Red Hat Enterprise Linux 9

9.1 Notes de mise à jour

Notes de mise à jour pour Red Hat Enterprise Linux 9.1

Last Updated: 2023-06-12

Red Hat Enterprise Linux 9 9.1 Notes de mise à jour

Notes de mise à jour pour Red Hat Enterprise Linux 9.1

Notice légale

Résumé

Les notes de mise à jour fournissent une couverture de haut niveau des améliorations et des ajouts qui ont été mis en œuvre dans Red Hat Enterprise Linux 9.1 et documentent les problèmes connus dans cette version, ainsi que les corrections de bogues notables, les aperçus technologiques, les fonctionnalités obsolètes et d'autres détails.

Table des matières

RENDRE L'OPEN SOURCE PLUS INCLUSIF 5

FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT 6

CHAPITRE 1. VUE D'ENSEMBLE 7
1.1. PRINCIPAUX CHANGEMENTS DANS RHEL 9.1 7
1.2. MISE À NIVEAU SUR PLACE 10
1.3. PORTAIL CLIENTS DE RED HAT 11
1.4. RESSOURCES SUPPLÉMENTAIRES 12

CHAPITRE 2. ARCHITECTURES 13

CHAPITRE 3. DISTRIBUTION DU CONTENU DANS RHEL 9 14
3.1. INSTALLATION 14
3.2. RÉFÉRENTIELS 14
3.3. FLUX D'APPLICATIONS 15
3.4. GESTION DES PAQUETS AVEC YUM/DNF 15

CHAPITRE 4. NOUVELLES FONCTIONNALITÉS 17
4.1. CRÉATION D'INSTALLATEURS ET D'IMAGES 17
4.2. RHEL POUR EDGE 19
4.3. GESTION DES ABONNEMENTS 19
4.4. GESTION DES LOGICIELS 20
4.5. SHELLS ET OUTILS DE LIGNE DE COMMANDE 20
4.6. SERVICES D'INFRASTRUCTURE 23
4.7. SÉCURITÉ 26
4.8. MISE EN RÉSEAU 29
4.9. NOYAU 32
4.10. CHARGEUR DE DÉMARRAGE 36
4.11. SYSTÈMES DE FICHIERS ET STOCKAGE 36
4.12. HAUTE DISPONIBILITÉ ET CLUSTERS 37
4.13. LANGAGES DE PROGRAMMATION DYNAMIQUES, SERVEURS WEB ET DE BASE DE DONNÉES 39
4.14. COMPILATEURS ET OUTILS DE DÉVELOPPEMENT 42
4.15. GESTION DE L'IDENTITÉ 49
4.16. INFRASTRUCTURES GRAPHIQUES 54
4.17. LA CONSOLE WEB 55
4.18. RÔLES DU SYSTÈME RED HAT ENTERPRISE LINUX 55
4.19. VIRTUALISATION 61
4.20. RHEL DANS LES ENVIRONNEMENTS EN NUAGE 62
4.21. CONTENEURS 63

CHAPITRE 5. CHANGEMENTS IMPORTANTS DANS LES PARAMÈTRES EXTERNES DU NOYAU 66
Nouveaux paramètres du noyau 66
Mise à jour des paramètres du noyau 68
Nouveaux paramètres sysctl 72
Modification des paramètres sysctl 72

CHAPITRE 6. PILOTES DE PÉRIPHÉRIQUES 74
6.1. NOUVEAUX CONDUCTEURS 74
6.2. PILOTES MIS À JOUR 75

CHAPITRE 7. CARACTÉRISTIQUES DU FBP DISPONIBLES 77

CHAPITRE 8. BUG FIXES 93

8.1. CRÉATION D'INSTALLATEURS ET D'IMAGES 93
8.2. GESTION DES ABONNEMENTS 93
8.3. GESTION DES LOGICIELS 93
8.4. SHELLS ET OUTILS DE LIGNE DE COMMANDE 94
8.5. SERVICES D'INFRASTRUCTURE 95
8.6. SÉCURITÉ 95
8.7. MISE EN RÉSEAU 97
8.8. NOYAU 98
8.9. CHARGEUR DE DÉMARRAGE 99
8.10. SYSTÈMES DE FICHIERS ET STOCKAGE 99
8.11. HAUTE DISPONIBILITÉ ET CLUSTERS 100
8.12. COMPILATEURS ET OUTILS DE DÉVELOPPEMENT 101
8.13. GESTION DE L'IDENTITÉ 102
8.14. BUREAU 102
8.15. INFRASTRUCTURES GRAPHIQUES 103
8.16. LA CONSOLE WEB 103
8.17. RÔLES DU SYSTÈME RED HAT ENTERPRISE LINUX 103
8.18. VIRTUALISATION 106
8.19. RHEL DANS LES ENVIRONNEMENTS EN NUAGE 106
8.20. CONTENEURS 107

CHAPITRE 9. APERÇUS TECHNOLOGIQUES 109
9.1. SHELLS ET OUTILS DE LIGNE DE COMMANDE 109
9.2. SÉCURITÉ 109
9.3. MISE EN RÉSEAU 110
9.4. NOYAU 111
9.5. SYSTÈMES DE FICHIERS ET STOCKAGE 111
9.6. COMPILATEURS ET OUTILS DE DÉVELOPPEMENT 112
9.7. GESTION DE L'IDENTITÉ 112
9.8. BUREAU 115
9.9. LA CONSOLE WEB 116
9.10. VIRTUALISATION 116
9.11. RHEL DANS LES ENVIRONNEMENTS EN NUAGE 117
9.12. CONTENEURS 117

CHAPITRE 10. FONCTIONNALITÉ OBSOLÈTE 119
10.1. CRÉATION D'INSTALLATEURS ET D'IMAGES 119
10.2. SÉCURITÉ 119
10.3. MISE EN RÉSEAU 121
10.4. NOYAU 122
10.5. SYSTÈMES DE FICHIERS ET STOCKAGE 122
10.6. LANGAGES DE PROGRAMMATION DYNAMIQUES, SERVEURS WEB ET DE BASE DE DONNÉES 122
10.7. COMPILATEURS ET OUTILS DE DÉVELOPPEMENT 123
10.8. GESTION DE L'IDENTITÉ 123
10.9. BUREAU 124
10.10. INFRASTRUCTURES GRAPHIQUES 124
10.11. RÔLES DU SYSTÈME RED HAT ENTERPRISE LINUX 125
10.12. VIRTUALISATION 125
10.13. CONTENEURS 127
10.14. PAQUETS OBSOLÈTES 127

CHAPITRE 11. PROBLÈMES CONNUS 129
11.1. CRÉATION D'INSTALLATEURS ET D'IMAGES 129
11.2. GESTION DES ABONNEMENTS 133

11.3. GESTION DES LOGICIELS 133
11.4. SHELLS ET OUTILS DE LIGNE DE COMMANDE 134
11.5. SERVICES D'INFRASTRUCTURE 134
11.6. SÉCURITÉ 135
11.7. MISE EN RÉSEAU 139
11.8. NOYAU 139
11.9. CHARGEUR DE DÉMARRAGE 144
11.10. SYSTÈMES DE FICHIERS ET STOCKAGE 144
11.11. LANGAGES DE PROGRAMMATION DYNAMIQUES, SERVEURS WEB ET DE BASE DE DONNÉES 145
11.12. COMPILATEURS ET OUTILS DE DÉVELOPPEMENT 146
11.13. GESTION DE L'IDENTITÉ 146
11.14. BUREAU 150
11.15. INFRASTRUCTURES GRAPHIQUES 151
11.16. LA CONSOLE WEB 152
11.17. VIRTUALISATION 152
11.18. RHEL DANS LES ENVIRONNEMENTS EN NUAGE 154
11.19. CAPACITÉ DE SOUTIEN 155
11.20. CONTENEURS 156

ANNEXE A. LISTE DES TICKETS PAR COMPOSANT **157**

ANNEXE B. HISTORIQUE DES RÉVISIONS **164**

RENDRE L'OPEN SOURCE PLUS INCLUSIF

Red Hat s'engage à remplacer les termes problématiques dans son code, sa documentation et ses propriétés Web. Nous commençons par ces quatre termes : master, slave, blacklist et whitelist. En raison de l'ampleur de cette entreprise, ces changements seront mis en œuvre progressivement au cours de plusieurs versions à venir. Pour plus de détails, voir le message de notre directeur technique Chris Wright.

FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT

Nous apprécions vos commentaires sur notre documentation. Faites-nous savoir comment nous pouvons l'améliorer.

- Soumettre des commentaires sur des passages spécifiques

    1. Consultez la documentation au format **Multi-page HTML** et assurez-vous que le bouton **Feedback** apparaît dans le coin supérieur droit après le chargement complet de la page.
    2. Utilisez votre curseur pour mettre en évidence la partie du texte que vous souhaitez commenter.
    3. Cliquez sur la fenêtre **Add Feedback** qui apparaît près du texte en surbrillance.
    4. Ajoutez vos commentaires et cliquez sur **Submit**.

- Soumettre un retour d'information via Bugzilla (action requise) :

    1. Connectez-vous au site Web de Bugzilla.
    2. Sélectionnez la version correcte dans le menu **Version**.
    3. Saisissez un titre descriptif dans le champ **Summary**.
    4. Cliquez sur **Submit Bug**.

CHAPITRE 1. VUE D'ENSEMBLE

1.1. PRINCIPAUX CHANGEMENTS DANS RHEL 9.1

Création d'installateurs et d'images

Voici les principales caractéristiques du constructeur d'images dans RHEL 9.1 GA :

- Image builder on-premise supporte maintenant :
  - Téléchargement d'images vers le GCP
  - Personnalisation de la partition /**boot**
  - Pousser une image de conteneur directement vers un registre
  - Les utilisateurs peuvent désormais personnaliser leurs plans au cours du processus de création d'images.

Pour plus d'informations, voir Section 4.1, « Création d'installateurs et d'images ».

RHEL pour Edge

Voici les principales caractéristiques de RHEL for Edge dans RHEL 9.1-GA :

- RHEL for Edge prend désormais en charge l'installation des services et leur exécution avec la configuration par défaut, à l'aide de l'utilitaire CLI **fdo-admin**

Pour plus d'informations, voir Section 4.2, « RHEL pour Edge ».

Sécurité

RHEL 9.1 introduit **Keylime**, un outil d'attestation de machine distante utilisant la technologie TPM (trusted platform module). Avec Keylime, vous pouvez vérifier et surveiller en permanence l'intégrité des machines distantes.

SELinux ont été mis à jour vers la version 3.4. Les changements les plus notables sont les suivants :

- Amélioration des performances de réétiquetage grâce au réétiquetage parallèle
- Prise en charge de SHA-256 dans l'outil **semodule**
- Nouveaux services publics dans le paquet **libsepol-utils**

Les modifications apportées à la configuration du système et au sous-paquet **clevis-luks-systemd** permettent au client de chiffrement Clevis de déverrouiller également les volumes chiffrés LUKS qui se montent tardivement dans le processus de démarrage sans utiliser la commande **systemctl enable clevis-luks-askpass.path** pendant le processus de déploiement.

Voir Nouveautés - Sécurité pour plus d'informations.

Shells et outils de ligne de commande

RHEL 9.1 introduit un nouveau paquetage **xmlstarlet**. Avec **XMLStarlet**vous pouvez analyser, transformer, interroger, valider et éditer des fichiers XML.

Les outils de ligne de commande suivants ont été mis à jour dans RHEL 9.1 :

- **opencryptoki** à la version 3.18.0

- **powerpc-utils** à la version 1.3.10
- **libvpd** à la version 2.2.9
- **lsvpd** à la version 1.7.14
- **ppc64-diag** à la version 2.7.8

Pour plus d'informations, voir Nouvelles fonctionnalités - Shells et outils de ligne de commande

Services d'infrastructure

Les outils de services d'infrastructure suivants ont été mis à jour dans RHEL 9.1 :

- **chrony** à la version 4.2
- **unbound** à la version 1.16.2
- **frr** à la version 8.2.2

Pour plus d'informations, voir Nouveautés - Services d'infrastructure.

Mise en réseau

NetworkManager supporte la migration des profils de connexion du format **ifcfg** vers le format keyfile.

NetworkManager indique désormais clairement que la prise en charge du WEP n'est pas disponible dans RHEL 9.

Le code MultiPath TCP (MPTCP) du noyau a été mis à jour à partir de la version amont de Linux 5.19.

Pour plus de détails, voir Nouveautés - Mise en réseau.

Langages de programmation dynamiques, serveurs web et de base de données

Les versions ultérieures des composants suivants sont désormais disponibles sous la forme de nouveaux modules :

- **PHP 8.1**
- **Ruby 3.1**
- **Node.js 18**

En outre, le site **Apache HTTP Server** a été mis à jour à la version 2.4.53.

Pour plus d'informations, voir Nouveautés - Langages de programmation dynamiques, serveurs web et de base de données.

Compilateurs et outils de développement

Mise à jour de la chaîne d'outils système

Les composants suivants de la chaîne d'outils système ont été mis à jour dans RHEL 9.1 :

- **GCC 11.2.1**
- **glibc 2.34**
- **binutils 2.35.2**

Mise à jour des outils de performance et des débogueurs

Les outils de performance et les débogueurs suivants ont été mis à jour dans RHEL 9.1 :

- GDB 10.2
- Valgrind 3.19
- SystemTap 4.7
- Dyninst 12.1.0
- elfutils 0.187

Mise à jour des outils de contrôle des performances
Les outils de surveillance des performances suivants ont été mis à jour dans RHEL 9.1 :

- PCP 5.3.7
- Grafana 7.5.13

Mise à jour des outils de compilation
Les ensembles d'outils de compilation suivants ont été mis à jour dans RHEL 9.1 :

- GCC Toolset 12
- LLVM Toolset 14.0.6
- Rust Toolset 1.62
- Go Toolset 1.18

Pour plus de détails sur les changements, voir Section 4.14, « Compilateurs et outils de développement ».

Implémentations Java dans RHEL 9
Le référentiel RHEL 9 AppStream comprend :

- Les paquets **java-17-openjdk**, qui fournissent l'environnement d'exécution Java OpenJDK 17 et le kit de développement logiciel Java OpenJDK 17.
- Les paquets **java-11-openjdk**, qui fournissent l'environnement d'exécution Java OpenJDK 11 et le kit de développement logiciel Java OpenJDK 11.
- Les paquets **java-1.8.0-openjdk**, qui fournissent l'environnement d'exécution Java OpenJDK 8 et le kit de développement logiciel Java OpenJDK 8.

Pour plus d'informations, voir la documentation OpenJDK.

Outils Java
RHEL 9.1 introduit **Maven 3.8** en tant que nouveau flux de modules.

Voir Section 4.14, « Compilateurs et outils de développement » pour plus d'informations.

Gestion de l'identité

Identity Management (IdM) dans RHEL 9.1 introduit un aperçu technologique dans lequel vous pouvez déléguer l'authentification des utilisateurs à des fournisseurs d'identité externes (IdP) qui prennent en charge le flux OAuth 2 Device Authorization Grant. Lorsque ces utilisateurs s'authentifient avec SSSD, et après avoir terminé l'authentification et l'autorisation auprès du fournisseur d'identité externe, ils bénéficient des fonctionnalités d'authentification unique de RHEL IdM avec des tickets Kerberos.

Pour plus d'informations, voir les aperçus technologiques - Gestion de l'identité

Rôles du système Red Hat Enterprise Linux

Nouveautés notables de la version 9.1 RHEL System Roles :

- Les rôles système RHEL sont désormais disponibles également dans les playbooks dont la collecte des faits est désactivée.
- Le rôle **ha_cluster** prend désormais en charge les clôtures SBD, la configuration des paramètres Corosync et la configuration des ressources de bundle.
- Le rôle **network** permet désormais de configurer les paramètres réseau pour les règles de routage, de prendre en charge la configuration du réseau à l'aide de **nmstate API**, et les utilisateurs peuvent créer des connexions avec la capacité IPoIB.
- Le rôle **microsoft.sql.server** comporte de nouvelles variables, telles que des variables permettant de contrôler la configuration d'un cluster à haute disponibilité, de gérer automatiquement les ports du pare-feu ou des variables permettant de rechercher les valeurs **mssql_tls_cert** et **mssql_tls_private_key** sur les nœuds gérés.
- Le rôle **logging** prend en charge plusieurs nouvelles options, par exemple **startmsg.regex** et **endmsg.regex** dans les fichiers d'entrée, ou les options **template**, **severity** et **facility**.
- Le rôle **storage** inclut désormais la prise en charge des volumes à provisionnement fin, et le rôle a également moins de verbosité par défaut.
- Le rôle **sshd** vérifie la directive include pour le répertoire drop-in, et le rôle peut maintenant être géré via /etc/ssh/sshd_config.
- Le rôle **metrics** peut désormais exporter les données de performance de Postfix.
- Le rôle **postfix** dispose désormais d'une nouvelle option permettant d'écraser la configuration précédente.
- Le rôle **firewall** ne nécessite pas le paramètre state lors de la configuration de masquerade ou icmp_block_inversion. Dans le rôle **firewall**, vous pouvez désormais ajouter, mettre à jour ou supprimer des services en utilisant les états absent et présent. Le rôle peut également fournir des faits Ansible, et ajouter ou supprimer une interface à la zone en utilisant l'ID de périphérique PCI. Le rôle **firewall** dispose d'une nouvelle option pour écraser la configuration précédente.
- Le rôle de **selinux** comprend désormais la définition des paramètres de **seuser** et **selevel**.

1.2. MISE À NIVEAU SUR PLACE

Mise à niveau en place de RHEL 8 à RHEL 9

Les chemins de mise à niveau en place pris en charge sont actuellement les suivants :

- De RHEL 8.6 à RHEL 9.0 sur les architectures suivantes :
  - 64-bit Intel
  - 64-bit AMD
  - aRM 64 bits
  - IBM POWER 9 (little endian)
  - Architectures IBM Z, à l'exception de z13

- De RHEL 8.6 à RHEL 9.0 sur des systèmes avec SAP HANA

Pour garantir la prise en charge de votre système après la mise à niveau vers RHEL 9.0, mettez à jour vers la dernière version de RHEL 9.1 ou activez les référentiels Extended Update Support (EUS) de RHEL 9.0.

Pour obtenir des instructions sur l'exécution d'une mise à niveau en place, voir Mise à niveau de RHEL 8 vers RHEL 9.

Pour obtenir des instructions sur l'exécution d'une mise à niveau en place sur des systèmes dotés d'environnements SAP, voir Comment mettre à niveau en place des environnements SAP de RHEL 8 à RHEL 9.

Parmi les améliorations notables, citons

- Les mises à niveau sur place sur Microsoft Azure et Google Cloud Platform avec Red Hat Update Infrastructure (RHUI) sont désormais possibles.
- Les configurations OpenSSH et OpenSSL sont désormais migrées lors de la mise à niveau en place.

Mise à niveau en place de RHEL 7 à RHEL 9
Il n'est pas possible d'effectuer une mise à niveau directement de RHEL 7 à RHEL 9. Toutefois, vous pouvez effectuer une mise à niveau de RHEL 7 à RHEL 8, puis une seconde mise à niveau vers RHEL 9. Pour plus d'informations, voir Mise à niveau de RHEL 7 à RHEL 8 .

1.3. PORTAIL CLIENTS DE RED HAT

Red Hat Customer Portal Labs est un ensemble d'outils dans une section du portail client disponible sur https://access.redhat.com/labs/. Les applications de Red Hat Customer Portal Labs peuvent vous aider à améliorer les performances, à résoudre rapidement les problèmes, à identifier les problèmes de sécurité et à déployer et configurer rapidement des applications complexes. Certaines des applications les plus populaires sont :

- Assistant d'inscription
- Générateur de démarrage
- Certificats de produits Red Hat
- Red Hat CVE Checker
- Analyseur d'erreurs du noyau
- Red Hat Code Browser
- Configurateur VNC
- Graphique de mise à jour de la plateforme de conteneurs Red Hat OpenShift
- Aide à la mise à niveau de Red Hat Satellite
- Outil de configuration des options de la JVM
- Outil de configuration de l'équilibreur de charge

- Red Hat OpenShift Data Foundation Supportability and Interoperability Checker (vérificateur de supportabilité et d'interopérabilité de Red Hat OpenShift Data Foundation)
- Assistant de mise à niveau de la plateforme d'automatisation Ansible
- Calculateur de groupes de placement de céphales (PG) par pool

1.4. RESSOURCES SUPPLÉMENTAIRES

Capabilities and limits de Red Hat Enterprise Linux 9 par rapport à d'autres versions du système sont disponibles dans l'article de la base de connaissances Capacités et limites de la technologie Red Hat Enterprise Linux.

Les informations relatives à Red Hat Enterprise Linux **life cycle** sont fournies dans le document Red Hat Enterprise Linux Life Cycle.

Le document Package manifest fournit une adresse **package listing** pour RHEL 9, y compris les licences et les niveaux de compatibilité des applications.

Application compatibility levels sont expliquées dans le document Red Hat Enterprise Linux 9 : Guide de compatibilité des applications.

Les principaux sites **differences between RHEL 8 and RHEL 9**, y compris les fonctionnalités supprimées, sont documentés dans le document Considerations in adopting RHEL 9 (considérations relatives à l'adoption de RHEL 9).

Le document Upgrading from RHEL 8 to RHEL 9 fournit des instructions sur la manière d'effectuer un **in-place upgrade from RHEL 8 to RHEL 9**.

Le service **Red Hat Insights**, qui vous permet d'identifier, d'examiner et de résoudre de manière proactive les problèmes techniques connus, est disponible avec tous les abonnements RHEL. Pour obtenir des instructions sur l'installation du client Red Hat Insights et l'enregistrement de votre système au service, consultez la page Red Hat Insights Get Started .

CHAPITRE 2. ARCHITECTURES

Red Hat Enterprise Linux 9.1 est distribué avec la version 5.14.0-162 du noyau, qui prend en charge les architectures suivantes à la version minimale requise :

- Architectures AMD et Intel 64 bits (x86-64-v2)
- L'architecture ARM 64 bits (ARMv8.0-A)
- IBM Power Systems, Little Endian (POWER9)
- 64 bits IBM Z (z14)

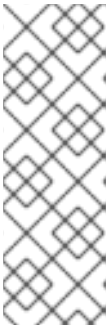
Assurez-vous d'acheter l'abonnement approprié pour chaque architecture. Pour plus d'informations, voir Démarrer avec Red Hat Enterprise Linux - architectures supplémentaires .

CHAPITRE 3. DISTRIBUTION DU CONTENU DANS RHEL 9

3.1. INSTALLATION

Red Hat Enterprise Linux 9 est installé à l'aide d'images ISO. Deux types d'images ISO sont disponibles pour les architectures AMD64, Intel 64 bits, ARM 64 bits, IBM Power Systems et IBM Z :

- ISO d'installation : une image d'installation complète qui contient les référentiels BaseOS et AppStream et vous permet de terminer l'installation sans référentiels supplémentaires. Sur la page Téléchargements de produits , le site **Installation ISO** est appelé **Binary DVD**.



NOTE

L'image ISO d'installation est d'une taille de plusieurs Go et, par conséquent, elle peut ne pas être compatible avec les formats de supports optiques. Il est recommandé d'utiliser une clé USB ou un disque dur USB lors de l'utilisation de l'image ISO d'installation pour créer un support d'installation amorçable. Vous pouvez également utiliser l'outil Image Builder pour créer des images RHEL personnalisées. Pour plus d'informations sur Image Builder, consultez le document *Composing a customized RHEL system image* document.

- ISO de démarrage : une image ISO de démarrage minimale qui est utilisée pour démarrer le programme d'installation. Cette option nécessite l'accès aux référentiels BaseOS et AppStream pour l'installation des logiciels. Les référentiels font partie de l'image ISO d'installation. Vous pouvez également vous enregistrer auprès de Red Hat CDN ou Satellite pendant l'installation afin d'utiliser les derniers contenus BaseOS et AppStream de Red Hat CDN ou Satellite.

Consultez le document Exécution d'une installation RHEL 9 standard pour obtenir des instructions sur le téléchargement d'images ISO, la création de supports d'installation et l'achèvement d'une installation RHEL. Pour les installations Kickstart automatisées et d'autres sujets avancés, voir le document Exécution d'une installation RHEL 9 avancée .

Pour obtenir la liste des utilisateurs et des groupes créés par les RPM dans une installation RHEL de base, ainsi que la marche à suivre pour obtenir cette liste, consultez l'article de la base de connaissances Quels sont tous les utilisateurs et groupes dans une installation RHEL de base ? Article de la base de connaissances.

3.2. RÉFÉRENTIELS

Red Hat Enterprise Linux 9 est distribué par le biais de deux dépôts principaux :

- BaseOS
- AppStream

Ces deux dépôts sont nécessaires pour une installation RHEL de base et sont disponibles avec tous les abonnements RHEL.

Le contenu du référentiel BaseOS est destiné à fournir l'ensemble des fonctionnalités du système d'exploitation sous-jacent qui constitue la base de toutes les installations. Ce contenu est disponible au format RPM et est soumis à des conditions de support similaires à celles des versions précédentes de RHEL. Pour plus d'informations, voir le document Scope of Coverage Details.

Le contenu du référentiel AppStream comprend des applications supplémentaires pour l'espace utilisateur, des langages d'exécution et des bases de données afin de prendre en charge les différentes charges de travail et les différents cas d'utilisation.

En outre, le référentiel CodeReady Linux Builder est disponible avec tous les abonnements RHEL. Il fournit des paquets supplémentaires à l'usage des développeurs. Les paquets inclus dans le dépôt CodeReady Linux Builder ne sont pas pris en charge.

Pour plus d'informations sur les dépôts RHEL 9 et les paquets qu'ils fournissent, voir le manifeste des paquets.

3.3. FLUX D'APPLICATIONS

Les versions multiples des composants de l'espace utilisateur sont fournies sous forme de flux d'applications et mises à jour plus fréquemment que les paquets du système d'exploitation principal. Cela offre une plus grande flexibilité pour personnaliser RHEL sans impacter la stabilité sous-jacente de la plateforme ou des déploiements spécifiques.

Les flux d'applications sont disponibles dans le format RPM habituel, en tant qu'extension du format RPM appelée modules, en tant que collections de logiciels ou en tant que Flatpaks.

Chaque composant Application Stream a un cycle de vie donné, soit identique à celui de RHEL 9, soit plus court. Pour plus d'informations sur le cycle de vie de RHEL, voir Red Hat Enterprise Linux Life Cycle.

RHEL 9 améliore l'expérience des flux d'applications en fournissant des versions initiales des flux d'applications qui peuvent être installées en tant que paquets RPM à l'aide de la commande traditionnelle **dnf install**.

NOTE

Certains flux d'applications initiaux au format RPM ont un cycle de vie plus court que Red Hat Enterprise Linux 9.

Certaines versions supplémentaires d'Application Stream seront distribuées sous forme de modules avec un cycle de vie plus court dans les prochaines versions mineures de RHEL 9. Les modules sont des ensembles de paquets représentant une unité logique : une application, une pile de langues, une base de données ou un ensemble d'outils. Ces paquets sont construits, testés et publiés ensemble.

Déterminez toujours la version d'un flux d'applications que vous souhaitez installer et assurez-vous de consulter d'abord le cycle de vie du flux d'applications de Red Hat Enterprise Linux .

Les contenus nécessitant une mise à jour rapide, tels que les compilateurs alternatifs et les outils de conteneur, sont disponibles dans des flux continus qui ne fourniront pas de versions alternatives en parallèle. Les flux roulants peuvent être conditionnés sous forme de RPM ou de modules.

Pour obtenir des informations sur les flux d'applications disponibles dans RHEL 9 et leur niveau de compatibilité avec les applications, consultez le manifeste du paquetage. Les niveaux de compatibilité des applications sont expliqués dans le document Red Hat Enterprise Linux 9 : Guide de compatibilité des applications.

3.4. GESTION DES PAQUETS AVEC YUM/DNF

Dans Red Hat Enterprise Linux 9, l'installation du logiciel est assurée par **DNF**. Red Hat continue à soutenir l'utilisation du terme **yum** par souci de cohérence avec les versions majeures précédentes de

RHEL. Si vous tapez **dnf** au lieu de **yum**, la commande fonctionne comme prévu car il s'agit dans les deux cas d'alias de compatibilité.

Bien que RHEL 8 et RHEL 9 soient basés sur **DNF**, ils sont compatibles avec **YUM** utilisé dans RHEL 7.

Pour plus d'informations, voir Gestion des logiciels avec l'outil DNF .

CHAPITRE 4. NOUVELLES FONCTIONNALITÉS

Cette partie décrit les nouvelles fonctionnalités et les améliorations majeures introduites dans Red Hat Enterprise Linux 9.1.

4.1. CRÉATION D'INSTALLATEURS ET D'IMAGES

Prise en charge de l'analyse automatique des LUN SCSI du FCP dans le programme d'installation

Le programme d'installation peut désormais utiliser l'analyse automatique des LUN lors de l'attachement des LUN SCSI FCP sur les systèmes IBM Z. L'analyse automatique des LUN est disponible pour les périphériques FCP fonctionnant en mode NPIV, si elle n'est pas désactivée par le paramètre du module du noyau **zfcp.allow_lun_scan**. Il est activé par défaut. Il permet d'accéder à tous les périphériques SCSI du réseau de stockage attachés au périphérique FCP avec l'ID de bus spécifié. Il n'est plus nécessaire de spécifier le WWPN et les LUN FCP et il suffit de fournir uniquement l'ID du bus du périphérique FCP.

(BZ#1937031)

Image builder on-premise prend désormais en charge la personnalisation de la partition /boot

La version sur site d'Image builder prend désormais en charge la construction d'images avec une taille de partition de point de montage **/boot** personnalisée. Vous pouvez spécifier la taille de la partition du point de montage **/boot** dans la personnalisation du plan, afin d'augmenter la taille de la partition **/boot** au cas où la taille de la partition de démarrage par défaut serait trop petite. Par exemple :

```
[[customizations.filesystem]]
mountpoint = "/boot"
size = "20 GiB"
```

(JIRA:RHELPLAN-130379)

Ajout de l'option --allow-ssh kickstart pour activer les connexions racine SSH basées sur un mot de passe

Lors de l'installation graphique, vous avez la possibilité d'activer les connexions racine SSH basées sur un mot de passe. Cette fonctionnalité n'était pas disponible dans les installations kickstart. Avec cette mise à jour, une option **--allow-ssh** a été ajoutée à la commande **rootpw** kickstart. Cette option permet à l'utilisateur root de se connecter au système en utilisant SSH avec un mot de passe.

(BZ#2083269)

Menu du chargeur de démarrage caché par défaut

Le chargeur de démarrage GRUB est désormais configuré pour masquer le menu de démarrage par défaut. Il en résulte une expérience de démarrage plus fluide. Le menu de démarrage est caché dans tous les cas suivants :

- Lorsque vous redémarrez le système à partir de l'environnement de bureau ou de l'écran de connexion.
- Lors du premier démarrage du système après l'installation.
- Lorsque le paquetage **greenboot** est installé et activé.

Si le démarrage précédent du système a échoué, GRUB affiche toujours le menu de démarrage lors du démarrage suivant.

Pour accéder manuellement au menu de démarrage, utilisez l'une des options suivantes :

- Appuyer plusieurs fois sur **Esc** pendant le démarrage.
- Appuyez plusieurs fois sur **F8** pendant le démarrage.
- Maintenir la **touche Shift** enfoncée pendant le démarrage.

Pour désactiver cette fonction et configurer l'affichage par défaut du menu du chargeur de démarrage, utilisez la commande suivante :

```
# grub2-editenv - unset menu_auto_hide
```

(BZ#2059414)

L'installation minimale de RHEL n'installe plus que le paquet s390utils-core

Dans RHEL 8.4 et les versions ultérieures, le paquet **s390utils-base** est divisé en un paquet **s390utils-core** et un paquet auxiliaire **s390utils-base**. Par conséquent, si vous définissez l'installation RHEL sur **minimal-environment**, vous n'installez que le paquet **s390utils-core** nécessaire et non le paquet auxiliaire **s390utils-base**. Si vous souhaitez utiliser le paquet **s390utils-base** avec une installation RHEL minimale, vous devez installer manuellement le paquet après avoir terminé l'installation RHEL ou installer explicitement **s390utils-base** à l'aide d'un fichier kickstart.

(BZ#1932480)

Image builder on-premise prend désormais en charge le téléchargement d'images vers GCP

Grâce à cette amélioration, vous pouvez utiliser l'interface CLI de construction d'images pour créer une image **gce**, en fournissant les informations d'identification de l'utilisateur ou du compte de service que vous souhaitez utiliser pour télécharger les images. En conséquence, image builder crée l'image et télécharge ensuite l'image **gce** directement dans l'environnement GCP que vous avez spécifié.

(BZ#2049492)

L'interface de programmation (CLI) d'Image builder on-premise permet de pousser une image de conteneur directement vers un registre

Grâce à cette amélioration, vous pouvez pousser les images de conteneurs RHEL for Edge directement vers un registre de conteneurs après leur construction, à l'aide de l'interface CLI de construction d'images. Pour construire l'image de conteneur :

1. Configurez un fournisseur de téléchargement et, éventuellement, ajoutez des informations d'identification.
2. Construit l'image du conteneur, en passant le registre du conteneur et le référentiel à **composer-cli** comme arguments.
   Une fois que l'image est prête, elle est disponible dans le registre de conteneurs que vous avez configuré.

(JIRA:RHELPLAN-130376)

Les utilisateurs d'Image builder sur site peuvent désormais personnaliser leurs plans au cours du processus de création d'images

Avec cette mise à jour, la page **Edit Blueprint** a été supprimée afin d'unifier l'expérience de l'utilisateur dans le service de création d'images et dans l'application de création d'images sur **cockpit-composer**. Les utilisateurs peuvent désormais créer leurs plans et les personnaliser, par exemple en ajoutant des paquets et en créant des utilisateurs, au cours du processus de création d'images. Le versionnage des blueprints a également été supprimé, de sorte que les blueprints n'ont qu'une seule version : la version actuelle. Les utilisateurs ont accès aux versions antérieures des blueprints par l'intermédiaire de leurs images déjà créées.

(JIRA:RHELPLAN-122735)

4.2. RHEL POUR EDGE

RHEL for Edge prend désormais en charge l'utilitaire fdo-admin cli

Avec cette mise à jour, vous pouvez configurer les services FDO directement dans tous les scénarios de déploiement en utilisant le CLI.

Exécutez les commandes suivantes pour générer les certificats et les clés des services :

NOTE

Cet exemple tient compte du fait que vous avez déjà installé le paquetage RPM **fdo-admin-cli**. Si vous avez utilisé le code source et l'avez compilé, le chemin correct est **./target/debug/fdo-admin-tool** ou **./target/debug/fdo-admin-tool**, en fonction de vos options de compilation.

```
$ mkdir keys
$ for i in "diun" "manufacturer" "device_ca" "owner"; do fdo-admin-tool generate-key-and-cert $i; done
$ ls keys
device_ca_cert.pem  device_ca_key.der  diun_cert.pem  diun_key.der  manufacturer_cert.pem
manufacturer_key.der  owner_cert.pem  owner_key.der
```

Par conséquent, une fois que vous avez installé et démarré le service, il s'exécute avec les paramètres par défaut.

(JIRA:RHELPLAN-122776)

4.3. GESTION DES ABONNEMENTS

L'utilitaire subscription-manager affiche l'état actuel des actions

L'utilitaire **subscription-manager** affiche désormais des informations sur la progression de l'opération en cours. Ceci est utile lorsque **subscription-manager** prend plus de temps que d'habitude pour terminer ses opérations liées à la communication avec le serveur, par exemple, l'enregistrement.

Pour revenir au comportement précédent, entrez :

```
# subscription-manager config --rhsm.progress_messages=0
```

(BZ#2092014)

4.4. GESTION DES LOGICIELS

La commande modulesync est désormais disponible pour remplacer certains flux de travail dans RHEL 9

Dans RHEL 9, les paquets modulaires ne peuvent pas être installés sans métadonnées modulaires. Auparavant, vous pouviez utiliser la commande **dnf** pour télécharger des paquets, puis utiliser la commande **createrepo_c** pour redistribuer ces paquets.

Cette amélioration introduit la commande **modulesync** pour garantir la présence de métadonnées modulaires, ce qui assure l'installabilité des paquets. Cette commande télécharge les paquets RPM des modules et crée un référentiel avec des métadonnées modulaires dans un répertoire de travail.

(BZ#2066646)

4.5. SHELLS ET OUTILS DE LIGNE DE COMMANDE

Cronie ajoute la prise en charge d'un temps aléatoire dans une plage sélectionnée

L'utilitaire **Cronie** supporte désormais l'opérateur ~ (random within range) pour l'exécution des cronjob. Par conséquent, vous pouvez démarrer un cronjob à un moment aléatoire dans l'intervalle sélectionné.

(BZ#2090691)

ReaR ajoute de nouvelles variables pour l'exécution des commandes avant et après la récupération

Avec cette amélioration, ReaR introduit deux nouvelles variables pour faciliter l'automatisation des commandes à exécuter avant et après la récupération :

- **PRE_RECOVERY_COMMANDS** accepte un tableau de commandes. Ces commandes seront exécutées avant le début de la récupération.
- **POST_RECOVERY_COMMANDS** accepte un tableau de commandes. Ces commandes seront exécutées une fois la récupération terminée.

Ces variables sont une alternative à **PRE_RECOVERY_SCRIPT** et **POST_RECOVERY_SCRIPT** avec les différences suivantes :

- Les variables précédentes **PRE_RECOVERY_SCRIPT** et **POST_RECOVERY_SCRIPT** acceptent une seule commande de l'interpréteur de commandes. Pour passer plusieurs commandes à ces variables, vous devez les séparer par des points-virgules.
- Les nouvelles variables **PRE_RECOVERY_COMMANDS** et **POST_RECOVERY_COMMANDS** acceptent des tableaux de commandes, et chaque élément du tableau est exécuté comme une commande distincte.

Par conséquent, il est désormais plus facile et moins sujet aux erreurs de fournir plusieurs commandes à exécuter dans le système de sauvetage avant et après la récupération.

Pour plus d'informations, voir le fichier **default.conf**.

(BZ#2111059)

Un nouveau paquet : xmlstarlet

XMLStarlet est un ensemble d'utilitaires en ligne de commande permettant d'analyser, de transformer,

d'interroger, de valider et d'éditer des fichiers XML. Le nouveau paquetage **xmlstarlet** fournit un ensemble simple de commandes shell que vous pouvez utiliser de la même manière que les commandes UNIX pour les fichiers de texte brut tels que **grep**, **sed**, **awk**, **diff**, **patch**, **join**, et autres.

(BZ#2069689)

opencryptoki repassé à la version 3.18.0

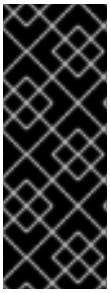
Le paquetage **opencryptoki**, qui est une implémentation de la norme de cryptographie à clé publique (PKCS) #11, a été mis à jour à la version 3.18.0. Les améliorations notables sont les suivantes :

- Format par défaut des données des jetons conformes aux normes du Federal Information Processing Standards (FIPS) (tokversion = 3.12).
- Ajout d'un support pour restreindre l'utilisation des mécanismes et des clés avec une politique globale.
- Ajout d'un support pour le comptage statistique de l'utilisation des mécanismes.
- Les jetons **ICA/EP11** prennent désormais en charge la version 4 de la bibliothèque **libica**.
- L'outil **p11sak** permet de définir différents attributs pour les clés publiques et privées.
- Le **C_GetMechanismList** ne renvoie pas le **CKR_BUFFER_TOO_SMALL** dans le jeton EP11.

openCryptoki prend en charge deux formats de données différents pour les jetons :

- le format de données antérieur, qui utilise des algorithmes non approuvés par la FIPS (tels que DES et SHA1)
- le nouveau format de données, qui utilise uniquement des algorithmes approuvés par le FIPS.

L'ancien format de données ne fonctionne plus car le fournisseur FIPS n'autorise l'utilisation que d'algorithmes approuvés par le FIPS.



IMPORTANT

Pour faire fonctionner openCryptoki sur RHEL 9, migrer les jetons pour utiliser le nouveau format de données avant d'activer le mode FIPS sur le système. Cette opération est nécessaire car l'ancien format de données est toujours utilisé par défaut dans **openCryptoki 3.17**. Les installations **openCryptoki** existantes qui utilisent l'ancien format de données des jetons ne fonctionneront plus lorsque le système passera en mode FIPS.

Vous pouvez migrer les jetons vers le nouveau format de données en utilisant l'utilitaire **pkcstok_migrate**, fourni avec **openCryptoki**. Notez que **pkcstok_migrate** utilise des algorithmes non approuvés par le FIPS pendant la migration. Par conséquent, il convient d'utiliser cet outil avant d'activer le mode FIPS sur le système. Pour plus d'informations, voir Migration vers la conformité FIPS - utilitaire pkcstok_migrate.

(BZ#2044179)

powerpc-utils repassé à la version 1.3.10

Le paquetage **powerpc-utils**, qui fournit divers utilitaires pour une plate-forme PowerPC, a été mis à jour à la version 1.3.10. Les améliorations notables sont les suivantes :

- Ajout de la possibilité d'analyser les informations relatives à la référence de la plate-forme d'architecture de puissance (PAPR) pour l'énergie et la fréquence dans l'outil **ppc64_cpu**.
- L'utilitaire **lparstat** a été amélioré afin d'afficher des messages d'erreur plus précis lorsque la commande **lparstat -E** échoue sur les systèmes de configuration maximale. La commande **lparstat** fournit des informations sur les partitions logiques.
- Correction de la mémoire en ligne signalée dans l'ancien format dans la commande **lparstat**.
- Ajout de la prise en charge de la commande **acc** pour modifier dynamiquement les crédits de qualité de service (QoS) pour l'accélérateur NX GZIP.
- Amélioration des spécificateurs de format dans les appels **printf()** et **sprintf()**.
- L'utilitaire **hcnmgr**, qui fournit les outils HMC au réseau virtuel hybride, comprend les améliorations suivantes :
  - Ajout de la fonction **wicked** à la liste Hybrid Network Virtualization **HNV FEATURE**. L'utilitaire **hcnmgr** prend en charge la virtualisation de réseau hybride (HNV) pour utiliser les fonctions **wicked** pour le bonding.
  - **hcnmgr** conserve un état **hcnid** en vue d'un nettoyage ultérieur.
  - **hcnmgr** exclut le code NetworkManager (NM) **nmcli**.
  - Le paramètre NM HNV **primary slave** a été corrigé.
  - **hcnmgr** prend en charge le contrôleur d'interface réseau virtuel (vNIC) en tant que périphérique de sauvegarde.
- Correction du message relatif au système de numérotation hexadécimale non valide sur le site **bootlist**.
- L'indicateur **-l** est inclus dans l'utilitaire **kpartx** en tant que valeur de délimitation **-p** dans la commande **bootlist**.
- Des corrections ont été apportées à l'utilitaire **sslot** afin d'éviter les fuites de mémoire lors de l'établissement de la liste des emplacements IO.
- Ajout des chaînes de description du type DRC pour les derniers types de fentes PCIe (peripheral component interconnect express) dans l'utilitaire **lsslot**.
- Correction de l'adresse de configuration invalide vers RTAS dans l'outil **errinjct**.
- Ajout de la prise en charge des dispositifs de mémoire non volatile sur tissus (NVMf) dans l'utilitaire **ofpathname**. L'utilitaire fournit un mécanisme pour convertir un nom de périphérique logique en un chemin de périphérique de micrologiciel ouvert et vice-versa.
- Ajout de correctifs pour la prise en charge de la mémoire non volatile (NVMe) en mode ANA (asymmetric namespace access) dans l'utilitaire **ofpathname**.
- Installation du fichier **smt.state** comme fichier de configuration.

(BZ#1920964)

Les modules Redfish font désormais partie de la collection Ansible **redhat.rhel_mgmt**

La collection Ansible de **redhat.rhel_mgmt** comprend désormais les modules suivants :

- **redfish_info**
- **redfish_command**
- **redfish_config**

Ainsi, les utilisateurs peuvent bénéficier de l'automatisation de la gestion en utilisant les modules Redfish pour récupérer l'état de santé du serveur, obtenir des informations sur l'inventaire du matériel et des microprogrammes, effectuer la gestion de l'alimentation, modifier les paramètres du BIOS, configurer les contrôleurs hors bande (OOB), configurer le RAID matériel et effectuer les mises à jour des microprogrammes.

(BZ#2112434)

libvpd repassé à la version 2.2.9

Le paquetage **libvpd**, qui contient des classes permettant d'accéder aux données sur les produits vitaux (VPD), a été mis à jour à la version 2.2.9. Les améliorations notables sont les suivantes :

- Correction du verrouillage de la base de données
- Mise à jour des informations sur la version de l'utilitaire **libtool**

(BZ#2051288)

lsvpd repassé à la version 1.7.14

Le paquetage **lsvpd**, qui fournit des commandes pour constituer un système d'inventaire matériel, a été mis à jour à la version 1.7.14. Avec cette mise à jour, l'utilitaire **lsvpd** empêche la corruption du fichier de base de données lorsque vous exécutez la commande **vpdupdate**.

(BZ#2051289)

ppc64-diag repassé à la version 2.7.8

Le paquetage **ppc64-diag** pour les diagnostics de plateforme a été mis à jour à la version 2.7.8. Les améliorations notables sont les suivantes :

- Mise à jour de la dépendance de construction pour utiliser l'utilitaire **libvpd** version 2.2.9 ou supérieure
- Correction du message d'erreur **extract_opal_dump** sur les plates-formes non prises en charge
- Correction d'un avertissement de compilation avec les compilateurs **GCC-8.5** et **GCC-11**

(BZ#2051286)

sysctl introduit la même syntaxe pour les arguments que systemd-sysctl

L'utilitaire **sysctl** du paquetage **procps-ng**, que vous pouvez utiliser pour modifier les paramètres du noyau au moment de l'exécution, utilise désormais la même syntaxe pour les arguments que l'utilitaire **systemd-sysctl**. Avec cette mise à jour, **sysctl** analyse désormais les fichiers de configuration qui contiennent des traits d'union (**-**) ou des globes (*****) sur les lignes de configuration. Pour plus d'informations sur la syntaxe de **systemd-sysctl**, consultez la page de manuel **sysctl.d(5)**.

(BZ#2052536)

4.6. SERVICES D'INFRASTRUCTURE

chrony utilise désormais des serveurs NTP DHCPv6

Le script NetworkManager dispatcher pour **chrony** met à jour les sources NTP (Network Time Protocol) transmises par les options DHCP (Dynamic Host Configuration Protocol). Depuis RHEL 9.1, le script utilise les serveurs NTP fournis par DHCPv6 en plus de DHCPv4. L'option DHCP 56 spécifie l'utilisation du DHCPv6, l'option DHCP 42 est spécifique au DHCPv4.

(BZ#2047415)

chrony repassé à la version 4.2

La suite **chrony** a été mise à jour à la version 4.2. Les améliorations notables par rapport à la version 4.1 sont les suivantes :

- Le mode entrelacé du serveur a été amélioré pour être plus fiable et prendre en charge plusieurs clients derrière un seul traducteur d'adresses (traduction d'adresses de réseau - NAT).
- La prise en charge expérimentale du champ d'extension du protocole NTPv4 (Network Time Protocol Version 4) a été ajoutée pour améliorer la stabilité de la synchronisation temporelle et la précision des erreurs estimées. Vous pouvez activer ce champ, qui étend les capacités du protocole NTPv4, en utilisant l'option **extfield F323**.
- Une prise en charge expérimentale de la transmission NTP sur le Precision Time Protocol (PTP) a été ajoutée pour permettre un horodatage matériel complet sur les cartes d'interface réseau (NIC) dont l'horodatage est limité aux paquets PTP. Vous pouvez activer le NTP sur le PTP en utilisant la directive **ptpport 319**.

(BZ#2051441)

unbound repassé à la version 1.16.2

Le composant **unbound** a été mis à jour vers la version 1.16.2. **unbound** est un résolveur DNS validant, récursif et de mise en cache. Les améliorations notables sont les suivantes :

- Grâce à la vérification des zones ZONEMD avec **RFC 8976**, les destinataires peuvent désormais vérifier l'intégrité des données et l'authenticité de l'origine du contenu de la zone.
- Avec **unbound**, vous pouvez désormais configurer des connexions TCP persistantes.
- Les types SVCB et HTTPS et leur traitement conformément à la spécification des paramètres et de la liaison de service via le document DNS **draft-ietf-dnsop-svcb-https** ont été ajoutés.
- **unbound** reprend les algorithmes de chiffrement TLS par défaut des stratégies cryptographiques.
- Vous pouvez utiliser un Special-Use Domain (domaine à usage spécial) **home.arpa.** conformément à **RFC8375**. Ce domaine est destiné à une utilisation non unique dans les réseaux résidentiels.
- **unbound** supporte désormais l'activation sélective des requêtes **tcp-upstream** pour les zones stub ou forward.
- La valeur par défaut de l'option **aggressive-nsec** est désormais **yes**.
- La logique de **ratelimit** a été mise à jour.

- Vous pouvez utiliser une nouvelle option **rpz-signal-nxdomain-ra** pour désactiver l'indicateur **RA** lorsqu'une requête est bloquée par une réponse nxdomain de la zone de politique de réponse non liée (RPZ).
- Avec la prise en charge de base des erreurs DNS étendues (EDE) conformément à **RFC8914**, vous pouvez bénéficier d'informations d'erreur supplémentaires.

(BZ#2087120)

La fonction de cryptage du mot de passe est désormais disponible dans whois

Le paquet **whois** fournit maintenant le binaire /**usr**/**bin**/**mkpasswd**, que vous pouvez utiliser pour crypter un mot de passe avec l'interface de la bibliothèque C **crypt**.

(BZ#2054043)

frr repassé à la version 8.2.2

Le paquetage **frr** pour la gestion de la pile de routage dynamique a été mis à jour à la version 8.2.2. Les changements et améliorations notables par rapport à la version 8.0 sont les suivants :

- Ajout d'une route Ethernet VPN (EVPN) de type 5 pour la passerelle IP Overlay Index.
- Ajout de la compression des routeurs frontières de systèmes autonomes (ASBR) dans le protocole Open-shortest-path-first (OSPFv3).
- Amélioration de l'utilisation des stub et des not-so-stubby-areas (NSSA) dans OSPFv3.
- Ajout de la capacité de redémarrage gracieux dans OSPFv2 et OSPFv3.
- La bande passante du lien dans le protocole BGP (border gateway protocol) est désormais encodée selon la norme IEEE 754. Pour utiliser la méthode d'encodage précédente, exécutez la commande **neighbor PEER disable-link-bw-encoding-ieee** dans la configuration existante.
- Ajout de la capacité de redémarrage gracieux à long terme dans BGP.
- Mise en œuvre de la communication de fermeture administrative étendue **rfc9003** et de la longueur des paramètres facultatifs étendue **rfc9072** dans BGP.

(BZ#2069563)

Les profils en temps réel de TuneD déterminent désormais automatiquement la configuration initiale de l'isolation du processeur

TuneD est un service qui permet de surveiller votre système et d'en optimiser le profil de performance. Vous pouvez également isoler les unités centrales de traitement (CPU) à l'aide du paquetage **tuned-profiles-realtime** afin de donner aux threads d'application le plus de temps d'exécution possible.

Auparavant, les profils temps réel des systèmes utilisant le noyau temps réel ne se chargeaient pas si vous ne spécifiez pas la liste des CPU à isoler dans le paramètre **isolated_cores**.

Avec cette amélioration, TuneD introduit la fonction intégrée **calc_isolated_cores** qui calcule automatiquement les listes de cœurs de maintien et de cœurs isolés, et applique le calcul au paramètre **isolated_cores**. Avec le préréglage automatique, un cœur de chaque socket est réservé pour le housekeeping, et vous pouvez commencer à utiliser le profil en temps réel sans aucune étape supplémentaire. Si vous souhaitez modifier le préréglage, personnalisez le paramètre **isolated_cores** en spécifiant la liste des CPU à isoler.

(BZ#2093847)

4.7. SÉCURITÉ

Nouveaux paquets : keylime

RHEL 9.1 introduit Keylime, un outil d'attestation des systèmes distants, qui utilise la technologie TPM (trusted platform module). Avec Keylime, vous pouvez vérifier et surveiller en permanence l'intégrité des systèmes distants. Vous pouvez également spécifier des charges utiles cryptées que Keylime délivre aux machines surveillées, et définir des actions automatisées qui se déclenchent chaque fois qu'un système échoue au test d'intégrité.

Pour plus d'informations, voir Ensuring system integrity with Keylime dans le document Security hardening de RHEL 9.

(JIRA:RHELPLAN-92522)

Une nouvelle option dans OpenSSH permet de définir la longueur minimale de la clé RSA

L'utilisation accidentelle de clés RSA courtes rend le système plus vulnérable aux attaques. Avec cette mise à jour, vous pouvez définir la longueur minimale des clés RSA pour les serveurs et les clients OpenSSH. Pour définir la longueur minimale des clés RSA, utilisez la nouvelle option **RequiredRSASize** dans le fichier **/etc/ssh/sshd_config** pour les serveurs OpenSSH et dans le fichier **/etc/ssh/ssh_config** pour les clients OpenSSH.

(BZ#2066882)

crypto-policies imposer une longueur minimale de clé RSA de 2048 bits pour OpenSSH par défaut

L'utilisation de clés RSA courtes rend le système plus vulnérable aux attaques. OpenSSH prenant désormais en charge la limitation de la longueur minimale des clés RSA, les règles cryptographiques applicables à l'ensemble du système imposent par défaut la longueur minimale de 2048 bits pour les clés RSA.

Si OpenSSH échoue les connexions avec un message d'erreur **Invalid key length**, commencez à utiliser des clés RSA plus longues.

Vous pouvez également assouplir cette restriction en utilisant une sous-politique personnalisée, au détriment de la sécurité. Par exemple, si la commande **update-crypto-policies --show** indique que la politique actuelle est **DEFAULT**:

1. Définissez une sous-politique personnalisée en insérant le paramètre **min_rsa_size@openssh = 1024** dans le fichier **/etc/crypto-policies/policies/modules/RSA-OPENSSH-1024.pmod**.

2. Appliquez la sous-politique personnalisée à l'aide de la commande **update-crypto-policies --set DEFAULT:RSA-OPENSSH-1024**.

(BZ#2102774)

Une nouvelle option d'OpenSSL prend en charge SHA-1 pour les signatures

OpenSSL 3.0.0 dans RHEL 9 ne prend pas en charge SHA-1 pour la création et la vérification des signatures par défaut (les fonctions de dérivation de clé SHA-1 (KDF) et les codes d'authentification de message basés sur le hachage (HMAC) sont toujours pris en charge). Toutefois, pour assurer la

rétrocompatibilité avec les systèmes RHEL 8 qui utilisent encore SHA-1 pour les signatures, une nouvelle option de configuration **rh-allow-sha1-signatures** est introduite dans RHEL 9. Cette option, si elle est activée dans **alg_section** de **openssl.cnf**, permet la création et la vérification de signatures SHA-1.

Cette option est automatiquement activée si la politique cryptographique du système LEGACY (pas le fournisseur LEGACY) est définie.

Notez que cela affecte également l'installation de paquets RPM avec des signatures SHA-1, ce qui peut nécessiter le passage à la politique cryptographique du système LEGACY.

(BZ#2060510, BZ#2055796)

crypto-policies soutient désormais sntrup761x25519-sha512@openssh.com

Cette mise à jour des politiques cryptographiques du système ajoute la prise en charge de la méthode d'échange de clés (KEX) **sntrup761x25519-sha512@openssh.com**. L'algorithme post-quantique **sntrup761** est déjà disponible dans la suite OpenSSH, et cette méthode offre une meilleure sécurité contre les attaques des ordinateurs quantiques. Pour activer **sntrup761x25519-sha512@openssh.com**, créez et appliquez une sous-politique, par exemple :

```
# echo 'key_exchange = +SNTRUP' > /etc/crypto-policies/policies/modules/SNTRUP.pmod
# update-crypto-policies --set DEFAULT:SNTRUP
```

Pour plus d'informations, voir la section Personnaliser les stratégies cryptographiques à l'échelle du système avec des sous-politiques dans le document de renforcement de la sécurité de RHEL 9.

(BZ#2070604)

Les NSS ne prennent plus en charge les clés RSA de moins de 1023 bits

La mise à jour des bibliothèques Network Security Services (NSS) modifie la taille minimale des clés pour toutes les opérations RSA de 128 à 1023 bits. Cela signifie que les NSS n'exécutent plus les fonctions suivantes :

- Générer des clés RSA plus courtes que 1023 bits.
- Signer ou vérifier des signatures RSA avec des clés RSA de moins de 1023 bits.
- Chiffrer ou déchiffrer des valeurs avec une clé RSA inférieure à 1023 bits.

(BZ#2091905)

La politique SELinux limite les services supplémentaires

Les paquets **selinux-policy** ont été mis à jour, et par conséquent les services suivants sont maintenant confinés par SELinux :

- **ksm**
- **nm-priv-helper**
- **rhcd**
- **stalld**
- **systemd-network-generator**
- **targetclid**

- **wg-quick**

(BZ#1965013, BZ#1964862, BZ#2020169, BZ#2021131, BZ#2042614, BZ#2053639, BZ#2111069)

SELinux prend en charge le mot-clé self dans les transitions de type

L'outil SELinux prend désormais en charge les règles de transition de type avec le mot-clé **self** dans les sources de politique. La prise en charge des transitions de type avec le mot-clé **self** prépare la politique SELinux à l'étiquetage des inodes anonymes.

(BZ#2069718)

Mise à jour des paquets SELinux pour l'espace utilisateur

Les paquets SELinux pour l'espace utilisateur **libsepol**, **libselinux**, **libsemanage**, **policycoreutils**, **checkpolicy**, et **mcstrans** ont été mis à jour vers la dernière version amont 3.4. Les changements les plus notables sont les suivants :

- Ajout de la prise en charge du réétiquetage parallèle grâce à l'option **-T** dans les outils **setfiles**, **restorecon** et **fixfiles**.
  - Vous pouvez soit spécifier le nombre de threads de processus dans cette option, soit utiliser **-T 0** pour utiliser le maximum de cœurs de processeurs disponibles. Cela permet de réduire considérablement le temps nécessaire au réétiquetage.
- Ajout de la nouvelle option **--checksum**, qui imprime les hashs SHA-256 des modules.
- Ajout de nouveaux utilitaires de politique dans le paquet **libsepol-utils**.

(BZ#2079276)

Le réétiquetage automatique SELinux est désormais parallèle par défaut

Étant donné que l'option de réétiquetage parallèle récemment introduite réduit considérablement le temps nécessaire au processus de réétiquetage SELinux sur les systèmes multicœurs, le script de réétiquetage automatique contient désormais l'option **-T 0** dans la ligne de commande **fixfiles**. L'option **-T 0** garantit que le programme **setfiles** utilise par défaut le maximum de cœurs de processeur disponibles pour le ré-étiquetage.

Pour n'utiliser qu'un seul thread de processus pour le réétiquetage, comme dans la version précédente de RHEL, remplacez ce paramètre en entrant la commande **fixfiles -T 1 onboot** au lieu de **fixfiles onboot** ou la commande **echo "-T 1" > /.autorelabel** au lieu de **touch /.autorelabel**.

(BZ#2115242)

Guide de sécurité SCAP repassé à la version 0.1.63

Les paquets du guide de sécurité SCAP (SSG) ont été rebasés vers la version amont 0.1.63. Cette version apporte diverses améliorations et corrections de bogues, notamment :

- De nouvelles règles de conformité pour **sysctl**, **grub2**, **pam_pwquality**, et la configuration du noyau au moment de la construction ont été ajoutées.
- Les règles de renforcement de la pile PAM utilisent désormais **authselect** comme outil de configuration. Note : Avec ce changement, les règles de renforcement de la pile PAM ne sont pas appliquées si la pile PAM a été modifiée par d'autres moyens.

(BZ#2070563)

Ajout d'une option de taille maximale pour les fichiers d'erreur Rsyslog

La nouvelle option **action.errorfile.maxsize** permet de spécifier le nombre maximal d'octets du fichier d'erreurs pour le système de traitement des journaux Rsyslog. Lorsque le fichier d'erreurs atteint la taille spécifiée, Rsyslog ne peut plus y écrire d'erreurs supplémentaires ou d'autres données. Cela permet d'éviter que le fichier d'erreurs ne remplisse le système de fichiers et ne rende l'hôte inutilisable.

(BZ#2064318)

clevis-luks-askpass est désormais activé par défaut

Le fichier **/lib/systemd/system-preset/90-default.preset** contient désormais l'option de configuration **enable clevis-luks-askpass.path** et l'installation du sous-paquet **clevis-systemd** garantit que le fichier d'unité **clevis-luks-askpass.path** est activé. Cela permet au client de chiffrement Clevis de déverrouiller également les volumes chiffrés LUKS qui se montent tardivement dans le processus de démarrage. Avant cette mise à jour, l'administrateur devait utiliser la commande **systemctl enable clevis-luks-askpass.path** pour permettre à Clevis de déverrouiller ces volumes.

(BZ#2107078)

fapolicyd repassé à la version 1.1.3

Les paquets **fapolicyd** ont été mis à jour vers la version 1.1.3. Les améliorations notables et les corrections de bogues incluent :

- Les règles peuvent désormais contenir le nouvel attribut PPID du sujet, qui correspond au PID parent (ID du processus) d'un sujet.
- La bibliothèque OpenSSL a remplacé la bibliothèque Libgcrypt en tant que moteur cryptographique pour les calculs de hachage.
- La commande **fagenrules --load** fonctionne désormais correctement.

(BZ#2100041)

4.8. MISE EN RÉSEAU

Le module du noyau act_ctinfo a été ajouté

Cette amélioration ajoute le module de noyau **act_ctinfo** à RHEL. En utilisant l'action **ctinfo** de l'utilitaire **tc**, les administrateurs peuvent copier la marque **conntrack** ou la valeur du point de code des services différenciés (DSCP) des paquets réseau dans le champ de métadonnées **mark** de la mémoire tampon de la socket. Par conséquent, vous pouvez utiliser des conditions basées sur la marque **conntrack** ou la valeur DSCP pour filtrer le trafic. Pour plus de détails, voir la page de manuel **tc-ctinfo(8)**.

(BZ#2027894)

cloud-init met à jour la configuration du réseau à chaque démarrage sur Microsoft Azure

Microsoft Azure ne modifie pas l'identifiant de l'instance lorsqu'un administrateur met à jour la configuration de l'interface réseau alors qu'une VM est hors ligne. Grâce à cette amélioration, le service **cloud-init** met toujours à jour la configuration du réseau lorsque la VM démarre afin de garantir que RHEL on Microsoft Azure utilise les derniers paramètres réseau.

Par conséquent, si vous configurez manuellement des paramètres sur les interfaces, tels qu'un domaine de recherche supplémentaire, **cloud-init** peut les remplacer lorsque vous redémarrez la VM. Pour plus de détails et une solution de contournement, voir la solution cloud-init-22.1-5 updates network config on

every boot.

(BZ#2144898)

Le pilote PTP prend désormais en charge les horloges virtuelles et l'horodatage

Grâce à cette amélioration, le pilote Precision Time Protocol (PTP) peut créer des horloges matérielles PTP virtuelles (PHC) au-dessus d'une PHC fonctionnant librement en écrivant à **/sys/class/ptp/ptp*/n_vclocks**. Par conséquent, les utilisateurs peuvent exécuter la synchronisation de plusieurs domaines avec des horodateurs matériels sur une seule interface.

(BZ#2066451)

firewalld est passé à la version 1.1.1

Les paquets **firewalld** ont été mis à jour vers la version 1.1.1. Cette version apporte de nombreuses corrections de bogues et des améliorations par rapport à la version précédente :

Nouvelles fonctionnalités :

- Les règles riches prennent en charge la cible NetFilter-log (NFLOG) pour la journalisation dans l'espace utilisateur. Notez qu'il n'existe pas de démon de journalisation capable de prendre en charge NFLOG dans RHEL. Cependant, vous pouvez utiliser la commande **tcpdump -i nflog** pour collecter les journaux dont vous avez besoin.
- Prise en charge de la redirection de port dans les politiques avec **ingress-zones=HOST** et **egress-zones={ANY, *source based zone*}**.

D'autres changements notables sont à signaler :

- Soutien aux services **afp**, **http3**, **jellyfin**, **netbios-ns**, **ws-discovery**, et **ws-discovery-client**
- Complétion des tabulations et sous-options dans Z Shell pour l'option **policy**

(BZ#2040689)

NetworkManager supporte désormais les attributs de route advmss, rto_min, et quickack

Grâce à cette amélioration, les administrateurs peuvent configurer le paramètre **ipv4.routes** avec les attributs suivants :

- **rto_min** (TIME) - configure le délai minimum de retransmission TCP en millisecondes lors de la communication avec la destination de l'itinéraire
- **quickack** (BOOL) - un paramètre par route pour activer ou désactiver les ACK rapides de TCP
- **advmss** (NOMBRE) - annonce la taille maximale du segment (MSS) à la destination de la route lors de l'établissement de connexions TCP. Si elle n'est pas spécifiée, Linux utilise une valeur par défaut calculée à partir de l'unité de transmission maximale (MTU) du périphérique de premier saut

L'avantage de la mise en œuvre de la nouvelle fonctionnalité de **ipv4.routes** avec les attributs mentionnés est qu'il n'est pas nécessaire d'exécuter le script **dispatcher**.

Notez qu'une fois que vous activez une connexion avec les attributs de route mentionnés, ces changements sont pris en compte dans le noyau.

(BZ#2068525)

Prise en charge de l'option 802.ad vlan-protocol en nmstate

L'API **nmstate** prend désormais en charge la création des interfaces **linux-bridge** à l'aide de l'option 802.ad **vlan-protocol**. Cette fonctionnalité permet la configuration de VLANs Service-Tag. L'exemple suivant illustre l'utilisation de cette fonctionnalité dans un fichier de configuration **yaml**.

```
---
interfaces:
  - name: br0
    type: linux-bridge
    state: up
    bridge:
      options:
        vlan-protocol: 802.1ad
      port:
        - name: eth1
          vlan:
            mode: trunk
            trunk-tags:
            - id: 500
```

(BZ#2084474)

Le service firewalld peut transférer les paquets NAT provenant de l'hôte local vers un autre hôte et un autre port

Vous pouvez transférer les paquets envoyés depuis l'hôte local qui exécute le service **firewalld** vers un port de destination et une adresse IP différents. Cette fonctionnalité est utile, par exemple, pour transférer les ports du périphérique **loopback** vers un conteneur ou une machine virtuelle. Avant cette modification, **firewalld** ne pouvait transférer des ports que lorsqu'il recevait un paquet provenant d'un autre hôte. Pour plus de détails et une configuration illustrative, voir Utilisation de DNAT pour transférer le trafic HTTPS vers un autre hôte.

(BZ#2039542)

NetworkManager supporte maintenant la migration de ifcfg-rh vers le fichier clé

Les utilisateurs peuvent migrer leurs fichiers de profil de connexion existants du format **ifcfg-rh** vers le format de fichier clé. De cette manière, tous les profils de connexion se trouveront à un seul endroit et dans le format préféré. Le format de fichier clé présente les avantages suivants :

- Ressemble beaucoup à la façon dont NetworkManager exprime la configuration du réseau
- Garantie de compatibilité avec les futures versions de RHEL
- Est plus facile à lire
- Prise en charge de tous les profils de connexion

Pour migrer les connexions, exécutez :

```
# nmcli connection migrate
```

Notez que les fichiers **ifcfg-rh** fonctionneront correctement pendant la durée de vie de RHEL 9. Toutefois, la migration de la configuration vers le format de fichier clé garantit la compatibilité au-delà de RHEL 9.

Pour plus de détails, consultez les pages de manuel **nmcli(1)**, nm-settings-keyfile(5), et nm-settings-ifcfg-rh(5).

(BZ#2059608)

D'autres attributs d'auto-configuration DHCP et IPv6 ont été ajoutés à l'API nmstate

Cette amélioration ajoute la prise en charge des attributs suivants à l'API nmstate :

- **dhcp-client-id** pour les connexions DHCPv4, comme décrit dans les RFC 2132 et 4361.
- **dhcp-duid** pour les connexions DHCPv6, comme décrit dans la RFC 8415.
- **addr-gen-mode** pour l'auto-configuration d'IPv6. Vous pouvez définir cet attribut sur :
  - **eui64** comme décrit dans le RFC 4862
  - **stable-privacy** comme décrit dans le RFC 7217

(BZ#2082043)

NetworkManager indique désormais clairement que la prise en charge du WEP n'est pas disponible dans RHEL 9

Les paquets **wpa_supplicant** de RHEL 9.0 et des versions ultérieures ne contiennent plus l'algorithme de sécurité WEP (Wired Equivalent Privacy), qui est obsolète et peu sûr. Cette amélioration met à jour NetworkManager pour refléter ces changements. Par exemple, la commande **nmcli device wifi list** renvoie désormais les points d'accès WEP à la fin de la liste en couleur grise, et la connexion à un réseau protégé par WEP renvoie un message d'erreur significatif.

Pour un cryptage sécurisé, n'utilisez que des réseaux wifi avec authentification Wi-Fi Protected Access 2 (WPA2) et WPA3.

(BZ#2030997)

Le code MPTCP a été mis à jour

Le code MultiPath TCP (MPTCP) dans le noyau a été mis à jour en amont de Linux 5.19. Cette mise à jour apporte un certain nombre de corrections de bogues et d'améliorations par rapport à la version précédente :

- L'option **FASTCLOSE** a été ajoutée pour fermer les connexions MPTCP sans une poignée de main tripartite complète.
- L'option **MP_FAIL** a été ajoutée pour permettre le repli sur TCP même après la poignée de main initiale.
- Les capacités de surveillance ont été améliorées par l'ajout de compteurs supplémentaires de la base d'informations de gestion (MIB).
- La prise en charge de la surveillance des sockets d'écoute MPTCP a été ajoutée. Utilisez l'utilitaire **ss** pour surveiller les sockets.

(BZ#2079368)

4.9. NOYAU

Version du noyau dans RHEL 9.1

Red Hat Enterprise Linux 9.1 est distribué avec la version 5.14.0-162 du noyau.

(BZ#2125549)

La consommation de mémoire du site list_lru a été optimisée

La structure de données interne du noyau, **list_lru**, suit l'état \N "Least Recently Used" (le moins récemment utilisé) des inodes du noyau et des entrées de répertoire pour les fichiers. Auparavant, le nombre de structures allouées **list_lru** était directement proportionnel au nombre de points de montage et au nombre de mémoires présentes **cgroups**. Ces deux chiffres augmentaient avec le nombre de conteneurs en cours d'exécution, ce qui entraînait une consommation de mémoire de **O(n^2)** où **n** est le nombre de conteneurs en cours d'exécution. Cette mise à jour optimise la consommation de mémoire de **list_lru** dans le système à **O(n)**. Par conséquent, les applications des utilisateurs disposent désormais de suffisamment de mémoire, en particulier sur les systèmes où un grand nombre de conteneurs sont en cours d'exécution.

(BZ#2013413)

BPF rebasé sur la version 5.16 du noyau Linux

Le filtre de paquets Berkeley (BPF) a été rebasé sur la version 5.16 du noyau Linux avec de nombreuses corrections de bogues et améliorations. Les changements les plus notables sont les suivants :

- Rationalisation de la gestion interne des sections du programme BPF et de l'API **bpf_program__set_attach_target()** dans la bibliothèque userspace **libbpf**. L'API **bpf_program__set_attach_target()** définit les objectifs de rattachement basés sur le BTF pour les programmes basés sur le FBP.
- Ajout de la prise en charge du type **BTF_KIND_TAG**, qui permet de baliser les déclarations.
- Ajout de la prise en charge de l'aide **bpf_get_branch_snapshot()**, qui permet au programme de traçage de capturer les derniers enregistrements de branche (LBR) à partir du matériel.
- Ajout de la prise en charge des anciens événements **kprobe** dans la bibliothèque de l'espace utilisateur **libbpf**, qui permet la création d'événements de points de contrôle **kprobe** par l'intermédiaire de l'interface existante.
- Ajout de la possibilité d'accéder aux horodatages matériels par le biais de structures spécifiques à BPF avec la fonction d'aide **__sk_buff**.
- Ajout de la prise en charge d'une interface par lots pour l'allocation de tampons RX dans le pool de tampons **AF_XDP**, avec la prise en charge des pilotes pour **i40e** et **ice**.
- Ajout du support de l'ancienne version de **uprobe** dans la bibliothèque de l'espace utilisateur de **libbpf** pour compléter l'ancienne version de **kprobe** récemment fusionnée.
- Ajout de l'aide **bpf_trace_vprintk()** comme aide variadique **printk**.
- Ajout de l'option **libbpf** pour une gestion plus stricte des noms de sections de programmes BPF dans le cadre de l'effort **libbpf** 1.0.
- Ajout du support **libbpf** pour localiser les cartes spécialisées, telles que **perf RB** et supprimer en interne les identifiants de type BTF lors de leur création.
- Ajout du type de carte **bloomfilter** BPF pour tester l'existence d'un élément dans un ensemble.
- Ajout de la prise en charge des appels de fonctions de modules du noyau à partir de BPF.

- Ajout de la prise en charge de **ksym** sans type et faible dans le squelette léger.
- Ajout de la prise en charge du type **BTF_KIND_DECL_TAG**.

Pour plus d'informations sur la liste complète des fonctionnalités BPF disponibles dans le noyau en cours d'exécution, utilisez la commande **bpftool feature**.

(BZ#2069045)

Les données BTF se trouvent désormais dans le module du noyau

BPF Type Format (BTF) est le format de métadonnées qui encode les informations de débogage relatives au programme et à la carte BPF. Auparavant, les données BTF pour les modules du noyau étaient stockées dans le paquetage **kernel-debuginfo**. Par conséquent, il était nécessaire d'installer le paquetage **kernel-debuginfo** correspondant afin d'utiliser le BTF pour les modules du noyau. Avec cette mise à jour, les données BTF sont maintenant situées directement dans le module du noyau. Par conséquent, vous n'avez pas besoin d'installer d'autres paquets pour que BTF fonctionne.

(BZ#2097188)

L'arborescence des sources de kernel-rt a été mise à jour vers l'arborescence RHEL 9.1

Les sources de **kernel-rt** ont été mises à jour pour utiliser la dernière arborescence des sources du noyau Red Hat Enterprise Linux. Le jeu de correctifs en temps réel a également été mis à jour vers la dernière version en amont, **v5.15-rt**. Ces mises à jour apportent un certain nombre de corrections de bogues et d'améliorations.

(BZ#2061574)

Programmation dynamique préemptive activée sur les architectures ARM, AMD et Intel 64 bits

RHEL 9 offre la fonction de planification dynamique sur les architectures ARM, AMD et Intel 64 bits. Cette amélioration permet de modifier le mode de préemption du noyau au démarrage ou au moment de l'exécution plutôt qu'au moment de la compilation. Le fichier **/sys/kernel/debug/sched/preempt** contient la configuration actuelle et permet de modifier **runtime**.

En utilisant l'option **DYNAMIC_PREEMPT**, vous pouvez définir la variable **preempt=** au démarrage sur **none**, **voluntary** ou **full**, la préemption **voluntary** étant la valeur par défaut. Grâce à la gestion dynamique de la préemption, vous pouvez remplacer le modèle de préemption par défaut afin d'améliorer la latence de l'ordonnancement.

(BZ#2065226)

stalld repassé à la version 1.17

Le programme **stalld**, qui fournit le démon **stall**, est un mécanisme permettant d'éviter l'état de famine des threads du système d'exploitation dans un système Linux. Cette version surveille les threads pour l'état de famine. L'état de famine survient lorsqu'un thread se trouve dans la file d'attente d'exécution du processeur pendant une durée supérieure au seuil de famine.

Cette version **stalld** comprend de nombreuses améliorations et corrections de bogues par rapport à la version précédente. Le changement le plus notable est la possibilité de détecter les tâches mourantes exécutables.

Lorsque **stalld** détecte un thread affamé, le programme change la classe d'ordonnancement du thread pour la politique **SCHED_DEADLINE**, qui donne au thread une petite tranche de temps pour que l'unité centrale spécifiée l'exécute. Lorsque **timeslice** est utilisé, le thread revient à sa politique

d'ordonnancement d'origine et **stalld** continue à surveiller les états du thread.

(BZ#2107275)

Le paquet tpm2-tools a été rebasé sur la version tpm2-tools-5.2-1

Le paquetage **tpm2-tools** a été rebasé vers la version **tpm2-tools-5.2-1**. Cette mise à jour apporte de nombreuses améliorations significatives et des corrections de bogues. Les changements les plus notables sont les suivants :

- Prise en charge de l'émission de clés publiques lors de la création d'objets primaires à l'aide des outils **tpm2_createprimary** et **tpm2_create**.
- Prise en charge de l'outil **tpm2_print** pour l'impression des formats de sortie des clés publiques. **tpm2_print** décode une structure de données Trusted Platform Module (TPM) et imprime les éléments inclus.
- L'outil **tpm2_eventlog** prend en charge la lecture des journaux de plus de 64 Ko.
- Ajout de l'outil **tpm2_sessionconfig** qui permet d'afficher et de configurer les attributs de la session.

Pour plus d'informations sur les changements notables, voir le fichier **/usr/share/doc/tpm2-tools/Changelog.md**.

(BZ#2090748)

Les appareils Intel E800 prennent désormais en charge les protocoles iWARP et RoCE

Avec cette amélioration, vous pouvez désormais utiliser les paramètres **enable_iwarp** et **enable_roce** devlink pour activer et désactiver la prise en charge des protocoles iWARP ou RoCE. Grâce à cette fonctionnalité obligatoire, vous pouvez configurer l'appareil avec l'un des protocoles. Les appareils Intel E800 ne supportent pas les deux protocoles simultanément sur le même port.

Pour activer ou désactiver le protocole iWARP pour un appareil E800 spécifique, il faut d'abord obtenir l'emplacement PCI de la carte :

```
$ lspci | awk '/E810/ {print $1}'
44:00.0
44:00.1
$
```

Ensuite, activez ou désactivez le protocole. Vous pouvez utiliser **pci/0000:44:00.0** pour le premier port, et **pci/0000:44:00.1** pour le second port de la carte comme argument à la commande devlink

```
$ devlink dev param set pci/0000:44:00.0 name enable_iwarp value true cmode runtime
$ devlink dev param set pci/0000:44:00.0 name enable_iwarp value false cmode runtime
```

Pour activer ou désactiver le protocole RoCE pour un périphérique E800 spécifique, obtenez l'emplacement PCI de la carte comme indiqué ci-dessus. Utilisez ensuite l'une des commandes suivantes :

```
$ devlink dev param set pci/0000:44:00.0 name enable_roce value true cmode runtime
$ devlink dev param set pci/0000:44:00.0 name enable_roce value false cmode runtime
```

(BZ#2096127)

4.10. CHARGEUR DE DÉMARRAGE

GRUB est signé par de nouvelles clés

Pour des raisons de sécurité, GRUB est désormais signé par de nouvelles clés. Par conséquent, vous devez mettre à jour le micrologiciel RHEL vers la version FW1010.30 (ou ultérieure) ou FW1020 pour pouvoir démarrer la variante little-endian des systèmes IBM Power avec la fonction Secure Boot activée.

(BZ#2074761)

4.11. SYSTÈMES DE FICHIERS ET STOCKAGE

Stratis permet désormais de définir la taille du système de fichiers lors de sa création

Vous pouvez désormais définir la taille requise lors de la création d'un système de fichiers. Auparavant, la taille automatique par défaut était de 1 TiB. Avec cette amélioration, les utilisateurs peuvent définir une taille de système de fichiers arbitraire. La limite inférieure ne doit pas être inférieure à 512 MiB.

(BZ#1990905)

Amélioration de la gestion de l'overprovisionnement des pools Stratis

Grâce aux améliorations apportées à la gestion du thin provisioning, vous pouvez désormais bénéficier de meilleurs avertissements, d'une allocation précise de l'espace pour les métadonnées du pool, d'une meilleure prévisibilité, d'une sécurité globale et d'une fiabilité accrue de la gestion du thin pool. Un nouveau mode distinct désactive l'overprovisioning. Grâce à cette amélioration, l'utilisateur peut désactiver l'overprovisioning pour s'assurer qu'un pool contient suffisamment d'espace pour prendre en charge tous ses systèmes de fichiers, même si ceux-ci sont complètement pleins.

(BZ#2040352)

Stratis offre désormais une meilleure gestion des piscines individuelles

Vous pouvez désormais arrêter et démarrer des pools Stratis individuels arrêtés. Auparavant, **stratisd** tentait de démarrer tous les pools disponibles pour tous les périphériques qu'il détectait. Cette amélioration permet une gestion plus souple des pools individuels au sein de Stratis, ainsi que de meilleures capacités de débogage et de récupération. Le système ne nécessite plus de redémarrage pour effectuer des opérations de récupération et de maintenance pour un seul pool.

(BZ#2039960)

Activation de la configuration spécifique au protocole des chemins d'accès des dispositifs à trajets multiples

Auparavant, en raison des différentes configurations optimales pour les différents protocoles, il était impossible de définir la configuration correctement sans définir une option pour chaque protocole individuel. Grâce à cette amélioration, les utilisateurs peuvent désormais configurer les chemins d'accès des dispositifs à trajets multiples en fonction de leur protocole de transport. Utilisez la sous-section **protocol** de la section **overrides** dans le fichier **/etc/multipath.conf** pour configurer correctement les chemins d'accès des périphériques à trajets multiples en fonction de leur protocole.

(BZ#2084365)

Nouvelle bibliothèque de fonctionnalités libnvme

Auparavant, l'utilitaire de l'interface de ligne de commande du stockage NVMe (**nvme-cli**) incluait toutes les fonctions et définitions d'aide. Cette amélioration apporte une nouvelle bibliothèque **libnvme** à RHEL 9.1. Cette bibliothèque comprend les éléments suivants

- Définitions de types pour les structures de la spécification NVMe
- Enumérations et champs de bits
- Fonctions d'aide pour construire, distribuer et décoder les commandes et les charges utiles
- Utilitaires pour connecter, scanner et gérer les périphériques NVMe

Avec cette mise à jour, les utilisateurs n'ont pas besoin de dupliquer le code dans plusieurs projets et paquets, tels que **nvme-stas**, et peuvent s'appuyer sur cette bibliothèque commune.

(BZ#2099619)

Une nouvelle bibliothèque libnvme est maintenant disponible

With this update, nvme-cli is divided in two different projects: * **nvme-cli** now only contains the code specific to the **nvme** tool * **libnvme** library now contains all type definitions for NVMe specification structures, enumerations, bit fields, helper functions to construct, dispatch, decode commands and payloads, and utilities to connect, scan, and manage NVMe devices.

(BZ#2090121)

4.12. HAUTE DISPONIBILITÉ ET CLUSTERS

Prise en charge de la haute disponibilité sur la plateforme Red Hat OpenStack

Vous pouvez désormais configurer un cluster de haute disponibilité sur la plateforme Red Hat OpenStack. Pour prendre en charge cette fonctionnalité, Red Hat fournit les nouveaux agents de cluster suivants :

- **fence_openstack**agent de clôture pour les clusters HA sur OpenStack
- **openstack-info**l'agent de ressource **openstack-info** pour configurer la ressource clonée, qui est nécessaire pour un cluster HA sur OpenStack
- **openstack-virtual-ip**agent de ressource pour configurer une ressource d'adresse IP virtuelle
- **openstack-floating-ip**agent de ressources pour configurer une ressource d'adresse IP flottante
- **openstack-cinder-volume**l'agent de ressources pour configurer une ressource de stockage en bloc

(BZ#2121838)

pcs prend en charge la mise à jour des périphériques SCSI à chemins multiples sans nécessiter un redémarrage du système

Vous pouvez désormais mettre à jour les périphériques SCSI multipath à l'aide de la commande **pcs stonith update-scsi-devices**. Cette commande met à jour les périphériques SCSI sans provoquer le redémarrage des autres ressources de la grappe fonctionnant sur le même nœud.

(BZ#2024522)

Prise en charge de l'UUID des clusters

Lors de l'installation d'un cluster, la commande **pcs** génère désormais un UUID pour chaque cluster. Comme le nom d'un cluster n'est pas un identifiant unique, vous pouvez utiliser l'UUID du cluster pour identifier les clusters portant le même nom lorsque vous administrez plusieurs clusters.

Vous pouvez afficher l'UUID du cluster actuel à l'aide de la commande **pcs cluster config [show]**. Vous pouvez ajouter un UUID à un cluster existant ou régénérer un UUID s'il existe déjà en utilisant la commande **pcs cluster config uuid generate**.

(BZ#2054671)

Nouvelle option de commande pcs resource config pour afficher les commandes pcs qui recréent les ressources configurées

La commande **pcs resource config** accepte désormais l'option **--output-format=cmd**. Cette option permet d'afficher les commandes **pcs** que vous pouvez utiliser pour recréer les ressources configurées sur un autre système.

(BZ#2058251)

Nouvelle option de commande pcs stonith config pour afficher les commandes pcs qui recréent les dispositifs de clôture configurés

La commande **pcs stonith config** accepte désormais l'option **--output-format=cmd**. Cette option permet d'afficher les commandes **pcs** que vous pouvez utiliser pour recréer les périphériques de clôture configurés sur un autre système.

(BZ#2058252)

Pacemaker passe à la version 2.1.4

Les paquets Pacemaker ont été mis à jour vers la version amont de Pacemaker 2.1.4. Les changements notables sont les suivants :

- Le paramètre de ressource **multiple-active** accepte désormais la valeur **stop_unexpected**, Le paramètre de ressource **multiple-active** détermine le comportement de récupération lorsqu'une ressource est active sur plus d'un nœud alors qu'elle ne devrait pas l'être. Par défaut, cette situation nécessite un redémarrage complet de la ressource, même si la ressource fonctionne correctement là où elle devrait être. Une valeur de **stop_unexpected** pour ce paramètre indique que seules les instances inattendues d'une ressource à activité multiple sont arrêtées. Il incombe à l'utilisateur de vérifier que le service et son agent de ressources peuvent fonctionner avec des instances actives supplémentaires sans nécessiter un redémarrage complet.

- Pacemaker prend désormais en charge le méta-attribut de ressource **allow-unhealthy-node**. Lorsque ce méta-attribut est défini sur **true**, la ressource n'est pas forcée de quitter un nœud en raison de la dégradation de son état de santé. Lorsque cet attribut est défini pour les ressources de santé, le cluster peut automatiquement détecter si l'état de santé du nœud se rétablit et déplacer les ressources vers ce nœud.

- Les utilisateurs peuvent désormais spécifier des listes de contrôle d'accès (ACLS) pour un groupe de systèmes à l'aide de la commande **pcs acl group**. Pacemaker permettait auparavant de spécifier des ACL pour des utilisateurs individuels, mais il est parfois plus simple et plus conforme aux politiques locales de spécifier des ACL pour un groupe de systèmes et de les appliquer à tous les utilisateurs de ce groupe. Cette commande était présente dans les versions antérieures mais n'avait aucun effet.

(BZ#2072108)

Samba n'est plus installé automatiquement avec les paquets cluster

À partir de cette version, l'installation des paquets pour le module complémentaire de haute disponibilité RHEL n'installe plus automatiquement les paquets Samba. Cela vous permet également de supprimer les paquets Samba sans supprimer automatiquement les paquets HA. Si votre cluster utilise des ressources Samba, vous devez maintenant les installer manuellement.

(BZ#1826455)

4.13. LANGAGES DE PROGRAMMATION DYNAMIQUES, SERVEURS WEB ET DE BASE DE DONNÉES

Le flux de modules nodejs:18 est désormais entièrement pris en charge

Le flux du module **nodejs:18**, précédemment disponible en tant qu'aperçu technologique, est entièrement pris en charge avec la publication de l'avis RHSA-2022:8832. Le flux de modules **nodejs:18** fournit désormais **Node.js 18.12**, qui est une version de support à long terme (LTS).

Node.js 18 inclus dans RHEL 9.1 apporte de nombreuses nouvelles fonctionnalités ainsi que des corrections de bogues et de sécurité sur **Node.js 16**.

Les changements les plus notables sont les suivants :

- Le moteur **V8** est passé à la version 10.2.
- Le gestionnaire de paquets **npm** a été mis à jour vers la version 8.19.2.
- **Node.js** propose désormais une nouvelle API expérimentale **fetch**.
- **Node.js** propose désormais un nouveau module expérimental **node:test**, qui facilite la création de tests dont les résultats sont présentés dans le format Test Anything Protocol (TAP).
- **Node.js** préfère désormais les adresses IPv6 aux adresses IPv4.

Pour installer le flux du module **nodejs:18**, utilisez :

```
# dnf module install nodejs:18
```

(BZ#2083072)

Un nouveau flux de modules : php:8.1

RHEL 9.1 ajoute **PHP 8.1** en tant que nouveau flux de modules **php:8.1**.

Avec **PHP 8.1**, vous pouvez :

- Définir un type personnalisé limité à une valeur parmi un nombre discret de valeurs possibles à l'aide de la fonctionnalité des énumérations (Enums)
- Déclarer une propriété avec le modificateur **readonly** pour empêcher la modification de la propriété après l'initialisation
- Utiliser des fibres, des fonctions interruptibles à pile complète

Pour installer le flux du module **php:8.1**, utilisez :

```
# dnf module install php:8.1
```

Pour plus d'informations sur l'utilisation de PHP sur RHEL 9, voir Utilisation du langage de script PHP.

(BZ#2070040)

Un nouveau flux de modules : ruby:3.1

RHEL 9.1 introduit **Ruby 3.1.2** dans un nouveau flux de modules **ruby:3.1**. Cette version apporte un certain nombre d'améliorations de performance, de corrections de bogues et de sécurité, ainsi que de nouvelles fonctionnalités par rapport à **Ruby 3.0** distribué avec RHEL 9.0.

Parmi les améliorations notables, citons

- L'utilitaire **Interactive Ruby** (IRB) propose désormais une fonction d'autocomplétion et une boîte de dialogue de documentation
- Une nouvelle gemme **debug**, qui remplace **lib/debug.rb**, améliore les performances et prend en charge le débogage à distance et le débogage multiprocessus/multithread
- La gem **error_highlight** fournit maintenant une localisation fine des erreurs dans le backtrace
- Les valeurs des types de données littérales de hachage et des arguments de mots clés peuvent désormais être omises
- L'opérateur d'épinglage (**^**) accepte désormais une expression dans la recherche de motifs
- Les parenthèses peuvent désormais être omises dans la recherche de motifs sur une seule ligne
- YJIT, un nouveau compilateur expérimental Just-in-Time (JIT) in-process, est désormais disponible sur les architectures AMD et Intel 64 bits
- L'utilitaire **TypeProf For IDE** a été introduit. Il s'agit d'un outil expérimental d'analyse statique de type pour le code **Ruby** dans les IDE

Les améliorations suivantes ont été apportées au compilateur Just-in-Time basé sur la méthode (MJIT) :

- Pour les charges de travail telles que **Rails**, la valeur maximale par défaut du cache JIT est passée de 100 à 10000
- Le code compilé à l'aide de JIT n'est plus annulé lorsque l'option **TracePoint** pour les événements de classe est activée

D'autres changements notables sont à signaler :

- Le fichier **tracer.rb** a été supprimé
- Depuis la version 4.0, l'analyseur YAML de **Psych** utilise par défaut la méthode **safe_load**

Pour installer le flux du module **ruby:3.1**, utilisez :

```
# dnf module install ruby:3.1
```

(BZ#2063773)

httpd repassé à la version 2.4.53

Le serveur HTTP Apache a été mis à jour vers la version 2.4.53, qui apporte des corrections de bogues, des améliorations et des correctifs de sécurité par rapport à la version 2.4.51 distribuée avec RHEL 9.0.

Les changements notables apportés aux modules **mod_proxy** et **mod_proxy_connect** sont les suivants :

- **mod_proxy**: La limite de longueur du nom du contrôleur a été augmentée
- **mod_proxy**: Vous pouvez maintenant configurer sélectivement les délais d'attente pour le backend et le frontend
- **mod_proxy**: Vous pouvez maintenant désactiver la redirection des connexions TCP en définissant le paramètre **SetEnv proxy-nohalfclose**
- **mod_proxy** et **mod_proxy_connect**: Il est interdit de modifier un code de statut après l'avoir envoyé à un client

En outre, une nouvelle fonction **ldap** a été ajoutée à l'API d'expression, ce qui peut aider à prévenir la vulnérabilité de l'injection LDAP.

(BZ#2079939)

Une nouvelle valeur par défaut pour la directive LimitRequestBody dans la configuration de httpd

Pour corriger CVE-2022-29404, la valeur par défaut de la directive **LimitRequestBody** dans le serveur HTTP Apache a été modifiée de **0** (illimité) à 1 GiB.

Sur les systèmes où la valeur de **LimitRequestBody** n'est pas explicitement spécifiée dans un fichier de configuration **httpd**, la mise à jour du paquet **httpd** fixe **LimitRequestBody** à la valeur par défaut de 1 GiB. Par conséquent, si la taille totale du corps de la requête HTTP dépasse cette limite par défaut de 1 GiB, **httpd** renvoie le code d'erreur **413 Request Entity Too Large**.

Si la nouvelle taille autorisée par défaut du corps d'un message de requête HTTP est insuffisante pour votre cas d'utilisation, mettez à jour vos fichiers de configuration **httpd** dans le contexte respectif (serveur, par répertoire, par fichier ou par emplacement) et définissez votre limite préférée en octets. Par exemple, pour définir une nouvelle limite de 2 GiB, utilisez :

```
LimitRequestBody 2147483648
```

Les systèmes déjà configurés pour utiliser n'importe quelle valeur explicite pour la directive **LimitRequestBody** ne sont pas affectés par ce changement.

(BZ#2128016)

Nouveau paquet : httpd-core

À partir de RHEL 9.1, le fichier binaire **httpd** contenant tous les fichiers essentiels a été déplacé vers le nouveau paquet **httpd-core** afin de limiter les dépendances du serveur HTTP Apache dans les scénarios où seule la fonctionnalité de base **httpd** est nécessaire, par exemple, dans les conteneurs.

Le paquet **httpd** fournit désormais des fichiers liés à **systemd**, notamment **mod_systemd**, **mod_brotli**, et de la documentation.

Avec cette modification, le paquet **httpd** ne fournit plus la valeur du numéro magique du module (MMN) **httpd**. C'est le paquet **httpd-core** qui fournit désormais la valeur **httpd-mmn**. Par conséquent, il n'est plus possible de récupérer **httpd-mmn** à partir du paquet **httpd**.

Pour obtenir la valeur **httpd-mmn** du binaire **httpd** installé, vous pouvez utiliser le binaire **apxs**, qui fait partie du paquetage **httpd-devel**. Pour obtenir la valeur de **httpd-mmn**, utilisez la commande suivante :

```
# apxs -q HTTPD_MMN
20120211
```

(BZ#2065677)

pcre2 repassé à la version 10.40

Le paquetage **pcre2**, qui fournit la bibliothèque Perl Compatible Regular Expressions v2, a été mis à jour à la version 10.40.

Avec cette mise à jour, l'utilisation de la séquence d'échappement **\K** dans les assertions de type "lookaround" est interdite, conformément au changement correspondant dans **Perl 5.32**. Si vous comptez sur le comportement précédent, vous pouvez utiliser l'option **PCRE2_EXTRA_ALLOW_LOOKAROUND_BSK**. Notez que lorsque cette option est activée, **\K** n'est accepté que dans les assertions positives et est ignoré dans les assertions négatives.

(BZ#2086494)

4.14. COMPILATEURS ET OUTILS DE DÉVELOPPEMENT

Le compilateur GCC mis à jour est désormais disponible pour RHEL 9.1

Le compilateur GCC du système, version 11.2.1, a été mis à jour pour inclure de nombreuses corrections de bogues et améliorations disponibles dans le GCC en amont.

La collection de compilateurs GNU (GCC) fournit des outils pour développer des applications avec les langages de programmation C, C , et Fortran.

Pour plus d'informations sur l'utilisation, voir Développement d'applications C et C dans RHEL 9 .

(BZ#2063255)

Nouveau jeu d'outils GCC 12

GCC Toolset 12 est un ensemble d'outils de compilation qui fournit des versions récentes d'outils de développement. Il est disponible en tant que flux d'application sous la forme d'une collection de logiciels dans le dépôt **AppStream**.

Le compilateur GCC a été mis à jour vers la version 12.1.1, qui apporte de nombreuses corrections de bogues et améliorations disponibles dans la version amont de GCC.

Les outils et versions suivants sont fournis par le Toolset 12 de GCC :

Outil	Version
CCG	12.1.1
GDB	11.2
binutils	2.35

Outil	Version
dwz	0.14
annobin	10.76

Pour installer GCC Toolset 12, exécutez la commande suivante en tant que root :

```
# dnf install gcc-toolset-12
```

Pour exécuter un outil du GCC Toolset 12 :

```
$ scl enable gcc-toolset-12 tool
```

Pour lancer une session shell dans laquelle les versions des outils du Toolset 12 de GCC remplacent les versions système de ces outils :

```
$ scl enable gcc-toolset-12 bash
```

Pour plus d'informations, voir GCC Toolset 12.

(BZ#2077465)

GCC Toolset 12 : Annobin rebasé à la version 10.76

Dans GCC Toolset 12, le paquetage Annobin a été mis à jour à la version 10.76.

Les corrections de bogues et les améliorations les plus notables sont les suivantes

- Une nouvelle option de ligne de commande pour annocheck lui permet d'éviter d'utiliser le service **debuginfod**, s'il n'est pas en mesure de trouver des informations de débogage d'une autre manière. L'utilisation de **debuginfod** permet à annocheck d'obtenir plus d'informations, mais elle peut également entraîner des ralentissements importants des performances d'annocheck si le serveur **debuginfod** n'est pas disponible.
- Les sources d'Annobin peuvent maintenant être compilées en utilisant **meson** et **ninja** plutôt que configure et make si désiré.
- Annocheck supporte désormais les binaires construits par le compilateur Rust 1.18.

En outre, le problème connu suivant a été signalé dans la version GCC Toolset 12 d'Annobin :

Dans certaines circonstances, il est possible qu'une compilation échoue avec un message d'erreur similaire au suivant :

```
cc1: fatal error: inaccessible plugin file
opt/rh/gcc-toolset-12/root/usr/lib/gcc/architecture-linux-gnu/12/plugin/gcc-annobin.so
expanded from short plugin name gcc-annobin: No such file or directory
```

Pour contourner le problème, créez un lien symbolique dans le répertoire du plugin de **annobin.so** vers **gcc-annobin.so**:

```
# cd /opt/rh/gcc-toolset-12/root/usr/lib/gcc/architecture-linux-gnu/12/plugin
# ln -s annobin.so gcc-annobin.so
```

Où *architecture* est remplacé par l'architecture utilisée :

- **aarch64**
- **i686**
- **ppc64le**
- **s390x**
- **x86_64**

(BZ#2077438)

GCC Toolset 12 : binutils rebasé à la version 2.38

Dans GCC Toolset 12, le paquet **binutils** a été mis à jour à la version 2.38.

Les corrections de bogues et les améliorations les plus notables sont les suivantes

- Tous les outils du paquet **binutils** prennent désormais en charge les options permettant d'afficher ou d'avertir de la présence de caractères multi-octets.
- Les outils **readelf** et **objdump** suivent désormais automatiquement et par défaut tous les liens vers des fichiers **debuginfo** distincts. Ce comportement peut être désactivé en utilisant l'option **--debug-dump=no-follow-links** pour **readelf** ou l'option **--dwarf=no-follow-links** pour **objdump**.

(BZ#2077445)

GCC 12 et les versions ultérieures prennent en charge _FORTIFY_SOURCE niveau 3

Avec cette amélioration, les utilisateurs peuvent construire des applications avec **-D_FORTIFY_SOURCE=3** dans la ligne de commande du compilateur lorsqu'ils construisent avec la version 12 ou ultérieure de GCC. **_FORTIFY_SOURCE** niveau 3 améliore la couverture de la fortification du code source, améliorant ainsi la sécurité des applications construites avec **-D_FORTIFY_SOURCE=3** dans la ligne de commande du compilateur. Cette fonctionnalité est prise en charge dans les versions 12 et ultérieures de GCC et dans tous les Clang de RHEL 9 avec l'intégration **__builtin_dynamic_object_size**.

(BZ#2033683)

L'option DNS stub resolver supporte désormais l'option no-aaaa

Avec cette amélioration, **glibc** reconnaît désormais l'option **no-aaaa** stub resolver dans **/etc/resolv.conf** et la variable d'environnement **RES_OPTIONS**. Lorsque cette option est active, aucune requête AAAA n'est envoyée sur le réseau. Les administrateurs système peuvent désactiver les recherches DNS AAAA à des fins de diagnostic, par exemple pour s'assurer que les recherches superflues sur les réseaux exclusivement IPv4 ne contribuent pas aux problèmes DNS.

(BZ#2096191)

Ajout de la prise en charge de la série IBM Z z16

La prise en charge du jeu d'instructions **s390** est désormais disponible sur la plate-forme **IBM z16**. **IBM**

z16 fournit deux capacités matérielles supplémentaires dans **glibc**, à savoir **HWCAP_S390_VXRS_PDE2** et **HWCAP_S390_NNPA**. Par conséquent, les applications peuvent désormais utiliser ces capacités pour fournir des bibliothèques et des fonctions optimisées.

(BZ#2077838)

Les applications peuvent utiliser les caractéristiques de la séquence redémarrable par l'intermédiaire des nouvelles interfaces glibc

Pour accélérer la fonction **sched_getcpu** (en particulier sur aarch64), il est nécessaire d'utiliser la fonction noyau de séquences redémarrables (rseq) par défaut dans **glibc**. Pour permettre aux applications d'utiliser en permanence la zone rseq partagée, **glibc** fournit maintenant les symboles **__rseq_offset**, **__rseq_size** et **__rseq_flags** qui ont été ajoutés pour la première fois dans la version amont **glibc** 2.35. Grâce à cette amélioration, les performances de la fonction **sched_getcpu** sont accrues et les applications peuvent désormais utiliser les caractéristiques de la séquence redémarrable par l'intermédiaire des nouvelles interfaces **glibc**.

(BZ#2085529)

GCC Toolset 12 : GDB repassé à la version 11.2

Dans GCC Toolset 12, le paquetage GDB a été mis à jour à la version 11.2.

Les corrections de bogues et les améliorations les plus notables sont les suivantes

- Nouvelle prise en charge de l'architecture ARM 64 bits Memory Tagging Extension (MTE). Voir les nouvelles commandes avec le préfixe **memory-tag**.
- **--qualified** pour **-break-insert** et **-dprintf-insert**. Cette option recherche une correspondance exacte avec le lieu de l'événement de l'utilisateur au lieu d'effectuer une recherche dans tous les champs d'application.
  Par exemple, **break --qualified *foo*** recherchera un symbole nommé *foo* dans la portée globale. Sans **--qualified**, GDB recherchera un symbole portant ce nom dans toutes les portées.
- **--force-condition**: Toute condition fournie est définie même si elle est actuellement invalide.
- **-break-condition --force**: De même pour la commande MI.
- **-file-list-exec-source-files** accepte l'option **REGEXP** pour limiter la production.
- **.gdbinit** le chemin de recherche inclut le répertoire de configuration. L'ordre est le suivant :
  a. **$XDG_CONFIG_HOME/gdb/gdbinit**
  b. **$HOME/.config/gdb/gdbinit**
  c. **$HOME/.gdbinit**
- Support pour **~/.config/gdb/gdbearlyinit** ou **~/.gdbearlyinit**.
- **-eix** et **-eiex** options du fichier d'initialisation précoce.

Interface utilisateur du terminal (TUI) :

- Prise en charge des actions de la souris dans les fenêtres de l'interface utilisateur du terminal (TUI).

- Les combinaisons de touches qui n'agissent pas sur la fenêtre ciblée sont désormais transmises à GDB.

Nouvelles commandes :

- **show print memory-tag-violations**
- **set print memory-tag-violations**
- **memory-tag show-logical-tag**
- **memory-tag with-logical-tag**
- **memory-tag show-allocation-tag**
- **memory-tag check**
- **show startup-quietly** et **set startup-quietly**: Une façon de spécifier **-q** ou **-quiet** dans les scripts GDB. Valable uniquement dans les premiers fichiers d'initialisation.
- **show print type hex** et **set print type hex**: indique à GDB d'imprimer les tailles ou les décalages des membres de la structure en hexadécimal plutôt qu'en décimal.
- **show python ignore-environment** et **set python ignore-environment**: Si cette option est activée, l'interpréteur Python de GDB ignore les variables d'environnement Python, un peu comme si vous passiez **-E** à l'exécutable Python. Valable uniquement dans les premiers fichiers d'initialisation.
- **show python dont-write-bytecode** et **set python dont-write-bytecode**: Si **off**, ces commandes empêchent l'interpréteur Python de GDB d'écrire des objets compilés en bytecode pour les modules importés, de la même manière qu'en passant **-B** à l'exécutable Python. Valable uniquement dans les premiers fichiers d'initialisation.

Commandes modifiées :

- **break *LOCATION* if *CONDITION***: Si *CONDITION* est invalide, GDB refuse de placer un point d'arrêt. L'option **-force-condition** permet d'annuler ce refus.
- ***CONDITION* -force N COND**: Identique à la commande précédente.
- **inferior [ID]**: Lorsque l'ID est omis, cette commande imprime des informations sur l'inférieur actuel. Dans le cas contraire, les informations restent inchangées.
- **ptype[*/FLAGS*] TYPE | EXPRESSION**: Utilisez l'option **/x** pour utiliser la notation hexadécimale lors de l'impression des tailles et des décalages des membres de la structure. Utilisez l'option **/d** pour faire la même chose mais en utilisant la notation décimale.
- **info sources**: La sortie a été restructurée.

API Python :

- Les objets inférieurs contiennent un attribut **connection_num** en lecture seule.
- Nouvelle méthode **gdb.Frame.level()**.
- Nouvelle méthode **gdb.PendingFrame.level()**.
- **gdb.BreakpoiontEvent** émis au lieu de **gdb.Stop**.

(BZ#2077494)

GDB supporte les instructions Power 10 PLT

GDB prend désormais en charge les instructions PLT Power 10. Grâce à cette mise à jour, les utilisateurs peuvent accéder aux fonctions de la bibliothèque partagée et inspecter les traces de la pile à l'aide de la version 10.2-10 de GDB et des versions ultérieures.

(BZ#1870017)

Le site dyninst est passé à la version 12.1

Le paquetage **dyninst** a été rebasé à la version 12.1. Les corrections de bogues et améliorations notables sont les suivantes :

- Prise en charge initiale de **glibc-2.35** plusieurs espaces de noms
- Corrections de la simultanéité pour l'analyse parallèle de DWARF
- Meilleure prise en charge des binaires GPU **CUDA** et **CDNA2**
- Meilleure prise en charge de l'accès aux registres des systèmes IBM POWER (little endian)
- Meilleure prise en charge des binaires PIE
- Correction de l'analyse des blocs catch
- Correction de l'accès aux registres à virgule flottante 64-bit Arm (**aarch64**)

(BZ#2057675)

Un nouveau jeu de fichiers /etc/profile.d/debuginfod.*

Ajout d'un nouveau jeu de fichiers pour l'activation des services de débogage organisationnel. Pour obtenir une activation du client **debuginfod** à l'échelle du système, vous devez ajouter l'URL au fichier **/etc/debuginfod/FOO.urls**.

(BZ#2088774)

Rust Toolset repassé à la version 1.62.1

Rust Toolset a été mis à jour à la version 1.62.1. Les changements notables sont les suivants :

- La déstructuration de l'affectation permet aux modèles d'affecter des variables existantes dans la partie gauche d'une affectation. Par exemple, une affectation de n-uplets peut échanger des variables : **(a, b) = (b, a);**
- L'assemblage en ligne est désormais pris en charge sur x86 64 bits et ARM 64 bits à l'aide de la macro **core::arch::asm!**. Pour plus de détails, voir le chapitre "Assemblage en ligne" de la référence, **/usr/share/doc/rust/html/reference/inline-assembly.html** (en ligne à https://doc.rust-lang.org/reference/inline-assembly.html).
- Les Enums peuvent désormais dériver le trait **Default** avec une variante **#[default]** explicitement annotée.
- **Mutex**, **CondVar**, et **RwLock** utilisent désormais une implémentation personnalisée basée sur **futex** plutôt que pthreads, avec de nouvelles optimisations rendues possibles par les garanties du langage Rust.

- Rust prend désormais en charge les codes de sortie personnalisés de **main**, y compris les types définis par l'utilisateur qui mettent en œuvre le trait **Termination** nouvellement stabilisé.
- Cargo permet de mieux contrôler les caractéristiques des dépendances. Le préfixe **dep:** peut faire référence à une dépendance optionnelle sans l'exposer en tant que fonctionnalité, et un **?** n'active une fonctionnalité de dépendance que si cette dépendance est activée ailleurs, comme **package-name?/feature-name**.
- Cargo dispose d'une nouvelle sous-commande **cargo add** pour ajouter des dépendances à **Cargo.toml**.
- Pour plus de détails, veuillez consulter la série d'annonces de versions en amont :
  - Annonce de Rust 1.59.0
  - Annonce de Rust 1.60.0
  - Annonce de Rust 1.61.0
  - Annonce de Rust 1.62.0
  - Annonce de Rust 1.62.1

(BZ#2075337)

Le jeu d'outils LLVM passe à la version 14.0.6

LLVM Toolset a été rebasé à la version 14.0.6. Les changements notables sont les suivants :

- Sur 64-bit x86, la prise en charge des instructions **AVX512-FP16** a été ajoutée.
- La prise en charge des architectures Armv9-A, Armv9.1-A et Armv9.2-A a été ajoutée.
- Sur PowerPC, ajout du type **__ibm128** pour représenter le format IBM double-double, également disponible sous la forme **__attribute__((mode(IF)))**.

clang changements :

- **if consteval** pour **C 2b** est désormais implémentée.
- Sur 64-bit x86, la prise en charge des instructions **AVX512-FP16** a été ajoutée.
- Fin de la prise en charge d'OpenCL C 3.0 et **C** pour OpenCL 2021 à l'état expérimental.
- La sortie du préprocesseur **-E -P** omet désormais toujours les lignes vides, ce qui correspond au comportement de GCC. Auparavant, jusqu'à 8 lignes blanches consécutives pouvaient apparaître dans la sortie.
- Prise en charge de **-Wdeclaration-after-statement** avec **C99** et les normes ultérieures, et pas seulement C89, ce qui correspond au comportement de GCC. Un cas d'utilisation notable est la prise en charge des guides de style qui interdisent de mélanger les déclarations et le code, mais qui veulent passer à des normes C plus récentes.

Pour plus d'informations, voir les notes de mise à jour en amont de LLVM Toolset et Clang.

(BZ#2061041)

Go Toolset passe à la version 1.18.2

Go Toolset est passé à la version 1.18.2.

Les changements les plus notables sont les suivants :

- L'introduction des génériques tout en maintenant la compatibilité avec les versions antérieures de Go.
- Une nouvelle bibliothèque de fuzzing.
- Nouveaux paquets **debug**/**buildinfo** et **net**/**netip**.
- L'outil **go get** ne construit ni n'installe plus de paquets. Il ne gère plus que les dépendances dans **go.mod**.
- Si le fichier **go.mod** du module principal spécifie **go 1.17** ou une version supérieure, la commande **go mod download** utilisée sans arguments supplémentaires ne télécharge que le code source des modules explicitement requis dans le fichier **go.mod** du module principal. Pour télécharger également le code source des dépendances transitives, utilisez la commande **go mod download all**.
- La sous-commande **go mod vendor** propose désormais une option **-o** pour définir le répertoire de sortie.
- La commande **go mod tidy** conserve désormais des sommes de contrôle supplémentaires dans le fichier **go.sum** pour les modules dont le code source est nécessaire pour vérifier qu'un seul module dans la liste de construction fournit chaque paquet importé. Ce changement n'est pas conditionné par la version de Go dans le fichier **go.mod** du module principal.

(BZ#2075169)

Un nouveau flux de modules : maven:3.8

RHEL 9.1 introduit **Maven 3.8** en tant que nouveau flux de modules.

Pour installer le flux du module **maven:3.8**, utilisez :

```
# dnf module install maven:3.8
```

(BZ#2083112)

la version 7.0 de .NET est disponible

Red Hat Enterprise Linux 9.1 est distribué avec **.NET** version 7.0. Les améliorations notables sont les suivantes :

- Support pour IBM Power (**ppc64le**)

Pour plus d'informations, voir les notes de mise à jour pour les paquets RPM .NET 7.0 et les notes de mise à jour pour les conteneurs .NET 7.0.

(BZ#2112027)

4.15. GESTION DE L'IDENTITÉ

SSSD prend désormais en charge la mise en cache de la mémoire pour les demandes de SID

Avec cette amélioration, SSSD prend désormais en charge la mise en cache de la mémoire pour les

requêtes SID, c'est-à-dire les recherches de GID et d'UID par SID et vice versa. La mise en cache de la mémoire permet d'améliorer les performances, par exemple, lors de la copie de grandes quantités de fichiers vers ou depuis un serveur Samba.

(JIRA:RHELPLAN-123369)

Les modules Ansible ipaservicedelegationtarget et ipaservicedelegationrule sont désormais disponibles

Vous pouvez désormais utiliser les modules **ipaservicedelegationtarget** et **ipaservicedelegationrule ansible-freeipa** pour, par exemple, configurer un client de console web afin de permettre à un utilisateur de gestion d'identité (IdM) qui s'est authentifié avec une carte à puce d'effectuer les opérations suivantes :

- Utilisez **sudo** sur l'hôte RHEL sur lequel le service de console Web est exécuté sans qu'il vous soit demandé de vous authentifier à nouveau.
- Accédez à un hôte distant à l'aide de **SSH** et accédez aux services de l'hôte sans devoir vous authentifier à nouveau.

Les modules **ipaservicedelegationtarget** et **ipaservicedelegationrule** utilisent la fonction Kerberos **S4U2proxy**, également connue sous le nom de délégation restreinte. IdM utilise traditionnellement cette fonctionnalité pour permettre au serveur web d'obtenir un ticket de service LDAP au nom de l'utilisateur. Le système de confiance IdM-AD utilise cette fonctionnalité pour obtenir un principal cifs.

(JIRA:RHELPLAN-117109)

Support SSSD pour PKINIT anonyme pour FAST

Grâce à cette amélioration, SSSD prend désormais en charge la PKINIT anonyme pour l'authentification flexible via un tunnel sécurisé (FAST), également appelée blindage Kerberos dans Active Directory. Jusqu'à présent, pour utiliser FAST, un keytab Kerberos était nécessaire pour demander les informations d'identification requises. Vous pouvez désormais utiliser PKINIT anonyme pour créer ce cache d'informations d'identification afin d'établir la session FAST.

Pour activer le PKINIT anonyme, procédez comme suit :

1. Définissez **krb5_fast_use_anonymous_pkinit** comme **true** dans la section **[domain]** du fichier **sssd.conf**.
2. Redémarrer SSSD.
3. Dans un environnement IdM, vous pouvez vérifier que le PKINIT anonyme a été utilisé pour établir la session FAST en vous connectant en tant qu'utilisateur IdM. Un fichier cache contenant le ticket FAST est créé et le site **Default principal: WELLKNOWN/ANONYMOUS@WELLKNOWN:ANONYMOUS** indique que le PKINIT anonyme a été utilisé :

```
klist /var/lib/sss/db/fast_ccache_IPA.VM
Ticket cache: FILE:/var/lib/sss/db/fast_ccache_IPA.VM
Default principal: WELLKNOWN/ANONYMOUS@WELLKNOWN:ANONYMOUS
Valid starting Expires Service principal
03/10/2022 10:33:45 03/10/2022 10:43:45 krbtgt/IPA.VM@IPA.VM
```

(JIRA:RHELPLAN-123368)

L'IdM prend désormais en charge les numéros de série aléatoires

Avec cette mise à jour, Identity Management (IdM) inclut désormais **dogtagpki 11.2.0**, qui vous permet d'utiliser Random Serial Numbers version 3 (RSNv3). Vous pouvez activer RSNv3 en utilisant l'option **--random-serial-numbers** lors de l'exécution de **ipa-server-install** ou **ipa-ca-install**. Lorsque RSNv3 est activé, IdM génère des numéros de série entièrement aléatoires pour les certificats et les demandes dans PKI sans gestion de plage. En utilisant RSNv3, vous pouvez éviter la gestion des plages dans les grandes installations IdM et prévenir les collisions communes lors de la réinstallation d'IdM.

IMPORTANT

RSNv3 n'est pris en charge que pour les nouvelles installations IdM. Si elle est activée, il est nécessaire d'utiliser RSNv3 pour tous les services PKI.

(BZ#747959)

IdM permet désormais de limiter le nombre de liaisons LDAP autorisées après l'expiration du mot de passe d'un utilisateur

Cette amélioration permet de définir le nombre de liaisons LDAP autorisées lorsque le mot de passe d'un utilisateur Identity Management (IdM) a expiré :

-1
: L'IdM accorde à l'utilisateur un nombre illimité de liaisons LDAP avant que l'utilisateur ne doive réinitialiser son mot de passe. Il s'agit de la valeur par défaut, qui correspond au comportement précédent.

0
: Cette valeur désactive toutes les liaisons LDAP lorsque le mot de passe a expiré. En effet, les utilisateurs doivent réinitialiser leur mot de passe immédiatement.

1-MAXINT
: La valeur saisie autorise exactement ce nombre de liaisons après l'expiration.

La valeur peut être définie dans la stratégie globale de mot de passe et dans les stratégies de groupe.

Notez que le décompte est stocké par serveur.

Pour qu'un utilisateur puisse réinitialiser son propre mot de passe, il doit se lier avec son mot de passe actuel, qui a expiré. Si l'utilisateur a épuisé toutes les liaisons après expiration, le mot de passe doit être réinitialisé administrativement.

(BZ#2091988)

Nouveaux rôles ipasmartcard_server et ipasmartcard_client

Avec cette mise à jour, le paquet **ansible-freeipa** fournit des rôles Ansible pour configurer les serveurs et les clients de gestion des identités (IdM) pour l'authentification par carte à puce. Les rôles **ipasmartcard_server** et **ipasmartcard_client** remplacent les scripts **ipa-advise** pour automatiser et simplifier l'intégration. Le même inventaire et le même schéma de dénomination sont utilisés que dans les autres rôles **ansible-freeipa**.

(BZ#2076567)

IdM prend désormais en charge la configuration d'AD Trust avec Windows Server 2022

Grâce à cette amélioration, vous pouvez établir une confiance inter-forêts entre les domaines Identity Management (IdM) et les forêts Active Directory qui utilisent des contrôleurs de domaine fonctionnant sous Windows Server 2022.

(BZ#2122716)

Les messages de débogage de ipa-dnskeysyncd et ipa-ods-exporter ne sont plus enregistrés dans /var/log/messages par défaut

Auparavant, **ipa-dnskeysyncd**, le service responsable de la synchronisation entre LDAP et OpenDNSSEC, et **ipa-ods-exporter**, le service d'exportation OpenDNSSEC de Identity Management (IdM), enregistraient par défaut tous les messages de débogage sur **/var/log/messages**. Par conséquent, les fichiers journaux augmentaient considérablement. Avec cette amélioration, vous pouvez configurer le niveau de journalisation en définissant **debug=True** dans le fichier **/etc/ipa/dns.conf**. Pour plus d'informations, consultez **default.conf(5)**, la page de manuel du fichier de configuration IdM.

(BZ#2083218)

samba repassé à la version 4.16.1

Les paquets **samba** ont été mis à jour vers la version amont 4.16.1, qui apporte des corrections de bogues et des améliorations par rapport à la version précédente :

- Par défaut, le processus **smbd** démarre automatiquement le nouveau processus **samba-dcerpcd** à la demande pour servir l'environnement informatique distribué / les appels de procédure à distance (DCERPC). Notez que Samba 4.16 et les versions ultérieures exigent toujours que **samba-dcerpcd** utilise DCERPC. Si vous désactivez le paramètre **rpc start on demand helpers** dans la section **[global]** du fichier **/etc/samba/smb.conf**, vous devez créer une unité de service **systemd** pour exécuter **samba-dcerpcd** en mode autonome.
- Le rôle de la base de données triviale en grappe (CTDB) **recovery master** a été renommé en **leader**. En conséquence, les sous-commandes suivantes de **ctdb** ont été renommées :
  - **recmaster** à **leader**
  - **setrecmasterrole** à **setleaderrole**
- La configuration CTDB **recovery lock** a été renommée **cluster lock**.
- La CTDB utilise désormais les diffusions du chef de file et un délai associé pour déterminer si une élection est nécessaire.

Notez que le protocole server message block version 1 (SMB1) est obsolète depuis Samba 4.11 et sera supprimé dans une prochaine version.

Sauvegardez les fichiers de base de données avant de démarrer Samba. Lorsque les services **smbd**, **nmbd**, ou **winbind** démarrent, Samba met automatiquement à jour ses fichiers de base de données **tdb**. Notez que Red Hat ne prend pas en charge la rétrogradation des fichiers de base de données **tdb**.

Après avoir mis à jour Samba, vérifiez le fichier **/etc/samba/smb.conf** à l'aide de l'utilitaire **testparm**.

Pour plus d'informations sur les changements notables, lisez les notes de version en amont avant de procéder à la mise à jour.

(BZ#2077487)

SSSD prend désormais en charge l'intégration directe avec Windows Server 2022

Grâce à cette amélioration, vous pouvez utiliser SSSD pour intégrer directement votre système RHEL aux forêts Active Directory qui utilisent des contrôleurs de domaine exécutant Windows Server 2022.

(BZ#2070793)

Amélioration des performances multithread de SSSD

Auparavant, SSSD sérialisait les requêtes parallèles provenant d'applications multithread, telles que Red Hat Directory Server et Identity Management. Cette mise à jour corrige toutes les bibliothèques client SSSD, telles que **nss** et **pam**, afin qu'elles ne sérialisent pas les requêtes, permettant ainsi aux requêtes provenant de plusieurs threads d'être exécutées en parallèle pour de meilleures performances. Pour activer le comportement précédent de la sérialisation, définissez la variable d'environnement **SSS_LOCKFREE** sur **NO**.

(BZ#1978119)

Directory Server permet désormais d'annuler la tâche du plug-in Auto Membership.

Auparavant, la tâche du plug-in Auto Membership pouvait générer une utilisation élevée du CPU sur le serveur si Directory Server avait une configuration complexe (grands groupes, règles complexes et interaction avec d'autres plugins). Avec cette amélioration, vous pouvez annuler la tâche du plug-in Auto Membership. Par conséquent, les problèmes de performance ne se produisent plus.

(BZ#2052527)

Directory Server prend désormais en charge les opérations de suppression récursive lors de l'utilisation de ldapdelete

Avec cette amélioration, Directory Server prend désormais en charge le contrôle OpenLDAP **Tree Delete Control [1.2.840.113556.1.4.805]**. Par conséquent, vous pouvez utiliser l'utilitaire **ldapdelete** pour supprimer de manière récursive les sous-entrées d'une entrée parentale.

(BZ#2057063)

Vous pouvez désormais définir des options de réplication de base lors de l'installation de Directory Server

Avec cette amélioration, vous pouvez configurer les options de réplication de base comme les identifiants d'authentification et le découpage du changelog pendant l'installation d'une instance à l'aide d'un fichier **.inf**.

(BZ#2057066)

Directory Server prend désormais en charge la création d'instances par un utilisateur non root

Auparavant, les utilisateurs non root ne pouvaient pas créer d'instances de Directory Server. Avec cette amélioration, un utilisateur non root peut utiliser la sous-commande **dscreate ds-root** pour configurer un environnement dans lequel les commandes **dscreate**,**dsctl**,**dsconf** sont utilisées comme d'habitude pour créer et administrer des instances de Directory Server.

(BZ#1872451)

pki les paquets ont été renommés en idm-pki

Les paquetages **pki** suivants sont maintenant renommés en **idm-pki** afin de mieux distinguer les paquetages IDM de ceux du Système de certification Red Hat :

- **idm-pki-tools**
- **idm-pki-acme**
- **idm-pki-base**

- **idm-pki-java**
- **idm-pki-ca**
- **idm-pki-kra**
- **idm-pki-server**
- **python3-idm-pki**

(BZ#2139877)

4.16. INFRASTRUCTURES GRAPHIQUES

Wayland est maintenant activé avec les GPU Matrox

La session de bureau active désormais le back-end Wayland avec les GPU Matrox.

Dans les versions précédentes, Wayland était désactivé avec les GPU Matrox en raison des performances et d'autres limitations. Ces problèmes ont été corrigés.

Vous pouvez toujours basculer la session de bureau de Wayland à Xorg. Pour plus d'informations, voir Vue d'ensemble des environnements GNOME.

(BZ#2097308)

les GPU Intel Core de 12e génération sont désormais pris en charge

Cette version ajoute la prise en charge de plusieurs GPU intégrés pour les CPU Intel Core de la 12ème génération. Cela inclut les GPU intégrés Intel UHD Graphics et Intel Xe que l'on trouve avec les modèles de CPU suivants :

- Intel Core i3 12100T à Intel Core i9 12900KS
- Intel Pentium Gold G7400 et G7400T
- Intel Celeron G6900 et G6900T
- Intel Core i5-12450HX à Intel Core i9-12950HX
- Intel Core i3-1220P à Intel Core i7-1280P

(JIRA:RHELPLAN-135601)

Prise en charge des nouveaux GPU AMD

Cette version ajoute la prise en charge de plusieurs GPU de la série AMD Radeon RX 6000 et des graphiques intégrés des CPU de la série AMD Ryzen 6000.

Les modèles de GPU de la série AMD Radeon RX 6000 suivants sont désormais pris en charge :

- AMD Radeon RX 6400
- AMD Radeon RX 6500 XT
- AMD Radeon RX 6300M
- AMD Radeon RX 6500M

La série AMD Ryzen 6000 comprend des GPU intégrés dans les modèles de CPU suivants :

- AMD Ryzen 5 6600U
- AMD Ryzen 5 6600H
- AMD Ryzen 5 6600HS
- AMD Ryzen 7 6800U
- AMD Ryzen 7 6800H
- AMD Ryzen 7 6800HS
- AMD Ryzen 9 6900HS
- AMD Ryzen 9 6900HX
- AMD Ryzen 9 6980HS
- AMD Ryzen 9 6980HX

(JIRA:RHELPLAN-135602)

4.17. LA CONSOLE WEB

La page de progression des mises à jour dans la console web propose désormais une option de redémarrage automatique

La page de progression des mises à jour comporte désormais un interrupteur **Reboot after completion**. Celui-ci redémarre automatiquement le système après l'installation des mises à jour.

(BZ#2056786)

4.18. RÔLES DU SYSTÈME RED HAT ENTERPRISE LINUX

Le rôle de système **network** RHEL prend en charge la configuration du réseau à l'aide de l'API **nmstate**

Avec cette mise à jour, le rôle de système **network** RHEL prend en charge la configuration du réseau via l'API **nmstate**. Les utilisateurs peuvent désormais appliquer directement la configuration de l'état du réseau requis à une interface réseau au lieu de créer des profils de connexion. Cette fonctionnalité permet également la configuration partielle d'un réseau. Il en résulte les avantages suivants :

- réduction de la complexité de la configuration du réseau
- manière fiable d'appliquer les modifications de l'état du réseau
- pas besoin de suivre l'ensemble de la configuration du réseau

(BZ#2072385)

Les utilisateurs peuvent créer des connexions avec la capacité IPoIB en utilisant le rôle de système RHEL **network**

Le type de connexion **infiniband** du rôle de système RHEL **network** prend désormais en charge le protocole Internet sur Infiniband (IPoIB). Pour activer cette fonctionnalité, définissez une valeur à

l'option **p_key** de la variable **infiniband**. Notez que si vous spécifiez **p_key**, l'option **interface_name** de la variable **network_connections** ne doit pas être définie. L'implémentation précédente du rôle de système RHEL **network** ne validait pas correctement la valeur **p_key** et l'option **interface_name** pour le type de connexion **infiniband**. Par conséquent, la fonctionnalité IPoIB n'a jamais fonctionné auparavant. Pour plus d'informations, voir le fichier README dans le répertoire **/usr/share/doc/rhel-system-roles/network/**.

(BZ#2086965)

Le rôle de système RHEL de cluster HA prend désormais en charge la clôture SBD et la configuration des paramètres Corosync

Le rôle de système de cluster HA prend désormais en charge les fonctionnalités suivantes :

Clôture SBD

La clôture est un élément essentiel de la configuration d'un cluster HA. Les SBD permettent aux nœuds de s'arrêter d'eux-mêmes de manière fiable lorsqu'une clôture est nécessaire. La clôture par SBD peut être particulièrement utile dans les environnements où les mécanismes de clôture traditionnels ne sont pas possibles. Il est désormais possible de configurer la clôture SBD avec le rôle de système de cluster HA.

Paramètres Corosync

Le rôle HA Cluster System prend désormais en charge la configuration des paramètres Corosync, tels que le transport, la compression, le cryptage, les liens, le totem et le quorum. Ces paramètres sont nécessaires pour adapter la configuration du cluster aux besoins et à l'environnement des clients lorsque les paramètres par défaut ne conviennent pas.

(BZ#2065337, BZ#2070452, BZ#2079626, BZ#2098212, BZ#2120709, BZ#2120712)

Le rôle network RHEL configure désormais les paramètres réseau pour les règles de routage

Auparavant, vous pouviez acheminer le paquet en fonction du champ de l'adresse de destination dans le paquet, mais vous ne pouviez pas définir le routage de la source et d'autres règles de routage de la politique. Avec cette amélioration, le rôle de **network** RHEL prend en charge les règles de routage afin que les utilisateurs puissent contrôler la transmission des paquets ou la sélection des itinéraires.

(BZ#2079622)

La nouvelle configuration de previous:replaced permet au rôle du système firewall de réinitialiser les paramètres du pare-feu à leur valeur par défaut

Les administrateurs système qui gèrent différents ensembles de machines, où chaque machine a des paramètres de pare-feu préexistants différents, peuvent désormais utiliser la configuration **previous: replaced** dans le rôle **firewall** pour s'assurer que toutes les machines ont les mêmes paramètres de configuration de pare-feu. La configuration **previous: replaced** peut effacer tous les paramètres de pare-feu existants et les remplacer par des paramètres cohérents.

(BZ#2043010)

Nouvelle option dans le rôle de système postfix RHEL pour écraser la configuration précédente

Si vous gérez un groupe de systèmes dont les configurations **postfix** sont incohérentes, vous voudrez peut-être rendre la configuration cohérente sur chacun d'entre eux. Avec cette amélioration, vous pouvez spécifier l'option **previous: replaced** dans le dictionnaire **postfix_conf** pour supprimer toute configuration existante et appliquer la configuration souhaitée sur une installation **postfix** propre. Ainsi, vous pouvez effacer toute configuration existante de **postfix** et garantir la cohérence sur tous les systèmes gérés.

(BZ#2065383)

Amélioration de microsoft.sql.server RHEL system role

Les nouvelles variables suivantes sont désormais disponibles pour le rôle de système **microsoft.sql.server** RHEL :

- Variables avec le préfixe **mssql_ha_** pour contrôler la configuration d'un cluster de haute disponibilité.
- La variable **mssql_tls_remote_src** pour rechercher les valeurs **mssql_tls_cert** et **mssql_tls_private_key** sur les nœuds gérés. Si vous conservez le paramètre par défaut **false**, le rôle recherche ces fichiers sur le nœud de contrôle.
- La variable **mssql_manage_firewall** permet de gérer automatiquement les ports du pare-feu. Si cette variable vaut **false**, vous devez activer manuellement les ports du pare-feu.
- Les variables **mssql_pre_input_sql_file** et **mssql_post_input_sql_file** permettent de contrôler si vous voulez exécuter les scripts SQL avant ou après l'exécution du rôle. Ces nouvelles variables remplacent l'ancienne variable **mssql_input_sql_file**, qui ne permettait pas d'influencer le moment de l'exécution des scripts SQL.

(BZ#2066337)

Le rôle de système RHEL logging prend en charge les options startmsg.regex et endmsg.regex dans les entrées de fichiers

Grâce à cette amélioration, vous pouvez désormais filtrer les messages de journalisation provenant de fichiers en utilisant des expressions régulières. Les options **startmsg_regex** et **endmsg_regex** sont désormais incluses dans l'entrée des fichiers. L'option **startmsg_regex** représente l'expression régulière qui correspond au début d'un message, et l'option **endmsg_regex** représente l'expression régulière qui correspond à la dernière partie d'un message. Par conséquent, vous pouvez désormais filtrer les messages en fonction de propriétés telles que la date et l'heure, la priorité et la gravité.

(BZ#2112145)

Le rôle système sshd RHEL vérifie la directive include pour le répertoire de dépôt

Le rôle système **sshd** RHEL sur RHEL 9 ne gère qu'un fichier dans le répertoire de dépôt, mais ne vérifiait pas auparavant que le répertoire était inclus dans le fichier principal **sshd_config**. Avec cette mise à jour, le rôle vérifie que **sshd_config** contient la directive include pour le répertoire de dépôt. Par conséquent, le rôle applique de manière plus fiable la configuration fournie.

(BZ#2052081)

Le rôle de système sshd RHEL peut être géré par l'intermédiaire de /etc/ssh/sshd_config

Le rôle de système RHEL **sshd** appliqué à un nœud géré par RHEL 9 place la configuration du SSHD dans un répertoire de dépôt (**/etc/ssh/sshd_config.d/00-ansible_system_role.conf** par défaut). Auparavant, toute modification apportée au fichier **/etc/ssh/sshd_config** écrasait les valeurs par défaut de **00-ansible_system_role.conf**. Avec cette mise à jour, vous pouvez gérer le SSHD en utilisant **/etc/ssh/sshd_config** au lieu de **00-ansible_system_role.conf** tout en préservant les valeurs par défaut du système dans **00-ansible_system_role.conf**.

(BZ#2052086)

Le rôle metrics utilise systématiquement le commentaire "Ansible_managed" dans ses fichiers de configuration gérés

Avec cette mise à jour, le rôle **metrics** insère le commentaire "Ansible managed" dans les fichiers de configuration, en utilisant la variable standard Ansible **ansible_managed**. Le commentaire indique que les fichiers de configuration ne doivent pas être modifiés directement car le rôle **metrics** peut écraser le fichier. En conséquence, les fichiers de configuration contiennent une déclaration indiquant que les fichiers de configuration sont gérés par Ansible.

(BZ#2065392)

Le rôle de système storage RHEL prend désormais en charge la gestion des membres du pool

Le rôle de système RHEL **storage** peut désormais ajouter ou supprimer des disques de pools LVM existants sans supprimer le pool au préalable. Pour augmenter la capacité du pool, le rôle système **storage** RHEL peut ajouter de nouveaux disques au pool et libérer les disques actuellement alloués au pool pour une autre utilisation.

(BZ#2072742)

La prise en charge des volumes à provisionnement fin est désormais disponible dans le rôle de système RHEL storage

Le rôle système **storage** RHEL permet désormais de créer et de gérer des volumes logiques LVM (LV) finement provisionnés. Les LV à provisionnement fin sont alloués au fur et à mesure de leur écriture, ce qui offre une plus grande souplesse lors de la création de volumes, car le stockage physique fourni pour les LV à provisionnement fin peut être augmenté ultérieurement en fonction des besoins. Le provisionnement fin de LVM permet également de créer des instantanés plus efficaces, car les blocs de données communs à un LV fin et à n'importe lequel de ses instantanés sont partagés.

(BZ#2072745)

Une meilleure prise en charge des volumes mis en cache est disponible sur le site storage RHEL System Role

Le rôle de système RHEL **storage** permet désormais d'attacher un cache aux volumes logiques LVM existants. Le cache LVM peut être utilisé pour améliorer les performances des volumes logiques plus lents en stockant temporairement des sous-ensembles de données d'un LV sur un périphérique plus petit et plus rapide, par exemple un SSD. Cela améliore la prise en charge précédemment ajoutée pour la création de volumes mis en cache en permettant d'ajouter (attacher) un cache à un volume existant, qui n'était pas mis en cache auparavant.

(BZ#2072746)

Le rôle de système RHEL logging prend désormais en charge les options template, severity et facility

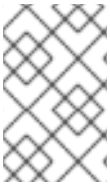
Le rôle de système RHEL **logging** comporte désormais de nouvelles options utiles **severity** et **facility** pour les entrées de fichiers, ainsi qu'une nouvelle option **template** pour les sorties de fichiers et de transferts. Utilisez l'option **template** pour spécifier le format d'heure traditionnel à l'aide du paramètre **traditional**, le format du protocole syslog 23 à l'aide du paramètre **syslog**, et le format de style moderne à l'aide du paramètre **modern**. Par conséquent, vous pouvez maintenant utiliser le rôle **logging** pour filtrer par gravité et facilité ainsi que pour spécifier le format de sortie par modèle.

(BZ#2075119)

Les rôles système RHEL sont désormais également disponibles dans les playbooks dont la collecte des faits est désactivée

La collecte de faits Ansible peut être désactivée dans votre environnement pour des raisons de

performances ou autres. Auparavant, il n'était pas possible d'utiliser les rôles système RHEL dans de telles configurations. Avec cette mise à jour, le système détecte le paramètre **ANSIBLE_GATHERING=explicit** dans votre configuration et le paramètre **gather_facts: false** dans vos playbooks, et utilise le module **setup:** pour collecter uniquement les faits requis par le rôle donné, s'ils ne sont pas disponibles dans le cache de faits.



NOTE

Si vous avez désactivé la collecte des faits Ansible pour des raisons de performance, vous pouvez activer la mise en cache des faits Ansible à la place, ce qui n'entraîne pas de baisse de performance en les récupérant à partir de la source.

(BZ#2078989)

Le rôle de stockage a maintenant moins de verbosité par défaut

La sortie des rôles de stockage est désormais moins verbeuse par défaut. Avec cette mise à jour, les utilisateurs peuvent augmenter la verbosité de la sortie du rôle de stockage pour ne produire qu'une sortie de débogage s'ils utilisent le niveau de verbosité Ansible 1 ou plus.

(BZ#2079627)

Le rôle de système RHEL **firewall** ne nécessite pas le paramètre **state** lors de la configuration de **masquerade** ou de **icmp_block_inversion**

Lors de la configuration des zones de pare-feu personnalisées, les variables **masquerade** et **icmp_block_inversion** sont des paramètres booléens. Une valeur de **true** implique **state: present** et une valeur de **false** implique **state: absent**. Par conséquent, le paramètre **state** n'est pas nécessaire lors de la configuration de **masquerade** ou **icmp_block_inversion**.

(BZ#2093423)

Vous pouvez désormais ajouter, mettre à jour ou supprimer des services à l'aide des états **absent** et **present** dans le rôle de système RHEL **firewall**

Grâce à cette amélioration, vous pouvez utiliser l'état **present** pour ajouter des ports, des modules, des protocoles, des services et des adresses de destination, ou utiliser l'état **absent** pour les supprimer. Notez que pour utiliser les états **absent** et **present** dans le rôle de système RHEL **firewall**, définissez l'option **permanent** sur **true**. Avec l'option **permanent** définie sur **true**, les paramètres d'état s'appliquent jusqu'à ce qu'ils soient modifiés et ne sont pas affectés par les rechargements de rôle.

(BZ#2100292)

Le rôle système **firewall** peut ajouter ou supprimer une interface à la zone en utilisant l'ID de périphérique PCI

En utilisant l'ID du périphérique PCI, le rôle système **firewall** peut désormais attribuer ou supprimer une interface réseau d'une zone. Auparavant, si seul l'ID du périphérique PCI était connu au lieu du nom de l'interface, les utilisateurs devaient d'abord identifier le nom de l'interface correspondante pour utiliser le rôle système **firewall**. Avec cette mise à jour, le rôle système **firewall** peut désormais utiliser l'ID du périphérique PCI pour gérer une interface réseau dans une zone.

(BZ#2100942)

Le rôle de système **firewall** RHEL peut fournir des faits Ansible

Grâce à cette amélioration, vous pouvez désormais rassembler les faits Ansible du rôle de système RHEL

firewall pour tous vos systèmes en incluant la variable **firewall:** dans le playbook sans aucun argument. Pour recueillir une version plus détaillée des données Ansible, utilisez l'argument **detailed: true**, par exemple :

```
vars:
  firewall:
    detailed: true
```

(BZ#2115154)

Ajout de la définition de seuser et selevel au rôle de système RHEL selinux

Il est parfois nécessaire de définir les paramètres **seuser** et **selevel** lors de la définition des mappages de systèmes de fichiers contextuels SELinux. Avec cette mise à jour, vous pouvez utiliser les arguments facultatifs **seuser** et **selevel** dans **selinux_fcontext** pour spécifier l'utilisateur et le niveau SELinux dans les mappages du système de fichiers du contexte SELinux.

(BZ#2115157)

Nouvelle variable cockpit System Role pour définir un port d'écoute personnalisé

Le rôle système **cockpit** introduit la variable **cockpit_port** qui vous permet de définir un port d'écoute personnalisé autre que le port par défaut 9090. Notez que si vous décidez de définir un port d'écoute personnalisé, vous devrez également ajuster votre politique SELinux pour permettre à la console web d'écouter sur ce port.

(BZ#2115152)

Le rôle metrics peut exporter les données de performance de postfix

Vous pouvez désormais utiliser la nouvelle variable booléenne **metrics_from_postfix** dans le rôle **metrics** pour l'enregistrement et l'analyse détaillée des performances. Avec cette amélioration, la définition de la variable active l'agent de mesure **pmdapostfix** sur le système, ce qui permet d'obtenir des statistiques sur **postfix**.

(BZ#2051737)

Le rôle postfix utilise systématiquement le commentaire "Ansible_managed" dans ses fichiers de configuration gérés

Le rôle **postfix** génère le fichier de configuration **/etc/postfix/main.cf**. Avec cette mise à jour, le rôle **postfix** insère le commentaire "Ansible managed" dans les fichiers de configuration, en utilisant la variable standard Ansible **ansible_managed**. Le commentaire indique que les fichiers de configuration ne doivent pas être modifiés directement car le rôle **postfix** peut écraser le fichier. En conséquence, les fichiers de configuration contiennent une déclaration indiquant que les fichiers de configuration sont gérés par Ansible.

(BZ#2065393)

Le rôle de système nbde-client RHEL prend en charge les adresses IP statiques

Dans les versions précédentes de RHEL, le redémarrage d'un système doté d'une adresse IP statique et configuré avec le rôle de système RHEL **nbde_client** modifiait l'adresse IP du système. Avec cette mise à jour, les systèmes avec des adresses IP statiques sont pris en charge par le rôle **nbde_client** et leurs adresses IP ne changent pas après un redémarrage.

Notez que par défaut, le rôle **nbde_client** utilise DHCP au démarrage et bascule sur l'IP statique configurée après le démarrage du système.

(BZ#2070462)

4.19. VIRTUALISATION

La console web RHEL propose désormais RHEL comme option pour le flux de travail de Download an OS VM

Avec cette amélioration, la console web RHEL prend désormais en charge l'installation de machines virtuelles (VM) RHEL à l'aide du flux de travail par défaut **Download an OS**. Vous pouvez ainsi télécharger et installer le système d'exploitation RHEL en tant que machine virtuelle directement dans la console web.

(JIRA:RHELPLAN-121982)

Amélioration de la conformité architecturale de KVM

Avec cette mise à jour, la conformité architecturale de l'hyperviseur KVM a été améliorée et rendue plus stricte. L'hyperviseur est ainsi mieux préparé à faire face aux changements futurs des systèmes d'exploitation basés sur Linux et d'autres systèmes.

(JIRA:RHELPLAN-117713)

ap-check est désormais disponible dans RHEL 9

L'outil **mdevctl** est désormais accompagné d'un nouvel utilitaire de support **ap-check**. Vous pouvez utiliser **mdevctl** pour configurer de manière persistante les adaptateurs cryptographiques et les domaines autorisés pour une utilisation pass-through dans les machines virtuelles, ainsi que les périphériques **matrix** et **vfio-ap**. Avec **mdevctl**, vous n'avez pas besoin de reconfigurer ces adaptateurs, domaines et périphériques après chaque IPL. En outre, **mdevctl** empêche le distributeur d'inventer d'autres façons de les reconfigurer.

Lors de l'invocation des commandes **mdevctl** pour les dispositifs **vfio-ap**, le nouvel utilitaire de support **ap-check** est invoqué dans le cadre de la commande **mdevctl** afin d'effectuer des contrôles de validité supplémentaires par rapport aux configurations des dispositifs **vfio-ap**.

En outre, l'outil **chzdev** permet désormais de gérer les paramètres de masque du processeur d'adjonction (AP) à l'échelle du système, qui déterminent les ressources AP disponibles pour les appareils **vfio-ap**. Lorsqu'il est utilisé, **chzdev** permet de conserver ces paramètres en générant une règle **udev** associée. En utilisant **lszdev**, vous pouvez désormais également interroger les paramètres de masque AP à l'échelle du système.

(BZ#1870699)

open-vm-tools repassé à la version 12.0.5

Les paquets **open-vm-tools** ont été mis à jour vers la version 12.0.5, qui introduit un certain nombre de corrections de bogues et de nouvelles fonctionnalités. En particulier, la prise en charge de l'outil Salt Minion a été ajoutée pour être gérée par des variables du système d'exploitation invité.

(BZ#2061193)

Les machines virtuelles sélectionnées sur IBM Z peuvent désormais démarrer avec des lignes de commande du noyau de plus de 896 octets

Auparavant, le démarrage d'une machine virtuelle (VM) sur un hôte IBM Z RHEL 9 échouait toujours si la ligne de commande du noyau de la VM dépassait 896 octets. Avec cette mise à jour, l'émulateur QEMU peut gérer les lignes de commande du noyau de plus de 896 octets. Par conséquent, vous pouvez désormais utiliser le démarrage direct du noyau de QEMU pour les machines virtuelles dont les lignes de

commande du noyau sont très longues, si le noyau de la machine virtuelle le prend en charge. Plus précisément, pour utiliser une ligne de commande de plus de 896 octets, la VM doit utiliser la version 5.16-rc1 du noyau Linux ou une version ultérieure.

(BZ#2044218)

La fonction Secure Execution d'IBM Z prend désormais en charge l'attestation à distance

La fonction Secure Execution sur l'architecture IBM Z prend désormais en charge l'attestation à distance. L'utilitaire **pvattest** peut créer une demande d'attestation à distance pour vérifier l'intégrité d'un invité pour lequel la fonction Secure Execution est activée.

En outre, il est désormais possible d'injecter des interruptions dans les invités avec Secure Execution grâce à l'utilisation de GISA.

(BZ#2001936, BZ#2044300)

Préallocation de la mémoire de la VM à l'aide de plusieurs threads

Vous pouvez désormais définir plusieurs threads de CPU pour l'allocation de la mémoire de la machine virtuelle (VM) dans la configuration XML du domaine, par exemple de la manière suivante :

```
<memoryBacking>
  <allocation threads='8'/>
</memoryBacking>
```

Cela permet de s'assurer que plus d'un thread est utilisé pour l'allocation des pages mémoire lors du démarrage d'une VM. Par conséquent, les machines virtuelles pour lesquelles plusieurs threads d'allocation ont été configurés démarrent beaucoup plus rapidement, en particulier si de grandes quantités de RAM ont été attribuées aux machines virtuelles et qu'elles sont soutenues par des pages de mémoire volumineuses.

(BZ#2064194)

Les invités RHEL 9 prennent désormais en charge SEV-SNP

Sur les machines virtuelles (VM) qui utilisent RHEL 9 comme système d'exploitation invité, vous pouvez désormais utiliser AMD Secure Encrypted Virtualization (SEV) avec la fonction Secure Nested Paging (SNP). Entre autres avantages, SNP améliore SEV en améliorant la protection de l'intégrité de la mémoire, ce qui permet de prévenir les attaques basées sur l'hyperviseur, telles que le rejeu de données ou le remappage de la mémoire. Notez que pour que SEV-SNP fonctionne sur une VM RHEL 9, l'hôte qui exécute la VM doit également prendre en charge SEV-SNP.

(BZ#2169738)

4.20. RHEL DANS LES ENVIRONNEMENTS EN NUAGE

Nouveau module SSH pour cloud-init

Avec cette mise à jour, un module SSH a été ajouté à l'utilitaire **cloud-init**, qui génère automatiquement des clés d'hôte lors de la création d'une instance.

Notez qu'avec ce changement, la configuration par défaut de **cloud-init** a été mise à jour. Par conséquent, si vous avez effectué une modification locale, assurez-vous que le fichier /etc/cloud/cloud.cfg contient la ligne "ssh_genkeytypes : ['rsa', 'ecdsa', 'ed25519']\N" ligne.

Sinon, **cloud-init** crée une image qui ne démarre pas le service **sshd**. Dans ce cas, procédez comme suit pour contourner le problème :

1. Assurez-vous que le fichier **/etc/cloud/cloud.cfg** contient la ligne suivante :

```
ssh_genkeytypes:  ['rsa', 'ecdsa', 'ed25519']
```

2. Vérifiez si les fichiers **/etc/ssh/ssh_host_*** existent dans l'instance.
3. Si les fichiers **/etc/ssh/ssh_host_*** n'existent pas, utilisez la commande suivante pour générer des clés d'hôte :

```
cloud-init single --name cc_ssh
```

4. Redémarrez le service sshd :

```
systemctl restart sshd
```

(BZ#2115791)

4.21. CONTENEURS

Les paquets Container Tools ont été mis à jour

Les paquets Container Tools qui contiennent les outils Podman, Buildah, Skopeo, crun et runc sont maintenant disponibles. Cette mise à jour fournit une liste de corrections de bogues et d'améliorations par rapport à la version précédente.

Les changements les plus notables sont les suivants :

- La commande **podman pod create** permet désormais de définir les limites de CPU et de mémoire. Vous pouvez définir une limite pour tous les conteneurs du pod, tandis que les conteneurs individuels du pod peuvent avoir leurs propres limites.
- La commande **podman pod clone** crée une copie d'un module existant.
- La commande **podman play kube** prend désormais en charge les paramètres du contexte de sécurité à l'aide des volumes **BlockDevice** et **CharDevice**.
- Les pods créés par **podman play kube** peuvent désormais être gérés par des fichiers unitaires systemd utilisant un **podman-kube@<service>.service** (par exemple **systemctl --user start podman-play-kube@$(systemd-escape my.yaml).service**).
- Les commandes **podman push** et **podman push manifest** prennent désormais en charge les signatures sigstore.
- Les réseaux Podman peuvent maintenant être isolés à l'aide de la commande **podman network --opt isolate**.

Podman a été mis à niveau vers la version 4.2. Pour plus d'informations sur les changements notables, voir les notes de version Podman v4.2.0 has been released ! et upstream.

(JIRA:RHELPLAN-118462)

GitLab Runner est maintenant disponible sur RHEL en utilisant Podman

A partir de GitLab Runner 15.1, vous pouvez utiliser Podman comme runtime de conteneur dans l'exécuteur Docker de GitLab Runner. Pour plus de détails, voir la note de version de GitLab .

(JIRA:RHELPLAN-101140)

Podman supporte désormais l'option --health-on-failure

Les commandes **podman run** et **podman create** prennent désormais en charge l'option **--health-on-failure** pour déterminer les actions à effectuer lorsque l'état d'un conteneur devient malsain.

L'option **--health-on-failure** prend en charge quatre actions :

- **none**: Ne rien faire, c'est l'action par défaut.
- **kill**: Tuer le conteneur.
- **restart**: Redémarrer le conteneur.
- **stop**: Arrêter le conteneur.

NOTE

Ne combinez pas l'action **restart** avec l'option **--restart**. Lorsque vous vous exécutez à l'intérieur d'une unité systemd, envisagez d'utiliser l'action **kill** ou **stop** à la place pour utiliser la politique de redémarrage de systemd.

(BZ#2097708)

La pile réseau Netavark est désormais disponible

La pile Netavark est un outil de configuration réseau pour les conteneurs. Dans RHEL 9, la pile Netavark est entièrement prise en charge et activée par défaut.

Cette pile de réseau possède les capacités suivantes :

- Configuration des réseaux de conteneurs à l'aide du fichier de configuration JSON
- Création, gestion et suppression d'interfaces réseau, y compris les interfaces de pont et MACVLAN
- Configuration des paramètres du pare-feu, tels que la traduction d'adresses réseau (NAT) et les règles de mappage des ports
- IPv4 et IPv6
- Capacité améliorée pour les conteneurs dans plusieurs réseaux
- Résolution DNS en conteneur à l'aide du projet aardvark-dns

NOTE

Vous devez utiliser la même version de la pile Netavark et le serveur DNS faisant autorité **aardvark-dns**.

(JIRA:RHELPLAN-132023)

Nouveau paquet : catatonit dans le référentiel CRB

Un nouveau paquetage **catatonit** est maintenant disponible dans le dépôt CodeReady Linux Builder (CRB). Le paquet **catatonit** est utilisé comme programme d'initialisation minimal pour les conteneurs et peut être inclus dans l'image du conteneur d'application. Notez que les paquets inclus dans le dépôt CodeReady Linux Builder ne sont pas pris en charge.

Notez que depuis RHEL 9.0, le paquet **podman-catonit** est disponible dans le dépôt AppStream. Le paquet **podman-catatonit** n'est utilisé que par l'outil Podman.

(BZ#2074193)

CHAPITRE 5. CHANGEMENTS IMPORTANTS DANS LES PARAMÈTRES EXTERNES DU NOYAU

Ce chapitre fournit aux administrateurs système un résumé des changements significatifs apportés au noyau distribué avec Red Hat Enterprise Linux 9.1. Ces changements peuvent inclure, par exemple, des entrées **proc** ajoutées ou mises à jour, des valeurs par défaut **sysctl** et **sysfs**, des paramètres de démarrage, des options de configuration du noyau ou tout autre changement de comportement notable.

Nouveaux paramètres du noyau

allow_mismatched_32bit_el0 = [ARM64]

Ce paramètre permet aux systèmes dont la prise en charge 32 bits au niveau EL0 n'est pas adaptée d'exécuter des applications 32 bits. L'ensemble des unités centrales prenant en charge le 32 bits au niveau EL0 est indiqué dans le fichier **/sys/devices/system/cpu/aarch32_el0**. Vous pouvez également limiter les opérations de débranchement à chaud.
Pour plus d'informations, voir **Documentation/arm64/asymmetric-32bit.rst**.

arm64.nomte = [ARM64]

Ce paramètre permet de désactiver inconditionnellement la prise en charge de l'extension de marquage de la mémoire (MTE).

i8042.probe_defer = [HW]

Ce paramètre permet d'autoriser le report du palpage en cas d'erreurs de palpage sur **i8042**.

idxd.tc_override = [HW]

Avec ce paramètre au format **<bool>**, vous pouvez autoriser l'annulation de la configuration par défaut des classes de trafic pour l'appareil.
La valeur par défaut est **false** (**0**).

kvm.eager_page_split = [KVM,X86]

Ce paramètre permet de contrôler si un KVM divise ou non de manière proactive toutes les pages volumineuses lors de la journalisation des données sales. Le fractionnement des pages réduit les interruptions de l'exécution du vCPU en éliminant les défauts de protection en écriture et les conflits de verrouillage de l'unité de gestion de la mémoire (MMU), qui sont autrement nécessaires pour fractionner les pages volumineuses de manière paresseuse.
Les charges de travail des machines virtuelles qui effectuent rarement des écritures ou qui n'écrivent que dans une petite région de la mémoire de la machine virtuelle peuvent bénéficier de la désactivation du fractionnement rapide des pages pour permettre aux pages volumineuses d'être encore utilisées pour les lectures.

Le comportement du fractionnement rapide des pages dépend de l'activation ou de la désactivation de l'option **KVM_DIRTY_LOG_INITIALLY_SET**.

- Si cette option est désactivée, toutes les pages volumineuses d'un site **memslot** sont scindées avec empressement lorsque la journalisation sale est activée sur ce site **memslot**.
- Si cette option est activée, le fractionnement des pages est effectué lors de l'appel système **KVM_CLEAR_DIRTY ioctl()** , et uniquement pour les pages en cours d'effacement. Actuellement, le fractionnement de pages ne permet de fractionner que les grandes pages mappées par le MMU de pagination bidimensionnelle (TDP).

  La valeur par défaut est **Y** (**on**).

kvm.nx_huge_pages_recovery_period_ms = [KVM]

Ce paramètre permet de contrôler le délai pendant lequel KVM ramène les pages de 4 KiB à des pages de grande taille.

- Si la valeur est différente de zéro (**N**), KVM zappe une partie des pages toutes les **N** millisecondes.
- Si la valeur est **0**, KVM choisit une période basée sur le ratio, de sorte qu'une page est supprimée après 1 heure en moyenne.
  La valeur par défaut est fixée à **0**.

l1d_flush = [X86,INTEL]

Ce paramètre permet de contrôler l'atténuation de la vulnérabilité du snooping basé sur L1D.
Certains processeurs sont vulnérables à un exploit contre les tampons internes du processeur qui peuvent, sous certaines conditions, transmettre des informations à un gadget de divulgation. Dans les processeurs vulnérables, les données transmises de manière spéculative peuvent être utilisées dans une attaque par canal latéral du cache, pour accéder à des données auxquelles l'attaquant n'a pas un accès direct.

L'option disponible est **on**, ce qui signifie **enable the interface for the mitigation**.

mmio_stale_data = [X86,INTEL]

Ce paramètre permet de contrôler l'atténuation des vulnérabilités liées aux données périmées de l'entrée/sortie en mémoire du processeur (MMIO).
Processor MMIO Stale Data est une classe de vulnérabilités qui peut exposer des données après une opération MMIO. Les données exposées peuvent provenir ou se terminer dans les mêmes tampons du processeur que ceux affectés par le serveur de métadonnées (MDS) et l'interruption asynchrone transactionnelle (TAA). Par conséquent, comme pour MDS et TAA, la solution consiste à effacer les tampons du processeur concernés.

Les options disponibles sont les suivantes :

- **full**les mesures d'atténuation sont activées sur les processeurs vulnérables
- **full,nosmt**les mesures d'atténuation sont activées et la technologie SMT est désactivée sur les processeurs vulnérables.
- **off**: désactivation inconditionnelle de l'atténuation
  Sur les machines affectées par MDS ou TAA, **mmio_stale_data=off** peut être empêché par une atténuation active de MDS ou TAA car ces vulnérabilités sont atténuées par le même mécanisme. Ainsi, pour désactiver cette atténuation, vous devez également spécifier **mds=off** et **tsx_async_abort=off**.

  Ne pas spécifier cette option équivaut à **mmio_stale_data=full**.

  Pour plus d'informations, voir **Documentation/admin-guide/hw-vuln/processor_mmio_stale_data.rst**.

random.trust_bootloader={on,off} = [KNL]

Ce paramètre permet d'activer ou de désactiver l'utilisation d'une graine transmise par le chargeur d'amorçage (si elle est disponible) pour ensemencer complètement le CRNG du noyau. Le comportement par défaut est contrôlé par l'option **CONFIG_RANDOM_TRUST_BOOTLOADER**.

rcupdate.rcu_task_collapse_lim = [KNL]

Ce paramètre permet de définir le nombre maximal de rappels présents au début d'une période de grâce qui permet aux saveurs de tâches de l'URC de revenir à l'utilisation d'une seule file d'attente de

rappels. Cette commutation ne se produit que lorsque l'option **rcupdate.rcu_task_enqueue_lim** est définie sur la valeur par défaut de **-1**.

rcupdate.rcu_task_contend_lim = [KNL]

Ce paramètre permet de définir le nombre minimum d'événements de rétention de verrou par jiffy requis pour que les saveurs de tâches RCU passent à la mise en file d'attente de rappel par unité centrale. Cette commutation ne se produit que lorsque l'option **rcupdate.rcu_task_enqueue_lim** est définie sur la valeur par défaut **-1**.

rcupdate.rcu_task_enqueue_lim = [KNL]

Ce paramètre permet de définir le nombre de files d'attente de rappel à utiliser pour la famille de saveurs RCU Tasks. Vous pouvez ajuster le nombre de files d'attente de rappel automatiquement et dynamiquement avec la valeur par défaut de **-1**.

Ce paramètre est destiné à être utilisé dans le cadre de tests.

retbleed = [X86]

Ce paramètre permet de contrôler l'atténuation de la vulnérabilité "Arbitrary Speculative Code Execution with Return Instructions" (RETBleed). Les options disponibles sont les suivantes :

- **off**: pas de mesures d'atténuation
- **auto**: sélection automatique d'une mesure d'atténuation
- **auto,nosmt**: sélectionne automatiquement une mesure d'atténuation, en désactivant SMT si nécessaire pour une atténuation complète (uniquement sur Zen1 et les versions antérieures sans STIBP).
- **ibpb**les données relatives à l'impact sur les performances : atténuent également les courtes fenêtres de spéculation sur les limites des blocs de base. Sûr, impact le plus élevé sur les performances.
- **unret**: force l'activation des thunks de retour non formés, efficace uniquement sur les systèmes basés sur AMD f15h-f17h.
- **unret,nosmt**: comme l'option **unret**, désactive SMT lorsque STIBP n'est pas disponible. L'option **auto** permet de choisir une méthode d'atténuation au moment de l'exécution en fonction de l'unité centrale.

  Ne pas spécifier cette option équivaut à **retbleed=auto**.

sev=option[,option...] = [X86-64]

Pour plus d'informations, voir **Documentation/x86/x86_64/boot-options.rst**.

Mise à jour des paramètres du noyau

acpi_sleep = [HW,ACPI]

Format : { s3_bios, s3_mode, s3_beep, s4_hwsig, s4_nohwsig, old_ordering, nonvs, sci_force_enable, nobl }

- Pour plus d'informations sur **s3_bios** et **s3_mode**, voir **Documentation/power/video.rst**.
- **s3_beep** est destiné au débogage ; il déclenche le bip du haut-parleur du PC dès que le point d'entrée en mode réel du noyau est appelé.
- **s4_hwsig** permet au noyau de vérifier la signature matérielle ACPI lors de la sortie d'hibernation, et de refuser de reprendre si elle a changé. Le comportement par défaut est

d'autoriser la reprise et d'avertir simplement lorsque la signature change, à moins que l'option **s4_hwsig** ne soit activée.

- **s4_nohwsig** empêche la signature matérielle ACPI d'être utilisée, ou même d'être prévenue, pendant la reprise. **old_ordering** fait en sorte que l'ordre ACPI 1.0 de la méthode de contrôle **_PTS**, en ce qui concerne le placement des périphériques dans des états de faible consommation, soit appliqué. L'ordre ACPI 2.0 de **_PTS** est utilisé par défaut.

- **nonvs** empêche le noyau de sauvegarder et de restaurer la mémoire ACPI NVS pendant la suspension, l'hibernation et la reprise.

- **sci_force_enable** fait en sorte que le noyau règle **SCI_EN** directement à la reprise de S1/S3. Bien que ce comportement soit contraire aux spécifications de l'ACPI, certains systèmes corrompus ne fonctionnent pas sans lui.

- **nobl** permet d'ignorer la liste interne des systèmes connus pour leur comportement incorrect en ce qui concerne la suspension et la reprise du système. Utilisez cette option à bon escient.
  Pour plus d'informations, voir **Documentation/power/video.rst**.

crashkernel=size[KMG],high = [KNL, X86-64, ARM64]

Ce paramètre permet d'allouer une région de mémoire physique en commençant par le haut, comme suit :

- Si le système dispose de plus de 4 Go de RAM, une région de mémoire physique peut dépasser 4 Go.

- Si le système dispose de moins de 4 Go de RAM, une région de mémoire physique sera allouée en dessous de 4 Go, si elle est disponible.
  Ce paramètre est ignoré si le paramètre **crashkernel=X** est spécifié.

crashkernel=size[KMG],low = [KNL, X86-64]

Lorsque vous passez **crashkernel=X,high**, le noyau peut allouer une région de mémoire physique supérieure à 4 Go. Cela provoque le deuxième plantage du noyau sur les systèmes qui nécessitent une certaine quantité de mémoire basse (par exemple, **swiotlb** nécessite au moins 64M 32K de mémoire basse) et suffisamment de mémoire basse supplémentaire pour s'assurer que les tampons DMA pour les périphériques 32 bits ne sont pas épuisés. Le noyau essaie d'allouer automatiquement au moins 256 M en dessous de 4 Go. Ce paramètre vous permet de spécifier la plage de mémoire inférieure à 4 Go pour le second noyau.

- **0:** désactive l'allocation basse. Elle sera ignorée si **crashkernel=X,high** n'est pas utilisé ou si la mémoire réservée est inférieure à 4 Go.

crashkernel=size[KMG],low = [KNL, ARM64]

Ce paramètre permet de spécifier une plage basse dans la zone DMA pour le noyau de crash dump. Il sera ignoré si **crashkernel=X,high** n'est pas utilisé ou si la mémoire réservée est située dans les zones DMA.

kvm.nx_huge_pages_recovery_ratio = [KVM]

Ce paramètre permet de contrôler le nombre de pages de 4 KiB qui sont périodiquement ramenées à des pages de grande taille :

- **0** désactive la récupération

- **N** KVM va zapper **1/Nth** des pages de 4 KiB à chaque période.

La valeur par défaut est **60**.

kvm-arm.mode = [KVM,ARM]

Ce paramètre permet de sélectionner l'un des modes de fonctionnement du KVM :

- **none**kVM : désactivation forcée de KVM.
- **nvhe**mode standard basé sur nVHE, sans prise en charge des invités protégés.
- **protected**mode basé sur **nVHE** avec prise en charge des invités dont l'état est gardé privé par rapport à l'hôte. Non valide si le noyau fonctionne au niveau EL2.
  La valeur par défaut est fixée à **VHE**/**nVHE** en fonction de la prise en charge matérielle.

mitigations = [X86,PPC,S390,ARM64]

Ce paramètre permet de contrôler les mesures d'atténuation optionnelles pour les vulnérabilités du processeur. Il s'agit d'un ensemble d'options curées, indépendantes de l'architecture, chacune d'entre elles étant une agrégation d'options existantes spécifiques à l'architecture :

- **off**désactivation de toutes les mesures d'atténuation facultatives pour le processeur. Cela améliore les performances du système, mais peut également exposer les utilisateurs à plusieurs vulnérabilités du processeur.
  - Équivalent à : **nopti [X86,PPC]**, **kpti=0 [ARM64]**, **nospectre_v1 [X86,PPC]**, **nobp=0 [S390]**, **nospectre_v2 [X86,PPC,S390,ARM64]**, **spectre_v2_user=off [X86]**, **spec_store_bypass_disable=off [X86,PPC]**, **ssbd=force-off [ARM64]**, **l1tf=off [X86]**, **mds=off [X86]**, **tsx_async_abort=off [X86]**, **kvm.nx_huge_pages=off [X86]**, **no_entry_flush [PPC]**, **no_uaccess_flush [PPC]**, **mmio_stale_data=off [X86]**.
  - Exceptions : Cela n'a aucun effet sur **kvm.nx_huge_pages** lorsque l'option **kvm.nx_huge_pages=force** est spécifiée.
- **auto** (par défaut) : atténue toutes les vulnérabilités du CPU, mais laisse SMT activé, même s'il est vulnérable.
  - Équivalent à : (comportement par défaut)
- **auto,nosmt**les vulnérabilités des CPU sont atténuées, en désactivant le SMT si nécessaire.
  - Équivalent à : **l1tf=flush,nosmt [X86]**, **mds=full,nosmt [X86]**, **tsx_async_abort=full,nosmt [X86]**, **mmio_stale_data=full,nosmt [X86]**

rcu_nocbs[=cpu-list] = [KNL]

L'argument facultatif est une liste de CPU.
Dans les noyaux construits avec **CONFIG_RCU_NOCB_CPU=y**, vous pouvez activer le mode CPU no-callback, qui empêche les callbacks de ces CPU d'être invoqués dans le contexte softirq. L'invocation des callbacks RCU de ces CPU sera à la place déchargée sur **rcuox**/**N kthreads** créé à cet effet, où **x** est **p** pour RCU-preempt, **s** pour RCU-sched, et **g** pour le **kthreads** qui gère les périodes de grâce ; et **N** est le numéro du CPU. Cela réduit la gigue du système d'exploitation sur les unités centrales déchargées, ce qui peut être utile pour les charges de travail HPC et en temps réel. Il peut également améliorer l'efficacité énergétique des multiprocesseurs asymétriques.

- Si un **cpulist** est fourni comme argument, la liste spécifiée de CPU est mise en mode no-callback à partir du démarrage.

- Si le signe **=** et les arguments **cpulist** sont omis, aucune unité centrale ne sera mise en mode "no-callback" au démarrage, mais vous pouvez basculer le mode au moment de l'exécution à l'aide de **cpusets**.

rcutree.kthread_prio = [KNL,BOOT]

Ce paramètre permet de définir la priorité **SCHED_FIFO** de l'unité RCU par CPU **kthreads** (**rcuc**/**N**). Cette valeur est également utilisée pour la priorité des threads boost de l'URC (**rcub**/**N**) et pour la période de grâce de l'URC **kthreads** (**rcu_bh**, **rcu_preempt**, et **rcu_sched**).

- Si **RCU_BOOST** est défini, les valeurs valables sont de 1 à 99 et la valeur par défaut est **1**, la priorité la moins favorisée.

- Si **RCU_BOOST** n'est pas défini, les valeurs valides sont 0-99 et la valeur par défaut est **0**, fonctionnement en temps réel.
  Lorsque **RCU_NOCB_CPU** est activé, vous devez ajuster la priorité du rappel **NOCB kthreads** .

rcuture.fwd_progress = [KNL]

Ce paramètre permet de spécifier le nombre de **kthreads** à utiliser pour les tests de progression de la période de grâce de l'UCR pour les types d'UCR supportant cette notion.
La valeur par défaut est **1 kthread** . Les valeurs inférieures à zéro ou supérieures au nombre de CPU entraînent l'utilisation du nombre de CPU.

spectre_v2 = [X86]

Ce paramètre permet de contrôler l'atténuation de la variante 2 de Spectre (spéculation indirecte sur les branches). Le fonctionnement par défaut protège le noyau des attaques de l'espace utilisateur.

- **on**: activation inconditionnelle, implique **spectre_v2_user=on**

- **off**: désactivation inconditionnelle, implique **spectre_v2_user=off**

- **auto**le noyau détecte si le modèle de votre processeur est vulnérable

- La sélection de **on** entraîne, et **auto** peut entraîner, le choix d'une méthode d'atténuation au moment de l'exécution en fonction du processeur, du microcode disponible, du réglage de l'option de configuration **CONFIG_RETPOLINE** et du compilateur avec lequel le noyau a été construit.

- La sélection de **on** permettra également de limiter les attaques de l'espace utilisateur contre les tâches de l'espace utilisateur.

- La sélection de **off** désactive les protections du noyau et de l'espace utilisateur.

- Des mesures d'atténuation spécifiques peuvent également être sélectionnées manuellement :

  - **retpoline**les branches indirectes sont remplacées par des branches indirectes

  - **retpoline,generic**: Retpolines

  - **retpoline,lfence**: LFENCE ; branche indirecte

  - **retpoline,amd**: alias de retpoline,lfence

  - **eibrs**: IBRS amélioré

- **eibrs,retpoline**les Retpolines de l'IBRS améliorées
- **eibrs,lfence**: amélioré IBRS LFENCE
- **ibrs**les entreprises de l'Union européenne : utiliser l'IBRS pour protéger le noyau
  Ne pas spécifier cette option équivaut à **spectre_v2=auto**.

Nouveaux paramètres sysctl

max_rcu_stall_to_panic

Lorsque vous définissez **panic_on_rcu_stall** à **1**, vous déterminez le nombre de fois que l'URC peut se bloquer avant que **panic()** ne soit appelé. Lorsque vous définissez **panic_on_rcu_stall** à **0**, cette valeur n'a aucun effet.

perf_user_access = [ARM64]

Ce paramètre permet de contrôler l'accès à l'espace utilisateur pour la lecture des compteurs d'événements de **perf**.

- Lorsqu'il est défini sur **1**, l'espace utilisateur peut lire directement les registres des compteurs du moniteur de performance.
- La valeur par défaut est **0**, ce qui signifie **access disabled**.
  Pour plus d'informations, voir **Documentation/arm64/perf.rst**.

gro_normal_batch

Avec ce paramètre, vous pouvez définir le nombre maximum de segments à mettre en lot à la sortie du GRO. Lorsqu'un paquet sort du GRO, que ce soit sous la forme d'une supertrame coalisée ou d'un paquet original que le GRO a décidé de ne pas coaliser, il est placé sur une liste par NAPI. Cette liste est ensuite transmise à la pile lorsque le nombre de segments atteint la limite de **gro_normal_batch**.

high_order_alloc_disable

Ce paramètre vous permet de choisir l'allocation de l'ordre 0. Par défaut, l'allocateur de fragments de pages tente d'utiliser des pages d'ordre élevé, c'est-à-dire d'ordre 3 sur les systèmes X86. Bien que le comportement par défaut donne de bons résultats, dans certaines situations, une contention dans l'allocation et la libération des pages se produit. Cela était particulièrement vrai avec les anciens noyaux (version 5.14 et plus) lorsque les pages d'ordre élevé n'étaient pas stockées dans des listes par unité centrale. Ce paramètre n'a plus qu'une importance historique.
La valeur par défaut est **0**.

page_lock_unfairness

En spécifiant la valeur de ce paramètre, vous pouvez déterminer le nombre de fois que le verrou de page peut être volé sous un serveur. Une fois que le verrou a été volé le nombre de fois spécifié dans ce fichier, la sémantique **fair lock handoff** s'applique et le serveur n'est réveillé que si le verrou peut être pris.
La valeur par défaut est **5**.

Modification des paramètres sysctl

urandom_min_reseed_secs

Vous pouvez utiliser ce paramètre pour déterminer le nombre minimum de secondes entre le réensemencement du pool **urandom**. Ce fichier est accessible en écriture pour des raisons de compatibilité, mais son écriture n'a aucun effet sur le comportement du RNG.

seuil d'écriture du réveil

Lorsque le compte d'entropie passe sous ce seuil en un certain nombre de bits, vous pouvez réveiller les processus qui attendent d'écrire dans le fichier /**dev**/**random**. Ce fichier est accessible en écriture à des fins de compatibilité, mais son écriture n'a aucun effet sur le comportement du RNG.

CHAPITRE 6. PILOTES DE PÉRIPHÉRIQUES

6.1. NOUVEAUX CONDUCTEURS

Pilotes de réseau

- Plate-forme Firmware Runtime Update Pilote de télémétrie (**pfr_telemetry**)
- Platform Firmware Runtime Update device driver (**pfr_update**)
- Prise en charge de Bluetooth pour les appareils MediaTek ver 0.1 (**btmtk**)
- Interface hôte de l'ICM (**mhi**)
- Modem Host Interface (MHI) PCI controller driver (**mhi_pci_generic**)
- Pilote IDXD Pilote de type dsa_bus (**idxd_bus**)
- Pilote AMD PassThru DMA (**ptdma**)
- Pilote FAN Mellanox (**mlxreg-fan**)
- Pilote Mellanox LED regmap (**leds-mlxreg**)
- Pilote Intel® LPSS ACPI (**intel-lpss-acpi**)
- Pilote Intel® LPSS PCI (**intel-lpss-pci**)
- Pilote Intel® LPSS core (**intel-lpss**)
- Pilote Maxlinear Ethernet GPY (**mxl-gpy**)
- Pilote Realtek 802.11ax wireless 8852A (**rtw89_8852a**)
- Pilote Realtek 802.11ax wireless 8852AE (**rtw89_8852ae**)
- Pilote Intel® PMT Class (**pmt_class**)
- Pilote Intel® PMT Crashlog (**pmt_crashlog**)
- Pilote Intel® PMT Telemetry (**pmt_telemetry**)
- Pilote de boîte aux lettres Intel® speed select interface (**isst_if_mbox_msr**)
- Intel® speed select interface pci mailbox driver (**isst_if_mbox_pci**)
- Intel® speed select interface mmio driver (**isst_if_mmio**)
- Pilote Intel® Software Defined Silicon (**intel_sdsi**)
- Pilote de bus auxiliaire Intel® Extended Capabilities (**intel_vsec**)
- ISH ISHTP eclite client opregion driver (**ishtp_eclite**)
- Pilote de la radiocommande sans fil Acer (**acer-wireless**)
- Pilote d'interface de plate-forme AMD HSMP (**amd_hsmp**)

- DESIGNWARE HS OTG Core (**dwc2**)
- Synopsys HAPS PCI Glue Layer (**dwc3-haps**)
- DesignWare USB3 PCI Glue Layer (**dwc3-pci**)
- DesignWare USB3 DRD Controller Driver (**dwc3**)
- pilote de contrôleur d'hôte de plate-forme xHCI (**xhci-plat-hcd**)
- Pilote ON Semiconductor FSA4480 (**fsa4480**)
- Richtek RT1719 Sink Only USBPD Controller Driver (**rt1719**)
- Pilote du contrôleur de port Type-C Willsemi WUSB3801 (**wusb3801**)
- Pilote principal pour les périphériques PCI basés sur VFIO (**vfio-pci-core**)
- Pilote invité AMD SEV (**sev-guest**)
- Pilote de chien de garde Mellanox (**mlx_wdt**)

Pilotes graphiques et divers

- Support DSP Cirrus Logic (**cs_dsp**)
- Aide DRM DisplayPort (**drm_dp_helper**)
- DRM Buddy Allocator (**drm_buddy**)
- DRM SHMEM aides à la gestion de la mémoire (**drm_shmem_helper**)
- Pilote DRM utilisant l'interface bochs dispi (**bochs**)
- Pilote pour tablette Letsketch (**hid-letsketch**)
- Pilote d'interface de sélection de vitesse Intel® (**isst_if_common**)
- Pilote SiGma Micro HID (**hid-sigmamicro**)
- Correction des boutons latéraux de la souris Xiaomi Mi Silent Mouse (**hid-xiaomi**)
- Pilote pour souris DEC VSXXX-AA et -GA et tablette VSXXX-AB (**vsxxxaa**)
- Pilote de plate-forme pour carte de ligne Nvidia (**mlxreg-lc**)
- Pilote thermique Intel PCH (**intel_pch_thermal**)
- Pilote Intel LPSS UART (**8250_lpss**)

6.2. PILOTES MIS À JOUR

Mises à jour des pilotes de réseau

- Le pilote VMware vmxnet3 virtual NIC (**vmxnet3**) a été mis à jour à la version 1.7.0.0-k.

Mises à jour des pilotes de stockage

- Le pilote SCSI Emulex LightPulse Fibre Channel (**lpfc**) a été mis à jour à la version 14.2.0.5.
- MPI3 Storage Controller Device Driver (**mpi3mr**) a été mis à jour à la version 8.0.0.69.0.
- Le pilote de périphérique LSI MPT Fusion SAS 3.0 (**mpt3sas**) a été mis à jour à la version 40.100.00.00.
- Le pilote pour Microchip Smart Family Controller (**smartpqi**) a été mis à jour à la version 2.1.18-045.

Mises à jour des pilotes graphiques et divers

- Le pilote drm autonome pour le périphérique VMware SVGA (**vmwgfx**) a été mis à jour à la version 2.20.0.0.

CHAPITRE 7. CARACTÉRISTIQUES DU FBP DISPONIBLES

Ce chapitre fournit la liste complète des fonctionnalités de **Berkeley Packet Filter** (**BPF**) disponibles dans le noyau de cette version mineure de Red Hat Enterprise Linux 9. Les tableaux incluent les listes de :

- Configuration du système et autres options
- Types de programmes disponibles et aides prises en charge
- Types de cartes disponibles

Ce chapitre contient les résultats générés automatiquement par la commande **bpftool feature**.

Tableau 7.1. Configuration du système et autres options

Option	Valeur
non privilégié_bpf_désactivé	2 (bpf() syscall restreint aux utilisateurs privilégiés, l'administrateur peut le modifier)
Compilateur JIT	1 (activé)
Durcissement du compilateur JIT	1 (activé pour les utilisateurs non privilégiés)
Compilateur JIT kallsyms exports	1 (activé pour la racine)
Limite de mémoire pour JIT pour les utilisateurs non privilégiés	264241152
CONFIG_BPF	y
CONFIG_BPF_SYSCALL	y
CONFIG_HAVE_EBPF_JIT	y
CONFIG_BPF_JIT	y
CONFIG_BPF_JIT_ALWAYS_ON	y
CONFIG_DEBUG_INFO_BTF	y
CONFIG_DEBUG_INFO_BTF_MODULES	y
CONFIG_CGROUPS	y
CONFIG_CGROUP_BPF	y
CONFIG_CGROUP_NET_CLASSID	y

Option	Valeur
CONFIG_SOCK_CGROUP_DATA	y
CONFIG_BPF_EVENTS	y
CONFIG_KPROBE_EVENTS	y
CONFIG_UPROBE_EVENTS	y
CONFIG_TRACING	y
CONFIG_FTRACE_SYSCALLS	y
CONFIG_FUNCTION_ERROR_INJECTION	y
CONFIG_BPF_KPROBE_OVERRIDE	n
CONFIG_NET	y
CONFIG_XDP_SOCKETS	y
CONFIG_LWTUNNEL_BPF	y
CONFIG_NET_ACT_BPF	m
CONFIG_NET_CLS_BPF	m
CONFIG_NET_CLS_ACT	y
CONFIG_NET_SCH_INGRESS	m
CONFIG_XFRM	y
CONFIG_IP_ROUTE_CLASSID	y
CONFIG_IPV6_SEG6_BPF	n
CONFIG_BPF_LIRC_MODE2	n
CONFIG_BPF_STREAM_PARSER	y
CONFIG_NETFILTER_XT_MATCH_BPF	m
CONFIG_BPFILTER	n

Option	Valeur
CONFIG_BPFILTER_UMH	n
CONFIG_TEST_BPF	m
CONFIG_HZ	1000
bpf() syscall	disponible
Limite de taille des programmes	disponible

Tableau 7.2. Types de programmes disponibles et aides prises en charge

Type de programme	Aides disponibles
filtre_socket	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_load_bytes_relative, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
kprobe	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_strtr, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_group_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot

Type de programme	Aides disponibles
sched_cls	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_skb_adjust_room, bpf_skb_get_xfrm_state, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_skb_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock

Type de programme	Aides disponibles
sched_act	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_skb_adjust_room, bpf_skb_get_xfrm_state, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_skb_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
point de traçage	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_strtr, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_group_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot

Type de programme	Aides disponibles
xdp	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_redirect, bpf_perf_event_output, bpf_csum_diff, bpf_get_current_task, bpf_get_numa_node_id, bpf_xdp_adjust_head, bpf_redirect_map, bpf_xdp_adjust_meta, bpf_xdp_adjust_tail, bpf_fib_lookup, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
perf_event	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_perf_prog_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_read_branch_records, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_group_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot

Type de programme	Aides disponibles
cgroup_skb	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_load_bytes_relative, bpf_skb_group_id, bpf_get_local_storage, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_sk_cgroup_id, bpf_sk_ancestor_group_id, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
cgroup_sock	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_storage_get, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_group_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs
lwt_in	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_lwt_push_encap, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock

Type de programme	Aides disponibles
lwt_out	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
lwt_xmit	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_lwt_push_encap, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock

Type de programme	Aides disponibles
sock_ops	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_setsockopt, bpf_sock_map_update, bpf_getsockopt, bpf_sock_ops_cb_flags_set, bpf_sock_hash_update, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_tcp_sock, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_load_hdr_opt, bpf_store_hdr_opt, bpf_reserve_hdr_opt, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
sk_skb	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_adjust_room, bpf_sk_redirect_map, bpf_sk_redirect_hash, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
cgroup_device	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_uid_gid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs

Type de programme	Aides disponibles
sk_msg	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_msg_redirect_map, bpf_msg_apply_bytes, bpf_msg_cork_bytes, bpf_msg_pull_data, bpf_msg_redirect_hash, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_msg_push_data, bpf_msg_pop_data, bpf_spin_lock, bpf_spin_unlock, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
raw_tracepoint	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_strtr, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_group_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot

Type de programme	Aides disponibles
cgroup_sock_addr	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_setsockopt, bpf_getsockopt, bpf_bind, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_group_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
lwt_seg6local	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
lirc_mode2	non pris en charge
sk_reuseport	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_skb_load_bytes_relative, bpf_sk_select_reuseport, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs

Type de programme	Aides disponibles
flow_dissector	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_strtr, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock
cgroup_sysctl	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_uid_gid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sysctl_get_name, bpf_sysctl_get_current_value, bpf_sysctl_get_new_value, bpf_sysctl_set_new_value, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs
raw_tracepoint_writable	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_strtr, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_group_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot

Type de programme	Aides disponibles
cgroup_sockopt	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_uid_gid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_tcp_sock, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs
traçage	non pris en charge

Type de programme	Aides disponibles
struct_ops	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_perf_event_read, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_stackid, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_xdp_adjust_head, bpf_probe_read_str, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_setsockopt, bpf_skb_adjust_room, bpf_redirect_map, bpf_sk_redirect_map, bpf_sock_map_update, bpf_xdp_adjust_meta, bpf_perf_event_read_value, bpf_perf_prog_read_value, bpf_getsockopt, bpf_override_return, bpf_sock_ops_cb_flags_set, bpf_msg_redirect_map, bpf_msg_apply_bytes, bpf_msg_cork_bytes, bpf_msg_pull_data, bpf_bind, bpf_xdp_adjust_tail, bpf_skb_get_xfrm_state, bpf_get_stack, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_sock_hash_update, bpf_msg_redirect_hash, bpf_sk_redirect_hash, bpf_lwt_push_encap, bpf_lwt_seg6_store_bytes, bpf_lwt_seg6_adjust_srh, bpf_lwt_seg6_action, bpf_rc_repeat, bpf_rc_keydown, bpf_skb_group_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_sk_select_reuseport, bpf_skb_ancestor_group_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_msg_push_data, bpf_msg_pop_data, bpf_rc_pointer_rel, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_sysctl_get_name, bpf_sysctl_get_current_value, bpf_sysctl_get_new_value, bpf_sysctl_set_new_value, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_send_signal, bpf_tcp_gen_syncookie, bpf_skb_output, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_strtr, bpf_probe_read_kernel_str, bpf_tcp_send_ack, bpf_send_signal_thread, bpf_jiffies64, bpf_read_branch_records, bpf_get_ns_current_pid_tgid, bpf_xdp_output, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_seq_printf, bpf_seq_write, bpf_sk_cgroup_id, bpf_sk_ancestor_cgroup_id, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_get_task_stack, bpf_load_hdr_opt, bpf_store_hdr_opt, bpf_reserve_hdr_opt, bpf_inode_storage_get, bpf_inode_storage_delete, bpf_d_path, bpf_copy_from_user, bpf_snprintf_btf, bpf_seq_printf_btf, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_bprm_opts_set, bpf_ktime_get_coarse_ns, bpf_ima_inode_hash, bpf_sock_from_file, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_sys_bpf, bpf_btf_find_by_name_kind, bpf_sys_close, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_skc_to_unix_sock, bpf_kallsyms_lookup_name
ext	non pris en charge

Type de programme	Aides disponibles
lsm	non pris en charge
sk_lookup	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock

Tableau 7.3. Types de cartes disponibles

Type de carte	Disponible
hachage	yes
réseau	yes
prog_array	yes
perf_event_array	yes
percpu_hash	yes
tableau percpu_	yes
trace_de_pile	yes
cgroup_array	yes
lru_hash	yes
lru_percpu_hash	yes
lpm_trie	yes
array_of_maps	yes

Type de carte	Disponible
hash_of_maps	yes
carte de données	yes
carte du stock	yes
cpumap	yes
xskmap	yes
sockhash	yes
cgroup_storage	yes
reuseport_sockarray	yes
percpu_cgroup_storage	yes
file d'attente	yes
pile	yes
sk_storage	yes
devmap_hash	yes
struct_ops	non
ringbuf	yes
inode_storage	yes
stockage_de_tâches	yes

CHAPITRE 8. BUG FIXES

Cette partie décrit les bogues corrigés dans Red Hat Enterprise Linux 9.1 qui ont un impact significatif sur les utilisateurs.

8.1. CRÉATION D'INSTALLATEURS ET D'IMAGES

Le programme d'installation n'installe plus les versions antérieures des paquets

Auparavant, le programme d'installation ne chargeait pas correctement le fichier de configuration DNF au cours du processus d'installation. En conséquence, le programme d'installation installait parfois des versions antérieures de certains paquets dans la transaction RPM.

Ce bogue a été corrigé et seules les dernières versions des paquets sont désormais installées à partir des référentiels d'installation. Dans les cas où il est impossible d'installer les dernières versions des paquets, l'installation échoue comme prévu.

(BZ#2053710)

L'installation d'Anaconda est réussie même si l'on modifie la configuration du réseau à l'étape 2

Auparavant, lors de l'utilisation de l'argument de démarrage **rd.live.ram**, Anaconda ne démontait pas un point de montage NFS utilisé dans **initramfs** pour récupérer l'image d'installation en mémoire. En conséquence, le processus d'installation pouvait ne plus répondre ou échouer avec une erreur de dépassement de délai si la configuration du réseau était modifiée à l'étape 2.

Pour résoudre ce problème, le point de montage NFS utilisé pour récupérer l'image d'installation en mémoire est démonté dans **initramfs** avant switchroot. Ainsi, le processus d'installation s'achève sans interruption.

(BZ#2082132)

8.2. GESTION DES ABONNEMENTS

virt-who se connecte désormais correctement aux serveurs ESX en mode FIPS

Auparavant, lors de l'utilisation de l'utilitaire **virt-who** sur un système RHEL 9 en mode FIPS, **virt-who** ne pouvait pas se connecter aux serveurs ESX. Par conséquent, **virt-who** ne signalait aucun serveur ESX, même s'il était configuré pour cela, et enregistrait le message d'erreur suivant :

```
ValueError: [digital envelope routines] unsupported
```

Avec cette mise à jour, **virt-who** a été corrigé pour gérer correctement le mode FIPS, et le problème décrit ne se produit plus.

(BZ#2054504)

8.3. GESTION DES LOGICIELS

Le DNF annule désormais correctement une transaction contenant un élément avec le type d'action Reason Change

Auparavant, l'exécution de la commande **dnf history rollback** sur une transaction contenant un élément avec le type d'action **Reason Change** échouait. Avec cette mise à jour, le problème a été corrigé et **dnf history rollback** fonctionne désormais comme prévu.

(BZ#2053014)

8.4. SHELLS ET OUTILS DE LIGNE DE COMMANDE

La commande vi dans ReaR n'entraîne plus de boucle infinie

Auparavant, le système de sauvetage ReaR ne contenait pas l'exécutable **vi**, mais seulement le script **/bin/vi**. Par conséquent, le script **/bin/vi** provoquait une boucle infinie lorsqu'il était invoqué. Avec cette mise à jour, le système de sauvetage ReaR contient l'exécutable **vi /usr/libexec/vi** , et l'exécution de la commande **vi** ne provoque plus de boucle infinie.

(BZ#2097437)

ReaR avec la méthode de sortie PXE n'échoue plus à stocker les fichiers de sortie dans l'emplacement rsync OUTPUT_URL

Auparavant, la gestion de la variable **OUTPUT_URL** avec les options **OUTPUT=PXE** et **BACKUP=RSYNC** a été supprimée. Par conséquent, lors de l'utilisation d'un emplacement rsync pour **OUTPUT_URL**, ReaR ne parvenait pas à copier les fichiers **initrd** et kernel à cet emplacement, bien qu'il les téléchargeait à l'emplacement spécifié par **BACKUP_URL**. Avec cette mise à jour, le comportement de RHEL 8.4 et des versions antérieures est rétabli. ReaR crée les fichiers requis à la destination **OUTPUT_URL** désignée à l'aide de rsync.

(BZ#2115958)

ReaR n'affiche plus de message d'erreur s'il ne met pas à jour l'UUID dans le fichier /etc/fstab

Auparavant, ReaR n'affichait pas de message d'erreur pendant la restauration lorsqu'il ne parvenait pas à mettre à jour l'identifiant universel unique (UUID) dans **/etc/fstab** pour qu'il corresponde à l'UUID de la partition nouvellement créée dans le cas où les UUID étaient différents. Cela aurait pu se produire si l'image de secours n'était pas synchronisée avec la sauvegarde. Avec cette mise à jour, un message d'erreur apparaît pendant la récupération si les fichiers du système de base restaurés ne correspondent pas au système recréé.

(BZ#2083272)

ReaR prend désormais en charge la restauration d'un système à l'aide de NetBackup version 9

Auparavant, la restauration d'un système à l'aide de la méthode NetBackup (NBU) avec NetBackup version 9 ou ultérieure échouait en raison de l'absence de bibliothèques et d'autres fichiers. Avec cette mise à jour, la variable **NBU_LD_LIBRARY_PATH** contient les chemins d'accès aux bibliothèques nécessaires et le système de secours intègre désormais les fichiers requis, et ReaR peut utiliser la méthode NetBackup.

(BZ#2120736)

ReaR n'affiche plus de faux message d'erreur concernant des cibles de liens symboliques manquantes

Auparavant, ReaR affichait des messages d'erreur incorrects concernant des cibles de liens symboliques manquantes pour les liens symboliques **build** et **source** sous **/usr/lib/modules/** lors de la création de l'image de secours. Cette situation était inoffensive et vous pouviez ignorer le message d'erreur. Avec

cette mise à jour, ReaR ne signale pas de faux message d'erreur concernant des cibles de liens symboliques manquantes dans cette situation.

(BZ#2119501)

L'opération cmx sans paramètre ne fait plus planter le client CIM

L'opération **cmx** appelle une méthode et renvoie du XML, un paramètre spécifie le nom de la méthode appelée. Auparavant, la ligne de commande **sblim-wbemcli** Common Information Model (CIM) Client se bloquait lors de l'exécution de l'opération **cmx** sans paramètre supplémentaire. Avec cette mise à jour, l'opération **cmx** nécessite le paramètre qui définit le nom de la méthode appelée. L'invocation de l'opération **cmx** sans ce paramètre génère un message d'erreur et le client CIM ne se bloque plus.

(BZ#2083577)

free la commande utilise une nouvelle méthode de calcul de la mémoire utilisée

Auparavant, le calcul de la mémoire utilisée dans l'utilitaire **free** soustrayait l'espace libre, l'espace cache et l'espace tampon de la mémoire totale. Par conséquent, une divergence survenait lorsque vous compariez la valeur de la mémoire utilisée avec les résultats d'un autre outil, car l'utilitaire **free** ne calculait pas la mémoire partagée. Avec cette mise à jour, la commande **free** utilise une nouvelle méthode de calcul qui fournit un état clair de la mémoire libre et prend en compte le cache non récupérable. La mémoire utilisée correspond désormais à toute mémoire qui n'est pas disponible et inclut également les objets **tmpfs** qui se trouvent dans la mémoire virtuelle.

(BZ#2003033)

8.5. SERVICES D'INFRASTRUCTURE

Unbound ne valide plus les signatures RSA basées sur SHA-1

Auparavant, OpenSSL ne validait pas les signatures RSA basées sur SHA-1 dans la politique cryptographique du système DEFAULT. Par conséquent, lorsque Unbound essayait de valider ces signatures, l'erreur d'OpenSSL entraînait l'échec de la résolution. Avec cette mise à jour, Unbound désactive la prise en charge de la validation de toutes les signatures RSA/SHA1 (algorithme numéro 5) et RSASHA1-NSEC3-SHA1 (algorithme numéro 7), ce qui résout la question. Notez que cela rend le résultat peu sûr dans le cadre de toutes les politiques cryptographiques à l'échelle du système.

(BZ#2071543)

8.6. SÉCURITÉ

La génération de clés OpenSSH utilise des interfaces compatibles FIPS

La bibliothèque cryptographique OpenSSL, utilisée par OpenSSH, propose deux interfaces : l'ancienne et la nouvelle. Auparavant, OpenSSH utilisait l'ancienne interface pour la génération de clés, qui n'était pas conforme aux exigences de la Federal Information Processing Standards (FIPS). Avec cette mise à jour, l'utilitaire **ssh-keygen** utilise l'API conforme aux normes FIPS au lieu de l'API de bas niveau incompatible avec les normes FIPS. Par conséquent, la génération de clés OpenSSH est conforme aux normes FIPS.

(BZ#2087121)

La cryptographie non approuvée par le FIPS ne fonctionne plus dans OpenSSL en mode FIPS

Auparavant, la cryptographie non approuvée par le FIPS fonctionnait dans la boîte à outils OpenSSL,

quels que soient les paramètres du système. Par conséquent, vous pouviez utiliser des algorithmes cryptographiques qui devraient être désactivés lorsque le système fonctionne en mode FIPS, par exemple :

- Les suites de chiffrement TLS utilisant l'échange de clés RSA ont fonctionné.
- Les algorithmes de cryptage et de décryptage à clé publique basés sur la technologie RSA ont fonctionné malgré l'utilisation des blocs PKCS #1 et SSLv23 ou de clés d'une longueur inférieure à 2048 bits.

Cette mise à jour contient des corrections garantissant que la cryptographie non approuvée par FIPS ne fonctionne plus dans OpenSSL en mode FIPS.

(BZ#2053289)

La spécification de courbes arbitraires est supprimée d'OpenSSL

Auparavant, les vérifications de la sécurité des paramètres explicites des courbes étaient incomplètes. Par conséquent, des courbes elliptiques arbitraires avec des valeurs **p** suffisamment grandes fonctionnaient dans RHEL. Avec cette mise à jour, les contrôles vérifient désormais que les paramètres explicites de la courbe correspondent à l'une des courbes bien connues prises en charge. Par conséquent, l'option permettant de spécifier des courbes arbitraires par l'utilisation de paramètres de courbe explicites a été supprimée d'OpenSSL. Les fichiers de paramètres, les clés privées, les clés publiques et les certificats qui spécifient des courbes explicites arbitraires ne fonctionnent plus dans OpenSSL. L'utilisation de paramètres de courbes explicites pour spécifier l'une des courbes bien connues et prises en charge telles que P-224, P-256, P-384, P-521 et **secp256k1** reste prise en charge en mode non-FIPS.

(BZ#2066412)

OpenSSL req utilise AES-256-CBC pour le cryptage des clés privées

Auparavant, l'outil OpenSSL **req** chiffrait les fichiers de clés privées en utilisant l'algorithme 3DES. L'algorithme 3DES n'étant pas sûr et n'étant pas autorisé par la norme FIPS 140 actuelle pour les modules cryptographiques, **req** génère désormais des fichiers de clés privées chiffrés à l'aide de l'algorithme AES-256-CBC. Le format général du fichier PKCS#8 reste inchangé.

(BZ#2063947)

OpenSSL n'échoue plus à se connecter lorsque FFDHE est utilisé

Auparavant, les connexions TLS qui utilisent le mécanisme d'échange de clés éphémères FFDHE (Diffie-Hellman à champ fini) échouaient parfois lors du traitement des partages de clés FFDHE provenant d'un client. Cela était dû à des vérifications trop restrictives dans OpenSSL. En conséquence, le serveur OpenSSL interrompait la connexion avec une alerte **internal_error**. Avec cette mise à jour, OpenSSL accepte des parts de clés plus petites mais toujours conformes. Par conséquent, les connexions entre OpenSSL et d'autres implémentations ne sont plus interrompues de manière aléatoire lors de l'utilisation d'échanges de clés FFDHE.

(BZ#2004915)

Les applications basées sur OpenSSL fonctionnent désormais correctement avec les paramètres linguistiques turcs

La bibliothèque **OpenSSL** utilisant des fonctions de comparaison de chaînes insensibles à la casse, les applications basées sur OpenSSL ne fonctionnaient pas correctement avec les paramètres linguistiques turcs, et les vérifications omises provoquaient le plantage des applications utilisant ces paramètres. Cette mise à jour fournit un correctif permettant d'utiliser les paramètres régionaux POSIX (Portable

Operating System Interface) pour la comparaison de chaînes insensibles à la casse. Par conséquent, les applications basées sur OpenSSL telles que curl fonctionnent correctement avec les paramètres linguistiques turcs.

(BZ#2071631)

Permissions pour **insights-client** ajoutées à la politique SELinux

Le nouveau service **insights-client** nécessite des autorisations qui n'existaient pas dans les versions précédentes de **selinux-policy**. En conséquence, certains composants de **insights-client** ne fonctionnaient pas correctement et signalaient des messages d'erreur AVC (Access Vector Cache). Cette mise à jour ajoute de nouvelles autorisations à la politique SELinux. En conséquence, **insights-client** fonctionne correctement sans signaler d'erreurs AVC.

(BZ#2081425, BZ#2077377, BZ#2087765, BZ#2107363)

Les utilisateurs de SELinux **staff_u** ne peuvent plus basculer par erreur vers la fonction **unconfined_r**

Auparavant, lorsque le booléen **secure_mode** était activé, les utilisateurs de **staff_u** pouvaient passer au rôle **unconfined_r**, ce qui n'était pas le comportement attendu. En conséquence, les utilisateurs de **staff_u** pouvaient effectuer des opérations privilégiées affectant la sécurité du système. Avec cette mise à jour, la politique SELinux a été corrigée et les utilisateurs de **staff_u** ne peuvent plus passer à tort à **unconfined_r**.

(BZ#2076681)

OpenSCAP ne produit plus d'erreurs incorrectes lors de la vérification de la mémoire disponible

Auparavant, lors de l'évaluation de certaines règles XCCDF, OpenSCAP affichait de manière incorrecte le message d'erreur **Failed to check available memory** et produisait des résultats d'analyse non valides. Par exemple, cela se produisait pour les règles **accounts_user_dot_no_world_writable_programs**, **accounts_user_dot_group_ownership** et **accounts_users_home_files_permissions**. Avec cette mise à jour, le bogue dans la gestion des erreurs est corrigé et le message d'erreur n'apparaît que pour les échecs réels.

(BZ#2109485)

fagenrules --load fonctionne désormais correctement

Auparavant, le service **fapolicyd** ne gérait pas correctement le signal de raccrochage (SIGHUP). Par conséquent, **fapolicyd** se terminait après avoir reçu le signal SIGHUP et la commande **fagenrules --load** ne fonctionnait pas correctement. Cette mise à jour contient un correctif pour ce problème. Par conséquent, **fagenrules --load** fonctionne désormais correctement et les mises à jour des règles ne nécessitent plus le redémarrage manuel de **fapolicyd**.

(BZ#2070655)

8.7. MISE EN RÉSEAU

Une instance conserve désormais l'adresse IP principale même après avoir démarré le service **nm-cloud-setup** dans Alibaba Cloud

Auparavant, après le lancement d'une instance dans Alibaba Cloud, le service **nm-cloud-setup** configurait l'adresse IP incorrecte comme adresse IP primaire en cas d'adresses IPv4 multiples. Par conséquent, cela affectait la sélection de l'adresse source IPv4 pour les connexions sortantes. Avec

cette mise à jour, après avoir configuré manuellement les adresses IP secondaires, le paquet **NetworkManager** récupère l'adresse IP primaire à partir des métadonnées **primary-ip-address** et configure correctement les adresses IP primaires et secondaires.

(BZ#2079849)

L'utilitaire NetworkManager permet d'ordonner correctement les adresses IPv6 ajoutées manuellement

En général, l'ordre des adresses IPv6 affecte la priorité de la sélection de l'adresse source. Par exemple, lorsque vous établissez une connexion TCP sortante. Auparavant, la priorité relative des adresses IPv6 ajoutées par les méthodes **manual**, **dhcpv6** et **autoconf6** n'était pas correcte. Cette mise à jour corrige le problème et l'ordre de priorité reflète désormais la logique suivante : **manual** > **dhcpv6** > **autoconf6**. En outre, l'ordre des adresses sous le paramètre **ipv6.addresses** a été inversé de sorte que l'adresse ajoutée en premier a la priorité la plus élevée.

(BZ#2097293)

8.8. NOYAU

Le marquage des prises de réseau fonctionne à nouveau

Certains contrôleurs **cgroup** v1 qui n'ont pas d'équivalent **cgroup** v2, tels que **net_prio** ou **net_cls**, interféraient auparavant avec le marquage de la prise **cgroup** v2 lorsqu'ils étaient montés avec d'autres contrôleurs **cgroup** v2 dans un environnement mixte **cgroup** v1/v2. Par conséquent, un environnement mixte **cgroup** v1/v2 utilisant le contrôleur **net_prio** ou **net_cls** v1 empêchait le marquage correct des sockets réseau avec **cgroup** v2. Cette mise à jour élimine cette limitation, ce qui permet d'utiliser le marquage des sockets réseau dans un environnement mixte cgroup v1/v2.

(BZ#2060150)

Le paquet kexec-tools prend désormais en charge les valeurs de réservation de mémoire par défaut de crashkernel

Le paquetage **kexec-tools** conserve désormais les valeurs par défaut de réservation de la mémoire de **crashkernel**. Le service **kdump** utilise la valeur par défaut pour réserver la mémoire du noyau de crash pour chaque noyau. Cette implémentation améliore également l'allocation de mémoire pour **kdump** lorsqu'un système dispose de moins de 4 Go de mémoire disponible.

Si la mémoire réservée par la valeur par défaut de **crashkernel** n'est pas suffisante sur votre système, vous pouvez utiliser la commande **kdumpctl estimate** pour obtenir une valeur estimée sans provoquer de plantage. La valeur estimée de **crashkernel=** peut ne pas être exacte et peut servir de référence pour définir une valeur appropriée de **crashkernel=**.

(BZ#1959203)

Les systèmes peuvent exécuter avec succès des opérations dynamiques sur les LPAR

Auparavant, les utilisateurs ne pouvaient pas exécuter d'opérations de partition logique dynamique (DLPAR) à partir de la console de gestion du matériel (HMC) si l'une ou l'autre de ces conditions était remplie :

- La fonction Secure Boot a été activée, ce qui permet d'activer implicitement le mécanisme **lockdown** du noyau en mode intégrité.
- Le mécanisme du noyau **lockdown** a été activé manuellement en mode intégrité ou confidentialité.

Dans RHEL 9, le noyau **lockdown** bloque complètement l'accès des Run Time Abstraction Services (RTAS) à la mémoire système accessible via le fichier de périphérique de caractères **/dev/mem**. Plusieurs appels RTAS nécessitaient un accès en écriture à **/dev/mem** pour fonctionner correctement. Par conséquent, les appels RTAS ne s'exécutaient pas correctement et les utilisateurs voyaient le message d'erreur suivant :

> HSCL2957 Il n'y a actuellement aucune connexion RMC entre la console de gestion et la partition <LPAR name> ou la partition ne prend pas en charge les opérations de partitionnement dynamique. Vérifiez la configuration du réseau sur la console de gestion et la partition et assurez-vous que l'authentification du pare-feu entre la console de gestion et la partition a eu lieu. Exécutez la commande diagrmc de la console de gestion pour identifier les problèmes qui pourraient être à l'origine de l'absence de connexion RMC.

Avec cette mise à jour, le problème a été résolu en fournissant une exception très étroite, spécifique au PowerPC, à **lockdown**. Cette exception permet à RTAS d'accéder aux zones requises de **/dev/mem**. Par conséquent, le problème ne se manifeste plus dans le scénario décrit.

(BZ#2046472)

Aucun avertissement du noyau après avoir réglé la valeur du tampon circulaire de rx à max

Le noyau produisait un message d'avertissement **Missing unregister, handled but fix driver** lorsqu'une fonction interne attendant une entrée propre était appelée avec une structure réutilisée et déjà initialisée. Avec cette mise à jour, le problème a été corrigé en réinitialisant la structure avant de l'enregistrer à nouveau.

(BZ#2054379)

8.9. CHARGEUR DE DÉMARRAGE

grubby transmet désormais les arguments aux futurs noyaux

Lors de l'installation d'une version plus récente du noyau, l'outil **grubby** ne transmettait pas les arguments de la ligne de commande du noyau de la version précédente. Par conséquent, le chargeur de démarrage GRUB ignorait les paramètres de l'utilisateur. Avec cette correction, les paramètres de l'utilisateur persistent maintenant après l'installation de la nouvelle version du noyau.

(BZ#1978226)

8.10. SYSTÈMES DE FICHIERS ET STOCKAGE

Les entrées de journal n'interrompent plus l'écriture du journal

Auparavant, dans le pilote VDO, pendant l'opération de suspension du device-mapper et après la reprise du fonctionnement du device, certains blocs de journal pouvaient encore être marqués comme étant en attente de certaines mises à jour de métadonnées avant de pouvoir être réutilisés, même si ces mises à jour avaient déjà été effectuées. Lorsqu'il y avait suffisamment d'entrées dans le journal pour que le journal puisse revenir au même bloc physique, celui-ci n'était pas disponible. Les écritures dans le journal s'arrêtaient, attendant que le bloc devienne disponible, ce qui ne se produisait jamais. Par conséquent, lorsque certaines opérations sur un périphérique VDO comprenaient un cycle de suspension ou de reprise, le périphérique se trouvait dans un état figé après certaines mises à jour du journal. Les mises à jour du journal avant cet état étaient imprévisibles car elles dépendaient des schémas d'allocation précédents au sein de la VDO, et des schémas d'écriture ou d'abandon entrants. Avec cette mise à jour, après le cycle de suspension ou de reprise de l'enregistrement des données, l'état de la structure de données interne est réinitialisé et les blocages ne se produisent plus.

(BZ#2064802)

L'ajout d'un périphérique de données ne déclenche plus d'échec d'assertion

Auparavant, lors de l'ajout de périphériques supplémentaires au cache, Stratis n'utilisait pas le cache immédiatement après l'initialisation. Par conséquent, le service **stratisd** renvoyait un message d'échec d'assertion chaque fois qu'un utilisateur tentait d'ajouter des périphériques de données supplémentaires à un pool. Avec cette correction, le cache est maintenant utilisé immédiatement après l'initialisation et aucun échec d'assertion ne se produit.

(BZ#2007018)

Résolution d'erreurs lors de l'ajout de nouveaux périphériques de données au pool crypté

Auparavant, lorsque l'utilisateur initialisait un pool chiffré avec des périphériques de données chiffrés, à l'aide d'une commande Clevis bind sur un serveur tang, spécifié avec l'option **--trust-url**, **stratisd** n'incluait pas la partie thumbprint de la configuration Clevis tang dans les structures de données internes. Par conséquent, un échec se produisait lors de la tentative d'ajout de nouveaux périphériques de données au pool. Avec cette mise à jour, les structures de données internes de **stratisd** incluent désormais la partie thumbprint de la configuration Clevis tang.

(BZ#2005110)

La connexion à des espaces de noms NVMe à partir d'initiateurs Broadcom sur des systèmes AMD EPYC ne nécessite plus de paramètres IOMMU autres que ceux par défaut

Par défaut, le noyau RHEL active l'IOMMU sur les plateformes basées sur AMD. Auparavant, le pilote **lpfc** n'utilisait pas les macros d'accesseur de la liste de dispersion. Par conséquent, certains serveurs équipés de processeurs AMD rencontraient des problèmes d'E/S NVMe, tels que des E/S échouant en raison de la non-concordance des longueurs de transfert.

Avec cette mise à jour, il n'est plus nécessaire de mettre IOMMU en mode passthrough avec une option de ligne de commande du noyau pour se connecter à des espaces de noms NVMe à partir d'initiateurs Broadcom.

(BZ#2073541)

8.11. HAUTE DISPONIBILITÉ ET CLUSTERS

pcs valide désormais la valeur de stonith-watchdog-timeout

Auparavant, il était possible de définir la propriété **stonith-watchdog-timeout** à une valeur incompatible avec la configuration du SBD. Cela pouvait entraîner une boucle de clôture ou faire en sorte que le cluster considère une action de clôture comme réussie même si l'action n'est pas terminée. Avec cette correction, **pcs** valide la valeur de **stonith-watchdog-property** lorsque vous la définissez, afin d'éviter une configuration incorrecte.

(BZ#2058246)

pcs reconnaît désormais l'option mode lors de la création d'un nouveau ticket Booth

Auparavant, lorsqu'un utilisateur spécifiait l'option **mode** lors de l'ajout d'un nouveau ticket de stand, **pcs** signalait l'erreur **invalid booth ticket option 'mode'**. Avec cette correction, vous pouvez maintenant spécifier l'option **mode** lors de la création d'un ticket de stand.

(BZ#2058243)

pcs fait désormais la distinction entre les ressources et les ressources en pierre

Auparavant, certaines commandes de **pcs** ne faisaient pas de distinction entre les ressources et les ressources stonith. Cela permettait aux utilisateurs d'utiliser les sous-commandes **pcs resource** pour les ressources stonith et d'utiliser les sous-commandes **pcs stonith** pour les ressources qui ne sont pas des ressources stonith. Cela pouvait entraîner une confusion chez les utilisateurs ou une mauvaise configuration des ressources. Avec cette mise à jour, **pcs** affiche un avertissement en cas de non concordance des types de ressources.

(BZ#1301204)

8.12. COMPILATEURS ET OUTILS DE DÉVELOPPEMENT

glibc restaure maintenant errno après le chargement d'un module NSS

Auparavant, l'implémentation de Name Service Switch (NSS) dans **glibc** définissait errno de manière incorrecte pendant l'énumération de la base de données à l'aide de fonctions telles que **getpwent()** si le dernier module NSS ne fournissait aucune donnée. Par conséquent, les applications utilisant ces fonctions d'énumération observaient des erreurs et échouaient. **glibc** rétablit désormais errno après le chargement d'un module NSS et, par conséquent, les applications utilisant ces fonctions n'échouent plus.

(BZ#2063142)

L'interface d'audit sauvegarde et restaure désormais le registre x8 et la totalité des registres NEON pour AArch64

Auparavant, un bogue dans l'implémentation de l'interface d'audit du chargeur dynamique faisait que l'état des registres sauvegardés sur **AArch64** était incomplet par rapport à la norme d'appel de procédure. Ce bogue a été corrigé et l'interface d'audit sauvegarde et restaure désormais le registre x8 et toute la largeur des registres NEON pour **AArch64**. Les applications utilisant l'interface d'audit du chargeur dynamique peuvent désormais inspecter et influencer le registre x8 pour **AArch64**. Pour utiliser ce nouveau registre x8 et avoir accès à toute la largeur des registres NEON sur **AArch64**, les modules d'audit doivent être recompilés pour utiliser la nouvelle version de l'interface (LAV_CURRENT est 2).

(BZ#2003291)

La fonction strncpy optimisée pour POWER9 ne donne plus de résultats incorrects

Auparavant, la fonction strncpy de POWER9 n'utilisait pas le registre correct comme source des octets NUL pour le remplissage. Par conséquent, le tampon de sortie contenait un contenu de registre non initialisé à la place du remplissage NUL. Avec cette mise à jour, la fonction strncpy a été corrigée et la fin du tampon de sortie est maintenant correctement remplie d'octets NUL.

(BZ#2091549)

Valgrind override of glibc memmem function installé sur l'architecture IBMz15

Auparavant, une surcharge valgrind manquante de la fonction **glibc memmem** conduisait à des avertissements faussement positifs :

```
Conditional jump or move depends on uninitialised value(s)
```

Cette mise à jour inclut un remplacement par valgrind de la fonction **glibc memmem** et, par conséquent, il n'y a plus d'avertissements faussement positifs lors de l'utilisation de la fonction **memmem** dans les programmes exécutés sous valgrind sur l'architecture IBMz15.

(BZ#1993976)

8.13. GESTION DE L'IDENTITÉ

La sortie de ipa user-del --preserve user_login n'indique plus que l'utilisateur a été supprimé

Auparavant, si vous exécutiez la commande **ipa user-del --preserve user_login** pour préserver un compte d'utilisateur, le résultat renvoyait à tort le message **Deleted user "user_login"**. Avec cette mise à jour, la sortie renvoie désormais **Preserved user "user_login"**.

(BZ#2100227)

L'authentification des utilisateurs PKINIT fonctionne désormais correctement dans le scénario client Kerberos RHEL 9 - KDC Heimdal

Auparavant, l'authentification PKINIT d'un utilisateur IdM sur un client Kerberos RHEL 9 contre le centre de distribution Kerberos (KDC) Heimdal échouait. Cet échec était dû au fait que le client Kerberos ne prenait pas en charge le champ **supportedCMSTypes** requis dans le contexte de la dépréciation de l'algorithme SHA-1 dans RHEL 9.

Avec cette mise à jour, le client Kerberos RHEL 9 envoie une liste d'algorithmes de signature comprenant **sha512WithRSAEncryption**, et **sha256WithRSAEncryption** en tant que **supportedCMSTypes** pendant PKINIT à Heimdal KDC. Heimdal KDC utilise **sha512WithRSAEncryption** et, par conséquent, l'authentification PKINIT fonctionne correctement.

(BZ#2068935)

Gestion des objets illisibles dans la liste des membres d'un groupe LDAP

Avant cette mise à jour, SSSD traitait de manière incohérente les objets illisibles dans la liste des membres d'un groupe LDAP, ce qui provoquait une erreur ou, dans certaines situations, les objets illisibles étaient ignorés.

Avec cette mise à jour, SSSD dispose d'une nouvelle option **ldap_ignore_unreadable_references** pour modifier ce comportement. Si l'option **ldap_ignore_unreadable_references** est définie sur **false**, les objets illisibles provoquent une erreur et si elle est définie sur **true**, les objets illisibles sont ignorés. La valeur par défaut est **false** et, en raison de l'incohérence du comportement initial, il se peut que certaines recherches de groupes échouent après la mise à jour. Dans ce cas, définissez **ldap_ignore_unreadable_references = True** dans la section **[domain/*name of the domain*]** correspondante dans le fichier **/etc/sssd/sssd.conf**.

Cela permet de gérer les objets illisibles de manière cohérente et le comportement peut être ajusté à l'aide de la nouvelle option **ldap_ignore_unreadable_references**.

(BZ#2069376)

8.14. BUREAU

L'enregistrement des abonnements avec des clés d'activation a été corrigé

Auparavant, vous ne pouviez pas enregistrer votre abonnement Red Hat dans **Settings** en utilisant des clés d'activation. **Settings** affichait l'erreur suivante après avoir appuyé sur **Register**:

> Failed to register system; Failed to RegisterWithActivationKeys: Unknown arguments: dict_keys(['enable_content'])

Avec cette mise à jour, le problème a été corrigé et vous pouvez maintenant inscrire votre abonnement en utilisant des clés d'activation comme prévu dans **Settings**.

(BZ#2100467)

8.15. INFRASTRUCTURES GRAPHIQUES

X.org active désormais l'extension X11 SECURITY

Auparavant, le serveur d'affichage X.org ne fournissait pas l'extension X11 **SECURITY**. Par conséquent, les applications qui utilisaient cette extension se terminaient de manière inattendue.

Avec cette mise à jour, X.org active l'extension X11 **SECURITY**. Par conséquent, les applications qui dépendent de cette extension fonctionnent désormais comme prévu.

(BZ#1894612)

Le GPU Matrox avec un écran VGA fonctionne désormais comme prévu

Avant cette version, votre écran n'affichait aucune sortie graphique si vous utilisiez la configuration système suivante :

- Un GPU de la famille Matrox MGA G200
- Un écran connecté au contrôleur VGA
- L'UEFI est passé en mode hérité

Par conséquent, il n'est pas possible d'utiliser ou d'installer RHEL sur cette configuration.

Avec cette mise à jour, le pilote **mgag200** a été réécrit de manière significative et, par conséquent, la sortie graphique fonctionne maintenant comme prévu.

(BZ#2100898)

8.16. LA CONSOLE WEB

La suppression de périphériques hôtes USB à l'aide de la console web fonctionne désormais comme prévu

Auparavant, lorsque vous attachiez un périphérique USB à une machine virtuelle (VM), le numéro de périphérique et le numéro de bus du périphérique USB changeaient après avoir été transmis à la VM. Par conséquent, l'utilisation de la console web pour supprimer ces périphériques échouait en raison de la corrélation incorrecte des numéros de périphérique et de bus. Avec cette mise à jour, le problème a été corrigé et vous pouvez supprimer les périphériques hôtes USB à l'aide de la console web.

(JIRA:RHELPLAN-109067)

L'attachement de plusieurs périphériques hôtes à l'aide de la console web fonctionne désormais comme prévu

Auparavant, lorsque vous sélectionniez plusieurs périphériques à attacher à une machine virtuelle (VM) à l'aide de la console web, seul un périphérique était attaché et les autres étaient ignorés. Avec cette mise à jour, le problème a été corrigé et vous pouvez désormais attacher simultanément plusieurs périphériques hôtes à l'aide de la console web.

(JIRA:RHELPLAN-115603)

8.17. RÔLES DU SYSTÈME RED HAT ENTERPRISE LINUX

Le rôle network RHEL gère le paramètre ansible_managed dans les fichiers de configuration

Auparavant, le rôle Ansible n'était pas en mesure de fournir l'en-tête **ansible_managed** correct pour les fichiers de configuration gérés par le rôle **network**. En conséquence, les administrateurs système n'étaient pas certains des fichiers gérés par Ansible. Avec cette correction, les fichiers gérés par le rôle ont un en-tête **ansible_managed** correct, et les administrateurs système peuvent dire de manière fiable quels fichiers sont gérés par Ansible.

(BZ#2065382)

Correction d'une erreur de typographie pour la prise en charge de active-backup pour le mode de liaison correct

Auparavant, il y avait une faute de frappe,**active_backup**, dans la prise en charge du port InfiniBand tout en spécifiant le mode de liaison **active-backup**. En raison de cette faute de frappe, la connexion ne prenait pas en charge le mode de liaison correct pour le port de liaison InfiniBand. Cette mise à jour corrige la faute de frappe en remplaçant le mode de liaison par **active-backup**. La connexion prend désormais en charge avec succès le port de liaison InfiniBand.

(BZ#2065394)

La fonction IPRouteUtils.get_route_tables_mapping() accepte désormais toute séquence d'espaces blancs

Auparavant, l'analyseur de la base de données de la table de routage **iproute2**, telle que **/etc/iproute2/rt_tables**, affirmait que les entrées du fichier étaient de la forme **254 main** et qu'un seul caractère d'espacement séparait l'identifiant numérique et le nom. Par conséquent, l'analyseur ne parvenait pas à mettre en cache toutes les correspondances entre le nom de la table de routage et l'identifiant de la table, ce qui empêchait l'utilisateur d'ajouter une route statique dans la table de routage en définissant le nom de la table de routage. Avec cette mise à jour, l'analyseur accepte toute séquence d'espaces blancs entre l'identifiant et le nom de la table. Par conséquent, comme l'analyseur met en cache toutes les correspondances entre le nom de la table d'itinéraires et l'identifiant de la table, les utilisateurs peuvent ajouter un itinéraire statique dans la table d'itinéraires en définissant le nom de la table d'itinéraires.

(BZ#2115886)

Le paramètre forward_port accepte désormais les options string et dict

Auparavant, dans le rôle **firewall** RHEL System, le paramètre **forward_port** n'acceptait que l'option **string**. Cependant, la documentation du rôle affirmait que les deux options **string and dict** étaient prises en charge. Par conséquent, les utilisateurs qui lisaient et suivaient la documentation obtenaient une erreur. Ce bogue a été corrigé en faisant en sorte que **forward_port** accepte les deux options. Par conséquent, les utilisateurs peuvent suivre la documentation en toute sécurité pour configurer la redirection de port.

(BZ#2100605)

La configuration par le rôle metrics suit désormais correctement les liens symboliques

Lorsque le paquet **mssql pcp** est installé, le fichier **mssql.conf** est situé dans **/etc/pcp/mssql/** et est ciblé par le lien symbolique **/var/lib/pcp/pmdas/mssql/mssql.conf**. Auparavant, cependant, le rôle **metrics** écrasait le lien symbolique au lieu de le suivre et de configurer **mssql.conf**. Par conséquent, l'exécution du rôle **metrics** a transformé le lien symbolique en un fichier normal et la configuration n'a donc affecté que le fichier **/var/lib/pcp/pmdas/mssql/mssql.conf**. Le lien symbolique a donc échoué et le fichier de configuration principal **/etc/pcp/mssql/mssql.conf** n'a pas été affecté par la configuration.

Le problème est maintenant résolu et l'option **follow: yes** pour suivre le lien symbolique a été ajoutée au rôle **metrics**. En conséquence, le rôle **metrics** préserve les liens symboliques et configure correctement le fichier de configuration principal.

(BZ#2060523)

Le site kernel_settings configobj est disponible sur les hôtes gérés

Auparavant, le rôle **kernel_settings** n'installait pas le paquetage **python3-configobj** sur les hôtes gérés. Par conséquent, le rôle renvoyait une erreur indiquant que le module Python **configobj** était introuvable. Avec cette correction, le rôle s'assure que le paquetage **python3-configobj** est présent sur les hôtes gérés et le rôle **kernel_settings** fonctionne comme prévu.

(BZ#2060525)

Le paramètre mount_options pour les volumes est désormais valable pour un volume

Auparavant, le paramètre avait été accidentellement supprimé de la liste des paramètres valides pour un volume. Par conséquent, les utilisateurs ne pouvaient pas définir le paramètre **mount_options** pour les volumes. Avec cette correction de bogue, le paramètre **mount_options** a été réintroduit dans la liste des paramètres valides et le code a été remanié pour détecter les erreurs. Par conséquent, le rôle système **storage** RHEL peut définir le paramètre **mount_options** pour les volumes.

(BZ#2083376)

Le rôle système storage RHEL prend désormais correctement en charge les niveaux striped et raid0 pour les volumes LVM

Le rôle système **storage** RHEL signalait auparavant de manière incorrecte que les niveaux RAID **striped** et **raid0** n'étaient pas pris en charge pour les volumes LVM. Ce problème est désormais corrigé et le rôle peut maintenant créer correctement des volumes LVM de tous les niveaux RAID pris en charge par LVM : **raid0**, **raid1**, **raid4**, **raid5**, **raid6**, **raid10**, **striped** et **mirror**.

(BZ#2083410)

Le README et la documentation de metrics RHEL System Role spécifient désormais clairement les versions de Redis et de Grafana prises en charge sur des versions spécifiques de RHEL par le rôle

Auparavant, lorsque l'on essayait d'utiliser le rôle **metrics** avec des versions non prises en charge de Redis et de Grafana sur des plates-formes non prises en charge, le rôle échouait. Cette mise à jour clarifie la documentation concernant les versions de Redis et de Grafana prises en charge par le rôle sur les différentes versions de RHEL. Par conséquent, vous pouvez éviter d'essayer d'utiliser des versions non prises en charge de Redis et de Grafana sur des plates-formes non prises en charge.

(BZ#2100286)

Option de longueur minimale des bits de la clé RSA dans les rôles de système RHEL ssh et sshd

L'utilisation accidentelle de clés RSA courtes peut rendre le système plus vulnérable aux attaques. Avec cette mise à jour, vous pouvez définir la longueur minimale des bits des clés RSA pour les clients et les serveurs OpenSSH en utilisant l'option **RequiredRSASize** dans les rôles système RHEL **ssh** et **sshd**.

(BZ#2109998)

Le rôle système nbde_client RHEL utilise désormais des espaces appropriés lorsqu'il spécifie des paramètres de ligne de commande Dracut supplémentaires

Le cadre Dracut exige un espacement approprié lors de la spécification de paramètres supplémentaires, tels que les paramètres de la ligne de commande du noyau. Si les paramètres ne sont pas spécifiés avec un espacement approprié, Dracut peut ne pas ajouter les paramètres supplémentaires spécifiés à la ligne de commande du noyau. Avec cette mise à jour, le rôle système **nbde_client** RHEL utilise un espacement approprié lors de la création de fichiers de configuration Dracut supplémentaires. Par conséquent, le rôle définit correctement les paramètres de ligne de commande de Dracut.

(BZ#2115156)

Le site tlog RHEL System Roles est désormais correctement superposé à SSSD

Auparavant, le rôle de système RHEL **tlog** s'appuyait sur le fournisseur de fichiers System Security Services Daemon (SSSD) et sur l'option **authselect** activée **with-files-domain** pour configurer les entrées **passwd** correctes dans le fichier **nsswitch.conf**. Dans RHEL 9.0, SSSD n'a pas implicitement activé le fournisseur de fichiers par défaut et, par conséquent, la superposition du shell **tlog-rec-session** par SSSD n'a pas fonctionné. Avec cette correction, le rôle **tlog** met désormais à jour le rôle **nsswitch.conf** pour s'assurer que **tlog-rec-session** est correctement superposé par SSSD.

(BZ#2071804)

Le rôle de système metrics RHEL redémarre automatiquement les services pmie et pmlogger après une mise à jour de leur configuration

Auparavant, les services **pmie** et **pmlogger** ne redémarraient pas après une modification de leur configuration et attendaient l'exécution du gestionnaire. Cela provoquait des erreurs avec d'autres services **metrics**, qui nécessitaient la configuration de **pmie** et **pmlogger** pour correspondre à leur comportement d'exécution. Avec cette mise à jour, le rôle redémarre **pmie** et **pmlogger** immédiatement après une mise à jour de la configuration, leur configuration correspond au comportement d'exécution des services de métriques dépendants et ils fonctionnent correctement.

(BZ#2100294)

8.18. VIRTUALISATION

Les performances du trafic réseau dans les machines virtuelles ne sont plus réduites en cas de forte charge

Auparavant, les machines virtuelles RHEL avaient, dans certains cas, des performances réduites lorsqu'elles géraient des niveaux élevés de trafic réseau. Le code sous-jacent a été corrigé et les performances du trafic réseau fonctionnent désormais comme prévu dans les circonstances décrites.

(BZ#1945040)

8.19. RHEL DANS LES ENVIRONNEMENTS EN NUAGE

La fonctionnalité SR-IOV d'une carte réseau attachée à une VM Hyper-V fonctionne désormais de manière fiable

Auparavant, lors de l'attachement d'un adaptateur réseau avec la virtualisation d'E/S à racine unique (SR-IOV) activée à une machine virtuelle (VM) RHEL 9 fonctionnant sur l'hyperviseur Microsoft Hyper-V, la fonctionnalité SR-IOV ne fonctionnait pas correctement dans certains cas. Un bogue dans le code d'allocation d'E/S en mémoire (MMIO) spécifique à Hyper-V a été corrigé et la fonctionnalité SR-IOV fonctionne désormais comme prévu sur les VM Hyper-V.

(BZ#2030922)

SR-IOV ne fonctionne plus de manière sous-optimale dans les machines virtuelles ARM 64 RHEL 9 sur Azure

Auparavant, les périphériques de mise en réseau SR-IOV présentaient un débit nettement plus faible et une latence plus élevée que prévu dans les machines virtuelles (VM) ARM 64 RHEL 9 fonctionnant sur une plateforme Microsoft Azure. Le problème a été corrigé et les machines virtuelles concernées fonctionnent désormais comme prévu.

(BZ#2068432)

8.20. CONTENEURS

podman system connection add et **podman image scp** n'échoue plus

Podman utilise des hachages SHA-1 pour l'échange de clés RSA. Auparavant, la connexion SSH normale entre les machines utilisant des clés RSA fonctionnait, tandis que les commandes **podman system connection add** et **podman image scp** ne fonctionnaient pas avec les mêmes clés RSA, car les hachages SHA-1 n'étaient pas acceptés pour l'échange de clés sur RHEL 9. Avec la mise à jour, le problème a été corrigé.

(JIRA:RHELPLAN-121180)

Les images de conteneurs signées avec une clé Beta GPG peuvent désormais être extraites

Auparavant, lorsque vous retiriez des images de conteneurs RHEL Beta, Podman échouait avec le message d'erreur : **Error: Source image rejected: None of the signatures were accepted**. Les images ne pouvaient pas être extraites car les versions actuelles étaient configurées pour ne pas faire confiance aux clés GPG de RHEL Beta par défaut. Avec cette mise à jour, le fichier **/etc/containers/policy.json** supporte un nouveau champ **keyPaths** qui accepte une liste de fichiers contenant les clés de confiance. De ce fait, les images de conteneurs signées avec les clés GPG GA et Beta sont maintenant acceptées dans la configuration par défaut.

(BZ#2094015)

Podman n'échoue plus à extraire un conteneur "X509 : certificat signé par une autorité inconnue"

Auparavant, si vous aviez votre propre registre interne signé par notre propre certificat d'autorité de certification, vous deviez importer le certificat sur votre machine hôte. Sinon, une erreur se produisait :

```
x509: certificate signed by unknown authority
```

Avec cette mise à jour, le problème a été corrigé.

(BZ#2027576)

DNF et YUM n'échouent plus à cause d'ID de référentiels non concordants

Auparavant, les ID de référentiel DNF et YUM ne correspondaient pas au format attendu par DNF ou YUM. Par exemple, si vous exécutez l'exemple suivant, l'erreur se produit :

```
# podman run -ti ubi8-ubi
# dnf debuginfo-install dnsmasq
...
This system is not registered with an entitlement server. You can use subscription-manager to
register.
```

–

Avec cette mise à jour, le problème a été corrigé. Le suffixe **--debug-rpms** a été ajouté à tous les noms de dépôts de débogage (par exemple **ubi-8-appstream-debug-rpms**), et le suffixe **-rpms** a été ajouté à tous les noms de dépôts UBI (par exemple **ubi-8-appstream-rpms**).

Pour plus d'informations, voir Images de base universelles (UBI) : Images, dépôts, paquets et code source.

(BZ#2120378)

CHAPITRE 9. APERÇUS TECHNOLOGIQUES

Cette partie fournit une liste de tous les aperçus technologiques disponibles dans Red Hat Enterprise Linux 9.

Pour plus d'informations sur l'étendue de l'assistance de Red Hat pour les fonctionnalités de l'aperçu technologique, voir l'étendue de l'assistance pour les fonctionnalités de l'aperçu technologique .

9.1. SHELLS ET OUTILS DE LIGNE DE COMMANDE

ReaR disponible sur l'architecture IBM Z 64 bits en tant qu'aperçu technologique

La fonctionnalité de base Relax and Recover (ReaR) est désormais disponible sur l'architecture IBM Z 64 bits en tant qu'aperçu technologique. Vous pouvez créer une image de secours ReaR sur IBM Z uniquement dans l'environnement z/VM. La sauvegarde et la récupération des partitions logiques (LPAR) n'ont pas été testées.

La seule méthode de sortie actuellement disponible est l'Initial Program Load (IPL). L'IPL produit un noyau et un ramdisk initial (initrd) qui peut être utilisé avec le bootloader **zIPL**.

AVERTISSEMENT

Actuellement, le processus de récupération reformate tous les DASD (Direct Attached Storage Devices) connectés au système. Ne tentez pas de récupérer le système si des données de valeur se trouvent sur les périphériques de stockage du système. Cela inclut également le périphérique préparé avec le chargeur de démarrage **zIPL**, le noyau ReaR et l'initrd qui ont été utilisés pour démarrer dans l'environnement de secours. Veillez à en conserver une copie.

Pour plus d'informations, voir Utilisation d'une image de secours ReaR sur l'architecture IBM Z 64 bits .

(BZ#2046653)

GIMP disponible en avant-première technologique dans RHEL 9

GNU Image Manipulation Program (GIMP) 2.99.8 est maintenant disponible dans RHEL 9 en tant qu'aperçu technologique. La version 2.99.8 du paquet **gimp** est une pré-version avec un ensemble d'améliorations, mais un ensemble limité de fonctionnalités et aucune garantie de stabilité. Dès que la version officielle de GIMP 3 sera publiée, elle sera introduite dans RHEL 9 en tant que mise à jour de cette version préliminaire.

Dans RHEL 9, vous pouvez facilement installer **gimp** en tant que paquetage RPM.

(BZ#2047161)

9.2. SÉCURITÉ

gnutls utilise désormais KTLS en tant qu'avant-première technologique

Les paquets **gnutls** mis à jour peuvent utiliser Kernel TLS (KTLS) pour accélérer le transfert de données

sur des canaux cryptés en tant qu'aperçu technologique. Pour activer KTLS, ajoutez le module de noyau **tls.ko** à l'aide de la commande **modprobe**, et créez un nouveau fichier de configuration **/etc/crypto-policies/local.d/gnutls-ktls.txt** pour les politiques cryptographiques du système avec le contenu suivant :

```
[global]
ktls = true
```

Notez que la version actuelle ne prend pas en charge la mise à jour des clés de trafic par le biais des messages TLS **KeyUpdate**, ce qui a une incidence sur la sécurité des suites de chiffrement AES-GCM. Voir le document RFC 7841 - TLS 1.3 pour plus d'informations.

(BZ#2042009)

9.3. MISE EN RÉSEAU

WireGuard VPN est disponible en avant-première technologique

WireGuard, que Red Hat fournit en tant qu'aperçu technologique non pris en charge, est une solution VPN de haute performance qui fonctionne dans le noyau Linux. Elle utilise une cryptographie moderne et est plus facile à configurer que d'autres solutions VPN. En outre, la petite base de code de WireGuard réduit la surface d'attaque et, par conséquent, améliore la sécurité.

Pour plus de détails, voir Configuration d'un VPN WireGuard .

(BZ#1613522)

Configurer le TCP Multipath avec NetworkManager est disponible en avant-première technologique

Avec cette mise à jour, l'utilitaire NetworkManager vous offre la fonctionnalité Multipath TCP (MPTCP). Vous pouvez utiliser les commandes **nmcli** pour contrôler MPTCP et rendre ses paramètres persistants.

Pour plus d'informations, voir Understanding Multipath TCP : High availability for endpoints and the networking highway of the future et RFC 8684 : TCP Extensions for Multipath Operation with Multiple Addresses.

(BZ#2029636)

KTLS disponible en avant-première technologique

RHEL fournit Kernel Transport Layer Security (KTLS) en tant qu'aperçu technologique. KTLS traite les enregistrements TLS à l'aide des algorithmes de chiffrement ou de déchiffrement symétriques du noyau pour le chiffrement AES-GCM. KTLS inclut également l'interface permettant de décharger le chiffrement des enregistrements TLS sur les contrôleurs d'interface réseau (NIC) qui fournissent cette fonctionnalité.

(BZ#1570255)

Le service systemd-resolved est disponible en tant qu'aperçu technologique

Le service **systemd-resolved** fournit la résolution de noms aux applications locales. Le service met en œuvre un résolveur de stub DNS avec mise en cache et validation, un résolveur et un répondeur de DNS multidiffusion (Link-Local Multicast Name Resolution - LLMNR) et de DNS multidiffusion.

Notez que **systemd-resolved** est un aperçu technologique non pris en charge.

(BZ#2020529)

9.4. NOYAU

Le pilote Intel data streaming accelerator pour le noyau est disponible en tant qu'aperçu technologique

Le pilote de l'accélérateur de flux de données Intel (IDXD) pour le noyau est actuellement disponible en tant qu'aperçu technologique. Il s'agit d'un accélérateur intégré au processeur Intel qui comprend la file d'attente partagée avec la soumission de l'espace d'adressage du processus (pasid) et la mémoire virtuelle partagée (SVM).

(BZ#2030412)

SGX disponible en avant-première technologique

Software Guard Extensions(SGX) est une technologie Intel® destinée à protéger le code et les données des logiciels contre la divulgation et la modification. Le noyau RHEL fournit partiellement les fonctionnalités SGX v1 et v1.5. La version 1 permet aux plateformes utilisant le mécanisme **Flexible Launch Control** d'utiliser la technologie SGX.

(BZ#1874182)

Le pilote Soft-iWARP est disponible en tant qu'aperçu technologique

Soft-iWARP (siw) est un logiciel, Internet Wide-area RDMA Protocol (iWARP), pilote de noyau pour Linux. Soft-iWARP met en œuvre la suite de protocoles iWARP sur la pile réseau TCP/IP. Cette suite de protocoles est entièrement mise en œuvre dans le logiciel et ne nécessite pas de matériel RDMA (Remote Direct Memory Access) spécifique. Soft-iWARP permet à un système doté d'un adaptateur Ethernet standard de se connecter à un adaptateur iWARP ou à un autre système sur lequel Soft-iWARP est déjà installé.

(BZ#2023416)

9.5. SYSTÈMES DE FICHIERS ET STOCKAGE

DAX est maintenant disponible pour ext4 et XFS en tant qu'aperçu technologique

Dans RHEL 9, le système de fichiers DAX est disponible en tant qu'aperçu technologique. DAX permet à une application de mapper directement la mémoire persistante dans son espace d'adressage. Pour utiliser DAX, un système doit disposer d'une certaine forme de mémoire persistante, généralement sous la forme d'un ou plusieurs modules de mémoire double en ligne non volatile (NVDIMM), et un système de fichiers compatible DAX doit être créé sur le(s) module(s) NVDIMM. Le système de fichiers doit également être monté avec l'option de montage **dax**. Ensuite, **mmap** d'un fichier sur le système de fichiers monté sur dax entraîne un mappage direct du stockage dans l'espace d'adressage de l'application.

(BZ#1995338)

Stratis est disponible en avant-première technologique

Stratis est un gestionnaire de stockage local. Il fournit des systèmes de fichiers gérés au-dessus des pools de stockage avec des fonctionnalités supplémentaires pour l'utilisateur :

- Gérer les snapshots et le thin provisioning
- Augmentation automatique de la taille du système de fichiers en fonction des besoins

- Maintenir les systèmes de fichiers

Pour administrer le stockage Stratis, utilisez l'utilitaire **stratis**, qui communique avec le service d'arrière-plan **stratisd**.

Stratis est fourni en tant qu'aperçu technologique.

Pour plus d'informations, voir la documentation Stratis : Configuration des systèmes de fichiers Stratis .

(BZ#2041558)

Fonctionnalités du service de découverte NVMe-oF disponibles en tant qu'aperçu technologique

Les fonctions du service de découverte NVMe-oF, définies dans les propositions techniques (TP) 8013 et 8014 de NVMexpress.org, sont disponibles en tant qu'aperçu technologique. Pour obtenir un aperçu de ces fonctionnalités, utilisez le paquetage **nvme-cli 2.0** et attachez l'hôte à un périphérique cible NVMe-oF qui implémente le TP-8013 ou le TP-8014. Pour plus d'informations sur les TP-8013 et TP-8014, voir les TPs NVM Express 2.0 Ratifiés sur le site web https://nvmexpress.org/developers/nvme-specification/.

(BZ#2021672)

nvme-stas disponible en avant-première technologique

Le paquetage **nvme-stas**, qui est un client Central Discovery Controller (CDC) pour Linux, est maintenant disponible en tant qu'aperçu technologique. Il gère les notifications d'événements asynchrones (AEN), les contrôles automatisés des connexions au sous-système NVMe, la gestion et le signalement des erreurs, ainsi que la configuration automatique (**zeroconf**) et manuelle.

Ce paquetage se compose de deux démons, Storage Appliance Finder (**stafd**) et Storage Appliance Connector (**stacd**).

(BZ#1893841)

9.6. COMPILATEURS ET OUTILS DE DÉVELOPPEMENT

jmc-core et owasp-java-encoder disponible en avant-première technologique

RHEL 9 est distribué avec les paquets **jmc-core** et **owasp-java-encoder** en tant que fonctionnalités d'aperçu technologique.

jmc-core est une bibliothèque fournissant des API de base pour le contrôle de mission du kit de développement Java (JDK), y compris des bibliothèques pour l'analyse et l'écriture de fichiers d'enregistrement de vol JDK, ainsi que des bibliothèques pour la découverte de la machine virtuelle Java (JVM) par le biais du protocole de découverte Java (JDP).

Le paquetage **owasp-java-encoder** fournit une collection d'encodeurs contextuels à haute performance et à faible surcharge pour Java.

(BZ#1980981)

9.7. GESTION DE L'IDENTITÉ

DNSSEC disponible en tant qu'aperçu technologique dans IdM

Les serveurs de gestion de l'identité (IdM) avec DNS intégré mettent désormais en œuvre les

extensions de sécurité DNS (DNSSEC), un ensemble d'extensions du DNS qui renforcent la sécurité du protocole DNS. Les zones DNS hébergées sur les serveurs IdM peuvent être automatiquement signées à l'aide de DNSSEC. Les clés cryptographiques sont générées automatiquement et font l'objet d'une rotation.

Il est conseillé aux utilisateurs qui décident de sécuriser leurs zones DNS avec DNSSEC de lire et de suivre ces documents :

- Pratiques opérationnelles DNSSEC, version 2
- Guide de déploiement du système de noms de domaine sécurisé (DNS)
- Considérations sur le calendrier de renouvellement des clés DNSSEC

Notez que les serveurs IdM avec DNS intégré utilisent DNSSEC pour valider les réponses DNS obtenues d'autres serveurs DNS. Cela peut affecter la disponibilité des zones DNS qui ne sont pas configurées conformément aux pratiques recommandées en matière de dénomination.

(BZ#2084180)

L'API JSON-RPC de gestion des identités est disponible en avant-première technologique

Une API est disponible pour la gestion des identités (IdM). Pour visualiser l'API, IdM fournit également un navigateur API en tant qu'aperçu technologique.

Auparavant, l'API IdM était améliorée pour permettre plusieurs versions des commandes de l'API. Ces améliorations pouvaient modifier le comportement d'une commande de manière incompatible. Les utilisateurs peuvent désormais continuer à utiliser les outils et les scripts existants même si l'API IdM change. Cela permet :

- Aux administrateurs d'utiliser des versions antérieures ou postérieures d'IdM sur le serveur par rapport au client de gestion.
- Les développeurs peuvent utiliser une version spécifique d'un appel IdM, même si la version IdM change sur le serveur.

Dans tous les cas, la communication avec le serveur est possible, même si l'une des parties utilise, par exemple, une version plus récente qui introduit de nouvelles options pour une fonctionnalité.

Pour plus de détails sur l'utilisation de l'API, voir Utilisation de l'API de gestion des identités pour communiquer avec le serveur IdM (AVANT-PROPOS TECHNOLOGIQUE).

(BZ#2084166)

RHEL IdM permet de déléguer l'authentification des utilisateurs à des fournisseurs d'identité externes en tant qu'aperçu technologique

Dans RHEL IdM, vous pouvez désormais associer des utilisateurs à des fournisseurs d'identité externes (IdP) qui prennent en charge le flux d'autorisation de périphérique OAuth 2. Lorsque ces utilisateurs s'authentifient avec la version SSSD disponible dans RHEL 9.1, ils bénéficient des fonctionnalités d'authentification unique de RHEL IdM avec des tickets Kerberos après avoir effectué l'authentification et l'autorisation auprès du fournisseur d'identité externe.

Parmi les caractéristiques notables, on peut citer

- Ajout, modification et suppression de références à des IdP externes à l'aide des commandes **ipa idp-***

- Activation de l'authentification IdP pour les utilisateurs avec la commande **ipa user-mod --user-auth-type=idp**

Pour plus d'informations, voir Utilisation de fournisseurs d'identité externes pour s'authentifier auprès de l'IdM.

(BZ#2069202)

le sous-paquet sssd-idp est disponible en tant qu'aperçu technologique

Le sous-paquet **sssd-idp** pour SSSD contient les plugins **oidc_child** et krb5 **idp**, qui sont des composants côté client qui effectuent l'authentification OAuth2 contre les serveurs de gestion d'identité (IdM). Cette fonctionnalité n'est disponible qu'avec les serveurs IdM sur RHEL 8.7 et plus, et RHEL 9.1 et plus.

(BZ#2065693)

Le plugin idp krb5 interne de SSSD est disponible en tant qu'aperçu technologique

Le plugin SSSD krb5 **idp** vous permet de vous authentifier auprès d'un fournisseur d'identité externe (IdP) à l'aide du protocole OAuth2. Cette fonctionnalité n'est disponible qu'avec les serveurs IdM sur RHEL 8.7 et plus, et RHEL 9.1 et plus.

(BZ#2056482)

ACME disponible en avant-première technologique

Le service Automated Certificate Management Environment (ACME) est désormais disponible dans Identity Management (IdM) en tant qu'aperçu technologique. ACME est un protocole de validation automatisée des identifiants et d'émission de certificats. Son objectif est d'améliorer la sécurité en réduisant la durée de vie des certificats et en évitant les processus manuels de gestion du cycle de vie des certificats.

Dans RHEL, le service ACME utilise le répondeur ACME PKI de Red Hat Certificate System (RHCS). Le sous-système RHCS ACME est automatiquement déployé sur chaque serveur d'autorité de certification (CA) dans le déploiement IdM, mais il ne traite pas les demandes jusqu'à ce que l'administrateur l'active. RHCS utilise le profil **acmeIPAServerCert** lors de l'émission de certificats ACME. La période de validité des certificats émis est de 90 jours. L'activation ou la désactivation du service ACME affecte l'ensemble du déploiement IdM.

IMPORTANT

Il est recommandé d'activer ACME uniquement dans un déploiement IdM où tous les serveurs utilisent RHEL 8.4 ou une version ultérieure. Les versions antérieures de RHEL n'incluent pas le service ACME, ce qui peut entraîner des problèmes dans les déploiements de versions mixtes. Par exemple, un serveur CA sans ACME peut faire échouer les connexions des clients, car il utilise un Subject Alternative Name (SAN) DNS différent.

AVERTISSEMENT

Actuellement, le RHCS ne supprime pas les certificats expirés. Comme les certificats ACME expirent après 90 jours, les certificats expirés peuvent s'accumuler, ce qui peut affecter les performances.

- Pour activer l'ACME dans l'ensemble du déploiement IdM, utilisez la commande **ipa-acme-manage enable**:

```
# ipa-acme-manage enable
The ipa-acme-manage command was successful
```

- Pour désactiver ACME dans l'ensemble du déploiement IdM, utilisez la commande **ipa-acme-manage disable**:

```
# ipa-acme-manage disable
The ipa-acme-manage command was successful
```

- Pour vérifier si le service ACME est installé et s'il est activé ou désactivé, utilisez la commande **ipa-acme-manage status**:

```
# ipa-acme-manage status
ACME is enabled
The ipa-acme-manage command was successful
```

(BZ#2084181)

9.8. BUREAU

GNOME pour l'architecture ARM 64 bits disponible en tant qu'aperçu technologique

L'environnement de bureau GNOME est disponible pour l'architecture ARM 64 bits en tant qu'aperçu technologique.

Vous pouvez désormais vous connecter à la session de bureau d'un serveur ARM 64 bits à l'aide de VNC. Vous pouvez ainsi gérer le serveur à l'aide d'applications graphiques.

Un ensemble limité d'applications graphiques est disponible sur ARM 64 bits. Par exemple :

- Le navigateur web Firefox
- Red Hat Subscription Manager (**subscription-manager-cockpit**)
- Configuration du pare-feu (**firewall-config**)
- Analyseur d'utilisation du disque (**baobab**)

Avec Firefox, vous pouvez vous connecter au service Cockpit sur le serveur.

Certaines applications, comme LibreOffice, ne fournissent qu'une interface en ligne de commande, et leur interface graphique est désactivée.

(JIRA:RHELPLAN-27394)

GNOME pour l'architecture IBM Z disponible en avant-première technologique

L'environnement de bureau GNOME est disponible pour l'architecture IBM Z en tant qu'aperçu technologique.

Vous pouvez désormais vous connecter à la session de bureau d'un serveur IBM Z à l'aide de VNC. Vous pouvez ainsi gérer le serveur à l'aide d'applications graphiques.

Un ensemble limité d'applications graphiques est disponible sur IBM Z. Par exemple :

- Le navigateur web Firefox
- Red Hat Subscription Manager (**subscription-manager-cockpit**)
- Configuration du pare-feu (**firewall-config**)
- Analyseur d'utilisation du disque (**baobab**)

Avec Firefox, vous pouvez vous connecter au service Cockpit sur le serveur.

Certaines applications, comme LibreOffice, ne fournissent qu'une interface en ligne de commande, et leur interface graphique est désactivée.

(JIRA:RHELPLAN-27737)

9.9. LA CONSOLE WEB

Stratis disponible en tant qu'aperçu technologique dans la console web RHEL

Avec cette mise à jour, la console web de Red Hat Enterprise Linux permet de gérer le stockage Stratis en tant qu'aperçu technologique.

Pour en savoir plus sur Stratis, voir Qu'est-ce que Stratis ?

(JIRA:RHELPLAN-122345)

9.10. VIRTUALISATION

Les machines virtuelles RHEL peuvent désormais être déployées sur des instances VMware ESXi fonctionnant sur des processeurs ARM64

En tant qu'aperçu technologique, il est désormais possible de déployer des machines virtuelles RHEL sur des instances d'hyperviseur VMware ESXi fonctionnant sur des processeurs ARM 64 bits.

(JIRA:RHELPLAN-95456)

AMD SEV et SEV-ES pour les machines virtuelles KVM

En tant qu'aperçu technologique, RHEL 9 fournit la fonction Secure Encrypted Virtualization (SEV) pour les machines hôtes AMD EPYC qui utilisent l'hyperviseur KVM. Si elle est activée sur une machine virtuelle (VM), SEV crypte la mémoire de la VM pour la protéger contre l'accès de l'hôte. La sécurité de la VM s'en trouve renforcée.

En outre, la version améliorée Encrypted State de SEV (SEV-ES) est également fournie en tant qu'aperçu technologique. SEV-ES crypte tous les contenus des registres de l'unité centrale lorsqu'une machine virtuelle cesse de fonctionner. Cela empêche l'hôte de modifier les registres de l'unité centrale de la machine virtuelle ou de lire les informations qu'ils contiennent.

Notez que SEV et SEV-ES ne fonctionnent que sur la deuxième génération de processeurs AMD EPYC (nom de code Rome) ou plus récents. Notez également que RHEL 9 inclut le chiffrement SEV et SEV-ES, mais pas l'attestation de sécurité SEV et SEV-ES.

(JIRA:RHELPLAN-65217)

La virtualisation est désormais disponible sur ARM 64

En tant qu'aperçu technologique, il est désormais possible de créer des machines virtuelles KVM sur des systèmes utilisant des processeurs ARM 64.

(JIRA:RHELPLAN-103993)

virtio-mem est désormais disponible sur AMD64, Intel 64 et ARM 64

En tant qu'aperçu technologique, RHEL 9 introduit la fonctionnalité **virtio-mem** sur les systèmes AMD64, Intel 64 et ARM 64. L'utilisation de **virtio-mem** permet d'ajouter ou de supprimer dynamiquement de la mémoire hôte dans les machines virtuelles (VM).

Pour utiliser **virtio-mem**, définissez les périphériques de mémoire **virtio-mem** dans la configuration XML d'une VM et utilisez la commande **virsh update-memory-device** pour demander des modifications de la taille des périphériques de mémoire lorsque la VM est en cours d'exécution. Pour connaître la taille actuelle de la mémoire exposée par ces dispositifs de mémoire à une VM en cours d'exécution, consultez la configuration XML de la VM.

(BZ#2014487, BZ#2044162, BZ#2044172)

9.11. RHEL DANS LES ENVIRONNEMENTS EN NUAGE

Les VM confidentielles RHEL sont désormais disponibles sur Azure en tant qu'aperçu technologique

Avec le noyau RHEL mis à jour, vous pouvez désormais créer et exécuter des machines virtuelles (VM) confidentielles sur Microsoft Azure en tant qu'aperçu technologique. Toutefois, il n'est pas encore possible de chiffrer les images de machines virtuelles confidentielles RHEL pendant le démarrage sur Azure.

(JIRA:RHELPLAN-122321)

9.12. CONTENEURS

La possibilité d'utiliser plusieurs clés GPG de confiance pour la signature des images est disponible en tant qu'aperçu technologique

Le fichier **/etc/containers/policy.json** prend en charge un nouveau champ **keyPaths** qui accepte une liste de fichiers contenant les clés de confiance. De ce fait, les images de conteneurs signées avec les clés GPG GA et Beta sont désormais acceptées dans la configuration par défaut.

Par exemple :

```
"registry.redhat.io": [
```

```
        {
            "type": "signedBy",
            "keyType": "GPGKeys",
            "keyPaths": ["/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release", "/etc/pki/rpm-gpg/RPM-GPG-
KEY-redhat-beta"]
        }
]
```

(JIRA:RHELPLAN-129327)

Les signatures sigstore sont désormais disponibles en avant-première technologique

À partir de Podman 4.2, vous pouvez utiliser le format sigstore pour les signatures d'images de conteneurs. Les signatures sigstore sont stockées dans le registre des conteneurs avec l'image du conteneur, sans qu'il soit nécessaire d'avoir un serveur de signatures distinct pour stocker les signatures des images.

(JIRA:RHELPLAN-74672)

CHAPITRE 10. FONCTIONNALITÉ OBSOLÈTE

Cette partie fournit une vue d'ensemble des fonctionnalités qui ont été *deprecated* dans Red Hat Enterprise Linux 9.

Les fonctionnalités obsolètes ne seront probablement plus prises en charge dans les prochaines versions majeures de ce produit et ne sont pas recommandées pour les nouveaux déploiements. Pour obtenir la liste la plus récente des fonctionnalités obsolètes dans une version majeure particulière, consultez la dernière version de la documentation.

Le statut de prise en charge des fonctionnalités dépréciées reste inchangé dans Red Hat Enterprise Linux 9. Pour plus d'informations sur la durée de la prise en charge, voir Red Hat Enterprise Linux Life Cycle et Red Hat Enterprise Linux Application Streams Life Cycle .

Les composants matériels obsolètes ne sont pas recommandés pour les nouveaux déploiements sur les versions majeures actuelles ou futures. Les mises à jour des pilotes de matériel sont limitées aux correctifs de sécurité et aux correctifs critiques. Red Hat recommande de remplacer ce matériel dès que possible.

Un paquet peut être déprécié et son utilisation déconseillée. Dans certaines circonstances, un paquetage peut être retiré d'un produit. La documentation du produit identifie alors les paquets plus récents qui offrent des fonctionnalités similaires, identiques ou plus avancées que celles du paquet supprimé, et fournit d'autres recommandations.

Pour plus d'informations sur les fonctionnalités présentes dans RHEL 8, mais qui ont été *removed* dans RHEL 9, voir les considérations relatives à l'adoption de RHEL 9 .

10.1. CRÉATION D'INSTALLATEURS ET D'IMAGES

Commandes Kickstart obsolètes

Les commandes Kickstart suivantes sont obsolètes :

- **timezone --ntpservers**
- **timezone --nontp**
- **logging --level**
- **%packages --excludeWeakdeps**
- **%packages --instLangs**
- **aconda**
- **pwpolicy**

Notez que lorsque seules des options spécifiques sont listées, la commande de base et ses autres options sont toujours disponibles et ne sont pas dépréciées. L'utilisation des commandes obsolètes dans les fichiers Kickstart entraîne l'affichage d'un avertissement dans les journaux. Vous pouvez transformer les avertissements des commandes obsolètes en erreurs avec l'option **inst.ksstrict** boot.

(BZ#1899167)

10.2. SÉCURITÉ

SHA-1 est déprécié à des fins cryptographiques

L'utilisation du condensé de message SHA-1 à des fins cryptographiques a été supprimée dans RHEL 9. Le condensé produit par SHA-1 n'est pas considéré comme sûr en raison des nombreuses attaques réussies documentées basées sur la recherche de collisions de hachage. Les composants cryptographiques de base de RHEL ne créent plus de signatures à l'aide de SHA-1 par défaut. Les applications de RHEL 9 ont été mises à jour pour éviter d'utiliser SHA-1 dans les cas d'utilisation liés à la sécurité.

Parmi les exceptions, le code d'authentification des messages HMAC-SHA1 et les valeurs UUID (Universal Unique Identifier) peuvent encore être créés à l'aide de SHA-1, car ces cas d'utilisation ne présentent actuellement aucun risque pour la sécurité. SHA-1 peut également être utilisé dans des cas limités liés à d'importants problèmes d'interopérabilité et de compatibilité, tels que Kerberos et WPA-2. Pour plus de détails, consultez la section Liste des applications RHEL utilisant une cryptographie non conforme à la norme FIPS 140-3 dans le document de renforcement de la sécurité de RHEL 9 .

Si votre scénario nécessite l'utilisation de SHA-1 pour la vérification des signatures cryptographiques existantes ou de tiers, vous pouvez l'activer en entrant la commande suivante :

```
# update-crypto-policies --set DEFAULT:SHA1
```

Vous pouvez également basculer les stratégies cryptographiques du système vers la stratégie **LEGACY**. Notez que **LEGACY** active également de nombreux autres algorithmes qui ne sont pas sécurisés.

(JIRA:RHELPLAN-110763)

SCP est obsolète dans RHEL 9

Le protocole de copie sécurisée (SCP) est obsolète car il présente des failles de sécurité connues. L'API SCP reste disponible pour le cycle de vie de RHEL 9, mais son utilisation réduit la sécurité du système.

- Dans l'utilitaire **scp**, SCP est remplacé par défaut par le protocole de transfert de fichiers SSH (SFTP).
- La suite OpenSSH n'utilise pas SCP dans RHEL 9.
- SCP est obsolète dans la bibliothèque **libssh**.

(JIRA:RHELPLAN-99136)

Digest-MD5 dans SASL est obsolète

Le mécanisme d'authentification Digest-MD5 du cadre Simple Authentication Security Layer (SASL) est obsolète et pourrait être supprimé des paquets **cyrus-sasl** dans une prochaine version majeure.

(BZ#1995600)

OpenSSL supprime MD2, MD4, MDC2, Whirlpool, RIPEMD160, Blowfish, CAST, DES, IDEA, RC2, RC4, RC5, SEED et PBKDF1

Le projet OpenSSL a déprécié un ensemble d'algorithmes cryptographiques parce qu'ils ne sont pas sûrs, qu'ils sont peu utilisés, ou les deux. Red Hat déconseille également l'utilisation de ces algorithmes, et RHEL 9 les fournit pour migrer les données chiffrées afin d'utiliser de nouveaux algorithmes. Les utilisateurs ne doivent pas dépendre de ces algorithmes pour la sécurité de leurs systèmes.

Les implémentations des algorithmes suivants ont été déplacées vers l'ancien fournisseur d'OpenSSL : MD2, MD4, MDC2, Whirlpool, RIPEMD160, Blowfish, CAST, DES, IDEA, RC2, RC4, RC5, SEED et PBKDF1.

Consultez le fichier de configuration **/etc/pki/tls/openssl.cnf** pour savoir comment charger l'ancien fournisseur et activer la prise en charge des algorithmes obsolètes.

(BZ#1975836)

/etc/system-fips est désormais obsolète

La prise en charge de l'indication du mode FIPS par le fichier **/etc/system-fips** a été supprimée et le fichier ne sera pas inclus dans les versions futures de RHEL. Pour installer RHEL en mode FIPS, ajoutez le paramètre **fips=1** à la ligne de commande du noyau lors de l'installation du système. Vous pouvez vérifier si RHEL fonctionne en mode FIPS à l'aide de la commande **fips-mode-setup --check**.

(JIRA:RHELPLAN-103232)

libcrypt.so.1 est désormais obsolète

La bibliothèque **libcrypt.so.1** est désormais obsolète et pourrait être supprimée dans une prochaine version de RHEL.

(BZ#2034569)

fapolicyd.rules est obsolète

Le répertoire **/etc/fapolicyd/rules.d/**, qui contient les fichiers contenant les règles d'exécution d'autorisation et de refus, remplace le fichier **/etc/fapolicyd/fapolicyd.rules**. Le script **fagenrules** fusionne désormais tous les fichiers de règles de ce répertoire dans le fichier **/etc/fapolicyd/compiled.rules**. Les règles contenues dans **/etc/fapolicyd/fapolicyd.trust** sont toujours traitées par le cadre **fapolicyd**, mais uniquement dans un souci de compatibilité ascendante.

(BZ#2054740)

10.3. MISE EN RÉSEAU

Les équipes réseau sont obsolètes dans RHEL 9

Le service **teamd** et la bibliothèque **libteam** sont obsolètes dans Red Hat Enterprise Linux 9 et seront supprimés dans la prochaine version majeure. En remplacement, configurez un lien au lieu d'une équipe réseau.

Red Hat concentre ses efforts sur le bonding basé sur le noyau afin d'éviter de maintenir deux fonctionnalités, les bonds et les teams, qui ont des fonctions similaires. Le code de bonding a été adopté par un grand nombre de clients, est robuste et est développé par une communauté active. Par conséquent, le code de bonding reçoit des améliorations et des mises à jour.

Pour plus d'informations sur la migration d'une équipe vers un lien, voir Migration d'une configuration d'équipe réseau vers un lien réseau.

(BZ#1935544)

NetworkManager stocke les nouvelles configurations de réseau sur /etc/NetworkManager/system-connections/ dans un fichier clé

Auparavant, NetworkManager stockait les nouvelles configurations réseau à l'adresse **/etc/sysconfig/network-scripts/** au format **ifcfg**. À partir de RHEL 9.0, RHEL stocke les nouvelles

configurations réseau à l'adresse **/etc/NetworkManager/system-connections/** dans un format de fichier clé. Les connexions pour lesquelles les configurations sont stockées sur **/etc/sysconfig/network-scripts/** dans l'ancien format continuent de fonctionner sans interruption. Les modifications apportées aux profils existants continuent de mettre à jour les anciens fichiers.

(BZ#1894877)

Le back-end iptables dans firewalld est obsolète

Dans RHEL 9, le cadre **iptables** est obsolète. Par conséquent, le backend **iptables** et le **direct interface** dans **firewalld** sont également obsolètes. Au lieu de **direct interface**, vous pouvez utiliser les fonctionnalités natives de **firewalld** pour configurer les règles requises.

(BZ#2089200)

10.4. NOYAU

L'encapsulation ATM est obsolète dans RHEL 9

L'encapsulation du mode de transfert asynchrone (ATM) permet une connectivité de couche 2 (protocole point à point, Ethernet) ou de couche 3 (IP) pour la couche d'adaptation ATM 5 (AAL-5). Red Hat ne fournit plus de support pour les pilotes ATM NIC depuis RHEL 7. La prise en charge de l'implémentation ATM est abandonnée dans RHEL 9. Ces protocoles ne sont actuellement utilisés que dans les chipsets, qui prennent en charge la technologie ADSL et sont progressivement abandonnés par les fabricants. Par conséquent, l'encapsulation ATM est dépréciée dans Red Hat Enterprise Linux 9.

Pour plus d'informations, voir PPP Over AAL5, Multiprotocol Encapsulation over ATM Adaptation Layer 5, et Classical IP and ARP over ATM .

(BZ#2058153)

10.5. SYSTÈMES DE FICHIERS ET STOCKAGE

lvm2-activation-generator et ses services générés sont supprimés dans RHEL 9.0

Le programme **lvm2-activation-generator** et ses services générés **lvm2-activation**, **lvm2-activation-early**, et **lvm2-activation-net** sont supprimés dans RHEL 9.0. Le paramètre **lvm.conf event_activation**, utilisé pour activer les services, n'est plus fonctionnel. La seule méthode d'activation automatique des groupes de volumes est l'activation basée sur les événements.

(BZ#2038183)

10.6. LANGAGES DE PROGRAMMATION DYNAMIQUES, SERVEURS WEB ET DE BASE DE DONNÉES

libdb a été supprimé

RHEL 8 et RHEL 9 fournissent actuellement la version 5.3.28 de Berkeley DB (**libdb**), qui est distribuée sous la licence LGPLv2. La version 6 de Berkeley DB en amont est disponible sous la licence AGPLv3, qui est plus restrictive.

Le paquet **libdb** est obsolète depuis RHEL 9 et pourrait ne plus être disponible dans les prochaines versions majeures de RHEL.

En outre, les algorithmes cryptographiques ont été retirés de **libdb** dans RHEL 9 et plusieurs dépendances de **libdb** ont été supprimées de RHEL 9.

Il est conseillé aux utilisateurs de **libdb** de migrer vers une autre base de données clé-valeur. Pour plus d'informations, voir l'article de la base de connaissances Remplacements disponibles pour Berkeley DB (libdb) dans RHEL.

(BZ#1927780, BZ#1974657, JIRA:RHELPLAN-80695)

10.7. COMPILATEURS ET OUTILS DE DÉVELOPPEMENT

Les clés de taille inférieure à 2048 sont dépassées par openssl 3.0

Les tailles de clés inférieures à 2048 bits sont dépréciées par **openssl** 3.0 et ne fonctionnent plus dans le mode FIPS de Go.

(BZ#2111072)

Certains modes de PKCS1 v1.5 sont désormais obsolètes

Certains modes de **PKCS1** v1.5 ne sont pas approuvés dans **FIPS-140-3** pour le cryptage et sont désactivés. Ils ne fonctionneront plus dans le mode FIPS de Go.

(BZ#2092016)

10.8. GESTION DE L'IDENTITÉ

SHA-1 dans OpenDNSSec est maintenant obsolète

OpenDNSSec prend en charge l'exportation de signatures numériques et d'enregistrements d'authentification à l'aide de l'algorithme **SHA-1**. L'utilisation de l'algorithme **SHA-1** n'est plus prise en charge. Avec la version RHEL 9, **SHA-1** dans OpenDNSSec est déprécié et pourrait être supprimé dans une future version mineure. En outre, la prise en charge d'OpenDNSSec est limitée à son intégration avec Red Hat Identity Management. OpenDNSSec n'est pas pris en charge de manière autonome.

(BZ#1979521)

Le domaine du fournisseur de fichiers implicites SSSD est désactivé par défaut

Le domaine fournisseur implicite SSSD **files**, qui récupère les informations sur les utilisateurs à partir de fichiers locaux tels que **/etc/shadow** et les informations sur les groupes à partir de **/etc/groups**, est désormais désactivé par défaut.

Pour récupérer des informations sur les utilisateurs et les groupes à partir de fichiers locaux avec SSSD :

1. Configurer SSSD. Choisissez l'une des options suivantes :
   a. Configurez explicitement un domaine local avec l'option **id_provider=files** dans le fichier de configuration **sssd.conf**.

```
[domain/local]
id_provider=files
...
```

   b. Activez le fournisseur **files** en définissant **enable_files_domain=true** dans le fichier de configuration **sssd.conf**.

```
[sssd]
enable_files_domain = true
```

2. Configurer le commutateur des services de noms.

```
# authselect enable-feature with-files-provider
```

(JIRA:RHELPLAN-100639)

-h et -p ont été supprimées dans les utilitaires clients OpenLDAP.

Le projet OpenLDAP en amont a déprécié les options **-h** et **-p** dans ses utilitaires et recommande d'utiliser l'option **-H** pour spécifier l'URI LDAP. Par conséquent, RHEL 9 a supprimé ces deux options dans tous les utilitaires clients OpenLDAP. Les options **-h** et **-p** seront supprimées des produits RHEL dans les prochaines versions.

(JIRA:RHELPLAN-137660)

10.9. BUREAU

GTK 2 est désormais obsolète

L'ancienne boîte à outils GTK 2 et les paquets suivants ont été supprimés :

- **adwaita-gtk2-theme**
- **gnome-common**
- **gtk2**
- **gtk2-immodules**
- **hexchat**

Plusieurs autres paquets dépendent actuellement de GTK 2. Ils ont été modifiés de manière à ne plus dépendre des paquets dépréciés dans une future version majeure de RHEL.

Si vous maintenez une application qui utilise GTK 2, Red Hat vous recommande de porter l'application vers GTK 4.

(JIRA:RHELPLAN-131882)

10.10. INFRASTRUCTURES GRAPHIQUES

Le serveur X.org est désormais obsolète

Le serveur d'affichage **X.org** est obsolète et sera supprimé dans une prochaine version majeure de RHEL. La session de bureau par défaut est désormais la session **Wayland** dans la plupart des cas.

Le protocole **X11** reste entièrement supporté par le back-end **XWayland**. Par conséquent, les applications qui nécessitent **X11** peuvent fonctionner dans la session **Wayland**.

Red Hat travaille à la résolution des problèmes et des lacunes restants dans la session **Wayland**. Pour les problèmes en suspens dans **Wayland**, voir la section Problèmes connus.

Vous pouvez basculer votre session utilisateur vers le back-end **X.org**. Pour plus d'informations, voir Sélection de l'environnement GNOME et du protocole d'affichage .

(JIRA:RHELPLAN-121048)

Motif a été supprimé

La boîte à outils Motif a été supprimée dans RHEL, car le développement de la communauté Motif en amont est inactif.

Les paquets Motif suivants ont été supprimés, y compris leurs variantes de développement et de débogage :

- **motif**
- **openmotif**
- **openmotif21**
- **openmotif22**

En outre, le paquet **motif-static** a été supprimé.

Red Hat recommande d'utiliser la boîte à outils GTK en remplacement. GTK est plus facile à entretenir et offre de nouvelles fonctionnalités par rapport à Motif.

(JIRA:RHELPLAN-98983)

10.11. RÔLES DU SYSTÈME RED HAT ENTERPRISE LINUX

Le rôle de système networking affiche un avertissement de dépréciation lors de la configuration des équipes sur les nœuds RHEL 9

Par conséquent, l'utilisation du rôle de système **networking** RHEL sur un contrôleur RHEL 8 pour configurer une équipe réseau sur des nœuds RHEL 9 affiche un avertissement concernant son obsolescence.

(BZ#1999770)

10.12. VIRTUALISATION

La vérification de l'image SecureBoot à l'aide de signatures basées sur SHA1 est obsolète

La vérification de l'image SecureBoot à l'aide de signatures basées sur l'algorithme SHA1 sur les exécutables UEFI (PE/COFF) est devenue obsolète. Red Hat recommande plutôt d'utiliser des signatures basées sur l'algorithme SHA2 ou plus récent.

(BZ#1935497)

Prise en charge limitée des instantanés de machines virtuelles

La création d'instantanés de machines virtuelles (VM) n'est actuellement prise en charge que pour les VM n'utilisant pas le micrologiciel UEFI. En outre, pendant l'opération de snapshot, le moniteur QEMU peut se bloquer, ce qui a un impact négatif sur les performances de l'hyperviseur pour certaines charges de travail.

Notez également que le mécanisme actuel de création d'instantanés de VM est obsolète et que Red Hat

ne recommande pas l'utilisation d'instantanés de VM dans un environnement de production. Cependant, un nouveau mécanisme d'instantané de VM est en cours de développement et devrait être entièrement mis en œuvre dans une prochaine version mineure de RHEL 9.

(JIRA:RHELPLAN-15509, BZ#1621944)

virt-manager a été supprimé

L'application Virtual Machine Manager, également connue sous le nom de **virt-manager**, a été supprimée. La console web RHEL, également connue sous le nom de **Cockpit**, est destinée à la remplacer dans une version ultérieure. Il est donc recommandé d'utiliser la console web pour gérer la virtualisation dans une interface graphique. Notez toutefois que certaines fonctionnalités disponibles sur **virt-manager** peuvent ne pas être encore disponibles dans la console web RHEL.

(JIRA:RHELPLAN-10304)

libvirtd est devenu obsolète

Le démon monolithique **libvirt**, **libvirtd**, a été abandonné dans RHEL 9 et sera supprimé dans une prochaine version majeure de RHEL. Notez que vous pouvez toujours utiliser **libvirtd** pour gérer la virtualisation sur votre hyperviseur, mais Red Hat recommande de passer aux démons modulaires **libvirt** récemment introduits. Pour obtenir des instructions et des détails, consultez le document RHEL 9 Configuring and Managing Virtualization (Configuration et gestion de la virtualisation).

(JIRA:RHELPLAN-113995)

Le pilote de disquette virtuelle est devenu obsolète

Le pilote **isa-fdc**, qui contrôle les périphériques de disquette virtuels, est désormais obsolète et ne sera plus pris en charge dans une prochaine version de RHEL. Par conséquent, pour assurer la compatibilité avec les machines virtuelles (VM) migrées, Red Hat déconseille l'utilisation de périphériques à disquette dans les VM hébergées sur RHEL 9.

(BZ#1965079)

le format d'image qcow2-v2 est obsolète

Avec RHEL 9, le format qcow2-v2 pour les images de disques virtuels est devenu obsolète et ne sera plus pris en charge dans une prochaine version majeure de RHEL. En outre, RHEL 9 Image Builder ne peut pas créer d'images de disque au format qcow2-v2.

Au lieu de qcow2-v2, Red Hat recommande fortement d'utiliser qcow2-v3. Pour convertir une image qcow2-v2 en une version de format plus récente, utilisez la commande **qemu-img amend**.

(BZ#1951814)

Les anciens modèles de CPU sont désormais obsolètes

Un nombre important de modèles de CPU sont devenus obsolètes et ne seront plus pris en charge pour une utilisation dans des machines virtuelles (VM) dans une prochaine version majeure de RHEL. Les modèles obsolètes sont les suivants :

- Pour Intel : modèles antérieurs aux familles de processeurs Intel Xeon 55xx et 75xx (également connus sous le nom de Nehalem)
- Pour AMD : modèles antérieurs à AMD Opteron G4
- Pour IBM Z : modèles antérieurs à IBM z14

Pour vérifier si votre VM utilise un modèle de processeur obsolète, utilisez l'utilitaire **virsh dominfo** et recherchez une ligne similaire à la suivante dans la section **Messages**:

```
tainted: use of deprecated configuration settings
deprecated configuration: CPU model 'i486'
```

(BZ#2060839)

10.13. CONTENEURS

L'exécution de conteneurs RHEL 9 sur un hôte RHEL 7 n'est pas prise en charge

L'exécution de conteneurs RHEL 9 sur un hôte RHEL 7 n'est pas prise en charge. Cela peut fonctionner, mais ce n'est pas garanti.

Pour plus d'informations, voir la Matrice de compatibilité des conteneurs Red Hat Enterprise Linux .

(JIRA:RHELPLAN-100087)

L'algorithme de hachage SHA1 utilisé dans Podman est obsolète

L'algorithme SHA1 utilisé pour générer le nom de fichier de l'espace de noms du réseau sans racine n'est plus pris en charge dans Podman. Par conséquent, les conteneurs sans racine démarrés avant la mise à jour vers Podman 4.1.1 ou une version ultérieure doivent être redémarrés s'ils sont reliés à un réseau (et pas seulement en utilisant **slirp4netns**) pour s'assurer qu'ils peuvent se connecter aux conteneurs démarrés après la mise à jour.

(BZ#2069279)

rhel9/pause a été supprimé

L'image du conteneur **rhel9/pause** a été supprimée.

(BZ#2106816)

10.14. PAQUETS OBSOLÈTES

Cette section répertorie les paquetages qui ont été dépréciés et qui ne seront probablement pas inclus dans une prochaine version majeure de Red Hat Enterprise Linux.

Pour les modifications apportées aux paquets entre RHEL 8 et RHEL 9, voir Changements apportés aux paquets dans le document *Considerations in adopting RHEL 9* .

IMPORTANT

Le statut de support des paquetages dépréciés reste inchangé dans RHEL 9. Pour plus d'informations sur la durée du support, voir Red Hat Enterprise Linux Life Cycle et Red Hat Enterprise Linux Application Streams Life Cycle.

Les paquets suivants sont obsolètes dans RHEL 9 :

- iptables-devel
- iptables-libs

- iptables-nft
- iptables-nft-services
- iptables-utils
- libdb
- mcpp
- mod_auth_mellon
- python3-pytz
- xorg-x11-server-Xorg

CHAPITRE 11. PROBLÈMES CONNUS

Cette partie décrit les problèmes connus dans Red Hat Enterprise Linux 9.1.

11.1. CRÉATION D'INSTALLATEURS ET D'IMAGES

Les commandes reboot --kexec et inst.kexec ne fournissent pas un état prévisible du système

L'installation de RHEL à l'aide de la commande **reboot --kexec** Kickstart ou des paramètres de démarrage du noyau **inst.kexec** n'offre pas le même état prévisible du système qu'un redémarrage complet. Par conséquent, le passage au système installé sans redémarrage peut produire des résultats imprévisibles.

Notez que la fonctionnalité **kexec** est obsolète et sera supprimée dans une prochaine version de Red Hat Enterprise Linux.

(BZ#1697896)

Local Media la source d'installation n'est pas détectée lors du démarrage de l'installation à partir d'une clé USB créée à l'aide d'un outil tiers

Lors du démarrage de l'installation RHEL à partir d'une clé USB créée à l'aide d'un outil tiers, le programme d'installation ne détecte pas la source d'installation **Local Media** (seule *Red Hat CDN* est détectée).

Ce problème survient parce que l'option de démarrage par défaut **int.stage2=** tente de rechercher le format d'image **iso9660**. Cependant, un outil tiers peut créer une image ISO avec un format différent.

En guise de solution de contournement, utilisez l'une ou l'autre des solutions suivantes :

- Lors du démarrage de l'installation, cliquez sur la touche **Tab** pour modifier la ligne de commande du noyau et remplacez l'option de démarrage **inst.stage2=** par **inst.repo=**.
- Pour créer un périphérique USB amorçable sous Windows, utilisez Fedora Media Writer.
- Si vous utilisez un outil tiers tel que Rufus pour créer un périphérique USB amorçable, régénérez d'abord l'image ISO RHEL sur un système Linux, puis utilisez l'outil tiers pour créer un périphérique USB amorçable.

Pour plus d'informations sur les étapes à suivre pour exécuter l'une des solutions de contournement spécifiées, voir, Le support d'installation n'est pas détecté automatiquement lors de l'installation de RHEL 8.3.

(BZ#1877697)

Les commandes auth et authconfig Kickstart nécessitent le dépôt AppStream

Le paquetage **authselect-compat** est requis par les commandes Kickstart **auth** et **authconfig** lors de l'installation. Sans ce paquet, l'installation échoue si **auth** ou **authconfig** est utilisé. Cependant, par conception, le paquet **authselect-compat** n'est disponible que dans le dépôt AppStream.

Pour contourner ce problème, vérifiez que les dépôts BaseOS et AppStream sont disponibles pour le programme d'installation ou utilisez la commande **authselect** Kickstart pendant l'installation.

(BZ#1640697)

Politiques SELinux inattendues sur les systèmes où Anaconda s'exécute en tant qu'application

Lorsqu'Anaconda est exécuté en tant qu'application sur un système déjà installé (par exemple pour effectuer une autre installation sur un fichier image à l'aide de l'option **–image** anaconda), il n'est pas interdit au système de modifier les types et attributs SELinux au cours de l'installation. Par conséquent, certains éléments de la politique SELinux peuvent changer sur le système où Anaconda est exécuté. Pour contourner ce problème, n'exécutez pas Anaconda sur le système de production et exécutez-le dans une machine virtuelle temporaire. Ainsi, la politique SELinux sur un système de production n'est pas modifiée. L'exécution d'Anaconda dans le cadre du processus d'installation du système, tel que l'installation à partir de **boot.iso** ou **dvd.iso**, n'est pas concernée par ce problème.

(BZ#2050140)

Le lecteur de CD-ROM USB n'est pas disponible comme source d'installation dans Anaconda

L'installation échoue lorsque le lecteur de CD-ROM USB en est la source et que la commande Kickstart **ignoredisk --only-use=** est spécifiée. Dans ce cas, Anaconda ne peut pas trouver et utiliser ce disque source.

Pour contourner ce problème, utilisez la commande **harddrive --partition=sdX --dir=**/ pour effectuer l'installation à partir d'un lecteur de CD-ROM USB. L'installation n'échoue alors pas.

(BZ#1914955)

Échec des installations de disques durs partitionnés avec le système de fichiers iso9660

Vous ne pouvez pas installer RHEL sur des systèmes dont le disque dur est partitionné avec le système de fichiers **iso9660**. Cela est dû à la mise à jour du code d'installation qui est configuré pour ignorer tout disque dur contenant une partition du système de fichiers **iso9660**. Cela se produit même lorsque RHEL est installé sans utiliser de DVD.

Pour contourner ce problème, ajoutez le script suivant dans le fichier kickstart pour formater le disque avant le début de l'installation.

Remarque : avant d'exécuter la solution de contournement, sauvegardez les données disponibles sur le disque. La commande **wipefs** formate toutes les données existantes sur le disque.

```
%pre
wipefs -a /dev/sda
%end
```

Par conséquent, les installations fonctionnent comme prévu, sans aucune erreur.

(BZ#1929105)

Anaconda ne parvient pas à vérifier l'existence d'un compte d'utilisateur administrateur

Lors de l'installation de RHEL à l'aide d'une interface graphique, Anaconda ne vérifie pas si le compte administrateur a été créé. En conséquence, les utilisateurs peuvent installer un système sans aucun compte d'utilisateur administrateur.

Pour contourner ce problème, veillez à configurer un compte d'utilisateur administrateur ou à définir le mot de passe root et à déverrouiller le compte root. Ainsi, les utilisateurs peuvent effectuer des tâches administratives sur le système installé.

(BZ#2047713)

De nouvelles fonctionnalités XFS empêchent le démarrage des systèmes PowerNV IBM POWER dont le microprogramme est antérieur à la version 5.10

Les systèmes PowerNV IBM POWER utilisent un noyau Linux pour le micrologiciel et Petitboot en remplacement de GRUB. Ainsi, le noyau du microprogramme monte **/boot** et Petitboot lit la configuration de GRUB et démarre RHEL.

Le noyau RHEL 9 introduit les fonctionnalités **bigtime=1** et **inobtcount=1** dans le système de fichiers XFS, que les noyaux dotés d'un microprogramme antérieur à la version 5.10 ne comprennent pas.

Pour contourner ce problème, vous pouvez utiliser un autre système de fichiers pour **/boot**, par exemple ext4.

(BZ#1997832)

Impossible d'installer RHEL lorsque la taille du PReP n'est pas de 4 ou 8 MiB

Le programme d'installation RHEL ne peut pas installer le chargeur de démarrage si la partition PowerPC Reference Platform (PReP) est d'une taille différente de 4 MiB ou 8 MiB sur un disque qui utilise des secteurs de 4 kiB. Par conséquent, vous ne pouvez pas installer RHEL sur le disque.

Pour contourner le problème, assurez-vous que la taille de la partition PReP est exactement de 4 ou 8 Mo et qu'elle n'est pas arrondie à une autre valeur. En conséquence, le programme d'installation peut maintenant installer RHEL sur le disque.

(BZ#2026579)

Le programme d'installation affiche un espace disque total incorrect lors d'un partitionnement personnalisé avec des périphériques à chemins multiples

Le programme d'installation ne filtre pas les chemins individuels des périphériques à chemins multiples lors du partitionnement personnalisé. Le programme d'installation affiche donc les chemins d'accès individuels aux périphériques à chemins d'accès multiples et les utilisateurs peuvent sélectionner les chemins d'accès individuels aux périphériques à chemins d'accès multiples pour les partitions créées. En conséquence, une somme incorrecte de l'espace disque total est affichée. Elle est calculée en ajoutant la taille de chaque chemin d'accès individuel à l'espace disque total.

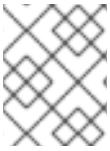
En guise de solution, utilisez uniquement les périphériques à chemins multiples et non les chemins individuels lors du partitionnement personnalisé, et ignorez l'espace disque total calculé de manière incorrecte.

(BZ#2052938)

Échec de l'installation avec des périphériques NVMe sur Fibre Channel

Lors de l'installation de RHEL, le programme d'installation affiche et permet de sélectionner des périphériques Non-volatile Memory Express (NVMe) sur Fibre Channel. L'utilisation de ces périphériques pendant le processus d'installation n'est pas prise en charge. Par conséquent, le processus d'installation peut échouer ou le système installé peut ne pas démarrer correctement.

Pour contourner ce problème, n'utilisez pas de périphériques NVMe over Fibre Channel lors d'une installation interactive (mode texte ou graphique). Lors de l'exécution d'une installation Kickstart, configurez le système pour qu'il ignore les périphériques NVMe over Fibre Channel à l'aide de la commande **ignoredisk --drives=<IGNORE_DISKS>** Kickstart, en remplaçant **<IGNORE_DISKS>** par les périphériques NVMe over Fibre Channel. Vous pouvez également définir les disques que Kickstart utilise pendant l'installation à l'aide de la commande **ignoredisk --only-use=<ONLY_USE_DISKS>**, en remplaçant **<ONLY_USE_DISKS>** par les périphériques pris en charge.



NOTE

L'installation échoue uniquement pour les périphériques NVMe sur Fibre Channel. Les périphériques NVMe attachés localement fonctionnent correctement.

Pour des informations détaillées sur la commande Kickstart **ignoredisk**, voir Commandes Kickstart pour la gestion du stockage dans le guide Exécution d'une installation RHEL 9 avancée.

(BZ#2107346)

L'image d'installation de RHEL for Edge ne parvient pas à créer des points de montage lors de l'installation d'une charge utile rpm-ostree

Lors du déploiement des charges utiles **rpm-ostree**, utilisées par exemple dans une image d'installation RHEL for Edge, le programme d'installation ne crée pas correctement certains points de montage pour les partitions personnalisées. En conséquence, l'installation est interrompue avec l'erreur suivante :

```
The command 'mount --bind /mnt/sysimage/data /mnt/sysroot/data' exited with the code 32.
```

Pour contourner ce problème :

- Utilisez un schéma de partitionnement automatique et n'ajoutez pas de points de montage manuellement.
- Attribuer manuellement des points de montage uniquement dans le répertoire **/var**. Par exemple, **/var/*my-mount-point***), et les répertoires standard suivants : / **/boot** , **/var**.

Le processus d'installation se termine donc avec succès.

(BZ#2125542)

NetworkManager ne démarre pas après l'installation lorsqu'il est connecté à un réseau mais qu'il n'y a pas d'adresse DHCP ou d'adresse IP statique configurée

À partir de RHEL 9.0, Anaconda active automatiquement les périphériques réseau lorsqu'il n'y a pas de configuration réseau spécifique **ip=** ou kickstart. Anaconda crée un fichier de configuration persistant par défaut pour chaque périphérique Ethernet. Dans le profil de connexion, les valeurs **ONBOOT** et **autoconnect** sont définies sur **true**. Par conséquent, lors du démarrage du système installé, RHEL active les périphériques réseau et le service **networkManager-wait-online** échoue.

En guise de solution de contournement, procédez de l'une des manières suivantes :

- Supprimez toutes les connexions à l'aide de l'utilitaire **nmcli**, à l'exception de celle que vous souhaitez utiliser. Par exemple :
  a. Liste de tous les profils de connexion :

     ```
     # nmcli connection show
     ```

  b. Supprimez les profils de connexion dont vous n'avez pas besoin :

     ```
     # nmcli connection delete <connection_name>
     ```

     Remplacez <nom_de_la_connexion> par le nom de la connexion que vous souhaitez supprimer.

- Désactive la fonction de connexion automatique au réseau dans Anaconda si aucune configuration réseau spécifique **ip=** ou kickstart n'est définie.

  a. Dans l'interface graphique d'Anaconda, naviguez jusqu'à **Network & Host Name**

  b. Sélectionnez un périphérique réseau à désactiver.

  c. Cliquez sur **Configure**.

  d. Dans l'onglet **General**, désélectionnez l'option **Connect automatically with priority**

  e. Cliquez sur **Save**.

(BZ#2115783)

11.2. GESTION DES ABONNEMENTS

L'utilitaire subscription-manager conserve le texte non essentiel dans le terminal après la fin d'une commande

À partir de RHEL 9.1, l'utilitaire **subscription-manager** affiche des informations sur la progression du traitement d'une opération. Pour certaines langues (généralement non latines), les messages de progression peuvent ne pas être effacés à la fin de l'opération. Par conséquent, vous pouvez voir des parties d'anciens messages de progression dans le terminal.

Notez qu'il ne s'agit pas d'une défaillance fonctionnelle pour **subscription-manager**.

Pour contourner ce problème, effectuez l'une des opérations suivantes :

- Inclure l'option **--no-progress-messages** lors de l'exécution des commandes `subscription-manager` dans le terminal
- Configurez **subscription-manager** pour qu'il fonctionne sans afficher de messages de progression en entrant la commande suivante :

```
# subscription-manager config --rhsm.progress_messages=0
```

(BZ#2136694)

11.3. GESTION DES LOGICIELS

Le processus d'installation ne répond parfois plus

Lorsque vous installez RHEL, le processus d'installation ne répond parfois plus. Le fichier **/tmp/packaging.log** affiche le message suivant à la fin :

```
10:20:56,416 DDEBUG dnf: RPM transaction over.
```

Pour contourner ce problème, redémarrez le processus d'installation.

(BZ#2073510)

Une mise à niveau DNF de sécurité échoue pour les paquets qui changent d'architecture au cours de la mise à niveau

Le correctif pour BZ#2108969, publié avec l'avis RHBA-2022:8295, introduit la régression suivante : La

mise à niveau DNF utilisant des filtres de sécurité échoue pour les paquets dont l'architecture passe de ou à **noarch** au cours de la mise à niveau. Par conséquent, elle peut laisser le système dans un état vulnérable.

Pour contourner ce problème, effectuez la mise à jour normale sans les filtres de sécurité.

(BZ#2108969)

11.4. SHELLS ET OUTILS DE LIGNE DE COMMANDE

Le renommage des interfaces réseau à l'aide des fichiers ifcfg échoue

Sur RHEL 9, le paquetage **initscripts** n'est pas installé par défaut. Par conséquent, le renommage des interfaces réseau à l'aide des fichiers **ifcfg** échoue. Pour résoudre ce problème, Red Hat vous recommande d'utiliser les règles **udev** ou les fichiers de liens pour renommer les interfaces. Pour plus de détails, reportez-vous à Nommage cohérent des périphériques d'interface réseau et à la page de manuel **systemd.link(5)**.

Si vous ne pouvez pas utiliser l'une des solutions recommandées, installez le paquetage **initscripts**.

(BZ#2018112)

Le paquet chkconfig n'est pas installé par défaut dans RHEL 9

Le paquet **chkconfig**, qui met à jour et interroge les informations de niveau d'exécution des services système, n'est pas installé par défaut dans RHEL 9.

Pour gérer les services, utilisez les commandes **systemctl** ou installez manuellement le paquet **chkconfig**.

Pour plus d'informations sur **systemd**, voir Introduction à systemd. Pour savoir comment utiliser l'utilitaire **systemctl**, voir Gérer les services système avec systemctl.

(BZ#2053598)

11.5. SERVICES D'INFRASTRUCTURE

Les deux sites bind et unbound désactivent la validation des signatures basées sur SHA-1

Les composants **bind** et **unbound** désactivent la prise en charge de la validation de toutes les signatures RSA/SHA1 (algorithme numéro 5) et RSASHA1-NSEC3-SHA1 (algorithme numéro 7), et l'utilisation de SHA-1 pour les signatures est restreinte dans la politique cryptographique DEFAULT applicable à l'ensemble du système.

Par conséquent, certains enregistrements DNSSEC signés avec les algorithmes SHA-1, RSA/SHA1 et RSASHA1-NSEC3-SHA1 ne sont pas vérifiés dans Red Hat Enterprise Linux 9 et les noms de domaine concernés deviennent vulnérables.

Pour contourner ce problème, passez à un algorithme de signature différent, tel que RSA/SHA-256 ou des clés à courbe elliptique.

Pour plus d'informations et une liste des domaines de premier niveau concernés et vulnérables, voir la solution " DNSSEC records signed with RSASHA1 fail to verify" (enregistrements DNSSEC signés avec RSASHA1 sans vérification).

(BZ#2070495)

named ne démarre pas si le même fichier de zone inscriptible est utilisé dans plusieurs zones

BIND n'autorise pas l'utilisation du même fichier de zone inscriptible dans plusieurs zones. Par conséquent, si une configuration comprend plusieurs zones qui partagent un chemin d'accès à un fichier qui peut être modifié par le service **named**, **named** ne démarre pas. Pour contourner ce problème, utilisez la clause **in-view** pour partager une zone entre plusieurs vues et veillez à utiliser des chemins différents pour les différentes zones. Par exemple, incluez les noms des vues dans le chemin d'accès.

Notez que les fichiers de zone inscriptibles sont généralement utilisés dans les zones où les mises à jour dynamiques sont autorisées, dans les zones esclaves ou dans les zones gérées par DNSSEC.

(BZ#1984982)

11.6. SÉCURITÉ

OpenSSL ne détecte pas si un jeton PKCS #11 prend en charge la création de signatures RSA ou RSA-PSS brutes

Le protocole TLS 1.3 nécessite la prise en charge des signatures RSA-PSS. Si un jeton PKCS #11 ne prend pas en charge les signatures RSA ou RSA-PSS brutes, les applications serveur qui utilisent la bibliothèque **OpenSSL** ne fonctionnent pas avec une clé **RSA** si la clé est détenue par le jeton **PKCS #11**. Par conséquent, la communication TLS échoue dans le scénario décrit.

Pour contourner ce problème, configurez les serveurs et les clients de manière à ce qu'ils utilisent la version 1.2 du protocole TLS, qui est la version la plus élevée disponible.

(BZ#1681178)

OpenSSL traite incorrectement les jetons PKCS #11 qui ne prennent pas en charge les signatures RSA ou RSA-PSS brutes

La bibliothèque **OpenSSL** ne détecte pas les capacités liées aux clés des jetons PKCS #11. Par conséquent, l'établissement d'une connexion TLS échoue lorsqu'une signature est créée avec un jeton qui ne prend pas en charge les signatures RSA ou RSA-PSS brutes.

Pour contourner le problème, ajoutez les lignes suivantes après la ligne **.include** à la fin de la section **crypto_policy** dans le fichier **/etc/pki/tls/openssl.cnf**:

```
SignatureAlgorithms =
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384
MaxProtocol = TLSv1.2
```

Par conséquent, une connexion TLS peut être établie dans le scénario décrit.

(BZ#1685470)

scp vide les fichiers copiés sur eux-mêmes lorsqu'une syntaxe spécifique est utilisée

L'utilitaire **scp** est passé du protocole de copie sécurisée (SCP) au protocole de transfert de fichiers SSH (SFTP), plus sûr. Par conséquent, la copie d'un fichier d'un emplacement vers le même emplacement efface le contenu du fichier. Le problème concerne la syntaxe suivante :

scp localhost:/myfile localhost:/myfile

Pour contourner ce problème, ne copiez pas de fichiers vers une destination identique à l'emplacement source en utilisant cette syntaxe.

Le problème a été corrigé pour les syntaxes suivantes :

- **scp /myfile localhost:/myfile**
- **scp localhost:~/myfile ~/myfile**

(BZ#2056884)

Les suites de chiffrement PSK ne fonctionnent pas avec la politique cryptographique FUTURE

Les suites de chiffrement à clé pré-partagée (PSK) ne sont pas reconnues comme des méthodes d'échange de clés à secret parfait (PFS). Par conséquent, les ciphersuites **ECDHE-PSK** et **DHE-PSK** ne fonctionnent pas avec OpenSSL configuré à **SECLEVEL=3**, par exemple avec la politique cryptographique **FUTURE**. Comme solution de contournement, vous pouvez définir une politique cryptographique moins restrictive ou un niveau de sécurité inférieur (**SECLEVEL**) pour les applications qui utilisent des suites de chiffrement PSK.

(BZ#2060044)

GnuPG permet incorrectement d'utiliser des signatures SHA-1 même si cela est interdit par la norme crypto-policies

Le logiciel cryptographique GNU Privacy Guard (GnuPG) peut créer et vérifier des signatures qui utilisent l'algorithme SHA-1 indépendamment des paramètres définis par les politiques cryptographiques du système. Par conséquent, vous pouvez utiliser SHA-1 à des fins cryptographiques dans la politique cryptographique **DEFAULT**, ce qui n'est pas cohérent avec la dépréciation de cet algorithme peu sûr pour les signatures à l'échelle du système.

Pour contourner ce problème, n'utilisez pas les options de GnuPG qui impliquent SHA-1. Vous empêcherez ainsi GnuPG d'abaisser la sécurité du système par défaut en utilisant les signatures SHA-1 non sécurisées.

(BZ#2070722)

gpg-agent ne fonctionne pas comme agent SSH en mode FIPS

L'outil **gpg-agent** crée des empreintes MD5 lors de l'ajout de clés au programme **ssh-agent**, même si le mode FIPS désactive le condensé MD5. Par conséquent, l'utilitaire **ssh-add** ne parvient pas à ajouter les clés à l'agent d'authentification.

Pour contourner le problème, créez le fichier **~/.gnupg/sshcontrol** sans utiliser la commande **gpg-agent --daemon --enable-ssh-support**. Par exemple, vous pouvez coller la sortie de la commande **gpg --list-keys** au format **<FINGERPRINT> 0** dans **~/.gnupg/sshcontrol**. Ainsi, **gpg-agent** fonctionne comme agent d'authentification SSH.

(BZ#2073567)

La politique SELinux par défaut autorise les exécutables non confinés à rendre leur pile exécutable

L'état par défaut du booléen **selinuxuser_execstack** dans la politique SELinux est activé, ce qui signifie que les exécutables non confinés peuvent rendre leur pile exécutable. Les exécutables ne devraient pas utiliser cette option, qui pourrait indiquer des exécutables mal codés ou une attaque possible. Cependant, en raison de la compatibilité avec d'autres outils, paquetages et produits tiers, Red Hat ne peut pas modifier la valeur du booléen dans la stratégie par défaut. Si votre scénario ne dépend pas de ces aspects de compatibilité, vous pouvez désactiver l'option booléenne dans votre politique locale en entrant la commande **setsebool -P selinuxuser_execstack off**.

(BZ#2064274)

La remédiation des règles d'audit SCAP échoue de manière incorrecte

La remédiation Bash de certaines règles SCAP liées à la configuration de l'Audit n'ajoute pas la clé Audit lors de la remédiation. Ceci s'applique aux règles suivantes :

- **audit_rules_login_events**
- **audit_rules_login_events_faillock**
- **audit_rules_login_events_lastlog**
- **audit_rules_login_events_tallylog**
- **audit_rules_usergroup_modification**
- **audit_rules_usergroup_modification_group**
- **audit_rules_usergroup_modification_gshadow**
- **audit_rules_usergroup_modification_opasswd**
- **audit_rules_usergroup_modification_passwd**
- **audit_rules_usergroup_modification_shadow**
- **audit_rules_time_watch_localtime**
- **audit_rules_mac_modification**
- **audit_rules_networkconfig_modification**
- **audit_rules_sysadmin_actions**
- **audit_rules_session_events**
- **audit_rules_sudoers**
- **audit_rules_sudoers_d**

Par conséquent, si la règle d'audit concernée existe déjà mais n'est pas entièrement conforme à la vérification OVAL, la remédiation corrige la partie fonctionnelle de la règle d'audit, c'est-à-dire les bits de chemin et d'accès, mais n'ajoute pas la clé d'audit. Par conséquent, la règle d'audit résultante fonctionne correctement, mais la règle SCAP signale incorrectement FAIL. Pour contourner ce problème, ajoutez manuellement les clés correctes aux règles d'audit.

(BZ#2120978)

Les règles de délai SSH dans les profils STIG configurent des options incorrectes

Une mise à jour d'OpenSSH a affecté les règles des profils suivants du Guide de mise en œuvre technique de la sécurité de l'Agence des systèmes d'information de la défense (DISA STIG) :

- DISA STIG pour RHEL 9 (**xccdf_org.ssgproject.content_profile_stig**)
- DISA STIG avec GUI pour RHEL 9 (**xccdf_org.ssgproject.content_profile_stig_gui**)

Dans chacun de ces profils, les deux règles suivantes sont affectées :

> Title: Set SSH Client Alive Count Max to zero
> CCE Identifier: CCE-90271-8
> Rule ID: xccdf_org.ssgproject.content_rule_sshd_set_keepalive_0
>
> Title: Set SSH Idle Timeout Interval
> CCE Identifier: CCE-90811-1
> Rule ID: xccdf_org.ssgproject.content_rule_sshd_set_idle_timeout

Lorsqu'elles sont appliquées aux serveurs SSH, chacune de ces règles configure une option (**ClientAliveCountMax** et **ClientAliveInterval**) qui ne se comporte plus comme auparavant. En conséquence, OpenSSH ne déconnecte plus les utilisateurs SSH inactifs lorsqu'il atteint le délai configuré par ces règles. Comme solution de contournement, ces règles ont été temporairement supprimées des profils DISA STIG pour RHEL 9 et DISA STIG avec GUI pour RHEL 9 jusqu'à ce qu'une solution soit développée.

(BZ#2038978)

Keylime pourrait échouer dans l'attestation des systèmes qui accèdent à plusieurs fichiers mesurés par l'IMA

Si un système qui exécute l'agent Keylime accède à plusieurs fichiers mesurés par l'architecture de mesure de l'intégrité (IMA) en succession rapide, le vérificateur Keylime peut traiter incorrectement les ajouts au journal de l'IMA. En conséquence, le hachage en cours d'exécution ne correspond pas à l'état correct du registre de configuration de la plate-forme (PCR), et le système échoue à l'attestation. Il n'existe actuellement aucune solution de contournement.

(BZ#2138167)

Le script de génération de la politique de démarrage mesurée par Keylime peut provoquer une erreur de segmentation et un vidage du noyau

Le script **create_mb_refstate**, qui génère des politiques pour l'attestation de démarrage sur mesure dans Keylime, peut calculer incorrectement la longueur des données dans le champ **DevicePath** au lieu d'utiliser la valeur du champ **LengthOfDevicePath** lors du traitement de la sortie de l'outil **tpm2_eventlog** en fonction de l'entrée fournie. En conséquence, le script tente d'accéder à une mémoire invalide en utilisant la longueur incorrectement calculée, ce qui entraîne une erreur de segmentation et un vidage du noyau. La fonctionnalité principale de Keylime n'est pas affectée par ce problème, mais il se peut que vous ne puissiez pas générer une politique de démarrage mesurée.

Pour contourner ce problème, n'utilisez pas de politique de démarrage mesurée ou écrivez le fichier de politique manuellement à partir des données obtenues à l'aide de l'outil **tpm2_eventlog** du paquetage **tpm2-tools**.

(BZ#2140670)

Certains certificats TPM provoquent le plantage du registraire Keylime

L'option de configuration **require_ek_cert** dans **tenant.conf**, qui devrait être activée dans les déploiements de production, détermine si le locataire Keylime nécessite un certificat de clé d'endossement (EK) du Trusted Platform Module (TPM). Lors de la cotation initiale de l'identité avec **require_ek_cert** activé, Kelime tente de vérifier si le dispositif TPM sur l'agent est authentique en comparant le certificat EK avec les certificats de confiance présents dans le magasin de certificats Keylime TPM. Cependant, certains certificats du magasin sont des certificats x509 malformés et provoquent le plantage du registre Keylime. Il n'existe actuellement aucune solution simple à ce problème, si ce n'est de configurer **require_ek_cert** en **false** et de définir un script personnalisé dans l'option **ek_check_script** qui effectuera la validation EK.

(BZ#2142009)

11.7. MISE EN RÉSEAU

Le service nm-cloud-setup supprime les adresses IP secondaires configurées manuellement sur les interfaces

Sur la base des informations reçues de l'environnement cloud, le service **nm-cloud-setup** configure les interfaces réseau. Désactivez **nm-cloud-setup** pour configurer manuellement les interfaces. Cependant, dans certains cas, d'autres services sur l'hôte peuvent également configurer les interfaces. Par exemple, ces services peuvent ajouter des adresses IP secondaires. Pour éviter cela, **nm-cloud-setup** supprime les adresses IP secondaires :

1. Arrêtez et désactivez le service et la minuterie **nm-cloud-setup**:

```
# systemctl disable --now nm-cloud-setup.service nm-cloud-setup.timer
```

2. Affiche les profils de connexion disponibles :

```
# nmcli connection show
```

3. Réactive les profils de connexion concernés :

```
# nmcli connection up "<profile_name>"
```

Par conséquent, le service ne supprime plus les adresses IP secondaires configurées manuellement sur les interfaces.

(BZ#2151040)

L'absence de mise à jour de la clé de session entraîne l'interruption de la connexion

Le protocole Kernel Transport Layer Security (kTLS) ne prend pas en charge la mise à jour de la clé de session, qui est utilisée par le chiffrement symétrique. Par conséquent, l'utilisateur ne peut pas mettre à jour la clé, ce qui entraîne une interruption de la connexion. Pour contourner ce problème, il faut désactiver le protocole kTLS. Par conséquent, avec la solution de contournement, il est possible de mettre à jour la clé de session avec succès.

(BZ#2013650)

Le paquet initscripts n'est pas installé par défaut

Par défaut, le paquetage **initscripts** n'est pas installé. Par conséquent, les utilitaires **ifup** et **ifdown** ne sont pas disponibles. Vous pouvez utiliser les commandes **nmcli connection up** et **nmcli connection down** pour activer et désactiver les connexions. Si l'alternative proposée ne fonctionne pas, signalez le problème et installez le paquetage **NetworkManager-initscripts-updown**, qui fournit une solution NetworkManager pour les utilitaires **ifup** et **ifdown**.

(BZ#2082303)

11.8. NOYAU

Le pilote mlx5 échoue lors de l'utilisation de l'adaptateur Mellanox ConnectX-5

Dans le modèle de pilote de périphérique de commutateur Ethernet (**switchdev**), le pilote **mlx5** échoue

lorsqu'il est configuré avec le paramètre DMFS (device managed flow steering) et que l'adaptateur **ConnectX-5** est pris en charge par le matériel. En conséquence, vous pouvez voir le message d'erreur suivant :

```
BUG: Bad page cache in process umount pfn:142b4b
```

Pour contourner ce problème, vous devez utiliser le paramètre SMFS (software managed flow steering) au lieu de DMFS.

(BZ#2180665)

L'activation de FADump avec Secure Boot peut entraîner une perte de mémoire (OOM) dans GRUB

Dans l'environnement Secure Boot, GRUB et PowerVM allouent ensemble une zone de mémoire de 512 Mo, connue sous le nom de Real Mode Area (RMA), pour la mémoire de démarrage. La région est divisée entre les composants de démarrage et, si l'un d'entre eux dépasse son allocation, des pannes de mémoire se produisent.

En général, le système de fichiers **initramfs** installé par défaut et la table de symboles **vmlinux** sont dans les limites permettant d'éviter de telles défaillances. Toutefois, si la fonction Firmware Assisted Dump (FADump) est activée dans le système, la taille par défaut de **initramfs** peut augmenter et dépasser 95 Mo. Par conséquent, chaque redémarrage du système entraîne un état GRUB OOM.

Pour éviter ce problème, n'utilisez pas Secure Boot et FADump en même temps. Pour plus d'informations et de méthodes sur la manière de contourner ce problème, voir https://www.ibm.com/support/pages/node/6846531.

(BZ#2149172)

weak-modules from kmod ne fonctionne pas avec les interdépendances de modules

Le script **weak-modules** fourni par le paquet **kmod** détermine quels modules sont compatibles avec les noyaux installés. Cependant, lors de la vérification de la compatibilité des modules avec le noyau, **weak-modules** traite les dépendances des symboles des modules de la version supérieure à la version inférieure du noyau pour lequel ils ont été construits. Par conséquent, les modules avec des interdépendances construits pour différentes versions du noyau peuvent être interprétés comme non compatibles, et le script **weak-modules** ne fonctionne donc pas dans ce cas.

Pour contourner le problème, compilez ou installez les modules supplémentaires avec le dernier noyau stocké avant d'installer le nouveau noyau.

(BZ#2103605)

Le service kdump ne parvient pas à créer le fichier initrd sur les systèmes IBM Z

Sur les systèmes IBM Z 64 bits, le service **kdump** ne parvient pas à charger le disque RAM initial (**initrd**) lorsque les informations de configuration liées à **znet**, telles que **s390-subchannels**, se trouvent dans un profil de connexion **NetworkManager** inactif. Par conséquent, le mécanisme **kdump** échoue avec l'erreur suivante :

```
dracut: Failed to set up znet
kdump: mkdumprd: failed to make kdump initrd
```

En guise de solution de contournement, utilisez l'une des solutions suivantes :

- Configurer un lien ou un pont réseau en réutilisant le profil de connexion qui contient les informations de configuration **znet**:

```
$ nmcli connection modify enc600 master bond0 slave-type bond
```

- Copier les informations de configuration de **znet** du profil de connexion inactif vers le profil de connexion actif :

  a. Exécutez la commande **nmcli** pour interroger les profils de connexion **NetworkManager**:

```
# nmcli connection show

NAME                    UUID              TYPE   Device

bridge-br0           ed391a43-bdea-4170-b8a2 bridge   br0
bridge-slave-enc600  caf7f770-1e55-4126-a2f4 ethernet enc600
enc600               bc293b8d-ef1e-45f6-bad1 ethernet --
```

  b. Mettre à jour le profil actif avec les informations de configuration de la connexion inactive :

```
#!/bin/bash
 inactive_connection=enc600
 active_connection=bridge-slave-enc600
 for name in nettype subchannels options; do
 field=802-3-ethernet.s390-$name
 val=$(nmcli --get-values "$field"connection show "$inactive_connection")
 nmcli connection modify "$active_connection" "$field" $val"
 done
```

  c. Redémarrez le service **kdump** pour que les modifications soient prises en compte :

```
# kdumpctl restart
```

(BZ#2064708)

Le mécanisme kdump ne parvient pas à capturer le fichier vmcore sur les cibles chiffrées par LUKS

Lors de l'exécution de **kdump** sur des systèmes dotés de partitions chiffrées Linux Unified Key Setup (LUKS), les systèmes requièrent une certaine quantité de mémoire disponible. Lorsque la mémoire disponible est inférieure à la quantité de mémoire requise, le service **systemd-cryptsetup** ne parvient pas à monter la partition. Par conséquent, le second noyau ne parvient pas à capturer le fichier de vidage d'urgence (**vmcore**) sur les cibles chiffrées LUKS.

La commande **kdumpctl estimate** vous permet d'interroger **Recommended crashkernel value**, qui est la taille de mémoire recommandée pour **kdump**.

Pour contourner ce problème, procédez comme suit pour configurer la mémoire requise pour **kdump** sur les cibles cryptées LUKS :

1. Imprimer la valeur estimée de **crashkernel**:

```
# kdumpctl estimate
```

2. Configurez la quantité de mémoire requise en augmentant la valeur de **crashkernel**:

```
# grubby --args=crashkernel=652M --update-kernel=ALL
```

3. Redémarrez le système pour que les modifications soient prises en compte.

```
# reboot
```

Par conséquent, **kdump** fonctionne correctement sur les systèmes dotés de partitions chiffrées par LUKS.

(BZ#2017401)

L'allocation de la mémoire du noyau de crash échoue au démarrage

Sur certains systèmes Ampere Altra, l'allocation de la mémoire du noyau de crash pour l'utilisation de **kdump** échoue au démarrage lorsque la mémoire disponible est inférieure à 1 Go. Par conséquent, la commande **kdumpctl** ne parvient pas à démarrer le service **kdump**.

Pour contourner ce problème, procédez de l'une des manières suivantes :

- Diminuez la valeur du paramètre **crashkernel** d'un minimum de 240 MB pour répondre à l'exigence de taille, par exemple **crashkernel=240M**.
- Utilisez l'option **crashkernel=x,high** pour réserver la mémoire du noyau de crash supérieure à 4 Go pour **kdump**.

Par conséquent, l'allocation de mémoire au noyau de crash pour **kdump** n'échoue pas sur les systèmes Ampere Altra.

(BZ#2065013)

La fonctionnalité Delay Accounting n'affiche pas par défaut les colonnes de statistiques SWAPIN et IO%

La fonctionnalité **Delayed Accounting**, contrairement aux premières versions, est désactivée par défaut. Par conséquent, l'application **iotop** n'affiche pas les colonnes de statistiques **SWAPIN** et **IO%** et affiche l'avertissement suivant :

```
CONFIG_TASK_DELAY_ACCT not enabled in kernel, cannot determine SWAPIN and IO%
```

La fonctionnalité **Delay Accounting**, qui utilise l'interface **taskstats**, fournit des statistiques sur les retards pour toutes les tâches ou threads qui appartiennent à un groupe de threads. Les retards dans l'exécution des tâches surviennent lorsqu'elles attendent qu'une ressource du noyau devienne disponible, par exemple, une tâche qui attend qu'une unité centrale soit libre pour s'exécuter. Les statistiques aident à définir la priorité de l'unité centrale d'une tâche, la priorité des entrées/sorties et les valeurs limites de **rss** de manière appropriée.

Pour contourner le problème, vous pouvez activer l'option de démarrage **delayacct** soit au moment de l'exécution, soit au moment du démarrage.

- Pour activer **delayacct** au moment de l'exécution, entrez :

```
echo 1 > /proc/sys/kernel/task_delayacct
```

  Notez que cette commande active la fonctionnalité à l'échelle du système, mais uniquement pour les tâches que vous démarrez après avoir exécuté cette commande.

- Pour activer **delayacct** de manière permanente au démarrage, utilisez l'une des procédures suivantes :
  - Modifiez le fichier **/etc/sysctl.conf** pour remplacer les paramètres par défaut :
    a. Ajoutez l'entrée suivante au fichier **/etc/sysctl.conf**:

    ```
    kernel.task_delayacct = 1
    ```

    Pour plus d'informations, voir Comment définir les variables sysctl sur Red Hat Enterprise Linux.

    b. Redémarrez le système pour que les modifications soient prises en compte.
  - Modifiez le fichier de configuration de GRUB 2 pour remplacer les paramètres par défaut :
    a. Ajouter l'option **delayacct** à l'entrée **GRUB _CMDLINE_LINUX** du fichier **/etc/default/grub**.

    b. Exécutez l'utilitaire **grub2-mkconfig** pour régénérer la configuration de démarrage :

    ```
    # grub2-mkconfig -o /boot/grub2/grub.cfg
    ```

    Pour plus d'informations, voir Comment modifier de façon permanente la ligne de commande du noyau ?

    c. Redémarrez le système pour que les modifications soient prises en compte.

Par conséquent, l'application **iotop** affiche les colonnes de statistiques **SWAPIN** et **IO%**.

(BZ#2132480)

kTLS ne prend pas en charge le délestage de TLS 1.3 vers les cartes réseau

Kernel Transport Layer Security (kTLS) ne prend pas en charge le délestage de TLS 1.3 vers les cartes réseau. Par conséquent, le chiffrement logiciel est utilisé avec TLS 1.3 même lorsque les cartes réseau prennent en charge le délestage TLS. Pour contourner ce problème, désactivez TLS 1.3 si le délestage est nécessaire. Par conséquent, vous ne pouvez décharger que TLS 1.2. Lorsque TLS 1.3 est utilisé, les performances sont moindres, car TLS 1.3 ne peut pas être déchargé.

(BZ#2000616)

Le site iwl7260-firmware interrompt le Wi-Fi sur Intel Wi-Fi 6 AX200, AX210, et Lenovo ThinkPad P1 Gen 4

Après la mise à jour du pilote **iwl7260-firmware** ou **iwl7260-wifi** vers la version fournie par RHEL 8.7 et/ou RHEL 9.1 (et plus), le matériel se retrouve dans un état interne incorrect. Par conséquent, les cartes Intel Wifi 6 peuvent ne pas fonctionner et afficher le message d'erreur :

```
kernel: iwlwifi 0000:09:00.0: Failed to start RT ucode: -110
kernel: iwlwifi 0000:09:00.0: WRT: Collecting data: ini trigger 13 fired (delay=0ms)
kernel: iwlwifi 0000:09:00.0: Failed to run INIT ucode: -110
```

Une solution non confirmée consiste à éteindre le système et à le rallumer. Ne pas redémarrer.

(BZ#2129288)

11.9. CHARGEUR DE DÉMARRAGE

Les nouveaux noyaux perdent les options de ligne de commande précédentes

Le chargeur d'amorçage GRUB n'applique pas aux nouveaux noyaux les options de ligne de commande personnalisées et configurées précédemment. Par conséquent, lorsque vous mettez à jour le paquetage du noyau, le comportement du système peut changer après le redémarrage en raison des options manquantes.

Pour contourner le problème, ajoutez manuellement toutes les options personnalisées de la ligne de commande du noyau après chaque mise à jour du noyau. Ainsi, le noyau applique les options personnalisées comme prévu, jusqu'à la prochaine mise à jour du noyau.

(BZ#1969362)

Le comportement de grubby diverge de sa documentation

Lorsque vous ajoutez un nouveau noyau à l'aide de l'outil **grubby** et que vous ne spécifiez aucun argument, **grubby** transmet les arguments par défaut à la nouvelle entrée. Ce comportement se produit même sans passer l'argument **--copy-default**. L'utilisation des options **--args** et **--copy-default** permet de s'assurer que ces arguments sont ajoutés aux arguments par défaut, comme indiqué dans la documentation **grubby**.

Cependant, lorsque vous ajoutez des arguments supplémentaires, tels que **$tuned_params**, l'outil **grubby** ne transmet pas ces arguments à moins que l'option **--copy-default** ne soit invoquée.

Dans cette situation, deux solutions sont possibles :

- Soit vous définissez l'argument **root=** et laissez **--args** vide :

```
# grubby --add-kernel /boot/my_kernel --initrd /boot/my_initrd --args "root=/dev/mapper/rhel-root" --title "entry_with_root_set"
```

- Ou définir l'argument **root=** et les arguments spécifiés, mais pas les arguments par défaut :

```
# grubby --add-kernel /boot/my_kernel --initrd /boot/my_initrd --args "root=/dev/mapper/rhel-root some_args and_some_more" --title "entry_with_root_set_and_other_args_too"
```

(BZ#2127453)

11.10. SYSTÈMES DE FICHIERS ET STOCKAGE

Les instances RHEL sur Azure ne démarrent pas si elles sont provisionnées par **cloud-init** et configurées avec une entrée de montage NFSv3

Actuellement, le démarrage d'une machine virtuelle RHEL (VM) sur la plateforme cloud Microsoft Azure échoue si la VM a été provisionnée par l'outil **cloud-init** et que le système d'exploitation invité de la VM possède une entrée de montage NFSv3 dans le fichier **/etc/fstab**.

(BZ#2081114)

Anaconda ne parvient pas à se connecter au serveur iSCSI à l'aide de la méthode **no authentication** après une tentative d'authentification CHAP infructueuse

Lorsque vous ajoutez des disques iSCSI à l'aide de l'authentification CHAP et que la tentative de connexion échoue en raison d'informations d'identification incorrectes, une nouvelle tentative de

connexion aux disques à l'aide de la méthode **no authentication** échoue. Pour contourner ce problème, fermez la session en cours et connectez-vous à l'aide de la méthode **no authentication**.

(BZ#1983602)

Device Mapper Multipath n'est pas pris en charge avec NVMe/TCP

L'utilisation de Device Mapper Multipath avec le pilote **nvme-tcp** peut entraîner des avertissements Call Trace et une instabilité du système. Pour contourner ce problème, les utilisateurs de NVMe/TCP doivent activer le multipathing NVMe natif et ne pas utiliser les outils **device-mapper-multipath** avec NVMe.

Par défaut, le multipathing NVMe natif est activé dans RHEL 9. Pour plus d'informations, voir Activation du multipathing sur les périphériques NVMe.

(BZ#2033080)

Le service blk-availability systemd désactive les piles de périphériques complexes

Dans **systemd**, le code de désactivation des blocs par défaut ne gère pas toujours correctement les piles complexes de blocs virtuels. Dans certaines configurations, les périphériques virtuels peuvent ne pas être supprimés lors de l'arrêt, ce qui entraîne l'enregistrement de messages d'erreur. Pour contourner ce problème, désactivez les piles de blocs complexes en exécutant la commande suivante :

```
# systemctl enable --now blk-availability.service
```

Par conséquent, les piles de dispositifs virtuels complexes sont correctement désactivées lors de l'arrêt et ne produisent pas de messages d'erreur.

(BZ#2011699)

supported_speeds sysfs l'attribut signale des valeurs de vitesse incorrectes

Auparavant, en raison d'une définition incorrecte dans le pilote **qla2xxx**, l'attribut **supported_speeds sysfs** pour le HBA indiquait une vitesse de 20 Gb/s au lieu de la vitesse attendue de 64 Gb/s. Par conséquent, si l'adaptateur de bus hôte prenait en charge une vitesse de liaison de 64 Gb/s, la valeur de **sysfs supported_speeds** était incorrecte, ce qui affectait la valeur de la vitesse signalée.

Mais maintenant, l'attribut **supported_speeds** sysfs pour le HBA renvoie une vitesse de 100 Gb/s au lieu des 64 Gb/s prévus, et une vitesse de 50 Gb/s au lieu des 128 Gb/s prévus. Cela n'affecte que la valeur de la vitesse rapportée, et les débits de liaison réels utilisés sur la connexion Fibre sont corrects.

(BZ#2069758)

11.11. LANGAGES DE PROGRAMMATION DYNAMIQUES, SERVEURS WEB ET DE BASE DE DONNÉES

L'option --ssl-fips-mode dans MySQL et MariaDB ne modifie pas le mode FIPS

L'option **--ssl-fips-mode** dans **MySQL** et **MariaDB** dans RHEL fonctionne différemment que dans upstream.

Dans RHEL 9, si vous utilisez **--ssl-fips-mode** comme argument pour le démon **mysqld** ou **mariadbd**, ou si vous utilisez **ssl-fips-mode** dans les fichiers de configuration des serveurs **MySQL** ou **MariaDB**, **--ssl-fips-mode** ne modifie pas le mode FIPS pour ces serveurs de base de données.

Au lieu de cela :

- Si vous attribuez la valeur **ON** à **--ssl-fips-mode**, le démon du serveur **mysqld** ou **mariadbd** ne démarre pas.
- Si vous remplacez **--ssl-fips-mode** par **OFF** sur un système compatible FIPS, les démons de serveur **mysqld** ou **mariadbd** s'exécutent toujours en mode FIPS.

Cela est normal car le mode FIPS doit être activé ou désactivé pour l'ensemble du système RHEL, et non pour des composants spécifiques.

Par conséquent, n'utilisez pas l'option **--ssl-fips-mode** dans **MySQL** ou **MariaDB** dans RHEL. Assurez-vous plutôt que le mode FIPS est activé sur l'ensemble du système RHEL :

- De préférence, installez RHEL avec le mode FIPS activé. L'activation du mode FIPS pendant l'installation garantit que le système génère toutes les clés à l'aide d'algorithmes approuvés par le FIPS et de tests de surveillance continue en place. Pour plus d'informations sur l'installation de RHEL en mode FIPS, voir Installation du système en mode FIPS.
- Vous pouvez également passer en mode FIPS pour l'ensemble du système RHEL en suivant la procédure décrite à la section Passage du système en mode FIPS.

(BZ#1991500)

11.12. COMPILATEURS ET OUTILS DE DÉVELOPPEMENT

Certaines sondes basées sur des symboles ne fonctionnent pas dans SystemTap sur l'architecture ARM 64 bits

La configuration du noyau désactive certaines fonctionnalités nécessaires à **SystemTap**. Par conséquent, certaines sondes basées sur des symboles ne fonctionnent pas sur l'architecture ARM 64 bits. Par conséquent, les scripts **SystemTap** concernés peuvent ne pas s'exécuter ou ne pas recueillir de résultats sur les points de sonde souhaités.

Ce bogue a été corrigé pour les autres architectures avec la publication de l'avis RHBA-2022:5259.

(BZ#2083727)

11.13. GESTION DE L'IDENTITÉ

MIT Kerberos ne prend pas en charge les certificats ECC pour PKINIT

MIT Kerberos n'implémente pas le document RFC5349 request for comments, qui décrit la conception de la prise en charge de la cryptographie à courbe elliptique (ECC) dans la cryptographie à clé publique pour l'authentification initiale (PKINIT). Par conséquent, le paquet MIT **krb5-pkinit**, utilisé par RHEL, ne prend pas en charge les certificats ECC. Pour plus d'informations, voir Prise en charge de la cryptographie à courbe elliptique (ECC) dans la cryptographie à clé publique pour l'authentification initiale dans Kerberos (PKINIT).

(BZ#2106043)

La sous-politique DEFAULT:SHA1 doit être définie sur les clients RHEL 9 pour que PKINIT fonctionne contre les KDC AD

L'algorithme de condensé SHA-1 a été supprimé dans RHEL 9, et les messages CMS pour la cryptographie à clé publique pour l'authentification initiale (PKINIT) sont désormais signés avec l'algorithme SHA-256, plus puissant.

Cependant, le centre de distribution Kerberos (KDC) d'Active Directory (AD) utilise toujours l'algorithme de condensé SHA-1 pour signer les messages CMS. Par conséquent, les clients Kerberos RHEL 9 ne parviennent pas à authentifier les utilisateurs en utilisant PKINIT contre un KDC AD.

Pour contourner le problème, activez la prise en charge de l'algorithme SHA-1 sur vos systèmes RHEL 9 à l'aide de la commande suivante :

```
# update-crypto-policies --set DEFAULT:SHA1
```

(BZ#2060798)

L'authentification PKINIT d'un utilisateur échoue si un agent Kerberos RHEL 9 communique avec un agent Kerberos non RHEL-9 et non AD

Si un agent Kerberos RHEL 9, qu'il s'agisse d'un client ou d'un centre de distribution Kerberos (KDC), interagit avec un agent Kerberos non RHEL-9 qui n'est pas un agent Active Directory (AD), l'authentification PKINIT de l'utilisateur échoue. Pour contourner le problème, effectuez l'une des actions suivantes :

- Définissez la politique cryptographique de l'agent RHEL 9 sur **DEFAULT:SHA1** pour autoriser la vérification des signatures SHA-1 :

  ```
  # update-crypto-polices --set DEFAULT:SHA1
  ```

- Mettez à jour l'agent non RHEL-9 et non AD pour vous assurer qu'il ne signe pas les données CMS à l'aide de l'algorithme SHA-1. Pour cela, mettez à jour votre client Kerberos ou les paquets KDC avec les versions qui utilisent SHA-256 au lieu de SHA-1 :
  - CentOS 9 Stream : krb5-1.19.1-15
  - RHEL 8.7 : krb5-1.18.2-17
  - RHEL 7.9 : krb5-1.15.1-53
  - Fedora Rawhide/36 : krb5-1.19.2-7
  - Fedora 35/34 : krb5-1.19.2-3

Par conséquent, l'authentification PKINIT de l'utilisateur fonctionne correctement.

Notez que pour les autres systèmes d'exploitation, c'est la version krb5-1.20 qui garantit que l'agent signe les données du CMS avec SHA-256 au lieu de SHA-1.

Voir aussi La sous-politique DEFAULT:SHA1 doit être définie sur les clients RHEL 9 pour que PKINIT fonctionne contre les KDC AD.

(BZ#2077450)

La prise en charge FIPS de la confiance AD nécessite la sous-politique cryptographique AD-SUPPORT

Active Directory (AD) utilise des types de chiffrement AES SHA-1 HMAC, qui ne sont pas autorisés par défaut en mode FIPS sur RHEL 9. Si vous souhaitez utiliser des hôtes IdM RHEL 9 avec une confiance AD, activez la prise en charge des types de chiffrement AES SHA-1 HMAC avant d'installer le logiciel IdM.

La conformité FIPS étant un processus qui implique des accords techniques et organisationnels,

consultez votre auditeur FIPS avant d'activer la sous-politique **AD-SUPPORT** pour permettre aux mesures techniques de prendre en charge les types de chiffrement AES SHA-1 HMAC, puis installez RHEL IdM :

```
# update-crypto-policies --set FIPS:AD-SUPPORT
```

(BZ#2057471)

Le client Heimdal ne parvient pas à authentifier un utilisateur à l'aide de PKINIT contre le KDC RHEL 9

Par défaut, un client Heimdal Kerberos initie l'authentification PKINIT d'un utilisateur IdM en utilisant Modular Exponential (MODP) Diffie-Hellman Group 2 pour Internet Key Exchange (IKE). Cependant, le Centre de distribution Kerberos (KDC) du MIT sur RHEL 9 ne prend en charge que les groupes MODP 14 et 16.

Par conséquent, la demande de pré-authentification échoue avec l'erreur **krb5_get_init_creds: PREAUTH_FAILED** sur le client Heimdal et **Key parameters not accepted** sur le KDC MIT RHEL.

Pour contourner ce problème, assurez-vous que le client Heimdal utilise le groupe MODP 14. Définissez le paramètre **pkinit_dh_min_bits** dans la section **libdefaults** du fichier de configuration du client sur 1759 :

```
[libdefaults]
pkinit_dh_min_bits = 1759
```

En conséquence, le client Heimdal complète la pré-authentification PKINIT contre le KDC MIT de RHEL.

(BZ#2106296)

IdM en mode FIPS ne prend pas en charge l'utilisation du protocole NTLMSSP pour établir une confiance bidirectionnelle entre forêts

L'établissement d'une confiance bidirectionnelle entre Active Directory (AD) et Identity Management (IdM) avec le mode FIPS activé échoue parce que l'authentification NTLMSSP (New Technology LAN Manager Security Support Provider) n'est pas conforme à la norme FIPS. IdM en mode FIPS n'accepte pas le hachage NTLM RC4 que le contrôleur de domaine AD utilise lors de la tentative d'authentification.

(BZ#2124243)

Échec des demandes de certificats de transfert de technologie entre IdM et AD

Les informations du certificat d'attributs de privilèges (PAC) dans les tickets IdM Kerberos sont désormais signées avec le cryptage AES SHA-2 HMAC, qui n'est pas pris en charge par Active Directory (AD).

Par conséquent, les demandes de TGS entre IdM et AD, c'est-à-dire les configurations de confiance à double sens, échouent avec l'erreur suivante :

```
"Erreur générique (voir texte électronique) lors de l'obtention des informations d'identification pour <service principal>"
```

(BZ#2060421)

Le chiffrement et le déchiffrement de la chambre forte IdM échouent en mode FIPS

Le chiffrement OpenSSL RSA-PKCS1v15 est bloqué si le mode FIPS est activé. Par conséquent, les chambres fortes de gestion de l'identité (IdM) ne fonctionnent pas correctement, car IdM utilise actuellement le cryptage PKCS1v15 pour envelopper la clé de session avec le certificat de transport.

(BZ#2089907)

Les utilisateurs IdM migrés peuvent être incapables de se connecter en raison de la non-concordance des identifiants de domaine

Si vous avez utilisé le script **ipa migrate-ds** pour migrer des utilisateurs d'un déploiement IdM à un autre, ces utilisateurs peuvent avoir des problèmes pour utiliser les services IdM parce que leurs identifiants de sécurité (SID) n'ont pas le SID du domaine de l'environnement IdM actuel. Par exemple, ces utilisateurs peuvent récupérer un ticket Kerberos avec l'utilitaire **kinit**, mais ils ne peuvent pas se connecter. Pour résoudre ce problème, consultez l'article suivant de la base de connaissances : Migrated IdM users unable to log in due to mismatching domain SIDs (Utilisateurs IdM migrés incapables de se connecter en raison de la non-concordance des SID de domaine).

(JIRA:RHELPLAN-109613)

Directory Server se termine de manière inattendue lorsqu'il est démarré en mode de référence

En raison d'un bogue, le mode de renvoi global ne fonctionne pas dans Directory Server. Si vous démarrez le processus **ns-slapd** avec l'option **refer** en tant qu'utilisateur **dirsrv**, Directory Server ignore les paramètres du port et se termine de manière inattendue. Essayer d'exécuter le processus en tant qu'utilisateur **root** modifie les étiquettes SELinux et empêche le service de démarrer à l'avenir en mode normal. Il n'y a pas de solution de rechange disponible.

(BZ#2053204)

La configuration d'un renvoi pour un suffixe échoue dans Directory Server

Si vous définissez une référence de back-end dans Directory Server, la définition de l'état du back-end à l'aide de la commande **dsconf <instance_name> backend suffix set --state referral** échoue avec l'erreur suivante :

```
Error: 103 - 9 - 53 - Server is unwilling to perform - [] - need to set nsslapd-referral before moving to referral state
```

Par conséquent, la configuration d'un renvoi pour les suffixes échoue. Pour contourner le problème :

1. Réglez manuellement le paramètre **nsslapd-referral**:

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com

dn: cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
changetype: modify
add: nsslapd-referral
nsslapd-referral: ldap://remote_server:389/dc=example,dc=com
```

2. Définir l'état du back-end :

```
# dsconf <instance_name> backend suffix set --state referral
```

Par conséquent, avec la solution de contournement, vous pouvez configurer un renvoi pour un suffixe.

(BZ#2063140)

L'utilitaire dsconf n'a pas d'option pour créer des tâches de correction pour le plug-in entryUUID

L'utilitaire **dsconf** ne propose pas d'option permettant de créer des tâches de correction pour le plug-in **entryUUID**. Par conséquent, les administrateurs ne peuvent pas utiliser **dsconf** pour créer une tâche permettant d'ajouter automatiquement des attributs **entryUUID** aux entrées existantes. Une solution de contournement consiste à créer une tâche manuellement :

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=entryuuid_fixup_<time_stamp>,cn=entryuuid task,cn=tasks,cn=config
objectClass: top
objectClass: extensibleObject
basedn: <fixup base tree>
cn: entryuuid_fixup_<time_stamp>
filter: <filtered_entry>
```

Une fois la tâche créée, Directory Server corrige les entrées dont les attributs **entryUUID** sont manquants ou invalides.

(BZ#2047175)

Risque potentiel lié à l'utilisation de la valeur par défaut de l'option ldap_id_use_start_tls

L'utilisation de **ldap://** sans TLS pour les recherches d'identité peut constituer un risque pour un vecteur d'attaque. En particulier une attaque de type "man-in-the-middle" (MITM) qui pourrait permettre à un pirate d'usurper l'identité d'un utilisateur en modifiant, par exemple, l'UID ou le GID d'un objet renvoyé lors d'une recherche LDAP.

Actuellement, l'option de configuration SSSD pour appliquer TLS, **ldap_id_use_start_tls**, est par défaut **false**. Assurez-vous que votre installation fonctionne dans un environnement de confiance et décidez s'il est sûr d'utiliser une communication non chiffrée pour **id_provider = ldap**. Notez que **id_provider = ad** et **id_provider = ipa** ne sont pas concernés car ils utilisent des connexions cryptées protégées par SASL et GSSAPI.

S'il n'est pas sûr d'utiliser des communications non chiffrées, appliquez le protocole TLS en définissant l'option **ldap_id_use_start_tls** sur **true** dans le fichier **/etc/sssd/sssd.conf**. Il est prévu de modifier le comportement par défaut dans une prochaine version de RHEL.

(JIRA:RHELPLAN-155168)

11.14. BUREAU

Les modules complémentaires de Firefox sont désactivés après la mise à niveau vers RHEL 9

Si vous passez de RHEL 8 à RHEL 9, tous les modules complémentaires que vous avez précédemment activés dans Firefox sont désactivés.

Pour contourner le problème, réinstallez ou mettez à jour manuellement les modules complémentaires. Les modules complémentaires sont alors activés comme prévu.

(BZ#2013247)

VNC ne fonctionne pas après la mise à niveau vers RHEL 9

Après une mise à niveau de RHEL 8 vers RHEL 9, le serveur VNC ne démarre pas, même s'il était activé auparavant.

Pour contourner le problème, activez manuellement le service **vncserver** après la mise à niveau du système :

```
# systemctl enable --now vncserver@ :port-number
```

En conséquence, VNC est maintenant activé et démarre après chaque démarrage du système, comme prévu.

(BZ#2060308)

11.15. INFRASTRUCTURES GRAPHIQUES

Matrox G200e n'affiche aucune sortie sur un écran VGA

Il se peut que votre écran n'affiche aucune sortie graphique si vous utilisez la configuration suivante :

- Le GPU Matrox G200e
- Un écran connecté au contrôleur VGA

Par conséquent, vous ne pouvez pas utiliser ou installer RHEL sur cette configuration.

Pour contourner le problème, suivez la procédure suivante :

1. Amorcez le système dans le menu du chargeur de démarrage.
2. Ajouter l'option **module_blacklist=mgag200** à la ligne de commande du noyau.

Par conséquent, RHEL démarre et affiche la sortie graphique comme prévu, mais la résolution maximale est limitée à 1024x768 avec une profondeur de couleur de 16 bits.

(BZ#1960467)

Les utilitaires de configuration X.org ne fonctionnent pas sous Wayland

Les utilitaires X.org permettant de manipuler l'écran ne fonctionnent pas dans la session Wayland. En particulier, l'utilitaire **xrandr** ne fonctionne pas sous Wayland en raison de son approche différente de la gestion, des résolutions, des rotations et de la mise en page.

(JIRA:RHELPLAN-121049)

Les pilotes NVIDIA pourraient revenir à X.org

Dans certaines conditions, les pilotes propriétaires de NVIDIA désactivent le protocole d'affichage Wayland et reviennent au serveur d'affichage X.org :

- Si la version du pilote NVIDIA est inférieure à 470.
- Si le système est un ordinateur portable qui utilise des graphiques hybrides.
- Si vous n'avez pas activé les options requises du pilote NVIDIA.

En outre, Wayland est activé mais la session de bureau utilise X.org par défaut si la version du pilote NVIDIA est inférieure à 510.

(JIRA:RHELPLAN-119001)

Night Light n'est pas disponible sur Wayland avec NVIDIA

Lorsque les pilotes NVIDIA propriétaires sont activés sur votre système, la fonction **Night Light** de GNOME n'est pas disponible dans les sessions Wayland. Les pilotes NVIDIA ne prennent pas actuellement en charge **Night Light**.

(JIRA:RHELPLAN-119852)

11.16. LA CONSOLE WEB

La console VNC fonctionne mal à certaines résolutions

Lorsque vous utilisez la console Virtual Network Computing (VNC) sous certaines résolutions d'affichage, vous pouvez rencontrer un problème de décalage de la souris ou ne voir qu'une partie de l'interface. Par conséquent, l'utilisation de la console VNC peut s'avérer impossible. Pour contourner ce problème, vous pouvez essayer d'agrandir la taille de la console VNC ou utiliser la visionneuse de bureau dans l'onglet Console pour lancer la visionneuse à distance à la place.

(BZ#2030836)

11.17. VIRTUALISATION

L'installation d'une machine virtuelle via https ou ssh échoue dans certains cas

Actuellement, l'utilitaire **virt-install** échoue lorsqu'il tente d'installer un système d'exploitation invité (OS) à partir d'une source ISO via une connexion https ou ssh - par exemple en utilisant **virt-install --cdrom https://example/path/to/image.iso**. Au lieu de créer une machine virtuelle (VM), l'opération décrite se termine de manière inattendue par un message **internal error: process exited while connecting to monitor**.

De même, l'utilisation de la console web RHEL 9 pour installer un système d'exploitation invité échoue et affiche une erreur **Unknown driver 'https'** si vous utilisez une URL https ou ssh, ou la fonction **Download OS**.

Pour contourner ce problème, installez **qemu-kvm-block-curl** et **qemu-kvm-block-ssh** sur l'hôte pour activer la prise en charge des protocoles https et ssh, respectivement. Vous pouvez également utiliser un autre protocole de connexion ou une autre source d'installation.

(BZ#2014229)

L'utilisation des pilotes NVIDIA dans les machines virtuelles désactive Wayland

Actuellement, les pilotes NVIDIA ne sont pas compatibles avec la session graphique Wayland. Par conséquent, les systèmes d'exploitation invités RHEL qui utilisent des pilotes NVIDIA désactivent automatiquement Wayland et chargent une session Xorg à la place. Cela se produit principalement dans les scénarios suivants :

- Lorsque vous faites passer un périphérique GPU NVIDIA dans une machine virtuelle RHEL (VM)
- Lorsque vous affectez un périphérique NVIDIA vGPU à une VM RHEL

(JIRA:RHELPLAN-117234)

Le type de CPU Milan VM n'est parfois pas disponible sur les systèmes AMD Milan

Sur certains systèmes AMD Milan, les options Enhanced REP MOVSB (**erms**) et Fast Short REP MOVSB (**fsrm**) sont désactivées par défaut dans le BIOS. Par conséquent, le type de CPU **Milan** peut ne pas être disponible sur ces systèmes. En outre, la migration en direct de VM entre des hôtes Milan avec des paramètres de drapeaux de fonctionnalités différents peut échouer. Pour résoudre ces problèmes, activez manuellement **erms** et **fsrm** dans le BIOS de votre hôte.

(BZ#2077767)

La désactivation d'AVX rend les machines virtuelles non amorçables

Sur une machine hôte qui utilise un processeur avec support Advanced Vector Extensions (AVX), la tentative de démarrage d'une VM avec AVX explicitement désactivé échoue actuellement et déclenche une panique du noyau dans la VM.

(BZ#2005173)

VNC ne peut pas se connecter aux machines virtuelles UEFI après la migration

Si vous activez ou désactivez une file d'attente de messages lors de la migration d'une machine virtuelle (VM), le client Virtual Network Computing (VNC) ne parviendra pas à se connecter à la VM une fois la migration terminée.

Ce problème ne concerne que les machines virtuelles basées sur l'UEFI qui utilisent l'Open Virtual Machine Firmware (OVMF).

(JIRA:RHELPLAN-135600)

Les cartes réseau virtio de basculement ne reçoivent pas d'adresse IP sur les machines virtuelles Windows

Actuellement, lors du démarrage d'une machine virtuelle (VM) Windows avec seulement une carte d'interface réseau virtio de basculement, la VM ne parvient pas à attribuer une adresse IP à la carte d'interface réseau. Par conséquent, la carte d'interface réseau n'est pas en mesure d'établir une connexion réseau. Il n'existe actuellement aucune solution de contournement.

(BZ#1969724)

La VM Windows ne parvient pas à obtenir l'adresse IP après la réinitialisation de l'interface réseau

Il arrive que les machines virtuelles Windows ne parviennent pas à obtenir une adresse IP après une réinitialisation automatique de l'interface réseau. Par conséquent, la machine virtuelle ne parvient pas à se connecter au réseau. Pour résoudre ce problème, désactivez et réactivez le pilote de la carte réseau dans le gestionnaire de périphériques de Windows.

(BZ#2084003)

Les adaptateurs réseau Broadcom ne fonctionnent pas correctement sur les machines virtuelles Windows après une migration en direct

Actuellement, les adaptateurs réseau de la famille Broadcom, tels que Broadcom, Qlogic ou Marvell, ne peuvent pas être débranchés à chaud pendant la migration en direct des machines virtuelles (VM) Windows. Par conséquent, les adaptateurs ne fonctionnent pas correctement une fois la migration terminée.

Ce problème ne concerne que les adaptateurs connectés à des machines virtuelles Windows utilisant la virtualisation d'E/S à racine unique (SR-IOV).

(BZ#2090712, BZ#2091528, BZ#2111319)

Une interface **hostdev** avec des paramètres de basculement ne peut pas être branchée à chaud après avoir été débranchée à chaud

Après avoir supprimé une interface réseau **hostdev** avec une configuration de basculement d'une machine virtuelle (VM) en cours d'exécution, l'interface ne peut actuellement pas être réattachée à la même VM en cours d'exécution.

(BZ#2052424)

Échec de la migration post-copie en direct de VM avec des VF de basculement

Actuellement, la tentative de migration post-copie d'une machine virtuelle (VM) en cours d'exécution échoue si la VM utilise un périphérique dont la capacité de basculement de la fonction virtuelle (VF) est activée. Pour contourner le problème, utilisez le type de migration standard plutôt que la migration post-copie.

(BZ#1817965)

Le réseau de l'hôte ne peut pas envoyer de ping aux VMs avec VFs pendant la migration en direct

Lors de la migration en direct d'une machine virtuelle (VM) avec une fonction virtuelle (VF) configurée, telle qu'une VM qui utilise le logiciel SR-IOV virtuel, le réseau de la VM n'est pas visible pour les autres périphériques et la VM ne peut pas être atteinte par des commandes telles que **ping**. Cependant, une fois la migration terminée, le problème ne se produit plus.

(BZ#1789206)

L'utilisation d'un grand nombre de files d'attente peut entraîner l'échec des machines virtuelles Windows

Les machines virtuelles (VM) Windows peuvent échouer lorsque le périphérique virtuel Trusted Platform Module (vTPM) est activé et que la fonction *multi-queue virtio-net* est configurée pour utiliser plus de 250 files d'attente.

Ce problème est dû à une limitation du dispositif vTPM. Le périphérique vTPM a une limite codée en dur sur le nombre maximum de descripteurs de fichiers ouverts. Étant donné que plusieurs descripteurs de fichiers sont ouverts pour chaque nouvelle file d'attente, la limite interne du vTPM peut être dépassée, ce qui entraîne l'échec de la VM.

Pour contourner ce problème, choisissez l'une des deux options suivantes :

- Gardez le dispositif vTPM activé, mais utilisez moins de 250 files d'attente.
- Désactivez le dispositif vTPM pour qu'il utilise plus de 250 files d'attente.

(BZ#2020146)

11.18. RHEL DANS LES ENVIRONNEMENTS EN NUAGE

Le clonage ou la restauration de machines virtuelles RHEL 9 utilisant LVM sur Nutanix AHV entraîne la disparition des partitions non root

Lors de l'exécution d'un système d'exploitation invité RHEL 9 sur une machine virtuelle (VM) hébergée sur l'hyperviseur Nutanix AHV, la restauration de la VM à partir d'un snapshot ou le clonage de la VM provoque actuellement la disparition des partitions non racine dans la VM si l'invité utilise Logical

Volume Management (LVM). En conséquence, les problèmes suivants se produisent :

- Après avoir restauré la VM à partir d'un instantané, la VM ne peut pas démarrer et passe en mode d'urgence.
- Une VM créée par clonage ne peut pas démarrer et passe en mode d'urgence.

Pour contourner ces problèmes, procédez comme suit en mode d'urgence de la VM :

1. Supprimez le fichier des périphériques du système LVM : **rm /etc/lvm/devices/system.devices**
2. Recréer les paramètres du périphérique LVM : **vgimportdevices -a**
3. Redémarrer la VM

Cela permet à la VM clonée ou restaurée de démarrer correctement.

Pour éviter que ce problème ne se produise, procédez comme suit avant de cloner une VM ou de créer un instantané de VM :

1. Décommenter la ligne **use_devicesfile = 0** dans le fichier **/etc/lvm/lvm.conf**
2. Redémarrer la VM

(BZ#2059545)

La personnalisation des invités RHEL 9 sur ESXi entraîne parfois des problèmes de réseau

Actuellement, la personnalisation d'un système d'exploitation invité RHEL 9 dans l'hyperviseur VMware ESXi ne fonctionne pas correctement avec les fichiers clés NetworkManager. Par conséquent, si l'invité utilise un tel fichier clé, il aura des paramètres réseau incorrects, tels que l'adresse IP ou la passerelle.

Pour plus d'informations et des instructions de contournement, consultez la base de connaissances VMware.

(BZ#2037657)

La définition d'une IP statique dans une machine virtuelle RHEL sur un hôte VMware ne fonctionne pas

Actuellement, lors de l'utilisation de RHEL en tant que système d'exploitation invité d'une machine virtuelle (VM) sur un hôte VMware, la fonction DatasourceOVF ne fonctionne pas correctement. Par conséquent, si vous utilisez l'utilitaire **cloud-init** pour configurer le réseau de la VM en IP statique et que vous redémarrez ensuite la VM, le réseau de la VM passera en DHCP.

(BZ#1750862)

11.19. CAPACITÉ DE SOUTIEN

Délai d'attente lors de l'exécution de sos report sur IBM Power Systems, Little Endian

Lors de l'exécution de la commande **sos report** sur des systèmes IBM Power, Little Endian avec des centaines ou des milliers de CPU, le plugin processeur atteint son délai d'attente par défaut de 300 secondes lors de la collecte de l'énorme contenu du répertoire **/sys/devices/system/cpu**. Pour contourner ce problème, augmentez le délai d'attente du plugin en conséquence :

- Pour un réglage unique, exécuter :

```
# sos report -k processor.timeout=1800
```

- Pour une modification permanente, modifiez la section **[plugin_options]** du fichier **/etc/sos/sos.conf**:

```
[plugin_options]
# Specify any plugin options and their values here. These options take the form
# plugin_name.option_name = value
#rpm.rpmva = off
processor.timeout = 1800
```

La valeur de l'exemple est fixée à 1800. La valeur particulière du délai d'attente dépend fortement d'un système spécifique. Pour définir le délai d'attente du plugin de manière appropriée, vous pouvez d'abord estimer le temps nécessaire pour collecter le plugin sans délai d'attente en exécutant la commande suivante :

```
# time sos report -o processor -k processor.timeout=0 --batch --build
```

(BZ#1869561)

11.20. CONTENEURS

L'exécution de systemd dans une ancienne image de conteneur ne fonctionne pas

L'exécution de systemd dans une ancienne image de conteneur, par exemple, **centos:7**, ne fonctionne pas :

```
$ podman run --rm -ti centos:7 /usr/lib/systemd/systemd
 Storing signatures
 Failed to mount cgroup at /sys/fs/cgroup/systemd: Operation not permitted
 [!!!!!!] Failed to mount API filesystems, freezing.
```

Pour contourner ce problème, utilisez les commandes suivantes :

```
# mkdir /sys/fs/cgroup/systemd
# mount none -t cgroup -o none,name=systemd /sys/fs/cgroup/systemd
# podman run --runtime /usr/bin/crun --annotation=run.oci.systemd.force_cgroup_v1=/sys/fs/cgroup -
-rm -ti centos:7 /usr/lib/systemd/systemd
```

(JIRA:RHELPLAN-96940)

ANNEXE A. LISTE DES TICKETS PAR COMPOSANT

Les identifiants Bugzilla et JIRA sont listés dans ce document à titre de référence. Les bugs Bugzilla qui sont accessibles au public incluent un lien vers le ticket.

Composant	Billets
389-ds-base	BZ#2052527, BZ#2057063, BZ#2057066, BZ#1872451, BZ#2053204, BZ#2063140, BZ#2047175
NetworkManager	BZ#2068525, BZ#2059608, BZ#2030997, BZ#2079849, BZ#2097293, BZ#2029636, BZ#1894877, BZ#2151040
anaconda	BZ#2059414, BZ#2053710, BZ#2082132, BZ#2050140, BZ#1877697, BZ#1914955, BZ#1929105, BZ#1997832, BZ#2052938, BZ#2107346, BZ#2125542, BZ#2115783
ansible-collection-microsoft-sql	BZ#2066337
ansible-collection-redhat-rhel_mgmt	BZ#2112434
ansible-freeipa	BZ#2076567
bind	BZ#1984982
catatonit	BZ#2074193
chrony	BZ#2047415, BZ#2051441
clevis	BZ#2107078
cloud-init	BZ#1750862
cockpit-appstream	BZ#2030836
cockpit	BZ#2056786
cronie	BZ#2090691
crypto-policies	BZ#2102774, BZ#2070604
cyrus-sasl	BZ#1995600
device-mapper-multipath	BZ#2084365, BZ#2033080, BZ#2011699
distribution	BZ#2063773

Composant	Billets
dnf-plugins-core	BZ#2066646
dnf	BZ#2053014, BZ#2073510
dotnet7.0	BZ#2112027
dyninst	BZ#2057675
edk2	BZ#1935497
elfutils	BZ#2088774
fapolicyd	BZ#2100041, BZ#2054740, BZ#2070655
firefox	BZ#2013247
firewalld	BZ#2040689, BZ#2039542
frr	BZ#2069563
gcc-toolset-12-annobin	BZ#2077438
gcc-toolset-12-binutils	BZ#2077445
gcc-toolset-12-gcc	BZ#2077465
gcc-toolset-12-gdb	BZ#2077494
gcc	BZ#2063255
gdb	BZ#1870017
gdm	BZ#2097308
gimp	BZ#2047161
glibc	BZ#2033683, BZ#2096191, BZ#2063142, BZ#2077838, BZ#2085529, BZ#2003291, BZ#2091549
gnome-settings-daemon	BZ#2100467
gnupg2	BZ#2070722, BZ#2073567
gnutls	BZ#2042009

Composant	Billets
golang	BZ#2075169, BZ#2111072, BZ#2092016
grub2	BZ#2074761, BZ#2026579
grubby	BZ#1978226, BZ#1969362, BZ#2127453
httpd	BZ#2079939, BZ#2065677
ipa	BZ#747959, BZ#2091988, BZ#2083218, BZ#2100227, BZ#2084180, BZ#2084166, BZ#2069202, BZ#2057471, BZ#2124243, BZ#2089907
jmc-core	BZ#1980981
kdump-anaconda-addon	BZ#1959203, BZ#2017401
kernel-rt	BZ#2061574
kernel	JIRA :RHELPLAN-117713, BZ#2027894, BZ#2066451, BZ#2079368, BZ#2065226, BZ#2013413, BZ#2069045, BZ#2001936, BZ#2097188, BZ#2096127, BZ#2054379, BZ#2073541, BZ#2030922, BZ#1945040, BZ#2100898, BZ#2068432, BZ#2046472, BZ#1613522, BZ#1874182, BZ#1995338, BZ#1570255, BZ#2023416, BZ#2021672, BZ#2000616, BZ#2013650, BZ#2132480, BZ#2060150, BZ#2059545, BZ#2069758, BZ#1960467, BZ#2005173, BZ#2129288
kexec-tools	BZ#2064708, BZ#2065013
keylime	BZ#2138167, BZ#2140670, BZ#2142009
kmod-kvdo	BZ#2064802
kmod	BZ#2103605
krb5	BZ#2068935, BZ#2106043, BZ#2060798, BZ#2077450, BZ#2106296, BZ#2060421
libdnf	BZ#2108969
libnvme	BZ#2099619
libsepol	BZ#2069718, BZ#2079276
libvirt	BZ#2064194, BZ#2014487
libvpd	BZ#2051288

Composant	Billets
libxcrypt	BZ#2034569
llvm-toolset	BZ#2061041
lsvpd	BZ#2051289
lvm2	BZ#2038183
maven	BZ#2083112
mysql	BZ#1991500
nfs-utils	BZ#2081114
nmstate	BZ#2084474, BZ#2082043
nodejs	BZ#2083072
nss	BZ#2091905
nvme-cli	BZ#2090121
nvme-stas	BZ#1893841
open-vm-tools	BZ#2061193, BZ#2037657
opencryptoki	BZ#2044179
openscap	BZ#2109485
openssh	BZ#2066882, BZ#2087121, BZ#2056884
openssl	BZ#2060510, BZ#2053289, BZ#2066412, BZ#2063947, BZ#2004915, BZ#2058663, BZ#1975836, BZ#1681178, BZ#1685470, BZ#2060044, BZ#2071631
pacemaker	BZ#2121838, BZ#2072108
pause-container	BZ#2106816
pcre2	BZ#2086494
pcs	BZ#2024522, BZ#2054671, BZ#2058251, BZ#2058252, BZ#2058246, BZ#2058243, BZ#1301204

Composant	Billets
php	BZ#2070040
pki-core	BZ#2084181
podman	BZ#2097708, BZ#2027576, BZ#2069279
policycoreutils	BZ#2115242
powerpc-utils	BZ#1920964
ppc64-diag	BZ#2051286
procps-ng	BZ#2052536, BZ#2003033
pykickstart	BZ#2083269
qemu-kvm	BZ#2044218, BZ#1965079, BZ#1951814, BZ#2060839, BZ#2014229, BZ#2052424, BZ#1817965, BZ#1789206, BZ#2090712, BZ#2020146
rear	BZ#2111059, BZ#2097437, BZ#2115958, BZ#2083272, BZ#2120736, BZ#2119501
resource-agents	BZ#1826455
rhel-system-roles	BZ#2072385, BZ#2086965, BZ#2065337, BZ#2079622, BZ#2043010, BZ#2065383, BZ#2112145, BZ#2052081, BZ#2052086, BZ#2065392, BZ#2072742, BZ#2072745, BZ#2072746, BZ#2075119, BZ#2078989, BZ#2079627, BZ#2093423, BZ#2100292, BZ#2100942, BZ#2115154, BZ#2115157, BZ#2115152, BZ#2051737, BZ#2065382, BZ#2065394, BZ#2115886, BZ#2100605, BZ#2060523, BZ#2060525, BZ#2065393, BZ#2070462, BZ#2083376, BZ#2083410, BZ#2100286, BZ#2109998, BZ#2115156, BZ#2071804, BZ#2100294, BZ#1999770
rsyslog	BZ#2064318
rust	BZ#2075337
s390utils	BZ#1870699, BZ#1932480
samba	BZ#2077487
sblim-wbemcli	BZ#2083577
scap-security-guide	BZ#2070563, BZ#2120978, BZ#2038978
selinux-policy	BZ#1965013, BZ#2081425, BZ#2076681, BZ#2064274

Composant	Billets
sos	BZ#1869561
sssd	BZ#1978119, BZ#2065693, BZ#2056482
stalld	BZ#2107275
stratisd	BZ#1990905, BZ#2040352, BZ#2039960, BZ#2007018, BZ#2005110, BZ#2041558
subscription-manager	BZ#2092014, BZ#2136694
systemd	BZ#2018112
systemtap	BZ#2083727
tigervnc	BZ#2060308
tpm2-tools	BZ#2090748
tuned	BZ#2093847
ubi8-container	BZ#2120378
udisks2	BZ#1983602
unbound	BZ#2087120, BZ#2071543, BZ#2070495
valgrind	BZ#1993976
virt-who	BZ#2054504
virtio-win	BZ#1969724, BZ#2084003
whois	BZ#2054043
xmlstarlet	BZ#2069689
xorg-x11-server	BZ#1894612

Composant	Billets
autres	JIRA:RHELPLAN-92522, BZ#2125549, BZ#2128016, BZ#1937031, JIRA:RHELPLAN-121982, JIRA:RHELPLAN-95456, JIRA:RHELPLAN-122321, JIRA:RHELPLAN-118462, JIRA :RHELPLAN-101140, JIRA:RHELPLAN-132023, JIRA:RHELPLAN-123369, JIRA:RHELPLAN-117109, JIRA:RHELPLAN-130379, BZ#2049492, JIRA:RHELPLAN-130376, JIRA :RHELPLAN-122735, BZ#2070793, BZ#2122716, JIRA:RHELPLAN-123368, JIRA:RHELPLAN-135601, JIRA:RHELPLAN-135602, BZ#2139877, JIRA:RHELPLAN-122776, JIRA :RHELPLAN-121180, BZ#2094015, JIRA:RHELPLAN-109067, JIRA:RHELPLAN-115603, JIRA:RHELPLAN-65217, BZ#2020529, BZ#2030412, BZ#2046653, JIRA:RHELPLAN-103993, JIRA:RHELPLAN-122345, JIRA :RHELPLAN-129327, JIRA:RHELPLAN-74672, BZ#1927780, JIRA:RHELPLAN-110763, BZ#1935544, BZ#2089200, JIRA:RHELPLAN-15509, JIRA:RHELPLAN-99136, JIRA :RHELPLAN-103232, BZ#1899167, BZ#1979521, JIRA:RHELPLAN-100087, JIRA:RHELPLAN-100639, JIRA:RHELPLAN-10304, BZ#2058153, JIRA:RHELPLAN-113995, JIRA:RHELPLAN-121048, JIRA :RHELPLAN-98983, JIRA:RHELPLAN-131882, JIRA:RHELPLAN-137660, BZ#1640697, BZ#1697896, BZ#2047713, JIRA:RHELPLAN-96940, JIRA:RHELPLAN-117234, JIRA :RHELPLAN-119001, JIRA:RHELPLAN-119852, BZ#2077767, BZ#2053598, BZ#2082303, JIRA:RHELPLAN-121049, JIRA:RHELPLAN-109613, JIRA:RHELPLAN-135600, BZ#2149172

ANNEXE B. HISTORIQUE DES RÉVISIONS

0.0-10

Mercredi 17 mai 2023, Gabriela Fialová(gfialova@redhat.com)

- Mise à jour du fichier deprecated-packages.adoc avec les informations relatives à la fin de vie.

0.0-9

Jeu. 27 avril 2023, Gabriela Fialová(gfialova@redhat.com)

- Ajout d'un problème connu JIRA:RHELPLAN-155168 (Identity Management).

0.0-8

Tue Apr 25, 2023, Lucie Vařáková(lvarakova@redhat.com)

- Ajout d'un problème connu BZ#2180665 (Kernel).

0.0-7

Mon Feb 20, 2023, Gabriela Fialová(gfialova@redhat.com)

- Ajout d'informations sur les environnements SAP pour la mise à niveau en place de RHEL 8 à RHEL 9.

0.0-6

Thu Feb 16, 2023, Gabriela Fialová(gfialova@redhat.com)

- Mise à jour d'un problème connu BZ#2132480 (Kernel).

0.0-5

Tue Feb 14, 2023, Gabriela Fialová(gfialova@redhat.com)

- Une petite modification de formatage a été apportée à la section Changements importants dans les paramètres externes du noyau.

0.0-4

Mar. 14 Fév. 2023, Marc Muehlfeld(mmuehlfeld@redhat.com)

- Ajout d'une amélioration BZ#2144898 (Mise en réseau).

0.0-3

Mer Déc 07, 2022, Gabriela Fialová(gfialova@redhat.com)

- Le flux de modules **nodejs:18** BZ#2083072 a été déplacé des aperçus technologiques vers les fonctionnalités entièrement prises en charge (langages de programmation dynamiques, serveurs web et de base de données).

0.0-2

Mercredi 16 novembre 2022, Gabriela Fialová(gfialova@redhat.com)

- Publication des notes de mise à jour de Red Hat Enterprise Linux 9.1.

0.0-1

Mercredi 28 septembre 2022, Gabriela Fialová(gfialova@redhat.com)

- Publication des notes de mise à jour de Red Hat Enterprise Linux 9.1 Beta.