



Red Hat Enterprise Linux 9

Composition d'une image système RHEL personnalisée

Création d'images système personnalisées avec le constructeur d'images sur Red Hat
Enterprise Linux 9

Red Hat Enterprise Linux 9 Composition d'une image système RHEL personnalisée

Création d'images système personnalisées avec le constructeur d'images sur Red Hat Enterprise Linux 9

Notice légale

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Résumé

Image builder est un outil permettant de créer des images système personnalisées et prêtes à être déployées : disques d'installation, machines virtuelles, images spécifiques à un fournisseur de cloud, etc. Le constructeur d'images permet de créer ces images plus rapidement que les procédures manuelles, car il élimine les configurations spécifiques requises pour chaque type de sortie. Ce document décrit comment configurer le constructeur d'images et créer des images avec lui.

Table des matières

RENDRE L'OPEN SOURCE PLUS INCLUSIF	4
FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT	5
CHAPITRE 1. DESCRIPTION DU CONSTRUCTEUR D'IMAGES	6
1.1. QU'EST-CE QUE LE CONSTRUCTEUR D'IMAGES ?	6
1.2. TERMINOLOGIE DES CONSTRUCTEURS D'IMAGES	6
1.3. FORMATS DE SORTIE DU CONSTRUCTEUR D'IMAGES	6
1.4. CONFIGURATION REQUISE POUR LE CONSTRUCTEUR D'IMAGES	7
CHAPITRE 2. INSTALLATION DU CONSTRUCTEUR D'IMAGES	9
2.1. INSTALLER LE CONSTRUCTEUR D'IMAGES DANS UNE MACHINE VIRTUELLE	9
CHAPITRE 3. GESTION DES RÉFÉRENTIELS DE CONSTRUCTEURS D'IMAGES	11
3.1. DÉPÔTS PAR DÉFAUT DU CONSTRUCTEUR D'IMAGES	11
3.2. REMPLACER UN RÉFÉRENTIEL SYSTÈME	11
3.3. REMPLACER UN RÉFÉRENTIEL SYSTÈME PAR UN SUPPORT POUR LES ABONNEMENTS	12
CHAPITRE 4. CRÉATION D'IMAGES SYSTÈME À L'AIDE DE L'INTERFACE DE LIGNE DE COMMANDE IMAGE BUILDER	14
4.1. PRÉSENTATION DE L'INTERFACE EN LIGNE DE COMMANDE DU CONSTRUCTEUR D'IMAGES	14
4.2. CRÉATION D'UN PLAN DE CONSTRUCTION D'IMAGES À L'AIDE DE L'INTERFACE DE LIGNE DE COMMANDE	14
4.3. ÉDITION D'UN PLAN DE CONSTRUCTION D'IMAGES À L'AIDE DE L'INTERFACE EN LIGNE DE COMMANDE	16
4.4. CRÉATION D'UNE IMAGE SYSTÈME AVEC IMAGE BUILDER DANS L'INTERFACE DE LIGNE DE COMMANDE	18
4.5. COMMANDES DE BASE DE LA LIGNE DE COMMANDE DU CONSTRUCTEUR D'IMAGES	19
4.6. FORMAT DU PLAN DE CONSTRUCTION DE L'IMAGE	20
4.7. PERSONNALISATIONS D'IMAGES PRISES EN CHARGE	21
4.8. PAQUETS INSTALLÉS PAR LE CONSTRUCTEUR D'IMAGES	29
4.9. ACTIVATION DES SERVICES SUR LES IMAGES PERSONNALISÉES	33
CHAPITRE 5. CRÉATION D'IMAGES SYSTÈME À L'AIDE DE L'INTERFACE DE LA CONSOLE WEB IMAGE BUILDER	34
5.1. ACCÈS AU TABLEAU DE BORD DU CONSTRUCTEUR D'IMAGES DANS LA CONSOLE WEB RHEL	34
5.2. CRÉATION D'UN PLAN DE CONSTRUCTION D'IMAGES DANS L'INTERFACE DE LA CONSOLE WEB	34
5.3. IMPORTATION D'UN BLUEPRINT DANS L'INTERFACE DE LA CONSOLE WEB DU CONSTRUCTEUR D'IMAGES	38
5.4. EXPORTATION D'UN BLUEPRINT À PARTIR DE L'INTERFACE DE LA CONSOLE WEB DU CONSTRUCTEUR D'IMAGES	39
5.5. CRÉATION D'UNE IMAGE SYSTÈME À L'AIDE DU CONSTRUCTEUR D'IMAGES DANS L'INTERFACE DE LA CONSOLE WEB	39
CHAPITRE 6. CRÉATION D'UNE IMAGE D'INSTALLATION ISO DE DÉMARRAGE AVEC LE CONSTRUCTEUR D'IMAGES	41
6.1. CRÉATION D'UNE IMAGE D'INSTALLATION ISO DE DÉMARRAGE À L'AIDE DU CONSTRUCTEUR D'IMAGES DANS L'INTERFACE DE LIGNE DE COMMANDE	41
6.2. CRÉATION D'UNE IMAGE D'INSTALLATION ISO DE DÉMARRAGE À L'AIDE DU GÉNÉRATEUR D'IMAGES DANS L'INTERFACE GRAPHIQUE	42
6.3. INSTALLATION D'UNE IMAGE ISO D'UN CONSTRUCTEUR D'IMAGES SUR UN SYSTÈME BARE METAL	43
CHAPITRE 7. CRÉATION D'IMAGES PRÉ-DURCIES AVEC L'INTÉGRATION DE L'IMAGE BUILDER OPENS CAP	45

7.1. DIFFÉRENCES ENTRE LES IMAGES KICKSTART ET LES IMAGES PRÉ-DURCIES	45
7.2. LA PERSONNALISATION DU MODÈLE OPENSCAP	45
7.3. CRÉATION D'UNE IMAGE PRÉ-DURCIE AVEC IMAGE BUILDER	46
CHAPITRE 8. PRÉPARATION ET DÉPLOIEMENT D'IMAGES D'INVITÉS KVM À L'AIDE DU CONSTRUCTEUR D'IMAGES	48
8.1. CRÉATION D'IMAGES D'INVITÉS KVM PERSONNALISÉES À L'AIDE DU CONSTRUCTEUR D'IMAGES	48
8.2. CRÉER UNE MACHINE VIRTUELLE À PARTIR D'UNE IMAGE D'INVITÉ KVM	49
CHAPITRE 9. POUSSER UN CONTENEUR VERS UN REGISTRE ET L'INTÉGRER DANS UNE IMAGE	52
9.1. PERSONNALISATION DU BLUEPRINT POUR INTÉGRER UN CONTENEUR DANS UNE IMAGE	52
9.2. LES INFORMATIONS D'IDENTIFICATION DU REGISTRE DES CONTENEURS	52
9.3. POUSSER UN ARTEFACT DE CONTENEUR DIRECTEMENT VERS UN REGISTRE DE CONTENEURS	53
9.4. CONSTRUCTION D'UNE IMAGE ET INTÉGRATION DU CONTENEUR DANS L'IMAGE	54
CHAPITRE 10. PRÉPARATION ET TÉLÉCHARGEMENT D'IMAGES DE NUAGES À L'AIDE DE L'OUTIL DE CRÉATION D'IMAGES	57
10.1. PRÉPARATION DU TÉLÉCHARGEMENT DES IMAGES AWS AMI	57
10.2. TÉLÉCHARGEMENT D'UNE IMAGE AMI SUR AWS À L'AIDE DE LA CLI	58
10.3. POUSSER DES IMAGES VERS AWS CLOUD AMI	59
10.4. PRÉPARATION AU TÉLÉCHARGEMENT D'IMAGES VHD MICROSOFT AZURE	61
10.5. TÉLÉCHARGEMENT D'IMAGES VHD DANS LE NUAGE MICROSOFT AZURE	63
10.6. POUSSER DES IMAGES VHD VERS LE NUAGE MICROSOFT AZURE	64
10.7. TÉLÉCHARGEMENT D'IMAGES VMDK ET CRÉATION D'UNE MACHINE VIRTUELLE RHEL DANS VSPHERE	66
10.8. TÉLÉCHARGEMENT D'IMAGES SUR GCP AVEC IMAGE BUILDER	68
10.9. POUSSER DES IMAGES VMDK VERS VSPHERE À L'AIDE DE L'OUTIL DE CRÉATION D'IMAGES GUI	71
10.10. TÉLÉCHARGEMENT D'IMAGES QCOW2 VERS OPENSTACK	73
10.11. PRÉPARATION DU TÉLÉCHARGEMENT D'IMAGES RHEL PERSONNALISÉES SUR ALIBABA	75
10.12. TÉLÉCHARGEMENT D'IMAGES RHEL PERSONNALISÉES SUR ALIBABA	76
10.13. IMPORTATION D'IMAGES SUR ALIBABA	77
10.14. CRÉATION D'UNE INSTANCE D'UNE IMAGE RHEL PERSONNALISÉE À L'AIDE D'ALIBABA	79

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. DESCRIPTION DU CONSTRUCTEUR D'IMAGES

Pour déployer un système sur une plateforme cloud, créez une image système. Pour créer des images système RHEL, utilisez l'outil de création d'images.

1.1. QU'EST-CE QUE LE CONSTRUCTEUR D'IMAGES ?

Vous pouvez utiliser Image builder pour créer des images système personnalisées de RHEL, y compris des images système préparées pour le déploiement sur des plateformes en nuage. Image builder gère automatiquement les détails de configuration pour chaque type de sortie et est donc plus facile à utiliser et plus rapide que les méthodes manuelles de création d'images. Vous pouvez accéder à la fonctionnalité de construction d'images par le biais d'une interface de ligne de commande dans l'outil **composer-cli**, ou d'une interface utilisateur graphique dans la console web RHEL.



NOTE

À partir de RHEL 8.3, le back-end **osbuild-composer** remplace **lorax-composer**. Le nouveau service fournit des API REST pour la création d'images.

1.2. TERMINOLOGIE DES CONSTRUCTEURS D'IMAGES

Schéma directeur

Un plan est une description de l'image d'un système personnalisé. Il énumère les paquets et les personnalisations qui feront partie du système. Vous pouvez modifier les plans avec des personnalisations et les enregistrer dans une version particulière. Lorsque vous créez une image système à partir d'un blueprint, l'image est associée au blueprint dans l'interface de construction d'image de la console web RHEL.

Vous pouvez créer des plans au format TOML.

Composer

Les compositions sont des constructions individuelles d'une image système, basées sur une version spécifique d'un plan particulier. Le terme "composition" fait référence à l'image du système, aux journaux de sa création, aux entrées, aux métadonnées et au processus lui-même.

Personnalisations

Les personnalisations sont des spécifications de l'image qui ne sont pas des paquets. Cela inclut les utilisateurs, les groupes et les clés SSH.

1.3. FORMATS DE SORTIE DU CONSTRUCTEUR D'IMAGES

Image builder peut créer des images dans plusieurs formats de sortie présentés dans le tableau suivant. Pour vérifier les types pris en charge, exécutez la commande :

```
# composer-cli compose types
```

Tableau 1.1. Formats de sortie du constructeur d'images

Description	Nom du CLI	extension de fichier
QEMU QCOW2 Image	qcow2	.qcow2

Description	Nom du CLI	extension de fichier
Archives TAR	tar	.tar
Disque Amazon Machine Image	ami	.raw
Image disque Azure	vhd	.vhd
Google Cloud Platform	gce	.vhd
Disque de machine virtuelle VMware	vmdk	.vmdk
Openstack	openstack	.qcow2
RHEL pour Edge Commit	edge-commit	.tar
RHEL pour Edge Container	edge-container	.tar
RHEL pour Edge Installer	edge-installer	.iso
RHEL pour Edge Raw	edge-raw-image	.tar
Installation simplifiée de RHEL for Edge	edge-simplified-installer	.iso
Image ISO	image-installer	.iso

1.4. CONFIGURATION REQUISE POUR LE CONSTRUCTEUR D'IMAGES

L'environnement dans lequel s'exécute le constructeur d'images, par exemple une machine virtuelle dédiée, doit répondre aux exigences énumérées dans le tableau suivant.

Tableau 1.2. Configuration requise pour le constructeur d'images

Paramètres	Valeur minimale requise
Type de système	Une machine virtuelle dédiée. Notez que le constructeur d'images n'est pas pris en charge par les conteneurs, y compris les images de base universelles de Red Hat (UBI).
Processeur	2 cœurs
Mémoire	4 GiB
Espace disque	20 GiB d'espace libre dans le système de fichiers /var
Privilèges d'accès	Niveau administrateur (root)

Paramètres	Valeur minimale requise
Réseau	Connectivité Internet



NOTE

Si vous ne disposez pas d'une connectivité Internet, vous pouvez utiliser le générateur d'images dans des réseaux isolés si vous le reconfigurez de manière à ce qu'il ne se connecte pas à Red Hat Content Delivery Network (CDN). Pour cela, vous devez remplacer les référentiels par défaut pour qu'ils pointent vers vos référentiels locaux. Assurez-vous que votre contenu est mis en miroir en interne ou utilisez Red Hat Satellite. Voir [Gestion des référentiels](#) pour plus de détails.

Ressources supplémentaires

- [Provisioning to Satellite using a Red Hat image builder image](#)

CHAPITRE 2. INSTALLATION DU CONSTRUCTEUR D'IMAGES

Avant d'utiliser le constructeur d'images, vous devez l'installer dans une machine virtuelle.

2.1. INSTALLER LE CONSTRUCTEUR D'IMAGES DANS UNE MACHINE VIRTUELLE

Pour installer le constructeur d'images sur une machine virtuelle dédiée, procédez comme suit :

Conditions préalables

- Vous devez être connecté à une VM RHEL.
- La VM pour le constructeur d'images doit être en cours d'exécution et souscrite au gestionnaire d'abonnement Red Hat (RHSM) ou à Red Hat Satellite.

Procédure

1. Installez le générateur d'images et les autres paquets nécessaires sur la VM :

- **osbuild-composer** - pris en charge à partir de RHEL 8.3
- **composer-cli**
- **cockpit-composer**
- **bash-completion**

```
# dnf install osbuild-composer composer-cli cockpit-composer bash-completion
```

La console web est installée en tant que dépendance du paquetage *cockpit-composer*.

2. Activer le démarrage du constructeur d'images après chaque redémarrage :

```
# systemctl enable --now osbuild-composer.socket  
# systemctl enable --now cockpit.socket
```

Les services **osbuild-composer** et **cockpit** démarrent automatiquement lors du premier accès.

3. Charger le script de configuration de l'interpréteur de commandes pour que la fonction d'autocomplétion de la commande **composer-cli** fonctionne immédiatement sans redémarrage :

```
$ source /etc/bash_completion.d/composer-cli
```



IMPORTANT

Le paquetage **osbuild-composer** est le nouveau moteur dorsal qui sera le moteur par défaut préféré et le point central de toutes les nouvelles fonctionnalités à partir de Red Hat Enterprise Linux 8.3 et des versions ultérieures. L'ancien paquetage backend **lorax-composer** est considéré comme déprécié, ne recevra que certaines corrections pour le reste du cycle de vie de Red Hat Enterprise Linux 8 et sera omis dans les futures versions majeures. Il est recommandé de désinstaller **lorax-composer** en faveur d'**osbuild-composer**.

Vérification

Vous pouvez utiliser un journal système pour suivre les activités du service de construction d'images. En outre, vous pouvez trouver les messages du journal dans le fichier.

- Pour trouver la sortie du journal pour traceback, exécutez les commandes suivantes :

```
$ journalctl | grep osbuild
```

- Pour afficher les travailleurs locaux ou distants :

```
$ journalctl -u osbuild-worker*
```

- Pour afficher les services en cours d'exécution :

```
$ journalctl -u osbuild-composer.service
```

CHAPITRE 3. GESTION DES RÉFÉRENTIELS DE CONSTRUCTEURS D'IMAGES

3.1. DÉPÔTS PAR DÉFAUT DU CONSTRUCTEUR D'IMAGES

Le back-end **osbuild-composer** n'hérite pas des référentiels système situés dans le répertoire **/etc/yum.repos.d/**. Au lieu de cela, il dispose de son propre ensemble de dépôts officiels définis dans le répertoire **/usr/share/osbuild-composer/repositories**. Cela inclut le dépôt officiel Red Hat, qui contient les RPMs du système de base pour installer des logiciels supplémentaires ou mettre à jour des programmes déjà installés vers des versions plus récentes. Si vous souhaitez remplacer les référentiels officiels, vous devez définir des dérogations dans le répertoire **/etc/osbuild-composer/repositories**. Ce répertoire est destiné aux dérogations définies par l'utilisateur et les fichiers qui s'y trouvent sont prioritaires sur ceux du répertoire **/usr**.

Les fichiers de configuration n'ont pas le format habituel des référentiels DNF, que l'on connaît grâce aux fichiers de **/etc/yum.repos.d/**. Il s'agit plutôt de simples fichiers JSON.

3.2. REMPLACER UN RÉFÉRENTIEL SYSTÈME

Vous pouvez configurer un remplacement de référentiel pour le constructeur d'images dans le répertoire **/etc/osbuild-composer/repositories** en suivant les étapes suivantes.

Conditions préalables

- Vous disposez d'un référentiel personnalisé accessible depuis le système hôte

Procédure

1. Créez un répertoire dans lequel vous souhaitez stocker vos dérogations au référentiel :

```
$ sudo mkdir -p /etc/osbuild-composer/repositories
```

2. Vous pouvez créer votre propre structure de fichier JSON.
3. Créez un fichier JSON, en utilisant un nom correspondant à votre version de RHEL. Vous pouvez également copier le fichier correspondant à votre distribution à partir de **/usr/share/osbuild-composer/** et en modifier le contenu.
Pour RHEL 9, utilisez **/etc/osbuild-composer/repositories/rhel-92.json**.
4. Ajoutez la structure suivante à votre fichier JSON, par exemple :

```
{
  "<ARCH>": [
    {
      "name": "baseos",
      "baseurl": "http://mirror.example.com/composes/released/RHEL-9/9.0/BaseOS/x86_64/os/",
      "gpgkey": "-----BEGIN PGP PUBLIC KEY BLOCK-----\n\n (...)",
      "check_gpg": true,
      "metadata_expire": ""
    }
  ]
}
```

Ne spécifiez qu'un seul des attributs suivants :

- **baseurl** - chaîne de caractères : URL de base du référentiel.
 - **metalink** - string : URL d'un fichier metalink contenant une liste de dépôts miroirs valides.
 - **mirrorlist** - string : URL d'un fichier mirrorlist contenant une liste de dépôts miroirs valides
- Les autres champs sont facultatifs.

a. Vous pouvez également copier le fichier JSON pour votre distribution.

- i. Copiez le fichier du référentiel dans le répertoire que vous avez créé. Dans la commande suivante, remplacez **rhel-version.json** par votre version de RHEL, par exemple : rhel-9.json.

```
$ cp /usr/share/osbuild-composer/repositories/rhel-version.json /etc/osbuild-composer/repositories/
```

5. À l'aide d'un éditeur de texte, modifiez les chemins d'accès **baseurl** dans le fichier **rhel-9.json** et enregistrez-le. Par exemple :

```
$ vi /etc/osbuild-composer/repositories/rhel-version.json
```

6. Redémarrer le site **osbuild-composer.service**:

```
$ sudo systemctl restart osbuild-composer.service
```

Vérification

- Vérifier que le référentiel pointe vers les bonnes URL :

```
$ cat /etc/yum.repos.d/redhat.repo
```

Vous pouvez voir que le référentiel pointe vers les bonnes URLs qui sont copiées depuis le fichier **/etc/yum.repos.d/redhat.repo**.

Ressources supplémentaires

- [La dernière version des RPMs disponible dans le dépôt n'est pas visible pour **osbuild-composer**.](#)

3.3. REMPLACER UN RÉFÉRENTIEL SYSTÈME PAR UN SUPPORT POUR LES ABONNEMENTS

Le service **osbuild-composer** peut utiliser des abonnements système définis dans le fichier **/etc/yum.repos.d/redhat.repo**. Pour utiliser un abonnement système dans **osbuild-composer**, définissez une surcharge de référentiel qui a :

- Le même **baseurl** que le référentiel défini dans **/etc/yum.repos.d/redhat.repo**.
- La valeur de **"rhsm"**: **true** définie dans l'objet JSON.

Conditions préalables

- Votre système dispose d'un abonnement défini dans `/etc/yum.repos.d/redhat.repo`
- Vous avez créé une substitution de référentiel. Voir [Remplacement d'un référentiel système](#).

Procédure

1. Obtenez le **baseurl** à partir du fichier `/etc/yum.repos.d/redhat.repo`:

```
# cat /etc/yum.repos.d/redhat.repo
[AppStream]
name = AppStream mirror example
baseurl = https://mirror.example.com/RHEL-9/9.0/AppStream/x86_64/os/
enabled = 1
gpgcheck = 0
sslverify = 1
sslcacert = /etc/pki/ca1/ca.crt
sslclientkey = /etc/pki/ca1/client.key
sslclientcert = /etc/pki/ca1/client.crt
metadata_expire = 86400
enabled_metadata = 0
```

2. Configurez le référentiel pour qu'il utilise le même **baseurl** et attribuez la valeur `true` à **rhsm**:

```
{
  "x86_64": [
    {
      "name": "AppStream mirror example",
      "baseurl": "https://mirror.example.com/RHEL-9/9.0/AppStream/x86_64/os/",
      "gpgkey": "-----BEGIN PGP PUBLIC KEY BLOCK-----\n\n (...)",
      "check_gpg": true,
      "rhsm": true
    }
  ]
}
```



NOTE

osbuild-composer n'utilise pas automatiquement les référentiels définis à l'adresse `/etc/yum.repos.d/`. Vous devez les spécifier manuellement, soit en tant que remplacement du référentiel système, soit en tant que **source** supplémentaire à l'aide de **composer-cli**. Les dérogations au référentiel système sont généralement utilisées pour les référentiels "BaseOS" et "AppStream", tandis que les sources **composer-cli** sont utilisées pour tous les autres référentiels.

Par conséquent, le constructeur d'images lit le fichier `/etc/yum.repos.d/redhat.repo` à partir du système hôte et l'utilise comme source d'abonnements.

Ressources supplémentaires

- [Le constructeur d'images utilise les dépôts CDN lorsque l'hôte est enregistré sur Satellite 6](#)

CHAPITRE 4. CRÉATION D'IMAGES SYSTÈME À L'AIDE DE L'INTERFACE DE LIGNE DE COMMANDE IMAGE BUILDER

Image builder est un outil permettant de créer des images système personnalisées. Pour contrôler Image builder et créer vos images système personnalisées, vous pouvez utiliser l'interface de ligne de commande (CLI) ou l'interface de la console web. Actuellement, l'interface de ligne de commande est la méthode préférée pour utiliser le constructeur d'images.

4.1. PRÉSENTATION DE L'INTERFACE EN LIGNE DE COMMANDE DU CONSTRUCTEUR D'IMAGES

L'interface de ligne de commande (CLI) d'image builder est actuellement la méthode préférée pour utiliser image builder. Elle offre plus de fonctionnalités que l'[interface de la console web](#). Pour utiliser l'interface de commande, exécutez la commande **composer-cli** avec les options et sous-commandes appropriées.

Le déroulement des opérations pour l'interface de ligne de commande peut être résumé comme suit :

1. Exporter (*save*) la définition du plan vers un fichier texte
2. Modifier ce fichier dans un éditeur de texte
3. Importer (*push*) le fichier texte du plan dans Image Builder
4. Exécuter une composition pour créer une image à partir du plan
5. Exporter le fichier image pour le télécharger

Outre les sous-commandes de base permettant de réaliser cette procédure, la commande **composer-cli** propose de nombreuses sous-commandes permettant d'examiner l'état des plans et des compositions configurés.

Pour exécuter les commandes **composer-cli** en tant que non-root, l'utilisateur doit faire partie des groupes **weldr** ou **root**.

- Pour ajouter un utilisateur aux groupes **weldr** ou **root**, exécutez les commandes suivantes :

```
$ sudo usermod -a -G weldr user  
$ newgrp weldr
```

4.2. CRÉATION D'UN PLAN DE CONSTRUCTION D'IMAGES À L'AIDE DE L'INTERFACE DE LIGNE DE COMMANDE

Vous pouvez créer un nouveau modèle de construction d'image à l'aide de l'interface de ligne de commande (CLI). Le blueprint décrit l'image finale et ses personnalisations, telles que les paquets et les personnalisations du noyau.

Prérequis

- Accès à l'outil de création d'images.

Procédure

1. Créez un fichier de texte brut avec le contenu suivant :

```
name = "BLUEPRINT-NAME"
description = "LONG FORM DESCRIPTION TEXT"
version = "0.0.1"
modules = []
groups = []
```

Remplacez *BLUEPRINT-NAME* et *LONG FORM DESCRIPTION TEXT* par un nom et une description de votre plan.

Remplacer *0.0.1* par un numéro de version conformément au schéma de [versionnement sémantique](#).

2. Pour chaque paquet que vous souhaitez inclure dans le plan, ajoutez les lignes suivantes au fichier :

```
[[packages]]
name = "package-name"
version = "package-version"
```

Remplacez *package-name* par le nom du paquet, par exemple **httpd, gdb-docou coreutils**.

Remplacez *package-version* par la version à utiliser. Ce champ prend en charge les spécifications de la version **dnf**:

- Pour une version spécifique, utilisez le numéro de version exact, par exemple **8.7.0**.
- Pour obtenir la dernière version disponible, utilisez l'astérisque *****.
- Pour la dernière version mineure, utilisez des formats tels que **8.***.

3. Personnalisez vos plans en fonction de vos besoins. Par exemple, pour désactiver le multithreading simultané (SMT), ajoutez les lignes suivantes au fichier blueprint :

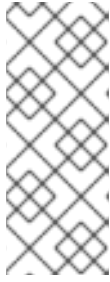
```
[customizations.kernel]
append = "nosmt=force"
```

Pour connaître les autres personnalisations disponibles, voir [Personnalisations d'images prises en charge](#).

4. Enregistrez le fichier, par exemple, sous *BLUEPRINT-NAME.toml* et fermez l'éditeur de texte.
5. Pousser (importer) le plan :

```
# composer-cli blueprints push BLUEPRINT-NAME.toml
```

Remplacez *BLUEPRINT-NAME* par la valeur que vous avez utilisée dans les étapes précédentes.



NOTE

Pour créer des images en utilisant **composer-cli** en tant que non-root, ajoutez votre utilisateur aux groupes **weldr** ou **root**.

```
# usermod -a -G weldr user
$ newgrp weldr
```

Vérification

- Lister les plans existants pour vérifier que le plan a été poussé et qu'il existe :

```
# composer-cli blueprints list
```

- Affichez la configuration du plan que vous venez d'ajouter :

```
# composer-cli blueprints show BLUEPRINT-NAME
```

- Vérifiez si les composants et les versions répertoriés dans le plan et leurs dépendances sont valides :

```
# composer-cli blueprints depsolve BLUEPRINT-NAME
```

Si le constructeur d'images n'est pas en mesure de résoudre un paquet à partir de vos dépôts personnalisés, suivez les étapes suivantes :

- Supprimer le cache de osbuild-composer :

```
$ sudo rm -rf /var/cache/osbuild-composer/*
$ sudo systemctl restart osbuild-composer
```

Ressources supplémentaires

- [osbuild-composer](#) n'est pas en mesure de résoudre un paquet provenant de mon dépôt personnalisé.
- [Composer une image système RHEL personnalisée avec un serveur proxy](#) .

4.3. ÉDITION D'UN PLAN DE CONSTRUCTION D'IMAGES À L'AIDE DE L'INTERFACE EN LIGNE DE COMMANDE

Vous pouvez modifier un plan de construction d'images existant dans l'interface de ligne de commande (CLI) pour, par exemple, ajouter un nouveau paquet, ou définir un nouveau groupe, et créer vos images personnalisées. Pour cela, suivez les étapes suivantes :

Conditions préalables

- Vous avez créé un schéma directeur.

Procédure

1. Enregistrer (exporter) le plan dans un fichier texte local :

-

```
# composer-cli blueprints save BLUEPRINT-NAME
```

2. Modifiez le fichier *BLUEPRINT-NAME.toml* à l'aide d'un éditeur de texte et apportez vos modifications.
3. Avant de terminer les modifications, vérifiez que le fichier est un plan valide :
 - a. Supprimer cette ligne, si elle existe :

```
packages = []
```

- b. Augmentez le numéro de version, par exemple de 0.0.1 à 0.1.0. N'oubliez pas que les versions des plans de construction d'images doivent utiliser le système de [versionnement sémantique](#). Notez également que si vous ne modifiez pas la version, le composant de version **patch** augmente automatiquement.
- c. Vérifier si le contenu est une spécification TOML valide. Voir la [documentation TOML](#) pour plus d'informations.



NOTE

La documentation TOML est un produit communautaire et n'est pas pris en charge par Red Hat. Vous pouvez signaler tout problème lié à l'outil à l'adresse suivante : <https://github.com/toml-lang/toml/issues>

4. Enregistrez le fichier et fermez l'éditeur de texte.
5. Repousser (importer) le plan dans le constructeur d'images :

```
# composer-cli blueprints push BLUEPRINT-NAME.toml
```



NOTE

Pour réimporter le plan dans Image Builder, il faut fournir le nom du fichier avec l'extension **.toml**, alors que dans les autres commandes, il faut utiliser uniquement le nom du plan.

6. Pour vérifier que le contenu téléchargé dans le générateur d'images correspond à vos modifications, dressez la liste du contenu du plan :

```
# composer-cli blueprints show BLUEPRINT-NAME
```

7. Vérifiez si les composants et les versions répertoriés dans le plan et leurs dépendances sont valides :

```
# composer-cli blueprints depsolve BLUEPRINT-NAME
```

Ressources supplémentaires

- [Personnalisations d'images prises en charge](#) .

4.4. CRÉATION D'UNE IMAGE SYSTÈME AVEC IMAGE BUILDER DANS L'INTERFACE DE LIGNE DE COMMANDE

Vous pouvez créer une image personnalisée à l'aide de l'interface de ligne de commande du constructeur d'images.

Conditions préalables

- Vous avez préparé un plan pour l'image. Voir [Création d'un plan de construction d'image à l'aide de l'interface de ligne de commande](#).

Procédure

1. Commencez la composition :

```
# composer-cli compose start BLUEPRINT-NAME IMAGE-TYPE
```

Remplacez *BLUEPRINT-NAME* par le nom du plan et *IMAGE-TYPE* par le type d'image. Pour connaître les valeurs disponibles, voir la sortie de la commande **composer-cli compose types**.

Le processus de composition démarre en arrière-plan et affiche l'identifiant universel unique (UUID) du compositeur.

2. Attendez que le processus de composition soit terminé. La création d'une image peut prendre jusqu'à dix minutes.
Pour vérifier l'état de la composition :

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

Une composition terminée affiche la valeur de l'état **FINISHED**. Pour identifier votre composition dans la liste, utilisez son UUID.

3. Une fois le processus de composition terminé, téléchargez le fichier image résultant :

```
# composer-cli compose image UUID
```

Remplacez *UUID* par la valeur UUID indiquée dans les étapes précédentes.

Vérification

Après avoir créé votre image, vous pouvez vérifier la progression de la création de l'image à l'aide des commandes suivantes :

- Vérifier l'état de la composition :

```
$ sudo composer-cli compose status
```

- Télécharger les métadonnées de l'image :

```
sudo composer-cli compose metadata UUID
```

- Télécharger les logs de l'image :

```
sudo composer-cli compose logs UUID
```

La commande crée un fichier **.tar** qui contient les journaux de la création de l'image. Si les journaux sont vides, vous pouvez vérifier le journal.

- Vérifier le journal :

```
$ journalctl | grep osbuild
```

- Vérifier le manifeste :

```
$ sudo cat /var/lib/osbuild-composer/jobs/job_UUID.json
```

Vous trouverez le fichier *job_UUID.json* dans le journal.

Ressources supplémentaires

- [Constructeur d'images de traçage](#)

4.5. COMMANDES DE BASE DE LA LIGNE DE COMMANDE DU CONSTRUCTEUR D'IMAGES

L'interface de ligne de commande du constructeur d'images propose les sous-commandes suivantes.

Manipulation de plans

Liste de tous les plans disponibles

```
# composer-cli blueprints list
```

Afficher le contenu d'un plan au format TOML

```
# composer-cli blueprints show BLUEPRINT-NAME
```

Sauvegarder (exporter) le contenu du plan au format TOML dans un fichier **BLUEPRINT-NAME.toml**

```
# composer-cli blueprints save BLUEPRINT-NAME
```

Supprimer un plan

```
# composer-cli blueprints delete BLUEPRINT-NAME
```

Introduire (importer) un fichier de plan au format TOML dans le constructeur d'images

```
# composer-cli blueprints push BLUEPRINT-NAME
```

Composer des images à partir de plans

Liste des types d'images disponibles

```
# composer-cli compose types
```

Commencer une composition

```
# composer-cli compose start BLUEPRINT COMPOSE-TYPE
```

Remplacez *BLUEPRINT* par le nom du plan à construire et *COMPOSE-TYPE* par le type d'image de sortie.

Liste de tous les compositeurs

```
# composer-cli compose list
```

Liste de toutes les compositions et de leur statut

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

Annuler une composition en cours

```
# composer-cli composer cancel COMPOSE-UUID
```

Supprimer une composition terminée

```
# composer-cli composer delete COMPOSE-UUID
```

Afficher des informations détaillées sur une composition

```
# composer-cli composer info COMPOSE-UUID
```

Télécharger le fichier image d'une composition

```
# composer-cli composer image COMPOSE-UUID
```

Voir d'autres sous-commandes et options

```
# composer-cli help
```

Ressources supplémentaires

- [composer-cli\(1\)](#) page de manuel

4.6. FORMAT DU PLAN DE CONSTRUCTION DE L'IMAGE

Les plans des constructeurs d'images sont présentés à l'utilisateur sous forme de texte brut au format TOML.

Les éléments d'un fichier de plan type sont les suivants :

Les métadonnées du plan directeur

```
name = "BLUEPRINT-NAME"  
description = "LONG FORM DESCRIPTION TEXT"
```



```
version = "VERSION"
```

Les champs *BLUEPRINT-NAME* et *LONG FORM DESCRIPTION TEXT* donnent un nom et une description à votre plan.

Le site *VERSION* est un numéro de version selon le schéma de [versionnement sémantique](#).

Cette partie n'est présente qu'une seule fois pour l'ensemble du fichier de plan.

L'entrée *modules* répertorie les noms et les versions des paquets à installer dans l'image.

L'entrée *group* décrit un groupe de paquets à installer dans l'image. Les groupes utilisent les catégories de paquets suivantes :

- Obligatoire
- Défaut
- En option

Les Blueprints installent les paquets obligatoires et les paquets par défaut. Il n'y a pas de mécanisme de sélection des paquets optionnels.

Groupes à inclure dans l'image

```
[[groups]]
name = "group-name"
```

Le *group-name* est le nom du groupe, par exemple, **anaconda-tools**, **widget**, **wheel** ou **users**.

Paquets à inclure dans l'image

```
[[packages]]
name = "package-name"
version = "package-version"
```

package-name est le nom du paquet, par exemple **httpd**, **gdb-docou** **coreutils**.

package-version est une version à utiliser. Ce champ prend en charge les spécifications de la version **dnf**:

- Pour une version spécifique, utilisez le numéro de version exact, par exemple **8.7.0**.
- Pour connaître la dernière version disponible, utilisez l'astérisque *****.
- Pour la dernière version mineure, utilisez un format tel que **8.***.

Répétez ce bloc pour chaque paquet à inclure.



NOTE

Actuellement, il n'y a pas de différence entre les paquets et les modules dans l'outil de construction d'images. Les deux sont traités comme des dépendances de paquets RPM.

4.7. PERSONNALISATIONS D'IMAGES PRISES EN CHARGE

Vous pouvez personnaliser votre image en ajoutant à votre plan un paquetage RPM supplémentaire, en activant un service ou en personnalisant un paramètre de la ligne de commande du noyau. Vous pouvez utiliser plusieurs personnalisations d'image dans les blueprints. Pour utiliser ces options, vous devez configurer les personnalisations dans le blueprint et l'importer (push) dans le constructeur d'images.



NOTE

Ces personnalisations ne sont pas prises en charge lors de l'utilisation du constructeur d'images dans la console web.

Sélectionner un groupe de paquets

```
[[packages]]
name = "package_group_name"
```

Remplacez `"package_group_name"` par le nom du groupe. Par exemple, `"@server with gui"`.

Définir le nom d'hôte de l'image

```
[customizations]
hostname = "baseimage"
```

Spécifications utilisateur pour l'image système résultante

```
[[customizations.user]]
name = "USER-NAME"
description = "USER-DESCRIPTION"
password = "PASSWORD-HASH"
key = "PUBLIC-SSH-KEY"
home = "/home/USER-NAME"
shell = "/usr/bin/bash"
groups = ["users", "wheel"]
uid = NUMBER
gid = NUMBER
```

Le GID est facultatif et doit déjà exister dans l'image. Il peut être créé par un paquet ou par le plan directeur à l'aide de l'entrée `[[customizations.group]]`.



IMPORTANT

Pour générer le fichier **password hash**, vous devez installer **python3** sur votre système.

```
# dnf install python3
```

Remplacez `PASSWORD-HASH` par l'actuel **password hash**. Pour générer **password hash**, utilisez une commande telle que :

```
$ python3 -c 'import crypt,getpass;pw=getpass.getpass();print(crypt.crypt(pw) if
(pw==getpass.getpass("Confirm: ")) else exit())'
```

Remplacez `PUBLIC-SSH-KEY` par la clé publique réelle.

Remplacez les autres espaces réservés par des valeurs appropriées.

Vous devez saisir l'adresse **name**. Vous pouvez omettre toutes les lignes dont vous n'avez pas besoin.

Répétez ce bloc pour chaque utilisateur à inclure.

Spécifications du groupe pour l'image système résultante

```
[[customizations.group]]
name = "GROUP-NAME"
gid = NUMBER
```

Répétez ce bloc pour chaque groupe à inclure.

Définir la clé SSH d'un utilisateur existant

```
[[customizations.sshkey]]
user = "root"
key = "PUBLIC-SSH-KEY"
```



NOTE

La personnalisation "Définir la clé SSH d'un utilisateur existant" ne s'applique qu'aux utilisateurs existants. Pour créer un utilisateur et définir une clé SSH, voir la personnalisation **User specifications for the resulting system image**

Ajouter une option de paramètre de démarrage du noyau aux valeurs par défaut

```
[customizations.kernel]
append = "KERNEL-OPTION"
```

Par défaut, le constructeur d'images intègre un noyau par défaut dans l'image. Mais vous pouvez personnaliser le noyau avec la configuration suivante dans le blueprint

```
[customizations.kernel]
name = "KERNEL-rt"
```

Définir un nom de noyau à utiliser dans une image

```
[customizations.kernel.name]
name = "KERNEL-NAME"
```

Définir le fuseau horaire et les serveurs *Network Time Protocol* (NTP) pour l'image système résultante

```
[customizations.timezone]
timezone = "TIMEZONE"
ntpservers = "NTP_SERVER"
```

Si vous ne définissez pas de fuseau horaire, le système utilise par défaut *Universal Time, Coordinated* (UTC). La définition de serveurs NTP est facultative.

Définir les paramètres linguistiques de l'image système résultante

```
[customizations.locale]
languages = ["LANGUAGE"]
keyboard = "KEYBOARD"
```

Le paramétrage de la langue et des options du clavier est obligatoire. Vous pouvez ajouter de nombreuses autres langues. La première langue que vous ajoutez sera la langue principale et les autres langues seront secondaires. Par exemple, la première langue ajoutée sera la langue principale et les autres langues seront secondaires :

```
[customizations.locale]
languages = ["en_US.UTF-8"]
keyboard = "us"
```

Pour dresser la liste des valeurs prises en charge par les langues, exécutez la commande suivante :

```
$ localectl list-locales
```

Pour dresser la liste des valeurs prises en charge par le clavier, exécutez la commande suivante :

```
$ localectl list-keymaps
```

Définir le pare-feu pour l'image système résultante

```
[customizations.firewall]
port = ["PORTS"]
```

Pour activer les listes, vous pouvez utiliser des ports numériques ou leurs noms dans le fichier **/etc/services**.

Personnaliser les services de pare-feu

Examinez les services de pare-feu disponibles.

```
$ firewall-cmd --get-services
```

Dans le plan, sous la section **customizations.firewall.service**, indiquez les services de pare-feu que vous souhaitez personnaliser.

```
[customizations.firewall.services]
enabled = ["SERVICES"]
disabled = ["SERVICES"]
```

Les services répertoriés dans **firewall.services** sont différents des noms de services disponibles dans le fichier **/etc/services**.



NOTE

Si vous ne souhaitez pas personnaliser les services de pare-feu, omettez les sections **[customizations.firewall]** et **[customizations.firewall.services]** du plan.

Définir les services à activer pendant le démarrage

```
[customizations.services]
enabled = ["SERVICES"]
disabled = ["SERVICES"]
```

Vous pouvez contrôler les services à activer pendant le démarrage. Certains types d'images ont déjà des services activés ou désactivés pour s'assurer que l'image fonctionne correctement et cette configuration ne peut pas être remplacée. La personnalisation de **[customizations.services]** dans le blueprint ne remplace pas ces services, mais les ajoute à la liste des services déjà présents dans les modèles d'image.



NOTE

Chaque fois qu'une compilation démarre, elle clone le référentiel du système hôte. Si vous vous référez à un référentiel avec un historique important, le clonage peut prendre un certain temps et utiliser une quantité importante d'espace disque. De plus, le clone est temporaire et la compilation le supprime après avoir créé le paquetage RPM.

Spécifier une configuration personnalisée du système de fichiers

Vous pouvez spécifier une configuration de système de fichiers personnalisée dans vos blueprints et créer ainsi des images avec une disposition de disque spécifique, au lieu de la configuration par défaut. L'utilisation d'une configuration différente de la configuration par défaut dans vos blueprints vous permet de bénéficier des avantages suivants

- sécurité benchmark conformité
- protection contre les erreurs hors disque
- amélioration des performances
- cohérence avec les installations existantes

Pour personnaliser la configuration du système de fichiers dans votre projet :

```
[[customizations.filesystem]]
mountpoint = "MOUNTPOINT"
size = MINIMUM-PARTITION-SIZE
```

Le plan directeur prend en charge les **mountpoints** suivants et leurs sous-répertoires :

- / - le point de montage racine
- **/var**
- **/home**
- **/opt**
- **/srv**
- **/usr**
- **/app**

- **/data**
- **/boot** - Pris en charge à partir de RHEL 8.7 et RHEL 9.1.



NOTE

La personnalisation des points de montage n'est prise en charge qu'à partir des distributions RHEL 8.5 et RHEL 9.0, à l'aide de la CLI. Dans les distributions antérieures, vous pouvez uniquement spécifier la partition **root** comme point de montage et spécifier l'argument **size** comme alias pour la taille de l'image.

Si vous avez plus d'une partition dans l'image personnalisée, vous pouvez créer des images avec une partition de système de fichiers personnalisé sur LVM et redimensionner ces partitions au moment de l'exécution. Pour ce faire, vous pouvez spécifier une configuration de système de fichiers personnalisée dans votre plan et créer ainsi des images avec l'agencement de disque souhaité. La configuration par défaut du système de fichiers reste inchangée si vous utilisez des images simples sans personnalisation du système de fichiers, et **cloud-init** redimensionne la partition racine.



NOTE

À partir de la version 8.6, pour le RPM **osbuild-composer-46.1-1.el8** et les versions ultérieures, les partitions physiques ne sont plus disponibles et les personnalisations du système de fichiers créent des volumes logiques.

Le schéma directeur convertit automatiquement la personnalisation du système de fichiers en une partition LVM.

La valeur **MINIMUM-PARTITION-SIZE** n'a pas de format de taille par défaut. La personnalisation du plan directeur prend en charge les valeurs et unités suivantes : kB à TB et KiB à TiB. Par exemple, vous pouvez définir la taille du point de montage en octets :

```
[[customizations.filesystem]]
mountpoint = "/var"
size = 1073741824
```

Vous pouvez également définir la taille du point de montage en utilisant des unités.



NOTE

Vous ne pouvez définir la taille du point de montage qu'en utilisant les unités de la version du paquetage fournie pour les distributions RHEL 8.6 et RHEL 9.0.

Par exemple :

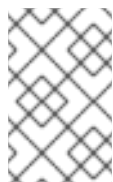
```
[[customizations.filesystem]]
mountpoint = "/opt"
size = "20 GiB"
```

or

```
[[customizations.filesystem]]
mountpoint = "/boot"
size = "1 GiB"
```

Créez des répertoires et des fichiers personnalisés pour votre image dans le répertoire `/etc`

Pour créer des fichiers et des répertoires personnalisés dans votre image, utilisez les personnalisations **[[customizations.files]]** et **[[customizations.directories]]**. Actuellement, vous ne pouvez utiliser ces personnalisations que dans le répertoire `/etc`.



NOTE

Ces personnalisations du plan directeur sont prises en charge par tous les types d'images, à l'exception des types d'images qui déploient des commits OSTree, tels que **edge-raw-image**, **edge-installer**, et **edge-simplified-installer**.

Create a custom directory blueprint customization

Avec la personnalisation de **[[customizations.directories]]** blueprint, vous pouvez créer des répertoires personnalisés dans le répertoire `/etc` de votre image.



AVERTISSEMENT

Si vous utilisez **customizations.directories** avec un chemin de répertoire qui existe déjà dans l'image avec **mode**, **user** ou **group**, la construction de l'image échoue pour empêcher la modification de la propriété ou des autorisations du répertoire existant.

Grâce à la personnalisation de **[[customizations.directories]]** blueprint, vous pouvez :

- Créer de nouveaux répertoires.
- Définissez la propriété de l'utilisateur et du groupe pour le répertoire que vous créez.
- Définit l'autorisation du mode répertoire au format octal.
- Veillez à ce que les répertoires parents soient créés si nécessaire.

Pour personnaliser la configuration d'un répertoire dans votre plan, créez un fichier avec le contenu suivant, par exemple :

```
[[customizations.directories]]
path = "/etc/directory_name"
mode = "octal_access_permission"
user = "user_string_or_integer"
group = "group_string_or_integer"
ensure_parents = boolean
```

Les entrées du plan sont décrites comme suit :

- **path** - Obligatoire - indiquez le chemin d'accès au répertoire que vous souhaitez créer. Il doit s'agir d'un chemin absolu sous le répertoire **/etc**.
- **mode** - Facultatif - définit l'autorisation d'accès au répertoire, au format octal. Si vous ne spécifiez pas d'autorisation, la valeur par défaut est 0755. Le zéro initial est facultatif.
- **user** - Facultatif - définit un utilisateur comme propriétaire du répertoire. Si vous ne spécifiez pas d'utilisateur, la valeur par défaut est **root**. Vous pouvez spécifier l'utilisateur sous la forme d'une chaîne de caractères ou d'un nombre entier.
- **group** - Facultatif - définit un groupe comme propriétaire du répertoire. Si vous ne spécifiez pas de groupe, la valeur par défaut est **root**. Vous pouvez spécifier le groupe sous la forme d'une chaîne de caractères ou d'un nombre entier.
- **ensure_parents** - Facultatif - Indiquez si vous souhaitez créer des répertoires parents si nécessaire. Si vous ne spécifiez pas de valeur, la valeur par défaut est **false**.

Create a custom file blueprint customization

Vous pouvez utiliser la personnalisation du plan de fichier pour créer de nouveaux fichiers ou remplacer des fichiers existants. Le répertoire parent du fichier que vous spécifiez doit exister, sinon la construction de l'image échoue. Assurez-vous que le répertoire parent existe en le spécifiant dans la personnalisation **[[customizations.directories]]**.



AVERTISSEMENT

Si vous combinez les personnalisations de fichiers avec d'autres personnalisations de plans, cela peut affecter le fonctionnement des autres personnalisations ou remplacer les personnalisations de fichiers actuelles. Si vous n'êtes pas sûr des personnalisations, utilisez la personnalisation de modèle appropriée.

Grâce à la personnalisation de **[[customizations.files]]** blueprint, vous pouvez :

- Créer de nouveaux fichiers texte.
- Modifier les fichiers existants. ATTENTION : cette modification peut remplacer le contenu existant.
- Définissez la propriété de l'utilisateur et du groupe pour le fichier que vous créez.
- Définit l'autorisation de mode au format octal.



NOTE

Vous ne pouvez pas créer ou remplacer les fichiers suivants :

- **/etc/fstab**
- **/etc/shadow**

- `/etc/passwd`
- `/etc/group`

Pour personnaliser un fichier dans votre plan, créez un fichier avec le contenu suivant, par exemple :

```
[[customizations.files]]
path = "/etc/directory_name"
mode = "octal_access_permission"
user = "user_string_or_integer"
group = "group_string_or_integer"
data = "Hello world!"
```

Les entrées du plan sont décrites comme suit :

- **path** - Obligatoire - indiquez le chemin d'accès au fichier que vous souhaitez créer. Il doit s'agir d'un chemin absolu sous le répertoire **/etc**.
- **mode** Facultatif - définit l'autorisation d'accès au fichier, au format octal. Si vous ne spécifiez pas d'autorisation, la valeur par défaut est 0644. Le zéro initial est facultatif.
- **user** - Facultatif - définit un utilisateur comme propriétaire du fichier. Si vous ne spécifiez pas d'utilisateur, la valeur par défaut est **root**. Vous pouvez spécifier l'utilisateur sous la forme d'une chaîne de caractères ou d'un nombre entier.
- **group** - Facultatif - définit un groupe comme propriétaire du fichier. Si vous ne spécifiez pas de groupe, la valeur par défaut est **root**. Vous pouvez spécifier le groupe sous la forme d'une chaîne de caractères ou d'un nombre entier.
- **data** - Facultatif - Spécifiez le contenu d'un fichier de texte brut. Si vous n'indiquez pas de contenu, un fichier vide est créé.

Ressources supplémentaires

- [L'importation du plan directeur échoue après l'ajout d'une personnalisation du système de fichiers.](#)
- [Référence du plan.](#)

4.8. PAQUETS INSTALLÉS PAR LE CONSTRUCTEUR D'IMAGES

Lorsque vous créez une image système à l'aide de l'outil de création d'images, le système installe un ensemble de paquets de base.

Tableau 4.1. Paquets par défaut pour la création de types d'images

Type d'image	Paquets par défaut
ami	checkpolicy, chrony, cloud-init, cloud-utils-growpart, @Core, dhcp-client, gdisk, insights-client, kernel, langpacks-en, net-tools, NetworkManager, redhat-release, redhat-release-eula, rng-tools, rsync, selinux-policy-targeted, tar, yum-utils

Type d'image	Paquets par défaut
openstack	@core, langpacks-en
qcow2	@core, chrony, dnf, kernel, dnf, nfs-utils, dnf-utils, cloud-init, python3-jjsonschema, qemu-guest-agent, cloud-utils-growpart, dracut-norescue, tar, tcpdump, rsync, dnf-plugin-spacewalk, rhn-client-tools, rhnlib, rhnsd, rhn-setup, NetworkManager, dhcp-client, cockpit-ws, cockpit-system, subscription-manager-cockpit, redhat-release, redhat-release-eula, rng-tools, insights-client
tar	polycoreutils, selinux-policy-targeted
vhd	@core, langpacks-en
vmdk	@core, chrony, cloud-init, firewalld, langpacks-en, open-vm-tools, selinux-policy-targeted
edge-commit	attr, audit, basesystem, bash, bash-completion, chrony, clevis, clevis-dracut, clevis-luks, container-selinux, coreutils, criu, cryptsetup, curl, dnsmasq, dosfstools, dracut-config-generic, dracut-network, e2fsprogs, firewalld, fuse-overlayfs, fwupd, glibc, glibc-minimal-langpack, gnupg2, greenboot, gzip, hostname, ima-evm-utils, iproute, iptables, iputils, keyutils, less, lvm2, NetworkManager, NetworkManager-wifi, NetworkManager-wwan, nss-altfiles, openssh-clients, openssh-server, passwd, pinentry, platform-python, podman, polycoreutils, polycoreutils-python-utils, polkit, procps-ng, redhat-release, rootfiles, rpm, rpm-ostree, rsync, selinux-policy-targeted, setools-console, setup, shadow-utils, shadow-utils, skopeo, slirp4netns, sudo, systemd, tar, tmux, traceroute, usbguard, util-linux, vim-minimal, wpa_supplicant, xz
edge-container	dnf, dosfstools, e2fsprogs, glibc, lorax-templates-generic, lorax-templates-rhel, lvm2, polycoreutils, python36, python3-iniparse, qemu-img, selinux-policy-targeted, systemd, tar, xfsprogs, xz

Type d'image	Paquets par défaut
edge-installer	aajohan-comfortaa-fonts, abattis-cantarell-fonts, alsa-firmware, alsa-tools-firmware, anaconda, anaconda-install-env-deps, anaconda-widgets, audit, bind-utils, bitmap-fangsongti-fonts, bzip2, cryptsetup, dbus-x11, dejavu-sans-fonts, dejavu-sans-mono-fonts, device-mapper-persistent-data, dnf, dump, ethtool, fcoe-utils, ftp, gdb-gdbserver, gdisk, gfs2-utils, glibc-all-langpacks, google-noto-sans-cjk-ttc-fonts, gsettings-desktop-schemas, hdparm, hexedit, initscripts, ipmitool, iwl3945-firmware, iwl4965-firmware, iwl6000g2a-firmware, iwl6000g2b-firmware, jomolhari-fonts, kacst-farsi-fonts, kacst-qurn-fonts, kbd, kbd-misc, kdump-anaconda-addon, khmeros-base-fonts, libblockdev-lvm-dbus, libertas-sd8686-firmware, libertas-sd8787-firmware, libertas-usb8388-firmware, libertas-usb8388-olpc-firmware, libibverbs, libreport-plugin-bugzilla, libreport-plugin-reportuploader, libreport-rhel-anaconda-bugzilla, librsvg2, linux-firmware, lklug-fonts, lldpad, lohit-assamese-fonts, lohit-bengali-fonts, lohit-devanagari-fonts, lohit-gujarati-fonts, lohit-gurmukhi-fonts, lohit-kannada-fonts, lohit-odia-fonts, lohit-tamil-fonts, lohit-telugu-fonts, lsof, madan-fonts, metacity, mtr, mt-st, net-tools, nmap-ncat, nm-connection-editor, nss-tools, openssh-server, oscap-anaconda-addon, pciutils, perl-interpreter, pigz, python3-pyatspi, rdma-core, redhat-release-eula, rpm-ostree, rsync, rsyslog, sg3_utils, sil-abyssinica-fonts, sil-padauk-fonts, sil-scheherazade-fonts, smartmontools, smc-meera-fonts, spice-vdagent, strace, system-storage-manager, thai-scalable-waree-fonts, tigervnc-server-minimal, tigervnc-server-module, udisks2, udisks2-iscsi, usbutils, vim-minimal, volume_key, wget, xfsdump, xorg-x11-drivers, xorg-x11-fonts-misc, xorg-x11-server-utils, xorg-x11-server-Xorg, xorg-x11-xauth

Type d'image	Paquets par défaut
edge-simplified-installer	attr, basesystem, binutils, bsdtar, clevis-dracut, clevis-luks, cloud-utils-growpart, coreos-installer, coreos-installer-dracut, coreutils, device-mapper-multipath, dnsmasq, dosfstools, dracut-live, e2fsprogs, fcoe-utils, fdo-init, gzip, ima-evm-utils, iproute, iptables, iputils, iscsi-initiator-utils, keyutils, lldpad, lvm2, passwd, policycoreutils, policycoreutils-python-utils, procps-ng, rootfiles, setools-console, sudo, traceroute, util-linux
image-installer	anaconda-dracut, curl, dracut-config-generic, dracut-network, hostname, iwl100-firmware, iwl1000-firmware, iwl105-firmware, iwl135-firmware, iwl2000-firmware, iwl2030-firmware, iwl3160-firmware, iwl5000-firmware, iwl5150-firmware, iwl6000-firmware, iwl6050-firmware, iwl7260-firmware, kernel, less, nfs-utils, openssh-clients, ostree, plymouth, prefixdevname, rng-tools, rpcbind, selinux-policy-targeted, systemd, tar, xfsprogs, xz
edge-raw-image	dnf, dosfstools, e2fsprogs, glibc, lorax-templates-generic, lorax-templates-rhel, lvm2, policycoreutils, python36, python3-iniparse, qemu-img, selinux-policy-targeted, systemd, tar, xfsprogs, xz
gce	@core, langpacks-en, acpid, dhcp-client, dnf-automatic, net-tools, python3, rng-tools, tar, vim



NOTE

Lorsque vous ajoutez des composants supplémentaires à votre plan, assurez-vous que les paquets des composants que vous avez ajoutés n'entrent pas en conflit avec d'autres composants de paquets. Dans le cas contraire, le système ne parvient pas à résoudre les dépendances et la création de votre image personnalisée échoue. Vous pouvez vérifier qu'il n'y a pas de conflit entre les paquets en exécutant la commande :

```
# composer-cli blueprints depsolve BLUEPRINT-NAME
```

Ressources supplémentaires

- [Description du constructeur d'images](#)

4.9. ACTIVATION DES SERVICES SUR LES IMAGES PERSONNALISÉES

Lorsque vous utilisez le constructeur d'images pour configurer une image personnalisée, les services par défaut utilisés par l'image sont déterminés par les éléments suivants :

- La version de RHEL sur laquelle vous utilisez l'utilitaire **osbuild-composer**
- Le type d'image

Par exemple, le type d'image **ami** active les services **sshd**, **chronyd** et **cloud-init** par défaut. Si ces services ne sont pas activés, l'image personnalisée ne démarre pas.

Tableau 4.2. Services activés pour soutenir la création de types d'images

Type d'image	Services activés par défaut
ami	sshd, cloud-init, cloud-init-local, cloud-config, cloud-final
openstack	sshd, cloud-init, cloud-init-local, cloud-config, cloud-final
qcow2	cloud-init
rhel-edge-commit	Aucun service supplémentaire n'est activé par défaut
tar	Aucun service supplémentaire n'est activé par défaut
vhd	sshd, chronyd, waagent, cloud-init, cloud-init-local, cloud-config, cloud-final
vmdk	sshd, chronyd, vmttoolsd, cloud-init

Remarque : vous pouvez personnaliser les services à activer lors du démarrage du système. Cependant, la personnalisation ne remplace pas les services activés par défaut pour les types d'images mentionnés.

Ressources supplémentaires

- [Personnalisations d'images prises en charge](#)

CHAPITRE 5. CRÉATION D'IMAGES SYSTÈME À L'AIDE DE L'INTERFACE DE LA CONSOLE WEB IMAGE BUILDER

Image builder est un outil permettant de créer des images système personnalisées. Pour contrôler Image builder et créer vos images système personnalisées, vous pouvez utiliser l'interface de la console web. Notez que l'[interface en ligne de commande](#) est l'alternative actuellement préférée, car elle offre plus de fonctionnalités.

5.1. ACCÈS AU TABLEAU DE BORD DU CONSTRUCTEUR D'IMAGES DANS LA CONSOLE WEB RHEL

Avec le plugin **cockpit-composer** pour la console web RHEL, vous pouvez gérer les blueprints et les composites des constructeurs d'images à l'aide d'une interface graphique.

Conditions préalables

- Vous devez avoir un accès root au système.
- Vous avez installé image builder.

Procédure

1. Ouvrir <https://localhost:9090/> dans un navigateur web sur le système où le générateur d'images est installé.
Pour plus d'informations sur l'accès à distance au constructeur d'images, voir [Managing systems using the RHEL web console](#) document.
2. Connectez-vous à la console web en tant qu'utilisateur root.
3. Pour afficher les commandes du constructeur d'images, cliquez sur l'icône **image builder**, dans le coin supérieur gauche de la fenêtre.
Le tableau de bord du constructeur d'images s'ouvre et répertorie les plans existants, le cas échéant.

Ressources supplémentaires

- [Création d'images système avec l'interface de ligne de commande image builder](#)

5.2. CRÉATION D'UN PLAN DE CONSTRUCTION D'IMAGES DANS L'INTERFACE DE LA CONSOLE WEB

La création d'un plan est une étape nécessaire avant de créer l'image système RHEL personnalisée. Toutes les personnalisations sont facultatives.



NOTE

Ces personnalisations de plans sont disponibles pour Red Hat Enterprise Linux 9.2 ou les versions ultérieures et Red Hat Enterprise Linux 8.8 ou les versions ultérieures.

Conditions préalables

- Vous avez ouvert l'application de construction d'images à partir de la console Web dans un navigateur. Voir [Accès à l'interface graphique du constructeur d'images dans la console web RHEL](#).

Procédure

1. Cliquez sur **Créer un plan** dans le coin supérieur droit.
Un assistant de dialogue contenant des champs pour le nom et la description du plan s'ouvre.
2. Sur la page **Details**:
 - a. Saisissez le nom du plan et, éventuellement, sa description. Cliquez sur **Suivant**.
3. Facultatif : dans la page **Packages**:
 - a. Sur la page de recherche **Available packages**, saisissez le nom du paquet et cliquez sur le bouton **>** pour le déplacer dans le champ Paquets choisis. Recherchez et incluez autant de paquets que vous le souhaitez. Cliquez sur **Suivant**.



NOTE

Ces personnalisations sont toutes facultatives, sauf indication contraire.

4. Sur la page **Kernel**, entrez un nom de noyau et les arguments de la ligne de commande.
5. Sur la page **File system**, vous pouvez sélectionner **Use automatic partitioning** ou **Manually configure partitions** pour votre système de fichiers image. Pour configurer manuellement les partitions, procédez comme suit :
 - a. Cliquez sur le bouton **Configurer manuellement les partitions**.
La section **Configure partitions** s'ouvre, montrant la configuration basée sur les standards et les guides de sécurité de Red Hat.
 - b. Dans le menu déroulant, fournissez les détails pour configurer les partitions :
 - i. Dans le champ **Mount point**, sélectionnez l'une des options de type de point de montage suivantes :
 - **/** - le point de montage racine
 - **/var**
 - **/home**
 - **/opt**
 - **/srv**
 - **/usr**
 - **/app**
 - **/data**
 - **/tmp**
 - **/usr/local**

Vous pouvez également ajouter un chemin supplémentaire au **Mount point**, tel que **/tmp**. Par exemple : **/var** comme préfixe et **/tmp** comme chemin supplémentaire donne **/var/tmp**.



NOTE

Selon le type de point de montage choisi, le type de système de fichiers devient **xfs**, et ainsi de suite.

- ii. Dans le champ **Minimum size partition** du système de fichiers, entrez la taille minimale de partition souhaitée. Dans le menu déroulant Taille minimale, vous pouvez utiliser des unités de taille courantes telles que **GiB**, **MiB** ou **KiB**. L'unité par défaut est **GiB**.



NOTE

Minimum size signifie que le constructeur d'images peut encore augmenter la taille des partitions, au cas où elles seraient trop petites pour créer une image fonctionnelle.

- c. Pour ajouter d'autres partitions, cliquez sur le bouton **Ajouter une partition**. Si le message d'erreur suivant s'affiche : "Duplicate partitions : Une seule partition peut être créée à chaque point de montage", vous pouvez.. :
 - i. Cliquez sur le bouton **Supprimer** pour supprimer la partition dupliquée.
 - ii. Choisissez un nouveau point de montage pour la partition que vous souhaitez créer.
 - d. Après avoir terminé la configuration du partitionnement, cliquez sur **Suivant**.
6. Sur la page **Services**, vous pouvez activer ou désactiver des services :
 - a. Saisissez les noms des services que vous souhaitez activer ou désactiver, en les séparant par une virgule, un espace ou en appuyant sur la touche **Entrée**. Cliquez sur **Suivant**.
 7. Sur la page **Firewall**, configurez votre pare-feu :
 - a. Saisissez l'adresse **Ports** et les services de pare-feu que vous souhaitez activer ou désactiver.
 - b. Cliquez sur le bouton **Ajouter une zone** pour gérer vos règles de pare-feu pour chaque zone indépendamment. Cliquez sur **Suivant**.
 8. Sur la page **Users**, ajoutez un utilisateur en suivant les étapes suivantes :
 - a. Cliquez sur **Ajouter un utilisateur**.
 - b. Saisissez un **Username**, un **password** et un **SSH key**. Vous pouvez également marquer l'utilisateur en tant qu'utilisateur privilégié en cochant la case **Server administrator**. Cliquez sur **Suivant**.
 9. Sur la page **Groups**, ajoutez des groupes en suivant les étapes suivantes :
 - a. Cliquez sur le bouton **Ajouter des groupes**:
 - i. Saisissez un **Group name** et un **Group ID**. Vous pouvez ajouter d'autres groupes. Cliquez sur **Suivant**.

10. Sur la page **SSH keys**, ajoutez une clé :
 - a. Cliquez sur le bouton **Ajouter une clé**.
 - i. Entrez la clé SSH.
 - ii. Saisissez une adresse **User**. Cliquez sur **Suivant**.
11. Sur la page **Timezone**, définissez vos paramètres de fuseau horaire :
 - a. Dans le champ **Timezone**, entrez le fuseau horaire que vous souhaitez ajouter à votre image système. Par exemple, ajoutez le format de fuseau horaire suivant : "US/Eastern".
Si vous ne définissez pas de fuseau horaire, le système utilise par défaut le temps universel coordonné (UTC).
 - b. Saisissez les serveurs **NTP**. Cliquez sur **Suivant**.
12. Sur la page **Locale**, effectuez les étapes suivantes :
 - a. Dans le champ de recherche **Keyboard**, saisissez le nom du paquet que vous souhaitez ajouter à votre image système. Par exemple, `[\N- "en_US.UTF-8\N"] : [\N-"en_US.UTF-8\N"]`.
 - b. Dans le champ de recherche **Languages**, saisissez le nom du paquet que vous souhaitez ajouter à votre image système. Par exemple : "us". Cliquez sur **Suivant**.
13. Sur la page **Others**, effectuez les étapes suivantes :
 - a. Dans le champ **Hostname**, entrez le nom d'hôte que vous souhaitez ajouter à votre image système. Si vous n'ajoutez pas de nom d'hôte, le système d'exploitation détermine le nom d'hôte.
 - b. Obligatoire uniquement pour l'image Simplifier Installer : Dans le champ **Installation Devices**, entrez un nœud valide pour votre image système. Par exemple : **dev/sda1**. Cliquez sur **Suivant**.
14. Obligatoire uniquement lors de la création d'images FIDO : Sur la page **FIDO device onboarding**, effectuez les étapes suivantes :
 - a. Dans le champ **Manufacturing server URL**, entrez les informations suivantes :
 - i. Dans le champ **DIUN public key insecure**, entrez la clé publique non sécurisée.
 - ii. Dans le champ **DIUN public key hash**, entrez le hachage de la clé publique.
 - iii. Dans le champ **DIUN public key root certs**, entrez les certificats racine de la clé publique. Cliquez sur **Suivant**.
15. Sur la page **OpenSCAP**, effectuez les étapes suivantes :
 - a. Dans le champ **Datastream**, entrez les instructions de remédiation **datastream** que vous souhaitez ajouter à votre image système.
 - b. Dans le champ **Profile ID**, entrez le profil de sécurité **profile_id** que vous souhaitez ajouter à votre image système. Cliquez sur **Suivant**.
16. Obligatoire uniquement lors de la création d'images Ignition : Sur la page **Ignition**, effectuez les étapes suivantes :

- a. Dans le champ **Firstboot URL**, saisissez le nom du paquet que vous souhaitez ajouter à votre image système.
- b. Dans le champ **Embedded Data**, faites glisser ou téléchargez votre fichier. Cliquez sur **Suivant**.

17. Sur la page **Review**, passez en revue les détails du plan. Cliquez sur **Créer**.

La vue du constructeur d'images s'ouvre et répertorie les plans existants.

5.3. IMPORTATION D'UN BLUEPRINT DANS L'INTERFACE DE LA CONSOLE WEB DU CONSTRUCTEUR D'IMAGES

Vous pouvez importer et utiliser un plan déjà existant. Le système résout automatiquement toutes les dépendances.

Conditions préalables

- Vous avez ouvert l'application de construction d'images à partir de la console web dans un navigateur.
- Vous avez un plan que vous souhaitez importer pour l'utiliser dans l'interface de la console web de construction d'images.

Procédure

1. Dans le tableau de bord du constructeur d'images, cliquez sur **Importer un projet**. L'assistant **Import blueprint** s'ouvre.
2. Dans le champ **Upload**, faites glisser ou téléchargez un plan existant. Ce plan peut être au format **TOML** ou **JSON**.
3. Cliquez sur **Importer**. Le tableau de bord répertorie le plan que vous avez importé. Lorsque vous cliquez sur le plan que vous avez importé, vous avez accès à un tableau de bord contenant toutes les personnalisations du plan que vous avez importé.

Vérification

- Pour vérifier les paquets qui ont été sélectionnés pour le plan importé, naviguez jusqu'à l'onglet **Packages**.
 - Pour dresser la liste de toutes les dépendances du paquet, cliquez sur **Toutes**. La liste est consultable et peut être ordonnée.
- Facultatif : Pour modifier toute personnalisation :
 - Dans le tableau de bord **Customizations**, cliquez sur la personnalisation que vous souhaitez modifier. Si vous le souhaitez, vous pouvez cliquer sur **Modifier le plan** pour accéder à toutes les options de personnalisation disponibles.

Ressources supplémentaires

- [Création d'une image système à l'aide du constructeur d'images dans l'interface de la console web](#).

5.4. EXPORTATION D'UN BLUEPRINT À PARTIR DE L'INTERFACE DE LA CONSOLE WEB DU CONSTRUCTEUR D'IMAGES

Vous pouvez exporter un modèle pour utiliser les personnalisations dans un autre système. Vous pouvez exporter le plan au format **TOML** ou **JSON**. Les deux formats fonctionnent dans l'interface de commande et dans l'interface API.

Conditions préalables

- Vous avez ouvert l'application de construction d'images à partir de la console web dans un navigateur.
- Vous avez un plan que vous souhaitez exporter.

Procédure

1. Dans le tableau de bord du constructeur d'images, sélectionnez le plan que vous souhaitez exporter.
2. Cliquez sur **Export blueprint**. L'assistant **Export blueprint** s'ouvre.
3. Pour enregistrer le site **Exported blueprint**, cliquez sur le bouton **Exporter**.
 - a. En option, cliquez sur le bouton **Copier** pour copier le plan.

Vérification

1. Ouvrez le plan exporté pour l'inspecter et le réviser.

5.5. CRÉATION D'UNE IMAGE SYSTÈME À L'AIDE DU CONSTRUCTEUR D'IMAGES DANS L'INTERFACE DE LA CONSOLE WEB

Vous pouvez créer une image système RHEL personnalisée à partir d'un modèle en suivant les étapes suivantes.

Conditions préalables

- Vous avez ouvert l'application de construction d'images à partir de la console web dans un navigateur.
- Vous avez créé un schéma directeur.

Procédure

1. Accéder au tableau de bord du constructeur d'images.
2. Sur la table des plans, trouvez le plan que vous voulez construire.
3. Dans la partie droite du plan choisi, cliquez sur **Créer une image**. L'assistant de la boîte de dialogue **Create image** s'ouvre.
4. Naviguez jusqu'à l'onglet **Images** et cliquez sur **Create Image**. L'assistant **Create image** s'ouvre.

5. Sur la page **Image output**, effectuez les étapes suivantes :
 - a. Dans la liste **Select a blueprint**, sélectionnez le type d'image souhaité.
 - b. Dans la liste **Image output type**, sélectionnez le type de sortie d'image que vous souhaitez. Selon le type d'image sélectionné, vous devez ajouter des détails supplémentaires.
6. Cliquez sur **Suivant**.
7. Sur la page **Review**, vérifiez les détails de la création de l'image et cliquez sur **Créer une image**. La construction de l'image démarre et dure jusqu'à 20 minutes.

Vérification

Une fois la construction de l'image terminée, vous pouvez :

- Télécharger l'image.
- Téléchargez les journaux de l'image pour inspecter les éléments et vérifier s'il y a un problème.

CHAPITRE 6. CRÉATION D'UNE IMAGE D'INSTALLATION ISO DE DÉMARRAGE AVEC LE CONSTRUCTEUR D'IMAGES

Vous pouvez utiliser Image builder pour créer des images ISO Installer amorçables. Ces images se composent d'un fichier **.tar** qui possède un système de fichiers racine. Vous pouvez utiliser l'image ISO amorçable pour installer le système de fichiers sur un serveur bare metal.

Image builder construit un manifeste qui crée une image ISO de démarrage contenant un système de fichiers racine. Pour créer l'image ISO, sélectionnez le type d'image **image-installer**. Image builder crée un fichier **.tar** avec le contenu suivant :

- un programme d'installation ISO standard d'Anaconda
- un fichier tar du système RHEL intégré
- un fichier Kickstart par défaut qui installe le commit avec des exigences minimales par défaut

L'image ISO d'installation créée comprend une image système préconfigurée que vous pouvez installer directement sur un serveur bare metal.

6.1. CRÉATION D'UNE IMAGE D'INSTALLATION ISO DE DÉMARRAGE À L'AIDE DU CONSTRUCTEUR D'IMAGES DANS L'INTERFACE DE LIGNE DE COMMANDE

Vous pouvez créer une image d'installation ISO de démarrage personnalisée à l'aide de l'interface de ligne de commande Image Builder.

Conditions préalables

- Vous avez créé un modèle pour l'image avec un utilisateur inclus et l'avez transféré dans le générateur d'images. Voir [Personnalisation de l'image pour les utilisateurs](#) .

Procédure

1. Créer l'image ISO :

```
# composer-cli compose start BLUEPRINT-NAME image-installer
```

- *BLUEPRINT-NAME* avec le nom du plan que vous avez créé
- *image-installer* est le type d'image
Le processus de composition démarre en arrière-plan et l'UUID de la composition est affiché.

2. Attendez que la composition soit terminée. Cela peut prendre plusieurs minutes.
Pour vérifier l'état de la composition :

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

Une composition terminée affiche une valeur d'état de **FINISHED**. Identifiez la composition dans la liste par son UUID.

3. Une fois la composition terminée, téléchargez le fichier image créé :

```
# composer-cli composer image UUID
```

Remplacez *UUID* par la valeur UUID obtenue dans les étapes précédentes.

En conséquence, le constructeur d'images construit un fichier **.iso** qui contient un fichier **.tar**. Le fichier **.tar** est l'image qui sera installée pour le système d'exploitation. Le fichier **.iso** est configuré pour démarrer Anaconda et installer le fichier tar pour configurer le système.

Vérification

1. Naviguez jusqu'au dossier dans lequel vous avez téléchargé le fichier image.
2. Localisez l'image **.iso** que vous avez téléchargée.
3. Monter l'ISO.

```
$ mount -o ro path_to_ISO /mnt
```

Vous trouverez le fichier **.tar** dans le répertoire **/mnt/liveimg.tar.gz**.

4. Extraire le contenu du fichier **.tar**.

```
$ tar xzf /mnt/liveimg.tar.gz
```

Vous pouvez utiliser le fichier image ISO créé sur un disque dur ou pour démarrer une machine virtuelle, par exemple dans le cadre d'une installation HTTP Boot ou USB.

Ressources supplémentaires

- [Création d'images système avec l'interface de ligne de commande image builder](#)
- [Création d'un support d'installation amorçable pour RHEL](#)

6.2. CRÉATION D'UNE IMAGE D'INSTALLATION ISO DE DÉMARRAGE À L'AIDE DU GÉNÉRATEUR D'IMAGES DANS L'INTERFACE GRAPHIQUE

Vous pouvez créer une image d'installation ISO de démarrage personnalisée à l'aide de l'interface graphique du constructeur d'images.

Conditions préalables

- Vous avez créé un modèle pour votre image. Voir [Création d'un modèle de construction d'image dans l'interface de la console web](#).

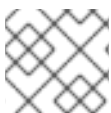
Procédure

1. Ouvrez l'interface du constructeur d'images de la console web RHEL dans un navigateur. Voir [Création d'un modèle de constructeur d'images à l'aide de l'interface de ligne de commande](#).
2. Localisez le plan que vous souhaitez utiliser pour construire votre image. Pour ce faire, saisissez le nom du plan ou une partie de celui-ci dans la zone de recherche située en haut à gauche, puis cliquez sur **Entrée**.
3. Sur le côté droit du plan, cliquez sur le bouton **Créer une image**.

L'assistant de dialogue **Create image** s'ouvre.

4. Dans l'assistant de dialogue **Create image**:
 - a. Dans la liste **Image Type**, sélectionnez **"RHEL Installer (.iso)"**.
 - b. Cliquez sur **Créer**.

Image builder ajoute la composition d'une image ISO RHEL à la file d'attente.



NOTE

Le processus de création de l'image prend quelques minutes.

Une fois le processus terminé, vous pouvez voir l'état de la construction de l'image. Le constructeur d'images crée l'image ISO.

Vérification

Une fois l'image créée, vous pouvez la télécharger.

1. Cliquez sur **Download** pour enregistrer l'image **"RHEL Installer (.iso)"** sur votre système.
2. Naviguez jusqu'au dossier dans lequel vous avez téléchargé l'image **"RHEL Installer (.iso)"**.
3. Localisez l'image .tar que vous avez téléchargée.
4. Extraire le contenu de l'image **"RHEL Installer (.iso)"**.

```
$ tar -xf content.tar
```

Vous pouvez utiliser le fichier image ISO résultant sur un disque dur ou pour démarrer une machine virtuelle, par exemple, dans le cadre d'un démarrage HTTP ou d'une installation USB.

Ressources supplémentaires

- [Création d'un plan de construction d'image dans l'interface de la console web](#) .
- [Création d'images système avec l'interface de ligne de commande image builder](#) .
- [Création d'un support d'installation amorçable pour RHEL](#) .

6.3. INSTALLATION D'UNE IMAGE ISO D'UN CONSTRUCTEUR D'IMAGES SUR UN SYSTÈME BARE METAL

Installez l'image ISO amorçable que vous avez créée à l'aide de l'outil de création d'images sur un système métallique nu.

Conditions préalables

- Vous avez créé l'image ISO amorçable à l'aide du générateur d'images. Voir [Création d'une image d'installation ISO de démarrage à l'aide du générateur d'images dans l'interface de ligne de commande](#).
- Vous avez téléchargé l'image ISO amorçable.

- Vous avez installé l'outil **dd**.
- Vous disposez d'une clé USB d'une capacité suffisante pour l'image ISO. La taille requise varie en fonction des paquets que vous avez sélectionnés dans votre plan, mais la taille minimale recommandée est de 8 Go.

Procédure

1. Écrire l'image ISO amorçable directement sur la clé USB à l'aide de l'outil **dd**. Par exemple :

```
dd if=installer.iso of=/dev/sdX
```

Où **installer.iso** est le nom du fichier image ISO et **/dev/sdX** est le chemin d'accès de votre clé USB.

2. Insérez la clé USB dans un port USB de l'ordinateur que vous souhaitez démarrer.
3. Démarrez l'image ISO à partir de la clé USB.
Lorsque l'environnement d'installation démarre, il se peut que vous deviez terminer l'installation manuellement, de la même manière que pour l'installation par défaut de Red Hat Enterprise Linux.

Ressources supplémentaires

- [Démarrage de l'installation](#)
- [Personnalisation de l'installation](#)
- [Création d'un périphérique USB amorçable sous Linux](#)

CHAPITRE 7. CRÉATION D'IMAGES PRÉ-DURCIES AVEC L'INTÉGRATION DE L'IMAGE BUILDER OPENSAP

Image builder on-premises prend en charge l'intégration d'OpenSCAP pour produire des images RHEL pré-durcies. Avec Image builder on-premises intégré à OpenSCAP, vous pouvez produire des images RHEL pré-durcies. Vous pouvez configurer un modèle, choisir parmi un ensemble de profils de sécurité prédéfinis, ajouter un ensemble de paquets ou de fichiers complémentaires et créer une image RHEL personnalisée prête à être déployée sur la plate-forme de votre choix, mieux adaptée à votre environnement.

Red Hat fournit des versions régulièrement mises à jour des profils de renforcement de la sécurité que vous pouvez choisir lorsque vous construisez vos systèmes afin de répondre à vos directives de déploiement actuelles.

7.1. DIFFÉRENCES ENTRE LES IMAGES KICKSTART ET LES IMAGES PRÉ-DURCIES

Pour la création traditionnelle d'une image à l'aide d'un fichier Kickstart, vous devez choisir les paquets à installer et vous assurer que le système n'est pas affecté par une vulnérabilité. L'intégration du constructeur d'images OpenSCAP vous permet de créer des images renforcées sur le plan de la sécurité. Au cours du processus de construction d'une image, un OSBuild **oscap remediation stage** exécute l'outil **OpenSCAP** dans le chroot, sur l'arborescence du système de fichiers. L'outil **OpenSCAP** exécute l'évaluation standard pour le profil que vous avez choisi et applique les remédiations à l'image. Ainsi, vous pouvez construire une image plus complètement durcie, si vous la comparez à l'exécution de la remédiation sur un système vivant.

7.2. LA PERSONNALISATION DU MODÈLE OPENSAP

Grâce à la prise en charge de la personnalisation des plans par OpenSCAP, vous pouvez créer des plans et les utiliser pour créer vos propres images pré-durcies. Pour créer une image pré-durcie, vous pouvez personnaliser les points de montage et configurer la disposition du système de fichiers en fonction du profil de sécurité sélectionné. Pendant la construction de l'image, OpenSCAP applique une remédiation au premier démarrage.

Après avoir sélectionné le profil OpenSCAP, la personnalisation du plan directeur OpenSCAP configure l'image pour déclencher la remédiation pendant la construction de l'image avec le profil sélectionné.

Pour utiliser la personnalisation du plan OpenSCAP dans vos plans d'image, entrez les informations suivantes :

- Le chemin du flux de données vers les instructions de remédiation de **datastream**. Vous pouvez le trouver dans le répertoire **/usr/share/xml/scap/ssg/content/**.
- Le **profile_id** du profil de sécurité requis. Le champ **profile_id** accepte à la fois la forme longue et la forme courte, par exemple : **cis** ou **xccdf_org.ssgproject.content_profile_cis**. Pour plus de détails, voir les [profils du Guide de sécurité SCAP pris en charge par RHEL 9](#) .

Le schéma suivant est un exemple de personnalisation d'OpenSCAP :

```
[customizations]
[customizations.openscap]
datastream = "/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml"
profile_id = "xccdf_org.ssgproject.content_profile_cis"
```

Le type de fichier SCAP le plus courant est un flux de données source SCAP. Pour afficher les détails du flux de données source SCAP à partir du paquetage **scap-security-guide**, entrez la commande :

```
$ oscap info /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

L'outil **oscap** s'exécute sur l'arborescence de l'image pour effectuer une analyse hors ligne d'un système de fichiers monté sur un chemin arbitraire. Vous pouvez l'utiliser pour l'analyse d'objets personnalisés qui ne sont pas pris en charge par **oscap-docker** ou **oscap-vm**, tels que les conteneurs dans des formats autres que Docker. **oscap-chroot** imite l'utilisation et les options de l'outil **oscap**.

Image builder génère les configurations nécessaires pour l'étape **osbuild** sur la base des personnalisations de votre plan. En outre, le constructeur d'images ajoute deux paquets à l'image :

- **openscap-scanner** - l'outil **OpenSCAP**.
- **scap-security-guide** - qui contient les instructions de remédiation.



NOTE

L'étape de remédiation utilise le paquetage **scap-security-guide** pour le flux de données car ce paquetage est installé par défaut sur l'image. Si vous souhaitez utiliser un flux de données différent, ajoutez le paquetage nécessaire au blueprint et indiquez le chemin d'accès au flux de données dans la configuration **oscap**.

Ressources supplémentaires

- [Profils du guide de sécurité SCAP pris en charge dans RHEL 9](#) .

7.3. CRÉATION D'UNE IMAGE PRÉ-DURCIE AVEC IMAGE BUILDER

Grâce à l'intégration d'OpenSCAP et de l'outil de création d'images, vous pouvez créer des images pré-durcies.

Procédure

1. Créer un schéma directeur au format TOML, avec le contenu suivant :

```
name = "blueprint_name"
description = "blueprint_description"
version = "0.0.1"
modules = []
groups = []
distro = ""

[customizations]
[[customizations.user]]
name = "scap-security-guide"
description = "Admin account"
password = secure_password_hash
key = ssh-key
home = "/home/scap-security-guide"
group = ["wheel"]
```

```
[[customizations.filesystem]]
mountpoint = "/tmp"
size = 13107200
[customizations.openscap]
datastream = "/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml "
profile_id = "cis"
```

2. Lance la construction d'une image OpenSCAP :

```
# composer-cli compose start blueprint_name qcow2
```

Où *blueprint_name* est le nom du plan.

Une fois que la construction de l'image est prête, vous pouvez utiliser votre image pré-durcie pour vos déploiements. Voir [Création d'une machine virtuelle](#).

Vérification

Après avoir déployé votre image pré-durcie dans une VM, vous pouvez effectuer une analyse de conformité de la configuration pour vérifier que l'image est alignée sur le profil de sécurité sélectionné.



IMPORTANT

L'exécution d'une analyse de conformité de la configuration ne garantit pas la conformité du système. Pour plus d'informations, voir [Analyse de la conformité](#) de la configuration.

1. Connectez-vous à l'image en utilisant SSH.
2. Lancez le scanner **oscap**.


```
# scap-workbench
```
3. Sélectionnez la version du système que vous souhaitez analyser. Cliquez sur **Charger le contenu**.
4. Sélectionnez le profil à analyser et cliquez sur **Analyser**. OpenSCAP vérifie toutes les exigences du système.
5. Une fois l'analyse terminée, cliquez sur **Afficher le rapport**.
Les résultats montrent que le système est sûr.

Ressources supplémentaires

- [OpenSCAP](#).
- [Profils de sécurité disponibles pour OpenSCAP](#).

CHAPITRE 8. PRÉPARATION ET DÉPLOIEMENT D'IMAGES D'INVITÉS KVM À L'AIDE DU CONSTRUCTEUR D'IMAGES

Pour créer une image spécialement conçue pour être utilisée sur Red Hat Virtualization, vous pouvez utiliser le constructeur d'images pour composer une image d'invité **KVM**. Notez cependant que les images d'invités KVM ne sont prises en charge que par **rhel-guest-image** dans Red Hat Virtualization.

L'utilisation d'une image d'invité KVM personnalisée implique les étapes de haut niveau suivantes :

1. Création d'une image d'invité KVM **.qcow2** image en utilisant le constructeur d'image.
2. Création d'une machine virtuelle à partir de l'image de l'invité KVM.

8.1. CRÉATION D'IMAGES D'INVITÉS KVM PERSONNALISÉES À L'AIDE DU CONSTRUCTEUR D'IMAGES

Vous pouvez créer une image d'invité KVM **.qcow2** personnalisée à l'aide de l'outil de création d'images.

Conditions préalables

- Vous devez avoir accès au système en tant que root ou utilisateur du groupe wheel.
- Le paquet **cockpit-composer** est installé.
- Sur un système RHEL, vous avez ouvert le tableau de bord du constructeur d'images de la console web.
- Vous avez créé un modèle. Voir [Création d'un modèle de constructeur d'images dans l'interface de la console web](#).

Procédure

1. Cliquez sur le nom du plan que vous avez créé.
2. Sélectionnez l'onglet **Images**.
3. Cliquez sur **Create Image** pour créer votre image personnalisée. Une fenêtre pop-up s'ouvre.
4. Dans la liste du menu déroulant **Type**, sélectionnez **QEMU Image(.qcow2)**.
5. Définissez la taille que vous souhaitez donner à l'image lorsqu'elle est instanciée et cliquez sur **Create**.
6. Une petite fenêtre contextuelle en haut à droite de la fenêtre vous informe que la création d'image a été ajoutée à la file d'attente. Une fois le processus de création d'image terminé, vous pouvez consulter l'état de **Image build complete**.

Vérification

1. Cliquez sur l'icône du fil d'Ariane et sélectionnez l'option **Download**. Image builder télécharge le fichier de l'image de l'invité KVM **.qcow2** à votre emplacement de téléchargement par défaut.

Ressources supplémentaires

- [Création d'un plan de construction d'image dans l'interface de la console web](#) .

8.2. CRÉER UNE MACHINE VIRTUELLE À PARTIR D'UNE IMAGE D'INVITÉ KVM

Avec le constructeur d'images, vous pouvez créer une image **.qcow2** et utiliser une image d'invité KVM pour créer une machine virtuelle. Les images d'invités KVM créées à l'aide de l'outil de création d'images ont déjà installé et activé **cloud-init**.

Conditions préalables

- Vous avez créé une image **.qcow2** à l'aide de Image builder. Voir [Création d'un modèle de constructeur d'images dans l'interface de la console web](#).
- Le paquet **qemu-kvm** est installé sur votre système. Vous pouvez vérifier si le dossier **/dev/kvm** est disponible sur votre système.
- Les paquets **libvirt** et **virt-install** sont installés sur votre système.
- L'utilitaire **genisoimage** est installé sur votre système.

Procédure

1. Déplacez l'image KVM Guest que vous avez créée à l'aide de Image builder dans le répertoire **/var/lib/libvirt/images**.
2. Créez un répertoire, par exemple **cloudinitiso**, et naviguez jusqu'à ce répertoire nouvellement créé :

```
$ mkdir cloudinitiso
$ cd cloudinitiso
```

3. Créez un fichier nommé **meta-data**. Ajoutez les informations suivantes à ce fichier :

```
instance-id: citest
local-hostname: vmname
```

4. Créez un fichier nommé **user-data**. Ajoutez les informations suivantes au fichier :

```
#cloud-config
user: admin
password: password
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...fhHQ== your.email@example.com
```

Où **ssh_authorized_keys** est votre clé publique SSH. Vous trouverez votre clé publique SSH à l'adresse **~/.ssh/id_rsa.pub**.

5. Utilisez la commande **genisoimage** pour créer une image ISO qui inclut les fichiers **user-data** et **meta-data**.

```
# genisoimage -output cloud-init.iso -volid cidata -joliet -rock user-data meta-data
```

```

l: -input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 331
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 0
183 extents written (0 MB)

```

6. Créez une nouvelle VM à partir de l'image KVM Guest à l'aide de la commande **virt-install**. Incluez l'image ISO que vous avez créée à l'étape 4 en tant que pièce jointe à l'image de la VM.

```

# virt-install \
  --memory 4096 \
  --vcpus 4 \
  --name myvm \
  --disk rhel-9-x86_64-kvm.qcow2,device=disk,bus=virtio,format=qcow2 \
  --disk cloud-init.iso,device=cdrom \
  --os-variant rhel 9 \
  --virt-type kvm \
  --graphics none \
  --import

```

Où ?

- --graphics none - signifie qu'il s'agit d'une VM RHEL 9 sans tête.
- --vcpus 4 - signifie qu'il utilise 4 CPU virtuels.
- --Mémoire 4096 - signifie qu'il utilise 4096 Mo de RAM.

7. L'installation de la VM démarre :

```

Starting install...
Connected to domain mytestcvm
...
[ OK ] Started Execute cloud user/final scripts.
[ OK ] Reached target Cloud-init target.

Red Hat Enterprise Linux 9 (Ootpa)
Kernel 4.18.0-221.el8.x86_64 on an x86_64

```

Vérification

Une fois le démarrage terminé, la VM affiche une interface de connexion textuelle. Pour se connecter à la VM :

1. Saisissez **admin** comme nom d'utilisateur et appuyez sur **Entrée**.
2. Saisissez **password** comme mot de passe et appuyez sur **Entrée**.
Une fois l'authentification terminée, vous pouvez accéder à la VM à l'aide de la CLI.

Ressources supplémentaires

- [Permettre la virtualisation.](#)

- Configurer et gérer cloud-init pour RHEL 9 .
- cloud-init des répertoires et des fichiers importants .

CHAPITRE 9. POUSSER UN CONTENEUR VERS UN REGISTRE ET L'INTÉGRER DANS UNE IMAGE

Grâce à la prise en charge de la personnalisation des conteneurs dans les plans, vous pouvez créer un conteneur et l'intégrer directement dans l'image que vous créez.

9.1. PERSONNALISATION DU BLUEPRINT POUR INTÉGRER UN CONTENEUR DANS UNE IMAGE

Pour intégrer un conteneur à partir du registre registry.access.redhat.com, vous devez ajouter une personnalisation de conteneur à votre plan directeur. Par exemple :

```
[[containers]]
source = "registry.access.redhat.com/ubi9/ubi:latest"
name = "local-name"
tls-verify = true
```

- **source** - Champ obligatoire. Il s'agit d'une référence à l'image du conteneur dans un registre. Cet exemple utilise le registre **registry.access.redhat.com**. Vous pouvez spécifier une version de balise. La version de balise par défaut est **latest**.
- **name** - le nom du conteneur dans le registre local.
- **tls-verify** - Champ booléen facultatif. Le champ booléen **tls-verify** contrôle la sécurité de la couche transport. La valeur par défaut est **true**.
Le constructeur d'image extrait le conteneur pendant la construction de l'image et le stocke dans l'image. L'emplacement de stockage local par défaut du conteneur dépend du type d'image, de sorte que tous les supports **container-tools** tels que Podman sont en mesure de travailler avec lui. Les conteneurs intégrés ne sont pas démarrés. Pour accéder aux ressources protégées du conteneur, vous pouvez utiliser un fichier **containers-auth.json**.

9.2. LES INFORMATIONS D'IDENTIFICATION DU REGISTRE DES CONTENEURS

Le service **osbuild-worker** est responsable de la communication avec le registre des conteneurs. Pour l'activer, vous pouvez configurer le fichier de configuration **/etc/osbuild-worker/osbuild-worker.toml**.



NOTE

Après avoir défini le fichier de configuration **/etc/osbuild-worker/osbuild-worker.toml**, vous devez redémarrer le service **osbuild-worker**, car il ne lit le fichier de configuration **/etc/osbuild-worker/osbuild-worker.toml** qu'une seule fois, lors du démarrage du service **osbuild-worker**.

Le fichier de configuration **/etc/osbuild-worker/osbuild-worker.toml** comporte une section sur les conteneurs avec une entrée **auth_field_path** qui est une chaîne renvoyant au chemin d'un fichier **containers-auth.json** à utiliser pour accéder aux ressources protégées. Les informations d'identification du registre des conteneurs ne sont utilisées que pour extraire une image de conteneur d'un registre, lors de l'intégration du conteneur dans l'image.

Voici un exemple :


```
[containers]
auth_file_path = "/etc/osbuild-worker/containers-auth.json"
```

Ressources supplémentaires

- [containers-auth.json](#) page de manuel.

9.3. POUSSER UN ARTEFACT DE CONTENEUR DIRECTEMENT VERS UN REGISTRE DE CONTENEURS

Vous pouvez pousser des artefacts de conteneur, tels que des images de conteneur RHEL for Edge, directement vers un registre de conteneur après l'avoir construit, à l'aide de l'interface CLI de construction d'image. Pour cela, vous devez configurer une adresse **upload provider** et, éventuellement, des informations d'identification, puis vous pouvez construire l'image de conteneur en passant le registre et le référentiel à **composer-cli** en tant qu'arguments. Une fois que l'image est prête, elle est disponible dans le registre de conteneurs que vous avez configuré.

Conditions préalables

- Accès au [registre quay.io](#). Cet exemple utilise le registre de conteneurs **quay.io** comme registre cible, mais vous pouvez utiliser le registre de conteneurs de votre choix.

Procédure

1. Créez un fichier **registry-config.toml** pour sélectionner le fournisseur de conteneurs.

```
provider = "container_provider"

[settings]
tls_verify = false
username = "admin"
password = "your_password"
```

2. Créez un plan au format **.toml**. Il s'agit d'un plan pour le conteneur dans lequel vous installez un paquet **nginx** dans le plan.

```
name = "simple-container"
description = "Simple RHEL container"
version = "0.0.1"

[[packages]]
name = "nginx"
version = ""
```

3. Pousser le plan d'action :

```
# composer-cli blueprints push blueprint.toml
```

4. Construire l'image du conteneur :

```
# composer-cli compose start simple-container container "quay.io:8080/osbuild/repository"
registry-config.toml
```

- `simple-container` - est le nom du modèle.
- `container` - est le type d'image.
- `\N "quay.io:8080/osbuild/repository\N"` est le registre cible, est l'organisation et est l'emplacement pour pousser le conteneur lorsque la construction est terminée - **quay.io** est le registre cible, **osbuild** est l'organisation et **repository** est l'emplacement où pousser le conteneur à la fin de sa construction. En option, vous pouvez définir une valeur pour **tag**. Si vous ne définissez pas de valeur pour **:tag**, la balise **:latest** est utilisée par défaut.



NOTE

La construction de l'image du conteneur prend du temps en raison de la suppression des paquets personnalisés.

5. Une fois la construction de l'image terminée, le conteneur que vous avez créé est disponible sur quay.io.
6. Accédez à quay.io et cliquez sur **Repository Tags**.

You can see details about the container you created, such as:

- last modified
- image size
- the `manifest ID`, that you can copy to the clipboard.

7. Copiez la valeur **manifest ID** pour construire l'image dans laquelle vous souhaitez intégrer un conteneur.

Ressources supplémentaires

- [Quay.io - Travailler avec des balises](#).

9.4. CONSTRUCTION D'UNE IMAGE ET INTÉGRATION DU CONTENEUR DANS L'IMAGE

Après avoir créé l'image du conteneur, vous pouvez construire votre image personnalisée et y intégrer l'image du conteneur. Pour cela, vous devez spécifier un **container customization** dans le blueprint, et le **container name** pour l'image finale. Pendant le processus de construction, l'image du conteneur est récupérée et placée dans le stockage local du conteneur Podman.

Conditions préalables

- Vous avez créé une image de conteneur et l'avez poussée dans votre instance locale de registre de conteneur **quay.io**.
- Vous avez accès à registry.access.redhat.com.
- Vous avez un conteneur **manifest ID**.
- Vous avez installé les paquets **qemu-kvm** et **qemu-img**. Pour l'installer, exécutez la commande :

```
# yum install qemu-kvm qemu-img
```

Procédure

1. Créez un modèle pour construire une image **qcow2**. Le modèle doit contenir la personnalisation.

```
name = "image"
description = "A qcow2 image with a container"
version = "0.0.1"
distro = "rhel-90"

[[packages]]
name = "podman"
version = "*"

[[containers]]
source = "registry.access.redhat.com/ubi9:8080/osbuild/container/container-
image@sha256:manifest-ID-from-Repository-tag: tag-version"
name = "source-name"
tls-verify = true
```

2. Pousser le plan d'action :

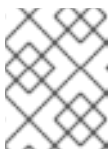
```
# composer-cli blueprints push blueprint-image.toml
```

3. Construire l'image du conteneur :

```
# composer-cli start compose image qcow2
```

Où ?

- *image* est le nom du plan.
- **qcow2** est le type d'image.



NOTE

La création de l'image prend du temps car elle vérifie le conteneur dans le registre **quay.io**.

Une fois que l'état de construction de l'image est "FINISHED", vous pouvez utiliser l'image **qcow2** que vous avez créée dans une VM.

Vérification

1. Transférez l'image **.qcow2** depuis le site **composer-cli** vers votre système de fichiers local :

```
# composer-cli compose image COMPOSE-UUID
```

2. Démarrez l'image **qcow2** dans une machine virtuelle. Voir [Création d'une machine virtuelle à partir d'une image d'invité KVM](#).
3. L'assistant **qemu** s'ouvre. Connectez-vous à l'image **qcow2**.

- a. Saisissez le nom d'utilisateur et le mot de passe. Il peut s'agir du nom d'utilisateur et du mot de passe que vous avez définis dans le projet **.qcow2** dans la section `\N` "customizations.user\`\N`", ou que vous avez créés au démarrage avec **cloud-init**.
4. Exécutez l'image du conteneur et ouvrez une invite shell à l'intérieur du conteneur :

```
# podman run -it registry.access.redhat.com/ubi9:8080/osbuild/repository /bin/bash/
```

Où ?

- **registry.access.redhat.com** est le registre cible, **osbuild** est l'organisation et **repository** est l'emplacement où pousser le conteneur à la fin de sa construction.

5. Vérifiez que les paquets que vous avez ajoutés au plan sont disponibles :

```
# type -a nginx
```

La sortie indique le chemin du paquet **nginx**.

Ressources supplémentaires

- [Red Hat Container Registry Authentication](#) .
- [Accès et configuration du registre Red Hat](#) .
- [Commandes de base de Podman](#) .
- [Exécution de Skopeo dans un conteneur](#) .

CHAPITRE 10. PRÉPARATION ET TÉLÉCHARGEMENT D'IMAGES DE NUAGES À L'AIDE DE L'OUTIL DE CRÉATION D'IMAGES

Image builder permet de créer des images système personnalisées prêtes à être utilisées sur diverses plates-formes en nuage. Pour utiliser votre image système RHEL personnalisée dans un nuage, 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 de nuage. Vous pouvez pousser des nuages d'images personnalisées via l'application **image builder** dans la console web RHEL, disponible pour un sous-ensemble de fournisseurs de services que nous prenons en charge, tels que les nuages **AWS** et **Microsoft Azure**. Voir [Pushing images to AWS Cloud AMI](#) et [Pushing VHD images to Microsoft Azure cloud](#).

10.1. 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](#)

10.2. 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](#)

10.3. 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](#)

10.4. PRÉPARATION AU TÉLÉCHARGEMENT D'IMAGES VHD MICROSOFT AZURE

Vous pouvez utiliser le générateur d'images pour préparer une image VHD qui peut être téléchargée sur le nuage **Microsoft Azure**.

Conditions préalables

- Vous devez disposer d'un groupe de ressources et d'un compte de stockage Microsoft Azure utilisables.
- Vous avez installé python2 car l'outil **AZ CLI** dépend spécifiquement de python 2.7.

Procédure

1. Importer la clé du référentiel Microsoft :

```
# rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

2. Créer un dépôt local d'informations **azure-cli**:

```
# sh -c 'echo -e \N-[azure-cli]\N-[azure-cli]\N-[azure-cli]\N-[azure-
cli]\Nbaseurl=https://packages.microsoft.com/yumrepos/azure-
cli\Nenabled=1\Npgcheck=1\Npgkey=https://packages.microsoft.com/keys/microsoft.asc\N" >
/etc/yum.repos.d/azure-cli.repo'
```

3. Installez la CLI Microsoft Azure :

```
# dnfdownloader azure-cli
# rpm -ivh --nodeps azure-cli-2.0.64-1.el7.x86_64.rpm
```



NOTE

La version téléchargée du paquet Microsoft Azure CLI peut varier en fonction de la version actuellement disponible.

4. Exécutez l'interface de programmation Microsoft Azure :

```
$ az login
```

Le terminal affiche le message suivant : **Note, we have launched a browser for you to login. For old experience with device code, use "az login --use-device-code**. Ensuite, le terminal ouvre un navigateur avec un lien vers <https://microsoft.com/devicelogin> à partir duquel vous pouvez vous connecter.



NOTE

Si vous exécutez une session à distance (SSH), le lien <https://microsoft.com/devicelogin> ne s'ouvrira pas dans le navigateur. Dans ce cas, vous pouvez copier le lien dans un navigateur et vous connecter pour authentifier votre session à distance. Pour vous connecter, utilisez un navigateur web pour ouvrir la page <https://microsoft.com/devicelogin> et entrez le code de l'appareil pour vous authentifier.

5. Listez les clés du compte de stockage dans Microsoft Azure :

```
$ GROUP=resource-group-name
$ ACCOUNT=storage-account-name
$ az storage account keys list --resource-group $GROUP --account-name $ACCOUNT
```

Remplacez *resource-group-name* par le nom de votre groupe de ressources Microsoft Azure et *storage-account-name* par le nom de votre compte de stockage Microsoft Azure.



NOTE

Vous pouvez dresser la liste des ressources disponibles à l'aide de la commande suivante :

```
$ az resource list
```

- Notez la valeur **key1** dans la sortie de la commande précédente, et affectez-la à une variable d'environnement :

```
KEY1=value
```

- Créer un conteneur de stockage :

```
$ CONTAINER=storage-account-name
$ az storage container create --account-name $ACCOUNT \
--account-key $KEY1 --name $CONTAINER
```

Remplacez *storage-account-name* par le nom du compte de stockage.

Ressources supplémentaires

- [Microsoft Azure CLI](#).

10.5. TÉLÉCHARGEMENT D'IMAGES VHD DANS LE NUAGE MICROSOFT AZURE

Après avoir créé votre image VHD personnalisée, vous pouvez la télécharger sur le nuage Microsoft Azure.

Conditions préalables

- Votre système doit être configuré pour télécharger des images VHD Microsoft Azure. Voir [Préparation au téléchargement d'images Microsoft Azure VHD](#).
- Vous devez disposer d'une image VHD Microsoft Azure créée par le constructeur d'images.
 - Dans le CLI, utilisez le type de sortie **vhd**.
 - Dans l'interface graphique, utilisez le type d'image **Azure Disk Image (.vhd)**.



NOTE

Lors de la création d'une image **.vhd** à l'aide de l'interface de commande, le constructeur d'images écrit des fichiers temporaires dans le sous-répertoire **/var**. Pour éviter que la création de l'image **.vhd** n'échoue, augmentez la capacité du sous-répertoire **/var** pour qu'il dispose d'au moins 15 à 20 Go d'espace libre afin d'en garantir la disponibilité.

Procédure

1. Transférez l'image vers Microsoft Azure et créez une instance à partir de celle-ci :

```
$ VHD=25ccb8dd-3872-477f-9e3d-c2970cd4bbaf-disk.vhd
$ az storage blob upload --account-name $ACCOUNT --container-name $CONTAINER --file
$VHD --name $VHD --type page
...
```

2. Une fois le téléchargement vers le stockage Microsoft Azure Blob terminé, créez une image Microsoft Azure à partir de celui-ci :

```
$ az image create --resource-group $GROUP --name $VHD --os-type linux --location eastus
--source https://$ACCOUNT.blob.core.windows.net/$CONTAINER/$VHD
- Running ...
```

Vérification

1. Créez une instance à l'aide du portail Microsoft Azure ou d'une commande similaire à la suivante :

```
$ az vm create --resource-group $GROUP --location eastus --name $VHD --image $VHD --
admin-username azure-user --generate-ssh-keys
- Running ...
```

2. Utilisez votre clé privée via SSH pour accéder à l'instance résultante. Connectez-vous en tant que **azure-user**.

Ressources complémentaires

- [La composition d'une image pour le format .vhd échoue.](#)

10.6. POUSSER DES IMAGES VHD VERS LE NUAGE MICROSOFT AZURE

Vous pouvez créer des images **.vhd** à l'aide de l'outil de création d'images. Ensuite, vous pouvez pousser les images **.vhd** vers un stockage Blob du fournisseur de services Microsoft Azure Cloud.

Conditions préalables

- Vous avez un accès root au système.
- Vous avez accès à l'interface de construction d'images de la console web RHEL.
- Vous avez créé un modèle. Voir [Création d'un modèle de constructeur d'images dans l'interface de la console web](#).
- Vous avez créé un [compte de stockage Microsoft](#).
- Vous avez préparé un [stockage Blob](#) accessible en écriture.

Procédure

1. Dans le tableau de bord du constructeur d'images, sélectionnez le modèle que vous souhaitez utiliser.
2. Cliquez sur l'onglet **Images**.
3. Cliquez sur **Créer une image** pour créer votre image **.vhd** personnalisée. L'assistant **Create image** s'ouvre.
 - a. Sélectionnez **Microsoft Azure (.vhd)** dans la liste du menu déroulant **Type**.
 - b. Cochez la case **Upload to Azure** pour télécharger votre image dans le nuage Microsoft Azure.
 - c. Saisissez le site **Image Size** et cliquez sur **Next**.
4. Sur la page **Upload to Azure**, saisissez les informations suivantes :
 - a. Sur la page **Authentification**, entrez :
 - i. Votre nom **Storage account**. Vous le trouverez sur la page **Storage account**, dans le [portail Microsoft Azure](#).
 - ii. Votre **Storage access key**. Vous pouvez le trouver sur la page **Access Key Storage**.
 - iii. Cliquez sur **Suivant**.
 - b. Sur la page **Authentication**, entrez :
 - i. Le nom de l'image.
 - ii. Le **Storage container** est le conteneur blob dans lequel vous allez télécharger l'image. Vous le trouverez sous la section **Blob service**, dans le [portail Microsoft Azure](#).
 - iii. Cliquez sur **Suivant**.
5. Sur la page **Review**, cliquez sur **Créer**. Les processus de construction et de téléchargement de l'image démarrent.

Prochaines étapes

1. Pour accéder à l'image que vous avez transférée sur **Microsoft Azure Cloud**, accédez au [portail Microsoft Azure](#).
2. Dans la barre de recherche, tapez "compte de stockage" et cliquez sur **Storage accounts** dans la liste.
3. Dans la barre de recherche, tapez "Images" et sélectionnez la première entrée sous **Services**. Vous êtes redirigé vers le site **Image dashboard**.
4. Dans le panneau de navigation, cliquez sur **Containers**.
5. Trouvez le conteneur que vous avez créé. Dans le conteneur se trouve le fichier **your_image_name.vhd** que vous avez créé et poussé.

Vérification

Vérifiez que vous pouvez créer une image de VM et la lancer.

1. Dans la barre de recherche, tapez `compte images` et cliquez sur **Images** dans la liste.
2. Cliquez sur **Créer**.
3. Dans la liste déroulante, choisissez le groupe de ressources que vous avez utilisé précédemment.
4. Entrez un nom pour l'image.
5. Pour le site **OS type**, sélectionnez **Linux**.
6. Pour le site **VM generation**, sélectionnez **Gen 2**.
7. Sous **Storage Blob**, cliquez sur **Parcourir** et cliquez sur les comptes de stockage et les conteneurs jusqu'à ce que vous atteigniez votre fichier VHD.
8. Cliquez sur **Select** à la fin de la page.
9. Choisissez un type de compte, par exemple, **Standard SSD**.
10. Cliquez sur **Réviser créer**, puis sur **Créer**. Attendez quelques instants pour la création de l'image.

Pour lancer la VM, procédez comme suit :

1. Cliquez sur **Aller à la ressource**.
2. Cliquez sur **Créer une VM** dans la barre de menu de l'en-tête.
3. Entrez un nom pour votre machine virtuelle.
4. Complétez les sections **Size** et **Administrator account**.
5. Cliquez sur **Examiner la création**, puis sur **Créer**. Vous pouvez voir la progression du déploiement.
Une fois le déploiement terminé, cliquez sur le nom de la machine virtuelle pour récupérer l'adresse IP publique de l'instance afin de vous connecter à l'aide de SSH.
6. Ouvrez un terminal pour créer une connexion SSH afin de vous connecter à la VM.

Ressources supplémentaires

- [Documentation sur le stockage Microsoft Azure](#) .
- [Créer un compte Microsoft Azure Storage](#) .
- [Ouvrez un dossier sur le portail client de Red Hat](#) .
- [Aide à l'assistance](#) .
- [Contacter Red Hat](#) .

10.7. TÉLÉCHARGEMENT D'IMAGES VMDK ET CRÉATION D'UNE MACHINE VIRTUELLE RHEL DANS VSPHERE

Téléchargez une image `.vmdk` sur VMware vSphere à l'aide de l'outil CLI `govc import.vmdk`.



NOTE

Le téléchargement de l'image via l'interface utilisateur n'est pas pris en charge.

Conditions préalables

- Vous avez créé un modèle avec des personnalisations de nom d'utilisateur et de mot de passe.
- Vous avez créé une image **.vmdk** à l'aide de l'outil de création d'images et l'avez téléchargée sur votre système hôte.
- Vous avez installé l'outil CLI **govc import.vmdk**.
- Vous avez configuré le client de l'outil CLI **govc import.vmdk**.
 - vous devez définir les valeurs suivantes dans l'environnement :

```
GOVC_URL
GOVC_DATACENTER
GOVC_FOLDER
GOVC_DATASTORE
GOVC_RESOURCE_POOL
GOVC_NETWORK
```

Procédure

1. Naviguez jusqu'au répertoire où vous avez téléchargé votre image **.vmdk**.
2. Lancer l'image sur vSphere en suivant les étapes :
 - a. Importez l'image **.vmdk** dans vSphere :

```
govc import.vmdk ./composer-api.vmdk foldername
```

- b. Créez la VM dans VSphere sans la mettre sous tension :

```
govc vm.create \
-net.adapter=vmxnet3 \
-m=4096 -c=2 -g=rhel8_64Guest \
-firmware=efi -disk="foldername/composer-api.vmdk" \
-disk.controller=scsi -on=false \
vmname
```

- c. Mettez la VM sous tension :

```
govc vm.power -on vmname
```

- d. Récupérer l'adresse IP de la VM :

```
HOST=$(govc vm.ip vmname)
```

- e. Utilisez SSH pour vous connecter à la VM, en utilisant le nom d'utilisateur et le mot de passe que vous avez spécifiés dans votre plan :

```
$ ssh admin@HOST
```



NOTE

Si vous avez copié l'image **.vmdk** de votre hôte local vers la destination à l'aide de la commande **govc datastore.upload**, l'utilisation de l'image n'est pas prise en charge. Il n'y a pas d'option pour utiliser la commande **import.vmdk** dans l'interface graphique vSphere et, par conséquent, l'interface graphique vSphere ne prend pas en charge le téléchargement direct, ce qui signifie que l'image **.vmdk** n'est pas directement utilisable à partir de l'interface graphique vSphere.

10.8. TÉLÉCHARGEMENT D'IMAGES SUR GCP AVEC IMAGE BUILDER

Avec le constructeur d'images, vous pouvez créer une image **gce**, fournir des informations d'identification pour votre utilisateur ou votre compte de service GCP, puis télécharger l'image **gce** directement dans l'environnement GCP.

10.8.1. Téléchargement d'une image gce sur GCP à l'aide de la CLI

Suivez la procédure pour configurer un fichier de configuration avec des informations d'identification pour télécharger votre image **gce** sur GCP.

Conditions préalables

- Vous disposez d'un compte utilisateur ou d'un compte de service dont les identifiants Google vous permettent de télécharger des images vers GCP. Le compte associé aux informations d'identification doit avoir au moins les rôles IAM suivants attribués :
 - **roles/storage.admin** - pour créer et supprimer des objets de stockage
 - **roles/compute.storageAdmin** - pour importer une image de VM dans Compute Engine.
- Vous disposez d'un panier GCP existant.

Procédure

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

```
provider = "gcp"

[settings]
bucket = "GCP_BUCKET"
region = "GCP_STORAGE_REGION"
object = "OBJECT_KEY"
credentials = "GCP_CREDENTIALS"
```

Où ?

- **GCP_BUCKET** pointe vers un godet existant. Il est utilisé pour stocker l'objet de stockage intermédiaire de l'image en cours de téléchargement.

- **GCP_STORAGE_REGION** est à la fois une région de stockage Google normale et une région double ou multiple.
- **OBJECT_KEY** est le nom d'un objet de stockage intermédiaire. Il ne doit pas exister avant le téléchargement et il est supprimé lorsque le processus de téléchargement est terminé. Si le nom de l'objet ne se termine pas par **.tar.gz**, l'extension est automatiquement ajoutée au nom de l'objet.
- **GCP_CREDENTIALS** est un schéma encodé en Base64 du fichier JSON des informations d'identification téléchargé depuis le GCP. Les informations d'identification déterminent le projet vers lequel le GCP télécharge l'image.



NOTE

La spécification de **GCP_CREDENTIALS** dans **gcp-config.toml** est facultative si vous utilisez un mécanisme différent pour vous authentifier auprès de GCP. Pour plus de détails sur les différentes manières de s'authentifier avec GCP, voir [Authentification avec GCP](#).

2. Créez une composition avec un nom d'image supplémentaire et un profil de fournisseur de cloud :

```
$ sudo composer-cli compose start BLUEPRINT-NAME gce IMAGE_KEY gcp-config.toml
```

Remarque : la création et le téléchargement de l'image ainsi que l'enregistrement dans le nuage peuvent prendre jusqu'à dix minutes.

Vérification

- Vérifiez que l'état de l'image est FINI :

```
$ sudo composer-cli compose status
```

Ressources supplémentaires

- [Gestion des identités et des accès](#).
- [Créez des godets de stockage](#).

10.8.2. Authentification avec GCP

Vous pouvez utiliser différents types d'informations d'identification avec Image Builder pour vous authentifier auprès de GCP. Si la configuration d'image builder est définie pour s'authentifier auprès de GCP à l'aide de plusieurs jeux d'informations d'identification, elle utilise les informations d'identification dans l'ordre de préférence suivant :

1. Informations d'identification spécifiées avec la commande **composer-cli** dans le fichier de configuration.
2. Informations d'identification configurées dans la configuration du travailleur **osbuild-composer**.
3. Application Default Credentials de la bibliothèque **Google GCP SDK**, qui tente de trouver automatiquement un moyen de s'authentifier à l'aide des options suivantes :

- a. Si la variable d'environnement `GOOGLE_APPLICATION_CREDENTIALS` est définie, Application Default Credentials tente de charger et d'utiliser les informations d'identification du fichier indiqué par la variable.
- b. Application Default Credentials tente de s'authentifier à l'aide du compte de service attaché à la ressource qui exécute le code. Par exemple, Google Compute Engine VM.



NOTE

Vous devez utiliser les informations d'identification GCP pour déterminer le projet GCP dans lequel l'image doit être téléchargée. Par conséquent, à moins que vous ne souhaitiez télécharger toutes vos images vers le même projet GCP, vous devez toujours spécifier les informations d'identification dans le fichier de configuration **`gcp-config.toml`** à l'aide de la commande **`composer-cli`**.

10.8.2.1. Spécifier les informations d'identification avec la commande `composer-cli`

Vous pouvez spécifier les informations d'authentification GCP dans la configuration de la cible de téléchargement fournie **`gcp-config.toml`**. Utilisez un schéma codé **Base64** du fichier JSON des informations d'identification du compte Google pour gagner du temps.

Procédure

- Dans la configuration de la cible de téléchargement fournie **`gcp-config.toml`**, définissez les informations d'identification :

```
provider = "gcp"

[settings]
provider = "gcp"

[settings]
...
credentials = "GCP_CREDENTIALS"
```

- Pour obtenir le contenu codé du fichier d'informations d'identification du compte Google avec le chemin d'accès stocké dans la variable d'environnement **`GOOGLE_APPLICATION_CREDENTIALS`**, exécutez la commande suivante :

```
$ base64 -w 0 "${GOOGLE_APPLICATION_CREDENTIALS}"
```

10.8.2.2. Spécifier les informations d'identification dans la configuration du travailleur `osbuild-composer`

Vous pouvez configurer les informations d'authentification GCP de manière à ce qu'elles soient utilisées pour GCP de manière globale pour toutes les constructions d'images. Ainsi, si vous souhaitez importer des images dans le même projet GCP, vous pouvez utiliser les mêmes informations d'identification pour tous les téléchargements d'images vers GCP.

Procédure

- Dans la configuration du travailleur **`/etc/osbuild-worker/osbuild-worker.toml`**, définissez la valeur d'identification suivante :

```
[gcp]  
credentials = "PATH_TO_GCP_ACCOUNT_CREDENTIALS"
```

10.9. POUSSER DES IMAGES VMDK VERS VSPHERE À L'AIDE DE L'OUTIL DE CRÉATION D'IMAGES GUI

Vous pouvez créer des images VMware à l'aide de l'outil de création d'images de l'interface utilisateur graphique et les envoyer directement à votre instance vSphere, afin d'éviter d'avoir à télécharger le fichier image et à l'envoyer manuellement. Pour créer des images **.vmdk** à l'aide de l'outil de création d'images directement vers le fournisseur de services d'instances vSphere, procédez comme suit :

Conditions préalables

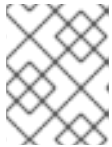
- Vous avez accès au système par l'intermédiaire du groupe d'utilisateurs **root** ou **wheel**.
- Vous avez ouvert l'interface de [construction d'images](#) de la console web RHEL dans un navigateur.
- Vous avez créé un modèle. Voir [Création d'un modèle de constructeur d'images dans l'interface de la console web](#).
- Vous avez un [compte vSphere](#).

Procédure

1. Pour le plan que vous avez créé, cliquez sur l'onglet **Images**.
2. Cliquez sur **Créer une image** pour créer votre image personnalisée. La fenêtre Type d'image s'ouvre.
3. Dans la fenêtre **Image type**:
 - a. Dans le menu déroulant, sélectionnez le type : VMware vSphere (.vmdk).
 - b. Cochez la case **Upload to VMware** pour télécharger votre image sur vSphere.
 - c. Facultatif : Définissez la taille de l'image que vous souhaitez instancier. La taille minimale par défaut est de 2 Go.
 - d. Cliquez sur **Suivant**.
4. Dans la fenêtre **Upload to VMware**, sous **Authentication**, entrez les détails suivants :
 - a. **Username** nom d'utilisateur du compte vSphere.
 - b. **Password** mot de passe du compte vSphere.
5. Dans la fenêtre **Upload to VMware**, sous **Destination**, entrez les détails suivants :
 - a. **Image name**: un nom pour l'image à télécharger.
 - b. **Host**: L'URL de votre VMware vSphere où l'image sera téléchargée.
 - c. **Cluster**: Le nom du cluster où l'image sera téléchargée.
 - d. **Data center**: Le nom du centre de données où l'image sera téléchargée.

- e. **Data store**: Le nom du magasin de données où l'image sera téléchargée.
 - f. Cliquez sur **Next**.
6. Dans la fenêtre **Review**, vérifiez les détails de la création de l'image et cliquez sur **Terminer**. Vous pouvez cliquer sur **Retour** pour modifier tout détail incorrect.

Image builder ajoute la composition d'une image RHEL vSphere à la file d'attente, crée et télécharge l'image vers le cluster sur l'instance vSphere que vous avez spécifiée.



NOTE

Les processus de création et de téléchargement de l'image prennent quelques minutes.

Une fois le processus terminé, vous pouvez consulter l'état de **Image build complete**.

Vérification

Une fois le téléchargement de l'image terminé, vous pouvez créer une machine virtuelle (VM) à partir de l'image que vous avez téléchargée et vous y connecter. Pour ce faire, procédez comme suit

1. Accédez au client VMware vSphere.
2. Recherchez l'image dans le cluster sur l'instance vSphere que vous avez spécifiée.
3. Vous pouvez créer une nouvelle VM à partir de l'image que vous avez téléchargée :
 - a. Sélectionnez l'image que vous avez téléchargée.
 - b. Cliquez avec le bouton droit de la souris sur l'image sélectionnée.
 - c. Cliquez sur **New Virtual Machine**.
Une fenêtre **New Virtual Machine** s'ouvre.

Dans la fenêtre **New Virtual Machine**, fournissez les informations suivantes :

- i. Sélectionnez **New Virtual Machine**.
- ii. Sélectionnez un nom et un dossier pour votre VM.
- iii. Sélectionner une ressource informatique : choisir une ressource informatique de destination pour cette opération.
- iv. Sélectionnez le stockage : Par exemple, sélectionnez NFS-Node1
- v. Sélectionnez la compatibilité : L'image doit être réservée au BIOS.
- vi. Sélectionnez un système d'exploitation invité : Par exemple, sélectionnez *Linux* et *Red Hat Fedora (64-bit)*.
- vii. **Customize hardware**: Lors de la création d'une VM, sur le bouton **Device Configuration** en haut à droite, supprimez le Nouveau disque dur par défaut et utilisez le menu déroulant pour sélectionner une image de disque dur existante :
- viii. Prêt à compléter : Passez en revue les détails et cliquez sur **Finish** pour créer l'image.

- d. Naviguez jusqu'à l'onglet **VMs**.
 - i. Dans la liste, sélectionnez la VM que vous avez créée.
 - ii. Cliquez sur le bouton **Start** dans le tableau de bord. Une nouvelle fenêtre apparaît, montrant le chargement de l'image VM.
 - iii. Connectez-vous avec les informations d'identification que vous avez créées pour le projet.
 - iv. Vous pouvez vérifier si les paquets que vous avez ajoutés au plan sont installés. Par exemple, vous pouvez vérifier si les paquets que vous avez ajoutés au plan directeur sont installés :

```
$ rpm -qa | grep firefox
```

Ressources supplémentaires

- [Installation du client vSphere.](#)

10.10. TÉLÉCHARGEMENT D'IMAGES QCOW2 VERS OPENSTACK

Avec l'outil de création d'images, vous pouvez créer des images **.qcow2** personnalisées qui conviennent aux déploiements de nuages OpenStack, et y démarrer des instances.



AVERTISSEMENT

Ne confondez pas le format de sortie du type d'image générique **QCOW2** que vous créez en utilisant le constructeur d'images avec le type d'image OpenStack, qui est également dans le format QCOW2, mais qui contient des modifications supplémentaires spécifiques à OpenStack.

Conditions préalables

- Vous avez créé un schéma directeur.
- créé une image **QCOW2** à l'aide de l'outil de création d'images. Voir

Procédure

1. Commencez la composition d'une image **QCOW2**.

```
# composer-cli compose start blueprint_name openstack
```

2. Vérifier l'état du bâtiment.

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

Une fois la construction de l'image terminée, vous pouvez télécharger l'image.

3. Télécharger l'image **QCOW2**:

```
# composer-cli composer image UUID
```

4. Accédez au tableau de bord OpenStack et cliquez sur **Créer une image**.5. Dans le menu de gauche, sélectionnez l'onglet **Admin**.

- a. Sur le site **System Panel**, cliquez sur **Image**.
L'assistant **Create An Image** s'ouvre.

6. Dans l'assistant **Create An Image**:

- a. Saisir un nom pour l'image
- b. Cliquez sur **Browse** pour télécharger l'image **QCOW2**.
- c. Dans la liste déroulante **Format**, sélectionnez le site **QCOW2 - QEMU Emulator**.
- d. Cliquez sur **Créer une image**.

Create An Image

Name: *
96268ffb-2c71-4e97-a855-7ac25e983a6e-disk.qcow2

Description:
Specify an image to upload to the Image Service.
Currently only images available via an HTTP URL are supported. The image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)
Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.

Image Source:
Image File

Image File
Browse... 96268ffb-2c71-4e97-a85...c25e98

Format: *
QCOW2 - QEMU Emulator

Architecture:
x86_64

Minimum Disk (GB):
5

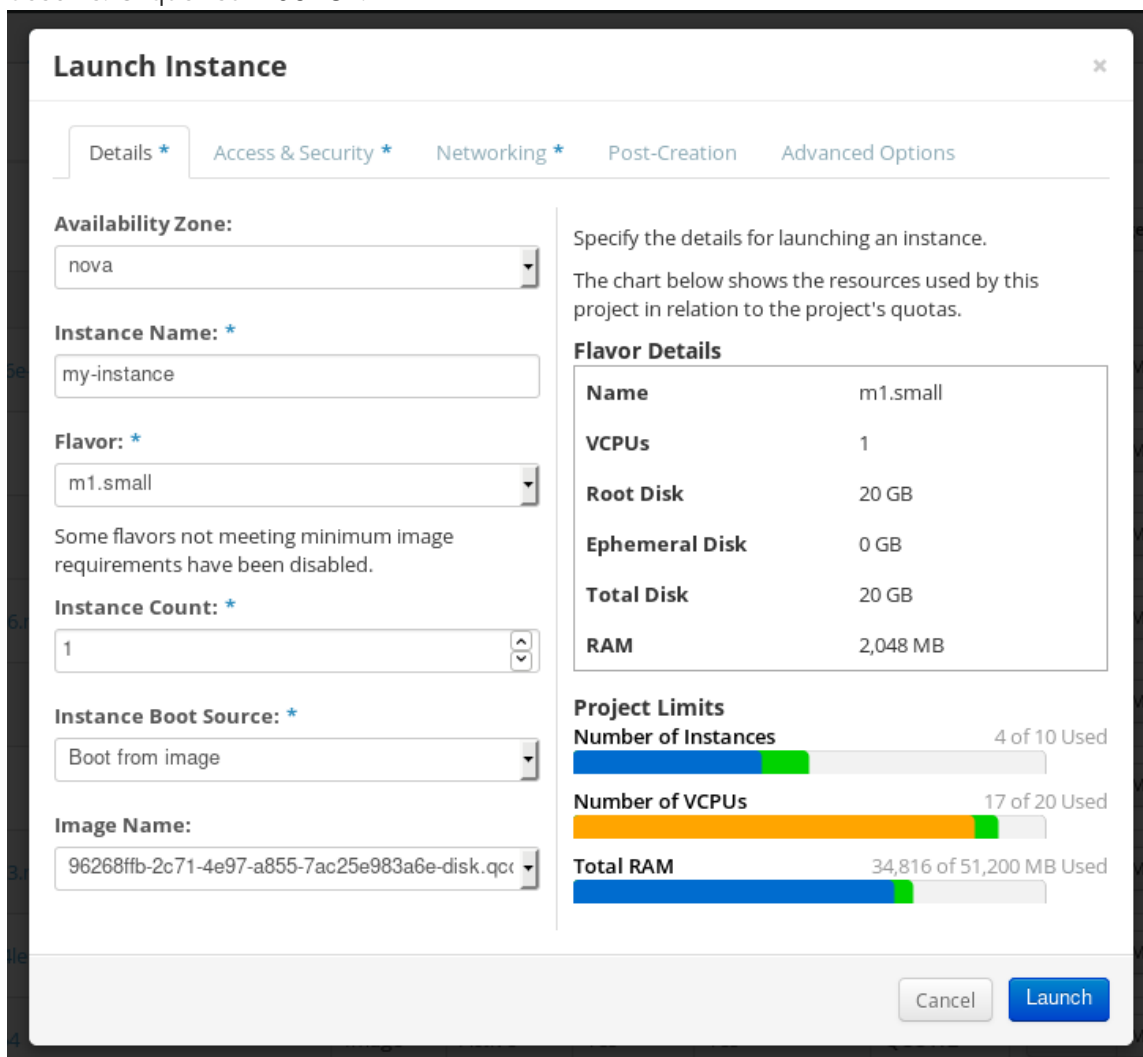
Minimum Ram (MB):
1024

Public:

Protected:

Cancel Create Image

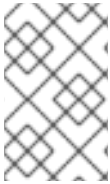
7. Dans le menu de gauche, sélectionnez l'onglet **Project**.
 - a. Dans le menu **Compute**, sélectionnez **Instances**.
 - b. Cliquez sur le bouton **Launch Instance** sur le bouton L'assistant **Launch Instance** s'ouvre.
 - c. Sur la page **Details**, saisissez un nom pour l'instance. Cliquez sur **Suivant**.
 - d. Sur la page **Source**, sélectionnez le nom de l'image que vous avez téléchargée. Cliquez sur **Suivant**.
 - e. Sur la page **Flavor**, sélectionnez les ressources machine qui correspondent le mieux à vos besoins. Cliquez sur **Launch**.



8. Vous pouvez exécuter l'instance de l'image en utilisant n'importe quel mécanisme (CLI ou OpenStack web UI) à partir de l'image. Utilisez votre clé privée via SSH pour accéder à l'instance résultante. Connectez-vous en tant que **cloud-user**.

10.11. PRÉPARATION DU TÉLÉCHARGEMENT D'IMAGES RHEL PERSONNALISÉES SUR ALIBABA

Pour déployer une image RHEL personnalisée sur le site **Alibaba Cloud**, vous devez d'abord vérifier l'image personnalisée. L'image a besoin d'une configuration spécifique pour démarrer avec succès, car Alibaba Cloud demande aux images personnalisées de répondre à certaines exigences avant de les utiliser.



NOTE

Image builder génère des images conformes aux exigences d'Alibaba. Cependant, Red Hat recommande également d'utiliser Alibaba **image_check tool** pour vérifier la conformité du format de votre image.

Conditions préalables

- Vous devez avoir créé une image Alibaba à l'aide de l'outil de création d'images.

Procédure

1. Connectez-vous au système contenant l'image que vous souhaitez vérifier en utilisant le site Alibaba **image_check tool**.
2. Télécharger le site **image_check tool**:

```
$ curl -O http://docs-aliyun.cn-hangzhou.oss.aliyun-inc.com/assets/attach/73848/cn_zh/1557459863884/image_check
```

3. Modifier les droits d'accès au fichier de l'outil de mise en conformité des images :

```
# chmod +x image_check
```

4. Exécutez la commande pour lancer le contrôle de l'outil de conformité de l'image :

```
# ./image_check
```

L'outil vérifie la configuration du système et génère un rapport qui s'affiche à l'écran. L'outil `image_check` enregistre ce rapport dans le même dossier que l'outil `image compliance`.

Résolution de problèmes

Si l'un des sites **Detection Items** échoue, suivez les instructions du terminal pour y remédier. Voir le lien : [Section sur les éléments de détection](#).

Ressources supplémentaires

- [Outil de conformité des images](#).

10.12. TÉLÉCHARGEMENT D'IMAGES RHEL PERSONNALISÉES SUR ALIBABA

Vous pouvez télécharger l'image **AMI** personnalisée que vous avez créée à l'aide de l'outil de création d'images vers le service de stockage d'objets (OSS).

Conditions préalables

- Votre système est configuré pour télécharger des images Alibaba. Voir [Préparation au téléchargement d'images vers Alibaba](#).
- Vous avez créé une image **ami** à l'aide de l'outil de création d'images.
- Vous avez un seau. Voir [Création d'un seau](#).

- Vous avez un [compte Alibaba actif](#).
- Vous avez activé [OSS](#).

Procédure

1. Connectez-vous à la [console OSS](#).
2. Dans le menu Bucket sur la gauche, sélectionnez le bucket dans lequel vous souhaitez télécharger une image.
3. Dans le menu supérieur droit, cliquez sur l'onglet **Fichiers**.
4. Cliquez sur **Upload**. Une fenêtre de dialogue s'ouvre à droite. Configurez les éléments suivants :
 - **Upload To**: Choisissez de télécharger le fichier dans le répertoire **Current** ou dans un répertoire **Specified**.
 - **File ACL**: Choisissez le type d'autorisation du fichier téléchargé.
5. Cliquez sur **Upload**.
6. Sélectionnez l'image que vous souhaitez télécharger.
7. Cliquez sur **Ouvrir**.

En conséquence, l'image personnalisée **AMI** est téléchargée dans la console OSS.

Ressources supplémentaires

- [Télécharger un objet](#).
- [Création d'une instance à partir d'images personnalisées](#).
- [Importation d'images](#).

10.13. IMPORTATION D'IMAGES SUR ALIBABA

Pour importer une image Alibaba RHEL personnalisée que vous avez créée à l'aide de l'outil de création d'images dans l'Elastic Cloud Console (ECS), procédez comme suit :

Conditions préalables

- Votre système est configuré pour télécharger des images Alibaba. Voir [Préparation au téléchargement d'images vers Alibaba](#).
- Vous avez créé une image **ami** à l'aide de l'outil de création d'images.
- Vous avez un seau. Voir [Création d'un seau](#).
- Vous avez un [compte Alibaba actif](#).
- Vous avez activé [OSS](#).
- Vous avez téléchargé l'image sur Object Storage Service (OSS). Voir [Téléchargement d'images sur Alibaba](#).

Procédure

1. Connectez-vous à la [console ECS](#).
 - i. Dans le menu de gauche, cliquez sur **Images**.
 - ii. En haut à droite, cliquez sur **Importer une image**. Une fenêtre de dialogue s'ouvre.
 - iii. Confirmez que vous avez configuré la région correcte où se trouve l'image. Saisissez les informations suivantes :
 - a. **OSS Object Address**: Voir comment obtenir l'[adresse de l'objet OSS](#).
 - b. **Image Name**
 - c. **Operating System**
 - d. **System Disk Size**
 - e. **System Architecture**
 - f. **Platform**: Red Hat
 - iv. Si vous le souhaitez, vous pouvez fournir les informations suivantes :
 - g. **Image Format** le format de l'image téléchargée est le suivant : **qcow2** ou **ami**, selon le format de l'image téléchargée.
 - h. **Image Description**
 - i. **Add Images of Data Disks**
L'adresse peut être déterminée dans la console de gestion OSS. Après avoir sélectionné le seau souhaité dans le menu de gauche :
2. Sélectionnez la section **Files**.
3. Cliquez sur le lien **Détails** à droite pour l'image appropriée.
Une fenêtre apparaît sur le côté droit de l'écran, affichant les détails de l'image. L'adresse de l'objet **OSS** se trouve dans la case **URL**.
4. Cliquez sur **OK**.



NOTE

La durée du processus d'importation peut varier en fonction de la taille de l'image.

L'image personnalisée est importée dans la console **ECS**.

Ressources supplémentaires

- [Notes pour l'importation d'images](#).
- [Création d'une instance à partir d'images personnalisées](#).
- [Télécharger un objet](#).

10.14. CRÉATION D'UNE INSTANCE D'UNE IMAGE RHEL PERSONNALISÉE À L'AIDE D'ALIBABA

Vous pouvez créer des instances d'une image RHEL personnalisée à l'aide de **Alibaba ECS Console**.

Conditions préalables

- Vous avez activé [OSS](#) et téléchargé votre image personnalisée.
- Vous avez importé avec succès votre image dans la console ECS. Voir [Importer des images sur Alibaba](#).

Procédure

1. Connectez-vous à la [console ECS](#).
2. Dans le menu de gauche, sélectionnez **Instances**.
3. Dans le coin supérieur droit, cliquez sur **Create Instance**. Vous êtes redirigé vers une nouvelle fenêtre.
4. Complétez toutes les informations requises. Pour plus de détails, voir [Création d'une instance à l'aide de l'assistant](#).
5. Cliquez sur **Create Instance** et confirmez la commande.



NOTE

Vous pouvez voir l'option **Create Order** au lieu de **Create Instance**, selon votre abonnement.

Vous disposez ainsi d'une instance active prête à être déployée à partir du site **Alibaba ECS Console**.

Ressources supplémentaires

- [Création d'une instance en utilisant une image personnalisée](#).
- [Créer une instance en utilisant l'assistant](#).