOG_DIR/jsvc.pid \
-user jboss \
-D[Standalone] -XX:+UseCompressedOops -Xms1303m \
-Xmx1303m -XX:MaxPermSize=256m \
-Djava.net.preferIPv4Stack=true \
-Djboss.modules.system.pkgs=org.jboss.byteman \
-Djava.awt.headless=true \
-Dorg.jboss.boot.log.file=LOG_DIR/server.log \
-Dlogging.configuration=file:CONF_DIR/logging.properties \
-Djboss.modules.policy-permissions \
-cp EAP_HOME/jboss-modules.jar:JSVC_JAR \

```

```
-Djboss.home.dir=EAP_HOME \
-Djboss.server.base.dir=EAP_HOME/standalone \
@org.jboss.modules.Main -start-method main \
-mp EAP_HOME/modules \
-jaxpmodule javax.xml.jaxp-provider \
org.jboss.as.standalone
```

### Stopper JBoss en serveur autonome

- ```
JSVC_BIN \
-stop \
-outfile LOG_DIR/jsvc.out.log \
-errfile LOG_DIR/jsvc.err.log \
-pidfile LOG_DIR/jsvc.pid \
-user jboss \
-D[Standalone] -XX:+UseCompressedOops -Xms1303m \
-Xmx1303m -XX:MaxPermSize=256m \
-Djava.net.preferIPv4Stack=true \
-Djboss.modules.system.pkgs=org.jboss.byteman \
-Djava.awt.headless=true \
-Dorg.jboss.boot.log.file=LOG_DIR/server.log \
-Dlogging.configuration=file:CONF_DIR/logging.properties \
-Djboss.modules.policy-permissions \
-cp EAP_HOME/jboss-modules.jar:JSVC_JAR \
-Djboss.home.dir=EAP_HOME \
-Djboss.server.base.dir=EAP_HOME/standalone \
@org.jboss.modules.Main -start-method main \
-mp EAP_HOME/modules \
-jaxpmodule javax.xml.jaxp-provider \
org.jboss.as.standalone
```

Mode Domaine

Les instructions suivantes sont utilisées pour démarrer ou pour stopper JBoss EAP en mode autonome. Notez qu'en mode de domaine, vous devrez remplacer la variable `JAVA_HOME` par le répertoire d'accueil de Java.

Tableau 8.3. Emplacements des fichiers Jsvc pour les installations zip - Mode de domaine

Référence Fichier en Instructions	Emplacement fichier
EAP-HOME	<code>\${eap-installation-location}/jboss-eap-\${version}</code>
JSVC-BIN	<code>EAP_HOME/modules/system/layers/base/native/sbin/jsvc</code>
JSVC-JAR	<code>EAP_HOME/modules/system/layers/base/native/sbin/commons-daemon.jar</code>
CONF-DIR	<code>EAP_HOME/domain/configuration</code>
LOG-DIR	<code>EAP_HOME/domain/log</code>

Tableau 8.4. Emplacements des fichiers Jsvc pour les installations RPM - Mode de domaine

Référence Fichier en Instructions	Emplacement fichier
EAP-HOME	/usr/share/jbossas
JSVC-BIN	/usr/bin/jsvc-eap6/jsvc
JSVC-JAR	EAP_HOME/modules/system/layers/base/native/sbin/commons-daemon.jar
CONF-DIR	/etc/jbossas/domain
LOG-DIR	/var/log/jbossas/domain

Démarrez JBoss EAP en mode de domaine

- ```

JSVC_BIN \
-outfile LOG_DIR/jsvc.out.log \
-errfile LOG_DIR/jsvc.err.log \
-pidfile LOG_DIR/jsvc.pid \
-user jboss \
-nodetach -D"[Process Controller]" -server -Xms64m \
-Xmx512m -XX:MaxPermSize=256m \
-Djava.net.preferIPv4Stack=true \
-Djboss.modules.system.pkgs=org.jboss.byteman \
-Djava.awt.headless=true \
-Dorg.jboss.boot.log.file=LOG_DIR/process-controller.log \
-Dlogging.configuration=file:CONF_DIR/logging.properties \
-Djboss.modules.policy-permissions \
-cp "EAP_HOME/jboss-modules.jar:JSVC_JAR" \
org.apache.commons.daemon.support.DaemonWrapper \
-start org.jboss.modules.Main -start-method main \
-mp EAP_HOME/modules org.jboss.as.process-controller \
-jboss-home EAP_HOME -jvm $JAVA_HOME/bin/java \
-mp EAP_HOME/modules -- \
-Dorg.jboss.boot.log.file=LOG_DIR/host-controller.log \
-Dlogging.configuration=file:CONF_DIR/logging.properties \
-Djboss.modules.policy-permissions \
-server -Xms64m -Xmx512m -XX:MaxPermSize=256m \
-Djava.net.preferIPv4Stack=true \
-Djboss.modules.system.pkgs=org.jboss.byteman \
-Djava.awt.headless=true -- -default-jvm $JAVA_HOME/bin/java

```

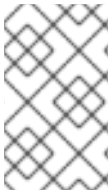
### Stopper JBoss EAP en mode de domaine

- ```

JSVC_BIN \
-stop \
-outfile LOG_DIR/jsvc.out.log \
-errfile LOG_DIR/jsvc.err.log \
-pidfile LOG_DIR/jsvc.pid \

```

```
-user jboss \  
-nodetach -D"[Process Controller]" -server -Xms64m \  
-Xmx512m -XX:MaxPermSize=256m \  
-Djava.net.preferIPv4Stack=true \  
-Djboss.modules.system.pkgs=org.jboss.byteman \  
-Djava.awt.headless=true \  
-Dorg.jboss.boot.log.file=LOG_DIR/process-controller.log \  
-Dlogging.configuration=file:CONF_DIR/logging.properties \  
-Djboss.modules.policy-permissions \  
-cp "EAP_HOME/jboss-modules.jar:JSVC_JAR" \  
org.apache.commons.daemon.support.DaemonWrapper \  
-start org.jboss.modules.Main -start-method main \  
-mp EAP_HOME/modules org.jboss.as.process-controller \  
-jboss-home EAP_HOME -jvm $JAVA_HOME/bin/java \  
-mp EAP_HOME/modules -- \  
-Dorg.jboss.boot.log.file=LOG_DIR/host-controller.log \  
-Dlogging.configuration=file:CONF_DIR/logging.properties \  
-Djboss.modules.policy-permissions \  
-server -Xms64m -Xmx512m -XX:MaxPermSize=256m \  
-Djava.net.preferIPv4Stack=true \  
-Djboss.modules.system.pkgs=org.jboss.byteman \  
-Djava.awt.headless=true -- -default-jvm $JAVA_HOME/bin/java
```

**NOTE**

Si JBoss EAP 6 est arrêté de façon anormale, comme un crash de la JVM, Jsvc le démarrera à nouveau automatiquement. Si JBoss EAP 6 s'arrête normalement, Jsvc s'arrêtera également.

[Rapporter un bogue](#)

CHAPITRE 9. VALVES GLOBALES

9.1. VALVES

Une valve est une classe Java insérée dans le pipeline de traitement de demande d'une application. Elle est insérée dans le pipeline avant les filtres servlet. Les valves peuvent apporter des modifications à la demande avant de les passer ou d'effectuer tout autre traitement comme l'authentification ou annuler la demande.

Les valves peuvent être configurées au niveau du serveur ou au niveau de l'application. La seule différence est la façon dont elles sont configurées ou empaquetées.

- Les valves globales sont configurées au niveau serveur et s'appliquent à toutes les applications déployées dans le serveur. Les instructions sur la façon de configurer les valves globales se trouvent dans le guide *Administration and Configuration Guide* de JBoss EAP.
- Les valves configurées au niveau de l'application sont empaquetées dans le déploiement de l'application et n'affectent que l'application en question. Des instructions sur la façon de configurer des valves au niveau de l'application se trouvent dans le guide *Development Guide* de JBoss EAP.

Les versions 6.1.0 et supérieures prennent en charge les valves globales.

[Rapporter un bogue](#)

9.2. VALVES GLOBALES

Une valve globale est une valve insérée dans le pipeline de traitement de requête de toutes les applications déployées. Une valve est rendue globale lorsqu'elle est mise en paquetage et qu'elle est installée comme module static dans JBoss EAP 6. Les valves globales sont configurées dans le sous-système web.

Seules les versions 6.1.0 et supérieures prennent en charge les valves globales.

Pour obtenir plus d'informations sur la façon de configurer Global Valves, voir [Section 9.5](#), « Configuration d'une valve globale ».

[Rapporter un bogue](#)

9.3. LES VALVES D'AUTHENTIFICATION

Une valve d'authentification est une valve qui authentifie les informations d'identification d'une requête. Cette valve est une sous-classe de `org.apache.catalina.authenticator.AuthenticatorBase` et elle remplace la méthode `authenticate(Request request, Response response, LoginConfig config)`.

Elle peut être utilisée pour implémenter des schémas d'authentification supplémentaires.

[Rapporter un bogue](#)

9.4. INSTALLATION D'UNE VALVE GLOBALE

Les valves globales doivent être empaquetées et installées sous forme de modules statics dans JBoss EAP 6. Cette tâche vous montre comment installer le module.

Conditions préalables :

- La valve doit déjà avoir été créée et empaquetée dans un fichier JAR.
- Un fichier `module.xml` doit déjà avoir été créé pour le module.

Voir [Section 7.1.1, « Modules »](#) pour un exemple de fichier `module.xml`.

Procédure 9.1. Installer un module global**1. Créer un répertoire d'installation de module**

Vous devrez créer un répertoire pour le module à installer dans le répertoire de modules du serveur d'applications.

```
EAP_HOME/modules/system/layers/base/MODULENAME/main
```

```
$ mkdir -P EAP_HOME/modules/system/layers/base/MODULENAME/main
```

2. Copier les fichiers

Copier le JAR et les fichiers `module.xml` dans le répertoire créé dans l'étape 1.

```
$ cp MyValves.jar module.xml
EAP_HOME/modules/system/layers/base/MODULENAME/main
```

Les classes de valve déclarées dans le module sont maintenant disponibles et peuvent être configurées dans le sous-système web.

[Rapporter un bogue](#)

9.5. CONFIGURATION D'UNE VALVE GLOBALE

Les valves globales sont activées et configurées dans le sous-système web par l'intermédiaire de l'interface CLI de JBoss.

Procédure 9.2. Configuration d'une valve globale**1. Activer la valve**

Utiliser l'opération `add` pour ajouter une nouvelle saisie de valve.

```
/subsystem=web/valve=VALVENAME:add(module="MODULENAME",class-
name="CLASSNAME")
```

Vous devrez indiquer les valeurs suivantes :

- **VALVENAME**, le nom utilisé pour cette valve dans la configuration de l'application.
- **MODULENAME**, le module qui contient la valeur en cours de configuration.
- **CLASSNAME**, le nom de classe de la valve spécifique du module.

Par exemple :

```
/subsystem=web/valve=clientlimiter:add(module="clientlimitermodule",  
class-name="org.jboss.samplevalves.RestrictedUserAgentsValve")
```

2. En option : spécifier les paramètres

Si la valve a des paramètres de configuration, spécifier les dans l'opération `add-param`.

```
/subsystem=web/valve=VALVENAME:add-param(param-name="PARAMNAME",  
param-value="PARAMVALUE")
```

Vous devrez indiquer les valeurs suivantes :

- **VALVENAME**, le nom utilisé pour cette valve dans la configuration de l'application.
- **PARAMNAME**, le nom du paramètre configuré pour une valve spécifique.
- **PARAMVALUE**, la valeur du paramètre spécifié.

Par exemple :

Exemple 9.1. Configuration de la valve

```
/subsystem=web/valve=clientlimiter:add-param(  
    param-name="restrictedUserAgents",  
    param-value="^.*MS Web Services Client Protocol.*$" )
```

Cette valve est maintenant activée et configurée pour toutes les applications déployées.

Voir la section *Create a Custom Valve* du *Guide de développement* pour obtenir plus d'informations sur la façon de créer une valve personnalisée.

[Rapporter un bogue](#)

CHAPITRE 10. DÉPLOIEMENT D'APPLICATIONS

10.1. LES DÉPLOIEMENTS D'APPLICATIONS

JBoss EAP 6 dispose d'une gamme d'options de déploiement et de configuration d'application pour répondre à la fois aux environnements administratifs et de développement. Pour les administrateurs, la console de gestion et l'interface CLI offrent un graphisme et des interfaces de ligne de commande idéals pour gérer le déploiement des applications dans un environnement de production. Pour les développeurs, la gamme des options de testing de déploiement d'application incluent un scanner de déploiement hautement configurable de système de fichiers, l'utilisation d'un IDE comme JBoss Developer Studio, ou le déploiement et l'annulation du déploiement via Maven.

Administration

- **Console de management**
 - [Section 10.2.2, « Activer une application déployée à l'aide de la console de gestion »](#)
 - [Section 10.2.3, « Désactiver une application déployée à l'aide de la console de gestion »](#)
- **Interface CLI**
 - [Section 10.3.4, « Déployer une application dans un domaine géré à l'aide de l'interface CLI »](#)
 - [Section 10.3.2, « Déployer une application dans un serveur autonome à l'aide de l'interface CLI »](#)
 - [Section 10.3.5, « Supprimer le déploiement d'une application dans un domaine géré à l'aide de l'interface CLI »](#)
 - [Section 10.3.3, « Supprimer le déploiement d'une application dans un serveur autonome à l'aide de l'interface CLI »](#)
 - [Section 10.3.1, « Gérer le déploiement d'une application à l'aide de l'interface CLI »](#)

Développement

- **Scanner de déploiement**
 - [Section 10.5.7, « Configurer le scanner de déploiement »](#)
 - [Section 10.5.2, « Déployer une application dans une instance de serveur autonome par un scanner de déploiement »](#)
 - [Section 10.5.3, « Supprimer le déploiement d'une application dans une instance de serveur autonome par un scanner de déploiement »](#)
 - [Section 10.5.4, « Redéploiement d'une application dans une instance de serveur autonome par le scanner de déploiement »](#)
 - [Section 10.5.8, « Configurer le scanner de déploiement avec l'interface CLI »](#)
 - [Section 10.5.6, « Référence pour attributs de scanner de déploiement »](#)

- [Section 10.5.5, « Référence pour les fichiers de marquage de scanneur de déploiement »](#)
- **Maven**
 - [Section 10.6.2, « Déployer une application dans Maven »](#)
 - [Section 10.6.3, « Supprimer le déploiement d'une application dans Maven »](#)

[Rapporter un bogue](#)

10.2. DÉPLOYER AVEC LA CONSOLE DE GESTION

10.2.1. Gérer le déploiement d'application à l'aide de la console de gestion

Le déploiement d'applications par l'intermédiaire de la console de gestion vous donne l'avantage d'une interface graphique facile à utiliser. Vous pouvez voir en un coup d'œil quelles applications sont déployées sur votre serveur ou les groupes de serveurs, et vous pouvez désactiver ou supprimer des applications dans le référentiel de contenu selon les besoins.

[Rapporter un bogue](#)

10.2.2. Activer une application déployée à l'aide de la console de gestion

Conditions préalables

- [Section 3.4.2, « Se connecter à la console de gestion »](#)
- [Section 3.4.7, « Ajouter un déploiement dans une console de management »](#)

Procédure 10.1. Activer une application déployée à l'aide de la console de gestion

1. Sélectionner l'onglet **Runtime** en haut de la console.
2.
 - Pour un domaine géré, étendre le menu **Domain**.
 - Pour un domaine autonome, étendre le menu **Server**.
3. Sélectionner **Manage Deployments**.
4. La méthode de déploiement des applications variera suivant que vous déployez dans une instance de serveur autonome ou dans un domaine géré.
 - **Activer une application sur une instance de serveur autonome**
Le tableau **Available Deployments** affiche tous les déploiements d'applications disponibles et leurs statuts.
 - a. Pour activer une application dans une instance de serveur autonome, sélectionner l'application, puis cliquer sur **En/Disable**.
 - b. Cliquer sur **confirm** pour confirmer que l'application puisse être activée dans l'instance du serveur.
 - **Activer une application dans un domaine géré**

L'onglet **Content Repository** contient un tableau **Available Deployment Content** qui montre tous les déploiements d'applications disponibles et leur statut.

- a. Pour activer une application dans un domaine géré, sélectionner l'application à déployer. Cliquer sur **Assign** au dessus du tableau **Available Deployment Content**.
- b. Cocher les cases pour chaque groupe de serveurs dans lesquels vous souhaitez ajouter l'application et cliquer sur le bouton **Save** pour terminer.
- c. Sélectionner l'onglet **Server Groups** pour afficher le tableau **Server Groups**. Votre application sera maintenant déployée dans les groupes de serveurs que vous avez sélectionnés.

Résultat

L'application est déployée sur le serveur qui convient ou dans le groupe de serveurs qui convient.

[Rapporter un bogue](#)

10.2.3. Désactiver une application déployée à l'aide de la console de gestion

Conditions préalables

- [Section 3.4.2, « Se connecter à la console de gestion »](#)
- [Section 3.4.7, « Ajouter un déploiement dans une console de management »](#)
- [Section 10.2.2, « Activer une application déployée à l'aide de la console de gestion »](#)

Procédure 10.2. Désactiver une application déployée à l'aide de la console de gestion

1.
 - a. Sélectionner l'onglet **Runtime** en haut de la console.
 - b.
 - Pour un domaine géré, étendre le menu **Domain**.
 - Pour un domaine autonome, étendre le menu **Server**.
 - c. Sélectionner **Manage Deployments**.
2. La méthode de suppression d'une application variera suivant que vous déployez dans une instance de serveur autonome ou dans un domaine géré.
 - **Désactiver une application déployée dans une instance de serveur autonome**
Le tableau **Available Deployments** affiche tous les déploiements d'applications disponibles et leurs statuts.
 - a. Sélectionner l'application à désactiver. Cliquer sur **En/Disable** pour désactiver l'application sélectionnée.
 - b. Cliquer sur **Confirm** pour confirmer que l'application puisse être désactivée dans l'instance du serveur.
 - **Désactiver l'application déployée dans un domaine géré**

L'affichage écran **Manage Deployment Content** contient un onglet **Content Repository**. Le tableau **Available Deployment Content** affiche tous les déploiements d'applications disponibles et leur statut.

- a. Sélectionner l'onglet **Server Groups** pour afficher les groupes de serveurs et le statut de leurs applications déployées.
- b. Sélectionner le nom du serveur dans le tableau **Server Group** pour supprimer un déploiement. Cliquer sur **View** pour voir les applications.
- c. Sélectionner l'application à désactiver et cliquer sur **En/Disable** pour désactiver l'application sur le serveur sélectionné.
- d. Cliquer sur **Confirm** pour confirmer que l'application puisse être désactivée dans l'instance du serveur.
- e. Répéter au besoin pour d'autres groupes de serveur. Le statut de l'application est confirmé pour chaque groupe de serveurs dans le tableau **Group Deployments** de ce groupe de serveurs.

Résultat

L'application n'est pas déployée à partir d'un serveur ou d'un groupe de serveurs qui convient.

[Rapporter un bogue](#)

10.2.4. Retirer le déploiement d'une application à l'aide de la Console de gestion

Conditions préalables

- [Section 3.4.2, « Se connecter à la console de gestion »](#)
- [Section 3.4.7, « Ajouter un déploiement dans une console de management »](#)
- [Section 10.2.2, « Activer une application déployée à l'aide de la console de gestion »](#)
- [Section 10.2.3, « Désactiver une application déployée à l'aide de la console de gestion »](#)

Procédure 10.3. Retirer le déploiement d'une application à l'aide de la Console de gestion

1.
 - a. Sélectionner l'onglet **Runtime** en haut de la console.
 - b.
 - Pour une domaine géré, étendre le menu **Domain**.
 - Pour une domaine autonome, étendre le menu **Server**.
 - c. Sélectionner **Manage Deployments**.
2. La méthode de suppression d'une application utilisée variera suivant que vous supprimez un déploiement dans une instance de serveur autonome ou dans un domaine géré.
 - **Supprimer une application déployée d'une instance de serveur autonome**
Le tableau **Available Deployments** affiche tous les déploiements d'applications disponibles et leurs statuts.

- a. Sélectionner l'application dont on doit supprimer le déploiement. Cliquer sur le bouton **Remove** pour supprimer le déploiement d'une application sélectionnée.
 - b. Cliquer sur **Confirm** pour confirmer que le déploiement de l'application puisse être supprimé dans l'instance du serveur.
- o **Supprimer le déploiement d'une application déployée dans un domaine géré**
L'affichage écran **Manage Deployment Content** contient un onglet **Content Repository**. Le tableau **Available Deployment Content** affiche tous les déploiements d'applications disponibles et leur statut.
- a. Sélectionner l'onglet **Server Groups** pour afficher les groupes de serveurs et le statut de leurs applications déployées.
 - b. Sélectionner le nom du serveur dans le tableau **Server Group** pour supprimer un déploiement. Cliquer sur **View** pour voir les applications.
 - c. Sélectionner l'application et cliquer sur **Remove** pour supprimer le déploiement de l'application d'un serveur sélectionné.
 - d. Cliquer sur **Confirm** pour confirmer que le déploiement de l'application puisse être supprimé dans l'instance du serveur.
 - e. Répéter au besoin pour d'autres groupes de serveur. Le statut de l'application est confirmé pour chaque groupe de serveurs dans le tableau **Group Deployments** de ce groupe de serveurs.

Résultat

Le déploiement de l'application est supprimé d'un serveur ou d'un groupe de serveurs particulier. Dans une instance autonome, le contenu du déploiement sera supprimé également. Dans un domaine géré, le contenu du déploiement restera dans le référentiel de contenu et le déploiement ne sera supprimé qu'à partir d'un groupe de serveurs.

[Rapporter un bogue](#)

10.3. DÉPLOYER PART L'INTERFACE DE COMMANDES CLI

10.3.1. Gérer le déploiement d'une application à l'aide de l'interface CLI

Le déploiement d'applications par l'intermédiaire de l'interface CLI vous donne l'avantage d'une interface à ligne de commande facile à utiliser. Vous pouvez utiliser les capacités de scripting pour configurer le déploiement d'applications spécifiques et des scénarios de gestion. Vous pouvez gérer le statut de déploiement d'un serveur dans le cas d'une instance autonome, ou d'un réseau entier de serveurs dans le cas d'un domaine géré.

[Rapporter un bogue](#)

10.3.2. Déployer une application dans un serveur autonome à l'aide de l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

- [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#)

Procédure 10.4. Déployer une application dans un serveur autonome

- **Exécuter la commande `deploy`**
À l'aide de l'interface CLI, saisir la commande `deploy` avec le chemin d'accès vers le déploiement de l'application.

Exemple 10.1. La commande « `deploy` »

```
[standalone@localhost:9999 /] deploy /path/to/test-application.war
```

Veillez noter qu'un déploiement réussi ne crée pas de sortie vers le CLI.

Résultat

L'application indiquée est maintenant déployée dans un serveur autonome.

[Rapporter un bogue](#)

10.3.3. Supprimer le déploiement d'une application dans un serveur autonome à l'aide de l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)
- [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#)
- [Section 10.3.2, « Déployer une application dans un serveur autonome à l'aide de l'interface CLI »](#)

Procédure 10.5. Supprimer le déploiement d'une application dans un serveur autonome

Par défaut, la commande `undeploy` supprimera le déploiement et supprimera le contenu de déploiement de l'instance autonome de JBoss EAP. Pour retenir le contenu du déploiement, ajouter le paramètre `--keep-content`.

- **Exécuter la commande `undeploy`**
Pour supprimer le déploiement de l'application et supprimer le contenu du déploiement, avec l'interface CLI, saisir la commande `undeploy` avec le nom du fichier du déploiement de l'application.

```
[standalone@localhost:9999 /] undeploy test-application.war
```

Pour supprimer le déploiement de l'application et conserver le contenu du déploiement, avec l'interface CLI, saisir la commande `undeploy` avec le nom du fichier du déploiement de l'application et le paramètre `--keep-content`.

```
[standalone@localhost:9999 /] undeploy test-application.war --keep-content
```

Résultat

Le déploiement de l'application spécifiée est maintenant retiré. Notez que la commande `undeploy` ne produit pas de sortie dans l'interface CLI si elle réussit.

[Rapporter un bogue](#)

10.3.4. Déployer une application dans un domaine géré à l'aide de l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)
- [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#)

Procédure 10.6. Déployer une application dans un domaine géré

- **Exécuter la commande `deploy`**

Par l'intermédiaire de l'interface CLI, saisir la commande `deploy` ainsi que le chemin d'accès vers le déploiement de l'application. Inclure le paramètre `--all-server-groups` afin de déployer tous les groupes de serveurs.

```
[domain@localhost:9999 /] deploy /path/to/test-application.war --all-server-groups
```

- Sinon, définir des groupes de serveurs particuliers de déploiement avec le paramètre `--server-groups`.

```
[domain@localhost:9999 /] deploy /path/to/test-application.war --server-groups=server_group_1,server_group_2
```

Veuillez noter qu'un déploiement réussi ne crée pas de sortie vers le CLI.

Résultat

L'application indiquée est maintenant déployée dans un groupe de serveurs dans votre domaine géré.

[Rapporter un bogue](#)

10.3.5. Supprimer le déploiement d'une application dans un domaine géré à l'aide de l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)
- [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#)
- [Section 10.3.4, « Déployer une application dans un domaine géré à l'aide de l'interface CLI »](#)

Procédure 10.7. Supprimer le déploiement d'une application dans un domaine géré

- **Exécuter la commande `undeploy`**

Par l'intermédiaire de l'interface CLI, saisir la commande `undeploy` ainsi que le nom de fichier

du déploiement de l'application. On peut retirer le déploiement de l'application à partir de n'importe quel groupe de serveur dans lequel elle a été déployée à l'origine en ajoutant le paramètre `--all-relevant-server-groups`.

```
[domain@localhost:9999 /] undeploy test-application.war --all-relevant-server-groups
```

Veillez noter qu'une suppression de déploiement réussie ne crée pas de sortie vers le CLI.

Résultat

L'application spécifiée n'est désormais plus déployée.

[Rapporter un bogue](#)

10.4. DÉPLOYER PAR L'API HTTP

10.4.1. Déployer une application par l'API HTTP

Résumé

Les applications peuvent être déployées via l'API HTTP en suivant les instructions suivantes.

Procédure 10.8. Déployer une application par `DeployDmrToJson.java`

1. Utiliser `DeployDmrToJson.java` pour créer une requête à JSON de déploiement de votre application.

Exemple 10.2. `DeployDmrToJson.java` class

```
import org.jboss.dmr.ModelNode;
import java.net.URL;

public class DeployDmrToJson
{
    public static void main(String[] args) throws Exception
    {
        if(args.length < 1)
            throw new IllegalArgumentException("The first argument must
be a URL");

        URL url = new URL(args[0]);
        String[] pathElements = url.getFile().split("/");
        String name = pathElements[pathElements.length-1];

        ModelNode deploy = getDeploy(url.toExternalForm(), name);
        ModelNode undeploy = getUndeploy(name);

        System.out.println("Deploy\n-----
\n");
        System.out.println("Formatted:\n" +
deploy.toJSONString(false));
        System.out.println("Unformatted:\n" +
deploy.toJSONString(true));
        System.out.println("\nUndeploy\n-----
```

```

--\n");
    System.out.println("Formatted:\n" +
undeploy.toJSONString(false));
    System.out.println("Unformatted:\n" +
undeploy.toJSONString(true));
    }

    public static ModelNode getUndeploy(String name)
    {
        ModelNode undeployRequest = new ModelNode();
        undeployRequest.get("operation").set("undeploy");
        undeployRequest.get("address", "deployment").set(name);

        ModelNode removeRequest = new ModelNode();
        removeRequest.get("operation").set("remove");
        removeRequest.get("address", "deployment").set(name);

        ModelNode composite = new ModelNode();
        composite.get("operation").set("composite");
        composite.get("address").setEmptyList();
        final ModelNode steps = composite.get("steps");
        steps.add(undeployRequest);
        steps.add(removeRequest);
        return composite;
    }

    public static ModelNode getDeploy(String url, String name)
    {
        ModelNode deployRequest = new ModelNode();
        deployRequest.get("operation").set("deploy");
        deployRequest.get("address", "deployment").set(name);

        ModelNode addRequest = new ModelNode();
        addRequest.get("operation").set("add");
        addRequest.get("address", "deployment").set(name);
        addRequest.get("content").get(0).get("url").set(url);

        ModelNode composite = new ModelNode();
        composite.get("operation").set("composite");
        composite.get("address").setEmptyList();
        final ModelNode steps = composite.get("steps");
        steps.add(addRequest);
        steps.add(deployRequest);
        return composite;
    }
}
}

```

2. Exécuter la classe par une commande basée sur les instructions suivantes :

Exemple 10.3. Commande Execute

```

java -cp .:$JBOSS_HOME/modules/org/jboss/dmr/main/jboss-
dmr-1.1.1.Final-redhat-1.jar DeployDmrToJson \
file:///Users/username/support/helloWorld.war/dist/helloWorld.war

```

3. Quand la classe est exécutée, les formats de commande suivants seront affichés. Utiliser la commande `deploy` ou la commande `undeploy` selon vos besoins.

Exemple 10.4. Commande de déploiement et de suppression de déploiement

Deploy

Formatted:

```
{
  "operation" : "composite",
  "address" : [],
  "steps" : [
    {
      "operation" : "add",
      "address" : {"deployment" : "helloWorld.war"},
      "content" : [{"url" :
"file:/Users/username/support/helloWorld.war/dist/helloWorld.war"}
]
    },
    {
      "operation" : "deploy",
      "address" : {"deployment" : "helloWorld.war"}
    }
  ]
}
```

Unformatted:

```
{"operation" : "composite", "address" : [], "steps" :
[{"operation" : "add", "address" : {"deployment" :
"helloWorld.war"}, "content" : [{"url" :
"file:/Users/username/support/helloWorld.war/dist/helloWorld.war"}
]},{ "operation" : "deploy", "address" : {"deployment" :
"helloWorld.war"}}]}
```

Undeploy

Formatted:

```
{
  "operation" : "composite",
  "address" : [],
  "steps" : [
    {
      "operation" : "undeploy",
      "address" : {"deployment" : "helloWorld.war"}
    },
    {
      "operation" : "remove",
      "address" : {"deployment" : "helloWorld.war"}
    }
  ]
}
```

Unformatted:

```
{"operation" : "composite", "address" : [], "steps" :
```

```
[{"operation" : "undeploy", "address" : {"deployment" :
"helloWorld.war"}}, {"operation" : "remove", "address" :
{"deployment" : "helloWorld.war"}}]
```

- Utiliser la commande suivante pour déployer ou supprimer le déploiement d'une application. Remplacer `json request` par la requête décrite ci-dessus.

Exemple 10.5. Commande Execute

```
curl -f --digest -u "<user>:<pass>" -H Content-Type:\
application/json -d 'json request'\
"http://localhost:9990/management"
```

[Rapporter un bogue](#)

10.5. DÉPLOYER AVEC LE SCANNEUR DE DÉPLOIEMENT

10.5.1. Gérer le déploiement d'applications dans le scanneur de déploiement

Déployer des applications dans une instance de serveur autonome par l'intermédiaire d'un scanneur de déploiement vous permet de créer et de tester des applications d'une manière adaptée aux cycles de développement rapides. Vous pouvez configurer le scanneur de déploiement en fonction de vos besoins de fréquence de déploiement et de comportement pour une variété de types d'applications.

[Rapporter un bogue](#)

10.5.2. Déployer une application dans une instance de serveur autonome par un scanneur de déploiement

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#)

Résumé

Cette tâche présente une méthode de déploiement des applications dans une instance de serveur autonome par le scanneur de déploiement. Comme il est indiqué dans la section [Section 10.1, « Les déploiements d'applications »](#), cette méthode est retenue pour la commodité des développeurs, où les méthodes de console de gestion ou d'interface CLI sont recommandées pour la gestion des applications dans les environnements de production.

Procédure 10.9. Utiliser le scanneur de déploiement pour déployer les applications.

1. Copier le contenu dans le dossier de déploiement

Copier le fichier de l'application dans un dossier de déploiement qui se situe `EAP_HOME/standalone/deployments/`.

2. Modes d'analyses de déploiements

Il existe deux méthodes de déploiement. Vous pouvez choisir entre les modes de traitement manuel ou automatique. Avant de démarrer une des méthodes de déploiement, lisez ceci [Section 10.5.8, « Configurer le scanneur de déploiement avec l'interface CLI »](#).

- **Déploiement automatique**

Le scanner de déploiement saisit un changement d'état d'un dossier et crée un fichier de marquage, comme expliqué dans la section [Section 10.5.8, « Configurer le scanner de déploiement avec l'interface CLI »](#).

- **Déploiement manuel**

Le scanner de déploiement a besoin d'un fichier de marqueurs pour déclencher le processus de déploiement. L'exemple suivant utilise la commande Unix `touch` pour créer un nouveau fichier `.dodeploy`.

Exemple 10.6. Déploiement par la commande `touch`

```
[user@host bin]$ touch  
$EAP_HOME/standalone/deployments/example.war.dodeploy
```

Résultat

Le fichier de l'application est déployé sur le serveur d'applications. Un fichier de marquage est créé dans le dossier de déploiement pour indiquer la réussite du déploiement, et l'application est marquée comme `Enabled` dans la console de gestion.

Exemple 10.7. Contenu du dossier de déploiement après le déploiement.

```
example.war  
example.war.deployed
```

[Rapporter un bogue](#)

10.5.3. Supprimer le déploiement d'une application dans une instance de serveur autonome par un scanner de déploiement

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#)
- [Section 10.5.2, « Déployer une application dans une instance de serveur autonome par un scanner de déploiement »](#)

Résumé

Cette tâche présente une méthode de retrait de déploiement d'applications en provenance d'une instance de serveur autonome qui ont été déployées par le scanner de déploiement. Comme il est indiqué dans la section [Section 10.1, « Les déploiements d'applications »](#), cette méthode est retenue pour la commodité des développeurs, où les méthodes de console de gestion ou d'interface CLI sont recommandées pour la gestion des applications dans les environnements de production.



NOTE

Le scanneur de déploiement ne devrait pas servir en conjonction avec d'autres méthodes de déploiement pour la gestion des applications. Les applications supprimées du serveur d'applications par la console de gestion seront retirées du runtime sans affecter les fichiers de marquage ou l'application contenue dans le répertoire de déploiement. Pour minimiser le risque de redéploiement accidentel ou autre erreur, utilisez l'interface CLI et la console de gestion pour l'administration dans des environnements de production.

Procédure 10.10. Retirer le déploiement d'une application à l'aide d'une des méthodes suivantes

- **Supprimer le déploiement de l'application**

Il existe deux méthodes pour supprimer un déploiement d'application suivant que vous souhaitez supprimer l'application d'un dossier de déploiement ou bien que vous souhaitez uniquement modifier son statut de déploiement.

- **Retirer un déploiement par suppression du fichier de marquage**

Supprimer le fichier de marquage `example.war.deployed` de l'application qui est déployée pour commencer le retrait de déploiement de l'application du runtime.

Résultat

Le scanneur de déploiement retire l'application et crée un fichier de marquage `example.war.undeployed`. L'application demeure dans le dossier de déploiement.

- **Retirer le déploiement en supprimant l'application**



NOTE

Annuler le déploiement d'un fichier WAR explosé à l'aide de cette méthode n'est pas valide. Seule une annulation en supprimant le fichier de marqueurs est valide. Tenter d'annuler le déploiement d'un fichier WAR explosé se traduira par un message ressemblant au message suivant enregistré.

```
WARN [org.jboss.as.server.deployment.scanner]
(DeploymentScanner-threads - 2) JBAS015006: The
deployment scanner found that the content for exploded
deployment EXAMPLE.war has been deleted, but auto-
deploy/undeploy for exploded deployments is not
enabled and the EXAMPLE.war.deployed marker file for
this deployment has not been removed. As a result, the
deployment is not being undeployed, but resources
needed by the deployment may have been deleted and
application errors may occur. Deleting the
EXAMPLE.war.deployed marker file to trigger undeploy
is recommended.
```

Supprimer l'application depuis le répertoire de déploiement pour encourager le scanneur de déploiement à commencer le retrait du déploiement de l'application du runtime.

Résultat

Le scanneur de déploiement retire l'application et crée un fichier de marquage `filename.filetype.undeployed`. L'application n'est plus présente dans le dossier

de déploiement.

Résultat

Le fichier de l'application n'est plus déployé dans le serveur d'applications et n'est plus visible dans l'écran `Deployments` de la console de gestion.

[Rapporter un bogue](#)

10.5.4. Redéploiement d'une application dans une instance de serveur autonome par le scanneur de déploiement

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#)
- [Section 10.5.2, « Déployer une application dans une instance de serveur autonome par un scanneur de déploiement »](#)

Résumé

Cette tâche présente une méthode de retrait de déploiement d'applications dans une instance de serveur autonome qui a été déployée par le scanneur de déploiement. Comme il est indiqué dans la section [Section 10.1, « Les déploiements d'applications »](#), cette méthode est retenue pour la commodité des développeurs, où les méthodes de console de gestion ou d'interface CLI sont recommandées pour la gestion des applications dans les environnements de production.

Procédure 10.11. Déployer à nouveau une application dans un serveur autonome

- **Redéploiement de l'application**
Il existe trois méthodes possibles pour redéployer une application déployée par le scanneur de déploiement. Ces méthodes déclenchent le scanneur de déploiement pour initier un cycle de déploiement, et peuvent être choisies en fonction des préférences personnelles.
 - **Redéploiement par modification du fichier de marquage**
Déclencher le redéploiement du scanneur de déploiement en modifiant l'horodatage d'accès du fichier. Dans l'exemple Linux suivant, on utilise une commande Unix `touch`.

Exemple 10.8. Redéployer par la commande Unix `touch`

```
[user@host bin]$ touch
EAP_HOME/standalone/deployments/example.war.dodeploy
```

Résultat

Le scanneur de déploiement a détecté un changement dans le fichier de marquage et a déployé à nouveau l'application. Un nouveau fichier de marquage `.deployed` remplace le précédent.

- **Déployer à nouveau en créant un nouveau fichier de marquage `.dodeploy`**
Déclencher le redéploiement du scanneur de déploiement en créant un nouveau fichier de marquage `.dodeploy`. Voir les instructions de déploiement du manuel qui se trouvent dans [Section 10.5.2, « Déployer une application dans une instance de serveur autonome »](#)

par un scanner de déploiement ».

- **Déployer à nouveau en supprimant le fichier de marquage**

Comme décrit dans [Section 10.5.5, « Référence pour les fichiers de marquage de scanner de déploiement »](#), la suppression du fichier de marquage `.deployed` va déclencher un retrait de déploiement et créera un marqueur `.undeployed`. Supprimer le marqueur de suppression de déploiement déclenchera le cycle de déploiement à nouveau. Voir [Section 10.5.3, « Supprimer le déploiement d'une application dans une instance de serveur autonome par un scanner de déploiement »](#) pour obtenir des informations supplémentaires.

Résultat

Le fichier de l'application est déployé à nouveau.

[Rapporter un bogue](#)

10.5.5. Référence pour les fichiers de marquage de scanner de déploiement

Les fichiers de marquage

Les fichiers de marquage font partie du sous-système du scanner de déploiement. Ces fichiers indiquent le statut d'une application dans un répertoire de déploiement de l'instance du serveur autonome. Un fichier de marquage a le même nom que l'application, avec un suffixe de fichier qui indique l'état du déploiement de l'application. Le tableau suivant définit les types et les réponses de chaque fichier de marquage.

Exemple 10.9. Exemple de fichier de marquage

L'exemple suivant montre un fichier de marquage d'une instance déployée réussie pour une application nommée `testapplication.war`.

```
testapplication.war.deployed
```

Tableau 10.1. Définitions de types de fichiers de marquage

Suffixe du nom de fichier	Origine	Description
<code>.dodeploy</code>	Utilisateur généré	Indique que le contenu doit être déployé ou redéployé dans le runtime.
<code>.skipdeploy</code>	Utilisateur généré	Désactive l'autodéploiement d'une application si présent. Utile pour bloquer temporairement l'auto-déploiement d'un contenu explosé, empêchant ainsi le risque de modifications de contenu incomplètes d'être rendues live. Peut être utile pour le contenu compressé, bien que le scanner détecte les changements en cours dans le contenu compressé et attend qu'ils soient terminés.
<code>.isdeploying</code>	Système généré	Indique l'initiation du déploiement. Le fichier de marquage sera effacé quand le processus de déploiement sera complété.

Suffixe du nom de fichier	Origine	Description
.deployed	Système généré	Indique que le contenu a été déployé. Le déploiement du contenu sera supprimé si le fichier est effacé.
.failed	Système généré	Indique les échecs de déploiement. Le fichier de marquage contient des informations sur la cause de l'échec. Si le fichier de marquage est supprimé, le contenu sera rendu visible dans l'auto-déploiement à nouveau.
.isundeploying	Système généré	Indique une réponse suite à la suppression d'un fichier .deployed . Le déploiement du contenu sera supprimé et le marqueur sera effacé automatiquement dès complétion.
.undeployed	Système généré	Indique si le contenu a été déployé. La suppression du fichier de marquage n'a pas d'impact sur le re-déploiement du contenu.
.pending	Système généré	Indique que les instructions de déploiement devront être envoyées au serveur suite à la résolution d'un problème qui a été détecté. Ce marqueur sert de blocage de déploiement global. Le scanneur ne demandera pas au serveur de déployer ou de supprimer un déploiement de contenu tant que cette condition existe.

[Rapporter un bogue](#)

10.5.6. Référence pour attributs de scanneur de déploiement

Le scanneur de déploiement contient les attributs suivants, qui sont exposés dans l'interface CLI et qui peuvent être configurés par l'opération `write-attribute`. Pour plus d'informations sur les options de configuration, se référer à la section suivante [Section 10.5.8, « Configurer le scanneur de déploiement avec l'interface CLI »](#).

Tableau 10.2. Attributs de scanneur de déploiement

Nom	Description	Type	Valeur par défaut
auto-deploy-exploded	Permet le déploiement automatique d'un contenu éclaté sans nécessiter un fichier de marquage .dodeploy . Recommandé pour les scénarios de développement de base uniquement pour empêcher le déploiement d'applications éclatées de se produire lors de changements du développeur ou du système d'exploitation.	Booléen	False
auto-deploy-xml	Autorise le déploiement automatique d'un contenu XML sans besoin de fichier de marquage .dodeploy .	Booléen	True

Nom	Description	Type	Valeur par défaut
auto-deploy-zipped	Autorise le déploiement automatique d'un contenu compressé sans besoin de fichier de marquage <code>.dodeploy</code> .	Booléen	True
deployment-timeout	La durée nécessaire en secondes pour que le scanneur de déploiement puisse permettre un déploiement avant annulation.	Long	600
path	Définit le chemin du système de fichier à scanner. Si l'attribut relative-to est spécifié, la valeur path agira comme un ajout relatif à ce répertoire ou chemin d'accès.	String	déploiements
relative-to	Référence à un chemin de système de fichier défini dans la section paths du fichier de configuration XML du serveur.	String	jboss.server.base.dir
scan-enabled	Autorise le scanning automatique des applications par scan-interval au démarrage.	Booléen	True
scan-interval	L'intervalle en millisecondes entre les balayages de référentiels. Une valeur inférieure à 1 empêche le scanneur d'opérer au démarrage.	Int	5000

[Rapporter un bogue](#)

10.5.7. Configurer le scanneur de déploiement

Le scanner de déploiement peut être configuré à l'aide de la console de gestion ou l'interface CLI. Vous pouvez créer un nouveau scanneur de déploiement ou bien gérer les attributs existants de scanneur, c'est à dire l'intervalle de balayage, l'emplacement du dossier de déploiement et les types de fichiers d'application qui déclencheront un déploiement.

[Rapporter un bogue](#)

10.5.8. Configurer le scanneur de déploiement avec l'interface CLI

Conditions préalables

- [Section 3.5.2, « Lancement de l'interface CLI »](#)

Résumé

Bien qu'il existe plusieurs méthodes de configuration du scanneur de déploiement, l'interface CLI permet d'exposer et de modifier les attributs par l'utilisation de scripts de lots ou en temps réel. Vous pouvez modifier le comportement du scanneur de déploiement par l'utilisation de l'attribut lecture

`read-attribute` et des opérations de ligne de commande `write-attribute`. Davantage d'informations sur les attributs de scanner de déploiement sont définis dans la rubrique [Section 10.5.6, « Référence pour attributs de scanner de déploiement »](#).

Le scanner de déploiement est un sous-système de JBoss EAP 6, que vous pouvez voir dans `standalone.xml`.

Exemple 10.10. Excerpt de `standalone.xml`

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">
  <deployment-scanner path="deployments" relative-
to="jboss.server.base.dir" scan-interval="5000"/>
</subsystem>
```

Procédure 10.12. Configurer le scanner de déploiement

1. Déterminer les attributs de scanner de déploiement à configurer

Pour configurer le scanner de déploiement par l'interface CLI, vous devrez tout d'abord exposer les noms d'attribut qui conviennent. Vous pouvez faire cela grâce à l'opération `read-resources` au nœud `root`, ou bien par la commande `cd` pour passer au nœud dépendant du sous-système. Vous pouvez également afficher les attributs par la commande `ls` à ce niveau.

- o **Exposer les attributs de scanner de déploiement par l'opération `read-resource`**
Utiliser l'opération `read-resource` pour exposer les attributs définis par la ressource de scanner de déploiement par défaut.

Exemple 10.11. Exemple de sortie `read-resource`

```
[standalone@localhost:9999 /]/subsystem=deployment-
scanner/scanner=default:read-resource
{
  "outcome" => "success",
  "result" => {
    "auto-deploy-exploded" => false,
    "auto-deploy-xml" => true,
    "auto-deploy-zipped" => true,
    "deployment-timeout" => 600,
    "path" => "deployments",
    "relative-to" => "jboss.server.base.dir",
    "scan-enabled" => true,
    "scan-interval" => 5000
  }
}
```

- o **Exposer les attributs de scanner de déploiement par la commande `ls`**
Utiliser la commande `ls` avec l'argument `-l` en option pour afficher une table de résultats qui incluent des attributs de nœud, des valeurs et types de sous-système. Vous pouvez en apprendre davantage sur la commande `ls` et ses arguments en exposant l'entrée `ls --help`. Pour plus d'informations sur le menu `help` de l'interface CLI, voir la section [Section 3.5.5, « Comment obtenir de l'aide par l'interface CLI »](#).

Exemple 10.12. Exemple de sortie `ls -l`

```
[standalone@localhost:9999 /] ls -l /subsystem=deployment-
scanner/scanner=default
ATTRIBUTE          VALUE          TYPE
auto-deploy-exploded false          BOOLEAN
auto-deploy-xml    true           BOOLEAN
auto-deploy-zipped true           BOOLEAN
deployment-timeout 600           LONG
path               deployments    STRING
relative-to        jboss.server.base.dir STRING
scan-enabled       true           BOOLEAN
scan-interval      5000          INT
```

2. Configurer le scanneur de déploiement par l'opération `write-attribute`

Une fois que vous avez déterminé le nom de l'attribut à modifier, utiliser la commande `write-attribute` pour spécifier le nom de l'attribut et la nouvelle valeur à indiquer. Les exemples suivants sont tous exécutés au niveau du nœud dépendant, qui peut être accédé en utilisant la commande `cd` et la saisie semi-automatique via la touche `TAB` pour passer au nœud de scanneur par défaut.

```
[standalone@localhost:9999 /] cd subsystem=deployment-
scanner/scanner=default
```

a. Activer le déploiement automatique du contenu explosé.

Utiliser l'opération `write-attribute` pour activer le déploiement automatique du contenu d'application explosé.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=auto-deploy-exploded,value=true)
{"outcome" => "success"}
```

b. Désactiver le déploiement automatique du contenu XML

Utiliser l'opération `write-attribute` pour désactiver le déploiement automatique du contenu d'application explosé.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=auto-deploy-xml,value=false)
{"outcome" => "success"}
```

c. Désactiver le déploiement automatique du contenu compressé

Utiliser la commande `write-attribute` pour désactiver le déploiement automatique du contenu d'applications compressé.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=auto-deploy-zipped,value=false)
{"outcome" => "success"}
```

d. Configurer l'attribut du chemin d'accès

Utiliser l'opération `write-attribute` pour modifier l'attribut de chemin d'accès, pour substituer la valeur de l'exemple `newpathname` par un nouveau nom de chemin d'accès

que le scanneur de déploiement puisse surveiller. Noter que le serveur aura besoin que le nouveau chargement prenne effet.

```
[standalone@localhost:9999 scanner=default] :write-attribute(name=path,value=newpathname)
{
  "outcome" => "success",
  "response-headers" => {
    "operation-requires-reload" => true,
    "process-state" => "reload-required"
  }
}
```

e. Configurer l'attribut du chemin relatif

Utiliser l'opération `write-attribute` pour modifier la référence relative du chemin du système de fichier ainsi définie dans la section des chemins d'accès du fichier de configuration XML. Notez que le serveur aura besoin que le nouveau chargement prenne effet.

```
[standalone@localhost:9999 scanner=default] :write-attribute(name=relative-to,value=new.relative.dir)
{
  "outcome" => "success",
  "response-headers" => {
    "operation-requires-reload" => true,
    "process-state" => "reload-required"
  }
}
```

f. Désactiver le scanneur de déploiement

Utiliser l'opération `write-attribute` pour désactiver le scanneur de déploiement en définissant la valeur `scan-enabled` à `false`.

```
[standalone@localhost:9999 scanner=default] :write-attribute(name=scan-enabled,value=false)
{"outcome" => "success"}
```

g. Changer l'intervalle de balayage

Utiliser l'opération `write-attribute` pour modifier l'intervalle de balayage de 5 000 millisecondes à 10 000 millisecondes.

```
[standalone@localhost:9999 scanner=default] :write-attribute(name=scan-interval,value=10000)
{"outcome" => "success"}
```

Résultat

Vos modifications de configuration sont sauvegardées dans le scanneur de déploiement.

[Rapporter un bogue](#)

10.6. DÉPLOYER AVEC MAVEN

10.6.1. Gestion du déploiement d'applications dans Maven

Le déploiement d'applications dans Maven vous permet d'incorporer un cycle de déploiement dans votre flux de développement existant.

[Rapporter un bogue](#)

10.6.2. Déployer une application dans Maven

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#)

Résumé

Cette tâche vous montre une méthode pour déployer des applications dans Maven. L'exemple fourni utilise l'application `jboss-helloworld.war` qui se trouve dans la collection Jboss EAP 6 Quickstarts. Le projet `helloworld` contient un fichier POM qui initialise le `jboss-as-maven-plugin`. Ce plugin fournit des simples opérations pour déployer ou supprimer le déploiement d'applications vers ou en provenance du serveur d'applications.

Procédure 10.13. Déployer une application dans Maven.

1. Ouvrir la session de terminal et naviguer dans le répertoire qui contient les exemples Quickstart.

Exemple 10.13. Passer au répertoire d'application helloworld

```
[localhost]$ cd /QUICKSTART_HOME/helloworld
```

2. Exécuter la commande de déploiement Maven pour déployer l'application. Si l'application est déjà en cours d'exécution, elle sera déployée à nouveau.

```
[localhost]$ mvn package jboss-as:deploy
```

3. Afficher les résultats.

- Le déploiement peut être confirmé si vous regardez les entrées de journalisation de l'opération dans la fenêtre du terminal.

Exemple 10.14. Confirmation Maven pour l'application helloworld

```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 32.629s
[INFO] Finished at: Fri Mar 14 09:09:50 EDT 2014
[INFO] Final Memory: 23M/204M
[INFO] -----
```


- o Le déploiement peut également être confirmé dans le flux de statut de l'instance du serveur d'applications actives.

Exemple 10.15. Confirmation du serveur d'applications pour l'application helloworld

```
09:09:49,167 INFO [org.jboss.as.repository] (management-
handler-thread - 1) JBAS014900: Content added at location
/home/username/EAP_HOME/standalone/data/content/32/4b4ef9a4bbe7
206d3674a89807203a2092fc70/content
09:09:49,175 INFO [org.jboss.as.server.deployment] (MSC
service thread 1-7) JBAS015876: Starting deployment of "jboss-
helloworld.war" (runtime-name: "jboss-helloworld.war")
09:09:49,563 INFO [org.jboss.weld.deployer] (MSC service
thread 1-8) JBAS016002: Processing weld deployment jboss-
helloworld.war
09:09:49,611 INFO [org.jboss.weld.deployer] (MSC service
thread 1-1) JBAS016005: Starting Services for CDI deployment:
jboss-helloworld.war
09:09:49,680 INFO [org.jboss.weld.Version] (MSC service thread
1-1) WELD-000900 1.1.17 (redhat)
09:09:49,705 INFO [org.jboss.weld.deployer] (MSC service
thread 1-2) JBAS016008: Starting weld service for deployment
jboss-helloworld.war
09:09:50,080 INFO [org.jboss.web] (ServerService Thread Pool -
- 55) JBAS018210: Register web context: /jboss-helloworld
09:09:50,425 INFO [org.jboss.as.server] (management-handler-
thread - 1) JBAS018559: Deployed "jboss-helloworld.war"
(runtime-name : "jboss-helloworld.war")
```

Résultat

L'application est déployée dans le serveur d'applications.

[Rapporter un bogue](#)

10.6.3. Supprimer le déploiement d'une application dans Maven

Conditions préalables

- [Section 2.2.1, « Démarrer JBoss EAP 6 »](#)

Résumé

Cette tâche vous montre une méthode pour supprimer le déploiement des applications dans Maven. L'exemple fourni utilise l'application `jboss-helloworld.war` qui se trouve dans la collection Jboss EAP 6 Quickstarts. Le projet `helloworld` contient un fichier POM qui initialise le `jboss-as-maven-plugin`. Ce plugin fournit des simples opérations pour déployer ou supprimer le déploiement d'applications vers ou en provenance du serveur d'applications.

Procédure 10.14. Supprimer le déploiement d'une application dans Maven

1. Ouvrir la session de terminal et naviguer dans le répertoire qui contient les exemples Quickstart.

Exemple 10.16. Passer au répertoire d'application helloworld

```
[localhost]$ cd /QUICKSTART_HOME/helloworld
```

2. Exécuter la commande de suppression de déploiement.

```
[localhost]$ mvn jboss-as:undeploy
```

3. Afficher les résultats.

- o La suppression du déploiement peut être confirmée si on observe les entrées de journalisation de la fenêtre de terminal.

Exemple 10.17. Confirmation Maven pour la suppression du déploiement de l'application helloworld.

```
[INFO] -----
-----
[INFO] BUILD SUCCESSFUL
[INFO] -----
-----
[INFO] Total time: 1 second
[INFO] Finished at: Mon Oct 10 17:33:02 EST 2011
[INFO] Final Memory: 11M/212M
[INFO] -----
-----
```

- o La suppression du déploiement peut également être confirmée dans le flux de statut de l'instance du serveur d'applications actives.

Exemple 10.18. Confirmation du serveur d'applications pour la suppression du déploiement de l'application helloworld.

```
09:51:40,512 INFO [org.jboss.web] (ServerService Thread Pool -
- 69) JBAS018224: Unregister web context: /jboss-helloworld
09:51:40,522 INFO [org.jboss.weld.deployer] (MSC service
thread 1-3) JBAS016009: Stopping weld service for deployment
jboss-helloworld.war
09:51:40,536 INFO [org.jboss.as.server.deployment] (MSC
service thread 1-1) JBAS015877: Stopped deployment jboss-
helloworld.war (runtime-name: jboss-helloworld.war) in 27ms
09:51:40,621 INFO [org.jboss.as.repository] (management-
handler-thread - 10) JBAS014901: Content removed from location
/home/username/EAP_HOME/jboss-eap-
6.4/standalone/data/content/44/e1f3c55c84b777b0fc201d69451223c0
9c9da5/content
```

```
09:51:40,621 INFO [org.jboss.as.server] (management-handler-
thread - 10) JBAS018558: Undeployed "jboss-helloworld.war"
(runtime-name: "jboss-helloworld.war")
```

Résultat

La suppression du déploiement de l'application provient du serveur d'applications.

[Rapporter un bogue](#)

10.7. CONTRÔLER L'ORDRE DES APPLICATIONS DÉPLOYÉES DANS JBOSS EAP 6

JBoss EAP 6 offre un contrôle à grain fin sur l'ordre de déploiement d'applications lors du démarrage du serveur. L'ordre strict de déploiement d'applications présentes dans plusieurs fichiers ear peut être activé avec la persistance de cet ordre après un redémarrage.

Procédure 10.15. Contrôler l'ordre de déploiement dans EAP 6.0

1. Crée des scripts CLI qui déploient et retirent les déploiements d'applications dans un ordre séquentiel quand le serveur est à l'arrêt/démarrage.
2. CLI prend également en charge le concept de mode batch qui permet de grouper les commandes et les opérations et les exécuter ensemble comme une unité atomique. Si au moins une des commandes ou opérations échoue, toutes les autres commandes et opérations exécutées avec succès dans le lot seront annulées.

Procédure 10.16. Contrôler l'ordre de déploiement dans EAP 6.1 et versions ultérieures

La nouvelle fonctionnalité nommée Inter Deployment Dependencies d'EAP 6.1, et versions ultérieures, vous permet de déclarer des dépendances entre les niveaux supérieurs de déploiement.

1. Créer (s'il n'existe pas encore) un fichier `jboss-all.xml` dans le dossier `app.ear/META-INF` où `app.ear` est l'archive d'application qui dépend d'une autre archive d'application à déployer avant.
2. Effectuer une saisie `jboss-deployment-dependencies` dans ce fichier comme indiqué ci-dessous. Notez que dans la liste ci-dessous, `framework.ear` est l'application de dépendance qui doit être déployée avant que l'archive d'application `app.ear` ne le soit.

```
<jboss xmlns="urn:jboss:1.0">
  <jboss-deployment-dependencies xmlns="urn:jboss:deployment-
dependencies:1.0">
    <dependency name="framework.ear" />
  </jboss-deployment-dependencies>
</jboss>
```

[Rapporter un bogue](#)

10.8. REMPLACEMENT DU DESCRIPTEUR DE DÉPLOIEMENT

Une nouvelle fonctionnalité présente à partir de la version EAP 6.1 vous permet de remplacer des descripteurs de déploiement, des JAR, des classes, des pages JSP, et d'autres fichiers lors du temps

d'exécution. Un *deployment overlay* représente un ensemble de règles de fichiers devant être remplacés dans les archives. Il fournit également des liens vers les nouveaux fichiers qui doivent être utilisés au lieu de ceux ayant été remplacés. Si le fichier remplacé n'est pas présent dans les archives de déploiement, il sera ajouté au déploiement.

Procédure 10.17. Remplacer le descripteur de déploiement en utilisant l'interface CLI

Les étapes suivantes partent du principe que vous possédez déjà une application déployée appelée `app.war` et que vous souhaitez remplacer son fichier `WEB-INF/web.xml` avec un autre fichier `web.xml` situé dans `/home/user/web.xml`.

1. Ajouter une superposition de déploiement (deployment overlay) et y ajouter du contenu. Vous pouvez y parvenir de deux façons :

- o **Utilisation d'une arborescence DMR**

a. `/deployment-overlay=myoverlay:add`

b. `/deployment-overlay=myoverlay/content=WEB-INF/web.xml:add(content={url=file:///home/user/web.xml})`

Vous pouvez ajouter des règles de contenu supplémentaires en utilisant le deuxième code.

- o **Utilisation des méthodes de facilité**

```
deployment-overlay add --name=myoverlay --content=WEB-INF/web.xml=/home/user/web.xml
```

2. Relie la superposition à une archive de déploiement. Vous pouvez y parvenir de deux façons :

- o **Utilisation d'une arborescence DMR**

`/deployment-overlay=myoverlay/deployment=app.war:add`

- o **Utilisation des méthodes de facilité**

```
deployment-overlay link --name=myoverlay --deployments=app.war
```

Pour spécifier plusieurs noms d'archives, séparez-les par des virgules.

Veillez noter que le nom d'archive de déploiement n'a pas besoin d'exister sur le serveur. Vous devez indiquer le nom, mais il ne sera pas encore lié à un déploiement.

3. **Redéploiement de l'application**

```
/deployment=app.war:redeploy
```

[Rapporter un bogue](#)

CHAPITRE 11. CONFIGURATION DE SOUS-SYSTÈMES

11.1. APERÇU DE LA CONFIGURATION DU SOUS-SYSTÈME

Introduction

JBoss EAP 6 utilise une configuration simplifiée, avec un fichier de configuration par domaine ou par serveur autonome. En mode de domaine, un fichier distinct existe pour chaque contrôleur hôte également. Les modifications apportées à la configuration persistent automatiquement, donc le XML ne doit pas être édité manuellement. La configuration est scannée et automatiquement remplacée par l'API de gestion. La ligne de commande de l'interface CLI et la console de gestion basée-web permettent de configurer chaque aspect de JBoss EAP 6.

JBoss EAP 6 repose sur le concept de chargement de classes modulaire. Chaque API ou service fourni par la plateforme est implémenté comme un module, qui est chargé et déchargé à la demande. La plupart des modules incluent un élément configurable appelé un sous-système. Les informations de configuration du sous-système sont stockées dans le fichier de configuration unifiée `EAP_HOME/domain/configuration/domain.xml` pour un domaine géré ou `EAP_HOME/standalone/configuration/standalone.xml` pour un serveur autonome. La plupart des sous-systèmes incluent les détails de configuration configurés par l'intermédiaire de descripteurs de déploiements dans les versions précédentes de JBoss EAP.

Schémas de configuration du sous-système

Chaque configuration de sous-système est définie dans un schéma XML. Les schémas de configuration se trouvent dans le répertoire `EAP_HOME/docs/schema/` de votre installation.

Les sous-systèmes suivants sont connus comme *sous-systèmes simples*, parce qu'ils n'ont pas d'attributs ou d'éléments configurables. Ils sont généralement répertoriés en haut du fichier de configuration.

Sous-systèmes simples

- `ee`– l'implémentation Java EE 6 API
- `ejb`– le sous-système d'Enterprise JavaBeans (EJB)
- `jaxrs`– l'API JAX-RS API, fourni par RESTeasy.
- `sar`– le sous-système qui supporte Service Archives.
- `threads`– le sous-système qui supporte le traitement des threads.
- `weld`– l'API Contexts and Dependency Injection, fourni par Weld.

[Rapporter un bogue](#)

CHAPITRE 12. LE SOUS-SYSTÈME DE JOURNALISATION

12.1. INTRODUCTION

12.1.1. Logging (Journalisation)

JBoss EAP 6 fournit des fonctionnalités de journalisation hautement configurables pour son propre usage et pour utilisation par des applications déployées. Le sous-système d'enregistrement est basé sur JBoss LogManager et prend en charge plusieurs frameworks de journalisation d'applications de tierce partie en plus du JBoss Logging.

Le sous-système de journalisation est configuré à l'aide d'un système de catégories de journaux et de gestionnaires de journaux. Les catégories de journalisation définissent quels messages capturer et les gestionnaires de journaux définissent comment procéder avec ces messages (écriture sur disque, envoyer à la console, etc.).

Le profils de journalisation permettent à des configurations de journalisation possédant un nom unique d'être créées et assignées à des applications indépendamment de toute autre configuration de journalisation. La configuration des profils de journalisation est presque identique pour le sous-système de journalisation principal.

[Rapporter un bogue](#)

12.1.2. Frameworks de journalisations d'applications pris en charge par JBoss LogManager

JBoss LogManager prend en charge les frameworks de journalisation suivants :

- JBoss Logging - inclus avec JBoss EAP 6
- Apache Commons Logging - <http://commons.apache.org/logging/>
- Simple Logging Facade Java (SLF4J) - <http://www.slf4j.org/>
- Apache log4j - <http://logging.apache.org/log4j/1.2/>
- Java SE Logging (java.util.logging) - <http://download.oracle.com/javase/6/docs/api/java/util/logging/package-summary.html>

[Rapporter un bogue](#)

12.1.3. Journalisation Bootup

JBoss EAP consigne les entrées du démarrage de l'environnement d'exécution Java et du démarrage de chaque service. Ce journal peut être utile pour les dépannages. Par défaut, toutes les entrées de journal sont écrites dans le fichier `server.log`, dont l'emplacement dépendra du mode d'exécution.

Mode autonome

```
EAP_HOME/standalone/log/server.log
```

Mode de domaine

```
EAP_HOME/domain/servers/SERVER_NAME/log/server.log
```

La configuration de la journalisation de Bootup est indiquée dans le fichier de configuration `logging.properties`, et son emplacement dépend du mode d'exécution.

Mode autonome

`EAP_HOME/standalone/configuration/logging.properties`

Mode de domaine

En mode de domaine, il y a un fichier `logging.properties` pour le contrôleur de domaine et pour chaque serveur.

Contrôleur de domaine : `EAP_HOME/domain/configuration/logging.properties`

Serveur : `EAP_HOME/domain/servers/SERVER_NAME/data/logging.properties`

Le fichier de configuration `logging.properties` est actif jusqu'à ce que le sous-système de journalisation démarre et prenne la relève.



AVERTISSEMENT

Il est conseillé de ne pas éditer le fichier `logging.properties` directement, à moins que vous connaissiez un cas d'utilisation particulier qui le requiert. Avant de procéder, on vous conseille de soumettre un dossier de prise en charge.

Les modifications apportées au fichier `logging.properties` sont remplacées au démarrage.

[Rapporter un bogue](#)

12.1.4. Voir les erreurs de démarrage initial (bootup)

Quand on tente de résoudre des problèmes dans JBoss EAP, on doit commencer par vérifier les erreurs survenues au cours du démarrage. Il existe deux méthodes d'affichage des erreurs de démarrage, chacune avec ses avantages. Chaque méthode produit une liste de toutes les erreurs qui se sont produites au cours du démarrage. Utilisez les informations fournies pour diagnostiquer et résoudre leurs causes. Contact Support client de Red Hat pour l'assistance en cas de problème.

- Examiner le fichier journal `server.log`.

Cette méthode vous permet de voir chaque message d'erreur et les messages liés dans certains cas, ce qui vous permet d'obtenir plus d'informations sur la *raison* pour laquelle une erreur a pu avoir lieu. Cela vous permet également de voir les erreurs en format brut.

- À partir de JBoss EAP 6.4, utiliser la commande d'interface de commande `read-boot-errors`.

Cette méthode ne nécessite pas d'accès au système de fichiers du serveur, ce qui est bien utile pour toute personne responsable de contrôler les erreurs et qui n'a pas d'accès au système de fichiers. Comme il s'agit d'une commande d'interface CLI, elle peut être utilisée dans un script.

Par exemple, vous pouvez écrire un script qui démarre par des instances multiples de JBoss EAP, puis vérifier les erreurs qui ont lieu au démarrage.

Procédure 12.1. Examiner server.log pour chercher les erreurs

1. Ouvrir le fichier `server.log` dans une visionneuse de fichiers.
2. Déplacez-vous en fin de fichier.
3. Rechercher l'identifiant du message **JBAS015899** rétroactivement, qui indique le début de la dernière séquence de démarrage.
4. Chercher les instances d'**ERROR** à partir de cet endroit dans le journal. Chaque instance devra inclure une description de l'erreur et la liste des modules concernés.

Exemple 12.1. Description de l'erreur de server.log

Ce qui suit est un exemple de description d'erreur du fichier journal `server.log`.

```
13:23:14,281 ERROR [org.apache.coyote.http11.Http11Protocol] (MSC
service thread 1-4) JBWEB003043: Error initializing endpoint:
java.net.BindException: Address already in use /127.0.0.1:8080
```

Procédure 12.2. Lister les erreurs de démarrage par l'interface CLI

- Exécuter la commande d'interface CLI suivante :

```
/core-service=management:read-boot-errors
```

Toute erreur qui a lieu lors du démarrage sera listée à cet endroit.

L'heure de chaque erreur est définie par la méthode `currentTimeMillis()`, qui correspond à la différence, mesurée en millisecondes, entre l'heure actuelle et minuit le 1er janvier 1970 UTC.

Exemple 12.2. Sortie de la commande `read-boot-errors`

```
{
  "outcome" => "success",
  "result" => [{
    "failed-operation" => {
      "operation" => "add",
      "address" => [
        ("subsystem" => "web"),
        ("connector" => "http")
      ]
    },
    "failure-timestamp" => 1417560953245L,
    "failure-description" => "{\\"JBAS014671: Failed services\\" =>
{\\"jboss.web.connector.http\\" => \\"org.jboss.msc.service.StartException
in service jboss.web.connector.http: JBAS018007: Error starting web
connector
Caused by: LifecycleException: JBWEB000023: Protocol handler
```



```

initialization failed\}}",
    "failed-services" => {"jboss.web.connector.http" =>
"org.jboss.msc.service.StartException in service
jboss.web.connector.http: JBAS018007: Error starting web connector
Caused by: LifecycleException: JBWEB000023: Protocol handler
initialization failed"}
    }}
}

```

[Rapporter un bogue](#)

12.1.5. Journalisation de Garbage collection

La journalisation de Garbage collection consigne toutes les activités de collecte de la poubelle dans les fichiers journaux en texte clair. Ces fichiers journaux peuvent être utiles à des fins de diagnostic. À partir de JBoss EAP 6, la journalisation de la Garbage collection est activée par défaut en mode `standalone` (autonome) sur toutes les configurations prises en charge *sauf* JDK d'IBM.

La journalisation correspond à la sortie du fichier `$EAP_HOME/standalone/log/gc.log.digit`. La rotation de la journalisation est activée, avec un nombre de fichiers de journalisation ne devant pas dépasser cinq, et une taille de trois MiB maximum par fichier.

[Rapporter un bogue](#)

12.1.6. Dépendances d'API de journalisation implicites

Le sous-système de journalisation de JBoss EAP 6 possède un attribut `add-logging-api-dependencies` qui contrôle la possibilité pour le conteneur d'ajouter des dépendances d'API de journalisation implicites aux déploiements. Par défaut, cet attribut est défini à `true`, ce qui signifie que toutes les dépendances d'API de journalisation implicites doivent être ajoutées aux déploiements. Si défini sur `false`, les dépendances d'API de journalisation se seront pas ajoutées.

L'attribut `add-logging-api-dependencies` peut être configuré par l'interface CLI. Par exemple :

```

/subsystem=logging:write-attribute(name=add-logging-api-dependencies,
value=false)

```

[Rapporter un bogue](#)

12.1.7. Emplacements de fichiers de journalisation par défaut

Il s'agit des fichiers de journalisation qui ont été créés pour les configurations de journalisation par défaut. La configuration par défaut écrit des fichiers de journalisation du serveur à l'aide de gestionnaires de journaux périodiques.

Tableau 12.1. Fichier de journalisation par défaut d'un serveur autonome

Fichier journal	Description
<code>EAP_HOME/standalone/log/server.log</code>	Journal du serveur. Contient les messages de journalisation de serveur, dont les messages de démarrage de serveur.

Fichier journal	Description
<code>EAP_HOME/standalone/log/gc.log</code>	Journalisation de Garbage collection. Contient des informations sur tous les nettoyages de mémoire.

Tableau 12.2. Fichiers de journalisation par défaut d'un domaine géré

Fichier journal	Description
<code>EAP_HOME/domain/log/host-controller.log</code>	Journal d'amorçage du contrôleur hôte. Contient les messages de journalisation liés au démarrage du contrôleur hôte.
<code>EAP_HOME/domain/log/process-controller.log</code>	Journal d'amorçage du contrôleur de processus. Contient les messages de journalisation liés au démarrage du contrôleur de processus.
<code>EAP_HOME/domain/servers/SERVERNAME/log/server.log</code>	Le journal du serveur pour le serveur nommé. Contient les messages de journalisation de ce serveur, dont les messages de démarrage de serveur.

[Rapporter un bogue](#)

12.1.8. Filtre les expressions de journalisation

Les expressions de filtrage sont utilisées pour enregistrer les messages de journalisation sur la base d'un certain nombre de critères. Le contrôle de filtrage est toujours effectué sur un message brut non formaté. Vous pouvez inclure un filtre de logger ou de gestionnaire, mais le filtre du logger aura la priorité sur le filtre du gestionnaire.



NOTE

Un `filter-spec` spécifié pour le root logger n'est pas hérité par les autres loggers. Au lieu de cela, un `filter-spec` devra être spécifié pour chaque gestionnaire.

Tableau 12.3. Filtre les expressions de journalisation

Type de filtre expression	Description	Paramètres
Accepter <code>accept</code>	Accepter tous les messages de journalisation	<code>accept</code>
Refuser <code>deny</code>	Refuser tous les messages de journalisation	<code>deny</code>

Type de filtre expression	Description	Paramètres
Not not[filter expression]	Renvoie la valeur inversée de l'expression du filtre	Prend une expression de filtre unique comme paramètre not(match("JBAS"))
Tout all[filter expression]	Renvoie la valeur concaténée à partir de multiples expressions de filtre.	Prends plusieurs expressions de filtre séparées par des virgules all(match("JBAS"), match("WEL D"))
Tous any[filter expression]	Renvoie une valeur à partir de multiples expressions de filtre.	Prends plusieurs expressions de filtre séparées par des virgules any(match("JBAS"), match("WEL D"))
Changement de niveau levelChange[level]	Modifie l'enregistrement de journalisation avec le niveau indiqué	Prend un niveau basé chaîne unique comme argument levelChange("WARN")
Niveaux levels[levels]	Filtre les messages de journalisation avec un niveau apparaissant sur la liste des niveaux	Prend plusieurs niveaux basés chaînes séparées par des virgules en tant qu'arguments levels("DEBUG", "INFO", "WARN", "ERROR")

Type de filtre expression	Description	Paramètres
Gamme de niveaux <code>levelRange[minLevel, maxLevel]</code>	Filtre les messages de journalisation dans la gamme de niveaux spécifiée.	L'expression de filtre utilise [pour indiquer un niveau inclusif minimum et] pour indiquer un niveau inclusif maximum. Sinon, vous pouvez utiliser (ou) respectivement pour indiquer une exclusivité. Le premier argument de l'expression est le niveau minimum autorisé, le deuxième argument est le niveau maximum autorisé. Voici des exemples ci-dessous. <ul style="list-style-type: none"> <code>levelRange("DEBUG", "ERROR")</code> Le niveau minimum doit être supérieur à DEBUG et le niveau maximum doit être inférieur à ERROR. <code>levelRange["DEBUG", "ERROR"]</code> Le niveau minimum doit être supérieur ou égal à DEBUG et le niveau maximum doit être inférieur à ERROR. <code>levelRange["INFO", "ERROR"]</code> Le niveau minimum doit être supérieur ou égal à INFO et le niveau maximum doit être inférieur ou égal à ERROR.
Match (<code>match["pattern"]</code>)	Filtre basé sur une expression régulière. Le message non formaté est utilisé à l'encontre du modèle spécifié dans l'expression.	Prend une expression régulière comme argument <code>match("JBAS\d+")</code>
Substitute (<code>substitute["pattern", "replacement value"]</code>)	Filtre qui remplace la première correspondance au modèle par une valeur de remplacement	Le premier argument de l'expression est le modèle, le deuxième est le texte de remplacement. <code>substitute("JBAS", "EAP")</code>

Type de filtre expression	Description	Paramètres
Remplacer tout (<code>substituteAll["pattern", "replacement value"]</code>)	Un filtre qui remplace toutes les correspondances du modèle avec la valeur de remplacement.	Le premier argument de l'expression est le modèle, le deuxième est le texte de remplacement. <code>substituteAll("JBAS", "EAP")</code>

[Rapporter un bogue](#)

12.1.9. Niveaux de journalisation

Les niveaux de journalisation sont des ensembles ordonnés de valeurs énumérées qui indiquent la nature et la sévérité d'un message de journalisation. Le niveau d'un message de journalisation donné est indiqué par le développeur par des méthodes qui conviennent dans un framework de journalisation particulier pour envoyer le message.

JBoss EAP 6 accepte tous les niveaux de journalisation utilisés par les frameworks de journalisation de l'application prise en charge. Les six niveaux de journalisation les plus utilisés sont (dans l'ordre croissant) : **TRACE**, **DEBUG**, **INFO**, **ATTENTION**, **ERREUR** et **FATAL**.

Les niveaux de journalisation sont utilisés par des catégories et des gestionnaires de journalisation pour limiter les messages dont ils sont responsables. Chaque niveau de journalisation possède une valeur numérique qui indique son ordre par rapport à d'autres niveaux de journalisation. Les catégories et gestionnaires de journalisation correspondent à un certain niveau de journalisation, et ils traitent les messages de journalisation du même niveau ou d'un niveau supérieur uniquement. Par exemple, un gestionnaire de journalisation du niveau **ATTENTION** enregistrera uniquement les messages des niveaux **ATTENTION**, **ERREUR** et **FATAL**.

[Rapporter un bogue](#)

12.1.10. Niveaux de journalisation pris en charge

Tableau 12.4. Niveaux de journalisation pris en charge

Niveau de journalisation	Valeur	Description
FINESSE MAX	300	-
PLUS FIN	400	-
TRACE	400	Utilisé pour des messages qui fournissent des informations détaillées sur l'état d'exécution d'une application. Les messages de journalisation TRACE sont habituellement capturés uniquement lors du débogage d'une application.

Niveau de journalisation	Valeur	Description
DEBOG	500	Utilisé pour des messages qui indiquent des demandes individuelles de progrès ou des activités d'une application. Les messages de journalisation DEBUG ne sont habituellement capturés que lors du débogage d'une application.
FINESSE	500	-
CONFIG	700	-
INFO	800	Utilisé pour des messages qui indiquent la progression globale de l'application. Souvent utilisé pour le démarrage de l'application, la fermeture et autres événements majeurs de cycle de vie.
AVERTISSEMENT	900	Utilisé pour indiquer une situation qui n'est pas en erreur, mais n'est pas considérée comme idéale. Peut indiquer des circonstances qui peuvent entraîner des erreurs dans le futur.
ATTENTION	900	-
ERREUR	1000	Utiliser pour indiquer une erreur qui s'est produite et qui puisse empêcher l'activité actuelle ou la demande de se remplir, mais qui n'empêchera pas l'application d'exécuter.
SÉVÈRE	1000	-
FATALE	1100	Utiliser pour indiquer les événements qui pourraient entraîner des défaillances de services critiques ou la fermeture de l'application, ou qui pourraient entraîner la fermeture de la plateforme JBoss EAP 6.

[Rapporter un bogue](#)

12.1.11. Catégories de journalisation

Les catégories de journalisation définissent les messages de journalisation à acquérir et un ou plusieurs gestionnaires de journalisation qui traitent les messages.

Les messages de journalisation à capturer sont définis par leur package Java d'origine et leur niveau de journalisation. Les messages de classes de ce package et de niveau de journalisation ou niveau inférieur sont capturés par la catégorie de journalisation et envoyés aux gestionnaires de journalisation spécifiés.

Les catégories ont la possibilité d'utiliser les gestionnaires de journalisation du root logger au lieu de leurs propres gestionnaires.

[Rapporter un bogue](#)

12.1.12. Root Logger

Le root logger capture tous les messages de journalisation qui sont envoyés au serveur (à un niveau indiqué) et qui ne sont pas capturés par une catégorie de journalisation particulière. Ces messages sont alors envoyés à un ou à plusieurs gestionnaires de journalisation.

Par défaut, le root logger est configuré pour utiliser une console et un gestionnaire de journalisation périodique. Le gestionnaire de journalisation périodique est configuré pour écrire sur le fichier `server.log`. On prénomme parfois ce fichier : journal du serveur (server log).

[Rapporter un bogue](#)

12.1.13. Gestionnaires de journaux

Les gestionnaires de journaux définissent la façon dont les messages de journalisation sont enregistrés dans JBoss EAP. Voici la liste des gestionnaires de journalisation : **Console**, **File**, **Periodic**, **Size**, **Async**, **syslog**, **Periodic Size** et **Custom**.



NOTE

Un gestionnaire de journaux doit être ajouté à un logger au moins pour être actif.

[Rapporter un bogue](#)

12.1.14. Types de gestionnaires de journalisation

Console

Les gestionnaires de journaux de console écrivent des messages de journalisation soit dans le système d'exploitation hôte (`stdout`) ou dans le flux d'erreurs standard (`stderr`). Ces messages sont affichés lorsque JBoss EAP 6 est exécuté à partir d'une invite de ligne de commande. Les messages d'un gestionnaire de journal de Console ne sont pas enregistrés à moins que le système d'exploitation ne soit spécifiquement configuré pour capturer `stdout` ou `stderr`.

Fichier

Les gestionnaires de journaux de fichiers inscrivent des messages de journalisation dans un fichier spécifique.

Périodique

Les gestionnaires de journaux périodiques écrivent des messages de journalisation dans un fichier nommé jusqu'à ce qu'une certaine durée se soit écoulée. Une fois que cette période a expiré, le fichier est renommé à nouveau en rajoutant l'horodatage et le gestionnaire continue d'écrire dans un fichier de journalisation nouvellement créé avec le nom d'origine.

Taille

Les gestionnaires de journaux de Taille écrivent les messages de journalisation dans un fichier jusqu'à ce que le fichier atteigne une taille spécifiée. Lorsque le fichier atteint une taille donnée, il est renommé avec un suffixe numérique et le gestionnaire continue d'écrire dans un fichier journal récemment créé avec le nom d'origine. Chaque gestionnaire de journaux de Taille doit spécifier le nombre maximal de fichiers contenus de cette façon.

Taille périodique

Disponible à partir de JBoss EAP 6.4. Correspond à une combinaison de gestionnaires Périodiques et de Taille, et prend en charge leurs attributs combinés.

Une fois que le fichier de journalisation a atteint une taille donnée, *ou* que la période donnée a expiré, le fichier est renommé à nouveau en rajoutant l'horodatage et le gestionnaire continue d'écrire dans un fichier de journalisation nouvellement créé avec le nom d'origine.

Async

Les gestionnaires de journaux async sont des gestionnaires de journaux wrapper qui fournissent un comportement asynchrone pour un ou plusieurs autres gestionnaires de journaux. Ils sont utiles pour les gestionnaires de journaux qui pourraient avoir une latence élevée ou autres problèmes de performances comme l'écriture d'un fichier journal à un système de fichiers réseau.

Personnalisé

Les gestionnaires d'informations personnalisés vous permettent de configurer de nouveaux types de gestionnaires de journaux mis en place. Un gestionnaire personnalisé doit être implémenté comme classe Java qui s'étend `java.util.logging.Handler` et doit être contenu dans un module.

syslog

Les gestionnaires de syslog peuvent être utilisés pour envoyer des messages à un serveur de journalisation à distance. Cela permet à plusieurs applications d'envoyer leurs messages de journalisation au même serveur, où ils peuvent être analysés en même temps.

[Rapporter un bogue](#)

12.1.15. Log Formatters

Un formateur de journalisation (log formatter) est une propriété de configuration d'un gestionnaire de journalisation qui détermine l'apparence des messages de journalisation. Il s'agit d'un string qui utilise une syntaxe basée sur la classe `java.util.Formatter`.

Ainsi, le string du formateur de journalisation de la configuration par défaut, `%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n`, crée des messages de journalisation qui ressemblent à ceci :

```
15:53:26,546 INFO [org.jboss.as] (Controller Boot Thread) JBAS015951:
Admin console listening on http://127.0.0.1:9990
```

[Rapporter un bogue](#)

12.1.16. Syntaxe de Formateur de journaux

Tableau 12.5. Syntaxe de Formateur de journaux

Symbol	Description
<code>%C</code>	La catégorie de l'événement de journalisation
<code>%p</code>	Le niveau de saisie de la journalisation (info/débogage/etc)
<code>%P</code>	Le niveau localisé de la saisie de journalisation

Symbol e	Description
%d	Les date/heure (yyyy-MM-dd HH:mm:ss,SSS form)
%r	L'heure relative (en millisecondes depuis l'initialisation de la journalisation)
%Z	Le réseau horaire
%k	Une clé de ressource de journalisation (utilisée pour la localisation de messages de journalisation)
%m	Le message de journalisation (avec trace d'exception)
%S	Le simple message de journalisation (sans trace d'exception)
%e	Suivi de la pile d'exceptions (sans informations sur les modules étendus)
%E	Suivi de la pile d'exceptions (avec informations sur les modules étendus)
%t	Le nom du thread en cours
%n	Un caractère de nouvelle ligne
%C	La classe du code appelant la méthode de journalisation (lente)
%F	Le nom de fichier de la classe appelant la méthode de journalisation (lente)
%l	L'emplacement d'origine du code appelant la méthode de journalisation (lente)
%L	Le numéro de ligne du code appelant la méthode de journalisation (lente)
%M	La méthode du code appelant la méthode de journalisation (lente)
%x	Contexte de diagnostic intégré
%X	Contexte de diagnostic du message
%%	Un pourcentage (caractère d'échappement)

[Rapporter un bogue](#)

12.2. CONFIGURER LA JOURNALISATION PAR LA CONSOLE DE GESTION

La console de gestion fournit une interface graphique utilisateur pour la configuration du root logger, des log handlers et des catégories de journalisation. Voir [Section 3.4.1, « Console de management »](#) pour obtenir plus d'informations sur la console de gestion.

Vous pourrez accéder à cette configuration en suivant les étapes suivantes :

1. Se connecter à la console de gestion
2. Naviguer dans la configuration du sous-système de logging. Cette étape varie suivant qu'il s'agisse de serveurs qui s'exécutent sur les serveurs autonomes ou des serveurs exécutant dans un domaine géré.
 - o **Serveur autonome**
Cliquer sur **Configuration**, étendre **Core** dans le menu **Subsystems**, et cliquer sur **Logging**.
 - o **Domaine géré**
Cliquer sur **Configuration**, sélectionner le profil que vous souhaitez modifier dans le menu déroulant. Étendre **Core** dans le menu **Subsystems**, et cliquer sur **Logging**.

Les tâches à effectuer pour configurer le root logger sont les suivantes :

- Modifier le niveau de journalisation.
- Ajouter et supprimer des log handlers.

Les tâches à effectuer pour configurer les catégories de journalisation sont les suivantes :

- Ajouter et supprimer les catégories de journalisation.
- Modifier les propriétés de catégories.
- Ajouter et supprimer les log handlers d'une catégorie.

Les tâches principales à effectuer pour configurer les log handlers sont les suivantes :

- Ajouter de nouveaux handlers.
- Configurer les handlers.

Tous les log handlers pris en charge (y compris le handler personnalisé) peuvent être configurés dans la console de gestion.

[Rapporter un bogue](#)

12.3. CONFIGURATION DE LOGGING DANS LE CLI

Conditions préalables

L'interface CLI doit exécuter et est connectée à l'instance JBoss EAP qui convient. Pour plus d'informations, voir [Section 3.5.2, « Lancement de l'interface CLI »](#)

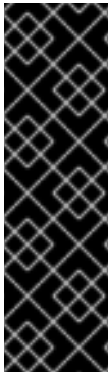
[Rapporter un bogue](#)

12.3.1. Configurer le root logger par le CLI

La configuration du Root Logger peut s'afficher ou être modifiée par l'interface CLI.

Les tâches principales à effectuer pour configurer le root logger sont les suivantes :

- Ajouter des log handler au root logger.
- Afficher la configuration du root logger.
- Modifier le niveau de journalisation.
- Supprimer des log handler du root logger.



IMPORTANT

Pour la configuration du Root Logger dans un profil de journalisation de système autonome, la racine (root) du chemin de configuration est `/subsystem=logging/logging-profile=NAME/` au lieu de `/subsystem=logging/`.

Dans un domaine géré, vous devez spécifier quel profil utiliser. Vous devez ajouter le nom de profil en début de chemin de configuration pour un domaine géré, et remplacer `/subsystem=logging/` par `/profile=NAME/subsystem=logging/`.

Ajouter un log handler au root logger

Utiliser l'opération `add-handler` avec la syntaxe suivante *HANDLER* dans le nom du gestionnaire de journalisation (Log Handler) à ajouter.

```
/subsystem=logging/root-logger=ROOT:add-handler(name="HANDLER")
```

Le log handler doit déjà avoir été créé avant de l'ajouter au root handler.

Exemple 12.3. Opération root logger add-handler

```
[standalone@localhost:9999 /] /subsystem=logging/root-logger=ROOT:add-handler(name="FILE")
{"outcome" => "success"}
```

Afficher le contenu de la configuration du root logger

Utiliser l'opération `read-resource` avec la syntaxe suivante.

```
/subsystem=logging/root-logger=ROOT:read-resource
```

Exemple 12.4. Opération root logger «read-resource»

```
[standalone@localhost:9999 /] /subsystem=logging/root-logger=ROOT:read-resource
{
  "outcome" => "success",
  "result" => {
    "filter" => undefined,
    "filter-spec" => undefined,
    "handlers" => [
      "CONSOLE",
```

```

        "FILE"
    ],
    "level" => "INFO"
}

```

Définir le niveau de journalisation du root logger

Utiliser l'opération `write-attribute` avec la syntaxe suivante avec `LEVEL` indiquant les niveaux de journalisation pris en charge.

```

/subsystem=logging/root-logger=ROOT:write-attribute(name="level",
value="LEVEL")

```

Exemple 12.5. L'opération «write-attribute» du root logger pour définir le niveau de journalisation

```

[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:write-attribute(name="level", value="DEBUG")
{"outcome" => "success"}

```

Supprimer un log handler du root logger

Utiliser `remove-handler` avec la syntaxe suivante, avec `HANDLER` comme nom du gestionnaire de journalisation à supprimer.

```

/subsystem=logging/root-logger=ROOT:remove-handler(name="HANDLER")

```

Exemple 12.6. Supprimer un log handler

```

[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:remove-handler(name="FILE")
{"outcome" => "success"}

```

Rapporter un bogue

12.3.2. Configurer une Catégorie dans l'interface CLI

Les catégories de journalisation peuvent être ajoutées, supprimées et modifiées dans le CLI.

Les tâches principales à effectuer pour configurer les catégories de journalisation sont les suivantes :

- Ajouter une nouvelle catégorie de journalisation :
- Afficher la configuration d'une catégorie de journalisation.
- Définir un niveau de journalisation.
- Ajouter des Log Handlers à une catégorie de journalisation.

- Supprimer des Log Handlers d'une catégorie de journalisation.
- Supprimer une catégorie de journalisation.



IMPORTANT

Pour la configuration d'une catégorie de journal dans un profil de journalisation de système autonome, la racine (root) du chemin de configuration est `/subsystem=logging/logging-profile=NAME/` au lieu de `/subsystem=logging/`.

Dans un domaine géré, vous devez spécifier quel profil utiliser. Vous devez ajouter le nom de profil en début de chemin de configuration pour un domaine géré, et remplacer `/subsystem=logging/` par `/profile=NAME/subsystem=logging/`.

Ajouter une catégorie de journalisation

Utiliser l'opération `add` avec la syntaxe suivante. Remplacer `CATEGORY` par la catégorie à ajouter.

```
/subsystem=logging/logger=CATEGORY:add
```

Exemple 12.7. Ajouter une nouvelle catégorie de journalisation

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:add
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Afficher une configuration de catégorie de journalisation

Utiliser l'opération `read-resource` avec la syntaxe suivante. Remplacer `CATEGORY` par le nom de la catégorie.

```
/subsystem=logging/logger=CATEGORY:read-resource
```

Exemple 12.8. Opération de lecture de ressource de la catégorie de journalisation

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=org.apache.tomcat.util.modeler:read-
resource
{
  "outcome" => "success",
  "result" => {
    "category" => "org.apache.tomcat.util.modeler",
    "filter" => undefined,
    "filter-spec" => undefined,
    "handlers" => undefined,
    "level" => "WARN",
    "use-parent-handlers" => true
  }
}
[standalone@localhost:9999 /]
```

Définir le niveau de journalisation

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `CATEGORY` par le nom de la catégorie de journalisation et `LEVEL` par le niveau de journalisation à définir.

```
/subsystem=logging/logger=CATEGORY:write-attribute(name="level",
value="LEVEL")
```

Exemple 12.9. Définir un niveau de journalisation

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:write-
attribute(name="level", value="DEBUG")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir la Catégorie de journalisation pour utiliser le Log Handler du Root Logger.

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `CATEGORY` par le nom de la catégorie de journalisation. Remplacer `BOOLEAN` par `true` pour cette catégorie de journalisation pour utiliser les handlers du Root Logger. Le remplacer par `false` s'il doit utiliser ses propres handlers.

```
/subsystem=logging/logger=CATEGORY:write-attribute(name="use-parent-
handlers", value="BOOLEAN")
```

Exemple 12.10. Configurer use-parent-handlers

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:write-
attribute(name="use-parent-handlers", value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Ajouter un Log Handler à une catégorie de journalisation.

Utiliser l'opération `add-handler` avec la syntaxe suivante. Remplacer `CATEGORY` par la catégorie à ajouter et `HANDLER` par le nom du handler à ajouter.

```
/subsystem=logging/logger=CATEGORY:add-handler(name="HANDLER")
```

Le Log Handler doit déjà avoir été créé avant de l'ajouter au Root Handler.

Exemple 12.11. Ajouter un Log Handler

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:add-
handler(name="AccountsNFSAsync")
{"outcome" => "success"}
```

Supprimer un Log Handler d'une catégorie de journalisation

Utiliser l'opération `remove-handler` avec la syntaxe suivante. Remplacer `CATEGORY` par le nom de la catégorie à `HANDLER` par le nom du log handler à supprimer.

```
/subsystem=logging/logger=CATEGORY:remove-handler(name="HANDLER")
```

Exemple 12.12. Supprimer un Log Handler

```
[standalone@localhost:9999 /] /subsystem=logging/logger=jacorb:remove-  
handler(name="AccountsNFSAsync")  
{"outcome" => "success"}  
[standalone@localhost:9999 /]
```

Supprimer une catégorie

Utiliser l'opération `remove` avec la syntaxe suivante. Remplacer `CATEGORY` par le nom de la catégorie à supprimer.

```
/subsystem=logging/logger=CATEGORY:remove
```

Exemple 12.13. Supprimer une catégorie de journalisation

```
[standalone@localhost:9999 /]  
/subsystem=logging/logger=com.company.accounts.rec:remove  
{"outcome" => "success"}  
[standalone@localhost:9999 /]
```

Rapporter un bogue

12.3.3. Configurer un log handler de console dans le CLI

Les log handlers de console peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un log handler de console :

- Ajouter un nouveau log handler de console.
- Afficher la configuration d'un log handler de console.
- Définir le niveau de journalisation du handler.
- Définir la cible de la sortie du handler.
- Définir la codification utilisée pour la sortie du handler.
- Définir le formateur utilisé pour la sortie du handler.
- Définir si le handler utilise autoflush ou non.

- Supprimer un handler de journalisation de console.



IMPORTANT

Pour la configuration d'un gestionnaire de journal dans un profil de journalisation de système autonome, la racine (root) du chemin de configuration est `/subsystem=logging/logging-profile=NAME/` au lieu de `/subsystem=logging/`.

Dans un domaine géré, vous devez spécifier quel profil utiliser. Vous devez ajouter le nom de profil en début de chemin de configuration pour un domaine géré, et remplacer `/subsystem=logging/` par `/profile=NAME/subsystem=logging/`.

Ajouter un log handler de console

Utiliser l'opération `add` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de journalisation de la console à ajouter.

```
/subsystem=logging/console-handler=HANDLER:add
```

Exemple 12.14. Ajouter un log handler de console

```
[standalone@localhost:9999 /] /subsystem=logging/console-  
handler=ERRORCONSOLE:add  
{"outcome" => "success"}
```

Afficher une configuration de log handler de journalisation de la console

Utiliser l'opération `read-resource` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de la console.

```
/subsystem=logging/console-handler=HANDLER:read-resource
```

Exemple 12.15. Afficher une configuration de log handler de journalisation de la console

```
[standalone@localhost:9999 /] /subsystem=logging/console-  
handler=CONSOLE:read-resource  
{  
  "outcome" => "success",  
  "result" => {  
    "autoflush" => true,  
    "enabled" => true,  
    "encoding" => undefined,  
    "filter" => undefined,  
    "filter-spec" => undefined,  
    "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",  
    "level" => "INFO",  
    "name" => "CONSOLE",  
    "named-formatter" => "COLOR-PATTERN",  
    "target" => "System.out"  
  }  
}
```


Définir le niveau de journalisation

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de la console et `LEVEL` par le niveau de journalisation à définir.

```
/subsystem=logging/console-handler=HANDLER:write-attribute(name="level",
value="INFO")
```

Exemple 12.16. Définir le niveau de journalisation

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="level", value="TRACE")
{"outcome" => "success"}
```

Définir la cible

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de la console et `TARGET` par `System.err` ou `System.out` pour le flux Erreurs système ou le flux Standard out.

```
/subsystem=logging/console-handler=HANDLER:write-
attribute(name="target", value="TARGET")
```

Exemple 12.17. Définir la cible

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="target",
value="System.err")
{"outcome" => "success"}
```

Définir le codage

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de la console et `ENCODING` par le nom de codification du caractère qui convient.

```
/subsystem=logging/console-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

Exemple 12.18. Définir le codage

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="encoding", value="utf-8")
{"outcome" => "success"}
```

Définir le formateur

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de la console et `FORMAT` par le string de formateur requis.

```
/subsystem=logging/console-handler=HANDLER:write-attribute(name="formatter", value="FORMAT")
```

Exemple 12.19. Définir le formateur

```
[standalone@localhost:9999 /] /subsystem=logging/console-handler=ERRORCONSOLE:write-attribute(name="formatter", value="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n") {"outcome" => "success"}
```

Définir auto flush

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de la console et `BOOLEAN` par `true` si le handler doit écrire sa sortie immédiatement.

```
/subsystem=logging/console-handler=HANDLER:write-attribute(name="autoflush", value="BOOLEAN")
```

Exemple 12.20. Définir auto flush

```
[standalone@localhost:9999 /] /subsystem=logging/console-handler=ERRORCONSOLE:write-attribute(name="autoflush", value="true") {"outcome" => "success"}
```

Supprimer un log handler de console

Utiliser l'opération `remove` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de la console à supprimer.

```
/subsystem=logging/console-handler=HANDLER:remove
```

Exemple 12.21. Supprimer un log handler de console

```
[standalone@localhost:9999 /] /subsystem=logging/console-handler=ERRORCONSOLE:remove {"outcome" => "success"}
```

[Rapporter un bogue](#)

12.3.4. Configurer un log handler de fichiers dans le CLI

Les log handlers de fichiers peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un log handler de fichiers :

- Ajouter un nouveau log handler de fichiers.

- Afficher la configuration d'un log handler de fichiers
- Définir le niveau de journalisation du handler.
- Définir le comportement d'ajout du handler.
- Définir si le handler utilise autoflush ou non.
- Définir la codification utilisée pour la sortie du handler.
- Indiquer le fichier dans lequel le log handler écrit.
- Définir le formateur utilisé pour la sortie du handler.
- Supprimer un log handler de fichiers.



IMPORTANT

Pour la configuration d'un gestionnaire de journal dans un profil de journalisation de système autonome, la racine (root) du chemin de configuration est `/subsystem=logging/logging-profile=NAME/` au lieu de `/subsystem=logging/`.

Dans un domaine géré, vous devez spécifier quel profil utiliser. Vous devez ajouter le nom de profil en début de chemin de configuration pour un domaine géré, et remplacer `/subsystem=logging/` par `/profile=NAME/subsystem=logging/`.

Ajouter un log handler de fichiers

Utiliser l'opération `add` avec la syntaxe suivante. Remplacer `PATH` par le nom du fichier dans lequel le log est écrit. Remplacer `DIR` par le nom du répertoire dans lequel le fichier se trouve. La valeur `DIR` peut correspondre à une variable de chemin.

```
/subsystem=logging/file-handler=HANDLER:add(file={"path"=>"PATH",
"relative-to"=>"DIR"})
```

Exemple 12.22. Ajouter un log handler de fichiers

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:add(file={"path"=>"accounts.log", "relative-
to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Afficher une configuration de log handler de fichiers

Utiliser l'opération `read-resource` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de fichiers.

```
/subsystem=logging/file-handler=HANDLER:read-resource
```

Exemple 12.23. Utiliser l'opération read-resource

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:read-resource
{
  "outcome" => "success",
  "result" => {
    "append" => true,
    "autoflush" => true,
    "enabled" => true,
    "encoding" => undefined,
    "file" => {
      "path" => "accounts.log",
      "relative-to" => "jboss.server.log.dir"
    },
    "filter" => undefined,
    "filter-spec" => undefined,
    "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
    "level" => undefined,
    "name" => "accounts_log",
    "named-formatter" => undefined
  }
}
```

Définir le niveau de journalisation

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de fichiers et `LOG_LEVEL_VALUE` par le niveau de journalisation à définir.

```
/subsystem=logging/file-handler=HANDLER:write-attribute(name="level",
value="LOG_LEVEL_VALUE")
```

Exemple 12.24. Changer le niveau de journalisation

```
/subsystem=logging/file-handler=accounts_log:write-
attribute(name="level", value="DEBUG")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le comportement d'ajout

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de fichiers. Remplacer `BOULÉEN` par `false` si vous souhaitez qu'un nouveau fichier de journalisation soit créé à chaque fois qu'un serveur d'applications est lancé. Remplacer `BOULÉEN` par `true` si le serveur d'applications doit continuer à utiliser le même fichier.

```
/subsystem=logging/file-handler=HANDLER:write-attribute(name="append",
value="BOULÉEN")
```

Exemple 12.25. Changer la propriété d'ajout

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="append", value="true")
```

```
{
  "outcome" => "success",
  "response-headers" => {
    "operation-requires-reload" => true,
    "process-state" => "reload-required"
  }
}
[standalone@localhost:9999 /]
```

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Définir auto flush

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de fichiers et *BOOLEAN* par `true` si le handler doit écrire sa sortie immédiatement.

```
/subsystem=logging/file-handler=HANDLER:write-attribute(name="autoflush", value="BOOLEAN")
```

Exemple 12.26. Changer la propriété autoflush

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="autoflush", value="false")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le codage

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de fichiers et *ENCODING* par le nom de codification du caractère qui convient.

```
/subsystem=logging/file-handler=HANDLER:write-attribute(name="encoding", value="ENCODING")
```

Exemple 12.27. Définir le codage

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="encoding", value="utf-8")
{"outcome" => "success"}
```

Changer le fichier dans lequel le log handler écrit.

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer *PATH* par le nom du fichier du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

```
/subsystem=logging/file-handler=HANDLER:write-attribute(name="file",
value={"path"=>"PATH", "relative-to"=>"DIR"})
```

Exemple 12.28. Changer le fichier dans lequel le log handler écrit.

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="file", value=
{"path"=>"accounts-debug.log", "relative-
to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le formateur

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de fichiers et `FORMAT` par le string de formateur requis.

```
/subsystem=logging/file-handler=HANDLER:write-
attribute(name="formatter", value="FORMAT")
```

Exemple 12.29. Définir le formateur

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="formatter",
value="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Supprimer un log handler de fichiers.

Utiliser l'opération `remove` avec la syntaxe suivante. Remplacer `HANDLER` par le nom du log handler de fichiers à supprimer.

```
/subsystem=logging/file-handler=HANDLER:remove
```

Exemple 12.30. Supprimer un log handler de fichiers.

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Un log handler ne peut être supprimé que s'il ne peut pas être référencé par une catégorie de journalisation ou par un log handler async.

[Rapporter un bogue](#)

12.3.5. Configurer un log handler périodique dans le CLI

Les log handlers périodiques peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un log handler périodique :

- Ajouter un nouveau log handler périodique.

- Afficher la configuration d'un log handler périodique
- Définir le niveau de journalisation du handler.
- Définir le comportement d'ajout du handler.
- Définir si le gestionnaire utilise `autoFlush` ou non.
- Définir la codification utilisée pour la sortie du handler.
- Indiquer le fichier dans lequel le log handler écrit.
- Définir le formateur utilisé pour la sortie du handler.
- Définir le suffixe pour les journaux en rotation.
- Supprimer un log handler périodique

Chacune de ces actions sont décrites ci-dessous.



IMPORTANT

Pour la configuration d'un gestionnaire de journal dans un profil de journalisation de système autonome, la racine (root) du chemin de configuration est `/subsystem=logging/logging-profile=NAME/` au lieu de `/subsystem=logging/`.

Dans un domaine géré, vous devez spécifier quel profil utiliser. Vous devez ajouter le nom de profil en début de chemin de configuration pour un domaine géré, et remplacer `/subsystem=logging/` par `/profile=NAME/subsystem=logging/`.

Ajouter un nouveau log handler périodique en rotation.

Utiliser l'opération `add` avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:add(file={
"path"=>"PATH", "relative-to"=>"DIR"}, suffix="SUFFIX")
```

Remplacer *HANDLER* par le nom du log handler. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin. Remplacer *SUFFIX* par le suffixe de rotation de fichiers à utiliser.

Exemple 12.31. Créer un nouvel handler

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:add(file={"path"=>"daily-debug.log",
"relative-to"=>"jboss.server.log.dir"}, suffix=".yyyy.MM.dd")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Afficher une configuration de log handler de fichiers en rotation périodique

Utiliser l'opération `read-resource` avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:read-resource
```

Remplacer *HANDLER* par le nom du log handler.

Exemple 12.32. Utiliser l'opération read-resource

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:read-resource
{
  "outcome" => "success",
  "result" => {
    "append" => true,
    "autoflush" => true,
    "enabled" => true,
    "encoding" => undefined,
    "file" => {
      "path" => "daily-debug.log",
      "relative-to" => "jboss.server.log.dir"
    },
    "filter" => undefined,
    "filter-spec" => undefined,
    "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
    "level" => undefined,
    "name" => "HOURLY_DEBUG",
    "named-formatter" => undefined,
    "suffix" => ".yyyy.MM.dd"
  }
}
```

Définir le niveau de journalisation

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="level". value="LOG_LEVEL_VALUE")
```

Remplacer *HANDLER* par le nom du log handler périodique et *LOG_LEVEL_VALUE* par le niveau de journalisation à définir.

Exemple 12.33. Définir le niveau de journalisation

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="level",
value="DEBUG")
{"outcome" => "success"}
```

Définir le comportement d'ajout

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-handler=HANDLER:write-
attribute(name="append", value="BOOLEAN")
```


Remplacer *HANDLER* par le nom du log handler périodique. Remplacer *BOULÉEN* par `false` si vous souhaitez qu'un nouveau fichier de journalisation soit créé à chaque fois qu'un serveur d'applications est lancé. Remplacer *BOULÉEN* par `true` si le serveur d'applications doit continuer à utiliser le même fichier.

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Exemple 12.34. Définir le comportement d'ajout

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="append", value="true")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
```

Définir auto flush

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="autoflush", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du log handler périodique et *BOOLEAN* par `true` si le handler doit écrire sa sortie immédiatement.

Exemple 12.35. Définir de comportement Auto Flush

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="autoflush",
value="false")
{
    "outcome" => "success",
    "response-headers" => {"process-state" => "reload-required"}
}
```

Définir le codage

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

Remplacer *HANDLER* par le nom du log handler périodique et *ENCODING* par le nom de codification du caractère qui convient.

Exemple 12.36. Définir le codage

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="encoding", value="utf-
8")
{"outcome" => "success"}
```

Changer le fichier dans lequel le log handler écrit.

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="file", value={"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer *HANDLER* par le nom du log handler périodique. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

Exemple 12.37. Changer le fichier dans lequel le log handler écrit.

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="file", value=
{"path"=>"daily-debug.log", "relative-to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
```

Définir le formateur

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="formatter", value="FORMAT")
```

Remplacer *HANDLER* par le nom du log handler périodique et *FORMAT* par le string de formateur à définir.

Exemple 12.38. Définir le formateur

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="formatter",
value="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le suffixe pour les journaux en rotation

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="suffix", value="SUFFIX")
```

Remplacer *HANDLER* par le nom du log handler et *SUFFIX* par le string de suffixe à définir.

Exemple 12.39. Définir le suffixe de journal en rotation.

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="suffix", value=".yyyy-
MM-dd-HH")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Supprimer un log handler périodique

Utiliser l'opération `remove` avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:remove
```

Remplacer *HANDLER* par le nom du log handler périodique.

Exemple 12.40. Supprimer un log handler périodique

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

[Rapporter un bogue](#)**12.3.6. Configurer un log handler de taille dans le CLI**

Les log handlers de taille de fichiers en rotation peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches qui vous reviendront pour configurer un log handler de taille de fichiers en rotations :

- Ajouter un nouveau log handler
- Afficher la configuration d'un log handler.
- Définir le niveau de journalisation du handler.
- Définir le comportement d'ajout du handler.
- Définir si le handler utilise autoflush ou non.
- Définir la codification utilisée pour la sortie du handler.
- Indiquer le fichier dans lequel le log handler écrit.
- Définir le formateur utilisé pour la sortie du handler.
- Définir la taille maximum de chaque fichier journal.
- Définir le nombre maximum de journaux de sauvegarde à conserver.

- Définit l'option de démarrage de la rotation à l'amorçage pour le log handler de taille de fichiers.
- Définir le suffixe pour les journaux en rotation.
- Supprimer un log handler.

Chacune de ces tâches sont décrites ci-dessous.



IMPORTANT

Pour la configuration d'un gestionnaire de journal dans un profil de journalisation de système autonome, la racine (root) du chemin de configuration est `/subsystem=logging/logging-profile=NAME/` au lieu de `/subsystem=logging/`.

Dans un domaine géré, vous devez spécifier quel profil utiliser. Vous devez ajouter le nom de profil en début de chemin de configuration pour un domaine géré, et remplacer `/subsystem=logging/` par `/profile=NAME/subsystem=logging/`.

Ajouter un nouveau log handler

Utiliser l'opération `add` avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:add(file=
{"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer `HANDLER` par le nom du log handler. Remplacer `PATH` par le nom du fichier dans lequel le log est écrit. Remplacer `DIR` par le nom du répertoire dans lequel le fichier se trouve. La valeur `DIR` peut correspondre à une variable de chemin.

Exemple 12.41. Ajouter un nouveau log handler

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:add(file={"path"=>"accounts_trace.log",
"relative-to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
```

Afficher la configuration du log handler

Utiliser l'opération `read-resource` avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:read-resource
```

Remplacer `HANDLER` par le nom du log handler.

Exemple 12.42. Afficher la configuration du log handler

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:read-resource
{
  "outcome" => "success",
  "result" => {
```

```

        "append" => true,
        "enabled" => true,
        "autoflush" => true,
        "encoding" => undefined,
        "file" => {
            "path" => "accounts_trace.log",
            "relative-to" => "jboss.server.log.dir"
        },
        "filter" => undefined,
        "filter-spec" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => "ALL",
        "name" => "ACCOUNTS_TRACE"
        "named-formatter" => undefined,
        "max-backup-index" => 1,
        "rotate-size" => "2m"
        "rotate-on-boot" => false
    }
}
[standalone@localhost:9999 /]

```

Définir le niveau de journalisation du handler

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```

/subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="level", value="LOG_LEVEL_VALUE")

```

Remplacer *HANDLER* par le nom du log handler et *LOG_LEVEL_VALUE* par le niveau de journalisation à définir.

Exemple 12.43. Définir le niveau de journalisation du handler

```

[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="level", value="TRACE")
{"outcome" => "success"}
[standalone@localhost:9999 /]

```

Définir le comportement d'ajout du handler.

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```

/subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="append", value="BOOLEAN")

```

Remplacer *HANDLER* par le nom du log handler. Remplacer *BOULÉEN* par `false` si vous souhaitez qu'un nouveau fichier de journalisation soit créé à chaque fois qu'un serveur d'applications est lancé. Remplacer *BOULÉEN* par `true` si le serveur d'applications doit continuer à utiliser le même fichier.

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Exemple 12.44. Définir le comportement d'ajout du handler.

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="append", value="true")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
[standalone@localhost:9999 /]
```

Définir si le handler utilise autoflush ou non

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="autoflush", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du log handler et *BOOLEAN* par `true` si le handler doit écrire sa sortie immédiatement.

Exemple 12.45. Définir si le handler utilise autoflush ou non

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="autoflush", value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir la codification utilisée pour la sortie du handler

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

Remplacer *HANDLER* par le nom du log handler et *ENCODING* par le nom de codification du caractère qui convient.

Exemple 12.46. Définir la codification utilisée pour la sortie du handler

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="encoding", value="utf-8")
{"outcome" => "success"}
```

Indiquer le fichier dans lequel le log handler écrit

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

■

```
/subsystem=logging/size-rotating-file-handler=HANDLER:write-attribute(name="file", value={"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer *HANDLER* par le nom du log handler. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

Exemple 12.47. Indiquer le fichier dans lequel le log handler écrit

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-handler=ACCOUNTS_TRACE:write-attribute(name="file", value={"path"=>"accounts_trace.log", "relative-to"=>"jboss.server.log.dir"}) {"outcome" => "success"}
```

Définir le formateur utilisé pour la sortie du handler.

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:write-attribute(name="formatter", value="FORMATTER")
```

Remplacer *HANDLER* par le nom du log handler et *FORMAT* par le string de formateur à définir.

Exemple 12.48. Définir le formateur utilisé pour la sortie du handler.

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-handler=ACCOUNTS_TRACE:write-attribute(name="formatter", value="%d{HH:mm:ss,SSS} %-5p (%c) [%t] %s%E%n") {"outcome" => "success"}
```

Définir la taille maximum de chaque fichier de journalisation

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:write-attribute(name="rotate-size", value="SIZE")
```

Remplacer *HANDLER* par le nom du log handler et *SIZE* par la taille de fichier maximum.

Exemple 12.49. Définir la taille maximum de chaque fichier de journalisation

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-handler=ACCOUNTS_TRACE:write-attribute(name="rotate-size", value="50m") {"outcome" => "success"} [standalone@localhost:9999 /]
```

Définir le nombre maximum de journaux de sauvegarde à conserver

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:write-attribute(name="max-backup-index", value="NUMBER")
```

Remplacer *HANDLER* par le nom du log handler et *NUMBER* par le nombre de fichiers de journalisation à conserver.

Exemple 12.50. Définir le nombre maximum de journaux de sauvegarde à conserver

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-handler=ACCOUNTS_TRACE:write-attribute(name="max-backup-index", value="5")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définit l'option de rotation au démarrage du `size-rotating-file-handler`

Cette option n'est disponible que pour le gestionnaire de fichiers `size-rotating-file-handler`. Une valeur par défaut de `false` indique qu'un nouveau fichier de journalisation n'est pas créé au redémarrage du serveur.

Pour changer cela, utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:write-attribute(name="rotate-on-boot", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du log handler `size-rotating-file-handler`. Remplacer *BOOLEAN* par `true` si un nouveau fichier de journalisation `size-rotating-file-handler` doit être créé au redémarrage.

Exemple 12.51. Indique qu'il faut créer un nouveau fichier de journalisation `size-rotating-file-handler` lors du redémarrage du serveur.

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-handler=ACCOUNTS_TRACE:write-attribute(name="rotate-on-boot", value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le suffixe pour les journaux en rotation

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:write-attribute(name="suffix", value="SUFFIX")
```

Remplacer *HANDLER* par le nom du log handler et *SUFFIX* par le string de suffixe à définir.

Exemple 12.52.

■


```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="suffix", value=".yyyy-MM-
dd-HH")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Supprimer un log handler

Utiliser l'opération **remove** avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:remove
```

Remplacer *HANDLER* par le nom du log handler.

Exemple 12.53. Supprimer un log handler

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:remove
{"outcome" => "success"}
```

Rapporter un bogue

12.3.7. Configurer un gestionnaire de journal de taille de fichiers périodique en rotation dans l'interface CLI

Les gestionnaire de journaux de taille de fichiers périodiques en rotation peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un gestionnaire de journal de taille de fichiers périodique en rotation :

- Ajouter un nouveau gestionnaire de journal de taille de fichiers périodique en rotation.
- Afficher la configuration d'un gestionnaire de taille de fichiers périodique en rotation.
- Définir le niveau de journalisation du gestionnaire
- Définir le comportement d'ajout du gestionnaire.
- Définir si le gestionnaire utilise `autoFlush` ou non.
- Définir la codification utilisée pour la sortie du gestionnaire.
- Indiquer le fichier dans lequel le gestionnaire de journal écrit.
- Définir le formateur utilisé pour la sortie du gestionnaire.
- Définir la taille maximum de chaque fichier journal.
- Définir le nombre maximum de journaux de sauvegarde à conserver.

- Définit l'option de démarrage de la rotation à l'amorçage pour le gestionnaire de taille de fichiers périodique en rotation.
- Définir le suffixe pour les journaux en rotation.
- Supprimer un gestionnaire de journal de taille de fichiers périodique en rotation.

Chacune de ces actions sont décrites ci-dessous.



IMPORTANT

Pour la configuration d'un gestionnaire de taille de fichiers périodique en rotation dans un profil de journalisation de système autonome, la racine (root) du chemin de configuration est `/subsystem=logging/logging-profile=NAME/` au lieu de `/subsystem=logging/`.

Dans un domaine géré, vous devez spécifier quel profil utiliser. Vous devez ajouter le nom de profil en début de chemin de configuration pour un domaine géré, et remplacer `/subsystem=logging/` par `/profile=NAME/subsystem=logging/`.

Ajouter un nouveau gestionnaire de taille de fichiers périodique en rotation.

Utiliser l'opération `add` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-
handler=HANDLER:add(file={"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer *gestionnaire* par le nom du gestionnaire de journal. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

Exemple 12.54. Ajouter un nouveau gestionnaire de journal

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-
rotating-file-handler=PERIODIC_SIZE:add(file=
{"path"=>"periodic_size.log", "relative-
to"=>"jboss.server.log.dir"}, suffix=".yyyy.MM.dd")
{"outcome" => "success"}
```

Afficher la configuration du gestionnaire de journal

Utiliser l'opération `read-resource` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:read-
resource
```

Remplacer *HANDLER* par le nom du gestionnaire de journal.

Exemple 12.55. Afficher la configuration du gestionnaire de journal

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-
rotating-file-handler=PERIODIC_SIZE:read-resource
{
```

```

"outcome" => "success",
"result" => {
  "append" => true,
  "autoflush" => true,
  "enabled" => true,
  "encoding" => undefined,
  "file" => {
    "relative-to" => "jboss.server.log.dir",
    "path" => "periodic_size.log"
  },
  "filter-spec" => undefined,
  "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
  "level" => "ALL",
  "max-backup-index" => 1,
  "name" => "PERIODIC_SIZE",
  "named-formatter" => undefined,
  "rotate-on-boot" => false,
  "rotate-size" => "2m",
  "suffix" => ".yyyy.MM.dd"
}
}
[standalone@localhost:9999 /]

```

Définir le niveau de journalisation du gestionnaire

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```

/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:write-
attribute(name="level", value="LOG_LEVEL_VALUE")

```

Remplacer *HANDLER* par le nom du gestionnaire de journal et *LOG_LEVEL_VALUE* par le niveau de journalisation à définir.

Exemple 12.56. Définir le niveau de journalisation du gestionnaire

```

[standalone@localhost:9999 /] /subsystem=logging/periodic-size-
rotating-file-handler=PERIODIC_SIZE:write-attribute(name="level",
value="TRACE")
{"outcome" => "success"}
[standalone@localhost:9999 /]

```

Définir le comportement d'ajout du gestionnaire.

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```

/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:write-
attribute(name="append", value="BOOLEAN")

```

Remplacer *HANDLER* par le nom du gestionnaire de journal. Remplacer *BOULÉEN* par `false` si vous souhaitez qu'un nouveau fichier journal soit créé à chaque fois qu'un serveur d'applications est lancé. Remplacer *BOULÉEN* par `true` si le serveur d'applications doit continuer à utiliser le même fichier.

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Exemple 12.57. Définir le comportement d'ajout du gestionnaire.

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-rotating-file-handler=PERIODIC_SIZE:write-attribute(name="append", value="true")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
[standalone@localhost:9999 /]
```

Définir si le gestionnaire utilise autoflush ou non

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:write-attribute(name="autoflush", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du gestionnaire de journal et *BOOLEAN* par `true` si le handler doit écrire sa sortie immédiatement.

Exemple 12.58. Définir si le gestionnaire utilise autoflush ou non

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-rotating-file-handler=PERIODIC_SIZE:write-attribute(name="autoflush", value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir la codification utilisée pour la sortie du gestionnaire

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:write-attribute(name="encoding", value="ENCODING")
```

Remplacer *HANDLER* par le nom du gestionnaire de journal et *ENCODING* par le nom de codification du caractère qui convient.

Exemple 12.59. Définir la codification utilisée pour la sortie du gestionnaire

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-rotating-file-handler=PERIODIC_SIZE:write-attribute(name="encoding", value="utf-8")
{"outcome" => "success"}
```

Indiquer le fichier dans lequel le gestionnaire de journal écrit

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:write-attribute(name="file", value={"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer *gestionnaire* par le nom du gestionnaire de journal. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

Exemple 12.60. Indiquer le fichier dans lequel le gestionnaire de journal écrit

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-rotating-file-handler=PERIODIC_SIZE:write-attribute(name="file", value={"path"=>"accounts_trace.log", "relative-to"=>"jboss.server.log.dir"}) {"outcome" => "success"}
```

Définir le formateur utilisé pour la sortie du gestionnaire.

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:write-attribute(name="formatter", value="FORMAT")
```

Remplacer *HANDLER* par le nom du gestionnaire de journal et *FORMAT* par le string de formateur à définir ou le nom du formateur spécifié dans le fichier de configuration.

Exemple 12.61. Définir le formateur utilisé pour la sortie du gestionnaire.

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-rotating-file-handler=PERIODIC_SIZE:write-attribute(name="formatter", value="%d{HH:mm:ss,SSS} %-5p (%c) [%t] %s%E%n") {"outcome" => "success"}
```

Définir la taille maximum de chaque fichier journal

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:write-attribute(name="rotate-size", value="SIZE")
```

Remplacer *HANDLER* par le nom du gestionnaire de journal et *SIZE* par la taille de fichier maximum.

Exemple 12.62. Définir la taille maximum de chaque fichier journal

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-rotating-file-handler=PERIODIC_SIZE:write-attribute(name="rotate-size", value="50m") {"outcome" => "success"} [standalone@localhost:9999 /]
```

Définir le nombre maximum de journaux de sauvegarde à conserver

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:write-attribute(name="max-backup-index", value="NUMBER")
```

Remplacer *HANDLER* par le nom du gestionnaire de journal et *NUMBER* par le nombre de fichiers de journalisation à conserver.

Exemple 12.63. Définir le nombre maximum de journaux de sauvegarde à conserver

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-rotating-file-handler=PERIODIC_SIZE:write-attribute(name="max-backup-index", value="5")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définit l'option de rotation au démarrage du `periodic-size-rotating-file-handler`

Une valeur par défaut de `false` indique qu'un nouveau fichier journal n'est pas créé au redémarrage du serveur.

Pour changer cela, utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:write-attribute(name="rotate-on-boot", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du gestionnaire de journal `periodic-size-rotating-file-handler`. Remplacer *BOOLEAN* par `true` si un nouveau fichier journal `periodic-size-rotating-file-handler` doit être créé au redémarrage.

Exemple 12.64. Indique qu'il faut créer un nouveau fichier journal `periodic-size-rotating-file-handler` lors du redémarrage du serveur.

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-rotating-file-handler=PERIODIC_SIZE:write-attribute(name="rotate-on-boot", value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le suffixe pour les journaux en rotation

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:write-attribute(name="suffix", value="SUFFIX")
```

Remplacer *HANDLER* par le nom du gestionnaire de journal et *SUFFIX* par le string de suffixe à définir.

Exemple 12.65.

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-rotating-file-handler=PERIODIC_SIZE:write-attribute(name="suffix",value=".yyyy-MM-dd-HH")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Supprimer un gestionnaire de journal

Utiliser l'opération `remove` avec la syntaxe suivante.

```
/subsystem=logging/periodic-size-rotating-file-handler=HANDLER:remove
```

Remplacer *HANDLER* par le nom du gestionnaire de journal.

Exemple 12.66. Supprimer un gestionnaire de journal

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-size-rotating-file-handler=PERIODIC_SIZE:remove
{"outcome" => "success"}
```

Rapporter un bogue

12.3.8. Configurer un log handler async dans le CLI

Les log handlers async peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Les tâches que vous allez effectuer pour configurer un log handler async sont les suivantes :

- Ajouter un nouveau log handler async
- Afficher la configuration d'un log handler async
- Modifier le niveau de journalisation
- Définir la longueur de la file d'attente
- Définir l'action de dépassement
- Ajouter les sous-handlers
- Supprimer les sous-handlers
- Supprimer un sous-handler async

Chacune de ces tâches sont décrites ci-dessous.



IMPORTANT

Pour la configuration d'un gestionnaire de journal dans un profil de journalisation de système autonome, la racine (root) du chemin de configuration est `/subsystem=logging/logging-profile=NAME/` au lieu de `/subsystem=logging/`.

Dans un domaine géré, vous devez spécifier quel profil utiliser. Vous devez ajouter le nom de profil en début de chemin de configuration pour un domaine géré, et remplacer `/subsystem=logging/` par `/profile=NAME/subsystem=logging/`.

Ajouter un nouveau log handler async

Utiliser l'opération `add` avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:add(queue-length="LENGTH")
```

Remplacer *HANDLER* par le nom du log handler et *LENGTH* par la valeur du nombre maximum de requêtes de journalisation pouvant tenir dans une file d'attente.

Exemple 12.67.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:add(queue-length="10")
{"outcome" => "success"}
```

Afficher la configuration d'un log handler async

Utiliser l'opération `read-resource` avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:read-resource
```

Remplacer *HANDLER* par le nom du log handler.

Exemple 12.68.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:read-resource
{
  "outcome" => "success",
  "result" => {
    "enabled" => true,
    "encoding" => undefined,
    "filter" => undefined,
    "filter-spec" => undefined,
    "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
    "level" => "ALL",
    "name" => "NFS_LOGS"
    "overflow-action" => "BLOCK",
    "queue-length" => "50",
    "subhandlers" => undefined
  }
}
```



```

    }
  }
[standalone@localhost:9999 /]

```

Modifier le niveau de journalisation

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```

/subsystem=logging/async-handler=HANDLER:write-attribute(name="level",
value="LOG_LEVEL_VALUE")

```

Remplacer *HANDLER* par le nom du log handler et *LOG_LEVEL_VALUE* par le niveau de journalisation à définir.

Exemple 12.69.

```

[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:write-attribute(name="level", value="INFO")
{"outcome" => "success"}
[standalone@localhost:9999 /]

```

Définir la longueur de la file d'attente

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```

/subsystem=logging/async-handler=HANDLER:write-attribute(name="queue-
length", value="LENGTH")

```

Remplacer *HANDLER* par le nom du log handler et *LENGTH* par la valeur du nombre maximum de requêtes de journalisation pouvant tenir dans une file d'attente.

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Exemple 12.70.

```

[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:write-attribute(name="queue-length", value="150")
{
  "outcome" => "success",
  "response-headers" => {
    "operation-requires-reload" => true,
    "process-state" => "reload-required"
  }
}

```

Définir l'action de dépassement

Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:write-attribute(name="overflow-action", value="ACTION")
```

Remplacer *HANDLER* par le nom du log handler et *ACTION* par *DISCARD* ou *BLOCK*.

Exemple 12.71.

```
[standalone@localhost:9999 /] /subsystem=logging/async-handler=NFS_LOGS:write-attribute(name="overflow-action", value="DISCARD") {"outcome" => "success"} [standalone@localhost:9999 /]
```

Ajouter les sous-handlers

Utiliser l'opération **add-handler** avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:add-handler(name="SUBHANDLER")
```

Remplacer *HANDLER* par le nom du log handler et *SUBHANDLER* par le nom du log handler qui doit être ajouté comme sous-handler de ce handler asynch.

Exemple 12.72.

```
[standalone@localhost:9999 /] /subsystem=logging/async-handler=NFS_LOGS:add-handler(name="NFS_FILE") {"outcome" => "success"} [standalone@localhost:9999 /]
```

Supprimer les sous-handlers

Utiliser l'opération **remove-handler** avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:remove-handler(name="SUBHANDLER")
```

Remplacer *HANDLER* par le nom du log handler et *SUBHANDLER* par le nom du log handler qui doit être supprimé.

Exemple 12.73.

```
[standalone@localhost:9999 /] /subsystem=logging/async-handler=NFS_LOGS:remove-handler(name="NFS_FILE") {"outcome" => "success"} [standalone@localhost:9999 /]
```

Supprimer un sous-handler asynch

Utiliser l'opération **remove** avec la syntaxe suivante.

-

```
/subsystem=logging/async-handler=HANDLER:remove
```

Remplacer *HANDLER* par le nom du log handler.

Exemple 12.74.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Rapporter un bogue

12.3.9. Configurer un gestionnaire personnalisé dans le CLI

Un gestionnaire personnalisé peut être ajouté, supprimé et modifié dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un gestionnaire personnalisé :

- Ajouter un nouveau gestionnaire personnalisé.
- Afficher la configuration d'un gestionnaire personnalisé.
- Définir un niveau de journalisation.
- Supprimer un gestionnaire personnalisé.

IMPORTANT

Pour la configuration d'un gestionnaire personnalisé dans un profil de journalisation de système autonome, la racine (root) du chemin de configuration est `/subsystem=logging/logging-profile=NAME/` au lieu de `/subsystem=logging/`.

Dans un domaine géré, vous devez spécifier quel profil utiliser. Vous devez ajouter le nom de profil en début de chemin de configuration pour un domaine géré, et remplacer `/subsystem=logging/` par `/profile=NAME/subsystem=logging/`.

Ajouter un nouveau gestionnaire personnalisé

Utiliser l'opération `add` avec la syntaxe suivante.

```
/subsystem=logging/custom-handler="MyCustomHandler":add
```

Exemple 12.75. Ajouter un nouveau gestionnaire personnalisé

```
[standalone@localhost:9999 /] /subsystem=logging/custom-
handler="MyCustomHandler":add(class="JdbcLogger",module="com.MyModule
",formatter="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",properties=
```

```
{ "maxNumberOfDays" => "90", "fileName" => "custom.log", "compressBackups" =>
  "true" })
[standalone@localhost:9999 /]
```

Afficher un gestionnaire personnalisé

Utiliser l'opération `read-resource` avec la syntaxe suivante. Remplacer `CUSTOMHANDLER` par le nom du gestionnaire personnalisé.

```
/subsystem=logging/custom-handler=CUSTOMHANDLER:read-resource
```

Exemple 12.76. Afficher une configuration de gestionnaire personnalisé

```
[standalone@localhost:9999 /] /subsystem=logging/custom-
handler="MyCustomHandler":read-resource
{
  "outcome" => "success",
  "result" => {
    "autoflush" => true,
    "enabled" => true,
    "encoding" => undefined,
    "filter" => undefined,
    "filter-spec" => undefined,
    "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
    "level" => "INFO",
    "name" => "CONSOLE",
    "named-formatter" => "COLOR-PATTERN",
    "target" => "System.out"
  }
}
```

Définir le niveau de journalisation

Utiliser l'opération `write-attribute` avec la syntaxe suivante. Remplacer `CUSTOMHANDLER` par le nom de la catégorie de journalisation et `LEVEL` par le niveau de journalisation à définir.

```
/subsystem=logging/custom-handler=CUSTOMHANDLER:write-
attribute(name="level", value="INFO")
```

Exemple 12.77. Définir le niveau de journalisation

```
[standalone@localhost:9999 /] /subsystem=logging/custom-
gestionnaire="MyCustomgestionnaire":write-attribute(name="level",
value="TRACE")
{"outcome" => "success"}
```

Supprimer un gestionnaire personnalisé

Utiliser l'opération `remove` avec la syntaxe suivante. Remplacer `CUSTOMgestionnaire` par le nom du gestionnaire personnalisé de fichiers à supprimer.

```
/subsystem=logging/custom-handler=CUSTOMHANDLER:remove
```

Exemple 12.78. Supprimer un gestionnaire personnalisé

```
[standalone@localhost:9999 /] /subsystem=logging/custom-
gestionnaire="MyCustomgestionnaire":remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

[Rapporter un bogue](#)

12.3.10. Configurer un gestionnaire Syslog dans le CLI

Le logmanager de JBoss EAP 6 contient désormais un gestionnaire syslog. Les gestionnaires syslog peuvent être utilisés pour envoyer des messages à un serveur de journalisation à distance qui supporte Syslog protocol (RFC-3164 or RFC-5424). Cela permet d'envoyer des messages de journalisation en provenance de plusieurs applications au même serveur, où ils peuvent être traités en même temps. Ce sujet traite de la création et de la configuration d'un gestionnaire syslog avec l'interface CLI, ainsi que des options de configuration disponibles.

Pour obtenir des informations sur les attributs du gestionnaire syslog, consulter [Section A.3, « Référence de Journalisation d'auditing de l'interface de gestion »](#).

Conditions préalables

- Accès et permissions correctes pour l'interface CLI.

Procédure 12.3. Ajouter un gestionnaire Syslog

- Exécuter la commande suivante pour ajouter un gestionnaire syslog :

```
/subsystem=logging/syslog-handler=HANDLER_NAME:add
```

Procédure 12.4. Configurer un gestionnaire Syslog

- Exécuter la commande suivante pour configurer un attribut de gestionnaire syslog :

```
/subsystem=logging/syslog-handler=HANDLER_NAME:write-
attribute(name=ATTRIBUTE_NAME,value=ATTRIBUTE_VALUE)
```

Procédure 12.5. Supprimer un gestionnaire Syslog

- Exécuter la commande suivante pour supprimer un gestionnaire syslog existant :

```
/subsystem=logging/syslog-handler=HANDLER_NAME:remove
```

[Rapporter un bogue](#)

12.3.11. Configurer un log handler personnalisé dans le CLI

Résumé

En plus de la syntaxe de formatage de journalisation spécifiée dans [Section 12.1.16, « Syntaxe de Formateur de journaux »](#), un module de formatage de journal personnalisé peut être créé pour une utilisation avec n'importe quel gestionnaire de journal. Cet exemple de procédure vous le montre en créant un formateur XML pour un gestionnaire de journal de console.

Conditions préalables

- Accéder au Management CLI dans le serveur JBoss EAP 6.
- Un log handler précédemment configuré. Cet exemple de procédure utilise un log handler de console.

Procédure 12.6. Configurer un formateur XML personnalisé de log handler

1. Créer un formateur personnalisé.

Dans cet exemple, la commande suivante crée un formateur personnalisé intitulé `XML_FORMATTER` qui utilise la classe `java.util.logging.XMLFormatter`.

```
[standalone@localhost:9999 /] /subsystem=logging/custom-formatter=XML_FORMATTER:add(class=java.util.logging.XMLFormatter, module=org.jboss.logmanager)
```

2. Enregistrer un formateur personnalisé pour le log handler que vous souhaitez utiliser avec.

Dans cet exemple, le formateur de l'étape suivante est ajouté au log handler de la console.

```
[standalone@localhost:9999 /] /subsystem=logging/console-handler=HANDLER:write-attribute(name=named-formatter, value=XML_FORMATTER)
```

3. Démarrer à nouveau le serveur JBoss EAP 6 pour que le changement puisse prendre effet :

```
[standalone@localhost:9999 /] shutdown --restart=true
```

Résultat

Le formateur XML personnalisé est ajouté au log handler de la console. La sortie du log de la console sera formaté en XML, comme suit :

```
<record>
  <date>2014-03-11T13:02:53</date>
  <millis>1394539373833</millis>
  <sequence>116</sequence>
  <logger>org.jboss.as</logger>
  <level>INFO</level>
  <class>org.jboss.as.server.BootstrapListener</class>
  <method>logAdminConsole</method>
  <thread>282</thread>
  <message>JBAS015951: Admin console listening on http://%s:%d</message>
  <param>127.0.0.1</param>
  <param>9990</param>
</record>
```

[Rapporter un bogue](#)

12.4. LA JOURNALISATION PAR DÉPLOIEMENT

12.4.1. La journalisation par déploiement

La journalisation par déploiement permet à un développeur de configurer à l'avance la configuration de journalisation de ses applications. Lorsque l'application est déployée, la journalisation commence selon la configuration définie. Les fichiers journaux créés par le biais de cette configuration contiennent uniquement des informations sur le comportement de l'application.

Cette approche a des avantages et des inconvénients pour l'utilisation de la journalisation dans l'ensemble du système. Un des avantages est que l'administrateur de l'instance JBoss EAP n'a pas besoin de configurer la journalisation. Un inconvénient est que la configuration de journalisation par déploiement est en lecture seule au démarrage et donc ne peut donc pas être modifiée pendant l'exécution.

[Rapporter un bogue](#)

12.4.2. Désactivation de la journalisation par déploiement

Procédure 12.7. Désactivation de la journalisation par déploiement

- Il existe deux méthodes pour désactiver la journalisation par déploiement. Il y en a une qui fonctionne pour toutes les versions de JBoss EAP 6, et une autre qui ne fonctionne que sur JBoss EAP 6.3 ou version supérieure.
 - **JBoss EAP 6 (toutes les versions)**
Ajouter la propriété système :


```
org.jboss.as.logging.per-deployment=false
```
 - **JBoss EAP 6.3 (ou version supérieure)**
Exclure le sous-système de journalisation via le fichier `jboss-deployment-structure.xml`. Pour obtenir des informations sur la façon de procéder, voir *Exclude a Subsystem from a Deployment* dans le *Development Guide*.

[Rapporter un bogue](#)

12.5. PROFILS DE JOURNALISATION

12.5.1. Profils de journalisation



IMPORTANT

Les profils de journalisation ne sont disponibles que dans les versions 6.1.0 ou versions supérieures. Ils ne peuvent pas être configurés par la console de gestion.

Les profils de journalisation sont des ensembles de configuration de journalisation indépendants qui peuvent être assignés à des applications déployées. Un profil de journalisation peut définir les handlers, les catégories et un root logger à la manière du sous-système de journalisation standard,

mais ne peut pas recommander la configuration à d'autres profils ou au sous-système de journalisation principal. La conception des profils de journalisation émule le sous-système de journalisation au niveau de l'aisance de configuration.

Les profils de journalisation permettent aux administrateurs de créer des configurations de journalisation spécifiques à une ou à plusieurs applications sans affecter une autre configuration de journalisation. Comme chaque profil est défini dans une configuration serveur, cela implique que la configuration de journalisation peut être modifiée sans exiger que les applications affectées ne soient redéployées.

Chaque profil de journalisation peut avoir la configuration suivante :

- Un nom unique. Ceci est requis.
- N'importe quel nombre de log handlers.
- N'importe quelle catégorie log.
- Un root logger maximum.

Une application peut spécifier un profil de journalisation à utiliser dans son fichier **MANIFEST.MF**, en utilisant l'attribut *logging-profile*.

[Rapporter un bogue](#)

12.5.2. Créer un nouveau profil de journalisation par le CLI

On peut créer un nouveau profil de journalisation par la commande CLI suivante, en remplaçant *NAME* par le nom de profil qui convient :

```
/subsystem=logging/logging-profile=NAME:add
```

Cela va créer un nouveau profil vide auquel les handlers, catégories et root logger peuvent être ajoutés.

[Rapporter un bogue](#)

12.5.3. Créer un profil de journalisation par le CLI

Un profil de journalisation peut être configuré par les log handlers, catégories et root logger en utilisant exactement la même syntaxe que lorsque l'on utilise le sous-système de journalisation principal.

Il existe uniquement deux différences entre configurer le sous-système de journalisation principal et le profil de journalisation :

1. Le chemin de configuration root est `/subsystem=logging/logging-profile=NAME`
2. Un profil de journalisation ne peut pas contenir d'autres profils de journalisation.

Référez vous à la tâche de gestion de journalisation qui convient :

- [Section 12.3.1, « Configurer le root logger par le CLI »](#)
- [Section 12.3.2, « Configurer une Catégorie dans l'interface CLI »](#)

- [Section 12.3.3, « Configurer un log handler de console dans le CLI »](#)
- [Section 12.3.4, « Configurer un log handler de fichiers dans le CLI »](#)
- [Section 12.3.5, « Configurer un log handler périodique dans le CLI »](#)
- [Section 12.3.6, « Configurer un log handler de taille dans le CLI »](#)
- [Section 12.3.8, « Configurer un log handler async dans le CLI »](#)

Exemple 12.79. Créer et configurer un profil de journalisation

Créer un profil de journalisation et ajouter une catégorie et un log handler de fichiers.

1. Créer le profil :

```
/subsystem=logging/logging-profile=accounts-app-profile:add
```

2. Créer un gestionnaire de fichiers

```
/subsystem=logging/logging-profile=accounts-app-profile/file-  
handler=ejb-trace-file:add(file={path=>"ejb-trace.log", "relative-  
to"=>"jboss.server.log.dir"})
```

```
/subsystem=logging/logging-profile=accounts-app-profile/file-  
handler=ejb-trace-file:write-attribute(name="level",  
value="DEBUG")
```

3. Créer une catégorie de logger

```
/subsystem=logging/logging-profile=accounts-app-  
profile/logger=com.company.accounts.ejbs:add(level=TRACE)
```

4. Assigner un gestionnaire de fichiers à une catégorie

```
/subsystem=logging/logging-profile=accounts-app-  
profile/logger=com.company.accounts.ejbs:add-handler(name="ejb-  
trace-file")
```

[Rapporter un bogue](#)

12.5.4. Spécifier un profil de journalisation dans une application

Une application spécifie le profil de journalisation à utiliser dans son fichier `MANIFEST.MF`.

Conditions préalables :

1. Vous devez connaître le nom du profil de journalisation qui a été défini sur le serveur pour cette application. Demandez à votre administrateur de systèmes le nom du profil à utiliser.

Procédure 12.8. Ajouter une configuration de profil de journalisation à une application

- **Modifier MANIFEST.MF**

Si votre application ne possède pas de fichier **MANIFEST.MF** : le créer avec le contenu suivant, en remplaçant *NAME* par le nom de profil qui convient.

```
Manifest-Version: 1.0
Logging-Profile: NAME
```

Si votre application contient déjà un fichier **MANIFEST.MF** : ajouter la ligne suivante, en remplaçant *NAME* par le nom du profil qui convient.

```
Logging-Profile: NAME
```

NOTE

Si vous utilisez Maven et **maven-war-plugin**, vous pourrez mettre votre fichier **MANIFEST.MF** dans **src/main/resources/META-INF/** et ajouter la configuration suivante à votre fichier **pom.xml**.

```
<plugin>
  <artifactId>maven-war-plugin</artifactId>
  <configuration>
    <archive>
      <manifestFile>src/main/resources/META-
INF/MANIFEST.MF</manifestFile>
    </archive>
  </configuration>
</plugin>
```

Quand l'application sera déployée, elle utilisera la configuration qui se trouve dans le profil de journalisation spécifié pour ses messages de journalisation.

[Rapporter un bogue](#)

12.5.5. Exemple de configuration de profil de journalisation

Cet exemple montre la configuration du profil de journalisation et l'application qui en fait usage. Cela comprend la session CLI affichée, la configuration XML qui est générée et le fichier **MANIFEST.MF** de l'application.

L'exemple de profil de journalisation a les caractéristiques suivantes :

- Le nom est **accounts-app-profile**.
- La catégorie de journalisation est **com.company.accounts.ejbs**.
- Le niveau de journalisations est **TRACE**.
- Le log handler est un gestionnaire de fichiers qui utilise **ejb-trace.log**.

Exemple 12.80. Session CLI

```
localhost:bin user$ ./jboss-cli.sh -c
```

```
[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile:add
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile/file-handler=ejb-trace-file:add(file=
{path=>"ejb-trace.log", "relative-to"=>"jboss.server.log.dir"})
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile/file-handler=ejb-trace-file:write-
attribute(name="level", value="DEBUG")
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-
profile/logger=com.company.accounts.ejbs:add(level=TRACE)
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile/logger=com.company.accounts.ejbs:add-
handler(name="ejb-trace-file")
{"outcome" => "success"}

[standalone@localhost:9999 /]
```

Exemple 12.81. Configuration XML

```
<logging-profiles>
  <logging-profile name="accounts-app-profile">
    <file-handler name="ejb-trace-file">
      <level name="DEBUG"/>
      <file relative-to="jboss.server.log.dir" path="ejb-
trace.log"/>
    </file-handler>
    <logger category="com.company.accounts.ejbs">
      <level name="TRACE"/>
      <handlers>
        <handler name="ejb-trace-file"/>
      </handlers>
    </logger>
  </logging-profile>
</logging-profiles>
```

Exemple 12.82. Fichier de l'application MANIFEST.MF

```
Manifest-Version: 1.0
Logging-Profile: accounts-app-profile
```

12.6. PROPRIÉTÉS DE LA CONFIGURATION DE JOURNALISATION

12.6.1. Propriétés root logger

Tableau 12.6. Propriétés root logger

Property	Datatype	Description
level	String	Le niveau maximum de messages log que le root logger souhaite enregistrer.
handlers	String[]	Une liste des log handlers utilisés par le root logger.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui exclut les entrées de journalisation qui <i>ne correspondent pas</i> à un modèle : <code>not (match("JBAS.*"))</code>



NOTE

Un `filter-spec` spécifié pour le root logger n'est *pas* hérité par les autres gestionnaires. Au lieu de cela, un `filter-spec` devra être spécifié pour chaque handler.

[Rapporter un bogue](#)

12.6.2. Propriétés de catégorie de journalisation

Tableau 12.7. Propriétés de catégorie de journalisation

Propriété	Datatype	Description
level	String	Le niveau maximum de messages log que le root logger souhaite enregistrer.
handlers	String[]	Une liste des log handlers associés au logger.
use-parent-handlers	Boolean	Si défini sur <code>true</code> , cette catégorie utilisera les log handlers du root logger en plus des handlers assignés.
catégorie	String	La catégorie de journalisation à partir de laquelle les messages de journalisation seront capturés.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : <code>not (match("JBAS.*"))</code>

[Rapporter un bogue](#)

12.6.3. Propriétés de log handlers de console

Tableau 12.8. Propriétés de log handlers de console

Propriété	Datatype	Description
level	String	Le niveau maximum de messages de journalisation que le log handler enregistre.
encoding	String	Définir la codification utilisée pour la sortie.
formatter	String	Le formateur de journalisation utilisé par ce log handler.
named-formatter	String	Le nom du formateur défini à utiliser avec le gestionnaire
target	String	Le flux de sortie du système vers lequel la sortie du log handler se dirige. Peut correspondre à System.err ou System.out pour le flux d'erreurs système ou standard out respectivement.
autoflush	Boolean	Si défini sur true, les messages de journalisation seront envoyés vers la cible des handlers immédiatement après la réception.
name	String	L'identifiant unique de ce log handler.
enabled	Boolean	Si défini sur true , le gestionnaire sera activé et fonctionnera normalement. Si défini sur false , le gestionnaire sera ignoré lors du traitement des messages de journalisation.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : <code>not (match("JBAS.*"))</code>

[Rapporter un bogue](#)

12.6.4. Propriétés de log handlers de fichiers

Tableau 12.9. Propriétés de log handlers de fichiers

Propriété	Datatype	Description
level	String	Le niveau maximum de messages de journalisation que le log handler enregistre.
encoding	String	Définir la codification utilisée pour la sortie.
formatter	String	Le formateur de journalisation utilisé par ce log handler.
named-formatter	String	Le nom du formateur défini à utiliser avec le gestionnaire

Propriété	Datatype	Description
append	Boolean	Si la valeur est true alors tous les messages rédigés par ce gestionnaire seront ajoutés au fichier si celui-ci existe déjà. Si la valeur est false, un nouveau fichier sera créé chaque fois que le serveur d'applications est lancé. Les modifications à append nécessitent un redémarrage du serveur pour qu'elles soient prises en compte.
autoflush	Boolean	Si défini sur true, les messages de journalisation seront envoyés vers les gestionnaires immédiatement après la réception.
name	String	L'identifiant unique de ce log handler.
file	Object	L'objet qui représente le fichier dans lequel la sortie de ce log handler est écrite. Il contient deux propriétés de configuration, relative-to et path .
relative-to	String	C'est une propriété de l'objet fichier qui correspond au répertoire où le fichier journal est écrit. Les variables de chemin d'accès de fichier JBoss EAP 6 peuvent être indiquées ici. La variable jboss.server.log.dir pointe vers le répertoire log/ du serveur.
path	String	C'est une propriété de l'objet fichier qui correspond au nom du fichier où seront écrits les messages du journal. C'est un nom de chemin d'accès relatif qui est ajouté à la valeur de la propriété relative-to pour déterminer le chemin d'accès complet.
enabled	Boolean	Si défini sur true , le gestionnaire sera activé et fonctionnera normalement. Si défini sur false , le gestionnaire sera ignoré lors du traitement des messages de journalisation.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : not (match("JBAS.*"))

Rapporter un bogue

12.6.5. Propriétés de log handlers périodiques

Tableau 12.10. Propriétés de log handlers périodiques

Propriété	Datatype	Description
append	Boolean	Si la valeur est définie sur true , alors tous les messages rédigés par ce gestionnaire seront ajoutés au fichier si celui-ci existe déjà. Si la valeur est false , un nouveau fichier sera créé chaque fois que le serveur d'applications est lancé. Les modifications à «append» (ajout) nécessitent un redémarrage du serveur pour qu'elles soient prises en compte.

Propriété	Datatype	Description
autoflush	Boolean	Si défini sur true , les messages de journalisation seront envoyés vers les gestionnaires immédiatement après la réception.
encoding	String	Définir la codification utilisée pour la sortie.
formatter	String	Le formateur de journalisation utilisé par ce log handler.
named-formatter	String	Le nom du formateur défini à utiliser avec le gestionnaire
level	String	Le niveau maximum de messages de journalisation que le log handler enregistre.
name	String	L'identifiant unique de ce log handler.
file	Object	L'objet qui représente le fichier dans lequel la sortie de ce log handler est écrite. Il contient deux propriétés de configuration, relative-to et path .
relative-to	String	C'est une propriété de l'objet fichier qui correspond au répertoire où le fichier journal est écrit. Les variables de chemin d'accès peuvent être indiquées ici. La variable jboss.server.log.dir pointe vers le répertoire log/ du serveur.
path	String	C'est une propriété de l'objet fichier qui correspond au nom du fichier où seront écrits les messages du journal. C'est un nom de chemin d'accès relatif qui est ajouté à la valeur de la propriété relative-to pour déterminer le chemin d'accès complet.
suffix	String	<p>Cette chaîne est ajoutée au nom de fichier des journaux en rotation et sert à déterminer la fréquence de rotation. Le format du suffixe est un point (.) suivi d'une chaîne date String qui puisse être interprétée par SimpleDateFormat. Le journal est mis en rotation sur la base de la plus petite unité de temps définie par le suffixe. Notez que la plus petite unité de temps autorisée pour l'attribut suffixe est la minute.</p> <p>Par exemple, une valeur de suffixe de .YYYY-MM-dd résultera en une rotation de journal quotidienne. Considérons que vous ayez un fichier journal server.log et un suffixe ayant comme valeur .YYYY-MM-dd, le fichier journal mis en rotation le 20 October 2014 se nommerait server.log.2014-10-19. Avec un gestionnaire de journal périodique, le suffixe inclut la valeur précédente de plus petite unité de temps. Dans cet exemple, la valeur de dd est 19, le jour précédent.</p>

Propriété	Datatype	Description
enabled	Boolean	Si défini sur true , le gestionnaire sera activé et fonctionnera normalement. Si défini sur false , le gestionnaire sera ignoré lors du traitement des messages de journalisation.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : <code>not (match("JBAS. *"))</code> .

[Rapporter un bogue](#)

12.6.6. Propriétés de log handlers de taille

Tableau 12.11. Propriétés de log handlers de taille

Propriété	Datatype	Description
append	Boolean	Si la valeur est définie sur true , alors tous les messages rédigés par ce gestionnaire seront ajoutés au fichier si celui-ci existe déjà. Si la valeur est false , un nouveau fichier sera créé chaque fois que le serveur d'applications est lancé. Les modifications à «append» (ajout) nécessitent un redémarrage du serveur pour qu'elles soient prises en compte.
autoflush	Boolean	Si défini sur true , les messages de journalisation seront envoyés vers les gestionnaires immédiatement après la réception.
encoding	String	Définir la codification utilisée pour la sortie.
formatter	String	Le formateur de journalisation utilisé par ce log handler.
named-formatter	String	Le nom du formateur défini à utiliser avec le gestionnaire
level	String	Le niveau maximum de messages de journalisation que le log handler enregistre.
name	String	L'identifiant unique de ce log handler.
file	Object	L'objet qui représente le fichier dans lequel la sortie de ce log handler est écrite. Il contient deux propriétés de configuration, relative-to et path .
relative-to	String	C'est une propriété de l'objet fichier qui correspond au répertoire où le fichier journal est écrit. Les variables de chemin d'accès peuvent être indiquées ici. La variable <code>jboss.server.log.dir</code> pointe vers le répertoire <code>log/</code> du serveur.

Propriété	Datatype	Description
path	String	C'est une propriété de l'objet fichier qui correspond au nom du fichier où seront écrits les messages du journal. C'est un nom de chemin d'accès relatif qui est ajouté à la valeur de la propriété relative-to pour déterminer le chemin d'accès complet.
rotate-size	Integer	La taille maximale que le fichier journal peut atteindre avant qu'il soit mis en rotation. Un seul caractère ajouté au nombre indique les unités de taille : b pour bytes, k pour kilobytes, m pour megabytes, g pour gigabytes. Par ex. 50m pour 50 megabytes.
max-backup-index	Integer	<p>Le nombre maximum de journaux en rotation conservés. Quand ce nombre est atteint, le journal le plus ancien est utilisé à nouveau. Valeur par défaut : 1.</p> <p>Disponible à partir de JBoss EAP 6.4. Si l'attribut suffix est utilisé, le suffixe des fichiers de journaux en rotation sera inclus dans l'algorithme de rotation. Quand le fichier journal est mis en rotation, le fichier le plus ancien dont le nom commence par name+suffix est supprimé, les fichiers journal en rotation restant auront leur suffixe numérique incrémenté et le nouveau fichier journal en rotation recevra le suffixe 1.</p> <p>Considérons que nous ayons le nom de fichier journal server.log, un suffixe ayant pour valeur .YYYY-mm, et un max-backup-index de valeur 3. Quand le fichier journal aura été mis en rotation deux fois, les noms de fichier pourront être server.log, server.log.2014-10.1 et server.log.2014-10.2. La prochaine fois que le fichier est mis en rotation, le fichier server.log.2014-10.2 sera supprimé, server.log.2014-10.1 sera renommée server.log.2014-10.2 et le fichier journal nouvellement mis en rotation sera nommé server.log.2014-10.1.</p>
enabled	Boolean	Si défini sur true , le gestionnaire sera activé et fonctionnera normalement. Si défini sur false , le gestionnaire sera ignoré lors du traitement des messages de journalisation.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : not(match("JBAS.*"))
rotate-on-boot	Boolean	Si la valeur est true , un nouveau fichier de journalisation sera créé au redémarrage du serveur. La valeur par défaut est false .

Propriété	Datatype	Description
suffix	String	<p>Disponible à partir de JBoss EAP 6.4. Ce string est inclus dans le suffixe ajouté aux fichiers de journaux en rotation. Le format du suffix est avec un (.) suivi d'un string de date qui peut être traité par la classe SimpleDateFormat</p> <p>Considérons que vous ayez un fichier journal server.log et un suffixe ayant comme valeur .YYYY-MM-dd, le fichier journal mis en rotation le 1 October 2014 se nommerait server.log.2014-10-1. Dans cet exemple, la portion 2014-10 du suffixe dérive de la valeur desuffixe.</p>

[Rapporter un bogue](#)

12.6.7. Propriétés de gestionnaires de journaux périodiques

Tableau 12.12. Propriétés des gestionnaires de journaux périodiques en rotation

Propriété	Datatype	Description
append	Boolean	<p>Si la valeur est définie sur true, alors tous les messages rédigés par ce gestionnaire seront ajoutés au fichier si celui-ci existe déjà. Si la valeur est false, un nouveau fichier sera créé chaque fois que le serveur d'applications est lancé. Les modifications à «append» (ajout) nécessitent un redémarrage du serveur pour qu'elles soient prises en compte.</p>
rotate-size	String	<p>La taille maximale que le fichier journal peut atteindre avant qu'il soit mis en rotation.</p>
max-backup-index	Integer	<p>Le nombre maximum de journaux en rotation conservés. Quand ce nombre est atteint, le journal le plus ancien est utilisé à nouveau. Valeur par défaut : 1.</p> <p>Si l'attribut suffix est utilisé, le suffixe des fichiers de journaux en rotation sera inclus dans l'algorithme de rotation. Quand le fichier journal est mis en rotation, le fichier le plus ancien dont le nom commence par name+suffix est supprimé, les fichiers journal en rotation restant auront leur suffixe numérique incrémenté et le nouveau fichier journal en rotation recevra le suffixe 1.</p> <p>Considérons que nous ayons le nom de fichier journal server.log, un suffixe ayant pour valeur .YYYY-mm, et un max-backup-index de valeur 3. Quand le fichier journal aura été mis en rotation deux fois, les noms de fichier pourront être server.log, server.log.2014-10.1 et server.log.2014-10.2. La prochaine fois que le fichier est mis en rotation, le fichier server.log.2014-10.2 sera supprimé, server.log.2014-10.1 sera renommée server.log.2014-10.2 et le fichier journal nouvellement mis en rotation sera nommé server.log.2014-10.1.</p>

Propriété	Datatype	Description
suffix	String	<p>Cette chaîne est ajoutée au nom de fichier des journaux en rotation et sert à déterminer la fréquence de rotation. Le format du suffixe est un point (.) suivi d'une chaîne date String qui puisse être interprétée par SimpleDateFormat. Le journal est mis en rotation sur la base de la plus petite unité de temps définie par le suffixe. Notez que la plus petite unité de temps atourisée pour l'attribut suffixe est la minute.</p> <p>Par exemple, une valeur de suffixe de .YYYY-MM-dd résultera en une rotation de journal quotidienne. Considérons que vous ayez un fichier journal server.log et un suffixe ayant comme valeur .YYYY-MM-dd, le fichier journal mis en rotation le 20 October 2014 se nommerait server.log.2014-10-19. Avec un gestionnaire de journal périodique, le suffixe inclut la valeur précédente de plus petite unité de temps. Dans cet exemple, la valeur de dd est 19, le jour précédent.</p>
autoflush	Boolean	Si défini sur true , les messages de journalisation seront envoyés vers les gestionnaires immédiatement après la réception.
file	Object	L'objet qui représente le fichier dans lequel la sortie de ce gestionnaire de journal est écrite. Il contient deux propriétés de configuration, relative-to et path .
relative-to	String	C'est une propriété de l'objet fichier qui correspond au répertoire où le fichier journal est écrit. Les variables de chemin d'accès peuvent être indiquées ici. La variable jboss.server.log.dir pointe vers le répertoire log/ du serveur.
path	String	C'est une propriété de l'objet fichier qui correspond au nom du fichier où seront écrits les messages du journal. C'est un nom de chemin d'accès relatif qui est ajouté à la valeur de la propriété relative-to pour déterminer le chemin d'accès complet.
enabled	Boolean	Si défini sur true , le gestionnaire sera activé et fonctionnera normalement. Si défini sur false , le gestionnaire sera ignoré lors du traitement des messages de journalisation.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : not(match("JBAS.*"))
rotate-on-boot	Boolean	Si la valeur est true , un nouveau fichier journal sera créé au redémarrage du serveur. La valeur par défaut est false .
formatter	String	Le formateur de journalisation utilisé par ce gestionnaire de journal.
level	String	Le niveau minimum de messages de journalisation que le gestionnaire de journal enregistre.

Propriété	Datatype	Description
name	String	L'identifiant unique de ce gestionnaire de journal.
named-formatter	String	Le nom du formateur défini à utiliser avec le gestionnaire
encoding	String	Définir la codification utilisée pour la sortie.

[Rapporter un bogue](#)

12.6.8. Propriétés de log handlers async

Tableau 12.13. Propriétés de log handlers async

Propriété	Datatype	Description
level	String	Le niveau maximum de messages de journalisation que le log handler enregistre.
name	String	L'identifiant unique de ce log handler.
encoding	String	Définir la codification utilisée pour la sortie.
formatter	String	Le formateur de journalisation utilisé par ce log handler.
queue-length	Integer	Nombre maximal de messages de journalisation gardés par le handler en attendant que les sub-handlers répondent.
overflow-action	String	La façon dont ce handler répond quand sa file d'attente est dépassée. Peut être défini sur BLOCK ou DISCARD . BLOCK fait patienter l'application de journalisation jusqu'à ce qu'il y ait suffisamment d'espace disponible dans la file d'attente. C'est le même comportement qu'avec un handler non-async. DISCARD permet à l'application de journalisation de continuer, mais le message de journalisation sera effacé.
subhandlers	String[]	Il s'agit de la liste de log handlers à laquelle cet handler async passe ses messages log.
enabled	Boolean	Si défini sur true , le gestionnaire sera activé et fonctionnera normalement. Si défini sur false , le gestionnaire sera ignoré lors du traitement des messages de journalisation.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : <code>not (match("JBAS. *"))</code>

[Rapporter un bogue](#)

12.7. EXEMPLE DE CONFIGURATION XML DE LOGGING

12.7.1. Échantillon de Configuration XML pour root logger

```
<root-logger>
  <level name="INFO"/>
  <handlers>
    <handler name="CONSOLE"/>
    <handler name="FILE"/>
  </handlers>
</root-logger>
```

[Rapporter un bogue](#)

12.7.2. Échantillon de Configuration XML pour une catégorie de journalisation

```
<logger category="com.company.accounts.rec">
  <handlers>
    <handler name="accounts-rec"/>
  </handlers>
</logger>
```

[Rapporter un bogue](#)

12.7.3. Échantillon de configuration XML pour un log handler de console

```
<console-handler name="CONSOLE">
  <level name="INFO"/>
  <formatter>
    <pattern-formatter pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n"/>
  </formatter>
</console-handler>
```

[Rapporter un bogue](#)

12.7.4. Échantillon de configuration XML pour un gestionnaire de journalisation de fichiers

```
<file-handler name="accounts-rec-trail" autoflush="true">
  <level name="INFO"/>
  <file relative-to="jboss.server.log.dir" path="accounts-rec-
trail.log"/>
  <append value="true"/>
</file-handler>
```

[Rapporter un bogue](#)

12.7.5. Échantillon de configuration XML pour un log handler périodique

```
<periodic-rotating-file-handler name="FILE">
  <formatter>
    <pattern-formatter pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t)
%s%E%n"/>
  </formatter>
```

```

<file relative-to="jboss.server.log.dir" path="server.log"/>
<suffix value=".yyyy-MM-dd"/>
<append value="true"/>
</periodic-rotating-file-handler>

```

[Rapporter un bogue](#)

12.7.6. Échantillon de configuration XML pour un log handler de taille

```

<size-rotating-file-handler name="accounts_debug" autoflush="false">
  <level name="DEBUG"/>
  <file relative-to="jboss.server.log.dir" path="accounts-debug.log"/>
  <rotate-size value="500k"/>
  <max-backup-index value="5"/>
  <append value="true"/>
</size-rotating-file-handler>

```

[Rapporter un bogue](#)

12.7.7. Échantillon de Configuration XML pour un gestionnaire de journal de taille de fichiers périodique en rotation

```

<periodic-size-rotating-file-handler name="Periodic_size_rotating_Handler"
autoflush="false">
  <level name="INFO"/>
  <file relative-to="jboss.server.log.dir" path="Sample.log"/>
  <max-backup-index value="1"/>
  <suffix value=".yyyy.MM.dd"/>
  <append value="false"/>
</periodic-size-rotating-file-handler>

```

[Rapporter un bogue](#)

12.7.8. Échantillon de Configuration XML pour un Log Handler Async

```

<async-handler name="Async_NFS_handlers">
  <level name="INFO"/>
  <queue-length value="512"/>
  <overflow-action value="block"/>
  <subhandlers>
    <handler name="FILE"/>
    <handler name="accounts-record"/>
  </subhandlers>
</async-handler>

```

[Rapporter un bogue](#)

CHAPITRE 13. INFINISPAN

13.1. INFINISPAN

Infinispan est une plateforme de grille de données Java. Il fournit une interface cache compatible [JSR-107](#) pour gérer les données mises en cache.

Les conteneurs cache d'infinispan sont utilisés dans JBoss Enterprise Application Platform 6 :

- `web` dans Web Session Clustering
- `ejb` dans Stateful Session Bean Clustering
- `hibernate` pour la mise en cache d'entité
- `singleton` pour la mise en cache de singleton

Chaque conteneur cache définit un "repl" et un "dist". Ces caches ne doivent pas être utilisés directement par les applications utilisateur.



IMPORTANT

Les utilisateurs peuvent ajouter les conteneurs cache et des caches, et les référencer via JNDI. Cependant, cela n'est pas pris en charge par JBoss EAP 6.

Pour obtenir plus d'informations sur la fonctionnalité Infinispan et ses options de configuration, consultez [Infinispan Documentation](#).

[Rapporter un bogue](#)

13.2. MODES DE CLUSTERING

Le clustering peut être configuré de deux façons différentes dans JBoss EAP 6 en utilisant Infinispan. La méthode qui convient à votre application dépendra de vos besoins. Il y a des considérations à prendre quand vous choisissez entre la disponibilité, la fiabilité et l'évolutivité de chaque mode. Avant de choisir un mode de clusterisation, vous devez identifier les principales caractéristiques de votre réseau, et équilibrer ces besoins.

Mode répliqué

Le mode répliqué détecte et ajoute automatiquement de nouvelles instances sur le cluster. Les changements faits à ces instances seront répliqués à tous les noeuds du cluster. Le mode répliqué fonctionne normalement mieux dans les petits clusters à cause du montant d'informations qui aura été répliqué sur le réseau. Infinispan peut être configuré pour utiliser UDP

Mode de distribution

Le mode de distribution permet à Infinispan de mettre le cluster à l'échelle linéairement. Le mode de distribution utilise un algorithme de hachage uniforme pour déterminer où un nouveau nœud doit être placé dans un cluster. Le nombre de copies d'informations à conserver est configurable. C'est un compromis entre le nombre de copies conservées, la durabilité des données et la performance : plus le nombre de copies qui restent sera grand, plus cela aura d'impact sur les performances, mais il sera alors moins probable que vous perdiez des données en cas de panne de serveur. L'algorithme de hachage permet également de réduire le trafic réseau en détectant les entrées sans multidiffusion ou en stockant des métadonnées.

Réplication synchrone et asynchrone

La réplication peut être effectuée soit en mode synchrone ou soit en mode asynchrone et le mode choisi dépend de vos besoins et de votre application. Avec la réplication synchrone, le thread qui gère la demande de l'utilisateur est bloqué jusqu'à ce que la réplication ait réussi. Quand la réplication a réussi, une réponse est envoyée au client, et le thread est libéré. La réplication synchrone a un impact sur le trafic réseau parce qu'elle requiert une réponse de chaque nœud du cluster. Elle a l'avantage, cependant, de veiller à ce que toutes les modifications soient apportées à tous les nœuds du cluster.

La réplication asynchrone est effectuée en arrière-plan. Infinispan implémente une file d'attente de réplication, qui est utilisée par un thread d'arrière-plan pour effectuer la réplication. La réplication est déclenchée sur une base de temps, ou sur la taille de la file d'attente. Une file d'attente de réplication permet de meilleures performances parce qu'il n'y a aucune conversation menée entre les nœuds du cluster. Le compromis avec la réplication asynchrone, c'est qu'elle n'est pas aussi précise. Les tentatives de réplication qui ont échoué sont inscrites dans un journal, et ne sont pas notifiées en temps réel.

[Rapporter un bogue](#)

13.3. CONTENEURS DE CACHE

Conteneurs de cache

Un conteneur de cache est le référentiel de caches utilisés par un sous-système. Dans Infinispan, les conteneurs de cache par défaut sont définis dans les fichiers de configuration xml (`standalone-ha.xml`, `standalone-full-ha.xml`, `domain.xml`). Un cache est défini comme le cache par défaut, et ce sera le cache qui sera utilisé pour le clustering.

Exemple 13.1. Définitions de conteneur cache du fichier de configuration `standalone-ha.xml`

```
<subsystem xmlns="urn:jboss:domain:infinispan:1.5">
  <cache-container name="singleton" aliases="cluster ha-
partition" default-cache="default">
    <transport lock-timeout="60000"/>
    <replicated-cache name="default" mode="SYNC"
batching="true">
      <locking isolation="REPEATABLE_READ"/>
    </replicated-cache>
  </cache-container>
  <cache-container name="web" aliases="standard-session-cache"
default-cache="repl" module="org.jboss.as.clustering.web.infinispan">
    <transport lock-timeout="60000"/>
    <replicated-cache name="repl" mode="ASYNC" batching="true">
      <file-store/>
    </replicated-cache>
    <replicated-cache name="sso" mode="SYNC" batching="true"/>
      <distributed-cache name="dist" l1-lifespan="0"
mode="ASYNC" batching="true">
        <file-store/>
      </distributed-cache>
    </cache-container>
```

Notez le cache par défaut défini dans chaque conteneur cache. Dans cet exemple, dans le conteneur cache `web`, le cache `repl` est défini par défaut. Le cache `repl` sera donc utilisé pour le clustering des sessions web.

Les conteneurs de cache et les attributs de cache peuvent être configurés par la console de gestion ou les commandes CLI, mais il n'est pas conseillé de modifier les noms des conteneurs cache, ni des caches eux-mêmes.

Configuration des conteneurs cache

Les conteneurs cache d'Infinispan peuvent être configurés par le CLI ou par la console de gestion.

Procédure 13.1. Configurer les conteneurs cache Infinispan dans la console de gestion

1. Sélectionner l'onglet **Configuration** en haut de l'écran.
2. En mode de domaine uniquement, sélectionner **ha** ou **full-ha** à partir du menu déroulant en haut et à gauche.
3. Étendre le menu **Subsystems**, puis étendre le menu **Infinispan**. Sélectionner **Cache Containers**.
4. Sélectionner un conteneur cache à partir du tableau **Cache Containers**.
5. **Ajouter, supprimer ou définir le conteneur de caches par défaut**
 - a. Pour créer un nouveau conteneur cache, cliquer sur **Add** dans le tableau **Cache Containers**.
 - b. Pour supprimer le conteneur cache, sélectionner le conteneur cache dans le tableau **Cache Containers**. Cliquer sur **Remove** puis sur **OK** pour confirmer.
 - c. Pour définir un conteneur cache comme défaut, cliquer sur **Set Default**, puis, saisir un nom de conteneur cache à partir de la liste de menu déroulant, puis sur **Save** pour confirmer.
6. Pour ajouter ou mettre à jour les attributs d'un conteneur de cache, sélectionner le conteneur cache dans le tableau **Cache Containers**. Sélectionnez en un à partir des onglets **Attributes**, **Transport** ou **Aliases** qui se trouvent dans la partie d'écran **Details**, puis cliquer sur **Edit**. Pour avoir des informations supplémentaires sur le contenu des onglets **Attributes**, **Transport** et **Aliases**, cliquer sur **Need Help?**.

Procédure 13.2. Configurer les conteneurs cache Infinispan dans l'interface CLI

1. Pour obtenir une liste des attributs configurables, saisir la commande CLI suivante :

```
/profile=profile name/subsystem=infinispan/cache-  
container=container name:read-resource
```

2. Vous pouvez utiliser l'interface CLI pour ajouter, supprimer, et mettre à jour les conteneurs cache. Avant d'utiliser des commandes avec les conteneurs cache, assurez-vous d'avoir le profil qui convient dans la commande CLI.

- a. **Ajouter un conteneur cache**

Pour ajouter un conteneur cache, basez votre commande sur l'exemple suivant :

```
/profile=profile-name/subsystem=infinispan/cache-  
container="cache container name":add
```

b. Supprimer un conteneur cache

Pour supprimer un conteneur cache, basez votre commande sur l'exemple suivant :

```
/profile=profile-name/subsystem=infinispan/cache-  
container="cache container name":remove
```

c. Mettez à jour les attributs de conteneur de cache

Utiliser l'opération `write-attribute` pour écrire une nouvelle valeur dans un attribut. Vous pouvez utiliser l'onglet de complétion pour terminer la chaîne de commande en cours, ainsi que pour exposer les attributs disponibles. L'exemple suivant met à jour `statistics-enable` à `true`.

```
/profile=profile name/subsystem=infinispan/cache-  
container=cache container name:write-attribute(name=statistics-  
enabled,value=true)
```

[Rapporter un bogue](#)

13.4. STATISTIQUES INFINISPAN

Les statistiques de temps d'exécution d'objets cache-container ou Infinispan cache peuvent être activés à but de surveillance. La collecte des statistiques n'est pas activée par défaut pour des raisons de performance.

**AVERTISSEMENT**

Activer les statistiques Infinispan peut avoir un impact négatif sur la performance des sous-systèmes Infinispan. Les statistiques ne devraient être activés que si besoin est.

La collecte de statistiques peut être activée pour chaque conteneur-cache, cache, ou les deux. L'option de statistiques pour chaque cache remplace l'option de cache-container. L'activation ou la désactivation de la collecte de statistiques d'un conteneur de cache entraînera tous les caches de ce conteneur à hériter la configuration, à moins qu'ils ne spécifient explicitement les leur. Si seulement un cache-container est activé pour les statistiques, des statistiques utiles seront disponibles.

[Rapporter un bogue](#)

13.5. ACTIVER LA COLLECTE DES STATISTIQUES D'INFINISPAN

La collecte des statistiques ne peut pas être activée à partir du fichier de configuration de démarrage (par exemple, `standalone.xml`, `standalone-ha.xml`, `domain.xml`) ou l'interface CLI.

[Rapporter un bogue](#)

13.5.1. Activer la collecte des statistiques d'Infinispan dans un fichier de configuration de démarrage

Procédure 13.3. Activer les statistiques d'Infinispan dans un fichier de configuration de démarrage

- Ajouter l'attribut `statistics-enabled=VALUE` dans la balise XML du `cache-container` ou du `cache` dans le sous-système d'Infinispan.

Exemple 13.2. Activer la collecte de statistiques pour un cache

```
<replicated-cache name="sso" mode="SYNC" batching="true" statistics-enabled="true"/>
```

Exemple 13.3. Activer la collecte de statistiques pour un cache-container

```
<cache-container name="singleton" aliases="cluster ha-partition" default-cache="default" statistics-enabled="true">
```

[Rapporter un bogue](#)

13.5.2. Active la collecte des statistiques d'Infinispan à partir de l'interface CLI**Procédure 13.4. Active la collecte des statistiques d'Infinispan à partir de l'interface CLI**

Dans cette procédure :

- `CACHE_CONTAINER` est le `cache-container` préféré (par exemple, `web`)
- `CACHE_TYPE` est le type de cache préféré (par exemple, `distributed-cache`)
- `CACHE` est le nom de cache (par exemple, `dist`)

1. Saisir la commande suivante :

```
/subsystem=infinispan/cache-container=CACHE_CONTAINER/CACHE_TYPE=CACHE:write-attribute(name=statistics-enabled,value=true)
```

2. Saisir la commande suivante pour recharger le serveur :

```
:reload
```

NOTE

Pour annuler la définition d'un attribut, saisir la commande suivante :

```
/subsystem=infinispan/cache-container=CACHE_CONTAINER/CACHE_TYPE=CACHE:undefine-attribute(name=statistics-enabled)
```

[Rapporter un bogue](#)

13.5.3. Vérifier que la collecte des statistiques d'Infinispan soit activée

Procédure 13.5. Vérifier que la collecte des statistiques d'Infinispan soit activée

Suivant que vous confirmiez que la collecte de statistiques est activée sur un **cache** ou sur un **cache-container**, utiliser une des commandes d'interface CLI suivantes :

- ○ **Pour un cache**

```
/subsystem=infinispan/cache-  
container=CACHE_CONTAINER/CACHE_TYPE=CACHE:read-  
attribute(name=statistics-enabled)
```

- ○ **Pour un cache-container**

```
/subsystem=infinispan/cache-container=CACHE_CONTAINER:read-  
attribute(name=statistics-enabled)
```

[Rapporter un bogue](#)

13.6. JGROUPS

13.6.1. JGroups

JGroups est un outil de messagerie qui permet aux développeurs de créer des applications de messagerie fiables où la fiabilité du système est en cause. JGroups peut être utilisé pour créer des clusters dont les nœuds peuvent envoyer des messages de l'un à l'autre.

Le sous-système de JGroups fournit tous les mécanismes de communication sur la façon dont les serveurs d'un cluster communiquent entre eux. EAP est préconfiguré avec deux piles de JGroups.

- UDP - les nœuds du cluster utilisent la multidiffusion UDP (User Datagram Protocol) pour communiquer entre eux. UDP est généralement plus rapide mais moins fiable que TCP.
- TCP - les nœuds du cluster utilisent TCP (Transmission Control Protocol) pour communiquer entre eux. TCP a tendance à être plus lent que l'UDP, mais plus fiable pour délivrer des données vers sa destination.

Les piles préconfigurées peuvent être utilisées, ou vous pouvez définir votre propre pile pour répondre aux besoins spécifiques de votre système.

[Rapporter un bogue](#)

13.7. RÉOLUTION DE PROBLÈMES JGROUPS

13.7.1. Les nœuds ne forment pas un cluster

Veillez à ce que votre machine soit configurée pour IP multidiffusion. Deux programmes de tests peuvent être utilisés dans ce but : McastReceiverTest et McastSenderTest. Par exemple :

```
java -cp $JBOSS_HOME/bin/client/jboss-client.jar
org.jgroups.tests.McastReceiverTest -mcast_addr 230.11.11.11 -port 5555 -
bind_addr $YOUR_BIND_ADDRESS
```

Dans un autre écran, démarrez **McastSenderTest**:

```
java -cp $JBOSS_HOME/bin/client/jboss-client.jar
org.jgroups.tests.McastSenderTest -mcast_addr 230.11.11.11 -port 5555 -
bind_addr $YOUR_BIND_ADDRESS
```

Si vous souhaitez vous relier à une carte d'interface de réseau particulière (NIC), utiliser **-bind_addr 192.168.0.2**, avec **192.168.0.2** pour adresse IP du NIC auquel vous souhaitez vous relier. Utiliser ce paramètre pour l'expéditeur et le receveur à la fois.

Vous devriez être en mesure de saisir vos entrées dans la fenêtre **McastSenderTest** et voir le résultat dans la fenêtre de **McastReceiverTest**. Si non, essayez d'utiliser **-ttl 32** dans l'expéditeur. Si cela échoue encore, contactez un administrateur de système pour vous aider à configurer correctement les adresses IP multidiffusion et demander à l'administrateur de veiller à ce que la multidiffusion fonctionne sur l'interface que vous avez sélectionnée ou, si les machines ont plusieurs interfaces, demandez qu'on leur dise quelle est l'interface appropriée. Une fois que vous saurez que multidiffusion fonctionne correctement sur chaque machine de votre cluster, vous pourrez répéter le test ci-dessus pour tester le réseau, en mettant l'expéditeur sur une machine et le récepteur sur une autre.

[Rapporter un bogue](#)

13.7.2. Les causes

Parfois, un membre est considéré suspect par FD parce qu'un « heartbeat ack » n'a pas été reçu depuis un moment *T* (défini par *timeout* et *max_tries*). Cela peut avoir plusieurs raisons, par ex., dans un cluster de A,B,C,D; C peut être suspecté si (notez que A ping B, B ping C, C ping D et D ping A) :

- B et C exécutent à 100% de CPU pendant plus de *T* secondes. Donc, même quand C envoie un accusé de réception de pulsation à B, B n'est sans doute pas à même de pouvoir le traiter car il est déjà à 100%.
- B ou C nettoient la mémoire, tout comme ci-dessus.
- Une combinaison des 2 cas ci-dessus
- Le réseau perd des packages. Cela a généralement lieu quand il y a beaucoup de trafic sur le réseau, et que le commutateur commence à lâcher des packages (diffusions tout d'abord, puis les IP multidiffusions, et enfin les paquets TCP).
- B ou C sont entrain de traiter un rappel. Disons que C a reçu un appel de méthode distant sur son canal et prend *T + 1* secondes afin de la traiter. Pendant ce temps, C ne traitera plus aucun message, ni les pulsations, et donc B ne recevra pas l'accusé de réception de pulsation et suspectera C.

[Rapporter un bogue](#)

CHAPITRE 14. JVM

14.1. JVM

14.1.1. Paramètres de configuration de JVM

Les paramètres de configuration de machines virtuelles Java (JVM) varient selon les instances de domaine géré et les instances de serveur autonome. Dans un domaine géré, les paramètres de JVM sont déclarés dans les fichiers de configuration `host.xml` et `domain.xml`, et ils sont déterminés par les composants de contrôleur de domaine chargés de lancer et d'arrêter le processus du serveur. Dans une instance de serveur autonome, les processus de démarrage de serveur peuvent passer des paramètres de ligne de commande au démarrage. Ceux-ci peuvent être déclarés depuis la ligne de commande ou via l'écran **System Properties** dans la console de gestion.

Domaine géré

Une caractéristique importante du domaine géré est la possibilité de définir des paramètres de la JVM à plusieurs niveaux. Vous pouvez configurer les paramètres de JVM personnalisés au niveau de l'hôte, par groupe de serveurs, ou par instance de serveur. Les éléments enfants plus spécialisés remplacent la configuration parent, permettant la déclaration des configurations de serveur spécifique sans nécessiter d'exclusions au niveau groupe ou hôte. Cela permet également à la configuration parent d'être héritée par les autres niveaux jusqu'à ce que les paramètres soient déclarés dans les fichiers de configuration ou transmis pendant le runtime.

Exemple 14.1. Les paramètres de configuration JVM du fichier de configuration du domaine

L'exemple suivant montre une déclaration JVM pour un groupe de serveurs dans le fichier de configuration `domain.xml`.

```
<server-groups>
  <server-group name="main-server-group" profile="default">
    <jvm name="default">
      <heap size="64m" max-size="512m"/>
    </jvm>
    <socket-binding-group ref="standard-sockets"/>
  </server-group>
  <server-group name="other-server-group" profile="default">
    <jvm name="default">
      <heap size="64m" max-size="512m"/>
    </jvm>
    <socket-binding-group ref="standard-sockets"/>
  </server-group>
</server-groups>
```

Dans cette instance, un groupe de serveurs appelé `main-server-group` déclare une taille de segment de 64 mégaoctets et une taille de segment maximale de 512 mégaoctets. Tout serveur qui appartient à ce groupe héritera de ces paramètres. Vous pouvez modifier ces paramètres pour le groupe dans son ensemble, par hôte ou serveur individuel.

Exemple 14.2. Les paramètres de configuration du domaine dans le fichier de configuration de l'hôte

L'exemple suivant montre une déclaration JVM pour un groupe de serveurs dans le fichier de configuration `host.xml`.

```
<servers>
  <server name="server-one" group="main-server-group" auto-start="true">
    <jvm name="default"/>
  </server>
  <server name="server-two" group="main-server-group" auto-start="true">
    <jvm name="default">
      <heap size="64m" max-size="256m"/>
    </jvm>
    <socket-bindings port-offset="150"/>
  </server>
  <server name="server-three" group="other-server-group" auto-
start="false">
    <socket-bindings port-offset="250"/>
  </server>
</servers>
```

Dans ce cas, un serveur appelé `server-two` appartient au groupe de serveurs nommé `main-server-group`, qui hérite les paramètres du groupe de JVM par défaut. Dans l'exemple précédent, la taille du segment principal de `main-server-group` a été fixée à 512 méga-octets. En déclarant une taille de segment basse de 256 méga-octets, `server-two` peut substituer les paramètres de `domain.xml` pour ajuster les performances comme vous le souhaitez.

Paramètres de configuration de serveur autonome en cours d'exécution

Les paramètres de JVM pour des instances de serveurs autonomes peuvent être déclarés pendant l'exécution en définissant la variable d'environnement `JAVA_OPTS` avant de démarrer le serveur. Un exemple de définition de la variable d'environnement `JAVA_OPTS` en ligne de commande Linux est :

```
[user@host bin]$ export JAVA_OPTS="-Xmx1024M"
```

La même configuration peut être utilisée dans un environnement Microsoft Windows, comme suit :

```
C:\> set JAVA_OPTS="Xmx1024M"
```

Alternativement, les paramètres de configuration JVM peuvent être ajoutés au fichier `standalone.conf` qui se trouve dans le dossier `EAP_HOME/bin` contenant des exemples d'options à passer à la JVM.



AVERTISSEMENT

En définissant la variable d'environnement `JAVA_OPTS`, vous redéfinissez les valeur par défaut de la variable d'environnement `JAVA_OPTS`. Cela peut compromettre ou résilier le démarrage de JBoss EAP.

14.1.2. Afficher le statut JVM dans la console de gestion

Conditions préalables

- [Section 2.2.2, « Démarrez JBoss EAP 6 comme un serveur autonome »](#)
- [Section 2.2.3, « Démarrez JBoss EAP 6 comme domaine géré »](#)
- [Section 3.4.2, « Se connecter à la console de gestion »](#)

Le statut de la Machine virtuelle Java (JVM) peut être affichée dans la console de gestion pour le serveur autonome ou un domaine géré. La console affiche l'utilisation de segments, leur non utilisation, et l'usage de threads du serveur. Malgré que les statistiques ne soient pas affichés en temps réel, vous pouvez actualiser l'affichage de la console pour donner un aperçu à jour des ressources de la machine virtuelle Java.

Le statut de la JVM affiche les valeurs suivantes.

Tableau 14.1. Attributs de Statut JVM

Type	Description
Max	Le montant de mémoire maximale pouvant être utilisé pour la gestion de la mémoire. La mémoire maximum disponible s'affiche sur la barre grise claire.
Utilisé	Le montant de mémoire utilisée. La mémoire utilisée s'affiche sur la barre grise claire.
Validé	Le montant de mémoire allouée à l'utilisation pour une machine virtuelle Java. La mémoire utilisée s'affiche sur la barre grise claire.

Procédure 14.1. Afficher le statut JVM dans la console de gestion

- **Affichage du statut de la JVM pour une instance de serveur autonome**
Sélectionner l'onglet **Runtime** en haut de l'écran. Étendre le menu **Status**, puis étendre le menu **Platform**. Sélectionner **JVM**.
 - **Afficher le statut de la JVM d'un domaine géré**
Sélectionner l'onglet **Runtime** en haut de l'écran. Étendre le menu **Server Status**, puis étendre le menu **Platform**. Sélectionner **JVM**.
- Le domaine géré peut rendre visibles toutes les instances de serveur dans le groupe de serveurs, mais ne vous permettra d'afficher qu'un seul serveur à la fois en sélectionnant dans le menu serveur. Pour afficher le statut des autres serveurs dans votre groupe de serveurs, cliquer sur **Change Server** en haut à gauche de l'écran pour sélectionner à partir de l'hôte et des serveurs affichés dans votre groupe. cliquer sur le bouton **Done** pour terminer.

Résultat

Le statut des paramètres de configuration de la JVM de l'instance de serveur est affiché.

[Rapporter un bogue](#)

14.1.3. Configuration d'une JVM

Les balises `<jvm></jvm>` prennent en charge l'utilisation des options `<jvm-options></jvm-options>`, pouvant être utilisées pour ajouter des paramètres à la configuration de la JVM en utilisant la balise `<option value="VALUE"/>`.

Exemple 14.3. Utilisation de `<jvm-options>`

```
<jvm name="default">
  <heap size="1303m" max-size="1303m"/>
  <permgen max-size="256m"/>
  <jvm-options>
    <option value="-XX:+UseCompressedOops"/>
  </jvm-options>
</jvm>
```

Configurer une JVM par le CLI

Pour configurer une JVM par le CLI, utiliser la syntaxe suivante :

```
# cd /server-group=main-server-group/jvm=default

# :add-jvm-option(jvm-option="-XX:+UseCompressedOops")
{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => {"main-server-group" => {"host" => {"master" => {
    "server-one" => {"response" => {
      "outcome" => "success",
      "response-headers" => {
        "operation-requires-restart" => true,
        "process-state" => "restart-required"
      }
    }
  }},
  "server-two" => {"response" => {
    "outcome" => "success",
    "response-headers" => {
      "operation-requires-restart" => true,
      "process-state" => "restart-required"
    }
  }
}
}}}
}

# :read-resource

# Expected Result:

[domain@localhost:9999 jvm=default] :read-resource
{
  "outcome" => "success",
  "result" => {
    "agent-lib" => undefined,
    "agent-path" => undefined,
    "env-classpath-ignored" => undefined,
    "environment-variables" => undefined,
```

```

    "heap-size" => "1303m",
    "java-agent" => undefined,
    "java-home" => undefined,
    "jvm-options" => ["-XX:+UseCompressedOops"],
    "max-heap-size" => "1303m",
    "max-permgen-size" => "256m",
    "permgen-size" => undefined,
    "stack-size" => undefined,
    "type" => undefined
  }
}

```

Supprimer l'entrée jvm-options

Pour supprimer l'entrée jvm-options, utiliser la syntaxe suivante :

```

# cd /server-group=main-server-group/jvm=default

# :remove-jvm-option(jvm-option="-XX:+UseCompressedOops")

# Expected Result:

[domain@localhost:9999 jvm=default] :remove-jvm-option(jvm-option="-
XX:+UseCompressedOops")
{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => {"main-server-group" => {"host" => {"master" => {
    "server-one" => {"response" => {
      "outcome" => "success",
      "response-headers" => {
        "operation-requires-restart" => true,
        "process-state" => "restart-required"
      }
    }
  }},
  "server-two" => {"response" => {
    "outcome" => "success",
    "response-headers" => {
      "operation-requires-restart" => true,
      "process-state" => "restart-required"
    }
  }
}
}}
}}}}
}

```

[Rapporter un bogue](#)

CHAPITRE 15. SOUS-SYSTÈME WEB

15.1. CONFIGURER LE SOUS-SYSTÈME WEB

Le sous-système web peut être configuré par la console de gestion ou par l'interface CLI.

Dans la console de gestion, cliquer sur l'onglet **Configuration** en haut de l'écran, développez le menu de **sous-systèmes**, puis développez le menu **Web**. cliquer sur l'élément de menu de **Servlet/HTTP** pour configurer **Global**, **JSP**, **Connector** et **Virtual Servers**. cliquer sur l'élément de menu **Web Services** pour configurer le sous-système des services web.

Dans l'interface CLI, tous les paramètres du sous-système web sont configurés en utilisant le format de commande standard.



NOTE

Le composant `mod_cluster` n'est disponible que si votre profil est `ha` ou `full-ha`, dans un domaine géré, ou si vous démarrez votre serveur autonome avec le profil `standalone-ha` ou `standalone-full-ha`. Pour obtenir des informations supplémentaires sur la configuration `mod_cluster` voir [Section 17.6.2, « Configurer le sous-système mod_cluster »](#).

[Rapporter un bogue](#)

15.2. CONFIGURER LE TIMEOUT DE SESSION HTTP

Le délai d'expiration (timeout) de la session HTTP définit la période après laquelle la session est considérée comme invalide parce qu'il n'y n'avait aucune activité dans le délai imparti. Modifier le délai d'expiration de la session HTTP exige que toutes les instances de JBoss EAP concernées soient redémarrées. Jusqu'à ce que cela soit fait, la valeur de timeout de session HTTP originale s'applique.

Le timeout de session HTTP peut être configuré à plusieurs endroits. Les voici par ordre de précedence :

- Application - défini dans le fichier de configuration `web.xml` de l'application. Pour plus de détails, consulter *Configure the HTTP Timeout per Application* dans le *Development Guide* (Guide de développement).
- Serveur - indiqué par l'attribut `default-session-timeout`. Cette configuration n'est disponible qu'à partir de JBoss EAP 6.4.
- Valeur par défaut - 30 minutes.

Procédure 15.1. Configurer le timeout de session HTTP par la console de gestion

1. Cliquer sur l'onglet **Configuration**, naviguer dans **Subsystems**, **Web**, puis dans l'élément de menu **Servlet/HTTP**.
2. Cliquer sur l'onglet **Global** qui se trouve dans **Servlet/HTTP Configuration**.
3. Cliquer sur l'option **Edit**.
4. Saisir la nouvelle valeur pour le timeout de session par défaut `Default session timeout`.

5. Cliquer sur le bouton **Enregistrer**.

6. Recharger le serveur JBoss EAP

Procédure 15.2. Configurer le timeout de session HTTP par l'interface de commandes CLI.



NOTE

Ajouter le préfixe `/host=HOST_NAME` à la commande si vous devez appliquer les changements à un domaine géré.

1. Indiquer la valeur du timeout de session HTTP désiré.

```
/subsystem=web:write-attribute(name=default-session-timeout,
value=timeout)
```

2. Recharger le serveur JBoss EAP

```
:reload
```

[Rapporter un bogue](#)

15.3. CONFIGURER LE SOUS-SYSTÈME DU SERVLET/HTTP

Dans la console de gestion, cliquer sur l'onglet **Configuration** en haut de l'écran, développer le menu de **sous-systèmes**, puis développer le menu **Web**. Cliquer sur l'élément de menu de **Servlet/HTTP**. cliquer sur le nom du composant, puis cliquer sur **modifier**. Cliquer sur le bouton **avancé** pour afficher les options avancées. Les options sont expliquées ci-dessous.

Si les commandes d'interface CLI doivent être appliquées à un profil, ajouter le préfixe `/profile=PROFILE`.

Exemple 15.1. Configurez les nom de cette instance

```
/profile=full-ha/subsystem=web:write-attribute(name=instance-
id,value=worker1)
```

Options de configuration globale

Timeout de session par défaut

Disponible à partir de JBoss EAP 6.4. Timeout de session par défaut du conteneur web.

Attribut d'interface CLI : `default-session-timeout`

Serveur virtuel par défaut

Le serveur virtuel par défaut du conteneur web.

L'attribut d'interface CLI : `default-virtual-server`

ID de l'instance

L'identificateur utilisé pour activer l'affinité de session dans les scénarios d'équilibrage de charge. L'identificateur doit être unique sur tous les serveurs du cluster JBoss EAP et est ajouté à l'identificateur de session généré. Cela permet au proxy frontal de transmettre la session spécifique à une même instance de JBoss EAP. L'ID d'instance n'est pas défini par défaut.

Attribut d'interface CLI : `instance-id`

Natif

Ajouter le listener d'initialisation natif au conteneur web.

Attribut d'interface CLI : `native`. Valeurs : `true` ou `false`. Valeur par défaut : `false`.

Options de configuration JSP

Délai entre les vérifications

Cliquer sur **Advanced** pour voir cette option, si elle est cachée. Valeur en secondes qui détermine la fréquence des vérifications de mises à jour JSP par un processus en arrière-plan. La valeur par défaut est 0.

Attribut d'interface CLI : `check-interval`

Développement

Si sur `true`, active Development Mode, qui produit davantage d'informations verbeuses de débogage. Valeur par défaut : `false`.

Attribut d'interface CLI : `development`

Désactivé

Si sur `true`, désactive le conteneur Java ServerPages (JSP). Valeur par défaut `false`. Utile si vous n'utilisez pas les pages JSP.

Attribut d'interface CLI : `disabled`

Garder Générés

Cliquer sur **Advanced** pour voir cette option, si elle est cachée. Si sur `true`, garde les servlets générés. Option définie sur `true` par défaut.

Attribut d'interface CLI : `instance-id`

Afficher les fragments de source

Cliquer sur **Advanced** pour voir cette option, si elle est cachée. Si sur `true`, le fragment de source JSP est affiché quand une erreur d'exécution a lieu. Valeur par défaut : `true`.

Attribut d'interface CLI : `display-source-fragment`

X Powered By

Si défini sur `true`, JSP engine est commercialisé `x-powered-by`. Valeur par défaut : `true`.

Attribut d'interface CLI : `x-powered-by`

Les connecteurs AJP and HTTP utilisent `mod_cluster`, `mod_jk`, `mod_proxy`, `ISAPI`, et `NSAPI` pour l'équilibrage des charges et pour le clustering HA. Pour configurer un connecteur, sélectionner l'onglet **Connectors** et cliquer sur **Add**. Pour supprimer un connecteur, le sélectionner et cliquer sur **Remove**. Pour modifier un connecteur, le sélectionner et cliquer sur **Edit**.

Quand vous créez un nouveau connecteur par l'interface CLI, ses options sont définies aussitôt, comme dans la commande suivante :

Exemple 15.2. Créer un nouveau connecteur

```
/profile=full-ha/subsystem=web/connector=ajp/:add(socket-binding=ajp,scheme=http,protocol=AJP/1.3,secure=false,name=ajp,max-post-size=2097152,enabled=true,enable-lookups=false,redirect-port=8433,max-save-post-size=4096)
```

Options de connecteurs

Nom

Un nom unique de connecteur, à but d'affichage.

Attribut d'interface CLI : `name`

Liaisons de sockets

La liaison de socket nommée à laquelle le connecteur doit se lier. La liaison de socket est un mappage entre un nom de socket et un port réseau. Les liaisons de sockets sont configurées pour chaque serveur autonome, ou par l'intermédiaire de groupes de liaisons de sockets dans un domaine géré. Un groupe de liaisons de sockets est appliqué à un groupe de serveurs.

Attribut d'interface CLI : `socket-binding`

Schéma

Le schéma de connecteur web, comme HTTP ou HTTPS.

Attribut d'interface CLI :

Protocol

Le protocole de connecteur web à utiliser, comme AJP ou HTTP.

Attribut d'interface CLI : `protocol`

Activé

Indique si le connecteur web connecté est activé.

Attribut d'interface CLI : `enabled`

Redirect Port

Utilisé pour indiquer un numéro de port à utiliser en cas de redirection, normalement pour une redirection vers un connecteur AJP ou HTTPS (sécurisé).

Attribut d'interface CLI : `redirect-port`

Redirect Binding

La redirection de liaison est similaire pour rediriger le port en termes de comportement, sauf qu'il faut la spécification d'un nom de liaison de socket comme *valeur* au lieu d'un numéro de port. L'option `redirect-liaison` fournit une plus grande flexibilité de configuration car il permet l'utilisation d'une liaison de socket pré-définie liaison (https, AJP etc.) vers le port spécifique de redirection. Elle donne les mêmes résultats que l'option `redirection-port`

Attribut d'interface CLI : `redirect-binding`

Pour configurer des serveurs virtuels, cliquer sur l'onglet **Virtual Servers**. Utiliser le bouton **Add** pour ajouter un nouveau serveur virtuel. Pour modifier ou supprimer un serveur virtuel, le sélectionner et cliquer le bouton **Edit** ou **Remove**.

Quand vous ajoutez un nouveau serveur virtuel par l'interface CLI, toutes les options requises sont définies en même temps, comme par la commande suivante.

Exemple 15.3. Ajouter un nouveau serveur virtuel

```
/profile=full-ha/subsystem=web/virtual-server=default-host/:add(enable-welcome-root=true,default-web-module=ROOT.war,alias=["localhost","example.com"],name=default-host)
```

Options de serveurs virtuels

Nom

Nom unique de serveur virtuel, à but d'affichage.

Attribut d'interface CLI : `name`

Alias

Une liste de noms d'hôtes qui doivent correspondre à ce serveur virtuel. Dans la console de gestion, utiliser un nom d'hôte par ligne.

Attribut d'interface CLI : `alias`

Module par défaut

Le module dont l'application web doit être déployée au nœud racine de ce serveur virtuel et qui sera affiché quand aucun répertoire n'est donné par la requête HTTP.

Attribut d'interface CLI : `default-web-module`

[Rapporter un bogue](#)

15.4. REMPLACER L'APPLICATION WEB WELCOME PAR DÉFAUT

JBoss EAP 6 inclut l'application Welcome, qui s'affiche quand on ouvre l'URL du serveur sur le port 8080. Vous pouvez remplacer cette application par votre propre application web, en suivant la procédure suivante.

Procédure 15.3. Remplacer l'application web Welcome par défaut par votre propre application web**1. Désactiver l'application Welcome**

Utiliser le script d'interface CLI `EAP_HOME/bin/jboss-cli.sh` pour exécuter la commande suivante. Vous aurez sans doute besoin de modifier un profil de domaine géré, ou retirer une portion de la commande `/profile=default` du serveur autonome.

```
/profile=default/subsystem=web/virtual-server=default-host:write-attribute(name=enable-welcome-root,value=false)
```

2. Configurer votre application web par le contexte root.

Afin de configurer votre application web, pour qu'elle utilise (/) comme adresse URL, modifier son fichier `jboss-web.xml`, qui se trouve dans le répertoire `META-INF/` ou `WEB-INF/`.

Remplacer sa directive `<context-root>` par une autre directive qui ressemble à ce qui suit.

```
<jboss-web>
  <context-root>/</context-root>
</jboss-web>
```

3. Déployer votre application.

Déployer votre application pour le groupe de serveurs ou le serveur que vous avez modifié lors de la première étape. L'application est maintenant disponible sur `http://SERVER_URL:PORT/`.

[Rapporter un bogue](#)

15.5. PROPRIÉTÉS SYSTÈME

Cette section répertorie les propriétés système qui peuvent être utilisées pour modifier le comportement de JBossWeb par défaut. Les propriétés système peuvent être définies dans la configuration de JBoss Enterprise Web Application. Vous devez le redémarrer pour qu'elles soient appliquées au sous-système web.

L'exemple suivant illustre comment modifier les propriétés système dans JBossWeb

```
standalone@localhost:9999 [/] ./system-
property=org.apache.catalina.JSESSIONID:add(value="MYID")
{"outcome" => "success"}
standalone@localhost:9999 [/] :shutdown
Communication error: Channel closed
Closed connection to localhost:9999
```

Pour certaines propriétés, vous pouvez les redémarrer en utilisant la commande « reload ».

Voici un exemple sur la façon de démarrer à nouveau en utilisant la commande « reload ».

```
[standalone@localhost:9999 /] :reload
{
  "outcome" => "success",
  "response-headers" => {
    "operation-requires-reload" => true,
    "process-state" => "reload-required"
  }
}
```


Tableau 15.1. Conteneur de servlets et de connecteurs

Attribut	Description
jvmRoute	Fournit une valeur par défaut pour l'attribut <i>jvmRoute</i> . Ne remplace pas la valeur générée automatiquement qui est utilisée quand on utilise <code>ha-read</code> avec une configuration comme <code>standalone-ha.xml</code> . Prend en charge <code>:reload</code> .
org.apache.tomcat.util.buf.StringCache.byte.enabled	Si sur <code>true</code> , le cache de String n'est pas activé pour <code>ByteChunk</code> . Si la valeur n'est pas spécifiée, la valeur par défaut (<code>false</code>) sera utilisée.
org.apache.tomcat.util.buf.StringCache.char.enabled	Si sur <code>true</code> , le cache de String est activé pour <code>CharChunk</code> . Si la valeur n'est pas spécifiée, la valeur par défaut (<code>false</code>) sera utilisée.
org.apache.tomcat.util.buf.StringCache.cacheSize	La taille du cache de String. Si la valeur n'est pas spécifiée, la valeur par défaut de 5000 sera utilisée.
org.apache.tomcat.util.buf.StringCache.maxStringLength	La longueur maximum de String à mettre en cache. Si la valeur n'est pas spécifiée, la valeur par défaut de 128 sera utilisée.
org.apache.tomcat.util.http.FastDateFormat.CACHE_SIZE	La taille du cache pour utiliser les valeurs de date analysées et formatées. Si la valeur n'est pas spécifiée, la valeur par défaut (1000) sera utilisée.
org.apache.catalina.core.StandardService.DELAY_CONNECTOR_STARTUP	Si sur <code>true</code> , le démarrage du connecteur aura lieu automatiquement. Est normalement en mode intégré.
org.apache.catalina.connector.Request.SESSION_ID_CHECK	Si défini sur <code>true</code> , le conteneur de servlet vérifiera que la session existe bien dans un contexte avec l'id de session indiqué avant de créer une session avec cet id.
org.apache.coyote.Constants.USE_CUSTOM_STATUS_MSG_IN_HEADER	Si sur <code>true</code> , les messages de statut HTTP personnalisés sont utilisés dans les en-têtes HTTP. Les utilisateurs doivent s'assurer que ce genre de message est codé ISO-8859-1, particulièrement si l'entrée fournie par l'utilisateur est incluse dans le message, afin d'éviter une vulnérabilité XSS possible. Si la valeur n'est pas spécifiée, la valeur <code>false</code> par défaut sera utilisée.

Attribut	Description
org.apache.tomcat.util.http.Parameters.MAX_COUNT	Le montant maximal de paramètres pouvant être traités dans un corps de message. En cas de dépassement, le traitement échoue en créant l'exception <code>IllegalStateException</code> . La valeur par défaut est 512 paramètres.
org.apache.tomcat.util.http.MimeHeaders.MAX_COUNT	Le montant maximal d'en têtes pouvant être envoyés dans la demande HTTP. En cas de dépassement, le traitement échoue en créant l'exception <code>IllegalStateException</code> . La valeur par défaut est 128 paramètres.
org.apache.tomcat.util.net.MAX_THREADS	Le nombre maximum de threads qu'un connecteur utilisera pour traiter des requêtes. La valeur par défaut est 32 x <code>Runtime.getRuntime().availableProcessors()</code> . (512 x <code>Runtime.getRuntime().availableProcessors()</code> pour le connecteur JIO)
org.apache.coyote.http11.Http11Protocol.MAX_HEADER_SIZE	La taille maximum d'en-têtes HTTP, en octets. Si cette valeur est dépassée, le traitement échoue accompagné de l'exception <code>ArrayOutOfBoundsException</code> . La valeur par défaut est de 8192 octets.
org.apache.coyote.http11.Http11Protocol.COMPRESSION	Permet d'utiliser la compression simple avec le connecteur HTTP. La valeur par défaut est désactivé, et la compression peut être activée à l'aide de la valeur « on » pour l'activer conditionnellement, ou « force » pour l'activer en continu.
org.apache.coyote.http11.Http11Protocol.COMPRESSION_RESTRICTED_UA	Agent regexps d'utilisateur qui ne recevront pas de contenu compressé. La valeur par défaut est vide.
org.apache.coyote.http11.Http11Protocol.COMPRESSION_MIME_TYPES	Prefixes de contenu compressible. La valeur par défaut est <code>text/html,text/xml,text/plain</code> .
org.apache.coyote.http11.Http11Protocol.COMPRESSION_MIN_SIZE	Taille minimum de contenu à comprimer. La valeur par défaut est de 2048 octets.
org.apache.coyote.http11.DEFAULT_CONNECTION_TIMEOUT	Timeout de socket par défaut. La valeur par défaut est de 60000 ms.
org.jboss.as.web.deployment.DELETE_WORK_DIR_ON_CONTEXT_DESTROY	Utilisez cette propriété pour supprimer les fichiers <code>.java</code> et <code>.class</code> pour que les sources JSP soient recompilés. La valeur par défaut est <code>false</code> . Timeout de socket par défaut pour les garder en vie. La valeur par défaut est de -1 ms, ce qui signifie qu'il utilisera le timeout de socket par défaut.

Attribut	Description
org.apache.tomcat.util.buf.StringCache.trainThreshold	Indique le nombre de fois que <code>toString()</code> devra être invoqué avant d'activer le cache. La valeur par défaut est de 100000 .

Tableau 15.2. EL

Attribut	Description
org.apache.el.parser.COERCE_TO_ZERO	Si sur true, lorsque l'on oblige les expression à correspondre à des nombres "" et que null sera obligé de correspondre à zéro comme l'exige la spécification. Si la valeur n'est pas spécifiée, la valeur par défaut (true) sera utilisée.

Tableau 15.3. JSP

Attribut	Description
org.apache.jasper.compiler.Generator.VAR_EXPRESSIONFACTORY	Le nom de la variable à utiliser pour l'usine d'expressions de langage d'expressions. Si la valeur n'est pas spécifiée, la valeur par défaut <code>_el_expressionfactory</code> sera utilisée.
org.apache.jasper.compiler.Generator.VAR_INSTANCEMANAGER	Le nom de la variable à utiliser pour l'usine d'expressions de langage d'expressions. Si la valeur n'est pas spécifiée, la valeur par défaut <code>_el_expressionfactory</code> sera utilisée.
org.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING	Si sur false, les exigences pour les signes de cotation d'échappement des attributs JSP sont assouplies afin qu'un signe de cotation requis manquant ne cause pas d'erreur. Si la valeur n'est pas spécifiée, la valeur par défaut de true conforme à la spécification sera utilisée.
org.apache.jasper.Constants.DEFAULT_TAG_BUFFER_SIZE	Toute mémoire tampon de balise qui s'étend au delà de la taille <code>org.apache.jasper.Constants.DEFAULT_TAG_BUFFER_SIZE</code> est détruite et une nouvelle mémoire tampon est créée à la taille par défaut. Si la valeur n'est pas spécifiée, la valeur par défaut de 512 sera utilisée.
org.apache.jasper.runtime.JspFactoryImpl.USE_POOL	Si sur true, un pool de <code>ThreadLocal PageContext</code> sera utilisé. Si la valeur n'est pas spécifiée, la valeur par défaut de true sera utilisée.

Attribut	Description
org.apache.jasper.runtime.JspFactoryImpl.POOL_SIZE	La taille de ThreadLocal PageContext. Si la valeur n'est pas spécifiée, la valeur par défaut de 8 sera utilisée.
org.apache.jasper.Constants.JSP_SERVLET_BASE	La taille de ThreadLocal PageContext. Si la valeur n'est pas spécifiée, la valeur par défaut de org.apache.jasper.runtime.HttpJspBase sera utilisée.
org.apache.jasper.Constants.SERVICE_METHOD_NAME	Le nom de la méthode de service à utiliser pour la classe de base. Si la valeur n'est pas spécifiée, la valeur par défaut _jspService sera utilisée.
org.apache.jasper.Constants.SERVLET_CLASSPATH	Le nom de l'attribut ServletContext qui contient le chemin de classe du JSP. Si la valeur n'est pas spécifiée, la valeur par défaut org.apache.catalina.jsp_classpath sera utilisée.
org.apache.jasper.Constants.JSP_FILE	Le nom de l'attribut de la demande de l'élément <jsp-file> d'une définition de servlet. S'il est présent sur une demande, cela remplacera la valeur retournée par request.getServletPath() pour sélectionner la page JSP à traiter. Si la valeur n'est pas spécifiée, la valeur par défaut org.apache.catalina.jsp_file sera utilisée.
org.apache.jasper.Constants.PRECOMPILE	Le nom du paramètre de requête qui indique au moteur JSP qui servira à pré-générer le servlet mais pas à l'invoquer. Si la valeur n'est pas spécifiée, la valeur par défaut org.apache.catalina.jsp_precompile sera utilisée.
org.apache.jasper.Constants.JSP_PACKAGE_NAME	Le nom de paquet par défaut à utiliser pour les pages jsp compilées. Si la valeur est non spécifiée, la valeur par défaut à utiliser sera org.apache.jsp.
org.apache.jasper.Constants.TAG_FILE_PACKAGE_NAME	Le nom de paquet par défaut à utiliser pour les gestionnaires de balises des fichiers de balises. Si la valeur est non spécifiée, la valeur par défaut à utiliser sera org.apache.jsp.tag
org.apache.jasper.Constants.TEMP_VARIABLE_NAME_PREFIX	Le préfixe à utiliser pour générer les noms de variables temporaires. Si la valeur n'est pas spécifiée, la valeur par défaut _jspx_temp sera utilisée.
org.apache.jasper.Constants.USE_INSTANCE_MANAGER_FOR_TAGS	Si sur true, le gestionnaire d'instances sera utilisé pour obtenir des instances de gestionnaires de balises. Si la valeur n'est pas spécifiée, la valeur true sera utilisée.

Attribut	Description
org.apache.jasper.Constants.INJECT_TAGS	Si sur true, les annotations spécifiées dans les balises seront traitées et injectées. Cela peut avoir un impact sur les performances lors de l'utilisation de balises simples, ou si « tag pooling » est désactivé. Si la valeur n'est pas spécifiée, false sera utilisé.

Tableau 15.4. Sécurité

Attribut	Description
org.apache.catalina.connector.RECYCLE_FACADES	Si sur true ou si un gestionnaire de sécurité est utilisé, un nouvel objet de façade sera créé pour chaque demande. Si la valeur n'est pas spécifiée, la valeur par défaut false sera utilisée.
org.apache.catalina.connector.CoyoteAdapter.ALLOW_BACKSLASH	Si sur true, le caractère '\' sera autorisé en tant que délimiteur de chemin d'accès. Si la valeur n'est pas spécifiée, la valeur par défaut false sera utilisée.
org.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH	Si sur true, les expressions '%2F' and '%5C' sont autorisés en tant que délimiteurs de chemin d'accès. Si la valeur n'est pas spécifiée, la valeur par défaut false sera utilisée.

Tableau 15.5. Spécification

Attribut	Description
----------	-------------

Attribut	Description
org.apache.catalina.STRICT_SERVLET_COMPLIANCE	<p>Si la valeur n'est pas spécifiée, la valeur true sera utilisée. Si sur true, les actions suivantes auront lieu :</p> <ul style="list-style-type: none"> • tout objet encapsulé de demande ou de réponse passé à un répartiteur d'applications est vérifié pour s'assurer qu'il a enveloppé la demande originale ou réponse. (SRV.8.2 / SRV.14.2.5.1) • un appel à Response.getWriter() si aucun codage de caractères qui n'a pas été spécifié résulte en appels ultérieurs à Response.getCharacterEncoding() retournant ISO-8859-1 et l'en-tête de la réponse de type de contenu inclura un charset = ISO-8859-1 composant. (SRV.15.2.22.1) • chaque requête associée à une session provoque la mise à jour de la dernière date d'accès indépendamment de si oui ou non la requête a explicitement accédé à la session. (SRV.7.6)
org.apache.catalina.core.StandardWrapperValve.STRICT_SERVLET_STATS	<p>Si sur true ou si org.apache.catalina.STRICT_SERVLET_COMPLIANCE a la valeur true, le wrapper recueillera les statistiques de JSR-77 pour les servlets individuels. Si la valeur n'est pas spécifiée, la valeur par défaut false est utilisée.</p>
org.apache.catalina.session.StandardSession.ACTIVITY_CHECK	<p>Si sur true ou si org.apache.catalina.STRICT_SERVLET_COMPLIANCE est sur true, Tomcat vérifie le nombre de requêtes actives pour chaque session. Quand on détermine si une session reste valide, toute séance comportant au moins une demande active est toujours considérée comme valide. Si la valeur n'est pas spécifiée, la valeur par défaut false est utilisée.</p>

[Rapporter un bogue](#)

15.6. LES COOKIES DE GESTION DE SESSIONS HTTP-ONLY

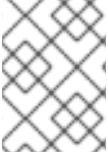
L'attribut `http-only` des cookies de gestion de sessions atténue les risques de failles de sécurité en limitant l'accès des API non - HTTP (comme JavaScript). Cette restriction permet d'atténuer la menace d'interception de cookies de session suite à des attaques de scripts inter-sites. Côté client, les cookies ne sont pas accessibles en JavaScript ou dans les autres méthodes de scripts. Cela s'applique uniquement aux cookies de gestion de sessions et non pas aux autres cookies de navigateur. Par défaut, l'attribut `http-only` est activé.

Vous devez ajouter SSO au serveur virtuel du sous-système web pour utiliser l'attribut `http-only`.

Exemple 15.4. Ajouter SSO au serveur virtuel

Saisir la commande CLI suivante pour ajouter SSO au serveur virtuel dans le sous-système web.

```
/subsystem=web/virtual-server=default-host/sso=configuration:add
```

**NOTE**

JSESSIONID et JSESSIONIDSSO sont des cookies de suivi de sessions. Par défaut, elles sont `http-only` et on ne doit pas y accéder par des scripts.

Exemple 15.5. Vérifier l'attribut `http-only`

Saisir la commande CLI suivante pour vérifier la valeur de l'attribut `http-only`.

```
/subsystem=web/virtual-server=default-host/sso=configuration:read-
resource-description

{
  "outcome" => "success",
  "result" => {
    "description" => "The SSO configuration for this virtual
server.",
    "access-constraints" => {"sensitive" => {"web-sso" => {"type"
=> "web"}}}},
    "attributes" => {
      "http-only" => {
        "type" => BOOLEAN,
        "description" => "The cookie http-only flag will be
set.",
        "expressions-allowed" => true,
        "nillable" => true,
        "default" => true,
        "access-type" => "read-write",
        "storage" => "configuration",
        "restart-required" => "all-services"
      }
      ...
    }
  }
}
```

Exemple 15.6. Activer l'attribut `http-only`

Saisir la commande CLI suivante pour activer l'attribut `http-only`.

```
/subsystem=web/virtual-server=default-host/sso=configuration:write-
attribute(name=http-only,value=true)
```

CHAPITRE 16. SOUS-SYSTÈME DE SERVICES WEB

16.1. CONFIGURER LES OPTIONS DE SERVICES WEB

Pour configurer les options de Services Web, cliquer sur **Web Services**. Les options sont expliquées dans le tableau ci-dessous.

Tableau 16.1. Options de configuration des Services Web

Option	Description	CLI Command
Modifier l'adresse WSDL	Indique si l'adresse WSDL peut être modifiée par les applications. Valeur par défaut true .	<pre>/profile=full-ha/subsystem=webservices/:write-attribute(name=modify-wsd-address,value=true)</pre>
Hôte WSDL	Le contrat WSDL d'un service Web JAX-WS inclut un élément <code><soap:address></code> qui pointe vers l'emplacement du point de terminaison. Si la valeur de <code><soap:address></code> est un URL valide, elle n'est pas remplacée à moins que modify-wsd-address soit défini à la valeur true . Si la valeur de <code><soap:address></code> n'est pas un URL valide, elle est remplacée en utilisant les valeurs wsd-host et wsd-port ou wsd-secure-port . Si wsd-host est défini sur jbossws.undefined.host , l'adresse hôte de l'auteur de la demande est utilisée lorsque <code><soap:address></code> est réécrite. Par défaut, \${jboss.bind.address:127.0.0.1} , qui utilise 127.0.0.1 si aucune adresse de liaison est spécifiée lors du démarrage de JBoss EAP 6.	<pre>/profile=full-ha/subsystem=webservices/:write-attribute(name=wsd-host,value=127.0.0.1)</pre>
Port WSDL	Le port non-sécurisé utilisé pour écrire à nouveau l'adresse SOAP. Si non défini (défaut), le port sera identifié en demandant la liste des connecteurs installés.	<pre>/profile=full-ha/subsystem=webservices/:write-attribute(name=wsd-port,value=80)</pre>

Option	Description	CLI Command
Port sécurisé WSDL	Le port sécurisé utilisé pour écrire à nouveau l'adresse SOAP. Si non défini (défaut), le port sera identifié en demandant la liste des connecteurs installés.	<code>/profile=full-ha/subsystem=webservices/:write-attribute(name=wsdl-secure-port,value=443)</code>

**NOTE**

Vous aurez sans doute besoin de modifier le profil pour modifier un profil de domaine géré différent, ou supprimer la partie `/profile=full-ha` de la commande d'un serveur autonome.

Sous-système de services web

Pour activer la journalisation dans Apache CXF, configurer la propriété système suivante dans le fichier `standalone/domain.xml`:

```
<system-properties>
<property name="org.apache.cxf.logging.enabled" value="true"/>
</system-properties>
```

[Rapporter un bogue](#)

CHAPITRE 17. HTTP CLUSTERING ET ÉQUILIBRAGE DES CHARGES

17.1. CONVENTIONS DE NOM DE SERVEUR HTTP

Les tableaux suivants contiennent des conventions de nommage ou d'emplacement utilisés dans on discute de sujets HTTPD.

Tableau 17.1. Le serveur Apache HTTP fourni par le système d'exploitation

Systèmes d'exploitation	<i>HTTPD_CONF.D</i>	<i>HTTPD_CONF</i>	<i>HTTPD_MODULES</i>
Red Hat Enterprise Linux (httpd)	<code>/etc/httpd/conf.d</code>	<code>/etc/httpd/conf</code>	<code>/etc/httpd/modules</code>
HPUX (Web Server Suite)	Voir la note ci-dessous.	<code>/opt/hpws/apache/conf</code>	<code>/opt/hpws/apache/modules</code>



NOTE

Il n'y a pas de fichier `conf.d` dans la suite du serveur Web HP-UX pour le serveur Apache HTTP. En revanche, vous pouvez en créer un :

Procédure 17.1. titre

1. Créer `/opt/hpws/apache/conf.d`.
2. Ajouter `Include conf.d/*.conf` dans le fichier `httpd.conf`.

Tableau 17.2. JBoss Enterprise Web Server

Systèmes d'exploitation	<i>HTTPD_CONF.D</i>	<i>HTTPD_CONF</i>	<i>HTTPD_MODULES</i>	Distribution
Red Hat Enterprise Linux	<code>/EWS_HOME/httpd/conf.d</code>	<code>/EWS_HOME/httpd/conf</code>	<code>/EWS_HOME/httpd/modules</code>	zip
Red Hat Enterprise Linux	<code>/etc/httpd/conf.d</code>	<code>/etc/httpd/conf</code>	<code>/usr/lib/httpd/modules</code>	rpm
Solaris	<code>/EWS_HOME/etc/httpd/conf.d</code>	<code>/EWS_HOME/etc/httpd/conf</code>	<code>/EWS_HOME/lib/httpd/modules</code>	zip
Windows	<code>/EWS_HOME/etc/httpd/conf.d</code>	<code>/EWS_HOME/etc/httpd/conf</code>	<code>/EWS_HOME/lib/httpd/modules</code>	zip



NOTE

Variations de chemins d'accès :

- Si vous utilisez une architecture 64-bit, modifiez les chemins de fichiers dans le tableau ci-dessus pour utiliser le répertoire `lib64/`.
- Si vous utilisez une installation Red Hat Enterprise Linux 7, le répertoire `httpd` s'appelle `httpd22`.

[Rapporter un bogue](#)

17.2. INTRODUCTION

17.2.1. Clusters haute disponibilité (HA) et clusters d'équilibrage des charges

Le terme *Clustering* se réfère à l'utilisation de ressources multiples, comme des serveurs, comme s'ils constituaient une seule entité. Les deux principaux types de clustering sont *Load balancing (LB)* et *High-availability (HA)*. Dans un groupement LB, toutes les ressources exécutent en même temps, et une couche de gestion se charge de répartir la charge de travail entre elles.

Dans le clustering HA, une ressource exécute, et une autre est prête à prendre position quand la première est rendue disponible. Le but du clustering HA est de réduire les chances de panne niveau matériel, logiciels ou réseau.

JBoss Enterprise Application Platform supporte le clustering à plusieurs niveaux. Certains sous-systèmes que l'on puisse rendre disponibles sont les suivants :

- Les instances du serveur d'applications
- Les applications web, lorsqu'elles sont utilisées en conjonction avec le serveur interne JBoss Web, Apache HTTP, Microsoft IIS ou Oracle iPlanet Web Server.
- Les EJB (Enterprise JavaBeans) avec, ou sans état
- Mécanismes Single Sign On (SSO)
- Cache distribué
- Sessions HTTP
- Les services JMS et les MDB (Messages Driven Beans)

Le clustering est rendu disponible par deux sous-systèmes : `jgroups` et `modcluster` dans JBoss EAP 6. Les profils `ha` et `full-ha` ont ces systèmes activés. Dans JBoss EAP 6, ces services démarrent et se ferment à la demande, mais ils ne démarreront que si une application configurée comme `distributable` est déployée sur les serveurs.

Infinispan est fourni comme fournisseur de cache dans JBoss EAP 6. Infinispan gère le clustering et la réplication des caches de JBoss EAP 6.

[Rapporter un bogue](#)

17.2.2. Composants pouvant bénéficier de la haute disponibilité (HA)

La haute disponibilité (HA) tombe dans un certain nombre de larges catégories de JBoss EAP 6.

Le conteneur

Plusieurs instances de JBoss EAP 6 (exécutant en tant que serveur autonome) ou les membres d'un groupe de serveurs (exécutant en tant que domaine géré) peuvent être configurés pour être hautement disponibles. Cela signifie que si une instance ou un membre est arrêté ou disparaît du groupement, sa charge de travail sera déplacée vers un père. La charge de travail peut être gérée de manière à fournir une fonctionnalité d'équilibrage de la charge, afin que les serveurs ou les groupes de serveurs avec des ressources plus ou moins supérieures puissent prendre une part plus importante à la charge de travail, ou qu'une capacité supplémentaire puisse être ajoutée pendant les périodes de forte charge.

Le serveur web

Le serveur web lui-même peut être groupé pour HA, à l'aide d'un des mécanismes d'équilibrage de charge compatible. Le plus souple est le connecteur `mod_cluster`, qui est intégré dans le conteneur de JBoss EAP. Les autres possibilités incluent les connecteurs Apache `mod_jk` ou `mod_proxy`, ou les re-directionneurs ISAPI et NSAPI.

L'application

Les application déployées peuvent être rendues HA (Highly Available) à cause de la spécification Java Enterprise Edition 6 (Java EE 6). Les EJB de session avec ou sans état peuvent être clusterisés, de façon à ce que si le nœud impliqué dans le travail disparaît, un autre nœud prendra sa place, et dans le cas de beans de session avec état, préservera l'état.

Rapporter un bogue

17.2.3. Connecteurs HTTP - Aperçu général

JBoss EAP 6 a la possibilité d'utiliser des mécanismes d'équilibrage de charge et de haute disponibilité intégrés à des serveurs web externes, tels que Apache Web Server, IIS de Microsoft et Oracle iPlanet. JBoss EAP 6 communique avec le serveur web externe à l'aide d'un connecteur HTTP. Ces connecteurs HTTP sont configurés dans le sous-système de web de JBoss EAP 6.

Les serveurs web incluent des modules informatiques qui contrôlent la façon dont les requêtes HTTP sont routées vers les nœuds de JBoss EAP 6. Chacun de ces modules varie dans la façon dont il fonctionne et sur la façon dont il est configuré. Les modules sont configurés pour équilibrer les charges de travail entre plusieurs nœuds de JBoss EAP 6, pour déplacer des charges de travail vers d'autres serveurs en cas d'échec, ou les deux. Ces fonctions sont appelées *équilibrage de charge et Haute disponibilité (HA)*.

JBoss EAP 6 prend en charge un certain nombre de connecteurs HTTP. Celui que vous choisirez dépendra du serveur web que vous utilisez et de la fonctionnalité dont vous aurez besoin.

Le tableau ci-dessous liste les différences entre les différents connecteurs HTTP compatibles avec JBoss EAP 6. Pour obtenir les dernières informations sur les configurations prises en charge pour les connecteurs HTTP, voir <https://access.redhat.com/site/articles/111663>.

Tableau 17.3. Caractéristiques et contraintes des connecteurs HTTP

Connec-teur	Web server	Systèmes d'exploitation pris en charge	Protocoles pris en charge	S'adapte au statut de déploiement	Prend en charge une sticky session
mod_cluster	httpd dans JBoss Enterprise Web Server, httpd fourni par un système d'exploitation (Red Hat Enterprise Linux, Hewlett-Packard HP-UX)	Red Hat Enterprise Linux, Microsoft Windows Server, Oracle Solaris, Hewlett-Packard HP-UX	HTTP, HTTPS, AJP	Oui. Détecte le déploiement et l'annulation du déploiement d'applications et décide dynamiquement s'il faut diriger les demandes clients vers un serveur, basé sur la question de savoir si l'application est déployée sur ce serveur.	Oui
mod_jk	httpd dans JBoss Enterprise Web Server, httpd fourni par un système d'exploitation (Red Hat Enterprise Linux, Hewlett-Packard HP-UX)	Red Hat Enterprise Linux, Microsoft Windows Server, Oracle Solaris, Hewlett-Packard HP-UX	AJP	Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application.	Oui
mod_proxy	httpd dans JBoss Enterprise Web Server	Red Hat Enterprise Linux, Microsoft Windows Server, Oracle Solaris	HTTP, HTTPS, AJP	Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application.	Oui
ISAPI	Microsoft IIS	Microsoft Windows Server	AJP	Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application.	Oui

Connecteur	Web server	Systèmes d'exploitation pris en charge	Protocoles pris en charge	S'adapte au statut de déploiement	Prend en charge une sticky session
NSAPI	Oracle iPlanet Web Server	Oracle Solaris	AJP	Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application.	Oui

Pour en savoir plus sur les connecteurs HTTP

- [Section 17.6.1, « Le connecteur HTTP mod_cluster »](#)
- [Section 17.7.1, « Le connecteur Apache mod_jk HTTP »](#)
- [Section 17.8.1, « Le connecteur Apache mod_proxy HTTP »](#)
- [Section 17.9.1, « Internet Server API \(ISAPI\) »](#)
- [Section 17.10.1, « Netscape Server API \(NSAPI\) »](#)

JBoss EAP 6 prend en charge les configurations disponibles ici : <https://access.redhat.com/site/articles/111663>.

[Rapporter un bogue](#)

17.2.4. Types de nœuds

Nœud de connecteur HTTP

Un *worker node* (connu sous le simple nom de *worker*) est un serveur JBoss EAP 6 qui accepte des requêtes d'un ou plusieurs serveurs Web faisant face au client, tel qu'Apache HTTP Server, Microsoft IIS, ou Oracle iPlanet Web Server.

Pour avoir un aperçu des connecteurs HTTP pris en charge par JBoss EAP 6 et sur la façon de les configurer, voir [Section 17.2.3, « Connecteurs HTTP - Aperçu général »](#).

Nœud de cluster

Un nœud de *cluster node* (ou *node*) est un membre d'un groupement de serveurs. Un tel cluster peut être en équilibrage de charge, en haute disponibilité, ou les deux. Dans un cluster d'équilibrage de charge, un gestionnaire central distribue également la charge de travail parmi ses nœuds, par mesure d'égalité suivant la situation particulière. Dans un cluster de haute disponibilité (HA), certains nœuds travaillent activement, tandis que d'autres sont en attente d'intervenir si un des nœuds actifs quitte le cluster.

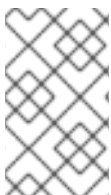
[Rapporter un bogue](#)

17.3. CONFIGURATION DE CONNECTEUR

17.3.1. Définir les pools de thread pour le connecteur HTTP dans JBoss EAP 6

Résumé

Les pools de threads de JBoss EAP 6 peuvent être partagés entre les différents éléments à l'aide du modèle `Executor`. Ces pools peuvent être partagés non seulement par les différents connecteurs (HTTP), mais aussi par d'autres composants de JBoss EAP 6 qui prennent en charge le modèle `Executor`. Obtenir le pool de threads de connecteurs HTTP qui corresponde à vos exigences de performance web actuel est un art délicat et nécessite une étroite surveillance du pool de threads en cours et des exigences de charge web en cours ou anticipées. Dans cette tâche, vous allez apprendre à définir un pool de threads de connecteur HTTP en utilisant le modèle `Executor`. Vous apprendrez comment définir cela en utilisant à la fois l'interface par ligne de commande (CLI) et en modifiant le fichier de configuration XML.



NOTE

Si vous exécutez JBoss EAP en mode de domaine, ajouter le préfixe `/profile=PROFILE_NAME` à toutes les commandes d'interface CLI dans cette procédure.

Procédure 17.2. Installation d'un pool de threads pour un connecteur HTTP

1. Définir une usine de threads

Ouvrir votre fichier de configuration (`standalone.xml` si vous modifiez un serveur autonome ou `domain.xml` si vous modifiez une configuration basée domaine. Ce fichier se trouve dans le dossier `EAP_HOME/standalone/configuration` ou dans `EAP_HOME/domain/configuration`).

Ajouter l'entrée de sous-système suivante, en modifiant les valeurs en fonction de vos besoins de serveur.

```
<subsystem xmlns="urn:jboss:domain:threads:1.1">
  <thread-factory name="http-connector-factory" thread-name-
pattern="HTTP-%t" priority="9" group-name="uq-thread-pool"/>
</subsystem>
```

Si vous préférez utiliser l'interface en ligne de commande pour cette tâche, alors exécutez la commande suivante dans une invite de commande CLI :

```
[standalone@localhost:9999 /] ./subsystem=threads/thread-
factory=http-connector-factory:add(thread-name-pattern="HTTP-%t",
priority="9", group-name="uq-thread-pool")
```

2. Créer un exécuteur

Vous pouvez utiliser une des six classes d'exécuteur intégrées pour qu'elle agisse en tant qu'exécuteur pour cette usine :

- o **unbounded-queue-thread-pool** : ce type de pool de threads accepte toujours les tâches. Si une valeur moindre que la valeur maximale de threads est en cours d'exécution, un nouveau thread démarre pour exécuter la tâche soumise. Sinon, la tâche sera placée dans une file d'attente FIFO illimitée qui sera exécutée lorsqu'un thread sera disponible.



NOTE

Le type d'exécuteur en single-thread fourni par `Executors.singleThreadExecutor()` est essentiellement un exécuteur de file d'attente illimitée avec une limite de thread d'un (1). Ce type d'exécuteur est déployé par l'élément `unbounded-queue-thread-pool-executor`.

- **bounded-queue-thread-pool** : ce type d'exécuteur maintient une file d'attente de longueur fixe et deux tailles de pool : une taille de base `core` et une taille maximale `maximum`. Lorsqu'une tâche est acceptée, si le nombre de pool de threads en cours d'exécution est inférieur à la taille de `core`, un nouveau thread est démarré pour exécuter la tâche. Si l'espace reste dans la file d'attente, la tâche est placée dans la file d'attente. Si le nombre de pool de threads en cours d'exécution est inférieur à la taille `maximum`, un nouveau thread est démarré pour exécuter la tâche. Si un blocage est activé sur l'exécuteur, le thread appelant se bloque jusqu'à ce que l'espace soit rendu disponible dans la file d'attente. La tâche est déléguée à l'exécuteur de la procédure de transfert si un exécuteur de la procédure de transfert est configuré. Dans le cas contraire, la tâche sera rejetée.
- **blocking-bounded-queue-thread-pool** : un exécuteur de pool de threads avec une file d'attente délimitée où les tâches de soumission des threads peuvent bloquer. Un tel pool de threads a une taille de base et une taille maximale, ainsi qu'une longueur de la file d'attente spécifiée. Lorsqu'une tâche est soumise, si le nombre de threads en cours d'exécution est inférieur à la taille de base, un nouveau thread sera créé. Sinon, s'il y a de la place dans la file d'attente, la tâche sera mise en file d'attente. Dans le cas contraire, si le nombre de threads en cours d'exécution est inférieur à la taille maximale, un nouveau thread sera créé. Dans le cas contraire, l'appelant sera bloqué jusqu'à ce que de la place se libère dans la file d'attente.
- **queueless-thread-pool** : parfois, il faut un pool de threads simple pour exécuter des tâches dans des threads séparés, en réutilisant les threads ayant terminé leurs tâches sans aucune file d'attente intermédiaire. Ce type de pool est idéal pour le traitement des tâches qui sont longues, en utilisant sans doute des E/S bloquantes, puisque les tâches sont toujours démarrées immédiatement après l'acceptation plutôt que d'accepter une tâche et puis retarder son exécution avant que d'autres tâches en cours d'exécution soient terminées. Ce type d'exécuteur est déclaré à l'aide de l'élément `queueless-thread-pool-executor`.
- **blocking-queueless-thread-pool** : un exécuteur de pool de threads sans file d'attente où les tâches de soumission de threads peuvent bloquer. Lorsqu'une tâche est soumise, si le nombre de threads en cours d'exécution est inférieur à la taille maximale, un nouveau thread sera créé. Dans le cas contraire, l'appelant sera bloqué jusqu'à ce qu'un autre thread se termine et accepte le nouveau.
- **scheduled-thread-pool** : il s'agit d'un type spécial d'exécuteur dont le but est d'exécuter des tâches à des moments et intervalles précis, basés sur la classe `java.util.concurrent.ScheduledThreadPoolExecutor`. Ce type d'exécuteur est configuré dans l'élément `scheduled-thread-pool-executor` :

Dans cet exemple, nous utiliserons `unbounded-queue-thread-pool` comme exécuteur. Modifier les valeurs des paramètres `max-threads` et `keepalive-time` selon les besoins de votre serveur.

```
<unbounded-queue-thread-pool name="uq-thread-pool">
```



```
<thread-factory name="http-connector-factory" />
<max-threads count="10" />
<keepalive-time time="30" unit="seconds" />
</unbounded-queue-thread-pool>
```

Ou bien, si vous préférez utiliser l'interface CLI :

```
[standalone@localhost:9999 /] ./subsystem=threads/unbounded-queue-
thread-pool=uq-thread-pool:add(thread-factory="http-connector-
factory", keepalive-time={time=30, unit="seconds"}, max-threads=30)
```

3. Forcez le connecteur web HTTP à utiliser ce pool de threads

Dans le même fichier de configurations, cherchez l'élément de connecteur HTTP dans le sous-système web et modifiez-le en utilisant le pool de threads défini dans les étapes suivantes.

```
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-
binding="http" executor="uq-thread-pool" />
```

Ou bien, si vous préférez utiliser l'interface CLI :

```
[standalone@localhost:9999 /] ./subsystem=web/connector=http:write-
attribute(name=executor, value="uq-thread-pool")
```

4. Redémarrer le serveur

Redémarrer le serveur (autonome ou domaine) pour que les changements puissent prendre effet. Utiliser les commandes d'interface CLI suivantes pour confirmer si les changements des étapes ci-dessus ont eu lieu :

```
[standalone@localhost:9999 /] ./subsystem=threads:read-
resource(recursive=true)
{
  "outcome" => "success",
  "result" => {
    "blocking-bounded-queue-thread-pool" => undefined,
    "blocking-queueless-thread-pool" => undefined,
    "bounded-queue-thread-pool" => undefined,
    "queueless-thread-pool" => undefined,
    "scheduled-thread-pool" => undefined,
    "thread-factory" => {"http-connector-factory" => {
      "group-name" => "uq-thread-pool",
      "name" => "http-connector-factory",
      "priority" => 9,
      "thread-name-pattern" => "HTTP-%t"
    }},
    "unbounded-queue-thread-pool" => {"uq-thread-pool" => {
      "keepalive-time" => {
        "time" => 30L,
        "unit" => "SECONDS"
      },
      "max-threads" => 30,
      "name" => "uq-thread-pool",
      "thread-factory" => "http-connector-factory"
    }
  }
}
```

```

}
[standalone@localhost:9999 /] ./subsystem=web/connector=http:read-
resource(recursive=true)
{
  "outcome" => "success",
  "result" => {
    "configuration" => undefined,
    "enable-lookups" => false,
    "enabled" => true,
    "executor" => "uq-thread-pool",
    "max-connections" => undefined,
    "max-post-size" => 2097152,
    "max-save-post-size" => 4096,
    "name" => "http",
    "protocol" => "HTTP/1.1",
    "proxy-name" => undefined,
    "proxy-port" => undefined,
    "redirect-port" => 443,
    "scheme" => "http",
    "secure" => false,
    "socket-binding" => "http",
    "ssl" => undefined,
    "virtual-server" => undefined
  }
}

```

Résultat

Vous avez pu créer une usine de threads et un exécuteur, tout en modifiant votre connecteur HTTP pour qu'il utilise ce pool de threads.

[Rapporter un bogue](#)

17.4. CONFIGURATION DU SERVEUR WEB

17.4.1. Le serveur Apache HTTP Autonome

JBoss EAP 6 est testé et pris en charge avec le serveur Apache HTTP qui est inclus avec les versions Red Hat Enterprise Linux 6 certifiées. Le serveur Apache HTTP est également disponible pour les autres systèmes d'exploitation, tels que Microsoft Windows Server. Cependant, comme le serveur Apache HTTP est un produit distinct, produit de la Fondation Apache, il était difficile auparavant d'être certain que la version de serveur Apache HTTP qu'un client utilisait était compatible avec JBoss EAP.

Le serveur Apache HTTP Autonome est maintenant disponible en tant que téléchargement séparé dans JBoss EAP 6. Ceci simplifie l'installation et la configuration dans les environnements autres que Red Hat Enterprise Linux, ou sur des systèmes qui déjà ont une configuration de serveur Apache HTTP et qui souhaitent utiliser une instance distincte pour les applications web. Vous pouvez télécharger ce serveur HTTP Apache en tant que téléchargement distinct dans le Portail de Services Client, qui se trouve dans la liste des téléchargements de JBoss EAP 6 disponibles pour votre plate-forme d'installation.

[Rapporter un bogue](#)

17.4.2. Installer le serveur Apache HTTP inclus dans JBoss EAP 6

Conditions préalables

- Accès root-level ou admin.
- Une version prise en charge de Java a été installée.
- Les packages suivants ont été installés :
 - `krb5-workstation`
 - `mod_auth_kerb` (requis pour la fonctionnalité Kerberos)
 - `eLinks` (requis pour la fonctionnalité `apachectl`)
 - Dans Red Hat Enterprise Linux 7, `apr-util-ldap` (fonction d'authentification LDAP)
- L'APR (Apache Portability Runtime) doit être installé. Sous Red Hat Enterprise Linux, installer le package `apr-util-devel`.

L'archive zip Apache HTTP Server contient des liens symboliques vers plusieurs modules Kerberos, et c'est la raison pour laquelle `mod_auth_kerb` est un prérequis. Si la fonctionnalité Kerberos n'est pas requise, vous n'aurez pas besoin d'installer le package `mod_auth_kerb` et le lien symbolique associé pourra être supprimé : `EAP_HOME/httpd/modules/mod_auth_kerb.so`.

Pour obtenir des informations sur l'installation du serveur Apache HTTP dans un environnement de serveur Microsoft Windows, voir le paragraphe *Configuring the Environment* qui se trouve dans la section *Installing Enterprise Web Server on Windows* du guide *JBoss Enterprise Web Server 2 Installation Guide*.



NOTE

Si vous utilisez le module CovalentSNMP dans Microsoft Windows Server, les fichiers suivants devront avoir l'extension `.sample` supprimée. Aussi, le fichier `snmpd.conf` devra être déplacé vers le répertoire `jboss-ews-2.1/etc/httpd/conf\`.

```
jboss-ews-2.1/etc/httpd/conf.d/mod_rt.conf.sample
jboss-ews-2.1/etc/httpd/conf.d/mod_snmp.conf.sample
jboss-ews-2.1/etc/httpd/conf.d/snmpd/snmpd.conf.sample
```

Procédure 17.3. Installer le serveur Apache HTTP

1. **Naviguer dans la liste des téléchargements de JBoss EAP de votre plateforme dans le portail clients de Red Hat.**

Connectez-vous dans le Portail clients à <https://access.redhat.com>. Cliquez sur **Downloads**, puis **Red Hat JBoss Enterprise Application Platform** dans la liste de **Product Downloads**. Sélectionner la version JBoss EAP qui convient à partir du menu déroulant **Version**.

2. **Sélectionner le binaire httpd de la liste.**

Chercher l'option **Apache HTTP Server** pour votre système d'exploitation et votre architecture. Cliquez sur le lien **Download**. Un fichier ZIP qui contient la distribution HTTP se télécharge dans votre ordinateur.

3. **Extraire le Zip dans le système où le binaire du serveur Apache HTTP exécutera.**

Extraire le fichier Zip sur votre serveur préféré à un emplacement temporaire. Le fichier Zip

contiendra le répertoire `httpd` sous le dossier `jboss-ews-version-number`. Copier le dossier `httpd` et le placer à l'intérieur du répertoire où vous avez installé JBoss EAP 6, couramment appelé `EAP_HOME`.

Votre serveur Apache HTTP se trouve maintenant dans le répertoire `EAP_HOME/httpd/`. Vous pouvez maintenant utiliser cet emplacement pour `HTTPD_HOME`, comme dans les autres documentations JBoss EAP 6.

4. Exécuter le script de post-installation et créer un utilisateur apache et des comptes de groupe.

Dans un émulateur de terminal, passer sur le compte utilisateur root, naviguer vers le répertoire `EAP_HOME/httpd` et exécuter la commande suivante.

```
./postinstall
```

Ensuite, vérifier si un utilisateur appelé `apache` existe sur le système en exécutant la commande suivante :

```
id apache
```

Si l'utilisateur n'existe pas, il devra être ajouté avec l'utilisateur approprié. Pour cela, exécuter la commande suivante :

```
getent group apache >/dev/null || groupadd -g 48 -r apache  
getent passwd apache >/dev/null || useradd -r -u 48 \  
-g apache -s /sbin/nologin -d HTTPD_HOME/httpd/www -c "Apache"  
apache
```

Une fois complété, si l'utilisateur `apache` compte exécuter le service `httpd`, la propriété des répertoires HTTP devra être changée pour indiquer ceci :

```
chown -R apache:apache httpd
```

Pour vérifier que les commandes ci-dessus ont été exécutées avec succès, vérifier que l'utilisateur `apache` possède une permission d'exécution pour le chemin d'installation du serveur Apache HTTP.

```
ls -l
```

Le résultat doit correspondre à cela :

```
drwxrwxr-- 11 apache apache 4096 Feb 14 06:52 httpd
```

5. Configurer le serveur Apache HTTP.

Avant de configurer le serveur Apache HTTP, configurez-le pour qu'il puisse répondre aux besoins de votre organisation. Vous pouvez utiliser la documentation disponible à partir de l'Apache Foundation à l'adresse <http://httpd.apache.org/> pour un guide général.

6. Démarrez le serveur HTTP Apache.

Démarrer le serveur Apache HTTP par la commande suivante :

```
HTTPD_HOME/httpd/sbin/apachectl start
```

7. Stopper le serveur Apache HTTP.

Pour stopper le serveur Apache HTTP, lancer la commande suivante :

```
HTTPD_HOME/httpd/sbin/apachectl stop
```

[Rapporter un bogue](#)

17.4.3. Installer le serveur Apache HTTP dans Red Hat Enterprise Linux (RHEL) 5, 6, et 7 (RPM)

Conditions préalables

- Accès root-level
- La dernière version du package `elinks` installé (requis pour la fonctionnalité `apachectl`)
- Inscrivez-vous sur les canaux Red Hat Enterprise Linux (RHEL) (pour installer le serveur Apache HTTP à partir de canaux RHEL).
- Abonnez-vous au canal `jbappplatform-6-ARCH-server-VERS-rpm` Red Hat Network (pour installer la distribution spéciale EAP du serveur Apache HTTP).

Vous pouvez installer le serveur Apache HTTP en utilisant une des méthodes suivantes :

- À partir des canaux Red Hat Enterprise Linux (RHEL) : un abonnement actif aux canaux Red Hat Enterprise Linux (RHEL) est obligatoire pour installer le serveur Apache HTTP.
- À partir du canal `jbappplatform-6-ARCH-server-VERS-rpm` (JBoss EAP specific distribution) : JBoss EAP distribue sa propre version du serveur Apache HTTP. Un abonnement actif au canal `jbappplatform-6-ARCH-server-VERS-rpm` est obligatoire pour installer la distribution spécifique de JBoss EAP du serveur Apache HTTP.

Procédure 17.4. Installer et configurer le serveur Apache HTTP dans Red Hat Enterprise Linux 5 et 6 (RPM)

1. Installer `httpd`

Pour installer la version spécifique de JBoss EAP du package `httpd`, exécutez la commande suivante :

```
yum install httpd
```

Pour installer `httpd` explicitement à partir des canaux Red Hat Enterprise Linux (RHEL), exécutez la commande suivante :

```
yum install httpd --disablerepo=jbappplatform-6-*
```



NOTE

Vous devez exécuter une des commandes ci-dessus uniquement pour installer `httpd` sur votre système.

2. Définir le comportement de démarrage du service

Vous pouvez définir le comportement de démarrage du service `httpd` en ligne de commande

ou par l'outil graphique de configuration du service. Exécutez la commande suivante pour définir le comportement :

```
chkconfig httpd on
```

Pour utiliser l'outil de configuration du service, exécutez la commande suivante et modifiez la configuration du service dans la fenêtre qui s'affiche :

```
system-config-services
```

3. Démarrer httpd

Démarrer le httpd par la commande suivante :

```
service httpd start
```

4. Stopper httpd

Stopper le httpd par la commande suivante :

```
service httpd stop
```

Procédure 17.5. Installer et configurer le serveur Apache HTTP dans Red Hat Enterprise Linux 7 (RPM)

1. Installer httpd22

Pour installer la version spécifique de JBoss EAP du package httpd22, exécutez la commande suivante :

```
yum install httpd22
```

2. Définir le comportement de démarrage du service

Exécutez la commande suivante pour démarrer le service httpd22 au démarrage :

```
systemctl enable httpd22.service
```

3. Démarrer httpd22

Démarrer le httpd22 par la commande suivante :

```
systemctl start httpd22.service
```

4. Stopper httpd22

Stopper le httpd22 par la commande suivante :

```
systemctl stop httpd22.service
```

[Rapporter un bogue](#)

17.4.4. Configuration mod_cluster sur httpd

Résumé

mod_cluster est un équilibreur de charges basé httpd. Il utilise un réseau de communication pour envoyer des requêtes de httpd vers un groupe de noeuds de serveur d'applications. Les dérivatifs suivants peuvent être définis pour configurer mod_cluster sur httpd.



NOTE

Il n'est nul besoin d'utiliser les directives ProxyPass car mod_cluster configure automatiquement les URL qui doivent être envoyés à JBossWEB.

Tableau 17.4. Dérivatifs mod_cluster

Dérivatif	Description	Valeurs
CreateBalancers	Définit comment les équilibreurs de charge sont créés dans les hôtes virtuels httpd. Cela active les directives comme : ProxyPass /balancer://mycluster1/ .	0: Créer tous les hôtes virtuels httpd 1: Ne pas créer d'équilibreurs (vous aurez besoin d'un ProxyPass ou d'un ProxyMatch au moins pour définir les noms des équilibreurs) 2: Ne créer que le serveur principal Par défaut: 2 Si vous utilisez la valeur 1, n'oubliez pas de configurer l'équilibreur dans la directive ProxyPass, car la valeur par défaut correspond à une session sticky vide. De plus nofailover=Off et les valeurs reçues via message MCMP CONFIG sont ignorées.
UseAlias	Vérifier que l'alias corresponde bien au nom du serveur.	0: Ignorer les alias 1: Vérifier les alias Par défaut : 0
LBstatusRecalTime	Intervalle d'équilibrage de charge (en secondes) de la logique pour recalculer le statut d'un nœud.	Par défaut : 5 secondes
WaitBeforeRemove	Durée en secondes avant qu'un nœud supprimé soit oublié par httpd.	Par défaut : 10 secondes

Dérivatif	Description	Valeurs
ProxyPassMatch/ProxyPass	<p>ProxyPassMatch et ProxyPass sont des directives mod_proxy qui, quand on utilise ! (à la place de l'url de back-end), évitent un proxy inverse sur le chemin d'accès. Utilisé pour autoriser httpd à fournir des informations statiques comme des images. Par exemple</p> <p>ProxyPassMatch ^(/.*\.gif)\$!</p> <p>L'exemple ci-dessus permet à httpd de servir les fichiers .gif directement.</p>	

Un nœud en hot-standby dans la logique mod_cluster est le dernier nœud vers qui toutes les demandes seront dirigées si tous les autres nœuds sont hors d'opération. Ceci est similaire à la logique hot-standby dans mod_proxy.

Pour configurer un nœud en hot-standby, remplacer le dynamic-load-provider en mod_cluster subsystem avec un simple-load-provider ayant pour facteur 0, comme par exemple :

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.2">
  <mod-cluster-config advertise-socket="modcluster" connector="ajp">
-     <dynamic-load-provider>
-       <load-metric type="busyness"/>
-     </dynamic-load-provider>
+     <simple-load-provider factor="0"/>
  </mod-cluster-config>
</subsystem>
```

Dans la console mod_cluster-manager, le nœud s'affiche avec le statut OK et une charge: 0. Pour plus d'informations, voir la section *Apache mod_cluster-manager Application* dans le guide *JBoss Enterprise Application Platform Development Guide*.

Ainsi, s'il y a trois nœuds :

- Nœud A, Charge : 10
- Nœud B, Charge : 10
- Nœud C, Charge : 10

La charge sera équilibrée entre les nœuds A et B. Si les deux nœuds ne sont pas rendus disponibles, le nœud C prendra la charge.

mod_manager

Le contexte d'une directive mod_manager est l'hôte virtuel dans tous les cas, sauf contre indication. Le contexte `server config` implique que la directive doit se trouver en dehors d'une configuration de l'hôte virtuel. Si tel n'est pas le cas, un message erreur apparaîtra et httpd ne démarrera pas.

Tableau 17.5. Dérivatifs mod_manager

Dérivatif	Description	Valeurs
EnableMCPMReceive	Autorise l'hôte virtuel à recevoir MCPM des noeuds. Inclure EnableMCPMReceive dans la configuration httpd pour permettre au mod_cluster de fonctionner. Le sauvegarder dans l'hôte virtuel de configuration d'advertise.	
MemManagerFile	Le nom de base pour les noms que mod_manager utilise pour stocker la configuration, générer des clés pour mémoire partagée ou fichiers verrouillés. Ce doit être un nom de chemin d'accès absolu ; les répertoires seront créés si nécessaire. Il est recommandé que ces fichiers soient placés sur un lecteur local et non pas en NFS share. Context: config serveur	<code>\$server_root/logs/</code>
Maxcontext	Le nombre maximum de contextes pris en charge par mod_cluster Context: config serveur	Par défaut: 100
Maxnode	Le nombre maximum de nœuds supportés par le mod_cluster. Context: config serveur	Par défaut: 20
Maxhost	Le nombre maximum d'hôtes (alias) supportés par mod_cluster. Inclut également le nombre maximum d'équilibreurs de charge. Context: config serveur	Par défaut: 20
Maxsessionid	Le nombre d'ID de sessions actives stockés afin de procurer le nombre de sessions actives du gestionnaire mod_cluster-manager. Une session est inactive quand mod_cluster ne reçoit aucune information de la session pendant 5 minutes. Context: config serveur Ce champ est à but de démonstration et de débogage uniquement.	0: la logique n'est pas activée.

Dérivatif	Description	Valeurs
MaxMCMPMaxMessSize	Taille maximum des messages MCMP en provenance d'autres directives max	Calculé sur la base d'autres directives max. Min: 1024
ManagerBalancerName	Le nom que l'équilibreur des charges utilise quand JBoss AS/JBossWeb/Tomcat ne fournit pas de nom d'équilibreur.	mycluster
PersistSlots	Indique à mod_slotmem de persister les nœuds, les alias et les contextes dans des fichiers. Context: config serveur	Off
CheckNonce	Contrôle de la vérification de la valeur unique avec le gestionnaire mod_cluster=manager.	on/off Par défaut : on - Nonce checked
AllowDisplay	Contrôle des affichages supplémentaires sur la page principale du mod_cluster-manager.	on/off Par défaut : off - seule la version sera affichée
AllowCmd	Autorise les commandes qui utilisent l'URL mod_cluster-manager.	on/off Par défaut: on - Les commandes sont autorisées
ReduceDisplay	Réduit le montant d'informations affichées sur la page principale de mod_cluster-manager, afin qu'un plus grand nombre de nœuds puissent être affichés sur la page.	on/off Par défaut : off - les informations complètes s'affichent

Dérivatif	Description	Valeurs
SetHandler mod_cluster-manager	<p>Affiche des informations sur le nœud que mod_cluster voit dans le cluster. L'information comprend des informations génériques et compte aussi le nombre de sessions actives.</p> <pre> <Location /mod_cluster- manager> SetHandler mod_cluster-manager Order deny,allow Allow from 127.0.0.1 </Location> </pre>	<p>on/off</p> <p>Par défaut : off</p>



NOTE

Quand on accède à l'emplacement défini dans httpd.conf :

Transferred : correspond aux données POST envoyées du serveur de back-end.

Connected : correspond au nombre de requêtes traitées au moment où la page de statuts du mod_cluster a été demandée.

Num_sessions : correspond au nombre de sessions que le rapport mod_cluster a reporté comme étant inactives (sur lesquelles il y a eu une requête au cours des 5 dernières minutes). Ce champ n'est pas présent quand Maxsessionid est égal à zéro. Ce champ est à but de démonstration et de débogage uniquement.

[Rapporter un bogue](#)

17.4.5. Utiliser un serveur web externe comme Web frontal pour les applications JBoss EAP 6.

Aperçu

Pour comprendre les raisons d'utiliser un service web externe comme le serveur web frontal, et pour connaître les avantages et inconvénients des différents connecteurs HTTP pris en charge par JBoss EAP 6, consulter [Section 17.2.3, « Connecteurs HTTP - Aperçu général »](#). Dans certaines situations, vous pouvez utiliser le serveur Apache HTTP de votre système d'exploitation. Sinon, vous pouvez utiliser le serveur Apache HTTP qui est fourni par JBoss Enterprise Web Server.

Une fois que vous aurez décidé quel serveur web ou quel connecteur HTTP utiliser, voir une des procédures suivantes :

- [Section 17.4.2, « Installer le serveur Apache HTTP inclus dans JBoss EAP 6 »](#)

- [Section 17.6.3, « Installer le module mod_cluster dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server \(Zip\) »](#)
- [Section 17.7.3, « Installer le module jk_mod dans un serveur Apache HTTP \(ZIP\) »](#)
- [Section 17.9.3, « Configurer Microsoft IIS pour qu'il puisse utiliser ISAPI »](#)
- [Section 17.10.2, « Configurer NSAPI dans Oracle Solaris »](#)
- [Section 17.4.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »](#)

[Rapporter un bogue](#)

17.4.6. Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes

Aperçu

JBoss EAP 6 n'a pas besoin de savoir de quel proxy elle accepte les requêtes, uniquement le port et le protocole. Ce n'est pas le cas pour `mod_cluster`, qui est davantage lié à la configuration de JBoss EAP 6. En revanche, les tâches suivantes fonctionnent pour `mod_jk`, `mod_proxy`, `ISAPI`, et `NSAPI`. Substituer les protocoles et les ports que vous aurez besoin de configurer par ceux des exemples.

Pour configurer JBoss EAP 6 pour `mod_cluster`, consulter [Section 17.6.6, « Configurer un nœud de worker de mod_cluster »](#).

Conditions préalables

- Vous devrez être connecté au Management CLI ou à la console de gestion pour effectuer cette tâche. Les étapes précises utilisent l'interface CLI, mais la même procédure de base est utilisée dans la console de gestion.
- Vous aurez besoin d'une liste des protocoles que vous devrez utiliser, que ce soit HTTP, HTTPS ou AJP.

Procédure 17.6. Modifier la configuration et ajouter les liaisons de socket

1. Configurer les propriétés de système `jvmRoute`

Par défaut, le `jvmRoute` est sur la même valeur que le nom du serveur. Si vous avez besoin de le personnaliser, vous pouvez utiliser une commande semblable à la suivante. Remplacer ou supprimer la partie de la commande `/profile=ha`, en fonction du profil que vous utilisez ou si vous utilisiez un serveur autonome. Remplacez la chaîne `CUSTOM_ROUTE_NAME` par votre nom `jvmRoute` personnalisé.

```
[user@localhost:9999 /] /profile=ha/subsystem=web:write-attribute(name="instance-id",value="CUSTOM_ROUTE_NAME")
```

2. Lister les connecteurs disponibles dans le sous-système.

**NOTE**

Cette étape est nécessaire uniquement si vous n'utilisez pas les profils **ha** ou **complète-ha** dans un serveur autonome ou un groupe de serveurs dans un domaine géré. Ces configurations incluent déjà tous les connecteurs nécessaires.

Pour qu'un service de serveur web externe puisse se connecter au serveur web de JBoss EAP 6, le sous-système web doit avoir un connecteur. Chaque protocole a besoin de son propre connecteur, lié à un groupe de sockets.

Pour avoir la liste des connecteurs actuellement disponibles, lancer la commande suivante :

```
[standalone@localhost:9999 /] /subsystem=web:read-children-
names(child-type=connector)
```

S'il n'y a aucune ligne sur le connecteur dont vous avez besoin (HTTP, HTTPS, AJP), vous devrez ajouter un connecteur.

3. Lire la configuration d'un connecteur.

Pour voir les détails de configuration d'un connecteur, vous pourrez lire sa configuration. La commande suivante lit la configuration du connecteur AJP. Les autres connecteurs ont des sorties de configuration semblables.

```
[standalone@localhost:9999 /] /subsystem=web/connector=ajp:read-
resource(recursive=true)
{
  "outcome" => "success",
  "result" => {
    "enable-lookups" => false,
    "enabled" => true,
    "max-post-size" => 2097152,
    "max-save-post-size" => 4096,
    "protocol" => "AJP/1.3",
    "redirect-port" => 8443,
    "scheme" => "http",
    "secure" => false,
    "socket-binding" => "ajp",
    "ssl" => undefined,
    "virtual-server" => undefined
  }
}
```

4. Ajouter les connecteurs utiles au sous-système web.

Pour ajouter un connecteur au sous-système web, il doit y avoir une liaison de sockets. La liaison de sockets est ajoutée au groupe de liaisons de sockets utilisées par votre serveur ou groupe de serveurs. Les étapes suivantes supposent que votre groupe de serveurs est **server-group-one** et que votre groupe de liaisons de sockets est **standard-sockets**.

a. Ajouter une liaison au groupe de liaisons de sockets.

Pour ajouter un groupe de liaison de sockets, lancer la commande suivante, en remplaçant le protocole et le port par ceux dont vous avez besoin.

```
[standalone@localhost:9999 /] /socket-binding-group=standard-sockets/socket-binding=ajp:add(port=8009)
```

b. Ajouter la liaison de socket au sous-système web.

Lancer la commande suivante pour ajouter un connecteur au sous-système web, en substituant le nom de liaison de socket et le protocole par ceux que vous avez besoin.

```
[standalone@localhost:9999 /] /subsystem=web/connector=ajp:add(socket-binding=ajp, protocol="AJP/1.3", enabled=true, scheme="http")
```

[Rapporter un bogue](#)

17.5. CLUSTERING

17.5.1. Utiliser la communication TCP dans le sous-système de clusterisation

Par défaut, les nœuds de cluster surveillent leurs statuts respectifs avec le protocole UDP. Certains réseaux ne permettent que TCP. Dans ce cas, vous pouvez ajouter la pile de protocole **TCP** à votre configuration et l'utiliser comme mécanisme par défaut. Ces options de configuration sont disponibles en ligne de commandes par l'interface CLI.

Le sous-système `mod_cluster` utilise également la communication UDP par défaut, et vous pouvez opter pour TCP également.

Voir les deux procédures suivantes pour configurer les sous-systèmes `mod_cluster` et `JGroups` pour qu'ils puissent utiliser TCP pour la communication réseau :

- [Section 17.5.2, « Configurer le sous-système JGroup pour une utilisation TCP »](#)
- [Section 17.5.3, « Désactiver les annonces dans le sous-système `mod_cluster`. »](#)

[Rapporter un bogue](#)

17.5.2. Configurer le sous-système JGroup pour une utilisation TCP

Par défaut, le sous-système `JGroups` communique à l'aide de la multidiffusion UDP. Utilisez la procédure suivante pour configurer le sous-système `JGroups` pour qu'il puisse utiliser la monodiffusion TCP à la place.

Pour configurer le sous-système `mod_cluster` pour qu'il puisse utiliser TCP également, consulter [Section 17.5.3, « Désactiver les annonces dans le sous-système `mod_cluster`. »](#).

1. Modifier le script suivant selon votre environnement.

Copier le script suivant dans un éditeur de texte. Si vous utilisez un profil différent sur un domaine géré, changer le nom du profil. Si vous utilisez un serveur autonome, supprimer la portion `/profile=full-ha` des commandes. Modifier les propriétés figurant au bas de la commande comme suit. Chacune de ces propriétés est facultative.

initial_hosts

Une liste des hôtes considérés comme connus, séparés par des virgules, sera à votre disposition pour rechercher l'adhésion de départ.

port_range

Si vous le souhaitez, vous pouvez attribuer une plage de ports. Si vous affectez une plage de ports de 2, et que le port initial est 7600, alors TCPING tentera de contacter chaque hôte sur les ports 7600-7601. Cette propriété est facultative.

timeout

Une valeur de timeout facultative, en millisecondes, pour les membres d'un cluster.

num_initial_members

Le nombre de nœuds avant qu'un cluster soit considéré comme complet. Cette propriété est facultative.

```
batch
## If tcp is already added then you can remove it ##
/profile=full-ha/subsystem=jgroups/stack=tcp:remove
/profile=full-ha/subsystem=jgroups/stack=tcp:add(transport={"type"
=>"TCP", "socket-binding" => "jgroups-tcp"})
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=TCPPING)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=MERGE2)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=FD_SOCKET, socket-binding=jgroups-tcp-fd)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-protocol(type=FD)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=VERIFY_SUSPECT)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=BARRIER)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=pbcast.NAKACK)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=UNICAST2)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=pbcast.STABLE)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=pbcast.GMS)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-protocol(type=UFC)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-protocol(type=MFC)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=FRAG2)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=RSVP)
/profile=full-ha/subsystem=jgroups:write-attribute(name=default-
stack,value=tcp)
run-batch
/profile=full-
ha/subsystem=jgroups/stack=tcp/protocol=TCPPING/property=initial_hos
ts/:add(value="HostA[7600],HostB[7600]")
/profile=full-
ha/subsystem=jgroups/stack=tcp/protocol=TCPPING/property=port_range/
:add(value=0)
/profile=full-
ha/subsystem=jgroups/stack=tcp/protocol=TCPPING/property=timeout/:ad
```

```
d(value=3000)
/profile=full-
ha/subsystem=jgroups/stack=tcp/protocol=TCPPING/property=num_initial
_members/:add(value=3)
```

2. Exécuter le script en mode de lot.



AVERTISSEMENT

Les serveurs qui exécutent le profil devront être fermés avant de pouvoir exécuter le fichier de commandes.

Dans un émulateur de terminal, naviguer vers le répertoire contenant le script `jboss-cli.sh` et saisir la commande

```
./jboss-cli.sh -c --file=SCRIPT_NAME
```

où le nom de script `SCRIPT_NAME` correspond au nom et au chemin contenant le script.

Résultat

La pile `TCPPING` est maintenant disponible pour les sous-systèmes JGroups. Si elle est utilisée, le sous-système JGroups utilisera TCP pour toute la communication de réseau. Pour configurer le sous-système `mod_cluster` à utiliser TCP également, consulter [Section 17.5.3, « Désactiver les annonces dans le sous-système `mod_cluster`. »](#).

[Rapporter un bogue](#)

17.5.3. Désactiver les annonces dans le sous-système `mod_cluster`.

Par défaut, l'équilibreur du sous-système `mod_cluster` utilise l'UDP multidiffusion pour annoncer sa présence aux workers d'arrière-plan. Si vous le souhaitez, vous pouvez désactiver les annonces. Utiliser la procédure suivante pour configurer ce comportement.

Procédure 17.7.

1. Modifier la configuration `httpd`.

Modifier la configuration `httpd` pour désactiver « server advertising » et pour utiliser un proxy à la place. La liste de proxys est configurée sur le worker, et contient tous les serveurs HTTPS activés-`mod_cluster` avec lesquels le worker peut communiquer.

La configuration de `mod_cluster` pour le serveur Web se trouve généralement dans le répertoire `/etc/httpd/` ou `etc/httpd/` au sein de l'installation `httpd`, s'il est installé dans un emplacement non standard. Reportez-vous à [Section 17.6.3, « Installer le module `mod_cluster` dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server \(Zip\) »](#) et [Section 17.6.5, « Configurer les propriétés de Server Advertisement de votre serveur web activé par votre `mod_cluster` »](#) pour plus d'informations sur le fichier lui-même. Ouvrez le fichier contenant l'hôte virtuel qui écoute les requêtes MCPM (à l'aide de la directive `EnableMCPMReceive`) et désactiver le serveur d'annonces en remplaçant la directive `ServerAdvertise` comme suit.

ServerAdvertise Off

2. Désactiver les annonces dans le sous-système `mod_cluster` de JBoss EAP 6, et fournir une liste de proxys.

Vous pouvez désactiver les annonces du sous-système `mod_cluster` et fournir une liste de proxys, en utilisant la console de gestion basée web ou l'interface CLI de lignes de commande. La liste de proxys est utile car le sous-système `mod_cluster` ne sera pas en mesure de découvrir les proxys automatiquement si les annonces sont désactivées.

o Console de gestion

Si vous utilisez un domaine géré, vous ne pourrez uniquement configurer que `mod_cluster` dans les profils où il est activé, tels que `ha` et `full-ha`.

1. Connectez-vous à la console de gestion et sélectionner **Configuration** en haut de l'écran. Si vous utilisez un domaine géré, sélectionner le profil `ha` ou `full-ha` à partir du menu déroulant **Profile** en haut et à gauche.
2. Étendre sur le menu **Subsystems**. Étendre **Web**, et sélectionner `mod_cluster`.
3. Cliquer sur **Edit** sous l'onglet **Advertising** dans `mod_cluster`. Pour désactiver la publicité, retirer la marque de la case qui se situe à côté d'**Advertise**, et cliquer sur **Save**.

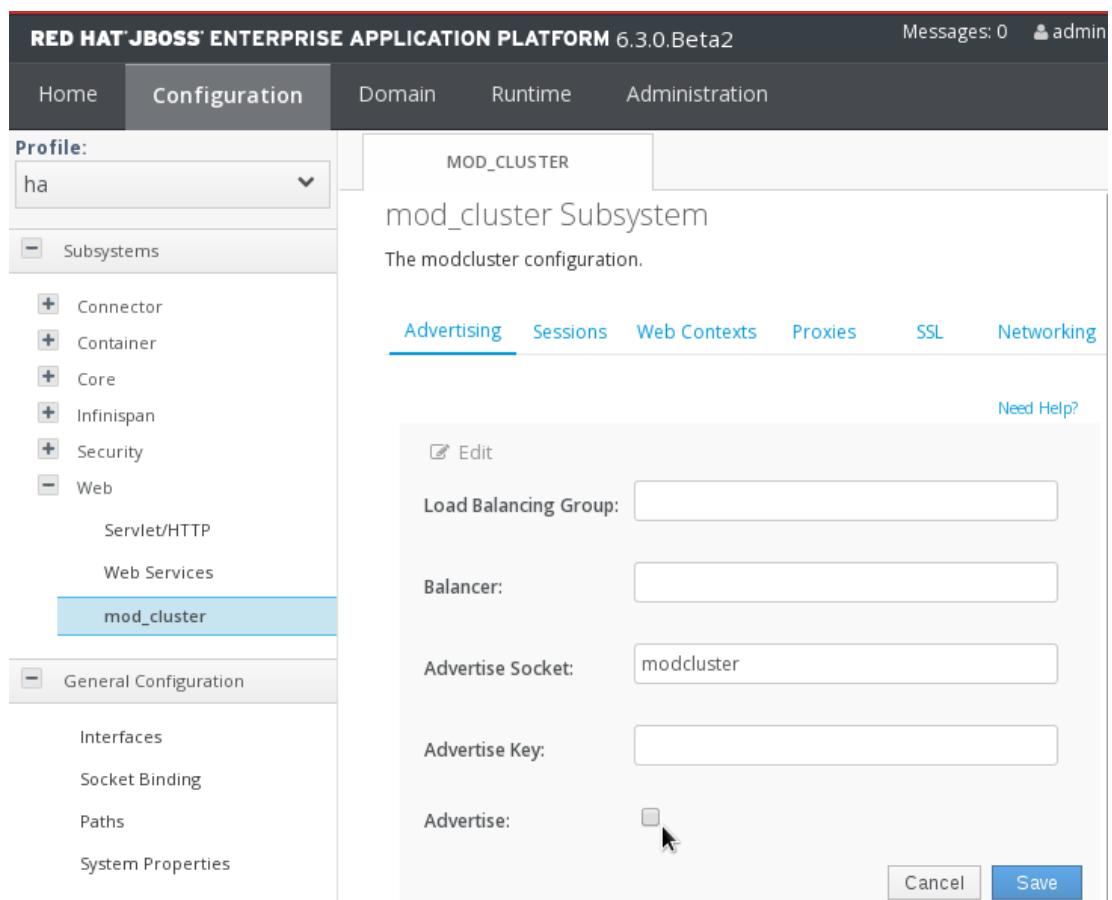


Figure 17.1. Écran de configuration de publicité `mod_cluster`

4. Cliquer sur l'onglet **Proxies**. Cliquer sur **Edit** et saisir une liste des serveurs proxy dans le champ **Proxy List**. La syntaxe qui convient est une liste séparée par des virgules de strings `HOSTNAME:PORT`, comme suit :

```
10.33.144.3:6666,10.33.144.1:6666
```

Cliquer sur le bouton **Enregistrer** pour terminer.

o Interface CLI

Les deux commandes d'interface CLI suivantes créent la même configuration que les instructions de la console de gestion ci-dessus. Elles supposent que vous exécutez un domaine géré et que votre groupe de serveurs utilise le profil **full-ha**. Si vous utilisez un profil différent, modifier son nom dans les commandes. Si vous utilisez un serveur autonome à l'aide du profil **standalone-ha** profil, supprimer la portion **/profile=full-ha** des commandes.

```
/profile=full-ha/subsystem=modcluster/mod-cluster-
config=configuration/:write-attribute(name=advertise,value=false)
```

```
/profile=full-ha/subsystem=modcluster/mod-cluster-
config=configuration/:write-attribute(name=proxy-
list,value="10.33.144.3:6666,10.33.144.1:6666")
```

Résultat

L'équilibreur httpd n'annonce plus sa présence aux nœuds de worker et la multidiffusion UDP n'est plus utilisée.



NOTE

Pour pouvoir définir l'attribut **advertise="false"**, vous devez également définir l'attribut **proxy-list="address:port"**. Si l'attribut **proxy-list** est vide, l'attribut **advertise="false"** sera ignoré. Pour désactiver le sous-système **mod_cluster subsystem**, vous pouvez le retirer de la configuration de serveur.

[Rapporter un bogue](#)

17.5.4. Passez d'UDP à TCP dans HornetQ Clustering

L'exemple suivant utilise le fichier **standalone-full-ha.xml** par défaut fourni dans EAP 6.



NOTE

Si la sécurité est activée, vous devrez définir l'attribut **cluster-password** :

```
<cluster-
password>${jboss.messaging.cluster.password:ChangeMe}</cluster-
password>
```

1. Supprimer les broadcast-groups et les discovery-groups :

```
<broadcast-groups>
  <broadcast-group name="bg-group1">
    <socket-binding>messaging-group</socket-binding>
    <broadcast-period>5000</broadcast-period>
    <connector-ref>netty</connector-ref>
```

```

    </broadcast-group>
</broadcast-groups>
<discovery-groups>
  <discovery-group name="dg-group1">
    <socket-binding>messaging-group</socket-binding>
    <refresh-timeout>10000</refresh-timeout>
  </discovery-group>
</discovery-groups>

```

2. En option, supprimer la liaison de socket de "messaging-group" :

```

<socket-binding name="messaging-group" port="0" multicast-
address="${jboss.messaging.group.address:231.7.7.7}" multicast-
port="${jboss.messaging.group.port:9876}"/>

```

3. Configurer les connecteur(s) Netty qui conviennent - un pour chacun des autres noeuds du cluster.

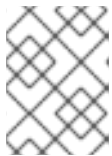
Ainsi, si le cluster a 3 noeuds, alors configurez 2 connecteurs Netty, etc. Si le cluster a 2 noeuds, alors configurez 1 connecteur Netty, etc. Voici un exemple de configuration d'un cluster 3-noeuds :

```

<netty-connector name="other-cluster-node1" socket-binding="other-
cluster-node1"/>
<netty-connector name="other-cluster-node2" socket-binding="other-
cluster-node2"/>

```

4. Configurer les liaisons de sockets correspondantes.



NOTE

La substitution de propriété système peut être utilisée soit pour "hôte", soit pour "port" selon les besoins.

```

<outbound-socket-binding name="other-cluster-node1">
  <remote-destination host="otherNodeHostName1" port="5445"/>
</outbound-socket-binding>
<outbound-socket-binding name="other-cluster-node2">
  <remote-destination host="otherNodeHostName2" port="5445"/>
</outbound-socket-binding>

```

5. Configurer la connexion au cluster pour qu'elle utilise ces connecteurs à la place du discovery-group utilisé par défaut :

```

<cluster-connection name="my-cluster">
  <address>jms</address>
  <connector-ref>netty</connector-ref>
  <static-connectors>
    <connector-ref>other-cluster-node1</connector-ref>
    <connector-ref>other-cluster-node2</connector-ref>
  </static-connectors>
</cluster-connection>

```

Ce processus devra être répété sur chaque noeud du groupement pour que chaque noeud ait des connecteurs sur chaque noeud du cluster.



NOTE

Ne pas configurer un noeud avec une connexion sur lui-même. Cela est considéré comme une erreur de configuration.

[Rapporter un bogue](#)

17.6. WEB, CONNECTEURS HTTP, ET HTTP CLUSTERING

17.6.1. Le connecteur HTTP `mod_cluster`

Le module `mod_cluster` permet l'équilibrage des charges et est connu sous le nom de `connector`. Pour en savoir plus sur les autres connecteurs, voir ce qui suit :

- [Section 17.7.1, « Le connecteur Apache `mod_jk` HTTP »](#)
- [Section 17.9.1, « Internet Server API \(ISAPI\) »](#)
- [Section 17.10.1, « Netscape Server API \(NSAPI\) »](#)

Le connecteur `mod_cluster` a plusieurs avantages par rapport aux autres connecteurs.

- Le MCMP ou *mod_cluster Management Protocol* est un lien supplémentaire entre les nœuds de serveurs de JBoss Enterprise Application Platform 6 et le serveur Apache HTTP avec le module `mod_cluster` activé. Il est utilisé par les serveurs de JBoss Enterprise Application Platform pour transmettre des facteurs d'équilibrage des charges côté serveur et des événements de cycle de vie de retour vers le serveur Apache HTTP via un ensemble personnalisé de méthodes HTTP.
- Une configuration dynamique du serveur Apache HTTP avec `mod_cluster` permet aux serveurs de JBoss EAP 6 de se joindre à l'arrangement d'équilibrage de charge sans aucune configuration manuelle.
- JBoss EAP 6 se charge des calculs de factorisation d'équilibre des charges, au lieu de s'en remettre au serveur Apache HTTP avec `mod_cluster`. Cela rend les métriques d'équilibrage des charges plus précis qu'avec les autres connecteurs.
- Le connecteur `mod_cluster` offre un contrôle précis du cycle de vie d'application. Chaque serveur JBoss EAP 6 transmet tout événement de cycle de vie du contexte d'application web au serveur Apache HTTP, l'informant de démarrer ou d'arrêter les demandes de routage dans un contexte donné. Ceci empêche les utilisateurs finaux de voir les erreurs HTTP en raison des ressources non disponibles.
- Les transports AJP, HTTP ou HTTPS peuvent être utilisés.

[Rapporter un bogue](#)

17.6.2. Configurer le sous-système `mod_cluster`

Le sous-système `mod_cluster` peut être configuré par la console de gestion ou par l'interface CLI. Dans cette section, les différentes options de configuration sont décrites, groupées telles qu'elles

apparaissent dans la console de gestion. Des exemples de commandes d'interface CLI sont données pour chaque option.



NOTE

La page de configuration de `mod_cluster` n'est visible que pour les profils `ha` et `full-ha`. Dans un domaine géré, ces profils sont `ha` et `full-ha`, et dans un domaine autonome, ces profils sont `standalone-ha` et `standalone-full-ha`.

Console de gestion

Cliquer sur l'onglet **Configuration**. Si vous configurez un domaine géré, sélectionner le bon profil à partir de **Profile**. Étendre **Subsystems**. Étendre **Web**, et sélectionner `mod_cluster`.

Tableau 17.6. Option de configuration de publicité de `mod_cluster`

Option	Description	CLI Command
Load Balancing Group	Si non nulles, les requêtes devront être envoyées vers un groupe d'équilibrage de charges sur l'équilibreur de charges. Laisser cet espace vide si vous ne souhaitez pas utiliser ces groupes d'équilibrage des charges.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=load-balancing-group,value=myGroup)</pre>
Balancer	Cet attribut spécifie quel équilibreur <code>mod_proxy</code> doit être configuré automatiquement par <code>mod_cluster</code> sur le serveur HTTP Apache. La valeur par défaut est none , dans lequel cas, la valeur par défaut de <code>mon_cluster</code> sera utilisée (balancer://mon-cluster/ si exprimée en termes de <code>mod_proxy</code>). Cette valeur par défaut est configurée côté serveur HTTP Apache avec la directive <code>ManagerBalancerName</code> Si vous utilisez deux valeurs d'attributs différents de balancer sur les instances de worker de JBoss EAP 6, il y aura deux équilibreurs de <code>mod_proxy</code> différents créés par <code>mod_cluster</code> automatiquement sur le serveur HTTP Apache.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=balancer,value=myBalancer)</pre>

Option	Description	CLI Command
Socket Advertise	Le nom de la liaison de sockets à utiliser pour les annonces de cluster.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise-socket,value=modcluster)</pre>
Advertise Security Key	Une chaîne contenant la clé de sécurité à annoncer.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise-security-key,value=myKey)</pre>
Advertise	Indique si les annonces sont activées. Valeur par défaut true .	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise,value=true)</pre>

Tableau 17.7. Options de configuration de session mod_cluster

Option	Description	CLI Command
Sticky Session	Si vous souhaitez utiliser des sessions pour les demandes. Cela signifie qu'après que le client ait établi une connexion sur un nœud spécifique, leur transmission ultérieure est routée vers ce même nœud à moins qu'il ne soit plus disponible. La valeur par défaut est true , qui est le paramètre recommandé.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=sticky-session,value=true)</pre>

Option	Description	CLI Command
Sticky Session Force	Si sur true , la demande ne sera pas redirigée vers un nouveau nœud si son nœud initial n'est plus disponible. Au lieu de cela, elle échouera. La valeur par défaut est false .	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=sticky-session-force,value=false)</pre>
Sticky Session Remove	Supprime les informations de la session après le basculement. Cela a comme valeur par défaut false .	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=sticky-session-remove,value=false)</pre>

Tableau 17.8. Options de configuration de contexte web `mod_cluster`

Option	Description	CLI Command
Auto Enable Contexts	Si on doit ajouter de nouveaux contextes à <code>mod_cluster</code> par défaut ou non. La valeur par défaut true . Si vous modifiez la valeur par défaut et que vous devez activer le contexte manuellement, l'application web peut activer son contexte à l'aide de la méthode MBean <code>enable()</code> , ou via le gestionnaire <code>mod_cluster</code> , une application web qui s'exécute sur le serveur proxy httpd, sur un hôte virtuel nommé ou le port qui est spécifié dans la configuration de cet httpd.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=auto-enable-contexts,value=true)</pre>
Excluded Contexts	Une liste séparée par des virgules de contextes que <code>mod_cluster</code> doit ignorer. Si aucun hôte n'est indiqué, l'hôte est censé être <code>localhost</code> . <code>ROOT</code> indique le contexte racine de l'application web. La valeur par défaut est <code>ROOT, invoker, jbossws, juddi, console</code> .	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=excluded-contexts,value="ROOT, invoker, jbossws, juddi, console")</pre>

Tableau 17.9. Options de configuration proxy de `mod_cluster`

Option	Description	CLI Command
Proxy URL	Si définie, cette valeur sera ajoutée à l'URL des commandes MCMP.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=proxy-url,value=myhost)</pre>
Proxy List	Une liste séparée par des virgules des adresses proxy httpd, dans le format hostname:port . Ceci indique la liste des serveurs proxy avec lesquels le processus de <code>mod_cluster</code> va tenter de communiquer au départ.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=proxy-list,value="127.0.0.1,127.0.0.2")</pre>


Configurer la communication SSL pour `mod_cluster`

Par défaut, la communication `mod_cluster` a lieu sur un lien HTTP crypté. Si vous définissez le schéma du connecteur à HTTPS (voir [Tableau 17.7, « Options de configuration de session `mod_cluster` »](#)), les paramètres ci-dessous indiquent à `mod_cluster` où trouver les informations pour encoder la connexion.

Tableau 17.10. Options de configuration SSL de `mod_cluster`

Option	Description	CLI Command
Key Alias	Clé alias choisie quand le certificat est créé.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=key-alias,value=jboss)</pre>
Password	Ce mot de passe est le mot de passe de keystore pour les deux keystores: <code>certificate-key-file</code> (Key File) et <code>ca-certificate-file</code> (Cert File) et l'entrée <code>key/certificate</code> spécifiée avec Key Alias dans Cert File.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=password,value=changeit)</pre>

Option	Description	CLI Command
CA Cert File/Trust Store	Trust store utilisé pour valider le certificat de serveur web.	<pre data-bbox="1034 255 1423 577">/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=ca-certificate-file,value=\${user.home}/jboss.crt)</pre>
Key Store	Keystore contenant le certificat et la clé privée qui identifient cette instance.	<pre data-bbox="1034 678 1423 1001">/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=certificate-key-file,value=\${user.home}/.keystore)</pre>
Cipher Suite	La suite cipher d'encodage autorisée.	<pre data-bbox="1034 1102 1423 1348">/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=cipher-suite,value=ALL)</pre>
Revocation URL	L'URL de la liste de révocation de l'autorité de certificat	<pre data-bbox="1034 1451 1423 1742">/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=ca-revocation-url,value=jboss.crl)</pre>

Option	Description	CLI Command
Protocol	<p>Les protocoles SSL activés.</p> <p>Vous pouvez également spécifier une combinaison de protocoles, séparés par des virgules. Par exemple, TLSv1, TLSv1.1,TLSv1.2.</p> <div style="background-color: #fff9c4; padding: 10px; border: 1px solid #ccc;">  <p>AVERTISSEMENT</p> <p>Red Hat recommande que vous désactiviez SSL explicitement en faveur de TLSv1.1 ou TLSv1.2 dans tous les packages affectés.</p> </div>	<pre>/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=protocol,value="TLSv1,TLSv1.1,TLSv1.2")</pre>

Configurer les options de réseautage de mod_cluster

Les options de réseautage de `mod_cluster` contrôlent des comportements de timeout différents pour des types de services variés avec lesquels le service `mod_cluster` communique.

Tableau 17.11. Options de configuration de réseautage de `mod_cluster`

Option	Description	CLI Command
--------	-------------	-------------

Option	Description	CLI Command
Node Timeout	<p>Timeout (en secondes) des connexions de proxy vers un worker. C'est que le temps mod_cluster attendra la réponse de dorsal avant de retourner l'erreur. Qui correspond au délai d'attente dans la documentation de mod_proxy de worker. Une valeur correspondant à -1 n'indique aucun délai d'attente. Notez que mod_cluster utilise toujours un cping/cpong avant d'adresser une demande et la valeur connectiontimeout utilisée par mod_cluster est la valeur de ping.</p>	<pre data-bbox="1034 255 1423 465">/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=node-timeout,value=-1)</pre>
Socket Timeout	<p>Nombre de millisecondes durant lesquelles patienter avant d'obtenir une réponse d'un proxy httpd suite à des commandes MCMP avant le timeout, et indiquer erreur de proxy.</p>	<pre data-bbox="1034 846 1423 1057">/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=socket-timeout,value=20)</pre>
Stop Context Timeout	<p>Durée, mesurée dans les unités spécifiées par stopContextTimeoutUnit, pendant laquelle attendre l'arrêt net d'un contexte (fin des demandes en attente pour un contexte distribuable; ou destruction/expiration des sessions actives pour un contexte non distribuable).</p>	<pre data-bbox="1034 1160 1423 1406">/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=stop-context-timeout,value=10)</pre>

Option	Description	CLI Command
Session Draining Strategy	<p>Indique si on doit drainer les sessions avant de retirer le déploiement d'une application web.</p> <p>DEFAULT</p> <p>Sessions de drainage avant qu'une application web retire son déploiement si l'application web n'est pas distribuable.</p> <p>ALWAYS</p> <p>Toujours drainer les sessions avant le retrait du déploiement d'une application web, même pour les applications web distribuables.</p> <p>NEVER</p> <p>Ne pas drainer les sessions avant le retrait du déploiement d'une application web, même pour les applications web non distribuables.</p>	<pre data-bbox="1034 255 1422 539">/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=session-draining-strategy,value=DEFAULT)</pre>
Max Attempts	<p>Nombre de fois qu'un proxy httpd va tenter d'envoyer une requête donnée à un noeud avant d'abandonner. La valeur minimale est 1, ce qui signifie essayer une seule fois. La valeur par défaut du module mod_proxy est également 1, ce qui signifie qu'aucune nouvelle tentative ne se produit.</p>	<pre data-bbox="1034 1236 1422 1447">/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=max-attempts,value=1)</pre>
Flush Packets	<p>Indique si on doit activer le vidage des paquets dans le serveur web. Valeur par défaut false.</p>	<pre data-bbox="1034 1617 1422 1861">/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=flush-packets,value=false)</pre>

Option	Description	CLI Command
Flush Wait	Durée, en secondes, pendant laquelle on doit attendre le vidage des paquets dans le serveur web. -1 est la valeur par défaut. Une valeur correspondant à -1 indique une attente indéfinie avant de vider les paquets.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=flush-wait,value=-1)</pre>
Ping	Durée, en secondes, pendant laquelle attendre un réponse au ping d'un worker. Valeur par défaut 10 secondes.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=ping,value=10)</pre>
SMAX	Le nombre de soft connexions inactives maximales (le même que smax dans la documentation de mod_proxy). La valeur maximale dépend de la configuration de thread httpd et peut être ThreadsPerChild ou 1 .	<pre>profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=smax,value=ThreadsPerChild)</pre>
TTL	Time To live (en secondes) pour les connexions inactives au dessus de smax , la valeur par défaut est 60 Quand nodeTimeout n'est pas défini, le Proxy de la directive ProxyTimeout est utilisé. Si ProxyTimeout n'est pas défini, alors le Timeout sera utilisé. La valeur par défaut est de 300 secondes. nodeTimeout , ProxyTimeout , et Timeout sont définis au niveau socket.	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=ttl,value=-1)</pre>
Node Timeout	La durée d'attente, en secondes, pour le traitement d'une requête par un processus de serveur web externe. Par défaut, -1 , ce qui signifie que mod_cluster attend indéfiniment la requête traitée par le worker httpd .	<pre>/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=node-timeout,value=-1)</pre>

Options de configuration de load provider de **mod_cluster**

Les options de configuration de `mod_cluster` suivantes ne sont pas disponibles dans la console de gestion, et peuvent être uniquement définies en utilisant l'interface CLI en ligne de commandes.

Un simple fournisseur de charge est utilisé si aucun processeur de charge dynamique n'est présent. Il donne à chaque membre du cluster un facteur de charge 1, et répartit uniformément les travaux sans prendre en compte un algorithme d'équilibrage de charges. Pour l'ajouter, utilisez la commande CLI suivante :

```
[standalone@localhost:9990 /] /subsystem=modcluster/mod-cluster-
config=configuration/:write-attribute(name=simple-load-provider, value=1)
```

Un fournisseur de charge dynamique peut être configuré pour utiliser une variété d'algorithmes, en combinaison, pour déterminer quel worker recevra la demande suivante. Vous pouvez créer un fournisseur de charge et le configurer pour qu'il soit adapté à votre environnement. et vous pourrez avoir plus d'un métrique de charge active en même temps, en les ajoutant par l'interface CLI. Le fournisseur de charge dynamique par défaut utilise `busyness` comme métrique de charge déterminant. Les options de fournisseur de charge dynamique et les métriques de charge possible se trouvent ci-dessous.

Tableau 17.12. Options de load provider dynamic de `mod_cluster`

Option	Description	CLI Command
Decay	Le facteur par lequel les métriques historiques se désintègrent de façon significative.	
History	Le nombre d'enregistrements de métriques de charge historique à considérer pour déterminer la charge.	

Option	Description	CLI Command
Load Metric	Le métrique de charge par défaut inclue avec le fournisseur de charge dynamique dans JBoss EAP 6 est busyness , qui calcule la charge du worker en fonction de la quantité de threads dans le pool de threads devant servir les requêtes. Vous pouvez définir la capacité de ce métrique par lequel la charge réelle est divisée : charge calculée / capacité. Vous pouvez définir plusieurs paramètres de charge dans le fournisseur de la charge dynamique.	<pre data-bbox="1034 255 1423 613">/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=busyness/:write-attribute(name=capacity,value=1.0)</pre> <pre data-bbox="1034 667 1423 1025">/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=busyness/:write-attribute(name=type,value=busyness)</pre> <pre data-bbox="1034 1079 1423 1438">/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=busyness/:write-attribute(name=weight,value=1)</pre>

Load Metric Algorithms

cpu

Le métrique de charge **cpu** utilise la charge CPU moyenne pour déterminer quel nœud reçoit la charge de travail suivante.

mem

Le métrique de charge **mem** utilise la mémoire native RAM comme facteur de charge. L'utilisation de ce métrique est déconseillée car elle fournit une valeur qui inclut les tampons et le cache. C'est donc toujours un chiffre très faible sur chaque système décent pourvu d'une bonne gestion de mémoire.

heap

Le métrique de charge **heap** utilise l'usage heap (de tas) pour déterminer quel worker reçoit la charge de travail suivante.

sessions

Le métrique de charge de **session** utilise le nombre de sessions actives comme métrique.

requests

Le métrique de charge **requests** utilise le nombre de requêtes en provenance des clients pour déterminer quel worker reçoit la charge de travail suivante. Par exemple, capacité 1000 signifie que 1000 requêtes/sec est considéré comme une « pleine charge ».

send-traffic

Le métrique de charge **send-traffic** (trafic envoyé) utilise le volume de trafic envoyé à partir d'un worker vers les clients. Par ex. une capacité par défaut de 512 indique que le nœud doit être considéré en pleine charge, si le trafic sortant moyen est 512 KB/s ou supérieur.

receive-traffic

Le métrique de charge **receive-traffic** (réception de trafic) utilise le volume de trafic envoyé vers le worker en provenance des clients. Par ex. une capacité par défaut de 1024 indique que le worker doit être considéré en pleine charge, si le trafic entrant moyen est 1024 KB/s ou supérieur.

busyness

Ce métrique représente le nombre de threads d'un pool de threads en train de répondre à des requêtes.

Exemple 17.1. Ajouter un métrique de charge

Pour ajouter un métrique de charge, utiliser la commande **add-metric**.

```
/subsystem=modcluster/mod-cluster-config=configuration/:add-metric(type=sessions)
```

Exemple 17.2. Définir une valeur de métrique existant

Pour définir la valeur d'un métrique existant, utiliser la commande **write-attribute**.

```
/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=cpu/:write-attribute(name="weight",value="3")
```

Exemple 17.3. Modifier une valeur de métrique existant

Pour changer la valeur d'un métrique existant, utiliser la commande **write-attribute**.

```
/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=cpu/:write-attribute(name="type",value="busyness")
```


Exemple 17.4. Supprimer un métrique existant

Pour supprimer un métrique existant, utiliser la commande `remove-metric`.

```
/subsystem=modcluster/mod-cluster-config=configuration/:remove-metric(type=sessions)
```

[Rapporter un bogue](#)

17.6.3. Installer le module mod cluster dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server (Zip)

Conditions préalables

- Pour cette tâche, vous devrez utiliser un serveur Apache HTTP installé dans Red Hat Enterprise Linux 6, ou JBoss Enterprise Web Server, ou encore un serveur Apache HTTP autonome comme composant de JBoss EAP 6 à télécharger séparément.
- Si vous avez besoin d'installer un serveur Apache HTTP dans Red Hat Enterprise Linux 6, utiliser les instructions dans *Red Hat Enterprise Linux 6 Deployment Guide*
- Si vous avez besoin d'installer un serveur Apache HTTP autonome en tant que composant téléchargeable de JBoss EAP 6, consulter [Section 17.4.2, « Installer le serveur Apache HTTP inclus dans JBoss EAP 6 »](#).
- Si vous avez besoin d'installer JBoss Enterprise Web Server, utiliser les instructions dans *JBoss Enterprise Web Server Installation Guide*
- Télécharger le package **Webserver Connecter Natives** correspondant à votre système d'exploitation et architecture depuis le portail client de Red Hat à <https://access.redhat.com>. Ce paquet contient les modules HTTPD mod_cluster binaires précompilés pour votre système d'exploitation. Après avoir extrait l'archive, les modules se trouvent dans le répertoire `EAP_HOME/modules/system/layers/base/native/lib/httpd/modules`.

Le répertoire `etc/` contient quelques exemples de fichiers de configuration et le répertoire `share/` contient une documentation supplémentaire.

- Vous devez être connectés avec des privilèges administratifs (root).



NOTE

Si vous utilisez un système 64 bit, les modules binaires de serveur web de mod_cluster se trouveront ici :

`EAP_HOME/modules/system/layers/base/native/lib64/httpd/modules`.

Vous devrez utiliser ce chemin si vous devez accéder aux modules.

Procédure 17.8. Installer le Module mod cluster

1. **Déterminer l'emplacement de votre configuration de serveur Apache HTTP.**
Votre emplacement de serveur Apache HTTP sera différent selon que vous utilisez Apache HTTP de Red Hat Enterprise Linux, le serveur Apache HTTP autonome inclus comme

composant séparé téléchargeable dans JBoss EAP 6 ou le serveur Apache HTTP disponible dans JBoss Enterprise Web Server. C'est l'une des trois options suivantes qui sera mentionnée au cours de cette tâche sous le nom *HTTPD_HOME*.

- o Apache HTTP Server - */etc/httpd/*
 - o JBoss EAP 6 HTTP Server - cet emplacement est choisi par vous-même sur la base des exigences de votre infrastructure.
 - o JBoss Enterprise Web Server Apache HTTP Server - *EWS_HOME/httpd/*
2. **Copier les modules dans le répertoire de modules d'Apache HTTP Server.**
Copier les quatre modules (les fichiers qui se terminent par *.so*) à partir du répertoire *EAP_HOME/modules/system/layers/base/native/lib/httpd/modules* de l'archive extraite Webserver Natives vers le répertoire *HTTPD_MODULES*.
 3. **Pour JBoss Enterprise Web Server, désactiver le module *mod_proxy_balancer*.**
Si vous utilisez JBoss Enterprise Web Server, le module *mod_proxy_balancer* sera activé par défaut. Il est incompatible avec *mod_cluster*. Pour le désactiver, modifier *HTTPD_CONF/httpd.conf* et décommenter la ligne suivante en mettant le symbole *#* (hachage) devant la ligne qui charge le module. La ligne apparaîtra sans le commentaire, puis avec, comme ci-dessous.

```
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
```

```
# LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
```

Sauvegarder et fermer le fichier.

4. **Configurer le module *mod_cluster*.**
L'archive Webserver Natives contient un échantillon de fichier *mod_cluster.conf* (*EAP_HOME/modules/system/layers/base/native/etc/httpd/conf*). Ce fichier peut être utilisé comme guide ou copié et modifié pour créer un fichier *HTTPD_CONF.DJBoss_HTTP.conf*.



NOTE

L'utilisation du nom *JBoss_HTTP.conf* est une convention arbitraire de ce document. le fichier de configuration sera chargé, indépendamment de son nom, s'il est enregistré dans le répertoire *conf.d/* avec l'extension *.conf*.

Ajouter l'entrée suivante à votre fichier de configuration :

```
LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module modules/mod_advertise.so
```

Cela oblige le serveur Apache HTTP à charger les modules dont *mod_cluster* a besoin automatiquement pour fonctionner.

5. **Créer un proxy de listener de serveur.**

Continuer à éditer `HTTPD_CONF.D/ht tpd/JBoss_HTTP.conf` et ajouter la configuration minimale suivante, en remplaçant les valeurs en lettres majuscules par des valeurs adaptées à votre environnement.

```
Listen IP_ADDRESS:PORT
<VirtualHost IP_ADDRESS:PORT>
  <Location />
    Order deny,allow
    Deny from all
    Allow from *.MYDOMAIN.COM
  </Location>

  KeepAliveTimeout 60
  MaxKeepAliveRequests 0
  EnableMCPMReceive

  ManagerBalancerName mycluster
  ServerAdvertise On

</VirtualHost>
```

Ces directives créent un nouveau serveur virtuel qui écoute sur le port `IP_ADDRESS:PORT`, permet des connexions de `MYDOMAIN.COM` et se présente comme un équilibreur de charge du nom `mycluster`. Ces directives sont traitées en détail dans la documentation pour le serveur Web Apache Server. Pour en savoir plus sur les directives `ServerAdvertise` et `EnableMCPMReceive` ou les implications des annonces de serveur, consulter [Section 17.6.5, « Configurer les propriétés de Server Advertisement de votre serveur web activé par votre mod_cluster »](#).

Enregistrer le fichier et sortir.

6. Démarrer à nouveau le serveur HTTP Apache.

La façon de redémarrer le serveur Apache HTTP dépend de savoir si vous utilisez le serveur Apache HTTP de Red Hat Enterprise Linux ou le serveur HTTP inclus dans JBoss Enterprise Web Server. Choisir une des deux méthodes ci-dessous.

- **Serveur Apache HTTP de Red Hat Enterprise Linux 6**

Exécuter la commande suivante :

```
[root@host]# service httpd restart
```

- **JBoss Enterprise Web Server HTTP Server**

JBoss Enterprise Web Server exécute à la fois sur Red Hat Enterprise Linux et Microsoft Windows Server. La méthode de redémarrage du serveur Apache HTTP est différente pour chacun.

- **Red Hat Enterprise Linux**

Dans Red Hat Enterprise Linux, JBoss Enterprise Web Server installe son serveur Apache HTTP en tant que service. Pour redémarrer le serveur Apache HTTP, lancer les deux commandes suivantes :

```
[root@host ~]# service httpd stop
[root@host ~]# service httpd start
```

■ Microsoft Windows Server

Lancer les commandes suivantes dans une invite de commande avec des privilèges administratifs :

```
C:\> net stop httpd
C:\> net start httpd
```

Résultat

Le serveur Apache HTTP est maintenant configuré comme équilibreur de charges, et peut fonctionner avec le sous-système `mod_cluster` qui exécute sur JBoss EAP 6. Pour configurer JBoss EAP 6 pour qu'il soit au fait de `mod_cluster`, consulter [Section 17.6.6, « Configurer un nœud de worker de `mod_cluster` »](#).

[Rapporter un bogue](#)

17.6.4. Installer le module `mod_cluster` dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server (RPM)

Conditions préalables

- Pour cette tâche, vous devrez utiliser le serveur Apache HTTP installé dans Red Hat Enterprise Linux 6, ou JBoss Enterprise Web Server, ou encore un serveur Apache HTTP autonome comme composant de JBoss EAP 6 à télécharger séparément.
- Si vous avez besoin d'installer un serveur Apache HTTP dans Red Hat Enterprise Linux 6, utiliser les instructions dans *Red Hat Enterprise Linux 6 Deployment Guide*
- Si vous avez besoin d'installer un serveur Apache HTTP autonome en tant que composant téléchargeable de JBoss EAP 6, consulter [Section 17.4.2, « Installer le serveur Apache HTTP inclus dans JBoss EAP 6 »](#).
- Si vous avez besoin d'installer JBoss Enterprise Web Server, utiliser les instructions dans *JBoss Enterprise Web Server Installation Guide*
- Vous devez être connectés avec des privilèges administratifs (root).
- Vous devez posséder un abonnement actif au canal RHN `jbappplat form-6-ARCH-server-VERS-rpm`.

La méthode d'installation RPM reste la même sur Red Hat Enterprise Linux 5 et 6 et ne requiert que de mineures modifications pour les utilisateurs de Red Hat Enterprise Linux 6 ayant installé un serveur Apache HTTP 2.2.15.

1. Installer le package `mod_cluster-native` avec YUM :

```
yum install mod_cluster-native
```

2. Apache HTTP Server:

- Si vous choisissez de rester sur un serveur HTTP 2.2.15, vous devez désactiver le module `mod_proxy_balancer` chargé par défaut en commentant la ligne `LoadModule proxy_balancer_module` dans le fichier `httpd.conf`.

Vous pouvez modifier le fichier manuellement ou utiliser la commande suivante :

```
sed -i 's/^LoadModule proxy_balancer_module/#LoadModule
proxy_balancer_module;/s/$//' /etc/httpd/conf/httpd.conf
```

- o Si vous choisissez d'effectuer une mise à niveau vers le serveur Apache HTTP 2.2.26, installer la dernière version par la commande suivante.

```
yum install httpd
```

3. Pour que le service du serveur Apache HTTP démarre automatiquement, saisir la commande suivante :

- o Pour Red Hat Enterprise Linux 5 et 6 :

```
service httpd add
```

- o Pour Red Hat Enterprise Linux 7 :

```
systemctl enable httpd22.service
```

4. Démarrer l'équilibreur `mod_cluster` par la commande suivante :

- o Pour Red Hat Enterprise Linux 5 et 6 :

```
service httpd start
```

- o Pour Red Hat Enterprise Linux 7 :

```
systemctl start httpd22.service
```

[Rapporter un bogue](#)

17.6.5. Configurer les propriétés de Server Advertisement de votre serveur web activé par votre `mod_cluster`

Résumé

Pour obtenir des instructions sur la façon de configurer votre serveur web pour qu'il interagisse avec l'équilibreur de charges, consulter [Section 17.6.3, « Installer le module `mod_cluster` dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server \(Zip\) »](#). L'élément de configuration `server advertisement` requiert davantage d'explications.

Quand Server Advertisement est inactif, le serveur web envoie des messages qui contiennent l'adresse IP et le numéro de port spécifié dans l'hôte virtuel du `mod_cluster`. Pour configurer ces valeurs, consulter [Section 17.6.3, « Installer le module `mod_cluster` dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server \(Zip\) »](#). Si UDP multicast n'est pas disponible sur votre réseau, ou si vous préférez configurer les worker avec une liste statique de serveurs proxy, vous pouvez désactiver Server Advertisement et configurer les noeuds proxy manuellement. Voir [Section 17.6.6, « Configurer un nœud de worker de `mod_cluster` »](#) pour obtenir des informations sur la façon de configurer un worker.

Vous devrez modifier le `httpd.conf` associé à votre instance de serveur Apache HTTP. Il s'agit le plus souvent de `/etc/httpd/conf/httpd.conf` dans Red Hat Enterprise Linux, ou peut-être du répertoire `etc/` de votre instance HTTP Apache autonome.

Procédure 17.9. Modifier le fichier httpd.conf et implémenter les changements

1. Désactivez le paramètre `AdvertiseFrequency`, s'il existe.

Si vous voyez une ligne qui ressemble à ceci dans votre énoncé `<VirtualHost>`, décommenter cette ligne en ajoutant un signe `#` (hachage) avant le premier caractère. La valeur ne devra pas correspondre à 5.

```
AdvertiseFrequency 5
```

2. Ajouter la directive qui permet de désactiver Server Advertisement.

Ajouter la directive suivante dans l'énoncé `<VirtualHost>` afin de désactiver Server Advertisement.

```
ServerAdvertise Off
```

3. Actionner la possibilité de recevoir des messages MCPM.

Ajouter la directive suivante pour permettre au serveur web de recevoir des messages MCPM de la part des worker nodes.

```
EnableMCPMReceive
```

4. Redémarrer le serveur web.

Redémarrer le serveur web en lançant une des commandes suivantes, selon que vous utilisez Red Hat Enterprise Linux ou Microsoft Windows Server.

o Red Hat Enterprise Linux

```
[root@host ]# service httpd restart
```

o Microsoft Windows Server

```
C:\> net service http
C:\> net service httpd start
```

Résultat

Le serveur web n'annonce plus l'adresse IP et le port de votre serveur proxy `mod_cluster`. Pour l'annoncer, vous devez configurer vos worker nodes pour qu'ils utilisent une adresse statique et un port pour communiquer avec le proxy. Consulter [Section 17.6.6, « Configurer un nœud de worker de `mod_cluster` »](#) pour plus de détails.

[Rapporter un bogue](#)

17.6.6. Configurer un nœud de worker de `mod_cluster`

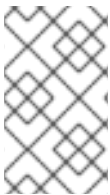
Résumé

Un nœud de worker de `mod_cluster` se compose d'un serveur JBoss EAP 6. Ce serveur peut faire partie d'un groupe de serveurs dans un domaine géré ou un serveur autonome. Un processus distinct s'exécute sur JBoss EAP 6, qui gère tous les nœuds de workers du cluster. C'est ce qu'on appelle le master. Pour plus d'informations conceptuelles sur les nœuds, consulter [Section 17.2.4, « Types de nœuds »](#). Pour une vue d'ensemble de l'équilibrage de la charge du serveur web, consulter [Section 17.2.3, « Connecteurs HTTP - Aperçu général »](#).

Les nœuds de worker d'un domaine géré partagent une configuration identique à travers un groupe de serveurs. Les nœuds de workers exécutant comme serveurs autonomes sont configurés individuellement. Les étapes de configuration, sinon, sont identiques.

Configuration d'un nœud de worker

- Un serveur autonome devra démarrer avec le profil `standalone-ha` ou `standalone-full-ha`.
- Un groupe de serveurs de domaine géré devra utiliser le profil `ha` ou `full-ha`, et le groupe de liaisons de sockets `ha-sockets` ou `full-ha-sockets`. JBoss EAP 6 est fourni avec un groupe de serveurs à clusterisation activée, nommé `other-server-group` qui remplit ces prérequis.



NOTE

Quand vous avez des commandes d'interface CLI, celles-ci présument que vous utilisez un domaine géré. Si vous utilisez un serveur autonome, supprimez la portion `/profile=full-ha` des commandes.

Procédure 17.10. Configurez un nœud de worker

1. Configurez les interfaces de réseau

Les interfaces de réseau ont toutes la valeur `127.0.0.1` par défaut. Chaque hôte physique, qui accueille un serveur autonome ou bien un ou plusieurs serveurs au sein d'un groupe de serveurs, a besoin d'interfaces configurées pour utiliser son adresse IP, que les autres serveurs peuvent apercevoir.

Pour changer l'adresse IP d'un hôte de JBoss EAP 6, vous devrez le fermer et modifier son fichier de configuration directement. C'est parce que l'API de gestion qui actionne la console de gestion et le CLI dépendent d'une adresse de gestion stable.

Suivez ces étapes pour changer l'adresse IP sur chaque serveur de votre cluster par votre adresse IP publique de master.

- Démarrez le serveur JBoss EAP en utilisant le profil décrit plus haut dans cette section.
- Lancez la Management CLI, avec la commande `EAP_HOME/bin/jboss-cli.sh` dans Linux ou bien la commande `EAP_HOME\bin\jboss-cli.bat` dans le serveur Microsoft Windows. Saisir la commande `connect` pour connecter le contrôleur de domaine sur l'hôte local, ou `connect IP_ADDRESS` pour vous connecter à un contrôleur de domaines sur un serveur éloigné.
- Modifier l'adresse IP externe des interfaces `management`, `public` et `unsecure` en saisissant les commandes suivantes. Veillez bien à remplacer `EXTERNAL_IP_ADDRESS` qui se trouve dans la commande par l'adresse IP externe de l'hôte.

```
/interface=management:write-attribute(name=inet-
address,value="{jboss.bind.address.management:EXTERNAL_IP_ADDRES
S}"
/interface=public:write-attribute(name=inet-
address,value="{jboss.bind.address.public:EXTERNAL_IP_ADDRESS}"
/interface=unsecure:write-attribute(name=inet-
address,value="{jboss.bind.address.unsecure:EXTERNAL_IP_ADDRESS}
"
```

```
┆ :reload
```

Vous devriez voir le résultat suivant pour chaque commande.

```
┆ "outcome" => "success"
```

- d. Pour les hôtes qui participent à un domaine géré, mais qui ne sont pas master, vous devez modifier le nom d'hôte du **master** par un nom unique. Ce nom devra être unique (parmi les noms d'esclave) et sera utilisé par l'esclave pour s'authentifier au cluster, donc notez bien le nom que vous utilisez.

- i. Démarrer l'hôte esclave JBoss EAP à l'aide de la syntaxe suivante :

```
┆ bin/domain.sh --host-config=HOST_SLAVE_XML_FILE_NAME
```

Par exemple :

```
┆ bin/domain.sh --host-config=host-slave01.xml
```

- ii. Lancer l'interface CLI.

- iii. Utiliser la syntaxe suivante pour remplacer le nom d'hôte :

```
┆ /host=master:write-attribute(name="name", value=UNIQUE_HOST_SLAVE_NAME)
```

Par exemple :

```
┆ /host=master:write-attribute(name="name", value="host-slave01")
```

Vous devriez voir apparaître le résultat suivant.

```
┆ "outcome" => "success"
```

Cela modifie le XML du fichier **host-slave01.xml** comme suit :

```
┆ <host name="host-slave01" xmlns="urn:jboss:domain:1.6">
```

- e. Pour les hôtes nouvellement configurés qui ont besoin de rejoindre un domaine géré, cherchez l'élément **local** et ajoutez l'attribut **host** de l'élément **remote** qui pointe en direction du contrôleur de domaine. Cette étape ne s'applique pas à un serveur autonome.

- i. Démarrer l'hôte esclave JBoss EAP à l'aide de la syntaxe suivante :

```
┆ bin/domain.sh --host-config=HOST_SLAVE_XML_FILE_NAME
```

Par exemple :

```
┆ bin/domain.sh --host-config=host-slave01.xml
```

- ii. Lancer l'interface CLI.

- iii. Utilisez la syntaxe suivante en spécifiant le contrôleur de domaine :


```
/host=UNIQUE_HOST_SLAVE_NAME/:write-remote-domain-
controller(host=DOMAIN_CONTROLLER_IP_ADDRESS,port=${jboss.doma
in.master.port:9999},security-realm="ManagementRealm")
```

Par exemple :

```
/host=host-slave01/:write-remote-domain-
controller(host="192.168.1.200",port=${jboss.domain.master.port
:9999},security-realm="ManagementRealm")
```

Vous devriez voir apparaître le résultat suivant.

```
"outcome" => "success"
```

Cela modifie le XML du fichier `host - slave01.xml` comme suit :

```
<domain-controller>
  <remote host="192.168.1.200"
port="${jboss.domain.master.port:9999}" security-
realm="ManagementRealm"/>
</domain-controller>
```

2. Configurez l'authentification pour chaque serveur esclave.

Chaque serveur esclave a besoin d'un nom d'utilisateur et d'un mot de passe créé dans le **ManagementRealm** du contrôleur de domaine ou du master autonome. Sur le contrôleur de domaine ou sur le master autonome, exécutez la commande `EAP_HOME/bin/add-user.sh`. Ajouter un utilisateur avec le même nom d'utilisateur comme esclave, au **ManagementRealm**. Quand on vous demandera si cet utilisateur doit s'authentifier auprès d'une instance de JBoss EAP 6 externe, répondez **Oui**. Vous trouverez un exemple de l'entrée et de la sortie de la commande ci-dessous, pour un esclave appelé `slave1`, et un mot de passe `changeme`.

```
user:bin user$ ./add-user.sh
```

```
What type of user do you wish to add?
```

```
a) Management User (mgmt-users.properties)
```

```
b) Application User (application-users.properties)
```

```
(a): a
```

```
Enter the details of the new user to add.
```

```
Realm (ManagementRealm) :
```

```
Username : slave1
```

```
Password : changeme
```

```
Re-enter Password : changeme
```

```
About to add user 'slave1' for realm 'ManagementRealm'
```

```
Is this correct yes/no? yes
```

```
Added user 'slave1' to file '/home/user/jboss-eap-
6.0/standalone/configuration/mgmt-users.properties'
```

```
Added user 'slave1' to file '/home/user/jboss-eap-
6.0/domain/configuration/mgmt-users.properties'
```

```
Is this new user going to be used for one AS process to connect to
another AS process e.g. slave domain controller?
```

```
yes/no? yes
```

```
To represent the user add the following to the server-identities
definition <secret value="Y2hhbmdlbWU=" />
```

3. Copiez l'élément codé-Base64 <secret> à partir de la sortie `add-user . sh`.

Si vous prévoyez de spécifier un mot de passe codé Base64 pour l'authentification, copiez l'élément <secret> à partir de la dernière ligne de la sortie `add-user . sh` car vous en aurez besoin à l'étape suivante.

4. Modifiez le domaine de sécurité de l'hôte esclave pour la nouvelle authentification.

Vous pourrez spécifier la valeur secrète d'une des manières suivantes :

- o Spécifier le mot de passe codé-Base64 dans le fichier de configuration du serveur par le CLI.
 - a. Lancez la Management CLI, avec la commande `EAP_HOME/bin/jboss-cli . sh` dans Linux ou bien la commande `EAP_HOME\bin\jboss-cli . bat` dans le serveur Microsoft Windows. Saisir la commande `connect` pour connecter le contrôleur de domaine sur l'hôte local, ou `connect IP_ADDRESS` pour vous connecter à un contrôleur de domaines sur un serveur éloigné.
 - b. Spécifiez la valeur secrète en saisissant la commande suivante. Veillez bien à remplacer `SECRET_VALUE` par la valeur secrète retournée de la sortie `add-user . sh` lors de l'étape précédente.

```
/host=master/core-service=management/security-
realm=ManagementRealm/server-
identity=secret:add(value="SECRET_VALUE")
:reload
```

Vous devriez voir le résultat suivant pour chaque commande.

```
"outcome" => "success"
```

- o Configurez l'hôte pour obtenir un mot de passe de l'archivage sécurisé.
 - a. Utilisez le script `vault . sh` pour générer un mot de passe masqué. Cela générera une chaîne comme la suivante :
`VAULT::secret::password::ODVmYmJjNGMtZDU2ZC00YmN1LWE4ODMtZjQ1NWNmNDU4ZDc1TE1ORV9CUkVBS3ZhdWx0.`
 - b. Lancez la Management CLI, avec la commande `EAP_HOME/bin/jboss-cli . sh` dans Linux ou bien la commande `EAP_HOME\bin\jboss-cli . bat` dans le serveur Microsoft Windows. Saisir la commande `connect` pour connecter le contrôleur de domaine sur l'hôte local, ou `connect IP_ADDRESS` pour vous connecter à un contrôleur de domaines sur un serveur éloigné.
 - c. Spécifiez la valeur secrète en saisissant la commande suivante. Veillez bien à remplacer `SECRET_VALUE` par le mot de passe masqué généré lors de l'étape précédente.

```
/host=master/core-service=management/security-
realm=ManagementRealm/server-
identity=secret:add(value="{VAULT::secret::password::SECRET_V
```

```
ALUE}"))
:reload
```

Vous devriez voir le résultat suivant pour chaque commande.

```
"outcome" => "success"
```



NOTE

Quand vous créez un mot de passe dans l'archivage sécurisé, celui-ci devra être spécifié en texte brut, et non pas codé Base64.

- o **Spécifiez le mot de passe en tant que propriété système.**

Les exemples suivants utilisent `server.identity.password` comme nom de propriété de système pour le mot de passe.

- a. Spécifiez la propriété système pour le mot de passe dans le fichier de configuration du serveur par le CLI.

- i. Lancez la Management CLI, avec la commande `EAP_HOME/bin/jboss-cli.sh` dans Linux ou bien la commande `EAP_HOME\bin\jboss-cli.bat` dans le serveur Microsoft Windows. Saisir la commande `connect` pour connecter le contrôleur de domaine sur l'hôte local, ou `connect IP_ADDRESS` pour vous connecter à un contrôleur de domaines sur un serveur éloigné.

- ii. Saisir la commande suivante pour configurer l'identité secrète pour utiliser la propriété système.

```
/host=master/core-service=management/security-
realm=ManagementRealm/server-
identity=secret:add(value="{server.identity.password}")
:reload
```

Vous allez voir le résultat suivant pour chaque commande.

```
"outcome" => "success"
```

- b. Quand vous spécifiez le mot de passe en tant que propriété système, vous pouvez configurer l'hôte d'une des manières suivantes :

- Démarrez le serveur en saisissant le mot de passe en texte brut comme argument de ligne de commande, comme par exemple :

```
-Dserver.identity.password=changeme
```



NOTE

Le mot de passe doit être saisi en texte brut et sera visible par quiconque lance la commande `ps -ef`.

- Mettez le mot de passe dans un fichier de propriétés et passez l'URL du fichier de propriétés sous forme d'argument de ligne de commande.

- i. Ajoutez la paire clé/valeur à un fichier de propriétés. Par exemple :

```
server.identity.password=changeme
```

- ii. Démarrez le serveur par les arguments de ligne de commande

```
--properties=URL_TO_PROPERTIES_FILE
```

5. Redémarrez le serveur.

L'esclave va maintenant authentifier le master en utilisant son nom d'hôte comme nom d'utilisateur et le string codifié comme mot de passe.

Résultat

Votre serveur autonome, ou les serveurs au sein d'un groupe de serveurs d'un domaine géré, sont désormais configurés en tant que nœuds de worker du `mod_cluster`. Si vous déployez une application en cluster, ses sessions seront répliquées sur tous les nœuds de cluster de basculement, et seront en mesure d'accepter les demandes provenant d'un serveur web externe ou d'un équilibreur de charges. Chaque nœud du cluster détecte les autres nœuds à l'aide d'automatic discovery, par défaut. Pour configurer la détection automatique et les autres paramètres spécifiques du sous-système `mod_cluster`, reportez-vous à [Section 17.6.2, « Configurer le sous-système mod_cluster »](#). Pour configurer le serveur Apache HTTP, reportez-vous à [Section 17.4.5, « Utiliser un serveur web externe comme Web frontal pour les applications JBoss EAP 6. »](#).

[Rapporter un bogue](#)

17.6.7. Migration du trafic entre les clusters

Résumé

Après avoir créé un nouveau cluster à l'aide de JBoss EAP 6, vous souhaitez sans doute migrer le trafic d'un ancien cluster vers un nouveau dans le cadre d'un processus de mise à niveau. Au cours de cette tâche, vous verrez la stratégie qui peut être utilisée pour migrer ce trafic avec un minimum de temps mort.

Conditions préalables

- Nouvelle installation de cluster : [Section 17.6.2, « Configurer le sous-système mod_cluster »](#) (nous appellerons ce cluster : NOUVEAU Cluster).
- Une ancienne installation de cluster a été rendue obsolète (nous appellerons ce cluster : ANCIEN Cluster).

Procédure 17.11. Mise à niveau du processus pour les clusters

1. Installez votre nouveau cluster en suivant les étapes décrites dans les conditions préalables.
2. Pour les NOUVEAUX et ANCIENS clusters à la fois, assurez-vous que l'option de configuration `sticky-session` est définie sur `true` (`true` par défaut). L'activation de cette option signifie que toutes les nouvelles demandes présentées à un nœud de cluster dans un de ces clusters continueront d'aller vers ce nœud.

```
/profile=full-ha/subsystem=modcluster/mod-cluster-
config=configuration/:write-attribute(name=sticky-
session,value=true)
```

3. Ajouter les nœuds dans le NOUVEAU cluster à la configuration de `mod_cluster` individuellement à l'aide du processus décrit ici : [Section 17.6.6, « Configurer un nœud de worker de mod_cluster »](#)
4. Configurer l'équilibrage de la charge (`mod_cluster`) pour arrêter les contextes individuels dans l'ANCIEN Cluster. L'arrêt des contextes (par opposition à leur désactivation) dans l'ANCIEN cluster permettra aux contextes individuels de s'arrêter gracieusement (et éventuellement à un arrêt total). Les sessions existantes seront toujours servies, mais aucune nouvelle session ne sera dirigée vers ces nœuds. Les contextes arrêtés peuvent prendre plusieurs minutes ou même plusieurs heures pour s'arrêter.

Vous pouvez utiliser le CLI suivant pour stopper un contexte. Remplacer les valeurs de paramètre par des valeurs adaptées à votre environnement.

```
[standalone@localhost:9999 subsystem=modcluster] :stop-
context(context=/myapp, virtualhost=default-host, waittime=50)
```

Résultat

Vous avez effectué la mise à niveau de JBoss EAP 6 Cluster avec succès.

[Rapporter un bogue](#)

17.6.8. Configurer `fail_on_status` pour `mod_cluster`

`fail_on_status` sert à spécifier un ou plusieurs codes de statut HTTP, de façon à ce que si un nœud de worker de cluster retourne un des codes de statuts spécifiés, ce worker échouera. L'équilibreur de charge enverra alors les demandes futures vers un autre nœud de worker du cluster. Le nœud de worker défaillant gardera le statut **NOTOK** jusqu'à ce qu'il envoie un message **STATUS** à l'équilibreur de charge.

`fail_on_status` doit être configuré dans le fichier `httpd` de votre équilibreur de charges. On peut spécifier plusieurs codes de statut HTTP pour `fail_on_status` dans une liste de codes de status séparés par des virgules. Les exemples suivants spécifient les codes de statut **203** et **204** pour `fail_on_status`.

Exemple 17.5. Exemple de configuration de `fail_on_status`

```
ProxyPass / balancer://MyBalancer stickysession=JSESSIONID|jsessionId
nofailover=on failonstatus=203,204
ProxyPassReverse / balancer://MyBalancer
ProxyPreserveHost on
```

[Rapporter un bogue](#)

17.7. APACHE MOD_JK

17.7.1. Le connecteur Apache mod_jk HTTP

Apache mod_jk est un connecteur HTTP fourni aux clients qui en ont besoin pour des raisons de compatibilité. Il fournit un équilibrage des charges et fait partie des packages natifs Red Hat JBoss Enterprise Application Platform 6.X.0 Webserver Connector Natives (zip installation) disponibles sur le portail clients de Red Hat <https://access.redhat.com>. mod_jk peut être installé par les RPM. Pour obtenir des informations sur la façon d'installer les RPM, voir [Section 17.7.4, « Installer le Module_jk_mod dans Apache HTTPD Server \(RPM\) »](#). Pour les plateformes prises en charge, consulter <https://access.redhat.com/site/articles/111663>. Le connecteur mod_jk est maintenu par Apache, et sa documentation se trouve à l'adresse suivante <http://tomcat.apache.org/connectors-doc/>.

JBoss EAP 6 peut accepter des charges de travail en provenance d'un serveur proxy Apache HTTP. Le serveur proxy accepte les requêtes des clients en provenance des serveurs frontaux web, et passe le travail à des serveurs JBoss Enterprise Application Platform participants. Quand les sessions sticky sont activées, une requête en provenance d'un même client va toujours vers le même serveur JBoss EAP 6, à moins que celui-ci ne soit pas rendu disponible.

À la différence du JBoss HTTP connector mod_cluster, un connecteur mod_jk HTTP ne connaît pas le statut des déploiements sur les serveurs ou groupes de serveurs, et ne peut donc pas ajuster les envois en fonction.

mod_jk communique par le protocole AJP 1.3. mod_cluster prend en charge d'autres protocoles. Pour obtenir plus d'informations, voir le tableau de fonctionnalités et de contraintes de connecteur HTTP dans [Section 17.2.3, « Connecteurs HTTP - Aperçu général »](#).



NOTE

mod_cluster est un équilibreur de charge plus avancé que mod_jk. mod_cluster fournit toute la fonctionnalité de mod_jk et quelques fonctionnalités supplémentaires. Pour plus d'informations sur mod_cluster, consulter [Section 17.6.1, « Le connecteur HTTP mod_cluster »](#).

Prochaine étape : configurer une instance de plateforme JBoss EAP 6 pour qu'elle puisse participer à un groupe d'équilibrage des charges mod_jk.

- [Section 17.4.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »](#)
- [Section 17.7.3, « Installer le module jk_mod dans un serveur Apache HTTP \(ZIP\) »](#)

[Rapporter un bogue](#)

17.7.2. Configurer JBoss EAP 6 pour qu'il communique avec Apache Mod_jk

Aperçu

Le connecteur mod_jk HTTP possède un simple composant, le module mod_jk.so, qui est chargé par le serveur web. Ce module reçoit les demandes des clients et les transfère vers le conteneur, en l'occurrence JBoss EAP 6. JBoss EAP 6 doit également être configuré pour accepter ces demandes et envoyer leurs réponses vers le serveur web.

La configuration du serveur Apache HTTPD est couverte dans [Section 17.7.3, « Installer le module jk_mod dans un serveur Apache HTTP \(ZIP\) »](#).

Pour que JBoss Enterprise Application Platform puisse communiquer avec HTTP Apache, il doit avoir le connecteur AJP/1.3 activé. Ce connecteur sera présent par défaut dans les configurations suivantes :

- Dans un domaine géré, dans les groupes de serveurs qui utilisent les profils **ha** et **full-ha**, et le groupe de liaisons de sockets **ha** ou **full-ha**. Le groupe de serveurs **other-server-group** est configuré correctement dans une installation par défaut.
- Dans un serveur autonome, les profils **standalone-ha** et **standalone-full-ha** sont configurés pour les configurations en cluster. Pour démarrer le serveur autonome avec un de ces profils, lancer la commande ci-dessous, à partir du répertoire **EAP_HOME/**. Remplacer par le nom de profil approprié.

```
EAP_HOME/bin/standalone.sh --server-config=standalone-ha.xml
```

Dans Windows, saisir ce qui suit en ligne de commande :

```
EAP_HOME\bin\standalone.bat --server-config=standalone-ha.xml
```

[Rapporter un bogue](#)

17.7.3. Installer le module `jk_mod` dans un serveur Apache HTTP (ZIP)

Conditions préalables

- Pour cette tâche, vous devrez utiliser Apache HTTPD installé dans un environnement pris en charge ou l'Apache HTTP installé sur JBoss Enterprise Web Server. Notez que JBoss Enterprise Web Server fait partie de la distribution JBoss EAP 6.
- Si vous devez installer un serveur Apache HTTP natif de Red Hat Enterprise Linux, utilisez les instructions qui se trouvent dans *Red Hat Enterprise Linux Deployment Guide*
- Si vous devez installer un serveur HP-UX native Apache HTTP Server, utiliser les instructions qui se trouvent dans le guide *HP-UX Web Server Suite Installation Guide* à l'adresse suivante <https://h20392.www2.hp.com/portal/swdepot/displayInstallInfo.do?productNumber=HPUXWSATW232>.
- Si vous avez besoin d'installer JBoss Enterprise Web Server, utiliser les instructions dans *JBoss Enterprise Web Server Installation Guide*
- Si vous utilisez le serveur Apache HTTP, télécharger le package JBoss EAP 6 Native Components pour votre plate-forme du portail clients de Red Hat à <https://access.redhat.com>. Ce paquet contient à la fois les binaires `mod_jk` et `mod_cluster` qui sont précompilés pour Red Hat Enterprise Linux. Si vous utilisez JBoss Enterprise Web Server, il comprend déjà le binaire pour `mod_jk`.
- Si vous utilisez Red Hat Enterprise Linux (RHEL) 5 et le serveur natif Apache HTTP (httpd 2.2.3), commencez par télécharger le module `mod_perl` pour charger le module `mod_jk`.
- Vous devez être connectés avec des privilèges administratifs (root).

Procédure 17.12. Installer le module `mod_cluster`

1. Configurer le module `mod_jk`.

- a. Créer un nouveau fichier nommé `HTTPD_HOME/conf.d/mod-jk.conf` et y ajouter ce qui suit.



NOTE

La directive `JkMount` indique quels URL Apache doivent aller vers le module `mod_jk`. Sur la base de la configuration de la directive, `mod_jk` transfère l'URL reçu aux conteneurs de servlet qui conviennent.

Pour servir le contenu directement, et pour n'utiliser que l'équilibreur de charges pour les applications Java, le chemin URL doit être `/application/*`. Pour utiliser `mod_jk` en tant qu'équilibreur des charges, utiliser la valeur `/*` pour transférer tous les URL au `mod_jk`.

```
# Load mod_jk module
# Specify the filename of the mod_jk lib
LoadModule jk_module modules/mod_jk.so

# Where to find workers.properties
JkWorkersFile conf/workers.properties

# Where to put jk logs
JkLogFile logs/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel info

# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"

# JkOptions indicates to send SSK KEY SIZE
JkOptions +ForwardKeySize -ForwardDirectories

# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Mount your applications
# The default setting only sends Java application data to mod_jk.
# Use the commented-out line to send all URLs through mod_jk.
# JkMount /* loadbalancer
JkMount /application/* loadbalancer

# Add shared memory.
# This directive is present with 1.2.10 and
# later versions of mod_jk, and is needed for
# for load balancing to work properly
JkShmFile logs/jk.shm

# Add jkstatus for managing runtime data
<Location /jkstatus/>
JkMount status
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>
```


■

Observer les valeurs et vérifier qu'elles conviennent à votre installation. Quand vous serez satisfait, sauvegarder le fichier.

b. Spécifier une directive JKMountFile

En plus de la directive JKMount de `mod-jk.conf`, vous pourrez spécifier un fichier qui contienne des modèles URL multiples à transférer au `mod_jk`.

i. Ajouter ce qui suit au fichier `HTTPD_HOME/conf/mod-jk.conf` :

```
# You can use external file for mount points.
# It will be checked for updates each 60 seconds.
# The format of the file is: /url=worker
# /examples/*=loadbalancer
JKMountFile conf/uriworkermap.properties
```

ii. Créer un nouveau fichier intitulé `HTTPD_CONF/uriworkermap.properties`, avec une ligne pour chaque modèle URL à faire correspondre. L'exemple suivant montre des exemples de syntaxe pour ce fichier.

```
# Simple worker configuration file
/*=loadbalancer
```

c. Copier le fichier `mod_jk.so` dans le répertoire de modules d'`httpd`



NOTE

Cela n'est utile que si le serveur HTTP Apache n'a pas de `mod_jk.so` dans son répertoire `modules/`. Vous pourriez éviter cette étape si vous utilisez le serveur Apache HTTP inclus comme téléchargement de JBoss EAP 6.

Extraire le paquet Native Web Server Connectors Zip. Localiser le fichier `mod_jk.so` soit dans le répertoire

`EAP_HOME/modules/system/layers/base/native/lib/httpd/modules/` ou dans le répertoire

`EAP_HOME/modules/system/layers/base/native/lib64/httpd/modules/` suivant que votre système d'exploitation est de 32-bit ou de 64-bit.

Copier le fichier dans le répertoire `HTTPD_MODULE/`.

2. Configurer les noeuds de worker `mod_jk`.

a. Créer un nouveau fichier nommé `HTTPD_CONF/workers.properties`. Utiliser l'exemple suivant comme point de départ, et modifier le fichier selon vos besoins.

```
# Define list of workers that will be used
# for mapping requests
worker.list=loadbalancer,status

# Define Node1
# modify the host as your host IP or DNS name.
worker.node1.port=8009
worker.node1.host=node1.mydomain.com
worker.node1.type=ajp13
```

```

worker.node1.ping_mode=A
worker.node1.lbfactor=1

# Define Node2
# modify the host as your host IP or DNS name.
worker.node2.port=8009
worker.node2.host=node2.mydomain.com
worker.node2.type=ajp13
worker.node2.ping_mode=A
worker.node2.lbfactor=1

# Load-balancing behavior
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1

# Status worker for managing load balancer
worker.status.type=status

```

Pour obtenir une description détaillée de la syntaxe du fichier `workers.properties`, et pour obtenir des options de configuration avancées, consulter [Section 17.7.5, « Référence de configuration pour les Apache Mod_jk Workers »](#).

3. Redémarrer le serveur web.

La façon de redémarrer le serveur web dépend de savoir si vous utilisez le serveur Apache HTTP de Red Hat Enterprise Linux ou le serveur HTTP inclus dans JBoss Enterprise Web Server. Choisir une des deux méthodes ci-dessous.

- o **Serveur Apache HTTPD de Red Hat Enterprise Linux**

Exécuter la commande suivante :

```
[root@host]# service httpd restart
```

- o **Serveur de JBoss Enterprise Web Server HTTP**

JBoss Enterprise Web Server exécute à la fois sur Red Hat Enterprise Linux et Microsoft Windows Server. La méthode de redémarrage du serveur web est différente pour chacun.

- **Red Hat Enterprise Linux, installé avec RPM**

Dans Red Hat Enterprise Linux, JBoss Enterprise Web Server installe son serveur web en tant que service. Pour redémarrer le serveur web, lancer les deux commandes suivantes :

```
[root@host ~]# service httpd stop
[root@host ~]# service httpd start
```

- **Red Hat Enterprise Linux, installé avec Zip**

Si vous avez installé le serveur HTTP Apache de JBoss Enterprise Web à partir d'une archive ZIP, utiliser la commande `apachectl` pour redémarrer le serveur web.

Remplacer `EWS_HOME` par le répertoire où vous avez décompressé le serveur JBoss Enterprise Web Server Apache HTTP.

```
[root@host ~]# EWS_HOME/httpd/sbin/apachectl restart
```

- **Microsoft Windows Server**

Lancer les commandes suivantes dans une invite de commande avec des privilèges administratifs :

```
C:\> net stop Apache2.2
C:\> net start Apache2.2
```

■ Solaris

Lancer les commandes suivantes dans l'invite de commandes avec des permissions admin. Remplacer *EWS_HOME* par le répertoire dans lequel vous avez décompressé le serveur HTTP Apache de JBoss Enterprise Web.

```
[root@host ~] EWS_HOME/httpd/sbin/apachectl restart
```

Résultat

Le serveur Apache HTTP est maintenant configuré pour pouvoir utiliser l'équilibreur de charges de `mod_jk`. Pour configurer JBoss EAP 6 pour qu'il soit au fait de `mod_jk`, consulter [Section 17.4.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »](#).

[Rapporter un bogue](#)

17.7.4. Installer le Module `_jk_mod` dans Apache HTTPD Server (RPM)

Conditions préalables

- Pour cette tâche, vous devrez utiliser Apache HTTPD installé dans un environnement pris en charge ou l'Apache HTTP installé sur JBoss Enterprise Web Server. Notez que l'Apache HTTP installé dans JBoss Enterprise Web Server fait partie de la distribution JBoss Enterprise Web Server faisant partie de la distribution JBoss EAP 6.
- Si vous devez installer Apache HTTP Server, utilisez les instructions qui se trouvent dans *Red Hat Enterprise Linux Deployment Guide*, dans <https://access.redhat.com/site/documentation/>.
- Si vous avez besoin d'installer le serveur JBoss Enterprise Web Server, utiliser les instructions dans *JBoss Enterprise Web Server Installation Guide* dans <https://access.redhat.com/site/documentation/>.
- Vous devez être connectés avec des privilèges administratifs (root).

Procédure 17.13. Red Hat Enterprise Linux 5 : `mod_jk` with Apache HTTP Server 2.2.3

1. Installer `mod_jk-ap22 1.2.37` et ses dépendances `mod_perl` de `jbappplatform-6-*-server-5-rpm` :

```
yum install mod_jk
```

2. **Option** : copier l'exemple de fichiers de configuration à utiliser :

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/mod_jk.conf.sample
/etc/httpd/conf.d/mod_jk.conf
```

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/workers.properties.sample
/etc/httpd/conf/workers.properties
```

Ces fichiers devront être modifiés pour pouvoir correspondre à vos besoins.

3. Démarrer le serveur :

```
service httpd start
```

NOTE

Le message erreur suivant indique que votre module `mod_jk` a été chargé avant que `mod_eri` ne soit présent :

```
Cannot load /etc/httpd/modules/mod_jk.so into server:
/etc/httpd/modules/mod_jk.so: undefined symbol:
ap_get_server_description
```

Pour s'assurer que le module `mod_perl` soit chargé avant `mod_jk`, ajouter ce qui suit à `/etc/httpd/conf.d/mod_jk.conf` :

```
<IfModule !perl_module>
    LoadModule perl_module modules/mod_perl.so
</IfModule>
LoadModule jk_module modules/mod_jk.so
```

Procédure 17.14. Red Hat Enterprise Linux 5 : mod_jk with JBoss EAP Apache HTTP Server 2.2.26

1. Installer à la fois `mod_jk` et le dernier Apache HTTP Server 2.2.26 fourni par le canal `jbappplatform-6-* -server-5-rpm` à l'aide de cette commande :

```
yum install mod_jk httpd
```

2. **Option** : copier l'exemple de fichiers de configuration à utiliser :

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/mod_jk.conf.sample
/etc/httpd/conf.d/mod_jk.conf
```

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/workers.properties.sample
/etc/httpd/conf/workers.properties
```

Ces fichiers devront être modifiés pour pouvoir correspondre à vos besoins.

3. Démarrer le serveur :

```
service httpd start
```

Procédure 17.15. Red Hat Enterprise Linux 6 : mod_jk et JBoss EAP Apache HTTP Server 2.2.26

1. Installer `mod_jk-ap22.1.2.37` et le package Apache HTTP Server 2.2.26 `httpd` du canal `jbappplatform-6-* -server-6-rpm` (toutes les versions existantes doivent être mises à jour) :

```
yum install mod_jk httpd
```

2. Option : copier l'exemple de fichiers de configuration à utiliser :

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/mod_jk.conf.sample
/etc/httpd/conf.d/mod_jk.conf
```

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/workers.properties.sample
/etc/httpd/conf/workers.properties
```

Ces fichiers devront être modifiés pour pouvoir correspondre à vos besoins.

3. Démarrer le serveur :

```
service httpd start
```

Procédure 17.16. Red Hat Enterprise Linux 6 : mod_jk with Apache HTTP Server 2.2.15**1. Installer mod_jk with Apache HTTP Server 2.2.15 par la commande suivante :**

```
yum install mod_jk
```

2. Option : copier l'exemple de fichiers de configuration à utiliser :

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/mod_jk.conf.sample
/etc/httpd/conf.d/mod_jk.conf
```

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/workers.properties.sample
/etc/httpd/conf/workers.properties
```

Ces fichiers devront être modifiés pour pouvoir correspondre à vos besoins.

3. Démarrer le serveur :

```
service httpd start
```

Procédure 17.17. Red Hat Enterprise Linux 7: mod_jk et JBoss EAP Apache HTTP Server 2.2.26**1. Installer mod_jk-ap22 1.2.37 et le package Apache HTTP Server 2.2.26 httpd22 du canal `jbappplatform-6-* -server-6-rpm` (tous les versions existantes doivent être mises à jour) :**

```
yum install mod_jk
```

2. Option : copier l'exemple de fichiers de configuration à utiliser :

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/mod_jk.conf.sample
/etc/httpd22/conf.d/mod_jk.conf
```

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/workers.properties.sample
/etc/httpd22/conf/workers.properties
```

Ces fichiers devront être modifiés pour pouvoir correspondre à vos besoins.

3. Démarrer le serveur :

```
systemctl start httpd22.service
```

[Rapporter un bogue](#)

17.7.5. Référence de configuration pour les Apache Mod_jk Workers

Le fichier `workers.properties` définit le comportement de workers à qui mod_jk passe les requêtes de clients. Dans Red Hat Enterprise Linux, le fichier se trouve dans `/etc/httpd/conf/workers.properties`. Le fichier `workers.properties` définit où les différents conteneurs de servlet se trouvent, et la façon dont la charge de travail doit être distribuée parmi eux.

La configuration est divisée en trois sections. La première section traite des propriétés globales, qui s'appliquent à tous les workers. La deuxième section contient des paramètres qui s'appliquent à un équilibreur de charge spécifique. La troisième section contient les paramètres qui s'appliquent à un nœud spécifique, équilibré par l'équilibreur de charge.

La structure générale d'une propriété est `worker.WORKER_NAME.DIRECTIVE`, avec `WORKER_NAME` comme nom unique de worker, et `DIRECTIVE` comme paramètre de configuration à appliquer au worker.

Référence de configuration pour les équilibreurs de charge d'Apache mod_jk

Les modèles spécifient les paramètres d'équilibrage de charge par défaut. Vous pouvez remplacer le modèle contenu dans les paramètres de l'équilibreur de charge lui-même. Vous pouvez voir un exemple de modèle d'équilibreur de charge dans [Exemple 17.6, « Exemple de fichier workers.properties »](#).

Tableau 17.13. Propriétés globales

Propriété	Description
<code>worker.list</code>	La liste des noms d'équilibreur de charge utilisés par mod_jk. Ces équilibreurs de charge sont prêts à recevoir des requêtes.

Tableau 17.14. Directives obligatoires

Propriété	Description
<code>type</code>	Le type d'équilibreur de charge. Le type par défaut est <code>ajp13</code> . Autres valeurs possibles <code>ajp14</code> , <code>lb</code> , <code>status</code> . Pour plus d'informations sur ces directives, voir la référence de protocole Apache Tomcat Connector AJP à http://tomcat.apache.org/connectors-doc/ajp/ajpv13a.html .

Tableau 17.15. Directives d'équilibreur de charge

Propriété	Description
<code>balance_workers</code>	Spécifie les nœuds de worker que l'équilibreur de charge doit gérer. Vous pouvez utiliser la directive plusieurs fois pour un même équilibrage de charge. Il se compose d'une liste séparée par des virgules des noms de workers. Ceci est défini par worker, et non pas par nœud. Elle affecte tous les nœuds équilibrés par ce type de worker.
<code>sticky_session</code>	Indique si les demandes d'une même session sont toujours acheminées vers le même équilibreur de charge. La valeur par défaut est 0 , ce qui signifie que les sticky sessions sont désactivées. Pour activer des sticky sessions, définir à 1 . Les sticky sessions doivent habituellement être activées, à moins que toutes vos demandes soient vraiment stateless. Ceci est défini par équilibreur de charge, et non pas par nœud. Affecte tous les nœuds équilibrés par ce type d'équilibreur de charge.

Tableau 17.16. Directives de connexion

Propriété	Description
<code>hôte</code>	Le nom d'hôte ou l'adresse IP de l'équilibreur de charge. L'équilibreur de charge doit supporter la pile de protocole ajp . La valeur par défaut est localhost .
<code>Important</code>	Le numéro de port de l'instance de serveur éloigné qui écoute les requêtes de protocoles définis. La valeur par défaut est 8009 , qui correspond au port d'écoute des équilibreurs de charge AJP13. La valeur par défaut des équilibreurs de charge AJP14 est 8011 .
<code>ping_mode</code>	<p>Les conditions dans lesquelles les connexions sont interrogées pour le statut du réseau. La sonde utilise un paquet AJP13 vide pour CPing et s'attend à un CPong en réponse. Spécifier les conditions à l'aide d'une combinaison d'indicateurs de la directive. Les indicateurs ne sont pas séparés par une virgule ou un espace blanc. Le ping_mode peut être n'importe quelle combinaison de C, P, I, ou A.</p> <ul style="list-style-type: none"> • C - Connect. Sonde la connexion une fois seulement suite à la connexion au serveur. Spécifier le timeout en utilisant la valeur de connect_timeout. Sinon, la valeur de ping_timeout sera utilisée. • P - Prepost. Sonde la connexion avant d'envoyer une requête au serveur. Spécifier le timeout en utilisant la directive prepost_timeout. Sinon, la valeur ping_timeout sera utilisée. • I - Interval. Interroge la connexion à un intervalle spécifié par connection_ping_interval, si présent. Sinon, utilise la valeur de ping_timeout. • A - All. Un raccourci pour CPI, qui indique que toutes les sondes de connexion sont utilisées.

Propriété	Description
ping_timeout, connect_timeout, prepost_timeout, connection_ping_interval	Les valeurs de timeout pour les paramètres de sonde de connexion ci-dessus. La valeur est spécifiée en millisecondes, et la valeur par défaut pour ping_timeout est de 10000.
lbfactor	Spécifie le facteur d'équilibrage des charges d'un équilibreur individuel et ne s'applique qu'à un noeud de worker membre d'un équilibreur de charge. Ceci est utile pour donner à un serveur plus puissant, une charge de travail supplémentaire. Pour donner à un worker 3 fois la charge de la valeur par défaut, définir cette valeur à 3:worker.my_worker.lbfactor=3

Exemple 17.6. Exemple de fichier workers.properties

```
worker.balancer1.type=lb
worker.balancer2.type=lb

worker.balancer1.sticky_sessions=1
worker.balancer1.balance_workers=node1
worker.balancer2.sticky_session=1
worker.balancer2.balance_workers=node2,node3

worker.nodetemplate.type=ajp13
worker.nodetemplate.port=8009

worker.node1.template=nodetemplate
worker.node1.host=localhost
worker.node1.ping_mode=CI
worker.node1.connection_ping_interval=9000
worker.node1.lbfactor=1

worker.node2.template=nodetemplate
worker.node2.host=192.168.1.1
worker.node2.ping_mode=A

worker.node3.template=nodetemplate
worker.node3.host=192.168.1.2
```

L'exemple ci-dessus vous montre l'utilisation de plusieurs équilibreurs de charge au service du contenu pour le compte du serveur web. Les raisons de telles configurations peuvent être :

- Avoir différents contextes pouvant être servis par différents équilibreurs de charge, fournissant un environnement de développement dans lequel tous les développeurs peuvent partager le même serveur web tout en ayant un équilibreur de charge propre.
- Avoir plusieurs hôtes virtuels servis par différents processus, offrant ainsi une séparation claire entre les sites appartenant à des sociétés diverses.
- Fournir un équilibrage des charges, c-a-d exécuter plusieurs équilibreurs de charge sur une même machine et répartir les demandes entre eux.

Les détails de configuration de ce document sont limités. Voir la documentation Apache à <http://tomcat.apache.org/connectors-doc/> pour obtenir des instructions supplémentaires.

[Rapporter un bogue](#)

17.8. APACHE MOD_PROXY

17.8.1. Le connecteur Apache mod_proxy HTTP

Apache offre deux modules différents d'équilibrage de charge et de proxying pour ses démons httpd : `mod_proxy` et `mod_jk`. Pour en savoir plus sur `mod_jk`, consulter [Section 17.7.1, « Le connecteur Apache mod_jk HTTP »](#). La plate-forme JBoss EAP 6 prend en charge l'utilisation de l'un d'entre eux, bien que `mod_cluster`, le connecteur HTTP de JBoss, couple plus étroitement JBoss EAP 6 et httpd externe, et est le connecteur HTTP recommandé. Reportez-vous à [Section 17.2.3, « Connecteurs HTTP - Aperçu général »](#) pour une vue d'ensemble des connecteurs HTTP pris en charge, y compris les avantages et les inconvénients.

À la différence de `mod_jk`, `mod_proxy` supporte les connexions via les protocoles HTTP et HTTPS. Chacun d'entre eux supporte le protocole AJP.

`mod_proxy` peut être configuré en autonome ou en configuration d'équilibrage de charge, et il prend en charge la notion de sticky sessions.

Le module `mod_proxy` nécessite que JBoss EAP 6 ait le connecteur web HTTP, HTTPS ou AJP configuré. Cela fait partie du sous-système web. Consulter [Section 15.1, « Configurer le sous-système web »](#) pour obtenir des informations sur la façon de configurer le sous-système web.



NOTE

`mod_cluster` est un équilibreur de charges plus avancé que `mod_proxy`. `mod_cluster` fournit toute la fonctionnalité de `mod_proxy` et quelques fonctionnalités supplémentaires. Pour plus d'informations sur `mod_cluster`, consulter [Section 17.6.1, « Le connecteur HTTP mod_cluster »](#).

[Rapporter un bogue](#)

17.8.2. Installer un connecteur Mod_proxy HTTP sur le serveur Apache HTTPD

Aperçu

`mod_proxy` est un module d'équilibrage de charges fourni par Apache. Cette tâche présente une configuration de base. Pour plus d'informations sur la configuration avancée, ou pour plus de détails, reportez-vous à la documentation Apache `mod_proxy` à https://httpd.apache.org/docs/2.2/mod/mod_proxy.html. Pour plus d'informations sur `mod_proxy` d'une perspective JBoss EAP 6, consulter [Section 17.8.1, « Le connecteur Apache mod_proxy HTTP »](#) et [Section 17.2.3, « Connecteurs HTTP - Aperçu général »](#).

Conditions préalables

- Un serveur Apache HTTP de JBoss Enterprise Web Server httpd ou bien fourni par un système d'exploitation doit être installé. Un serveur Apache HTTP autonome est fourni séparément dans le portail clients Red Hat, dans la zone de téléchargement de JBoss EAP 6. Voir

[Section 17.4.2, « Installer le serveur Apache HTTP inclus dans JBoss EAP 6 »](#) pour obtenir des informations sur ce serveur Apache HTTP si vous souhaitez l'utiliser.

- Les modules `mod_proxy` doivent être installés. Le serveur Apache HTTPD est généralement livré avec les modules `mod_proxy` déjà inclus. C'est le cas sur Red Hat Enterprise Linux et sur le serveur Apache HTTP qui vient avec le serveur Web de JBoss Enterprise.
- Vous devez avoir les permissions `root` ou administrateur pour pouvoir modifier la configuration de Serveur Apache HTTP.
- Dans notre exemple, on assume que JBoss EAP 6 est configuré avec le connecteur web HTTP ou HTTPS. Cela fait partie de la configuration du sous-système web. Voir [Section 15.1, « Configurer le sous-système web »](#) pour obtenir des informations sur la façon de configurer le sous-système web.

1. Activer les modules `mod_proxy` dans le démon `httpd`

Recherchez les lignes suivantes dans votre fichier `HTTPD_CONF/httpd.conf`. Si elles ne sont pas présentes, ajoutez-les en bas. Si elles sont présentes, mais que les lignes commencent par un caractère de commentaire (`#`), supprimer le caractère. Enregistrez le fichier par la suite. Habituellement, les modules sont déjà présents et activés.

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_http_module modules/mod_proxy_http.so
# Uncomment these to proxy FTP or HTTPS
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
```

2. Ajouter un proxy non équilibreur de charges.

Ajouter la configuration suivante à votre fichier `HTTPD_CONF/httpd.conf`, directement sous une directive `<VirtualHost>` que vous possédez sans doute. Remplacer les valeurs par des valeurs appropriées à votre installation.

Cet exemple utilise un hôte virtuel. Voir la nouvelle étape pour utiliser la configuration `httpd` par défaut.

```
<VirtualHost *:80>
# Your domain name
ServerName Domain_NAME_HERE

ProxyPreserveHost On

# The IP and port of JBoss EAP 6
# These represent the default values, if your httpd is on the same
host
# as your JBoss EAP 6 managed domain or server

ProxyPass / http://localhost:8080/
ProxyPassReverse / http://localhost:8080/

# The location of the HTML files, and access control information
DocumentRoot /var/www
<Directory /var/www>
Options -Indexes
Order allow,deny
```

```

Allow from all
</Directory>
</VirtualHost>

```

Après avoir appliqué vos changements, sauvegarder le fichier.

3. Ajouter le proxy d'équilibrage des charges.

Pour utiliser `mod_proxy` comme équilibreur de charges, et pour envoyer du travail à des instances multiples de JBoss EAP 6, ajouter la configuration suivante à votre fichier **`HTTPD_CONF/httpd.conf`**. Remplacer les par les valeurs qui correspondent à votre environnement.

```

<Proxy balancer://mycluster>

Order deny,allow
Allow from all

# Add each JBoss Enterprise Application Server by IP address and
port.
# If the route values are unique like this, one node will not fail
over to the other.
BalancerMember http://192.168.1.1:8080 route=node1
BalancerMember http://192.168.1.2:8180 route=node2
</Proxy>

<VirtualHost *:80>
# Your domain name
ServerName YOUR_DOMAIN_NAME

ProxyPreserveHost On
ProxyPass / balancer://mycluster/

# The location of the HTML files, and access control information
DocumentRoot /var/www
<Directory /var/www>
Options -Indexes
Order allow,deny
Allow from all
</Directory>

</VirtualHost>

```

Les exemples ci-dessus communiquent tous par le protocole HTTP. Vous pouvez également utiliser les protocoles AJP ou HTTPS si vous chargez les modules `mod_proxy`. Voir la documentation `mod_proxy` http://httpd.apache.org/docs/2.2/mod/mod_proxy.html pour plus d'informations.

4. Activer les sticky sessions.

Sticky sessions signifie que si la demande d'un client va initialement à un worker spécifique de JBoss EAP 6, toutes les demandes futures seront envoyées au même worker, sauf s'il n'est plus disponible. C'est presque toujours le comportement correct.

Pour activer des sticky sessions du `mod_proxy`, ajoutez le paramètre `stickysession` à l'énoncé `ProxyPass`. Cet exemple montre également d'autres paramètres que vous pouvez utiliser. Reportez-vous à documentation `mod_proxy` Apache à

http://httpd.apache.org/docs/2.2/mod/mod_proxy.html pour plus d'informations à leur sujet.

```
ProxyPass /MyApp balancer://mycluster stickysession=JSESSIONID  
lbmethod=bytraffic nofailover=Off
```

5. Redémarrer le serveur web.

Redémarrer le serveur web pour appliquer les changements.

Résultat

Votre serveur http Apache est configuré pour utiliser le `mod_proxy` pour envoyer des demandes de client aux instances de JBoss EAP 6, en configuration standard ou équilibrage de charge. Pour configurer la plate-forme JBoss EAP 6 pour répondre à ces demandes, reportez-vous à [Section 17.4.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »](#).

[Rapporter un bogue](#)

17.9. MICROSOFT ISAPI

17.9.1. Internet Server API (ISAPI)

Internet Server API (ISAPI) est un ensemble d'API utilisés pour écrire les extensions de Serveur OLE et des filtres pour les serveur Web comme IIS (Internet Information Services) de Microsoft.

`isapi_redirect.dll` est une extension de `mod_jk` ajustée à IIS. `isapi_redirect.dll` vous permet de configurer les instances de JBoss EAP 6 en tant que noeuds de worker avec un IIS comme équilibreur de charge.

[Rapporter un bogue](#)

17.9.2. Téléchargement et extraction de Webserver Connector Natives dans Microsoft IIS

1. Ouvrir un navigateur et connectez-vous au Portail clients de Red Hat à l'adresse suivante <https://access.redhat.com>.
2. Naviguer dans **Downloads**, puis **Red Hat JBoss Middleware Download Software**, et enfin, sélectionner **Enterprise Application Platform** à partir du menu déroulant **Product**.
3. Sélectionner la version qui convient à partir du menu déroulant **Version**.
4. Choisissez entre l'option **Download** et **Red Hat JBoss Enterprise Application Platform <VERSION> Webserver Connector Natives for Windows Server 2008 x86_64** ou encore **Red Hat JBoss Enterprise Application Platform <VERSION> Webserver Connector Natives for Windows Server 2008 i686** suivant l'architecture du serveur.
5. Extraire le fichier zip et copier son contenu dans le répertoire `jboss-eap-<VERSION>/modules/system/layers/base/native/sbin` vers une location sur votre serveur. Le reste de cette tâche assume que vous avez copié le contenu `C:\connectors\`.

[Rapporter un bogue](#)

17.9.3. Configurer Microsoft IIS pour qu'il puisse utiliser ISAPI

Conditions préalables :

- [Section 17.9.2, « Téléchargement et extraction de Webserver Connector Natives dans Microsoft IIS »](#)



NOTE

Voir <https://access.redhat.com/site/articles/111663> pour une liste complète des configurations prises en charge par Microsoft Windows Server et IIS.

Procédure 17.18. Configurer le re-directionneur IIS Redirector à l'aide du IIS Manager (IIS 7)

1. Ouvrir le IIS Manager en cliquant sur **Start** → **Run**, puis, saisir `inetmgr`.
2. Dans le panneau de vue d'arborescence, développer **IIS 7**.
3. Cliquer à deux fois sur **ISAPI and CGI Registrations** pour l'ouvrir sous forme d'une fenêtre séparée.
4. Dans le panneau **Actions**, cliquer sur **Add**. La fenêtre **Add ISAPI or CGI Restriction** s'ouvrira.
5. Indiquer les valeurs suivantes :
 - **ISAPI or CGI Path:** `c:\connectors\isapi_redirect.dll`
 - **Description:** `jboss`
 - **Allow extension path to execute:** sélectionner la case à cocher.
6. Cliquer sur **OK** pour fermer la fenêtre **Add ISAPI or CGI Restriction**.
7. Définir un répertoire virtuel **JBoss Native**
 - a. Cliquer à droite sur **Default Web Site**, puis cliquer sur **Add Virtual Directory**. La fenêtre **Add Virtual Directory** va s'ouvrir.
 - b. Indiquer les valeurs suivantes pour ajouter un répertoire virtuel :
 - **Alias:** `jboss`
 - **Physical Path:** `C:\connectors\`
 - c. Cliquer sur **OK** pour sauvegarder les valeurs et fermer la fenêtre **Add Virtual Directory**.
8. Définir un filtre **JBoss Native ISAPI Redirect**
 - a. Dans le panneau de vue d'arborescence, développer **Sites** → **Default Web Site**.
 - b. Cliquer à deux fois sur **Filtres ISAPI**. L'affichage **ISAPI Filters Features** apparaîtra.
 - c. Dans le panneau **Actions**, cliquer sur **Add**. La fenêtre **Add ISAPI Filter** s'ouvrira.

d. Indiquer les valeurs suivantes dans la fenêtre **Add ISAPI Filter**:

- **Filter name:** jboss
- **Executable:** C:\connectors\isapi_redirect.dll

e. Cliquer **OK** pour sauvegarder les valeurs et pour fermer la fenêtre **Add ISAPI Filters**.

9. Activer le gestionnaire ISAPI-dll

a. Cliquer deux fois sur l'élément **IIS 7** qui se trouve sur le panneau d'affichage : **IIS 7 Home Features View** s'ouvrira.

b. Cliquer deux fois sur **Handler Mappings:Handler Mappings Features View** s'ouvrira.

c. Dans la liste déroulante modifiable **Group by** sélectionner **State**, les **Handler Mappings** s'affichent dans **Enabled and Disabled Groups**.

d. Trouver **ISAPI-dll**. S'il se trouve dans le groupe **Disabled**, cliquer à droite, et sélectionner **Edit Feature Permissions**.

e. Activer les permissions suivantes :

- **Read**
- **Script**
- **Execute**

f. Cliquer sur **OK** pour sauvegarder les valeurs, et fermer la fenêtre **Edit Feature Permissions**.

Résultat

Microsoft IIS est maintenant configuré pour utiliser le re-directionneur ISAPI Redirector. Ensuite, [Section 17.4.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »](#), puis [Section 17.9.4, « Configurer le re-directionneur ISAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP 6 »](#) ou [Section 17.9.5, « Configurer le re-directionneur ISAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6 »](#).

[Rapporter un bogue](#)

17.9.4. Configurer le re-directionneur ISAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP 6

Aperçu

Cette tâche configure un groupe de serveurs de JBoss EAP 6 pour qu'ils puissent accepter les demandes du re-directionneur ISAPI. Il n'inclut pas la configuration d'équilibrage de charge ou de haute disponibilité avec basculement. Si vous avez besoin de ces fonctionnalités, reportez-vous à [Section 17.9.5, « Configurer le re-directionneur ISAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6 »](#).

Cette configuration est faite sur le serveur IIS, et assume que JBoss Enterprise Application Platform est déjà configurée, comme dans [Section 17.4.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »](#).

Conditions préalables

- Vous aurez besoin d'un accès administrateur pour accéder au serveur IIS
- [Section 17.4.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »](#)
- [Section 17.9.3, « Configurer Microsoft IIS pour qu'il puisse utiliser ISAPI »](#)

Procédure 17.19. Modifier les fichiers de propriété et configurer la redirection

1. Créer un répertoire pour stocker la journalisation, les fichiers de propriété, et les fichiers de verrouillage.

Le reste de cette procédure suppose que vous utilisez le répertoire `C:\connectors\` à cet effet. Si vous utilisez un autre répertoire, modifier les instructions en conséquence.

2. Créer le fichier `isapi_redirect.properties`.

Créer un nouveau fichier intitulé `C:\connectors\isapi_redirect.properties`. Copier les contenus suivants dans le fichier.

```
# Configuration file for the ISAPI Redirector
# Extension uri definition
extension_uri=/jboss/isapi_redirect.dll

# Full path to the log file for the ISAPI Redirector
log_file=c:\connectors\isapi_redirect.log

# Log level (debug, info, warn, error or trace)
log_level=info

# Full path to the workers.properties file
worker_file=c:\connectors\workers.properties

# Full path to the uriworkermap.properties file
worker_mount_file=c:\connectors\uriworkermap.properties

#Full path to the rewrite.properties file
rewrite_rule_file=c:\connectors\rewrite.properties
```

Si vous ne souhaitez pas utiliser un fichier `rewrite.properties`, dé-commentez la dernière ligne en plaçant un caractère `#` au début de la ligne. Voir [Étape 5](#) pour plus d'informations.

3. Créer le fichier `uriworkermap.properties`

Le fichier `uriworkermap.properties` contient les mappages entre les URL de l'application déployée et quel worker gère leurs demandes vers eux. Le fichier d'exemple suivant illustre la syntaxe du fichier. Placez votre fichier `uriworkermap.properties` dans `C:\connectors\`.

```
# images and css files for path /status are provided by worker01
/status=worker01
/images/=worker01
/css/=worker01
```

```
# Path /web-console is provided by worker02
# IIS (customized) error page is used for http errors with number
greater or equal to 400
# css files are provided by worker01
/web-console/*=worker02;use_server_errors=400
/web-console/css/*=worker01

# Example of exclusion from mapping, logo.gif won't be displayed
# /web-console/images/logo.gif=*

# Requests to /app-01 or /app-01/something will be routed to
worker01
/app-01|/*=worker01

# Requests to /app-02 or /app-02/something will be routed to
worker02
/app-02|/*=worker02
```

4. Créer le fichier `workers.properties`.

Le fichier `workers.properties` contient des définitions de mappage entre les étiquettes de `workers` et les instances de serveur. Le fichier d'exemple suivant illustre la syntaxe du fichier. Placez ce fichier dans le répertoire `C:\connectors\`.

```
# An entry that lists all the workers defined
worker.list=worker01, worker02

# Entries that define the host and port associated with these
workers

# First JBoss EAP 6 server definition, port 8009 is standard port
for AJP in EAP
worker.worker01.host=127.0.0.1
worker.worker01.port=8009
worker.worker01.type=ajp13

# Second JBoss EAP 6 server definition
worker.worker02.host=127.0.0.100
worker.worker02.port=8009
worker.worker02.type=ajp13
```

5. Créer le fichier `rewrite.properties`.

Le fichier `rewrite.properties` contient des dispositions relatives aux demandes spécifiques de réécriture d'URL simple pour certaines applications. Le chemin d'accès de réécriture est spécifié à l'aide de paires nom / valeur, comme illustré dans l'exemple ci-dessous. Placez ce fichier dans le répertoire `C:\connectors\`.

```
#Simple example
# Images are accessible under abc path
/app-01/abc/=/app-01/images/
```

6. Redémarrer le serveur IIS.

Redémarrer votre serveur IIS par les commandes `net stop` et `net start`.


```
C:\> net stop was /Y
C:\> net start w3svc
```

Résultat

Le serveur IIS est configuré pour envoyer des demandes de clients à des serveurs spécifiques de JBoss EAP 6 que vous aurez configurés, sur une base spécifique à l'application.

[Rapporter un bogue](#)

17.9.5. Configurer le re-directionneur ISAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6

Aperçu

Cette configuration équilibre les requêtes des clients entre les serveurs de JBoss EAP 6 que vous spécifiez. Si vous préférez envoyer des demandes de client à des serveurs JBoss EAP 6 spécifiques sur une base «par-déploiement», reportez-vous plutôt à [Section 17.9.4, « Configurer le re-directionneur ISAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP 6 »](#).

Cette configuration est faite sur le serveur IIS, et assume que JBoss Enterprise Application Platform est déjà configurée, comme dans [Section 17.4.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »](#).

Conditions préalables

- Vous aurez besoin d'un accès administrateur pour accéder au serveur IIS.
- [Section 17.4.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »](#)
- [Section 17.9.3, « Configurer Microsoft IIS pour qu'il puisse utiliser ISAPI »](#)

Procédure 17.20. Équilibrage des requêtes de clients entre des serveurs multiples.

1. **Créer un répertoire pour stocker la journalisation, les fichiers de propriété, et les fichiers de verrouillage.**

Le reste de cette procédure suppose que vous utilisez le répertoire `C:\connectors\` à cet effet. Si vous utilisez un autre répertoire, modifier les instructions en conséquence.

2. **Créer le fichier `isapi_redirect.properties`.**

Créer un nouveau fichier intitulé `C:\connectors\isapi_redirect.properties`. Copier les contenus suivants dans le fichier.

```
# Configuration file for the ISAPI Redirector
# Extension uri definition
extension_uri=/jboss/isapi_redirect.dll

# Full path to the log file for the ISAPI Redirector
log_file=c:\connectors\isapi_redirect.log

# Log level (debug, info, warn, error or trace)
log_level=info

# Full path to the workers.properties file
```

```

worker_file=c:\connectors\workers.properties

# Full path to the uriworkermap.properties file
worker_mount_file=c:\connectors\uriworkermap.properties

#OPTIONAL: Full path to the rewrite.properties file
rewrite_rule_file=c:\connectors\rewrite.properties

```

Si vous ne souhaitez pas utiliser un fichier `rewrite.properties`, dé-commentez la dernière ligne en plaçant un caractère `#` au début de la ligne. Voir [Étape 5](#) pour plus d'informations.

3. Créer le fichier `uriworkermap.properties`.

Le fichier `uriworkermap.properties` contient les mappages entre les URL de l'application déployée et quel worker gère les demandes à leur intention. Le fichier exemple suivant illustre la syntaxe du fichier, avec une configuration d'équilibrage de charge. Le caractère générique (*) envoie toutes les requêtes de divers sous-répertoires d'URL vers l'équilibreur de charge nommé `router`. La configuration de l'équilibreur de charge est couverte dans [Étape 4](#).

Mettez votre fichier `uriworkermap.properties` dans `C:\connectors\`.

```

# images, css files, path /status and /web-console will be
# provided by nodes defined in the load-balancer called "router"
/css/*=router
/images/*=router
/status=router
/web-console|/*=router

# Example of exclusion from mapping, logo.gif won't be displayed
# /web-console/images/logo.gif=*

# Requests to /app-01 and /app-02 will be routed to nodes defined
# in the load-balancer called "router"
/app-01|/*=router
/app-02|/*=router

# mapping for management console, nodes in cluster can be enabled or
# disabled here
/jkmanager|/*=status

```

4. Créer le fichier `workers.properties`.

Le fichier `workers.properties` contient les définitions de mappage entre les étiquettes de workers et les instances de serveur. Le fichier exemple suivant illustre la syntaxe du fichier. L'équilibrage de charge est configuré vers la fin du fichier, et comprend les workers `worker01` et `worker02`. Le fichier `workers.properties` suit la syntaxe du même fichier que celui utilisé pour la configuration d'Apache `mod_jk`. Pour plus d'informations sur la syntaxe du fichier `workers.properties`, reportez-vous à [Section 17.7.5, « Référence de configuration pour les Apache Mod_jk Workers »](#).

Mettez ce fichier dans le répertoire `C:\connectors\`.

```

# The advanced router LB worker
worker.list=router,status

# First EAP server definition, port 8009 is standard port for AJP in
EAP

```

```

#
# lbfactor defines how much the worker will be used.
# The higher the number, the more requests are served
# lbfactor is useful when one machine is more powerful
# ping_mode=A - all possible probes will be used to determine that
# connections are still working

worker.worker01.port=8009
worker.worker01.host=127.0.0.1
worker.worker01.type=ajp13
worker.worker01.ping_mode=A
worker.worker01.socket_timeout=10
worker.worker01.lbfactor=3

# Second EAP server definition
worker.worker02.port=8009
worker.worker02.host=127.0.0.100
worker.worker02.type=ajp13
worker.worker02.ping_mode=A
worker.worker02.socket_timeout=10
worker.worker02.lbfactor=1

# Define the LB worker
worker.router.type=lb
worker.router.balance_workers=worker01,worker02

# Define the status worker for jkmanager
worker.status.type=status

```

5. Créer le fichier `rewrite.properties`.

Le fichier `rewrite.properties` contient des dispositions relatives aux demandes spécifiques de réécriture d'URL simple pour certaines applications. Le chemin d'accès de réécriture est spécifié à l'aide de paires nom / valeur, comme illustré dans l'exemple ci-dessous. Placez ce fichier dans le répertoire `C:\connectors\`.

```

#Simple example
# Images are accessible under abc path
/app-01/abc/=/app-01/images/

```

6. Redémarrer le serveur IIS.

Redémarrer votre serveur IIS par les commandes `net stop` et `net start`.

```

C:\> net stop was /Y
C:\> net start w3svc

```

Résultat

Le serveur IIS est configuré pour envoyer des demandes de clients à des serveurs de JBoss EAP 6 référencés dans le fichier `workers.properties`, équilibrant la charge équitablement à travers les serveurs dans un ratio 1:3. Ce ratio est dérivé du facteur d'équilibre des charges (`lbfactor`) assigné à chaque serveur.

[Rapporter un bogue](#)

17.10. ORACLE NSAPI

17.10.1. Netscape Server API (NSAPI)

Netscape Server API (NSAPI) est un API fourni par le serveur Oracle iPlanet Web Server (anciennement connu sous le nom de Netscape Web Server) pour implémenter des extensions au serveur. Ces extensions sont connues en tant que plugins de serveur. L'API est utilisé dans le `nsapi_redirector.so` fourni par JBoss EAP 6 dans les packages `Native utilities`. Pour configurer ce connecteur, voir [Section 17.10.4, « Configurer le re-directionneur NSAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6 »](#).

[Rapporter un bogue](#)

17.10.2. Configurer NSAPI dans Oracle Solaris

Résumé

Le re-directionneur NSAPI est un module qui exécute dans le serveur Oracle iPlanet Web Server.

Conditions préalables

- Votre serveur exécute Oracle Solaris 10 ou version supérieure, soit en architecture 32-bit ou 64-bit Intel, soit en architecture SPARC64.
- Oracle iPlanet Web Server 7.0.15 ou versions supérieures pour architectures Intel, ou 7.0.14 ou versions supérieures pour les architectures SPARC, est installé ou configuré, indépendamment du re-directionneur NSAPI.
- La plate-forme JBoss EAP 6 est installée et configurée sur chaque serveur qui servira en tant que worker. Voir [Section 17.4.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »](#).
- Le package JBoss Native Components ZIP peut être téléchargé à partir du portail clients à <https://access.redhat.com>.

Procédure 17.21. Extraire et Installer le re-directionneur NSAPI

1. Extraire le package JBoss Native Components.

Le reste de cette procédure assume que le package de Native Components est extrait d'un répertoire nommé `EAP_HOME`. Pour le reste de cette procédure, le répertoire `/opt/oracle/webserver7/config/` sera identifié comme `IPLANET_CONFIG`. Si votre répertoire de configuration Oracle iPlanet est différent, modifier la procédure en fonction.

2. Désactiver les mappages du servlet.

Ouvrir le fichier `IPLANET_CONFIG/default.web.xml` et chercher la section avec l'en-tête **Built In Server Mappings**. Désactiver les mappages pour les trois servlets suivantes, en les enveloppant entre lignes de commentaires XML (`<!--` et `-->`).

- défaut
- invoker
- jsp

L'exemple de configuration suivant montre les mappages désactivés.

```

<!-- ===== Built In Servlet Mappings ===== -->
<!-- The servlet mappings for the built in servlets defined above. -
->
<!-- The mapping for the default servlet -->
<!--servlet-mapping>
  <servlet-name>default</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping-->
<!-- The mapping for the invoker servlet -->
<!--servlet-mapping>
  <servlet-name>invoker</servlet-name>
  <url-pattern>/servlet/*</url-pattern>
</servlet-mapping-->
<!-- The mapping for the JSP servlet -->
<!--servlet-mapping>
  <servlet-name>jsp</servlet-name>
  <url-pattern>*.jsp</url-pattern>
</servlet-mapping-->

```

Sauvegarder et sortir du fichier.

3. Configurer iPlanet Web Server pour qu'il puisse charger le module de re-directionneur NSAPI.

Ajouter les lignes suivantes à la fin de ce fichier *IPLANET_CONFIG/magnus.conf*, en modifiant les chemins de fichiers pour qu'ils s'accordent avec votre configuration. Ces lignes définissent l'emplacement du module *nsapi_redirector.so*, ainsi que celle du fichier *workers.properties*, qui liste les noeuds de worker et leurs propriétés.

```

Init fn="load-modules" funcs="jk_init,jk_service"
shlib="EAP_HOME/modules/system/layers/base/native/lib/nsapi_redirector.so" shlib_flags="(global|now)"

Init fn="jk_init"
worker_file="IPLANET_CONFIG/connectors/workers.properties"
log_level="info" log_file="IPLANET_CONFIG/connectors/nsapi.log"
shm_file="IPLANET_CONFIG/connectors/tmp/jk_shm"

```

La configuration ci-dessus est basée sur une architecture 32-bit. Si vous utilisez 64-bit Solaris, changez le string *lib/nsapi_redirector.so* en *lib64/nsapi_redirector.so*.

Sauvegarder et sortir du fichier.

4. Configurer le re-directionneur NSAPI.

Vous pouvez configurer le re-directionneur NSAPI pour une configuration de base, avec aucun équilibrage des charges, ou une configuration d'équilibrage des charges. Choisissez l'une des options suivantes, après quoi votre configuration sera terminée

- o [Section 17.10.3, « Configurer le re-directionneur NSAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP 6 »](#)
- o [Section 17.10.4, « Configurer le re-directionneur NSAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6 »](#)

[Rapporter un bogue](#)

17.10.3. Configurer le re-directionneur NSAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP 6

Aperçu

Cette tâche configure le re-directionneur NSAPI à rediriger les demandes des clients aux serveurs JBoss EAP 6 sans aucun équilibrage de charge ou basculement. La redirection se fait sur la base d'un déploiement (est donc basé-URL). Pour une configuration d'équilibrage des charges, consultez [Section 17.10.4, « Configurer le re-directionneur NSAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6 »](#) à la place.

Conditions préalables

- Vous devez compléter [Section 17.10.2, « Configurer NSAPI dans Oracle Solaris »](#) avant de continuer avec la tâche suivante.

Procédure 17.22. Installer le connecteur HHP de base

1. Définir les chemins URL pour redirection vers les serveurs de la plate-forme JBoss EAP 6.



NOTE

Dans *IPLANET_CONFIG/obj.conf*, les espaces ne sont pas autorisés en début de ligne, sauf quand la ligne est en continuation de la ligne précédente.

Modifier le fichier *IPLANET_CONFIG/obj.conf*. Chercher la section qui commence par `<Object name="default">`, et ajouter chaque modèle d'URL à son correspondant, sous le format montré dans le fichier exemple ci-dessous. Le string `jknsapi` fait référence au connecteur HTTP qui sera défini dans la prochaine étape. L'exemple montre comment utiliser les caractères génériques pour la correspondance de modèles.

```
<Object name="default">
[... ]
NameTrans fn="assign-name" from="/status" name="jknsapi"
NameTrans fn="assign-name" from="/images(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/css(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/nc(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/jmx-console(|/*)" name="jknsapi"
</Object>
```

2. Définir le worker qui sert chaque chemin d'accès.

Continuer à modifier le fichier *IPLANET_CONFIG/obj.conf*. Ajouter ce qui suit directement après la balise de fermeture de la section que vous venez de finir d'éditer : `</Object>`.

```
<Object name="jknsapi">
ObjectType fn=force-type type=text/plain
Service fn="jk_service" worker="worker01" path="/status"
Service fn="jk_service" worker="worker02" path="/nc(|/*)"
Service fn="jk_service" worker="worker01"
</Object>
```

L'exemple ci-dessus redirige les requêtes vers le chemin URL `/status` et le worker nommé `worker01`, et tous les URL en-dessous `/nc/` vers le worker `worker02`. La troisième ligne indique que tous les URL assignés à l'objet `jknsapi` qui n'ont pas de correspondance avec les

lignes précédentes sont servis à **worker01**.

Sauvegarder et sortir du fichier.

3. Définir les workers et leurs attributs.

Créer un fichier intitulé **workers.properties** dans le répertoire **IPLANET_CONFIG/connectors/**. Coller les commentaires suivants dans le fichier, et les modifier pour qu'il conviennent à votre environnement.

```
# An entry that lists all the workers defined
worker.list=worker01, worker02

# Entries that define the host and port associated with these
workers
worker.worker01.host=127.0.0.1
worker.worker01.port=8009
worker.worker01.type=ajp13

worker.worker02.host=127.0.0.100
worker.worker02.port=8009
worker.worker02.type=ajp13
```

Le fichier **workers.properties** utilise la même syntaxe qu'Apache **mod_jk**. Pour obtenir plus d'informations sur les options disponibles, voir [Section 17.7.5, « Référence de configuration pour les Apache Mod_jk Workers »](#).

Sauvegarder et sortir du fichier.

4. Redémarrer le serveur iPlanet Web Server.

Lancer les commandes suivantes pour redémarrer le serveur iPlanet Web Server.

```
IPLANET_CONFIG/../../bin/stopserv
IPLANET_CONFIG/../../bin/startserv
```

Résultat

iPlanet Web Server envoie maintenant les requêtes clients aux URL que vous avez configurés vers les déploiements de JBoss EAP 6.

[Rapporter un bogue](#)

17.10.4. Configurer le re-directionneur NSAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6

Aperçu

Cette tâche configure le re-directionneur NSAPI à envoyer les demandes des clients aux serveurs JBoss EAP 6 dans une configuration d'équilibrage des charges. Pour utiliser NSAPI comme simple connecteur HTTP sans équilibrage des charges, voir [Section 17.10.3, « Configurer le re-directionneur NSAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP 6 »](#).

Conditions préalables

- [Section 17.10.2, « Configurer NSAPI dans Oracle Solaris »](#)

Prochaines étapes : [Section 17.10.5, « Configurer le re-directionneur NSAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6 »](#)

Procédure 17.23. Configurer le connecteur pour l'équilibrage des charges**1. Définir les chemins URL pour redirection vers les serveurs de la plate-forme JBoss EAP 6.****NOTE**

Dans *IPLANET_CONFIG/obj.conf*, les espaces ne sont pas autorisés en début de ligne, sauf quand la ligne est en continuation de la ligne précédente.

Modifier le fichier *IPLANET_CONFIG/obj.conf*. Chercher la section qui commence par `<Object name="default">`, et ajouter chaque modèle d'URL à son correspondant, sous le format montré dans le fichier exemple ci-dessous. Le string `jknsapi` fait référence au connecteur HTTP qui sera défini dans la prochaine étape. L'exemple montre comment utiliser les caractères génériques pour la correspondance de modèles.

```
<Object name="default">
[...]
NameTrans fn="assign-name" from="/status" name="jknsapi"
NameTrans fn="assign-name" from="/images(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/css(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/nc(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/jmx-console(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/jkmanager/*" name="jknsapi"
</Object>
```

2. Définir le worker qui sert chaque chemin d'accès.

Continuer à modifier le fichier *IPLANET_CONFIG/obj.conf*. Ajouter ce qui suit directement après la balise de fermeture de la section que vous venez de finir d'éditer dans l'étape précédente (`</Object>`), et modifiez là suivant vos besoins :

```
<Object name="jknsapi">
ObjectType fn=force-type type=text/plain
Service fn="jk_service" worker="status" path="/jkmanager(/*)"
Service fn="jk_service" worker="router"
</Object>
```

Cet objet `jknsapi` définit les noeuds de worker utilisés pour servir chaque chemin relié au mappage `name="jknsapi"` de l'objet `default`. Tout sauf les URL correspondant à `/jkmanager/*` sont redirigés vers le worker nommé `router`.

3. Définir les workers et leurs attributs.

Créer un fichier intitulé `workers.properties` dans le répertoire *IPLANET_CONFIG/connector/*. Coller les commentaires suivants dans le fichier, et les modifier pour qu'il conviennent à votre environnement.

```
# The advanced router LB worker
# A list of each worker
worker.list=router,status

# First JBoss EAP server
# (worker node) definition.
# Port 8009 is the standard port for AJP
#
```



```
worker.worker01.port=8009
worker.worker01.host=127.0.0.1
worker.worker01.type=ajp13
worker.worker01.ping_mode=A
worker.worker01.socket_timeout=10
worker.worker01.lbfactor=3

# Second JBoss EAP server
worker.worker02.port=8009
worker.worker02.host=127.0.0.100
worker.worker02.type=ajp13
worker.worker02.ping_mode=A
worker.worker02.socket_timeout=10
worker.worker02.lbfactor=1

# Define the load-balancer called "router"
worker.router.type=lb
worker.router.balance_workers=worker01,worker02

# Define the status worker
worker.status.type=status
```

Le fichier `workers.properties` utilise la même syntaxe qu'Apache `mod_jk`. Pour obtenir plus d'informations sur les options disponibles, voir [Section 17.7.5, « Référence de configuration pour les Apache Mod_jk Workers »](#).

Sauvegarder et sortir du fichier.

4. Redémarrer le serveur iPlanet Web Server 7.0.

```
IPLANET_CONFIG/../../bin/stopserv
IPLANET_CONFIG/../../bin/startserv
```

Résultat

iPlanet Web Server redirige les modèles d'URL que vous avez configurés à vos serveurs JBoss EAP 6 dans une configuration d'équilibrage des charges.

[Rapporter un bogue](#)

CHAPITRE 18. MESSAGERIE

18.1. INTRODUCTION

18.1.1. HornetQ

HornetQ est un système de messagerie multiprotocole, asynchrone développé par Red Hat. HornetQ qui procure une haute disponibilité (HA) avec basculement automatique des clients pour garantir la fiabilité du message dans le cas d'une panne de serveur. HornetQ prend également en charge des solutions de clustering flexibles avec équilibrage de charge des messages.

HornetQ est le fournisseur JMS (Java Message Service) de JBoss EAP 6 et est configuré comme **Messaging Subsystem**

[Rapporter un bogue](#)

18.1.2. Java Messaging Service (JMS)

Les systèmes de messagerie vous permettent de coupler de façon informelle des systèmes hétérogènes avec une fiabilité supplémentaire. Les fournisseurs de Service JMS (Java Messaging) utilisent un système de transactions pour valider ou annuler les modifications atomiquement. Contrairement aux systèmes basés sur un modèle d'échange de données informatisé (RPC, Remote Procedure Call), les systèmes de messagerie utilisent principalement un modèle de passage de messages asynchrone avec aucune relation réelle entre les demandes et les réponses. La plupart des systèmes de messagerie supportent également un mode de requête-réponse, mais ce n'est pas une caractéristique principale des systèmes de messagerie.

Les systèmes de messagerie découplent les expéditeurs des messages des consommateurs de messages. Les expéditeurs et les consommateurs de messages sont complètement indépendants et ne savent rien l'un de l'autre. Cela vous permet de créer des systèmes flexibles et faiblement couplés. Souvent, les grandes entreprises utilisent un système de messagerie pour mettre en place un bus de messages qui couple légèrement des systèmes hétérogènes. Les bus de messages forment souvent la base d'une Enterprise Service Bus (ESB). En utilisant un Message Bus pour découpler des systèmes disparates, on peut permettre au système de croître et de s'adapter plus facilement. Cela permet également plus de souplesse pour ajouter de nouveaux systèmes ou en retirer des "anciens", puisqu'ils n'ont pas de dépendances fragiles les uns avec les autres.

[Rapporter un bogue](#)

18.1.3. Styles de messagerie pris en compte

HornetQ prend en charge les styles de messagerie suivants :

Modèle de file d'attente de messages

Le modèle de file d'attente de message consiste à envoyer un message à une file d'attente. Une fois dans la file d'attente, le message est normalement rendu persistant pour en garantir sa livraison. Une fois que le message s'est déplacé dans la file, le système de messagerie le livre à un consommateur de messages. Le consommateur de messages accuse réception de la livraison du message une fois qu'il a été traité.

En messagerie PPP, le modèle de file d'attente de messagerie autorise plusieurs consommateurs pour une même file d'attente, mais chaque message ne peut être reçu que par un seul consommateur.

Modèle Publish-Subscribe

Le modèle Publish-Subscribe permet à plusieurs émetteurs d'envoyer des messages vers une seule entité sur le serveur. Cette entité est connue sous le nom de "topic". Chaque topic peut être traité par plusieurs consommateurs, ce que l'on appelle des "abonnements" (ou "subscriptions" en anglais)

Chaque abonnement reçoit une copie des messages envoyés au topic. La différence avec le modèle de file d'attente de messages, c'est que chaque message n'est consommé que par un seul consommateur.

Les abonnements qui sont durables conservent une copie de chaque message envoyé à ce topic jusqu'à ce que l'abonné les consomme. Ces copies sont conservées même en cas d'un redémarrage du serveur. Les abonnements non durables durent le temps de la durée de la connexion qui les a créés.

[Rapporter un bogue](#)

18.2. CONFIGURATION DES TRANSPORTS

18.2.1. Accepteurs et connecteurs

HornetQ utilise le concept de connecteurs et d'accepteurs comme un élément clé du système de messagerie.

Accepteurs et connecteurs

Acceptor

Un accepteur définit quels types de connections sont acceptées par le serveur HornetQ.

Connector

Un connecteur définit comment se connecter au serveur HornetQ, et est utilisé par le client HornetQ.

Il y a deux sortes de connecteurs et d'accepteurs, suivant que le connecteur ou l'accepteur qui correspond sont dans la même JVM ou non.

Invm et Netty

Invm

Invm est un acronyme pour Intra Virtual Machine. Peut être utilisé quand le client et le serveur exécutent en même temps dans la même JVM.

Netty

Le nom d'un projet JBoss. Doit être utilisé quand le client et le serveur exécutent dans des JVM différentes.

Un client HornetQ doit utiliser un connecteur compatible avec un des accepteurs du serveur. Seul un connecteur Invm peut se connecter à un accepteur Invm, et seul un connecteur netty peut se connecter à un accepteur de netty. Les connecteurs et les accepteurs sont configurés sur le serveur

dans un `standalone.xml` et `domain.xml`. Vous pouvez utiliser la console de gestion ou l'interface CLI pour les définir.

[Rapporter un bogue](#)

18.2.2. Configuration de Netty TCP

Netty TCP est un simple transport de sockets basées TCP non chiffré. Netty TCP peut être configuré pour utiliser l'ancien blocage Java IO ou Java NIO non bloquant. Java NIO est recommandé pour une meilleure évolutivité avec un grand nombre de connexions simultanées sur le serveur. Si le nombre de connexions simultanées est moindre, l'ancien Java IO peut donner une meilleure latence que NIO.

Netty TCP n'est pas conseillé pour les connexions défilant sur un réseau non sécurisé car il est encodé. Avec le transport Netty TCP, toutes les connexions sont initiées côté client.

Exemple 18.1. Exemple de configuration de Netty TCP à partir de la configuration EAP par défaut

```
<connectors>
  <netty-connector name="netty" socket-binding="messaging"/>
  <netty-connector name="netty-throughput" socket-binding="messaging-
throughput">
    <param key="batch-delay" value="50"/>
  </netty-connector>
  <in-vm-connector name="in-vm" server-id="0"/>
</connectors>
<acceptors>
  <netty-acceptor name="netty" socket-binding="messaging"/>
  <netty-acceptor name="netty-throughput" socket-binding="messaging-
throughput">
    <param key="batch-delay" value="50"/>
    <param key="direct-deliver" value="false"/>
  </netty-acceptor>
  <in-vm-acceptor name="in-vm" server-id="0"/>
</acceptors>
```

La configuration de l'exemple montre également comment l'application JBoss EAP 6 de HornetQ utilise des liaisons de socket dans la configuration du connecteur et l'accepteur. Cela diffère de la version autonome de HornetQ, qui vous oblige à déclarer les ports et les hôtes spécifiques.

Le tableau suivant décrit les propriétés de configuration de Netty TCP :

Tableau 18.1. Propriétés de configuration de Netty TCP

Property	Par défaut	Description
batch-delay	0 millisecondes	Avant d'inscrire les paquets au transport, HornetQ peut être configuré pour regrouper les écritures pour un maximum de millisecondes de batch-delay. Cela augmentera le débit total pour les petits messages en augmentant la latence de transfert de messages.

Property	Par défaut	Description
direct-deliver	true	Lorsqu'un message arrive sur le serveur et est livré aux consommateurs qui attendent, par défaut, la livraison se fait sur le même thread que celui sur lequel le message est arrivé. Cela donne une bonne latence dans des environnements avec des messages relativement peu volumineux et un petit nombre de consommateurs, mais réduit le débit et la latence. Pour un débit plus élevé, vous pouvez définir cette propriété à « false »
local-address	[local address available]	Pour un connecteur netty, c'est utilisé pour indiquer l'adresse locale que le client va utiliser quand il se connectera à l'adresse distante. Si une adresse locale n'est pas spécifiée, le connecteur utilisera n'importe quelle adresse locale disponible.
local-port	0	Pour un connecteur netty, c'est utilisé pour indiquer le port local que le client va utiliser quand il se connectera à l'adresse distante. Si le port local par défaut est utilisé (0), le connecteur laissera le système collecter un port éphémère. Les ports valides sont 0 à 65535
nio-remoting-threads	-1	Si configuré pour utiliser NIO, HornetQ, par défaut, utilise un nombre de threads égal à trois fois le nombre de noyaux (ou hyper-threads) rapporté par <code>Runtime.getRuntime().availableProcessors()</code> pour le traitement des paquets entrants. Pour substituer cette valeur, vous pouvez définir une valeur personnalisée pour le nombre de threads
tcp-no-delay	true	Si sur true, alors l'algorithme sera activé. Cet algorithme aide à améliorer l'efficacité des réseaux TCP/IP en réduisant le nombre de paquets à envoyer sur le réseau

Property	Par défaut	Description
tcp-send-buffer-size	32768 bytes	Ce paramètre détermine la taille du tampon d'envoi TCP en octets
tcp-receive-buffer-size	32768 bytes	Ce paramètre détermine la taille du tampon de réception TCP en octets
use-nio	false	Si cela est sur true, alors les NIO Java non bloquantes seront utilisées. Si sur false, alors les anciennes e/s non bloquantes de Java seront utilisées. Si vous avez besoin que le serveur utilise plusieurs connexions concourantes, utiliser les NIO Java non bloquantes, sinon, choisissez les anciennes e/s (bloquantes)

**NOTE**

Les propriétés de Netty TCP sont valides pour tous les types de transport (Netty SSL, Netty HTTP et Netty Servlet).

[Rapporter un bogue](#)

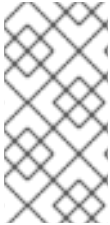
18.2.3. Configuration de Netty Secure Sockets Layer (SSL)

Netty TCP est un simple transport de sockets basé TCP non chiffré. Netty ressemble à Netty TCP mais il procure une sécurité supplémentaire en chiffrant les connexions TCP par SSL (Secure Sockets Layer).

**AVERTISSEMENT**

Utiliser le protocole SSLv3 pour les connecteurs/accepteurs SSL n'était pas autorisé à cause de la vulnérabilité "Poodle". Les clients JMS se connectant à ce protocole ne seront pas acceptés.

L'exemple suivant montre la configuration Netty pour SSL One Way :

**NOTE**

La plupart des paramètres suivants peuvent être utilisés avec des accepteurs comme connecteurs. Toutefois, certains paramètres fonctionnent uniquement avec les accepteurs. La description du paramètre explique la différence entre l'utilisation de ces paramètres dans des connecteurs ou des accepteurs.

```
<acceptors>
  <netty-acceptor name="netty" socket-binding="messaging"/>
    <param key="ssl-enabled" value="true"/>
    <param key="key-store-password" value="[keystore password]"/>
    <param key="key-store-path" value="[path to keystore file]"/>
  </netty-acceptor>
</acceptors>
```

Tableau 18.2. Propriétés de configuration de Netty SSL

Nom de propriété	Par défaut	Description
ssl-enabled	true	Active SSL
key-store-password	[keystore password]	Lorsqu'il est utilisé sur un accepteur, c'est le mot de passe du keystore côté serveur. Lorsqu'il est utilisé sur un connecteur, c'est le mot de passe du keystore côté client. C'est pertinent pour un connecteur si vous utilisez SSL Two Ways (authentification dans les deux sens). Cette valeur peut être configurée sur le serveur, mais elle sera téléchargée et utilisée par le client.
key-store-path	[path to keystore file]	Lorsqu'il est utilisé sur un accepteur, c'est le chemin du keystore SSL côté serveur qui détient les certificats du serveur (auto-signés ou signés par une autorité). Lorsqu'il est utilisé sur un connecteur, c'est le chemin du keystore SSL côté client qui détient les certificats des clients. C'est pertinent pour un connecteur si vous utilisez SSL Two Ways (authentification dans les deux sens). Cette valeur est configurée sur le serveur; elle sera téléchargée et utilisée par le client.

Si vous configurez Netty en SSL Two Ways (authentification mutuelle entre serveur et client), il y a trois paramètres supplémentaires en plus de ceux décrits dans l'exemple ci-dessus pour SSL One Way :

- ***need-client-auth*** : indique le besoin de l'authentification Two Way (dans les deux sens) pour les connexions client.
- ***trust-store-password*** : lorsqu'il est utilisé sur un accepteur, c'est le mot de passe du trust store côté serveur. Lorsqu'il est utilisé sur un connecteur, c'est le mot de passe du trust store côté client. C'est pertinent pour un connecteur à la fois en SSL One Way et SSL Two Ways. Cette valeur peut être configurée sur le serveur, mais elle sera téléchargée et utilisée par le client
- ***trust-store-path*** : lorsqu'il est utilisé sur un accepteur, c'est le chemin du trust store SSL côté serveur qui détient les clés de tous les clients à qui le serveur fait confiance. Lorsqu'il est utilisé sur un connecteur, c'est le chemin du keystore SSL côté client qui détient les clés publiques de tous les serveurs à qui le client fait confiance. C'est important pour un connecteur si vous utilisez SSL Two Ways ou One Way. Ce chemin peut être configuré sur le serveur, mais sera téléchargé et utilisé par le client.

[Rapporter un bogue](#)

18.2.4. Configuration de Netty HTTP

Netty HTTP conduit des paquets sur le protocole HTTP. Il peut être utile dans les scénarios où les pare-feux permettent uniquement le trafic HTTP. Netty HTTP utilise les mêmes propriétés que Netty TCP, ainsi que les quelques propriétés supplémentaires suivantes :



NOTE

Les paramètres suivants peuvent être utilisés par les accepteurs ainsi que par les connecteurs. Le transport Netty HTTP ne permet pas la réutilisation du standard HTTP port (8080 par défaut). L'utilisation du port HTTP standard entraîne une exception. Vous pouvez utiliser [Section 18.2.5, « Configuration de Netty Servlet »](#) (Netty Servlet Transport) pour le tunneling des connexions HornetQ via un port HTTP standard.

```
<socket-binding name="messaging-http" port="7080" />
```

```
<acceptors>
  <netty-acceptor name="netty" socket-binding="messaging-http">
    <param key="http-enabled" value="false"/>
    <param key="http-client-idle-time" value="500"/>
    <param key="http-client-idle-scan-period" value="500"/>
    <param key="http-response-time" value="10000"/>
    <param key="http-server-scan-period" value="5000"/>
    <param key="http-requires-session-id" value="false"/>
  </netty-acceptor>
</acceptors>
```

Le tableau suivant décrit les propriétés supplémentaires de configuration de Netty HTTP :

Tableau 18.3. Propriétés de configuration de Netty HTTP

Nom de propriété	Par défaut	Description
http-enabled	false	Si défini à true, HTTP est activé

Nom de propriété	Par défaut	Description
http-client-idle-time	500 millisecondes	La durée pendant laquelle un client peut être inactif avant d'envoyer une demande de connexion HTTP vide pour conserver la connexion active
http-client-idle-scan-period	500 millisecondes	La fréquence (en ms) de balayage des clients inactifs
http-response-time	10000 millisecondes	La durée pendant laquelle le serveur peut patienter avant d'envoyer une réponse HTTP vide pour conserver la connexion active
http-server-scan-period	5000 millisecondes	La fréquence, en millisecondes, de balayage des clients en attente de réponses
http-requires-session-id	false	Si défini sur true, le client devra attendre après le premier appel avant de recevoir une ID de session



AVERTISSEMENT

Failover automatique non pris en charge pour les clients se connectant via un transport Netty HTTP.

[Rapporter un bogue](#)

18.2.5. Configuration de Netty Servlet

Le transport de servlet autorise le trafic HornetQ via HTTP à un servlet exécutant dans un moteur de servlet qui le redirige ensuite vers un serveur de HornetQ in-VM. Le transport Netty HTTP agit comme un serveur web à l'écoute du trafic HTTP sur des ports spécifiques. Avec le transport de servlet, le trafic HornetQ est mandaté par un moteur de servlet qui puisse déjà servir à un site web ou à d'autres applications.

Pour pouvoir configurer un moteur de servlet qui puisse fonctionner sur un transport de servlet Netty, vous devrez suivre les étapes suivantes :

- Déployer le servlet : l'exemple suivant décrit une application web qui utilise le servlet :

```
<web-app>
```

```

<servlet>
  <servlet-name>HornetQServlet</servlet-name>
  <servlet-
class>org.jboss.netty.channel.socket.http.HttpTunnelingServlet</serv
let-class>
  <init-param>
    <param-name>endpoint</param-name>
    <param-value>local:org.hornetq</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>HornetQServlet</servlet-name>
  <url-pattern>/HornetQServlet</url-pattern>
</servlet-mapping>
</web-app>

```

Le paramètre init **endpoint** spécifie l'attribut d'hôte de l'accepteur Netty à qui le servlet va envoyer ses packages.

- Insérer l'accepteur de Netty servlet sur la configuration côté serveur : l'exemple suivant montre la définition d'un accepteur en serveur de fichiers de configuration (**standalone.xml** et **domain.xml**) :

```

<acceptors>
  <acceptor name="netty-servlet">
    <factory-class>
      org.hornetq.core.remoting.impl.netty.NettyAcceptorFactory
    </factory-class>
    <param key="use-servlet" value="true"/>
    <param key="host" value="org.hornetq"/>
  </acceptor>
</acceptors>

```

- La dernière étape consiste à définir un connecteur pour le client dans les fichiers de configuration du serveur (**standalone.xml** et **domain.xml**) :

```

<netty-connector name="netty-servlet" socket-binding="http">
  <param key="use-servlet" value="true"/>
  <param key="servlet-path" value="/messaging/HornetQServlet"/>
</netty-connector>

```

- Il est également possible d'utiliser le transport de servlet via SSL en ajoutant la configuration suivante au connecteur :

```

<netty-connector name="netty-servlet" socket-binding="https">
  <param key="use-servlet" value="true"/>
  <param key="servlet-path" value="/messaging/HornetQServlet"/>
  <param key="ssl-enabled" value="true"/>
  <param key="key-store-path" value="path to a key-store"/>
  <param key="key-store-password" value="key-store password"/>
</netty-connector>

```

**AVERTISSEMENT**

Failover automatique non pris en charge pour les clients se connectant via HTTP tunneling servlet.

**NOTE**

Le servlet Netty ne peut pas être utilisé pour configurer les serveurs EAP 6 pour mettre en place un cluster HornetQ.

[Rapporter un bogue](#)

18.3. JNDI (JAVA NAMING AND DIRECTORY INTERFACE)

L'API *Java Naming and Directory Interface (JNDI)* est un API standard Java pour les services de répertoire et de nommage. Il permet aux technologies basées-Java de découvrir ou d'organiser des composants de noms dans un environnement informatique distribué.

[Rapporter un bogue](#)

18.4. DÉTECTION DE CONNEXION MORTE

18.4.1. Fermer les ressources de connexions mortes

Un noyau HornetQ ou une application cliente JMS doivent fermer leurs ressources avant de s'arrêter. Vous pouvez configurer votre application de façon à ce qu'elle ferme ses ressources automatiquement en utilisant le bloc `finally` dans le code de l'application.

L'exemple suivant montre une application cliente qui ferme sa session et son usine de sessions dans le bloc `finally`:

```
ServerLocator locator = null;
ClientSessionFactory sf = null;
ClientSession session = null;

try
{
    locator = HornetQClient.createServerLocatorWithoutHA(..);

    sf = locator.createClientSessionFactory();

    session = sf.createSession(...);

    ... do some operations with the session...
}
finally
{
    if (session != null)
```

```

    {
        session.close();
    }

    if (sf != null)
    {
        sf.close();
    }

    if(locator != null)
    {
        locator.close();
    }
}

```

L'exemple suivant montre une application cliente JMS qui ferme sa session et son usine de sessions dans le bloc **finally** :

```

Connection jmsConnection = null;

try
{
    ConnectionFactory jmsConnectionFactory =
HornetQJMSClient.createConnectionFactoryWithoutHA(...);

    jmsConnection = jmsConnectionFactory.createConnection();

    ... do some operations with the connection...
}
finally
{
    if (connection != null)
    {
        connection.close();
    }
}

```

Utilisation du paramètre TTL (Connection Time to Live)

Le paramètre *connection-ttl* détermine la période pendant laquelle le serveur conserve la connexion vivante lorsqu'il ne reçoit pas de données, ni de paquets de ping du client. Ce paramètre garantit que les ressources de serveur mortes, comme les anciennes sessions, soient soutenues plus longtemps, permettant ainsi aux clients de se reconnecter lorsque qu'une connexion de réseau interrompue est ravivée.

Vous pouvez définir la connexion TTL pour les clients JMS en indiquant le paramètre *connexion-ttl* dans l'instance `HornetQConnectionFactory`. Si vous déployez des instances de fabrication de connexions JMS directement dans JNDI, vous pouvez définir le paramètre *connection-ttl* dans les fichiers de configuration de serveur `standalone.xml` et `domain.xml`.

La valeur par défaut du paramètre *connection-ttl* est 60000 millisecondes. Si vous n'avez pas besoin de clients pour spécifier leur propre connexion TTL ; vous pouvez définir le paramètre *connection-ttl-override* dans les fichiers de configuration du serveur pour remplacer toutes les valeurs. Le paramètre *connection-ttl-override* est désactivée par défaut et a une valeur de -1.

Garbage Collection

HornetQ utilise garbage collection pour détecter et fermer les sessions qui ne sont pas explicitement fermées dans un bloc `finally`. Le serveur HornetQ consigne un avertissement semblable à l'avertissement ci-dessous avant de fermer les sessions :

```
[Finalizer] 20:14:43,244 WARNING
[org.hornetq.core.client.impl.DelegatingSession] I'm closing a
ClientSession you left open. Please make sure you close all ClientSessions
explicitly before letting them go out of scope!
[Finalizer] 20:14:43,244 WARNING
[org.hornetq.core.client.impl.DelegatingSession] The session you didn't
close was created here:
java.lang.Exception
    at
org.hornetq.core.client.impl.DelegatingSession.<init>(DelegatingSession.java:83)
    at org.acme.yourproject.YourClass (YourClass.java:666)
```

Le message de journal contient l'information sur la partie de code où la connexion JMS ou la session utilisateur ont été créées et n'ont pas été fermées par la suite.

[Rapporter un bogue](#)

18.4.2. Détection des échecs côté-client

L'application du client envoie automatiquement des paquets de pings au serveur pour éviter que le client échoue. De la même manière, l'application du client considère que la connexion est vivante tant qu'elle reçoit des données du serveur.

Si le client ne reçoit pas de paquets de données du serveur pour une période de temps spécifiée par le paramètre *client-failure-check-period*, alors le client considère que la connexion a échoué. Ensuite, le client initie un basculement ou appelle les instances `FailureListener`.

Pour les clients JMS, « *client-failure-check-period* » est configuré à l'aide de l'attribut `ClientFailureCheckPeriod` sur l'instance `HornetQConnectionFactory`. Si vous déployez des instances d'usine de connexions JMS directement dans JNDI sur le serveur, vous pouvez spécifier le paramètre *client-failure-check-period* dans les fichiers de configuration de serveur `standalone.xml` et `domain.xml`.

La valeur par défaut de « *client-failure-check-period* » est de 3 000 millisecondes. Une valeur de -1 signifie que le client ne fermera jamais la connexion, si aucune donnée n'est reçue du serveur. La valeur de « *client-failure-check-period* » est beaucoup plus faible que le TTL de connexion afin que les clients puissent se reconnecter en cas d'un échec de transition.

Configuration de l'exécution de connexions asynchrones

Par défaut, les paquets reçus sur le serveur sont exécutés sur le thread d'accès distant. Il est possible de libérer le thread de l'accès distant en procédant à des opérations asynchrones sur n'importe quel thread du pool de threads. Vous pouvez configurer une exécution de connexion asynchrone grâce au paramètre *async-connection-execution-enabled* dans les fichiers de configuration de serveur `standalone.xml` et `domain.xml`. La valeur par défaut de ce paramètre est « `true` ».

**NOTE**

Si vous traitez les opérations de façon asynchrone sur n'importe quel thread du pool de threads, cela va ajouter un petit temps de latence. Les opérations en cours d'exécution sont toujours gérées sur le thread d'accès distant pour des raisons de performance.

[Rapporter un bogue](#)

18.5. TRAVAILLER AVEC DES MESSAGES VOLUMINEUX

18.5.1. Travailler avec des messages volumineux

HornetQ prend en charge l'utilisation de messages volumineux, même lorsque le client ou le serveur a limité la quantité de mémoire. Les messages volumineux peuvent être traités tels quels, ou comprimés davantage pour un transfert plus efficace. Un utilisateur peut envoyer un message volumineux en définissant un `InputStream` dans le corps du message. Lorsque le message est envoyé, HornetQ lit ce `InputStream` et transmet les données au serveur par fragments.

Le client et le serveur ne stockent jamais le corps complet d'un message volumineux en mémoire. Le consommateur reçoit initialement un message volumineux avec un corps vide et affecte par la suite un `OutputStream` au message pour obtenir des fragments dans un fichier disque.

[Rapporter un bogue](#)

18.5.2. Configurer des messages volumineux d'HornetQ

Configurer le serveur

En mode autonome, les messages volumineux sont stockés dans le répertoire `EAP_HOME/standalone/data/largemessages`. En mode domaine, les messages volumineux sont stockés dans le répertoire `EAP_HOME/domain/servers/SERVERNAME/data/largemessages`. La propriété de configuration `large-messages-directory` indique l'endroit où les messages volumineux sont stockés.

**IMPORTANT**

Pour une meilleure performance, nous vous recommandons de stocker le répertoire de messages volumineux sur un volume physique différent du journal de messages ou sur le répertoire de pagination.

[Rapporter un bogue](#)

18.5.3. Configurer les paramètres

Vous pouvez configurer des messages volumineux d'HornetQ en définissant les divers paramètres suivants :

Avec Hornet Core API côté client

Si vous utilisez HornetQ Core API côté client, vous devez définir le paramètre `ServerLocator.setMinLargeMessageSize` pour spécifier la taille minimale des messages volumineux. La taille minimale de messages (`min-large-message-size`) a la valeur 100KiB par défaut.

■

```

ServerLocator locator = HornetQClient.createServerLocatorWithoutHA(new
TransportConfiguration(NettyConnectorFactory.class.getName()))

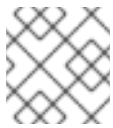
locator.setMinLargeMessageSize(25 * 1024);

ClientSessionFactory factory =
HornetQClient.createClientSessionFactory();

```

Configurer le serveur pour les clients JMS (Java Messaging Service)

Si vous utilisez Java Messaging Service (JMS), vous devez spécifier la taille minimale des messages volumineux avec l'attribut `min-large-message-size` de vos fichiers de configuration de serveur (`standalone.xml` et `domain.xml`). La taille minimale de messages (`min-large-message-size`) a la valeur 100KiB par défaut.



NOTE

La valeur de l'attribut `min-large-message-size` doit être en octets

Vous pouvez choisir de compresser les messages volumineux pour un transfert rapide et efficace. Toutes les opérations de compression/de-compression sont gérées sur le côté client. Si le message compressé est plus petit que `min-large-message-size`, il sera envoyé au serveur comme un message ordinaire. À l'aide de Java Messaging Service (JMS), vous pouvez compresser des messages volumineux en définissant la propriété booléenne de `compress-large-messages` à "true" sur l'indice de serveur ou `ConnectionFactory`.

```

<connection-factory name="ConnectionFactory">
  <connectors>
    <connector-ref connector-name="netty"/>
  </connectors>
  ...
  <min-large-message-size>204800</min-large-message-size>
  <compress-large-messages>true</compress-large-messages>
</connection-factory>

```

[Rapporter un bogue](#)

18.6. PAGINATION

18.6.1. La pagination

HornetQ supporte plusieurs files d'attente, comprenant chacune des millions de messages. Le serveur HornetQ exécute avec une mémoire limitée, ce qui rend le stockage de toutes les files d'attente de messages en mémoire à la fois difficile.

La pagination est un mécanisme utilisé par le serveur HornetQ pour paginer les messages vers et en provenance de la mémoire selon les besoins afin d'accommoder les files d'attente volumineuses dans un espace mémoire limité.

HornetQ commence par paginer les messages dans des disques quand la taille des messages en mémoire d'une adresse particulière commence à dépasser la taille de message configurée maximum.



NOTE

La pagination HornetQ est activée par défaut.

[Rapporter un bogue](#)

18.6.2. Les fichiers de pagination

Il y a un dossier individuel pour chaque adresse sur le système de fichiers qui stocke les messages dans plusieurs fichiers. Ces fichiers qui stockent les messages sont appelés fichiers de pagination. Chaque fichier contient des messages avec une taille maximale de message configurée (**page-size-bytes**).

Le système navigue dans les fichiers de pagination selon les besoins, et supprime les fichiers de pagination dès que tous les messages figurant sur la page sont reçus par le client.

[Rapporter un bogue](#)

18.6.3. Configuration d'un dossier de pagination

Des paramètres de pagination globaux sont spécifiés dans les fichiers de configuration du serveur (`standalone.xml` et `domain.xml`). Vous pourrez configurer l'emplacement du répertoire/dossier de pagination en utilisant le paramètre *paging-directory*.

```
<hornetq-server>
  ...
  <paging-directory>/location/paging-directory</paging-directory>
  ...
</hornetq-server>
```

Le paramètre *paging-directory* est utilisé pour spécifier un emplacement/dossier où stocker les fichiers de pagination. HornetQ crée un dossier pour chaque adresse de pagination dans le répertoire de pagination. Les fichiers de pagination sont stockés dans ces dossiers.

Le répertoire de pagination par défaut est `EAP_HOME/standalone/data/messagingpaging` (standalone mode) et `EAP_HOME/domain/servers/SERVERNAME/data/messagingpaging` (en mode de domaine).

[Rapporter un bogue](#)

18.6.4. Mode de pagination

Quand les messages qui sont envoyés à une adresse dépassent la taille configurée, cette adresse va en "page/paging mode".



NOTE

La pagination est faite individuellement par adresse. Si vous configurez un *max-size-bytes* d'adresse, cela signifie que chaque adresse correspondante aura une taille maximum que vous aurez spécifiée. Cependant, cela ne veut pas dire que la taille de toutes les adresses correspondantes soit limitée à *max-size-bytes*,

Même en mode *page*, le serveur peut se bloquer en raison d'une erreur d'insuffisance de mémoire. HornetQ conserve une référence à chaque fichier d'échange sur le disque. Dans une situation où on a des millions de fichiers d'échange, HornetQ peut faire face à un épuisement de la mémoire. Pour minimiser ce risque, il est important de définir l'attribut *page-size-bytes* à une valeur appropriée. Vous devez configurer une mémoire de serveur JBoss EAP 6 supérieure à $(\text{nombre de destinations}) * (\text{max-size-bytes})$, sinon une erreur « out-of-memory » (insuffisance de mémoire) peut se produire.

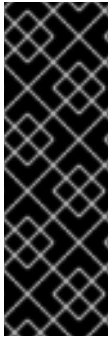
Vous pouvez configurer la taille maximum en bytes (*max-size-bytes*) pour une adresse dans les fichiers de configuration du serveur (*standalone.xml* et *domain.xml*) :

```
<address-settings>
  <address-setting match="jms.someaddress">
    <max-size-bytes>104857600</max-size-bytes>
    <page-size-bytes>10485760</page-size-bytes>
    <address-full-policy>PAGE</address-full-policy>
  </address-setting>
</address-settings>
```

Le tableau suivant décrit les paramètres utilisés pour la configuration de l'adresse :

Tableau 18.4. Configuration des paramètres de l'adresse de pagination

Élément	Valeur par défaut	Description
max-size-bytes	10485760	Ceci est utilisé pour indiquer la taille maximum de la mémoire que l'adresse peut avoir avant d'entrer en mode de pagination.
page-size-bytes	2097152	Cela est utilisé pour spécifier la taille de chaque fichier de pagination sur le système de pagination.
address-full-policy	PAGE	Cette valeur de cet attribut est utilisée pour les décisions de pagination. Vous pouvez définir n'importe quelle de ces valeurs pour cet attribut : PAGE : pour activer la pagination et les messages de pagination au delà de la limite définie sur le disque, DROP : pour supprimer silencieusement les messages qui excèdent la limite définie, FAIL : pour supprimer les messages et envoyer une exception aux producteurs de messages au client, BLOCK : pour bloquer les producteurs de messages client quand ils envoient des messages au delà de la limite définie
page-max-cache-size	5	Le système conservera des fichiers de pagination à hauteur de <i>page-max-cache-size</i> en mémoire pour optimiser l'Input/Output pendant la navigation de pagination



IMPORTANT

Si vous ne souhaitez pas paginer des messages quand la taille maximum est atteinte, vous pouvez configurer une adresse pour pouvoir supprimer les messages tout simplement, supprimer les messages avec une exception côté client ou bloquer les producteurs de messages pour qu'ils cessent d'envoyer des messages supplémentaires, en définissant *address-full-policy* à **DROP**, **FAIL** ou **BLOCK** respectivement. Dans la configuration par défaut, toutes les adresses devront être configurées pour paginer les messages une fois qu'une adresse atteint *max-size-bytes*.

Adresses avec files d'attente multiples

Quand un message est rerouté vers une adresse ayant plusieurs files d'attente, il n'y aura qu'un message en mémoire. Chaque file d'attente gère une référence du message. Ainsi, la mémoire n'est libérée que quand toutes les files d'attente référençant le message auront délivré le message.



NOTE

Un abonnement/ ou file d'attente lazy peuvent réduire la performance Input/Output de toute l'adresse car toutes les files d'attente auront des messages envoyés par l'intermédiaire d'un stockage supplémentaire dans le système de pagination.

[Rapporter un bogue](#)

18.7. DIVERTS

Les diverts (redirections) sont des objets configurés dans HornetQ qui aident à redirectionner des messages d'une adresse (vers laquelle le message est dirigé) vers d'autres adresses. Les redirections peuvent être configurées dans des fichiers de configuration de serveur (*standalone.xml* et *domain.xml*).

Les diverts peuvent être classifiés en types suivants :

- **Exclusive Divert** : un message est redirigé vers une nouvelle adresse uniquement et n'est pas dutout renvoyé vers une ancienne adresse
- **Non-exclusive Divert** : un message continue d'aller vers une ancienne adresse, et une copie de ce message est également envoyée vers la nouvelle adresse. Les redirections non exclusives (non-exclusive diverts) peuvent être utilisées pour diviser les flux de messages.

Les diverts peuvent être configurés pour appliquer un **Transformer** et un filtre optionnel de message. Un filtre optionnel de message ne redirige que des messages qui correspondent au filtre spécifié. Un transformateur est utilisé pour transformer les messages sous une autre forme. Quand un transformateur est spécifié, tous les messages redirigés sont transformés par le **Transformer**.

Un divert ne redirige un message vers une adresse que dans un même serveur. Si vous avez besoin de rediriger un message vers un serveur différent, vous pouvez suivre le modèle décrit ci-dessous :

- Rediriger les messages vers un store local et une file d'attente de redirection. Définir un pont qui consomme à partir de cette file d'attente et qui redirige les messages vers une adresse qui se trouve sur un serveur différent.

Vous pouvez combiner des diverts par des ponts pour créer divers routages.

[Rapporter un bogue](#)

18.7.1. Exclusive Divert

Un exclusive divert (redirection exclusive) redirige tous les messages d'une ancienne adresse vers une nouvelle adresse. Les messages correspondants ne sont pas redirigés vers l'ancienne adresse. Vous pouvez activer un exclusive divert en configurant l'attribut *exclusive* à *true* dans les fichiers de configuration `standalone.xml` et `domain.xml` du serveur.

L'exemple suivant vous présente un exclusive divert configuré dans les fichiers de configuration du serveur :

```
<divert name="prices-divert">
  <address>jms.topic.priceUpdates</address>
  <forwarding-address>jms.queue.priceForwarding</forwarding-address>
  <filter string="office='New York'"/>
  <transformer-class-name>
    org.hornetq.jms.example.AddForwardingTimeTransformer
  </transformer-class-name>
  <exclusive>>true</exclusive>
</divert>
```

La liste suivante décrit les attributs utilisés dans l'exemple ci-dessus :

- **address**: les messages envoyés à cette adresse sont redirigés vers une autre adresse
- **forwarding-address**: les messages sont redirigés vers cette adresse à partir d'une ancienne adresse
- **filter-string**: les messages qui correspondent à la valeur *filter-string* sont redirigés. Tous les autres messages sont redirigés vers l'adresse normale
- **transformer-class-name**: si vous spécifiez ce paramètre, une transformation aura lieu pour chaque message correspondant. Cela vous permettra de modifier le contenu ou la propriété d'un message avant qu'il soit redirigé.
- **exclusive**: utilisé pour activer ou désactiver un exclusive divert

Rapporter un bogue

18.7.2. Non-exclusive Divert

Un non-exclusive diverts (redirection non exclusive) redirige une copie de message d'origine vers une nouvelle adresse. Le message d'origine continue d'arriver à la nouvelle adresse. Vous pouvez configurer des redirections non exclusives en configurant la propriété *exclusive* à *false* dans les fichiers de configuration du serveur `standalone.xml` et `domain.xml`.

L'exemple suivant présente un non-exclusive divert :

```
<divert name="order-divert">
  <address>jms.queue.orders</address>
  <forwarding-address>jms.topic.spyTopic</forwarding-address>
  <exclusive>>false</exclusive>
</divert>
```

L'exemple ci-dessus fait une copie de chaque message envoyé à l'adresse *jms.queue.orders* et l'envoie à l'adresse *jms.topic.spyTopic*.

[Rapporter un bogue](#)

18.8. CONFIGURATION

18.8.1. Configurer le serveur JMS

Pour configurer le JMS Server d'HornetQ, modifier le fichier de configuration du serveur. La configuration du serveur se trouve dans le fichier

EAP_HOME/domain/configuration/domain.xml pour les serveurs de domaine, ou dans le fichier *EAP_HOME*/standalone/configuration/standalone-full.xml pour les serveurs autonomes.

L'élément `<subsystem xmlns="urn:jboss:domain:messaging:1.4">` du fichier de configuration du serveur contient toute la configuration JMS. Ajouter les instances `ConnectionFactory`, `Queue`, ou `Topic` requises pour le JNDI.

1. Activer le sous-système JMS dans JBoss EAP 6

Dans l'élément `<extensions>`, vérifier que la ligne suivante est bien présente et n'est pas décommentée :

```
<extension module="org.jboss.as.messaging"/>
```

2. Ajouter le sous-système JMS de base.

Si le sous-système de messagerie n'est pas présent dans votre fichier de configuration, ajoutez-le.

- Chercher le `<profile>` qui corresponde à celui que vous utilisez, et chercher sa balise de `<subsystems>`.
- Ajouter le XML suivant à la suite de la balise suivante `<profile>`.

```
<subsystem xmlns="urn:jboss:domain:messaging:1.4">
  <hornetq-server>
    <!-- ALL XML CONFIGURATION IS ADDED HERE -->
  </hornetq-server>
</subsystem>
```

Toutes les configurations supplémentaires pourront être ajoutées à la ligne vide ci-dessus.

3. Ajouter la configuration de base à JMS.

Ajouter l'XML suivant dans les lignes restées vides juste après la balise `<subsystem xmlns="urn:jboss:domain:messaging:1.4"><hornetq-server>` :

```
<journal-min-files>2</journal-min-files>
<journal-type>NIO</journal-type>
<persistence-enabled>>true</persistence-enabled>
```

Personnaliser les valeurs ci-dessus pour qu'elles correspondent à vos besoins.



AVERTISSEMENT

La valeur de `journal-file-size` doit être plus élevée ou égale à `min-large-message-size` (100KiB par défaut), ou bien le serveur ne pourra pas stocker le message.

4. Ajouter les instances de fabrique de connexion à HornetQ

Le client utilise un objet `ConnectionFactory` JMS pour faire des connexions au serveur. Pour ajouter un objet de fabrique de connexions JMS à HornetQ, inclure une simple balise `<jms-connection-factories>` et un élément `<connection-factory>` pour chaque fabrique de connexion comme suit :

```
<jms-connection-factories>
  <connection-factory name="InVmConnectionFactory">
    <connectors>
      <connector-ref connector-name="in-vm"/>
    </connectors>
    <entries>
      <entry name="java:/ConnectionFactory"/>
    </entries>
  </connection-factory>
  <connection-factory name="RemoteConnectionFactory">
    <connectors>
      <connector-ref connector-name="netty"/>
    </connectors>
    <entries>
      <entry
name="java:jboss/exported/jms/RemoteConnectionFactory"/>
    </entries>
  </connection-factory>
  <pooled-connection-factory name="hornetq-ra">
    <transaction mode="xa"/>
    <connectors>
      <connector-ref connector-name="in-vm"/>
    </connectors>
    <entries>
      <entry name="java:/JmsXA"/>
    </entries>
  </pooled-connection-factory>
</jms-connection-factories>
```

5. Configurer les connecteurs et accepteurs netty

La fabrique de connexion JMS utilise des connecteurs et accepteurs `netty`. Il s'agit de références à des objets de connecteurs ou d'accepteurs déployés dans le fichier de configuration du serveur. L'objet de connecteur détermine le transport et les paramètres utilisés pour vous connecter au serveur HornetQ. L'accepteur identifie le type de connexions acceptées par le serveur HornetQ.

Pour configurer les connecteurs `netty`, inclure les paramètres suivants :

```

<connectors>
  <netty-connector name="netty" socket-binding="messaging"/>
  <netty-connector name="netty-throughput" socket-
binding="messaging-throughput">
    <param key="batch-delay" value="50"/>
  </netty-connector>
  <in-vm-connector name="in-vm" server-id="0"/>
</connectors>

```

Pour configurer les accepteurs netty, inclure les paramètres suivants :

```

<acceptors>
  <netty-acceptor name="netty" socket-binding="messaging"/>
  <netty-acceptor name="netty-throughput" socket-
binding="messaging-throughput">
    <param key="batch-delay" value="50"/>
    <param key="direct-deliver" value="false"/>
  </netty-acceptor>
  <in-vm-acceptor name="in-vm" server-id="0"/>
</acceptors>

```

6. Vérifier la configuration

Si vous avez suivi les étapes suivantes, votre système de messagerie devra ressembler à ce qui suit :

```

<subsystem xmlns="urn:jboss:domain:messaging:1.4">
  <hornetq-server>
    <journal-min-files>2</journal-min-files>
    <journal-type>NIO</journal-type>
    <persistence-enabled>true</persistence-enabled>
    <jms-connection-factories>
      <connection-factory name="InVmConnectionFactory">
        <connectors>
          <connector-ref connector-name="in-vm"/>
        </connectors>
        <entries>
          <entry name="java:/ConnectionFactory"/>
        </entries>
      </connection-factory>
      <connection-factory name="RemoteConnectionFactory">
        <connectors>
          <connector-ref connector-name="netty"/>
        </connectors>
        <entries>
          <entry
name="java:jboss/exported/jms/RemoteConnectionFactory"/>
        </entries>
      </connection-factory>
      <pooled-connection-factory name="hornetq-ra">
        <transaction mode="xa"/>
        <connectors>
          <connector-ref connector-name="in-vm"/>
        </connectors>
        <entries>
          <entry name="java:/JmsXA"/>

```

```

        </entries>
    </pooled-connection-factory>
</jms-connection-factories>
<connectors>
    <netty-connector name="netty" socket-
binding="messaging"/>
    <netty-connector name="netty-throughput" socket-
binding="messaging-throughput">
        <param key="batch-delay" value="50"/>
    </netty-connector>
    <in-vm-connector name="in-vm" server-id="0"/>
</connectors>
<acceptors>
    <netty-acceptor name="netty" socket-
binding="messaging"/>
    <netty-acceptor name="netty-throughput" socket-
binding="messaging-throughput">
        <param key="batch-delay" value="50"/>
        <param key="direct-deliver" value="false"/>
    </netty-acceptor>
    <in-vm-acceptor name="in-vm" server-id="0"/>
</acceptors>
</hornetq-server>
</subsystem>

```

7. Configurer les groupes de liaisons de sockets

Les connecteurs netty réfèrent les liaisons de socket de **messaging** et de **messaging-throughput**. La liaison de socket de **messaging** utilise le port 5445, et la liaison de socket **messaging-throughput** utilise le port 5455. La balise **<socket-binding-group>** se situe dans une section séparée du fichier de configuration du serveur. Veillez à ce que les liaisons de socket suivantes soient présentes dans l'élément **<socket-binding-groups>** :

```

<socket-binding-group name="standard-sockets" default-
interface="public" port-offset="{jboss.socket.binding.port-
offset:0}">
    ...
    <socket-binding name="messaging" port="5445"/>
    <socket-binding name="messaging-throughput" port="5455"/>
    ...
</socket-binding-group>

```

8. Ajouter les instances de file d'attente à HornetQ

Il y a quatre façons de configurer les instances de files d'attente (ou destinations JMS) pour HornetQ.

- o Utiliser la console de gestion

Pour utiliser la console de gestion, le serveur devra être démarré sous le mode **Message-Enabled**. Vous y parviendrez en utilisant l'option **-c** et en forçant l'utilisation du fichier de configuration **standalone-full.xml** (pour les serveurs autonomes). Ainsi, en mode autonome, ce qui suit démarrera le serveur en mode activation de message.

```
./standalone.sh -c standalone-full.xml
```

Une fois que le serveur aura démarré, connectez-vous à la console de gestion, et

sélectionner l'onglet **Configuration**. Étendre le menu **Subsystems**, puis le menu **Messaging** et cliquer sur **Destinations**. À côté de **Default** dans le tableau JMS Messaging Provider, cliquer sur **View**, puis cliquer sur **Add** pour saisir les détails de la destination JSM.

- o Utiliser l'interface CLI :

Tout d'abord, connectez-vous à l'interface CLI :

```
bin/jboss-cli.sh --connect
```

Puis, passez au sous-système de messagerie :

```
cd /subsystem=messaging/hornetq-server=default
```

Finalement, exécuter une opération « add », en remplaçant les exemples de valeurs données ci-dessous par les vôtres :

```
./jms-queue=testQueue:add(durable=false,entries=
["java:jboss/exported/jms/queue/test"])
```

- o Créer un fichier de configuration JMS et l'ajouter au dossier de déploiements

Commencer à créer un fichier de configuration JMS : *example-jms.xml*. Ajouter y les entrées suivantes, en remplaçant les valeurs par les vôtres.

```
<?xml version="1.0" encoding="UTF-8"?>           <messaging-
deployment xmlns="urn:jboss:messaging-deployment:1.0">
  <hornetq-server>
    <jms-destinations>
      <jms-queue name="testQueue">
        <entry name="queue/test"/>
        <entry
name="java:jboss/exported/jms/queue/test"/>
      </jms-queue>
      <jms-topic name="testTopic">
        <entry name="topic/test"/>
        <entry
name="java:jboss/exported/jms/topic/test"/>
      </jms-topic>
    </jms-destinations>
  </hornetq-server>
</messaging-deployment>
```

Sauvegarder ce fichier dans le dossier de déploiements et faire un déploiement.

- o Ajouter les entrées dans le fichier de configuration de JBoss EAP 6.

En utilisant *standalone-full.xml* comme exemple, cherchez le sous-système de messagerie dans ce fichier.

```
<subsystem xmlns="urn:jboss:domain:messaging:1.4">
```


Y ajouter les entrées suivantes, encore une fois, en remplaçant les valeurs de l'exemple par les vôtres. Vous devez ajouter ces entrées après la balise de fin `</jms-connection-factories>` mais avant l'élément `</hornetq-server>` :

```
<jms-destinations>
  <jms-queue name="testQueue">
    <entry name="queue/test"/>
    <entry name="java:jboss/exported/jms/queue/test"/>
  </jms-queue>
  <jms-topic name="testTopic">
    <entry name="topic/test"/>
    <entry name="java:jboss/exported/jms/topic/test"/>
  </jms-topic>
</jms-destinations>
```

9. Procéder à une configuration supplémentaire

Si vous avez besoin de davantage de paramètres de configuration, revoir DTD dans `EAP_HOME/docs/schema/jboss-as-messaging_1_4.xsd`.

[Rapporter un bogue](#)

18.8.2. Configuration des paramètres de l'adresse JMS

Le sous-système JMS comprend plusieurs options configurables qui gèrent différents aspects de la transmission des messages, le nombre de tentatives d'envoi, et quand le message devra expirer. Ces options de configuration sont contenues dans l'élément de configuration `<address-settings>`.

Une des caractéristiques des configurations d'adresse est la syntaxe commune pour faire correspondre des adresses diverses, connue également sous le nom de Wildcard (caractères génériques).

Syntaxe Wildcard

Les adresses en syntaxe wildcard peuvent être utilisées pour faire correspondre plusieurs adresses similaires à une seule instruction, ce qui est semblable à la façon dont nombreux systèmes utilisent le caractère astérisque (*) pour faire correspondre plusieurs fichiers ou chaînes dans une seule recherche. Les caractères suivants ont une signification particulière dans un énoncé wildcard.

Tableau 18.5. Syntaxe Wildcard JMS

Caractère	Description
. (point simple)	Marque l'espace entre les mots au sein d'une expression wildcard.
# (a pound or hash symbol)	Fait correspondre une séquence de zéros ou de plusieurs mots.
* (un astérisque)	Faire correspondre à un mot unique.

Tableau 18.6. Exemples de JMS Wildcards

Exemple	Description
news.europe.#	Correspond à news . europe , news . europe . sport , news . europe . politic , mais pas à news . usa ou europe .
news.*	Correspond à news . europe mais pas à news . europe . sport .
news.*.sport	Correspond à news . europe . sport et news . usa . sport , mais pas à news . europe . politics .

Exemple 18.2. Configuration des paramètres d'adresse par défaut

Les valeurs dans cet exemple sont utilisées pour illustrer le reste de ce topic.

```
<address-settings>
  <!--default for catch all-->
  <address-setting match="#">
    <dead-letter-address>jms.queue.DLQ</dead-letter-address>
    <expiry-address>jms.queue.ExpiryQueue</expiry-address>
    <redelivery-delay>0</redelivery-delay>
    <max-size-bytes>10485760</max-size-bytes>
    <address-full-policy>BLOCK</address-full-policy>
    <message-counter-history-day-limit>10</message-counter-history-
day-limit>
  </address-setting>
</address-settings>
```

Tableau 18.7. Description de la configuration des paramètres de l'adresse JMS

Élément	Description	Valeur par défaut	Type
address-full-policy	Détermine ce qui se passe quand une adresse dont la max-size-bytes est spécifiée, est remplie.	PAGE	STRING

Élément	Description	Valeur par défaut	Type
dead-letter-address	Si une adresse de lettres mortes est spécifiée, les messages seront déplacés vers l'adresse de lettres mortes si les tentatives de livraison max-livraison-tentatives ont échoué. Dans le cas contraire, ces messages non remis seront ignorés. Les caractères génériques (wildcard) sont autorisés.	jms.queue.DLQ	STRING
expiry-address	Si l'adresse d'expiration est présente, les messages expirés seront envoyés à l'adresse ou aux adresses correspondantes, au lieu d'être jetés. Les caractères génériques (wildcards) sont autorisés.	jms.queue.ExpiryQueue	STRING
last-value-queue	Définit si une file d'attente utilise uniquement les dernières valeurs ou non.	false	BOULÉEN
max-delivery-attempts	Le nombre max de tentatives d'envoi d'un message avant qu'il soit envoyé à dead-letter-address ou qu'il soit ignoré.	10	INT
max-size-bytes	La taille maximum d'octets.	10485760L	LONG
message-counter-history-day-limit	Limite en Jours de l'historique du compteur de messages.	10	INT
page-max-cache-size	Le nombre de pages de fichiers à conserver en mémoire pour optimiser IO en cours de navigation de pagination.	5	INT

Élément	Description	Valeur par défaut	Type
page-size-bytes	La taille de pagination.	5	INT
redelivery-delay	La durée entre les tentatives de re-livraison, exprimée en millisecondes. Si défini sur la valeur 0, les tentatives de re-livraison auront lieu indéfiniment.	0L	LONG
redistribution-delay	Définit la durée à attendre jusqu'à ce que le dernier consommateur d'une file d'attente soit fermé, avant de pouvoir redistribuer des messages.	-1L	LONG
send-to-dla-on-no-route	Un paramètre pour une adresse qui définit la condition d'un message non acheminé vers une file d'attente pour qu'il soit envoyé à la place vers une file d'attente des messages morts ou DLA (de l'anglais Dead Letter Queue) indiquée pour cette adresse.	false	BOOLEEN

- **Configurer les paramètres de l'adresse et les attributs du modèle**

Choisir l'interface CLI ou la console de gestion pour configurer vos attributs de modèle selon les besoins.

- **Configurer les paramètres de l'adresse par l'interface CLI**

Utiliser l'interface CLI pour configurer les paramètres de l'adresse.

- Ajouter un nouveau Modèle**

Utiliser l'opération **add** pour créer un nouveau paramètre d'adresse, si nécessaire.

Vous pouvez exécuter cette commande à partir de la racine de la session d'interface CLI, qui, dans les exemples suivants, crée un nouveau modèle ou motif intitulé *patternname* avec un attribut **max-delivery-attempts** déclaré à 5. Voici des exemples pour les modifications sur le serveur autonome et le domaine géré pour le profil **full**.

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-server=default/address-setting=patternname/:add(max-delivery-attempts=5)
```

```
[domain@localhost:9999 /]
```

```
/profile=full/subsystem=messaging/hornetq-
server=default/address-setting=patternname/:add(max-delivery-
attempts=5)
```

b. Modifier les attributs de modèle

Utiliser l'opération `write` pour écrire une nouvelle valeur dans un attribut. Vous pouvez utiliser l'onglet de complétion pour terminer la chaîne de commande en cours, ainsi que pour exposer les attributs disponibles. L'exemple suivant met à jour la valeur de `max-delivery-attempts` à `10`

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-
server=default/address-setting=patternname/:write-
attribute(name=max-delivery-attempts,value=10)
```

```
[domain@localhost:9999 /]
/profile=full/subsystem=messaging/hornetq-
server=default/address-setting=patternname/:write-
attribute(name=max-delivery-attempts,value=10)
```

c. Confirmer les attributs de modèle

Confirmer que les valeurs ont changé en exécutant `read-resource` accompagné du paramètre `include-runtime=true` pour exposer toutes les valeurs actives en cours du modèle de serveur.

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-
server=default/address-setting=patternname/:read-resource
```

```
[domain@localhost:9999 /]
/profile=full/subsystem=messaging/hornetq-
server=default/address-setting=patternname/:read-resource
```

o Configurer les paramètres de l'adresse par la console de gestion

Utiliser la console de gestion pour configurer les paramètres de l'adresse.

- a. Connectez-vous à la console de gestion de votre domaine géré ou de votre serveur autonome.
- b. Sélectionner l'onglet **Configuration** en haut de l'écran. En mode de domaine, sélectionner un profil à partir du menu **Profile** en haut et à gauche. Seuls les profils **full** et **full-ha** ont le sous-système **messaging** activé.
- c. Étendre le menu **Messaging**, et sélectionner **Destinations**.
- d. Une liste de fournisseurs JMS s'affiche. Dans la configuration par défaut, on ne voit que le fournisseur par **default**. Cliquer sur **View** pour afficher les paramètres de ce fournisseur en détail.
- e. Cliquer sur l'onglet **Address Settings**. Ensuite, vous pourrez soit ajouter un nouveau modèle en cliquant sur **Add**, ou sélectionner un modèle existant et cliquer sur **Edit** pour mettre les paramètres à jour.
- f. Si vous ajoutez un modèle, le champ **Pattern** s'en référera au paramètre **match** de l'élément **address-setting**. Vous pourrez aussi modifier **Dead Letter Address**,

Expiry Address, Redelivery Delay, et Max Delivery Attempts. Les autres options doivent être configurées par l'interface CLI.

[Rapporter un bogue](#)

18.8.3. Configurer la messagerie dans HornetQ

La méthode recommandée pour configurer la messagerie dans JBoss EAP 6 est soit par la console de gestion, soit par l'interface CLI. Vous pouvez effectuer des modifications persistantes avec l'un ou l'autre de ces outils de gestion sans avoir besoin de modifier manuellement les fichiers de configuration `standalone.xml` ou `domain.xml`. Cependant, il est utile de se familiariser avec les composants de messagerie des fichiers de configuration par défaut, où les exemples de documentation utilisant des outils de gestion donnent des extraits de fichiers de configuration comme référence.

[Rapporter un bogue](#)

18.8.4. Activer la journalisation dans HornetQ

Vous pouvez activer la journalisation pour HornetQ dans EAP 6.x en utilisant une des approches suivantes :

- Éditing des fichiers de configuration du serveur (`standalone-full.xml` et `standalone-full-ha.xml`) manuellement
- Éditing des fichiers de configuration du serveur par le CLI

Procédure 18.1. Définir la journalisation d'HornetQ en modifiant les fichiers de configuration du serveur manuellement

1. Ouvrir le(s) fichier(s) de configuration du serveur pour l'éditing. Par exemple, `standalone-full.xml` ou `standalone-full-ha.xml`
2. Naviguer dans la configuration du sous-système de journalisation dans le(s) fichier(s). La configuration par défaut ressemble à ceci :

```
<logger category="com.arjuna">
  <level name="TRACE"/>
</logger>
...
<logger category="org.apache.tomcat.util.modeler">
  <level name="WARN"/>
</logger>
....
```

3. Ajouter la catégorie d'enregistreur d'événement `org.hornetq` ainsi que le niveau de journalisation désiré, comme le montre l'exemple suivant :

```
<logger category="com.arjuna">
  <level name="TRACE"/>
</logger>
...
<logger category="org.hornetq">
  <level name="INFO"/>
</logger>
....
```

Résultat

La journalisation HornetQ est active et les messages de journalisation sont traités selon le niveau de journalisation défini.

Définir la journalisation d'HornetQ en modifiant les fichiers de configuration du serveur par l'interface CLI

Vous pouvez également utiliser l'interface CLI pour ajouter la catégorie d'enregistreur d'événement (logger) `org.hornetq` ainsi que le niveau de journalisation désiré dans le(s) fichier(s) de configuration du serveur. Pour plus d'informations : [Section 12.3.2, « Configurer une Catégorie dans l'interface CLI »](#)

[Rapporter un bogue](#)

18.8.5. Configurer HornetQ Core Bridge

Exemple 18.3. Exemple de configuration d'Hornet Core Bridge :

Les valeurs de cet exemple sont utilisées pour illustrer le reste de cette section.

```
<bridges>
  <bridge name="myBridge">
    <queue-name>jms.queue.InQueue</queue-name>
    <forwarding-address>jms.queue.OutQueue</forwarding-address>
  </ha>true</ha>
  <reconnect-attempts>-1</reconnect-attempts>
  <use-duplicate-detection>true</use-duplicate-detection>
  <static-connectors>
    <connector-ref>
      bridge-connector
    </connector-ref>
  </static-connectors>
</bridge>
</bridges>
```

Tableau 18.8. Attributs d'HornetQ Core Bridge

Attribut	Description
name	Tous les ponts doivent posséder un nom unique dans le serveur :
queue-name	Ce paramètre obligatoire est le nom unique de la file d'attente locale utilisée par le pont. La file d'attente doit déjà exister au moment où que le pont est instancié au démarrage.
forwarding-address	Il s'agit de l'adresse qui se trouve sur le serveur cible où le message sera envoyé. Si aucune adresse n'est spécifiée, alors, l'adresse d'origine du message sera retenue.

Attribut	Description
ha	Ce paramètre optionnel déterminera si ce pont doit prendre en charge HA ou non. <code>true</code> indique qu'il se connectera à tout serveur disponible faisant partie d'un groupement ou qu'il supportera le basculement. La valeur par défaut est <code>false</code> .
reconnect-attempts	Ce paramètre optionnel détermine le nombre total de tentatives de reconnexion que le pont doit faire avant d'abandonner et se fermer. Une valeur <code>-1</code> indique un nombre d'essais illimité. La valeur par défaut correspond à <code>-1</code> .
use-duplicate-detection	Ce paramètre optionnel détermine si un pont doit ou non insérer automatiquement une propriété d'identifiant en duplicata dans chaque message qu'il fait suivre.
static-connectors	<code>static-connectors</code> correspond à une liste d'éléments <code>connector-ref</code> pointant vers des éléments de connecteur définis par ailleurs. Un connecteur encapsule les informations sur le transport à utiliser (TCP, SSL, HTTP etc.) ainsi que les paramètres de connexion de serveur (host, port, etc.).

[Rapporter un bogue](#)

18.8.6. Configurer un pontage JMS

HornetQ inclut un pontage de messages JMS qui fonctionne bien. La fonction de ce pontage est de consommer des messages à partir d'un sujet ou d'une file d'attente source, et de les envoyer vers un sujet ou une file d'attente cible, se trouvant normalement sur un serveur différent.

Les serveurs sources et cibles ne doivent pas forcément être dans le même cluster, ce qui permet d'envoyer des messages d'un cluster à l'autre de façon fiable, via un WAN ou quand il y a une connexion fiable.

On peut déployer un pontage en tant qu'application autonome, avec le serveur autonome HornetQ ou à l'intérieur de l'instance JBoss AS. La source et la cible peuvent se trouver dans la même machine virtuelle ou dans une autre.

Exemple 18.4. Exemple de configuration de pontage JMS :

Les valeurs de cet exemple sont utilisées pour illustrer le reste de cette section.

```
<subsystem>
  <subsystem xmlns="urn:jboss:domain:messaging:1.3">
    <hornetq-server>
      ...
    </hornetq-server>

    <jms-bridge name="myBridge">
```



```

        <source>
            <connection-factory name="ConnectionFactory"/>
            <destination name="jms/queue/InQueue"/>
        </source>
        <target>
            <connection-factory
name="jms/RemoteConnectionFactory"/>
            <destination name="jms/queue/OutQueue"/>
            <context>
                <property key="java.naming.factory.initial"
value="org.jboss.naming.remote.client.InitialContextFactory"/>
                <property key="java.naming.provider.url"
value="remote://192.168.40.1:4447"/>
            </context>
        </target>
        <quality-of-service>AT_MOST_ONCE</quality-of-service>
        <failure-retry-interval>1000</failure-retry-interval>
        <max-retries>-1</max-retries>
        <max-batch-size>10</max-batch-size>
        <max-batch-time>100</max-batch-time>
        <add-messageID-in-header>true</add-messageID-in-header>
    </jms-bridge>
    ...
</subsystem>

```

Tableau 18.9. Attributs JMS d'HornetQ Core

Attribut	Description
name	Tous les ponts doivent posséder un nom unique dans le serveur :
source connection-factory	Injecte le bean SourceCFF (comme défini dans le fichier de beans). Ce bean crée la ConnectionFactory d'origine.
source destination name	Injecte le bean SourceDestinationFactory (comme défini dans le fichier de beans). Ce bean crée la destination source.
target connection-factory	Injecte le bean TargetCFF (comme défini dans le fichier de beans). Ce bean crée la ConnectionFactory cible.
target destination name	Injecte le bean TargetDestinationFactory (comme défini dans le fichier de beans). Ce bean crée la destination cible.
quality-of-service	Ce paramètre représente la qualité de mode de service requise. Les valeurs possibles sont : AT_MOST_ONCE, DUPLICATES_OK, ONCE_AND_ONLY_ONCE

Attribut	Description
failure-retry-interval	Représente la durée en millisecondes qu'il faut attendre pour créer à nouveau des connexions vers les serveurs sources ou cibles quand le pontage a détecté qu'ils ont échoué.
max-retries	Représente le nombre de tentatives pour créer à nouveau des connexions vers les serveurs sources ou cibles quand le pontage a détecté qu'ils ont échoué. Le pontage échouera après un certain nombre de tentatives. -1 représente un nombre d'essais indéfini.
max-batch-size	Représente le nombre maximum de messages à consommer de la destination source avant de les envoyer en groupe vers une destination cible. Sa valeur doit être ≥ 1 .
max-batch-time	Cela représente la durée d'attente maximum en millisecondes avant d'envoyer un lot vers la cible, même si le nombre de messages consommés n'atteint pas la taille MaxBatchSize. Sa valeur doit être -1 pour correspondre à 'wait forever', ou bien ≥ 1 pour indiquer une durée précise.
add-messageID-in-header	<p>Si défini sur true, alors l'id du message original sera ajouté dans le message envoyé à la destination dans l'en-tête HORNETQ_BRIDGE_MSG_ID_LIST. Si le message est envoyé plus d'une fois, chaque id de message sera ajouté. Cela permettra l'utilisation d'un modèle de requête-réponse distribué.</p> <p>Quand vous recevrez le message, vous pourrez envoyer une réponse par l'id de corrélation de l'id du premier message, ce qui fait que quand l'émetteur d'origine recevra le message, il sera facile de faire une corrélation.</p>



NOTE

Quand on ferme un serveur qui a un pont JMS déployé avec un attribut *quality-of-service* défini à **ONCE_AND_ONLY_ONCE**, commencer par fermer le serveur avec le pont JMS pour éviter les exceptions inattendues.

Pour obtenir des informations plus en détail, consultez [Section 18.11.2, « Créer un pontage JMS »](#) .

[Rapporter un bogue](#)

18.8.7. Configurer la re-livraison différée

Introduction

La re-livraison différée est définie dans l'élément `<redelivery-delay>`, qui est un élément dépendant de l'élément de configuration `<address-setting>` de la configuration du sous-système JMS (Java Messaging Service).

```
<!-- delay redelivery of messages for 5s -->
<address-setting match="jms.queue.exampleQueue">
  <redelivery-delay>5000</redelivery-delay>
</address-setting>
```

Si un retard de livraison est spécifié, le système JMS attendra durant cette valeur de délai avant de re-livrer les messages. Si `<redelivery-delay>` est défini à `0`, il n'y aura pas de livraison à nouveau. Les caractères génériques (wildcards) peuvent être utilisés sur l'attribut *match* de l'élément `<address-match>` pour configurer la livraison à nouveau des adresses qui correspondent au(x) caractère(s) générique(s).

[Rapporter un bogue](#)

18.8.8. Configurer les adresses de lettres mortes

Introduction

Une adresse de lettres mortes est définie dans l'élément `<address-setting>` de configuration du sous-système de JMS (Java Messaging Service).

```
<!-- undelivered messages in exampleQueue will be sent to the dead letter
address
deadLetterQueue after 3 unsuccessful delivery attempts
-->
<address-setting match="jms.queue.exampleQueue">
  <dead-letter-address>jms.queue.deadLetterQueue</dead-letter-address>
  <max-delivery-attempts>3</max-delivery-attempts>
</address-setting>
```

Si `<dead-letter-address>` n'est pas spécifié, les messages sont supprimés au bout de `<max-delivery-attempts>` envois. Par défaut, les messages sont envoyés 10 fois. Si vous définissez `<max-delivery-attempts>` à `-1`, vous autorisez un nombre d'envois indéterminé. Ainsi, une lettre morte peut être définie globalement pour un ensemble d'adresses correspondantes et vous pouvez définir `<max-delivery-attempts>` à `-1` pour qu'une adresse particulière soit configurée sur un nombre d'envois indéfini. Les astérisques peuvent aussi être utilisés pour faire correspondre à un ensemble d'adresses particulières.

[Rapporter un bogue](#)

18.8.9. Configurer les adresses d'expiration de messages

Introduction

Les adresses d'expiration de messages sont définies dans la configuration `address-setting` de JMS (Java Messaging Service). Ainsi :

```
<!-- expired messages in exampleQueue will be sent to the expiry address
expiryQueue -->
<address-setting match="jms.queue.exampleQueue">
```

```
<expiry-address>jms.queue.expiryQueue</expiry-address>
</address-setting>
```

Si les messages ont expirés et qu'aucune adresse d'expiration n'est spécifiée, les messages sont tout simplement retirés de la file d'attente et abandonnés. Des *caractères de remplacement* peuvent également être utilisés pour configurer des plages de données d'adresses d'expiration spécifiques pour un ensemble d'adresses. Voir [Section 18.8.2, « Configuration des paramètres de l'adresse JMS »](#) pour la syntaxe de caractères de remplacement JMX et quelques exemples.

[Rapporter un bogue](#)

18.8.10. Référence pour les attributs de configuration d'HornetQ

L'implémentation d'HornetQ de JBoss EAP 6 expose les attributs de configuration suivants. Vous pouvez utiliser l'interface CLI pour exposer plus particulièrement les attributs configurables ou affichables par l'opération `read-resource`.

Exemple 18.5. Exemple

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-
server=default:read-resource
```

Tableau 18.10. Attributs HornetQ

Attribut	Valeur par défaut	Type	Description
<code>allow-failback</code>	true	BOOLÉEN	Il indique si le serveur de sauvegarde se fermera automatiquement quand le serveur live d'origine revient.
<code>async-connection-execution-enabled</code>	true	BOOLÉEN	Indique si les paquets entrants sur le serveur doivent être remis à un thread du pool de threads pour le traitement
<code>address-setting</code>			Un paramétrage de l'adresse définit certains attributs avec un caractère générique d'adresse plutôt qu'une file d'attente spécifique
<code>acceptor</code>			Un accepteur définit une façon dont les connections sont acceptées par le serveur HornetQ.

Attribut	Valeur par défaut	Type	Description
backup-group-name		STRING	Le nom d'un ensemble de live/sauvegardes qui doivent se répliquer ensemble.
backup	false	BOOLÉEN	Indique si ce serveur est un serveur de sauvegarde
check-for-live-server	false	BOOLÉEN	Il indique si un serveur répliqué live doit vérifier le cluster actuel pour voir s'il existe déjà un serveur live avec le même ID de nœud
clustered	false	BOOLÉEN	[Déprécié] Indique si le serveur est clusterisé
cluster-password	CHANGE ME !!	STRING	Le mot de passe utilisé par les connexions du cluster pour communiquer entre les nœuds clusterisés
cluster-user	HORNETQ. CLUSTER.A DMIN.USER	STRING	L'utilisateur utilisé par les connexions du cluster pour communiquer entre les nœuds clusterisés
cluster-connection			Les connexions du cluster groupe les serveurs en clusters pour que les messages puissent être équilibrés entre les nœuds du cluster
create-bindings-dir	true	BOOLÉEN	Indique si le serveur doit créer le répertoire de liaisons au démarrage
create-journal-dir	true	BOOLÉEN	Indique si le serveur doit créer le journal de liaisons au démarrage
connection-ttl-override	-1L	LONG	Si défini, cela remplacera la longueur (en ms) qu'il faut pour conserver une connexion vivante sans recevoir de ping

Attribut	Valeur par défaut	Type	Description
connection-factory			Définit une usine de connexions
connector			Un connecteur peut être utilisé par un client pour définir comment il doit se connecter à un serveur
connector-service			
divert			Une ressource de messagerie qui vous permet de rediriger en toute transparence des messages routés vers une adresse vers d'autre adresse, sans apporter aucune modification à une logique d'application client
discovery-group			Groupe multidiffusion à écouter pour recevoir les informations de la part des autres serveurs lorsqu'ils annonceront leurs connecteurs.
failback-delay	5000	LONG	Le temps, en millisecondes, qu'il faut attendre avant qu'un failback (restauration automatique) puisse avoir lieu en cas de redémarrage d'une serveur live
failover-on-shutdown	false	BOULÉEN	Indique si ce serveur de sauvegarde (s'il s'agit d'un serveur de sauvegarde) doit devenir live lors d'un arrêt normal du serveur.
grouping-handler			Prend des décisions sur quels noeuds d'un cluster doit gérer un message lié à une id de groupe
id-cache-size	20000	INT	La taille du cache pour pré-créeer les ID de message

Attribut	Valeur par défaut	Type	Description
in-vm-acceptor			Définit la façon dont les connexions in-VM peuvent être faites au serveur HornetQ
in-vm-connector			Utilisé par un client in-VM pour définir comment il doit se connecter à un serveur
jmx-domain	org.hornetq	STRING	Le domaine JMX utilisé pour enregistrer des MBeans HornetQ dans le MBeanServer
jmx-management-enabled	false	BOOLÉEN	Indique si HornetQ doit exposer sa gestion interne API via JMX. Cela n'est pas recommandé, car accéder à ces MBeans peut conduire à une configuration incohérente
journal-buffer-size	501760 (490KiB)	LONG	La taille du tampon interne sur la journal
journal-buffer-timeout	500000 (0.5 millisecond s) for ASYNCIO journal and 3333333 (3.33 millisecond s) for NIO journal	LONG	Le timeout (en nanoseconds) utilisé pour vider des tampons internes dans le journal
journal-compact-min-files	10	INT	Le nombre minimum de fichiers de données de journaux avant que nous démarrions le compactage
journal-compact-percentage	30	INT	Le pourcentage de données live sur lequel on considère compacter le journal
journal-file-size	10485760	LONG	La taille (en octets) de chaque fichier de journal

Attribut	Valeur par défaut	Type	Description
journal-max-io	1	INT	Le nombre maximal de requêtes d'écriture que vous pourrez avoir dans la file d'attente de l'AIO à chaque instant. La valeur par défaut passe à 500 lorsque le journal ASYNCIO est utilisé
journal-min-files	2	INT	Le nombre de fichiers de journaux à pré-crérer
journal-sync-non-transactional	true	BOOLÉEN	Indique si on doit attendre que les données non transactionnelles soient synchronisées avec le journal avant de renvoyer une réponse au client
journal-sync-transactional	true	BOOLÉEN	Indique si on doit attendre que les données de transaction soient synchronisées avec le journal avant de renvoyer une réponse au client
journal-type	ASYNCIO	String	Le type de journal à utiliser. Cet attribut peut prendre les valeurs "ASYNCIO" ou "NIO"
jms-topic			Définit un sujet JMS
live-connector-ref	référence	STRING	[Déprécié] Le nom du connecteur utilisé pour se connecter au connecteur live. Si ce serveur n'est pas une sauvegarde qui utilise «shared nothing HA», sa valeur est «undefined» (indéfinie)
log-journal-write-rate	false	BOOLÉEN	Indique si on doit consigner périodiquement le taux d'écriture au journal et le taux de vidage
mask-password	true	BOOLÉEN	

Attribut	Valeur par défaut	Type	Description
management-address	jms.queue.hornetq.management	STRING	Adresse à laquelle envoyer des messages de gestion
management-notification-address	hornetq.notifications	STRING	Le nom de l'adresse que les consommateurs utilisent pour recevoir des notifications de la part du management
max-saved-replicated-journal-size	2	INT	Le nombre maximum de journaux de sauvegarde à conserver suite à une restauration automatique.
memory-measure-interval	-1	LONG	Fréquence d'échantillonnage de mémoire JVM en ms (ou -1 pour désactiver l'échantillonnage de mémoire)
memory-warning-threshold	25	INT	Pourcentage de mémoire disponible qui, si dépassée, résulte en message d'avertissement
message-counter-enabled	false	BOULÉEN	Indique si le compteur de messages est activé
message-counter-max-day-history	10	INT	Le nombre de jours qu'il faut conserver l'historique du compteur de messages
message-counter-sample-period	10000	LONG	La période d'échantillonnage (en ms) à utiliser pour les compteurs de messages
message-expiry-scan-period	30000	LONG	La fréquence (en ms) de balayage des messages expirés
message-expiry-thread-priority	3	INT	La priorité du thread de messages expirés
page-max-concurrent-io	5	INT	Le nombre maximal de lectures simultanées autorisées sur la pagination

Attribut	Valeur par défaut	Type	Description
perf-blast-pages	-1	INT	
persist-delivery-count-before-delivery	false	BOOLÉEN	Si le nombre de livraison est rendu persistant avant la livraison. False signifie que cela se produit uniquement après qu'un message ait été annulé
persist-id-cache	true	BOOLÉEN	Indique si les ID sont persistés dans le journal
persistence-enabled	true	BOOLÉEN	Indique si le serveur utilisera le journal basé fichier pour la persistance
pooled-connection-factory			Définit une usine de connexions gérées
remoting-interceptors	Non défini	LIST	[Deprecated] La liste de classes d'intercepteur utilisée par ce serveur
remoting-incoming-interceptors	Non défini	LIST	La liste de classes d'intercepteurs entrants utilisée par ce serveur
remoting-outgoing-interceptors	Non défini	LIST	La liste de classes d'intercepteurs sortants utilisée par ce serveur
run-sync-speed-test	false	BOOLÉEN	Indique si l'on doit procéder à un diagnostic de la vitesse de sync de votre disque au démarrage. Utile quand vous déterminez les problèmes de performance.
replication-clustername		STRING	Le nom de la connexion de cluster à répliquer si plus d'une connexion de serveur est configurée
runtime-queue			Une file d'attente de runtime
remote-connector			Utilisé par un client distant pour définir comment il doit se connecter à un serveur

Attribut	Valeur par défaut	Type	Description
remote-acceptor			Définit la façon dont les connexions distantes peuvent être faites au serveur HornetQ
scheduled-thread-pool-max-size	5	INT	Le nombre de threads contenus dans le thread pool principal programmé
security-domain	autre	STRING	Le domaine de sécurité à utiliser pour pouvoir vérifier les informations rôle et utilisateur
security-enabled	true	BOOLÉEN	Indique si la sécurité est activée
security-setting			Un paramètre de sécurité permet à un groupe de permissions d'être définies en fonction de files d'attente sur la base de leur adresse.
security-invalidation-interval	10000	LONG	Le temps, en millisecondes, qu'il faut attendre avant d'invalider la cache de sécurité
server-dump-interval	-1	LONG	La fréquence de vidage d'information de runtime de base dans le journal du serveur. Une valeur inférieure à 1 désactive cette fonction
shared store	true	BOOLÉEN	Indique si le serveur utilise ou non un store partagé en cas de basculement
thread-pool-max-size	30	INT	Le nombre de threads contenus dans le thread pool. -1 signifie qu'il n'y a pas de limite

Attribut	Valeur par défaut	Type	Description
transaction-timeout	300000	LONG	Le temps, en millisecondes (ms), qu'il faut attendre avant qu'une transaction puisse être retirée du gestionnaire de ressources après sa création
transaction-timeout-scan-period	1000	LONG	La fréquence (en ms) de balayage des transactions de timeout
wild-card-routing-enabled	true	BOOLÉEN	Indique si le serveur prend en charge le le routage de wild-card



AVERTISSEMENT

La valeur de **journal-file-size** doit être plus élevée que celle de la taille du message envoyé au serveur, ou bien le serveur ne pourra pas stocker le message.

[Rapporter un bogue](#)

18.8.11. Définir l'expiration des messages

Introduction

Les messages envoyés peuvent être définis de façon à pouvoir expirer sur le serveur, s'ils ne sont pas livrés au consommateur après un certain laps de temps (en millisecondes). À l'aide de Java Messaging Service (JMS) ou de l'API de base de HornetQ, le délai d'expiration est réglable sur le message directement. Par exemple :

```
// message will expire in 5000ms from now
message.setExpiration(System.currentTimeMillis() + 5000);
```

JMS **MessageProducer** inclut un paramètre **TimeToLive** qui contrôle l'expiration de message du message qu'il envoie :

```
// messages sent by this producer will be retained for 5s (5000ms) before
expiration
producer.setTimeToLive(5000);
```

Les messages expirés qui sont consommés à partir d'une adresse d'expiration ont les propriétés suivantes :

- `_HQ_ORIG_ADDRESS`

Une propriété de string qui contient l'adresse d'origine du message expiré.

- `_HQ_ACTUAL_EXPIRY`

Une propriété longue qui contient l'heure d'expiration du message expiré.

Au lieu de définir le paramètre `time-to-live` sur le producer JMS, vous pouvez également le définir au fur et à mesure. Vous pouvez le faire en ajoutant le paramètre `TTL` à la méthode d'envoi du producer quand vous envoyez le message.

```
producer.send(message, DeliveryMode.PERSISTENT, 0, 5000)
```

Quand le dernier paramètre est TTL basé message.

Configurer les adresses d'expiration

Les adresses d'expiration de messages sont définies dans la configuration `address-setting` :

```
<!-- expired messages in exampleQueue will be sent to the expiry address
expiryQueue -->
<address-setting match="jms.queue.exampleQueue">
  <expiry-address>jms.queue.expiryQueue</expiry-address>
</address-setting>
```

Si les messages sont expirés et qu'aucune adresse d'expiration n'est spécifiée, les messages sont tout simplement retirés de la file d'attente et abandonnés.

Configurer un Thread Reaper d'expiration

Un thread reaper inspecte périodiquement les files d'attente pour valider si les messages ont expiré.

- `message-expiry-scan-period`

La fréquence de balayage des files d'attente pour détecter les messages expirés (en millisecondes, valeur par défaut 30000ms, définie à -1 pour désactiver le thread reaper).

- `message-expiry-thread-priority`

Priorité de thread reaper. Doit se situer entre 0 et 9, 9 étant la plus haute priorité, et la valeur par défaut étant 3.

[Rapporter un bogue](#)

18.9. GROUPEMENT DES MESSAGES

18.9.1. Groupement des messages

Un groupement de messages est un ensemble de messages qui partagent certaines caractéristiques :

- Tous les messages d'un groupe de messages sont groupés sous un id de groupe commun. Cela signifie qu'ils peuvent être identifiés par une propriété de groupe en commun.
- Tous les messages dans un groupe de messages sont traités en série et consommés par le même consommateur quel que soit le nombre de clients présents dans la file d'attente. Cela

signifie qu'un groupe de messages spécifique avec un id de groupe unique est toujours traité par un consommateur lorsque le consommateur l'ouvre. Si le consommateur ferme le groupe de messages, le groupe de messages entier sera redirigé vers un autre consommateur dans la file d'attente.



IMPORTANT

Les groupes de messages sont particulièrement utiles lorsque des messages possédant une certaine valeur de la propriété (id de groupe) doivent être traités en série par un même consommateur.

[Rapporter un bogue](#)

18.9.2. Utilisation d'un API Hornet Cor côté client

La propriété `_HQ_GROUP_ID` est utilisée pour identifier un groupe de messages dans l'API d'HornetQ côté client. Pour choisir un id de groupe unique de groupe de message, vous pouvez également définir la propriété `autogroup` sur "true" dans la `SessionFactory`.

[Rapporter un bogue](#)

18.9.3. Configurer le serveur pour les clients JMS (Java Messaging Service)

La propriété `JMSXGroupID` est utilisée pour identifier un groupe de messages pour les clients JMS (Java Messaging Service). Si vous souhaitez envoyer un groupe de messages avec des messages différents vers un consommateur, vous pouvez définir le même `JMSXGroupID` pour des messages différents :

```
Message message = ...
message.setStringProperty("JMSXGroupID", "Group-0");
producer.send(message);

message = ...
message.setStringProperty("JMSXGroupID", "Group-0");
producer.send(message);
```

La deuxième approche consiste à définir la propriété `autogroup` sur « true » sur le `HornetQConnectionFactory`. Le `HornetQConnectionFactory` reprendra ensuite un id de groupe de message unique aléatoire. Vous pouvez définir la propriété `autogroup` dans les fichiers de configuration du serveur (`standalone.xml` et `domain.xml`) comme suit :

```
<connection-factory name="ConnectionFactory">
  <connectors>
    <connector-ref connector-name="netty-connector"/>
  </connectors>
  <entries>
    <entry name="ConnectionFactory"/>
  </entries>
  <autogroup>true</autogroup>
</connection-factory>
```

Une alternative aux deux approches ci-dessus consiste à définir un id de groupe de messages spécifique par le biais de la fabrique de connexions. Cela définira la propriété `JMSXGroupID` à la

valeur spécifiée pour tous les messages envoyés par le biais de cette fabrique de connexions. Pour définir un id de groupe de messages spécifique sur la fabrique de connexion, modifiez la propriété **groupe-id** dans les fichiers de configuration du serveur (`standalone.xml` et `domain.xml`) comme suit :

```
<connection-factory name="ConnectionFactory">
  <connectors>
    <connector-ref connector-name="netty-connector"/>
  </connectors>
  <entries>
    <entry name="ConnectionFactory"/>
  </entries>
  <group-id>Group-0</group-id>
</connection-factory>
```

[Rapporter un bogue](#)

18.9.4. Groupement clusterisé

Un regroupement clusterisé suit une approche différente du groupement de messages normal. Dans un cluster, les groupes de messages avec des id de groupe spécifiques peuvent arriver sur n'importe quel nœud. Il est important pour un nœud de déterminer quels identifiants de groupes sont liés aux consommateurs et sur quel nœud. Chaque nœud est responsable de diriger les groupes de messages correctement vers le nœud qui possède le consommateur qui traite ces ID de groupes quel que soit l'endroit où les groupes de messages arrivent par défaut.

Cette situation a été corrigée par un handler de groupement. Chaque nœud possède un handler de groupement et ce gestionnaire de groupement (avec d'autres handlers) est responsable de l'acheminement des groupes de messages vers le nœud qui convient. Il existe deux types de regroupement des handlers, à savoir `local` (local) et `remote` (distant).

Le handler local est chargé de décider du chemin qu'un groupe de messages doit prendre. Les handlers distants communiquent avec le handler local et fonctionnent ainsi. Chaque cluster doit choisir un nœud spécifique pour avoir un handler de groupement local et tous les autres nœuds doivent avoir des handlers distants.



AVERTISSEMENT

Si le regroupement de messages est utilisée dans le cluster, le processus s'interrompt en cas d'échec du gestionnaire de groupement distant configuré. Créer une sauvegarde pour un gestionnaire de groupement distant ne fonctionne pas non plus.

Vous pouvez configurer les handlers de groupements "local" et "remote" dans les fichiers de configuration du serveur (`standalone.xml` et `domain.xml`) comme suit :

```
<grouping-handler name="my-grouping-handler">
  <type>LOCAL</type>
  <address>jms</address>
  <timeout>5000</timeout>
```

```
</grouping-handler>
<grouping-handler name="my-grouping-handler">
  <type>REMOTE</type>
  <address>jms</address>
  <timeout>5000</timeout>
</grouping-handler>
```

L'attribut « timeout » veille à ce qu'une décision de routage soit faite rapidement dans un délai imparti. Si une décision n'est pas faite dans ce délai, une exception sera levée.

Le nœud qui commence par recevoir un groupe de messages prend la décision de routage selon les conditions de routage de cluster ordinaire (disponibilité de la file d'attente de round robin). Le nœud propose cette décision au handler de groupement respectif, qui achemine ensuite les messages vers la file d'attente proposée, s'il accepte la proposition.

Si le handler de groupement rejette la proposition, il proposera un autre itinéraire et le routage aura lieu en conséquence. Les autres nœuds suivront et renverront les groupes de messages dans la file d'attente choisie. Après l'arrivée d'un message dans une file d'attente, ce message sera alloué à un client sur cette file d'attente.

[Rapporter un bogue](#)

18.9.5. Meilleures pratiques avec les groupements clusterisés

Voici quelques unes des meilleures pratiques avec les groupements clusterisés :

- Si vous créez et fermez des consommateurs régulièrement, assurez-vous que vos clients soient répartis uniformément sur les différents nœuds. Une fois qu'une file d'attente est épinglée, les messages seront transférés automatiquement vers cette file d'attente indépendamment du fait que l'on puisse supprimer des visiteurs dessus.
- Si vous souhaitez supprimer une file d'attente qui dispose d'un groupe de messages qui lui soit liée, assurez-vous que la file d'attente soit supprimée par la session qui envoie les messages. Cela fera en sorte que les autres nœuds ne tenteront pas de router les messages dans cette file d'attente suite à la suppression.
- Comme un mécanisme de failover réplique toujours le nœud qui a le handler de groupement local

[Rapporter un bogue](#)

18.10. LA DÉTECTION DE MESSAGES DUPLIQUÉS

18.10.1. Détection de messages dupliqués

La détection de messages dupliqués permet le filtrage des messages dupliqués sans besoin de codage de la logique de détection des doublons au sein de l'application. Vous pouvez configurer la détection de messages dupliqués dans HornetQ.

Lorsqu'un émetteur(client/server) envoie un message vers un autre serveur, on peut assister à une situation où le serveur(receveur) de la cible ou la connexion échoue suite à l'envoi du message, mais ceci, avant l'envoi d'une réponse à l'expéditeur indiquant que le processus a réussi. Dans de telles situations, il est très difficile pour l'émetteur (client) de déterminer si le message a été envoyé avec succès au destinataire prévu.

L'envoi de message peut ou peut ne pas réussir si le récepteur cible ou la connexion échouent (avant ou après l'envoi du message). Si l'expéditeur (client/serveur) décide de renvoyer le dernier message, cela peut provoquer un message envoyé deux fois à la même adresse.

HornetQ fournit une détection des messages doubles envoyés aux adresses.

[Rapporter un bogue](#)

18.10.2. Utiliser la détection des messages en double pour l'envoi des messages

Pour activer la détection des messages en double des messages envoyés, vous devrez définir une propriété spéciale sur le message à une valeur unique. Vous pourrez créer la valeur comme vous le souhaitez, mais cette valeur doit être unique.

Lorsque le serveur cible reçoit ce message, il vérifie si la propriété spéciale est définie. Si la propriété est définie, alors le serveur cible vérifiera son cache de mémoire pour vérifier la présence d'un message reçu avec cette valeur d'en-tête. Si le serveur détecte un message de cette valeur d'en-tête, il ignorera le message envoyé par un client.

Si vous envoyez des messages dans une transaction, alors vous n'aurez pas à définir la propriété pour chaque message que vous envoyez dans cette transaction. Il suffira de le mettre une fois pour toute dans la transaction. Si le serveur détecte un message dupliqué pour chaque message de la transaction, il ignorera l'ensemble de la transaction.

Le nom de la propriété que vous avez définie est donné par la valeur du *org.hornetq.api.core.HDR_DUPLICATE_DETECTION_ID*, qui est *_HQ_DUPL_ID*. La valeur de cette propriété peut être de type `byte[]` ou `SimpleString` pour l'API core. Pour les clients JMS (Java Messaging Service), il doit être de type `String` avec une valeur unique. Une simple façon de créer un id unique consiste à créer un UUID.

L'exemple suivant vous montre comment définir la propriété pour l'API core :

```
...
ClientMessage message = session.createMessage(true);

SimpleString myUniqueID = "This is my unique id"; // Can use a UUID for
this

message.setStringProperty(HDR_DUPLICATE_DETECTION_ID, myUniqueID);

...
```

L'exemple suivant vous montre comment définir la propriété pour les clients JMS :

```
...
Message jmsMessage = session.createMessage();

String myUniqueID = "This is my unique id"; // Could use a UUID for this

message.setStringProperty(HDR_DUPLICATE_DETECTION_ID.toString(),
myUniqueID);

...
```

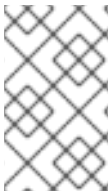
[Rapporter un bogue](#)

18.10.3. Configurer un cache d'ID dupliqué

Le serveur maintient les caches des valeurs reçues de la propriété `org.hornetq.core.message.impl.HDR_DUPLICATE_DETECTION_ID` envoyée à chaque adresse. Chaque adresse maintient son propre cache d'adresses.

Le cache est déterminé par rapport à sa taille. La taille maximum du cache est configurée par le paramètre `id-cache-size` dans les fichiers de configuration du serveur (`standalone.xml` et `domain.xml`). La valeur par défaut de ce paramètre est de 2000 éléments. Si le cache a une taille maximum de n éléments, alors le $(n + 1)$ ème ID stocké remplacera le 0ème élément du cache.

Les caches peuvent également être configurés pour persister dans un disque ou non. On doit pour cela configurer le paramètre `persist-id-cache` dans les fichiers de configuration du serveur (`standalone.xml` et `domain.xml`). Si la valeur est sur "true" alors chaque ID sera persisté en stockage permanent au fur et à mesure qu'il sera reçu. La valeur par défaut de ce paramètre est true.



NOTE

Définir la taille du cache de l'ID dupliqué à un nombre élevé pour pouvoir veiller à ce que le renvoi à nouveau des messages n'écrase pas les messages déjà envoyés qui sont stockés dans le cache.

[Rapporter un bogue](#)

18.10.4. Utilisation de la détection dupliquée par pontages et par connexions de cluster

Des ponts principaux peuvent être configurés pour ajouter automatiquement une valeur d'ID unique en double (si celle-ci n'existe pas déjà dans le message) avant de transférer le message à la cible. Pour configurer un pont principal de détection de duplication de messages, définir la propriété `use-duplicate-detection` sur "true" dans les fichiers de configuration du serveur (`standalone.xml` et `domain.xml`). La valeur par défaut de ce paramètre est "true".

Les connexions de cluster utilisent des ponts principaux en interne pour déplacer les messages entre les noeuds du cluster. Pour configurer une connexion de cluster à la détection de duplication de messages, définir la propriété `use-duplicate-detection` sur "true" dans les fichiers de configuration du serveur (`standalone.xml` et `domain.xml`). La valeur par défaut de ce paramètre est "true".

[Rapporter un bogue](#)

18.11. PONTAGES JMS

18.11.1. Les ponts

La fonction des ponts est de consommer des messages à partir d'une file d'attente de source, et de les envoyer vers une adresse cible, qui se trouve normalement sur un serveur HornetQ. Les ponts s'accommodent de connexions non fiables, et se reconnectent automatiquement quand les connexions sont rendues disponibles à nouveau. Les ponts HornetQ peuvent être configurés avec des expressions de filtre pour n'envoyer que certains messages.



IMPORTANT

Un pont JMS ne peut être déployé vers serveur EAP 6, qui inclut HornetQ configuré comme une sauvegarde dédiée. La raison est que Transaction Manager sur un serveur de sauvegarde dédié est incapable de réparer les opérations précédemment démarrées sur le serveur HornetQ.

[Rapporter un bogue](#)

18.11.2. Créer un pontage JMS

Résumé

Un pontage JMS consomme des messages d'une file d'attente ou une topic JMS source et les envoie à une file d'attente JMS de cible ou sujet, se trouvant en général sur un autre serveur. Il peut être utilisé pour faire un pontage entre les messages entre les serveurs JMS, tant qu'ils sont compatibles avec JMS 1.1. Les ressources JMS de source et de destination sont cherchées à l'aide de JNDI et les classes de client doivent être regroupées dans un module pour la recherche JNDI. Le nom du module est ensuite déclaré dans la configuration de pontage JMS.

Procédure 18.2. Créer un pontage JMS

Cette procédure montre comment configurer un pontage JMS pour faire migrer des messages d'un serveur JBoss EAP 5.x vers un serveur JBoss EAP 6.

1. Configurer le pontage sur le serveur de messagerie JMS source

Configurer le pontage JMS sur le serveur source grâce aux instructions fournies par ce type de serveur. Pour trouver un exemple sur la façon de configurer un pontage JMS sur un serveur JBoss EAP 5.x, voir la rubrique intitulée *Create a JMS Bridged* dans le *Migration Guide* de JBoss EAP 6.

2. Configurer un pontage déployé dans de serveur JBoss EAP 6.x de destination

Dans JBoss EAP 6.1 et dans les versions supérieures, le pontage JMS peut servir à combler des messages depuis n'importe quel serveur compatible JMS 1.1. Comme les ressources JMS sources et cibles sont recherchées par JNDI, les classes de recherche JNDI du fournisseur de messagerie source ou fournisseur de messages doivent être regroupées dans un module de JBoss. Les étapes suivantes utilisent le fournisseur de messages fictive « MyCustomMQ » à titre d'exemple.

a. Créer le module JBoss pour le fournisseur de messagerie.

i. Créer une structure de répertoire sous

`EAP_HOME/modules/system/layers/base/` pour le nouveau module. Le sous-répertoire `main/` contiendra les JAR du client et le fichier `module.xml`. L'exemple suivant est un exemple de structure de répertoires créé pour le fournisseur de messageries MyCustomMQ :

`EAP_HOME/modules/system/layers/base/org/mycustommq/main/`

ii. Dans le sous-répertoire `main/`, créer un fichier `module.xml` qui contienne la définition de module suivante pour le fournisseur de messagerie. Ce qui suit est un exemple de `module.xml` créé pour le fournisseur de messagerie MyCustomMQ.

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.mycustommq">
  <properties>
```

```

        <property name="jboss.api" value="private"/>
    </properties>

    <resources>
        <!-- Insert resources required to connect to the
source or target -->
        <resource-root path="mycustommq-1.2.3.jar" />
        <resource-root path="mylogapi-0.0.1.jar" />
    </resources>

    <dependencies>
        <!-- Add the dependencies required by JMS Bridge code
-->
        <module name="javax.api" />
        <module name="javax.jms.api" />
        <module name="javax.transaction.api"/>
        <!-- Add a dependency on the org.hornetq module since
we send -->
        <!-- messages to the HornetQ server embedded in the
local EAP instance -->
        <module name="org.hornetq" />
    </dependencies>
</module>

```

- iii. Copier les JAR de fournisseur de messagerie requises pour la recherche JNDI des ressources source vers le sous-répertoire `main/` du module. La structure du répertoire du module `MyCustomMQ` ne devra pas ressembler à ce qui suit.

```

modules/
  -- system
    -- layers
      -- base
        -- org
          -- mycustommq
            -- main
              -- mycustommq-1.2.3.jar
              -- mylogapi-0.0.1.jar
              -- module.xml

```

- b. Configurer le pontage JMS dans le sous-système de `messaging` du serveur JBoss EAP.

- i. Avant de commencer, arrêtez le serveur et sauvegardez les fichiers de configuration du serveur actuel. Si vous exécutez un serveur autonome, il s'agira du fichier `EAP_HOME /standalone/configuration/standalone-full-ha.xml`. Si vous exécutez un domaine géré, sauvegardez les fichiers `EAP_HOME /domain/configuration/host.xml` et `EAP_HOME /domain/configuration/domain.xml`.
- ii. Ajouter l'élément `jms-bridge` au sous-système `messaging` dans le fichier de configuration du serveur. Les éléments `source` et `target` procurent les noms des ressources JMS utilisées pour les recherches JNDI. Si les informations d'authentification `user` et `password` sont spécifiées, elles seront passées comme arguments quand une connexion JMS est créée.

Ce qui suit est un exemple d'élément `jms-bridge` configuré pour le fournisseur de messagerie `MyCustomMQ` :

```
<subsystem xmlns="urn:jboss:domain:messaging:1.3">
  ...
  <jms-bridge name="myBridge" module="org.mycustommq">
    <source>
      <connection-factory name="ConnectionFactory"/>
      <destination name="sourceQ"/>
      <user>user1</user>
      <password>pwd1</password>
      <context>
        <property key="java.naming.factory.initial"
value="org.mycustommq.jndi.MyCustomMQInitialContextFactory"/>
        <property key="java.naming.provider.url"
value="tcp://127.0.0.1:9292"/>
      </context>
    </source>
    <target>
      <connection-factory name="java:/ConnectionFactory"/>
      <destination name="/jms/targetQ"/>
    </target>
    <quality-of-service>DUPLICATES_OK</quality-of-service>
    <failure-retry-interval>500</failure-retry-interval>
    <max-retries>1</max-retries>
    <max-batch-size>500</max-batch-size>
    <max-batch-time>500</max-batch-time>
    <add-messageID-in-header>true</add-messageID-in-header>
  </jms-bridge>
</subsystem>
```

Dans l'exemple suivant, les propriétés JNDI sont définies dans l'élément `context` pour la `source`. Si l'élément `context` est omis, comme dans l'exemple `target` ci-dessus, les ressources JMS seront recherchées dans l'instance locale.

[Rapporter un bogue](#)

18.12. PERSISTANCE

18.12.1. Persistance dans HornetQ

HornetQ gère sa propre persistance. Il est livré avec un journal de haute performance, qui est optimisé pour les cas d'utilisation de messagerie spécifique.

Le journal HornetQ `journal` est en «append» (ajout) uniquement avec une taille de fichier configurable, ce qui améliore les performances en permettant des opérations d'écriture simples. Il se compose d'un ensemble de fichiers sur le disque, qui sont initialement pré-crés à une taille fixe et remplis au fur et à mesure que les opérations de serveur (ajouter un message, supprimer le message, mise à jour de message, etc.) sont effectuées, les enregistrements des opérations sont ajoutés au journal jusqu'à ce que le fichier `journal` soit plein, moment à partir duquel le fichier `journal` suivant est utilisé.

Un algorithme de nettoyage de la mémoire détermine si les fichiers journaux peuvent être extraits à nouveau ou réutilisés quand toutes les données auront été supprimées. Un algorithme de compaction supprime les espaces vides des fichiers journaux et compriment les données.

Le journal supporte également les transaction locales et XA.

La majorité du journal est imprimée en Java, mais l'interaction avec le système de fichier a été rendue abstraite pour autoriser plusieurs implémentations enfichables. Les deux implémentations livrées avec HornetQ sont :

- *Java New I/O (NIO)*

Utilise Java NIO pour l'interface avec le système de fichiers. Cela donne une excellence performance et exécute sur n'importe quelle plate-forme avec Java 6 ou un runtime plus récent.

- *Linux Asynchronous IO (AIO)*

Utilise un encapsuleur de code natif pour parler à la bibliothèque d'e/s asynchrone Linux (AIO). Avec AIO, HornetQ reçoit un message lorsque les données ont été rendues persistantes. Cette commande supprime le besoin de synchronisation explicite. AIO fournira généralement une meilleure performance que Java NIO, mais nécessite le noyau Linux 2.6 ou version ultérieure et le package libaio.

AIO nécessite également les systèmes de fichiers ext2, ext3, ext4, jfs ou xfs.

Le serveur standard HornetQ utilise les instances de journaux suivants :

- *bindings journal*

Stocke les données relatives à des liaisons, y compris l'ensemble des files d'attente déployées sur le serveur et leurs attributs. Il stocke également des données comme les compteurs de séquence ID. Le journal de liaisons est toujours un journal NIO, car il a généralement un débit faible en comparaison au journal de messages.

Les fichiers de ce journal ont comme préfixe hornetq-bindings. Chaque fichier a des extensions de liaison. La taille du fichier est de 1048576 octets, et le fichier se trouve dans le dossier de liaisons.

- *JMS journal*

Stocke toutes les données liées à JMS, comme les files d'attente JMS, les sujets ou fabriques de connexions et toutes les liaisons JNDI de ces ressources. Toutes les ressources JMS créées avec l'API de gestion sont persistées dans ce journal. Toutes les ressources configurées par des fichiers de configuration ne le sont pas. Ce journal n'est créé que si JMS est utilisé.

- *message journal*

Stocke toutes les données liées à des messages, y compris les messages eux-mêmes et les duplicate-id caches. Par défaut, HornetQ utilise AIO pour son journal. Si AIO n'est pas disponible, ce sera automatiquement NIO.

Les gros messages sont persistés en dehors du journal de messages. Dans les situations de moindre mémoire, configurer HornetQ pour qu'il envoie les messages sur le disque. Si la persistance n'est pas requise, HornetQ peut être configuré pour ne persister aucune donnée.

[Rapporter un bogue](#)

18.13. HORNETQ CLUSTERING

Les clusters HornetQ sont utilisés pour créer des groupes de serveurs HornetQ qui partagent la charge de traitement des messages. Chaque noeud actif du cluster agit comme serveur HornetQ indépendant et gère ses propres messages et connexions.

Pour former un cluster, chaque noeud (serveur indépendant HornetQ) déclare des connexions au cluster par un autre noeud avec des paramètres de configuration dans les fichiers de configuration (`standalone.xml` et `domain.xml`).

En clustering, des ponts principaux sont utilisés pour le pontage/routage des messages d'un cluster à l'autre. Les ponts principaux consomment des messages à partir d'une file d'attente de la source et ensuite, transfèrent ces messages vers un serveur cible HornetQ (noeud) qui peut ou peut ne pas être dans le même cluster.

Lorsqu'un noeud forme une connexion avec un autre noeud de cluster, il crée un pont principal en interne. Chaque noeud crée un pont principal explicite et vous n'avez pas besoin de le déclarer. Ces connexions au cluster permettent la transmission des messages entre les noeuds dans divers clusters d'équilibrage de charge de traitement des messages.

Vous pouvez configurer des noeuds de cluster dans les fichiers de configuration du serveur (`standalone.xml` et `domain.xml`).



IMPORTANT

Vous pouvez configurer un noeud par le biais de fichiers de configuration du serveur (`standalone.xml` et `domain.xml`) et copier cette configuration aux autres noeuds pour générer un cluster symétrique. Toutefois, vous devez être prudent lorsque vous copiez les fichiers de configuration de serveur. Vous ne devez pas copier les données de HornetQ (c'est-à-dire les liaisons, le journal et les répertoires de messages volumineux) d'un noeud à l'autre. Lorsqu'un noeud est démarré pour la première fois, il persiste un identificateur unique pour le répertoire de journal qui sert à la bonne formation des clusters.

[Rapporter un bogue](#)

18.13.1. Server Discovery

Les serveurs utilisent un mécanisme qui s'appelle "server discovery" pour :

- Transmettre leurs informations de connexion aux clients de messagerie : le but de la messagerie clients est de se connecter aux serveurs d'un cluster sans détails précis sur les serveurs en cours d'exécution à un moment donné
- Se connecter aux autres serveurs : les serveurs d'un cluster veulent établir des connexions au cluster avec d'autres serveurs, sans détails précis sur les autres serveurs d'un cluster

L'information sur les serveurs est envoyée aux clients de la message par l'intermédiaire de connexions HornetQ habituelles et aux autres services par l'intermédiaire des connexions du cluster.

La première connexion doit être établie et peut être établie par les techniques de découverte de serveur dynamique (discovery) comme UDP (User Datagram Protocol), JGroups ou en fournissant une liste de connecteurs.

[Rapporter un bogue](#)

18.13.2. Broadcast Groups

Des connecteurs sont utilisés sur le client pour définir comment et de quelle manière il se connecte au serveur. Les serveurs utilisent des groupes de diffusion (broadcast groups) pour diffuser des connecteurs sur le réseau. Le groupe de diffusion prend un groupe de paires de connecteur et les diffuse sur le réseau. Chaque paire de connecteurs contient des paramètres de connexion pour un serveur live et de sauvegarde.

Vous pouvez définir des groupes de diffusion dans l'élément *broadcast-groups* des fichiers de configuration du serveur (*standalone.xml* et *domain.xml*). Un seul serveur HornetQ peut posséder plusieurs groupes de diffusion. Vous pouvez définir un UDP (User Datagram Protocol) ou un groupe de diffusion JGroup.

Rapporter un bogue

18.13.2.1. Groupe de diffusion UDP (User Datagram Protocol)

L'exemple ci-dessous montre comment définir un groupe de diffusion UDP :

```
<broadcast-groups>
  <broadcast-group name="my-broadcast-group">
    <local-bind-address>172.16.9.3</local-bind-address>
    <local-bind-port>5432</local-bind-port>
    <group-address>231.7.7.7</group-address>
    <group-port>9876</group-port>
    <broadcast-period>2000</broadcast-period>
    <connector-ref>netty</connector-ref>
  </broadcast-group>
</broadcast-groups>
```



NOTE

Dans l'exemple de configuration ci-dessus, les attributs "local-bind-address", "local-bind-port", "group-address" et "group-port" sont dépréciés. Vous pourrez choisir d'utiliser l'attribut "socket-binding" à la place.

L'exemple indiqué ci-dessous définit un groupe de diffusion UDP qui remplace tous les attributs dépréciés par l'attribut "socket-binding" :

```
<broadcast-groups>
  <broadcast-group name="my-broadcast-group">
    <socket-binding>messaging-group</socket-binding>
    <broadcast-period>2000</broadcast-period>
    <connector-ref>netty</connector-ref>
  </broadcast-group>
</broadcast-groups>
```

Le tableau ci-dessous décrit tous les paramètres importants utilisés dans les exemples ci-dessus et, qui servent, en général, à définir un groupe de diffusion UDP :

Tableau 18.11. Paramètres de groupe de diffusion UDP

Attribut	Description
----------	-------------

Attribut	Description
attribut de nom	Dénote le nom de chaque groupe de diffusion d'un serveur. Chaque groupe de diffusion doit posséder un nom unique.
local-bind-address	[Déprécié] C'est un attribut spécifique UDP qui spécifie l'adresse de liaison locale à laquelle se relie le paquet de datagramme. Vous devez définir cette propriété pour définir l'interface que vous souhaitez utiliser pour vos diffusions. Si cette propriété n'est pas spécifiée, alors la liaison se relie à une adresse générique (une adresse générée par le noyau au hasard).
local-bind-port	[Déprécié] C'est un attribut spécifique UDP qui spécifie un port local auquel le socket de datagramme se relie. Une valeur "-1" par défaut indique un port anonyme à utiliser.
group-address	[Déprécié] C'est une adresse multidiffusion spécifique à UDP où les messages sont diffusés. Cette adresse IP possède un large éventail qui va de 224.0.0.0 à 239.255.255.255, inclus. L'adresse IP de 224.0.0 est réservée et ne peut pas être utilisée.
group-port	[Déprécié] Dénote le numéro de port UDP de diffusion.
socket-binding	Dénote la liaison de socket de groupe de diffusion
broadcast-period	Ce paramètre indique la durée entre deux diffusion (en millisecondes). Optionnel.
connector-ref	Se réfère au connecteur qui sera diffusé.

Rapporter un bogue

18.13.2.2. Groupe de diffusion JGroups

Vous pouvez utiliser JGroups pour la diffusion en spécifiant deux attributs *jgroups-stack* et *jgroups-channel*. L'exemple ci-dessous définit un groupe de diffusion JGroups :

```
<broadcast-groups>
  <broadcast-group name="bg-group1">
    <jgroups-stack>udp</jgroups-stack>
    <jgroups-channel>udp</jgroups-channel>
    <broadcast-period>2000</broadcast-period>
    <connector-ref>netty</connector-ref>
  </broadcast-group>
</broadcast-groups>
```

La définition du groupe de diffusion JGroups utilise deux attributs principaux :

- L'attribut *jgroups-stack* qui indique le nom de la pile définie dans le sous-système

org.jboss.as.clustering.jgroups

- L'attribut *jgroups-channel* qui indique le canal auquel les canaux de JGroups se connectent pour la multidiffusion

[Rapporter un bogue](#)

18.13.3. Les groupes discovery

Les groupes de diffusion sont utilisés pour diffuser les connecteurs sur le réseau. D'autre part, les groupes discovery définissent comment l'information sur les connecteurs est reçue à partir des points de terminaison de diffusion (Groupes de diffusion JGroups ou UDP). Un groupe discovery maintient une liste de paires de connecteurs, un par diffusion par un serveur différent.

Quand un groupe discovery reçoit des diffusions à partir d'un point de terminaison pour un serveur particulier, il met à jour les paires de connecteurs dans la liste du serveur spécifique. S'il ne reçoit pas de diffusion en provenance d'un serveur spécifique pendant un certain temps, il supprime l'entrée du serveur de la liste.

Les groupes discovery sont surtout utilisés par les connexions du cluster et les clients JMS (Java Messaging Service) afin d'obtenir les informations de connexion de départ pour pouvoir télécharger la topologie qui convient.



NOTE

Vous devez configurer chaque groupe discovery avec un point de terminaison de diffusion qui convient correspondant à son homologue de groupe de diffusion (UDP ou JGroups).

[Rapporter un bogue](#)

18.13.3.1. Configurer un groupe de diffusion UDP (User Datagram Protocol) sur le serveur

L'exemple ci-dessous montre comment définir un groupe discovery UDP :

```
<discovery-groups>
  <discovery-group name="my-discovery-group">
    <local-bind-address>172.16.9.7</local-bind-address>
    <group-address>231.7.7.7</group-address>
    <group-port>9876</group-port>
    <refresh-timeout>10000</refresh-timeout>
  </discovery-group>
</discovery-groups>
```



NOTE

Dans l'exemple de configuration ci-dessus, les attributs "local-bind-address", "group-address" et "group-port" sont dépréciés. Vous pourrez choisir d'utiliser l'attribut "socket-binding" à la place.

L'exemple indiqué ci-dessous définit un groupe de discovery UDP qui remplace tous les attributs dépréciés par l'attribut "socket-binding" :

```
<discovery-groups>
  <discovery-group name="my-discovery-group">
    <socket-binding>messaging-group</socket-binding>
    <refresh-timeout>10000</refresh-timeout>
  </discovery-group>
</discovery-groups>
```

Le tableau ci-dessous décrit tous les paramètres importants utilisés dans l'exemple ci-dessus et, qui servent, en général, à définir un groupe de diffusion :

Tableau 18.12. Paramètres de groupe discovery UDP

Attribut	Description
name attribute	Cet attribut indique le nom du groupe discovery. Chaque nom discovery doit avoir un nom unique par serveur.
local-bind-address	[Déprécié] C'est un attribut spécifique UDP optionnel. Il est utilisé pour configurer un groupe discovery pour qu'il écoute une interface spécifique lors de l'utilisation d'interfaces multiples sur la même machine.
group-address	[Déprécié] C'est un attribut spécifique UDP obligatoire. Il est utilisé pour configurer un groupe discovery pour qu'il écoute une adresse IP spécifique de groupe. La valeur de cet attribut doit correspondre à l'attribut group-address du groupe de diffusion que vous souhaitez écouter.
group-port	[Déprécié] C'est un attribut spécifique UDP obligatoire. Il est utilisé pour configurer le port UDP du groupe multidiffusion. La valeur de cet attribut doit correspondre à l'attribut group-port du groupe de diffusion que vous souhaitez écouter.
socket-binding	Dénote la liaison de socket de groupe discovery
refresh-timeout	Il s'agit d'un attribut spécifique UDP facultatif. Il est utilisé pour configurer la durée (en millisecondes) pendant laquelle le groupe discovery patientera avant de retirer l'entrée de paire de connecteurs d'un serveur de la liste après avoir reçu la dernière diffusion en provenance de ce serveur. La valeur de refresh-timeout doit être définie à un niveau nettement supérieur à la valeur de l'attribut broadcast-period sur le groupe de diffusion pour empêcher le déplacement rapide de serveurs de la liste lorsque le processus de diffusion est encore actif. La valeur par défaut de cet attribut est de 10 000 millisecondes.

[Rapporter un bogue](#)

18.13.3.2. Configurer un groupe discovery JGroups sur le serveur

L'exemple ci-dessous montre comment définir un groupe discovery JGroups :

```
<discovery-groups>
  <discovery-group name="dg-group1">
    <jgroups-stack>udp</jgroups-stack>
    <jgroups-channel>udp</jgroups-channel>
    <refresh-timeout>10000</refresh-timeout>
  </discovery-group>
</discovery-groups>
```

La définition du groupe discovery JGroups utilise deux attributs principaux :

- L'attribut *jgroups-stack* qui indique le nom de la pile définie dans le sous-système `org.jboss.as.clustering.jgroups`
- L'attribut *jgroups-channel* qui indique le canal auquel les canaux de JGroups se connectent pour la multidiffusion



NOTE

Les attributs JGroups et les attributs spécifiques UDP sont exclusifs. Vous pouvez utiliser les groupes d'attributs JGroups ou UDP dans la configuration d'un groupe discovery ou de diffusion

[Rapporter un bogue](#)

18.13.3.3. Configurer les groupes discovery pour les clients JMS (Java Messaging Service)

Les groupes discovery peuvent être configurés pour les clients JMS et pour les clients principaux. Vous pouvez spécifier le groupe discovery à utiliser pour une usine de connexions JMS dans les fichiers de configuration de serveur (`standalone.xml` et `domain.xml`) :

```
<connection-factory name="ConnectionFactory">
  <discovery-group-ref discovery-group-name="my-discovery-group"/>
  <entries>
    <entry name="ConnectionFactory"/>
  </entries>
</connection-factory>
```

L'élément *discovery-group-ref* est utilisé pour spécifier le nom d'un groupe discovery. Lorsqu'une application client télécharge cette usine de connexions de JNDI (Java Naming and Directory Interface) et crée des connexions JMS, ces connexions sont équilibrées sur tous les serveurs que le groupe discovery préserve en écoutant l'adresse de multidiffusion spécifiée dans la configuration de groupe discovery.

Si vous utilisez JMS et non pas JNDI pour chercher une usine de connexions, alors vous pourrez spécifier les paramètres de groupe de discovery directement quand vous créez l'usine de connexions JMS :

```
final String groupAddress = "231.7.7.7";
final int groupPort = 9876;
ConnectionFactory jmsConnectionFactory =
HornetQJMSClient.createConnectionFactory(new
DiscoveryGroupConfiguration(groupAddress, groupPort, new
```

```

UDPBroadcastGroupConfiguration(groupAddress, groupPort, null, -1)),
JMSFactoryType.CF);
Connection jmsConnection1 = jmsConnectionFactory.createConnection();
Connection jmsConnection2 = jmsConnectionFactory.createConnection();

```

La valeur par défaut de l'attribut *refresh-timeout* peut être définie sur `DiscoveryGroupConfiguration` en utilisant la méthode setter `setDiscoveryRefreshTimeout()`. Pour que la fabrique de connexions attende un certain temps avant de créer la première connexion, vous pouvez utiliser la méthode setter `setDiscoveryInitialWaitTimeout()` sur `DiscoveryGroupConfiguration`.

Cela garantit que la fabrique de connexions ait suffisamment de temps pour recevoir les diffusions de tous les nœuds du cluster. La valeur par défaut de ce paramètre est de 10 000 millisecondes.

[Rapporter un bogue](#)

18.13.3.4. Configuration de discovery pour l'API principal

Si vous utilisez l'API principal pour instancier les instances de *ClientSessionFactory*, alors vous pouvez spécifier les paramètres du groupe discovery directement quand vous créez une usine de session :

```

final String groupAddress = "231.7.7.7";
final int groupPort = 9876;
ServerLocator factory = HornetQClient.createServerLocatorWithHA(new
DiscoveryGroupConfiguration(groupAddress, groupPort, new
UDPBroadcastGroupConfiguration(groupAddress, groupPort, null, -1)));
ClientSessionFactory factory = locator.createSessionFactory();
ClientSession session1 = factory.createSession();
ClientSession session2 = factory.createSession();

```

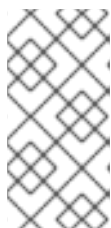
La valeur par défaut de l'attribut *refresh-timeout* peut être définie sur `DiscoveryGroupConfiguration` en utilisant la méthode setter `setDiscoveryRefreshTimeout()`. Vous pouvez utiliser la méthode setter `setDiscoveryInitialWaitTimeout()` sur `DiscoveryGroupConfiguration` pour que l'usine de sessions patiente pendant un certain temps avant de créer une session.

[Rapporter un bogue](#)

18.13.4. Équilibrage des charges côté serveur

Il y a une topologie de clusterisation importante :

- **Cluster symétrique** : dans un cluster symétrique, chaque nœud de cluster est connecté directement à chaque nœud du cluster. Pour créer un cluster symétrique, chaque nœud du cluster définit une connexion de cluster avec l'attribut *max-hops* défini sur 1.



NOTE

Dans un cluster symétrique, chaque nœud connaît toutes les files d'attente qui existent sur tous les autres nœuds et les consommateurs qu'ils ont. Avec ces connaissances, il peut déterminer comment équilibrer la charge et redistribuer les messages vers les nœuds.

[Rapporter un bogue](#)**18.13.4.1. Configuration des connexions du cluster**

Les connexions du cluster sont configurées dans les fichiers de configuration du serveur (`standalone.xml` et `domain.xml`) dans l'élément `cluster-connection`. Il peut y avoir zéro ou plusieurs connexions définies par serveur HornetQ.

```
<cluster-connections>
  <cluster-connection name="my-cluster">
    <address>jms</address>
    <connector-ref>netty-connector</connector-ref>
    <check-period>1000</check-period>
    <connection-ttl>5000</connection-ttl>
    <min-large-message-size>50000</min-large-message-size>
    <call-timeout>5000</call-timeout>
    <retry-interval>500</retry-interval>
    <retry-interval-multiplier>1.0</retry-interval-multiplier>
    <max-retry-interval>5000</max-retry-interval>
    <reconnect-attempts>-1</reconnect-attempts>
    <use-duplicate-detection>>true</use-duplicate-detection>
    <forward-when-no-consumers>>false</forward-when-no-consumers>
    <max-hops>1</max-hops>
    <confirmation-window-size>32000</confirmation-window-size>
    <call-failover-timeout>30000</call-failover-timeout>
    <notification-interval>1000</notification-interval>
    <notification-attempts>2</notification-attempts>
    <discovery-group-ref discovery-group-name="my-discovery-group"/>
  </cluster-connection>
</cluster-connections>
```

Le tableau suivant décrit les attributs configurables :

Tableau 18.13. Attributs configurables de connexions de cluster

Attribut	Description	Par défaut
<i>address</i>	Chaque connexion de cluster ne s'applique qu'aux messages envoyés à une adresse qui commence par cette valeur. L'adresse peut être n'importe quelle valeur et vous pouvez avoir plusieurs connexions au cluster avec des valeurs différentes des adresses, équilibrant simultanément les messages de ces adresses, potentiellement à différents clusters de serveurs. Cela n'utilise pas de correspondance de wild-card.	

Attribut	Description	Par défaut
<i>connector-ref</i>	C'est un attribut obligatoire qui fait référence à un connecteur envoyé dans d'autres noeuds du cluster pour qu'ils adoptent la topologie de cluster qui convient.	
<i>check-period</i>	Il s'agit d'une durée (en millisecondes) utilisée pour vérifier si une connexion de cluster a manqué des pings en provenance d'un autre cluster.	30 000 millisecondes
<i>connection-ttl</i>	Indique la durée pendant laquelle une connexion de cluster doit rester vivante s'il cesse de recevoir des messages en provenance d'un noeud spécifique du cluster.	60 000 millisecondes
<i>min-large-message-size</i>	Si la taille du message (en octets) est plus élevée que cette valeur, alors il sera divisé en plusieurs segments après avoir été envoyé à d'autres membres du cluster du réseau.	102400 octets
<i>call-timeout</i>	Indique la durée (en millisecondes) pendant laquelle un paquet envoyé sur une connexion de cluster doit attendre (une réponse) avant de lancer une exception.	30 000 millisecondes
<i>retry-interval</i>	Si la connexion de cluster est créée entre les noeuds d'un cluster et que le noeud cible n'a pas été démarré ou doit être redémarré, les connexions de cluster d'autres noeuds retenteront de se reconnecter à la cible jusqu'à ce qu'il revienne. Le paramètre <i>retry-interval</i> définit l'intervalle (en millisecondes) entre les tentatives.	500 millisecondes
<i>retry-interval-multiplier</i>	Utilisé pour incrémenter <i>retry-interval</i> après chaque tentative	1

Attribut	Description	Par défaut
<i>max-retry-interval</i>	Se réfère à la durée maximum (en millisecondes) des nouvelles tentatives	2000 millisecondes
<i>reconnect-attempts</i>	Définit le nombre de fois que le système va essayer de connecter un noeud au cluster	-1 (infinite retries)
<i>use-duplicate-detection</i>	Les connexions au cluster utilisent des ponts pour relier les nœuds, et les ponts peuvent être configurés pour ajouter une propriété id en double dans chaque message qui est transmis. Si le nœud cible du pont se bloque et puis récupère, les messages peuvent être renvoyés à partir du nœud de la source. En permettant la détection des doublons, tous les messages en double vont être filtrés et ignorés lors de la réception au niveau du nœud cible.	True
<i>forward-when-no-consumers</i>	Ce paramètre détermine si oui ou non les messages seront distribués de façon alternée entre d'autres nœuds du cluster indépendamment du fait qu'ils puissent correspondre à un consommateur sur d'autres nœuds	False
<i>max-hops</i>	Détermine comment les charges de messages sont équilibrées sur d'autres serveurs HornetQ connectés à ce serveur.	-1
<i>confirmation-window-size</i>	La taille (en octets) de la fenêtre utilisée pour envoyer des confirmations du serveur connecté à	1048576
<i>call-failover-timeout</i>	Utilisé quand un appel est fait lors d'une tentative de basculement	-1 (no timeout)

Attribut	Description	Par défaut
<i>notification-interval</i>	Détermine la fréquence (en millisecondes) à laquelle la connexion du cluster doit se diffuser quand elle s'attache au cluster.	1000 millisecondes
<i>notification-attempts</i>	Définit le nombre de fois que la connexion du cluster doit se diffuser quand elle se connecte au cluster.	2
<i>discovery-group-ref</i>	Ce paramètre détermine quel groupe discovery est utilisé pour obtenir la liste des autres serveurs du cluster vers laquelle la connexion de cluster va faire des connexions	

Lorsque vous créez des connexions entre les noeuds d'un cluster pour former une connexion de cluster, HornetQ utilise un cluster utilisateur et un mot de passe de cluster qui est défini dans les fichiers de configuration de serveur (`standalone.xml` et `domain.xml`) :

```
<cluster-user>HORNETQ.CLUSTER.ADMIN.USER</cluster-user>
<cluster-password>NEW USER</cluster-password>
```



AVERTISSEMENT

Il est important de changer les valeurs par défaut de ces informations d'identification pour empêcher les clients distants d'effectuer les connexions au serveur en utilisant les valeurs par défaut.

[Rapporter un bogue](#)

18.14. HAUTE DISPONIBILITÉ

18.14.1. Introduction à la haute disponibilité

HornetQ supporte la possibilité de continuer à fonctionner après un basculement d'un ou de plusieurs serveurs. Ceci est en partie réalisé grâce au support de basculement lorsque des connexions de clients migrent d'un serveur live vers un serveur de sauvegarde en cas de basculement de serveur live. Pour conserver le serveur de sauvegarde actuel, les messages sont répliqués du serveur live vers le serveur de sauvegarde en continu par deux stratégies: store partagé et réplication.

Il existe deux types de topologies en haute disponibilité :

- **Topologie dédiée** : cette topologie est composée de deux serveurs EAP. Dans le premier serveur, HornetQ est configuré comme un serveur live. Dans le second serveur, HornetQ est configuré comme un serveur de sauvegarde. Le serveur EAP ayant HornetQ configuré comme un serveur de sauvegarde agit seulement en tant que conteneur pour HornetQ. Ce serveur est inactif et ne peut pas héberger des déploiements comme les EJB, les MDB ou des servlets.
- **Topologie colocalisée** : cette topologie contient deux serveurs d'EAP. Chaque serveur EAP contient deux serveurs de HornetQ (un serveur live et un serveur de sauvegarde). Le serveur HornetQ sur le premier serveur EAP et le serveur de sauvegarde de HornetQ sur le second serveur EAP forment une paire de serveurs live de sauvegarde, tandis que le serveur live HornetQ sur le second serveur EAP et le serveur de sauvegarde HornetQ sur le premier serveur EAP forment une autre paire de serveurs de sauvegarde live.

En topologie colocalisée, dès qu'un serveur HornetQ (faisant partie de la paire live-sauvegarde) échoue, le serveur de sauvegarde HornetQ prend la relève et redevient actif. Lorsque le serveur de sauvegarde HornetQ s'arrête en cas de restauration automatique (failback), alors les usines de connexions et destinations configurées sur le serveur de sauvegarde sont dissociées de JNDI (Java Naming and Directory Interface).

L'interface JNDI (Java Naming and Directory Interface) est partagée avec l'autre serveur live HornetQ (faisant partie de l'autre paire live-sauvegarde). Ainsi, la dissociation des usines de connexion et destinations de JNDI a pour effet de dissocier également les fabriques de connexions et de destinations de ce serveur live HornetQ.



IMPORTANT

La configuration des serveurs de sauvegarde colocalisés ne peuvent pas contenir de configurations de destination ou d'usines de connexion.



NOTE

Les informations suivantes référencent `standalone-full-ha.xml`. Les changements de configuration peuvent s'appliquer à `standalone-full-ha.xml` ou à tout fichier de configuration dérivé.

[Rapporter un bogue](#)

18.14.2. HornetQ Shared Stores

Lorsque vous utilisez un magasin partagé (Shared Store), les serveurs live et de sauvegarde partagent le répertoire de données même, ensemble, à l'aide d'un système de fichiers partagé. Cela inclut le répertoire de pagination, le répertoire de journaux, des messages volumineux et le journal de liaison. Lorsque le basculement et le serveur de sauvegarde reprennent, il chargent le stockage persistant de système de fichiers partagé. Les clients peuvent alors s'y connecter.

Cette forme de haute disponibilité diffère de la réplication de données, car elle requiert que le système de fichiers soit accessible à la fois par les noeuds de sauvegarde live et de sauvegarde. Cela correspondra le plus souvent à un SAN de haute performance.

L'avantage de share-store haute disponibilité est qu'aucune réplication ne se produit entre les nœuds live et de sauvegarde. Autrement dit, il n'y a pas de dégradation des performances en raison de la surcharge de réplication pendant le fonctionnement normal.

L'inconvénient la réplication d'un shared-store est qu'elle nécessite un système de fichiers partagé, et que lorsque le serveur de sauvegarde est activé, il faut charger le journal à partir d'un shared-store. Cela peut prendre un certain temps, selon la quantité de données dans le store.

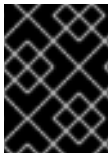
S'il est exigé d'avoir des performances élevées durant le fonctionnement normal, il y a accès à un réseau SAN rapide et un taux de basculement légèrement plus lent est acceptable (en fonction de la quantité de données). Shared-store haute disponibilité est recommandé.

[Rapporter un bogue](#)

18.14.3. Configurations de stockage d'HornetQ

HornetQ prend en charge deux styles de configuration différents pour les stores partagés :

- GFS2 sur un SAN, pour le type de journal ASYNCIO.
- NFSv4, pour les types de journaux ASYNCIO ou NIO.



IMPORTANT

NFS est pris en charge sous des directives strictes de configuration qui sont soulignées ci-dessous.

L'implémentation NFS de Red Hat Linux utilisée supporte à la fois le direct I/O (ouverture des fichiers avec l'indicateur `O_DIRECT` défini), et l'I/O asynchrone basé noyau. Avec ces deux fonctionnalités présentes, il est possible d'utiliser NFS comme option de stockage, sous conditions de configuration strictes :

- Le cache client Red Hat Enterprise Linux NFS doit être désactivé.



NOTE

Il est recommandé que, si vous utilisez NFS en vertu des stipulations ci-dessus, une configuration NFS hautement disponible soit utilisée.

[Rapporter un bogue](#)

18.14.4. Types de journaux HornetQ

Deux types de journaux sont disponibles pour HornetQ :

- ASYNCIO
- NIO

Le type de journal ASYNCIO, également connu sous le nom de AIO, est un wrapper de code natif thin de bibliothèque Linux d'e/s asynchrones (AIO). L'utilisation de fonctionnalités natives peut fournir de meilleures performances que NIO. Ce type de journal n'est supporté que dans Red Hat Enterprise Linux et exige que `libaio` et le package de composants natifs soient installés où JBoss EAP 6 est en cours d'exécution. Consultez le *Guide d'Installation* pour obtenir des instructions sur l'installation du package de composants natifs.



IMPORTANT

Consultez le journal de serveur après le démarrage de JBoss EAP 6, pour vous assurer que la bibliothèque native est chargée avec succès, et que le type de journal ASYNCIO est utilisé. Si la bibliothèque native ne se charge pas, HornetQ retournera à un type de journal NIO, et cela sera précisé dans le journal du serveur.

Le type de journal NIO utilise Java NIO pour créer l'interface avec le système de fichiers. Il fournit une très bonne performance et exécute sur toutes les plateformes prises en charge.

Pour spécifier le type de journal HornetQ, définir le paramètre `<journal-type>` dans le sous-système de `Messaging`.

[Rapporter un bogue](#)

18.14.5. Configurer HornetQ avec une topologie dédiée et un store partagé

Pour configurer les serveurs live ou de backup des stores partagés dans la topologie dédiée, configurer les fichiers `standalone-X.xml` sur chaque serveur avec ce qui suit :

```
<shared-store>>true</shared-store>
<paging-directory path="${shared.directory}/journal"/>
<bindings-directory path="${shared.directory}/bindings"/>
<journal-directory path="${shared.directory}/journal"/>
<large-messages-directory path="${shared.directory}/large-messages"/>
.
.
.
<cluster-connections>
  <cluster-connection name="my-cluster">
    ...
  </cluster-connection>
</cluster-connections>
```

Tableau 18.14. Les attributs de configuration des serveurs HornetQ (pour serveurs live et de backup à la fois)

Attribut	Description
shared-store	Indique si le serveur utilise un store partagé ou non. La valeur par défaut est false.
paging-directory path	Indique le chemin d'accès vers le répertoire de pagination. Ce chemin est le même pour les serveurs live ou de backup car ils partagent ce répertoire
bindings-directory path	Indique le chemin d'accès vers le journal des liaisons. Ce chemin est le même pour les serveurs live ou de backup car ils partagent ce journal.
journal-directory path	Indique le chemin d'accès vers le répertoire de journalisation. Ce chemin est le même pour les serveurs live ou de backup car ils partagent ce répertoire

Attribut	Description
large-messages-directory path	Indique le chemin d'accès vers le répertoire de messages volumineux Ce chemin est le même pour les serveurs live ou de backup car ils partagent ce répertoire
failover-on-shutdown	Indique si ce serveur devient actif quand un serveur de sauvegarde actuellement actif se ferme

Le serveur de sauvegarde doit également être marqué explicitement en tant que serveur de sauvegarde.

```
<backup>true</backup>
```

L'attribut de configuration dédié au serveur de sauvegarde d'HornetQ est : **allow-failback**. Il indique si le serveur de sauvegarde se ferme automatiquement quand le serveur live d'origine est de retour.

[Rapporter un bogue](#)

18.14.6. La réplication de messages HornetQ



AVERTISSEMENT

Seuls les messages persistés peuvent être répliqués. Tout message non persistant ne peut pas survivre à un basculement.

La réplication de messages entre un serveur direct et un serveur de sauvegarde est effectué par le biais du trafic réseau car les serveurs live et de sauvegarde ne partagent pas les mêmes stores de données. Toutes les revues sont répliquées entre les deux serveurs, tant que les deux serveurs sont dans le même cluster et ont le même nom d'utilisateur et mot de passe de cluster. Tout le trafic de données persistantes reçu par le serveur live est répliqué sur le serveur de sauvegarde.

Quand le serveur de sauvegarde est en ligne, il cherche à trouver et à se connecter à un serveur live pour tenter la synchronisation. Une fois synchronisé, il n'est plus disponible en tant que serveur de sauvegarde. La synchronisation peut prendre un long moment selon le volume de données à synchroniser et la vitesse du réseau. Si le serveur de sauvegarde apparaît en ligne et qu'aucun serveur live n'est disponible, le serveur de sauvegarde attendra jusqu'à ce que le serveur live soit disponible dans le cluster.

Pour activer les serveurs qui répliquent les données, il faudra définir un lien entre eux dans le fichier `standalone-full-ha.xml`. Un serveur de sauvegarde ne se répliquera qu'avec un serveur live du même nom de groupe. Le nom de groupe doit être défini dans le paramètre **backup-group-name** qui se trouve dans le fichier `standalone-full-ha.xml` de chaque serveur.

Dans le cas d'un serveur live ayant échoué, le serveur de sauvegarde correctement configuré et totalement synchronisé reprendra ses fonctions. Le serveur de sauvegarde ne s'activera que si le

serveur live a échoué, et si le serveur de sauvegarde est en mesure de se connecter à plus de la moitié des serveurs dans le cluster. Si plus de la moitié des autres serveurs du cluster manquent également de répondre, cela indique une panne générale de réseau et le serveur de sauvegarde attendra pour réessayer la connexion au serveur live.

Pour accéder à l'état initial après un basculement, il faut démarrer le serveur et attendre qu'il soit entièrement synchronisée avec le serveur de sauvegarde. Lorsque cela aura été réalisé, vous pourrez arrêter le serveur de sauvegarde pour que le serveur de départ s'active à nouveau. Cela se fait automatiquement si l'attribut *allow-failback* est défini sur true.

[Rapporter un bogue](#)

18.14.7. Configurer les serveurs HornetQ pour la réplication

Pour configurer les serveurs live et de sauvegarde en tant que paire de réplication, configurer les fichiers `standalone-full-ha.xml` sur chaque serveur pour qu'ils aient les paramètres de configuration suivants :

```
<shared-store>>false</shared-store>
<backup-group-name>NameOfLiveBackupPair</backup-group-name>
<check-for-live-server>true</check-for-live-server>
.
.
.
<cluster-connections>
  <cluster-connection name="my-cluster">
    ...
  </cluster-connection>
</cluster-connections>
```

Tableau 18.15. Attributs de configuration pour la réplication HornetQ

Attribut	Description
shared-store	Indique si le serveur utilise un store partagé ou non. La valeur par défaut est false.
backup-group-name	Il s'agit du nom unique qui identifie une paire live/sauvegarde qui doivent se répliquer ensemble
check-for-live-server	Indique si un serveur répliqué live doit vérifier le cluster actuel pour voir s'il existe déjà un serveur live avec le même id de nœud. La valeur par défaut est false.
failover-on-shutdown	Indique si ce serveur de sauvegarde (s'il s'agit d'un serveur de sauvegarde) doit devenir le serveur live lors d'un arrêt normal du serveur. La valeur par défaut est false.

Le serveur de sauvegarde doit également être marqué explicitement en tant que serveur de sauvegarde.

```
<backup>true</backup>
```

Tableau 18.16. Attributs de configuration de serveur de sauvegarde HornetQ

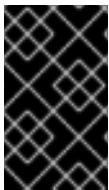
Attribut	Description
allow-failback	Indique si le serveur de sauvegarde se fermera automatiquement si le serveur live d'origine revient. La valeur par défaut est true.
max-saved-replicated-journal-size	Le nombre maximal de journaux de sauvegarde à conserver après que la restauration automatique se soit produite. La spécification de cet attribut n'est nécessaire que si allow-failback est sur true. La valeur par défaut est 2, ce qui signifie que, après 2 restaurations, le serveur de sauvegarde doit être redémarré pour pouvoir répliquer le journal du serveur live et redevenir la sauvegarde.

[Rapporter un bogue](#)

18.14.8. High-availability (HA) Failover

High-availability Failover est disponible, soit en basculement automatique des clients, ou en basculement au niveau des applications, grâce à une structure de live-backup. Chaque serveur live a un serveur de sauvegarde. Une sauvegarde par serveur live uniquement est prise en charge.

Le serveur de sauvegarde ne prend le relais que si le serveur se plante ou en cas de basculement. Après que le serveur ait été redémarré, et si l'attribut de `allow-failback` est défini sur true, il redevient le serveur live. Lorsque le serveur live d'origine prend la relève, le serveur de sauvegarde rétablit sa fonction de sauvegarde pour le serveur live.



IMPORTANT

La mise en cluster doit être activée même si vous n'utilisez pas les fonctionnalités de clustering. C'est parce que chaque nœud du cluster HA doit avoir une connexion de cluster pour tous les autres nœuds, afin de négocier des rôles avec d'autres serveurs.

La topologie de cluster haute disponibilité est atteinte par les serveurs directs et de sauvegarde car ils envoient des informations sur leurs informations de connexion en utilisant la multidiffusion IP. Si la multidiffusion IP ne peut pas être utilisée, il est également possible d'utiliser une configuration statique des connexions initiales. Après la connexion initiale, le client est informé de la topologie. Si la connexion en cours est périmée, le client établit une connexion vers un autre nœud.

Après qu'un serveur live a échoué et qu'un serveur de sauvegarde a pris la relève, vous devrez redémarrer le serveur live et procéder à une restauration automatique des clients. Pour ce faire, redémarrez le serveur d'origine et arrêter (kill) le nouveau serveur. Vous pouvez le faire en supprimant le processus proprement dit ou attendre que le serveur se plante par lui-même. Vous pouvez également provoquer un basculement lors de l'arrêt normal du serveur. Pour cela, définir la propriété `failover-on-shutdown` à true dans le fichier de configuration `standalone.xml` :

```
<failover-on-shutdown>true</failover-on-shutdown>
```

Par défaut, la propriété `failover-on-shutdown` est définie à false.

Vous pouvez également forcer l'arrêt du nouveau serveur live quand l'ancien serveur live revient, ce qui permet au serveur live d'origine de prendre la relève automatiquement en définissant la propriété `allow-failback` à `true` dans le fichier de configuration `standalone.xml` :

```
<allow-failback>true</allow-failback>
```

En mode de réplication HA, pour forcer l'arrêt du nouveau serveur live quand l'ancien serveur live revient, définir la propriété `check-for-live-server` à `true` dans le fichier de configuration `standalone.xml` :

```
<check-for-live-server>true</check-for-live-server>
```

[Rapporter un bogue](#)

18.14.9. Déploiements sur les serveurs de sauvegarde HornetQ

Dans un environnement dédié de HA, un serveur JBoss EAP 6 avec HornetQ configuré comme une sauvegarde ne doit pas servir à héberger les déploiements qui utilisent ou se connectent à la sauvegarde HornetQ sur ce serveur. Cela inclut les déploiements comme les Enterprise Java Beans (Stateless Session Beans, Message Driven Beans) ou servlets.

Si un serveur JBoss EAP 6 a une configuration de sauvegarde de cohabitation HornetQ (avec le serveur HornetQ configuré en « live » dans le sous-système de messagerie et un autre serveur HornetQ configuré en sauvegarde), alors le serveur JBoss EAP 6 peut héberger des déploiements tant qu'ils sont configurés pour se connecter au serveur HornetQ « live ».

[Rapporter un bogue](#)

18.14.10. Modes de basculements HornetQ

HornetQ définit deux types de basculement client :

- Le basculement client automatique
- Le basculement client niveau application

HornetQ fournit un rattachement automatique transparent des connexions sur le même serveur, par exemple, en cas de problèmes de réseau transitoire. Ceci est similaire à un basculement, sauf dans les cas de reconnexion au même serveur.

Pendant un basculement, si le client a des consommateurs sur une file d'attente temporaire ou persistante, ces files d'attentes sont automatiquement recrées lors du basculement sur un nœud de sauvegarde, puisque les nœuds de sauvegarde n'ont aucune information sur les files d'attente non persistantes.

[Rapporter un bogue](#)

18.14.11. Le basculement client automatique

Les clients HornetQ peuvent être configurés pour recevoir des informations sur les serveurs live et de sauvegarde. Cette information est utile en cas d'échec de connexion client. En cas de serveur live, le client détecte un basculement et reconnecte au serveur de sauvegarde. Le serveur de sauvegarde

recrée automatiquement toutes les sessions ou consommateurs qui existent pour chaque connexion avant le basculement, ce qui évite à l'utilisateur d'avoir à coder manuellement la logique de reconnexion.

Les clients de HornetQ détectent un échec de connexion si les paquets ne sont pas reçus par le serveur dans le délai indiqué par `client-failure-check-period`. Si le client ne reçoit pas de données à temps, le client assume que la connexion a échoué et il initie un basculement. Si la socket est fermée par le système d'exploitation, le processus serveur est tué plutôt que la machine se bloque, et le client immédiatement initie un basculement.

Les clients Hornet Q peuvent être configurés de différentes façons pour découvrir la liste des groupes de serveurs de sauvegarde live. Le client peut être configuré de manière explicite ou utiliser `server discovery` pour découvrir automatiquement la liste. Par ailleurs, les clients peuvent explicitement se connecter à un serveur spécifique et télécharger les serveurs et les sauvegardes en cours.

Pour permettre le basculement automatique des clients, le client doit être configuré pour autoriser les tentatives de reconnexion non-nulle.

Par défaut, le basculement ne se produit que lorsqu'au moins une connexion a été faite au serveur live. Le client fait des tentatives de connexion au serveur live comme spécifié dans la propriété de `reconnect-attempts` et échoue après le nombre spécifié de tentatives.

[Rapporter un bogue](#)

18.14.12. Le basculement niveau application

Sans certains cas et en fonction de vos besoins, vous pouvez gérer une erreur de connexion manuellement en spécifiant la logique de reconnexion dans un gestionnaire de basculement personnalisé. Vous pouvez définir cela pour le basculement au niveau application puisque le basculement est géré au niveau de l'application utilisateur.

Pour implémenter le basculement au niveau des applications, si vous utilisez JMS, vous devrez définir une classe `ExceptionListener` sur la connexion JMS. Si un échec de connexion est détecté, la classe `ExceptionListener` sera appelée par HornetQ. Dans votre `ExceptionListener`, fermer les anciennes connexions JMS, chercher les nouvelles instances d'usine de connexion de JNDI et créer de nouvelles connexions

Si vous utilisez l'API core, alors la procédure est similaire : définir un `FailureListener` sur les instances de la session client (`ClientSession`) principale.

[Rapporter un bogue](#)

CHAPITRE 19. SOUS-SYSTÈME DE TRANSACTION

19.1. CONFIGURATION DE SOUS-SYSTÈME DE TRANSACTIONS

19.1.1. Configuration des transactions

Introduction

Les procédures suivantes vous montrent comment configurer le sous-système de transactions de JBoss EAP 6.

- [Section 19.1.3, « Configurez votre source de données pour utiliser l'API de transaction JTA »](#)
- [Section 19.1.4, « Configuration d'une source de données XA »](#)
- [Section 19.1.2, « Configurer le Transaction Manager \(TM\) \(ou gestionnaire de transactions\) »](#)
- [Section 19.1.6, « Configurer la journalisation des sous-systèmes de transactions »](#)

[Rapporter un bogue](#)

19.1.2. Configurer le Transaction Manager (TM) (ou gestionnaire de transactions)

Vous pouvez configurer le Transaction Manager (TM) à l'aide de la console de gestion sur le web ou en ligne de commande. Pour chaque commande ou option donnée, on assume que vous exécutez JBoss EAP 6 comme un domaine géré. Si vous utilisez un serveur autonome ou que vous souhaitez modifier un profil différent de la valeur par défaut `default`, il se peut que vous ayez à modifier les étapes et les commandes de la manière suivante.

Notes sur les commandes d'exemple

- Pour la console de gestion, le profil par défaut `default` est celui qui sera sélectionné quand vous vous connectez. Si vous souhaitez modifier la configuration du gestionnaire de transactions (TM) dans un autre profile, sélectionnez votre profile à la place, et non pas `default`, pour chaque instruction.

De même, substituez votre profil à la place du profil par défaut `default` pour les commandes CLI de l'exemple.

- Si vous utilisez un serveur autonome, un seul profil existe. Ignorer toute instruction pour choisir un profil spécifique. Dans les commandes CLI, retirer la partie `/profile=default` des commandes d'échantillon.



NOTE

Pour que les options du TM soient visibles dans la console de gestion ou dans l'interface CLI, le sous-système `transactions` doit être activé. Il est activé par défaut, et il faut pour cela qu'un certain nombre d'autres sous-systèmes fonctionnent correctement, donc il est improbable qu'il soit désactivé.

Configurer le TM par la console de gestion

Pour configurer le gestionnaire de transactions (TM) à l'aide de la console de gestion sur le web, sélectionnez l'onglet **Configuration** en haut de l'écran. Si vous utilisez un domaine géré, vous avez

le choix de plusieurs profils. Choisir le bon **Profile** de la boîte de sélection dans la partie supérieure gauche de l'écran. Étendez le menu **Container**, et sélectionnez **Transactions**.

Vous verrez la plupart des options dans la page de configuration du gestionnaire de transactions. Les options **Recovery** sont cachées par défaut. Cliquer sur l'onglet **Recovery** pour voir les options de recouvrement. Cliquer sur **Edit** pour éditer une des options. Les changements prendront place immédiatement.

Cliquer sur l'étiquette **Need Help?** pour afficher le texte d'aide en ligne.

Configurer le TM par l'interface CLI

Dans l'interface CLI, vous pouvez configurer le TM en utilisant une série de commandes. Les commandes commencent toutes par `/profile=default/subsystem=transactions/` pour un domaine géré avec `default` de profil, ou par `/subsystem=transactions` pour un serveur autonome.



IMPORTANT

HornetQ n'autorise pas plusieurs instances à partager un message log store. Si vous configurez plusieurs instances d'HornetQ, chaque instance devra avoir son propre message log store.

Tableau 19.1. Options de configuration du TM

Option	Description	CLI Command
Activer les statistiques	Indique s'il faut activer les statistiques de transaction. Ces statistiques se trouvent dans la console de gestion dans la section Subsystem Metrics de l'onglet Runtime .	<code>/profile=default/subsystem=transactions/:write-attribute(name=enable-statistics,value=true)</code>
Activer le statut TSM	Indique si l'on doit activer le service de gestion du statut de transaction (TSM), qui est utilisé pour le recouvrement hors-processus. L'exécution d'un manager de recouvrement de processus pour contacter l'ActionStatusService à partir de processus différents n'est pas pris en charge (normalement contacté en mémoire).	Cette option de configuration n'est pas prise en charge.
Délai d'attente par défaut	Délai d'attente de transaction par défaut. La valeur par défaut est de 300 secondes. Vous pouvez la remplacer par programmation, sur la base d'une transaction.	<code>/profile=default/subsystem=transactions/:write-attribute(name=default-timeout,value=300)</code>

Option	Description	CLI Command
Chemin de Store Objet	Un chemin de système de fichiers relatif ou absolu où le store objet TM stocke des données. Relatif, par défaut, à la valeur du paramètre object-store-relative-to .	<code>/profile=default/subsystem=transactions/:write-attribute(name=object-store-path,value=tx-object-store)</code>
Chemin de Store Objet Relatif à	Référence une configuration de chemin global dans le modèle du domaine. La valeur par défaut correspond au répertoire de données de JBoss EAP 6, qui correspond à la valeur de la propriété jboss.server.data.dir , et qui a pour valeur par défaut EAP_HOME/domain/data/ pour un domaine géré, ou EAP_HOME/standalone/data/ pour une instance de serveur autonome. La valeur de l'attribut object-store-path de l'object store est relative à ce chemin.	<code>/profile=default/subsystem=transactions/:write-attribute(name=object-store-relative-to,value=jboss.server.data.dir)</code>
Liaisons de sockets	Indique le nom de la liaison du socket utilisé par le gestionnaire de transactions pour la récupération et la création des identificateurs de transactions, lorsque le mécanisme du socket est utilisé. Se référer à processus-id-socket-max-ports pour plus d'informations sur la génération de l'identificateur unique. Les liaisons de socket sont spécifiées par le groupe de serveurs dans l'onglet Serveur de la console de gestion.	<code>/profile=default/subsystem=transactions/:write-attribute(name=socket-binding,value=txn-recovery-environment)</code>
Statut de liaison de socket	Indique la liaison de socket à utiliser pour le gestionnaire de statuts de transactions.	Cette option de configuration n'est pas prise en charge.
Listener de recouvrement	Indique si oui ou non le processus de recouvrement de transactions doit écouter au socket de réseau. La valeur par défaut est false .	<code>/profile=default/subsystem=transactions/:write-attribute(name=recovery-listener,value=false)</code>

Les options suivantes sont pour une utilisation avancée et ne peuvent être modifiées que par ligne de commande. Soyez prudent lors de leur modification à partir de la configuration par défaut. Communiquer avec Red Hat Global Support Services pour plus d'informations.

Tableau 19.2. Options de configuration du TM avancées

Option	Description	CLI Command
jts	Indique si l'on doit utiliser les transactions Java Transaction Service (JTS). La valeur par défaut est false , qui utilise des transactions JTA uniquement.	<code>/profile=default/subsystem=transactions/:write-attribute(name=jts,value=false)</code>
Identificateur de nœud	<p>L'identificateur de nœud de gestionnaire de transactions. Cette option est requise dans les situations suivantes :</p> <ul style="list-style-type: none"> • Pour les communications de JTS à JTS • Quand deux gestionnaires de transactions accèdent à des gestionnaires de ressources partagées • Quand deux gestionnaires de transactions accèdent à des object stores partagés <p>Le node-identifiant doit être unique pour chaque gestionnaire de transaction car il doit assurer l'intégrité des données pendant le recouvrement. Le node-identifiant doit également être unique à JTA car plusieurs nœuds peuvent entrer en interaction avec le même gestionnaire de ressources ou partager un object store de transactions.</p>	<code>/profile=default/subsystem=transactions/:write-attribute(name=node-identifiant,value=1)</code>

Option	Description	CLI Command
process-id-socket-max-ports	<p>Le gestionnaire de transactions crée un identifiant unique pour chaque journal des transactions. Deux mécanismes différents sont fournis pour générer des identifiants uniques : un mécanisme basé sur le socket et un mécanisme fondé sur l'identificateur de processus du processus.</p> <p>Dans le cas de l'identifiant basé-socket, le socket est ouvert et son numéro de port est utilisé pour l'identifiant. Si le port est déjà utilisé, on cherchera le port suivant jusqu'à ce qu'un port libre soit trouvé. Les processus-id-socket-max-ports représentent le nombre maximal de sockets que le TM va essayer avant d'abandonner. La valeur par défaut est 10.</p>	<pre>/profile=default/subsystem=transactions/:write - attribute(name=process-id-socket-max-ports,value=10)</pre>
process-id-uuid	<p>Définir à true avec un identifiant de processus pour créer un identifiant unique pour chaque transaction. Sinon, le mécanisme basé socket sera utilisé. La valeur par défaut est true. Se référer à process-id-socket-max-ports pour obtenir davantage d'informations. Pour activer process-id-socket-binding, définir process-id-uuid à false.</p>	<pre>/profile=default/subsystem=transactions/:write - attribute(name=process-id-uuid,value=true)</pre>
process-id-socket-binding	<p>Le nom de la configuration de liaison de socket à utiliser si le gestionnaire de transactions doit utiliser un id de processus basé-socket. Correspondra à undefined si process-id-uuid est sur true; sinon il devra être défini.</p>	<pre>/profile=default/subsystem=transactions/:write - attribute(name=process-id-socket-binding,value=true)</pre>

Option	Description	CLI Command
use-hornetq-store	Utiliser les mécanismes de stockage journalisés de HornetQ au lieu du stockage basé sur des fichiers, pour les journaux de transactions. Ceci est désactivé par défaut, mais peut améliorer les performances I/O. Il n'est pas recommandé pour les transactions JTS sur les gestionnaires de transactions séparés. Quand vous changez cette option, le serveur doit démarrer à nouveau à l'aide de la commande shutdown pour que le changement puisse prendre effet.	<code>/profile=default/subsystem=transactions/:write-attribute(name=use-hornetq-store,value=false)</code>

[Rapporter un bogue](#)

19.1.3. Configurez votre source de données pour utiliser l'API de transaction JTA

Résumé

Cette tâche vous montre comment activer un JTA (Java Transaction API) dans votre source de données.

Conditions préalables

Vous devez remplir les conditions suivantes avant de continuer cette tâche :

- Votre base de données ou autre ressource devra supporter un API JTA. Dans le doute, veuillez consulter la documentation.
- Créer une source de données. Veuillez vous référer à [Section 6.3.1, « Créer une source de données non-XA avec les interfaces de gestion »](#).
- Stopper le serveur JBoss EAP 6
- Obtenir un accès pour pouvoir éditer les fichiers de configuration directement dans un éditeur de texte.

Procédure 19.1. Configurez la source de données pour utiliser l'API de Java Transaction

1. Ouvrir le fichier de configuration dans l'éditeur de texte.

Selon que vous exécutiez JBoss EAP 6 sur un domaine géré ou un serveur autonome, votre fichier de configuration ne se trouvera pas au même endroit.

o Domaine géré

Le fichier de configuration par défaut d'un domaine géré se trouve dans `EAP_HOME/domain/configuration/domain.xml` pour Red Hat Enterprise Linux, et `EAP_HOME\domain\configuration\domain.xml` pour Microsoft Windows Server.

o Serveur autonome

Le fichier de configuration par défaut d'un serveur autonome se trouve dans `EAP_HOME/standalone/configuration/domain.xml` pour Red Hat Enterprise Linux,

et `EAP_HOME\standalone\configuration\domain.xml` pour Microsoft Windows Server.

2. Chercher la balise `<datasource>` qui correspond à votre source de données.

La source de données aura un attribut `jndi-name` correspondant à celui que vous aviez indiqué quand vous l'avez créée. Par exemple, la source de données ExampleDS ressemble à ceci :

```
<datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="H2DS" enabled="true" jta="true" use-java-context="true" use-cm="true">
```

3. Définir l'attribut `jta` à `true`.

Ajouter l'élément suivant au contenu de votre balise `<datasource>`, tel qu'il apparaît à l'étape précédente : `jta="true"`

Sauf si vous avez un cas d'utilisation spécifique (comme par exemple, définir une source de données en lecture seule), Red Hat décourage la substitution de la valeur par défaut `jta=true`. Ce paramètre indique que la source de données honorera l'API Java Transaction et permettra un meilleur suivi des connexions par l'implémentation JCA.

4. Sauvegarder le fichier de configuration.

Sauvegarder le fichier de configuration et sortir de l'éditeur de texte.

5. Démarrer JBoss EAP 6.

Relancer le serveur JBoss EAP 6.

Résultat :

JBoss EAP 6 démarre, et votre source de données est configurée pour utiliser l'API de transactions JTA.

[Rapporter un bogue](#)

19.1.4. Configuration d'une source de données XA

Conditions préalables

Pour pouvoir ajouter une source de données XA, vous devrez vous connecter à la console de gestion. Voir [Section 3.4.2, « Se connecter à la console de gestion »](#) pour plus d'informations.

1. Ajouter une nouvelle source de données.

Ajouter une nouvelle source de données à la plateforme JBoss EAP 6. Suivre les instructions qui se trouvent dans [Section 6.3.1, « Créer une source de données non-XA avec les interfaces de gestion »](#), puis, cliquer sur l'onglet **XA Datasource** en haut.

2. Configurer les propriétés supplémentaires suivant les besoins.

Tous les paramètres de la source de données se trouvent dans [Section 6.7.1, « Paramètres de source de données »](#).

Résultat

Votre source de données XA est configurée et elle est prête à l'utilisation.

[Rapporter un bogue](#)

19.1.5. Messages de journalisation de transactions

Pour suivre le statut de la transaction tout en gardant les fichiers de journalisation lisibles, utiliser le niveau de journalisation **DEBUG** pour le logger de transaction. Pour un débogage détaillé, utiliser le niveau de journalisation **TRACE**. Veuillez consulter [Section 19.1.6, « Configurer la journalisation des sous-systèmes de transactions »](#) pour plus d'informations sur la configuration du logger de transaction.

Le gestionnaire de transaction peut générer beaucoup d'informations de journalisation si configuré pour se connecter au niveau de journalisation **TRACE**. Vous trouverez ci-dessous quelques-uns des messages les plus courants. Cette liste n'est pas exhaustive, il se peut que vous rencontriez d'autres messages.

Tableau 19.3. Changement d'état de transaction

Début de transaction	<p>Quand une transaction commence, le code suivant s'exécute :</p> <pre>com.arjuna.ats.arjuna.coordinator .BasicAction::Begin:1342 tsLogger.logger.trace("BasicActio n::Begin() for action-id "+ get_uid());</pre>
Validation de transaction	<p>Quand une transaction est validée, le code suivant s'exécute :</p> <pre>com.arjuna.ats.arjuna.coordinator .BasicAction::End:1342 tsLogger.logger.trace("BasicActio n::End() for action-id "+ get_uid());</pre>
Restauration de transaction	<p>Quand une transaction est restaurée, le code suivant s'exécute :</p> <pre>com.arjuna.ats.arjuna.coordinator .BasicAction::Abort:1575 tsLogger.logger.trace("BasicActio n::Abort() for action-id "+ get_uid());</pre>

<p>Délai d'expiration de transaction</p>	<p>Quand une transaction expire, le code suivant s'exécute :</p> <pre> com.arjuna.ats.arjuna.coordinator .TransactionReaper::doCancellatio ns:349 tsLogger.logger.trace("Reaper Worker " + Thread.currentThread() + " attempting to cancel " + e._control.get_uid()); </pre> <p>Vous verrez ensuite le même thread restaurer la transaction comme montré ci-dessus.</p>
--	--

[Rapporter un bogue](#)

19.1.6. Configurer la journalisation des sous-systèmes de transactions

Résumé

Utiliser cette procédure pour contrôler la quantité d'informations enregistrées sur les transactions, indépendamment des autres paramètres de journalisation dans JBoss EAP 6. La procédure montre comment procéder dans la console de gestion sur le web. La commande de gestion CLI est donnée par la suite.

Procédure 19.2. Configurer l'enregistreur de transactions (Transaction Logger) par la console de gestion

1. Naviguer vers la zone de configuration de la journalisation

Dans la console de gestion, cliquer sur l'onglet **Configuration**. Si vous utilisez un domaine géré, fermer le profil du serveur que vous souhaitez configurer, à partir de la case de sélection **Profile** qui se trouve en haut et à gauche.

Étendre le menu **Core**, et sélectionner **Logging**.

2. Modifier les attributs de `com.arjuna`.

Sélectionner l'onglet **Log Categories**. Sélectionner `com.arjuna`, puis **Edit** dans **Details**. Vous pourrez ajouter ici les informations de journalisation spécifiques à la classe. La classe `com.arjuna` est déjà présente. Vous pourrez modifier le niveau de journalisation et décider si vous souhaitez utiliser les gestionnaires parents.

Niveau de journalisation

Le niveau de journalisation est **WARN** par défaut. Comme les transactions peuvent produire une grande quantité de messages de journalisation, la signification des niveaux de journalisation standard est légèrement différente pour l'enregistreur de transactions. En général, les messages avec des niveaux de gravité moins élevés que le niveau choisi seront ignorés.

Niveaux de journalisation des transactions, du plus au moins détaillé.

- TRACE
- DEBOG
- INFO
- AVERTISSEMENT
- ERREUR
- FAILURE

Utiliser les gestionnaires parents

Indique si l'enregistreur d'événements doit envoyer ses sorties vers l'enregistreur d'événements parent. Le comportement par défaut est `true`.

3. Les changements prennent effet immédiatement.

[Rapporter un bogue](#)

19.2. ADMINISTRATION DES TRANSACTIONS

19.2.1. Naviguer et gérer les transactions

Le CLI prend en charge la capacité de naviguer et de manipuler les enregistrements des transactions. Cette fonctionnalité est fournie par l'interaction entre le gestionnaire de transactions et l'API de gestion de JBoss EAP 6.

Le gestionnaire de transactions stocke des informations sur chaque transaction en attente et les participants impliqués dans la transaction, dans un stockage persistant appelé *object store*. L'API de gestion expose le store objet sous forme de ressource appelée *log-store*. Une opération API nommée *probe* lit les journaux de transactions et crée un noeud pour chaque journal. Vous pouvez invoquer la commande *probe* manuellement, quand vous souhaitez réactualiser le *log-store*. Il est normal pour les journaux de transaction d'apparaître ou de disparaître rapidement.

Exemple 19.1. Réactualiser le log store

Cette commande réactualise le log store des groupes de serveurs qui utilisent le profil par défaut `default` dans un domaine géré. Dans le cas d'un serveur autonome, supprimer `profile=default` de la commande.

```
/profile=default/subsystem=transactions/log-store=log-store/:probe
```

Exemple 19.2. Voir toutes les transactions préparées

Pour voir toutes les transactions préparées, commencer par réactualiser le log store (voir [Exemple 19.1, « Réactualiser le log store »](#)), puis exécuter la commande suivante, qui fonctionne de la même manière qu'une commande `ls` de système de fichiers.

```
ls /profile=default/subsystem=transactions/log-store=log-store/transactions
```

Chaque transaction est visible, ainsi que son identifiant unique. Les opérations individuelles peuvent être exécutées pour une transaction individuelle (voir [Gérer une transaction](#)).

Gérer une transaction

Voir des attributs de transaction.

Pour voir des informations sur une transaction, comme son nom JNDI, son nom de produit EIS ou sa version, ou encore son statut, utiliser la commande CLI : `read-resource`.

```
/profile=default/subsystem=transactions/log-store=log-store/transactions=0\:ffff7f000001\:-b66efc2\:4f9e6f8f\:9:read-resource
```

Voir tous les participants à une transaction.

Chaque journal de transaction contient un élément enfant nommé **participants**. Utiliser la commande CLI `read-resource` sur cet élément pour voir les participants des transactions. Les participants sont identifiés par leur nom JNDI.

```
/profile=default/subsystem=transactions/log-store=log-store/transactions=0\:ffff7f000001\:-b66efc2\:4f9e6f8f\:9/participants=java\:\JmsXA:read-resource
```

Le résultat doit ressembler à ceci :

```
{
  "outcome" => "success",
  "result" => {
    "eis-product-name" => "HornetQ",
    "eis-product-version" => "2.0",
    "jndi-name" => "java:/JmsXA",
    "status" => "HEURISTIC",
    "type" => "/StateManager/AbstractRecord/XAResourceRecord"
  }
}
```

Le statut du résultat affiché ici est dans un état **HEURISTIC** et est susceptible d'être recouvert. Voir [Recouvrement d'une transaction](#). pour plus d'informations.

Dans certains cas, il est possible de créer des enregistrements orphelins dans l'object store, c-a-d XAResourceRecords, qui n'a pas d'enregistrement de transaction correspondante dans le journal. Par exemple, la ressource XA s'était préparée mais avait échoué avant le TM enregistré et est inaccessible à l'API de gestion du domaine. Pour accéder à de tels enregistrements, vous devez définir l'option de gestion `expose-all-logs` à `true`. Cette option n'est pas sauvegardée dans le modèle de gestion et est restaurée à `false` quand le serveur redémarre à nouveau.

```
/profile=default/subsystem=transactions/log-store=log-store:write-attribute(name=expose-all-logs, value=true)
```

Supprimer une transaction.

Chaque journal de transaction supporte une opération : `delete` pour effacer l'enregistrement qui représente la transaction.

```
/profile=default/subsystem=transactions/log-store=log-
store/transactions=0\:\ffff7f000001\:-b66efc2\:\4f9e6f8f\:\9:delete
```

Recouvrement d'une transaction.

Chaque participant de transaction supporte le recouvrement par la commande CLI : **recover**.

```
/profile=default/subsystem=transactions/log-store=log-
store/transactions=0\:\ffff7f000001\:-
b66efc2\:\4f9e6f8f\:\9/participants=2:recover
```

Recouvrement des transactions heuristiques et des participants

- Si le statut de la transaction est **HEURISTIC**, l'opération de recouvrement change l'état en **PREPARE** et déclenche un recouvrement.
- Si l'un des participants de la transaction est heuristique, l'opération de recouvrement tente de reprendre l'opération **commit** (validation) à nouveau. En cas de succès, le participant est retiré du journal des transactions. Vous pouvez vérifier cela en exécutant à nouveau l'opération : **probe** sur le **log-store** et en vérifiant que le participant n'est plus inscrit. Si c'est le dernier participant, la transaction sera également supprimée.

Réactualiser le statut de la transaction qui a besoin d'être recouverte.

Si une transaction a besoin d'être recouverte, vous pourrez utiliser la commande CLI : **refresh** pour vous assurer qu'elle a toujours besoin d'être recouverte, avant de tenter le recouvrement.

```
/profile=default/subsystem=transactions/log-store=log-
store/transactions=0\:\ffff7f000001\:-
b66efc2\:\4f9e6f8f\:\9/participants=2:refresh
```

Voir les statistiques de transaction

Si les statistiques de Transaction manager sont activées, vous pouvez consulter les statistiques à propos du gestionnaire de transactions et du sous-système de transaction. Veuillez consulter [Section 19.1.2, « Configurer le Transaction Manager \(TM\) \(ou gestionnaire de transactions\) »](#) pour plus d'informations sur l'activation des statistiques de Transaction manager.

Vous pouvez consulter les statistiques soit par la console de gestion basée-web, soit par l'interface CLI. de gestion Dans la console de gestion basée-web, les statistiques de transaction seront disponibles via **Runtime** → **Status** → **Subsystems** → **Transactions**. Les statistiques de transaction sont disponibles pour chaque serveur dans un domaine géré, également. Pour voir le statut d'un autre serveur, sélectionner **Change Server** situé en haut à gauche du menu, et sélectionner un serveur de la liste.

Le tableau suivant affiche chaque statistique disponible, sa description et la commande CLI de gestion pour afficher le statistique.

Tableau 19.4. Les statistiques de sous-système de transaction

Statistique	Description	CLI Command
-------------	-------------	-------------

Statistique	Description	CLI Command
Total	Le nombre total de transactions exécutées par le gestionnaire de transactions sur ce serveur.	<pre>/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-transactions,include-defaults=true)</pre>
Validé	Le nombre de transactions validées exécutées par le gestionnaire de transactions sur ce serveur.	<pre>/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-committed-transactions,include-defaults=true)</pre>
Abandonné	Le nombre de transactions interrompues exécutées par le gestionnaire de transactions sur ce serveur.	<pre>/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-aborted-transactions,include-defaults=true)</pre>
Délai expiré	Le nombre de transactions expirées exécutées par le gestionnaire de transactions sur ce serveur.	<pre>/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-timed-out-transactions,include-defaults=true)</pre>

Statistique	Description	CLI Command
Heuristiques	Pas disponible dans la console de gestion. Nombre de transactions dans un état heuristique.	<pre>/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-heuristics,include-defaults=true)</pre>
Transactions In-Flight	Pas disponible dans la console de gestion. Nombre de transactions commencées non achevées.	<pre>/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-inflight-transactions,include-defaults=true)</pre>
Origine de l'échec - Applications	Le nombre de transactions échouées dont l'origine de l'échec était une application.	<pre>/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-application-rollback,include-defaults=true)</pre>
Origine de l'échec - Ressources	Le nombre de transactions échouées dont l'origine de l'échec était une ressource.	<pre>/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-resource-rollback,include-defaults=true)</pre>

Statistique	Description	CLI Command
ID Participant	L'ID du participant.	<pre>/host=master/server=server-one/subsystem=transactions/log-store=log-store/transactions=0\n:ffff7f000001\n:-b66efc2\n:4f9e6f8f\n:9:read-children-names(child-type=participants)</pre>
Liste de toutes les transactions	La liste complète des transactions.	<pre>/host=master/server=server-one/subsystem=transactions/log-store=log-store:read-children-names(child-type=transactions)</pre>

[Rapporter un bogue](#)

19.3. RÉFÉRENCES DE TRANSACTIONS

19.3.1. Erreurs et exceptions pour les transactions JBoss

Pour obtenir des informations lancées par les méthodes de la classe `UserTransaction`, voir la spécification *UserTransaction API* dans <http://docs.oracle.com/javaee/6/api/javax/transaction/UserTransaction.html>.

[Rapporter un bogue](#)

19.3.2. Limitations sur les transactions JTA

Les transactions JTA ne peuvent pas être au courant de la distribution à travers les instances multiples de JBoss EAP 6. Pour ce comportement, utiliser les transactions JTS.

Pour pouvoir utiliser les transactions JTS, vous devrez configurer l'ORB, qui inclut l'activation des transactions dans le sous-système JacORB, puis configurez le sous-système JTS.

- [Section 19.4.2, « Configurer l'ORB pour les transactions JTS »](#)

[Rapporter un bogue](#)

19.4. CONFIGURATION ORB

19.4.1. CORBA (Common Object Request Broker Architecture)

Common Object Request Broker Architecture (CORBA) est une norme qui autorise les applications et services à travailler ensemble même lorsqu'ils sont écrits dans de multiples langages normalement incompatibles ou lorsqu'ils sont hébergés sur des plateformes différentes. Les requêtes CORBA sont émises par un composant côté serveur appelé *Object Request Broker (ORB)*. JBoss EAP 6 fournit une instance ORB au moyen du composant JacORB.

L'ORB est utilisé en interne pour les transactions *Java Transaction Service (JTS)*, et peut également être utilisé par votre propre application.

[Rapporter un bogue](#)

19.4.2. Configurer l'ORB pour les transactions JTS

Dans une installation JBoss EAP 6 par défaut, l'ORB est désactivé. Vous pouvez activer l'ORB en utilisant l'interface CLI de ligne de commande.



NOTE

Dans un domaine géré, le sous-système JacORB est disponible dans les profils `full` et `full-ha` uniquement. Dans un serveur autonome, il est disponible uniquement quand vous utilisez les configurations `standalone-full.xml` ou `standalone-full-ha.xml`.

Procédure 19.3. Configurer l'ORB par la console de gestion

1. Voir les paramètres de configuration du profil.

Sélectionner **Configuration** dans la partie supérieure de la console de gestion. Si vous utilisez un domaine géré, sélectionner soit le profil `full` ou `full-ha` à partir de la boîte de dialogue de sélection en haut à gauche.

2. Modifier les paramètres Initializers

Étendre sur le menu **Subsystems**. Étendre le menu **Container**, et sélectionner **JacORB**.

Sur le formulaire qui apparaît sur l'écran principal, sélectionner l'onglet **Initializers**, et cliquer sur le bouton **Edit**.

Activer les intercepteurs de sécurité en configurant la valeur de **Security** à **active**.

Pour activer ORB sur JTS, définir la valeur des **Transaction Interceptors** à **active**, au lieu de la valeur par défaut **spec**.

Voir le lien **Need Help?** sur le formulaire pour accéder à des explications sur ces valeurs. Cliquer sur **Save** quand vous aurez fini de modifier les valeurs.

3. Configuration ORB avancée

Voir les autres sections du formulaire pour les options de configuration avancées. Chaque section inclut un lien **Need Help?** avec des informations détaillées sur les paramètres.

Configurer l'ORB par l'interface CLI

Vous pouvez configurer chaque aspect de l'ORB à l'aide de l'interface CLI. Les commandes suivantes configurent les initialiseurs aux mêmes valeurs que celles de la procédure ci-dessus, pour la console de gestion. Il s'agit de la configuration minimale pour l'ORB, si utilisé avec JTS.

Ces commandes sont configurées pour un domaine de sécurité utilisant le profil `full`. Si nécessaire, modifier le profil pour qu'il convienne mieux à celui que vous aurez besoin de configurer. Si vous utilisez un serveur autonome, n'utilisez pas la portion `/profile=full` des commandes.

Exemple 19.3. Activer les intercepteurs de sécurité

```
/profile=full/subsystem=jacorb/:write-attribute(name=security,value=on)
```

Exemple 19.4. Activer les transactions dans le sous-système JacORB

```
/profile=full/subsystem=jacorb/:write-attribute(name=transactions,value=on)
```

Exemple 19.5. Activer JTS dans le sous-système de transactions

```
/profile=full/subsystem=transactions:write-attribute(name=jts,value=true)
```



NOTE

Pour l'activation JTS, le serveur doit être démarré à nouveau car le rechargement ne suffira pas.

[Rapporter un bogue](#)

19.5. JDBC OBJECT STORE SUPPORT

19.5.1. JDBC Store de transactions

Conditions préalables :

- [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#)

Les transactions peuvent utiliser une source de données JDBC comme magasin d'objets. Si la base de données à utiliser est configuré pour le basculement et la récupération, c'est peut-être une meilleure option que l'utilisation d'espace disque sur un serveur d'applications. Les avantages doivent être pesés contre le fait qu'un magasin d'objets JDBC brut est un magasin d'objets spéciaux et ne peut pas exécuter de la même façon qu'un système de fichiers ou qu'un HornetQ journal object store.

**NOTE**

Une source de données JDBC utilisée comme store d'objets de transactions *doit* spécifier `jta="false"` dans la section `datasource` du fichier de configuration du serveur.

Procédure 19.4. Active l'utilisation d'une source de données JDBC comme Transaction Object Store

1. Définir `use-jdbc-store` à `true`.

```
/subsystem=transactions:write-attribute(name=use-jdbc-store,
value=true)
```

2. Définir `jdbc-store-datasource` au nom JNDI pour la source de données à utiliser.

```
/subsystem=transactions:write-attribute(name=jdbc-store-datasource,
value=java:jboss/datasources/TransDS)
```

3. Démarrer à nouveau le serveur JBoss EAP 6 pour que les changements puissent prendre effet.

```
shutdown --restart=true
```

L'ensemble des attributs sont fournis ci-dessous.

Tableau 19.5. Propriétés des StoresJDBC de transactions

Property	Description
<code>use-jdbc-store</code>	Le définir à <code>true</code> pour activer le store JDBC de transactions.
<code>jdbc-store-datasource</code>	Le nom JNDI de la source de données JDBC utilisée pour le stockage.
<code>jdbc-action-store-drop-table</code>	Supprimer et recréer les tables de stores d'actions lors du lancement. En option, par défaut « <code>false</code> ».
<code>jdbc-action-store-table-prefix</code>	Le préfixe des noms de tables de stores d'actions. En option.
<code>jdbc-communication-store-drop-table</code>	Supprimer et recréer les tables de stores de communications lors du lancement. En option, par défaut « <code>false</code> ».
<code>jdbc-communication-store-table-prefix</code>	Le préfixe des noms de tables de stores de communications. En option.

Property	Description
<code>jdbc-state-store-drop-table</code>	Supprimez et recréez les tables de stores d'états lors du lancement. En option, par défaut « false ».
<code>jdbc-state-store-table-prefix</code>	Le préfixe des noms de tables de stores d'états. En option.

Voir également :

- [Section 19.1.3, « Configurez votre source de données pour utiliser l'API de transaction JTA »](#)

[Rapporter un bogue](#)

CHAPITRE 20. SOUS-SYSTÈME DE MESSAGERIE

20.1. UTILISER DES TRANSPORTS PERSONNALISÉS DANS LES SOUS-SYSTÈMES DE MESSAGERIE

Quand on utilise un serveur de messagerie standard (POP3, IMAP), le serveur possède un ensemble d'attributs qui peuvent être définis, dont certains obligatoires.

Le plus important étant `outbound-socket-binding-ref` qui fait référence à une liaison de socket de mail sortante et qui est défini par l'adresse d'hôte et le numéro de port.

Ce n'est pas la meilleure solution pour certains utilisateurs car leur configuration utilise des hôtes multiples à but d'équilibrage des charges. Cette configuration, cependant, n'est pas prise en charge par le JavaMail standard, ce qui signifie que certains utilisateurs devront mettre en place des moyens de transport personnalisés pour leur mail.

Ces transports personnalisés ne requièrent pas de `outbound-socket-binding-ref` et autorisent les formats de propriétés d'hôte personnalisés.

Un transport personnalisé peut être configuré par le CLI à l'aide des commandes suivantes :

Procédure 20.1.

1. Ajouter une nouvelle session mail. La commande ci-dessous crée une nouvelle session nommée `mySession` et définit JNDI à `java:jboss/mail/MySession` :

```
/subsystem=mail/mail-session=mySession:add(jndi-name=java:jboss/mail/MySession)
```

2. Ajouter une liaison de socket sortante. La commande ci-dessous ajoute une liaison de socket nommée `my-smtp-binding` qui pointe vers `localhost:25`.

```
/socket-binding-group=standard-sockets/remote-destination-outbound-socket-binding=my-smtp-binding:add(host=localhost, port=25)
```

3. Ajouter un serveur SMTP avec `outbound-socket-binding-ref`. La commande suivante ajoute un SMTP nommé `my-smtp-binding` et définit un nom d'utilisateur, un mot de passe et une configuration TLS.

```
/subsystem=mail/mail-session=mySession/server=smtp:add(outbound-socket-binding-ref=my-smtp-binding, username=user, password=pass, tls=true)
```

4. Répéter ce processus pour POP3 et IMAP :

```
/socket-binding-group=standard-sockets/remote-destination-outbound-socket-binding=my-pop3-binding:add(host=localhost, port=110)
```

```
/subsystem=mail/mail-session=mySession/server=pop3:add(outbound-socket-binding-ref=my-pop3-binding, username=user, password=pass)
```

```
/socket-binding-group=standard-sockets/remote-destination-outbound-socket-binding=my-imap-binding:add(host=localhost, port=143)
```

```
/subsystem=mail/mail-session=mySession/server=imap:add(outbound-socket-binding-ref=my-imap-binding, username=user, password=pass)
```

5. Pour utiliser un serveur personnalisé, créer un nouveau serveur mail personnalisé sans la liaison de socket sortante (comme c'est optionnel) et fournir à la place l'information hôte comme faisant partie des propriétés.

```
/subsystem=mail/mail-session=mySession/custom=myCustomServer:add(username=user, password=pass, properties={"host" => "myhost", "my-property" => "value"})
```

Lorsque vous définissez les protocoles personnalisés, n'importe quel nom de propriété qui contient un point (.) est considéré comme un nom complet et est passé tel qu'il est fourni. N'importe quel autre format (*my-property*, par exemple) sera traduit dans le format suivant : **mail. server-name .my-property**.

Vous trouverez ci-dessous un exemple de configuration XML complète qui montre un format personnalisé de l'attribut `custom-server` :

```
<subsystem xmlns="urn:jboss:domain:mail:1.1">
  <mail-session jndi-name="java:/Mail" from="user.name@domain.org">
    <smtp-server outbound-socket-binding-ref="mail-smtp" tls="true">
      <login name="user" password="password"/>
    </smtp-server>
    <pop3-server outbound-socket-binding-ref="mail-pop3"/>
    <imap-server outbound-socket-binding-ref="mail-imap">
      <login name="nobody" password="password"/>
    </imap-server>
  </mail-session>
  <mail-session debug="true" jndi-name="java:jboss/mail/Default">
    <smtp-server outbound-socket-binding-ref="mail-smtp"/>
  </mail-session>
  <mail-session debug="true" jndi-name="java:jboss/mail/Custom">
    <custom-server name="smtp">
      <login name="username" password="password"/>
      <property name="host" value="mail.example.com"/>
    </custom-server>
    <custom-server name="pop3" outbound-socket-binding-ref="mail-
pop3">
      <property name="custom_prop" value="some-custom-prop-value"/>
      <property name="some.fully.qualified.property" value="fully-
qualified-prop-name"/>
    </custom-server>
  </mail-session>
  <mail-session debug="true" jndi-name="java:jboss/mail/Custom2">
    <custom-server name="pop3" outbound-socket-binding-ref="mail-
pop3">
      <property name="custom_prop" value="some-custom-prop-value"/>
    </custom-server>
  </mail-session>
</subsystem>
```

[Rapporter un bogue](#)

CHAPITRE 21. ENTERPRISE JAVABEANS

21.1. INTRODUCTION

21.1.1. Entreprise JavaBeans

Enterprise JavaBeans (EJB) 3.1 est une API pour développer des applications de Java EE distribuées, transactionnelles, sécurisées et portables grâce à l'utilisation des composants côté serveur appelés Enterprise Beans. Enterprise Beans implémente la logique métier d'une application, de manière découplée, qui encourage la réutilisation. Enterprise JavaBeans 3.1 est documenté dans la spécification Java EE JSR-318.

JBoss EAP 6 prend en charge la génération d'applications qui utilisent la spécification Enterprise JavaBeans 3.1.

[Rapporter un bogue](#)

21.1.2. Entreprise JavaBeans pour Administrateurs

Les administrateurs JBoss ont de nombreuses options de configuration disponibles pour contrôler la performance des Beans Enterprise dans JBoss EAP 6. Ces options sont accessibles par la console de gestion ou par l'outil de configuration de ligne de commande. Éditer le fichier de configuration du serveur XML pour appliquer les modifications est également possible mais non recommandé

Les options de configuration EJB se situent dans des endroits légèrement différents de la console de gestion, selon que le serveur exécute ou non.

1. Cliquer sur l'onglet **Configuration** qui se trouve en haut de la console de gestion.
2. Si vous êtes en mode de domaine, sélectionner un profil à partir du menu déroulant **Profiles** en haut et à gauche.
3. Étendre le menu **Subsystems**.
4. Étendre le menu **Container**, puis sélectionner **EJB 3**.

[Rapporter un bogue](#)

21.1.3. Beans Enterprise

Les beans enterprise sont des composants d'applications côté serveur, définis dans la spécification Enterprise JavaBeans (EJB) 3.1, JSR-318. Les beans enterprise sont conçus pour l'implémentation d'une logique commerciale d'application d'une manière découplée, pour encourager sa réutilisation.

Les beans enterprise sont écrits comme des classes Java et sont annotés avec les annotations EJB appropriées. Ils peuvent être déployés sur le serveur d'applications dans leurs propres archives (un fichier JAR) ou être déployés dans le cadre d'une application Java EE. Le serveur d'applications gère le cycle de vie de chaque bean entreprise et leur fournit des services comme la sécurité, les transactions et la gestion de concurrence.

Un bean enterprise peut également définir un nombre d'interfaces de métier. Les interfaces de métier vous proposent un plus grand contrôle sur les méthodes bean disponibles aux clients et peut également vous donner accès aux clients qui exécutent dans les JVM à distance.

Il existe trois types de beans enterprise : les session beans, les message-driven beans et les entity beans.



IMPORTANT

Les entity beans sont maintenant obsolètes dans EJB 3.1 et Red Hat recommande d'utiliser des entités JPA à la place. Red Hat ne recommande d'utiliser des entity beans que pour les compatibilités rétroactives avec les systèmes hérités.

[Rapporter un bogue](#)

21.1.4. Session Beans

Les session beans sont des beans Enterprise qui encapsulent un ensemble de processus métier connexes ou des tâches, et qui sont injectés dans les classes qui en ont fait la demande. Il existe trois types de session beans : sans état, avec état et singleton.

[Rapporter un bogue](#)

21.1.5. Message-Driven Beans

Les Message-driven Beans (MDB) fournissent un modèle basé-événement pour le développement de l'application. Les méthodes des MDB ne sont pas injectées ou invoquées du code client mais sont déclenchées par la réception de messages d'un service de messagerie comme le serveur Java Messaging Service (JMS). La spécification Java EE 6 exige que JMS soit pris en charge, mais les autres systèmes de messagerie peuvent être supportés également.

[Rapporter un bogue](#)

21.2. CONFIGURER LES BEAN POOLS

21.2.1. Bean Pools

JBoss EAP 6 maintient un certain nombre d'instances de beans stateless enterprise déployés en mémoire pour procurer une performance plus rapide. Cette technique s'appelle le Bean Pooling. Quand on a besoin d'un bean, le serveur de l'application peut en prendre un du pool qui convient parmi les beans déjà existants au lieu d'en instancier un nouveau. Quand le bean n'est plus requis, il est renvoyé dans le pool en vue d'être réutilisé.

Les bean pools sont configurés et maintenus séparément dans le cas des stateless session beans et des beans basés messages.

L'annotation `@org.jboss.ejb3.annotation.Pool` peut être utilisée sur les EJB pour identifier le pool qui doit être utilisé pour cet EJB. Cette annotation pointe vers le nom de ce pool.

[Rapporter un bogue](#)

21.2.2. Créer un bean pool

Les bean pools peuvent être créés par l'intermédiaire de la console de gestion ou du CLI.

Les bean pools peuvent également être créés en ajoutant la configuration du bean pool requis au fichier de configuration du serveur en utilisant l'éditeur de texte. [Exemple 21.2, « Exemple de configuration XML »](#) est un exemple de configuration.

Procédure 21.1. Créer un bean pool par la console de gestion

1. Se connecter à la console de gestion. Consulter [Section 3.4.2, « Se connecter à la console de gestion »](#).
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Bean Pools**.
3. Cliquer sur **Add**. Le dialogue **Add EJB3 Bean Pools** apparaîtra.
4. Donner les informations requises, les valeurs de **Name**, **Max Pool Size**, **Timeout** et l'unité de **Timeout**.
5. Cliquer sur **Save** pour terminer.

Procédure 21.2. Créer un bean pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#).
2. Utiliser l'opération **add** avec la syntaxe suivante.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:add(max-pool-size=MAXSIZE, timeout=TIMEOUT, timeout-unit="UNIT")
```

- o Remplacer *BEANPOOLNAME* par le nom requis de bean pool.
- o Remplacer *MAXSIZE* par le nom requis de bean pool.
- o Remplacer *TIMEOUT*
- o Remplacer *UNIT* par l'unité de temps requise. Les valeurs permises sont les suivantes : **NANOSECONDS**, **MICROSECONDS**, **MILLISECONDS**, **SECONDS**, **MINUTES**, **HOURS**, et **DAYS**.

3. Utiliser l'opération **read-resource** pour confirmer la création d'un bean pool.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:read-resource
```

Exemple 21.1. Créer un bean pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/strict-max-bean-instance-pool=ACCTS_BEAN_POOL:add(max-pool-size=500, timeout=5000, timeout-unit="SECONDS")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Exemple 21.2. Exemple de configuration XML

```

<subsystem xmlns="urn:jboss:domain:ejb3:1.2">
    <pools>
        <bean-instance-pools>
            <strict-max-pool name="slsb-strict-max-pool" max-pool-
size="20"
                instance-acquisition-timeout="5"
                instance-acquisition-timeout-unit="MINUTES" />
            <strict-max-pool name="mdb-strict-max-pool" max-pool-size="20"
                instance-acquisition-timeout="5"
                instance-acquisition-timeout-unit="MINUTES" />
        </bean-instance-pools>
    </pools>
</subsystem>

```

[Rapporter un bogue](#)

21.2.3. Supprimer un bean pool

Les bean pool non utilisés peuvent être supprimés par la console de gestion.

Prérequis :

- Le bean pool que vous souhaitez supprimer ne peut pas être en cours d'utilisation. Consulter [Section 21.2.5, « Assigner des beans pools aux beans de session et aux beans basés messages »](#) pour vérifier qu'ils ne sont pas en cours d'utilisation.

Procédure 21.3. Supprimer un bean pool par la console de gestion.

1. Se connecter à la console de gestion. Consulter [Section 3.4.2, « Se connecter à la console de gestion »](#).
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Bean Pools**.
3. Cliquer sur le bean pool que vous souhaitez supprimer de la liste.
4. Cliquer sur **Remove**. La boîte de dialogue **Remove Item** apparaîtra.
5. Cliquer sur **Confirm** pour confirmer.

Procédure 21.4. Supprimer un bean pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#).
2. Utiliser l'opération `remove` avec la syntaxe suivante.

■

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:remove
```

- Remplacer *BEANPOOLNAME* par le nom requis de bean pool.

Exemple 21.3. Supprimer un bean pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/strict-max-bean-instance-
pool=ACCTS_BEAN_POOL:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

[Rapporter un bogue](#)

21.2.4. Modifier un bean pool

Les bean pools peuvent être modifiés par la console de gestion.

Procédure 21.5. Modifier un bean pool par la console de gestion

1. Connectez-vous à la console de gestion. [Section 3.4.2, « Se connecter à la console de gestion »](#)
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Bean Pools**.
3. Cliquer sur le bean pool que vous souhaitez modifier.
4. Cliquer sur **Edit**.
5. Modifier les informations que vous souhaitez. Seules les valeurs de **Max Pool Size**, **Timeout**, et de l'unité de **Timeout Unit** peuvent être modifiées.
6. Cliquer sur le bouton **Save** pour terminer.

Procédure 21.6. Modifier un bean pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#).
2. Utiliser l'opération `write-attribute` avec la syntaxe suivante pour chaque attribut du bean pool à modifier.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:write-
attribute(name="ATTRIBUTE", value="VALUE")
```

- Remplacer *BEANPOOLNAME* par le nom requis de bean pool.
 - Remplacer *ATTRIBUTE* par le nom de l'attribut à modifier. Les attributs ne pouvant pas être modifiés de cette façon sont **max-pool-size**, **timeout**, et **timeout-unit**.
 - Remplacer *VALUE* par la valeur requise de l'attribut.
3. Utiliser l'opération `read-resource` pour confirmer les changements au bean pool.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:read-resource
```

Exemple 21.4. Définir la valeur de timeout d'un bean pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/strict-max-bean-instance-pool=HSBeanPool:write-attribute(name="timeout", value="1500")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

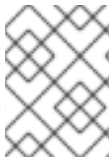
[Rapporter un bogue](#)

21.2.5. Assigner des beans pools aux beans de session et aux beans basés messages

Les administrateurs de systèmes JBoss peuvent assigner des beans pools individuels que les sessions beans et les beans basés-messages peuvent utiliser. Les beans pools peuvent être distribués par la console de gestion ou le CLI.

Par défaut, deux beans pools sont fournis, `slsb-strict-max-pool` et `mdb-strict-max-pool` pour les stateless sessions beans et les beans basés-messages respectivement.

Pour créer ou modifier des beans pools, consulter [Section 21.2.2, « Créer un bean pool »](#) et [Section 21.2.4, « Modifier un bean pool »](#).



NOTE

Si vous utilisez l'annotation `@Pool` sur un EJB en particulier, cela remplacera tous les paramètres par défaut créés en utilisant les procédures suivantes.

Procédure 21.7. Allouer des beans pools pour les session beans ou pour les beans basés-message par la console de gestion.

1. Connectez-vous à la console de gestion. [Section 3.4.2, « Se connecter à la console de gestion »](#)
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Container**.
3. Cliquer sur **Edit**.
4. Sélectionner le bean pool à utiliser pour chaque type de bean à partir de la combo-box appropriée.
5. Cliquer sur le bouton **Save** pour terminer.

Procédure 21.8. Allouer des beans pools pour les session beans ou pour les beans basés-message par le CLI.

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#).
2. Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=ejb3:write-attribute(name="BEANTYPE", value="BEANPOOL")
```

- - Remplacer *BEANTYPE* par `default-mdb-instance-pool` pour les beans basés-messages ou `default-slsb-instance-pool` pour les stateless sessions beans.
 - Remplacer *BEANPOOL* par le nom du bean pool à assigner.
3. Utiliser l'opération `read-resource` pour confirmer les changements.

```
/subsystem=ejb3:read-resource
```

Exemple 21.5. Assigner un bean pool pour les sessions beans par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3:write-attribute(name="default-slsb-instance-pool", value="LV_SLSB_POOL") {"outcome" => "success"} [standalone@localhost:9999 /]
```

Exemple 21.6. Exemple de configuration XML

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">
  <session-bean>
    <stateless>
      <bean-instance-pool-ref pool-name="slsb-strict-max-pool"/>
    </stateless>
    <stateful default-access-timeout="5000" cache-ref="simple"/>
    <singleton default-access-timeout="5000"/>
  </session-bean>
  <mdb>
    <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
    <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
  </mdb>
</subsystem>
```

[Rapporter un bogue](#)

21.3. CONFIGURER LES EJB THREAD POOLS

21.3.1. Enterprise Bean Thread Pools

JBoss EAP 6 maintient un certain nombre d'instances d'objets de thread Java en mémoire à utiliser par les services de beans enterprise, y compris l'invocation à distante, le service de minuteur et l'invocation asynchrone.

Cette technique s'appelle le «thread pooling». Elle fournit une meilleure performance en éliminant l'étape de création de threads et procure à l'administrateur de services un moyen de contrôler l'utilisation des ressources.

On peut créer des pools de threads multiples par divers paramètres et chaque service peut recevoir ainsi un thread pool différent.

[Rapporter un bogue](#)

21.3.2. Créer un thread pool

Les thread pool EJB peuvent être créés par la console de gestion ou le CLI.

Procédure 21.9. Créer un thread pool EJB par la console de gestion

1. Connectez-vous à la console de gestion. [Section 3.4.2, « Se connecter à la console de gestion »](#)
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Thread Pools**.
3. Cliquer sur **Add**. Le dialogue **Add EJB3 Thread Pools** apparaîtra.
4. Donnez les informations requises, les valeurs de **Name**, **Max Threads**, et **Keep-Alive Timeout**.
5. Cliquer sur le bouton **Save** pour terminer.

Procédure 21.10. Créer un thread pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#).
2. Utiliser l'opération **add** avec la syntaxe suivante.

```
/subsystem=ejb3/thread-pool=THREADPOOLNAME:add(max-threads=MAXSIZE,
keepalive-time={"time"=>"TIME", "unit"=>"UNIT"})
```

- o Remplacer *BEANPOOLNAME* par le nom requis de thread pool.
 - o Remplacer *MAXSIZE* par la taille maximum de thread Pool.
 - o Remplacer *UNIT* par l'unité de temps requise de «keep-alive time». Les valeurs permises sont les suivantes : **NANOSECONDS**, **MICROSECONDS**, **MILLISECONDS**, **SECONDS**, **MINUTES**, **HOURS**, et **DAYS**.
 - o Remplacer *TIME* par la valeur (entier relatif) du «keep-alive time». Cette valeur doit correspondre à un nombre d'unités *UNIT*.
3. Utiliser l'opération **read-resource** pour confirmer la création d'un bean pool.

```
/subsystem=ejb3/strict-max-bean-instance-pool=THREADPOOLNAME:read-
resource
```

Exemple 21.7. Créer un thread pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-
pool=testmepool:add(max-threads=50, keepalive-time={"time"=>"150",
"unit"=>"SECONDS"})
```

```
{"outcome" => "success"}  
[standalone@localhost:9999 /]
```

Exemple 21.8. Exemple de configuration XML

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">  
  <thread-pools>  
    <thread-pool name="default" max-threads="20" keepalive-  
time="150"/>  
  </thread-pools>  
</subsystem>
```

[Rapporter un bogue](#)

21.3.3. Supprimer un thread pool

Les thread pool non utilisés peuvent être supprimés par la console de gestion.

Conditions préalables

- La thread pool que vous souhaitez supprimer ne peut pas être en cours d'utilisation. Voir les tâches suivantes pour vérifier que le thread pool n'est pas en cours d'utilisation :
 - [Section 21.6.2, « Configurer le service de minuterie EJB3 »](#)
 - [Section 21.7.2, « Configurer le thread pool du service d'invocations asynchrones EJB3 »](#)
 - [Section 21.8.2, « Configurer EJB3 Remote Service »](#)

Procédure 21.11. Supprimer un thread pool EJB par la console de gestion

1. Connectez-vous à la console de gestion. [Section 3.4.2, « Se connecter à la console de gestion »](#).
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Thread Pools**.
3. Cliquer sur le thread pool que vous souhaitez supprimer.
4. Cliquer sur **Remove**. La boîte de dialogue **Remove Item** apparaîtra.
5. Cliquer sur **Confirm**.

Procédure 21.12. Supprimer un thread pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#).
2. Utiliser l'opération `remove` avec la syntaxe suivante.


```
/subsystem=ejb3/thread-pool=THREADPOOLNAME:remove
```

- Remplacer *THREADPOOLNAME* par le nom requis de thread pool.

Exemple 21.9. Supprimer un thread pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-
pool=ACCTS_THREADS:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

[Rapporter un bogue](#)

21.3.4. Modifier un thread pool

Les administrateurs JBoss peuvent modifier les thread pools par la console de gestion ou le CLI.

Procédure 21.13. Modifier un thread pool par la console de gestion

1. Connectez-vous à la console de gestion. [Section 3.4.2, « Se connecter à la console de gestion »](#).
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Thread Pools**.
3. Cliquer sur le thread pool que vous souhaitez modifier.
4. Cliquer sur **Edit**.
5. Modifier les détails que vous souhaitez modifier. Vous ne pourrez modifier que les valeurs suivantes : **Thread Factory**, **Max Threads**, **Keepalive Timeout**, et **Keepalive Timeout Unit**.
6. Cliquer sur le bouton **Save** pour terminer.

Procédure 21.14. Modifier un thread pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#).
2. Utiliser l'opération `write_attribute` avec la syntaxe suivante pour chaque attribut du thread pool à modifier.

```
/subsystem=ejb3/thread-pool=THREADPOOLNAME:write-
attribute(name="ATTRIBUTE", value="VALUE")
```

- Remplacer *THREADPOOLNAME* par le nom requis de thread pool.
- Remplacer *ATTRIBUTE* par le nom de l'attribut à modifier. Les attributs ne pouvant pas être modifiés de cette façon sont `keepalive-time`, `max-thread`, et `thread-factory`.
- Remplacer *VALUE* par la valeur requise de l'attribut.

3. Utiliser l'opération `read-resource` pour confirmer les changements au thread pool.

```
/subsystem=ejb3/thread-pool=THREADPOOLNAME:read-resource
```

IMPORTANT

Quand vous changez la valeur de l'attribut `keepalive-time` par le CLI, la valeur requise correspond à une représentations d'objet. La syntaxe sera la suivante :

```
/subsystem=ejb3/thread-pool=THREADPOOLNAME:write-attribute(name="keepalive-time", value={"time" => "VALUE", "unit" => "UNIT"})
```

Exemple 21.10. Définir la xaleur maximum d'un thread pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-pool=HSThreads:write-attribute(name="max-threads", value="50") {"outcome" => "success"}
[standalone@localhost:9999 /]
```

Exemple 21.11. Définir la valeur `keepalive-time` d'un thread pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-pool=HSThreads:write-attribute(name="keepalive-time", value={"time"=>"150"}) {"outcome" => "success"}
[standalone@localhost:9999 /]
```

[Rapporter un bogue](#)

21.4. CONFIGURER LES SESSION BEANS

21.4.1. Timeout d'accès au session bean

Les seesion beans stateful et singleton ont une valeur de délai d'accès précise pour les accès simultanés. Cette valeur correspond à la période pendant laquelle une demande de méthode de bean de session peut être bloquée avant qu'il y ait timeout.

La valeur de timeout et l'unité de temps utilisées peut être spécifiée grâce à l'annotation `@javax.ejb.AccessTimeout` sur la méthode. Elle peut être spécifiée sur le bean de session (qui s'applique à toutes les méthodes de beans) et sur certaines méthodes pour remplacer la configuration du bean.

Si non spécifié, JBoss EAP 6 fournit une valeur de timeout de 5000 millisecondes.

Consulter Javadocs pour `AccessTimeout` à l'adresse suivante :

<http://docs.oracle.com/javaee/6/api/javax/ejb/AccessTimeout.html>

[Rapporter un bogue](#)

21.4.2. Définir les valeurs de timeout d'accès aux beans de session par défaut

Les administrateurs de systèmes JBoss peuvent spécifier les valeurs de timeout par défaut des session beans de stateful ou singleton. Les valeurs de timeout par défaut peuvent être modifiées par la console de gestion ou le CLI. La valeur par défaut est de 5000 millisecondes.

Procédure 21.15. Définir les valeurs de timeout d'accès aux session beans par défaut par la console de gestion

1. Connectez-vous à la console de gestion. Voir [Section 3.4.2, « Se connecter à la console de gestion »](#).
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Container**.
3. Cliquer sur **Edit**. Le champ de la zone **Details** est maintenant modifiable.
4. Saisir les valeurs qui conviennent dans **Stateful Access Timeout** et/ou dans les cases de texte **Singleton Access Timeout**.
5. Cliquer sur le bouton **Save** pour terminer.

Procédure 21.16. Définir les valeurs de timeout d'accès aux session beans par le CLI.

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#).
2. Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=ejb3:write-attribute(name="BEANTYPE", value=TIME)
```

- o Remplacer *BEANTYPE* par `default-stateful-bean-access-timeout` pour les sessions beans stateful, ou `default-singleton-bean-access-timeout` pour les sessions bean singleton.
- o Remplacer *TIME* par la valeur de timeout qui convient.

3. Utiliser l'opération `read-resource` pour confirmer les changements.

```
/subsystem=ejb3:read-resource
```

Exemple 21.12. Définir la valeur de timeout d'accès aux beans stateful par le CLI à 9000.

```
[standalone@localhost:9999 /] /subsystem=ejb3:write-attribute(name="default-stateful-bean-access-timeout", value=9000)
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Exemple 21.13. Exemple de configuration XML

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">
  <session-bean>
```

```

    <stateless>
      <bean-instance-pool-ref pool-name="slsb-strict-max-pool"/>
    </stateless>
    <stateful default-access-timeout="5000" cache-ref="simple"/>
    <singleton default-access-timeout="5000"/>
  </session-bean>
</subsystem>

```

[Rapporter un bogue](#)

21.4.3. Timeout de transaction de Session Bean

L'annotation `TransactionTimeout` est utilisée pour spécifier le timeout de transaction pour une méthode donnée. La valeur de l'annotation est le timeout utilisé pour un élément d'unité donnée. Elle doit correspondre à un nombre entier positif ou à 0. Quand 0 est indiqué, le timeout configuré de domaine par défaut sera utilisé.

L'élément d'unité indique la façon dont on mesure la valeur.



NOTE

Une valeur inférieure à une valeur en secondes est considéré comme une erreur, même si la valeur calculée peut résulter en un nombre entier de secondes. Par exemple :
`@TransactionTimeout(value = 1000, unit=TimeUnit.MILLISECONDS)`

Indiquer un timeout de transactions dans le descripteur de déploiement

L'élément `trans-timeout` est utilisé pour définir le timeout de transaction pour les méthodes d'interface de listener de message, composant, home et business; aucune méthode d'affichage d'interface; méthodes de points de terminaison de service web; et les méthodes de rappel de timeout. L'élément `trans-timeout` réside dans l'espace-nom `urn:trans-timeout` et fait partie de l'élément `container-transaction` standard défini dans l'espace-nom `jboss`.

Exemple 21.14. Extrait de configuration de trans-timeout XML

```

<ejb-name>*</ejb-name>
<tx:trans-timeout>
  <tx:timeout>2</tx:timeout>
  <tx:unit>Seconds</tx:unit>
</tx:trans-timeout>

```

`ejb-name` peut être spécifié pour un nom EJB particulier, ou un caractère générique (*). Spécifier un caractère générique (*) pour `ejb-name` signifie que le timeout de transaction particulière sera le timeout par défaut de tous les EJB de l'application.

[Rapporter un bogue](#)

21.5. CONFIGURER LES MESSAGE-DRIVEN BEANS

21.5.1. Définir l'adaptateur de ressources par défaut des beans basés-message

Les administrateurs de systèmes JBoss peuvent spécifier l'adaptateur de ressources par défaut utilisé par les beans basés-message. L'adaptateur de ressources par défaut peut être spécifié par la console de gestion ou le CLI. L'adaptateur de ressources fourni par défaut dans JBoss EAP 6 est `hornetq-ra`.

Procédure 21.17. Définir l'adaptateur de ressources par défaut pour les beans basés-message par la console de gestion.

1. Connectez-vous à la console de gestion. [Section 3.4.2, « Se connecter à la console de gestion »](#)
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Container**.
3. Cliquer sur **Edit**. Le champ de la zone **Details** est maintenant modifiable.
4. Saisir le nom de l'adaptateur de la ressource à utiliser dans la case de texte **Default Resource Adapter**.
5. Cliquer sur le bouton **Save** pour terminer.

Procédure 21.18. Définir l'adaptateur de ressources par défaut pour les beans basés-messages par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir [Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »](#).
2. Utiliser l'opération `write-attribute` avec la syntaxe suivante.

```
/subsystem=ejb3:write-attribute(name="default-resource-adapter-name", value="RESOURCE-ADAPTER")
```

Remplacer `RESOURCE-ADAPTER` par le nom de l'adaptateur de ressources à utiliser.

3. Utiliser l'opération `read-resource` pour confirmer les changements.

```
/subsystem=ejb3:read-resource
```

Exemple 21.15. Définir l'adaptateur de ressources par défaut pour les beans basés-messages par le CLI

```
[standalone@localhost:9999 subsystem=ejb3] /subsystem=ejb3:write-attribute(name="default-resource-adapter-name", value="EDIS-RA")
{"outcome" => "success"}
[standalone@localhost:9999 subsystem=ejb3]
```

Exemple 21.16. Exemple de configuration XML

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">
  <mdb>
    <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
  </mdb>
</subsystem>
```

```

    <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
  </mdb>
</subsystem>

```

[Rapporter un bogue](#)

21.6. CONFIGURER LE SERVICE EJB3 TIMER

21.6.1. Service de minuterie EJB3

Le service de minuterie EJB3 est un service standard Java EE 6 pour programmer l'invocation de méthodes à partir de beans enterprise. Les beans de sessions Stateless, les beans de sessions Singleton et les beans basés-messages peuvent tous programmer un rappel de n'importe quelle méthode qui leur appartient à un moment précis, après un intervalle de temps, ou à un intervalle récurrent, ou encore sur la base d'un calendrier.

[Rapporter un bogue](#)

21.6.2. Configurer le service de minuterie EJB3

Le service Timer Service EJB3 peut être configuré par la console de gestion ou par l'interface CLI. Vous pouvez configurer le thread pool utilisé pour l'invocation de bean programmée, et le répertoire ou la source de données utilisée pour stocker les données du service de minuterie. Vous pourrez modifier le répertoire par défaut du Timer Service s'il y a un stockage plus rapide que le répertoire par défaut.

Procédure 21.19. Configurer le Thread Pool du service de minuterie EJB3 par la console de gestion

Conditions préalables

- Le Thread Pool qui doit être utilisé par le service de minuterie EJB3 doit déjà avoir été créé.
1. Connectez-vous à la console de gestion.
 2. Cliquez sur l'onglet **Configuration** en haut de l'écran. Étendez le menu **Container** et sélectionnez **EJB 3**. Sélectionnez l'onglet **Services**, cliquez sur **Timer Services**. Cliquez sur **Edit**.
 3. Cliquez sur la liste de menu déroulant du service **EJB3 Thread Pool** et cliquez sur le nom du Thread Pool que vous souhaitez.
 4. Démarrez à nouveau l'instance JBoss EAP.

Procédure 21.20. Configurez le Thread Pool du service de minuterie EJB3 par l'interface de commandes CLI



NOTE

Ajouter le préfixe `/profile=PROFILE_NAME` à la commande si vous devez appliquer les changements à un domaine géré.

1. Exécuter la commande d'interface CLI suivante.

■

```
/subsystem=ejb3/service=timer-service:write-attribute(name=thread-pool-name,value="thread-pool-name")
```

2. Démarrez à nouveau l'instance JBoss EAP.

Procédure 21.21. Configurer le répertoire du service de minuterie EJB3 par la console de gestion

1. Connectez-vous à la console de gestion.
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Services**, cliquer sur **Timer Services**. Cliquer sur **Edit**.
3. Saisir les valeurs souhaitées dans les champs **Path** et **Relative To**.
4. Cliquer sur le bouton **Enregistrer**.
5. Démarrez à nouveau l'instance JBoss EAP.

Procédure 21.22. Configurer le répertoire du service de minuterie EJB3 par l'interface CLI

1. Selon le chemin que vous souhaitez suivre, exécuter une ou deux des commandes CLI suivantes. Quel que soit le chemin choisi, vous pouvez utiliser une valeur système - par exemple, `${jboss.server.data.dir}`.



NOTE

Ajouter le préfixe `/profile=PROFILE_NAME` à la commande si vous devez appliquer les changements à un domaine géré.

```
/subsystem=ejb3/service=timer-service/file-data-store=default-file-store:write-attribute(name=path,value="path")
```

```
/subsystem=ejb3/service=timer-service/file-data-store=default-file-store:write-attribute(name=relative-to,value="relative-path")
```

2. Démarrez à nouveau l'instance JBoss EAP.

Procédure 21.23. Configurez le service de minuterie EJB3 à utiliser une source de données par l'interface de commandes CLI

À partir de JBoss EAP 6.4, vous pourrez configurer le service de minuterie EJB3 pour qu'il utilise une source de données à la place d'un répertoire local. Il y a un moindre coût de performance à cette option, mais cela a comme avantage de diminuer le risque aux données du minuteur en cas de problème de stockage local.

Une fois que le service de minuterie EJB3 est configuré pour utiliser une source de données, vous devrez soit configurer un déploiement EJB pour qu'il utilise le datastore ou le configurer comme valeur par défaut pour tous les déploiements. Pour obtenir des instructions sur la façon de procéder, voir la procédure *Configure one or all EJB3 Deployments to use the Datasource*

Conditions préalables

- La source de données à utiliser par le service de minuterie EJB3 doit déjà exister et la base de données sous-jacente doit être compatible pour et être configurée en mode s'isolation READ_COMMITTED or SERIALIZABLE.



NOTE

Ajouter le préfixe `/profile=PROFILE_NAME` à la commande si vous devez appliquer les changements à un domaine géré.

- Exécuter la commande d'interface CLI suivante.
 - `datastore_name` - Un nom de votre choix
 - `datasource_name` - le nom JNDI de la source de données JDBC utilisée pour le stockage.
 - `database` - soit `postgresql`, `mssql`, `sybase`, `mysql`, `oracle`, `db2`, ou `hsq1`.
 - `partition_name` - un nom de votre choix. Cet attribut est utilisé pour distinguer les minuteriers d'instance particulière de serveur quand plusieurs instances de JBoss EAP partagent la même base de données pour stocker des minuteriers EJB. Dans ce cas, chaque instance de serveur doit avoir son propre nom de partition. Si la base de données est utilisée par une seule instance de serveur, vous pouvez laisser cet attribut vide.

```
/subsystem=ejb3/service=timer-service/database-data-store=datastore_name:add(datasource-jndi-name='java:/datasource_name', database='database', partition='partition_name')
```

Procédure 21.24. Configurer un ou tous les déploiements EJB3 pour qu'ils puissent utiliser la source de données

Vous pouvez soit configurer un déploiement EJB3 pour utiliser la source de données du Timer Service ou configurez-le comme la valeur par défaut pour *tous* les déploiements.

- ◦ Pour configurer un déploiement EJB3 pour qu'il utilise une source de données, modifier le fichier `jboss-ejb3.xml` du déploiement pour que la section `timer` ressemble à ce qui suit. Remplacer `datastore_name` par le nom du datastore.

```
[<assembly-descriptor>
  <timer:timer>
    <ejb-name>*</ejb-name>
    <timer:persistence-store-name>datastore_name</timer:persistence-store-name>
  </timer:timer>
</assembly-descriptor>
```

- Pour configurer la source de données par défaut pour *tous les* déploiements, exécutez la commande d'interface CLI suivante, puis redémarrez l'instance de JBoss EAP. Remplacer `datastore_name` par le nom du datastore.

**NOTE**

Ajouter le préfixe `/profile=PROFILE_NAME` à la commande si vous devez appliquer les changements à un domaine géré.

```
[/subsystem=ejb3/service=timer-service:write-attribute(name=default-data-store,value=datastore_name)
```

[Rapporter un bogue](#)

21.7. CONFIGURER LE SERVICE D'INVOCATION ASYNCHRONE EJB

21.7.1. Service d'invocations asynchrones EJB3

Le service d'invocations asynchrones est un service de conteneurs JavaBeans Enterprise qui gère l'invocation asynchrone des méthodes de beans de sessions. Ce service maintient un certain nombre d'invocations asynchrones configurables (Thread Pool) qui sont allouées pour l'exécution de méthodes asynchrones.

Enterprise JavaBeans 3.1 permet à toute méthode de bean de session (stateful, stateless, ou singleton) d'être annotée pour permettre l'exécution asynchrone.

[Rapporter un bogue](#)

21.7.2. Configurer le thread pool du service d'invocations asynchrones EJB3

Les administrateurs JBoss peuvent configurer le service d'invocations asynchrones EJB3 dans la console de gestion JBoss EAP 6 pour permettre l'utilisation d'un thread pool spécifique.

Procédure 21.25. Configurer `http://francegourmet.com.au/product-category/snails-and-mushrooms/`

1. Connectez-vous à la console de gestion. Voir [Section 3.4.2, « Se connecter à la console de gestion »](#).
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Services**, cliquer sur **Async Service**.
3. Cliquer sur **Edit**.
4. Sélectionner le thread pool à utiliser dans la liste. Le thread pool devra déjà avoir été créé.
5. Cliquer sur le bouton **Save** pour terminer.

[Rapporter un bogue](#)

21.8. CONFIGURER EJB3 REMOTE INVOCATION SERVICE

21.8.1. EJB3 Remote Service

Le service EJB3 Remote gère l'exécution à distance des Beans Enterprise dans les interfaces commerciales à distance.

[Rapporter un bogue](#)

21.8.2. Configurer EJB3 Remote Service

Les administrateurs JBoss peuvent configurer EJB3 Remote Service dans la console de gestion de JBoss EAP 6. Les fonctions pouvant être configurées sont le thread pool utilisé pour l'invocation de beans programmés et le connecteur sur lequel le réseau EJB3 Remoting est enregistré.

Procédure 21.26. Configurer EJB3 Remote Service

1. Connectez-vous à la console de gestion. Voir [Section 3.4.2, « Se connecter à la console de gestion »](#).
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Services**, cliquer sur **Remote Service**.
3. Cliquer sur **Edit**.
4. Vous pouvez sélectionner un EJB3 Thread Pool différent pour Remote Service si les Thread Pools supplémentaires ont été configurés. Vous pouvez changer le connecteur utilisé pour enregistrer le Canal EJB Remoting.
5. Cliquer sur le bouton **Save** pour terminer.

[Rapporter un bogue](#)

21.9. CONFIGURER LES EJB 2.X ENTITY BEANS

21.9.1. EJB Entity Beans

Les EJB Entity Beans sont un type de bean entreprise de la version 2.x de la spécification EJB qui représentait des données persistantes maintenues dans une base de données. Les entity beans ont été remplacées par les entités JPA et ont été officiellement listées pour être retirées (nettoyées) des versions futures de la spécification. Red Hat ne recommande pas l'utilisation des entity beans, sauf pour raison de compatibilité rétro-active.

Le support des entity beans est désactivé par défaut dans JBoss EAP 6.

[Rapporter un bogue](#)

21.9.2. Container-Managed Persistence

Container-Managed Persistence (CMP) est un service fourni par un serveur d'applications qui procure la persistance des données pour les beans entity.

[Rapporter un bogue](#)

21.9.3. Activer EJB 2.x Container-Managed Persistence

Container-Managed Persistence (CMP) est géré par l'extension `org.jboss.as.cmp`. CMP est activé par défaut dans le domaine géré et dans la configuration complète du serveur autonome, par ex. `standalone-full.xml`.

Pour activer CMP dans une configuration différente, ajouter le module `org.jboss.as.cmp` à la liste d'extensions actives dans le fichier de configuration du serveur.

```
<extensions>
    <extension module="org.jboss.as.cmp"/>
</extensions>
```

Une fois que l'extension est ajoutée, vous devrez aussi ajouter l'élément suivant au fichier de configuration XML du profil sous l'élément `<profile>`.

```
<subsystem xmlns="urn:jboss:domain:cmp:1.1"/>
```

Pour désactiver CMP dans une configuration de serveur, supprimer l'entrée d'extension du module `org.jboss.as.cmp`.

[Rapporter un bogue](#)

21.9.4. Configurer EJB 2.x Container-Managed Persistence

Le sous-système EJB 2.x Container Managed Persistence (CMP) peut être configuré pour spécifier un certain nombre de générateurs de clés. Les générateurs de clés sont utilisés pour produire des clés uniques pour identifier chaque entité persistée par le service CMP.

Il existe deux types de générateurs de clés : les générateurs de clés basés-UUID et les générateurs de clés HiLO

Les générateurs de clés basés-UUID

Un générateur de clés basé-UUID crée des clés qui utilisent un Identifiant Unique Universel (UUI). Les générateurs de clés UUID ont besoin uniquement d'avoir un nom unique; ils n'ont pas d'autre configuration.

Les générateurs de clés basé-UUID peuvent être ajoutés par le CLI avec la syntaxe suivante.

```
/subsystem=cmp/uuid-keygenerator=UNIQUE_NAME:add
```

Exemple 21.17. Ajouter le générateur de clés UUID

Pour ajouter un générateur de clés basé-UUID ayant pour nom `uuid_identities`, utiliser cette commande CLI :

```
/subsystem=cmp/uuid-keygenerator=uuid_identities:add
```

La configuration XML créée par cette commande est :

```
<subsystem xmlns="urn:jboss:domain:cmp:1.0">
    <key-generators>
        <uuid name="uuid_identities" />
    </key-generators>
</subsystem>
```

Générateurs de clés HiLo

Les générateurs de clés HiLo utilisent une base de données pour créer et stocker des clés d'identité des entités. Le générateur de clés HiLo doivent posséder des noms uniques et sont configurés avec des propriétés qui indiquent la source de données utilisée pour stocker les données, ainsi que les noms du tableau et colonnes qui stockent les clés.

Les générateurs de clés HiLo peuvent être ajoutés par le CLI grâce à la syntaxe de commande suivante:

```
/subsystem=cmp/hilo-keygenerator=UNIQUE_NAME/:add(property=value,
property=value, ...)
```

Exemple 21.18. Ajouter un générateur de clés HiLo

```
/subsystem=cmp/hilo-keygenerator=HiLoKeyGeneratorFactory:add(create-
table=true,create-table-ddl="create table HILOSEQUENCES (SEQUENCENAME
varchar(50) not null, HIGHVALUES integer not null, constraint hilo_pk
primary key (SEQUENCENAME))",data-
source=java:jboss/datasources/ExampleDS, id-
column=HIGHVALUES, sequence-column=SEQUENCENAME, table-
name=HILOSEQUENCES, sequence-name=general, block-size=10)
```

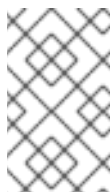
La configuration XML créée par cette commande est :

```
<subsystem xmlns="urn:jboss:domain:cmp:1.1">
  <key-generators>
    <hilo name="HiLoKeyGeneratorFactory">
      <block-size>10</block-size>
      <create-table>true</create-table>
      <create-table-ddl>create table HILOSEQUENCES
(SEQUENCENAME varchar(50) not null, HIGHVALUES integer not null,
constraint hilo_pk primary key (SEQUENCENAME))</create-table-ddl>
      <data-source>java:jboss/datasources/ExampleDS</data-
source>
      <id-column>HIGHVALUES</id-column>
      <sequence-column>SEQUENCENAME</sequence-column>
      <sequence-name>general</sequence-name>
      <table-name>HILOSEQUENCES</table-name>
    </hilo>
  </key-generators>
</subsystem>
```



NOTE

La taille de bloc doit être sur une valeur != 0, sinon la clé PKey générée ne sera pas incrémentée et, par conséquent, la création d'entités échouera accompagnée de l'exception DuplicateKeyException.



NOTE

Le select-hi-ddl doit être défini avec 'FOR UPDATE' en cas de cluster pour veiller à l'homogénéité. Toutes les bases de données ne prennent pas en charge la fonctionnalité de verrouillage.

[Rapporter un bogue](#)

21.9.5. Les propriétés de sous-système CMP pour les générateurs de clés HiLo

Tableau 21.1. Les propriétés de sous-système CMP pour les générateurs de clés HiLo

Propriété	Type des données	Description
block-size	long	La taille du bloc.
create-table	booléen	Si défini sur TRUE , le tableau table-name sera créé avec le contenu create-table-ddl si le tableau n'est pas trouvé.
create-table-ddl	chaîne	Les commandes DDL utilisées pour créer le tableau spécifié dans table-name si on ne trouve pas le tableau et create-table est défini à TRUE .
data-source	token	La source de données utilisée pour se connecter à la base de données.
drop-table	booléen	Pour déterminer si on doit supprimer les tableaux.
id-column	token	Le nom de la colonne ID.
select-hi-ddl	chaîne	La commande SQL qui retournera la plus grande clé actuellement stockée.
sequence-column	token	Le nom de la colonne de séquences.
sequence-name	token	Le nom de la séquence.
table-name	token	Nom de la table utilisée pour stocker les informations sur les clés

[Rapporter un bogue](#)

CHAPITRE 22. JAVA CONNECTOR ARCHITECTURE (JCA)

22.1. INTRODUCTION

22.1.1. Java EE Connector API (JCA)

JBoss EAP 6 fournit un support complet à la spécification Java EE Connector API (JCA). Voir [JSR 322: Java EE Connector Architecture 1.6](#) pour obtenir plus d'informations sur la spécification JCA.

Un adaptateur de ressources est un composant qui implémente l'architecture de Java EE Connector API. Il ressemble à un objet de source de données, mais fournit une connectivité à partir d'EIS (Enterprise Information System) vers un grand nombre de systèmes hétérogènes, comme des bases de données, systèmes de messagerie, traitement de transactions, et systèmes ERP (Enterprise Resource Planning).

[Rapporter un bogue](#)

22.1.2. Java Connector Architecture (JCA)

La Java EE Connector Architecture (JCA) définit une architecture standard pour les systèmes de Java EE pour les systèmes externes hétérogènes Enterprise Information Systems (EIS). Exemples de systèmes EIS : Enterprise Resource Planning (ERP), transaction central de traitement (TP), bases de données et systèmes de messagerie.

JCA 1.6 fournit des fonctionnalités de gestion :

- connexions
- transactions
- sécurité
- cycle de vie
- Instances de travail
- Flux interne de transactions
- Flux interne de messages

JCA 1.6 a été développé en tant que Java Community Process JSR-322, <http://jcp.org/en/jsr/detail?id=313>.

Autre pool de connexions gérées

JBoss EAP 6.4 comprend les autres implémentations de pool suivantes :

- `org.jboss.jca.core.connectionmanager.pool.mcp.SemaphoreArrayListManagedConnectionPool`: C'est le pool de connexion par défaut.
- `org.jboss.jca.core.connectionmanager.pool.mcp.SemaphoreConcurrentLinkedQueueManagedConnectionPool`: Ce pool de connexions fournit parfois un meilleur profil de performance et est activé par l'intermédiaire de la propriété -
`Dironjacamar.mcp=org.jboss.jca.core.connectionmanager.pool.mcp.SemaphoreConcurrentLinkedQueueManagedConnectionPool`

- `org.jboss.jca.core.connectionmanager.pool.mcp.LeakDumperManagedConnectionPool`: Ce pool de connexion n'est utilisé qu'à but de débogage. Pour obtenir davantage d'informations sur le `LeakDetectorPool`, consulter www.ironjacamar.org/doc/userguide/1.2/en-US/html/ch04.html#configuration_ironjacamar_leakpool

[Rapporter un bogue](#)

22.1.3. Adaptateurs de ressources

Un adaptateur de ressources est un composant Java EE déployable qui permet la communication entre une application Java EE et une entreprise d'informations système (EIS) à l'aide de la spécification Java Connector Architecture (JCA). Un adaptateur de ressources est souvent fourni par les fournisseurs de l'EIS pour permettre une intégration facile de leurs produits aux applications Java EE.

Un système d'information Enterprise peut être n'importe quel autre système de logiciel au sein d'une organisation. Les exemples incluent les systèmes ERP (Enterprise Resource Planning), les systèmes de base de données, les serveurs d'e-mails et les systèmes de messagerie propriétaires.

Un adaptateur de ressources est emballé dans un fichier de Ressources Adaptateur Archive (RAR) qui peut être déployé dans JBoss EAP 6. Un fichier RAR peut également être inclus dans un déploiement Enterprise Archive (EAR).

[Rapporter un bogue](#)

22.2. CONFIGURATION DU SOUS-SYSTÈME JAVA CONNECTOR ARCHITECTURE (JCA)

Le sous-système JCA du fichier de configuration de JBoss EAP 6 contrôle les paramètres de configuration généraux du conteneur JCA et déploiements d'adaptateurs de ressources.

Éléments clés du sous-système JCA

Archive validation

- Ce paramétrage indique si la validation d'archivage doit avoir lieu sur les unités de déploiement.
- Le tableau suivant décrit les attributs que vous pouvez définir pour la validation d'archivage.

Tableau 22.1. Attributs du paramètre de validation d'archivage

Attribut	Valeur par défaut	Description
<code>enabled</code>	<code>true</code>	Indique si la validation d'archivage est activée.
<code>fail-on-error</code>	<code>true</code>	Indique si un rapport d'erreur de validation d'archivage a fait échouer le développement.
<code>fail-on-warn</code>	<code>false</code>	Indique si un rapport d'avertissement de validation d'archivage a fait échouer le développement.

- Si une archive n'implémente pas la spécification Java EE Connector Architecture

correctement, et que la validation d'archivage est activée, un message d'erreur s'affichera pendant le déploiement pour décrire le problème, comme par exemple :

```
Severity: ERROR
Section: 19.4.2
Description: A ResourceAdapter must implement a "public int hashCode()" method.
Code: com.mycompany.myproject.ResourceAdapterImpl

Severity: ERROR
Section: 19.4.2
Description: A ResourceAdapter must implement a "public boolean equals(Object)" method.
Code: com.mycompany.myproject.ResourceAdapterImpl
```

- Si la validation d'archivage n'est pas spécifiée, on la considérera comme présente et l'attribut **enabled** aura comme valeur true par défaut.

Bean validation

- Ce paramètre indique si la validation de bean (JSR-303) aura lieu sur les unités de déploiement.
- Le tableau ci-dessous décrit les attributs que vous pouvez déterminer pour la validation de bean.

Tableau 22.2. Attributs du paramètre de validation de bean

Attribut	Valeur par défaut	Description
enabled	true	Indique si la validation de bean est activée.

- Si la validation de bean n'est pas spécifiée, on la considérera comme présente et l'attribut **enabled** aura comme valeur true par défaut.

Work Managers

- Il y a deux types de Work Managers :

Work Manager par défaut

Le Work Manager par défaut et ses Thread Pools.

Work Manager personnalisé

Une définition de Work Manager et ses Thread Pools.

- Le tableau suivant décrit les attributs que vous pouvez définir pour les Work Managers.

Tableau 22.3. Attributs de Work Managers

Attribut	Description
name	Indique le nom du Work Manager. Requis pour les Work Managers personnalisés.
short-running-threads	Thread Pool pour les instances Work standards. Chaque Work Manager a un Thread Pool à exécution courte.
long-running-threads	Les instances Work de Thread pool de JCA 1.6 qui définissent LONG_RUNNING . Chaque Work Manager peut avoir un Thread pool de longue durée en option.

- Le tableau ci-dessous décrit les attributs que vous pouvez définir pour les Thread pools de Work Managers.

Tableau 22.4. Attributs de Thread pool

Attribut	Description
allow-core-timeout	Paramètre booléen qui détermine quels threads principaux risquent d'expirer. La valeur par défaut est false.
core-threads	La taille du pool de threads. Doit être inférieure ou égale à la taille de pool de threads maximum.
queue-length	La longueur maximum de la file d'attente.
max-thread	Taille de pool de threads maximum.
keepalive-time	Indique le durée pendant laquelle les threads de pool doivent être conservés après avoir complété leur tâche.
thread-factory	Référence à la fabrique de threads.

Bootstrap Contexts

- Utilisé pour définir les contextes de bootstrapping (démarrage) personnalisés.
- Le tableau suivant décrit les attributs à définir pour les contextes de bootstrapping.

Tableau 22.5. Attributs de contexte de bootstrapping

Attribut	Description
name	Indique le nom du contexte de bootstrapping
workmanager	Indique le nom du Work Manager à utiliser dans ce contexte.

Cached connection manager

- Utilisé pour déboguer les connexions et pour supporter l'inscription tardive d'une connexion dans une transaction, pour vérifier leur bonne utilisation et fonctionnement.
- Le tableau suivant décrit les attributs que vous pouvez définir pour le manager de connexions mis en cache.

Tableau 22.6. Attributs du paramètre manager de connexion mis en cache

Attribut	Valeur par défaut	Description
debug	false	Sorties avertissement en cas d'échec de fermeture explicite des connexions
error	false	Envoie une exception en cas d'échec de fermeture explicite des connexions.

Procédure 22.1. Configurer le sous-système JCA par la console de management

Le sous-système JCA de JBoss EAP 6 peut être configuré dans la console de gestion. Les options de configuration de JCA sont situées dans des endroits légèrement différents dans la console de gestion, selon la façon dont le serveur est exécuté.

1. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Connector** et sélectionner **JCA**.
2. Si le serveur exécute en mode de domaine, sélectionner un profil à partir du menu déroulant **Profile** en haut et à gauche.
3. Configurer les paramètres du sous-système JCA à l'aide des trois onglets.
 - a. **Config courante**
L'onglet **Common Config** contient des paramètres pour le gestionnaire de connexions en cache, la validation de l'archive et la validation de bean (JSR-303). Chacun d'entre eux est contenu dans son onglet propre. Ces réglages peuvent être changés en ouvrant l'onglet correspondant, en cliquant sur le bouton **Edit**, en effectuant les changements nécessaires et puis en cliquant sur le bouton **Save** de sauvegarde.

RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 6.3.0.Alpha3 Messages: 0 admin

Home Configuration Domain Runtime Administration

Profile: full

Subsystems

- Connector
 - JCA**
 - Datasources
 - Resource Adapters
 - Mail
- Container
- Core
- Infinispan
- Messaging
- Security
- Web

General Configuration

COMMON CONFIG BOOTSTRAP CONTEXTS WORK MANAGER

JCA Subsystem

The Java EE Connector Architecture (JCA) subsystem providing general configuration for resource adapters.

[Connection Manager](#) [Archive Validation](#) [Bean Validation](#)

[Need Help?](#)

[Edit](#)

Is Enabled?: true

Figure 22.1. Configuration commune JCA

b. Work Managers

L'onglet **Work Manager** contient la liste des Work Managers (gestionnaires de travail) configurés. Les nouveaux Work Managers peuvent être ajoutés, supprimés, et leurs pools de threads configurés ici. Chaque Work Manager peut avoir un pool de threads à exécution courte et un pool de threads de longue durée en option.

RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 6.3.0.Alpha3 Messages: 0 admin

Home Configuration Domain Runtime Administration

Profile: full

Subsystems

- Connector
 - JCA**
 - Datasources
 - Resource Adapters
 - Mail
- Container
- Core
- Infinispan
- Messaging
- Security
- Web

General Configuration

COMMON CONFIG BOOTSTRAP CONTEXTS WORK MANAGER

JCA Workmanager

Work manager for resource adapters.

Available Work Manager

[Add](#) [Remove](#)

Name	Option
default	View >

« < 1-1 of 1 > »

Figure 22.2. Work Managers

Les attributs de thread pool peuvent être configurés en cliquant sur **View** sur l'adaptateur de ressources sélectionnées.

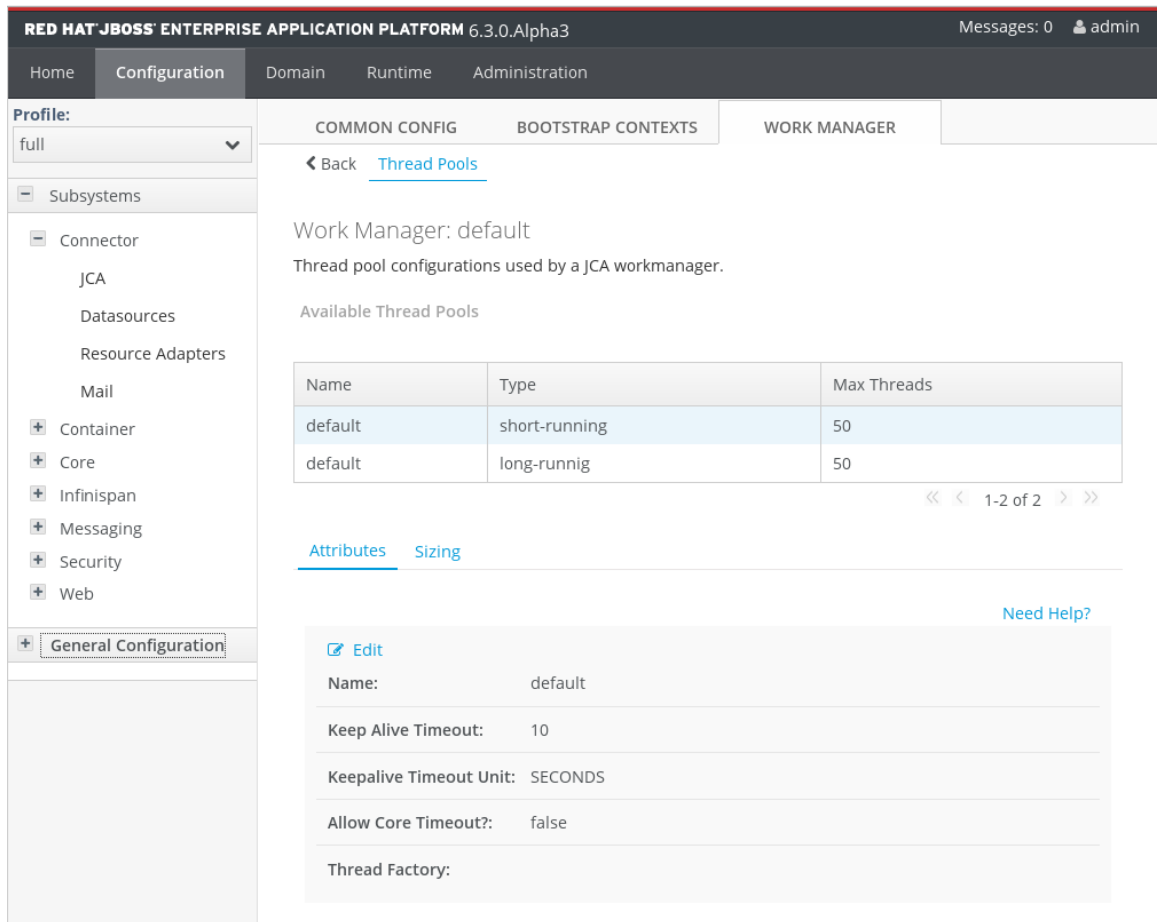


Figure 22.3. Work Manager Thread Pools

c. Bootstrap Contexts

L'onglet **Bootstrap Contexts** contient la liste des contextes d'amorçage configurés. De nouveaux objets de contexte d'amorçage peuvent être ajoutés, supprimés ou configurés. Un Work Manager doit être assigné à chaque contexte de Bootstrap.

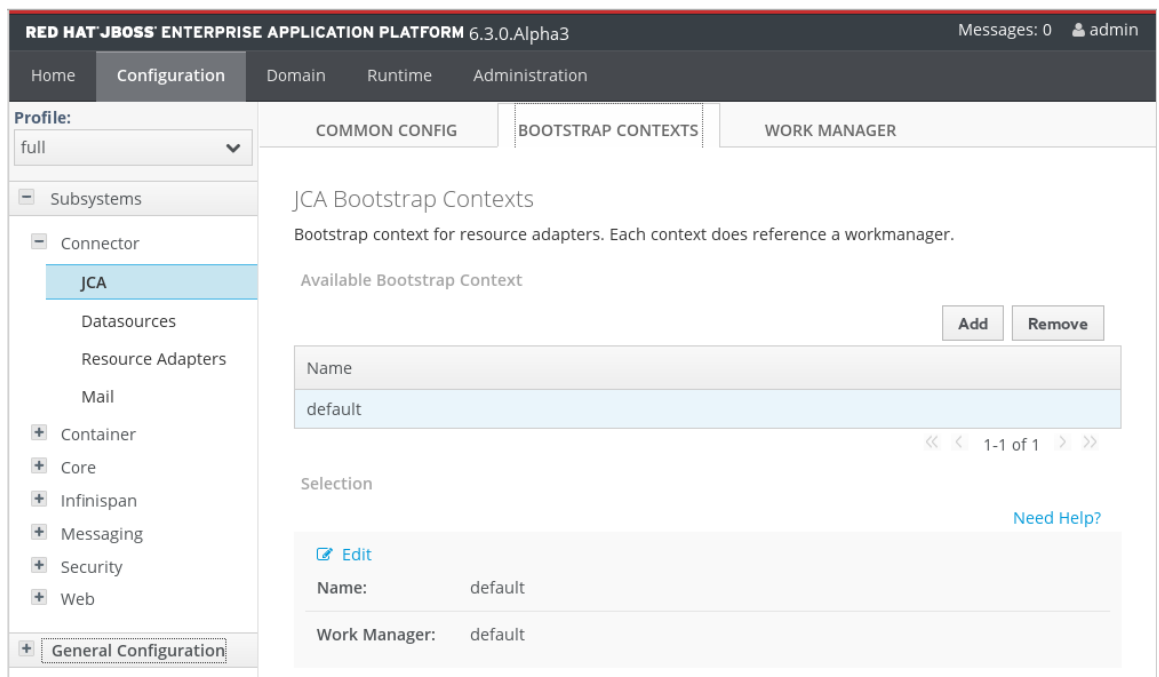


Figure 22.4. Bootstrap Contexts

22.3. DÉPLOYER UN ADAPTATEUR DE RESSOURCES

Les adaptateurs de ressources peuvent être déployés dans JBoss EAP 6 à l'aide de l'interface CLI, de la console de management basée-web, ou en copiant manuellement les fichiers. Le processus est le même que celui d'autres artefacts déployables.

Procédure 22.2. Déployer un adaptateur de ressources par l'interface CLI

1. Ouvrir une invite de commande de votre système d'exploitation.
2. Connectez-vous à l'interface CLI.
 - o Dans Linux, saisir ce qui suit au niveau de la ligne de commande :

```
$ EAP_HOME/bin/jboss-cli.sh --connect
$ Connected to standalone controller at localhost:9999
```

- o Dans Windows, saisir ce qui suit au niveau de la ligne de commande :

```
C:\>EAP_HOME\bin\jboss-cli.bat --connect
C:\> Connected to standalone controller at localhost:9999
```

3. Déployer l'adaptateur de ressources.
 - o Pour déployer l'adaptateur de ressources dans un serveur autonome, saisir ce qui suit dans une ligne de commande :

```
$ deploy path/to/resource-adapter-name.rar
```

- o Pour déployer l'adaptateur de ressources dans tous les serveurs d'un domaine géré, saisir ce qui suit dans une ligne de commande :

```
$ deploy path/to/resource-adapter-name.rar --all-server-groups
```

Procédure 22.3. Déployer un adaptateur de ressources par la console de gestion

1. Connectez-vous à la console de gestion. Voir [Section 3.4.2, « Se connecter à la console de gestion »](#).
2. Cliquer sur l'onglet **Runtime** qui se trouve en haut de l'écran. Sélectionner **Manage Deployments**. Cliquer sur **Add**.
3. Naviguer dans l'archive d'adaptateur de ressources et le sélectionner. Puis, cliquer sur le bouton **Next**.
4. Vérifier les noms de déploiement, puis cliquer sur le bouton **Save**.
5. L'archive d'adaptateur de ressources doit maintenant apparaître dans la liste dans un état de désactivation.
6. Activer l'adaptateur de ressources.

- Dans le mode de domaine, cliquer sur **Assign**. Sélectionner à quels groupes de serveurs il faut assigner l'adaptateur de ressources. Cliquer sur **Save** pour terminer.
- En mode autonome, sélectionner le composant d'application de la liste. Cliquer sur **En/Disable**. Cliquer sur **Confirm** après **Are You Sure?** pour activer le composant.

Procédure 22.4. Déployer un adaptateur de ressources manuellement

- Copier l'archive d'adaptateur de ressources dans le répertoire des déploiements de serveur,
 - Pour un serveur autonome, copier l'archive d'adaptateur de ressources dans le répertoire `EAP_HOME/standalone/deployments/`.
 - Dans un domaine géré, vous devez utiliser la console de gestion ou le CLI pour déployer l'archive d'adaptateur de ressources dans les groupes de serveurs.

[Rapporter un bogue](#)

22.4. CONFIGURATION D'UN ADAPTATEUR DE RESSOURCES DÉPLOYÉES

Les administrateurs JBoss peuvent configurer les adaptateurs de ressources pour JBoss EAP 6 à l'aide de l'interface CLI, de la console de management basée-web, ou en modifiant manuellement la configuration des fichiers.

Voir le document du fournisseur pour votre adaptateur de ressources pour obtenir des informations sur les propriétés prises en charge et autres informations.



NOTE

Dans la procédure suivante, la ligne de commande que vous devez saisir suit l'invite suivante `[standalone@localhost:9999 /]`. Ne pas saisir le texte qui se trouve à l'intérieur des accolades. Voici la sortie que vous devriez apercevoir comme résultat, ainsi, `{"outcome" => "success"}`.

Procédure 22.5. Configurer un adaptateur de ressources par l'interface CLI

1. Ouvrir une invite de commande de votre système d'exploitation.
2. Connectez-vous à l'interface CLI.
 - Dans Linux, saisir ce qui suit au niveau de la ligne de commande :

```
$ EAP_HOME/bin/jboss-cli.sh --connect
```

Vous devriez voir le résultat de sortie suivant :

```
$ Connected to standalone controller at localhost:9999
```

- Dans Windows, saisir ce qui suit au niveau de la ligne de commande :

```
C:\>EAP_HOME\bin\jboss-cli.bat --connect
```

Vous devriez voir le résultat de sortie suivant :

```
C:\> Connected to standalone controller at localhost:9999
```

3. Ajouter la configuration d'adaptateur de ressource.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-adapter=eis.rar:add(archive=eis.rar, transaction-support=XATransaction) {"outcome" => "success"}
```

4. Configurer la <config-property> **server** niveau adaptateur de ressources.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-adapter=eis.rar/config-properties=server/:add(value=localhost) {"outcome" => "success"}
```

5. Configurer la <config-property> **port** niveau adaptateur de ressources

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-adapter=eis.rar/config-properties=port/:add(value=9000) {"outcome" => "success"}
```

6. Ajouter une définition de connexion à la fabrique de connexions gérées.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-adapter=eis.rar/connection-definitions=cfName:add(class-name=com.acme.eis.ra.EISManagedConnectionFactory, jndi-name=java:/eis/AcmeConnectionFactory) {"outcome" => "success"}
```

7. Configurer <config-property> **port** niveau usine de connexions gérées.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-adapter=eis.rar/connection-definitions=cfName/config-properties=name/:add(value=Acme Inc) {"outcome" => "success"}
```

8. Ajouter un objet admin.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-adapter=eis.rar/admin-objects=aoName:add(class-name=com.acme.eis.ra.EISAdminObjectImpl, jndi-name=java:/eis/AcmeAdminObject) {"outcome" => "success"}
```

9. Configurer la propriété **threshold** de l'objet admin.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-adapter=eis.rar/admin-objects=aoName/config-properties=threshold/:add(value=10) {"outcome" => "success"}
```

10. Activer l'adaptateur de ressource.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-adapter=eis.rar:activate
{"outcome" => "success"}
```

11. Voir l'adaptateur de ressources nouvellement configuré et activé.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-adapter=eis.rar:read-resource(recursive=true)
{
  "outcome" => "success",
  "result" => {
    "archive" => "eis.rar",
    "beanvalidationgroups" => undefined,
    "bootstrap-context" => undefined,
    "transaction-support" => "XATransaction",
    "admin-objects" => {"aoName" => {
      "class-name" => "com.acme.eis.ra.EISAdminObjectImpl",
      "enabled" => true,
      "jndi-name" => "java:/eis/AcmeAdminObject",
      "use-java-context" => true,
      "config-properties" => {"threshold" => {"value" => 10}}
    }},
    "config-properties" => {
      "server" => {"value" => "localhost"},
      "port" => {"value" => 9000}
    },
    "connection-definitions" => {"cfName" => {
      "allocation-retry" => undefined,
      "allocation-retry-wait-millis" => undefined,
      "background-validation" => false,
      "background-validation-millis" => undefined,
      "blocking-timeout-wait-millis" => undefined,
      "class-name" =>
"com.acme.eis.ra.EISManagedConnectionFactory",
      "enabled" => true,
      "flush-strategy" => "FailingConnectionOnly",
      "idle-timeout-minutes" => undefined,
      "interleaving" => false,
      "jndi-name" => "java:/eis/AcmeConnectionFactory",
      "max-pool-size" => 20,
      "min-pool-size" => 0,
      "no-recovery" => undefined,
      "no-tx-separate-pool" => false,
      "pad-xid" => false,
      "pool-prefill" => false,
      "pool-use-strict-min" => false,
      "recovery-password" => undefined,
      "recovery-plugin-class-name" => undefined,
      "recovery-plugin-properties" => undefined,
      "recovery-security-domain" => undefined,
      "recovery-username" => undefined,
      "same-rm-override" => undefined,

```



```

        "security-application" => undefined,
        "security-domain" => undefined,
        "security-domain-and-application" => undefined,
        "use-ccm" => true,
        "use-fast-fail" => false,
        "use-java-context" => true,
        "use-try-lock" => undefined,
        "wrap-xa-resource" => true,
        "xa-resource-timeout" => undefined,
        "config-properties" => {"name" => {"value" => "Acme
Inc"}}}
    }}
}
}

```

Procédure 22.6. Configurer un adaptateur de ressources par la console de management basée-web

1. Connectez-vous à la console de gestion. Voir [Section 3.4.2, « Se connecter à la console de gestion »](#).
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Connectors** et sélectionner **Resource Adapters**.
 - a. En mode de domaine, sélectionner **Profile** à partir du menu déroulant qui se trouve en haut et à gauche.

Cliquer sur **Add**.
3. Saisir le nom de l'archive et choisir le type de transaction **XATransaction** à partir du menu déroulant **TX**:. Ensuite, cliquer sur **Save**.
4. Sélectionner l'onglet **Properties**. Cliquer sur **Add**.
5. Saisir le **serveur** pour le **Name** (nom) et le nom d'hôte, par exemple **localhost**, pour la valeur **Value**. Puis cliquer sur **Save** pour terminer.
6. Cliquer sur **Add** à nouveau. Saisir le **port** pour le **Name** (nom) et le nom de port, par exemple **9000**, pour la valeur **Value**. Puis cliquer sur **Save** pour terminer.
7. Les propriétés **server** et **port** apparaissent maintenant dans le panneau **Properties**. Cliquer sur le lien **View** (Vue) sous la colonne **Option** pour l'adaptateur de ressources listées pour visualiser les définitions de connexion or **Connection Definitions**.
8. Cliquer sur **Add** qui se trouve au dessus du tableau **Available Connection Definitions** pour ajouter une définition de connexion.
9. Saisir le **JNDI Name** et le nom de classe complet de la **Connection Class**. Puis cliquer sur **Save** pour terminer.
10. Sélectionner la nouvelle définition de connexion, puis sélectionner l'onglet **Properties**. Cliquer sur le bouton **Add** pour saisir les données de **Key** et **Value** pour cette définition de connexion. Cliquer sur **Save** pour terminer.
11. La définition de connexion est terminée, mais non activée. Sélectionner la définition de connexion et cliquer sur le bouton **Enable** pour activer la définition de connexion.

12. Un dialogue vous demande **Souhaitez-vous réellement modifier la définition de connexion?** du nom JNDI. Cliquer sur **Confirm**. La définition de connexion devrait maintenant afficher **Enabled** (activée).
13. Cliquer sur l'onglet **Admin Objects** qui se trouve dans la partie supérieure de la page pour créer et configurer des objets admin. Puis, cliquer sur **Add**.
14. Saisir le **JNDI Name** et le nom de classe **Class Name** complet de l'objet admin. Puis cliquer sur **Save**.
15. Sélectionner l'onglet **Properties**, puis cliquer sur **Add** pour ajouter des propriétés d'objet admin.
16. Saisir une propriété de configuration d'objet admin, comme par exemple la limite **threshold**, dans le champ **Name** (nom). Saisir la valeur de la propriété de configuration, comme par exemple **10**, pour la valeur **Value**. Puis cliquer sur **Save** pour sauvegarder la propriété.
17. L'objet admin est maintenant complété, mais non actif. Cliquer sur **Enable** pour activer l'objet admin.
18. Un dialogue vous demande **Souhaitez-vous réellement modifier l'Objet admin?** du nom JNDI. Cliquer sur **Confirm**. L'objet admin devrait maintenant afficher **Enabled** (activé).
19. Vous devez charger à nouveau la configuration du serveur pour terminer ce processus. Cliquer sur le lien **Runtime**. Étendre le menu **Server**. Sélectionner **Overview** dans le panneau de navigation de gauche.
 - a. Charger à nouveau les serveurs
 - En mode de domaine, faire glisser le curseur sur le groupe de serveurs. Sélectionner **Restart Group**.
 - En mode standalone, il y aura un bouton **Reload** disponible. Cliquer sur **Reload**.
20. Un dialogue vous demande **Souhaitez-vous charger à nouveau la configuration du serveur ?** pour le serveur indiqué. Cliquer sur **Confirm**. La configuration du serveur sera à jour.

Procédure 22.7. Configurer un adaptateur de ressources manuellement

1. Stopper le serveur JBoss EAP 6.



IMPORTANT

Vous devez interrompre le serveur avant de modifier le fichier de configuration du serveur pour que votre changement puisse être persisté au redémarrage du serveur.

2. Ouvrir le fichier de configuration du serveur pour l'édition.
 - o Pour les serveurs autonomes, il s'agit du fichier **EAP_HOME/standalone/configuration/standalone.xml**.
 - o Si vous exécutez dans un domaine géré, il s'agira du fichier

`EAP_HOME/domain/configuration/domain.xml`.

3. Chercher le sous-système `urn:jboss:domain:resource-adapters` dans le fichier de configuration.
4. Il n'y a pas d'adaptateur de ressources défini pour ce système. Veuillez commencer par remplacer :

```
<subsystem xmlns="urn:jboss:domain:resource-adapters:1.1"/>
```

par ceci :

```
<subsystem xmlns="urn:jboss:domain:resource-adapters:1.1">
  <resource-adapters>
    <!-- <resource-adapter> configuration listed below -->
  </resource-adapters>
</subsystem>
```

5. Remplacer la configuration `<!-- <resource-adapter> listée ci-dessous -->` par la définition XML de l'adaptateur de ressources. Ce qui suit représente la représentation XML de la configuration de l'adaptateur de ressources créé par l'interface CLI et la console de management basée-web décrite ci-dessus.

```
<resource-adapter>
  <archive>
    eis.rar
  </archive>
  <transaction-support>XATransaction</transaction-support>
  <config-property name="server">
    localhost
  </config-property>
  <config-property name="port">
    9000
  </config-property>
  <connection-definitions>
    <connection-definition class-
name="com.acme.eis.ra.EISManagedConnectionFactory"
      jndi-name="java:/eis/AcmeConnectionFactory"
      pool-name="java:/eis/AcmeConnectionFactory">
      <config-property name="name">
        Acme Inc
      </config-property>
    </connection-definition>
  </connection-definitions>
  <admin-objects>
    <admin-object class-
name="com.acme.eis.ra.EISAdminObjectImpl"
      jndi-name="java:/eis/AcmeAdminObject"
      pool-name="java:/eis/AcmeAdminObject">
```

```

        <config-property name="threshold">
            10
        </config-property>
    </admin-object>
</admin-objects>
</resource-adapter>

```

6. Démarrer le serveur

Lancer à nouveau le serveur JBoss EAP 6 pour qu'il commence à exécuter avec la nouvelle configuration.

[Rapporter un bogue](#)

22.5. RÉFÉRENCE DE DESCRIPTION D'ADAPTATEUR DE RESSOURCES

Les tableaux suivants décrivent les éléments de description d'adaptateurs de ressources.

Tableau 22.7. Éléments principaux

Élément	Description
bean-validation-groups	Indique le groupe de validation du bean qui doit être utilisé
bootstrap-context	Indique le nom unique du contexte de bootstrapping qui doit être utilisé
config-property	Config-property spécifie les propriétés de configuration de l'adaptateur de ressources.
transaction-support	Indique le type de transactions pris en charge par l'adaptateur de ressources. La valeurs valides sont : NoTransaction , LocalTransaction , XATransaction
connection-definitions	Indique les définitions de connexion
admin-objects	Indique les objets d'administration

Tableau 22.8. Éléments de groupes de validation de beans

Élément	Description
bean-validation-group	Indique le nom de classe complet d'un groupe de validation de beans devant être utilisé pour la validation.

Tableau 22.9. Définition de connexion / attributs d'objets admin

Attribut	Description
class-name	Indique le nom de classe complet d'une usine de connexions gérée ou d'un objet admin
jndi-name	Indique le nom JNDI
enabled	L'objet doit-il être activé ?
use-java-context	Indique si on doit utiliser un contexte java:/ JNDI
pool-name	Indique le nom de pool de l'objet
use-ccm	Active le gestionnaire de connexion mis en cache

Tableau 22.10. Éléments de définition de connexion

Élément	Description
config-property	Config-property spécifie les propriétés de configuration de l'usine de connexions.
pool	Indique les paramètres de pooling
xa-pool	Indique les paramètres de pooling XA
security	Indique les paramètres de sécurité
timeout	Indique les paramètres de timeout
validation	Indique les paramètres de validation
recovery	Indique les paramètres de recouvrement XA

Tableau 22.11. Éléments de pooling

Élément	Description
min-pool-size	L'élément min-pool-size indique le nombre minimal de connexions qu'un pool peut contenir. Celles-ci ne sont pas créées tant que l'on ne connaît pas le sujet de la demande de connexion. La valeur par défaut est 0
max-pool-size	L'élément max-pool-size indique le nombre maximal de connexions d'un pool. On ne pourra pas créer plus de connexions que ce nombre indiqué pour chaque sub-pool. Cette valeur par défaut est à 20 .

Élément	Description
prefill	Indique si l'on doit essayer de pré-remplir le pool de connexion. La valeur par défaut est false .
use-strict-min	Indique si la min-pool-size doit être considérée sérieusement. La valeur par défaut est false .
flush-strategy	Indique comment le pool doit être vidé en cas d'erreur. Les valeurs valides sont : FailingConnectionOnly (default), IdleConnections , EntirePool

Tableau 22.12. Éléments de pool XA

Élément	Description
min-pool-size	L'élément min-pool-size indique le nombre minimal de connexions qu'un pool peut contenir. Celles-ci ne sont pas créées tant que l'on ne connaît pas le sujet de la demande de connexion. Cette valeur par défaut à 0
max-pool-size	L'élément max-pool-size indique le nombre maximal de connexions d'un pool. On ne pourra pas créer plus de connexions que ce nombre indiqué pour chaque sub-pool. Cette valeur par défaut est à 20 .
prefill	Indique si l'on doit essayer de pré-remplir le pool de connexion. La valeur par défaut est false .
use-strict-min	Indique si la min-pool-size doit être considérée sérieusement. La valeur par défaut est false .
flush-strategy	Indique comment le pool doit être vidé en cas d'erreur. Les valeurs valides sont : FailingConnectionOnly (default), IdleConnections , EntirePool
is-same-rm-override	L'élément is-same-rm-override element permet de définir inconditionnellement si <code>javax.transaction.xa.XAResource.isSameRM(XAResource)</code> doit renvoyer true or false
interleaving	Élément qui permet ou non l'entrelacement des usines de connexions XA
no-tx-separate-pools	Oracle n'aime pas que les connexions XA soient utilisées à la fois à l'intérieur et à l'extérieur d'une connexion JTA. Pour résoudre ce problème, vous pourrez créer des sub-pools pour ces contextes différents.

Élément	Description
pad-xid	Est-ce que le Xid doit être mis en tampon ?
wrap-xa-resource	Est-ce que les instances XAResource doivent être encapsulées dans une instance org.jboss.tm.XAResourceWrapper

Tableau 22.13. Éléments de sécurité

Élément	Description
application	Indique si les paramètres de sécurité fournis, (comme par exemple, à partir de <code>getConnection(user, pw)</code>), sont utilisés pour distinguer les connexions d'un pool.
security-domain	Indique si des sujets (de domaine de sécurité) sont utilisés pour distinguer les connexions d'un pool. Le contenu du domaine de sécurité correspond au nom du gestionnaire de sécurité JAAS qui gère l'authentification. Ce nom est en corrélation avec l'attribut <code>application-policy/name</code> du descripteur JAAS <code>login-config.xml</code> .
security-domain-and-application	Indique que les paramètres de l'application fournis (par exemple, à partir de <code>getConnection(user, pw)</code>) ou que le sujet (du domaine de la sécurité) soient utilisés pour distinguer les connexions du pool. Le contenu du domaine de sécurité est le nom du gestionnaire de sécurité JAAS qui gère l'authentification. Ce nom est en corrélation avec l'attribut <code>application-policy/name</code> du descripteur JAAS <code>login-config.xml</code> .

Tableau 22.14. Éléments de timeout

Élément	Description
blocking-timeout-millis	L'élément « <code>blocking-timeout-millis</code> » indique la durée maximale en millisecondes de blocage pendant que vous attendez une connexion, avant de lever une exception. Notez que cela bloque uniquement pendant que vous attendez un permis de connexion, et ne soulèvera pas d'exception si la création d'une nouvelle connexion prend un temps excessivement long. La valeur par défaut est 30000 (30 secondes).
idle-timeout-minutes	Les éléments <code>idle-timeout-minutes</code> indiquent la durée maximum, en minutes, avant qu'une connexion inutile puisse être fermée. La durée maximum dépend du temps de balayage de l' <code>idleRemove</code> , qui correspond à la moitié du temps « <code>idle-timeout-minutes</code> » le plus petit de n'importe quel pool.

Élément	Description
<code>allocation-retry</code>	Cet élément de tentative d'allocation indique le nombre de fois que l'on doit allouer un connexion avant de lancer une exception. La valeur par défaut est <code>0</code> .
<code>allocation-retry-wait-millis</code>	Le temps, en millisecondes, qu'il faut attendre avant de retenter d'allouer une connexion. La valeur par défaut est <code>5000</code> , soit 5 secondes.
<code>xa-resource-timeout</code>	Passé à <code>XAResource.setTimeout()</code> . La valeur par défaut est 0 sans invoquer le setter. Indiqué en secondes.

Tableau 22.15. Éléments de validation

Élément	Description
<code>background-validation</code>	Élément pour spécifier que les connexions doivent être validées en arrière-plan plutôt qu'avant utilisation
<code>background-validation-minutes</code>	L'élément « <code>background-validation-minutes</code> » indique la durée, en minutes, d'exécution de la validation d'arrière-plan.
<code>use-fast-fail</code>	Indique s'il y a échec d'allocation de connexion à la première connexion si invalide (<code>true</code>) ou s'il y a de nouvelles tentatives jusqu'à ce que le pool soit épuisé de tout essai de connexion possible (<code>false</code>). La valeur par défaut est <code>false</code>

Tableau 22.16. Éléments d'objets admin

Élément	Description
<code>config-property</code>	Spécifie une propriété de configuration d'objet d'administration.

Tableau 22.17. Éléments de recouvrement

Élément	Description
<code>recover-credential</code>	Indique la paire nom / mot de passe ou le domaine de sécurité qui doit être utilisé pour le recouvrement.
<code>recover-plugin</code>	Spécifie l'implémentation de <code>org.jboss.jca.core.spi.recovery.RecoveryPlugin</code> class.

Les schéma de déploiement sont définis dans `jboss-as-resource-adapters_1_0.xsd` et http://www.ironjacamar.org/doc/schema/ironjacamar_1_0.xsd pour l'activation automatique.

[Rapporter un bogue](#)

22.6. AFFICHAGES DES STATISTIQUES DE CONNEXION

Vous pouvez lire les statistiques d'une connexion définie à partir de la sous-arborescence `deployment=name.rar`.

Les statistiques sont définis à ce niveau et non pas au niveau `/subsystem` afin d'être accessibles à partir de tout rar non défini dans les configurations de fichiers `standalone.xml` ou `domain.xml`.

Par exemple :

Exemple 22.1.

```
/deployment=example.rar/subsystem=resource-  
adapters/statistics=statistics/connection-  
definitions=java\:\testMe:read-resource(include-runtime=true)
```



NOTE

Veillez à spécifier l'argument `include-runtime=true` car tous les statistiques sont en runtime uniquement et la valeur par défaut est `false`.

[Rapporter un bogue](#)

22.7. STATISTIQUES D'ADAPTATEUR DE RESSOURCES

Statistiques principaux

Le tableau suivant contient une liste de statistiques principaux d'adaptateurs de ressources pris en charge :

Tableau 22.18. Statistiques principaux

Nom	Description
ActiveCount	Le nombre de connexions actives. Chacune de ces connexions est soit utilisée par une application, soit disponible via pool
AvailableCount	Le nombre de connexions disponibles dans le pool
AverageBlockingTime	Le durée moyenne passée à bloquer l'obtention d'un verrou exclusif sur le pool. La valeur est en millisecondes.
AverageCreationTime	Le durée moyenne passée à créer une connexion. La valeur est en millisecondes.
CreatedCount	Le nombre de connexions créées.
DestroyedCount	Le nombre de connexions détruites.

Nom	Description
InUseCount	Le nombre de connexions actuellement utilisées.
MaxCreationTime	La durée maximum pour créer une connexion. La valeur est en millisecondes.
MaxUsedCount	Le nombre maximum de connexions utilisées
MaxWaitCount	Le nombre maximum de requêtes attendant une connexion en même temps.
MaxWaitTime	Le durée maximum à attendre un verrou exclusif sur le pool.
TimedOut	Le nombre de connexions expirées
TotalBlockingTime	Le durée à attendre un verrou exclusif sur le pool. La valeur est en millisecondes.
TotalCreationTime	La durée passée à créer des connexions. La valeur est en millisecondes.
WaitCount	Le nombre de requêtes en attente de connexion.

[Rapporter un bogue](#)

22.8. DÉPLOYER L'ADAPTATEUR DE RESSOURCES WEBSHERE MQ

Websphere MQ

WebSphere MQ est un logiciel de messagerie Oriented Middleware (MOM) d'IBM qui permet à des applications sur des systèmes distribués de communiquer entre eux. Ceci est accompli grâce à l'utilisation des messages et des files d'attente de messages. WebSphere MQ est chargé de remettre des messages à des files d'attente de messages et pour transférer des données à d'autres gestionnaires de file d'attente à l'aide de canaux de message. Pour plus d'informations sur WebSphere MQ, voir [WebSphere MQ](#).

Résumé

Cette section couvre les étapes à suivre pour déployer et configurer l'adaptateur de ressource Websphere MQ dans Red Hat JBoss Enterprise Application Platform 6. Cela peut se faire manuellement en modifiant les fichiers de configuration, par l'interface CLI, ou par la Console de gestion basée-web.



NOTE

Il y a un problème connu dans WebSphere MQ Resource Adapter version 7.5.0.3 et versions antérieures qui entraîne l'échec du processus de recouvrement périodique accompagné de l'exception XA qui laisse un message similaire à celui-ci dans le journal du serveur JBoss EAP :

```
WARN [com.arjuna.ats.jta] (Periodic Recovery) ARJUNA016027:
Local XARecoveryModule.xaRecovery got XA exception
XAException.XAER_INVALID: javax.transaction.xa.XAException: The
method 'xa_recover' has failed with errorCode '-5'.
```

Il y a un correctif dans la version 7.5.0.4. Vous trouverez une description détaillée de ce problème ici : <http://www-01.ibm.com/support/docview.wss?uid=swg11C97579>.

Sachez que WebSphere MQ 8.0, et versions supérieures, est non pris en charge dans EAP 6.x.

Conditions préalables

Avant de démarrer, vous devez vérifier votre version d'adaptateur de ressource WebSphere MQ et comprendre certaines propriétés de configuration de WebSphere MQ.

- L'adaptateur de ressources WebSphere MQ est fourni en tant que fichier RAR (Resource Archive) nommé `wmq.jmsra-VERSION.rar`. Vous devez utiliser la version 7.5.0. x. Voir la note sur la version requise.
- Vous devez connaître les valeurs des propriétés de configuration Websphere MQ suivantes. Voir la documentation de produit WebSphere MQ pour obtenir des détails sur ces propriétés.
 - `MQ.QUEUE.MANAGER`: le nom du gestionnaire de files d'attentes de WebSphere MQ
 - `MQ.HOST.NAME`: le nom d'hôte utilisé pour se connecter au gestionnaire de files d'attente de WebSphere MQ
 - `MQ.CHANNEL.NAME`: le canal de serveur utilisé pour se connecter au gestionnaire de files d'attente de WebSphere MQ
 - `MQ.QUEUE.NAME`: le nom de la file d'attente de destination
 - `MQ.TOPIC.NAME`: le nom du sujet de destination
 - `MQ.PORT`: le port utilisé pour se connecter au gestionnaire de files d'attente de WebSphere MQ
 - `MQ.CLIENT`: le type de transport
- Pour les connexions sortantes, vous devez vous familiariser avec la propriété de configuration suivante :
 - `MQ.CONNECTIONFACTORY.NAME`: le nom de l'instance d'usine de connexion qui fournit la connexion vers le système à distance.

**NOTE**

Voici les configurations par défaut fournies par IBM. Elles sont assujetties au changement. Veuillez vous référer à la documentation Websphere MQ pour plus d'informations.

Procédure 22.8. Déployer l'adaptateur de ressources manuellement

1. Si vous avez besoin d'un support de transactions avec l'adaptateur de ressources de WebSphereMQ, vous devrez re-paquager l'archive `wmq.jmsra-VERSION.rar` pour qu'elle inclue `mqetclient.jar`. Vous devrez utiliser la commande suivante :

```
[user@host ~]$ jar -uf wmq.jmsra-VERSION.rar mqetclient.jar
```

Soyez certain de remplacer `VERSION` par le numéro de version correct.

2. Copier le fichier `wmq.jmsra-VERSION.rar` dans le répertoire `EAP_HOME/standalone/deployments/`.
3. Ajouter l'adaptateur de ressources au fichier de configuration du serveur.
 - a. Ouvrir le fichier `EAP_HOME/standalone/configuration/standalone-full.xml` dans un éditeur.
 - b. Chercher le sous-système `urn:jboss:domain:resource-adapters` dans le fichier de configuration.
 - c. Il n'y a pas d'adaptateur de ressources défini pour ce système. Veuillez commencer par remplacer :

```
<subsystem xmlns="urn:jboss:domain:resource-adapters:1.1"/>
```

par ceci :

```
<subsystem xmlns="urn:jboss:domain:resource-adapters:1.1">
  <resource-adapters>
    <!-- <resource-adapter> configuration listed below -->
  </resource-adapters>
</subsystem>
```

- d. La configuration de l'adaptateur de ressources dépend de si vous avez besoin de la restauration et du support de transactions. Si vous n'avez pas besoin de support de transaction, choisissez la première étape de configuration ci-dessous. Si vous avez besoin de support de transaction, choisissez la deuxième étape de la configuration.
 - Pour les déploiements non transactionnels, veuillez remplacer la configuration `<!-- <resource-adapter> listée ci-dessous -->` par ce qui suit :

```
<resource-adapter>
  <archive>
    wmq.jmsra-VERSION.rar
  </archive>
  <transaction-support>NoTransaction</transaction-support>
  <connection-definitions>
    <connection-definition
```

```

        class-
name="com.ibm.mq.connector.outbound.ManagedConnectionFactoryIm
pl"
        jndi-
name="java:jboss/MQ.CONNECTIONFACTORY.NAME"
        pool-name="MQ.CONNECTIONFACTORY.NAME">
        <config-property name="hostName">
            MQ.HOST.NAME
        </config-property>
        <config-property name="port">
            MQ.PORT
        </config-property>
        <config-property name="channel">
            MQ.CHANNEL.NAME
        </config-property>
        <config-property name="transportType">
            MQ.CLIENT
        </config-property>
        <config-property name="queueManager">
            MQ.QUEUE.MANAGER
        </config-property>
        <security>
            <security-domain>MySecurityDomain</security-
domain>
        </security>
        </connection-definition>
    </connection-definitions>
    <admin-objects>
        <admin-object
            class-
name="com.ibm.mq.connector.outbound.MQQueueProxy"
            jndi-name="java:jboss/MQ.QUEUE.NAME"
            pool-name="MQ.QUEUE.NAME">
            <config-property name="baseQueueName">
                MQ.QUEUE.NAME
            </config-property>
            <config-property name="baseQueueManagerName">
                MQ.QUEUE.MANAGER
            </config-property>
        </admin-object>
        <admin-object class-
name="com.ibm.mq.connector.outbound.MQTopicProxy"
            jndi-name="java:jboss/MQ.TOPIC.NAME" pool-
name="MQ.TOPIC.NAME">
            <config-property name="baseTopicName">
                MQ.TOPIC.NAME
            </config-property>
            <config-property name="brokerPubQueueManager">
                MQ.QUEUE.MANAGER
            </config-property>
        </admin-object>
    </admin-objects>
</resource-adapter>

```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR.

- Pour les déploiements transactionnels, veuillez remplacer la configuration `<!-- <resource-adapter>` listée ci-dessous `-->` par ce qui suit :

```

<resource-adapter>
  <archive>
    wmq.jmsra-VERSION.rar
  </archive>
  <transaction-support>XATransaction</transaction-support>
  <connection-definitions>
    <connection-definition
      class-
name="com.ibm.mq.connector.outbound.ManagedConnectionFactoryIm
pl"
      jndi-
name="java:jboss/MQ.CONNECTIONFACTORY.NAME"
      pool-name="MQ.CONNECTIONFACTORY.NAME">
      <config-property name="hostName">
        MQ.HOST.NAME
      </config-property>
      <config-property name="port">
        MQ.PORT
      </config-property>
      <config-property name="channel">
        MQ.CHANNEL.NAME
      </config-property>
      <config-property name="transportType">
        MQ.CLIENT
      </config-property>
      <config-property name="queueManager">
        MQ.QUEUE.MANAGER
      </config-property>
    <security>
      <security-domain>MySecurityDomain</security-
domain>
    </security>
    <recovery>
      <recover-credential>
        <user-name>USER_NAME</user-name>
        <password>PASSWORD</password>
      </recover-credential>
    </recovery>
    </connection-definition>
  </connection-definitions>
  <admin-objects>
    <admin-object
      class-
name="com.ibm.mq.connector.outbound.MQQueueProxy"
      jndi-name="java:jboss/MQ.QUEUE.NAME"
      pool-name="MQ.QUEUE.NAME">
      <config-property name="baseQueueName">
        MQ.QUEUE.NAME
      </config-property>
      <config-property name="baseQueueManagerName">
        MQ.QUEUE.MANAGER
      </config-property>
    </admin-object>

```

```

        <admin-object class-
name="com.ibm.mq.connector.outbound.MQTopicProxy"
        jndi-name="java:jboss/MQ.TOPIC.NAME" pool-
name="MQ.TOPIC.NAME">
        <config-property name="baseTopicName">
            MQ.TOPIC.NAME
        </config-property>
        <config-property name="brokerPubQueueManager">
            MQ.QUEUE.MANAGER
        </config-property>
        </admin-object>
    </admin-objects>
</resource-adapter>

```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR. Vous devrez également remplacer *USER_NAME* et *PASSWORD* avec le nom et le mot de passe valides.



NOTE

Pour supporter les transactions, l'élément `<transaction-support>` a été défini à `XATransaction`. Pour supporter XA recovery, l'élément `<recovery>` a été ajouté à une définition de connexion.

- e. Si vous souhaitez changer le fournisseur par défaut en système de messagerie EJB3 dans JBoss EAP 6 de HornetQ vers WebSphere MQ, modifier le sous-système `urn:jboss:domain:ejb3:1.2` comme suit :

Remplacer :

```

<mdb>
    <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
    <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>

```

par :

```

<mdb>
    <resource-adapter-ref resource-adapter-name="wmq.jmsra-
VERSION.rar"/>
    <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>

```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR.

Procédure 22.9. Modifier le code MDB pour utiliser l'adaptateur de ressources

- Configurer `ActivationConfigProperty` et `ResourceAdapter` du code MDB comme suit :

```

@MessageDriven(name="WebSphereMQMDB", activationConfig = {
    @ActivationConfigProperty(propertyName =
"destinationType",propertyValue = "javax.jms.Queue"),
    @ActivationConfigProperty(propertyName = "useJNDI",

```

```

propertyValue = "false"),
    @ActivationConfigProperty(propertyName = "hostName",
propertyValue = "MQ.HOST.NAME"),
    @ActivationConfigProperty(propertyName = "port", propertyValue =
"MQ.PORT"),
    @ActivationConfigProperty(propertyName = "channel",
propertyValue = "MQ.CHANNEL.NAME"),
    @ActivationConfigProperty(propertyName = "queueManager",
propertyValue = "MQ.QUEUE.MANAGER"),
    @ActivationConfigProperty(propertyName = "destination",
propertyValue = "MQ.QUEUE.NAME"),
    @ActivationConfigProperty(propertyName = "transportType",
propertyValue = "MQ.CLIENT")
})

@ResourceAdapter(value = "wmq.jmsra-VERSION.rar")
@TransactionAttribute(TransactionAttributeType.NOT_SUPPORTED)
public class WebSphereMQMDB implements MessageListener {
}

```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR.

[Rapporter un bogue](#)

22.9. INSTALLER L'ADAPTATEUR DE RESSOURCES DE JBOSS ACTIVE MQ

Afin d'installer l'adaptateur de ressources JBoss Active MQ (A-MQ) dans JBoss EAP 6 pour qu'il puisse fonctionner avec JBoss Fuse A-MQ 6.1.0, suivre les étapes suivantes :

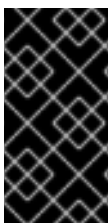
https://access.redhat.com/documentation/en-US/Red_Hat_JBoss_A-MQ/6.1/html/Integrating_with_JBoss_Enterprise_Application_Platform/DeployRar-InstallRar.html.

[Rapporter un bogue](#)

22.10. CONFIGURER UN ADAPTATEUR DE RESSOURCES JMS STANDARD À UTILISER AVEC UN FOURNISSEUR JMS DE TIERCE PARTIE

Résumé

JBoss EAP 6 peut être configuré pour fonctionner avec des fournisseurs tiers de JMS, cependant tous les fournisseurs JMS ne produisent pas un adaptateur de ressources JMS JCA pour l'intégration avec les plateformes d'applications Java. Cette procédure couvre les étapes requises pour configurer l'adaptateur de ressources JMS générique inclus dans JBoss EAP 6 pour se connecter à un fournisseur JMS. Dans cette procédure, Tibco EMS 6.3 est utilisé comme un exemple de fournisseur JMS, mais d'autres fournisseurs JMS peuvent avoir besoin d'une configuration différente.



IMPORTANT

Avant d'utiliser l'adaptateur de ressources JMS générique, vous devez d'abord vérifier auprès du fournisseur JMS pour savoir si ils ont leur propre adaptateur de ressources qui puisse être utilisé avec JBoss EAP 6. L'adaptateur de ressources JCA JMS ne doit être utilisé que lorsqu'un fournisseur JMS ne procure pas l'adaptateur qui lui est propre.

Conditions préalables

- Le serveur de l'adaptateur JMS est déjà configuré et prêt à l'utilisation. Il nous faudra tous les binaires utiles à l'implémentation JMS du fournisseur.
- Vous devrez également connaître les valeurs des propriétés suivantes du fournisseur JMS pour pouvoir chercher ses ressources JMS (usines de connexion, files d'attente, et topics)
 - `java.naming.factory.initial`
 - `java.naming.provider.url`
 - `java.naming.factory.url.pkgs`

Dans l'exemple XML utilisé dans cette procédure, ces paramètres sont notés **PROVIDER_FACTORY_INITIAL**, **PROVIDER_URL**, and **PROVIDER_CONNECTION_FACTORY** respectivement. Remplir ces espaces réservés avec les valeurs du fournisseur JMS pour votre environnement.

Procédure 22.10. Configurer un adaptateur de ressources JMS standard à utiliser avec un fournisseur JMS de tierce partie

1. Créer le module JBoss pour le fournisseur JMS

Créer un module JBoss qui puisse contenir toutes les bibliothèques requises pour se connecter et communiquer avec le fournisseur JMS. Ce module sera nommé `org.jboss.genericjms.provider`.

- a. Créer la structure de répertoire suivante :
`EAP_HOME/modules/system/layers/base/org/jboss/genericjms/provider/main`
- b. Copier les binaires requis pour l'implémentation JMS du fournisseur dans
`EAP_HOME/modules/system/layers/base/org/jboss/genericjms/provider/main`.



NOTE

Pour Tibco EMS, les binaires requis sont `tibjms.jar` et `tibcrypt.jar` du répertoire `lib` de l'installation Tibco.

- c. Créer un fichier `module.xml` dans
`EAP_HOME/modules/system/layers/base/org/jboss/genericjms/provider/main` comme ci-dessous, en énumérant les fichiers JAR des étapes précédentes comme ressources :

```
<module xmlns="urn:jboss:module:1.1"
name="org.jboss.genericjms.provider">
  <resources>
    <!-- all jars required by the JMS provider, in this case
Tibco -->
    <resource-root path="tibjms.jar"/>
    <resource-root path="tibcrypt.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
</module>
```

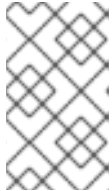
```

    <module name="javax.jms.api"/>
  </dependencies>
</module>

```

2. Configurer un contexte externe JNDI pour le fournisseur JMS

Les ressources JMS (usines de connexion et destinations) sont vérifiées par le fournisseur JMS. Nous ajouterons un contexte externe dans l'instance JBoss EAP 6 pour que toute recherche *locale* de cette ressource puisse être conduite automatiquement dans le fournisseur JMS distant.



NOTE

Dans cette procédure, ***EAP_HOME/standalone/configuration/standalone-full.xml*** est utilisé comme fichier de configuration JBoss EAP 6.

Dans ***EAP_HOME/standalone/configuration/standalone-full.xml***, sous **`<subsystem xmlns="urn:jboss:domain:naming:1.4">`**, ajouter :

```

<bindings>
  <external-context name="java:global/remoteJMS/"
    module="org.jboss.genericjms.provider"
    class="javax.naming.InitialContext">
    <environment>
      <property name="java.naming.factory.initial"
value="\${PROVIDER_FACTORY_INITIAL}"/>
      <property name="java.naming.provider.url"
value="\${PROVIDER_URL}"/>
      <property name="java.naming.factory.url.pkgs"
value="\${PROVIDER_URL_PKGS}"/>
    </environment>
    </external-context>
</bindings>

```

Les valeurs de ces trois propriétés doivent être remplacées par la bonne valeur pour pouvoir se connecter au fournisseur JMS distant. Veillez bien à garder `\{\}` intact quand vous remplissez l'espace réservé.

3. Activer la recherche par chaîne

Il y a des fournisseurs JMS (comme EMS Tibco) qui ne prennent pas la méthode **lookup(Name)** de JNDI. Dans ces cas, ajouter à la propriété **`org.jboss.as.naming.lookup.by.string`** la valeur **`true`** pour régler ce problème.

Exemple 22.2. Activer la recherche par chaîne dans EMS Tibco

Voici une définition complète d'un contexte externe dans EMS Tibco.

```

<bindings>
  <external-context name="java:global/remoteJMS/"
    module="org.jboss.genericjms.provider"
    class="javax.naming.InitialContext">
    <environment>
      <property name="java.naming.factory.initial"
value="com.tibco.tibjms.naming.TibjmsInitialContextFactory"/>

```

```

        <property name="java.naming.provider.url"
value="TIBCO_EMS_SERVER_HOST_NAME:PORT"/>
        <property name="java.naming.factory.url.pkgs"
value="com.tibco.tibjms.naming"/>
        <property name="org.jboss.as.naming.lookup.by.string"
value="true"/>
    </environment>
</external-context>
</bindings>

```

Dans ce contexte externe, toute recherche JNDI vers une ressource commençant par `java:global/remoteJMS/` sera effectuée sur le fournisseur JMS distant (après avoir supprimé le suffixe) `java:global/remoteJMS/Queue1`, le contexte externe se connectera au fournisseur JMS distant, et engagera la recherche pour la ressource `Queue1`.

4. Configuration d'un adaptateur de ressources générique JMS

Dans `EAP_HOME/standalone/configuration/standalone-full.xml`, ajouter la configuration d'adaptateur de ressources générique dans `<subsystem xmlns="urn:jboss:domain:resource-adapters:1.1">`.

Exemple 22.3. Configuration de l'adaptateur de ressources EMS Tibco

Voici une définition complète d'un adaptateur de ressource dans EMS Tibco.

```

<resource-adapter id="org.jboss.genericjms">
  <module slot="main" id="org.jboss.genericjms"/>
  <transaction-support>NoTransaction</transaction-support>
  <connection-definitions>
    <connection-definition class-
name="org.jboss.resource.adapter.jms.JmsManagedConnectionFactory"
  jndi-name="java:/jms/XAQCF"
  pool-name="XAQCF">
      <config-property name="ConnectionFactory">
        XAQCF
      </config-property>
      <config-property name="JndiParameters">

java.naming.factory.initial=com.tibco.tibjms.naming.TibjmsInitialC
ontextFactory;java.naming.provider.url=TIBCO_EMS_SERVER_HOST_NAME:
PORT
      </config-property>
      <security>
        <application/>
      </security>
    </connection-definition>
  </connection-definitions>
</resource-adapter>

```

5. Configurer le pool de beans basés-messages par l'adaptateur de ressource standard.

Dans `EAP_HOME/standalone/configuration/standalone-full.xml`, de `<subsystem xmlns="urn:jboss:domain:ejb3:1.4">`, mettez la configuration `<mdb>` à jour avec :

```
<mdb>
  <resource-adapter-ref resource-adapter-
name="org.jboss.genericjms"/>
  <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

Résultat

L'adaptateur de ressources JMS standard est maintenant configuré et prêt à l'utilisation. Quand on crée un bean basé-message, utiliser un code qui ressemble à ceci pour pouvoir utiliser l'adaptateur de ressources.

```
@MessageDriven(name = "HelloWorldQueueMDB", activationConfig = {
    // The generic JMS resource adapter requires the JNDI bindings
    // for the actual remote connection factory and destination
    @ActivationConfigProperty(propertyName = "connectionFactory",
propertyValue = "java:global/remoteJMS/XAQCF"),
    @ActivationConfigProperty(propertyName = "destination",
propertyValue = "java:global/remoteJMS/Queue1"),
    @ActivationConfigProperty(propertyName = "destinationType",
propertyValue = "javax.jms.Queue"),
    @ActivationConfigProperty(propertyName = "acknowledgeMode",
propertyValue = "Auto-acknowledge") })
public class HelloWorldQueueMDB implements MessageListener {
    public void onMessage(Message message) {
        // called every time a message is received from the `Queue1`
queue on the JMS provider.
    }
}
```

Vous pouvez également utiliser l'usine de connexions groupées de l'adaptateur de ressources.

```
@Resource(lookup = "java:/jms/XAQCF")
private ConnectionFactory cf;
```

Il n'est pas possible d'injecter directement une ressource d'un contexte externe, mais il est possible d'injecter un contexte externe, puis de faire une recherche. Par exemple, voici à quoi ressemble une recherche de file d'attente déployée dans un broker EMS Tibco.

```
@Resource(lookup = "java:global/remoteJMS")
private Context context;

...

Queue queue = (Queue) context.lookup("Queue1")
```

[Rapporter un bogue](#)

CHAPITRE 23. HIBERNATE SEARCH

23.1. INTRODUCTION À HIBERNATE SEARCH

23.1.1. Hibernate Search

Hibernate Search fournit la fonctionnalité de recherche de texte brut aux applications d'Hibernate. Cette fonctionnalité est particulièrement adaptée aux recherches de demandes pour lesquelles des solutions basées SQL ne sont pas adaptées, c'est à dire les recherches de texte brut, approximations, et géolocalisation. Hibernate Search utilise Apache Lucene comme moteur de recherche de texte intégral, mais est conçu pour minimiser la charge de maintenance. Une fois configuré, l'indexation, le clustering et la synchronisation de données sont maintenus en toute transparence, ce qui vous permet de mettre l'accent sur les besoins de votre entreprise.

[Rapporter un bogue](#)

23.1.2. Aperçu

Hibernate Search se compose d'un composant d'indexation, ainsi que d'un composant de recherche d'index, tous deux soutenus par Apache Lucene. Chaque fois qu'une entité est insérée, mise à jour ou supprimée de la base de données, Hibernate Search assure le suivi de cet événement (par le système d'événements d'Hibernate Search) et planifie une mise à jour de l'index. Toutes ces mises à jour sont gérées sans que vous ayez à interagir directement avec les API d'Apache Lucene. Au lieu de cela, l'interaction avec les indexes de Lucene sous-jacents est gérée par l'intermédiaire d'un `IndexManager`.

Une fois que l'index est créé, vous pouvez rechercher des entités et retourner des listes d'entités gérées, au lieu de vous pencher sur l'infrastructure sous-jacente de Lucene. Le même contexte de persistance est partagé entre Hibernate et Hibernate Search. La classe `FullTextSession` est construite au-dessus de la classe `Hibernate Session` afin que le code d'application puisse utiliser les API unifiés `org.hibernate.Query` ou `javax.persistence.Query` exactement de la même manière qu'une requête native, HQL, ou JPA-QL le ferait.



NOTE

Il est conseillé - à la fois pour votre base de données et pour Hibernate Search - d'exécuter vos opérations dans une transaction, que ce soit JDBC ou JTA.



NOTE

Hibernate Search fonctionne parfaitement dans un modèle de conversation long Hibernate / `EntityManager`, que l'on appelle une conversation atomique.

[Rapporter un bogue](#)

23.1.3. Index Manager

Chaque fois qu'une entité est insérée, mise à jour ou supprimée de la base de données, Hibernate Search assure le suivi de cet événement à travers le système d'événements d'Hibernate et planifie une mise à jour de l'index. L'interaction avec les index de Lucene sous-jacents est gérée par un

IndexManager, qui les identifie chacun par son nom. Par défaut, il existe une relation one-to-one entre l'*IndexManager* et l'index de Lucene. L'*IndexManager* extrait la configuration d'un index spécifique, dont le *backend* sélectionné, la *reader strategy* (stratégie de lecteur) et le *DirectoryProvider* choisi.

[Rapporter un bogue](#)

23.1.4. Fournisseur de répertoires

Apache Lucene, qui fait partie de l'infrastructure d'Hibernate Search, possède un concept de répertoire pour le stockage d'index. Hibernate Search gère l'initialisation et la configuration d'une instance d'un répertoire de Lucene via un *Fournisseur de répertoires*.

La propriété `directory_provider` indique le fournisseur de répertoires à utiliser pour stocker les index. Le fournisseur de répertoires de système de fichier par défaut est `filesystem`, qui utilise le système de fichiers local pour stocker les index.

[Rapporter un bogue](#)

23.1.5. Worker

Les mises à jour aux indexes de Lucene sont gérées par le *worker* d'Hibernate Search, qui reçoit toutes les modifications de l'entité, les met en files d'attente de contexte et les applique une fois qu'un contexte se termine. Le contexte le plus courant est la transaction, mais peut dépendre du nombre de modifications de l'entité ou de certains autres événements de l'application (cycle de vie).

Pour une plus grande efficacité, les interactions sont groupées et sont généralement appliquées une fois que le contexte se termine. En dehors d'une transaction, l'opération de mise à jour d'index est exécutée juste après l'opération de base de données réelle. Dans le cas d'une transaction en cours, l'opération de mise à jour d'index est prévue pour la phase de validation de la transaction et sera ignorée en cas de restauration de transaction. Un worker peut être configuré avec une limite de taille de lot spécifique, après quoi l'indexation se produit quel que soit le contexte.

Pour obtenir des informations sur les options de configuration du worker, voir [Section 23.2.5, « Configuration d'un worker »](#).

Il y a deux avantages immédiats à cette méthode de gérer les mises à jour d'index :

- Performance : l'indexation de Lucene fonctionne mieux quand l'opération est exécutée en lots.
- ACIDity : Le travail exécuté a le même scoping que celui qui est exécuté par la transaction de base de données et sera exécuté si et seulement si la transaction est validée. Ce n'est pas considéré comme ACID au sens strict, mais un comportement acide est rarement utile pour les indexes de recherche de texte intégral, car ils peuvent être reconstruits à partir de la source à tout moment.

Les deux modes de lots - no scope vs transactionnel - sont l'équivalent d'autocommit versus comportement transactionnel. D'un point de vue performance, le mode *transactionnel* est recommandé. Le choix de scoping se fait en toute transparence. Hibernate Search détecte la présence d'une transaction et ajuste le scoping (voir [Section 23.2.5, « Configuration d'un worker »](#)).

[Rapporter un bogue](#)

23.1.6. Installation et opérations de Back End

23.1.6.1. Backend

Hibernate Search utilise plusieurs backends pour traiter des lots de travail. Le backend n'est pas limité à l'option de configuration `default.worker.backend`. Cette propriété spécifie une implémentation de l'interface `BackendQueueProcessor` qui fait partie d'une configuration de backend. Des paramètres supplémentaires sont nécessaires pour mettre en place un backend, par exemple le backend JMS.

[Rapporter un bogue](#)

23.1.6.2. Lucene

En mode Lucene, toutes mises à jour d'index d'un noeud (JVM) sont exécutées par le même nœud vers les répertoires de Lucene en utilisant les fournisseurs de répertoire. Utilisez ce mode dans un environnement non clusterisé ou dans des environnements clusterisés ayant un store de répertoires partagé.

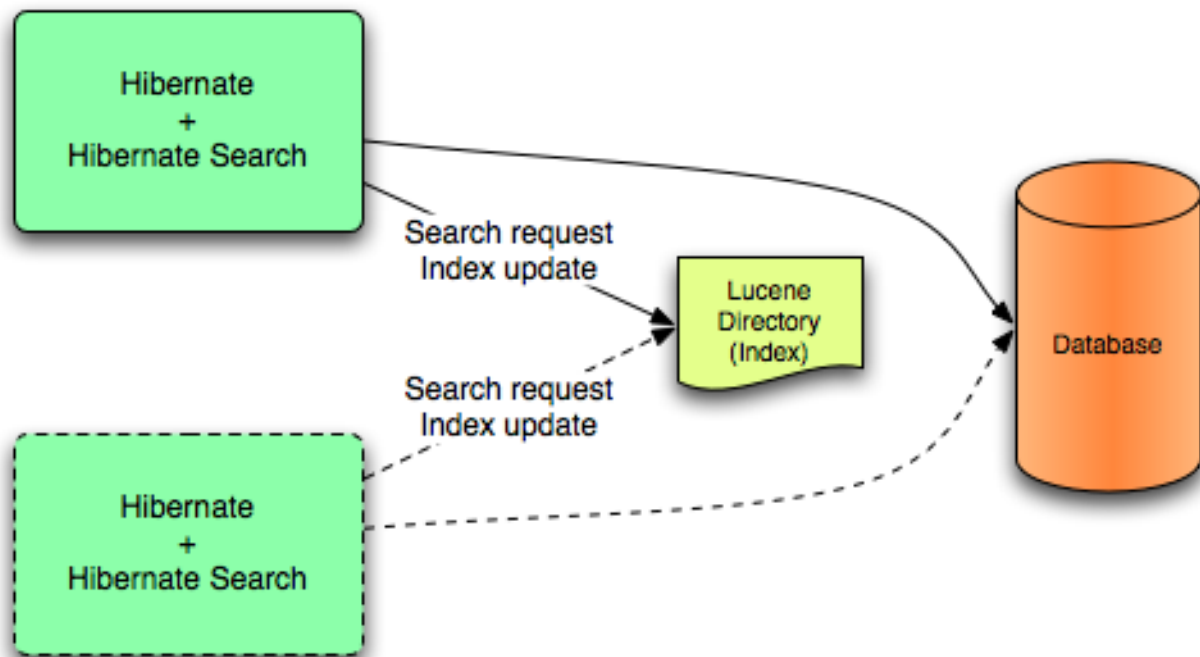


Figure 23.1. Configuration de backend de Lucene

Le mode Lucene cible les applications en cluster ou non clusterisées où le `Directory` gère la stratégie de verrouillage. Le principal avantage du mode de Lucene est sa simplicité et une visibilité instantanée des changements dans les requêtes de Lucene. Le backend NRT est un autre backend pour les configurations d'index non clusterisés ou non partagés.

[Rapporter un bogue](#)

23.1.6.3. JMS

Les mises à jour de l'index d'un noeud sont envoyées à la file d'attente JMS. Un lecteur unique traite la file d'attente et met à jour l'index master. L'index principal est répliqué par la suite régulièrement dans les copies d'esclave pour établir le modèle master/esclave. Le master est chargé des mises à jour des indexes de Lucene. Les esclaves acceptent de lire et d'écrire des opérations mais traite des opérations de lecture sur l'index local. Le master est le seul responsable de la mise à jour de l'index de Lucene. Seul le master applique les modifications locales dans une opération de mise à jour.

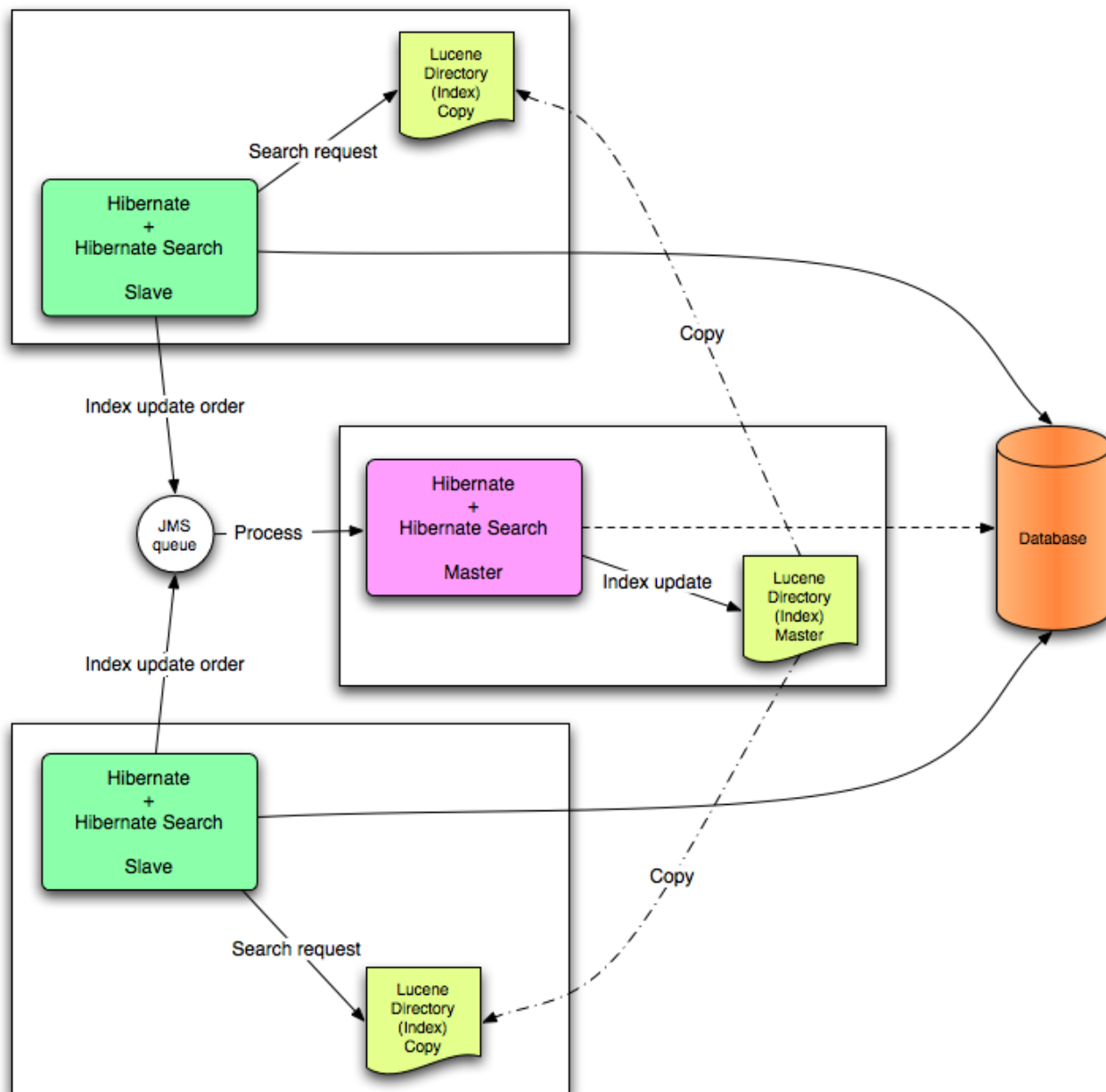


Figure 23.2. Configuration de JMS Backend

Ce mode cible les environnements clusterisés où le débit est critique et dans lesquels on peut se permettre des délais de mise à jour d'index. Le fournisseur JMS veille au niveau de la fiabilité et utilise les esclaves pour modifier les copies d'index locales.

[Rapporter un bogue](#)

23.1.7. Stratégies Reader

Lorsque vous exécutez une requête, Hibernate Search, utiliser une stratégie de lecture pour pouvoir interagir avec les indexes d'Apache Lucene. Choisissez une stratégie de lecture basée sur le profil de la demande (mises à jour fréquentes, lecture principalement, mise à jour asynchrone d'index, etc.).

[Rapporter un bogue](#)

23.1.7.1. Stratégie de partage

En utilisant la stratégie de **partage**, Hibernate Search partage le même `IndexReader` pour un index

de Lucene donné à travers plusieurs requêtes et chaînes, dans la mesure où l' `IndexReader` reste à jour. Si `IndexReader` n'est pas mis à jour, un nouveau sera ouvert et fourni. Chaque `IndexReader` est composé de plusieurs `SegmentReader`. La stratégie de partage rouvre les segments qui ont été modifiés ou créés après la dernière ouverture et partage les segments déjà chargés de l'instance précédente. Il s'agit de la stratégie par défaut.

[Rapporter un bogue](#)

23.1.7.2. Stratégie de non-partage

Avec la stratégie de non partage, un `IndexReader` de Lucene s'ouvre à chaque fois qu'une requête s'exécute. L'ouverture et le démarrage d'un `IndexReader` sont des opérations coûteuses. Ainsi, ouvrir un `IndexReader` pour chaque exécution de la requête d'ouverture n'est pas une stratégie efficace.

[Rapporter un bogue](#)

23.1.7.3. Stratégies de lecture personnalisées

Vous pouvez écrire une stratégie de lecture personnalisée en utilisant une implémentation de `org.hibernate.search.reader.ReaderProvider`. L'implémentation doit être thread-safe.

[Rapporter un bogue](#)

23.1.7.4. Configuration de stratégie de lecture

Modifier la stratégie de la valeur par défaut (`shared`) à `not-shared`, comme suit :

```
hibernate.search.[default|<indexname>].reader.strategy = not-shared
```

Sinon, vous pouvez personnaliser la stratégie de lecture en remplaçant `my.corp.myapp.CustomReaderProvider` par une implémentation de stratégie personnalisée :

```
hibernate.search.[default|<indexname>].reader.strategy =
my.corp.myapp.CustomReaderProvider
```

[Rapporter un bogue](#)

23.2. CONFIGURATION

23.2.1. Configuration minimum

Hibernate Search a été conçu pour fournir une flexibilité de configuration et de fonctionnement, avec des valeurs par défaut choisies avec soin pour répondre à la majorité des cas d'utilisation. Un `Directory Provider` au moins doit être configuré, ainsi que de ses propriétés. Le fournisseur de répertoires par défaut est le système de fichiers qui utilise le système de fichiers local pour le stockage de l'index. Pour plus d'informations sur les fournisseurs de répertoires disponibles et leur configuration, consultez [Section 23.2.3, « Configuration de DirectoryProvider »](#).

Si vous utilisez Hibernate directement, les paramètres de configuration comme `DirectoryProvider` doivent être définis dans le fichier de configuration, soit `hibernate.properties` ou `hibernate.cfg.xml`. Si vous utilisez Hibernate via JPA, le fichier de configuration sera `persistence.xml`.

[Rapporter un bogue](#)

23.2.2. Configurer l'IndexManager

Hibernate Search propose plusieurs implémentations pour cette interface :

- **directory-based** : l'implémentation par défaut qui utilise l'abstraction **Directory** de Lucène pour gérer les fichiers d'index.
- **near-real-time** : éviter de vider les écritures sur le disque à chaque validation. Le gestionnaire d'index est aussi basé **Directory** (répertoire), mais utilise la fonctionnalité (NRT) en temps réel de Lucène.

Pour spécifier un IndexManager autre que la valeur par défaut, spécifiez la propriété suivante :

```
hibernate.search.[default|<indexname>].indexmanager = near-real-time
```

[Rapporter un bogue](#)

23.2.2.1. Basée répertoire

L'implémentation basée **annuaire** est l'implémentation d'**IndexManager** par défaut. Elle est hautement configurable et permet des configurations distinctes de stratégie de lecture, des serveurs de backend et des fournisseurs de répertoires.

[Rapporter un bogue](#)

23.2.2.2. Near Real Time (NRT)

La classe **NRTIndexManager** est une extension de l'**IndexManager** par défaut, et tire profit de la fonctionnalité Lucene NRT (Near Real Time) pour les écritures d'index de faible latence. Cependant, elle ignore les paramètres de configuration des serveurs alternatifs de backend non **Lucene** et obtient des verrous d'écriture exclusifs pour la classe **Directory**.

L'**IndexWriter** ne vide pas de tous les changements sur le disque pour fournir une latence faible. Les requêtes peuvent lire les états mis à jour dans les tampons d'écrivain d'index non vidés. Cependant, cela signifie que si **IndexWriter** est arrêté ou si l'application se bloque, les mises à jour peuvent être perdues si les index doivent être reconstruits.

La configuration NRT est conseillée pour les site web non clusterisés à données limitées, en raison des inconvénients mentionnés et parce qu'un noeud maître peut être configuré individuellement pour une meilleure performance également.

[Rapporter un bogue](#)

23.2.2.3. Personnalisé

Spécifiez un nom de classe qualifié complet d'implémentation personnalisée pour créer un **IndexManager** personnalisé. Créer un constructeur no-argument pour l'implémentation, comme suit :

```
[default|<indexname>].indexmanager = my.corp.myapp.CustomIndexManager
```

L'implémentation de gestionnaire d'index personnalisé ne nécessite pas les mêmes composants que les implémentations par défaut. Par exemple, elle délègue un service d'indexation distant qui n'expose pas une interface de répertoire.

[Rapporter un bogue](#)

23.2.3. Configuration de DirectoryProvider

Un `DirectoryProvider` est l'abstraction d'Hibernate Search autour d'un `Directory` de Lucene qui gère la configuration et l'initialisation des ressources de Lucene sous-jacentes. [Les fournisseurs de répertoires et leurs propriétés](#) affiche la liste des fournisseurs de répertoires disponibles dans Hibernate Search, ainsi que leurs options correspondantes.

Chaque entité indexée est associée à un index de Lucene (sauf quand plusieurs entités partagent un même index). Le nom de l'index est donné par la propriété `index` de l'annotation `@Indexed`. Si la propriété `index` n'est pas spécifiée, le nom complet de la classe indexée servira de nom (recommandé).

Le `DirectoryProvider` et toute option supplémentaire peuvent être configurés en utilisant le préfixe `hibernate.search.<indexname>`. Le nom par défaut (`hibernate.search.default`) est réservé et peut être utilisé pour définir les propriétés qui s'appliquent à tous les indexes.

[Exemple 23.2, « Configurer les fournisseurs de répertoires »](#) montre comment `hibernate.search.default.directory_provider` est utilisée pour définir le fournisseur de répertoire par défaut du système de fichiers. `hibernate.search.default.indexBase` définit ensuite le répertoire de base par défaut de l'index. Ainsi, l'index de l'entité `Status` est créé dans `/usr/lucene/indexes/org.hibernate.example.Status`.

L'index de l'entité `Rule`, cependant, utilise un répertoire en mémoire, car le fournisseur de répertoires par défaut de cette entité est substitué par la propriété `hibernate.search.Rules.directory_provider`.

Finalement, l'entité `Action` utilise un fournisseur de répertoire personnalisé `CustomDirectoryProvider` spécifié par `hibernate.search.Actions.directory_provider`.

Exemple 23.1. Spécifier le nom d'index

```
package org.hibernate.example;

@Indexed
public class Status { ... }

@Indexed(index="Rules")
public class Rule { ... }

@Indexed(index="Actions")
public class Action { ... }
```

Exemple 23.2. Configurer les fournisseurs de répertoires

```
hibernate.search.default.directory_provider = filesystem
hibernate.search.default.indexBase=/usr/lucene/indexes
hibernate.search.Rules.directory_provider = ram
```

```
hibernate.search.Actions.directory_provider =
com.acme.hibernate.CustomDirectoryProvider
```



NOTE

En utilisant le schéma de configuration décrit, vous pourrez facilement définir des règles communes comme le fournisseur de répertoire ou le répertoire de base, et remplacer ces valeurs par défaut plus tard, un index à la fois.

Les fournisseurs de répertoires et leurs propriétés

ram

Aucun

système de fichiers

Répertoire basé système de fichiers. Le répertoire utilisé sera `<indexBase>/< indexName >`

- **indexBase** : répertoire de base
- **indexName**: remplace `@Indexed.index` (utile pour les index partitionnés)
- **locking_strategy** : en option, voir [Section 23.2.7, « Configuration LockFactory »](#)
- **filesystem_access_type**: permet de déterminer le type exact de l'implémentation `FSDirectory` utilisée par ce fournisseur de répertoire `DirectoryProvider`. Les valeurs autorisées sont `auto` (la valeur par défaut, `NIOFSDirectory` sur les systèmes non Windows, `SimpleFSDirectory` sur les systèmes Windows), `simple` (`SimpleFSDirectory`), `nio` (`NIOFSDirectory`), `mmap` (`MMapDirectory`). Consulter les Javadocs de ces implémentations de `Directory` avant de modifier cette configuration. Malgré le fait que `NIOFSDirectory` ou `MMapDirectory` peuvent apporter des gains de performance appréciables, ils ont aussi leurs problèmes associés.

filesystem-master

Répertoire basé système de fichiers. Tout comme `filesystem`, il copie également l'index dans un répertoire source régulièrement.

La valeur recommandée pour la période d'actualisation est (au moins) 50 % plus élevée que le temps qu'il faut pour copier les informations (par défaut 3 600 secondes - 60 minutes).

Notez que la copie est basée sur un mécanisme de copie incrémental qui réduit le temps moyen de copie.

`DirectoryProvider` utilisé normalement sur le noeud maître dans un cluster JMS Backend.

La taille `buffer_size_on_copy` optimum dépendra de votre système d'exploitation et de la mémoire dont vous disposez. La plupart des gens rapportent qu'ils ont de bons résultats en utilisant des valeurs entre 16 et 64Mo.

- **indexBase**: répertoire de base
- **indexName**: remplace `@Indexed.index` (utile pour les index partitionnés)

- **sourceBase**: répertoire de base source (copie).
- **source**: suffixe du répertoire source (valeur par défaut `@Indexed.index`). Le nom du répertoire source étant `<sourceBase>/<source>`
- **refresh**: période de réactualisation en secondes (la copie aura lieu tous les `refresh` secondes). Si une copie est toujours en cours quand la période `refresh` a expiré, la seconde opération de copie sera ignorée.
- **buffer_size_on_copy**: le montant de mégaoctets à déplacer dans une seule instruction de copie de bas niveau; la valeur par défaut est 16 Mo.
- **locking_strategy** : en option, voir [Section 23.2.7, « Configuration LockFactory »](#)
- **filesystem_access_type**: permet de déterminer le type exact de l'implémentation `FSDirectory` utilisée par ce fournisseur de répertoire `DirectoryProvider`. Les valeurs autorisées sont `auto` (la valeur par défaut, `NIOFSDirectory` sur les systèmes non Windows, `SimpleFSDirectory` sur les systèmes Windows), `simple` (`SimpleFSDirectory`), `nio` (`NIOFSDirectory`), `mmap` (`MMapDirectory`). Consulter les Javadocs de ces implémentations de `Directory` avant de modifier cette configuration. Malgré le fait que `NIOFSDirectory` ou `MMapDirectory` peuvent apporter des gains de performance appréciables, ils ont aussi leurs problèmes associés que vous devez connaître.

filesystem-slave

Répertoire basé système de fichiers. Ressemble au système de fichiers, mais extrait une version maître (source) régulièrement. Pour éviter le verrouillage et les résultats de recherche inconsistants, 2 copies locales sont conservées.

La valeur recommandée pour la période d'actualisation est (au moins) 50 % plus élevée que le temps qu'il faut pour copier les informations (par défaut 3 600 secondes - 60 minutes).

Notez que la copie est basée sur un mécanisme de copie incrémentielle, réduisant le temps de copie moyen. Si une copie est en cours d'exécution quand la période `refresh` expire, la seconde opération de copie sera ignorée.

`DirectoryProvider` normalement utilisé sur des noeuds esclaves utilisant un JMS Backend.

La taille `buffer_size_on_copy` optimum dépendra de votre système d'exploitation et de la mémoire dont vous disposez. La plupart des gens rapportent qu'ils ont de bons résultats en utilisant des valeurs entre 16 et 64Mo.

- **indexBase**: répertoire de base
- **indexName**: remplace `@Indexed.index` (utile pour les index partitionnés)
- **sourceBase**: répertoire de base source (copie)
- **source**: suffixe du répertoire source (valeur par défaut `@Indexed.index`). Le nom du répertoire source étant `<sourceBase>/<source>`
- **refresh**: période de réactualisation en secondes
- **buffer_size_on_copy**: le montant de mégaoctets à déplacer dans une seule instruction de copie de bas niveau; la valeur par défaut est 16 Mo.

- **locking_strategy** : en option, voir [Section 23.2.7, « Configuration LockFactory »](#)
- **retry_marker_lookup** : optionnel, valeur par défaut 0. Définit le nombre de fois qu'Hibernate Search ira chercher les fichiers de marqueurs dans le répertoire source avant d'échouer. Attend 5 secondes entre chaque tentative.
- **retry_initialize_period** : facultatif, définissez une valeur entière en quelques secondes pour permettre à la nouvelle tentative initialiser caractéristique : si l'esclave ne peut pas trouver l'index maître il réessayera jusqu'à ce qu'il se trouve dans le fond, sans pour autant empêcher l'application à démarrer : les requêtes de texte intégral effectuées avant l'initialisation de l'index ne sont pas bloqués, mais retournera les résultats vides. Quand pas l'activation de l'option ou explicitement mise à zéro, il échoue avec une exception au lieu de planifier une minuterie de réessayer. Pour empêcher l'application de démarrer sans un index non valide, mais toujours pas à contrôler un délai d'initialisation, consultez plutôt le **retry_marker_lookup**.
- **filesystem_access_type**: permet de déterminer le type exact de l'implémentation **FSDirectory** utilisée par ce fournisseur de répertoire **DirectoryProvider**. Les valeurs autorisées sont **auto** (la valeur par défaut, **NIOFSDirectory** sur les systèmes non Windows, **SimpleFSDirectory** sur les systèmes Windows), **simple** (**SimpleFSDirectory**), **nio** (**NIOFSDirectory**), **mmap** (**MMapDirectory**). Consulter les Javadocs de ces implémentations de **Directory** avant de modifier cette configuration. Malgré le fait que **NIOFSDirectory** ou **MMapDirectory** peuvent apporter des gains de performance appréciables, vous devez vous familiariser avec les problèmes associés.



NOTE

Si les fournisseurs de répertoire intégré ne correspondent pas à vos besoins, vous pouvez écrire votre propre fournisseur de répertoire, en implémentant l'interface `org.hibernate.store.DirectoryProvider`. Dans ce cas, passez le nom de classe complet de votre fournisseur dans la propriété `directory_provider`. Vous pouvez passer des propriétés supplémentaires en utilisant le préfixe `hibernate.search.<indexname>`.

[Rapporter un bogue](#)

23.2.4. Partitionnement d'indexes

Dans certains cas, il est utile de séparer (sharding) les données indexées d'une entité donnée en plusieurs indexes Lucene.



AVERTISSEMENT

Le partitionnement n'a lieu d'être que si les avantages dépassent les inconvénients. Rechercher des indexes partitionnés sera normalement plus lent car pour une simple recherche, toutes les partitions doivent être ouvertes.

Cas possibles d'utilisation du partitionnement :

- Un index unique est trop volumineux, donc les temps de mises à jour ralentissent l'application.
- Une recherche typique ne touchera qu'un sous-ensemble d'indexes, comme quand les données sont segmentées naturellement par client, région ou application.

Par défaut, le partitionnement n'est pas activé à moins que le nombre de partitions soit configuré. Pour ce faire, utiliser la propriété `hibernate.search`.

`<indexName>.sharding_strategy.nbr_of_shards`.

Exemple 23.3. Activer le partitionnement de l'index

Dans cet exemple, 5 partitions sont activées.

```
hibernate.search.<indexName>.sharding_strategy.nbr_of_shards = 5
```

La stratégie `IndexShardingStrategy` est chargée de diviser les données en sous-index. La stratégie de partitionnement par défaut divise les données en fonction de la valeur de hachage de la représentation de la chaîne d'ID (générée par le `FieldBridge`). Cela garantit un partitionnement assez équilibré. Vous pouvez remplacer la stratégie par défaut avec une `IndexShardingStrategy` personnalisée. Pour utiliser votre stratégie personnalisée, vous devez définir la propriété `hibernate.search.<indexName>.sharding_strategy`.

Exemple 23.4. Spécifier une stratégie de partitionnement personnalisée

```
hibernate.search.<indexName>.sharding_strategy =  
my.shardingstrategy.Implementation
```

La propriété `IndexShardingStrategy` permet également d'optimiser les recherches en sélectionnant dans quelle partition (shard) exécuter la requête. En activant un filtre, une stratégie de partitionnement peut sélectionner un sous-ensemble de sections permettant de répondre à une requête (`IndexShardingStrategy.getIndexManagersForQuery`) et ainsi accélérer l'exécution de la requête.

Chaque partition correspond à un `IndexManager` indépendant, et donc peut être configuré pour utiliser un fournisseur et un serveur de sauvegarde différents. Les noms d'index `IndexManager` pour l'entité `Animal` dans [Exemple 23.5, « Configuration de partitionnement pour l'entité Animal »](#) sont `Animal.0` à `Animal.4`. En d'autres termes, chaque partition porte le nom de l'index auquel il appartient, suivi de `.` (point) et son numéro d'index.

Exemple 23.5. Configuration de partitionnement pour l'entité Animal

```
hibernate.search.default.indexBase = /usr/lucene/indexes  
hibernate.search.Animal.sharding_strategy.nbr_of_shards = 5  
hibernate.search.Animal.directory_provider = filesystem  
hibernate.search.Animal.0.indexName = Animal00  
hibernate.search.Animal.3.indexBase = /usr/lucene/sharded  
hibernate.search.Animal.3.indexName = Animal03
```

Dans [Exemple 23.5, « Configuration de partitionnement pour l'entité Animal »](#), la configuration utilise la stratégie de hachage de chaîne d'identification par défaut et partitionne l'index `Animal` en 5 sous

indexes. Tous les sous-indexes sont des instances du système de fichiers et du répertoire où est stockée chaque sous-index comme suit :

- pour sous-index 0: `/usr/lucene/indexes/Animal00` (indexBase partagée mais indexName substitué)
- pour sous-index 1: `/usr/lucene/indexes/Animal.1` (indexBase partagé, indexName par défaut)
- pour sous-index 2: `/usr/lucene/indexes/Animal.1` (indexBase partagé, indexName par défaut)
- pour sous-index 3: `/usr/lucene/indexes/Animal03` (indexBase substitué, indexName substitué)
- pour sous-index 4: `/usr/lucene/indexes/Animal.4` (indexBase partagé, indexName par défaut)

Lorsque vous implémentez une `IndexShardingStrategy`, n'importe quel champ peut être utilisé pour déterminer la sélection de partitionnement. Considérons que pour traiter des suppressions, avec les opérations `purge` et `purgeAll`, l'implémentation soit amenée à retourner un ou plusieurs index sans être en mesure de lire toutes les valeurs de champ ou l'identificateur principal. Dans ce cas, l'information ne suffit pas pour choisir un seul indice, tous les indexes doivent être retournés, afin que l'opération de suppression puisse être propagée à tous index susceptibles de contenir les documents devant être supprimés.

[Rapporter un bogue](#)

23.2.5. Configuration d'un worker

Il est possible de préciser comment Hibernate Search interagit avec Lucene par la configuration de worker. Il existe plusieurs composants architecturaux et points d'extension possibles. Voyons cela de plus près.

Il y a tout d'abord un `Worker`. Une implémentation de l'interface `Worker` est responsable de recevoir tous les changements d'entité, de les mettre en file d'attente selon le contexte et de les appliquer une fois que le contexte prend fin. Le contexte le plus intuitif, en connexion à ORM, est la transaction. Pour cette raison, Hibernate Search utilisera par défaut le `TransactionalWorker` pour scoper tous les changements par transaction. On peut, cependant, imaginer un scénario où le contexte dépend, par exemple, du nombre de changements d'entités ou d'autres événements d'applications (cycle de vie). Pour cette raison, l'implémentation `Worker` est configurable, comme on peut le voir dans [Tableau 23.1, « Configuration de l'étendue »](#).

Tableau 23.1. Configuration de l'étendue

Property	Description
<code>hibernate.search.worker.scope</code>	Le nom de classe complet de l'implémentation de <code>Worker</code> à utiliser. Si cette propriété n'est pas définie, est vide ou <code>transaction</code> , le <code>TransactionalWorker</code> par défaut sera utilisé.

hibernate.search.worker.*	Toutes les propriétés de configuration ayant comme préfixe <code>hibernate.search.worker</code> sont passées au worker pendant l'initialisation. Cela permet d'ajouter des paramètres spécifiques au worker et personnalisés.
hibernate.search.worker.batch_size	Définit le nombre maximal d'opérations d'indexation par lots de contexte. Une fois la limite atteinte, l'indexation se déclenche même si le contexte n'est pas encore terminé. Cette propriété ne fonctionne que si l'implémentation du <code>worker</code> délègue le travail en file d'attente à <code>BatchedQueueingProcessor</code> (ce que fait le <code>TransactionalWorker</code>)

Une fois qu'un contexte se termine, il est temps de préparer et d'appliquer les modifications d'index. Cela peut être fait, de façon synchrone ou asynchrone, à partir d'un nouveau thread. Les mises à jour synchrones ont l'avantage que l'index est en synchronisation avec les bases de données en permanence. Les mises à jour asynchrones, en revanche, peuvent aider à minimiser le temps de réponse utilisateur. L'inconvénient est qu'il puisse y avoir des divergences entre les états d'index et de bases de données. Étudions les options de configuration indiquées dans [Tableau 23.2, « Configuration d'exécution »](#).



NOTE

Les options suivantes peuvent être différentes pour chaque index; en fait, elles ont besoin du préfixe `indexName`, ou bien, utiliser `default` pour définir la valeur par défaut pour tous les indexes.

Tableau 23.2. Configuration d'exécution

Property	Description
hibernate.search.<indexName>.worker.execution	sync : exécution synchronisée (valeur par défaut) async : exécution asynchronisée
hibernate.search.<indexName>.worker.thread_pool.size	Le backend peut appliquer des mises à jour depuis le même contexte de transaction (ou lot) en parallèle, à l'aide d'un pool de threads. La valeur par défaut est 1. Vous pouvez expérimenter avec des valeurs plus grandes si vous avez beaucoup d'opérations par transaction.
hibernate.search.<indexName>.worker.buffer_queue.max	Définit le nombre maximal de files d'attente de travail si le pool de threads n'est pas alimenté. Utile uniquement pour une exécution asynchrone. Par défaut à l'infini. Si la limite est atteinte, le travail sera effectué par le thread principal.

Jusqu'à présent, tout le travail est fait dans la même Machine virtuelle (VM), quel que soit le mode

d'exécution. Le montant total de travail ne change pas pour la machine virtuelle. Heureusement, il y a une meilleure approche: la délégation. Il est possible d'envoyer le travail d'indexation sur un autre serveur en configurant `hibernate.search.default.worker.backend` - voir [Tableau 23.3, « Configuration Backend »](#). Encore une fois, cette option peut être configurée différemment pour chaque index.

Tableau 23.3. Configuration Backend

Property	Description
<code>hibernate.search.<indexName>.worker.backend</code>	<p>lucene: le backend par défaut qui exécute les mises à jour des indexes dans une même VM. Utilisé également quand la propriété est indéfinie ou vide.</p> <p>jms: JMS backend. Les mises à jour d'indexes sont envoyées dans une file d'attente JMS afin d'être traitées par un master d'indexation. Voir Tableau 23.4, « Configuration de JMS Backend » pour obtenir des options de configuration supplémentaires et Section 23.2.5.1, « JMS Master/Esclave Backend » pour obtenir une description plus détaillée pour cette installation.</p> <p>blackhole: une configuration test/developper qui ignore tout le travail d'indexation</p> <p>Vous pouvez également spécifier le nom complet d'une classe qui implémente BackendQueueProcessor. De cette façon, vous pouvez implémenter votre propre couche de communication. L'implémentation est chargée de retourner une instance Runnable (exécutable) qui, sur l'exécution, traitera le travail de l'index.</p>

Tableau 23.4. Configuration de JMS Backend

Property	Description
<code>hibernate.search.<indexName>.worker.jndi.*</code>	Définit les propriétés JNDI pour initier InitialContext (si besoin est). JNDI n'est utilisé que par JMS Backend.
<code>hibernate.search.<indexName>.worker.jms.connection_factory</code>	Obligatoire avec JMS Backend. Définit le nom JNDI pour rechercher la fabrique de connexions JMS (/ ConnectionFactory par défaut dans Red Hat JBoss Enterprise Application Platform)
<code>hibernate.search.<indexName>.worker.jms.queue</code>	Obligatoire avec JMS Backend. Définit le nom JNDI pour rechercher la file d'attente JMS. La file d'attente sera utilisée pour poster les messages travail.



AVERTISSEMENT

Comme vous l'avez sans doute remarqué, certaines des propriétés indiquées sont corrélées, ce qui signifie que toutes les combinaisons de valeurs de propriété ne sont pas forcément logiques. En fait, vous pouvez vous retrouver avec une configuration non fonctionnelle. Cela est particulièrement vrai dans le cas où vous fournissiez vos propres implémentations de certains des interfaces illustrées. Assurez-vous d'étudier le code existant avant d'écrire votre propre implémentation de `Worker` ou de `BackendQueueProcessor`.

[Rapporter un bogue](#)

23.2.5.1. JMS Master/Esclave Backend

Cette section décrit comment configurer l'architecture d'Hibernate Search Master/Esclave.

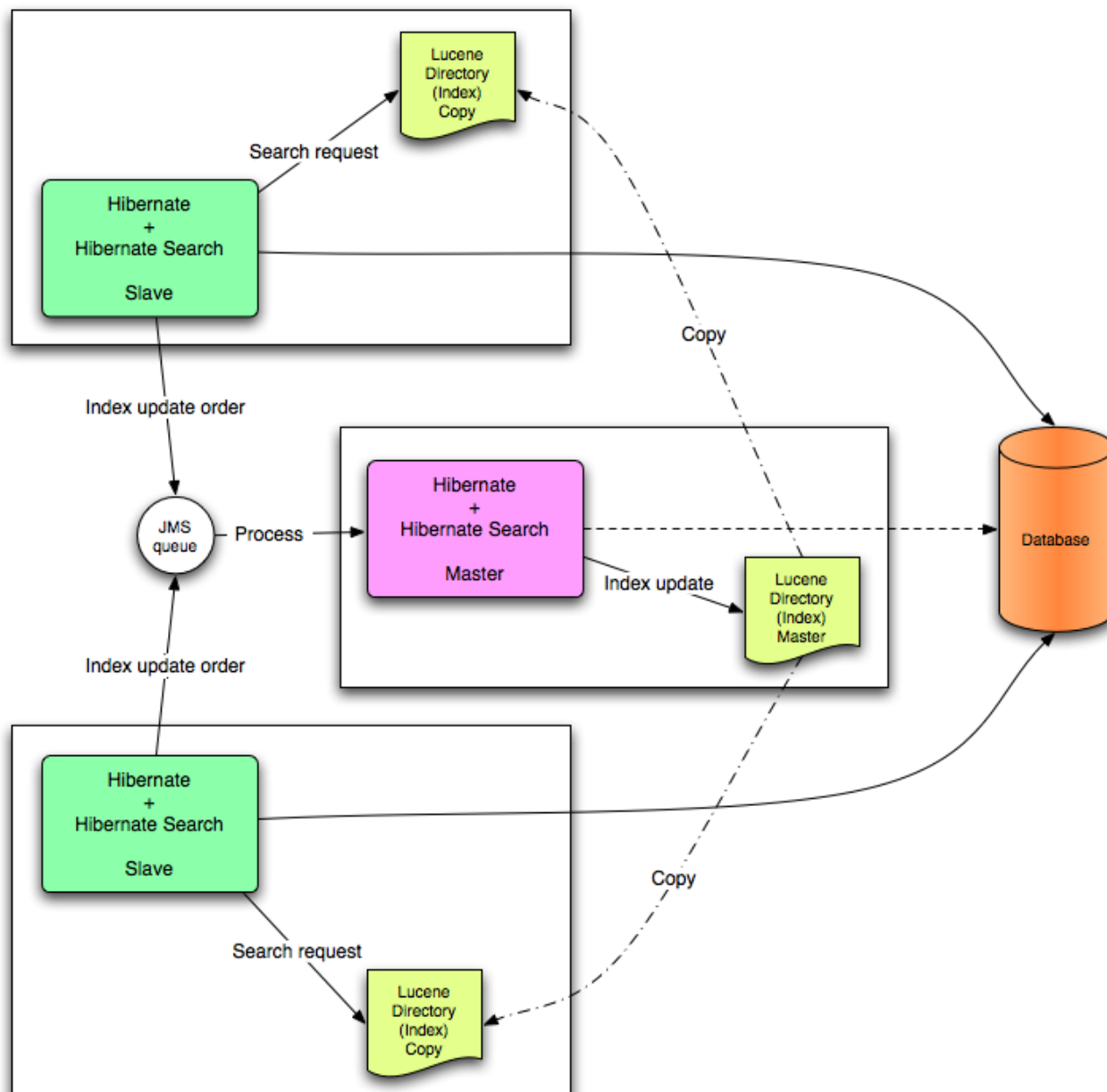


Figure 23.3. Configuration de JMS Backend

[Rapporter un bogue](#)

23.2.5.2. Noeuds esclaves

Chaque opération de mise à jour des indexes est envoyée dans une file d'attente JMS. Les opérations de recherche d'index sont exécutées sur une copie d'index locale.

Exemple 23.6. Configuration d'esclave JMS

```

### slave configuration

## DirectoryProvider
# (remote) master location
hibernate.search.default.sourceBase =
/mnt/mastervolume/lucenedirs/mastercopy

# local copy location
  
```

```
hibernate.search.default.indexBase = /Users/prod/lucenedirs

# refresh every half hour
hibernate.search.default.refresh = 1800

# appropriate directory provider
hibernate.search.default.directory_provider = filesystem-slave

## Backend configuration
hibernate.search.default.worker.backend = jms
hibernate.search.default.worker.jms.connection_factory =
/ConnectionFactory
hibernate.search.default.worker.jms.queue = queue/hibernatesearch
#optional jndi configuration (check your JMS provider for more
information)

## Optional asynchronous execution strategy
# hibernate.search.default.worker.execution = async
# hibernate.search.default.worker.thread_pool.size = 2
# hibernate.search.default.worker.buffer_queue.max = 50
```

**NOTE**

On conseille de faire une copie locale du système de fichiers pour accélérer les recherches.

[Rapporter un bogue](#)

23.2.5.3. Noeud Master

Chaque opération de mise à jour des indexes provient d'une file d'attente JMS avant d'être exécutée. L'index master est copié régulièrement.

Exemple 23.7. La configuration de JMS Master

```
### master configuration

## DirectoryProvider
# (remote) master location where information is copied to
hibernate.search.default.sourceBase =
/mnt/mastervolume/lucenedirs/mastercopy

# local master location
hibernate.search.default.indexBase = /Users/prod/lucenedirs

# refresh every half hour
hibernate.search.default.refresh = 1800

# appropriate directory provider
hibernate.search.default.directory_provider = filesystem-master

## Backend configuration
#Backend is the default lucene one
```

En plus de la configuration de framework d'Hibernate Search, on doit écrire et définir le Message Driven Bean pour traiter la file d'attente de travaux d'indexation par JMS.

Exemple 23.8. Message Driven Bean traitant la file d'attente d'indexation

```
@MessageDriven(activationConfig = {
    @ActivationConfigProperty(propertyName="destinationType",
        propertyValue="javax.jms.Queue"),
    @ActivationConfigProperty(propertyName="destination",
        propertyValue="queue/hibernatesearch"),
    @ActivationConfigProperty(propertyName="DLQMaxResent",
        propertyValue="1")
})
public class MDBSearchController extends
    AbstractJMSHibernateSearchController
        implements MessageListener {
    @PersistenceContext EntityManager em;

    //method retrieving the appropriate session
    protected Session getSession() {
        return (Session) em.getDelegate();
    }

    //potentially close the session opened in #getSession(), not needed
    here
    protected void cleanSessionIfNeeded(Session session)
    {
    }
}
```

Cet exemple hérite de la classe abstraite de contrôleur JMS qui se trouve dans le code source d'Hibernate Search et implémente un MDB JavaEE. Cette implémentation est donnée à titre d'exemple et peut être ajustée pour utiliser un MDB non Java EE. Pour plus d'informations sur `getSession()` et `cleanSessionIfNeeded()`, voir `AbstractJMSHibernateSearchController`'s javadoc.

[Rapporter un bogue](#)

23.2.6. Réglage de l'indexation de Lucene

23.2.6.1. Optimiser la performance d'indexation de Lucene

Hibernate Search est utilisée pour optimiser les performances d'indexation Lucene en spécifiant un ensemble de paramètres qui sont passés à l'`IndexWriter` de Lucene sous-jacent, comme `mergeFactor`, `maxMergeDocs` et `maxBufferedDocs`. Spécifiez ces paramètres soit en tant que valeurs par défaut pour tous les index, sur une base par index ou par partition.

Il existe plusieurs configurations d'`IndexWriter` de bas niveau qui peuvent être ajustées suivant les cas d'utilisation. Ces paramètres sont regroupés sous le mot clé `indexwriter` :

```
hibernate.search.[default|<indexname>].indexwriter.<parameter_name>
```

S'il n'y a pas de valeur pour `indexwriter` de déterminée pour une configuration de partition, alors Hibernate Search regardera dans la section de l'index, puis dans la section par défaut.

La configuration dans la table suivante s'applique à la seconde partition de l'index `Animal` :

- `max_merge_docs` = 10
- `merge_factor` = 20
- `ram_buffer_size` = 64MB
- `term_index_interval` = Lucene default

Toutes les autres valeurs utiliseront les valeurs par défaut utilisées dans Lucene.

Les valeurs par défaut de Lucene sont les valeurs de configuration par défaut. Les valeurs listées dans [Tableau 23.5, « Liste des propriétés de comportement et de performance d'indexation »](#) dépendent de la version de Lucene utilisée. Les valeurs indiquées se rapportent à la version 2.4. Pour plus d'informations sur la performance d'indexation de Lucene, voir la documentation de Lucene.



NOTE

Les versions précédentes d'Hibernate Search avaient la notion de propriété `batch` (lot) et de `transaction`. Ce n'est plus le cas car le backend effectuera toujours les travaux avec les mêmes paramètres.

Tableau 23.5. Liste des propriétés de comportement et de performance d'indexation

Propriété	Description	Valeur par défaut
<code>hibernate.search.[default <indexname>].exclusive_index_use</code>	Définir à true quand aucun autre processus n'aura besoin d'écrire le même index. Cela permet à Hibernate Search de fonctionner en mode exclusif sur l'index et d'améliorer la performance quand on fait des modifications dans l'index.	true (meilleure performance, débloque les verrous uniquement lors de la fermeture)
<code>hibernate.search.[default <indexname>].max_queue_length</code>	Chaque index a un « pipeline » séparé qui contient les mises à jour à appliquer à l'index. Quand cette file d'attente est pleine, ajouter des opérations supplémentaires à la file d'attente se traduit en opération de blocage. Cette configuration de paramètre n'est pas très logique à moins que <code>worker.execution</code> soit configuré à async .	1000
<code>hibernate.search.[default <indexname>].indexwriter.max_buffered_delete_terms</code>	Détermine le nombre minimum de d'expressions à supprimer avant que les expressions à supprimer en mémoire tampon s'appliquent et soient vidées. S'il y a des documents en mémoire tampon à ce moment là, ils sont fusionnés et un nouveau segment est créé.	Désactivé (vidages par utilisation RAM)

Propriété	Description	Valeur par défaut
<code>hibernate.search.[default <indexname>].indexwriter.max_buffered_docs</code>	Contrôle le nombre de documents présents en mémoire tampon lors de l'indexation. Plus il y en a, plus il y a de RAM utilisée.	Désactivé (vidages par utilisation RAM)
<code>hibernate.search.[default <indexname>].indexwriter.max_merge_docs</code>	Définit le nombre maximum de documents autorisés par segment. Les petites valeurs fonctionnent mieux sur les indexes changeant fréquemment; les valeurs plus élevées permettent une meilleure performance de recherche si l'index ne change pas trop souvent.	Illimité (Integer.MAX_VALUE)
<code>hibernate.search.[default <indexname>].indexwriter.merge_factor</code>	Contrôle la taille et la fréquence des fusions de segments. Détermine le nombre de fois que les indexes de segments sont fusionnés quand une insertion a lieu. Pour les petites valeurs, on utilise moins de RAM lors de l'indexation, et les recherches sur indexes non optimisés sont plus rapides, mais la vitesse d'indexation est lente. Ainsi, les valeurs plus élevées, (> 10) sont plus appropriées pour la création d'index de lots, et les plus petites valeurs (< 10) sont plus appropriées pour les indexes maintenus de façon interactive. La valeur doit être inférieure à 2.	10
<code>hibernate.search.[default <indexname>].indexwriter.merge_min_size</code>	Contrôle la taille et la fréquence des fusions de segments. Les segments inférieurs à cette taille (en Mo) sont toujours pris en compte pour la prochaine opération de fusionnement de segments. Si vous configurez une taille trop élevée, vous obtiendrez d'énormes opérations de fusionnement, même si elles sont moins fréquentes. Voir également <code>org.apache.lucene.index.LogDocMergePolicy.minMergeSize</code> .	0 Mo (en réalité ~1K)

Propriété	Description	Valeur par défaut
<code>hibernate.search.[default <indexname>].indexwriter.merge_max_size</code>	<p>Contrôle la taille et la fréquence des fusions de segments.</p> <p>Les segments inférieures à cette taille (en Mo) ne sont jamais fusionnés en segments plus importants.</p> <p>Cela fait des économies de mémoire et évite les opérations de fusionnement qui ont lieu aux dépens d'une vitesse optimale de recherche. Quand on optimise un index, cette valeur est ignorée.</p> <p>Voir également <code>org.apache.lucene.index.LogDocMergePolicy.maxMergeSize</code>.</p>	Illimité
<code>hibernate.search.[default <indexname>].indexwriter.merge_max_optimize_size</code>	<p>Contrôle la taille et la fréquence des fusions de segments.</p> <p>Les segments supérieurs en taille à ceci (en Mo) ne sont pas fusionnés en segments plus importants même quand on optimise l'index (voir la configuration <code>merge_max_size</code> également).</p> <p>S'appliquant à <code>org.apache.lucene.index.LogDocMergePolicy.maxMergeSizeForOptimize</code>.</p>	Illimité
<code>hibernate.search.[default <indexname>].indexwriter.merge_calibrate_by_deletes</code>	<p>Contrôle la taille et la fréquence des fusions de segments.</p> <p>Définir à false afin de ne pas considérer les documents supprimés quand on estime la stratégie de fusionnement.</p> <p>S'appliquant à <code>org.apache.lucene.index.LogMergePolicy.calibrateSizeByDeletes</code>.</p>	true

Propriété	Description	Valeur par défaut
hibernate.search.[default <indexname>].indexwriter.ram_buffer_size	<p>Contrôle le montant de RAM en Mo qui est utilisé pour les tampons de documents. Quand ils sont utilisés avec <code>max_buffered_docs</code>, on assiste à un vidage, selon qu'un événement a lieu avant l'autre.</p> <p>Normalement, pour augmenter la performance d'indexation, il vaut mieux vider par l'utilisation RAM et non pas le nombre de documents, et utiliser le plus grand tampon RAM possible.</p>	16 Mo
hibernate.search.[default <indexname>].indexwriter.term_index_interval	<p>Expert : définir l'intervalle entre les expressions indexées.</p> <p>Avec les valeurs élevées, <code>IndexReader</code> utilise moins de mémoire, mais l'accès au hasard aux termes est plus lent. Avec les valeurs moins élevées, <code>IndexReader</code> utilise plus de mémoire, et la vitesse d'accès aux termes est plus élevée. Voir la documentation Lucene pour obtenir plus d'informations.</p>	128
hibernate.search.[default <indexname>].indexwriter.use_compound_file	<p>L'avantage d'utiliser un format de fichier composite, c'est qu'il y a moins de descripteurs de fichiers utilisés. L'inconvénient, c'est que l'indexation prend plus de temps et d'espace de disque temporaire. Vous pouvez définir ce paramètre à false en vue d'améliorer le temps d'indexation, mais vous pourriez manquer de descripteurs de fichiers si mergeFactor est trop élevé.</p> <p>Pour les paramètres booléens, utiliser "true" ou "false". La valeur par défaut pour cette option est true.</p>	true

Propriété	Description	Valeur par défaut
hibernate.search.enable_dirty_check	<p>Tous les changements d'entité ne nécessitent pas une mise à jour d'index Lucene. Si toutes les propriétés d'entité mises à jour (propriétés dirty) ne sont pas indexées, Hibernate Search évitera le processus de ré-indexation.</p> <p>Désactiver cette option si vous personnalisez des FieldBridge qui ont besoin d'être évoqués à chaque mise à jour d'événement (même si la propriété pour laquelle le pontage de champ est configuré n'a pas changé).</p> <p>Cette optimisation ne sera plus utilisée sur les classes qui utilisent un @ClassBridge ou un @DynamicBoost.</p> <p>Pour les paramètres booléens, utiliser "true" ou "false". La valeur par défaut pour cette option est true.</p>	true



AVERTISSEMENT

Le back end **blackhole** n'est pas conçu pour être utilisé en production, sauf en tant qu'outil pour identifier les étranglements d'indexation.

[Rapporter un bogue](#)

23.2.6.2. L'IndexWriter de Lucene

Il existe plusieurs configurations d'**IndexWriter** de bas niveau qui peuvent être ajustées suivant les cas d'utilisation. Ces paramètres sont regroupés sous le mot clé **indexwriter** :

```
default.<indexname>.indexwriter.<parameter_name>
```

S'il n'y a pas de valeur pour **indexwriter** de déterminée pour une configuration de partition, alors Hibernate Search regardera dans la section de l'index, puis dans la section par défaut.

[Rapporter un bogue](#)

23.2.6.3. Configuration des options de performance

La configuration suivante s'applique à l'index de la seconde partition d'**Animal** :

Exemple 23.9. Exemple de configuration d'option de performance

```

default.Animals.2.indexwriter.max_merge_docs = 10
default.Animals.2.indexwriter.merge_factor = 20
default.Animals.2.indexwriter.term_index_interval = default
default.indexwriter.max_merge_docs = 100
default.indexwriter.ram_buffer_size = 64

```

- `max_merge_docs = 10`
- `merge_factor = 20`
- `ram_buffer_size = 64MB`
- `term_index_interval = Lucene default`

Toutes les autres valeurs utiliseront les valeurs par défaut utilisées dans Lucene.

Les valeurs par défaut de Lucene sont les valeurs de configuration par défaut d'Hibernate Search. De ce fait, les valeurs listées dans le tableau suivant dépendent de la version de Lucene utilisée. Les valeurs indiquées se rapportent à la version 2.4. Pour plus d'informations sur la performance d'indexation de Lucene, voir la documentation de Lucene.



NOTE

Le back end fonctionnera toujours en utilisant les mêmes paramètres de configuration.

Tableau 23.6. Liste des propriétés de comportement et de performance d'indexation

Propriété	Description	Valeur par défaut
default.<indexname>.exclusive_index_use	Définir à true quand aucun autre processus n'aura besoin d'écrire le même index. Cela permet à Hibernate Search de fonctionner en mode exclusif sur l'index et d'améliorer la performance quand on fait des modifications dans l'index.	true (meilleure performance, débloque les verrous uniquement lors de la fermeture)
default.<indexname>.max_queue_length	Chaque index a un « pipeline » séparé qui contient les mises à jour à appliquer à l'index. Quand cette file d'attente est pleine, ajouter des opérations supplémentaires à la file d'attente se traduit en opération de blocage. Cette configuration de paramètre n'est pas très logique à moins que worker.execution soit configuré à async .	1000
default.<indexname>.indexwriter.max_buffered_delete_terms	Détermine le nombre minimum de d'expressions à supprimer avant que les expressions à supprimer en mémoire tampon s'appliquent et soient vidées. S'il y a des documents en mémoire tampon à ce moment là, ils sont fusionnés et un nouveau segment est créé.	Désactivé (vidages par utilisation RAM)

Propriété	Description	Valeur par défaut
default. <indexname>.indexwriter.max_buffered_docs	Contrôle le nombre de documents présents en mémoire tampon lors de l'indexation. Plus il y en a, plus il y a de RAM utilisée.	Désactivé (vidages par utilisation RAM)
default. <indexname>.indexwriter.max_merge_docs	Définit le nombre maximum de documents autorisés par segment. Les petites valeurs fonctionnent mieux sur les indexes changeant fréquemment; les valeurs plus élevées permettent une meilleure performance de recherche si l'index ne change pas trop souvent.	Illimité (Integer.MAX_VALUE)
default. <indexname>.indexwriter.merge_factor	Contrôle la taille et la fréquence des fusions de segments. Détermine le nombre de fois que les indexes de segments sont fusionnés quand une insertion a lieu. Pour les petites valeurs, on utilise moins de RAM lors de l'indexation, et les recherches sur indexes non optimisés sont plus rapides, mais la vitesse d'indexation est lente. Ainsi, les valeurs plus élevées, (> 10) sont plus appropriées pour la création d'index de lots, et les plus petites valeurs (< 10) sont plus appropriées pour les indexes maintenus de façon interactive. La valeur doit être inférieure à 2.	10
default. <indexname>.indexwriter.merge_min_size	Contrôle la taille et la fréquence des fusions de segments. Les segments inférieurs à cette taille (en Mo) sont toujours pris en compte pour la prochaine opération de fusionnement de segments. Si vous configurez une taille trop élevée, vous obtiendrez d'énormes opérations de fusionnement, même si elles sont moins fréquentes. Voir également org.apache.lucene.index.LogDocMergePolicy.minMergeSize.	0 Mo (en réalité ~1K)

Propriété	Description	Valeur par défaut
default. <indexname>.indexwriter.merge_max_size	<p>Contrôle la taille et la fréquence des fusions de segments.</p> <p>Les segments inférieures à cette taille (en Mo) ne sont jamais fusionnés en segments plus importants.</p> <p>Cela fait des économies de mémoire et évite les opérations de fusionnement qui ont lieu aux dépens d'une vitesse optimale de recherche. Quand on optimise un index, cette valeur est ignorée.</p> <p>Voir également org.apache.lucene.index.LogDocMergePolicy.maxMergeSize.</p>	Illimité
default. <indexname>.indexwriter.merge_max_optimize_size	<p>Contrôle la taille et la fréquence des fusions de segments.</p> <p>Les segments supérieure en taille à ceci (en Mo) ne sont pas fusionnés en segments plus importants même quand on optimise l'index (voir la configuration merge_max_size également).</p> <p>S'appliquant à org.apache.lucene.index.LogDocMergePolicy.maxMergeSizeForOptimize.</p>	Illimité
default. <indexname>.indexwriter.merge_calibrate_by_deletes	<p>Contrôle la taille et la fréquence des fusions de segments.</p> <p>Définir à false afin de ne pas considérer les documents supprimés quand on estime la stratégie de fusionnement.</p> <p>S'appliquant à org.apache.lucene.index.LogDocMergePolicy.calibrateSizeByDeletes.</p>	true
default. <indexname>.indexwriter.ram_buffer_size	<p>Contrôle le montant de RAM en Mo qui est utilisé pour les tampons de documents. Quand ils sont utilisés avec max_buffered_docs, on assiste à un vidage, selon qu'un événement a lieu avant l'autre.</p> <p>Normalement, pour augmenter la performance d'indexation, il vaut mieux vider par l'utilisation RAM et non pas le nombre de documents, et utiliser le plus grand tampon RAM possible.</p>	16 Mo

Propriété	Description	Valeur par défaut
default. <indexname>.indexwriter.term_index_interval	<p>Expert : définir l'intervalle entre les expressions indexées.</p> <p>Avec les valeurs élevées, IndexReader utilise moins de mémoire, mais l'accès au hasard aux termes est plus lent. Avec les valeurs moins élevées, IndexReader utilise plus de mémoire, et la vitesse d'accès aux termes est plus élevée. Voir la documentation Lucene pour obtenir plus d'informations.</p>	128
default. <indexname>.indexwriter.use_compound_file	<p>L'avantage d'utiliser un format de fichier composite, c'est qu'il y a moins de descripteurs de fichiers utilisés. L'inconvénient, c'est que l'indexation prend plus de temps et d'espace de disque temporaire. Vous pouvez définir ce paramètre à false en vue d'améliorer le temps d'indexation, mais vous pourriez manquer de descripteurs de fichiers si mergeFactor est trop élevé.</p> <p>Pour les paramètres booléens, utiliser "true" ou "false". La valeur par défaut pour cette option est true.</p>	true
default.enable_dirty_check	<p>Tous les changements d'entité ne nécessitent pas une mise à jour d'index Lucene. Si toutes les propriétés d'entité mises à jour (propriétés dirty) ne sont pas indexées, Hibernate Search évitera le processus de ré-indexation.</p> <p>Désactiver cette option si vous personnalisez des FieldBridge qui ont besoin d'être évoqués à chaque mise à jour d'événement (même si la propriété pour laquelle le pontage de champ est configuré n'a pas changé).</p> <p>Cette optimisation ne sera plus utilisée sur les classes qui utilisent un @ClassBridge ou un @DynamicBoost.</p> <p>Pour les paramètres booléens, utiliser "true" ou "false". La valeur par défaut pour cette option est true.</p>	true

[Rapporter un bogue](#)

23.2.6.4. Ajuster la vitesse d'indexation

Quand l'architecture le permet, conserver `default.exclusive_index_use=true` pour améliorer l'efficacité de l'écriture de l'index.

Quand vous ajustez la vitesse d'indexation, la façon de procéder conseillée consiste à se pencher tout d'abord sur le chargement de l'objet, puis à utiliser les temps obtenus comme base pour l'ajustement du processus d'indexation. Définir `blackhole` en tant que worker de back end et démarrer vos routines d'indexation. Le back end ne désactivera pas Hibernate Search: il générera les ensembles de changements à l'index, mais les ignorera au lieu de les vider dans l'index. Au lieu de définir `hibernate.search.indexing_strategy` à `manual`, en utilisant `blackhole`, vous chargerez sans doute plus de données en provenance de votre base de données car les entités associées seront indexées à nouveau également.

```
hibernate.search.[default|<indexname>].worker.backend blackhole
```



AVERTISSEMENT

Le back end `blackhole` ne doit pas être utilisé en production, sauf en tant qu'outil de diagnostic pour identifier les étranglements d'indexation.

[Rapporter un bogue](#)

23.2.6.5. Contrôle de la taille de segment

Les options suivantes configurent la taille maximum de segments créés :

- `merge_max_size`
- `merge_max_optimize_size`
- `merge_calibrate_by_deletes`

Exemple :

```
//to be fairly confident no files grow above 15MB, use:  
hibernate.search.default.indexwriter.ram_buffer_size = 10  
hibernate.search.default.indexwriter.merge_max_optimize_size = 7  
hibernate.search.default.indexwriter.merge_max_size = 7
```

Définir le paramètre `max_size` pour les opérations de fusion, à moins de la moitié de la taille de segment de limite stricte, comme la fusion des segments combine deux segments dans un segment plus large.

Un nouveau segment peut être d'une taille plus grande que prévue au départ; un segment n'est toutefois jamais créé significativement plus gros que la taille `ram_buffer_size` définie. Ce seuil est coché en tant qu'estimation.

[Rapporter un bogue](#)

23.2.7. Configuration LockFactory

Le répertoire Lucene Directory peut être configuré avec une stratégie de verrouillage personnalisée via `LockingFactory` pour chaque index géré par Hibernate Search.

Certaines stratégies de verrouillage ont besoin d'un verrou niveau système de fichiers, et peuvent être utilisées sur des indexes basés RAM. Lorsque vous utilisez cette stratégie, l'option de configuration `IndexBase` doit être spécifiée pour pointer vers un emplacement de système de fichiers dans lequel stocker les fichiers de marqueurs de verrouillage.

Pour sélectionner une fabrique de verrouillage, définissez l'option `hibernate.search.<index>.locking_strategy` à l'une des options suivantes :

- *simple*
- *native*
- *single*
- *none*

Tableau 23.7. Liste des implémentations `LockFactory` disponibles

name	Class	Description
simple	org.apache.lucene.store.SimpleFSLockFactory	<p>Implémentation sécurisée basée sur l'API Java's File, marque l'utilisation de l'index en créant un fichier de marqueurs.</p> <p>Si pour une raison ou une autre, vous deviez forcer la fermeture de votre application, vous devrez retirer ce fichier avant de démarrer à nouveau.</p>
native	org.apache.lucene.store.NativeFSLockFactory	<p>Tout comme avec simple, cela marque aussi l'utilisation de l'index en créant un fichier de marqueurs, mais celui-ci utilise des verrous de fichiers SE natifs, de façon à ce que même si la JVM est fermée, les verrous soient nettoyés.</p> <p>Cette implémentation a des problèmes bien connus avec NFS, donc évitez-la sur les partages NFS</p> <p>native est l'implémentation par défaut des fournisseurs de répertoires de filesystem, filesystem-master et filesystem-slave.</p>
single	org.apache.lucene.store.SingleInstanceLockFactory	<p>Cette usine de verrouillage n'utilise pas de marqueur de fichier mais est un verrou d'objet Java qui s'est tenu en mémoire ; par conséquent, il est possible de l'utiliser uniquement lorsque vous êtes sûr que l'index ne va pas être partagé par tout autre procédé.</p> <p>Il s'agit de l'implémentation par défaut du fournisseur de répertoire ram.</p>
none	org.apache.lucene.store.NoLockFactory	<p>Les modifications à cet index ne sont pas coordonnées par un verrou.</p>

Ce qui suit est un exemple de configuration de stratégie de verrouillage :

```
hibernate.search.default.locking_strategy = simple
hibernate.search.Animals.locking_strategy = native
hibernate.search.Books.locking_strategy =
org.custom.components.MyLockingFactory
```

[Rapporter un bogue](#)

23.2.8. Configuration pour la gestion des exceptions

Hibernate Search permet de configurer comment les exceptions sont gérées durant le processus d'indexation. Si aucune configuration n'est fournie, alors les exceptions sont consignées à la sortie du journal par défaut. Il est possible de déclarer explicitement le mécanisme de journalisation d'exception comme suit :

```
hibernate.search.error_handler = log
```

La gestion des exceptions par défaut se produit à la fois pour l'indexation synchrone et asynchrone. Hibernate Search fournit un mécanisme simple pour substituer l'implémentation de gestion des erreurs par défaut.

Afin de fournir votre propre implémentation, vous devez implémenter l'interface **ErrorHandler**, qui fournit la méthode de gérer (**cadre ErrorContext**). **ErrorContext** fournit une référence à l'instance principale de **LuceneWork**, à l'exception sous-jacente et à toute instance **LuceneWork** suivante qui n'aurait pas pu être traitée à cause de l'exception principale.

```
public interface ErrorContext {
    List<LuceneWork> getFailingOperations();
    LuceneWork getOperationAtFault();
    Throwable getThrowable();
    boolean hasErrors();
}
```

Pour enregistrer ce gestionnaire d'erreurs avec Hibernate Search, vous devez déclarer le classname complet de votre implémentation **ErrorHandler** dans les propriétés de configuration :

```
hibernate.search.error_handler = CustomerErrorHandler
```

[Rapporter un bogue](#)

23.2.9. Compatibilité de format d'index

Hibernate Search n'offre pas actuellement d'API de compatibilité rétroactive ou un outil pour faciliter le portage des applications pour les versions plus récentes. L'API utilise l'index d'écriture et de recherche Apache Lucene. Parfois, une mise à jour au format de l'index doit être effectuée. Dans ce cas, il y a une possibilité que les données doivent être ré-indexées si Lucene est incapable de lire l'ancien format.

**AVERTISSEMENT**

Sauvegarder les indexes avant de tenter de mettre à jour le format de l'index.

Hibernate Search expose la propriété de configuration `hibernate.search.lucene_version`. Cette propriété indique aux Analyzers et aux autres classes de Lucene de se conformer à leur comportement, tel qu'il est défini dans une version antérieure de Lucene. Voir aussi la classe `org.apache.lucene.util.Version` contenue dans le `lucene-core`. Si l'option n'est pas spécifiée, Hibernate Search demande à Lucene d'utiliser la valeur par défaut de la version. Il est recommandé que la version utilisée soit définie explicitement dans la configuration pour empêcher les changements automatiques lorsqu'une mise à niveau se produit. Après une mise à niveau, les valeurs de configuration peuvent être mises à jour explicitement si nécessaire.

Exemple 23.10. Forcer les Analyzers à être compatibles avec un index Lucene 3.0 créé

```
hibernate.search.lucene_version = LUCENE_30
```

La configuration `SearchFactory` est globale et affecte toutes les APIs de Lucene contenant le paramètre pertinent. Si Lucene est utilisé et Hibernate Search est contourné, appliquer-lui la même valeur pour obtenir des résultats uniformes.

[Rapporter un bogue](#)

23.2.10. Désactiver Hibernate Search

Hibernate Search peut être partiellement ou complètement désactivée selon les besoins. L'indexation d'Hibernate Search peut être désactivée, comme par exemple, dans le cas où l'index est en lecture-seule, ou si vous préférez faire d'index manuellement, plutôt qu'automatiquement. Il est aussi possible de désactiver complètement Hibernate Search, ce qui empêche les indexations et les recherches.

Désactivation de l'indexation

Pour désactiver Hibernate Search, définir l'option de configuration `indexing_strategy` à `manual`, puis redémarrez JBoss EAP.

```
hibernate.search.indexing_strategy = manual
```

Désactiver Hibernate Search totalement

Pour désactiver Hibernate Search totalement, désactiver tous les listeners en modifiant l'option de configuration `autoregister_listeners` à `false`, puis redémarrez JBoss EAP.

```
hibernate.search.autoregister_listeners = false
```

[Rapporter un bogue](#)

23.3. MONITORING

23.3.1. Monitoring

Hibernate Search offre un accès à un objet `statistics` via `SearchFactory.getStatistics()`. Il vous permet, par exemple, de déterminer quelles classes sont indexées et combien d'entités sont dans l'index. Cette information est toujours disponible. Cependant, en spécifiant la propriété `hibernate.search.generate_statistics` dans votre configuration, vous pouvez également cumuler les moyennes et les totaux de durées de chargement des objets et des requêtes Lucene.

Hibernate Search fournit plusieurs méthodes permettant de contrôler ses activités. La liste des catégories indexées et le nombre d'entités par index sont toujours disponibles dans l'objet `statistics` via la méthode `SearchFactory.getStatistics()`. Pour obtenir les moyennes et les totaux de durées de chargement des objets et des requêtes Lucene, spécifiez la propriété `hibernate.search.generate_statistics` dans votre configuration.

Accès aux statistiques via JMX

Pour autoriser l'accès aux statistiques via JMX, définir la propriété `hibernate.search.jmx_enabled` à `true`. Cela enregistrera le bean `StatisticsInfoMBean` automatiquement, donnant accès aux statistiques par l'objet `Statistics`. Selon votre configuration, le bean `IndexingProgressMonitorMBean` peut également être enregistré.

Contrôle de l'indexation

Si l'API d'indexeur de masse est utilisé, vous pourrez contrôler les progrès d'indexation par le bean `IndexingProgressMonitorMBean`. Le bean est lié à JMX uniquement quand l'indexation est en cours.



NOTE

On peut accéder aux beans JMX à distance par la JConsole en définissant la propriété système `com.sun.management.jmxremote` à `true`.

[Rapporter un bogue](#)

CHAPITRE 24. DÉPLOYER JBOSS EAP 6 DANS AMAZON EC2

24.1. INTRODUCTION

24.1.1. Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) est un service exploité par amazon.com qui offre aux clients un environnement informatique virtuel personnalisable. Une Image de Machine Amazon (AMI) peut être démarrée en utilisant le service pour créer une instance ou une machine virtuelle. Les utilisateurs peuvent installer n'importe quel logiciel dont ils ont besoin sur une instance et sont facturés en fonction de l'usage. Amazon EC2 est conçu pour être flexible et permettre aux utilisateurs de déployer rapidement leurs applications.

Vous pourrez en savoir davantage sur le site web Amazon EC2, <http://aws.amazon.com/ec2/>.

[Rapporter un bogue](#)

24.1.2. Amazon Machine Instances (AMIs)

Une Amazon Machine Image (AMI) est un modèle d'instance de machine virtuelle EC2. Les utilisateurs créent des instances EC2 en sélectionnant une AMI appropriée pour créer l'instance. La composante primaire d'une AMI est un système de fichiers en lecture seule qui contient un système d'exploitation installé, mais aussi des autres logiciels. Chaque AMI a différents logiciels installés pour des cas d'utilisation différents. Amazon EC2 comprend beaucoup d'AMIs au choix offerts par amazon.com et des tierces parties. Les utilisateurs peuvent également créer leurs propres AMIs personnalisées.

[Rapporter un bogue](#)

24.1.3. JBoss Cloud Access

JBoss Cloud Access est une fonctionnalité de Red Hat qui fournit un support à JBoss EAP 6 aux fournisseurs cloud certifiés Red Hat comme Amazon EC2. JBoss Cloud Access vous permet de déplacer vos abonnements entre les serveurs traditionnels et les ressources publiques basées-cloud d'une façon simple et peu coûteuse.

Vous trouverez des informations supplémentaires à l'adresse suivante <http://www.redhat.com/solutions/cloud/access/jboss/>.

[Rapporter un bogue](#)

24.1.4. Fonctionnalités de JBoss Cloud Access

L'abonnement au programme JBoss Cloud Access donne accès aux AMI (Amazon Machine Images) privées créées par Red Hat.

Les AMI de Red Hat ont le logiciel suivant pré-installé et complètement pris en charge par Red Hat :

- Red Hat Enterprise Linux 6
- JBoss EAP 6
- L'agent JBoss Operations Network (JON) 3

- Mises à jour de produit par les RPM par l'intermédiaire de l'infrastructure de mise à jour de Red Hat.

Chaque AMI de Red Hat n'est qu'un point de départ, qui requiert une configuration supplémentaire pour se conformer aux besoins de votre application.



IMPORTANT

JBoss Cloud Access n'apporte pas actuellement de support au profil full-ha, ni pour les instances standalone, ni pour les domaines gérés.

[Rapporter un bogue](#)

24.1.5. Types d'instances Amazon EC2 prises en charge

JBoss Cloud Access prend en charge les types d'instance Amazon EC2 suivantes. Voir *Amazon EC2 User Guide* pour obtenir davantage de détails sur chaque type d'instance, <http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/instance-types.html>.

Tableau 24.1. Types d'instances Amazon EC2 prises en charge

Type d'instance	Description
Instance standard	Les instances standard sont des environnements d'ordre général ayant un ratio de mémoire-à-CPU équilibré.
Instance de mémoire élevée	Les instances de mémoire élevée possèdent davantage de mémoire allouée que les instances standards. Les instances de mémoire élevée conviennent aux applications à haut débit telles que les bases de données ou les applications de mise en cache de mémoire.
Instance Haut CPU	Les instance Haut CPU ont davantage de ressources CPU allouées que de mémoire et conviennent à des débits moindres et à des applications intensives en CPU.



IMPORTANT

Le type d'instance **Micro** (`t1.micro`) ne convient pas au déploiement de la plateforme JBoss EAP 6.

[Rapporter un bogue](#)

24.1.6. Les AMI Red Hat prises en charge

Les AMI Red Hat prises en charge peuvent être identifiées par leur nom AMI.

Les AMI de JBoss EAP 6 sont composées ainsi :

```
RHEL-osversion-JBEAP-version-arch-creationdate
```

version est le nom de version de JBoss EAP installé dans l'AMI. Exemple **6.3**.

osversion est le nom de version de Red Hat Enterprise Linux installé dans l'AMI. Exemple **6.2**.

arch est l'architecture de l'AMI. Correspondra à `x86_64` ou `i386`.

creationdate est la date de création de l'AMI sous le format `YYYYMMDD`. Exemple `20120501`.

Exemple de nom d'AMI : `RHEL-6.2-JBEAP-6.0.0-x86_64-20120501`.

[Rapporter un bogue](#)

24.2. DÉPLOYER JBOSS EAP 6 DANS AMAZON EC2

24.2.1. Aperçu du déploiement de JBoss EAP 6 sur Amazon EC2

JBoss EAP 6 peut être déployé avec l'AMI Amazon EC2. L'AMI contient tout ce qui est requis pour le déploiement des instances clusterisées ou non clusterisées.

Le déploiement d'instances non clusterisées est le scénario le plus facile. Cela exige uniquement quelques changements de configuration pour spécifier votre déploiement d'application quand vous créez l'instance.

Le déploiement d'instances en cluster nécessite un peu plus de configuration. Il est recommandé de créer un Virtual Private Cloud qui puisse contenir votre cluster. Il n'est pas obligatoire d'utiliser une instance de JBoss EAP comme proxy, mais si vous prenez cette option, un compartiment S3 pour le protocole `ddiscovery` de `S3_PING` JGroups est également requis.

Chacune de ces étapes est expliquée ci-dessous mais on assume que vous avez de l'expérience avec JBoss EAP 6, Red Hat Enterprise Linux 6 et Amazon EC2.

La documentation supplémentaire suivante est recommandée :

- JBoss EAP 6
https://access.redhat.com/documentation/JBoss_Enterprise_Application_Platform/
- Red Hat Enterprise Linux 6
https://access.redhat.com/documentation/Red_Hat_Enterprise_Linux/
- Services Web Amazon
<http://aws.amazon.com/documentation/>

[Rapporter un bogue](#)

24.2.2. JBoss EAP 6 non clusterisée

24.2.2.1. Instances non-clusterisées

Une instance non clusterisée est une instance simple Amazon EC2 exécutant sur JBoss EAP 6. Elle ne fait pas partie d'un cluster.

[Rapporter un bogue](#)

24.2.2.2. Instances non clusterisées

24.2.2.2.1. Lancer l'instance de JBoss EAP 6 non clusterisée

Résumé

Ce sujet couvre les étapes requises pour lancer une instance de JBoss EAP 6 non clusterisée sur une AMI (Amazon Machine Image) Red Hat.

Conditions préalables

- Pour une Red Hat AMI qui convient, consulter [Section 24.1.6, « Les AMI Red Hat prises en charge »](#).
- Groupe de sécurité pré-configuré qui autorise les requêtes entrantes sur les ports 22, 8080, et 9990 au moins.

Procédure 24.1. Lancer une instance non clusterisée de JBoss EAP 6 sur Red Hat AMI (Amazon Machine Image).

1. Configurer le champ **User Data**. Les paramètres configurables sont disponibles ici : [Section 24.4.1, « Paramètres de configuration permanente »](#), [Section 24.4.2, « Paramètres de scripts personnalisés »](#).

Exemple 24.1. Exemple de champ de données utilisateur

L'exemple montre le champ de données utilisateur d'une instance JBoss EAP 6 non clusterisée. Le mot de passe de l'utilisateur `admin` a été défini à `adminpwd`.

```
JBOSSAS_ADMIN_PASSWORD=adminpwd
JBOSS_IP=0.0.0.0 #listen on all IPs and interfaces

# In production, access to these ports needs to be restricted for
security reasons
PORTS_ALLOWED="9990 9443"

cat> $USER_SCRIPT << "EOF"

# Get the application to be deployed from an Internet URL
# mkdir -p /usr/share/java/jboss-ec2-eap-applications
# wget https://<your secure storage hostname>/<path>/<app
name>.war -O /usr/share/java/jboss-ec2-eap-applications/<app
name>.war

# Create a file of CLI commands to be executed after starting the
server
cat> $USER_CLI_COMMANDS << "EOC"
# deploy /usr/share/java/jboss-ec2-eap-applications/<app name>.war
EOC

EOF
```

2. **Pour les instances de production**

Pour une instance de production, ajouter la ligne suivante sous la ligne `USER_SCRIPT` du champ **User Data** pour que les mises à jour de sécurité s'appliquent à l'amorçage.


```
yum -y update
```

**NOTE**

`yum -y update` doit être exécuté régulièrement pour appliquer les correctifs de sécurité et les améliorations.

3. Lancement de l'instance AMI Red Hat

Résultat

Une instance non clusterisée de JBoss EAP 6 a été configurée, et lancée sur une AMI Red Hat.

[Rapporter un bogue](#)

24.2.2.2.2. Déployer une application sur une instance de JBoss EAP 6 non clusterisée

Résumé

Ce sujet couvre le déploiement d'une application sur une instance de JBoss EAP 6 sur une AMI Red Hat.

1. **o Déploiement d'un exemple d'application**
Ajouter les lignes suivantes au champ **User Data** :

```
# Deploy the sample application from the local filesystem
deploy --force /usr/share/java/jboss-ec2-eap-samples/hello.war
```

Exemple 24.2. Champs de données d'utilisateur avec un exemple d'application

Cet exemple utilise l'exemple d'application fourni sur l'AMI Red Hat. Il inclut également une configuration de base d'une instance non clusterisée de JBoss EAP 6. Le mot de passe `admin` de l'utilisateur a été défini à `adminpwd`.

```
JBOSSAS_ADMIN_PASSWORD=adminpwd
JBOSS_IP=0.0.0.0 #listen on all IPs and interfaces

# In production, access to these ports needs to be restricted
for security reasons
PORTS_ALLOWED="9990 9443"

cat> $USER_SCRIPT << "EOF"

# Create a file of CLI commands to be executed after starting
the server
cat> $USER_CLI_COMMANDS << "EOC"

# Deploy the sample application from the local filesystem
deploy --force /usr/share/java/jboss-ec2-eap-samples/hello.war
EOC

EOF
```

- **Déployer une application personnalisée**

Ajouter les lignes suivantes au champ **User Data** (données utilisateur), pour configurer le nom de l'URL de l'application :

```
# Get the application to be deployed from an Internet URL
mkdir -p /usr/share/java/jboss-ec2-eap-applications
wget https://<your secure storage hostname>/<path>/<app name>.war
-o /usr/share/java/jboss-ec2-eap-applications/<app name>.war
```

Exemple 24.3. Exemple de champ de données utilisateur avec application personnalisée

Cet exemple utilise une application nommée **MyApp**, et inclut une configuration de base pour une instance JBoss EAP 6 non clusterisée. Le mot de passe **admin** de l'utilisateur a été défini à **adminpwd**.

```
JBOSAS_ADMIN_PASSWORD=adminpwd
JBOSS_IP=0.0.0.0 #listen on all IPs and interfaces

# In production, access to these ports needs to be restricted
for security reasons
PORTS_ALLOWED="9990 9443"

cat> $USER_SCRIPT << "EOF"

# Get the application to be deployed from an Internet URL
mkdir -p /usr/share/java/jboss-ec2-eap-applications
wget https://PATH_TO_MYAPP/MyApp.war -O /usr/share/java/jboss-
ec2-eap-applications/MyApp.war

# Create a file of CLI commands to be executed after starting
the server
cat> $USER_CLI_COMMANDS << "EOC"
deploy /usr/share/java/jboss-ec2-eap-applications/MyApp.war
EOC

EOF
```

2. Lancement de l'instance AMI Red Hat

Résultat

L'application a été déployée avec succès dans JBoss EAP 6.

[Rapporter un bogue](#)

24.2.2.2.3. Lancer l'instance de JBoss EAP 6 non clusterisée

Résumé

Cette rubrique couvre les étapes nécessaires pour tester la plateforme non clusterisée de JBoss EAP 6.

Procédure 24.2. Vérifier que l'instance de JBoss EAP 6 non clusterisée exécute correctement

1. Déterminer le **Public DNS** de l'instance, qui se situe dans le panneau d'informations de l'instance.
2. Naviguer dans `http://<public-DNS>:8080`.
3. Confirmer que la page d'accueil de JBoss EAP apparaît, y compris le lien vers la console admin. Si la page d'accueil n'est pas disponible, consulter : [Section 24.5.1, « Résolution de problèmes dans Amazon EC2 »](#).
4. Cliquer sur l'hyperlien **Admin Console**.
5. Se connecter :
 - o Nom d'utilisateur : `admin`
 - o Mot de passe : spécifié dans le champ **User Data** qui se trouve ici : [Section 24.2.2.1, « Lancer l'instance de JBoss EAP 6 non clusterisée »](#).
6. **Tester l'exemple d'application**
Naviguer dans `http://<public-DNS>:8080/hello` pour tester que l'exemple d'application exécute avec succès. Le texte **Hello World!** doit apparaître dans le navigateur. Si le texte n'apparaît pas, voir : [Section 24.5.1, « Résolution de problèmes dans Amazon EC2 »](#).
7. Se déconnecter de la console admin de JBoss EAP.

Résultat

L'instance de JBoss EAP 6 exécute correctement.

[Rapporter un bogue](#)

24.2.2.3. Domaines gérés non clusterisés

24.2.2.3.1. Lancer une instance pour qu'elle serve de contrôleur de domaine

Résumé

Cette rubrique couvre les étapes requises pour lancer un domaine géré non clusterisé de JBoss EAP 6 sur une AMI Red Hat (Amazon Machine Image de Red Hat)

Conditions préalables

- Pour une Red Hat AMI qui convient, consulter [Section 24.1.6, « Les AMI Red Hat prises en charge »](#).
- [Section 24.2.3.3, « Créer un VPC \(Virtual Private Cloud\) »](#)
- [Section 24.2.3.4, « Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC »](#)
- [Section 24.2.3.5, « Configurer le routage par défaut du sous-système privé VPC »](#)
- [Section 24.2.3.7, « Configurer l'installation IAM »](#)
- [Section 24.2.3.9, « Configurer l'installation S3 Bucket »](#)

Procédure 24.3. Lancer un domaine géré de JBoss EAP 6 non clusterisé sur une Red Hat AMI

1. Dans l'onglet « groupe de sécurité », veillez bien à ce que tout le trafic soit autorisé. Les capacités de pare-feu intégrées de Red Hat Enterprise Linux peuvent être utilisées pour restreindre l'accès si nécessaire.
2. Définir le sous-réseau public du VPC à *running*.
3. Sélectionner une adresse IP statique.
4. Configurer le champ **User Data**. Les paramètres configurables sont disponibles ici : [Section 24.4.1, « Paramètres de configuration permanente »](#), [Section 24.4.2, « Paramètres de scripts personnalisés »](#). Pour plus d'informations sur le contrôleur de domaine discovery d'Amazon EC2, voir [Section 24.2.2.3.4, « Configurer Domain Controller Discovery et Failover dans Amazon EC2 »](#).

Exemple 24.4. Exemple de champ de données utilisateur

L'exemple montre le champ de données utilisateur d'un domaine géré de JBoss EAP 6 non clusterisé. Le mot de passe de l'utilisateur `admin` a été défini sur `admin`.

```
## password that will be used by slave host controllers to connect
to the domain controller
JBOSSAS_ADMIN_PASSWORD=admin

## subnet prefix this machine is connected to
SUBNET=10.0.0.

## S3 domain controller discovery setup
# JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY=<your secret key>
# JBOSS_DOMAIN_S3_ACCESS_KEY=<your access key>
# JBOSS_DOMAIN_S3_BUCKET=<your bucket name>

##### to run the example no modifications below should be needed
#####
JBOSS_DOMAIN_CONTROLLER=true
PORTS_ALLOWED="9999 9990 9443"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Get the application to be deployed from an Internet URL
# mkdir -p /usr/share/java/jboss-ec2-eap-applications
# wget https://<your secure storage hostname>/<path>/<app
name>.war -O /usr/share/java/jboss-ec2-eap-applications/<app
name>.war

## Create a file of CLI commands to be executed after starting the
server
cat> $USER_CLI_COMMANDS << "EOC"

# Add the modcluster subsystem to the default profile to set up a
proxy
/profile=default/subsystem=web/connector=ajp:add(name=ajp,protocol
=AJP/1.3,scheme=http,socket-binding=ajp)
/:composite(steps=[ {"operation" => "add", "address" => [
```

```
( "profile" => "default"), ("subsystem" => "modcluster") ] }, {
"operation" => "add", "address" => [ ("profile" => "default"),
("subsystem" => "modcluster"), ("mod-cluster-config" =>
"configuration") ], "advertise" => "false", "proxy-list" =>
"${jboss.modcluster.proxyList}", "connector" => "ajp"}, {
"operation" => "add", "address" => [ ("profile" => "default"),
("subsystem" => "modcluster"), ("mod-cluster-config" =>
"configuration"), ("dynamic-load-provider" => "configuration") ]},
{ "operation" => "add", "address" => [ ("profile" => "default"),
("subsystem" => "modcluster"), ("mod-cluster-config" =>
"configuration"), ("dynamic-load-provider" => "configuration"),
("load-metric" => "busyness")], "type" => "busyness" } ]

# Deploy the sample application from the local filesystem
deploy /usr/share/java/jboss-ec2-eap-samples/hello.war --server-
groups=main-server-group
EOF

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e ">:::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
    echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF
```

5. Pour les instances de production

Pour une instance de production, ajouter la ligne suivante sous la ligne **USER_SCRIPT** du champ **User Data** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

```
yum -y update
```



NOTE

yum -y update doit être exécuté régulièrement pour appliquer les correctifs de sécurité et les améliorations.

6. Lancement de l'instance Red Hat AMI

Résultat

Un domaine géré non clusterisé de JBoss EAP 6 a été configuré, et lancé sur une Red Hat AMI.

[Rapporter un bogue](#)

24.2.2.3.2. Lancer une ou plusieurs instances pour qu'elles servent de contrôleurs hôtes

Résumé

Cette rubrique couvre les étapes requises pour lancer une ou plusieurs instances de JBoss EAP 6 en tant que contrôleurs hôtes non clusterisée sur Red Hat AMI (Amazon Machine Image de Red Hat).

Conditions préalables

- Configurer et lancer le contrôleur de domaine non clusterisé. Consulter [Section 24.2.2.3.1, « Lancer une instance pour qu'elle serve de contrôleur de domaine »](#).
- [Section 24.2.3.7, « Configurer l'installation IAM »](#)
- [Section 24.2.3.9, « Configurer l'installation S3 Bucket »](#)

Procédure 24.4. Lancer les contrôleurs hôtes

Pour chaque instance que vous souhaitez créer, répétez les étapes suivantes :

1. Sélectionner une AMI.
2. Définir le nombre d'instances que vous souhaitez (le nombre de contrôleurs hôtes esclaves)
3. Sélectionner le VPC et le type d'instance.
4. Cliquer sur le groupe de sécurité.
5. Veillez à ce que tout le trafic en provenance du sous-système de JBoss EAP 6 soit autorisé.
6. Définir les autres restrictions suivant les besoins.
7. Ajouter ce qui suit dans le champ User Data :

```
## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## host controller setup
### static domain controller discovery setup
JBOSS_DOMAIN_MASTER_ADDRESS=10.0.0.5
### S3 domain controller discovery setup
# JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY=<your secret key>
# JBOSS_DOMAIN_S3_ACCESS_KEY=<your access key>
# JBOSS_DOMAIN_S3_BUCKET=<your bucket name>

JBOSS_HOST_PASSWORD=<password for slave host controllers>

## subnet prefix this machine is connected to
SUBNET=10.0.1.

##### to run the example no modifications below should be needed #####
JBOSS_HOST_USERNAME=admin
PORTS_ALLOWED="1024:65535"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Server instance configuration
sed -i "s/other-server-group/main-server-group/"
$JBOSS_CONFIG_DIR/$JBOSS_HOST_CONFIG

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
```

```

echo -e " :::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
    echo -e "$SUBNET$i\tip-{$SUBNET//./-}$i" ;
done >> /etc/hosts

EOF

```

Pour plus d'informations sur le contrôleur de domaine discovery d'Amazon EC2, voir [Section 24.2.2.3.4, « Configurer Domain Controller Discovery et Failover dans Amazon EC2 »](#) .

8. Pour les instances de production

Pour une instance de production, ajouter la ligne suivante sous la ligne `USER_SCRIPT` du champ `User Data` pour que les mises à jour de sécurité s'appliquent à l'amorçage.

```
yum -y update
```



NOTE

`yum -y update` doit être exécuté régulièrement pour appliquer les correctifs de sécurité et les améliorations.

9. Lancement de l'instance Red Hat AMI

Résultat

Les contrôleurs hôtes non clusterisés de JBoss EAP 6 ont été configurés, et lancés sur une Red Hat AMI.

[Rapporter un bogue](#)

24.2.2.3.3. Tester le domaine géré de JBoss EAP 6 non clusterisée

Résumé

Cette rubrique couvre les étapes requises pour tester le domaine géré non clusterisé de JBoss EAP 6 sur une Red Hat AMI (Amazon Machine Image de Red Hat)

Pour tester le domaine géré, vous devrez connaître les adresses IP élastiques de Apache HTTP et du contrôleur de domaines de JBoss EAP 6 à la fois.

Conditions préalables

- Configurer et lancer le contrôleur de domaines. Consulter [Section 24.2.2.3.1, « Lancer une instance pour qu'elle serve de contrôleur de domaine »](#) .
- Configurer et lancer les contrôleur hôtes. Consulter [Section 24.2.2.3.2, « Lancer une ou plusieurs instances pour qu'elles servent de contrôleurs hôtes »](#) .

Procédure 24.5. Tester le Serveur Web

- Naviguer dans `http://ELASTIC_IP_OF_APACHE_HTTPD` avec un navigateur pour confirmer que le serveur web exécute avec succès.

Procédure 24.6. Tester le contrôleur de domaine

1. Naviguer dans `http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console`
2. Connectez-vous en utilisant le nom d'utilisateur `admin` et le mot de passe spécifiés dans le champ « données d'utilisateur » pour le contrôleur de domaine et la page d'accueil de la console admin du domaine géré s'affichera (`http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console/App.html#server-instances`).
3. Cliquer sur l'étiquette **Server** du serveur en haut et à droite de l'écran, et sélectionner un contrôleur hôte dans le menu déroulant **Host** en haut et à gauche de l'écran.
4. Vérifier que chaque contrôleur hôte ait deux configurations de serveurs nommées respectivement `server-one` et `server-two` qui appartiennent toutes deux au `main-server-group`.
5. Se déconnecter de la console admin de JBoss EAP.

Procédure 24.7. Tester les contrôleur hôte

1. Naviguer dans `http://ELASTIC_IP_OF_APACHE_HTTPD/hello` pour tester que l'exemple d'application exécute. Le texte **Hello World!** devrait apparaître dans le navigateur.

Si le texte n'est pas visible, voir : Section 18.5.1, "About Troubleshooting Amazon EC2".

2. Connectez-vous à l'instance Apache HTTP :

```
$ ssh -L7654:localhost:7654 ELASTIC_IP_OF_APACHE_HTTPD
```

3. Naviguer dans `http://localhost:7654/mod_cluster-manager` pour confirmer que toutes les instances exécutent correctement.

Résultat

Le serveur web de JBoss EAP 6, le contrôleur de domaine, et les contrôleurs hôtes exécutent correctement sur une Red Hat AMI.

[Rapporter un bogue](#)

24.2.2.3.4. Configurer Domain Controller Discovery et Failover dans Amazon EC2

Pour un domaine géré s'exécutant sur Amazon EC2, en plus de la découverte de contrôleur de domaine statique, les contrôleurs hôtes peuvent découvrir dynamiquement un contrôleur de domaine en utilisant le système de stockage d'Amazon S3. En particulier, les contrôleurs hôtes et le contrôleur de domaine peuvent être configurés avec les informations nécessaires pour accéder à un package Amazon S3.

En utilisant cette configuration, lorsqu'un contrôleur de domaine est démarré, il écrit ses coordonnées dans un fichier S3 dans le package. Chaque fois qu'un contrôleur hôte tente de contacter le contrôleur de domaine, il obtient des informations de contact du contrôleur de domaine du fichier S3.

Cela signifie que si les coordonnées du contrôleur de domaine changent (par exemple, il est fréquent que l'adresse IP de l'instance EC2 change quand il est arrêté et démarré), les contrôleurs hôtes n'ont pas besoin d'être reconfigurés. Les contrôleurs hôtes sont en mesure d'obtenir les nouvelles coordonnées du contrôleur de domaine dans le fichier S3.

Vous pouvez activer Domain Controller Discovery en passant les paramètres

`JBOSS_DOMAIN_S3_ACCESS_KEY`, `JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY`, et `JBOSS_DOMAIN_S3_BUCKET` à l'instance de JBoss EAP 6 quand vous la lancez. Voir [Section 24.4.1, « Paramètres de configuration permanente »](#) pour voir les paramètres configurables. Sinon, vous pouvez configurer Domain Discovery manuellement par la configuration suivante.

La configuration manuelle de Domain Controller Discovery est spécifiée par les propriétés suivantes :

access-key

La clé d'accès au compte utilisateur Amazon AWS

secret-access-key

La clé d'accès secrète au compte utilisateur Amazon AWS

location

Le package Amazon S3 à utiliser

Voici des exemples de configuration de contrôleurs hôtes et de contrôleurs de domaines. Bien qu'une option discovery soit illustrée dans les exemples ci-dessous, il est possible de configurer un nombre quelconque de discovery statiques ou d'options discovery S3. Pour plus d'informations sur le processus discovery et le processus de basculement du domaine, voir [Section 1.7, « Domain Controller Discovery et Failover »](#).

Exemple 24.5. Configuration du contrôleur hôte

```
<domain-controller>
  <remote security-realm="ManagementRealm">
    <discovery-options>
      <discovery-option name="s3-discovery"
code="org.jboss.as.host.controller.discovery.S3Discovery"
module="org.jboss.as.host-controller">
        <property name="access-key" value="S3_ACCESS_KEY"/>
        <property name="secret-access-key"
value="S3_SECRET_ACCESS_KEY"/>
        <property name="location" value="S3_BUCKET_NAME"/>
      </discovery-option>
    </discovery-options>
  </remote>
</domain-controller>
```

Exemple 24.6. Configuration du contrôleur de domaine

```
<domain-controller>
  <local>
    <discovery-options>
      <discovery-option name="s3-discovery"
code="org.jboss.as.host.controller.discovery.S3Discovery"
module="org.jboss.as.host-controller">
        <property name="access-key" value="S3_ACCESS_KEY"/>
        <property name="secret-access-key"
value="S3_SECRET_ACCESS_KEY"/>
        <property name="location" value="S3_BUCKET_NAME"/>
      </discovery-option>
    </discovery-options>
  </local>
</domain-controller>
```

```
</discovery-option>
</discovery-options>
</local>
</domain-controller>
```

[Rapporter un bogue](#)

24.2.3. JBoss EAP 6 clusterisé

24.2.3.1. Instances clusterisées

Une instance clusterisée est une instance Amazon EC2 exécutant sur JBoss EAP 6 avec le clustering activé. Une autre instance exécutant sur le serveur Apache HTTP agira en tant que proxy pour les instances dans le cluster.

Les AMI de JBoss EAP 6 comprennent deux fichiers de configuration à utiliser dans les instances en cluster, `standalone-ec2-ha.xml` et `standalone-mod_cluster-ec2-ha.xml`. Chacun de ces fichiers de configuration fournit du clustering sans multidiffusion car Amazon EC2 ne prend pas en charge la multidiffusion. Cela se fait par monodiffusion TCP pour les communications de clusters et S3_PING comme protocole de découverte. La configuration `autonome-mod_cluster-ec2-ha.xml` fournit également un enregistrement simple par les proxys de `mod_cluster`.

De même, le fichier de configuration `domain-ec2.xml` fournit deux profils à utiliser dans les domaines gérés clusterisés : `ec2-ha`, et `mod_cluster-ec2-ha`.

[Rapporter un bogue](#)

24.2.3.2. Clouds privés virtuels

Amazon VPC (Amazon Virtual Private Cloud) est une fonctionnalité d'AWS (Amazon Web Service) qui vous permet d'isoler un ensemble de ressources AWS dans un réseau privé. La topologie et la configuration de ce réseau privé peuvent être personnalisées.

Voir le site Amazon Virtual Private Cloud pour obtenir plus d'informations
<http://aws.amazon.com/vpc/>.

[Rapporter un bogue](#)

24.2.3.3. Créer un VPC (Virtual Private Cloud)

Résumé

Cette rubrique décrit les étapes requises pour créer un cloud privé virtuel, en prenant comme exemple une base de données externe au VPC. Vos stratégies de sécurité peuvent exiger la connexion à la base de données à crypter. Veuillez consulter *RDS FAQ* d'Amazon pour plus d'informations sur le cryptage des connexions de base de données.



IMPORTANT

Un VPC est recommandé pour une installation de cluster dans JBoss EAP 6 car cela simplifie grandement une communication sécurisée entre les nœuds du cluster, un Server JON et le proxy mod_cluster. Sans un VPC, ces canaux de communication doivent être chiffrés et authentifiés.

Pour obtenir des instructions sur la façon de configurer SSL, consulter le guide *Core Management Security Guide*.

1. Aller dans l'onglet VPC de la console AWS.
2. Abonnez-vous au service si nécessaire.
3. Cliquer sur "Create new VPC".
4. Sélectionner un VPC avec un sous-système public et un privé.
 - a. Définir le sous-système public à 10.0.0.0/24.
 - b. Définir le sous-système privé à 10.0.1.0/24.
5. Aller dans **Elastic IPs**.
6. Créer un IP élastique pour que l'instance mod_cluster proxy/NAT puisse l'utiliser.
7. Aller dans **Security groups** et créer un groupe de sécurité pour autoriser le trafic entrant et sortant.
8. Aller sur les ACL de réseau
 - a. Créer un ACL pour autoriser le trafic entrant et sortant.
 - b. Créer un ACL pour autoriser le trafic vers et depuis les ports TCP 22, 8009, 8080, 8443, 9443, 9990 et 16163 uniquement.

Résultat

Le Cloud privé virtuel (VPC) a été créé.

[Rapporter un bogue](#)

24.2.3.4. Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC

Résumé

Cette section couvre toutes les étapes requises pour lancer une instance de serveur Apache HTTP qui puisse servir de proxy mod_cluster et d'instance NAT au Virtuel Private Cloud.

Conditions préalables

- [Section 24.2.3.3, « Créer un VPC \(Virtual Private Cloud\) »](#)

Procédure 24.8. Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC

1. Créer un IP élastique pour cette instance.
2. Sélectionner une AMI.
3. Allez dans le **Security Group** et autoriser tout le trafic (utiliser les capacités de pare-feu intégrées de Red Hat Enterprise Linux pour restreindre l'accès si nécessaire).
4. Choisir "running" dans le sous-système public du VPC.
5. Sélectionner un IP statique (comme par ex 10.0.0.4).
6. Mettez ce qui suit dans le champ **User Data**:

```
JBOSSCONF=disabled

cat > $USER_SCRIPT << "EOS"

echo 1 > /proc/sys/net/ipv4/ip_forward
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
echo 0 > /proc/sys/net/ipv4/conf/eth0/rp_filter

iptables -I INPUT 4 -s 10.0.1.0/24 -p tcp --dport 7654 -j ACCEPT
iptables -I INPUT 4 -p tcp --dport 80 -j ACCEPT

iptables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -I FORWARD -s 10.0.1.0/24 -j ACCEPT
iptables -t nat -A POSTROUTING -o eth0 ! -s 10.0.0.4 -j MASQUERADE

# balancer module incompatible with mod_cluster
sed -i -e 's/LoadModule proxy_balancer_module/#\0/'
/etc/httpd/conf/httpd.conf

cat > /etc/httpd/conf.d/mod_cluster.conf << "EOF"
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module modules/mod_advertise.so

Listen 7654

# workaround JBPAPP-4557
MemManagerFile /var/cache/mod_proxy/manager

<VirtualHost *:7654>
  <Location /mod_cluster-manager>
    SetHandler mod_cluster-manager
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
  </Location>

  <Location />
    Order deny,allow
    Deny from all
```

```

        Allow from 10.
        Allow from 127.0.0.1
    </Location>

    KeepAliveTimeout 60
    MaxKeepAliveRequests 0
    ManagerBalancerName mycluster
    ServerAdvertise Off
    EnableMCPMReceive
</VirtualHost>
EOF

echo "`hostname | sed -e 's/ip-//' -e 'y/-/./'`          `hostname`"
>> /etc/hosts

semanage port -a -t http_port_t -p tcp 7654 #add port in the apache
port list for the below to work
setsebool -P httpd_can_network_relay 1 #for mod_proxy_cluster to
work
chcon -t httpd_config_t -u system_u
/etc/httpd/conf.d/mod_cluster.conf

##### Uncomment the following line when launching a managed domain
#####
# setsebool -P httpd_can_network_connect 1

service httpd start

EOS

```

7. Décochez la case Amazon EC2 cloud source/destination pour cette instance pour qu'elle puisse agir en tant que router.
 - a. Cliquer à droite sur l'instance de serveur Apache HTTP et sélectionner "**Change Source/Dest check**".
 - b. Cliquer sur **Yes, Disable**.
8. Créer un IP élastique pour cette instance.

Résultat

L'instance de serveur Apache HTTP aura été lancée avec succès.

[Rapporter un bogue](#)

24.2.3.5. Configurer le routage par défaut du sous-système privé VPC

Résumé

Cette rubrique couvre les étapes requises pour configurer le routage par défaut du sous-système privé VPC. Les noeuds de cluster de JBoss EAP 6 exécuteront dans le sous-système privé du VPC, mais les noeuds de cluster ont besoin d'un accès internet pour la connectivité S3. Un routage par défaut doit être défini pour aller dans l'instance NAT.

Procédure 24.9. Configurer le routage par défaut du sous-système privé VPC

1. Naviguer dans l'instance de serveur Apache HTTP de la console Amazon AWS.
2. Naviguer dans **VPC** → **tables de routage**.
3. Cliquer sur le tableau de routage utilisé par le sous-système privé.
4. Dans le champ de nouveau routage, saisir **0 . 0 . 0 . 0 / 0**.
5. Cliquer sur **"Select a target"**.
6. Sélectionner **"Enter Instance ID"**.
7. Choisir l'ID de l'instance du serveur Apache HTTP en cours d'exécution.

Résultat

Le routage par défaut a été configuré correctement pour le sous-système VPC.

[Rapporter un bogue](#)

24.2.3.6. IAM (Identity and Access Management)

IAM (Identity and Access Management) fournit une sécurité configurable pour vos ressources AWS. IAM peut être configuré pour utiliser les comptes créés dans IAM ou pour fournir une fédération d'identité entre IAM et vos propres services d'identité.

Consultez le site web AWS Identity and Access Management pour plus d'informations <http://aws.amazon.com/iam/>.

[Rapporter un bogue](#)

24.2.3.7. Configurer l'installation IAM

Résumé

Cette rubrique couvre les étapes de configuration requises pour installer IAM pour les instances de JBoss EAP 6. Le protocole **S3_PING** utilise le compartiment S3 pour découvrir d'autres membres du cluster. La version 3.0.x de **JGroups** a besoin d'un compte d'accès Amazon AWS et de clés secrètes pour s'authentifier dans le service S3.

Comme la découverte de contrôleur de domaine S3 utilise le compartiment S3, il a besoin d'un accès à un compte Amazon AWS et à des clés secrètes pour authentifier le service S3 (similaire au protocole **S3_PING** utilisé par JGroups). L'utilisateur IAM et le compartiment S3 utilisés pour la découverte de S3 doivent être différents de l'utilisateur IAM et du compartiment S3 utilisés pour le clustering.

Il y a un risque de sécurité à entrer vos informations d'identification du compte principal dans le domaine de l'utilisateur des données, de les stocker en ligne ou dans une AMI. Pour contourner cela, un compte distinct peut être créé en utilisant la fonction Amazon IAM qui donnerait accès qu'à un seul compartiment de S3.

Procédure 24.10. Configurer l'installation IAM

1. Aller dans l'onglet IAM de la console AWS.
2. Cliquer sur **users** (utilisateurs).
3. Sélectionner **Create New Users** (Créer Nouveaux Utilisateurs).

4. Choisir un nom, et veillez à ce que l'option **Generate an access key for each User** (Générer une clé d'accès pour chaque utilisateur) soit cochée.
5. Sélectionner **Download credentials**, et les sauvegarder dans un emplacement sécurisé.
6. Fermer la fenêtre.
7. Cliquer sur un utilisateur nouvellement créé.
8. Prenez note de la valeur de **User ARM**. Cette valeur est requise pour configurer Bucket S3, et est documentée ici: [Section 24.2.3.9, « Configurer l'installation S3 Bucket »](#).

Résultat

Le compte IAM a été créé avec succès.

[Rapporter un bogue](#)

24.2.3.8. S3 Bucket

Les compartiments S3 représentent une unité de stockage d'organisation de base d'Amazon S3 (Amazon Simple Storage System). Un compartiment peut stocker un certain nombre d'objets arbitraires et doit posséder un nom unique pour l'identifier avec Amazon S3.

Consulter le site web Amazon S3 pour obtenir plus d'informations, <http://aws.amazon.com/s3/>.

[Rapporter un bogue](#)

24.2.3.9. Configurer l'installation S3 Bucket

Résumé

Cette rubrique couvre les étapes nécessaires de configuration d'un nouveau compartiment S3.

Conditions préalables

- [Section 24.2.3.7, « Configurer l'installation IAM »](#).

Procédure 24.11. Configurer l'installation S3 Bucket

1. Ouvrir l'onglet **S3** dans la console AWS.
2. Cliquer sur **Créer Bucket**.
3. Choisir un nom pour le compartiment et cliquer sur **Create** (Créer).



NOTE

Les noms de compartiments sont uniques dans tout S3. Les noms ne peuvent pas être réutilisés.

4. Cliquer à droite sur le nouveau compartiment et sélectionner **Properties** (propriétés).
5. Cliquer sur **Add bucket policy** (ajouter la règle de compartiment) dans l'onglet de permissions.

6. Cliquer sur **New policy** (Nouvelle police) pour ouvrir l'assistant de création de police.

- a. Copier le texte suivant dans la nouvelle police, en remplaçant `arn:aws:iam::055555555555:user/jbosscluster*` par la valeur définie ici : [Section 24.2.3.7, « Configurer l'installation IAM »](#). Changer les deux instances de `clusterbucket123` au nom de compartiment défini dans l'étape 3 de cette procédure.

```
{
  "Version": "2008-10-17",
  "Id": "Policy1312228794320",
  "Statement": [
    {
      "Sid": "Stmt1312228781799",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::055555555555:user/jbosscluster"
        ]
      },
      "Action": [
        "s3:ListBucketVersions",
        "s3:GetObjectVersion",
        "s3:ListBucket",
        "s3:PutBucketVersioning",
        "s3>DeleteObject",
        "s3>DeleteObjectVersion",
        "s3:GetObject",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:PutObject",
        "s3:GetBucketVersioning"
      ],
      "Resource": [
        "arn:aws:s3:::clusterbucket123/*",
        "arn:aws:s3:::clusterbucket123"
      ]
    }
  ]
}
```

Résultat

Un nouveau compartiment a maintenant été créé, et configuré.

[Rapporter un bogue](#)

24.2.3.10. Instances clusterisées

24.2.3.10.1. Lancer les AMI de JBoss EAP 6 clusterisée

Résumé

Cette rubrique couvre les étapes requises pour lancer les AMI JBoss EAP 6.

Conditions préalables

- [Section 24.2.3.3, « Créer un VPC \(Virtual Private Cloud\) »](#).
- [Section 24.2.3.4, « Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC »](#).
- [Section 24.2.3.5, « Configurer le routage par défaut du sous-système privé VPC »](#).
- [Section 24.2.3.7, « Configurer l'installation IAM »](#).
- [Section 24.2.3.9, « Configurer l'installation S3 Bucket »](#).



AVERTISSEMENT

Exécuter le cluster JBoss EAP 6 dans un sous-système avec un masque de réseau inférieur à 24 bits ou bien fractionner plusieurs sous-systèmes compliquent l'obtention d'un ID homologue unique pour chaque membre du cluster.

Voir la variable `JBOSS_CLUSTER_ID` pour obtenir des informations sur la façon d'effectuer ce travail de configuration de façon fiable : [Section 24.4.1, « Paramètres de configuration permanente »](#).



IMPORTANT

La fonctionnalité de Amazon EC2 AutoScaling peut être utilisée avec des nœuds de cluster JBoss EAP 6. Cependant, s'assurer que c'est testé avant le déploiement. Vous devez vous assurer que vos charges de travail particulières sont à l'échelle du nombre de nœuds désirés, et que la performance répond à vos besoins avant d'envisager d'utiliser le type d'instance (des types d'instances différentes reçoivent une part de ressources cloud EC2 différentes).

De plus, l'emplacement de l'instance et l'utilisation machine/RDS réseau/stockage/machine hôte/RDS peuvent affecter la performance d'un cluster. Tester avec vos charges réelles attendues pour essayer de pallier à l'avance aux conditions inattendues.



AVERTISSEMENT

L'action *scale-down* Amazon EC2 termine les nœuds sans élégance, et, comme certaines transactions pourraient être interrompues, les autres nœuds de cluster (et équilibres de charge) auront besoin de temps pour basculer. Cela est susceptible d'influencer l'expérience des utilisateurs de votre application.

Il est recommandé que vous réduisiez le cluster d'applications manuellement en désactivant le serveur de l'interface de gestion du `mod_cluster` jusqu'à ce que toutes les sessions traitées soient complétées, ou bien que vous fermiez l'instance JBoss EAP 6 avec grâce (on peut utiliser l'accès SSH vers l'instance ou JON).

Vérifier que votre procédure de réduction choisie n'ait pas d'effets négatifs sur l'expérience utilisateur. Il est possible que vous ayez besoin de prendre des mesures supplémentaires pour certaines charges de travail, équilibrages de charge ou installations.

Procédure 24.12. Lancer les AMI de JBoss EAP 6 clusterisée

1. Sélectionner une AMI.
2. Définir le nombre d'instances voulues (la taille du cluster).
3. Sélectionner le VPC et le type d'instance.
4. Cliquer sur le groupe **Security Group**.
5. Veillez à ce que tout le trafic en provenance du cluster JBoss EAP 6 soit autorisé.
6. Définir les autres restrictions suivant les besoins.
7. Ajouter ce qui suit dans le champ **User Data** :

Exemple 24.7. Exemple de champ de données utilisateur

```
## mod_cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## clustering setup
JBOSS_JGROUPS_S3_PING_SECRET_ACCESS_KEY=<your secret key>
JBOSS_JGROUPS_S3_PING_ACCESS_KEY=<your access key>
JBOSS_JGROUPS_S3_PING_BUCKET=<your bucket name>

## password to access admin console
JBOSSAS_ADMIN_PASSWORD=<your password for opening admin console>

## database credentials configuration
JAVA_OPTS="$JAVA_OPTS -
Ddb.host=instancename.something.rds.amazonaws.com -
Ddb.database=mydatabase -Ddb.user=<user> -Ddb.passwd=<pass>"
```

```

## subnet prefix this machine is connected to
SUBNET=10.0.1.

#### to run the example no modifications below should be needed
####
PORTS_ALLOWED="1024:65535"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Get the application to be deployed from an Internet URL
# mkdir -p /usr/share/java/jboss-ec2-eap-applications
# wget https://<your secure storage hostname>/<path>/<app
name>.war -O /usr/share/java/jboss-ec2-eap-applications/<app
name>.war

## install the JDBC driver as a core module
yum -y install mysql-connector-java
mkdir -p /usr/share/jbossas/modules/com/mysql/main
cp -v /usr/share/java/mysql-connector-java-*.jar
/usr/share/jbossas/modules/com/mysql/main/mysql-connector-java.jar

cat > /usr/share/jbossas/modules/com/mysql/main/module.xml <<"EOM"
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
</module>
EOM

cat > $USER_CLI_COMMANDS << "EOC"
## Deploy sample application from local filesystem
deploy --force /usr/share/java/jboss-ec2-eap-samples/cluster-
demo.war

## ExampleDS configuration for MySQL database
data-source remove --name=ExampleDS
/subsystem=datasources/jdbc-driver=mysql:add(driver-
name="mysql",driver-module-name="com.mysql")
data-source add --name=ExampleDS --connection-
url="jdbc:mysql://${db.host}:3306/${db.database}" --jndi-
name=java:jboss/datasources/ExampleDS --driver-name=mysql --user-
name="${db.user}" --password="${db.passwd}"
/subsystem=datasources/data-source=ExampleDS:enable
/subsystem=datasources/data-source=ExampleDS:test-connection-in-
pool
EOC

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e ">:::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts

```

```

for (( i=1 ; i<255 ; i++ )); do
    echo -e "$SUBNET$i\tip-`${SUBNET//./-}`$i" ;
done >> /etc/hosts

EOF

```

Résultat

Les AMI JBoss EAP 6 clusterisés ont été configurés et lancés avec succès.

[Rapporter un bogue](#)

24.2.3.10.2. Lancer l'instance de JBoss EAP 6 clusterisée

Résumé

Cette rubrique couvre les étapes pour s'assurer que les instances de la plateforme clusterisée de JBoss EAP 6 exécutent correctement.

Procédure 24.13. Tester l'instance clusterisée

1. Naviguer dans http://ELASTIC_IP_OF_APACHE_HTTPD pour confirmer que le serveur web exécute correctement.
2. **Tester les noeuds clusterisés**
 - a. Naviguer dans http://ELASTIC_IP_OF_APACHE_HTTPD/cluster-demo/put.jsp à l'aide d'un navigateur.
 - b. Vérifier que l'un des noeuds de cluster journalise le message suivant :


```
Putting date now
```
 - c. Stopper le noeud de cluster qui a journalisé le message dans l'étape précédente.
 - d. Naviguer dans http://ELASTIC_IP_OF_APACHE_HTTPD/cluster-demo/get.jsp à l'aide d'un navigateur.
 - e. Vérifier que l'heure indiquée est la même que l'heure PUT (mise) par `put.jsp` à l'étape 2-a.
 - f. Vérifier que l'un des noeuds de cluster en cours d'exécution journalise le message suivant :


```
Getting date now
```
 - g. Démarrer à nouveau le noeud clusterisé qui est arrêté.
 - h. Connectez-vous à l'instance Apache HTTP :


```
ssh -L7654:localhost:7654 <ELASTIC_IP_OF_APACHE_HTTPD>
```
 - i. Naviguer dans http://localhost:7654/mod_cluster-manager pour confirmer que toutes les instances exécutent correctement.

Résultat

Les instances clusterisées de JBoss EAP 6 ont été testées, et il a été confirmé qu'elle exécutait correctement.

[Rapporter un bogue](#)

24.2.3.11. Domaines gérés clusterisés

24.2.3.11.1. Lancer une instance pour qu'elle serve de contrôleur de domaine de cluster

Résumé

Cette rubrique couvre les étapes requises pour lancer un domaine géré clusterisé de JBoss EAP 6 sur une Red Hat AMI (Amazon Machine Image de Red Hat)

Conditions préalables

- Une Red Hat AMI qui convient. Voir [Section 24.1.6, « Les AMI Red Hat prises en charge »](#) .
- [Section 24.2.3.3, « Créer un VPC \(Virtual Private Cloud\) »](#)
- [Section 24.2.3.4, « Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC »](#)
- [Section 24.2.3.5, « Configurer le routage par défaut du sous-système privé VPC »](#)
- [Section 24.2.3.7, « Configurer l'installation IAM »](#)
- [Section 24.2.3.9, « Configurer l'installation S3 Bucket »](#)

Procédure 24.14. Lancer un contrôleur de domaine clusterisé

1. Créer un IP élastique pour cette instance.
2. Sélectionner une AMI.
3. Allez dans le groupe de sécurité et autoriser tout le trafic (utiliser les capacités de pare-feu intégrées de Red Hat Enterprise Linux pour restreindre l'accès si nécessaire).
4. Choisir "running" dans le sous-système public du VPC.
5. Sélectionner un IP statique (comme par ex 10.0.0.5).
6. Mettez ce qui suit dans le champ User Data :

```
## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## password that will be used by slave host controllers to connect
to the domain controller
JBOSSAS_ADMIN_PASSWORD=<password for slave host controllers>

## subnet prefix this machine is connected to
SUBNET=10.0.0.

## S3 domain controller discovery setup
```

```

# JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY=<your secret key>
# JBOSS_DOMAIN_S3_ACCESS_KEY=<your access key>
# JBOSS_DOMAIN_S3_BUCKET=<your bucket name>

##### to run the example no modifications below should be needed #####
JBOSS_DOMAIN_CONTROLLER=true
PORTS_ALLOWED="9999 9990 9443"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Get the application to be deployed from an Internet URL
# mkdir -p /usr/share/java/jboss-ec2-eap-applications
# wget https://<your secure storage hostname>/<path>/<app name>.war
-O /usr/share/java/jboss-ec2-eap-applications/<app name>.war

## Install the JDBC driver as a core module
yum -y install mysql-connector-java
mkdir -p /usr/share/jbossas/modules/com/mysql/main
cp -v /usr/share/java/mysql-connector-java-*.jar
/usr/share/jbossas/modules/com/mysql/main/mysql-connector-java.jar

cat > /usr/share/jbossas/modules/com/mysql/main/module.xml <<"EOM"
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
</module>
EOM

cat > $USER_CLI_COMMANDS << "EOC"
## Deploy the sample application from the local filesystem
deploy /usr/share/java/jboss-ec2-eap-samples/cluster-demo.war --
server-groups=other-server-group

## ExampleDS configuration for MySQL database
data-source --profile=mod_cluster-ec2-ha remove --name=ExampleDS
/profile=mod_cluster-ec2-ha/subsystem=datasources/jdbc-
driver=mysql:add(driver-name="mysql",driver-module-name="com.mysql")
data-source --profile=mod_cluster-ec2-ha add --name=ExampleDS --
connection-url="jdbc:mysql://${db.host}:3306/${db.database}" --jndi-
name=java:jboss/datasources/ExampleDS --driver-name=mysql --user-
name="${db.user}" --password="${db.passwd}"
/profile=mod_cluster-ec2-ha/subsystem=datasources/data-
source=ExampleDS:enable
EOC

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e ">:::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do

```

```

echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF

```

7. Pour les instances de production

Pour une instance de production, ajouter la ligne suivante sous la ligne **USER_SCRIPT** du champ **User Data** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

```

yum -y update

```



NOTE

yum -y update doit être exécuté régulièrement pour appliquer les correctifs de sécurité et les améliorations.

8. Lancement de l'instance Red Hat AMI

Résultat

Un domaine géré clusterisé de JBoss EAP 6 a été configuré, et lancé sur une Red Hat AMI.

[Rapporter un bogue](#)

24.2.3.11.2. Lancer une ou plusieurs instances pour qu'elles servent en tant que contrôleurs hôtes de cluster

Résumé

Cette rubrique couvre les étapes requises pour lancer une ou plusieurs instances de JBoss EAP 6 en tant que contrôleurs hôtes clusterisés sur une AMI de Red Hat (Amazon Machine Image de Red Hat).

Conditions préalables

- Configurer et lancer le contrôleur de domaine clusterisé. Consulter [Section 24.2.3.11.1, « Lancer une instance pour qu'elle serve de contrôleur de domaine de cluster »](#).
- [Section 24.2.3.7, « Configurer l'installation IAM »](#)
- [Section 24.2.3.9, « Configurer l'installation S3 Bucket »](#)

Procédure 24.15. Lancer les contrôleur hôte

Pour chaque instance que vous souhaitez créer, répétez les étapes suivantes :

1. Sélectionner une AMI.
2. Définir le nombre d'instances que vous souhaitez (le nombre de contrôleurs hôtes esclaves)
3. Sélectionner le VPC et le type d'instance.
4. Cliquer sur le groupe de sécurité.
5. Veillez à ce que tout le trafic en provenance du cluster JBoss EAP 6 soit autorisé.

6. Définir les autres restrictions suivant les besoins.

7. Ajouter ce qui suit dans le champ User Data :

```

## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## clustering setup
JBOSS_JGROUPS_S3_PING_SECRET_ACCESS_KEY=<your secret key>
JBOSS_JGROUPS_S3_PING_ACCESS_KEY=<your access key>
JBOSS_JGROUPS_S3_PING_BUCKET=<your bucket name>

## host controller setup
### static domain controller discovery setup
JBOSS_DOMAIN_MASTER_ADDRESS=10.0.0.5
### S3 domain controller discovery setup
# JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY=<your secret key>
# JBOSS_DOMAIN_S3_ACCESS_KEY=<your access key>
# JBOSS_DOMAIN_S3_BUCKET=<your bucket name>

JBOSS_HOST_PASSWORD=<password for slave host controllers>

## database credentials configuration
JAVA_OPTS="$JAVA_OPTS -
Ddb.host=instancename.something.rds.amazonaws.com -
Ddb.database=mydatabase -Ddb.user=<user> -Ddb.passwd=<pass>"

## subnet prefix this machine is connected to
SUBNET=10.0.1.

##### to run the example no modifications below should be needed #####
JBOSS_HOST_USERNAME=admin
PORTS_ALLOWED="1024:65535"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Server instance configuration
sed -i "s/main-server-group/other-server-group/"
$JBOSS_CONFIG_DIR/$JBOSS_HOST_CONFIG

## install the JDBC driver as a core module
yum -y install mysql-connector-java
mkdir -p /usr/share/jbossas/modules/com/mysql/main
cp -v /usr/share/java/mysql-connector-java-*.jar
/usr/share/jbossas/modules/com/mysql/main/mysql-connector-java.jar

cat > /usr/share/jbossas/modules/com/mysql/main/module.xml <<"EOM"
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>

```



```

</module>
EOM

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e ">:::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
    echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF

```

8. Pour les instances de production

Pour une instance de production, ajouter la ligne suivante sous la ligne **USER_SCRIPT** du champ **User Data** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

```
yum -y update
```



NOTE

yum -y update doit être exécuté régulièrement pour appliquer les correctifs de sécurité et les améliorations.

9. Lancement de l'instance AMI Red Hat

Résultat

Les contrôleurs hôtes clusterisés de JBoss EAP 6 ont été configurés, et lancés sur une AMI Red Hat.

[Rapporter un bogue](#)

24.2.3.11.3. Tester le domaine géré de JBoss EAP 6 clusterisée

Résumé

Cette rubrique couvre les étapes requises pour tester le domaine géré clusterisé de JBoss EAP 6 sur une Red Hat AMI (Amazon Machine Image de Red Hat)

Pour tester le domaine géré, vous devrez connaître les adresses IP élastiques de Apache HTTP et du contrôleur de domaines de JBoss EAP 6 à la fois.

Conditions préalables

- Configurer et lancer le contrôleur de domaine clusterisé. Consulter [Section 24.2.3.11.1, « Lancer une instance pour qu'elle serve de contrôleur de domaine de cluster »](#).
- Configurer et lancer les contrôleur hôte du cluster. Consulter [Section 24.2.3.11.2, « Lancer une ou plusieurs instances pour qu'elles servent en tant que contrôleurs hôtes de cluster »](#).

Procédure 24.16. Tester l'instance de serveur Apache HTTP

- Naviguer dans `http://ELASTIC_IP_OF_APACHE_HTTP_SERVER` avec un navigateur pour confirmer que le serveur web exécute avec succès.

Procédure 24.17. Tester le contrôleur de domaine

1. Naviguer dans `http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console`
2. Connectez-vous en utilisant le nom d'utilisateur `admin` et le mot de passe spécifiés dans le champ Données d'utilisateur pour le contrôleur de domaine. Une fois connecté, la page d'accueil de la console d'administration d'un domaine géré s'affichera (`http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console/App.html#server-instances`).
3. Cliquer sur l'étiquette `Server` du serveur en haut et à droite de l'écran. Sélectionner un contrôleur hôte dans le menu déroulant `Host` en haut et à gauche de l'écran.
4. Vérifier que ce contrôleur hôte ait deux configurations de serveurs nommées respectivement `server-one` et `server-two` et vérifier qu'elles appartiennent toutes deux à `other-server-group`.

Procédure 24.18. Tester les contrôleurs hôtes

1. Naviguer dans `http://ELASTIC_IP_OF_APACHE_HTTP_SERVER/cluster-demo/put.jsp`
2. Vérifier que l'un des contrôleurs hôtes journalise le message suivant : `Putting date now`.
3. Stopper l'instance du serveur qui a journalisé le message dans l'étape précédente (voir *Stop a Server Using the Management Console*).
4. Naviguer dans `http://ELASTIC_IP_OF_APACHE_HTTP_SERVER/cluster-demo/get.jsp`.
5. Vérifier que l'heure indiquée est la même que l'heure `PUT` (mise) par `put.jsp` à l'étape 2-a.
6. Vérifier que l'une des instances de serveur en cours d'exécution journalise le message suivant : `Getting date now`.
7. Re-démarrer l'instance du serveur arrêtée (voir [Section 2.3.2, « Démarrer un serveur par la console de gestion »](#))
8. Connectez-vous à l'instance Apache HTTP .

```
$ ssh -L7654:localhost:7654 ELASTIC_IP_OF_APACHE_HTTP_SERVER
```

9. Naviguer dans `http://localhost:7654/mod_cluster-manager` pour confirmer que toutes les instances exécutent correctement.

Résultat

Le serveur web de JBoss EAP 6, le contrôleur de domaine, et les contrôleurs hôtes exécutent correctement sur une Red Hat AMI.

[Rapporter un bogue](#)

24.3. METTRE EN PLACE LE MONITORING DANS JBOSS OPERATIONS NETWORK (JON)

24.3.1. AMI Monitoring

Une fois que vous avez votre application commerciale déployée dans une instance AMI configurée correctement, l'étape suivante est d'instaurer le monitoring de la plateforme avec JON (JBoss Operations Network).

Le serveur JON est généralement situé à l'intérieur d'un réseau d'entreprise, donc il est nécessaire d'établir une connexion sécurisée entre le serveur et chacun de ses agents. La création d'un réseau privé virtuel entre les deux points est la solution la plus courante, mais cela complique la configuration de réseau requise. Ce chapitre fournit des directives de configuration de réseau permettant d'établir la communication entre l'agent de JON et le serveur JON. Pour plus d'informations sur la configuration, la gestion et l'utilisation, veuillez vous référer à la documentation officielle de Red Hat pour JBoss Operations Network (JON).

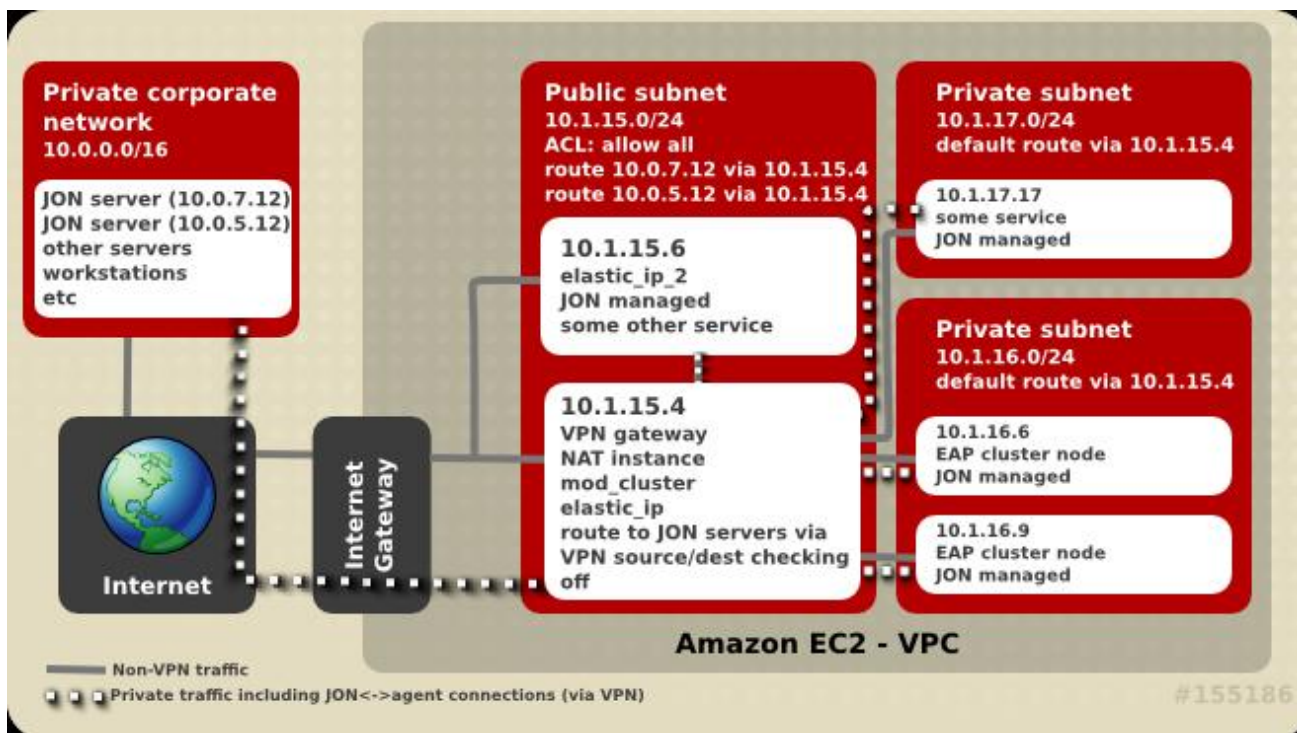


Figure 24.1. La connectivité du serveur JON

[Rapporter un bogue](#)

24.3.2. Prérequis de connectivité

L'inscription d'un agent JON avec ses serveurs requiert une communication bidirectionnelle entre l'agent et les serveurs. L'agent JON a besoin d'accéder au port 7080 sur tous les serveurs de JON, sauf dans le cas de SSL quand le port 7443 est utilisé. Chaque serveur de JON doit pouvoir accéder à chacun des agents connectés sur un hôte unique et son port homologue. Le port de l'agent est habituellement 16163.

S'il y a plusieurs serveurs JON clusterisés, veillez à ce que chaque agent puisse communiquer avec tous les serveurs dans le cluster JON via les paires IP et nom d'hôte comme configurés par la console d'administration du serveur JON. Le serveur JON utilisé par l'agent à enregistrer n'est sans doute pas le serveur qu'il essaie d'utiliser après l'initialisation.

[Rapporter un bogue](#)

24.3.3. Network Address Translation (NAT)

Une passerelle VPN entreprise agissant en mode routé simplifie grandement la configuration du réseau. Si toutefois votre gateway VPN entreprise opère en mode NAT, le serveur JON n'a pas de visibilité directe des agents. Dans ce cas, le réacheminement de port devra être configuré pour chaque agent.

Les configurations NAT VPN ont besoin d'un port sur la passerelle à transmettre à l'adresse de port de l'agent JON sur l'ordinateur dont il s'agit. L'agent JON doit également être configuré pour indiquer au serveur le numéro de port transféré et l'adresse IP. Vous trouverez de plus amples informations dans la description de `rhq.communications.connector.*` du fichier de configuration de `agent-configuration.xml`

[Rapporter un bogue](#)

24.3.4. Amazon EC2 et DNS

Les serveurs JON et les agents JON doivent être en mesure de résoudre les noms d'hôtes des uns et des autres. La résolution DNS est plus compliquée dans le cas d'une configuration VPN. Les serveurs connectés ont plusieurs options possibles. Une des options consiste à utiliser les serveurs DNS du réseau de l'entreprise ou Amazon EC2. Une autre option est d'utiliser une configuration divisée de DNS avec les serveurs DNS de l'entreprise utilisés pour résoudre les noms dans des domaines particuliers, et les serveurs d'Amazon EC2 DNS utilisés pour la résolution de tous les autres noms.

[Rapporter un bogue](#)

24.3.5. Le routage dans EC2

Tous les serveurs de Amazon EC2 ont une fonctionnalité de routage `source/destination checking` activée par défaut. Cette fonction supprime tous les paquets envoyés au serveur qui ont une destination séparée de l'adresse IP de la machine. Si la solution VPN sélectionnée pour la connexion des agents au serveur JON inclut un routeur, cette fonctionnalité devra être désactivée pour le ou les serveur(s) agissant en tant que routeurs ou passerelles VPN. Ce paramètre de configuration est accessible via la console d'Amazon AWS. La désactivation de `source/destination checking` est également requise dans un cloud privé virtuel (VPC).

Certaines configurations VPN renvoient le trafic internet général par le VPN entreprise par défaut. Il est conseillé de l'éviter car cela risque de ralentir et de rendre moins la configuration moins performante pour vos besoins particuliers.

Malgré que l'utilisation d'un schéma d'adressage approprié n'est pas un problème spécifique à JON, les mauvais schémas peuvent l'affecter. Amazon EC2 attribue des adresses IP à partir du réseau `10.0.0.0/8`. Les instances ont généralement une adresse IP publique également, mais seul le trafic réseau sur l'adresse IP interne dans la même zone de disponibilité est gratuit. Pour éviter d'utiliser le réseau `10.0.0.0/8` en adressage privé, il y a plusieurs choses à considérer.

- Quand vous créez un VPC, évitez d'allouer des adresses déjà utilisées dans le réseau privé afin d'éviter les problèmes de connectivité.
- Si une instance a besoin d'accéder à des ressources locales de zone disponible, veillez à ce que les adresses privées Amazon EC2 soient utilisées et que le trafic ne soit pas dirigé par le VPN.
- Si une instance d'Amazon EC2 a accès à un petit sous-ensemble d'adresses de réseau privé d'entreprise (par exemple les serveurs JON uniquement), seules ces adresses devront être acheminées par le VPN. Cela augmentera la sécurité et réduira les chances de collision d'espaces d'adresse de réseau privé et Amazon EC2.

[Rapporter un bogue](#)

24.3.6. Quitter ou Re-démarrer JON

Un des avantages des environnements de Cloud Computing est la facilité avec laquelle vous pouvez résilier ou lancer une instance de la machine. Vous pouvez également lancer une instance identique à l'instance initiale. Cela peut entraîner des problèmes si la nouvelle instance tente de s'enregistrer par les serveurs JON en utilisant le même nom d'agent que celui de l'agent déjà en cours d'exécution. Dans un tel cas, le serveur JON ne permettra pas à un agent de se reconnecter avec un token d'identification manquant ou non correspondant.

Afin d'éviter cela, veillez à ce que les agents qui ont fait leur travail soient retirés de l'inventaire JON avant d'essayer de connecter un agent du même nom ou de spécifier le token d'identification qui convient quand vous démarrerez un nouvel agent.

Un autre problème que vous pourriez rencontrer est lorsqu'une machine d'agent reçoit une nouvelle adresse IP VPN qui ne correspond plus à l'adresse enregistrée dans la configuration de JON. Un exemple pourrait inclure une machine qui redémarre ou lorsque une connexion VPN a été interrompue. Dans ce cas, il est recommandé que vous liez le cycle de vie de l'agent JON au cycle de vie de la connexion VPN. Si la connexion tombe, vous pouvez arrêter l'agent. Lorsque la connexion est rétablie à nouveau, mettre à jour `JON_AGENT_ADDR` dans `/etc/sysconfig/jon-agent-ec2` pour refléter la nouvelle adresse IP, puis redémarrez l'agent.

Les informations sur la façon de changer l'adresse IP de l'agent se trouve dans le guide «Configuring JON Servers and Agents Guide» à l'adresse suivante https://access.redhat.com/site/documentation/JBoss_Operations_Network/.

S'il existe un grand nombre d'instances lancées ou résiliées, cela risque d'être difficile d'ajouter ou de supprimer les instances manuellement dans l'inventaire de JON. Les capacités de scripting de JON peuvent être utilisées pour automatiser ces étapes. Voir la documentation JON pour plus d'informations.

[Rapporter un bogue](#)

24.3.7. Configurer une instance pour vous enregistrer dans le JBoss Operations Network

Utiliser la procédure suivante pour enregistrer une instance de JBoss EAP 6 dans JBoss Operations Network.

- Dans JBoss EAP 6, ajouter ceci dans le champ User Data (données utilisateur).

```
JON_SERVER_ADDR=jon2.it.example.com
## if instance not already configured to resolve its hostname
JON_AGENT_ADDR=`ip addr show dev eth0 primary to 0/0 | sed -n
's#.*inet \([0-9.]\+\)/.*#\1#p'`
PORTS_ALLOWED=16163
# insert other JON options when necessary.
```

Voir les paramètres [Section 24.4.1, « Paramètres de configuration permanente »](#), commençant par `JON_` pour le format des options JON.

[Rapporter un bogue](#)

24.4. CONFIGURATION DU SCRIPT UTILISATEUR

24.4.1. Paramètres de configuration permanente

Résumé

Les paramètres suivants peuvent être utilisés pour influencer la configuration et les opérations de JBoss EAP 6. Leur contenu se trouve dans `/etc/sysconfig/jbossas` et `/etc/sysconfig/jon-agent-ec2`.

Tableau 24.2. Paramètres configurables

Nom	Description	Par défaut
JBOSS_JGROUPS_S3_PING_ACCESS_KEY	Clé d'accès de compte utilisateur Amazon AWS pour S3_PING Discovery quand on utilise le clustering.	S/O
JBOSS_JGROUPS_S3_PING_SECRET_ACCESS_KEY	Clé d'accès secrète au compte utilisateur Amazon AWS	S/O
JBOSS_JGROUPS_S3_PING_BUCKET	Amazon S3 Bucket à utiliser dans S3_PING Discovery.	S/O
JBOSS_CLUSTER_ID	<p>ID des noeuds de membres d'un groupement. Utilisé uniquement pour le clustering. Les valeurs acceptées sont (dans l'ordre) :</p> <ul style="list-style-type: none"> • Un nombre d'ID de groupement valide entre 0 - 1023. • Un nom d'interface de réseau, avec le dernier octet de l'IP utilisé comme valeur. • "S3" comme valeur coordonnerait l'utilisation de l'ID par la S3 Bucket utilisée par S3_PING des jgroups. <p>Il est conseillé d'utiliser le dernier octet de l'IP (par défaut) quand tous les noeuds de cluster sont situés dans le sous-système de 24 octets ou davantage (par exemple, dans un sous-ensemble VPC).</p>	Dernier octet de l'adresse IP d'eth0
MOD_CLUSTER_PROXY_LIST	Liste délimitée par des virgules d'IP/Noms d'hôte de proxies mod_cluster si mod_cluster doit être utilisé.	S/O

Nom	Description	Par défaut
PORTS_ALLOWED	Liste des ports entrants qui seront utilisés par le pare-feu en plus des ports par défaut.	S/O
JBOSSAS_ADMIN_PASSWORD	Mot de passe pour l'utilisateur admin .	S/O
JON_SERVER_ADDR	IP ou nom d'hôte du serveur JON dans lequel s'enregistrer. Uniquement utilisé pour l'enregistrement, ensuite, l'agent peut communiquer avec les autres serveurs dans le groupement JON.	S/O
JON_SERVER_PORT	Port utilisé par l'agent pour communiquer avec le serveur.	7080
JON_AGENT_NAME	Nom de l'agent JON, doit être unique.	ID de l'instance
JON_AGENT_PORT	Port que l'agent écoute.	16163
JON_AGENT_ADDR	Adresse IP à laquelle l'agent JON est relié. Utilisé quand le serveur a plus d'une adresse publique, (par ex VPN).	L'agent JON choisit l'IP ou le nom d'hôte local par défaut.
JON_AGENT_OPTS	Propriétés de système d'agent JON supplémentaire pouvant être utilisées pour configurer SSL, NAT et d'autres paramètres avancés.	S/O

Nom	Description	Par défaut
JBOSS_SERVER_CONFIG	<p>Nom du fichier de configuration de serveur JBoss EAP 6 à utiliser. Si JBOSS_DOMAIN_CONTROLLER=true, alors domain-ec2.xml sera utilisé. Sinon :</p> <ul style="list-style-type: none"> • Si la config S3 est présente, alors standalone-ec2-ha.xml sera utilisé. • Si MOD_CLUSTER_PROXY_LIST est spécifié, alors standalone-mod_cluster-ec2-ha.xml sera sélectionné. • Si aucune des deux premières options n'est utilisée, alors le fichier standalone.xml sera utilisé. • Peut également être défini à standalone-full.xml. 	standalone.xml , standalone-full.xml , standalone-ec2-ha.xml , standalone-mod_cluster-ec2-ha.xml , domain-ec2.xml suivant les autres paramètres.
JAVA_OPTS	Valeurs personnalisées à ajouter à la variable avant que JBoss EAP 6 démarre.	JAVA_OPTS est créé à partir de valeurs appartenant à d'autres paramètres.
JBOSS_IP	Adresse IP à laquelle le serveur est lié.	127.0.0.1
JBOSSCONF	Le nom du profil de JBoss EAP 6 pour démarrer. Pour empêcher JBoss EAP 6 de démarrer, JBOSSCONF peut être défini à disabled	standalone
JBOSS_DOMAIN_CONTROLLER	Indique si cette instance exécute ou non comme contrôleur de domaine.	false
JBOSS_DOMAIN_MASTER_ADDR ESS	Adresse IP de contrôleur de domaine distant.	S/O
JBOSS_HOST_NAME	Le nom d'hôte logique (du domaine). Doit être un nom séparé.	La valeur de la variable d'environnement HOSTNAME.

Nom	Description	Par défaut
JBOSS_HOST_USERNAME	Le nom d'utilisateur du contrôleur hôte doit être utilisé quand on enregistre dans le contrôleur de domaine. Si non fourni, le JBOSS_HOST_NAME sera utilisé à la place.	JBOSS_HOST_NAME
JBOSS_HOST_PASSWORD	Le mot de passe que le contrôleur hôte doit utiliser quand il s'enregistre avec le contrôleur de domaine.	S/O
JBOSS_HOST_CONFIG	Si JBOSS_DOMAIN_CONTROLLER=true, alors host-master.xml sera utilisé. Si JBOSS_DOMAIN_MASTER_ADDRESS est présent, alors host-slave.xml sera utilisé.	host-master.xml ou host-slave.xml , suivant les autres paramètres.
JBOSS_DOMAIN_S3_ACCESS_KEY	Clé d'accès de compte utilisateur AWS Amazon pour la découverte de contrôleurs de domaines S3.	S/O
JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY	Clé secrète d'accès de compte utilisateur AWS Amazon pour la découverte de contrôleurs de domaines S3.	S/O
JBOSS_DOMAIN_S3_BUCKET	Amazon S3 Bucket à utiliser pour la découverte de contrôleur de domaine S3.	S/O

[Rapporter un bogue](#)

24.4.2. Paramètres de scripts personnalisés

Résumé

Les paramètres suivants peuvent être utilisés dans la section de personnalisation utilisateur du champ **User Data** :

Tableau 24.3. Paramètres configurables

Nom	Description
-----	-------------

Nom	Description
JBOSS_DEPLOY_DIR	Déployer le répertoire du profil actif (par exemple, <code>/var/lib/jbossas/standalone/deployments/</code>). Les archives déployables de ce répertoire seront déployées. Red Hat recommande d'utiliser la console de gestion ou le CLI pour gérer les déploiements au lieu d'utiliser le répertoire de déploiement.
JBOSS_CONFIG_DIR	Répertoire de config du profile actif (par exemple, <code>/var/lib/jbossas/standalone/configuration</code>).
JBOSS_HOST_CONFIG	Nom du fichier de configuration de l'hôte actif (par exemple, <code>host-master.xml</code>). Red Hat conseille d'utiliser la console de gestion ou le CLI pour configurer le serveur au lieu d'éditer le fichier de configuration.
JBOSS_SERVER_CONFIG	Nom du fichier de configuration du serveur actif (par exemple, <code>standalone-ec2-ha.xml</code>). Red Hat conseille d'utiliser la console de gestion ou le CLI pour configurer le serveur au lieu d'éditer le fichier de configuration.
USER_SCRIPT	Chemin d'accès au script de configuration personnalisé disponible avant de trouver la configuration user-data.
USER_CLI_COMMANDS	Chemin d'accès à un fichier personnalisé des commandes CLI, qui est disponible avant de trouver la configuration user-data.

[Rapporter un bogue](#)

24.5. RÉOLUTION DE PROBLÈMES

24.5.1. Résolution de problèmes dans Amazon EC2

EC2 donne un statut « Alarm » pour chaque instance, indiquant un dysfonctionnement grave d'instance, mais l'absence de ce statut d'alarme ne donne aucune garantie que l'instance a démarré correctement ou que les services s'exécutent correctement. Il est possible d'utiliser Amazon Cloud Watch avec ses fonctionnalités de métriques personnalisées pour surveiller la santé des instance de services mais l'utilisation d'une solution de gestion d'entreprise est conseillée.

Pour simplifier la résolution de problème, Red Hat conseille de gérer l'instance EC2 par JON (JBoss Operations Network) qui peut automatiquement découvrir, surveiller et gérer de nombreux services sur une instance EC2 avec l'agent JON installé, y compris JBoss Enterprise Application Platform et ses services installés, avec : JBoss EAP 6, Tomcat, Apache HTTP Server, and PostgreSQL. Pour obtenir plus d'informations sur le monitoring de JBoss EAP par JON, consulter [Section 24.3.1, « AMI Monitoring »](#).

[Rapporter un bogue](#)

24.5.2. Information de diagnostic

En cas de problème détecté par JBoss Operations Network, Amazon CloudWatch ou une inspection manuelle, voici les sources habituelles d'informations de diagnostic :

- `/var/log/jboss_user-data.out` est la sortie du script init de jboss-ec2-eap et du script de configuration personnalisée utilisateur.
- `/var/cache/jboss-ec2-eap/` contient les données utilisateur réelles, le script personnalisé, et les commandes CLI utilisées au démarrage de l'instance.
- `/var/log` contient également tous les journaux collectés au démarrage de la machine, JBoss EAP 6, httpd et la plupart des autres services.

L'accès à ces fichiers est disponible uniquement via une session SSH. Voir Amazon EC Getting Started Guide pour obtenir des informations sur la façon de configurer ou d'établir une session SSH avec une instance Amazon EC2.

[Rapporter un bogue](#)

ANNEXE A. RÉFÉRENCES SUPPLÉMENTAIRES

A.1. TÉLÉCHARGER LES FICHIERS DU PORTAIL CLIENT DE RED HAT

Conditions préalables

- Avant de commencer cette tâche, vous aurez besoin d'un compte de Portail clients. Rendez-vous dans <https://access.redhat.com> et cliquer sur le lien **Register** qui se trouve en haut et à droite pour créer un compte.

Procédure A.1. Connectez-vous et téléchargez les fichiers du Portail clients de Red Hat

1. Aller dans <https://access.redhat.com> et cliquer sur le lien **Log in** en haut et à gauche. Saisir vos identifiants, et cliquer sur **Log In**.

Résultat

Vous êtes connecté dans RHN et on vous renvoie à la page web principale à <https://access.redhat.com>.

2. **Rendez-vous à la page Downloads.**

Utiliser une des options de pour aller dans la page **Downloads**.

- Cliquer sur le lien **Downloads** qui se trouve en haut de la barre de navigation.
- Rendez-vous directement dans <https://access.redhat.com/downloads/>.

3. **Sélectionner le produit et la version à télécharger.**

Utiliser une de ces méthodes pour choisir le produit et la version qui conviennent à télécharger.

- Procéder une étape à la fois.
- Chercher votre produit à partir de la zone de recherche qui se trouve en haut et à droite de l'écran.

4. **Télécharger le fichier qui convient pour votre système d'exploitation et la méthode d'installation de votre choix.**

Suivant le produit, vous aurez le choix entre un installateur natif, un RPM ou une archive Zip suivant le système d'exploitation et l'architecture. Cliquer soit sur le nom de fichier ou sur le lien **Download** à droite du fichier que vous souhaitez télécharger.

Résultat

Le fichier sera téléchargé dans votre ordinateur.

[Rapporter un bogue](#)

A.2. CONFIGURER LE JDK PAR DÉFAUT DANS RED HAT ENTERPRISE LINUX

Il est possible d'avoir plusieurs Java Development Kits (JDKs) installés sur votre système Red Hat Enterprise Linux. Cette tâche vous montre comment spécifier quel kit est utilisé par votre environnement actuel. Nécessite la commande `alternatives`.

**IMPORTANT**

Cette tâche ne s'applique que pour Red Hat Enterprise Linux.

**NOTE**

Il est possible de ne pas avoir à passer par cette étape. Red Hat Enterprise Linux utilise OpenJDK 1.6 comme option par défaut. Si c'est ce qui vous convient et que vous utilisez votre système correctement, vous n'aurez pas besoin d'indiquer quel JDK utiliser.

Conditions préalables

- Pour compléter cette tâche, vous devrez avoir l'accès super utilisateur, soit en connexion directe, ou par la commande `sudo`.

Procédure A.2. Configurer le JDK par défaut

1. Déterminez les chemins qui vous conviennent le mieux pour vos binaires `java` et `javac`. Vous pouvez utiliser la commande `rpm -ql packagename |grep bin` pour trouver les emplacements des binaires installés à partir des RPM. Les locations par défaut des binaires `java` et `javac` des systèmes Red Hat Enterprise Linux 32-bit sont les suivantes :

Tableau A.1. Emplacements par défaut des binaires `java` et `javac`

JDK	Chemin
OpenJDK 1.6	<pre>/usr/lib/jvm/jre-1.6.0- openjdk/bin/java</pre> <pre>/usr/lib/jvm/java-1.6.0- openjdk/bin/javac</pre>
Oracle JDK 1.6	<pre>/usr/lib/jvm/jre-1.6.0- sun/bin/java</pre> <pre>/usr/lib/jvm/java-1.6.0- sun/bin/javac</pre>

2. Définir chaque solution alternative que vous souhaitez utiliser pour chacun. Exécutez les commandes suivantes pour que votre système utilise `java` et `javac`: `/usr/sbin/alternatives --config java` ou `/usr/sbin/alternatives --config javac`. Suivez les instructions sur l'écran.
3. Option: définir un choix alternatif `java_sdk_1.6.0`.
Si vous souhaitez utiliser Oracle JDK, vous devrez configurer la solution alternative `java_sdk_1.6.0`. également. Utiliser la commande suivante : `/usr/sbin/alternatives --config java_sdk_1.6.0`. Le chemin d'accès qui convient est normalement `/usr/lib/jvm/java-1.6.0-sun`. Vous pourrez faire une liste de fichiers pour vérifier.

Résultat :

Le JDK alternatif est sélectionné et actif.

[Rapporter un bogue](#)

A.3. RÉFÉRENCE DE JOURNALISATION D'AUDITING DE L'INTERFACE DE GESTION

Référence d'attributs de logger

En plus d'activer ou de désactiver la Journalisation de l'auditing de l'interface de gestion, vous pouvez utiliser les attributs de configuration de **logger** suivants.

log-boot

Si défini sur **true**, quand on démarre le serveur, les opérations de gestion seront incluses dans le log d'auditing, sinon **false**. La valeur par défaut est **true**.

log-read-only

Si défini sur **true**, toutes les opérations d'auditing seront journalisées. Si défini sur **false**, seules les opérations qui changeront le modèle seront journalisées. La valeur par défaut est : **false**.

Référence d'attributs de Formateur de journeaux

Le formateur indique le format des entrées de journalisation. Il n'y a qu'un formateur disponible, qui transforme les entrées de journalisation en format JASON.

Exemple A.1. Inclut l'horodatage dans les archives log

```
/core-service=management/access=audit/json-formatter=json-  
formatter:write-attribute(name=include-date,value=true)
```

Attributs de Formateurs de journeaux

include-date

Une valeur booléenne qui indique si oui ou non l'horodatage est inclus dans les archives log formatées. La valeur par défaut est **true**.

date-separator

Une chaîne contenant des caractères à utiliser pour séparer la date du reste des messages log formatés. Sera ignoré si **include-date=false**. La valeur par défaut est `-` (espace, suivi d'un trait d'union, puis espace à nouveau).

date-format

Format de date utilisé pour les horodatages compris par `java.text.SimpleDateFormat`. Ignoré si **include-date=false**. La valeur par défaut est `yyyy-MM-dd HH:mm:ss`.

compact

Si défini sur **true**, formatera le JSON sur une seule ligne. Il y a sans doute encore des valeurs contenant de nouvelles lignes, donc s'il est important pour vous d'avoir l'enregistrement total sur une seule ligne, définir **escape-new-line** ou **escape-control-characters** à **true**. La valeur par défaut est **false**.

escape-control-characters

Si défini sur `true`, échappera tous les caractères de contrôle (entrées ASCII à valeur décimales < 32) avec le code ASCII en octale ; par exemple, une nouvelle ligne devient `#012`. Si défini sur `true`, remplacera `escape-new-line=false`. La valeur par défaut est `false`.

escape-new-line

Si défini sur `true`, échappera toutes les nouvelles lignes avec le code ASCII en octale, comme par exemple `#012`. La valeur par défaut est `false`.

Référence d'attributs de Gestionnaire de fichiers de journalisation de l'auditing de l'interface de gestion

Un gestionnaire de fichiers indique les paramètres par lesquels les enregistrements de journalisation sont mis dans les fichiers. Il indique plus particulièrement le formateur, le nom du fichier et le chemin d'accès du fichier.

Attributs de Gestionnaire de fichiers

formatter

Le nom d'un formateur JSON à utiliser pour formater les enregistrements de journalisation. La valeur par défaut est `json-formatter`.

path

Le chemin d'accès du fichier de journalisation de l'auditing. La valeur par défaut est `audit-log.log`.

relative-to

Le nom d'un chemin d'accès déjà nommé, ou d'un des chemins standard fournis par le système. Si `relative-to` est donné, la valeur de l'attribut du chemin d'accès sera traitée comme relative au chemin spécifié par cet attribut. La valeur par défaut est `jboss.server.data.dir`.

failure-count

Le nombre d'échecs de journalisation depuis l'initialisation du gestionnaire. La valeur par défaut est 0.

max-failure-count

Le nombre maximum d'échecs de journalisation avant de désactiver ce gestionnaire. La valeur par défaut est 10.

disabled-due-to-failure

Sur `true` si ce gestionnaire était désactivé à cause des échecs de journalisation. La valeur par défaut est `false`.

Référence d'attributs du gestionnaire de syslog de l'interface de gestion

Un gestionnaire de syslog indique les paramètres par lesquels les entrées de journalisations d'auditing sont envoyées à un serveur syslog, comme le nom d'hôte du serveur syslog et le nom du port sur lequel le serveur syslog écoute.

Envoyer une journalisation d'auditing à un serveur de syslog fournit davantage d'options de sécurité que de journaliser dans un fichier local ou sur un serveur syslog local. On peut définir plusieurs gestionnaires syslog qui soient actifs en même temps.

Les serveurs de syslog peuvent varier dans leur implémentation, donc tous les paramètres ne sont pas forcément applicables à tous les serveurs de syslog. Le testing a été conduit par l'implémentation syslog *rsyslog*.

Les listes *Syslog Handler Attributes* répertorient uniquement les attributs de haut niveau. Chaque attribut possède des paramètres de configuration, et certains ont des paramètres de configuration enfant. Pour obtenir les détails des attributs d'un gestionnaire syslog, exécutez la commande suivante.

```
/core-service=management/access=audit/syslog-handler=mysyslog:read-resource-description(recursive=true)
```

Attributs de Gestionnaire Syslog

app-name

Le nom de l'application à ajouter aux archives syslog ainsi défini dans la section 6.2.5 de RFC-5424. Si non spécifié, aura comme valeur par défaut le nom du produit.

disabled-due-to-failure

Sur `true` si ce gestionnaire était désactivé à cause des échecs de journalisation. La valeur par défaut est `false`.

fonctionnalité

La fonctionnalité à utiliser pour la journalisation de syslog ainsi définie dans la section 6.2.1 de RFC-5424, et la section 4.1.1 de RFC-3164.

failure-count

Le nombre d'échecs de journalisation depuis l'initialisation du gestionnaire. La valeur par défaut est `0`.

formatter

Le nom d'un formateur à utiliser pour formater les enregistrements de journalisation. La valeur par défaut est `json-formatter`.

max-failure-count

Le nombre maximum d'échecs de journalisation avant de désactiver ce gestionnaire. La valeur par défaut est `10`.

max-length

La taille maximum d'un message de journalisation (en octets), comprenant l'en-tête. Si non défini, la valeur par défaut sera de `1024` octets si le `syslog-format` est `RFC3164`, ou `2048` octets si le `syslog-format` est `RFC5424`.

protocol

Le protocole à utiliser pour le gestionnaire de syslog. Doit correspondre à l'un des protocoles suivants `udp`, `tcp` ou `tls`.

reconnect-timeout

Disponible à partir de JBoss EAP 6.4. Le nombre de secondes à attendre avant de tenter de se reconnecter au serveur de syslog, en cas de perte de connectivité. La valeur par défaut est : **-1** (désactivé).

syslog-format

Format de Syslog : *RFC-5424* ou *RFC-3164*. Valeur par défaut : **RFC-5424**.

truncate

Indique si un message et son en-tête doivent ou non être tronqués quand la longueur en octets est supérieure à la **max-length** (longueur maximale) de l'attribut. Si la valeur est définie sur **false**, les messages seront divisés et envoyés avec les mêmes valeurs d'en-tête. La valeur par défaut est **false**.

[Rapporter un bogue](#)

ANNEXE B. HISTORIQUE DES RÉVISIONS

Version 6.4.0-25

Tuesday April 14 2015

Lucas Costi

Red Hat JBoss Enterprise Application Platform 6.4.0.GA