



# Red Hat Enterprise Linux Atomic Host 7

## インストールと設定ガイド

インストールと設定ガイド



# Red Hat Enterprise Linux Atomic Host 7 インストールと設定ガイド

---

## インストールと設定ガイド

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 © 2023 | You need to change the HOLDER entity in the en-US/Installation\_and\_Configuration\_Guide.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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js<sup>®</sup> 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<sup>®</sup> 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

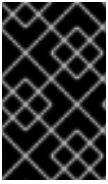
各種方法を使用したインストール、アップグレードまたはダウングレード、サンプルアプリケーションの作成

## 目次

1. 非推奨の通知 .....	3
<b>第1章 ATOMIC HOST の概要 .....</b>	<b>4</b>
1.1. オペレーティングシステムの内容	4
1.2. システム要件	4
<b>第2章 インストールの種類 .....</b>	<b>6</b>
2.1. 開発者モード	6
2.2. 物理マシンのインストール	6
2.2.1. 手動パーティション設定	6
2.2.2. Anaconda のインストール	7
2.2.3. キックスタートを使ったインストール	7
2.3. 仮想マシンのインストール	9
2.3.1. qcow2 メディアを使用した Linux ハイパーバイザーのインストール	9
2.3.1.1. インストールの準備	10
2.3.1.2. Red Hat Enterprise Linux Atomic Host の初回起動	11
2.3.2. Red Hat Enterprise Virtualization 環境のインストール	11
2.3.2.1. .ova ファイルからの Red Hat Enterprise Linux Atomic Host のインストール	12
2.3.2.2. ISO イメージからの Red Hat Enterprise Linux Atomic Host のインストール	15
2.3.3. Red Hat Enterprise Linux OpenStack Platform Installation	17
2.3.4. VMware のインストール	19
2.3.4.1. VMware での Red Hat Enterprise Linux Atomic Host 仮想マシンの設定*	21
2.3.5. Microsoft Hyper-V Installation	23
2.3.6. Microsoft Azure インストール	25
2.3.7. Google Compute Engine のインストール	31
2.3.7.1. Google Compute Engine の有効化	31
2.3.7.2. Red Hat Enterprise Linux Atomic Host インスタンスの起動	32
2.3.7.3. Red Hat Enterprise Linux Atomic Host インスタンスへのログイン	34
2.3.7.4. Red Hat Enterprise Linux Atomic Host インスタンスのモニターリング	36
2.3.7.5. ファイアウォールルールの作成	37
2.3.7.6. Red Hat Enterprise Linux Atomic Host インスタンスの削除	38
2.3.8. Amazon Web Services のインストール	38
2.4. PXE インストール	41
<b>第3章 CLOUD-INIT の設定 .....</b>	<b>42</b>
<b>第4章 インストール後の設定 .....</b>	<b>48</b>
4.1. ネットワークの設定	48
4.2. RHEL Atomic Host の登録	48
4.3. ユーザーアカウントの管理	49
<b>第5章 アップグレードおよびダウングレード .....</b>	<b>50</b>
5.1. Atomic Compose サーバーの設定	50
5.2. 新規バージョンへのアップグレード	51
5.3. 以前のバージョンへのロールバック	51
5.4. クライアントでの initramfs イメージの生成	52
<b>第6章 ATOMIC HOST の管理 .....</b>	<b>53</b>
6.1. Atomic Host	53
6.2. パッケージの階層化	54
6.2.1. RHEL Atomic Host への新規 RPM パッケージのインストール	54
6.2.2. 後でインストールするための RPM のダウンロードおよびキャッシュ	58
6.2.3. リポジトリメタデータの更新	58

6.2.4. 既存の RPM パッケージの上書き	58
6.3. "ostree admin unlock"	59
6.4. System Containers and Super-Privileged Containers (SPC)	59

## 1. 非推奨の通知



### 重要

2020年8月6日の時点で、Red Hat Enterprise Linux Atomic Host は廃止され、アクティブなサポートは提供されなくなりました。したがって、本書は非推奨となり、更新を受信しなくなります。

## 第1章 ATOMIC HOST の概要

### 1.1. オペレーティングシステムの内容

Red Hat Enterprise Linux Atomic Host は、Linux コンテナの実行に最適化された Red Hat Enterprise Linux 7 の様々なタイプです。軽量で効率的なオペレーティングシステムとして構築され、クラウド環境用のコンテナランタイムシステムとして使用することが特に最適になりました。RHEL Atomic Host には、事前にインストール済みのコンテナを実行するための多くのツールが含まれています(**docker**、**atomic**、**etcd**、**flannel**)。オールインワンの **kubernetes** インストールは引き続きサポートされますが、Red Hat は Kubernetes クラスタをサポートしなくなりました。

Red Hat Enterprise Linux Atomic Host はオープンソースツールである rpm-OSTree を使用して、RPM コンテンツで設定される起動可能で、変更不能で、バージョン管理されたファイルシステムツリーを管理します。これらのツリーは、パッケージから Red Hat によって設定され、**rpm-ostree** ツールはアトミックに複製します。これにより、アトミック更新に関するアップグレードおよびメンテナーストラテジーが作成されます。**Yum** の代わりに **rpm-ostree** を使用してソフトウェアをアップグレードおよび維持すると、Red Hat Enterprise Linux Atomic Host は他の Red Hat Enterprise Linux 7 バリエーションとは異なる方法で管理されます。

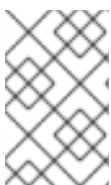
具体的には、Red Hat Enterprise Linux Atomic Host を使用する場合、オペレーティングシステムのコンテンツは読み取り専用モードでマウントされます。ローカルシステム設定には **/etc/** と **/var/** の2つの書き込み可能なディレクトリーのみがあります。更新は以下の方法で行われます。新規の起動可能なファイルシステムツリーが生成されます。これは、現在のファイルシステムツリーとストレージを共有します。新しいシステムツリーをダウンロードすると、古いツリーは並行して保持されます。つまり、ファイルシステムツリーの最初のアップグレード前のバージョンは、必要に応じてアトミックに復元できます。

コンテナやデータなど、アップグレード後も維持されるユーザーファイルは **/var/** ディレクトリーに置く必要があります。オペレーティングシステム自体は **/usr/** ディレクトリーに保存され、読み取り専用です。**ls -l** コマンドを使用して、**root** ディレクトリーで長いファイル一覧を実行すると、従来のルートレベルのディレクトリーの多くは、このいずれかの場所へのシンボリックリンクであることが分かります。たとえば、**/home/** ディレクトリーは **/var/home/** ディレクトリーへのシンボリックリンクです。したがって、このディレクトリーはアップグレード後も維持されます。

Red Hat Enterprise Linux Atomic Host 7.4.2 以降では、**/var** ディレクトリーをマウントポイントとして設定できます。これにより、**/var** を別のパーティションに配置できます。これにより、**/var** が満杯になると他のマウントポイントが満杯にならないようにします。

デフォルトのパーティション設定では、Red Hat Enterprise Linux にあるように、デフォルトのループバックではなく、ストレージバックエンドとして直接 LVM を使用して、コンテナで利用可能なほとんどの領域が示されます。ストレージは、インストール時に論理ボリュームを2つ作成する **docker-storage-setup** デモン、ファイルシステムコンテンツの **root**、およびイメージとコンテナの **docker-pool** により管理されます。

Red Hat Enterprise Linux Atomic Host は SELinux を使用してマルチテナント環境で強力な保護手段を提供します。iptables サービスはファイアウォールとして利用できます。iptables はデフォルトでオフになっています。



#### 注記

一部の RHEL Atomic Host バージョンでは、**docker** または **docker-latest** のいずれかを実行できます。ただし、Red Hat では、同じマシンで **docker** と **docker-latest** の両方を同時に実行することはサポートしていません。

### 1.2. システム要件



---

Red Hat Enterprise Linux Atomic Host は、過去 2 年以内に出荷されたほとんどのハードウェアと互換性がある必要があります。ハードウェアの互換性は、古いシステムをお使いの場合やシステムをカスタマイズした場合にとりわけ重要になります。ハードウェアの仕様はほぼ毎日変更されるため、すべてのシステムの互換性を確認することが推奨されます。サポートされるハードウェアの最新一覧は、[Red Hat Hardware Compatibility List](#) を参照してください。システム要件の一般的な情報は、[Red Hat Enterprise Linux テクノロジーの機能と制限](#) も参照してください。

Red Hat Enterprise Linux Atomic Host には、Red Hat Enterprise Linux と同じランタイム要件があります。ただし、ベアメタルまたは仮想環境での Anaconda ベースのインストール（対話型またはキックスタート）および PXE インストールには、最低 2GB のメモリーが必要です。

## 第2章 インストールの種類

Red Hat Enterprise Linux Atomic Host は複数の形式で配布されており、ベアメタル、複数の仮想環境、パブリックおよびプライベートクラウドインフラストラクチャーにインストールできます。

インストールメディアの **ダウンロード** ボタンをクリックすると、[Red Hat Enterprise Linux Atomic Host 製品ページ](#) で **インストールメディアを見つけることができます**。完全なインストール手順は [Red Hat Enterprise Linux インストールガイド](#) を参照してください。

### 注記

RHEL Atomic Host のすべてのバージョンで `.iso` イメージが利用できる訳ではありません。たとえば、`rhel-atomic-installer-7.3.3-1.x86_64.iso` は Atomic Host 7.3.3 で利用できますが、バージョン 7.3.4 から 7.3.6 には `.iso` は利用できません。

最新の Atomic Host を `.iso` からインストールするには、以下を実行します。

1. 利用可能な最新の `.iso` をダウンロードします。
2. インストールします。
3. 登録します。
4. サブスクリプションを割り当てます。
5. 実行コマンド

```
# atomic host upgrade
```

### 2.1. 開発者モード

Developer Mode では、実際にはインストールを行わずに Atomic Host を試すことができます。クラウドイメージの GRUB ブートメニュー（ベアメタル ISO にはない）でオプションとして利用でき、これにより **メタデータファイルおよびユーザーデータ ファイルのセットアップ** や `cloud-init` の設定も回避できます。

Atomic Host マシンが起動したら、GRUB ブートメニューで Developer Mode を選択して Developer Mode に入ります。

開発者モードは、`cloud-init` に以下を自動的に提供するローカルデータソースを提供します。

- 無作為に生成される root パスワード
- root アカウントの `tmux` セッションへの自動ログイン
- `rhel7/cockpit-ws` コンテナのプルおよび起動

### 2.2. 物理マシンのインストール

#### 2.2.1. 手動パーティション設定

RHEL Atomic Host と RHEL の物理マシンのインストールは通常類似していますが、いくつかの重要な違いがあります。このような違いの1つは、使用できるカスタムパーティション設定スキームです。

RHEL Atomic Host では、RHEL とは異なり、`/var` ディレクトリは書き込み可能な唯一のディレクトリ

リーです（小さな `/etc` ディレクトリーを除く）。そのため、`root` ディレクトリーのほとんどの書き込み可能なサブディレクトリーは実際には `/var` に保存されます。これにより、通常 `/var` が一番大きいディレクトリーになります。したがって、`/var` をマウントポイントとして設定する必要がある場合があります。これにより、`/var` を別のパーティションに配置できます。これにより、`/var` が満杯になると、他のマウントポイントが満杯にならないようにします。

RHEL Atomic Host 7.4.2 以降では、これを実行できます。手動パーティション設定を行う場合は、以下の点を考慮してください。

- コンテナとそのデータは `/var` に保存されます。システムコンテナは通常 `/ostree` にプルされ、`/var` にハードリンクされますが、`/var` が別のパーティションにある場合は、システムコンテナが `/var` のみにプルされます。つまり、`/var` が大きいことを意味します。十分な大きさのパーティションを専用にご覧ください。
- ストレージの場合は、LVM シンプルと `devicemapper` (RHEL Atomic Host のデフォルト) を使用する場合は、シンプル論理ボリュームを作成して使用できるように、ボリュームグループに十分な領域を確保してください。この手順については、[How to Leave Space in the Volume Group Backing Root During Installation](#) を参照してください。
- 追加の予防措置としては、さらに高度なスキームを使用して、`/var` のサブディレクトリーが異なるパーティションに配置することもできます。以下に例を示します。
  - `/var/lib/docker/`: `docker` または `cri-o` コンテナのイメージ用（通常は大きい領域）
  - `/var/lib/containers/atomic/`: システムコンテナおよびイメージ用
  - `/var/lib/docker/volumes/` - 実行中のコンテナからのデータ用

## 2.2.2. Anaconda のインストール

Anaconda を使用した RHEL Atomic Host のインストール手順は、Red Hat Enterprise Linux [インストールガイドの Anaconda を使用したインストール](#) の章を参照してください。

RHEL Atomic Host と RHEL のインストールにおける重要な違いは、どのカスタムパーティション設定スキームが利用できるかです。通常、RHEL Atomic Host は少ないパーティション設定スキームをサポートします。RHEL Atomic Host 7.4.2 以降、`/var` ディレクトリーをマウントポイントとして設定できます。これにより、`/var` を別のパーティションに配置できます。これにより、`/var` が満杯になると他のマウントポイントが満杯にならないようにします。[手動パーティション設定の手順は、Red Hat Enterprise Linux インストールガイドの手動パーティション設定](#) セクションを参照してください。

## 2.2.3. キックスタートを使ったインストール

キックスタートを使ったインストールの準備は、Red Hat Enterprise Linux インストールガイドの [キックスタートを使ったインストール](#) の章の説明に従ってください。Red Hat Enterprise Linux Atomic Host のキックスタートインストールは、特定の考慮事項を除き、Red Hat Enterprise Linux インストールとはあまり異なります。

Red Hat Enterprise Linux Atomic Host は、パッケージの管理および更新に `rpm-ostree` テクノロジーを使用します。したがって、キックスタートファイルでは `%packages` セクションは使用されません。代わりに、ファイルにはインストールメディアから `interactive-defaults.ks` ファイルを追加するためのコマンドが含まれている必要があります。このファイルには、メディア上の OSTree リポジトリを参照するキックスタートコマンドが含まれ、`cloud-init` サービスも無効にされます。

以下は、参照として使用できる Atomic Host のキックスタートファイルの例です。

```

lang en_US.UTF-8
keyboard us
timezone America/Chicago
#rootpw --iscrypted password_hash
rootpw --plaintext atomic
auth --enablesshadow --passalgo=sha512
ostreesetup --nogpg --osname=rhel-atomic-host --remote=rhel-atomic-host --url=file:///install/ostree --
ref=rhel-atomic-host/7/x86_64/standard
services --disabled cloud-init,cloud-config,cloud-final,cloud-init-local
clearpart --all --initlabel
zerombr
autopart
#%include /usr/share/anaconda/interactive-defaults.ks
%post --erroronfail
fn=/etc/ostree/remotes.d/rhel-atomic-host.conf; if test -f ${fn} && grep -q -e '^url=file:///install/ostree'
${fn}$; then rm ${fn}; fi
%end
%post --erroronfail
rm -f /etc/ostree/remotes.d/*.conf
echo 'unconfigured-state=This system is not registered to Red Hat Subscription Management. You
can use subscription-manager to register.' >> $(ostree admin --print-current-dir).origin
%end"

```

以下は、キックスタートファイル内のコマンドです。

- **rootpw** コマンドは、以下のプレーンテキスト引数を使用して root ユーザーのパスワードを設定するようにインストーラーに指示します（この場合は、パスワードは **atomic** に設定されます）。以前に作成したパスワードハッシュがある場合は、代わりに **--iscrypted** オプションを使用できます。
- **auth** コマンドは、**--enablesshadow** を使用してユーザーパスワードを **/etc/shadow** ファイルに保存するよう指示し、**--passalgo=sha512** は SHA512 アルゴリズムを使用してこれらのパスワードを暗号化することを指示します。
- **ostreesetup** コマンドはインストーラーに対し、ostree ファイルシステムを取得および設定する方法を指示します。
- **services** コマンドは、Atomic ホストに不適切な一部のサービスを無効にします。
- **clearpart --all --initlabel** コマンドは、アタッチされたネットワークストレージを含む、インストーラーが到達できるディスクをすべて削除します。
- **zerombr** を使用すると、Anaconda が無人インストールを可能にする確認を求めるプロンプトを出さないようにします。
- **autopart** コマンドは、自動的にパーティションを設定します（後で追加）。
- **%include** コマンドは、OSTree リポジトリを参照するコマンドが含まれ、cloud-init サービスを無効にするコマンドが含まれるファイルを参照します。このコマンドは、RHEL Atomic Host には必須です。
- ファイルの最後にある **%post** セクションは、いくつかのコマンドを実行して、インストール完了後にシステムをさらに設定します。

デフォルトでは、Red Hat Enterprise Linux Atomic Host のパーティション設定は、**autopart** コマンドで自動的に行われ、論理ボリューム管理(LVM)スタイルのパーティション設定を設定します。**autopart**

パーティション設定が推奨されますが、物理ボリューム、ボリュームグループ、論理ボリュームの名前、およびこれらのエンティティに関連付けられたディスク領域の容量などを独自にパーティション設定するオプションがあります。以下は、上記のキックスタート例に示されている **autopart** エントリーを置き換えるためにパーティションを手動で設定する方法の例になります。

```
zerombr
part /boot --ondisk=sda --asprimary --fstype="xfs" --size=512
part pv.01 --ondisk=sda --asprimary --fstype="lvm" --grow
volgroup vg.atomic --pesize=16384 pv.01
logvol swap --fstype="swap" --name=lv.swap --vgname=vg.atomic --size=4096
logvol / --fstype="xfs" --name=lv.root --vgname=vg.atomic --grow
```

この例では、`/boot` パーティションに割り当てられたディスク `/dev/sda` に `xfs` ファイルシステムタイプで 512MB のプライマリパーティションを設定します。残りのディスクは、`pv.01` という名前の LVM 物理ボリューム(`lvm`)に割り当てられます。その物理ボリュームは、`vg.atomic` という名前のボリュームグループに割り当てられます。そのボリュームグループから 2 つの論理ボリュームが作成されます。4G swap パーティションと、ボリュームグループの残りの領域を消費する XFS ファイルシステムを持つルートパーティション(`/`)です。

## 2.3. 仮想マシンのインストール

本章では、いくつかの異なる仮想化環境およびパブリッククラウドサービスに Red Hat Enterprise Linux Atomic Host をインストールする方法を説明します。以下の手順を実行する前に、Red Hat Enterprise Linux [7 インストールガイド](#) から [Red Hat Enterprise Linux](#) のダウンロードの説明に従って、お使いの環境に適切な ISO イメージをダウンロードします。

### 2.3.1. qcow2 メディアを使用した Linux ハイパーバイザーのインストール

以下のセクションでは、Red Hat Enterprise Linux 7 システムの Linux ハイパーバイザー環境で **qcow2** ディスクイメージを使用した Red Hat Enterprise Linux Atomic Host のインストール方法を説明します。

#### 概要

Red Hat Enterprise Linux Atomic Host は、Linux ハイパーバイザーで使用する準備が整った完全なディスクイメージとして利用できます。このバリエーションは圧縮された **gzip** アーカイブとして配布されます。以下のコマンドを使用して展開します。

```
# gzip -d rhel-atomic-host-7.qcow2.gz
```

結果として圧縮されていない **qcow2** イメージを使用して、Red Hat Enterprise Linux Atomic Host のインスタンスを作成できます。つまり、仮想マシンを起動すると、ファイルは書き込まれます。そのインスタンスを使用してインスタンスを起動すると、別のインスタンスを起動したり、**cloud-init** を使用して再設定したりすることはできません。したがって、最初のインスタンスを起動する前に、元の **qcow2** ファイルをバックアップする必要があります。**qemu-img** コマンドを使用して、変更されていないファイルのスナップショットを作成できます。

```
# qemu-img create -f qcow2 -o backing_file=rhel-atomic-host-standard.qcow2 atomic-beta-instance-0.qcow2
```

このコマンドは、元の、変更されていないイメージである **rhel-atomic-host-standard.qcow2** という名前のスナップショットと、実際の仮想マシンに使用できる **atomic-beta-instance-0.qcow2** という名前の新規ファイルを作成します。

### 2.3.1.1. インストールの準備

インストール設定オプションは、cloud-init 設定ファイルのペアで設定されます。

- **meta-data**

インストールされている Red Hat Enterprise Linux Atomic Host のインスタンスを識別する情報を提供するプレーンテキストファイル。そのコンテンツは以下の例のようになります。

```
instance-id: Atomic0
local-hostname: atomic-00
```

**instance-id** には任意の識別名を指定でき、**local-hostname** はサイトの標準に準拠するホスト名である必要があります。

- **user-data**

システム上のユーザーに関する情報を提供するプレーンテキストファイル。この情報は、Red Hat Enterprise Linux Atomic Host インスタンスへのアクセスを有効にするために使用されます。デフォルトでは、**root** ユーザーはパスワードロックされます。そのため、**user-data** ファイルを作成しないとログインできなくなります。

ユーザーデータファイルの例を以下に示します。

```
#cloud-config
password: atomic
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
- ssh-rsa AAA...SDvz user1@yourdomain.com
- ssh-rsa AAB...QTuo user2@yourdomain.com
```



#### 注記

例の最初の行(**#cloud-config**)はコメントやコマンドの例ではありません。これは、設定ファイルの必須の行です。

この例では、**cloud-user** ユーザーがパスワードまたは **SSH** キーのいずれかでログインできるようになります。両方の方法を使用できますが、必須ではありません。初期パスワードは **パスワード** 行に設定されます。ユーザーがこのインスタンスで初めてログインすると、**chpasswd** 行で定義されたパスワードを変更するように求められます。最初にパスワードがプレーンテキストで保存されるため、初回ログイン後にユーザーにパスワードの変更を強制することが推奨されます。

この例の最後の 4 行は、**SSH** を使用してリモートログインを設定します。**ssh\_pwauth: True** 行はパスワードを使用して **SSH** を有効にし、**ssh\_authorized\_keys** は 1 つ以上の認証済み公開鍵のブロックを開始します。このファイルで説明されているキーは `~/.ssh/authorized_keys` ファイルに追加されます。各承認キーは別々の行に指定し、2 つのスペースの後にハイフン(-)と別のスペースが続きます。

これらのファイルについての詳細は、cloud-init ISO ファイルの作成のセクションを参照してください。

上記の両方のファイルを作成したら、それらを ISO イメージにパッケージ化する必要があります。このイメージは、仮想マシンの仮想設定 CD として使用されます。ファイルをイメージにパッケージ化するには、以下のコマンドを使用します。

■

```
# genisoimage -output atomic0-cidata.iso -volid cidata -joliet -rock user-data meta-data
```

これにより、**atomic0-cidata.iso** という名前の新しい ISO イメージファイルが作成されます。

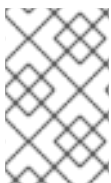
### 2.3.1.2. Red Hat Enterprise Linux Atomic Host の初回起動

分散 **qcow2** イメージを展開し、前のセクションで説明した設定イメージを作成したら、仮想マシンを作成してインストールプロセスを開始できます。本セクションでは、**virt-install** コマンドを使用してインスタンスを作成する方法を説明します。**virt-manager** グラフィカルインターフェイスを使用することもできます。いずれも [Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#) に記載されています。[Red Hat Enterprise Linux 7 での仮想化の概要](#)は、[Red Hat Enterprise Linux 7 仮想化スタートガイド](#)も参照してください。

以下のコマンドは、Red Hat が配布する **qcow2** イメージと、以前に作成した設定イメージを使用して、新しい仮想マシンを作成します。

```
# virt-install --import --name Atomic0 --ram 4096 --vcpus 2 --disk path=/path/to/rhel-atomic-host-standard.qcow2,format=qcow2,bus=virtio --disk path=/path/to/atomic0-cidata.iso,device=cdrom --network bridge=virbr0 --graphics vnc
```

2つの **--disk-path=** オプションは、作成する必要があるイメージファイルとデバイスタイプの場所を指定します（メインイメージ用の **virtio** デバイスと設定イメージの仮想 CD ドライブ）。また、4 GB の RAM (**-ram 4096**)と 2つの仮想 CPU (**--vcpus 2**)を仮想マシンに割り当て、VNC グラフィカルインターフェイス(**--graphics vnc**)およびネットワークブリッジ(**--network bridge=virbr0**)を設定します。これらの設定はニーズに合わせて変更できますが、常に両方のイメージファイルを使用する必要があります。



#### 注記

現在、**DHCP** は Red Hat Enterprise Linux Atomic Host で使用する推奨ネットワーク設定方法です。ネットワーク設定は、初回起動後に **/etc** ディレクトリーの設定ファイルを編集して変更できます。



#### 注記

ホストマシン外から仮想マシンにアクセスできるようにする場合。ダイレクトネットワークインターフェイスを使用する必要があります。たとえば、**--network bridge=virbr0** を **--network type=direct,source=em1** に置き換えることができます。**em1** は、ホストシステム上のアクティブなネットワークインターフェイスの名前です。

この時点で、**ユーザーデータ** ファイルで設定した認証情報を使用して Red Hat Enterprise Linux Atomic Host 仮想マシンにログインできます。**root** シェルにアクセスするには、**sudo -i** コマンドを使用します。ホストシステムから仮想マシンのコンソールに接続するには、次のコマンドを使用します。

```
# virsh console Atomic0
```

**Atomic0** は、仮想マシンの名前(**virt-install** コマンドの **--name** オプション)に置き換えます。

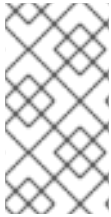
新しい Red Hat Enterprise Linux Atomic Host インスタンスの使用方法については、[Red Hat Enterprise Linux Atomic Host 7 Getting Started Guide](#) を参照してください。

### 2.3.2. Red Hat Enterprise Virtualization 環境のインストール



次のセクションでは、Red Hat Enterprise Virtualization (RHEV)を使用して、**.ova** ファイルと ISO ファイルを使用して RHEL Atomic Host を実行する仮想マシンを作成する方法を説明します。

## .OVA ベースのインストール



### 注記

現在、Atomic Host の RHEV OVA イメージを RHEV にインポートすることはできません。

詳細は、この [Bugzilla](#) を参照してください。

**.ova** ベースのインストール方法では、Red Hat Enterprise Linux Atomic Host インストールを迅速にデプロイできますが、後続のセクションで説明されている ISO ベースのインストールよりもカスタマイズが少なくなります。

1. [Red Hat Enterprise Linux のダウンロード](#) から RHEL Atomic Host **.ova** メディアを取得します。
2. **.ova** ファイルを Red Hat Enterprise Virtualization Manager にコピーします。
3. **engine-image-uploader** コマンドを使用して、**.ova** ファイルを Export ストレージドメインにアップロードします。
4. Red Hat Enterprise Virtualization インスタンスにアップロードされた **.ova** ファイルから Red Hat Enterprise Linux のインスタンスを作成します。

## ISO ベースのインストール

**.iso** ベースのインストール方法では、**.ova** ベースのインストール方法よりも多くのインストールをカスタマイズできますが、Atomic 環境をホストする仮想マシンの設定が必要になります。

1. [Download Red Hat Enterprise Linux](#) から Red Hat Enterprise Linux Atomic Host インストールメディアを取得し、Red Hat Enterprise Virtualization Manager のファイルシステムにコピーします。
2. **engine-image-uploader** を使用して、ISO イメージを Red Hat Enterprise Virtualization 環境のストレージドメインに追加します。
3. アップロードした Red Hat Enterprise Linux Atomic Host ISO イメージを新しい仮想マシンに割り当て、その仮想マシンに Red Hat Enterprise Linux Atomic Host をインストールします。
4. 新たに作成した Red Hat Enterprise Linux Atomic Host 仮想マシンを使用します。

詳細は、[Red Hat Enterprise Virtualization](#) 用に設定されたドキュメント を参照してください。

### 2.3.2.1. .ova ファイルからの Red Hat Enterprise Linux Atomic Host のインストール

次のセクションでは、**.ova** (Open Virtualization Appliance) ソースから Red Hat Enterprise Virtualization に Red Hat Enterprise Linux Atomic Host をインストールする方法を説明します。この操作は、3つの段階の手順で設定されます。最初の段階では、Red Hat Enterprise Virtualization 環境のエクスポートストレージドメインで **.ova** ファイルを展開する方法と、Red Hat Enterprise Virtualization が展開されたファイルの所有権を持つようにパーミッションを設定する方法を説明します。2番目のステージでは、エクスポートドメインから Red Hat Enterprise Virtualization 環境に仮想マシンテンプレートをインポートする方法を説明します。3番目のステージでは、インポートされたテンプレートから仮想マシンを作成する方法を説明します。



## engine-image-uploader を使用した .ova ファイルのインポート

この手順では、**rhevnm-image-uploader** を使用して、Red Hat Enterprise Linux Atomic Host の仮想マシンテンプレートを Export ストレージドメインにアップロードする方法を説明します。Red Hat Enterprise Virtualization Manager 環境内から以下の手順を実行します。

1. **.ova** ファイルを Red Hat Enterprise Virtualization Manager に転送します。

```
[a computer that is not the RHEV Manager]# scp filename.ova root@rhevhostname.com:/
```

2. Red Hat Enterprise Virtualization Manager マシンに root としてログインします。

```
[a computer that is not the RHEV Manager]# ssh root@rhevhostname.com
```

3. **.ova** ファイルを転送したディレクトリーに移動します。この例では、ディレクトリーが root (/)であることを前提としています。

```
[RHEVM]# cd /
```

4. 以下のコマンドを使用して、**.ova** ファイルを Export ストレージドメインにアップロードします。

```
[RHEVM]# engine-image-uploader -N imagename -e Export upload filename.ova
```

**-N imagename** を追加して、イメージに人間が判読できるファイル名を指定します。それ以外の場合は、イメージの名前は長い英数字の文字列になります。また、エクスポートドメイン名を Export に置き換え、**.ova** ファイルの名前を filename.ova ファイルに置き換えます。

5. プロンプトが表示されたら、admin@internal oVirt engine ユーザーの REST API パスワードを入力します。アップロードされたファイルのサイズによっては、アップロードに時間がかかる場合があります。アップロードがサイレントに成功し、完了するとコマンドプロンプトに戻ります。

## 仮想マシンテンプレートの Red Hat Enterprise Virtualization へのインポート

**.ova** ファイルが展開され、それに含まれる仮想マシンテンプレートにそのパーミッションが設定された後に、Red Hat Enterprise Virtualization が操作できるように仮想マシンテンプレートがそのパーミッションが設定されていると、管理ポータルของผู้ใช้ 인터フェイスから仮想マシンテンプレートを Red Hat Enterprise Virtualization 環境にインポートする必要があります。この手順が完了すると、インポートされたテンプレートから仮想マシンを作成できます。

1. Red Hat Enterprise Virtualization Manager 管理者ポータルに管理 者 としてサインインします。
2. Red Hat Enterprise Virtualization Manager ユーザーインターフェイスのナビゲーションペイン（インターフェイス上部のペイン）の **Storage** タブをクリックします。
3. Red Hat Enterprise Virtualization Manager ユーザーインターフェイスのナビゲーションメニューで、Export Domain の名前をクリックします。
4. Red Hat Enterprise Virtualization Manager ユーザーインターフェイスの詳細ペイン（インターフェイス下部のペイン）の **Template Import** タブをクリックします。
5. Red Hat Enterprise Virtualization Manager ユーザーインターフェイスの **詳細ペイン（インターフェイス下部のペイン）** で、インポートする予定のファイルの名前をクリックします。

6. Red Hat Enterprise Virtualization Manager ユーザーインターフェイスで、Details パネルの左上にある **Import** をクリックします。
7. **Import Template** ウィンドウで、インポートする仮想マシンの名前をクリックします。
8. **Import Template** ウィンドウで、右下隅で **OK** をクリックします。

#### cloud-init ISO の ISO ドメインへの追加

1. cloud-init ISO ファイルの作成セクションの手順に従って cloud-init ISO を作成します。
2. Red Hat Enterprise Virtualization 環境の RHEV Manager マシンにリモートから、**scp** を使用して cloud-init ISO を Red Hat Enterprise Virtualization 環境の RHEV Manager マシンのファイルシステムにコピーします。

```
[a computer that is not the RHEV Manager]# scp atomic-cloud.iso root@rhevhostname.com:/
```

1. Red Hat Enterprise Virtualization Manager マシンに **root** としてログインします。

```
[a computer that is not the RHEV Manager]# ssh root@rhevhostname.com
```

1. **atomic-cloud.iso** をアップロードしたディレクトリーに移動します。

```
[RHEVM]# cd /
```

1. **rhev-iso-uploader** を使用して cloud-init ISO を ISO ドメインにアップロードします。

```
[RHEVM]# rhev-iso-uploader --iso-domain=domain_name upload atomic-cloud.iso
```

1. Red Hat Enterprise Virtualization Manager 管理者ポータルに管理 **者** としてサインインします。
2. Red Hat Enterprise Virtualization Manager ユーザーインターフェイスで、**Navigation** ペインの **Storage** タブを選択します。
3. **Details** ペイン（インターフェイス下部のペイン）で、**Images** タブを選択します。
4. **.iso** ファイルが ISO ドメインに存在することを確認します（**詳細** ペインの **Images** サブタブに存在する場合は一覧に表示されます）。

#### インポートされたテンプレートからの仮想マシンの作成

これで、Red Hat Enterprise Linux Atomic Host 仮想マシンテンプレートが展開され、Red Hat Enterprise Virtualization 環境にインポートされ、cloud-init ISO ファイルが Red Hat Enterprise Virtualization ISO ドメインに存在し、以下の手順で Red Hat Enterprise Linux Atomic Host 仮想マシンを作成できます。

1. Red Hat Enterprise Virtualization Manager ユーザーインターフェイスにログインします。
2. **ナビゲーション** ペインで **Virtual Machines** タブを開きます。
3. Red Hat Enterprise Virtualization ユーザーインターフェイスのナビゲーションメニューで、**New VM** をクリックします。
4. **New Virtual Machine** ウィンドウの **Based on Template** ドロップダウンメニューで、先ほどインポートした Red Hat Enterprise Linux Atomic Host テンプレートの名前を選択します。

5. **New Virtual Machine** ウィンドウで、Name フィールド、Description フィールド、および コメントフィールドに入力します。
6. **New Virtual Machine** ウィンドウの **Boot Options** タブで Attach CD チェックボックスを選択し、この仮想マシンで使用するユーザーの認証情報が含まれる cloud-init ISO の名前を選択します。
7. **OK** をクリックします。

### Atomic Host VM での RHEV ゲストエージェントの更新

RHEV Manager が Atomic Host 仮想マシンを制御できるようにするには、その仮想マシンでゲストエージェントを実行する必要があります。ovirt-guest-agent は RHEV Manager と連携してランタイムデータと中心情報を提供し、RHEV Manager が仮想マシンの操作（シャットダウンや再起動を含む）を制御できるようにします。

RHEV の最新の Atomic Host ova イメージには、JBL-guest-agent という名前のコンテナ形式で ovirt `-guest-agent` が含まれています。インポートされた ova イメージから仮想マシンを作成すると（前述のように）、仮想マシンに含まれる rhvm-guest-agent コンテナイメージが、仮想マシンの起動時に実行されるように自動的に設定されます。

RHEV 環境の Atomic Host 仮想マシンにログインし、次のコマンドを実行して、nsN-guest-agent コンテナのステータスを確認し（必要に応じてコンテナを更新する）ことができます。

1. rhvm-guest-agent が使用可能であり、実行中であることを一覧表示します。

```
# runc list
ID          PID STATUS BUNDLE          CREATED
rhvm-guest-agent 674 running /var/lib/containers/atomic/rhvm-guest-agent.0 2017-06-...
```

1. systemd サービスとして実行されている rhvm-guest-agent のステータスを確認します。

```
# systemctl status rhvm-guest-agent
● rhvm-guest-agent.service - oVirt Guest Agent Container
   Loaded: loaded (/etc/systemd/system/rhvm-guest-agent.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2017-06-19 19:06:58 UTC; 1 weeks 0 days ago
   Main PID: 644 (runc)
   Memory: 5.8M
   CGroup: /system.slice/rhvm-guest-agent.service
           └─644 /bin/runc --systemd-cgroup run rhvm-guest-agent
```

1. rhvm-guest-agent を更新します。新しいバージョンの rhvm-guest-agent コンテナが利用できる場合は、以下のコマンドを実行してコンテナを更新できます（この例では新しいバージョンは利用できません）。

```
# atomic containers update rhvm-guest-agent
Latest version already installed.
```

### 2.3.2.2. ISO イメージからの Red Hat Enterprise Linux Atomic Host のインストール

#### ISO のアップロード



## 注記

このセクションでは、ISO イメージからの Red Hat Enterprise Linux Atomic Host システムのインストール手順のみを説明します。このセクションでは、**.ova** ファイルからの Red Hat Enterprise Linux Atomic Host システムの作成は含まれません。

1. ISO ファイルを Red Hat Enterprise Virtualization Manager のファイルシステムに転送します。

```
[a computer that is not the RHEV Manager]# scp filename.iso root@rhevhostname.com:/
```

2. Red Hat Enterprise Virtualization Manager のバックエンドに root としてログインします。これは、Red Hat Enterprise Virtualization Manager 管理者ポータルにログインする必要があるわけではありません。

```
[a computer that is not the RHEV Manager]# ssh root@rhevhostname.com
```

3. ISO ファイルを転送したディレクトリーに移動します。

```
[RHEVM]# cd /
```

4. Red Hat Enterprise Virtualization Manager 上の ISO ストレージドメインの名前を確認します。この例では、ISO ストレージドメインの名前は **ISO\_DOMAIN** です。

```
# rhevm-iso-uploader list
ISO Storage Domain Name | Datacenter          | ISO Domain Status
ISO_DOMAIN              | Default             | active
```

5. **rhev-iso-uploader** を使用して、Red Hat Enterprise Linux Atomic Host インストール ISO イメージを Red Hat Enterprise Virtualization ストレージドメインにアップロードします。

```
[RHEVM]# rhevm-iso-uploader upload -i ISO_DOMAIN filename.iso
```

Red Hat Enterprise Virtualization での ISO ドメインへの ISO ファイルのアップロードに関する詳細は、[Red Hat Enterprise Virtualization Installation Guide](#) を参照してください。

## Creating a Red Hat Enterprise Linux Atomic Virtual Machine

1. Red Hat Enterprise Virtualization Manager にログインします。
2. **Virtual Machines** タブをクリックします。
3. **New VM** ボタンをクリックして、**New Virtual Machine** ウィンドウを開きます。
4. **New Virtual Machine** ウィンドウの左下にある **Show Advanced Options** ボタンをクリックします。
5. **General** タブで、Name および Operating System フィールドに入力します。他のフィールドのデフォルト設定を使用するか、必要に応じて変更できます。
6. **New Virtual Machine** ウィンドウの左側にあるメニューで **Boot Options** をクリックします。
7. **Boot Sequence** メニューで、**First Device** ドロップダウンメニューで **CD-ROM** を選択します。

8. **Boot Sequence** メニューで、**2番目のデバイス** ドロップダウンメニューで **Hard Disk** を選択します。
9. **Attach CD** チェックボックスを選択します。
10. **Attach CD** チェックボックスの右側にあるドロップダウンメニューで、Red Hat Enterprise Linux Atomic Host インストール ISO の名前を選択します。
11. **New Virtual Machine** ウィンドウ右下の **OK** をクリックします。
12. **New Virtual Machine - Guide Me**画面が開き、**Configure Network Interfaces**と **Configure Virtual Disks** の2つのボタンが表示されます。
13. ネットワークインターフェイスの **設定** をクリックします。
14. **New Network Interface** ウィンドウが開きます。仮想マシンの仮想ネットワークインターフェイスを作成するには、このウィンドウのデフォルト値で十分です。
15. **New Network Interface** ウィンドウ右下の **OK** をクリックします。
16. **New Virtual Machine - Guide Me**ウィンドウで、**Configure Virtual Disks** ボタンをクリックします。
17. **New Virtual Disk** ウィンドウが開きます。**Size (GB)** フィールドに、仮想ハードドライブのサイズを入力します。
18. **New Virtual Disk** ウィンドウ右下の **OK** をクリックします。
19. **New Virtual Machine - Guide Me**ウィンドウで、右下の **Configure Later** をクリックします。

### 2.3.3. Red Hat Enterprise Linux OpenStack Platform Installation

本項では、**QCOW2** イメージを使用して、Red Hat Enterprise Linux OpenStack Platform で Red Hat Enterprise Linux Atomic Host のインスタンスを起動する方法を説明します。この手順を開始する前に、[Red Hat Enterprise Linux のダウンロード](#) から **QCOW2** イメージをダウンロードします。

#### QCOW2 イメージからの Red Hat Enterprise Linux Atomic Host インスタンスの作成

以下の手順は、Red Hat Enterprise Linux OpenStack Platform に精通していることを前提としています。Red Hat Enterprise Linux OpenStack Platform についての詳しい情報は、[Red Hat Enterprise Linux OpenStack Platform End User Guide](#) を参照してください。

1. プロジェクトを作成します。
  - a. Red Hat Enterprise Linux OpenStack Platform Dashboard にログインします。
  - b. **Admin** タブに移動し、**Identity Panel** の下にある **Projects** をクリックしてプロジェクトを作成します。
  - c. **Create Project** をクリックして、サイトの要件を満たす Project Name を指定します。追加の設定は必要ありませんが、サイトの要件を満たすために実行する必要があります。
2. プロジェクトのネットワークを設定します。これは、サイトの設定によって異なります。通常、以下の手順が必要です。
  - a. プロジェクトの内部ネットワーク用のネットワークおよびサブネットを作成します。

- b. ルーターを作成し、ゲートウェイを割り当て、内部ネットワークを外部ネットワークに接続するように設定するインターフェイスを作成します。
3. インスタンスで使用するキーペアを作成またはアップロードします。キーペアの設定は、**キーペア タブのアクセスとセキュリティーのコンピュートの管理のプロジェクトタブ**にあります。
4. **QCOW2** イメージを Red Hat Enterprise Linux OpenStack Platform に読み込みます。
  - a. **Manage Compute** の **Project** タブにある **Images & Snapshots** をクリックします。
  - b. **イメージの作成** をクリックして、以下の情報を提供します。
    - **名前**: 意味のあるイメージ名
    - **イメージソース**: イメージファイルを選択して、ローカルワークステーションからファイルをアップロードできるようにします。
    - **形式**: QCOW2 を選択します。
    - **最小ディスク(GB)**: このイメージで許可されるディスク領域の最小量。詳細は、[Disk Space and Memory Requirements](#) を参照してください。
    - **最小 RAM (MB)**: このイメージで使用できる最小メモリー量。詳細は、[Disk Space and Memory Requirements](#) を参照してください。
  - c. 最後に、**Choose File** をクリックしてアップロードする **QCOW2** イメージを選択し、**Create Image** をクリックしてアップロードを開始します。
5. cloud-init を使用した基本的な初回起動設定など、起動するインスタンスを設定します。
  - a. **Instances Screen** の **Projects** タブにある **Launch Instance** ボタンをクリックして、インスタンスの起動ダイアログボックスにアクセスします。
  - b. **Details** タブの **Launch Instance** ダイアログボックスで、以下の情報を入力します。
    - **インスタンス名**: 意味のあるインスタンス名
    - **フレーバー**: Red Hat Enterprise Linux Atomic Host の最小要件を満たすアプリケーション要件に適したサイズインスタンス。
    - **インスタンスのブートソース**: 前の手順で読み込んだイメージを選択します。詳細は、[Disk Space and Memory Requirements](#) を参照してください。
  - c. **アクセスとセキュリティータブのインスタンスの起動** ダイアログボックスに、以下の情報を入力します。
    - **キー ペア**: このインスタンスで使用するキーペアを選択します。
  - d. **Networking** タブの **Launch Instance** ダイアログボックスで、以下の情報を入力します。
    - **選択した ネットワーク**: このインスタンスで使用するネットワークを選択します。
  - e. **Post-Creation** タブの **Launch Instance** ダイアログボックスで、以下の情報を入力します。
    - **カスタマイズスクリプト**: このフィールドで、cloud-init のユーザーデータ ファイルと同等のものを貼り付けます。user-data は、システムのユーザーおよび設定に関する情報を提供するプレーンテキストファイルです。この情報は、Red Hat Enterprise Linux

Atomic Host インスタンスへのアクセスを有効にするために使用されます。デフォルトでは、**root** ユーザーはパスワードで保護されるため、**ユーザーデータ** ファイルを作成しないとログインできなくなります。

An example of a `user-data` file is below:

```
#cloud-config
password: atomic
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
- ssh-rsa AAA...SDvz user1@yourdomain.com
- ssh-rsa AAB...QTuo user2@yourdomain.com
```



### 注記

例の最初の行(**#cloud-config**)はコメントやコマンドの例ではありません。これは、設定ファイルの必須の行です。

この例では、**cloud-user** ユーザーがパスワードまたは SSH キーのいずれかでログインできるようになります。両方の方法を使用できますが、必須ではありません。初期パスワードは **password** 行に設定されます。ユーザーがこのインスタンスで初めてログインすると、**chpasswd** 行で定義されたパスワードを変更するように求められます。最初にパスワードがプレーンテキストで保存されるため、初回ログイン後にユーザーにパスワードの変更を強制することが推奨されます。

この例の最後の4行は、SSH を使用してリモートログインを設定します。**ssh\_pwauth: True** 行はパスワードを使用してSSH を有効にし、**ssh\_authorized\_keys** は1つ以上の認証済み公開鍵のブロックを開始します。このファイルで説明されているキーは `~/.ssh/authorized_keys` ファイルに追加されます。各承認キーは別々の行に指定し、2つのスペースの後にハイフン(-)と別のスペースが続きます。

このファイルの詳細は、cloud-init ISO ファイルの作成のセクションを参照してください。

- a. **Launch** ボタンをクリックしてインスタンスを起動します。

### 2.3.4. VMware のインストール

VMware vSphere は、仮想マシンリソースのデプロイおよび管理手段を提供します。本セクションでは、VMware vSphere クライアントを使用して Red Hat Enterprise Linux Atomic Host を実行する方法を説明します。この記事の例では、ISO イメージが Red Hat Enterprise Linux 7 システムで作成されており、Red Hat Enterprise Linux Atomic Host は、Microsoft Windows システムで実行されている単一の ESXi 5.5 ハイパーバイザーおよび vCenter ホストとして設定された VMware vSphere で実行されました。

#### Getting a Red Hat Enterprise Linux Atomic Host Image

VMware vSphere で実行できる Red Hat Enterprise Linux Atomic Host 仮想マシンイメージを作成するには、最初に Red Hat Enterprise Linux のダウンロード ページから VMware 用の [Red Hat Enterprise Linux Atomic Host OVA ファイル](#) をダウンロード します。

vSphere OVA プラグインには、設定可能なネットワークコントローラーと設定可能な SCSI コントローラーがあります。

設定可能なパラメーターは以下のとおりです。



### vsphere\_scsi\_controller\_type

有効な設定は **Isilogic** および **VirtualSCSI** です。

### vsphere\_network\_controller\_type

有効な設定は、**E1000** および **VmxNet3** です。

これらのパラメーターが明示的に設定されていない場合は、デフォルトで準仮想化以外の設定に設定されます。SCSI コントローラーの非準仮想化設定は **Isilogic** です。ネットワークコントローラーの非準仮想化設定は **E1000** です。

## cloud-init ISO ファイルの作成

Red Hat Enterprise Linux Atomic Host システムの設定に使用される情報を含む cloud-init ISO イメージを作成する必要があります。この情報には、ホスト名、ユーザー名とパスワード、およびその他の設定を含めることができます。以下の手順で説明するように、必要な設定情報を作成し、ISO イメージを生成します。

1. cloud-init **メタデータデータ** ファイルを作成します。

最後のインストール設定オプションは、cloud-init 設定ファイルのペアで設定されます。最初のインストール設定ファイルには、メタデータが含まれます。テキストエディターでこのファイルを作成し、**メタデータ** を呼び出します。このファイルは、インストールされている Red Hat Enterprise Linux Atomic Host のインスタンスを識別する情報を提供します。**instance-id** には任意の識別名を指定でき、**local-hostname** はサイトの標準に準拠するホスト名である必要があります。以下に例を示します。

```
instance-id: Atomic0
local-hostname: atomic-00
```

2. cloud-init **ユーザーデータ** ファイルを作成します。

2 つ目のインストール設定オプションファイルは、ユーザーデータファイルです。このファイルは、システム上のユーザーに関する情報を提供します。テキストエディターでこれを作成し、**user-data** を呼び出します。このファイルは、Red Hat Enterprise Linux Atomic Host のインストールを可能にするために使用されます。デフォルトでは、root ユーザーはパスワードがロックされ、この手順が省略されている場合はログインできません。以下は、**user-data** ファイルの例です。

```
#cloud-config
password: atomic
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
- ssh-rsa AAA...SDvz user1@yourdomain.com
- ssh-rsa AAB...QTuo user2@yourdomain.com
```

この **user-data** ファイルにより、デフォルトのユーザー **cloud-user** がパスワードまたは SSH キーを使用してログインできるようになります。両方の方法を使用できますが、必須ではありません。パスワードログインは、**password** と **chpasswd** の行で有効になります。パスワードには、**cloud-user** ユーザーのプレーンテキストのパスワードが含まれます。**chpasswd** 行は、最初のログインがすぐにパスワードの変更を要求しないように、パスワードの有効期限をオフにします。これは任意です。パスワードを設定する場合は、パスワードがプレーンテキストファイルに保存されているため、初回ログイン時にパスワードを変更することが推奨されます。

SSH ログインは、ファイルの最後の 3 行で有効になります。**ssh\_pwauth** 行は、SSH ログインを有効にします。**ssh\_authorized\_keys** 行は、1 つ以上の認証キーのブロックを開始しま



す。**ssh-rsa** 行に一覧表示されている各パブリック SSH キーは、`cloud-user` `~/.ssh/authorized_keys` ファイルに追加されます。この例では、2つのキーが一覧表示されます。この例では、鍵は切り捨てられており、実際のファイルで公開鍵全体を一覧表示する必要があります。**ssh-rsa** の行の前にスペースを2つ追加します。その後、ハイフンの後に別のスペースが続く必要があります。

### 3. ISO ファイルを作成します。

ファイルが完了したら、ISO イメージにパッケージ化する必要があります。この ISO イメージは、仮想マシンと共に仮想設定 CD として使用されます。**atomic0-cidata.iso** という ISO イメージは、Red Hat Enterprise Linux で以下のコマンドで作成されます。

```
# genisoimage -output atomic0-cidata.iso -volid cidata -joliet -rock user-data meta-data
```

### 4. 新しく作成された ISP イメージを、VMware を実行しているホストに転送します。

#### 2.3.4.1. VMware での Red Hat Enterprise Linux Atomic Host 仮想マシンの設定\*

VMware vSphere クライアントで Red Hat Enterprise Linux Atomic Host を実行する手順は次のとおりです。

1. 以前に作成した ISO イメージを VMware vSphere データストアに追加します。
2. vSphere で OVF テンプレートとして OVA ファイルをデプロイします。
3. ISO イメージを CD/DVD ドライブとして vSphere テンプレートに割り当てます。
4. Red Hat Enterprise Linux Atomic Host 仮想マシンを実行します。



#### 注記

この手順では、VMware vSphere に精通しており、VMware vSphere の特定バージョンへの参照では記述されていないことを前提としています。

#### データストアへのイメージの追加

1. VMware vSphere クライアントを開きます。
2. 左側のペインで、データストアに **アクセス** します。
3. ターゲットデータストアを選択します。
4. **Browse this datastore** を選択します。
5. フォルダーアイコンを選択し、新しいフォルダーを作成します。この例では、**atomic01/** という名前です。
6. 新規フォルダー **atomic01/** が **強調表示** された状態で、GUI オプションを選択して、データをデータストア（およびフォルダー）にアップロードします。
7. 以前に作成した cloud-init ISO ファイル（例：**atomic01-cid.iso**）を参照し、これを選択してデータストアにアップロードします。同じ名前が付けられたファイルがデータストアにすでに存在する場合は、上書きするかどうか尋ねられます。
8. Datastore Browser を閉じます。

## OVF テンプレートのデプロイ

1. **Home** を選択し、**Inventory** を選択し、**Hosts and Clusters** オプションを選択します。
2. **File and Deploy OVF Template** を選択します。
3. OVA ファイルがある場所（例：**rhel-atomic-cloud-7.1-6.x86\_64.vsphere.ova**）を参照し、これを選択して **Open** をクリックします。
4. **Next** ボタンを選択します。OVF テンプレートの詳細画面が表示されます。
5. **OVF テンプレートの詳細画面** から、再度 **Next** を選択します。
6. Red Hat Enterprise Linux Atomic Host 仮想マシンの名前を入力します。
7. 仮想マシンを実行するホストまたはクラスターを選択し、**Next** をクリックします。
8. **Disk Format** オプションを選択します。デフォルトは残すことができます。**次へ** をクリックします。



### 注記

**Power on after deployment** チェックボックスを選択しないでください。これを選択すると仮想マシンが起動し、cloud-init ISO をアタッチした後に起動する必要があります。

9. **Finish** をクリックしてテンプレートのデプロイを開始します。これには2分以上かかるはずで

## ISO イメージを CD/DVD として仮想マシンにアタッチ

1. 新たに追加した Red Hat Enterprise Linux Atomic Host テンプレートを右クリックし、**Edit Settings** を選択します。（仮想マシンを表示するために、**Virtual Machines** タブを選択するか、Tree View でサーバーを展開します。）
2. **Virtual Machine Properties** 画面で **Add** を選択し、**CD/DVD Drive** を選択して **Next** をクリックします。
3. **Use an ISO image** オプションを選択し、**Next** をクリックします。
4. 以前に作成した ISO イメージ(**atomic0-cidata.iso**と呼ばれます)を参照して選択し、**Next** をクリックします。ISO は、作成したフォルダーにある、アップロードしたデータストアにあります。
5. **Advanced** オプションが表示されたら、**Next** をクリックして続行します。
6. **Ready to Complete** 画面が表示されたら、**Finish** をクリックして設定を完了します。これで、Red Hat Enterprise Linux Atomic Host 仮想マシンを実行する準備が整いました。
7. **OK** をクリックして **Properties** 画面を終了します。

## Red Hat Enterprise Linux Atomic Host 仮想マシンの実行

1. Red Hat Enterprise Linux Atomic Host 仮想マシンを起動するには、 をクリックしてこれを強調表示し、**Power On** ボタンを選択します。
2. **Console** タブを選択して、仮想マシンが起動していることを監視します。

ここで説明するように Red Hat Enterprise Linux Atomic Host を設定している場合は、cloud-init ISO の作成時に定義したユーザー名 **cloud-user** とパスワード **atomic** を使用して仮想マシンにログインできるはずです。

### 2.3.5. Microsoft Hyper-V Installation

本セクションでは、Microsoft Hyper-V を使用して、Red Hat Enterprise Linux Atomic Host を実行する仮想マシンを作成する方法を説明します。インストールプロセスを開始する前に、[Download Red Hat Enterprise Linux](#) ページからインストールメディアをダウンロードするようにしてください。Red Hat が提供する VHD イメージは、生成1 Hyper-V 仮想マシンを迅速にデプロイするために使用できる事前デプロイディスクイメージです。または、Red Hat Enterprise Linux Atomic Host ISO インストーラーを使用することもできます。これにより、カスタマイズされたインストールが可能になります。

Microsoft Hyper-V の全ドキュメントは、Microsoft TechNet Library の [Hyper-V Getting Started](#) セクションを参照してください。

#### Hyper-V での仮想マシンの作成

1. **Actions** メニューで、**New** を選択します。次に、ドロップダウンメニューから **Virtual Machine** を選択し、**Next** をクリックします。**New Virtual Machine Wizard** という名前の新しいダイアログウィンドウが開きます。
2. を開始する前に、**を開始** します。**Next** をクリックします。
3. **Name** および **Location** を指定します。新しい仮想マシンに名前を付け、**Next** をクリックします。
4. **Generation** を指定します。Red Hat が提供する VHD ディスクイメージを使用する場合は **Generation 1** を指定し、必要な場合は **Generation 2** を指定します。（世代1および **Generation 2** 仮想マシンの情報については、**第2 世代1と生成2のの違いのセクション**を参照してください。）
5. **Next** をクリックして先に進みます。
6. **メモリー** を割り当てます。仮想マシンに割り当てるメモリー容量を選択し、**Next** をクリックします。
7. **ネットワークの設定**。**接続** ドロップダウンメニューで、**外部** を選択します。**Next** をクリックします。
8. **仮想ハードディスクを接続** します。Red Hat が提供する VHD ディスクイメージを使用している場合は、**Use an existing virtual hard disk** を選択し、Red Hat カスタマーポータルからダウンロードした VHD ファイルの場所を指定します。**Next** をクリックします。
9. **概要** 選択内容を確認し、**Finish** をクリックして仮想マシンを作成します。

#### インストールの準備

Hyper-V イメージを実行すると、ログイン認証情報が要求されます。これらは、cloud-init ファイルのペアを使用して事前設定でき、ファイルを使用して他のインストール設定オプションを設定することもできます。手順の例を以下に示します。

- **meta-data**

インストールされている Red Hat Enterprise Linux Atomic Host のインスタンスを識別する情報を提供するプレーンテキストファイル。そのコンテンツは以下の例のようになります。

```
instance-id: Atomic0
local-hostname: atomic-00
```

**instance-id** には任意の識別名を指定でき、**local-hostname** はサイトの標準に準拠するホスト名である必要があります。

- **user-data**

システム上のユーザーに関する情報を提供するプレーンテキストファイル。この情報は、Red Hat Enterprise Linux Atomic Host インスタンスへのアクセスを有効にするために使用されます。デフォルトでは、**root** ユーザーはパスワードで保護されるため、**ユーザーデータ** ファイルを作成しないとログインできなくなります。

**ユーザーデータファイルの例を** 以下に示します。

```
#cloud-config
password: atomic
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
- ssh-rsa AAA...SDvz user1@yourdomain.com
- ssh-rsa AAB...QTuo user2@yourdomain.com
```



### 注記

例の最初の行(**#cloud-config**)はコメントやコマンドの例ではありません。これは、設定ファイルの必須の行です。

この例では、**cloud-user** ユーザーがパスワードまたは **SSH** キーのいずれかでログインできるようになります。両方の方法を使用できますが、必須ではありません。初期パスワードは **パスワード** 行に設定されます。ユーザーがこのインスタンスで初めてログインすると、**chpasswd** 行で定義されたパスワードを変更するように求められます。最初にパスワードがプレーンテキストで保存されるため、初回ログイン後にユーザーにパスワードの変更を強制することが推奨されます。

この例の最後の4行は、**SSH** を使用してリモートログインを設定します。**ssh\_pwauth: True** 行はパスワードを使用して **SSH** を有効にし、**ssh\_authorized\_keys** は1つ以上の認証済み公開鍵のブロックを開始します。このファイルで説明されているキーは `~/.ssh/authorized_keys` ファイルに追加されます。各承認キーは別々の行に指定し、2つのスペースの後にハイフン(-)と別のスペースが続きます。

上記の両方のファイルを作成したら、それらを ISO イメージにパッケージ化する必要があります。このイメージは、仮想マシンの仮想設定 CD として使用されます。ファイルをイメージにパッケージ化するには、以下のコマンドを使用します。

```
# genisoimage -output atomic0-cidata.iso -volid cidata -joliet -rock user-data meta-data
```

これにより、**atomic0-cidata.iso** という名前の新しい ISO イメージファイルが作成されます。

### 世代1と生成2の違い

Microsoft Hyper-V には、世代1と Generation 2 の2つの異なる **世代** (モードとも呼ばれます)があります。これらの生成の相違点は、Red Hat Enterprise Linux Atomic Host のインストールプロセスに影響します。

Generation 1 のディスクイメージは、すべての Microsoft Hyper-V ホストでサポートされます。世代 2 のディスクイメージは、Microsoft Windows 2012 および Microsoft Windows 8.1 でのみサポートされません。

Red Hat が提供するイメージは Generation 1 カテゴリーに分類されます。これらのディスクイメージを使用すると、セクション 25.5.1, "Creating a Virtual Machine in Hyper-V" で説明されているように、Red Hat Enterprise Linux Atomic Host の事前設定されたインスタンスをすぐにデプロイできます。

事前に設定された Generation 2 ディスクイメージは Red Hat では提供されません。Red Hat Enterprise Linux Atomic Host を Generation 2 仮想マシンとしてデプロイする場合は、対話型インストーラーの ISO イメージを使用し、Anaconda を使用してインストールを実行できます（手動またはキックスタートファイルを使用した自動のいずれか）。このプロセスについては、第 6 章の第 6 章の Anaconda を使用したインストールから始めており、キックスタートインストールについては第 23 章キックスタートを使ったインストールで説明されています。

### 2.3.6. Microsoft Azure インストール

以下の手順を使用して、RHEL Atomic Host イメージを Microsoft Azure Cloud にアップロードし、そのイメージを仮想マシンとして実行します。RHEL サーバーシステムから実行する基本的な手順は次のとおりです。

- Azure CLI の [インストール](#) で説明されているように、Azure CLI ツール(az コマンド)を取得します。
- [Red Hat Atomic Host Download](#) ページから Red Hat Atomic Cloud (qcow2) イメージを取得します。
- イメージを VHD 形式に変換します。
- Azure ログインアカウントを取得してログインします。
- 以下の Azure リソースを作成します（または既存のリソースを使用します）。
  - ストレージアカウントの作成
  - コンテナを作成します。
  - 仮想ネットワークおよびサブネットワークの作成
- Atomic VHD イメージをアップロードします。
- ゴールドイメージを作成します（オプション）。
- RHEL Atomic VM を起動します。
- Azure エージェントを仮想マシンに追加します（オプション）。

以下の例で使用されている以下のリソース名を、実際の設定に適したものに置き換えます。

リソース名	例
Azure グループ	myazgroup
Azure ストレージ	myatomicstorage

リソース名	例
Azure コンテナ	myatomiccontainer
Azure 仮想ネットワーク	myazatomicnet01
Azure subnetwork	myazatomicsubnet01
Atomic イメージ	rhel-atomic-cloud-7.4.vhd
Azure リージョン	southcentralus (ユーザーに適した地域を使用)
Azure のアトミックイメージ	atomiccloud-74.vhd
Azure ゴールドイメージグループ	myatomicgold

Azure アカウントで、以下の手順を使用し、そのイメージを使用して Azure に Atomic 仮想マシンを作成します。

1. Azure CLI ツールの取得 : [Install the Azure CLI](#) の手順に従って `az` コマンドを取得します。
2. [Red Hat Atomic Host Download](#) ページから Red Hat Atomic Cloud (qcow2) イメージを取得します。
3. 以下のように、イメージを VHD 形式に変換します。

```
$ qemu-img convert -f qcow2 -o subformat=fixed,force_size -O vpc \
  rhel-atomic-cloud-7.4.4-2.x86_64.qcow2 rhel-atomic-cloud-7.4.vhd
```



### 注記

Azure では、VHD イメージを修正して調整する必要があります。ここで説明されているイメージは正常に動作するはずですが、後のステップでイメージをアップロードして実行できない場合は、[RHEL 仮想マシンイメージの VHD への変換の説明に従って、そのアライメントを確認し、修正](#)します。

4. Azure Cloud にログインします。

```
$ az login
```

To sign in, use a web browser to open the page <https://aka.ms/devicelogin> and enter the code ABCDEFGH9 to authenticate.

```
[
  {
    "cloudName": "AzureCloud",
    ...
    "user": {
      "name": "joe@example.com",
      "type": "user"
    }
  }
]
```

```

    }
  }
]

```

ブラウザからの指示に従い、ブラウザウィンドウを閉じ、コマンドラインから続行します。

5. Azure グループリソースの作成 : Azure グループがまだない場合には、以下のように作成します。

```

$ az group create --name myazgroup --location southcentralus

{
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx.../resourceGroups/myazgroup",
  "location": "southcentralus",
  "managedBy": null,
  "name": "myazgroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}

```

6. 適切な Azure リージョンを選択します。 [Microsoft Azure リージョン](#) を参照して利用可能なリージョンを表示し、以下を入力してリージョンを特定するために必要な名前を表示します。

```

$ az account list-locations -o table
DisplayName  Latitude  Longitude  Name
-----
...
South Central US 29.4167  -98.5    southcentralus
...

```

7. Azure ストレージアカウントの作成 : グループの場合は、ストレージアカウントを作成し、southcentralus をリージョンに置き換え、 [SKU タイプ](#) を選択します。

```

$ az storage account create -l southcentralus -n myatomicstorage \
-g myazgroup --sku Standard_LRS

{
  "accessTier": null,
  "creationTime": "2018-01-23T16:14:51.478598+00:00",
  ...
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx.../resourceGroups/myazgroup/providers/Microsoft.Storage/storageAccounts/myatomicstorage",
  "name": "myatomicstorage",
  ...
  "provisioningState": "Succeeded",
  "resourceGroup": "myazgroup",
  ...
}

```

8. ストレージアカウントの接続文字列を取得します。

```

$ az storage account show-connection-string -n myatomicstorage -g myazgroup

```

```
{
  "connectionString":
  "DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=myatomicsto
  age;AccountKey=xxxxxxxx/xxxxx+xxx/w=="
}
```

9. 接続文字列をエクスポートします。接続文字列を `AZURE_STORAGE_CONNECTION_STRING` 変数にコピーアンドペーストします。

```
$ export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;EndpointSuffi
x=core.windows.net;AccountName=myatomicstorage;AccountKey=xxxxxxxx/xxxxx+xxx/w=="
```

10. ストレージコンテナを作成します。

```
$ az storage container create -n myatomiccontainer

{
  "created": true
}
```

11. 仮想ネットワークとサブネットワークを作成します。

```
$ az network vnet create -g myazgroup -n myazatomicnet01 \
  --subnet-name myazatomicsubnet01

{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
      ]
    }
    ...
    "id": "/subscriptions/xxxxxxxx-
    xxxx.../resourceGroups/myazgroup/providers/Microsoft.Network/virtualNetworks/myazatomicnet
    01",
    ...
  }
}
```

12. Atomic VHD イメージをアップロードします。

```
$ az storage blob upload --account-name myatomicstorage \
  --container-name myatomiccontainer --type page \
  --file rhel-atomic-cloud-7.4.vhd --name myatomiccloud-74.vhd

Finished[#####] 100.0000%
{
  "etag": "\"0x8D123456789ABCD\"",
  "lastModified": "2018-01-25T16:30:41+00:00"
}
```

13. アップロードした VHD の URL を取得します。



```
$ az storage blob url -c myatomiccontainer -n myatomiccloud-74.vhd
```

```
"https://myatomicstorage.blob.core.windows.net/myatomiccontainer/myatomiccloud-74.vhd"
```

14. ゴールド Azure カスタム Atomic イメージの新規リソースグループを作成します (オプション) : このオプションの手順では、作成した永続以外のリソースとは別のゴールドイメージを維持します。新しいリソースグループは、vhd ファイルをアップロードしたのと同じリージョンに作成する必要があります。

```
$ az group create --name myatomicgold --location southcentralus
```

```
{
  "id": "/subscriptions/xxxxxxxx-xxxx-.../resourceGroups/myatomicgold",
  "location": "southcentralus",
  "managedBy": null,
  "name": "myatomicgold",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

15. Azure 用のゴールドイメージを作成します。

```
$ az image create -n myrhelatomcloud74 -g myatomicgold -l southcentralus \
  --source \
  "https://myatomicstorage.blob.core.windows.net/myatomiccontainer/myatomiccloud-74.vhd" \
  --os-type linux
```

```
{
  "additionalProperties": {},
  "id": "/subscriptions/xxxxxxxx-xxxx-.../resourceGroups/myatomicgold/providers/.../images/myrhelatomcloud74",
  ...
  "additionalProperties": {},
  "blobUri":
  "https://myatomicstorage.blob.core.windows.net/myatomiccontainer/myatomiccloud-74.vhd",
  ...
}
```

16. 仮想マシンの作成 : この例では、myatomic74vm-1 という名前の実行中の仮想マシンを作成します。(注記 : cloud-init スクリプトを作成し、コマンドラインに追加して、このコマンドラインをさらに設定できます。例 : `--custom-data RHELCloudInit.yml` 詳細は、[Cloud-Init Support](#) を参照してください。)

```
$ az vm create -g myatomicgold -l southcentralus -n myatomic74vm-1 \
  --size Standard_A2 --os-disk-name vm-1-osdisk \
  --admin-username clouduser --generate-ssh-keys --image myrhelatomcloud74
```

```
{
  "fqdns": "",
  "id": "/subscriptions/xxxxxxxx-xxxx-.../resourceGroups/myatomicgold/providers/Microsoft.Compute/virtualMachines/myatomic74vm-1",
  ...
}
```

```
"powerState": "VM running",
"privateIpAddress": "10.0.0.5",
"publicIpAddress": "49.82.154.297",
"resourceGroup": "myatomicgold",
"zones": ""
}
```

17. 仮想マシンにログインします。publicIpAddress（この偽のアドレスの49.82.154.297）をメモし、これを使用して仮想マシンにログインします。

```
$ ssh clouduser@49.82.154.297
The authenticity of host '49.82.154.297 (49.82.154.297)' can't be established.
ECDSA key fingerprint is bd:fe:12:1b:3c:d3:e2:4c:9f:b5:4a:87:10:48:5d:92.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '49.82.154.297' (ECDSA) to the list of known hosts.
[clouduser@myatomic74vm-1 ~]$
```

18. システムをサブスクライブする : Red Hat サブスクリプションアカウントを使用して、システムをサブスクライブして最新バージョンのAtomicに更新します。

```
$ sudo subscription-manager register
Registering to: subscription.rhsm.redhat.com:443/subscription
Username: yourusername
Password: *****
The system has been registered with ID: e4da51cb-4b89-3c94-30b5-946e5c222e91

$ sudo subscription-manager subscribe --auto
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux Atomic Host
Status: Subscribed
Product Name: Red Hat Enterprise Linux Server
Status: Subscribed

$ atomic host upgrade

$ sudo reboot
```

19. Azure エージェントを追加します (オプション) : Azure Web ポータルでさらに高度なAzure機能および精度が必要な場合は、Azure エージェントをAtomicホストに追加することを検討してください。これを実行するには、以下のように、必要なリポジトリを有効にし、WALinuxAgentパッケージを追加し、プロビジョニングを無効にし、Azure エージェントが起動するように設定します。

```
$ ssh clouduser@41.89.184.287
[clouduser@myatomic74vm-1 ~]$

$ sudo subscription-manager repos --disable=*
$ sudo subscription-manager repos --enable rhel-7-server-rpms \
--enable rhel-7-server-extras-rpms
$ sudo rpm-ostree install WALinuxAgent
$ sudo systemctl reboot
$ sudo vi /etc/waagent.conf
# Enable instance creation
Provisioning.Enabled=n
# Create and use swapfile on resource disk.
```

```
ResourceDisk.EnableSwap=y
# Size of the swapfile.
ResourceDisk.SwapSizeMB=2048 <-- or choose different size

$ sudo systemctl enable waagent
$ sudo systemctl start waagent
```

これでAtomic 仮想マシンが実行され、Azure ダッシュボードからアクセスできるはずですが。

### 2.3.7. Google Compute Engine のインストール

Google Compute Engine (GCE) は、Google インフラストラクチャーで実行される仮想マシンを提供するサービスです。本書では、GCE でRed Hat Enterprise Linux Atomic Host を実行する方法を説明します。

Red Hat Enterprise Linux Atomic Host は、Docker エンジンを使用して実行されるLinux コンテナ向けに最適化されたRed Hat Enterprise Linux 7のバリエーションで、Red Hat Enterprise Linux 7で利用可能な強力な技術を利用するように設計されています。Google Compute Engine (GCE)は、Google インフラストラクチャーで実行される仮想マシンを提供するサービスです。これらの仮想マシンは、Red Hat Enterprise Linux Atomic Host の実行に使用できます。このセクションでは、GCE でRed Hat Enterprise Linux Atomic Host の仮想マシンインスタンスを起動する方法について説明します。

詳細は、[The official documentation for Google Compute Engine](#) を参照してください。

#### 2.3.7.1. Google Compute Engine の有効化

##### プロジェクトの作成と請求の設定

プロジェクトを作成し、Google Compute Engine の請求を設定するには、以下の手順を実施します。

1. Google アカウントにログインし、<https://console.developers.google.com/project> でGoogle Developers Console に移動します。Developers コンソールは、利用可能なプロジェクトの一覧を提供します。
2. 有効にするプロジェクトを選択します。新規プロジェクトを作成する場合は、赤色の **Create Project** ボタンをクリックします。プロジェクト名とID を選択するように求められます。プロジェクトが特定のドメインに属する場合、プロジェクトID は `<domain>:\<your-chosen-project-id>` の形式になります。次に、プロジェクトダッシュボードに移動します。
3. Google Compute Engine を有効にするには、右側のバーの **Billing & Settings** メニューアイテムをクリックして請求を設定します。次に、**Enable Billing** をクリックします。後で表示されるフォームに入力します。Google Compute Engine は、サービスを使用する前に請求の設定を求めるプロンプトを表示します。請求をアクティベートせずにGoogle Compute Engine を使用することはできません。アクティベート後、アカウントの準備ができるまでに約5分かかる場合があります。

##### GCE ツールのダウンロードおよび設定

Google Compute Engine リソースを管理するには、最初にgcloud コマンドラインツールをダウンロードしてインストールします。

1. 以下のコマンドを実行して、Google Cloud SDK をインストールします。

```
$ curl https://sdk.cloud.google.com | bash
```

2. インストール時に、必要な情報が複数回指定するよう求められます。まず、Google Cloud SDK の宛先ディレクトリーを指定するよう求められます。

```
Directory to extract under (this will create a directory google-cloud-sdk) (/home/user):
```

3. 次に、Google への使用状況の報告を許可するかどうか尋ねられ、このデータを使用してツールを改善できるかどうかを尋ねられます。
4. 次に、Google Cloud SDK がインストールされます。その後、プロファイルの設定を求めるプロンプトが複数あります。**rc** ファイルを指定し、**\$PATH** 変数を変更して bash 補完を有効にすることができます。これらのプログラムを **\$PATH** 変数に追加すると、完全パスを提供せずに実行できるためです。また、bash 補完を有効にすると、コマンドが補完で入力しやすい複数の引数で設定されているため便利です。
5. 端末を再起動して、PATH への変更が反映されるようにします。たとえば、以下を行うことができます。

```
$ source ~/.bash-profile-file
```

6. **bash-profile-file** は、bash プロファイルファイルへのパスに置き換えます。通常、これは **~/.bashrc** ファイル です。

## GCE への認証

以下を実行して Google Cloud プラットフォームに対して認証します。

```
$ gcloud auth login
```

上記のコマンドは、Google アカウントのサインアップダイアログを使用して Web ブラウザーを起動します。サインインして続行します。サインインプロセス中に、Google Compute Engine が Google アカウントに関する情報にアクセスできるようにする必要があります。**--no-launch-browser** オプションを使用すると、ブラウザーを起動しなくても認証を行うことができます。詳細は、<https://cloud.google.com/compute/docs/gcloud-compute/#auth> を参照してください。

## プロジェクトのデフォルトの設定

コマンドテンプレートを使用して、**gcloud config set default default\_value** を使用すると、一般的に使用されるフラグのコマンドオプションをすべてのコマンドに渡す必要がなくなります。現在のデフォルトを一覧表示するには、**gcloud config list** コマンドを実行します。テンプレート **gcloud config unset default** により、プロジェクトのデフォルトが削除されます。以下のコマンドを実行してデフォルトのプロジェクトを設定します。

```
$ gcloud config set project project_id
```

**project\_id** は、プロジェクトの作成および請求の設定で作成したプロジェクトの ID を表します。

以下のコマンドを実行してデフォルトゾーンを設定します。

```
$ gcloud config set compute/zone zone
```

ここで、**ゾーン** は、インスタンスを配置する地理的な場所を決定します。利用可能なゾーンの一覧は、<https://developers.google.com/compute/docs/zones#available> を参照してください。

### 2.3.7.2. Red Hat Enterprise Linux Atomic Host インスタンスの起動

Red Hat Enterprise Linux Atomic Host イメージを GCE で使用する前に、qcow2 ファイルから RAW イメージに変換する必要があります。これは、qcow2 ファイルをダウンロードして tar ファイルに変換することで行います。このファイルは GCE にアップロードされ、インスタンスが作成されます。

### Creating a Red Hat Enterprise Linux Atomic Host RAW File

GCE にアップロードできる RAW ファイルを作成するには、以下の手順を実行します。

1. Red Hat Enterprise Linux のダウンロード から Red Hat Enterprise Linux Atomic Host qcow2 ファイルを [ダウンロード](#) します。
2. qcow2 イメージが **xz** で圧縮されている。イメージを展開するには、以下のコマンドを入力します。

```
$ xz -d rhel-atomic-cloud-7.1-0.x86_64.qcow2.xz
```

1. GCE で使用するには、qcow2 イメージを RAW ディスクファイルに変換する必要があります。これは、`qemu` で行います。

```
$ qemu-img convert -S 4096 -f qcow2 -O raw rhel-atomic-cloud-7.1-0.x86_64.qcow2 disk.raw
```

1. raw ディスクファイルは、GCE にアップロードする前に tar でパッケージ化する必要があります。raw ファイルの名前は `disk.raw` である必要があります。

```
$ tar -Szc rhel-atomic-cloud-7.1-0.x86_64.tar.gz disk.raw
```

1. アップロードした raw ディスクファイルは Google Cloud Storage バケットに保存されます。バケットを作成していない場合は、**gsutil** を使用して作成できます。

```
$ gsutil mb gs://bucketname
```

1. **gsutil** を使用して raw ディスクファイルをアップロードします。

```
$ gsutil cp rhel-atomic-cloud-7.1-0.x86_64.tar.gz gs://bucketname
```

1. raw ディスクファイルを使用する前に、GCE イメージとして作成する必要があります。

```
$ gcloud compute images create GCE_IMAGE_NAME --source-uri gs://bucketname/rhel-atomic-cloud-7.1-0.x86_64.tar.gz
```

1. イメージをアップロードして利用できるようにするには、`gcloud compute images list` の出力でイメージを探します。

### Creating a Red Hat Enterprise Linux Atomic Host Instance

以下のコマンドを実行して Atomic Host インスタンスを作成します。

```
$ gcloud compute instances create my-atomic-instance --machine-type n1-standard-1 --image GCE_IMAGE_NAME --metadata-from-file startup-script=<your-startup-script>
```

ここでは、以下のようになります。

**my-atomic-instance** は、この例のインスタンスの名前です。インスタンス名には、小文字、数字、ダッシュ（ダッシュは指定できない最後の文字を除く）のみを含めることができます。

**--machine-type** は、必要なマシンタイプを設定します。マシンタイプは、インスタンスのメモリ、仮想コア数、および永続ディスクの制限を決定します。詳細

は、<https://developers.google.com/compute/docs/machine-types> を参照してください。

**--image** は、使用するイメージを設定します。イメージには、インスタンスの起動に必要なオペレーティングシステムとルートファイルシステムが含まれます。GCE は root 永続ディスクを自動的に作成し、ルートファイルシステムを保存します。GCE\_IMAGE\_NAME は、直前の手順で作成したイメージです。

**--metadata-from-file** は、ローカルのメタデータサーバーを介してインスタンス環境で利用可能になるメタデータを指定します。このフラグを使用して、Red Hat Enterprise Linux Atomic Host インスタンスの初回起動時に自動的に実行されるスクリプトを指定します。詳細は、インスタンス作成時のカスタムスクリプトの実行セクションを参照してください。Red Hat Enterprise Linux Atomic Host インスタンスの起動スクリプトは GCE エージェントではなく **cloud-init** ユーティリティーで処理されるため、**user-data** キーが必要で、カスタムキーに置き換えることができないことに注意してください。



### 注記

このコマンドは、インスタンスが実行されるまでブロックします。インスタンスが最初に作成されると、インスタンスを起動してから自己設定する必要があります。これにはしばらく時間がかかり、インスタンスへのログインが遅延する可能性があります。

### インスタンス作成時のカスタムスクリプトの実行

上記のように、**--metadata-from-file** オプションを使用して、インスタンスの初回起動時にそのインスタンスで実行するカスタムスクリプトを指定できます。これらのコマンドは root 権限で実行されるため、このスクリプトで必要なシステムコマンドを実行できます。以下に例を示します。

```
--metadata-from-file startup-script=<your-startup-script>
```

以下の内容で `startup.sh` スクリプトを実行します。

```
#!/bin/sh
touch newfile
```

この行は、**newfile** という名前の新しいファイルを作成します。



### 注記

Red Hat Enterprise Linux Atomic Host インスタンスの起動スクリプトは GCE エージェントではなく、**cloud-init** ユーティリティーにより処理されます。したがって、**--metadata-from-file** でカスタムキーを使用することはできません。Red Hat Enterprise Linux Atomic Host インスタンスのカスタムスクリプトを設定する場合は、常に **user-data** キーを使用してください。

ローカルに保存された起動スクリプトの代わりに、スクリプトを Google Cloud Storage にアップロードし、**--metadata** オプションを使用してこれにアクセスできます。これは、スクリプトがメタデータ値の長さ制限である 32,768 バイトを超える場合に必要です。詳細

は、<http://developers.google.com/compute/docs/howtos/startupscript#googlecloudstorage> を参照してください。

### 2.3.7.3. Red Hat Enterprise Linux Atomic Host インスタンスへのログイン



`gcloud` ツールには、インスタンス名を使用してインスタンスにログインできる `ssh` コマンドが組み込まれています。

インスタンスにログインするには、以下のコマンドを実行します。

```
$ gcloud compute ssh cloud-user@my-atomic-instance
```

ここで、**cloud-user** はデフォルトのユーザー名になります。SSH キーを作成していない場合は、作成するように求められます。詳細は、[Password Protecting Your SSH Keys](#) を参照してください。



### 注記

セキュリティ上の理由から、標準の Google イメージは SSH を使用して `root` として直接接続することができません。インスタンス作成者および `--authorized_ssh_keys` フラグを使用して追加されたユーザー、またはメタデータの `sshKeys` 値はアカウントに自動的に管理者となり、パスワードなしに `sudo` を実行できます。これは推奨されていませんが、上級ユーザーは `/etc/ssh/sshd_config` を変更し、`sshd` を再起動してこのポリシーを変更できます。



### 警告

GNOME ユーザーにメッセージが表示されることがあります。

+

```
Agent admitted failure to sign using the key
```

+ GCE インスタンスへの接続を試みる際に SSH をトークします。これは、誤った SSH キーの使用を試みる GNOME キーリング管理が原因です。これは、GCE 環境の `rhel-atomic-host-20141111` イメージに固有のものであります。

+ この問題を回避するには、`gcutil` を実行する前に以下のコマンドを入力します。

+

```
$ ssh-add ~/.ssh/google_compute_engine
```

ログインしたら、他の Red Hat Enterprise Linux マシンと同じように作業できます。インスタンスに `root` パーミッションがあり、すべてを完全に制御している。`root` になるには、以下を実行します。

```
cloud-user@my-atomic-instance$ sudo -i
```

インスタンスからログアウトする必要がある場合は、以下のコマンドを実行します。

```
cloud-user@my-atomic-instance$ exit
```

Red Hat Enterprise Linux Atomic Host をインストールすると、Linux コンテナを実行する準備が整います。

## SSH 鍵のパスワード保護

SSH を使用してインスタンスに初めてログインすると、gcloud はローカルマシンに ssh 公開鍵/秘密鍵のペアを作成し、公開鍵をプロジェクトにコピーします。これらのキーは、ssh を使用してインスタンスにログインするために必要です。これらの鍵を初めて作成すると、gcutil によりパスワードの入力と確認が求められます。

```
WARNING: You don't have an ssh key for Google Compute Engine. Creating one now...
Enter passphrase (empty for no passphrase):
```

パスワードは空欄のままにしておくことはできますが、SSH 鍵を保護するためのパスワードを入力することを強く推奨します。パスワードの入力を要求することはほとんどなく、これらの鍵をパスワードで保護しない場合は、悪意のあるユーザーがその鍵を使用してインスタンスにアクセスできます。

### 2.3.7.4. Red Hat Enterprise Linux Atomic Host インスタンスのモニタリング

Google Cloud SDK は、インスタンスのパラメーターを監視する複数の方法を提供します。現在の gcloud 環境に関する一般的な情報を表示するには、以下のコマンドを実行します。

```
$ gcloud info
```

**describe** コマンドを実行して、特定のインスタンスに関する詳細情報を検索します。

```
$ gcloud compute instances describe my-atomic-instance
canIpForward: false
creationTimestamp: '2014-11-11T02:15:58.372-08:00'
disks:
- autoDelete: true
  boot: true
  deviceName: persistent-disk-0
  index: 0
  interface: SCSI
  kind: compute#attachedDisk
  mode: READ_WRITE
  source: https://www.googleapis.com/compute/v1/projects/eighth-saga-761/zones/europe-west1-b/disks/my-atomic-instance2
  type: PERSISTENT
  id: '6632858316955862880'
kind: compute#instance
machineType: https://www.googleapis.com/compute/v1/projects/eighth-saga-761/zones/europe-west1-b/machineTypes/n1-standard-1
metadata:
  fingerprint: owFsCDPRikY=
  kind: compute#metadata
  name: my-atomic-instance2
networkInterfaces:
- accessConfigs:
- kind: compute#accessConfig
  name: external-nat
  natIP: 23.251.142.75
  type: ONE_TO_ONE_NAT
  name: nic0
  network: https://www.googleapis.com/compute/v1/projects/eighth-saga-761/global/networks/default
  networkIP: 10.240.184.150
```



```
scheduling:
  automaticRestart: true
  onHostMaintenance: MIGRATE
selfLink: https://www.googleapis.com/compute/v1/projects/eighth-saga-761/zones/europe-west1-b/instances/my-atomic-instance2
serviceAccounts:
- email: 464767924601-compute@developer.gserviceaccount.com
  scopes:
  - https://www.googleapis.com/auth/devstorage.read_only
status: RUNNING
tags:
  fingerprint: 42WmSpB8rSM=
zone: https://www.googleapis.com/compute/v1/projects/eighth-saga-761/zones/europe-west1-b
```

Red Hat Enterprise Linux Atomic Host インスタンスのシリアルポートからデータを取得するには、以下を実行します。

```
$ gcloud compute instances get-serial-port-output my-atomic-instance
```

このコマンドは、GCE インスタンスのシリアルポートの出力を返します。このコマンドにより、ログインせずにインスタンスに関する情報を取得できます。これは診断の目的で役立ちます。

### インスタンスの外部IPアドレスの検索

デフォルトでは、インスタンスには新規の一時外部IPアドレスが割り当てられます。このアドレスと、上記の **gcutil getinstance** の出力で、他の情報を確認できます。または、以下のコマンドを入力してすべてのインスタンスのアドレスを取得することもできます。

```
$ gcloud compute instances list
NAME          ZONE          MACHINE_TYPE  INTERNAL_IP  EXTERNAL_IP  STATUS
my-atomic-instance  us-central1-a  n1-standard-1  10.240.184.150  23.251.142.75  RUNNING
```

### 2.3.7.5. ファイアウォールルールの作成

デフォルトでは、Google Compute Engine はインスタンス間の接続をすべてブロックし、インスタンスからインターネットへの接続をすべてブロックします。**httpd** などのサービスのポートを開くには、ファイアウォールルールを手動で作成する必要があります。すべてのプロジェクトには、デフォルトの3つのファイアウォールがあります。

1. 任意のインスタンスへのSSHアクセスを可能にするファイアウォール。
2. 同じネットワーク内のインスタンス間のすべての通信を許可するファイアウォール。
3. 任意のソースからネットワーク上の任意のインスタンスへのICMPトラフィックを許可するファイアウォール。

たとえば、インスタンスへのHTTP要求を許可するには、以下の **gcutil** コマンドを使用して新しいファイアウォールを作成します。

```
$ gcloud compute firewall-rules create http-allow --allow tcp:80
```

上記のコマンドを実行すると、以下が可能になります。

1. ポート80のtcpトラフィックを許可する **http-allow** という名前の新しいファイアウォールを作成しました。

2. プロジェクトのデフォルトネットワークにファイアウォールを割り当てました。
3. ネットワーク内外のすべてのソース（インターネット経由を含む）がサーバーへの要求を行うことを許可。ファイアウォールに許可されたソースを指定していないため、すべてのソースはデフォルトのネットワークに割り当てられたインスタンスへの要求を行うことができます。
4. このファイアウォールルールをネットワーク上のすべてのインスタンスに適用します。ファイアウォールにターゲットを指定していないため、ファイアウォールはネットワーク内のすべてのインスタンスにこのルールを適用します。

ファイアウォールに関する情報を確認するには、次のコマンドを実行します。

```
$ gcloud compute firewall-rules list
NAME                NETWORK SRC_RANGES  RULES                               SRC_TAGS TARGET_TAGS
default-allow-icmp  default 0.0.0.0/0  icmp
default-allow-internal default 10.240.0.0/16 tcp:1-65535,udp:1-65535,icmp
default-allow-rdp   default 0.0.0.0/0  tcp:3389
default-allow-ssh   default 0.0.0.0/0  tcp:22
http-allow          default 0.0.0.0/0  tcp:80
```

適切な **addfirewall** フラグを使用して、ソースとターゲットを特定の呼び出し元およびインスタンスに制限することができます。サポートされるフラグの完全な一覧を表示するには、**gcutil help addfirewall** コマンドを実行するか、<https://cloud.google.com/sdk/gcloud/reference/compute/firewall-rules/> を参照してください。

ファイアウォールは、インスタンスへの受信トラフィックのみを調整します。送信パケットをブロックできません。インスタンスで接続を確立すると、トラフィックはその接続上の両方で許可されます。インスタンスが送信パケットを送信できないようにするには、**iptables** 等の別のテクノロジーを使用します。



### 注記

デフォルトでは、GCE は 10 分間非アクティブになると、インスタンスへの TCP 接続を切断します。これを防ぐには、<https://developers.google.com/compute/docs/troubleshooting#communicatewithinterr> 説明に従って TCP キープアライブを設定します。

### 2.3.7.6. Red Hat Enterprise Linux Atomic Host インスタンスの削除

以下のコマンドを実行して **my-atomic-instance** を削除します。

```
$ gcloud compute instances delete my-atomic-instance
```

インスタンスを削除する前に、決定を確認するように求められます。インスタンスの削除には数秒かかる場合があります。削除の確認ダイアログの一部として、gcloud は、別のインスタンスによっても使用されていない限り、ディスクが削除されることを通知します。

### 2.3.8. Amazon Web Services のインストール

Amazon Web Services (AWS) は、Amazon インフラストラクチャーで実行される仮想マシンを提供するサービスです。このセクションでは、AWS で Red Hat Enterprise Linux Atomic Host を実行する方法を説明します。

AWS の詳細は、[Amazon Elastic Compute Cloud のドキュメント](#) を参照してください。

## Amazon Web Services での Red Hat Enterprise Linux Atomic Host インスタンスの起動

以下の手順では、Amazon Web Services で Red Hat Enterprise Linux Atomic Host の新しいインスタンスを作成する方法を説明します。この手順では、AWS にユーザーアカウントがあることを前提としています。この手順では、AWS についてある程度理解していることを前提としています。



### 注記

この手順を機能させるには、まず Cloud Access プログラム を使用してサブスクリプションを Amazon に移動する必要があります。Cloud Access Program でサブスクリプションを Amazon に移動するには、<https://engage.redhat.com/forms/cloud-access-registration> のフォームを完了します。Cloud Access プログラムについては、<http://www.redhat.com/en/technologies/cloud-computing/cloud-access> で詳しく説明されています。

1. にログインし、[Amazon EC2 コンソール](#) を開きます。
2. 画面上部のナビゲーションバーに、現在のリージョンが表示されます。Red Hat Enterprise Linux Atomic Host のインスタンスを起動するリージョンを選択します。一部の Amazon EC2 リソースはリージョン間で共有でき、他の Amazon EC2 リソースはリージョン間で共有できないため、この選択が重要になります。
3. コンソールのダッシュボードから **Launch Instance** をクリックします。
4. **My AMIs** を選択し、**Shared with Me** チェックボックスを選択します。AMI を検索できるようになりました。  
**Community AMIs** を選択し、特定のリージョンの Red Hat Enterprise Atomic Host AMI インスタンスを検索します。



### 警告

選択した AMI の ID が [Atomic Host Release Notes](#) に一覧表示されていることを確認してください。AWS コマンドラインインターフェイスで以下のコマンドを実行して、Red Hat が提供する AMI の ID を取得することもできます。

```
aws ec2 describe-images --owners 309956199498
```

このコマンドは、アカウント **309956199498** によって公開される AMI に関する情報を表示します。これは、AMI を公開するための Red Hat の AWS アカウントです。

Red Hat が提供する AMI の検索に関する詳細は、[ナレッジベースア티クル](#) を参照してください。

5. AMI の横にある **Select** ボタンをクリックします。
6. **Choose an Instance Type** ページで Instance Type を選択します。インスタンスタイプは、Red Hat Enterprise Linux Atomic Host の最小要件を満たす必要があります。詳細は、[ディスク領域およびメモリの要件](#) を参照してください。

7. **Review and Launch** をクリックします。**注記**

一部の Amazon EC2 リージョン（例：US East (N. Virginia)）では、EBS ストレージを使用するインスタンスタイプでは、開始する前に VPC を作成する必要があります。この場合、**Review and Launch** はクリックできません。**Next: Configure Instance Details** を代わりにクリックし、Instance Details 画面に進みます。デフォルト値を確認し、環境に必要な場合は変更し、続行する準備ができたなら **Review and Launch** をクリックします。

8. **Review Instance Launch** ページで、**Edit security groups** をクリックしてセキュリティグループを割り当てます。既存のセキュリティグループを選択するか、インスタンスに必要なポートを開く必要があります。SSH が機能するように、ポート 22 を開いたままにしておくことをお勧めします。AWS アカウントは、そのアカウントのユーザーのユーザーがセキュリティグループを作成または追加する機能を制限する方法で設定できます。これが発生した場合は、AWS アカウントの管理者にお問い合わせください。
9. 設定が適切であれば、**Review and Launch** をクリックして、**インスタンス起動の確認** ページに移動します。すべての設定が適切になったら、**起動** をクリックしてインスタンスを起動します。
10. **Select an existing key pair or create a new key pair** モーダルダイアログで、既存のキーペアを選択するか、新しいキーペアを作成します。起動したインスタンスへのすべてのアクセスは SSH 秘密鍵を使用するので、キーペアは重要です。キーペアは、すでにアップロードされているものか、この時点で作成するキーペアです。AWS アカウントは、そのアカウントのユーザーがキーペアを作成または追加する機能を制限する方法で設定できます。これが発生した場合は、AWS アカウントの管理者にお問い合わせください。
11. インスタンスの **表示 ボタン** をクリックして、インスタンスの起動の進捗を追跡します。

**Logging into a Red Hat Enterprise Linux Atomic Host Instance**

インスタンスがの **実行** 時に一覧表示されたら、以下の手順に従ってインスタンスに接続できます。

1. コマンドプロンプトから、SSH を使用してインスタンスに接続します。

```
$ ssh cloud-user@instancedns.compute.amazonaws.com
```

**注記**

適切な秘密鍵ファイルを指定するには、**-i /path/key\_pair.pem** オプションを含める必要がある場合があります。

2. 下部の **Description** タブで、**パブリック DNS** 情報を見つけます。
3. **Instances** ページで、インスタンスを選択します。
4. この時点でインスタンスにログインし、Red Hat Enterprise Linux Atomic Host を引き続き使用して Linux コンテナを実行する可能性があります。

**AWS での Atomic Host インスタンスの信頼性の確認**

Atomic Host インスタンスが Red Hat が提供する認証ソフトウェアであることを確認できます。これを実行するには、Atomic Host インスタンスで以下のコマンドを実行します。

```
ostree show rhel-atomic-host/7/x86_64/standard
```

出力の最後の行が以下の場合：

```
Good signature from "Red Hat, Inc. <security@redhat.com>"
```

次に、Atomic Host インスタンスが検証に合格しました。

## 2.4. PXE インストール

PXE サーバーを Red Hat Enterprise Linux Atomic Host から起動するように設定することは、Red Hat Enterprise Linux の標準手順とは異なります。Red Hat Enterprise Linux インストールガイドの [ネットワークインストールの準備](#) の章の詳細な手順を使用できます。

以下は、`/var/lib/tftpboot/pxelinux/pxelinux.cfg/default` ファイルの Atomic のエントリーの例です。

```
label Atomic-7.3
  menu label ^1. RHEL Atomic Host 7.3 kickstart
  kernel atomic7.3/vmlinuz
  append initrd=atomic7.3/initrd.img inst.stage2=http://192.168.122.1/distros/atomic xdriver=vesa
  nomodeset quiet ks=http://192.168.122.1/ks/atomic.ks
```

カーネル、不要なイメージ、インストールプログラムのランタイムイメージ(`inst.stage2`)、およびキックスタートファイルが指定される場所に存在することを確認します。

## 第3章 CLOUD-INIT の設定

Red Hat Enterprise Linux Atomic Host は cloud-init を使用して、インストールおよび初回起動時にシステムを設定します。cloud-init は、クラウドインスタンスの初期初期化を提供するために最初に開発されました。Red Hat Enterprise Linux Atomic Host では、仮想マシンのインストールにも使用できます。

cloud-init が使用するファイルは YAML 形式のファイルです。



### 注記

cloud-init は、マシンの初回起動時にのみ実行されます。ファイルの構文エラーが原因で cloud-init が失敗した場合や、ユーザー認証情報などの必要なディレクティブがすべて含まれていない場合は、新規インスタンスを作成し、起動する必要があります。新しい cloud-init ファイルで障害が発生したインスタンスの再起動は機能し**ません**。

以下は、cloud-init で一般的なタスクを実行する例です。

- cloud-init でユーザーを作成する方法

cloud-init を使用してユーザーを作成するには、`meta-data` と `user-data` の2つのファイルを作成してから、ISO イメージにパッケージ化する必要があります。

- ディレクトリーを作成して、そのディレクトリーに移動します。

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

- `meta-data` という名前のファイルを作成します。`meta-data` という名前のファイルに以下を追加します。

```
instance-id: Atomic0
local-hostname: atomic-00
```

- `user-data` という名前のファイルを作成します。`user-data` という名前のファイルに以下を追加します。

```
#cloud-config
password: atomic
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...SDvZ user1@domain.com
```

注記：上記の `user-data` ファイルの最後の行は、SSH 公開鍵です。SSH 公開鍵は `~/.ssh/id_rsa.pub` にあります。

- メタ データおよびユーザーデータを含む ISO イメージを作成します。

```
# genisoimage -output atomic0cidata.iso -volid cidata -joliet -rock user-data meta-data
```

- `atomic0cidata.iso` という名前のファイルが生成されます。このファイルを Red Hat Enterprise Linux Atomic Host をインストールする予定のマシンにアタッチします。ユーザー名は `cloud-user` になり、パスワードは `atomic` になります。

- cloud-user のパスワードを失効させ、ユーザーが初回ログイン時に変更する必要はありますか？

最初のログイン時に cloud-user を強制的にパスワードを変更するには、**user-data** ファイルで **chpasswd: {expire: False}** の行を **chpasswd: {expire: True}** に変更します。

```
#cloud-config
password: atomic
chpasswd: {expire: True}
ssh_pwauth: True
ssh_authorized_keys:
- ssh-rsa AAA...SDvz user1@yourdomain.com
- ssh-rsa AAB...QTuo user2@yourdomain.com
```

これは、特に指定がない限り、パスワードと chpasswd がデフォルトのユーザーで動作するため機能します。

注記：これはグローバル設定です。これを True に設定すると（以下を参照）、作成したすべてのユーザーにはパスワードを変更する必要があります。

- デフォルトのユーザー名を変更するにはどうすればよいですか？  
デフォルトのユーザー名を cloud-user から他のものに変更するには、**user: username** の行を **user-data** ファイルに追加します。

```
#cloud-config
user: username
password: atomic
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
- ssh-rsa AAA...SDvz user1@yourdomain.com
- ssh-rsa AAB...QTuo user2@yourdomain.com
```

- root パスワードの設定方法  
root パスワードを設定するには、ユーザーデータファイルの **chpasswd** セクションにユーザー一覧を作成する必要があります。リストの形式を以下に示します。空白は重要であるため、コロン(:)の両側に何も含めないでください。スペースでパスワードを設定するためです。この方法を使用してユーザーパスワードを設定する場合は、本セクションですべてのパスワードを設定する必要があります。つまり、**password:** 行は上部からこのセクションに移動する必要があります。

```
#cloud-config
ssh_pwauth: True
ssh_authorized_keys:
- ssh-rsa AAA...SDvz user1@yourdomain.com
- ssh-rsa AAB...QTuo user2@yourdomain.com
chpasswd:
list: |
  root:password
  cloud-user:atomic
expire: False
```

- How do I manage Red Hat subscriptions with cloud-init?  
**rh\_subscription** ディレクティブは、システムの登録に関するさまざまな操作を実行できます (RHEL Atomic 7.4 以降の場合)。以下は、利用可能なオプションのさまざまな例になります。

-

```
rh_subscription:
  username: atomic@redhat.com
  password: '<password>'
  auto-attach: True
  service-level: self-support
```

`service-level` は、`auto-attach` オプションでのみ使用されます。または、ユーザー名とパスワードの代わりにアクティベーションキーと `org` を使用できます。

```
rh_subscription:
  activation-key: example_key
  org: 12345
  auto-attach: True
```

プールの追加もサポートされています。以下は、**`subscription-manager attach --pool=XYZ01234567`** コマンドと同じです。

```
rh_subscription:
  username: atomic@redhat.com
  password: '<password>'
  add-pool: XYZ01234567
```

以下を使用して、`/etc/rhsm/rhsm.conf` にサーバーのホスト名を設定できます。

```
rh_subscription:
  username: atomic@redhat.com
  password: '<password>'
  server-hostname: atomic.example.com
  auto-attach: True
```

- 初期システム設定中にユーザーを追加するにはどうすればよいですか?追加のユーザーオプションの設定方法  
ユーザーが作成され、`user-data` ファイルの `users` セクションで説明されています。このセクションを追加するには、デフォルトのユーザーのオプションもここで設定する必要があります。

`users` セクションの最初のエントリがデフォルトの場合、デフォルトのユーザーである `cloud-user` が他のユーザーとともに作成されます。デフォルトの行を省略すると、`cloud-user` は作成されません。

```
#cloud-config
users:
  - default
  - name: foobar
    gecol: User N. Ame
    selinux-user: staff_u
    groups: users,wheel
    ssh_pwauth: True
    ssh_authorized_keys:
      - ssh-rsa AA..vz user@domain.com
chpasswd:
  list: |
    root:password
```



```
cloud-user:atomic
foobar:foobar
expire: False
```

注記：デフォルトでは、`se-linux-user` 値がない場合、ユーザーには `unconfined_u` というラベルが付けられます。

注記：この例では、ユーザー `foobar` を `users` と `wheel` の2つのグループに配置します。cloud-init 0.7.5 の時点で、グループリストでは空白はサポートされません。BZ 1126365

- 最初の起動コマンドを実行するにはどうすればよいですか？

`user-data` ファイルの `runcmd` セクションおよび `bootcmd` セクションを使用して、起動および初期化中に任意のコマンドを実行できます。`bootcmd` セクションは初期化プロセスの早い段階で実行されます。`runcmd` セクションは、`init` によってプロセスの末尾の近くで実行されます。これらのコマンドは次回の起動時に保存されず、初回の初期化時にのみ実行されます。

```
#cloud-config
users:
  - default
  - name: foobar
    gecos: User N. Ame
    groups: users
chpasswd:
  list: |
    root:password
    fedora:atomic
    foobar:foobar
    expire: False
bootcmd:
  - echo New MOTD >> /etc/motd
runcmd:
  - echo New MOTD2 >> /etc/motd
```

- `sudoers` の追加方法

以下のように `user-data` ファイルの `users` セクションに `sudo` および `groups` エントリーを追加して、ユーザーを `sudoer` として設定できます。

```
#cloud-config
users:
  - default
  - name: foobar
    gecos: User D. Two
    sudo: ["ALL=(ALL) NOPASSWD:ALL"]
    groups: wheel,adm,systemd-journal
    ssh_pwauth: True
    ssh_authorized_keys:
      - ssh-rsa AA...vz user@domain.com
chpasswd:
  list: |
    root:password
    cloud-user:atomic
    foobar:foobar
    expire: False
```

- 静的ネットワーク設定の設定方法

**network-interfaces** セクションをメタデータファイルに追加します。本セクションでは、通常のネットワーク設定オプションセットを説明します。

cloud-init の現在の [バグ](#) により、静的ネットワーク設定は自動的に開始されません。代わりに、デフォルトの DHCP 設定はアクティブなままになります。推奨される回避策は、**bootcmd** ディレクティブを介してネットワークインターフェイスを手動で停止および再起動することです。

```
network-interfaces: |
  iface eth0 inet static
  address 192.168.1.10
  network 192.168.1.0
  netmask 255.255.255.0
  broadcast 192.168.1.255
  gateway 192.168.1.254
bootcmd:
  - ifdown eth0
  - ifup eth0
```

- cloud-user を削除して、root だけがあり、その他のユーザーがないのか？ root ユーザーのみを作成するには、**user-data** ファイルの **users** セクションに root のエントリーを作成します。このセクションは、単に **name** オプションとして簡単に実行できます。

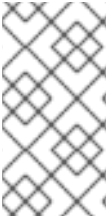
```
users:
  - name: root
chpasswd:
  list: |
    root:password
  expire: False
```

必要に応じて、以下のように root ユーザーの SSH キーを設定できます。

```
users:
  - name: root
    ssh_pwauth: True
    ssh_authorized_keys:
      - ssh-rsa AA..vz user@domain.com
```

- container-storage-setup でストレージをセットアップするにはどうすればよいですか？ たとえば、デフォルトの 3GB ではなく、root 論理ボリュームのサイズを 6GB に設定するには、**user-data** で **write\_files** ディレクティブを使用します。

```
write_files:
  - path: /etc/sysconfig/docker-storage-setup
    permissions: 0644
    owner: root
    content: |
      ROOT_SIZE=6G
```

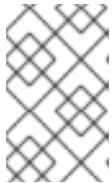


### 注記

RHEL 7.4 より前のバージョンでは、`container-storage-setup` は `docker-storage-setup` と呼ばれていました。ストレージに OverlayFS を使用している場合は、RHEL 7.4 の時点で、SELinux でそのタイプのファイルシステムを Enforcing モードで使用できるようになりました。

- Overlay グラフドライバーを有効にするにはどうすればよいですか？  
Overlay グラフドライバーは、`container-storage-setup` で有効化されます。`runcmd` ディレクティブを使用して、`STORAGE_DRIVER` オプションを "overlay2" に変更します。

```
runcmd:
- echo "STORAGE_DRIVER=overlay2" >> /etc/sysconfig/docker-storage-setup
```



### 注記

バックエンドストレージドライバーの変更は破壊的な操作であることに注意してください。さらに、OverlayFS は POSIX に準拠しておらず、制限と共に使用できます。詳細は、[RHEL 7.2 リリースノート](#) を参照してください。

- インスタンスで `cloud-init` を再実行するにはどうすればよいですか？  
ほとんどの場合、`cloud-init` を再実行して、すでに作成されている仮想マシンの設定を変更することはできません。

インスタンス ID を変更できる環境で `cloud-init` が使用されている場合（例：Atomic0 から Atomic1 へ）、インスタンス ID を変更して再起動して `cloud-init` を再実行することで、既存の仮想マシンを再設定することができます。`cloud-init` は、完全に適切に設定された初回起動システムでを作成するように設定されるため、実稼働環境では推奨されません。

ほとんどの IAAS 実装では、インスタンス ID を変更することはできません。`cloud-init` を再実行する必要がある場合は、新しいインスタンス ID を取得するためにインスタンスのクローンを作成する必要があります。

- シェルスクリプトを `bootcmd` および `runcmd` に配置することはできますか？  
有効。`bootcmd` または `runcmd` のリスト値を使用する場合は、各リスト項目は `execve` を使用して順番に実行されます。文字列の値を使用する場合、文字列全体がシェルスクリプトとして実行されます。または、`cloud-init` を使用してシェルスクリプトを実行する場合は、`cloud-init` に `.yaml` ファイルを指定する代わりに、（シバン(!) で完全）シェルスクリプトを指定できます。

シェルスクリプトを `bootcmd` および `runcmd` に配置する方法の例については、この [Web サイト](#) を参照してください。

## 第4章 インストール後の設定

Red Hat Enterprise Linux Atomic Host は、`/etc/` ディレクトリーの設定ファイルを使用して設定されます。これは、Red Hat Enterprise Linux 7 の設定方法と似ています。Red Hat Enterprise Linux Atomic Host はデスクトップのない最小限のサーバー製品であるため、GUI にあるグラフィカルな設定ツールは利用できません。

### 4.1. ネットワークの設定

インストール時にネットワークを設定しなかった場合は、`nmcli` ツールを使用してインストール後にネットワークを設定することができます。以下のコマンドは、`atomic` という名前のネットワーク接続を作成し、ホスト名を設定してから、その接続を有効にします。

```
# nmcli con add type ethernet con-name atomic ifname eth0
# nmcli con modify my-office my-office ipv4.dhcp-hostname atomic ipv6.dhcp-hostname atomic
# nmcli con up atomic
```

`nmcli` ツールの使用方法は、[Section 2.3.2](#) を参照してください。Red Hat Enterprise Linux 7 ネットワークガイドの `nmcli` を使用したネットワーク接続

### 4.2. RHEL Atomic Host の登録

ソフトウェア更新を有効にするには、Red Hat Enterprise Linux Atomic Host インストールを登録する必要があります。これは、以下で説明されているように `subscription-manager` コマンドを使用して行います。デフォルトでは、`subscription-manager` は、システムが Red Hat ソフトウェアリポジトリーにアクセスできる機能でインターネットにアクセスできることを前提としています。直接インターネットアクセスがない場合は、いくつかのオプションがあります。

- **HTTP プロキシ:** システムが HTTP プロキシの使用を必要とするネットワークにある場合は、Red Hat ナレッジベースの記事 [HTTP プロキシを使用するようにサブスクリプションマネージャーを設定する](#) を参照してください。サブスクリプションレコードの確認時に使用する簡単な名前を指定する場合は、`--name=` オプションを含めることができます。
- **内部 OSTree ミラー:** (HTTP プロキシ経由でも) インターネットにアクセスできない場合は、環境に対してローカルとなるように、公式の Red Hat Enterprise Linux Atomic Host ostree リポジトリーをミラーリングできます。このようなミラーを設定する手順は、[Using Atomic Host as an internal ostree mirror](#) を参照してください。

```
$ sudo subscription-manager register --username=<username> --auto-attach
```



#### 注記

Red Hat Enterprise Linux Atomic Host は、Red Hat Subscription Manager (RHSM) でのみ機能します。Red Hat Enterprise Linux Atomic Host は RHN では機能しません。



#### 注記

Red Hat Enterprise Linux Atomic Host は、2 つの製品 ID を登録します。1 つ目は、製品 ID 271、Red Hat Enterprise Linux Atomic Host です。2 つ目は、製品 ID 69、Red Hat Enterprise Linux Server です。いずれも同じエンタイトルメントを使用します。

適切に登録されたシステムは、以下のように両方の ID を表示します。

```
$ sudo subscription-manager list
+-----+
  Installed Product Status
+-----+
Product Name:  Red Hat Enterprise Linux Atomic Host
Product ID:    271
Version:      7
Arch:         x86_64
Status:       Subscribed
Status Details:
Starts:       02/27/2015
Ends:         02/26/2016

Product Name:  Red Hat Enterprise Linux Server
Product ID:    69
Version:      7.1
Arch:         x86_64
Status:       Subscribed
Status Details:
Starts:       02/27/2015
Ends:         02/26/2016
```

`subscription-manager` コマンドについては、Red Hat Subscription Management のコマンド [ラインからの登録](#) にも記載されています。

### 4.3. ユーザーアカウントの管理

現在、Red Hat Enterprise Linux 7 の一部のシステムユーザーは、`/etc/passwd` ファイルにリストされています。読み取り専用 `/usr/lib/passwd` ファイルに再配置されています。Red Hat Enterprise Linux Atomic Host 上のアプリケーションはLinux コンテナ内で実行されるため、これはデプロイメントには影響しません。`useradd` などの従来のユーザー管理ツールは、期待どおりにユーザーをローカルに追加したユーザー `/etc/passwd` ファイルに書き込みます。

## 第5章 アップグレードおよびダウングレード

### 5.1 Atomic Compose サーバーの設定

この手順では、Atomic Compose サーバーを設定する方法を説明します。Atomic Compose サーバーを使用してアトミック更新ツリーを作成できます。以下の手順では、アップストリームの OSTree リポジトリのローカルミラーを作成する Atomic Compose サーバーを設定する方法を説明します。

1. ホストのシェルにログインし、Atomic Tools コンテナを実行します。

```
# atomic run rhel7/rhel-tools
```

2. tools コンテナ内から、非特権ユーザーを作成します。

```
# adduser container
```

3. エンタイトルメント証明書を取得し、**chown** を使用して、特権のないコンテナユーザーが所有するようにします。

```
# cd ~container
# cp /host/etc/pki/entitlement/*.pem .
# chown container: *.pem
# runuser -u container bash
```

4. root アカウントからログアウトします。

```
# exit
```



#### 注記

データはコンテナの外にあるように `/host/var/tmp/repo` を使用します。これは、Ceph/etc へのリモートマウントポイントである可能性があります。

5. repo ディレクトリー内にエンタイトルメント証明書を配置します。

```
$ cd /host/var/tmp
$ mkdir repo && ostree --repo=repo init --mode=archive-z2
$ mv ~/.pem repo/
```

6. ホストからリポジトリにリモート設定をコピーします。

```
$ cat /host/etc/ostree/remotes.d/redhat.conf >> repo/config
```

7. 変数の変更

`repo/config` を編集し、`tls-client-*` 変数を以下のように変更します。これは、CDN にアクセスするために必要なクライアント証明書を検索する場所をコマンドに指示します。

```
tls-client-cert-path = ./repo/123451234512345.pem
tls-client-key-path = ./repo/123451234512345-key.pem
```

8. 最終手順

すべて設定されました。以下のコマンドは、すべてのコンテンツを段階的にミラーリングします。cron ジョブまたは systemd タイマーからコマンドを実行できます。

```
$ ostree --repo=repo pull --mirror rhel-atomic-host-ostree
```

クライアントマシンの場合は、`/etc/ostree/remotes.d/redhat.conf` を変更して、リポジトリディレクトリーをエクスポートする静的 Web サーバーを参照します。

## 5.2. 新規バージョンへのアップグレード

Yum を使用し、従来のパッケージ管理モデルを持つ Red Hat Enterprise Linux 7 とは異なり、RHEL Atomic Host は OSTree を使用し、新しいオペレーティングシステムのルートを準備し、次回の起動のデフォルトを作成してアップグレードされます。

アップグレードを実行するには、以下のコマンドを実行します。

```
# atomic host upgrade
# systemctl reboot
```



### 注記

OSTree は安全にダウンロードされます。ただし、必要な場合は、アップグレードする OSTree の証明を手動で検証できます。[OS ツリーの手動検証](#) を参照してください。

HTTP プロキシを必要とするシステムを使用している場合、プロキシは環境変数で設定されます。環境変数を設定するには、以下のようなコマンドを実行します。

```
# env http_proxy=http://proxy.example.com:port/ atomic host upgrade
```

## 5.3. 以前のバージョンへのロールバック

Red Hat Enterprise Linux Atomic Host の以前のインストールに戻すには、以下のコマンドを実行します。

```
# atomic host rollback
# systemctl reboot
```

Red Hat Enterprise Linux Atomic Host の 2 つのバージョンが、初回のアップグレード後にシステムで利用できます。1 つは現在実行中のバージョンです。もう 1 つは、アップグレードから最近インストールした新しいバージョン、または最後のアップグレード前に行われたバージョンです。



### 重要

設定は更新後も保持されますが、`forward-preserved` のみになります。つまり、設定変更を行い、後で以前のバージョンにロールバックすると、加えた変更は元に戻されます。



### 注記

`atomic host upgrade` コマンドを実行すると、実行されていないバージョンの Red Hat Enterprise Linux Atomic Host が置き換えられます。このバージョンは、次回の起動時に使用されるように設定されます。



実行中のオペレーティングシステムのバージョンを確認するには、以下のコマンドを実行します。

```
# atomic host status
```

`/ostree/deploy/rhel-atomic-host/` ディレクトリーにディレクトリーのハッシュ名が含まれる出力は以下ようになります。

```
# atomic host status
State: idle
Deployments:
* rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard
  Version: 7.3 (2016-09-27 17:53:07)
  BaseCommit: d3fa3283db8c5ee656f78dcfc0fcffe6cd5aa06596dac6ec5e436352208a59cb
  Commit: f5e639ce8186386d74e2558e6a34f55a427d8f59412d47a907793e046875d8dd
  OSName: rhel-atomic-host

rhel-atomic-host-ostree:rhel-atomic-host/7.2/x86_64/standard
  Version: 7.2.7 (2016-09-15 22:28:54)
  BaseCommit: dbbc8e805f0003d8e55658dc220f1fe1397caf80221cc050eeb1bbf44bef56a1
  Commit: 5cd426fa86bd1652ecd8f7d489f89f13ecb7d36e66003b0d7669721cb79545a8
  OSName: rhel-atomic-host
```

この出力例では、バージョン 7.3 が次回の再起動時に起動することを示しています。次回の再起動時に起動するバージョンが最初に出力されます。

また、このサンプルでは、バージョン 7.2.7 が現在稼働しているバージョンであることも示しています。現在実行中のバージョンにはアスタリスク(\*)が付いています。

この出力は、`atomic host upgrade` コマンドの実行直後に作成され、次回の再起動時に新しいバージョンが適用されるようにステージングされたことを意味します。

#### 5.4. クライアントでの `initramfs` イメージの生成

デフォルトでは、Atomic Host はサーバー側でビルドされた汎用 `initramfs` イメージを使用します。これは、インストールごとに `initramfs` が生成される `yum` ベースの Red Hat Enterprise Linux とは異なります。ただし、状況によっては、追加の設定またはコンテンツを追加する必要がある場合があります。これには、クライアント側で `initramfs` を生成する必要があります。

Atomic Host クライアントマシンがアップグレードされるたびに `initramfs` を生成するようにするには、以下を実行します。

```
$ rpm-ostree initramfs --enable
```

この後、アップグレードごとにクライアントが `dracut` プログラムを実行し、新しい `initramfs` を構築します。

クライアントで `initramfs` の生成を無効にするには、以下を実行します。

```
$ rpm-ostree initramfs --disable
```



## 第6章 ATOMIC HOST の管理

### 6.1. Atomic Host

**atomic** コマンドラインツールを使用すると、Atomic Host システムのステータスを確認し、アップグレードおよびロールバックを実行したり、特定のオペレーティングシステムツリーをデプロイしたりできます。

**atomic host status** を使用して、システムにダウンロードしたオペレーティングシステムのツリーを一覧表示し、現在実行中のバージョンを確認します。アスタリスク(\*)は、現在実行中のツリーをマークします。

```
# atomic host status
State: idle
Deployments:
* rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard
  Version: 7.3 (2016-09-27 17:53:07)
  BaseCommit: d3fa3283db8c5ee656f78dcfc0fcffe6cd5aa06596dac6ec5e436352208a59cb
  Commit: f5e639ce8186386d74e2558e6a34f55a427d8f59412d47a907793e046875d8dd
  OSName: rhel-atomic-host

rhel-atomic-host-ostree:rhel-atomic-host/7.2/x86_64/standard
  Version: 7.2.7 (2016-09-15 22:28:54)
  BaseCommit: dbbc8e805f0003d8e55658dc220f1fe1397caf80221cc050eeb1bbf44bef56a1
  Commit: 5cd426fa86bd1652ecd8f7d489f89f13ecb7d36e66003b0d7669721cb79545a8
  OSName: rhel-atomic-host
```

システムをアップグレードするには、**atomic host upgrade** を使用します。このコマンドは、利用可能な最新の **ostree** をダウンロードし、システムを再起動した後にデプロイします。再びアップグレードすると、新たにダウンロードした **ostree** が最も古いものを置き換えます。アップグレードには数分かかる場合があります。

```
# atomic host upgrade
# systemctl reboot
```

システムのダウンロードした他のツリーに戻すには、**atomic host rollback** を使用します。このコマンドは、アップグレード後に問題が発生した場合（たとえば、新しいパッケージが設定したサービスを壊すなど）場合に特に便利です。これは、すぐに以前の状態に切り替えることができます。**-r** オプションを使用して、直ちに再起動を開始できます。

```
# atomic host rollback -r
```

**ostree** の特定バージョンをデプロイするには、**atomic host deploy** を使用します。バージョンまたはコミット ID が分かっている場合は、それを指定できます。

```
# atomic host deploy <version/commit ID>
```

**--preview** オプションを使用して、指定したバージョンと現在実行中のツリー間のパッケージの違いを確認します。

```
# atomic host deploy 7.3 --preview
```

詳細なタスクについては、**ostree** ツールを使用して Atomic Host を管理できます。たとえば、ツリーのバージョン番号が不明な場合は、以下のコマンドを使用して **ostree** ログを取得し、利用可能なバージョンを一覧表示できます。

```
# ostree pull --commit-metadata-only --depth -1 rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard
# ostree log rhel-atomic-host/7/x86_64/standard
```

以下のコマンドのいずれかを使用して、**ostree** デプロイメントを削除できます。

```
# rpm-ostree cleanup -r
# rpm-ostree cleanup -p
```

**-p** オプションを使用すると、保留中のデプロイメントが削除されますが、**-r** はロールバックデプロイメントを削除します。

一時ファイルとキャッシュされたデータを消去するには、次のいずれかのコマンドを使用します。

```
# rpm-ostree -m
# rpm-ostree -b
```

**-m** オプションはキャッシュされた RPM リポジトリメタデータを削除しますが、**-b** は一時ファイルをクリアしますが、デプロイメントはそのまま残されます。

**atomic** および **ostree** ツールにはどちらも詳細な **--help** オプションがあり、システムで利用可能なすべてのコマンドを確認するには、**atomic host --help** または **ostree --help** を使用します。

## 6.2. パッケージの階層化

**rpm-ostree install** を使用すると、元の OSTree の一部ではない RPM ソフトウェアパッケージをシステムに永続的に追加できます。**rpm-ostree override** を使用すると、ベースシステムレイヤーの既存の RPM パッケージを、そのパッケージの別のバージョンで上書きできます。これらの機能は、パッケージ階層化機能を使用して行われます。

パッケージ層は、同じ OSTree を共有する他のマシンに影響を与えずに、1 台のマシンに特定のパッケージをインストールする必要がある場合に役立ちます。パッケージ階層化のユースケースの例は、**strace** などの診断ツールのインストールです。ベースシステムでパッケージを上書きする例は、別のバージョンの **docker** パッケージを使用する場合です。

### 6.2.1. RHEL Atomic Host への新規 RPM パッケージのインストール

階層型パッケージとその依存関係を RHEL Atomic Host にインストールするには、以下のコマンドを実行します。

```
# rpm-ostree install <package>
```

レイヤー化されたパッケージを削除するには、次のコマンドを使用します。

```
# rpm-ostree uninstall <package>
```

**rpm-ostree install** または **rpm-ostree uninstall** を実行しても、パッケージをすぐにインストールまたはアンインストールすることはできません。実際にパッケージをインストールまたはアンインストールするには、以下の 2 つのオプションがあります。

- システムを再起動します。
- LiveFS を使用して直ちに変更を適用します。



### 注記

LiveFS は依然としてテクノロジープレビュー機能であるため、実稼働環境では依存しないでください。

パッケージのみをインストールする場合は、**rpm-ostree ex livefs** コマンドを使用します。

パッケージを削除またはアップグレードする場合は、**rpm-ostree ex livefs --replace** コマンドを使用します。

**atomic host status** を実行して、システムに手動でインストールされているパッケージを確認することができます。

以下は、RHEL Atomic Host に **strace** をインストールする例と、OSTree に含まれることを確認する方法を示しています。**yum** でパッケージをインストールするのと同様に、パッケージをインストールする前に Atomic Host システムを登録してサブスクライブする必要があります。

1. Atomic Host のデプロイメントの現在のステータスを確認します。

```
-bash-4.2# rpm-ostree status
State: idle
Deployments:
● rhelah-7.4:rhel-atomic-host/7/x86_64/standard
  Version: 7.4.0 (2017-07-28 00:26:01)
  Commit: 846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a

rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard
  Version: 7.4.0 (2017-07-13 17:46:26)
  Commit: c28cad0d4144d91a3c206574e9341cd5bdf7d34cfaa2acb74dd84c0bf022593a
  GPGSignature: 1 signature
  Signature made Thu 13 Jul 2017 01:54:13 PM EDT using RSA key ID
199E2F91FD431D51
  Good signature from "Red Hat, Inc. <security@redhat.com>"
```

2. 以下のように strace パッケージをインストールします。

```
-bash-4.2# rpm-ostree install strace
Checking out tree 846fb0e... done
...
Importing metadata [=====] 100%
Resolving dependencies... done
Will download: 1 package (470.0 kB)
  Downloading from rhel-7-server-rpms: [=====] 100%
Importing: [=====] 100%
Overlaying... done
Writing rpmdb... done
Writing OSTree commit... done
Copying /etc changes: 20 modified, 5 removed, 43 added
Transaction complete; bootconfig swap: yes deployment count change: 0
Freed objects: 388.5 MB
```

```
Added:
strace-4.12-4.el7.x86_64
Run "systemctl reboot" to start a reboot
```

- ステータスを再度確認し、`strace` をインストールして作成されたレイヤー化されたパッケージを表示します。

```
-bash-4.2# rpm-ostree status
State: idle
Deployments:
  rhelrh-7.4:rhel-atomic-host/7/x86_64/standard
    Version: 7.4.0 (2017-07-28 00:26:01)
    BaseCommit:
      846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
    LayeredPackages: strace

  ● rhelrh-7.4:rhel-atomic-host/7/x86_64/standard
    Version: 7.4.0 (2017-07-28 00:26:01)
    Commit: 846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
```

- `strace` パッケージがまだインストールされているように見えることに注意してください。

```
# rpm -q strace
package strace is not installed
```

- いくつかの問題を考慮します。パッケージ自体のレイヤーにインストールされていても、システムにインストールされているようには表示されません。この時点で、**`rpm-ostree ex livefs`** を使用して直ちに再起動または適用することにより、保留中のデプロイメントを適用する必要があります。ただし、この決定を行う前に、以下の注記と制限を考慮してください。

- 変更を再起動または適用せずに、**`rpm-ostree install`** を複数回連続して実行すると、最新のコマンドのみが有効になります。**`wget`** と **`strace`** を順番にインストールしてから再起動すると、再起動後に **`strace`** のみが存在します。複数のパッケージを新しいデプロイメントに追加するには、コマンドを使用して、すべてのパッケージを同じ行で指定します。以下に例を示します。

```
# rpm-ostree install wget strace
```

- 現在、`root` 以外のユーザーが所有するファイルを含むパッケージをインストールすることはサポートされていません。たとえば、**`httpd`** パッケージには、**`apache`** のグループ所有権を持つファイルが含まれ、インストールは失敗します。

```
# rpm-ostree install httpd
notice: pkg-add is a preview command and subject to change.
```

```
Downloading metadata: [=====] 100%
Resolving dependencies... done
error: Unpacking httpd-2.4.6-40.el7_2.4.x86_64: Non-root ownership currently
unsupported: path "/run/httpd" marked as root:apache)
```

- `rpm-ostree install`** の後に、**`atomic host deploy`** または **`rpm-ostree deploy`** を使用して 7.2.6 よりも古いバージョン OSTree バージョンをデプロイしないでください。**`rpm-ostree install`** の後にそのようなバージョンにデプロイしようとする、システムは、**`atomic host`**

**upgrade** または **rpm-ostree upgrade** を使用して次のリリースにアップグレードできない状態のままになります。ただし、**atomic host rollback** または **rpm-ostree rollback** は引き続き成功し、システムを以前のデプロイメントに戻します。

6. Reboot または LiveFS: デプロイメントを有効にするために再起動するか、`livefs` 機能を使用して、以下のようにすぐに有効にします。

```
# rpm-ostree ex livefs
notice: "livefs" is an experimental command and subject to change.
Diff Analysis: 846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
=> 97f937f3789d0f25b887bcd4fcc03d33b76ee4c87095af48c602b5826519ce1b
Files:
modified: 0
removed: 0
added: 11
Packages:
modified: 0
removed: 0
added: 1
Preparing new rollback matching currently booted deployment
Copying /etc changes: 20 modified, 5 removed, 43 added
Transaction complete; bootconfig swap: yes deployment count change: 1
Overlaying /usr... done
```

7. 再度、`strace` パッケージがインストールされていることを確認し、新しい LiveCommit を含むデプロイメントのステータスを確認します。

```
# rpm -q strace
strace-4.12-4.el7.x86_64

# rpm-ostree status
State: idle
Deployments:
  rhelah-7.4:rhel-atomic-host/7/x86_64/standard
    Version: 7.4.0 (2017-07-28 00:26:01)
    BaseCommit:
      846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
    Commit:
      97f937f3789d0f25b887bcd4fcc03d33b76ee4c87095af48c602b5826519ce1b
    LayeredPackages: strace

  ● rhelah-7.4:rhel-atomic-host/7/x86_64/standard
    Version: 7.4.0 (2017-07-28 00:26:01)
    BootedCommit:
      846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
    LiveCommit:
      97f937f3789d0f25b887bcd4fcc03d33b76ee4c87095af48c602b5826519ce1b

  rhelah-7.4:rhel-atomic-host/7/x86_64/standard
    Version: 7.4.0 (2017-07-28 00:26:01)
    Commit:
      846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
```

この時点で、インストールしたソフトウェアの使用を開始できます。`rpm-ostree` および Live の更新の詳細は、[Project Atomic rpm-ostree](#) を参照してください。

## 6.2.2. 後でインストールするための RPM のダウンロードおよびキャッシュ

`--download-only` オプションおよび `--cache-only` オプションを使用すると、`rpm-ostree` インストールプロセスを2つのステージに分割できます。

1. レイヤー化された RPM をダウンロードし、キャッシュします。
2. キャッシュされたデータからのインストール。

これらのオプションにより、一度に RPM をダウンロードし、選択した場合はオフラインでいつでもインストールできます。

## 6.2.3. リポジトリメタデータの更新

`rpm-ostree refresh-md` サブコマンドは、最新のリポジトリメタデータをダウンロードしてキャッシュします。これは、Yum パッケージマネージャーの `yum makecache` コマンドに似ています。

## 6.2.4. 既存の RPM パッケージの上書き

Atomic ベースにある RPM パッケージを上書きし、別のバージョンをインストールするには、`rpm-ostree override` コマンドを使用します。その仕組みを以下に示します。

- 使用する RPM パッケージを Atomic ホストにコピーします。RPM で必要な依存パッケージも含めます。パッケージは、現在のパッケージからアップグレードまたはダウングレードできます。
- `rpm-ostree override` コマンドを実行します。
- 変更を有効にするために Atomic ホストを再起動します。



### 注記

`rpm-ostree オーバーライド` を使用して Atomic の Docker ランタイムを置き換える方法の例については、[RHEL Atomic Host での docker パッケージのバージョンのロック](#) を参照してください。

以下は、RHEL Atomic Host で `openssh-server` パッケージ（および依存パッケージ）を置き換える例です。

1. 置き換える RPM パッケージ（および依存パッケージ）を取得し、Atomic Host のディレクトリに配置します。
2. 現在のディレクトリのパッケージ（この場合は `openssh-server`、`openssh-clients`、および `openssh`）をダウングレードします。以下を入力し、これらのパッケージを置き換えます。

```
# rpm-ostree override replace \
  openssh-server-6.6.1p1-35.el7_3.x86_64.rpm \
  openssh-clients-6.6.1p1-35.el7_3.x86_64.rpm \
  openssh-6.6.1p1-35.el7_3.x86_64.rpm

Checking out tree 5df677d... done
...
Transaction complete; bootconfig swap: yes deployment count change: 1
Downgraded:
  openssh 7.4p1-16.el7 -> 6.6.1p1-35.el7_3
```

```
openssh-clients 7.4p1-16.el7 -> 6.6.1p1-35.el7_3
openssh-server 7.4p1-16.el7 -> 6.6.1p1-35.el7_3
Run "systemctl reboot" to start a reboot
```

3. Atomic Host システムを再起動します。

```
# systemctl reboot
```

4. パッケージがインストールされ、利用できることを確認します。

```
# atomic host status
State: idle
Deployments:
  • ostree://rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard
    Version: 7.5.0 (2018-04-05 10:29:00)
    BaseCommit: 5df677dcfef08a87dd0ace55790e184a35716cf11260239216bfeba2eb7c60b0
    ReplacedBasePackages: openssh openssh-server openssh-clients
                          7.4p1-16.el7 -> 6.6.1p1-35.el7_3

# rpm -q openssh openssh-clients openssh-server
openssh-6.6.1p1-35.el7_3.x86_64
openssh-clients-6.6.1p1-35.el7_3.x86_64
openssh-server-6.6.1p1-35.el7_3.x86_64
```

以前のパッケージバージョンに戻す場合は、**rpm-ostree override reset** を使用して実行できます。 **rpm-ostree override reset <packagename>** を使用して個々のパッケージを削除するか、 **rpm-ostree override reset --all** を使用して上書きされたパッケージをすべて削除します。

### 6.3. "ostree admin unlock"

OSTree **admin** は、現在の **ostree** デプロイメントのロックを解除し、書き込み可能な overlayfs を `/usr` にマウントして、パッケージを一時的にインストールできるようにします。ただし、後でインストールしたパッケージは再起動後に維持されません。**ostree admin unlock** には、overlayfs および SELinux に関する特定の制限や既知の問題があるため、テストにのみ使用してください。ソフトウェアを追加する場合は、**rpm-ostree install** を長期使用に推奨されます。

### 6.4. System Containers and Super-Privileged Containers (SPC)

場合によっては、コンテナ化されたサービスまたはアプリケーションは、Docker 形式のコンテナのデフォルト方法とは異なる方法でビルドされるコンテナイメージから実行される必要があります。このような特別なケースコンテナは Super-Privileged Containers (SPC) およびシステムコンテナです。これらは2つの状況で必要になります。

- **SPC: 特定のコンテナがより多くの権限とホストへのアクセスを必要とする場合。**

スーパー特権コンテナはホストコンピューターへの特別な権限で実行されるため、デフォルトの Docker 形式のコンテナとは異なり、ホストを変更できます。監視とロギング用のツールは SPC に含まれているため、必要なホストにアクセスできます。このような SPC は **rsyslog**、**sadc**、および **atomic-tools** コンテナです。SPC の詳細は、Red Hat Enterprise Linux Atomic Host Managing Containers Guide の [Running Super-Privileged Containers](#) の章を参照してください。

- **システムコンテナ**：特定のコンテナは docker デーモンとは別に実行する必要があります。

システムコンテナは、docker デーモンを実行する前に必要なサービスをコンテナ化する方法です。

このようなサービスは、`docker` の環境（ネットワークのセットアップなど）を設定して、`Docker` デーモンによって実行できず、`Docker` 形式のイメージではないためです。これらはランタイムに **runc** を使用し、ストレージに **ostree**、レジストリーの検索およびプルに **skopeo** を使用し、管理に `systemd` を使用します。システムコンテナは、**runc** ユーティリティを使用して `systemd` UNIT ファイルから実行されます。さらに、このようなサービスのコンテナ化は、`ostree` システムイメージを小さくする方法です。このようなシステムコンテナは **etcd** および **Flannel** です。詳細は、Red Hat Enterprise Linux Atomic Host Managing Containers Guide の [Running System Containers](#) の章を参照してください。