Red Hat Enterprise Linux 9

Déploiement de RHEL 9 sur Amazon Web Services

Obtention d'images système RHEL et création d'instances RHEL sur AWS

Last Updated: 2024-02-20

Red Hat Enterprise Linux 9 Déploiement de RHEL 9 sur Amazon Web Services

Obtention d'images système RHEL et création d'instances RHEL sur AWS

Notice légale

Résumé

Pour utiliser Red Hat Enterprise Linux (RHEL) dans un environnement de cloud public, vous pouvez créer et déployer des images système RHEL sur diverses plateformes de cloud, y compris Amazon Web Services (AWS). Vous pouvez également créer et configurer un cluster Red Hat High Availability (HA) sur AWS. Les chapitres suivants fournissent des instructions pour créer des instances RHEL en nuage et des clusters HA sur AWS. Ces processus comprennent l'installation des packages et des agents requis, la configuration de la clôture et l'installation des agents de ressources réseau.

Table des matières

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 le bouton **Add Feedback** qui apparaît près du texte en surbrillance.
4. Ajoutez vos commentaires et cliquez sur **Submit**.

Soumettre des commentaires via Bugzilla (compte requis)

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. Saisissez votre suggestion d'amélioration dans le champ **Description**. Incluez des liens vers les parties pertinentes de la documentation.
5. Cliquez sur **Submit Bug**.

CHAPITRE 1. PRÉSENTATION DE RHEL SUR LES PLATES-FORMES DE CLOUD PUBLIC

Les plateformes de cloud public fournissent des ressources informatiques en tant que service. Au lieu d'utiliser du matériel sur site, vous pouvez exécuter vos charges de travail informatiques, y compris les systèmes Red Hat Enterprise Linux (RHEL), en tant qu'instances de cloud public.

Pour en savoir plus sur RHEL sur les plates-formes de cloud public, voir :

- Avantages de l'utilisation de RHEL dans un nuage public
- Cas d'utilisation de RHEL dans les nuages publics
- Problèmes fréquents lors de la migration vers un nuage public
- Obtenir RHEL pour les déploiements dans les nuages publics
- Méthodes de création d'instances de cloud RHEL

1.1. AVANTAGES DE L'UTILISATION DE RHEL DANS UN NUAGE PUBLIC

RHEL en tant qu'instance cloud située sur une plateforme cloud publique présente les avantages suivants par rapport à RHEL sur des systèmes physiques ou des machines virtuelles (VM) sur site :

- **Flexible and fine-grained allocation of resources**
  Une instance cloud de RHEL s'exécute en tant que VM sur une plateforme cloud, ce qui signifie généralement une grappe de serveurs distants gérée par le fournisseur du service cloud. Par conséquent, l'attribution de ressources matérielles à l'instance, telles qu'un type spécifique d'unité centrale ou de stockage, se fait au niveau du logiciel et est facilement personnalisable.

  Par rapport à un système RHEL local, vous n'êtes pas non plus limité par les capacités de votre hôte physique. Au lieu de cela, vous pouvez choisir parmi une variété de fonctionnalités, sur la base de la sélection proposée par le fournisseur de services en nuage.

- **Space and cost efficiency**
  Vous n'avez pas besoin de posséder des serveurs sur site pour héberger vos charges de travail en nuage. Vous n'avez donc pas besoin de l'espace, de l'énergie et de la maintenance associés au matériel physique.

  En revanche, sur les plateformes de cloud public, vous payez directement le fournisseur de cloud pour l'utilisation d'une instance de cloud. Le coût est généralement basé sur le matériel alloué à l'instance et sur le temps que vous passez à l'utiliser. Vous pouvez donc optimiser vos coûts en fonction de vos besoins.

- **Software-controlled configurations**
  L'ensemble de la configuration d'une instance en nuage est sauvegardé sous forme de données sur la plateforme en nuage et est contrôlé par un logiciel. Vous pouvez donc facilement créer, supprimer, cloner ou migrer l'instance. Une instance en nuage est également gérée à distance dans une console du fournisseur de nuage et est connectée par défaut à un stockage à distance.

  En outre, vous pouvez à tout moment sauvegarder l'état actuel d'une instance cloud sous la forme d'un instantané. Ensuite, vous pouvez charger l'instantané pour restaurer l'instance à l'état sauvegardé.

- **Separation from the host and software compatibility**

Comme pour une VM locale, le système d'exploitation invité RHEL sur une instance cloud s'exécute sur un noyau virtualisé. Ce noyau est distinct du système d'exploitation hôte et du système *client* que vous utilisez pour vous connecter à l'instance.

Par conséquent, n'importe quel système d'exploitation peut être installé sur l'instance de cloud. Cela signifie que sur une instance de cloud public RHEL, vous pouvez exécuter des applications spécifiques à RHEL qui ne peuvent pas être utilisées sur votre système d'exploitation local.

En outre, même si le système d'exploitation de l'instance devient instable ou est compromis, votre système client n'est en aucun cas affecté.

Ressources supplémentaires

- Qu'est-ce que l'informatique dématérialisée ?
- Qu'est-ce qu'un hyperscaler ?
- Types d'informatique en nuage
- Cas d'utilisation de RHEL dans les nuages publics
- Obtenir RHEL pour les déploiements dans les nuages publics
- Pourquoi utiliser Linux sur AWS ?

1.2. CAS D'UTILISATION DE RHEL DANS LES NUAGES PUBLICS

Le déploiement sur un cloud public présente de nombreux avantages, mais n'est pas forcément la solution la plus efficace dans tous les cas de figure. Si vous envisagez de migrer vos déploiements RHEL vers le cloud public, demandez-vous si votre cas d'utilisation bénéficiera des avantages du cloud public.

Beneficial use cases

- Le déploiement d'instances de cloud public est très efficace pour augmenter et réduire de manière flexible la puissance informatique active de vos déploiements, également connue sous les noms de *scaling up* et *scaling down*. Par conséquent, l'utilisation de RHEL sur le cloud public est recommandée dans les scénarios suivants :
  - Clusters avec des charges de travail de pointe élevées et de faibles exigences de performance générale. La mise à l'échelle en fonction de vos besoins peut s'avérer très efficace en termes de coûts de ressources.
  - Mise en place ou extension rapide de vos clusters. Cela permet d'éviter les coûts initiaux élevés liés à la mise en place de serveurs locaux.
- Les instances en nuage ne sont pas affectées par ce qui se passe dans votre environnement local. Vous pouvez donc les utiliser pour la sauvegarde et la reprise après sinistre.

Potentially problematic use cases

- Vous utilisez un environnement existant qui ne peut pas être adapté. La personnalisation d'une instance en nuage pour répondre aux besoins spécifiques d'un déploiement existant peut ne pas être rentable par rapport à votre plateforme hôte actuelle.
- Votre budget est limité. Le maintien de votre déploiement dans un centre de données local offre généralement moins de flexibilité mais plus de contrôle sur les coûts maximaux des ressources que le nuage public.

Prochaines étapes

- Obtenir RHEL pour les déploiements dans les nuages publics

Ressources supplémentaires

- Dois-je migrer mon application vers l'informatique dématérialisée ? Voici comment décider.

1.3. PROBLÈMES FRÉQUENTS LORS DE LA MIGRATION VERS UN NUAGE PUBLIC

Le transfert de vos charges de travail RHEL d'un environnement local vers une plateforme de cloud public peut susciter des inquiétudes quant aux changements qu'il implique. Voici les questions les plus fréquemment posées.

Will my RHEL work differently as a cloud instance than as a local virtual machine?

À la plupart des égards, les instances RHEL sur une plateforme de cloud public fonctionnent de la même manière que les machines virtuelles RHEL sur un hôte local, tel qu'un serveur sur site. Les exceptions notables sont les suivantes :

- Au lieu d'interfaces d'orchestration privées, les instances de cloud public utilisent des interfaces de console spécifiques au fournisseur pour gérer vos ressources de cloud.
- Certaines fonctionnalités, telles que la virtualisation imbriquée, peuvent ne pas fonctionner correctement. Si une fonctionnalité spécifique est essentielle pour votre déploiement, vérifiez au préalable sa compatibilité avec le fournisseur de cloud public que vous avez choisi.

Will my data stay safe in a public cloud as opposed to a local server?

Les données de vos instances de cloud RHEL sont votre propriété et votre fournisseur de cloud public n'y a pas accès. En outre, les principaux fournisseurs de cloud prennent en charge le cryptage des données en transit, ce qui améliore la sécurité des données lors de la migration de vos machines virtuelles vers le cloud public.

La sécurité générale de vos instances de cloud public RHEL est gérée comme suit :

- Votre fournisseur de cloud public est responsable de la sécurité de l'hyperviseur du cloud
- Red Hat fournit les fonctions de sécurité des systèmes d'exploitation invités RHEL dans vos instances
- Vous gérez les paramètres et les pratiques de sécurité spécifiques de votre infrastructure en nuage

What effect does my geographic region have on the functionality of RHEL public cloud instances?

Vous pouvez utiliser des instances RHEL sur une plateforme de cloud public quelle que soit votre situation géographique. Par conséquent, vous pouvez exécuter vos instances dans la même région que votre serveur sur site.

Toutefois, l'hébergement de vos instances dans une région physiquement éloignée peut entraîner une latence élevée lors de leur fonctionnement. En outre, selon le fournisseur de cloud public, certaines régions peuvent offrir des fonctionnalités supplémentaires ou être plus rentables. Avant de créer vos instances RHEL, examinez les propriétés des régions d'hébergement disponibles pour le fournisseur de cloud choisi.

1.4. OBTENIR RHEL POUR LES DÉPLOIEMENTS DANS LES NUAGES PUBLICS

Pour déployer un système RHEL dans un environnement de cloud public :

1. Sélectionnez le fournisseur de services en nuage optimal pour votre cas d'utilisation, en fonction de vos besoins et de l'offre actuelle sur le marché.
   Les fournisseurs de services en nuage actuellement certifiés pour l'exécution d'instances RHEL sont les suivants :

   - Amazon Web Services (AWS)
   - Google Cloud Platform (GCP)
   - Microsoft Azure

   NOTE

   Ce document traite spécifiquement du déploiement de RHEL sur AWS.

2. Créez une instance de cloud RHEL sur la plateforme de cloud choisie. Pour plus d'informations, voir Méthodes de création d'instances de cloud RHEL .
3. Pour maintenir votre déploiement RHEL à jour, utilisez Red Hat Update Infrastructure (RHUI).

Ressources supplémentaires

- Documentation RHUI
- Red Hat Open Hybrid Cloud

1.5. MÉTHODES DE CRÉATION D'INSTANCES DE CLOUD RHEL

Pour déployer une instance RHEL sur une plateforme de cloud public, vous pouvez utiliser l'une des méthodes suivantes :

Create a system image of RHEL and import it to the cloud platform.

- Pour créer l'image système, vous pouvez utiliser le générateur d'images RHEL ou construire l'image manuellement.
- Cette méthode utilise votre abonnement RHEL existant et est également appelée *bring your own subscription* (BYOS).
- Vous payez à l'avance un abonnement annuel et vous pouvez utiliser votre réduction client Red Hat.
- Votre service clientèle est assuré par Red Hat.
- Pour créer plusieurs images de manière efficace, vous pouvez utiliser l'outil **cloud-init**.

Purchase a RHEL instance directly from the cloud provider marketplace.

- Vous post-payez un taux horaire pour l'utilisation du service. Cette méthode est donc également appelée *pay as you go* (PAYG).
- Votre service clientèle est assuré par le fournisseur de la plateforme en nuage.

NOTE

Pour des instructions détaillées sur l'utilisation de différentes méthodes pour déployer des instances RHEL sur Amazon Web Services, voir les chapitres suivants de ce document.

Ressources supplémentaires

- Qu'est-ce qu'une image en or ?
- Configurer et gérer cloud-init pour RHEL 9

CHAPITRE 2. CRÉATION ET TÉLÉCHARGEMENT D'IMAGES AMI AWS

Pour utiliser votre image système RHEL personnalisée dans le nuage Amazon Web Services (AWS), créez l'image système avec Image Builder en utilisant le type de sortie correspondant, configurez votre système pour le téléchargement de l'image et téléchargez l'image sur votre compte AWS.

2.1. PRÉPARATION DU TÉLÉCHARGEMENT DES IMAGES AWS AMI

Avant de télécharger une image AWS AMI, vous devez configurer un système pour télécharger les images.

Conditions préalables

- Vous devez avoir un ID de clé d'accès configuré dans le gestionnaire de compte AWS IAM .
- Vous devez avoir préparé un seau S3 accessible en écriture.

Procédure

1. Installez Python 3 et l'outil **pip**:

```
# dnf install python3
# dnf install python3-pip
```

2. Installez les outils en ligne de commande AWS avec **pip**:

```
# pip3 install awscli
```

3. Exécutez la commande suivante pour définir votre profil. Le terminal vous invite à fournir vos informations d'identification, votre région et le format de sortie :

```
$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]:
Default output format [None]:
```

4. Définissez un nom pour votre seau et utilisez la commande suivante pour créer un seau :

```
$ BUCKET=bucketname
$ aws s3 mb s3://$BUCKET
```

Remplacez *bucketname* par le nom du seau. Il doit s'agir d'un nom unique au niveau mondial. Le résultat est la création d'un seau.

5. Pour autoriser l'accès au seau S3, créez un rôle S3 **vmimport** dans le système de gestion des identités et des accès (IAM) d'AWS, si vous ne l'avez pas déjà fait par le passé :

```
$ printf '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service":
"vmie.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition": { "StringEquals":{
"sts:Externalid": "vmimport" } } } ] }' > trust-policy.json
$ printf '{ "Version":"2012-10-17", "Statement":[ { "Effect":"Allow", "Action":[
```

```
"s3:GetBucketLocation", "s3:GetObject", "s3:ListBucket" ], "Resource":[ "arn:aws:s3:::%s",
"arn:aws:s3:::%s/*" ] }, { "Effect":"Allow", "Action":[ "ec2:ModifySnapshotAttribute",
"ec2:CopySnapshot", "ec2:RegisterImage", "ec2:Describe*" ], "Resource":"*" } ] }' $BUCKET
$BUCKET > role-policy.json
$ aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-
policy.json
$ aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document
file://role-policy.json
```

Ressources supplémentaires

- Utiliser des commandes de haut niveau (s3) avec la CLI AWS

2.2. TÉLÉCHARGEMENT D'UNE IMAGE AMI SUR AWS À L'AIDE DE LA CLI

Vous pouvez utiliser le constructeur d'images pour créer des images **ami** et les envoyer directement au fournisseur de services Amazon AWS Cloud à l'aide de l'interface de programmation.

Conditions préalables

- Vous avez un **Access Key ID** configuré dans le gestionnaire de compte AWS IAM.
- Vous avez préparé un seau S3 accessible en écriture.
- Vous disposez d'un plan défini.

Procédure

1. À l'aide d'un éditeur de texte, créez un fichier de configuration avec le contenu suivant :

```
provider = "aws"

[settings]
accessKeyID = "AWS_ACCESS_KEY_ID"
secretAccessKey = "AWS_SECRET_ACCESS_KEY"
bucket = "AWS_BUCKET"
region = "AWS_REGION"
key = "IMAGE_KEY"
```

   Remplacez les valeurs des champs par vos informations d'identification pour **accessKeyID**, **secretAccessKey**, **bucket**, et **region**. La valeur **IMAGE_KEY** est le nom de votre image VM à télécharger sur EC2.

2. Enregistrez le fichier sous *CONFIGURATION-FILE*.toml et fermez l'éditeur de texte.

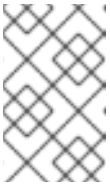
3. Commencez la composition :

```
# composer-cli composer start BLUEPRINT-NAME IMAGE-TYPE IMAGE_KEY
CONFIGURATION-FILE .toml
```

   Remplacer :

   - *BLUEPRINT-NAME* avec le nom du plan que vous avez créé

- *IMAGE-TYPE* avec le type d'image **ami**.
- *IMAGE_KEY* avec le nom de l'image de la VM à télécharger sur EC2.
- *CONFIGURATION-FILE*.toml avec le nom du fichier de configuration du fournisseur de cloud.



NOTE

Vous devez avoir les paramètres IAM corrects pour le panier dans lequel vous allez envoyer votre image personnalisée. Vous devez définir une politique pour votre panier avant de pouvoir y télécharger des images.

4. Vérifiez l'état de la création de l'image et téléchargez-la sur AWS :

```
# composer-cli compose status
```

Une fois le processus de téléchargement de l'image terminé, vous pouvez voir l'état "FINISHED".

Vérification

Pour confirmer que le téléchargement de l'image a réussi :

1. Accédez à EC2 dans le menu et sélectionnez la région correcte dans la console AWS. L'image doit avoir le statut **available**, pour indiquer qu'elle a été téléchargée avec succès.
2. Dans le tableau de bord, sélectionnez votre image et cliquez sur **Launch**.

Ressources complémentaires

- Rôle de service requis pour importer une VM

2.3. POUSSER DES IMAGES VERS AWS CLOUD AMI

Vous pouvez envoyer l'image de sortie que vous créez directement au fournisseur de services **Amazon AWS Cloud AMI**.

Conditions préalables

- Vous devez avoir accès au système par l'intermédiaire du groupe d'utilisateurs **root** ou **wheel**.
- Vous avez ouvert l'interface de construction d'images de la console web RHEL dans un navigateur.
- Vous avez créé un modèle. Voir Création d'un modèle de constructeur d'images dans l'interface de la console web.
- Vous devez avoir un ID de clé d'accès configuré dans le gestionnaire de compte AWS IAM.
- Vous devez avoir préparé un seau S3 accessible en écriture.

Procédure

1. Cliquez sur le site **blueprint name**.

2. Sélectionnez l'onglet **Images**.

3. Cliquez sur **Créer une image** pour créer votre image personnalisée.
   Une fenêtre pop-up s'ouvre.

   a. Dans la liste du menu déroulant **Type**, sélectionnez **Amazon Machine Image Disk (.raw)**.

   b. Cochez la case **Upload to AWS** pour télécharger votre image dans le nuage AWS et cliquez sur **Suivant**.

   c. Pour authentifier votre accès à AWS, saisissez votre **AWS access key ID** et **AWS secret access key** dans les champs correspondants. Cliquez sur **Suivant**.

   NOTE

   Vous ne pouvez consulter votre clé d'accès secrète AWS que lorsque vous créez un nouvel ID de clé d'accès. Si vous ne connaissez pas votre clé secrète, générez un nouvel ID de clé d'accès.

   d. Saisissez le nom de l'image dans le champ **Image name**, le nom du panier Amazon dans le champ **Amazon S3 bucket name** et le champ **AWS region** pour le panier dans lequel vous allez ajouter votre image personnalisée. Cliquez sur **Suivant**.

   e. Vérifiez les informations et cliquez sur **Terminer**.
   En option, vous pouvez cliquer sur **Retour** pour modifier tout détail incorrect.

   NOTE

   Vous devez disposer des paramètres IAM corrects pour le panier dans lequel vous allez envoyer votre image personnalisée. Cette procédure utilise l'importation et l'exportation IAM, vous devez donc configurer **a policy** pour votre panier avant de pouvoir y télécharger des images. Pour plus d'informations, voir Autorisations requises pour les utilisateurs IAM .

4. Une petite fenêtre contextuelle en haut à droite vous informe de la progression de l'enregistrement. Elle indique également que la création d'une image a été lancée, l'état d'avancement de cette création et le téléchargement ultérieur vers le nuage AWS.
   Une fois le processus terminé, vous pouvez consulter l'état de **Image build complete**.

5. Cliquez sur Service→EC2 dans le menu et choisissez la bonne région dans la console AWS.
   L'image doit avoir le statut **Available**, pour indiquer qu'elle est téléchargée.

6. Dans le tableau de bord, sélectionnez votre image et cliquez sur **Launch**.

7. Une nouvelle fenêtre s'ouvre. Choisissez un type d'instance en fonction des ressources dont vous avez besoin pour démarrer votre image. Cliquez sur **Review et Launch**.

8. Passez en revue les détails de votre début d'instance. Vous pouvez éditer chaque section si vous avez besoin de faire des changements. Cliquez sur **Launch**

9. Avant de démarrer l'instance, sélectionnez une clé publique pour y accéder.
   Vous pouvez utiliser la paire de clés que vous possédez déjà ou en créer une nouvelle.

   Suivez les étapes suivantes pour créer une nouvelle paire de clés dans EC2 et l'attacher à la nouvelle instance.

a. Dans la liste du menu déroulant, sélectionnez **Create a new key pair**.

b. Entrez le nom de la nouvelle paire de clés. Une nouvelle paire de clés est générée.

c. Cliquez sur **Download Key Pair** pour enregistrer la nouvelle paire de clés sur votre système local.

10. Ensuite, vous pouvez cliquer sur **Launch Instance** pour démarrer votre instance.
Vous pouvez vérifier l'état de l'instance, qui s'affiche comme suit : **Initializing**.

11. Une fois que l'état de l'instance est **running**, le bouton **Connecter** devient disponible.

12. Cliquez sur **Connecter**. Une fenêtre contextuelle s'affiche avec des instructions sur la manière de se connecter en utilisant SSH.

a. Sélectionnez **A standalone SSH client** comme méthode de connexion préférée et ouvrez un terminal.

b. À l'endroit où vous stockez votre clé privée, assurez-vous qu'elle est publiquement visible pour que SSH fonctionne. Pour ce faire, exécutez la commande suivante

```
$ chmod 400 <nom-de-votre-instance.pem>_
```

c. Connectez-vous à votre instance en utilisant son DNS public :

```
$ ssh -i "<_votre-nom-d'instance.pem_"> ec2-user@<_votre-adresse-IP-d'instance_>
```

d. Tapez **yes** pour confirmer que vous voulez continuer à vous connecter.
Par conséquent, vous êtes connecté à votre instance en utilisant SSH.

Vérification

1. Vérifiez si vous pouvez effectuer une action lorsque vous êtes connecté à votre instance à l'aide de SSH.

Ressources supplémentaires

- Ouvrir un cas sur le portail client de Red Hat
- Connexion à votre instance Linux à l'aide de SSH
- Cas d'assistance en cours

CHAPITRE 3. DÉPLOYER UNE IMAGE RED HAT ENTERPRISE LINUX EN TANT QU'INSTANCE EC2 SUR AMAZON WEB SERVICES

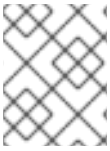
Vous disposez d'un certain nombre d'options pour déployer une image Red Hat Enterprise Linux (RHEL) 9 en tant qu'instance EC2 sur Amazon Web Services (AWS). Ce chapitre aborde les options de choix d'une image et énumère ou fait référence à la configuration requise pour votre système hôte et votre machine virtuelle (VM). Ce chapitre fournit également des procédures pour créer une VM personnalisée à partir d'une image ISO, la télécharger sur EC2 et lancer une instance EC2.

Pour déployer Red Hat Enterprise Linux 9 (RHEL 9) en tant qu'instance EC2 sur Amazon Web Services (AWS), suivez les informations ci-dessous. Ce chapitre :

- Examine les options qui s'offrent à vous pour le choix d'une image
- Liste ou fait référence à la configuration requise pour votre système hôte et votre machine virtuelle (VM)
- Fournit des procédures pour créer une VM personnalisée à partir d'une image ISO, la télécharger sur EC2 et lancer une instance EC2

IMPORTANT

Bien que vous puissiez créer une VM personnalisée à partir d'une image ISO, Red Hat vous recommande d'utiliser le produit Red Hat Image Builder pour créer des images personnalisées à utiliser sur des fournisseurs de cloud spécifiques. Avec Image Builder, vous pouvez créer et télécharger une image de machine Amazon (AMI) au format **ami**. Voir Composer une image système RHEL personnalisée pour plus d'informations.



NOTE

Pour une liste des produits Red Hat que vous pouvez utiliser en toute sécurité sur AWS, voir Red Hat sur Amazon Web Services .

Conditions préalables

- Créez un compte sur le portail client de Red Hat .
- Inscrivez-vous à AWS et configurez vos ressources AWS. Pour plus d'informations, reportez-vous à la section Configuration d'Amazon EC2 .

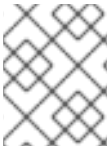
3.1. OPTIONS D'IMAGES RED HAT ENTERPRISE LINUX SUR AWS

Le tableau suivant énumère les choix d'images et indique les différences entre les options d'images.

Tableau 3.1. Options d'images

Option d'image	Abonnements	Exemple de scénario	Considérations

Option d'image	Abonnements	Exemple de scénario	Considérations
Déployez une image Red Hat Gold.	Utilisez vos abonnements Red Hat existants.	Sélectionnez une Red Hat Gold Image sur AWS. Pour plus de détails sur les Gold Images et la manière d'y accéder sur Azure, consultez le Guide de référence de Red Hat Cloud Access.	L'abonnement comprend le coût du produit Red Hat ; vous payez Amazon pour tous les autres coûts d'instance. Red Hat fournit une assistance directe pour les images Cloud Access.
Déployez une image personnalisée que vous déplacez sur AWS.	Utilisez vos abonnements Red Hat existants.	Téléchargez votre image personnalisée et joignez vos abonnements.	L'abonnement comprend le coût du produit Red Hat ; vous payez Amazon pour tous les autres coûts d'instance. Red Hat fournit une assistance directe pour les images RHEL personnalisées.
Déployer une image Amazon existante qui inclut RHEL.	Les images AWS EC2 incluent un produit Red Hat.	Sélectionnez une image RHEL lorsque vous lancez une instance sur la console de gestion AWS, ou choisissez une image sur la place de marché AWS.	Vous payez Amazon à l'heure sur la base du modèle *pay-as-you-go*. Ces images sont appelées "images à la demande". Amazon fournit une assistance pour les images à la demande. Red Hat fournit des mises à jour aux images. AWS met les mises à jour à disposition via l'infrastructure de mise à jour de Red Hat (RHUI).



NOTE

Vous pouvez créer une image personnalisée pour AWS à l'aide de Red Hat Image Builder. Voir Composer une image système RHEL personnalisée pour plus d'informations.

IMPORTANT

Vous ne pouvez pas convertir une instance à la demande en une instance RHEL personnalisée. Pour passer d'une image à la demande à une image personnalisée RHEL *bring-your-own-subscription* (BYOS) :

1. Créez une nouvelle instance RHEL personnalisée et migrez les données de votre instance à la demande.
2. Annulez votre instance à la demande après avoir migré vos données pour éviter une double facturation.

Ressources supplémentaires

- Composition d'une image système RHEL personnalisée
- Console de gestion AWS
- Place de marché AWS

3.2. COMPRENDRE LES IMAGES DE BASE

Cette section contient des informations sur l'utilisation d'images de base préconfigurées et leurs paramètres de configuration.

3.2.1. Utilisation d'une image de base personnalisée

Pour configurer manuellement une machine virtuelle (VM), créez d'abord une image VM de base (starter). Vous pouvez ensuite modifier les paramètres de configuration et ajouter les paquets dont la VM a besoin pour fonctionner sur le nuage. Vous pouvez apporter des modifications supplémentaires à la configuration pour votre application spécifique après avoir téléchargé l'image.

Ressources supplémentaires

- Red Hat Enterprise Linux

3.2.2. Paramètres de configuration de la machine virtuelle

Les VM du nuage doivent avoir les paramètres de configuration suivants.

Tableau 3.2. Paramètres de configuration de la VM

Paramètres	Recommandation
ssh	ssh doit être activé pour permettre l'accès à distance à vos machines virtuelles.
dhcp	L'adaptateur virtuel primaire doit être configuré pour dhcp.

3.3. CRÉATION D'UNE VM DE BASE À PARTIR D'UNE IMAGE ISO

Suivez les procédures de cette section pour créer une image de base RHEL 9 à partir d'une image ISO.

Conditions préalables

- La virtualisation est activée sur votre machine hôte.
- Vous avez téléchargé la dernière image ISO de Red Hat Enterprise Linux à partir du portail client de Red Hat et déplacé l'image sur **/var/lib/libvirt/images**.

3.3.1. Création d'une VM à partir de l'image ISO RHEL

Procédure

1. Assurez-vous d'avoir activé la virtualisation de votre machine hôte. Voir Activation de la virtualisation dans RHEL 9 pour plus d'informations et de procédures.
2. Créez et démarrez une VM Red Hat Enterprise Linux de base. Pour obtenir des instructions, voir Création de machines virtuelles.
   a. Si vous utilisez la ligne de commande pour créer votre VM, veillez à définir la mémoire et les processeurs par défaut en fonction de la capacité souhaitée pour la VM. Définissez votre interface réseau virtuelle sur **virtio**.
   Par exemple, la commande suivante crée une VM **kvmtest** à l'aide de l'image **/home/username/Downloads/rhel9.iso**:

```
# virt-install \
    --name kvmtest --memory 2048 --vcpus 2 \
    --cdrom /home/username/Downloads/rhel9.iso,bus=virtio \
    --os-variant=rhel9.0
```

   b. Si vous utilisez la console web pour créer votre machine virtuelle, suivez la procédure décrite dans la section Création de machines virtuelles à l'aide de la console web , avec les mises en garde suivantes :
      - Ne pas vérifier **Immediately Start VM**.
      - Modifiez la taille de votre site **Memory** en fonction de vos préférences.
      - Avant de commencer l'installation, assurez-vous que vous avez changé **Model** sous **Virtual Network Interface Settings** en **virtio** et changez votre **vCPUs** en fonction des paramètres de capacité que vous souhaitez pour la VM.

3.3.2. Terminer l'installation de RHEL

Effectuez les étapes suivantes pour terminer l'installation et activer l'accès root une fois la VM lancée.

Procédure

1. Choisissez la langue que vous souhaitez utiliser pendant la procédure d'installation.
2. Sur la vue **Installation Summary**:
   a. Cliquez sur **Software Selection** et cochez **Minimal Install**.
   b. Cliquez sur **Done**.
   c. Cliquez sur **Installation Destination** et cochez **Custom** sous **Storage Configuration**.

- Vérifiez qu'il y a au moins 500 Mo pour /**boot**. Vous pouvez utiliser l'espace restant pour la racine /.
- Les partitions standard sont recommandées, mais vous pouvez utiliser la gestion des volumes logiques (LVM).
- Vous pouvez utiliser xfs, ext4 ou ext3 pour le système de fichiers.
- Cliquez sur **Done** lorsque vous avez terminé les modifications.

3. Cliquez sur **Begin Installation**.
4. Définir un **Root Password**. Créer d'autres utilisateurs le cas échéant.
5. Redémarrez la VM et connectez-vous en tant que **root** une fois l'installation terminée.
6. Configurer l'image.
   a. Enregistrez la VM et activez le référentiel Red Hat Enterprise Linux 9.

```
# subscription-manager register --auto-attach
```

   b. Assurez-vous que le paquetage **cloud-init** est installé et activé.

```
# dnf install cloud-init
# systemctl enable --now cloud-init.service
```

7. **Important: This step is only for VMs you intend to upload to AWS.**
   a. Pour les machines virtuelles AMD64 ou Intel 64 (x86_64), installez les pilotes **nvme**, **xen-netfront** et **xen-blkfront**.

```
# dracut -f --add-drivers "nvme xen-netfront xen-blkfront"
```

   b. Pour les machines virtuelles ARM 64 (aarch64), installez le pilote **nvme**.

```
# dracut -f --add-drivers "nvme"
```

   L'inclusion de ces pilotes élimine la possibilité d'un délai d'attente pour Dracut.

   Vous pouvez également ajouter les pilotes à /**etc/dracut.conf.d**/, puis saisir **dracut -f** pour écraser le fichier **initramfs** existant.

8. Mettez la VM hors tension.

Ressources supplémentaires

- Comprendre le rattachement automatique des abonnements sur le portail client
- Introduction à cloud-init

3.4. TÉLÉCHARGEMENT DE L'IMAGE RED HAT ENTERPRISE LINUX SUR AWS

Suivez les procédures de cette section pour télécharger votre image sur AWS.

3.4.1. Installation de l'interface de programmation AWS

La plupart des procédures requises pour gérer les clusters HA dans AWS incluent l'utilisation de la CLI AWS. Suivez les étapes suivantes pour installer la CLI AWS.

Conditions préalables

- Vous avez créé un identifiant de clé d'accès AWS et une clé d'accès secrète AWS, et vous y avez accès. Pour obtenir des instructions et des détails, voir Configuration rapide de la CLI AWS.

Procédure

1. Installez les outils de ligne de commande AWS à l'aide de la commande **dnf**.

```
# dnf install awscli
```

2. Utilisez la commande **aws --version** pour vérifier que vous avez installé la CLI AWS.

```
$ aws --version
aws-cli/1.19.77 Python/3.6.15 Linux/5.14.16-201.fc34.x86_64 botocore/1.20.77
```

3. Configurez le client de ligne de commande AWS en fonction de vos informations d'accès à AWS.

```
$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]:
Default output format [None]:
```

Ressources supplémentaires

- Configuration rapide de la CLI AWS
- Outils de ligne de commande AWS

3.4.2. Création d'un seau S3

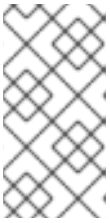
L'importation vers AWS nécessite un bac Amazon S3. Un bac Amazon S3 est une ressource Amazon dans laquelle vous stockez des objets. Dans le cadre du processus de téléchargement de votre image, vous créez un bac S3 et déplacez ensuite votre image vers le bac. Effectuez les étapes suivantes pour créer un seau.

Procédure

1. Lancez la console Amazon S3.
2. Cliquez sur **Create Bucket**. La boîte de dialogue **Create Bucket** apparaît.
3. Dans la vue **Name and region**:
   a. Saisissez une adresse **Bucket name**.
   b. Saisissez une adresse **Region**.

c. Cliquez sur **Next**.

4. Dans la vue **Configure options**, sélectionnez les options souhaitées et cliquez sur **Next**.
5. Dans la vue **Set permissions**, modifiez ou acceptez les options par défaut et cliquez sur **Next**.
6. Examinez la configuration de votre seau.
7. Cliquez sur **Create bucket**.



NOTE

Vous pouvez également utiliser l'interface de commande AWS pour créer un godet. Par exemple, la commande **aws s3 mb s3://my-new-bucket** crée un godet S3 nommé **my-new-bucket**. Pour plus d'informations sur la commande **mb**, reportez-vous à la référence des commandes de l'interface de commande AWS .

Ressources supplémentaires

- Console Amazon S3
- Référence de la commande CLI AWS

3.4.3. Création du rôle vmimport

Effectuez la procédure suivante pour créer le rôle **vmimport**, nécessaire à l'importation de VM. Pour plus d'informations, reportez-vous à la section Rôle du service d'importation de VM dans la documentation Amazon.

Procédure

1. Créez un fichier nommé **trust-policy.json** et incluez la politique suivante. Enregistrez le fichier sur votre système et notez son emplacement.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": { "Service": "vmie.amazonaws.com" },
         "Action": "sts:AssumeRole",
         "Condition": {
            "StringEquals":{
               "sts:Externalid": "vmimport"
            }
         }
      }
   ]
}
```

2. Utilisez la commande **create role** pour créer le rôle **vmimport**. Indiquez le chemin d'accès complet à l'emplacement du fichier **trust-policy.json**. Attribuez le préfixe **file://** au chemin d'accès. Par exemple :

```
$ aws iam create-role --role-name vmimport --assume-role-policy-document
file:///home/sample/ImportService/trust-policy.json
```

3. Créez un fichier nommé **role-policy.json** et incluez la politique suivante. Remplacez **s3-bucket-name** par le nom de votre panier S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::s3-bucket-name",
            "arn:aws:s3:::s3-bucket-name/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:ModifySnapshotAttribute",
            "ec2:CopySnapshot",
            "ec2:RegisterImage",
            "ec2:Describe*"
         ],
         "Resource":"*"
      }
   ]
}
```

4. Utilisez la commande **put-role-policy** pour attacher la politique au rôle que vous avez créé. Indiquez le chemin complet du fichier **role-policy.json**. Par exemple :

```
$ aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document
file:///home/sample/ImportService/role-policy.json
```

Ressources supplémentaires

- Rôle du service d'importation de VM
- Rôle de service requis

3.4.4. Conversion et transfert de l'image vers S3

Suivez la procédure suivante pour convertir et pousser votre image vers S3. Les exemples sont représentatifs ; ils convertissent une image au format **qcow2** en format **raw**. Amazon accepte les images aux formats **OVA**, **VHD**, **VHDX**, **VMDK** et **raw**. Voir Comment fonctionne l'importation/exportation de VM pour plus d'informations sur les formats d'image acceptés par Amazon.

Procédure

1. Exécutez la commande **qemu-img** pour convertir votre image. Par exemple :

```
# qemu-img convert -f qcow2 -O raw rhel-9.0-sample.qcow2 rhel-9.0-sample.raw
```

2. Pousser l'image vers S3.

```
$ aws s3 cp rhel-9.0-sample.raw s3://s3-bucket-name
```

NOTE

Cette procédure peut prendre quelques minutes. Une fois la procédure terminée, vous pouvez vérifier que votre image a bien été téléchargée dans votre panier S3 à l'aide de la console AWS S3.

Ressources supplémentaires

- Fonctionnement de l'importation/exportation de VM
- Console AWS S3

3.4.5. Importation d'une image en tant qu'instantané

La procédure suivante permet d'importer une image en tant qu'instantané.

Procédure

1. Créez un fichier pour spécifier un répertoire et un chemin d'accès pour votre image. Nommez le fichier **containers.json**. Dans l'exemple qui suit, remplacez **s3-bucket-name** par le nom de votre godet et **s3-key** par votre clé. Vous pouvez obtenir la clé de l'image à l'aide de la console Amazon S3.

```
{
    "Description": "rhel-9.0-sample.raw",
    "Format": "raw",
    "UserBucket": {
        "S3Bucket": "s3-bucket-name",
        "S3Key": "s3-key"
    }
}
```

2. Importez l'image en tant qu'instantané. Cet exemple utilise un fichier Amazon S3 public ; vous pouvez utiliser la console Amazon S3 pour modifier les paramètres d'autorisation de votre panier.

```
aws ec2 import-snapshot --disk-container file://containers.json
```

Le terminal affiche un message tel que le suivant. Notez le **ImportTaskID** dans le message.

```
{
    "SnapshotTaskDetail": {
        "Status": "active",
        "Format": "RAW",
        "DiskImageSize": 0.0,
```

```
        "UserBucket": {
            "S3Bucket": "s3-bucket-name",
            "S3Key": "rhel-9.0-sample.raw"
        },
        "Progress": "3",
        "StatusMessage": "pending"
    },
    "ImportTaskId": "import-snap-06cea01fa0f1166a8"
}
```

3. Suivez la progression de l'importation à l'aide de la commande **describe-import-snapshot-tasks**. Inclure la commande **ImportTaskID**.

```
$ aws ec2 describe-import-snapshot-tasks --import-task-ids import-snap-06cea01fa0f1166a8
```

Le message renvoyé indique l'état actuel de la tâche. Lorsqu'elle est terminée, **Status** affiche **completed**. Dans l'état, notez l'ID de l'instantané.

Ressources supplémentaires

- Console Amazon S3
- Importation d'un disque en tant qu'instantané à l'aide de VM Import/Export

3.4.6. Création d'une AMI à partir de l'instantané téléchargé

Dans EC2, vous devez choisir une Amazon Machine Image (AMI) lorsque vous lancez une instance. Suivez la procédure suivante pour créer une AMI à partir de votre instantané téléchargé.

Procédure

1. Accédez au tableau de bord AWS EC2.
2. Sous **Elastic Block Store**, sélectionnez **Snapshots**.
3. Recherchez l'identifiant de votre instantané (par exemple, **snap-0e718930bd72bcda0**).
4. Cliquez avec le bouton droit de la souris sur l'instantané et sélectionnez **Create image**.
5. Nommez votre image.
6. Sous **Virtualization type**, choisissez **Hardware-assisted virtualization**.
7. Cliquez sur **Create**. Dans la note concernant la création d'images, il y a un lien vers votre image.
8. Cliquez sur le lien de l'image. Votre image apparaît sous **Images>AMIs**.

NOTE

Vous pouvez également utiliser la commande AWS CLI **register-image** pour créer une AMI à partir d'un instantané. Voir register-image pour plus d'informations. Voici un exemple.

```
$ aws ec2 register-image \
   --name "myimagename" --description "myimagedescription" --architecture
x86_64 \
   --virtualization-type hvm --root-device-name "/dev/sda1" --ena-support \
   --block-device-mappings "{\"DeviceName\": \"/dev/sda1\",\"Ebs\":
{\"SnapshotId\": \"snap-0ce7f009b69ab274d\"}}"
```

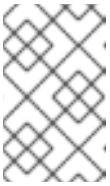
Vous devez spécifier le volume de périphérique racine **/dev/sda1** en tant que **root-device-name**. Pour obtenir des informations conceptuelles sur le mappage des périphériques pour AWS, voir Exemple de mappage des périphériques en bloc.

3.4.7. Lancement d'une instance à partir de l'AMI

Suivez la procédure suivante pour lancer et configurer une instance à partir de l'AMI.

Procédure

1. Dans le tableau de bord AWS EC2, sélectionnez **Images** puis **AMIs**.
2. Cliquez avec le bouton droit de la souris sur votre image et sélectionnez **Launch**.
3. Choisissez un site **Instance Type** qui répond ou dépasse les exigences de votre charge de travail.
   Voir Amazon EC2 Instance Types pour plus d'informations sur les types d'instances.
4. Cliquez sur **Next: Configure Instance Details**.
   a. Saisissez l'adresse **Number of instances** que vous souhaitez créer.
   b. Pour **Network**, sélectionnez le VPC que vous avez créé lors de la configuration de votre environnement AWS. Sélectionnez un sous-réseau pour l'instance ou créez-en un nouveau.
   c. Sélectionnez **Enable** pour Auto-assign Public IP.



NOTE

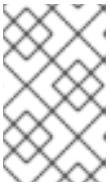
Il s'agit des options de configuration minimales nécessaires pour créer une instance de base. Examinez les options supplémentaires en fonction des exigences de votre application.

5. Cliquez sur **Next: Add Storage**. Vérifiez que le stockage par défaut est suffisant.
6. Cliquez sur **Next: Add Tags**.

NOTE

Les balises peuvent vous aider à gérer vos ressources AWS. Pour plus d'informations sur les balises, consultez la section Baliser vos ressources Amazon EC2.

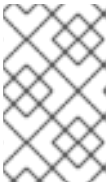
7. Cliquez sur **Next: Configure Security Group**. Sélectionnez le groupe de sécurité que vous avez créé lors de la configuration de votre environnement AWS.
8. Cliquez sur **Review and Launch**. Vérifiez vos sélections.
9. Cliquez sur **Launch**. Vous êtes invité à sélectionner une paire de clés existante ou à créer une nouvelle paire de clés. Sélectionnez la paire de clés que vous avez créée lors de la configuration de votre environnement AWS.



NOTE

Vérifiez que les droits d'accès à votre clé privée sont corrects. Utilisez les options de la commande **chmod 400 <keyname>.pem** pour modifier les autorisations, si nécessaire.

10. Cliquez sur **Launch Instances**.
11. Cliquez sur **View Instances**. Vous pouvez nommer la ou les instances.
Vous pouvez maintenant lancer une session SSH vers votre/vos instance(s) en sélectionnant une instance et en cliquant sur **Connect**. Utilisez l'exemple fourni pour **A standalone SSH client**.



NOTE

Vous pouvez également lancer une instance à l'aide de la CLI AWS. Pour plus d'informations, voir Lancer, lister et terminer les instances Amazon EC2 dans la documentation d'Amazon.

Ressources supplémentaires

- Console de gestion AWS
- Configuration avec Amazon EC2
- Instances Amazon EC2
- Types d'instances Amazon EC2

3.4.8. Attacher des abonnements Red Hat

Pour attacher votre abonnement Red Hat à une instance RHEL, suivez les étapes suivantes.

Conditions préalables

- Vous devez avoir activé vos abonnements.

Procédure

1. Enregistrez votre système.

```
# subscription-manager register --auto-attach
```

2. Joignez vos abonnements.

   - Vous pouvez utiliser une clé d'activation pour attacher des abonnements. Pour plus d'informations, reportez-vous à la section Créer des clés d'activation pour le portail client Red Hat.

   - Vous pouvez également rattacher manuellement un abonnement à l'aide de l'ID du pool d'abonnements (Pool ID). Voir Attacher et supprimer des abonnements via la ligne de commande.

Ressources supplémentaires

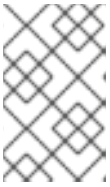
- Création de clés d'activation pour le portail client de Red Hat
- Attacher et supprimer des abonnements via la ligne de commande
- Utilisation et configuration du Gestionnaire d'abonnements Red Hat

3.4.9. Configuration de l'enregistrement automatique sur AWS Gold Images

Pour accélérer et faciliter le déploiement des machines virtuelles RHEL 8 sur Amazon Web Services (AWS), vous pouvez configurer les images Gold de RHEL 8 pour qu'elles soient automatiquement enregistrées dans le gestionnaire d'abonnements Red Hat (RHSM).

Conditions préalables

- Vous avez téléchargé la dernière Gold Image RHEL 8 pour AWS. Pour plus d'informations, voir Utilisation des Gold Images sur AWS.



NOTE

Un compte AWS ne peut être rattaché qu'à un seul compte Red Hat à la fois. Par conséquent, assurez-vous qu'aucun autre utilisateur n'a besoin d'accéder au compte AWS avant de l'attacher à votre compte Red Hat.

Procédure

1. Téléchargez l'image Gold sur AWS. Pour obtenir des instructions, voir Téléchargement de l'image Red Hat Enterprise Linux vers AWS.

2. Créez des machines virtuelles à l'aide de l'image téléchargée. Elles seront automatiquement abonnées au RHSM.

Vérification

- Dans une VM RHEL 9 créée à l'aide des instructions ci-dessus, vérifiez que le système est enregistré dans le RHSM en exécutant la commande **subscription-manager identity**. Sur un système enregistré avec succès, cette commande affiche l'UUID du système. Par exemple :

```
# subscription-manager identity
system identity: fdc46662-c536-43fb-a18a-bbcb283102b7
name: 192.168.122.222
```

```
org name: 6340056
org ID: 6340056
```

Ressources supplémentaires

- Console de gestion AWS
- Configuration des sources cloud pour les services Red Hat

3.5. RESSOURCES SUPPLÉMENTAIRES

- Guide de référence de Red Hat Cloud Access
- Red Hat dans le nuage public
- Red Hat Enterprise Linux sur Amazon EC2 - FAQs
- Configuration avec Amazon EC2
- Red Hat sur Amazon Web Services

CHAPITRE 4. CONFIGURATION D'UN CLUSTER RED HAT HIGH AVAILABILITY SUR AWS

Ce chapitre fournit des informations et des procédures pour configurer un cluster Red Hat High Availability (HA) sur Amazon Web Services (AWS) en utilisant des instances EC2 comme nœuds de cluster. Notez que vous disposez d'un certain nombre d'options pour obtenir les images Red Hat Enterprise Linux (RHEL) que vous utilisez pour votre cluster. Pour plus d'informations sur les options d'images pour AWS, voir Options d'images Red Hat Enterprise Linux sur AWS .

Ce chapitre comprend

- Procédures préalables pour configurer votre environnement pour AWS. Une fois votre environnement configuré, vous pouvez créer et configurer des instances EC2.
- Procédures spécifiques à la création de clusters HA, qui transforment des nœuds individuels en un cluster de nœuds HA sur AWS. Il s'agit notamment des procédures d'installation des packages et des agents de haute disponibilité sur chaque nœud de cluster, de la configuration des clôtures et de l'installation des agents de ressources réseau AWS.

Conditions préalables

- Créez un compte sur le portail client de Red Hat .
- Inscrivez-vous à AWS et configurez vos ressources AWS. Pour plus d'informations, reportez-vous à la section Configuration d'Amazon EC2.

4.1. CRÉATION DE LA CLÉ D'ACCÈS AWS ET DE LA CLÉ D'ACCÈS SECRÈTE AWS

Vous devez créer une clé d'accès AWS et une clé d'accès secrète AWS avant d'installer la CLI AWS. Les API de clôture et d'agent de ressources utilisent la clé d'accès AWS et la clé d'accès secrète pour se connecter à chaque nœud du cluster.

Suivez les étapes suivantes pour créer ces clés.

Conditions préalables

- Votre compte utilisateur IAM doit avoir un accès programmatique. Voir Configuration de l'environnement AWS pour plus d'informations.

Procédure

1. Lancez la console AWS.
2. Cliquez sur votre ID de compte AWS pour afficher le menu déroulant et sélectionnez **My Security Credentials**.
3. Cliquez sur **Users**.
4. Sélectionnez l'utilisateur et ouvrez l'écran **Summary**.
5. Cliquez sur l'onglet **Security credentials**.
6. Cliquez sur **Create access key**.

7. Téléchargez le fichier **.csv** (ou sauvegardez les deux clés). Vous devez saisir ces clés lors de la création du dispositif de clôture.

4.2. INSTALLATION DE L'INTERFACE DE PROGRAMMATION AWS

La plupart des procédures requises pour gérer les clusters HA dans AWS incluent l'utilisation de la CLI AWS. Suivez les étapes suivantes pour installer la CLI AWS.

Conditions préalables

- Vous avez créé un identifiant de clé d'accès AWS et une clé d'accès secrète AWS, et vous y avez accès. Pour obtenir des instructions et des détails, voir Configuration rapide de la CLI AWS.

Procédure

1. Installez les outils de ligne de commande AWS à l'aide de la commande **dnf**.

```
# dnf install awscli
```

2. Utilisez la commande **aws --version** pour vérifier que vous avez installé la CLI AWS.

```
$ aws --version
aws-cli/1.19.77 Python/3.6.15 Linux/5.14.16-201.fc34.x86_64 botocore/1.20.77
```

3. Configurez le client de ligne de commande AWS en fonction de vos informations d'accès à AWS.

```
$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]:
Default output format [None]:
```

Ressources supplémentaires

- Configuration rapide de la CLI AWS
- Outils de ligne de commande AWS

4.3. CRÉATION D'UNE INSTANCE EC2 HA

Effectuez les étapes suivantes pour créer les instances que vous utilisez comme nœuds de cluster HA. Notez que vous disposez d'un certain nombre d'options pour obtenir les images RHEL que vous utilisez pour votre cluster. Voir Red Hat Enterprise Linux Image options on AWS pour plus d'informations sur les options d'images pour AWS.

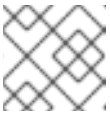
Vous pouvez créer et télécharger une image personnalisée que vous utiliserez pour vos nœuds de cluster, ou vous pouvez utiliser une image Gold ou une image à la demande.

Conditions préalables

- Vous devez avoir mis en place un environnement AWS. Voir Configuration avec Amazon EC2 pour plus d'informations.

Procédure

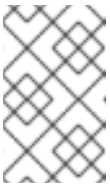
1. Dans le tableau de bord AWS EC2, sélectionnez **Images** puis **AMIs**.
2. Cliquez avec le bouton droit de la souris sur votre image et sélectionnez **Launch**.
3. Choisissez un site **Instance Type** qui répond ou dépasse les exigences de votre charge de travail. En fonction de votre application HA, chaque instance peut avoir besoin d'une capacité plus élevée.
   Voir Amazon EC2 Instance Types pour plus d'informations sur les types d'instances.
4. Cliquez sur **Next: Configure Instance Details**.
   a. Saisissez le site **Number of instances** que vous souhaitez créer pour le cluster. Cet exemple de procédure utilise trois nœuds de cluster.

   NOTE

   Ne pas se lancer dans un groupe de mise à l'échelle automatique.

   b. Pour **Network**, sélectionnez le VPC que vous avez créé dans Configurer l'environnement AWS. Sélectionnez le sous-réseau de l'instance pour créer un nouveau sous-réseau.
   c. Sélectionnez **Enable** pour Auto-assign Public IP. Il s'agit des sélections minimales que vous devez effectuer pour **Configure Instance Details**. En fonction de votre application HA spécifique, il se peut que vous deviez effectuer des sélections supplémentaires.

   NOTE

   Il s'agit des options de configuration minimales nécessaires pour créer une instance de base. Examinez les options supplémentaires en fonction des exigences de votre application HA.

5. Cliquez sur **Next: Add Storage** et vérifiez que le stockage par défaut est suffisant. Il n'est pas nécessaire de modifier ces paramètres, sauf si votre application HA nécessite d'autres options de stockage.
6. Cliquez sur **Next: Add Tags**.

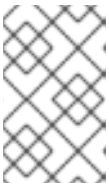
   

   NOTE

   Les balises peuvent vous aider à gérer vos ressources AWS. Pour plus d'informations sur les balises, consultez la section Baliser vos ressources Amazon EC2.

7. Cliquez sur **Next: Configure Security Group**. Sélectionnez le groupe de sécurité existant que vous avez créé dans Configuration de l'environnement AWS .
8. Cliquez sur **Review and Launch** et vérifiez vos choix.
9. Cliquez sur **Launch**. Vous êtes invité à sélectionner une paire de clés existante ou à créer une nouvelle paire de clés. Sélectionnez la paire de clés que vous avez créée lors de la configuration de l'environnement AWS.
10. Cliquez sur **Launch Instances**.

11. Cliquez sur **View Instances**. Vous pouvez nommer la ou les instances.



NOTE

Vous pouvez également lancer des instances à l'aide de la CLI AWS. Pour plus d'informations, voir Lancer, lister et terminer des instances Amazon EC2 dans la documentation d'Amazon.

Ressources supplémentaires

- Console de gestion AWS
- Configuration avec Amazon EC2
- Instances Amazon EC2
- Types d'instances Amazon EC2

4.4. CONFIGURATION DE LA CLÉ PRIVÉE

Effectuez les tâches de configuration suivantes pour utiliser le fichier de clés SSH privées (**.pem**) avant de pouvoir l'utiliser dans une session SSH.

Procédure

1. Déplacez le fichier clé du répertoire **Downloads** vers votre répertoire **Home** ou vers votre répertoire **~/.ssh directory**.
2. Modifiez les autorisations du fichier clé de manière à ce que seul l'utilisateur root puisse le lire.

```
# chmod 400 KeyName.pem
```

4.5. CONNEXION À UNE INSTANCE EC2

Effectuez les étapes suivantes sur tous les nœuds pour vous connecter à une instance EC2.

Procédure

1. Lancez la console AWS et sélectionnez l'instance EC2.
2. Cliquez sur **Connect** et sélectionnez **A standalone SSH client**.
3. Depuis votre session de terminal SSH, connectez-vous à l'instance en utilisant l'exemple AWS fourni dans la fenêtre contextuelle. Ajoutez le chemin d'accès correct à votre fichier **KeyName.pem** si le chemin d'accès n'est pas indiqué dans l'exemple.

4.6. INSTALLATION DES PAQUETS ET DES AGENTS DE HAUTE DISPONIBILITÉ

Effectuez les étapes suivantes sur tous les nœuds pour installer les paquets et les agents de haute disponibilité.

Procédure

1. Supprimez le client AWS Red Hat Update Infrastructure (RHUI).

```
$ sudo -i
# dnf -y remove rh-amazon-rhui-client*
```

2. Enregistrez la VM auprès de Red Hat.

```
# subscription-manager register --auto-attach
```

3. Désactiver tous les dépôts.

```
# subscription-manager repos --disable=*
```

4. Activer les référentiels RHEL 9 Server HA.

```
# subscription-manager repos --enable=rhel-9-for-x86_64-highavailability-rpms
```

5. Mettre à jour l'instance RHEL AWS.

```
# dnf update -y
```

6. Installez les paquets logiciels Red Hat High Availability Add-On, ainsi que tous les agents de clôture disponibles dans le canal High Availability.

```
# dnf install pcs pacemaker fence-agents-aws
```

7. L'utilisateur **hacluster** a été créé lors de l'installation de **pcs** et **pacemaker** à l'étape précédente. Créez un mot de passe pour **hacluster** sur tous les nœuds du cluster. Utilisez le même mot de passe pour tous les nœuds.

```
# passwd hacluster
```

8. Ajouter le service **high availability** au pare-feu RHEL si **firewalld.service** est installé.

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --reload
```

9. Démarrer le service **pcs** et l'autoriser à démarrer au démarrage.

```
# systemctl start pcsd.service
# systemctl enable pcsd.service
```

10. Modifiez **/etc/hosts** et ajoutez les noms d'hôtes RHEL et les adresses IP internes. Voir Comment le fichier /etc/hosts doit-il être configuré sur les nœuds de cluster RHEL ? pour plus d'informations.

Vérification

- Assurez-vous que le service **pcs** est en cours d'exécution.

```
# systemctl status pcsd.service
```

```
pcsd.service - PCS GUI and remote configuration interface
Loaded: loaded (/usr/lib/systemd/system/pcsd.service; enabled; vendor preset: disabled)
Active: active (running) since Thu 2018-03-01 14:53:28 UTC; 28min ago
Docs: man:pcsd(8)
man:pcs(8)
Main PID: 5437 (pcsd)
CGroup: /system.slice/pcsd.service
    └─5437 /usr/bin/ruby /usr/lib/pcsd/pcsd > /dev/null &
Mar 01 14:53:27 ip-10-0-0-48.ec2.internal systemd[1]: Starting PCS GUI and remote
configuration interface…
Mar 01 14:53:28 ip-10-0-0-48.ec2.internal systemd[1]: Started PCS GUI and remote
configuration interface.
```

4.7. CRÉATION D'UN CLUSTER

Effectuez les étapes suivantes pour créer la grappe de nœuds.

Procédure

1. Sur l'un des nœuds, entrez la commande suivante pour authentifier l'utilisateur pcs **hacluster**. Dans la commande, indiquez le nom de chaque nœud de la grappe.

```
# pcs host auth <hostname1> <hostname2> <hostname3>
```

Exemple :

```
[root@node01 clouduser]# pcs host auth node01 node02 node03
Username: hacluster
Password:
node01: Authorized
node02: Authorized
node03: Authorized
```

2. Create the cluster.

```
# pcs cluster setup <cluster_name> <hostname1> <hostname2> <hostname3>
```

Exemple :

```
[root@node01 clouduser]# pcs cluster setup new_cluster node01 node02 node03

[...]

Synchronizing pcsd certificates on nodes node01, node02, node03...
node02: Success
node03: Success
node01: Success
Restarting pcsd on the nodes in order to reload the certificates...
node02: Success
node03: Success
node01: Success
```

Vérification

1. Activer le cluster.

```
[root@node01 clouduser]# pcs cluster enable --all
node02: Cluster Enabled
node03: Cluster Enabled
node01: Cluster Enabled
```

2. Démarrer le cluster.

```
[root@node01 clouduser]# pcs cluster start --all
node02: Starting Cluster...
node03: Starting Cluster...
node01: Starting Cluster...
```

4.8. CONFIGURATION DES CLÔTURES

La configuration de la clôture permet d'isoler automatiquement un nœud défaillant de votre cluster AWS, ce qui empêche le nœud de consommer les ressources du cluster ou d'en compromettre la fonctionnalité.

Vous pouvez configurer la clôture sur un cluster AWS à l'aide de plusieurs méthodes. Cette section fournit les informations suivantes :

- Une procédure standard pour la configuration par défaut.
- Une procédure de configuration alternative pour une configuration plus avancée, axée sur l'automatisation.

Procédure standard

1. Saisissez la requête de métadonnées AWS suivante pour obtenir l'ID d'instance de chaque nœud. Vous avez besoin de ces identifiants pour configurer le dispositif de clôture. Voir Métadonnées d'instance et données utilisateur pour plus d'informations.

```
# echo $(curl -s http://169.254.169.254/latest/meta-data/instance-id)
```

   Exemple :

```
[root@ip-10-0-0-48 ~]# echo $(curl -s http://169.254.169.254/latest/meta-data/instance-id) i-07f1ac63af0ec0ac6
```

2. Entrez la commande suivante pour configurer le périphérique de clôture. Utilisez la commande **pcmk_host_map** pour faire correspondre le nom d'hôte RHEL à l'ID d'instance. Utilisez la clé d'accès AWS et la clé d'accès secrète AWS que vous avez configurées précédemment.

```
# pcs stonith \
    create <name> fence_aws access_key=access-key secret_key=<secret-access-key> \
    region=<region> pcmk_host_map="rhel-hostname-1:Instance-ID-1;rhel-hostname-2:Instance-ID-2;rhel-hostname-3:Instance-ID-3" \
    power_timeout=240 pcmk_reboot_timeout=480 pcmk_reboot_retries=4
```

   Exemple :

```
[root@ip-10-0-0-48 ~]# pcs stonith \
create clusterfence fence_aws access_key=AKIAI123456MRMJA
secret_key=a75EYIG4RVL3hdsdAslK7koQ8dzaDyn5yoIZ/ \
region=us-east-1 pcmk_host_map="ip-10-0-0-48:i-07f1ac63af0ec0ac6;ip-10-0-0-46:i-
063fc5fe93b4167b2;ip-10-0-0-58:i-08bd39eb03a6fd2c7" \
power_timeout=240 pcmk_reboot_timeout=480 pcmk_reboot_retries=4
```

Procédure alternative

1. Obtenez l'ID VPC du cluster.

```
# aws ec2 describe-vpcs --output text --filters "Name=tag:Name,Values=clustername-vpc" --
query 'Vpcs[*].VpcId'
vpc-06bc10ac8f6006664
```

2. En utilisant l'ID VPC du cluster, obtenez les instances VPC.

```
$ aws ec2 describe-instances --output text --filters "Name=vpc-id,Values=vpc-
06bc10ac8f6006664" --query 'Reservations[*].Instances[*].{Name:Tags[? Key==Name]|
[0].Value,Instance:InstanceId}' | grep "\-node[a-c]"
i-0b02af8927a895137     clustername-nodea-vm
i-0cceb4ba8ab743b69     clustername-nodeb-vm
i-0502291ab38c762a5     clustername-nodec-vm
```

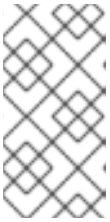
3. Utilisez les identifiants d'instance obtenus pour configurer la clôture sur chaque nœud du cluster. Par exemple :

```
[root@nodea ~]# CLUSTER=clustername && pcs stonith create fence${CLUSTER}
fence_aws access_key=XXXXXXXXXXXXXXXXXXX pcmk_host_map=$(for NODE \
in node{a..c}; do ssh ${NODE} "echo -n \${HOSTNAME}:\$(curl -s
http://169.254.169.254/latest/meta-data/instance-id)\;"; done) \
pcmk_reboot_retries=4 pcmk_reboot_timeout=480 power_timeout=240 region=xx-xxxx-x
secret_key=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[root@nodea ~]# pcs stonith config fence${CLUSTER}

Resource: clustername (class=stonith type=fence_aws)
Attributes: access_key=XXXXXXXXXXXXXXXXXXX pcmk_host_map=nodea:i-
0b02af8927a895137;nodeb:i-0cceb4ba8ab743b69;nodec:i-0502291ab38c762a5;
pcmk_reboot_retries=4 pcmk_reboot_timeout=480 power_timeout=240 region=xx-xxxx-x
secret_key=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Operations: monitor interval=60s (clustername-monitor-interval-60s)
```

Vérification

1. Testez l'agent de clôture pour l'un des nœuds de la grappe.

```
# pcs stonith fence awsnodename
```



NOTE

La réponse à la commande peut mettre plusieurs minutes à s'afficher. Si vous observez la session de terminal active pour le nœud clôturé, vous constaterez que la connexion de terminal est immédiatement interrompue après la saisie de la commande de clôture.

Exemple :

```
[root@ip-10-0-0-48 ~]# pcs stonith fence ip-10-0-0-58

Node: ip-10-0-0-58 fenced
```

2. Vérifier l'état pour s'assurer que le nœud est clôturé.

```
# pcs status
```

Exemple :

```
[root@ip-10-0-0-48 ~]# pcs status

Cluster name: newcluster
Stack: corosync
Current DC: ip-10-0-0-46 (version 1.1.18-11.el7-2b07d5c5a9) - partition with quorum
Last updated: Fri Mar  2 19:55:41 2018
Last change: Fri Mar  2 19:24:59 2018 by root via cibadmin on ip-10-0-0-46

3 nodes configured
1 resource configured

Online: [ ip-10-0-0-46 ip-10-0-0-48 ]
OFFLINE: [ ip-10-0-0-58 ]

Full list of resources:
clusterfence  (stonith:fence_aws):    Started ip-10-0-0-46

Daemon Status:
corosync: active/disabled
pacemaker: active/disabled
pcsd: active/enabled
```

3. Démarrez le nœud qui a été clôturé à l'étape précédente.

```
# pcs cluster start awshostname
```

4. Vérifier l'état pour s'assurer que le nœud a démarré.

```
# pcs status
```

Exemple :

```
[root@ip-10-0-0-48 ~]# pcs status
```

```
Cluster name: newcluster
Stack: corosync
Current DC: ip-10-0-0-46 (version 1.1.18-11.el7-2b07d5c5a9) - partition with quorum
Last updated: Fri Mar  2 20:01:31 2018
Last change: Fri Mar  2 19:24:59 2018 by root via cibadmin on ip-10-0-0-48

3 nodes configured
1 resource configured

Online: [ ip-10-0-0-46 ip-10-0-0-48 ip-10-0-0-58 ]

Full list of resources:

 clusterfence  (stonith:fence_aws):    Started ip-10-0-0-46

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

4.9. INSTALLATION DE L'INTERFACE DE PROGRAMMATION AWS SUR LES NŒUDS DE CLUSTER

Auparavant, vous avez installé la CLI AWS sur votre système hôte. Vous devez installer AWS CLI sur les nœuds de cluster avant de configurer les agents de ressources réseau.

Effectuez la procédure suivante sur chaque nœud du cluster.

Conditions préalables

- Vous devez avoir créé une clé d'accès AWS et une clé d'accès secrète AWS. Voir Création de la clé d'accès AWS et de la clé d'accès secrète AWS pour plus d'informations.

Procédure

1. Installez l'interface de programmation AWS. Pour plus d'informations, voir Installation de l'interface de programmation AWS.

2. Vérifiez que la CLI AWS est correctement configurée. Les identifiants et les noms des instances doivent s'afficher.
   Exemple :

```
[root@ip-10-0-0-48 ~]# aws ec2 describe-instances --output text --query
'Reservations[].Instances[].[InstanceId,Tags[?Key==Name].Value]'

i-07f1ac63af0ec0ac6
ip-10-0-0-48
i-063fc5fe93b4167b2
ip-10-0-0-46
i-08bd39eb03a6fd2c7
ip-10-0-0-58
```

4.10. INSTALLATION DES AGENTS DE RESSOURCES RÉSEAU

Pour que les opérations HA fonctionnent, le cluster utilise des agents de ressources réseau AWS pour activer la fonctionnalité de basculement. Si un nœud ne répond pas à un contrôle de battement de cœur dans un délai déterminé, le nœud est clôturé et les opérations basculent sur un autre nœud du cluster. Les agents de ressources réseau doivent être configurés pour que cela fonctionne.

Ajoutez les deux ressources au même groupe pour appliquer les contraintes **order** et **colocation**.

Create a secondary private IP resource and virtual IP resource

Suivez la procédure suivante pour ajouter une adresse IP privée secondaire et créer une IP virtuelle. Vous pouvez effectuer cette procédure à partir de n'importe quel nœud du cluster.

Procédure

1. Affichez la description de l'agent de ressources **AWS Secondary Private IP Address** (awsvip). Elle indique les options et les opérations par défaut de cet agent.

```
# pcs resource describe awsvip
```

2. Créez l'adresse IP privée secondaire en utilisant une adresse IP privée inutilisée dans le bloc **VPC CIDR**.

```
# pcs resource create privip awsvip secondary_private_ip=Unused-IP-Address --group group-name
```

Exemple :

```
[root@ip-10-0-0-48 ~]# pcs resource create privip awsvip
secondary_private_ip=10.0.0.68 --group networking-group
```

3. Créer une ressource IP virtuelle. Il s'agit d'une adresse IP VPC qui peut être rapidement transférée du nœud clôturé au nœud de basculement, masquant ainsi la défaillance du nœud clôturé dans le sous-réseau.

```
# pcs resource create vip IPaddr2 ip=secondary-private-IP --group group-name
```

Exemple :

```
root@ip-10-0-0-48 ~]# pcs resource create vip IPaddr2 ip=10.0.0.68 --group networking-group
```

Vérification

- Vérifiez que les ressources fonctionnent.

```
# pcs status
```

Exemple :

```
[root@ip-10-0-0-48 ~]# pcs status
Cluster name: newcluster
Stack: corosync
Current DC: ip-10-0-0-46 (version 1.1.18-11.el7-2b07d5c5a9) - partition with quorum
```

```
Last updated: Fri Mar  2 22:34:24 2018
Last change: Fri Mar  2 22:14:58 2018 by root via cibadmin on ip-10-0-0-46

3 nodes configured
3 resources configured

Online: [ ip-10-0-0-46 ip-10-0-0-48 ip-10-0-0-58 ]

Full list of resources:

clusterfence    (stonith:fence_aws):    Started ip-10-0-0-46
 Resource Group: networking-group
     privip (ocf::heartbeat:awsvip):    Started ip-10-0-0-48
     vip    (ocf::heartbeat:IPaddr2):   Started ip-10-0-0-58

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

Create an elastic IP address

Une adresse IP élastique est une adresse IP publique qui peut être rapidement transférée du nœud bloqué au nœud de basculement, masquant ainsi la défaillance du nœud bloqué.

Notez que cette adresse est différente de la ressource IP virtuelle créée précédemment. L'adresse IP élastique est utilisée pour les connexions Internet publiques au lieu des connexions de sous-réseau.

1. Ajoutez les deux ressources au même groupe que celui qui a été créé précédemment pour appliquer les contraintes **order** et **colocation**.

2. Entrez la commande CLI AWS suivante pour créer une adresse IP élastique.

```
[root@ip-10-0-0-48 ~]# aws ec2 allocate-address --domain vpc --output text
eipalloc-4c4a2c45   vpc 35.169.153.122
```

3. Voir la description de l'agent de ressources AWS Secondary Elastic IP Address (awseip). La commande suivante présente les options et les opérations par défaut de cet agent.

```
# pcs resource describe awseip
```

4. Créez la ressource Adresse IP élastique secondaire en utilisant l'adresse IP allouée créée à l'étape 1.

```
# pcs resource create elastic awseip elastic_ip=Elastic-IP-Address
allocation_id=Elastic-IP-Association-ID --group networking-group
```

   Exemple :

```
# pcs resource create elastic awseip elastic_ip=35.169.153.122 allocation_id=eipalloc-
4c4a2c45 --group networking-group
```

Vérification

- Entrez la commande **pcs status** pour vérifier que la ressource est en cours d'exécution.

```
# pcs status
```

Exemple :

```
[root@ip-10-0-0-58 ~]# pcs status
Cluster name: newcluster
Stack: corosync
Current DC: ip-10-0-0-58 (version 1.1.18-11.el7-2b07d5c5a9) - partition with quorum
Last updated: Mon Mar  5 16:27:55 2018
Last change: Mon Mar  5 15:57:51 2018 by root via cibadmin on ip-10-0-0-46

3 nodes configured
4 resources configured

Online: [ ip-10-0-0-46 ip-10-0-0-48 ip-10-0-0-58 ]

Full list of resources:

 clusterfence   (stonith:fence_aws):    Started ip-10-0-0-46
 Resource Group: networking-group
     privip (ocf::heartbeat:awsvip):  Started ip-10-0-0-48
     vip    (ocf::heartbeat:IPaddr2):    Started ip-10-0-0-48
     elastic (ocf::heartbeat:awseip):    Started ip-10-0-0-48

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

Test the elastic IP address

Entrez les commandes suivantes pour vérifier que les ressources IP virtuelles (awsvip) et IP élastiques (awseip) fonctionnent.

Procédure

1. Lancez une session SSH depuis votre poste de travail local vers l'adresse IP élastique précédemment créée.

```
$ ssh -l ec2-user -i ~/.ssh/<KeyName>.pem elastic-IP
```

Exemple :

```
$ ssh -l ec2-user -i ~/.ssh/cluster-admin.pem 35.169.153.122
```

2. Vérifiez que l'hôte auquel vous vous êtes connecté via SSH est l'hôte associé à la ressource élastique créée.

Ressources supplémentaires

- Aperçu du module complémentaire de haute disponibilité

- Configurer et gérer des clusters à haute disponibilité

4.11. CONFIGURATION DU STOCKAGE EN BLOC PARTAGÉ

Pour configurer le stockage en bloc partagé pour un cluster Red Hat High Availability avec des volumes Amazon Elastic Block Storage (EBS) Multi-Attach, utilisez la procédure suivante. Notez que cette procédure est facultative et que les étapes ci-dessous supposent trois instances (un cluster à trois nœuds) avec un disque partagé de 1 To.

Conditions préalables

- Vous devez utiliser une instance Amazon EC2 basée sur AWS Nitro System .

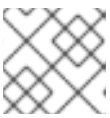
Procédure

1. Créez un volume de blocs partagé à l'aide de la commande AWS create-volume.

```
$ aws ec2 create-volume --availability-zone <availability_zone> --no-encrypted --size 1024 --volume-type io1 --iops 51200 --multi-attach-enabled
```

Par exemple, la commande suivante crée un volume dans la zone de disponibilité **us-east-1a**.

```
$ aws ec2 create-volume --availability-zone us-east-1a --no-encrypted --size 1024 --volume-type io1 --iops 51200 --multi-attach-enabled

{
    "AvailabilityZone": "us-east-1a",
    "CreateTime": "2020-08-27T19:16:42.000Z",
    "Encrypted": false,
    "Size": 1024,
    "SnapshotId": "",
    "State": "creating",
    "VolumeId": "vol-042a5652867304f09",
    "Iops": 51200,
    "Tags": [ ],
    "VolumeType": "io1"
}
```



NOTE

Vous aurez besoin du site **VolumeId** pour l'étape suivante.

2. Pour chaque instance de votre cluster, attachez un volume de blocs partagé à l'aide de la commande AWS attach-volume. Utilisez vos **<instance_id>** et **<volume_id>**.

```
$ aws ec2 attach-volume --device /dev/xvdd --instance-id <instance_id> --volume-id <volume_id>
```

Par exemple, la commande suivante attache un volume de blocs partagés **vol-042a5652867304f09** à **instance i-0eb803361c2c887f2**.

```
$ aws ec2 attach-volume --device /dev/xvdd --instance-id i-0eb803361c2c887f2 --volume-id vol-042a5652867304f09
```

```
{
    "AttachTime": "2020-08-27T19:26:16.086Z",
    "Device": "/dev/xvdd",
    "InstanceId": "i-0eb803361c2c887f2",
    "State": "attaching",
    "VolumeId": "vol-042a5652867304f09"
}
```

Vérification

1. Pour chaque instance de votre cluster, vérifiez que le périphérique de bloc est disponible en utilisant la commande **ssh** avec votre instance **<ip_address>**.

```
# ssh <ip_address> "hostname ; lsblk -d | grep ' 1T '"
```

Par exemple, la commande suivante répertorie les détails, y compris le nom d'hôte et le périphérique de bloc pour l'instance IP **198.51.100.3**.

```
# ssh 198.51.100.3 "hostname ; lsblk -d | grep ' 1T '"

nodea
nvme2n1 259:1    0   1T  0 disk
```

2. Utilisez la commande **ssh** pour vérifier que chaque instance de votre cluster utilise le même disque partagé.

```
# ssh <ip_address> "hostname ; lsblk -d | grep ' 1T ' | awk '{print \$1}' | xargs -i udevadm info --query=all --name=/dev/{} | grep '^E: ID_SERIAL='"
```

Par exemple, la commande suivante répertorie les détails, y compris le nom d'hôte et l'ID du volume de disque partagé pour l'adresse IP de l'instance **198.51.100.3**.

```
# ssh 198.51.100.3 "hostname ; lsblk -d | grep ' 1T ' | awk '{print \$1}' | xargs -i udevadm info --query=all --name=/dev/{} | grep '^E: ID_SERIAL='"

nodea
E: ID_SERIAL=Amazon Elastic Block Store_vol0fa5342e7aedf09f7
```

Ressources supplémentaires

- Configuration d'un système de fichiers GFS2 dans un cluster
- Configuration des systèmes de fichiers GFS2

4.12. RESSOURCES SUPPLÉMENTAIRES

- Guide de référence de Red Hat Cloud Access
- Red Hat dans le nuage public
- Red Hat Enterprise Linux sur Amazon EC2 - FAQs

- Configuration avec Amazon EC2
- Red Hat sur Amazon Web Services