



Red Hat Enterprise Linux Atomic Host 7 スタートガイド

スタートガイド

Red Hat Atomic Host Documentation Team

スタートガイド

法律上の通知

Copyright © 2016 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

RHEL Atomic Host の Red Hat スタートガイド

目次

第1章 RHEL ATOMIC HOST のインストール	3
1.1. 概要	3
1.2. システム要件	3
1.3. RHEL ATOMIC HOST の登録	4
1.4. RHEL ATOMIC HOST の設定	4
1.5. インストールのアップグレードおよびダウングレード	5
第2章 コンテナの使用方法	7
2.1. APACHE WEB SERVER コンテナのインストールおよびデプロイ	7
2.2. MARIADB コンテナのインストールおよびデプロイ	10

第1章 RHEL ATOMIC HOST のインストール

1.1. 概要

Red Hat Enterprise Linux Atomic Host は、Linux コンテナを Docker フォーマットで実行するために最適化された Red Hat Enterprise Linux 7 製品です。Red Hat Enterprise Linux Atomic Host は、Red Hat Enterprise Linux 7 で提供されている高度な技術を使用できるように設計されました。

Red Hat Enterprise Linux Atomic Host は、SELinux によるマルチテナント環境での強固な保護機能、およびアトミックなアップグレードとロールバック機能によるダウンタイムの少ない迅速で簡単なメンテナンスを提供しています。Red Hat Enterprise Linux Atomic Host は、Red Hat Enterprise Linux 7 と同じ RPM パッケージから配信されるアップストリームプロジェクトを使用します。

Red Hat Enterprise Linux Atomic Host には、Linux コンテナをサポートするために以下のツールがプリインストールされています。

- ※ Docker
- ※ Kubernetes

Red Hat Enterprise Linux Atomic Host は、以下のテクノロジーを使用します。

- ※ OSTree および rpm-OSTree: アトミックなアップグレードと、アップグレードをロールバックする機能を提供します。
- ※ systemd: Linux システムにおける強力な新しい init システムにより、起動時間が速くなり、システムのオーケストレーションが容易になります。
- ※ SELinux: 完全なマルチテナントセキュリティを提供するため、デフォルトで有効にされます。さらに Integrity Measurement Architecture (IMA)、audit、および libwrap が systemd から利用できます。

重要: Red Hat Enterprise Linux Atomic Host の管理方法は、その他の Red Hat Enterprise Linux 7 製品とは異なります。主に以下の点が異なります。

- ※ yum を使用せずにシステムをアップグレードします。
- ※ ローカルシステム設定の 2 つの書き込み可能なディレクトリー (/etc/ および /var/) があります。/usr/ ディレクトリーは読み取り専用でマウントされます。その他のディレクトリーは書き込み可能な場所へのシンボリックリンクです。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。
- ※ デフォルトのパーティション方法では、利用可能なほとんどの領域をコンテナに割り当て、デフォルトのループバックの代わりに direct LVM を使用します。

ユーザーおよびホスト固有のデータは、**/var/** ディレクトリーにのみ保存される必要があります。**/etc/** ディレクトリー内の設定ファイルのみが変更対象になります。

1.2. システム要件

Red Hat Enterprise Linux Atomic Host は、過去 2 年以内に工場で生産された大半のハードウェアと互換性があります。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リストについては、「[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については、「[Red Hat Enterprise Linux テクノロジーの機能と制限](#)」を参照してください。

Red Hat Enterprise Linux Atomic Host のランタイム要件は Red Hat Enterprise Linux と同じです。ただし、ベアメタルまたは仮想環境の Anaconda ベースのインストール (インタラクティブ、キックスタート、および PXE) の場合はメモリーが 2GB 必要です。

1.3. RHEL ATOMIC HOST の登録

ソフトウェアアップデートを有効にするには、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、以下に説明したとおりに **subscription-manager** コマンドで実行できます。お使いのシステムは、HTTP プロキシを使用する必要があるネットワークに置かれます。詳細は、Red Hat ナレッジベースの記事「[Red Hat Subscription Manager に HTTP Proxy を設定する](#)」を参照してください。--name= オプションを使用すると、サブスクリプションレコードを表示する際に覚えやすい名前を使用できます。

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

注意: Red Hat Enterprise Linux Atomic Host は Red Hat Subscription Manager (RHSM) とのみ連携します。Red Hat Enterprise Linux Atomic Host は RHN とは連携しません。

注意: Red Hat Enterprise Linux Atomic Host を使用するには、製品 ID 271 の Red Hat Enterprise Linux Atomic Host と製品 ID 61 の Red Hat Enterprise Linux Server を登録する必要があります。これらはどちらも同じエンタイトルメントを使用します。

システムが適切に登録されると、以下のように両方の ID が表示されます。

```
$ 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 Guide](#)』の 3.2 「Registering from the Command Line」にも記載されています。

1.4. RHEL ATOMIC HOST 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、Red Hat Enterprise Linux 7 と同様の方法で設定されます。Red Hat Enterprise Linux Atomic Host は、デスクトップを使用しない最小のサーバー製品です。したがって、GUI で使用されるグラフィカル設定ツールは利用できません。

現時点で Red Hat Enterprise Linux 7 の **/etc/passwd** ファイルに一覧表示されている一部のシステムユーザーは、読み取り専用の **/usr/lib/passwd** ファイルに移動しています。Red Hat Enterprise Linux Atomic Host のアプリケーションは Linux コンテナ内で実行されるため、これがデプロイメントに影響することはありません。**useradd** などの従来のユーザー管理ツールについては、**/etc/passwd** ファイルにローカルで追加したユーザーを書き込みます。

インストール時にネットワークを設定していない場合は、**nmcli** ツールを使用し、インストール後に設定することができます。以下のコマンドは、**atomic** というネットワーク接続の作成、ホスト名のセットアップ、およびその接続のアクティブ化を実行します。

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

nmcli ツールの使用方法についての詳細は、『Red Hat Enterprise Linux 7 ネットワークガイド』の「[2.3.2. nmcli を使用したネットワーク接続](#)」を参照してください。

Red Hat Enterprise Linux 7 の設定方法についての詳細は、『[Red Hat Enterprise Linux 7 システム管理者のガイド](#)』を参照してください。

1.5. インストールのアップグレードおよびダウングレード

RHEL Atomic Host は、オープンソースツールの 1 つである **rpm-OSTree** を使用し、起動可能で不変の、バージョン管理された RPM コンテンツのファイルシステムツリーを管理します。**yum** を使用し、従来のパッケージ管理モデルを持つその他の Red Hat Enterprise Linux 7 製品とは異なり、Red Hat Enterprise Linux Atomic Host は OSTree を使用します。新しいオペレーティングシステムの root を用意し、次の起動時のデフォルトに設定するとアップグレードが行われます。

1.5.1. 新規バージョンへのアップグレード

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。

```
$ sudo atomic host upgrade
$ sudo systemctl reboot
```

HTTP プロキシが必要なシステムを使用している場合は、環境変数でプロキシを設定できます。環境変数を設定するには、以下のようなコマンドを使用します。

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

1.5.2. 直前バージョンへのロールバック

Red Hat Enterprise Linux Atomic Host の直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。

```
$ sudo atomic host rollback
$ sudo systemctl reboot
```

初回アップグレード後は、Red Hat Enterprise Linux Atomic Host の 2 つのバージョンがシステムで利用可能になります。1 つ目は現在実行されているバージョンです。2 つ目は、アップグレードでインストールされたばかりの新規バージョンか、または最新のアップグレード前にインストールされていたバージョンです。

重要

設定は複数回のアップデート後も保持されますが、以前のバージョンに戻す場合は保存されません。つまり、設定を変更した後に直前のバージョンにロールバックする場合は、変更した設定が元に戻ります。

注記

`atomic host upgrade` コマンドを実行すると、Red Hat Enterprise Linux Atomic Host の実行されていないバージョンの置換が実行されます。さらに、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。

実行されているオペレーティングシステムのバージョンを判別するには、以下のコマンドを実行します。

```
$ sudo 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。

```
$ sudo atomic host status
TIMESTAMP (UTC)      VERSION  ID                OSNAME
REFSPEC
* 2015-05-07 19:00:48  7.1.2    203dd666d3       rhel-atomic-host
rhel-atomic-host:rhel-atomic-host/7/x86_64/standard
2015-04-02 20:14:06      7.1.1-1  21bd99f9f3       rhel-atomic-host
rhel-atomic-host:rhel-atomic-host/7/x86_64/standard
```

この出力例は、次回の再起動時にバージョン 7.1.1-1 が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は先頭に表示されます。

また、この例ではバージョン 7.1.2 が現在実行中であることを示しています。現在実行中のバージョンにはアスタリスク (*) が付けられます。

この出力は `atomic host upgrade` コマンドの実行後すぐに作成されました。したがって、新規バージョンは次回の再起動時に適用されるように設定されています。

第2章 コンテナの使用法

2.1. APACHE WEB SERVER コンテナのインストールおよびデプロイ

2.1.1. 概要

コンテナがどのように機能するかを説明する際に、Web サーバーを例に挙げることがよくあります。このトピックでは以下の手順について説明します。

- ※ コンテナ内に Apache (httpd) Web サーバーを構築する
- ※ ホストのポート 80 でサービスを公開する
- ※ 簡単な index.html ファイルを提供する
- ※ バックエンドサーバーからデータを表示する (後述する追加の MariaDB コンテナが必要)

2.1.2. Apache Web Server コンテナの作成および実行

1. **システムのインストール:** docker パッケージを含む RHEL 7 または RHEL Atomic システムをインストールして、docker サービスを起動します。
2. **イメージの取得:** 以下のコマンドを実行して rhel7 イメージを取得します。

```
# docker pull rhel7:latest
```

3. **サポートファイルが含まれる tarball の取得:** 本書に添付されている tarball ファイル ([web_cont_2.tgz](#)) をダウンロードします。新規の mywebcontainer ディレクトリーにダウンロードして、以下のように untar を実行します。

```
# mkdir ~/mywebcontainer
# cp web_cont*.tgz ~/mywebcontainer
# cd ~/mywebcontainer
# tar xvf web_cont*.tgz
action
Dockerfile
```

4. **CGI スクリプト action の変更:** action ファイルを随時編集します。このファイルは、バックエンドデータベースサーバーのコンテナからデータを取得するために使用されます。このスクリプトでは、ホストシステムの docker0 インターフェースの IP アドレスは **172.17.42.1**、ログイン用のユーザーアカウントは **dbuser1**、そのパスワードは **redhat**、および使用するデータベースの名前は **gss** としてデータベースにログインできると想定します。IP アドレスに変更がなく、後述のデータベースコンテナをお使いの場合は、このスクリプトを変更する必要はありません (このスクリプトを使用せずに、Web サーバーを使用して HTML コンテンツを取得することもできます)。

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import MySQLdb as mdb
import os

con =
mdb.connect(os.getenv('DB_SERVICE_SERVICE_HOST', '172.17.42.1'),
```

```
'dbuser1', 'redhat', 'gss')

with con:

    cur = con.cursor()
    cur.execute("SELECT MESSAGE FROM atomic_training")

    rows = cur.fetchall()

    print 'Content-type:text/html\r\n\r\n'
    print '<html>'
    print '<head>'
    print '<title>My Application</title>'
    print '</head>'
    print '<body>'

    for row in rows:
        print '<h2>' + row[0] + '</h2>'

    print '</body>'
    print '</html>'

con.close()
```

5. **Dockerfile の確認:** `~/mywebcontainer` ディレクトリーにある Dockerfile ファイルを随時変更します (おそらく変更が必要なのは、名前を追加する際の `Maintainer_Name` のみ)。以下は、そのファイルの内容です。

```
# Webserver container with CGI python script
# Using RHEL 7 base image and Apache Web server
# Version 1

# Pull the rhel image from the local registry
FROM rhel7:latest
USER root

MAINTAINER Maintainer_Name

# Fix per https://bugzilla.redhat.com/show_bug.cgi?id=1192200
RUN yum -y install deltarpm yum-utils --disablerepo=*-eus-* --
disablerepo=*-htb-* \n      --disablerepo=*-ha-* --disablerepo=*-
rt-* --disablerepo=*-lb-* --disablerepo=*-rs-* --disablerepo=*-
sap-*

RUN yum-config-manager --disable *-eus-* *-htb-* *-ha-* *-rt-* *-
lb-* *-rs-* *-sap-* > /dev/null

# Update image
RUN yum update -y
RUN yum install httpd procps-ng MySQL-python -y

# Add configuration file
ADD action /var/www/cgi-bin/action
RUN echo "PassEnv DB_SERVICE_SERVICE_HOST" >>
/etc/httpd/conf/httpd.conf
RUN chown root:apache /var/www/cgi-bin/action
```

```

RUN chmod 755 /var/www/cgi-bin/action
RUN echo "The Web Server is Running" > /var/www/html/index.html
EXPOSE 80

# Start the service
CMD ["-D", "FOREGROUND"]
ENTRYPOINT ["/usr/sbin/httpd"]

```

6. **Web サーバーコンテナの構築:** Dockerfile ファイルとその他のコンテンツを含むディレクトリで、以下のコマンドを実行します。

```

# docker build -t webwithdb .
Sending build context to Docker daemon 4.096 kB
Sending build context to Docker daemon
Step 0 : FROM rhel7:latest
----> bef54b8f8a2f
Step 1 : USER root
----> Running in 00c28d347131
----> cd7ef0fc55
...

```

7. **Web サーバーコンテナの起動:** コンテナイメージを起動するには、以下のコマンドを実行します。

```
# docker run -d -p 80:80 --name=mywebwithdb webwithdb
```

8. **Web サーバーコンテナのテスト:** Web サーバーが動作することを確認するには、以下に示す最初の curl コマンドを実行します。バックエンドデータベースコンテナを実行している場合は、2 つ目のコマンドを実行します。

```

# curl http://localhost/index.html
The Web Server is Running
# curl http://localhost/cgi-bin/action
<html>
<head>
<title>My Application</title>
</head>
<body>
<h2>RedHat rocks</h2>
<h2>Success</h2>
</body>
</html>
</tt></pre>

```

ローカルホストに Web ブラウザーをインストールしている場合は、Web ブラウザーを開いて、出力内の数行がきちんと表示されていることを確認します。ブラウザーで以下の URL を開きます。 <http://localhost/cgi-bin/action>

2.1.3. コンテナを使用するためのヒント

ここでは、Web サーバーコンテナを使用するためのヒントを紹介します。

- ※ **MariaDB の変更:** MariaDB コンテナ (後述) でこのコンテナを使用するには、**action** スクリプトを編集し、IP アドレスを 172.17.42.1 から docker0 インターフェースのホスト IP に変更しなければならない場合があります。そのアドレスがホストにあることを確認するには、以下のコマンドを実行します。

```
# ip a | grep docker0 | grep inet
    inet 172.17.42.1/16 scope global docker0
```

- ※ **コンテンツの追加:** docker run コマンドラインに -v オプションを使用すると、ローカルホストからマウントした独自のコンテンツを組み込むことができます。たとえば、次のようになります。

```
# docker run -d -p 80:80 -v /var/www/html:/var/www/html \n      --
name=mywebwithdb webwithdb
```

2.1.4. 添付ファイル

- ※ [Apache Web container tar file: action CGI script and Dockerfile](#)

2.2. MARIADB コンテナのインストールおよびデプロイ

2.2.1. 概要

MariaDB を使用して、他のアプリケーションからアクセスできる基本的なデータベースをコンテナにセットアップすることができます。本書では以下の手順について説明します。

- ※ docker フォーマットのコンテナに MariaDB データベースサーバーを構築する
- ※ ホストのポート 3306 でサービスを公開する
- ※ データベースサービスを起動して情報を共有する
- ※ Web サーバーからデータベースをクエリーするためのスクリプトを実行できるようにする (後述の追加 Web サーバーコンテナが必要)
- ※ このコンテナを使用し、拡張するためのヒント

2.2.2. MariaDB データベースサーバーコンテナの作成および実行

1. **システムのインストール:** docker パッケージを含む Red Hat Enterprise Linux 7 または Red Hat Enterprise Linux Atomic Host システムをインストールし、docker サービスを起動します。
2. **イメージの取得:** 以下のコマンドを実行して rhel7 イメージを取得します。

```
# docker pull rhel7:latest
```

3. **サポートファイルが含まれる tarball の取得:** 本書に添付されている tarball ファイル ([mariadb_cont.tgz](#)) をダウンロードします。新規の **mydbcontainer** ディレクトリーにダウンロードして、以下のように untar を実行します。

```
# mkdir ~/mydbcontainer
# cp mariadb_cont*.tgz ~/mydbcontainer
```

```
# cd ~/mydbcontainer
# tar xvf mariadb_cont*.tgz
gss_db.sql
Dockerfile
```

4. **Dockerfile の確認:** `~/mydbcontainer` ディレクトリーにある Dockerfile ファイルを随時変更します (おそらく変更が必要なのは、名前を追加する際の `Maintainer_Name` のみ)。以下は、そのファイルの内容です。

```
# Database container with simple data for a Web application
# Using RHEL 7 base image and MariahDB database
# Version 1

# Pull the rhel image from the local registry
FROM rhel7:latest
USER root

MAINTAINER Maintainer_Name

# Update image
RUN yum update -y

# Add Mariahdb software
RUN yum -y install mariadb-server

# Set up Mariahdb database
ADD gss_db.sql /tmp/gss_db.sql
RUN /usr/libexec/mariadb-prepare-db-dir
RUN /usr/bin/mysqld_safe --basedir=/usr & \n    sleep 10s && \n
/usr/bin/mysqladmin -u root password 'redhat' && \n    mysql --
user=root --password=redhat < /tmp/gss_db.sql && \n    mysqladmin
shutdown --password=redhat

# Expose Mysql port 3306
EXPOSE 3306

# Start the service
CMD ["--basedir=/usr"]
ENTRYPOINT ["/usr/bin/mysqld_safe"]
```

5. **データベースサーバーコンテナの構築:** Dockerfile ファイルとその他のコンテンツを含むディレクトリーで、以下のコマンドを実行します。

```
# docker build -t dbforweb .
Sending build context to Docker daemon 528.4 kB
Sending build context to Docker daemon
Step 0 : FROM rhel7:latest
---> bef54b8f8a2f
Step 1 : USER root
...
```

6. **データベースサーバーコンテナの起動:** コンテナイメージを起動するには、以下のコマンドを実行します。

```
# docker run -d -p 3306:3306 --name=mydbforweb dbforweb
```

7. **データベースサーバーコンテナのテスト:** たとえば、ホストの docker0 インターフェイスが 172.17.42.1 の場合 (これと異なる場合もある)、以下のように **nc** コマンド (RHEL 7 の場合は **yum install nc** と入力する) を実行して、データベースコンテナが動作することを確認します。

```
# nc -v 172.17.42.1 3306
Ncat: Version 6.40 ( http://nmap.org/ncat )
Ncat: Connected to 172.17.42.1:3306.
R
5.5.40-MariaDB?acL3YF31?X?FWbiiTI02Kd6mysql_native_password Ctrl-
C
```

2.2.3. コンテナを使用するためのヒント

ここでは、Web サーバーコンテナを使用するためのヒントを紹介します。

- ※ **独自のデータベースの追加:** お使いのデータベースファイルをビルドディレクトリーにコピーし、データベースファイル名を `gss_db.sql` からお使いのデータベース名に (Dockerfile ファイル内の複数箇所) 変更することにより、独自の MariaDB コンテンツを組み込むことができます。
- ※ **コンテナのオーケストレーション:** このコンテナを他のコンテナと共に管理する効果的な方法として、Kubernetes を使用して Pod にそれらをオーケストレーションすることができます。

2.2.4. 添付ファイル

- ※ [Tar file containing gss_db.sql database and Dockerfile files for MariaDB container](#)