



Red Hat Enterprise Linux 8

RHEL for Edge イメージのインストールおよび 管理

Red Hat Enterprise Linux 8 での RHEL for Edge イメージの作成、インストール、および管理

Red Hat Enterprise Linux 8 RHEL for Edge イメージのインストールおよび管理

Red Hat Enterprise Linux 8 での RHEL for Edge イメージの作成、インストール、および管理

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2021 | You need to change the HOLDER entity in the en-US/Composing_Installing_and_Managing_RHEL_for_Edge_images.ent file |.

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.

概要

本ガイドは、Image Builder を使用してカスタマイズした RHEL (rpm-ostree) イメージを作成し、Edge サーバーでイメージをリモートでインストールおよび管理するユーザーを対象としています。

目次

多様性を受け入れるオープンソースの強化	4
RED HAT ドキュメントへのフィードバック (英語のみ)	5
第1章 RHEL FOR EDGE イメージの概要	6
1.1. RHEL FOR EDGE 対応アーキテクチャー	6
1.2. RHEL FOR EDGE イメージの作成とデプロイ方法	6
1.3. RHEL RPM イメージと RHEL FOR EDGE イメージの違い	9
第2章 IMAGE BUILDER の設定	11
2.1. IMAGE BUILDER のシステム要件	11
2.2. IMAGE BUILDER のインストール	11
第3章 RHEL WEB コンソールで IMAGE BUILDER を使用した RHEL FOR EDGE イメージの作成	13
3.1. RHEL WEB コンソールでの IMAGE BUILDER へのアクセス	13
3.2. RHEL WEB コンソールで IMAGE BUILDER を使用した RHEL FOR EDGE イメージの BLUEPRINT の作成	14
3.3. RHEL FOR EDGE イメージの BLUEPRINT へのソースの追加	14
3.4. RHEL FOR EDGE イメージの BLUEPRINT の管理ユーザーアカウントの作成	15
3.5. RHEL WEB コンソールで IMAGE BUILDER を使用した RHEL FOR EDGE イメージの作成	16
3.6. RHEL FOR EDGE イメージのダウンロード	17
第4章 IMAGE BUILDER コマンドラインを使用した RHEL FOR EDGE イメージの作成	18
4.1. IMAGE BUILDER コマンドラインインターフェースを使用した RHEL FOR EDGE イメージの BLUEPRINT の作成	18
4.2. IMAGE BUILDER コマンドラインインターフェースを使用した RHEL FOR EDGE イメージの作成	20
4.3. ネットワーク以外のデプロイメント向けのコマンドラインインターフェースを使用した RHEL FOR EDGE インストーラーイメージの作成	21
4.4. IMAGE BUILDER コマンドラインインターフェースを使用した RHEL FOR EDGE イメージのダウンロード	22
第5章 RHEL FOR EDGE イメージのデプロイ	23
5.1. RHEL FOR EDGE イメージのコミットの展開	23
5.2. RHEL FOR EDGE イメージをインストールするための WEB サーバーの設定	25
5.3. RHEL BOOT.ISO イメージのダウンロード	26
5.4. キックスタートファイルを使用した RHEL FOR EDGE イメージのインストール	26
5.5. ネットワークベース以外のデプロイメント用の RHEL FOR EDGE コンテナイメージの作成	27
5.6. ネットワーク以外のデプロイメント用の RHEL FOR EDGE インストーラーイメージの作成	28
5.7. ネットワークベース以外のデプロイメント向けの RHEL FOR EDGE イメージのインストール	29
第6章 RHEL FOR EDGE イメージの管理	31
6.1. RHEL WEB コンソールで IMAGE BUILDER を使用した RHEL FOR EDGE イメージの BLUEPRINT の編集	31
6.1.1. RHEL Web コンソールで Image Builder を使用した RHEL for Edge イメージの Blueprint へのコンポーネントの追加	31
6.1.2. RHEL Web コンソールを使用した RHEL for Edge イメージの Blueprint 内の既存コンポーネントのバージョンの変更	32
6.1.3. RHEL Web コンソールで Image Builder を使用した RHEL for Edge イメージの Blueprint からのコンポーネントの削除	32
6.2. コマンドラインインターフェースを使用した RHEL FOR EDGE イメージの BLUEPRINT の編集	33
6.3. RHEL FOR EDGE イメージの更新	34
6.3.1. RHEL for Edge イメージの更新のデプロイ方法	34
6.3.2. RHEL for Edge イメージの更新の手動でのデプロイ	35
6.3.3. ネットワーク以外のデプロイメント向けの RHEL for Edge イメージ更新の手動デプロイ	37
6.3.4. RHEL for Edge の自動イメージ更新のデプロイ	40
6.3.5. RHEL for Edge イメージの更新ポリシーの更新	40

6.3.6. RHEL for Edge の更新の自動ダウンロードとステージングの有効化	41
6.4. RHEL FOR EDGE イメージのロールバック	42
6.4.1. RHEL for Edge イメージのロールバック方法	42
6.4.2. RHEL for Edge イメージの手動でのロールバック	42
6.4.3. 自動化プロセスを使用した RHEL for Edge イメージのロールバック	43
付録A 用語とコマンド	45
A.1. OSTREE と RPM-OSTREE の用語	45
A.2. OSTREE コマンド	45
A.3. RPM-OSTREE コマンド	46

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[弊社](#) の CTO、Chris Wright の [メッセージ](#) を参照してください。

RED HAT ドキュメントへのフィードバック (英語のみ)

ご意見ご要望をお聞かせください。ドキュメントの改善点はございませんか。改善点を報告する場合は、以下のように行います。

- 特定の文章に簡単なコメントを記入する場合は、以下の手順を行います。
 1. ドキュメントの表示が **Multi-page HTML** 形式になっていて、ドキュメントの右上端に **Feedback** ボタンがあることを確認してください。
 2. マウスカーソルで、コメントを追加する部分を強調表示します。
 3. そのテキストの下に表示される **Add Feedback** ポップアップをクリックします。
 4. 表示される手順に従ってください。
- より詳細なフィードバックを行う場合は、Bugzilla のチケットを作成します。
 1. [Bugzilla](#) の Web サイトにアクセスします。
 2. Component で **Documentation** を選択します。
 3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも記入してください。
 4. **Submit Bug** をクリックします。

第1章 RHEL FOR EDGE イメージの概要

RHEL for Edge イメージは、Edge サーバーで RHEL をリモートにインストールするシステムパッケージを含む rpm-ostree イメージです。

システムパッケージには以下が含まれます。

- ベース OS パッケージ
- コンテナエンジンとしての podman
- 追加の RPM コンテンツ

イメージをカスタマイズして、要件に従って OS コンテンツを設定できます。また、それらをオンプレミス、クラウド、またはコンテナ環境にデプロイできます。

Bare Metal、Appliance、および Edge サーバーに RHEL for Edge イメージをデプロイすることができます。

RHEL for Edge イメージを使用すると、以下を実行できます。

<p>Atomic upgrades</p> <p>State of each update is known / no changes are seen until reboot</p>	<p>Custom health checks and intelligent rollbacks</p> <p>Resiliency in case of failed upgrades</p>	<p>Container-focused workflow</p> <p>Separate core OS updates</p>	<p>Optimized OTA payloads</p> <p>Transfer only delta updates</p>
---	---	--	---

125_RHEL_1020

1.1. RHEL FOR EDGE 対応アーキテクチャー

現在、AMD と Intel 64 ビットシステムに RHEL for Edge イメージをデプロイすることができます。

1.2. RHEL FOR EDGE イメージの作成とデプロイ方法

RHEL for Edge イメージを作成およびデプロイする全体的なプロセスは、デプロイメント環境がネットワークベースであるか、そうでないかにより異なります。

RHEL for Edge イメージの作成およびデプロイには、以下の 2 つのフェーズが含まれます。

1. Image Builder を使用して RHEL rpm-ostree イメージの作成。composer-cli ツールのコマンドラインインターフェースで Image Builder にアクセスするか、RHEL 8 Web コンソールでグラフィカルユーザーインターフェースを使用できます。
2. RHEL インストーラーを使用したイメージのデプロイ

RHEL for Edge イメージの作成中、以下のイメージタイプのいずれかを選択できます。

- RHEL for Edge Commit(.tar)- ネットワークベースの環境に適しています。コミットイメージタイプは、フルオペレーティングシステムが含まれている場合でも直接起動できません。コミットイメージの種類を起動する場合は、これをデプロイする必要があります。
- RHEL for Edge Container(.tar)- ネットワークベース以外のデプロイメントに適しています。イメージタイプは OSTree コミットを作成し、これを Web サーバーのある OCI コンテナに組み込む必要があります。コンテナが起動すると、Web サーバーはコミットを OSTree リポジ

トリーとして提供します。

- RHEL for Edge Installer(.iso)- ネットワーク以外のデプロイメントに適しています。イメージタイプは実行中のコンテナからコミットをプルするため、埋め込み OSTree コミットを使用するように設定されたキックスタートファイルでインストール可能なブート ISO を作成します。

イメージタイプはその内容によって異なるため、各種デプロイメント環境に適しています。RHEL for Edge Commit はネットワークベースの環境に適していますが、RHEL for Edge Container および RHEL for Edge Installer イメージはネットワークベース以外のデプロイメントに適しています。

デプロイメント環境ごとに概要手順を以下に示します。

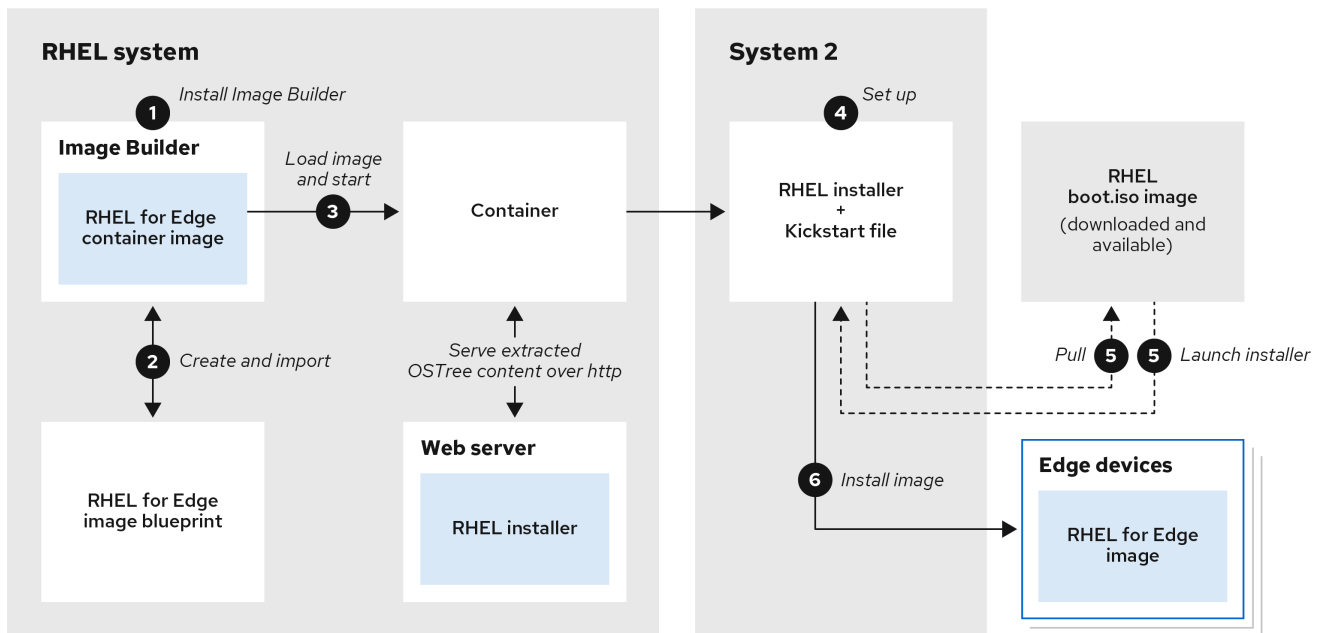
ネットワーク以外のデプロイメント

ネットワークベース以外のデプロイメントに RHEL for Edge イメージを作成し、デプロイするには、以下の概要手順を実施します。

1. RHEL システムをインストールして登録する
2. Image Builder をインストールする
3. Image Builder を使用して RHEL for Edge コンテナイメージの Blueprint を作成する
4. RHEL for Edge の Blueprint を Image Builder にインポートする
5. OSTree リポジトリとしてコミットをデプロイする準備が整った Web サーバーで、OCI コンテナに組み込む RHEL for Edge イメージを作成する
6. RHEL for Edge イメージコンテナファイルをダウンロードする
7. RHEL for Edge Commit でリポジトリを提供するコンテナをデプロイする
8. Image Builder を使用して RHEL for Edge インストーラーイメージの空の Blueprint を作成する
9. RHEL boot.iso イメージをダウンロードする
10. インストールを実行する
11. イメージを作成する

以下の図は、RHEL for Edge イメージ以外のデプロイメントワークフローを示しています。

図1.1 ネットワーク以外の環境での RHEL for Edge のデプロイ



164_RHEL_D621

ネットワークベースのデプロイメント

ネットワークベースのデプロイメントに RHEL for Edge イメージを作成し、デプロイするには、以下の概要手順を実施します。

Image Builder を使用して、要件に沿った柔軟な RHEL **rpm-ostree** イメージを作成し、Anaconda を使用してお使いの環境にデプロイします。Image Builder は、デプロイメント設定の詳細を自動的に識別し、**.tar** ファイルの **rhel-edge-commit** としてイメージの出力を生成します。

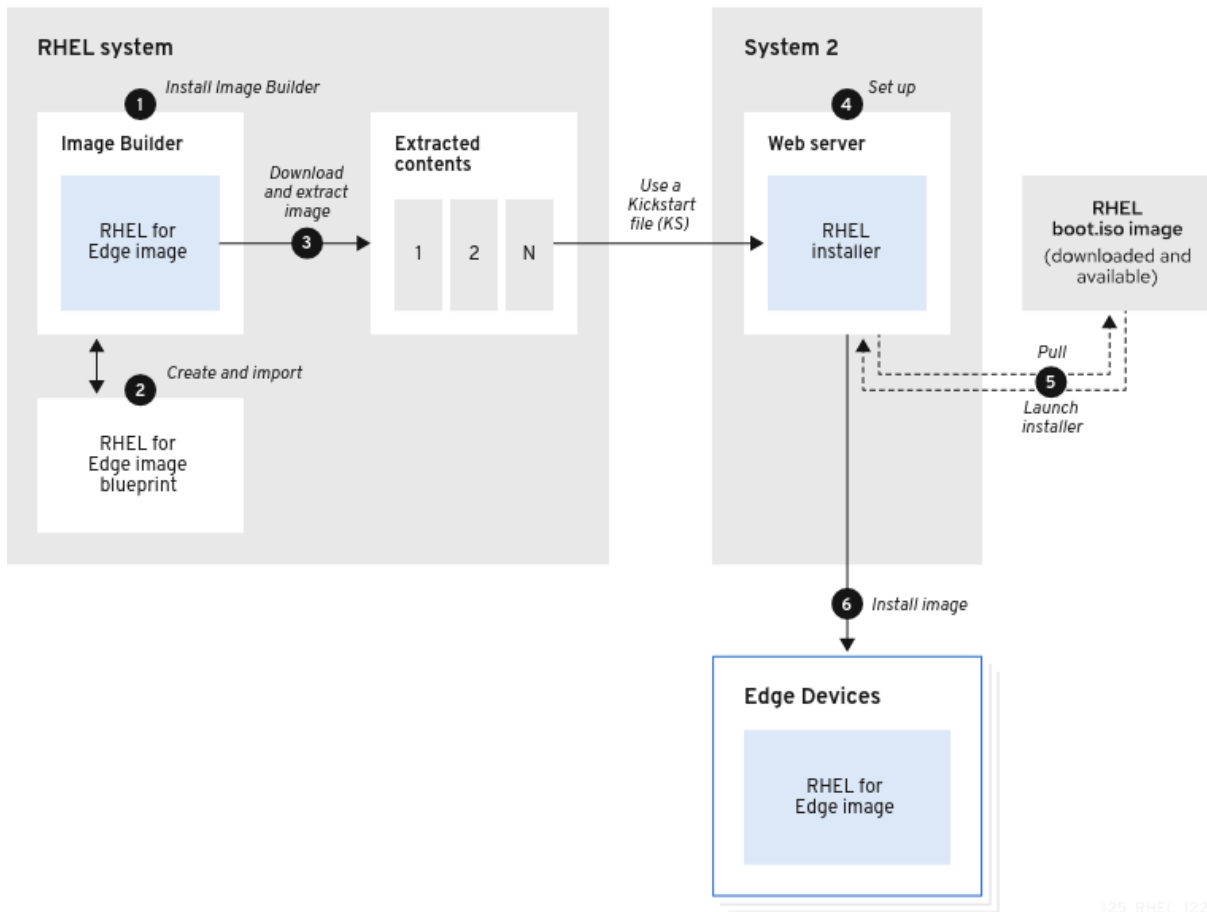
composer-cli ツールのコマンドラインインターフェース経由で Image Builder にアクセスするか、または RHEL 8 Web コンソールのグラフィカルユーザーインターフェースを使用することができます。

RHEL for Edge イメージは、以下の高レベルの手順を実行することで、作成およびデプロイすることができます。

1. RHEL システムのインストールと登録
2. Image Builder のインストール
3. Image Builder を使用した RHEL for Edge イメージの Blueprint の作成
4. RHEL for Edge の Blueprint の Image Builder へのインポート
5. RHEL for Edge イメージの作成
6. RHEL for Edge イメージファイルのダウンロード
7. Web サーバーの設定
8. RHEL boot.iso イメージのダウンロード
9. Web サーバーを使用した Edge デバイスへの RHEL for Edge イメージのインストール

以下の図は、RHEL for Edge ネットワークイメージのデプロイメントワークフローを示しています。

図1.2 ネットワークベース環境での RHEL for Edge のデプロイ



125_RHEL_1221

関連情報

RHEL システムのインストールと登録の詳細については、『[標準的な RHEL インストールの実行](#)』を参照してください。

1.3. RHEL RPM イメージと RHEL FOR EDGE イメージの違い

従来のパッケージベースの RPM 形式で RHEL システムイメージや、Edge (rpm-ostree) イメージ用の RHEL システムイメージを作成できます。

従来のパッケージベースの RPM を使用して、従来のデータセンターに RHEL をデプロイすることができます。ただし、RHEL for Edge イメージを使用すると、従来のデータセンター以外のサーバーに RHEL をデプロイすることができます。これらのサーバーには、大量のデータの処理が、データが生成されるソース (Edge サーバー) に最も近いところで行われるシステムが含まれます。

RHEL for Edge イメージがパッケージベースの RHEL RPM イメージとどのように異なるかを確認するには、以下の表を参照してください。

表1.1 RHEL RPM イメージと RHEL for Edge イメージの違い

キーの属性	RHEL RPM イメージ	RHEL for Edge イメージ

OS アセンブリー	パッケージをローカルでアセンブルして、イメージを形成できません。	パッケージは、システムにインストールできる ostree でアセンブルされます。
OS の更新	yum update を使用して、有効なリポジトリから利用可能な更新を適用することができます。	/etc/ostree/remotes.d/ の ostree リモートで新しいコミットが利用可能な場合は、 rpm-ostree upgrade を使用して更新をステージできます。更新はシステムの再起動時に有効になります。
リポジトリ	パッケージに Yum リポジトリが含まれる	パッケージに Ostree リモートリポジトリが含まれる
ユーザーアクセスパーミッション	読み書き	読み取り専用 (/usr)
データの永続性	tmpfs 以外のマウントポイントにイメージをマウントできます。	/etc および /var は読み書き可能で、永続的なデータを含みます。

第2章 IMAGE BUILDER の設定

Image Builder を使用して、カスタマイズした RHEL for Edge イメージを作成します。RHEL システムに Image Builder をインストールすると、Image Builder は RHEL Web コンソールでアプリケーションとして利用できます。**composer-cli** ツールのコマンドラインインターフェースで Image Builder にアクセスすることもできます。



注記

Image Builder を仮想マシンにインストールすることが推奨されます。

Image Builder をインストールする環境で、最初にシステム要件を満たしてから、インストールするようにしてください。

2.1. IMAGE BUILDER のシステム要件

Image Builder が実行する環境 (仮想マシンなど) は、次の表に記載されている要件を満たす必要があります。

表2.1 Image Builder のシステム要件

パラメーター	最低要求値
システムのタイプ	専用の仮想マシン
プロセッサ	2 コア
メモリー	4 GiB
ディスク容量	20 GiB
アクセス権限	管理者レベル (root)
ネットワーク	インターネットへの接続

2.2. IMAGE BUILDER のインストール

Image Builder を専用の仮想マシンにインストールするには、以下の手順を行います。

前提条件

- 仮想マシンが作成され、オンの状態になっている。
- RHEL をインストールし、RHSM または Redhat Satellite にサブスクライブしている。

手順

1. 仮想マシンに以下のパッケージをインストールします。
 - `osbuild-composer`

- composer-cli
- cockpit-composer
- Bash-completion

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

Image Builder は、RHEL Web コンソールでアプリケーションとしてインストールされていません。

2. 仮想マシンを再起動します。
3. Image Builder を有効にします。

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

lorax-composer サービスおよび cockpit サービスは、最初のアクセスで自動的に起動します。

4. Web コンソールへのアクセスを許可するように、システムのファイアウォールを設定します。

```
# firewall-cmd --add-service=cockpit && firewall-cmd --add-service=cockpit --permanent
```

5. システムを再起動しなくても、composer-cli コマンドのオートコンプリート機能がすぐに動作するように、シェル設定スクリプトを読み込みます。

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


第3章 RHEL WEB コンソールで IMAGE BUILDER を使用した RHEL FOR EDGE イメージの作成

Image Builder を使用して、カスタマイズした RHEL for Edge イメージ (OSTree コミット) を作成します。

Image Builder にアクセスし、カスタマイズした RHEL for Edge イメージを作成するには、RHEL Web コンソールインターフェースまたはコマンドラインインターフェースを使用します。

本セクションでは、Image Builder へのアクセスと、RHEL Web コンソールを使用した RHEL for Edge イメージの作成について説明します。Image Builder にアクセスし、CLI を使用して RHEL for Edge イメージを作成するには、「[4章 Image Builder コマンドラインを使用した RHEL for Edge イメージの作成](#)」を参照してください。

以下の概要手順を実行して、RHEL Web コンソールで Image Builder を使用し、RHEL for Edge イメージを作成することができます。

1. RHEL Web コンソールでの Image Builder へのアクセス
2. RHEL for Edge イメージの Blueprint の作成
3. RHEL for Edge イメージの Blueprint のユーザーアカウントの作成
4. RHEL for Edge イメージの作成
5. RHEL for Edge イメージのダウンロード

3.1. RHEL WEB コンソールでの IMAGE BUILDER へのアクセス

RHEL Web コンソールで Image Builder にアクセスするには、以下の前提条件を満たしていることを確認してから、手順に従ってください。

前提条件

- RHEL システムをインストール済みである。
- システムの管理者権限を持っている。
- RHEL システムを Redhat Subscription Manager (RHSM) または Red Hat Satellite Server にサブスクライブしている。
- システムに電源が入り、ネットワーク経由でアクセスできる。
- システムに Image Builder がインストール済みである。

手順

1. RHEL システムで、Web ブラウザーで <https://localhost:9090/> にアクセスします。
2. Image Builder にリモートでアクセスする方法の詳細は、『[RHEL 8 で Web コンソールを使用したシステムの管理](#)』を参照してください。
3. 管理ユーザーアカウントを使用して、Web コンソールにログインします。
4. Web コンソールで、左側のメニューの **Apps** をクリックします。

5. **Image Builder** をクリックします。
6. 右側のペインに Image Builder のダッシュボードが表示されます。
7. これで、RHEL for Edge イメージの Blueprint の作成に進むことができます。

3.2. RHEL WEB コンソールで IMAGE BUILDER を使用した RHEL FOR EDGE イメージの BLUEPRINT の作成

RHEL Web コンソールで Image Builder を使用して RHEL for Edge イメージの Blueprint を作成するには、以下の前提条件を満たしていることを確認してから、手順に従ってください。

前提条件

- RHEL システムで、Image Builder のダッシュボードを開いている。

手順

1. Image Builder のダッシュボードで **Create Blueprint** をクリックします。
Create Blueprint のダイアログボックスが表示されます。
2. 作成する Blueprint の名前と説明を指定します。
3. **Create** をクリックします。
ダッシュボードには、利用可能なコンポーネントの一覧が表示されます。
4. Available Components から、Blueprint に含めたいコンポーネントの **+** をクリックします。
特定のコンポーネントを検索するには、**Filter By Name** のテキストボックスにコンポーネント名を入力してから **Enter** キーを押します。**Component Details** ペインには、コンポーネントの詳細とそれに依存するコンポーネントが表示されます。必要なバージョンを選択することができます。Web コンソールでは、デフォルトで最新バージョンが選択されます。

Blueprint からコンポーネントを削除するには、**Available Components** ペインで、コンポーネント名に対応する **-** をクリックします。



注記

RHEL for Edge Installer(.tar)"を作成する Blueprint は空である必要があります。つまり、パッケージやユーザー作成などでカスタマイズしなようにします。

5. **Components Details** ペインで、**Add** をクリックします。
6. **Commit** をクリックして、Blueprint を保存します。
Blueprint の概要を示すダイアログボックスが表示されます。
7. **Commit** をクリックします。
8. のブレッドクラムから、**Back to Blueprints** をクリックします。
Image Builder のダッシュボードには、作成した Blueprint が一覧表示されます。

3.3. RHEL FOR EDGE イメージの BLUEPRINT へのソースの追加

Image Builder で定義したソースは、Blueprint に追加できるコンテンツを提供します。これらのソースはグローバルであるため、すべての Blueprint で利用可能です。システムソースは、コンピューターに

ローカルで設定されているリポジトリで、Image Builder からは削除できません。追加のカスタムソースを追加できることから、システムで利用できる **システムソース** 以外のコンテンツにアクセスできます。Blueprint に **ソース** を追加するには、以下の手順を実行します。

前提条件

- ブラウザーで、RHEL 8 Web コンソールの Image Builder インターフェースを開いている。

手順

1. 右上隅にある **Manage Sources** ボタンをクリックします。
利用可能なソース、その名前、および説明が記載されたポップアップウィンドウが表示されます。
2. ポップアップウィンドウの右側で、**Add Source** ボタンをクリックします。
3. 必要な **Source name**、**Source path**、および **Source Type** を追加します。
セキュリティーフィールドはオプションです。
4. **ソースの追加** をクリックします。画面には、利用可能なソースウィンドウが表示され、追加したソースが一覧表示されます。

これにより、新しいシステムソースを使用でき、変更を加えることができます。

3.4. RHEL FOR EDGE イメージの BLUEPRINT の管理ユーザーアカウントの作成

作成した RHEL for Edge イメージをビルドしてデプロイするには、イメージの Blueprint の管理ユーザー権限が必要です。Image Builder を使用して、イメージの Blueprint の管理ユーザーアカウントを作成します。続いて、このユーザーアカウントを使用して、作成したイメージをデプロイすることができます。管理ユーザーアカウントでは、パスワードベースのアクセスと SSH 鍵ベースのアクセスのどちらかを選択できます。

前提条件

- 作成するユーザーアカウントに使用する SSH 鍵を作成済みである。
- RHEL Web コンソールで Image Builder のダッシュボードにアクセス済みである。
- RHEL for Edge イメージの Blueprint を作成済みである。

手順

1. Image Builder のダッシュボードで、RHEL for Edge イメージの Blueprint を見つけます。
必要な Blueprint を検索するには、Filter by Name のテキストボックスに Blueprint 名を指定してから **Enter** キーを押します。
2. Blueprint 名をクリックします。
Image Builder は Blueprint の詳細を表示します。
3. Customizations タブで、**Create User Account** をクリックします。
4. Create User Account のダイアログボックスで、ユーザーアカウントに必要な詳細情報とパスワードを指定します。
パスワードベースのアクセスの場合は、ユーザーアカウントのパスワードを指定します。

SSH ベースのアクセスの場合は、ユーザーアカウントの SSH 鍵を指定します。



注記

作成するユーザーアカウントに管理者権限を付与する場合は、**Server administrator** チェックボックスを選択してください。

Image Builder は、指定したユーザーアカウントを作成し、詳細を表示します。

必要に応じて、追加のユーザーアカウントを作成することができます。

3.5. RHEL WEB コンソールで IMAGE BUILDER を使用した RHEL FOR EDGE イメージの作成

以下の手順に従って、RHEL Web コンソールで Image Builder を使用して RHEL for Edge イメージを作成します。

前提条件

- RHEL システムで、Image Builder ダッシュボードにアクセス済みである。
- RHEL for Edge イメージの Blueprint を作成済みである。

手順

1. Image Builder のダッシュボードで、RHEL for Edge イメージ用に作成した Blueprint の **Create Image** を **click** します。特定の Blueprint を検索するには、**Filter By Name** のテキストボックスに Blueprint 名を入力してから **Enter** キーを押します。
2. **Create Image** ウィンドウで、以下の手順を実行します。
 - a. **Type** のドロップダウンリストから、ネットワークベースのデプロイメント向けに **RHEL for Edge Commit(.tar)** を選択します。ネットワーク以外のデプロイメントの場合は、**RHEL for Edge Container(.tar)** を選択してコミットするか、**RHEL for Edge Installer(.tar)** を選択してそのコミットを使用して OS をコミットし、デプロイします。
 - b. **Parent commit** テキストボックスは、現時点でコミットがない場合は以前のコミットを指定するか、空欄のままにします。
 - c. **Ref** テキストボックスで、コミットを作成する場所の参照を指定します。デフォルトでは、Web コンソールは **rhel/8/arch_name/edge** を指定します。
 - d. **Create** をクリックします。



注記

Blueprint をすでに編集しているが、まだコミットしていない場合は、**Commit** をクリックして作成してください。

Image Builder は、作成した Blueprint 用の RHEL for Edge イメージを作成します。

RHEL for Edge イメージの作成の進捗状況を確認するには、ブレッドグラムから **blueprint name** をクリックし、次に **Images** タブをクリックします。



注記

イメージ作成プロセスの完了まで数分かかります。イメージ作成プロセスを中止するには、More Options メニューの Stop をクリックします。

3.6. RHEL FOR EDGE イメージのダウンロード

Image Builder が RHEL for Edge イメージを正常に作成したら、ローカルホストにイメージをダウンロードします。

手順

以下は、画像をダウンロードするための手順です。

1. **More Options** メニューの **Download** をクリックします。

Image Builder は、デフォルトのダウンロード場所にファイルをダウンロードします。

ダウンロードされたファイルは、OSTree リポジトリを持つ **.tar** ファイルで構成されています。このリポジトリには、コミットと、リポジトリのコンテンツに関する情報のメタデータを含む **json** ファイルが含まれています。

第4章 IMAGE BUILDER コマンドラインを使用した RHEL FOR EDGE イメージの作成

Image Builder を使用して、カスタマイズした RHEL for Edge イメージ (OSTree コミット) を作成します。

Image Builder にアクセスし、カスタマイズした RHEL for Edge イメージを作成するには、RHEL Web コンソールインターフェースまたはコマンドラインインターフェースを使用します。

この章では、CLI を使用して RHEL for Edge イメージを作成する方法について説明します。RHEL Web コンソールを使用して RHEL for Edge イメージを作成するには、「[3章 RHEL Web コンソールで Image Builder を使用した RHEL for Edge イメージの作成](#)」を参照してください。

ネットワークベースのデプロイメントの場合、CLI を使用して RHEL for Edge イメージを作成するワークフローでは、以下の概要手順を実施します。

1. RHEL for Edge イメージの Blueprint の作成
2. RHEL for Edge イメージの作成
3. RHEL for Edge イメージのダウンロード

手順を実行するには、**composer-cli** パッケージを使用します。



注記

composer-cli コマンドを root 以外のユーザーとして実行するには、weldr グループの一員であるか、システムへの管理者アクセス権を持っている必要があります。

ネットワーク以外のデプロイメントの場合、CLI を使用して RHEL for Edge イメージを作成するワークフローでは、以下の概要手順を実施します。

1. RHEL for Edge イメージの Blueprint の作成
2. RHEL for Edge コンテナイメージを作成する
3. RHEL for Edge Installer イメージを作成する
4. RHEL for Edge イメージをダウンロードする

4.1. IMAGE BUILDER コマンドラインインターフェースを使用した RHEL FOR EDGE イメージの BLUEPRINT の作成

RHEL for Edge イメージの Blueprint を作成するには、以下の手順を実行します。

手順

1. TOML (Tom's Obvious, Minimal Language) 形式で、以下のコンテンツのプレーンテキストファイルを作成します。

```
name = "blueprint-name"
description = "blueprint-text-description"
version = "0.0.1"
```

```
modules = [ ]
groups = [ ]
```

詳細は以下のようになります。

- **blueprint-name** は名前、**blueprint-text-description** は Blueprint の説明になります。
 - **0.0.1** は、Semantic Versioning スキームに従って、バージョン番号に置き換えます。
 - **モジュール** は、イメージにインストールされるパッケージ名とそれに対応するバージョングロブを記述します (例: パッケージ名 = "tmux"、対応するバージョングロブのバージョン = "2.9a")。現在、パッケージとモジュールには違いがないことに注意してください。
 - **グループ** は、イメージにインストールされるパッケージグループです (例: グループパッケージ `anaconda-tools`)。このとき、モジュールおよびグループがわからない場合は、空欄のままにしておきます。
2. 要件に沿うように、必要なパッケージを含め、Blueprint の他の詳細をカスタマイズします。Blueprint に含めたり、カスタマイズしたりできるパッケージの詳細は、[「サポートされているイメージのカスタマイズ」](#) 参照してください。

Blueprint に含むすべてのパッケージについて、以下の行をファイルに追加します。

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

詳細は以下のようになります。

- **package-name** は、パッケージ名に置き換えます (例: `httpd`, `gdb-doc`, `coreutils` など)。
 - **package-version** は、使用するパッケージのバージョン番号に置き換えます。**package-version** は、以下の `dnf` バージョン仕様をサポートしています。
 - 特定のバージョンを指定する場合は、バージョン番号を正確に指定してください (例: `8.3.0` など)。
 - 利用可能な最新バージョンを指定する場合は、アスタリスク `*` を使用します。
 - 最新のマイナーバージョンを指定する場合は、`8.*` などの形式を使用してください。
3. Blueprint を Image Builder サーバーにプッシュ (インポート) します。

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

4. 既存の Blueprint を一覧表示して、作成した Blueprint が正常にプッシュされて存在するかどうかを確認します。

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

5. Blueprint に一覧表示されているコンポーネントおよびバージョンと、その依存関係が有効かどうかを確認します。

```
# composer-cli blueprints depsolve blueprint-name
```

4.2. IMAGE BUILDER コマンドラインインターフェースを使用した RHEL FOR EDGE イメージの作成

Image Builder コマンドラインインターフェースを使用して RHEL for Edge イメージを作成するには、以下の前提条件を満たしていることを確認してから、手順に従ってください。

前提条件

- RHEL for Edge イメージの Blueprint を作成済みである。

手順

1. RHEL for Edge イメージの作成を開始します。

```
# composer-cli compose start-ostree --ref rhel/8/x86_64/edge --url URL-OSTree-repository  
blueprint-name image-type
```

詳細は以下のようになります。

- `--ref` は、お客様が `ostree` リポジトリの構築に使用した値と同じです。
- `--url` は、イメージに埋め込むコミットの OSTree リポジトリへの URL です。例:
<http://10.0.2.2:8080/repository/>「[RHEL for Edge イメージをインストールするための Web サーバーの設定](#)」を参照してください。
- `blueprint-name` は RHEL for Edge の Blueprint 名です。
- `image-type` は、ネットワークベースのデプロイメントにおける `rhel-edge-commit` か、ネットワークベース以外のデプロイメントの場合は `rhel-edge-container` を選択します。`composer` プロセスがキューに追加されたことを確認する画面が表示されます。また、作成されたイメージの UUID (Universally Unique Identifier) 番号も表示されます。UUID 番号を使用してビルドを追跡します。また、更なるタスクのために UUID 番号を手元に保管しておきます。

サポートされるイメージタイプの詳細は、「[コマンドラインインターフェースで Image Builder を使用したシステムイメージの作成](#)」を参照してください。

2. イメージの作成状態を確認します。

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

出力には、以下の形式で状態が表示されます。

```
<UUID> RUNNING date blueprint-name blueprint-version image-type
```



注記

イメージ作成プロセスの完了まで数分かかります。

イメージ作成プロセスを中断するには、以下を実行します。

```
# composer-cli compose cancel <UUID>
```


既存のイメージを削除するには、以下のコマンドを実行します。

```
# composer-cli compose delete <UUID>
```

4.3. ネットワーク以外のデプロイメント向けのコマンドラインインターフェースを使用した RHEL FOR EDGE インストーラーイメージの作成

Image Builder コマンドラインインターフェースを使用して OSTree コミットを組み込む RHEL for Edge インストーラーイメージを作成するには、以下の前提条件を満たしていることを確認してから、手順に従ってください。

前提条件

- RHEL for Edge イメージの **空** の Blueprint を作成済みである。
空の Blueprint にはカスタマイズがありません。つまり、パッケージが追加されず、ユーザーは作成されません。追加されたパッケージおよび作成されたユーザーは、ISO イメージのビルドに使用されるリポジトリからプルされます。

手順

1. RHEL for Edge イメージの作成を開始します。

```
# composer-cli compose start-ostree --ref rhel/8/x86_64/edge --url URL-OSTree-repository  
blueprint-name image-type
```

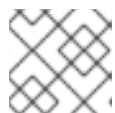
詳細は以下のようになります。

- **ref** は、お客様が ostree リポジトリの構築に使用した値と同じです。
 - **URL-OSTree-repository** は、イメージに埋め込むコミットの OSTree リポジトリへの URL です。例: <http://10.0.2.2:8080/repository/> 「[RHEL for Edge イメージをインストールするための Web サーバーの設定](#)」を参照してください。
 - **blueprint-name** は RHEL for Edge の Blueprint 名です。
 - **image-type** は **rhel-edge-installer** です。
composer プロセスがキューに追加されたことを確認する画面が表示されます。また、作成されたイメージの UUID (Universally Unique Identifier) 番号も表示されます。UUID 番号を使用してビルドを追跡します。また、更なるタスクのために UUID 番号を手元に保管しておきます。
2. イメージの作成状態を確認します。

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

コマンド出力には、以下の形式で状態が表示されます。

```
<UUID> RUNNING date blueprint-name blueprint-version image-type
```



注記

イメージ作成プロセスの完了まで数分かかります。

イメージ作成プロセスを中断するには、以下を実行します。

```
# composer-cli compose cancel <UUID>
```

既存のイメージを削除するには、以下のコマンドを実行します。

```
# composer-cli compose delete <UUID>
```

4.4. IMAGE BUILDER コマンドラインインターフェースを使用した RHEL FOR EDGE イメージのダウンロード

Image Builder コマンドラインインターフェースを使用して RHEL for Edge イメージをダウンロードするには、以下の前提条件を満たしていることを確認してから、手順に従ってください。

前提条件

- RHEL for Edge イメージを作成済みである。

手順

1. RHEL for Edge イメージの状態を確認します。

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

出力には、以下が表示される必要があります。

```
$ <UUID> FINISHED date blueprint-name blueprint-version image-type
```

2. イメージをダウンロードしてください。

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

Image Builder は、イメージを tar ファイルとして **<var/lib/osbuild-composer/composer/results/<UUID>/compose/>** にダウンロードします。

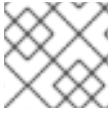
UUID 番号とイメージサイズは並んで表示されます。

```
$ <UUID>-commit.tar: size MB
```

イメージには、コミットと、リポジトリのコンテンツに関する情報のメタデータを含む **json** ファイルが含まれています。

第5章 RHEL FOR EDGE イメージのデプロイ

RHEL インストーラーのグラフィカルユーザーインターフェースまたはキックスタートファイルを使用して、RHEL for Edge イメージをデプロイできます。RHEL for Edge イメージをデプロイする全体的なプロセスは、デプロイメント環境がネットワークベースであるかそうでないかによって異なります。



注記

ベアメタルにイメージをデプロイするには、キックスタートファイルを使用します。

ネットワークベースのデプロイメント

ネットワークベースの環境に RHEL for Edge イメージをデプロイするには、以下の概要手順を実施します。

1. イメージファイルのコンテンツの展開。RHEL for Edge イメージのダウンロード方法の詳細は、「[RHEL for Edge イメージのダウンロード](#)」を参照してください。
2. Web サーバーを設定する。
3. イメージをインストールする。

ネットワーク以外のデプロイメント

ネットワークベース以外の環境に RHEL for Edge イメージをデプロイするには、以下の概要手順を実施します。

1. RHEL for Edge Container および RHEL for Edge イメージをダウンロードする。RHEL for Edge イメージのダウンロード方法の詳細は、「[RHEL for Edge イメージのダウンロード](#)」を参照してください。
2. RHEL for Edge コンテナイメージを Podman にロードする。
3. RHEL for Edge コンテナイメージを Podman で実行する。
4. 空の Blueprint をロードする。
5. RHEL for Edge Installer イメージをビルドする。
6. qcow ディスクを準備する。
7. 仮想マシンを起動する。
8. イメージをインストールする。

5.1. RHEL FOR EDGE イメージのコミットの展開

コミットをダウンロードしたら、.tar ファイルを展開して ref 名とコミット ID にアクセスします。

ダウンロードしたコミットファイルは、OSTree リポジトリを持つ .tar ファイルで構成されています。OSTree リポジトリには、コミットと **compose.json** ファイルが含まれます。

compose.json ファイルには、「Ref」(リファレンス ID) やコミット ID などの情報が含まれるコミットに関する情報のメタデータが含まれています。コミットには rpm パッケージが含まれます。

パッケージのコンテンツを展開するには、以下の手順を実行します。

前提条件

1. キックスタートファイルを作成するか、既存のファイルを使用する。

手順

1. ダウンロードしたイメージの tar ファイルを展開します。

```
# tar xvf <UUID>-commit.tar
```

2. **.tar** ファイルを展開したディレクトリーに移動します。
compose.json ファイルと OSTree ディレクトリーが含まれます。**compose.json** ファイルにはコミット番号が含まれ、OSTree ディレクトリーには rpm パッケージが含まれます。

3. **compose.json** ファイルを開き、コミット ID 番号を書き留めます。Web サーバーの設定を進める際には、この番号が必要になります。
jq JSON プロセッサがインストールされている場合は、**jq ツール**を使用してコミットハッシュ ID を取得することもできます。

```
# jq '!.["ostree-commit"]' < compose.json
```

4. コミットの **rpm** パッケージを一覧表示します。

```
# rpm-ostree db list rhel/8/x86_64/edge --repo=repo
```

5. キックスタートファイルを使用して、RHEL インストーラーを実行します。既存のファイルを使用するか、キックスタートジェネレーターツールを使用してファイルを作成できます。
キックスタートファイルでは、ファイルシステムのプロビジョニング、ユーザーの作成、RHEL for Edge イメージの取得とデプロイの方法についての詳細を含めるようにしてください。
RHEL インストーラーは、インストールプロセス中にこの情報を使用します。

以下は、キックスタートファイルの例です。

```
lang en_US.UTF-8
keyboard us
timezone Etc/UTC --isUtc
text
zerombr
clearpart --all --initlabel
autopart
reboot
user --name=core --group=wheel
sshkey --username=core "ssh-rsa AAAA3Nza...."

ostresetup --nogpg --osname=rhel --remote=edge
--url=https://mirror.example.com/repo/
--ref=rhel/8/x86_64/edge
```

関連情報

キックスタートファイルの作成方法は以下を参照してください。

- [高度な RHEL インストールの実行](#)

5.2. RHEL FOR EDGE イメージをインストールするための WEB サーバーの設定

RHEL for Edge イメージのコンテンツを展開したら、HTTP 経由で RHEL インストーラーにイメージのコミット詳細を提供するための Web サーバーを設定します。

以下の例では、コンテナを使用して Web サーバーを設定する手順を説明します。

前提条件

- システムに Podman をインストール済みである。 [「How do I install podman in RHEL 8」](#) を参照してください。

手順

- 以下の手順により Dockerfile を作成します。

```
FROM registry.access.redhat.com/ubi8/ubi
RUN yum -y install httpd && yum clean all
ADD kickstart.ks /var/www/html/
ARG commit=commit.tar
ADD $commit /var/www/html/
EXPOSE 80
CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
```

詳細は以下のようになります。

- kickstart.ks** は、RHEL for Edge イメージのキックスタートファイルの名前です。キックスタートファイルには、ディレクティブの情報が含まれています。後でイメージを管理しやすくするためにも、Greenboot チェックのチェックおよび設定を含むことが推奨されます。そのためには、以下の設定を含むようにキックスタートファイルを更新します。

```
/etc/greenboot/check/required.d/check-dns.sh

#!/bin/bash

DNS_SERVER=$(grep nameserver /etc/resolv.conf | cut -f2 -d" ")
COUNT=0

# check DNS server is available
ping -c1 $DNS_SERVER
while [ $? != '0' ] && [ $COUNT -lt 10 ]; do
((COUNT++))
echo "Checking for DNS: Attempt $COUNT ."
sleep 10
ping -c 1 $DNS_SERVER
done
```

- ARG commit=commit.tar** 行は、コンテナビルド時に **commit** 引数に使用されるデフォルト値です。コンテナビルド時に、異なる **tar** ファイルをビルド引数として指定できます。
- ADD \$commit /var/www/html** コマンドは **commit.tar** ファイルをコピーし、選択した場所に自動的に展開します。コミットを展開して保存することもできます。

OStree リポジトリは、任意の HTTP サービスからホストでき、コンテナの使用はホスティング方法の一例です。docker ファイルは、以下のタスクを実行します。

- a. 最新の Universal Base Image (ubi) の使用
 - b. Web サーバー (httpd) のインストール
 - c. キックスタートファイルのサーバーへの追加
 - d. RHEL for Edge イメージのコミットのサーバーへの追加
ARG commit=commit.tar の行では、コマンドラインを使って将来のコミットを指定することができます。
2. docker コンテナをビルドします。

```
# podman build -t name-of-container-image --build-arg commit=uuid-commit.tar .
```

3. コンテナを実行します。

```
# podman run --rm -d -p port:80 localhost/name-of-container-image
```

これにより、サーバーがセットアップされ、**commit.tar** リポジトリとキックスタートファイルを使用して、RHEL インストーラーを起動する準備が整いました。

5.3. RHEL BOOT.ISO イメージのダウンロード

Red Hat Boot ISO イメージは、Red Hat カスタマーポータルからダウンロードできます。Red Hat Boot ISO イメージは、RHEL インストーラーの起動に使用されます。インストーラーは、RHEL for Edge イメージをインストールするために提供されたキックスタートファイルを取得します。

前提条件

- アクティブな Red Hat サブスクリプションがある。
- Red Hat カスタマーポータルの [製品のダウンロード](#) セクションにログインしている。

手順

1. ブラウザーを開き、<https://access.redhat.com/downloads> にアクセスします。
2. Red Hat Enterprise Linux 8 を **クリック** します。
3. 「Red Hat Enterprise Linux 8.2 Boot ISO」オプションの場合は、**Download Now** のボタンをクリックします。

Boot.iso イメージのダウンロードの詳細は、[「ISO のインストールイメージのダウンロード」](#) を参照してください。

5.4. キックスタートファイルを使用した RHEL FOR EDGE イメージのインストール

キックスタートファイルを使用して RHEL for Edge イメージをインストールするには、Web サーバーを使用します。Web サーバーは、RHEL for Edge イメージの commit.tar リポジトリとキックスタートファイルを使用して、RHEL インストーラーを起動します。

前提条件

- インストーラーのコミットを取得するためのサーバーが利用可能であり、実行されている。
- 作成したコミットをインストールするためのディスクイメージ。

手順

1. libvirt virt-install を使用して Anaconda インストーラーを実行します。

```
virt-install --name rhel-edge-test-1 --memory 2048 --vcpus 2 --disk
path=prepared_disk_image.qcow2,format=qcow2 --os-variant rhel8.0 --cdrom
/home/username/Downloads/rhel8-edge.iso
```

2. インストール画面で **TAB** を押して、追加のカーネルパラメーターを指定します。

```
inst.ks=http://edge_device_ip:port/kickstart.ks
```

カーネルパラメーターは、RHEL インストーラーに含まれる RHEL イメージではなく、キックスタートファイルを使用して RHEL をインストールすることを指定します。

RHEL インストーラーが起動し、サーバー (HTTP) エンドポイントからキックスタートファイルを取得してコマンドを実行します。このコマンドには、HTTP エンドポイントから RHEL for Edge イメージコミットをインストールするコマンドが含まれます。インストールが完了すると、インストーラーはログインの詳細を要求します。

3. ログイン画面で、ユーザーアカウントの認証情報を指定し、**Enter** をクリックします。
4. RHEL for Edge イメージが正常にインストールされているかどうかを確認します。

```
$ rpm-ostree status
```

コマンドの出力には、イメージのコミット ID が表示され、正常にインストールされたことがわかります。

以下は出力例です。

```
State: idle
Deployments:
* ostree://edge:rhel/8/x86_64/edge
  Timestamp: 2020-09-18T20:06:54Z
  Commit: 836e637095554e0b634a0a48ea05c75280519dd6576a392635e6fa7d4d5e96
```

5.5. ネットワークベース以外のデプロイメント用の RHEL FOR EDGE コンテナイメージの作成

RHEL for Edge Container コミットをダウンロードした後に、ダウンロードしたコミットを Podman にロードして、コンテナを構築できます。これには、以下の手順を実行します。

前提条件

- システムに **Podman** をインストール済みである。「[How do I install podman in RHEL 8](#)」を参照してください。

手順

1. RHEL for Edge コンテナイメージをダウンロードしたディレクトリーに移動します。
2. RHEL for Edge コンテナイメージを **Podman** に読み込みます。

```
$ cat ./commit_ID-container.tar | sudo podman load
```

コマンド出力にはイメージ ID が提供されます (例:
@8e0d51f061ff1a51d157804362bc875b649b27f2ae1e66566a15e7e6530cec63)。

3. 前のステップで生成したイメージ ID を使用して、新しい RHEL for Edge コンテナイメージにタグ付けします。

```
$ sudo podman tag image-ID localhost/edge-container
```

podman tag コマンドは、ローカルイメージに別の名前を割り当てます。

4. **edge-container** という名前のコンテナを実行します。

```
$ sudo podman run --name=edge-container -p 8080:80 localhost/edge-container
```

podman run --name=edge-container コマンドは、localhost/edge-container イメージに基づいて、コンテナに名前を割り当てます。

5. コンテナを一覧表示します。

```
$ podman ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
2988198c4c4b ....localhost/edge-container /bin/bash 3 seconds ago Up 2 seconds ago
edge-container
```

その結果、**Podman** はコンテナを実行し、RHEL for Edge Container コミットのリポジトリーを提供します。

5.6. ネットワーク以外のデプロイメント用の RHEL FOR EDGE インストーラーイメージの作成

実行中のコンテナをビルドして、RHEL for Edge Container コミットでリポジトリーを提供してから、**RHEL for Edge Installer (.iso)** イメージを作成します。インストーラー (.iso) は、HTTP で RHEL for Edge Container (.tar) によるコミットをプルします。以下の例では、コンテナを使用して Web サーバーを設定する手順を説明します。

前提条件

- RHEL システムで、Image Builder ダッシュボードにアクセス済みである。
- RHEL for Edge イメージの空の Blueprint を作成済みである。「[RHEL for Edge イメージのダウンロード](#)」を参照してください。

手順

1. Image Builder のダッシュボードで、RHEL for Edge イメージ用に作成した空の Blueprint の **Create Image** を **クリック** します。
2. **Create Image** ウィンドウで、以下の手順を実行します。
 - a. **リポジトリ** テキストボックスで、イメージに埋め込むコミットの OSTree リポジトリへの URL を指定します。例: <http://10.0.2.2:8080/repository/>
 - b. **Ref** テキストボックスで、イメージに組み込む RHEL for Edge Container コミットの作成時に指定したものと同一参照を指定します。例: **rhel/edge/test**
 - c. **Create** をクリックします。

Image Builder は、イメージのビルド時に、実行中のコンテナが提供するコミットをプルします。イメージのビルドが完了したら、作成されたイメージをダウンロードできます。「[RHEL for Edge イメージのダウンロード](#)」を参照してください。

CLI に同じワークフローを使用できます。[ネットワーク以外のデプロイメント向けのコマンドラインインターフェースを使用した RHEL for Edge Installer イメージの作成](#) について参照してください。

5.7. ネットワークベース以外のデプロイメント向けの RHEL FOR EDGE イメージのインストール

RHEL for Edge イメージをインストールするには、以下の手順に従います。

前提条件

- 実行中のコンテナを停止している。
- 作成したコミットをインストールするためのディスクイメージ。

手順

1. 空の qcow イメージを作成して、(.iso) をインストールします。これは、仮想マシン用のハードドライブのイメージです。以下に例を示します。

```
$ qemu-img create -f qcow2 diskfile.qcow2 5G
```

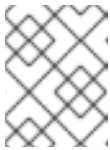
2. ディスクをドライブとして、インストーラー ISO を CD-ROM として使用している仮想マシンを起動します。以下に例を示します。

```
$ qemu-system-x86_64 \
  -enable-kvm \
  -m 4G \
  -cpu host \
  -net nic,model=virtio \
  -net user,hostfwd=tcp::2223-:22 \
  -cdrom $HOME/Downloads/Fedora-Server-netinst-x86_64-33-1.2.iso \
  disk-image.img
```

このコマンドにより、qemu (ハイパーバイザー) が以下のように設定されます。

- KVM 仮想化を使用します。

- メモリーを 4000MB に増やします (一部のプロセスは、dnf など、多くのメモリーを消費します)。
- ホストが使用する CPU タイプと同じタイプを使用します。
- ホストの仮想ネットワークブリッジに、ゲストをホスト上の仮想ネットワークブリッジに接続し、ホストからゲストの SSH ポート (22) に転送します。
- インストール ISO をマウントします。
- 上記で作成したハードドライブイメージを使用します。
Anaconda インストーラーを指定します。RHEL インストーラーが起動し、ISO からキックスタートファイルを読み込み、RHEL for Edge イメージのコミットのインストールなど、コマンドを実行します。インストールが完了すると、インストーラーはログインの詳細を要求します。



注記

Anaconda は、インストール時にコンテナコミットを使用するように事前設定されています。

3. ログイン画面で、ユーザーアカウントの認証情報を指定し、**Enter** をクリックします。

検証手順

1. RHEL for Edge イメージが正常にインストールされているかどうかを確認します。

```
$ rpm-ostree status
```

結果として、コマンドの出力にはイメージのコミット ID が表示され、正常にインストールされたことが分かります。

第6章 RHEL FOR EDGE イメージの管理

RHEL for Edge イメージを管理するために、以下のいずれかの管理タスクを実行することができます。

- RHEL Web コンソールで Image Builder を使用した RHEL for Edge イメージの Blueprint の編集
- Image Builder コマンドラインを使用した RHEL for Edge イメージの Blueprint の編集
- RHEL for Edge イメージの更新
- rpm-ostree リモートをノードに設定/ノードポリシーの更新
- RHEL for Edge イメージの手動での復元、または Greenboot を使用した自動的な復元

6.1. RHEL WEB コンソールで IMAGE BUILDER を使用した RHEL FOR EDGE イメージの BLUEPRINT の編集

RHEL for Edge イメージの Blueprint を編集して、以下を実行することができます。

- 必要に応じた追加コンポーネントの追加
- 既存コンポーネントのバージョンの変更
- 既存コンポーネントの削除

6.1.1. RHEL Web コンソールで Image Builder を使用した RHEL for Edge イメージの Blueprint へのコンポーネントの追加

RHEL for Edge イメージの Blueprint にコンポーネントを追加するには、以下の前提条件を満たしていることを確認してから、対応する Blueprint を編集する手順に従ってください。

前提条件

- RHEL システムで、Image Builder ダッシュボードにアクセス済みである。
- RHEL for Edge イメージの Blueprint を作成済みである。

手順

1. Image Builder のダッシュボードで、編集したい RHEL for Edge イメージの Blueprint をクリックします。
特定の Blueprint を検索するには、Filter by Name のテキストボックスに Blueprint 名を入力してから **Enter** キーを押します。
2. Blueprint の右上で、**Edit Packages** をクリックします。
画面が Edit Packages モードに切り替わります。
3. Filter by Name のテキストボックスに、追加するコンポーネント名を入力してから Enter キーを押します。
コンポーネント名の一覧が表示されます。
4. コンポーネントの横の **+** 記号をクリックします。
Blueprint にコンポーネントが追加されます。
5. **Commit** をクリックします。

Blueprint の更新が保存され、保留中のコミットと共にメッセージが表示されます。

6. サマリーダイアログボックスで、変更内容を確認してから **Commit** をクリックします。コミットが正常に実行されたことを確認するメッセージが表示されます。

これにより、新しいバージョンの Blueprint が作成され、右ペインに最新のコンポーネントが一覧表示されます。

6.1.2. RHEL Web コンソールを使用した RHEL for Edge イメージの Blueprint 内の既存コンポーネントのバージョンの変更

デフォルト (最新) のバージョンを選択したか、あるいは Blueprint に含めたコンポーネントのバージョンを選択しました。必要に応じて、任意のコンポーネントのバージョンを変更できるようになりました。

そのためには、以下の前提条件を満たしていることを確認してから、対応する Blueprint のコンポーネントのバージョンを変更する手順に従ってください。

前提条件

- RHEL システムで、Image Builder ダッシュボードにアクセス済みである。
- RHEL for Edge イメージの Blueprint を作成済みである。
- RHEL for Edge の Blueprint に1つ以上のコンポーネントを追加済みである。

手順

1. Image Builder のダッシュボードで、編集する Blueprint をクリックします。特定の Blueprint を検索するには、Filter by Name のテキストボックスに Blueprint 名を入力してから **Enter** キーを押します。
2. Blueprint の右上で、**Edit Packages** をクリックします。画面が Edit Packages モードに切り替わり、右側のパネルには、現在 Blueprint にコミットされているコンポーネント名が一覧表示されます。
3. コンポーネント名をクリックします。
4. Component Options Version のドロップダウンリストから希望のバージョンを選択します。
5. **Apply Changes** をクリックします。変更が保存され、右ペインには最新の変更が一覧表示されます。
6. **Commit** をクリックします。新しいバージョンは Blueprint に保存されています。保留中のコミットを示すメッセージが表示されます。
7. サマリーダイアログボックスで、変更内容を確認してから **Commit** をクリックします。コミットが正常に実行されたことを確認するメッセージが表示されます。

その結果、Blueprint の新しいバージョンが作成され、右側のペインには最新のコンポーネントが表示されます。

6.1.3. RHEL Web コンソールで Image Builder を使用した RHEL for Edge イメージの Blueprint からのコンポーネントの削除

作成した RHEL for Edge イメージの Blueprint から不要なコンポーネントを1つ以上削除するには、以下の前提条件を満たしていることを確認してから、手順に従ってください。

前提条件

- RHEL システムで、Image Builder ダッシュボードにアクセス済みである。
- RHEL for Edge イメージの Blueprint を作成済みである。
- RHEL for Edge の Blueprint に1つ以上のコンポーネントを追加済みである。

手順

1. Image Builder のダッシュボードで、編集する Blueprint をクリックします。
特定の Blueprint を検索するには、Filter by Name のテキストボックスに Blueprint 名を入力してから Enter キーを押します。
2. Blueprint の右上で、**Edit Packages** をクリックします。
画面が Edit Packages モードに切り替わります。右側のパネルには、現在 Blueprint にコミットされているコンポーネント名が一覧表示されます。
3. More Options メニューの **Remove** をクリックします。
オプションで、コンポーネント名をクリックしてから **Remove** をクリックします。
4. **Commit** をクリックします。
保留中のコミットを示すメッセージが表示されます。
5. 変更内容を確認してから、**Commit** をクリックします。
コミットが正常に実行されたことを確認するメッセージが表示されます。

これにより、新しいバージョンの Blueprint が作成され、右ペインに最新のコンポーネントが一覧表示されます。

6.2. コマンドラインインターフェースを使用した RHEL FOR EDGE イメージの BLUEPRINT の編集

Image Builder のコマンドラインを使用して、RHEL for Edge イメージの Blueprint の仕様を変更することができます。そのためには、以下の前提条件を満たしていることを確認してから、手順に従って対応する Blueprint を編集してください。

前提条件

- Image Builder のコマンドラインへアクセスできる。
- RHEL for Edge イメージの Blueprint を作成済みである。

手順

1. ローカルのテキストファイルに Blueprint を保存 (エクスポート) します。

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

2. 選択したテキストエディターで **BLUEPRINT-NAME.toml** ファイルを編集し、変更を加えます。

編集を終了する前に、ファイルが有効な Blueprint であることを確認してください。

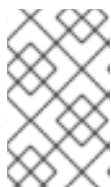
- バージョン番号を大きくしてください。
Semantic Versioning スキームを使用していることを確認してください。



注記

バージョンを変更しない場合、バージョンのパッチコンポーネントが自動的に増えます。

- コンテンツが有効な TOML 仕様かどうかを確認します。詳細は、TOML のドキュメントを参照してください。



注記

TOML のドキュメントはコミュニティが提供しているため、Red Hat のサポート対象外となります。このツールの問題は、<https://github.com/toml-lang/toml/issues> から報告できます。

- ファイルを保存してエディターを閉じます。
- Blueprint を Image Builder のコマンドラインにプッシュ (インポート) します。

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



注記

Blueprint を Image Builder のコマンドラインにプッシュしなおす場合は、**.toml** の拡張子を含めてファイル名を指定します。

- Image Builder にアップロードしたコンテンツが、編集したものと一致していることを確認します。

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

- Blueprint に一覧表示されているコンポーネントおよびバージョンと、その依存関係が有効かどうかを確認します。

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

6.3. RHEL FOR EDGE イメージの更新

6.3.1. RHEL for Edge イメージの更新のデプロイ方法

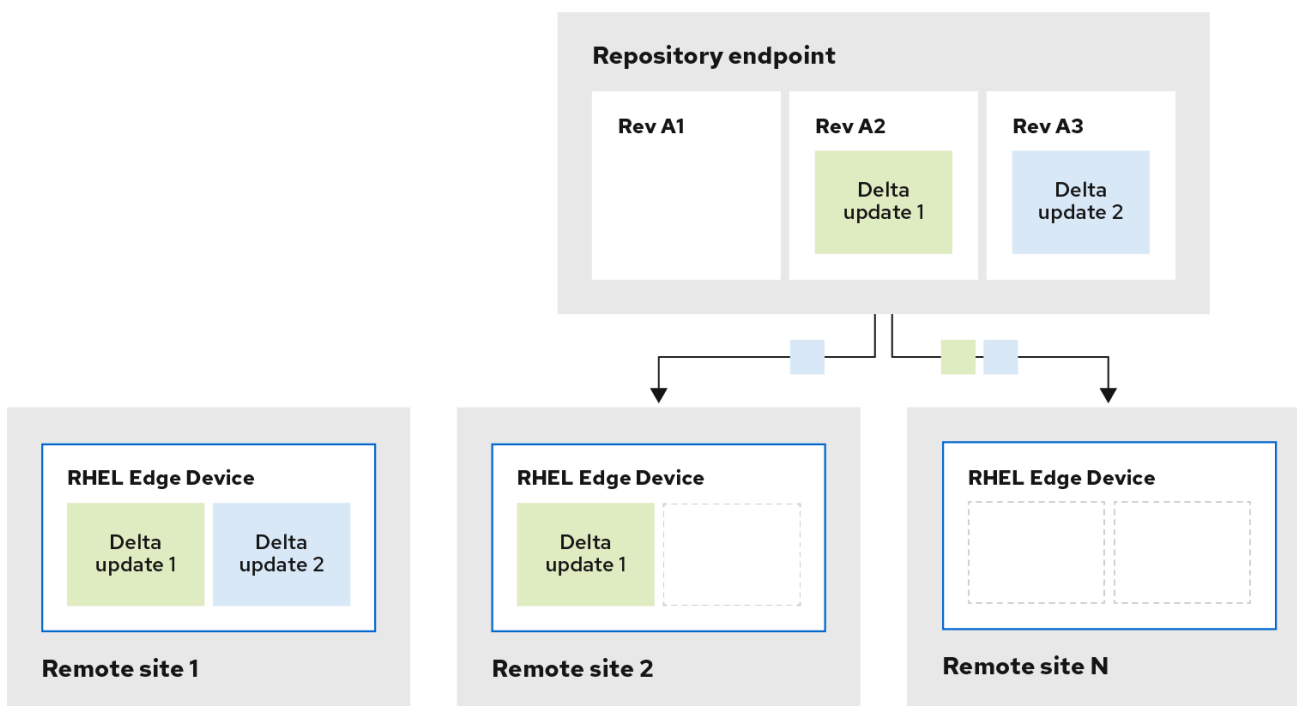
RHEL for Edge イメージでは、更新を手動でデプロイするか、デプロイメントプロセスを自動化することができます。更新はアトミックな方法で適用されます。つまり、各更新の状態が周知され、更新は段階的に行われ、再起動時にのみ適用されます。デバイスを再起動するまで変更を確認できないため、可能な限り最大限のアップタイムを確保するために再起動をスケジュールすることができます。

イメージの更新時には、更新された OS のコンテンツのみがネットワーク経由で転送されるため、イメージ全体を転送する場合と比較して、デプロイメントプロセスはより効率的になります。OS のバイ

ナリーおよびライブラリー (/usr) は読み込み専用で、読み書き状態は /var ディレクトリーと /etc ディレクトリーで管理されています。

デルタ転送を利用することで、断続的な接続および低帯域幅での接続の場合でも更新をデプロイでき、あるいは、切断されたデバイスにもローカルメディアを利用して更新をデプロイすることができます。さらに、static-delta を作成して、ネットワークの使用量をさらに減らすこともできます。static-delta は、個々の更新をすべて単一のファイルアーカイブにプルし、複数の接続を介した更新ごとの個別転送とは対照的に、単一の TCP 接続を介して OS の更新を転送する際の TCP ネットワークのオーバーヘッドを大幅に削減します。

以下の図は、RHEL for Edge イメージの更新デプロイメントプロセスを説明しています。



125_RHEL_1020

6.3.2. RHEL for Edge イメージの更新の手動でのデプロイ

RHEL for Edge の Blueprint を編集した後、イメージコミットを更新することができます。Image Builder は、更新された RHEL for Edge イメージの新しいコミットを生成します。この新しいコミットを使用して、最新のパッケージバージョンまたは追加パッケージのイメージをデプロイしてください。

RHEL for Edge イメージの更新をデプロイするには、前提条件を満たしていることを確認し、手順に従ってください。

前提条件

- RHEL システムで、Image Builder ダッシュボードにアクセス済みである。
- RHEL for Edge イメージの Blueprint を作成済みである。
- RHEL for Edge イメージの Blueprint を編集済みである。??? を参照してください。

手順

1. Image Builder のダッシュボードで、編集した Blueprint の **Create Image** をクリックします。

2. Create Image のウィンドウで、以下の手順を実行します。
 - a. **Type** のドロップダウンリストから、「RHEL for Edge Commit (.tar)」を選択します。
 - b. **Parent commit** テキストボックスで、以前に生成された親コミット ID を指定します。「[RHEL for Edge イメージのコミットの展開](#)」を参照してください。
 - c. Ref テキストボックスでは、コミットの名前を指定するか、空欄のままにしておくことができます。デフォルトでは、Web コンソールは Ref を **rhel/8/arch_name/edge** と指定します。
 - d. **Create** をクリックします。Image Builder は、更新された Blueprint 用に RHEL for Edge イメージを作成します。
RHEL for Edge イメージの作成の進捗状況を確認するには、ブレッドグラムから Blueprint 名をクリックし、次に **Images** タブをクリックします。



注記

イメージ作成プロセスの完了まで数分かかります。

作成したイメージには、追加した最新のパッケージがある場合は含まれ、親としてオリジナルの **commit ID** を持っています。

3. 作成した RHEL for Edge イメージをダウンロードします。RHEL for Edge イメージのダウンロードに関する詳細は、「[RHEL for Edge イメージのダウンロード](#)」を参照してください。
4. OSTree コミットを展開します。OSTree コミットの展開に関する詳細は、「[RHEL for Edge イメージのコミットの展開](#)」を参照してください。
5. 今回は子コミット ID を提供し、docker コンテナをビルドします。

```
# podman build -t <_name-of-server_> --build-arg commit=<uuid>-child_commit.tar .
```

6. コンテナを実行します。

```
# podman run --rm -p 8000:80 <_name-of-server_>
```

7. プロビジョニングされた RHEL システム上で、オリジナルの Edge イメージから、現在の状態を確認します。

```
$ rpm-ostree status
```

新しいコミット ID がない場合は、以下のコマンドを実行して、利用可能なアップグレードの有無を確認します。

```
$ rpm-ostree upgrade --check
```

コマンドの出力は、現在アクティブな OSTree コミット ID を提供します。

8. OSTree を更新して、新しい OSTree のコミット ID を利用できるようにします。

```
$ rpm-ostree upgrade
```


OSTree は、リポジトリに更新があるかどうかを確認します。ある場合は、更新を取得し、この新しいコミット更新のデプロイメントを有効化できるように、システムの再起動を要求します。

9. 再度、現在の状態を確認します。

```
$ rpm-ostree status
```

これで、2つのコミットが利用可能であることが確認できます。

- アクティブな親コミット。
- アクティブではなく、1つの違いを含む新しいコミット。

10. 新しいデプロイメントを有効にし、新しいコミットを有効にするには、システムを再起動します。

```
# systemctl reboot
```

Anaconda インストーラーは、新しいデプロイメントで再起動します。ログイン画面では、起動に利用可能な新しいデプロイメントが表示されます。

11. 最新のコミット/デプロイメントで起動すると、rpm-ostree upgrade コマンドは、新しいデプロイメントが一覧内で最初に表示されるようにブートエントリを自動的に順序を指定します。必要に応じて、キーボードの矢印キーを使用して GRUB メニューエントリを選択し、**Enter** を押します。
12. ログインユーザーのアカウント認証情報を提供してください。
13. OSTree の状態を確認します。

```
$ rpm-ostree status
```

コマンド出力は、アクティブなコミット ID を提供します。

14. 変更したパッケージが存在する場合は、親コミットと新しいコミットの間で diff を実行します。

```
$ rpm-ostree db diff parent_commit new_commit
```

更新により、インストールしたパッケージが利用可能になり、すぐに使用できる状態になっていることがわかります。

6.3.3. ネットワーク以外のデプロイメント向けの RHEL for Edge イメージ更新の手動デプロイ

RHEL for Edge の Blueprint を編集した後、イメージコミットを更新することができます。Image Builder は、更新された RHEL for Edge イメージの新しいコミットを生成します。この新しいコミットを使用して、最新のパッケージバージョンまたは追加パッケージでイメージをデプロイします。

RHEL for Edge イメージの更新をデプロイするには、前提条件を満たしていることを確認し、手順に従ってください。

前提条件

- RHEL for Edge システムが稼働している。
- OSTree リポジトリは HTTP 上で提供されます。
- RHEL for Edge イメージの Blueprint を作成済みである。
- RHEL for Edge イメージの Blueprint を編集済みである。??? を参照してください。

手順

1. Image Builder のダッシュボードで、編集した Blueprint の **Create Image** をクリックします。
2. **Create Image** ウィンドウで、以下の手順を実行します。
 - a. **Type** ドロップダウンリストから、**RHEL for Edge Container(.tar)** を選択します。
 - b. **Parent commit** テキストボックスで、以前に生成された親コミット ID を指定します。「[RHEL for Edge イメージのコミットの展開](#)」を参照してください。
 - c. **リポジトリ** テキストボックスで、イメージに埋め込むコミットの OSTree リポジトリへの URL を指定します。例: <http://10.0.2.2:8080/repository/>
 - d. **Ref** テキストボックスで、イメージに組み込む RHEL for Edge Container コミットの作成時に指定したものと同一参照を指定します。例: 'rhel/edge/test'。
 - e. **Create** をクリックします。Image Builder は、更新された Blueprint 用に RHEL for Edge イメージを作成します。
RHEL for Edge イメージの作成の進捗状況を確認するには、ブレイクダウンから blueprint name をクリックし、次に **Images** タブをクリックします。



注記

イメージ作成プロセスの完了まで数分かかります。

作成したイメージには、最新パッケージがあれば、そのパッケージが追加され、オリジナルの **commit ID** を親とします。

3. 作成した RHEL for Edge イメージをダウンロードします。RHEL for Edge イメージのダウンロードに関する詳細は、「[RHEL for Edge イメージのダウンロード](#)」を参照してください。
4. 今回は子コミット ID を提供して Podman に RHEL for Edge コンテナイメージを読み込みます。

```
$ cat ./child-commit_ID-container.tar | sudo podman load
```

5. **Podman** を実行します。

```
# sudo podman run -p 8080:80 localhost/edge-test
```

6. プロビジョニングされた RHEL システム上で、オリジナルの Edge イメージから、現在の状態を確認します。

```
$ rpm-ostree status
```

新しいバージョンの RHEL for Edge イメージが提供された場合、この手順を繰り返して、最新のイメージを使用する必要があります。

新しいコミット ID がない場合は、以下のコマンドを実行して、利用可能なアップグレードの有無を確認します。

```
$ rpm-ostree upgrade --check
```

利用可能な更新がある場合は、コマンドの出力は、現在アクティブな OSTree コミット ID など、OSTree リポジトリで利用可能な更新に関する情報を提供します。その他に、利用可能な更新がないことをメッセージに通知するよう求められます。

- OSTree を更新して、新しい OSTree のコミット ID を利用できるようにします。

```
$ rpm-ostree upgrade
```

OSTree は、リポジトリに更新があるかどうかを確認します。ある場合は、更新を取得し、この新しいコミット更新のデプロイメントを有効化できるように、システムの再起動を要求します。

- 現在の状態を確認します。

```
$ rpm-ostree status
```

これで、2つのコミットが利用可能であることが確認できます。

- アクティブな親コミット。
- アクティブではなく、1つの違いを含む新しいコミット。

- 新しいデプロイメントを有効にし、新しいコミットを有効にするには、システムを再起動します。

```
# systemctl reboot
```

Anaconda インストーラーは、新しいデプロイメントで再起動します。ログイン画面では、起動に利用可能な新しいデプロイメントが表示されます。

- 最新のコミット/デプロイメントで起動すると、**rpm-ostree upgrade** コマンドは、新しいデプロイメントが一覧内で最初に表示されるようにブートエントリを自動的に順序を指定します。必要に応じて、キーボードの矢印キーを使用して GRUB メニューエントリを選択し、**Enter** を押します。

- ログインユーザーのアカウント認証情報を提供してください。

- OSTree の状態を確認します。

```
$ rpm-ostree status
```

コマンド出力は、アクティブなコミット ID を提供します。

- 変更したパッケージが存在する場合は、親コミットと新しいコミットの間で diff を実行します。

```
$ rpm-ostree db diff parent_commit new_commit
```

更新により、インストールしたパッケージが利用可能になり、すぐに使用できる状態になっていることがわかります。

6.3.4. RHEL for Edge の自動イメージ更新のデプロイ

Edge デバイスに RHEL for Edge イメージをインストールした後、利用可能なイメージ更新がある場合は確認し、自動適用することができます。

`rpm-ostreed-automatic.service` (systemd service) および `rpm-ostreed-automatic.timer` (systemd timer) は、チェックとアップグレードの頻度を制御します。利用可能な更新がある場合は、ステージングされたデプロイメントとして表示されます。

イメージの自動更新をデプロイするには、以下の高レベルの手順を実施します。

- イメージ更新ポリシーの更新
- 更新の自動ダウンロードとステージングの有効化

6.3.5. RHEL for Edge イメージの更新ポリシーの更新

イメージ更新ポリシーを更新するには、Edge デバイスの `/etc/rpm-ostreed.conf` ロケーションにある `rpm-ostreed.conf` ファイルから「**AutomaticUpdatePolicy**」と「**IdleExitTimeout**」の設定を使用します。

Automatic Evaluator 設定では、自動更新ポリシーを制御し、以下のオプションを持ちます。

- `none`: 自動更新を無効にします。デフォルトでは、「**AutomaticUpdatePolicy**」設定は「**none**」に設定されています。
- `check`: `rpm-ostree` 状態で利用可能な更新を表示するために十分なメタデータをダウンロードします。
- `stage`: 再起動時に適用される更新をダウンロードして展開します。

The **IdleExitTimeout** 設定は、デーモンが終了するまでの非アクティブ時間（秒単位）を制御し、以下のオプションを持ちます。

- `0`: 自動終了を無効にします。
- `60`: デフォルトでは、**IdleExitTimeout** 設定は **60** に設定されます。

自動更新を有効にするには、以下の手順を実行します。

手順

1. `/etc/rpm-ostreed.conf` ファイルで、以下を更新します。
 - **AutomaticUpdatePolicy** の値を変更して **チェック** します。
 - 更新チェックを実行するには、**IdleExitTimeout** の値を秒単位で指定します。
2. **rpm-ostreed** サービスを再読み込みし、**systemd** タイマーを有効にします。

```
# systemctl reload rpm-ostreed
# systemctl enable rpm-ostreed-automatic.timer --now
```

3. `rpm-ostree` 状態を確認し、自動更新ポリシーが設定され、時間がアクティブであることを確認します。

```
# rpm-ostree status
```

コマンド出力は以下を表示します。

```
State: idle; auto updates enabled (check; last run <minutes> ago)
```

さらに、出力には利用可能な更新情報も表示されます。

6.3.6. RHEL for Edge の更新の自動ダウンロードとステージングの有効化

イメージ更新ポリシーを更新してイメージの更新を確認すると、更新がある場合は更新の詳細と一緒に表示されます。更新の適用を決定した場合は、ポリシーを有効にして、更新を自動的にダウンロードしてステージします。続いて、利用可能なイメージの更新がダウンロードされ、デプロイメントのためにステージングされます。更新は、Edge デバイスを再起動すると適用され、有効になります。

更新の自動ダウンロードとステージングのポリシーを有効にするには、以下の更新を実行します。

手順

1. `/etc/rpm-ostreed.conf` ファイルで、「AutomaticUpdatePolicy」を **stage** に更新します。
2. rpm-ostreed サービスを再読み込みします。

```
# systemctl enable rpm-ostreed-automatic.timer --now
```

3. rpm-ostree の状態を確認します。

```
# rpm-ostree status
```

コマンド出力は以下を表示します。

```
State: idle
AutomaticUpdates: stage; rpm-ostreed-automatic.timer: last run <time> ago
```

4. 更新を開始するには、タイマーによる更新の開始を待つか、手動でサービスを開始することができます。

```
# systemctl start rpm-ostreed-automatic.service
```

更新が開始すると、rpm-ostree の状態は以下のようになります。

```
# rpm-ostree status
State: busy
AutomaticUpdates: stage; rpm-ostreed-automatic.service: running
Transaction: automatic (stage)
```

更新が完了すると、デプロイメントの一覧に新しいデプロイメントがステージングされ、オリジナルの起動したデプロイメントはそのままになります。新しいデプロイメントを使用してシステムを起動するか、次の更新を待つかを定めることができます。

デプロイメントの一覧を表示するには、**rpm-ostree status** コマンドを実行してください。

以下は出力例です。

```
# rpm-ostree status
State: idle
AutomaticUpdates: stage; rpm-ostreed-automatic.timer: last run <time> ago
Deployments:
```

更新されたパッケージの詳細を含むデプロイメントの一覧を表示するには、**rpm-ostree status -v** コマンドを実行してください。

6.4. RHEL FOR EDGE イメージのロールバック

更新されたイメージが正常にデプロイされたかどうかを確認することができます。デプロイが失敗した場合は、以前のバージョンにロールバックすることができます (コミット)。以前の機能状態にロールバックするには、手動で手順を実行するか、自動化されたプロセスを使用することができます。

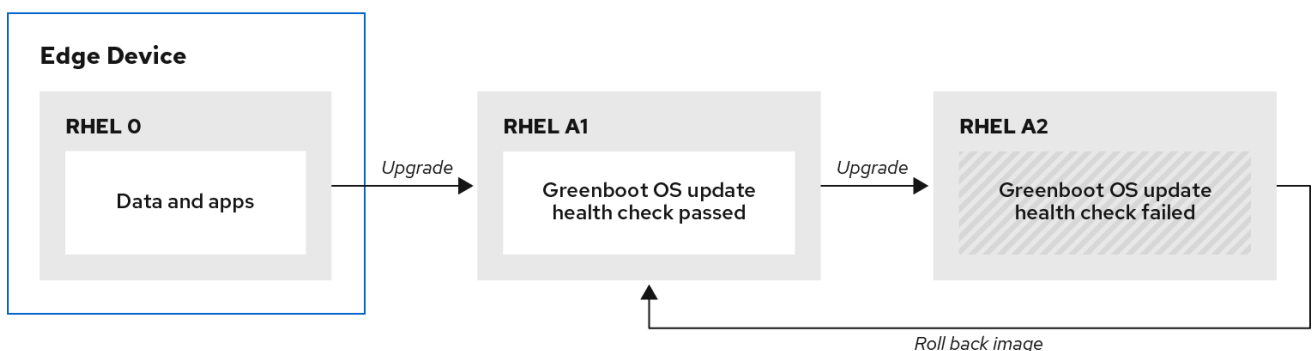
6.4.1. RHEL for Edge イメージのロールバック方法

RHEL for Edge イメージでは、トランザクション更新のみがオペレーティングシステムに適用されます。トランザクション更新では、失敗した更新を最後の既知の良い状態に簡単にロールバックすることができます、更新中のシステム障害を防ぐことができます。

Greenboot でインテリジェントなロールバックを使用して、アプリケーションの安定性を選択するか、またはセキュリティー更新のあるアプリケーションを選択するかという問題をなくすことができます。

Greenboot は rpm-ostree を活用し、システム起動時に実行されるカスタムヘルスチェックを実行します。問題が発生した場合、システムは変更をロールバックし、最後の作業状態を保持します。

以下の図は、RHEL for Edge イメージのロールバックプロセスを説明しています。



125_RHEL_1020

6.4.2. RHEL for Edge イメージの手動でのロールバック

RHEL for Edge イメージの更新のためのデプロイメントが失敗した場合、または更新が正常に動作しない場合は、手動で以前のバージョンのデプロイメントにロールバックすることができます。

前のバージョンにロールバックするには、以下の手順を実行します。

手順

1. **rollback** コマンドを実行します。

```
# rpm-ostree rollback
```

コマンド出力では、移動しているコミット ID の詳細と、削除されたパッケージの詳細を含む完全なトランザクションが表示されます。

2. システムを再起動します。

```
# systemctl reboot
```

このコマンドは、安定したコンテンツの前のコミットを有効にします。変更が適用され、以前のバージョンが復元されます。

6.4.3. 自動化プロセスを使用した RHEL for Edge イメージのロールバック

Greenboot チェックは、起動プロセスに統合されたフレームワークを提供し、ヘルスチェックが失敗した場合は **rpm-ostree** ロールバックを起動できます。ヘルスチェックについては、ヘルスチェックの可否を示すカスタムスクリプトを作成できます。結果をもとに、ロールバックをトリガーするタイミングを決めることができます。

ヘルスチェックスクリプトを作成するには、以下の手順を実行します。

手順

1. 標準の終了コード **0** を返すスクリプトを作成します。
たとえば、以下のスクリプトにより、構成された DNS サーバーを必ず利用できます。

```
#!/bin/bash

DNS_SERVER=$(grep ^nameserver /etc/resolv.conf | head -n 1 | cut -f2 -d" ")
COUNT=0
# check DNS server is available
ping -c1 $DNS_SERVER
while [ $? != '0' ] && [ $COUNT -lt 10 ]; do
((COUNT++))
echo "Checking for DNS: Attempt $COUNT ."
sleep 10
ping -c 1 $DNS_SERVER
done
```

2. `/etc/greenboot/check/required.d/` でヘルスチェック用の実行ファイルを追加します。

```
chmod +x check-dns.sh
```

次の再起動時には、システムが `boot-complete.target` に入る前に、スクリプトが起動プロセスの一環として実行されます。ヘルスチェックが正常に行われた場合は何もしません。ヘルスチェックに失敗した場合は、システムが数回再起動されてから、更新が失敗したとマークされ、前回の更新にロールバックされます。

検証手順

デフォルトゲートウェイにアクセスできるかどうかを確認するには、以下のヘルスチェックスクリプトを実行します。

1. 標準の終了コード **0** を返すスクリプトを作成します。

```
#!/bin/bash
```

```
DEF_GW=$(ip r | awk '/^default/ {print $3}')
SCRIPT=$(basename $0)

count=10
connected=0
ping_timeout=5
interval=5

while [ $count -gt 0 -a $connected -eq 0 ]; do
  echo "$SCRIPT: Pinging default gateway $DEF_GW"
  ping -c 1 -q -W $ping_timeout $DEF_GW > /dev/null 2>&1 && connected=1 || sleep
  $interval
  ((--count))
done

if [ $connected -eq 1 ]; then
  echo "$SCRIPT: Default gateway $DEF_GW is reachable."
  exit 0
else
  echo "$SCRIPT: Failed to ping default gateway $DEF_GW!" 1>&2
  exit 1
fi
```

2. `/etc/greenboot/check/required.d/` ディレクトリーでヘルスチェックの実行ファイルを追加します。

```
chmod +x check-gw.sh
```


付録A 用語とコマンド

本セクションでは、rpm-ostree の用語およびコマンドについて説明します。

A.1. OSTREE と RPM-OSTREE の用語

以下は、OSTree および rpm-ostree のイメージで使用される便利な用語の一部になります。

表A.1 OSTree と rpm-ostree の用語

用語	定義
OSTree	Linux ベースのオペレーティングシステムのバージョンを管理するために使用するツールです。OSTree のツリービューは Git に似ていて、同様のコンセプトに基づいています。
rpm-ostree	オペレーティングシステムの更新をホストするハイブリッドイメージまたはシステムパッケージ。
Commit	オペレーティングシステムのリリースバージョンまたはイメージバージョン。Image Builder は、RHEL for Edge イメージの ostree コミットを生成します。これらのイメージを使用して、Edge サーバーに RHEL をインストールしたり、更新したりすることができます。
Refs	ostree 内のブランチを表します。Refs は常に最新のコミットに解決します。(例: rhel/8/x86_64/edge)。
Revision (Rev)	特定のコミットの SHA-256。
Remote	ostree コンテンツをホストする http または https エンドポイント。これは yum リポジトリの baseurl に似ています。
static-delta	ostree イメージへの更新は常にデルタ更新です。RHEL for Edge イメージの場合、更新されるファイル数により、TCP のオーバーヘッドが予想以上に大きくなる可能性があります。TCP のオーバーヘッドを回避するために、特定のコミット間で static-delta を生成し、単一の接続で更新を送信することができます。この最適化は、接続性に制約のある大規模なデプロイメントをサポートします。

A.2. OSTREE コマンド

本セクションでは、ostree イメージのインストールまたは管理時に使用可能な ostree コマンドをいくつか紹介します。

表A.2 OSTree コマンド

ostree pull	<pre>ostree pull-local --repo [path] src</pre> <pre>ostree pull-local <path> <rev> --repo=<repo-path></pre> <pre>ostree pull <URL> <rev> --repo=<repo-path></pre>
ostree summary	<pre>ostree summary -u --repo=<repo-path></pre>
参照先の表示	<pre>ostree refs --repo ~/Code/src/osbuild-iot/build/repo/ --list</pre>
リポジトリでのコミットの表示	<pre>ostree log --repo=/home/gicmo/Code/src/osbuild-iot/build/repo/ <REV></pre>
コミットの検査	<pre>ostree show --repo build/repo <REV></pre>
リポジトリのリモートの一覧表示	<pre>ostree remote list --repo <repo-path></pre>
REV の解決	<pre>ostree rev-parse --repo ~/Code/src/osbuild-iot/build/repo fedora/x86_64/osbuild-demo</pre> <pre>ostree rev-parse --repo ~/Code/src/osbuild-iot/build/repo b3a008eceeddd0cfd</pre>
static-delta の作成	<pre>ostree static-delta generate --repo=[path] --from=REV --to=REV</pre>
GPG 鍵を使用した 既存 の ostree コミットの署名	<pre>ostree gpg-sign --repo=<repo-path> --gpg-homedir <gpg_home> COMMIT KEY-ID...</pre>

A.3. RPM-OSTREE コマンド

本セクションでは、ostree イメージのインストールまたは管理時に使用可能な **rpm-ostree** コマンドをいくつか紹介します。

表A.3 rpm-ostree コマンド

コマンド	説明
<pre>rpm-ostree --repo=/home/gicmo/Code/src/osbuild-iot/build/repo/ db list <REV></pre>	このコマンドは、リポジトリへの <REV> コミットに存在するパッケージを一覧表示します。

コマンド	説明
rpm-ostree rollback	OSTree は、 デプロイメント と呼ばれるブートローダーエントリーの順序リストを管理します。インデックス 0 のエントリーは、デフォルトのブートローダーエントリーです。エントリーごとに個別の /etc ディレクトリーがありますが、エントリーはすべて 1 つの /var ディレクトリーを共有します。Tab を押して起動を中断し、ブートローダーを使用してエントリーを選択できます。これにより、以前の状態にロールバックされます。つまり、デフォルトのデプロイメントはデフォルト以外のデプロイメントに変更されます。
rpm-ostree status	このコマンドは、使用中の現在のデプロイメントに関する情報を提供します。起動時にリストで一番上にあるデプロイメントがデフォルトになるなど、利用可能な全デプロイメントの名前と refspec が順番に記載されます。* のマークが付いたデプロイメントは現在の起動されるデプロイメントを、「r」のマークが付いたものは最新のアップグレードを示します。
rpm-ostree db list	このコマンドを使用して、コミットされたパッケージを確認します。1 つ以上のコミットを指定する必要がありますが、範囲や複数指定したコミットも機能します。
rpm-ostree db diff	このコマンドを使用して、2 つの rev (リビジョン) のツリー間でパッケージがどのように異なるかを表示します。rev を指定しないと、起動したコミットが保留中のコミットと比較されます。rev を 1 つだけ指定すると、起動したコミットはその rev と比較されます。
rpm-ostree upgrade	このコマンドは、現在のツリーの最新バージョンをダウンロードしてデプロイし、現在のツリーを次回の起動のデフォルトとして設定します。これは、実行中のファイルシステムツリーには影響はありません。変更を適用するには、再起動する必要があります。

関連情報

- 詳細は、**rpm-ostree** の man ページを参照してください。