



# Red Hat Enterprise Linux Atomic Host 7

## CLI リファレンス

Atomic CLI リファレンス



# Red Hat Enterprise Linux Atomic Host 7 CLI リファレンス

---

## Atomic CLI リファレンス

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/CLI\_Reference.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

atomic コマンドラインツールのガイド

---

# 目次

1. 非推奨の通知 .....	2
第1章 概要 .....	3
第2章 前提条件 .....	4
第3章 ATOMIC コマンド .....	5
3.1. Atomic Host	5
3.2. アトミックな差分	6
3.3. atomic install	7
3.4. アトミックアンインストール	8
3.5. atomic run	8
3.6. atomic stop	8
3.7. Atomic イメージ	9
3.8. Atomic コンテナ	11
3.9. atomic info	12
3.10. atomic top	12
3.11. アトミックマウント	13
3.12. atomic unmount	14
3.13. atomic pull	14
3.14. atomic push	15
3.15. アトミックストレージ (移行)	15
3.16. アトミックスキャン	16
3.17. atomic sign	18
3.18. atomic verify	18
3.19. アトミックの信頼	18
3.20. atomic update	19
3.21. atomic --help および man ページ	20

## 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 概要

**atomic** コマンドラインツールは、Atomic Host システムおよびコンテナを操作し、管理する方法を提供します。これにより、システムに対する高レベルかつ一貫したエントリーポイントが提供され、スーパー特権コンテナやデバッグツールなどの特殊な種類のコンテナとの対話が容易になります。

**atomic** コマンドは、**docker**、**ostree**、**skopeo** などのツールを使用してコンテナおよびコンテナのホストシステムを管理します。また、**atomic** コマンドに組み込まれており、**docker** コマンドで利用できない多くの機能もあります。これらの機能により、イメージの署名、イメージ検証、コンテナのインストール機能、ファイルシステムのマウント、権限のオープンに特別なコマンドを使用できます。

**LABEL** について: Dockerfiles は、**atomic** が読み取り、実行できる一部のコマンドのデフォルト値の保存をサポートします。これらは "LABEL" 命令と呼ばれ、イメージを独自の推奨値で簡単に提供でき、複雑な **docker** コマンドの実行が容易になります。たとえば、Dockerfile に **LABEL RUN** が含まれる場合は、**atomic run <image>** を実行してそのコンテンツを実行します。ラベルを使用する **アトミック** のコマンドは **install**、**アンインストール**、**mount** のマウント、**をアンマウントし**、**を実行し**、**を停止** します。

## 第2章 前提条件

- RHEL Atomic Host では、**atomic** は OSTree の一部であり、使用できる状態になります。
- Red Hat Enterprise Linux では、以下がカバーされていることを確認します。
  - **atomic** パッケージを提供する Extras チャンネルにシステムをサブスクライブします。

Red Hat Subscription Management の場合は、以下のコマンドを実行します。

```
# subscription-manager repos --enable rhel-7-server-extras-rpms
```

Satellite を使用している場合は、以下を実行します。

```
# rhn-channel --add --channel rhel-x86_64-server-extras-7
```

- Yum を使用して **atomic** をインストールします。

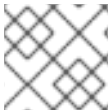
```
# yum install atomic
```

- **docker サービス** が実行していることを確認します。

```
# systemctl status docker
```

出力に **inactive** が表示される場合は、以下のコマンドを使用します。

```
# systemctl start docker
```



### 注記

どちらのシステムでも、**atomic** を使用するには root 権限が必要です。



## 第3章 ATOMIC コマンド

### 3.1. Atomic Host

このサブコマンドは、rpm-ostree の高レベルのラッパーで、アップグレード、ロールバック、およびシステム状態の検査を実行します。

- **Atomic Host のステータス**

すべてのデプロイメントに関する情報を表示します。アスタリスク(\*)は現在実行中のデプロイメントをマークします。

```
# 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/autobrew/buildmaster
  Version: 7.2.7 (2016-09-15 22:28:54)
  BaseCommit: dbbc8e805f0003d8e55658dc220f1fe1397caf80221cc050eeb1bbf44bef56a1
  Commit: 5cd426fa86bd1652ecd8f7d489f89f13ecb7d36e66003b0d7669721cb79545a8
  OSName: rhel-atomic-host
```

- **Atomic host rollback**

次の起動時にインストールされた他のツリーに切り替えます。-r オプションを使用して、ロールバックの完了後に再起動を開始できます。

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

- **Atomic Host のアップグレード**

利用可能な場合は、最新の OSTree にアップグレードします。これには数分の時間がかかる場合があります。完了すると、**変更された**、**削除された**パッケージの完全なリストが表示されます。新たにダウンロードしたツリーは、次の再起動時に自動的に起動します。

- **Atomic host deploy**

OSTree の特定のバージョンを指定してデプロイできます。このコマンドは、インストールされている 2 つの OSTree 間のみでの代替となるため、**アップグレード** または **ロールバック** よりも柔軟性があります。新たにダウンロードしたツリーは、現在デプロイされていないツリーを置き換えます。構文は次のとおりです。

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

たとえば、以下のコマンドを使用して 7.2.1 OSTree をデプロイし、ツリーのダウンロード後に再起動を開始します。

```
# atomic host deploy 7.2.1 -r
```

`--preview` オプションを使用して、現在デプロイされているツリーと、指定したツリーとパッケージの違いを確認します。

```
# atomic host deploy 7.2.1 --preview
```

バージョン番号が不明な場合は、以下の `ostree` コマンドを使用して、サブスクライブしているリポジトリのコミット履歴をプルします。

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

バージョン番号がある場合は、`atomic host <version> --preview` コマンドを使用して、パッケージの違いを確認することができます。

システムに最大 2 つのデプロイメントを設定できます。**アップグレード** または **デプロイ** は、新しいツリーをダウンロードし、現在デプロイされていないツリーを置き換えます。その後、**ロールバック** を使用して、システム上の両方のツリーを交換できます。

特定のバージョンのコミット ID を使用することもできます。カスタマーポータルからのソリューションには、リリースされたすべてのコミット ID の一覧が含まれています。[Red Hat Enterprise Linux Atomic Host の特定のバージョンをデプロイ](#) します。

## 3.2. アトミックな差分

ファイルレベルで 2 つのイメージまたはコンテナを比較し、その違いの詳細の一覧を表示します。デフォルトでは、ファイルの完全なリストが表示されます。

```
atomic diff <image1> <image2>
```

複数のオプションの組み合わせで出力を変更できます。

```
# atomic diff --rpms --no-files rhel7 centos

rhel7                | centos
-----|-----
Red Hat Enterprise Linux Server | CentOS Linux release 7.2.1511 (
release 7.2 (Maipo)      | Core)
-----|-----
                        | bind-license-32-9.9.4
                        | centos-release-0-7
dmidecode-1-2.12      |
gdb-gdbserver-0-7.6.1 |
                        | hostname-0-3.13
                        | iputils-0-20121221
libnl-0-1.1.4         |
libxml2-python-0-2.9.1 |
m2crypto-0-0.21.1     |
python-dateutil-0-1.5 |
python-dmidecode-0-3.10.13 |
python-ethtool-0-0.8  |
python-rhsm-0-1.15.4  |
redhat-release-server-0-7.2 |
subscription-manager-0-1.15.9 |
```

```

usermode-0-1.111 | tar-2-1.26
virt-what-0-1.13 |
                  | yum-plugin-fastestmirror-0-1.1.31

```

--rpms オプションは、2つのイメージの RPM 間で違いがあるテーブルを追加します。--no-files オプションと組み合わせると、そのテーブルのみを出力するように出力を制限できます。



#### 警告

--no-files オプションは出力を生成しないため、単独で使用しないでください。

--names-only オプションは、バージョンなしでパッケージ名のみを比較します。

ファイルの完全なリストが長すぎてターミナルが切り捨てられる可能性があるため、出力をより小さいテキストビューアーにリダイレクトすることが推奨されます。

```
# atomic diff rhel7 centos | less
```

--json オプションを使用して JSON 形式で出力し、リダイレクトします（例：未滿）。

```
# atomic diff --rpms --json rhel7 centos | less
```

### 3.3. atomic install

```
atomic install <image>
```

イメージのインストール方法を実行します。install メソッドは、コンテナイメージの LABEL **INSTALL** フィールドで説明されています。これは通常、イメージを実行するホストシステムを準備するために使用されます。イメージが削除された場合に編集および保存できるように、イメージに必要な設定ファイルをホストに公開することがよくあります。たとえば、以下のインストール方法は以下のようになります。

```

# atomic images info rhel7/rsyslog
[output truncated]
INSTALL: docker run --rm --privileged -v /:/host -e HOST=/host -e IMAGE=IMAGE -e NAME=NAME
IMAGE /bin/install.sh

```

以下のコマンドを実行します。

```

# atomic install rhel7/rsyslog
docker run --rm --privileged -v /:/host -e HOST=/host -e IMAGE=rhel7/rsyslog -e NAME=rsyslog
rhel7/rsyslog /bin/install.sh

```

この命令により、atomic install はホスト上の root ディレクトリー(/)からコンテナ内の /host/ ディレクトリーにファイルをマウントし、\$HOST 変数をコンテナ内の /host/ に設定します。たとえば、コンテナの /usr/bin は /host/usr/bin で、\$IMAGE は rhel7/rsyslog、\$NAME は rsyslog で

す。`/bin/install.sh` スクリプトは、`/etc/rsyslog.conf` ファイルをホストシステムに公開するため、コンテナ外から編集できます。

イメージがローカルにない場合、`atomic install` は設定済みのレジストリーからイメージをプルします。`--display` オプションを使用して、イメージのインストール方法を表示します。`--display` が指定されている場合、`install` コマンドは実行しません。

`-n` オプションを使用して、イメージの複数のコピーをインストールします。

```
# atomic install -n name1 rhel7/rsyslog
# atomic install -n name2 rhel7/rsyslog
```

### 3.4. アトミックアンインストール

```
atomic uninstall <image>
```

のインストールと同様に、アンインストールで `UNINSTALL` 命令からイメージのアンインストール方法を読み取り、実行します。

### 3.5. atomic run

```
atomic run <image>
```

イメージの `run` メソッドを実行します。`run` メソッドは、コンテナイメージの `RUN` フィールドで説明されています。`RUN` を使用すると、開発者は特定のアプリケーションを実行する方法を定義できます。たとえば、`ntpd` サービスを使用するコンテナには `--cap_add SYS_TIME` オプションが必要で、ユーザーは以下の `full` コマンドを入力する代わりに `RUN` ラベルにオプションを埋め込むことができます。

```
# docker run -d -n --cap_add SYS_TIME ntpd
```

`RUN` フィールドが存在しない場合、`atomic` の実行はデフォルトで以下を実行します。

```
docker create -ti -n <image_name> <container_name>
```

`--spc` オプションを使用して、コンテナをスーパー特権モードで実行します。Super-Privileged コンテナの詳細は、[Chapter 9 を参照してください](#)。RHEL Atomic Host Getting Started with Containers Guideの [Running Super-privileged Containers](#)

### 3.6. atomic stop

```
atomic stop <image_name>/<container_name>
```

イメージの `stop` メソッドを実行します。このコマンドを使用して、実行中のコンテナを停止します。イメージ名またはコンテナ名を引数として取ります。以下に例を示します。

```
# atomic stop cranky_wright
```

または

```
# atomic stop rhel7/rsyslog
```

### 3.7. Atomic イメージ

イメージに対してコマンドを実行します。イメージの表示、LABEL 情報、ヘルプファイルの表示、新しいバージョンの確認が可能です。

- **atomic images list**

システムにダウンロードしたコンテナイメージを一覧表示します。> 記号は、イメージがコンテナで使用されていることを示します。

```
# atomic images list

REPOSITORY                                TAG  IMAGE ID  CREATED  VIRTUAL
SIZE
registry.access.redhat.com/rhel7/openscap  latest sha256:da0d5 2016-06-20 14:24
363.37 MB
> registry.access.redhat.com/rhel7/sadc    latest sha256:7ed99 2016-05-08 16:31
215.23 MB
> registry.access.redhat.com/rhel7/kubernetes-controller-mgr latest sha256:feb3d 2016-05-06
20:12 347.29 MB
> registry.access.redhat.com/rhel7/kubernetes-apiserver  latest sha256:c3ac0 2016-05-06
20:12 347.29 MB
registry.access.redhat.com/rhel7/kubernetes-scheduler  latest sha256:d6c72 2016-05-06
20:12 347.29 MB
> registry.access.redhat.com/rhel7/cockpit-ws          latest sha256:f1ea2 2016-05-06 18:54
220.3 MB
registry.access.redhat.com/rhel7/rhel-tools            latest sha256:00211 2016-05-06 17:49
1.27 GB
> registry.access.redhat.com/rhel7/rsyslog             latest sha256:92bd7 2016-05-06 17:40
215.93 MB
```

- **atomic images delete**

```
# atomic images delete <image>
```

システムから指定したイメージを削除します。デフォルトでは、コンテナをベースとするイメージを削除することはできません。-f オプションを使用して、そのイメージを強制的に削除します。これにより、そのイメージに基づいて実行中のコンテナが停止しません。

```
# atomic images delete -f rhel7/rsyslog
```

--remote オプションを使用して、リモートリポジトリからイメージを削除します。ただし、リモートディスク領域は、レジストリーの **garbage-collection** コマンドがリモートレジストリーに対して実行されるまで解放されません。

- **Atomic images info**

```
atomic images info <image>
```

イメージに定義されたラベルを表示します。

```
$ atomic images info rhel7/rhel-tools
Image Name: registry.access.redhat.com/rhel7/rhel-tools:latest
License: GPLv3
```

```

architecture: x86_64
authoritative-source-url: registry.access.redhat.com
build-date: 2017-06-23T11:07:50.586605
com.redhat.build-host: ip-10-29-120-119.ec2.internal
com.redhat.component: rhel-tools-docker
description: The Red Hat Enterprise Linux Atomic Tools Container (RHEL Tools Container) is a
docker-formatted image that includes hundreds of software tools for troubleshooting and investigating
a Red Hat Enterprise Linux Atomic (RHEL Atomic) Host. Designed to run as a privileged container,
the RHEL Tools Container allows you to interact directly with the RHEL Atomic Host system to
uncover and solve problems. Inside the RHEL Tools Container are popular tools such as sosreport,
kdump, and many others (most of which are not included with RHEL Atomic).
distribution-scope: public
io.k8s.display-name: Red Hat Enterprise Linux Tools
io.openshift.tags: base rhel7
name: rhel7/rhel-tools
release: 29
run: docker run -it --name NAME --privileged --ipc=host --net=host --pid=host -e HOST=/host -e
NAME=NAME -e IMAGE=IMAGE -v /run:/run -v /var/log:/var/log -v /etc/machine-id:/etc/machine-id -v
/etc/localtime:/etc/localtime -v /:/host IMAGE
summary: Provides the latest release of Red Hat Enterprise Linux 7 in a fully featured and supported
base image.
vcs-ref: c046119740d92e03ba17e27e806e46865fe85838
vcs-type: git
vendor: Red Hat, Inc.
version: 7.3

```

デフォルトでは、ローカルイメージを最初にチェックしてから、システムに設定されたレジストリーを試行します。--remote オプションを使用してローカルイメージを無視し、設定されたレジストリーのみを検索します。

```
# atomic images info --remote rhel7/rhel-tools
```

--storage オプションを使用して、イメージのストレージを指定します。

```
$ atomic images info --storage ostree rhel7/rhel-tools
```

- **atomic images prune**

**atomic images prune** を使用して、未使用の **dangling** イメージを削除して、ディスク領域を解放します。イメージの変更とは、名前やタグがなく、他のイメージで使用されていないイメージです。これらは使用されていないため、システム領域を占有します。イメージの変更は通常、**docker build** コマンドを使用して、古いバージョンのイメージも削除せずにイメージを更新することによって生じます。アスタリスク(\*)は、dangling イメージを示します。

```
# atomic images list -a
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
registry.access.redhat.com/rhel7/openscap		latest sha256:da0d5	2016-06-20 14:24	363.37 MB
> registry.access.redhat.com/rhel7/sadc		latest sha256:7ed99	2016-05-08 16:31	215.23 MB
> registry.access.redhat.com/rhel7/kubernetes-controller-mgr	latest	sha256:feb3d	2016-05-06 20:12	347.29 MB
> registry.access.redhat.com/rhel7/kubernetes-apiserver	latest	sha256:c3ac0	2016-05-06	

```

20:12 347.29 MB
registry.access.redhat.com/rhel7/kubernetes-scheduler latest sha256:d6c72 2016-05-06
20:12 347.29 MB
*<none> <none> sha256:bad41 2016-05-06 17:55 125.08 MB
*<none> <none> sha256:9339b 2016-05-06 23:55 125.08 MB
> registry.access.redhat.com/rhel7/cockpit-ws latest sha256:f1ea2 2016-05-06 18:54
220.3 MB
registry.access.redhat.com/rhel7/rhel-tools latest sha256:00211 2016-05-06 17:49
1.27 GB
> registry.access.redhat.com/rhel7/rsyslog latest sha256:92bd7 2016-05-06 17:40
215.93 MB

```

- **Atomic イメージの検証**

```
atomic images verify <image>
```

新しいイメージが利用可能かどうかを確認します。また、すべてのレイヤーをスキャンして、サブ層に新しいバージョンが利用可能かどうかを確認します。

**-v** オプションを使用して、すべてのレイヤーの情報を表示します。

```

# atomic images verify -v rhel7/rhel-tools

rhel7/rhel-tools contains the following images:

NAME                LOCAL VERSION    REMOTE VERSION    DIFFERS
rhel7/rhel-tools    7.3-29           7.3-29            NO

```

**--storage** オプションを使用して、イメージのストレージを指定します。

```
$ atomic images verify --storage ostree rhel7/rhel-tools
```

**--no-validate** オプションを使用して、システムイメージの検証を無効にします。

```
atomic images verify --no-validate --storage ostree registry.access.redhat.com/rhel7/etcd
```

- **Atomic イメージバージョン**

```
atomic images version <image>
```

イメージの Name Version Release ラベルを表示します。

```

# atomic version rhel7/rsyslog
00b31ffda5e92737fe07aecaa972d6fb4bda7cc8eca225f6a12e06db1ac5ba39 rhel7/rsyslog-7.1-29
registry.access.redhat.com/rhel7/rsyslog:latest

```

### 3.8. Atomic コンテナ

コンテナでコマンドを実行します。このコマンドとサブコマンドを使用すると、現在実行中のコンテナの一覧表示、削除、またはトリミングを行うことができます。

- **Atomic containers list**

```
# atomic containers list
CONTAINER ID IMAGE          COMMAND                CREATED   STATUS    RUNTIME
flannel   rhel7/flannel    /usr/bin/flanneld-ru 2016-10-06 14:36 running  runc
etcd      rhel7/etcd      /usr/bin/etcd-env.sh 2016-10-13 14:21 running  runc
1cf730472572 rhel7/cockpit-ws /container/atomic-ru 2016-10-13 17:55 running  Docker
```

コンテナが使用するランタイム、**Docker**、または **runc** など、システムで **実行中** のコンテナとその情報を一覧表示します(**docker ps** は Docker 形式のコンテナのみを一覧表示します)。**atomic containers list -a** はすべてのコンテナを表示します。

```
# atomic containers list -a
CONTAINER ID IMAGE          COMMAND                CREATED   STATUS    RUNTIME
etcd      rhel7/etcd      /usr/bin/etcd-env.sh 2016-10-13 14:21 running  runc
flannel   rhel7/flannel    /usr/bin/flanneld-ru 2016-10-13 15:12 failed   runc
1cf730472572 rhel7/cockpit-ws /container/atomic-ru 2016-10-13 17:55 exited  Docker
9a2bb24e5978 rhel7/rsyslog    /bin/rsyslog.sh      2016-10-13 17:49 created  Docker
34f95af8f8f9 rhel7/cockpit-ws /container/atomic-ru 2016-09-27 19:10 exited  Docker
```

Atomic **containers** 一覧 では、**-f** オプションを使用した出力のフィルターリングもサポートされます。フィルターは、**コンテナ ID**、**イメージ**、**コマンド**、**作成された**、**ステータス**、**ランタイム** です。以下に例を示します。

```
# atomic containers list -f status=exited
1cf730472572 rhel7/cockpit-ws /container/atomic-ru 2016-10-13 17:55 exited  Docker
34f95af8f8f9 rhel7/cockpit-ws /container/atomic-ru 2016-09-27 19:10 exited  Docker
```

- **Atomic コンテナの更新**

**Atomic コンテナの更新** により、コンテナをプルできる最新のコンテナを更新できます。以下に例を示します。

```
# atomic containers update rhel7/flannel
```

- **Atomic コンテナの削除**

指定したコンテナを削除します。以下に例を示します。

```
# atomic containers delete rhel7/flannel
```

- **Atomic コンテナのトリミング**

このコマンドは、実行中のコンテナからの未使用のブロックを破棄します。ファイルシステムによって使用されていないブロックを破棄する **fstrim** コマンドを使用します。これは、RHEL Atomic Host で使用されるオプションである Thinly-Provisioned ストレージに特に便利です。このコマンドを 1 週間後に使用して、未使用のファイルシステムブロックからシステムをクリーンアップします。詳細は、**fstrim (8)** man ページを参照してください。

### 3.9. atomic info

**atomic info** コマンドは、コマンドの **atomic images** セットの下に移動されました。**atomic images info** の詳細は、「[Atomic イメージ](#)」を参照してください。

### 3.10. atomic top



**atomic top [<container>]**

top ユーティリティーなど、アクティブなコンテナで実行しているプロセスのインタラクティブなビューを表示します。デフォルトでは、**atomic top** はすべてのコンテナを監視しますが、オプションでコンテナ名または ID を使用して必要なコンテナのみを指定できます。デフォルトフィールドの表は以下のようになります。

**ATOMIC TOP**

CONTA(I)NER*	(N)AME	(P)ID	(C)PU	(M)EM	(U)ID	(G)ID	CMD
ec56d2f1fb10	httpd	2087	0.0	0.2	0	0	httpd -DFOREGROUND
ec56d2f1fb10	httpd	2095	0.0	0.1	1	1	httpd -DFOREGROUND
ec56d2f1fb10	httpd	2096	0.0	0.1	1	1	httpd -DFOREGROUND
ec56d2f1fb10	httpd	2097	0.0	0.1	1	1	httpd -DFOREGROUND
fa7586391e42	fedora	1913	0.0	0.1	0	0	/bin/sh

列ヘッダーから括弧内のキーを押すと、プロセスをソートできます。たとえば、P を押して、PID でプロセスを分類します。

```
# atomic top -d 5 -n 3
```

このコマンドを使用すると、3 回の反復で 5 秒間隔でプロセスを監視できます。

デフォルトのフィールドにフィールドを追加するには、親 PID や UID などの **--optional** オプションを使用します。

```
# atomic top --optional ppid uid
```

**3.11. アトミックマウント**

```
atomic mount <image> <mountpoint>
```

コンテナまたはイメージの基礎となるファイルシステムをホストファイルシステムにマウントします。これにより、コンテンツを検査できます。たとえば、これを使用して設定ファイルを確認することができます。

イメージの UUID、コンテナ UUID、コンテナ NAME、またはイメージリポジトリの 1 つ（オプションでレジストリーおよびタグ情報を使用）のいずれかを受け入れます。指定した UUID または NAME がコンテナであり、**--live** オプションが設定されていない場合には、**atomic mount** は一時的なイメージにコミットし、そのイメージから一時コンテナを生成することで、コンテナのスナップショットを作成します。UUID または REPO がイメージを参照する場合、**atomic mount** は指定されたイメージから一時的なコンテナを作成します。一時的なアーティファクトはすべてアンマウント時にクリーンアップされます。

```
# mkdir /root/tmp
# atomic mount rhel7/rsyslog /root/tmp
# cd /root/tmp
# ls
```



## 注記

アトミックマウントは、**devicemapper** および **overlayfs** ストレージバックエンドでのみサポートされます。

### 3.12. atomic unmount

```
atomic unmount <mountpoint>
```

**atomic mount** で以前にマウントされたコンテナまたはイメージをアンマウントします。

```
# atomic unmount /root/tmp
```

### 3.13. atomic pull

```
atomic pull <image>
```

リポジトリからイメージを取得し、システムにダウンロードします。

```
# atomic pull rhel7/rsyslog
```

**source:image** 形式を使用してソースを指定することもできます。ソースの場合、以下のオプションがあります。

- **OCI - skopeo** ツールを使用して、Docker レジストリーからイメージを取得します。これは、ソースが指定されていない場合に想定されるデフォルトのオプションです。デフォルトを変更するには、**default\_storage** キーワードで **/etc/atomic.conf** ファイルを編集します。

```
# atomic pull oci:rhel7/etcd
```

- **Docker**: ネットワークにアクセスせずに、ローカルの Docker レジストリーからイメージをインポートします。これは、**docker (docker save IMAGE)** のイメージを保存し、OSTree リポジトリにインポートすることに相当します。

```
# atomic pull --storage=ostree docker:fedora
```

- **dockertar**: ローカルの Docker レジストリーから tarball をインポートします。

```
# atomic pull --storage=ostree dockertar:path/to/image.tar
```

- **OSTree**: リモートの OSTree リポジトリからイメージを取得します。リモートは、ローカルの OSTree リポジトリですすでに設定されている必要があります。

```
# atomic pull --storage=ostree ostree:<remote>/branch
```

**--storage** オプションを使用して、イメージの宛先ストレージを指定します。2つのオプションは **docker** と **ostree** です。指定しない場合、コマンドは **docker** であることを前提としています。システムコンテナイメージをプルする場合は、**ostree** オプションを使用します。

```
# atomic pull --storage=ostree rhel7/etcd
```

**--type** オプションを使用して、別のレジストリータイプを指定します。レジストリーの **アトミック** タイプに切り換えることができます。以下に例を示します。

```
# atomic pull --type atomic <atomic_registry_address>:namespace/image
```

### 3.14. atomic push

```
atomic push <new_image>
```

ローカルにビルドしたイメージをリポジトリにプッシュします。デフォルトの動作では、docker リポジトリにプッシュされますが、**--satellite** または **--pulp** オプションを使用して Satellite リポジトリまたは Pulp リポジトリにプッシュするように設定することもできます。

### 3.15. アトミックストレージ（移行）

コンテナストレージを管理します。

- **Atomic ストレージのエクスポート/インポート**

export コマンドおよび import コマンドを使用すると、すべてのイメージ、ボリューム、およびコンテナをあるバージョンの atomic から別のバージョンに移行するか、またはあるストレージバックエンドから別のストレージバックエンドに移行できます。**atomic export** を使用すると、現在のアトミックインスタンスのすべてのデータを保存し、環境を変更してから、**atomic import** ですべての古いデータを新規システムにインポートできます。このコマンドは以前は migrate と呼ばれていました。

```
# atomic storage export
```

`/images/`、`/volumes/`、および `/containers/` サブディレクトリー下の現在のイメージ、ボリューム、およびコンテナをすべて `/var/lib/atomic/migrate/`（または他の指定されたディレクトリー）にエクスポートします。

```
# atomic storage import
```

`/var/lib/atomic/migrate/` に以前に保存されたイメージ、ボリューム、およびコンテナ、または別の指定されたディレクトリーを新規の atomic インスタンスにインポートします。

カスタムの場所(`/var/lib/docker/`ではなく)から docker を実行している場合は、カスタムの場所を指定する **--graph** オプションを設定する必要があります。データを標準以外のディレクトリーに保存するには、**--dir** オプションを使用します。

- **Atomic ストレージの変更**

デフォルトのストレージ設定を変更します。

ブロックデバイスをストレージプールに追加できます。このコマンドは、ブロックデバイスを追加して devicemapper ストレージプールを拡張します。devicemapper ドライバーでのみ機能します。以下に例を示します。

```
# atomic storage modify --add-device vda3/rhelah-expand
```

バックエンドストレージドライバーを変更するには、**--driver** オプションを使用します。サポートされるドライバーは `devicemapper`、`overlay`、および `overlay2` です。

```
# atomic storage modify --driver overlay2
```

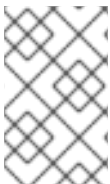
- **Atomic Storage のリセット**

このコマンドは、システムからすべてのコンテナおよびイメージを削除し、ストレージ設定をデフォルト値にリセットします。

### 3.16. アトミックスキャン

```
atomic scan <image>/<container>
```

イメージおよびコンテナで Common Vulnerabilities and Exposures (CVE)をスキャンします。デフォルトでは、**atomic** スキャンは **openscap** スキャナーを使用してイメージをスキャンしますが、プラグ可能な設計ではカスタムを含む追加のスキャナーの追加がサポートされます。**atomic scan** を初めて実行すると、**openscap** スキャナーを提供する **rhel7/openscap** コンテナをダウンロードします。**openscap** のデフォルトのスキャンタイプは、脆弱性をチェックすることです。**openscap** は、RHEL ベースのイメージおよびコンテナでのみ機能することに注意してください。



#### 注記

**openscap** コンテナを適切に使用するには、**atomic install rhel7/openscap** を実行してから、そのコンテナを **atomic scan** で使用します。詳細は、Atomic Release Notes の [既知の問題](#) の **atomic scan issue** を参照してください。

たとえば、rhel7 ベースイメージをスキャンするには、以下を実行します。

```
# atomic scan rhel7/rhel
```

コンテナおよびイメージをすべてスキャンして詳細なレポートを作成するには、以下を実行します。

```
# atomic scan --all --verbose
```

結果が正である場合、出力は以下のようになります。

```
# atomic scan rhel7/rhel
docker run -it --rm -v /etc/localtime:/etc/localtime -v /run/atomic/2016-06-21-10-10-28-942890:/scanin
-v /var/lib/atomic/openscap/2016-06-21-10-10-28-942890:/scanout:rw,Z -v /etc/oscaped:/etc/oscaped:ro
rhel7/openscap oscaped-evaluate scan --no-standard-compliance --targets chroots-in-dir:///scanin --
output /scanout
```

```
rhel7/rhel (sha256:bf203442)
```

The following issues were found:

```
RHSA-2016:1025: pcre security update (Important)
Severity: Important
RHSA URL: https://rhn.redhat.com/errata/RHSA-2016-1025.html
RHSA ID: RHSA-2016:1025-00
Associated CVEs:
CVE ID: CVE-2015-2328
CVE URL: https://access.redhat.com/security/cve/CVE-2015-2328
CVE ID: CVE-2015-3217
CVE URL: https://access.redhat.com/security/cve/CVE-2015-3217
```

```

CVE ID: CVE-2015-5073
CVE URL: https://access.redhat.com/security/cve/CVE-2015-5073
CVE ID: CVE-2015-8385
CVE URL: https://access.redhat.com/security/cve/CVE-2015-8385
CVE ID: CVE-2015-8386
CVE URL: https://access.redhat.com/security/cve/CVE-2015-8386
CVE ID: CVE-2015-8388
CVE URL: https://access.redhat.com/security/cve/CVE-2015-8388
CVE ID: CVE-2015-8391
CVE URL: https://access.redhat.com/security/cve/CVE-2015-8391
CVE ID: CVE-2016-3191
CVE URL: https://access.redhat.com/security/cve/CVE-2016-3191

```

Files associated with this scan are in /var/lib/atomic/openscap/2016-06-21-10-10-28-942890.

設定済みのスキャナーの一覧を表示するには、以下を使用します。

```

# atomic scan --list
Scanner: openscap *
Image Name: rhel7/openscap
Scan type: cve *
Description: Performs a CVE scan based on known CVE data

Scan type: standards_compliance
Description: Performs a standards scan

```

\* denotes defaults

この出力では、各スキャナーで利用可能なスキャンタイプを確認することもできます。OpenSCAP には 2 つの定義があり、`--scan_type` オプションを使用して両方の間でスイッチすることができます。

```

# atomic scan --scan_type standards_compliance rhel7/rhel
docker run -it --rm -v /etc/localtime:/etc/localtime -v /run/atomic/2016-07-12-16-08-03-011887:/scanin -v /var/lib/atomic/openscap/2016-07-12-16-08-03-011887:/scanout:rw,Z -v /etc/oscaped:/etc/oscaped:ro rhel7/openscap oscaped-evaluate scan --targets chroots-in-dir:///scanin --output /scanout --no-cve-scan

```

rhel7 (sha256:5fbb7430)

The following issues were found:

```

Ensure Software Patches Installed
Severity: Important
XCCDF result: notchecked

```

Files associated with this scan are in /var/lib/atomic/openscap/2016-07-12-16-08-03-011887.

新規スキャナーを追加するだけで、そのスキャナーに `atomic install` を提供し、これがローカルでカスタムのイメージである場合は以下を使用します。

```

# atomic install localhost:5000/custom_scanner

```

`--scanner` オプションを指定して、新しいスキャナーを使用できます。

```
# atomic scan --scanner custom_scanner rhel7/rhel
```

デフォルトのスキャナーを変更するには、`/etc/atomic.conf` の `default_scanner` 行を編集します。この行を使用して、`openscap` を明示的にデフォルトとして設定することもできます。これが明示的に設定されていない場合、`atomic scan` は `openscap` を使用します。

```
default_scanner: custom_scanner
```

`atomic` スキャンのもう1つの機能は、ホストファイルシステムもスキャンできることです。これは、`--rootfs` オプションを使用してホストにパスを提供するように設定できます。以下に例を示します。

```
# atomic scan --rootfs /tmp/chroot
```

### 3.17. atomic sign

```
# atomic sign <registry>/<image>
```

レジストリーからプルされた1つ以上のローカルイメージのローカル署名を作成します。デフォルトでは、署名は、`registry_confdir` キーワードを使用して `/etc/atomic.conf` ファイルに設定されたレジストリー設定ファイルから派生したディレクトリーに書き込まれます。



#### 警告

イメージが含まれるリモートレジストリーを信頼する場合にのみ `atomic sign` を使用します。これは管理するレジストリーであることが推奨されます。

`-d` オプションを使用して、デフォルトの場所とは異なる署名を保存します。

```
# atomic sign -d /tmp/signatures myregistry.example.com/my_image
```

`/etc/atomic.conf` ファイルで指定されたデフォルトのアイデンティティーに `--sign-by` オプションを使用して、`--gnupghome` を使用してその ID のキーリングに場所を指定します。

```
# atomic sign --sign-by user@example.com --gnupghome=~/.gnupg  
myregistry.example.com/my_image
```

イメージの署名に関する詳細は、Red Hat Enterprise Linux Atomic Host Managing Containers Guide の [Signing Container Images](#) の章を参照してください。

### 3.18. atomic verify

`atomic verify` コマンドは、コマンドの `atomic images` セットの下に移動されました。`atomic images verify` の説明は、「[Atomic イメージ](#)」を参照してください。

### 3.19. アトミックの信頼

`atomic trust` コマンドは、ホストシステムの信頼ポリシーを管理します。信頼ポリシーは

`/etc/containers/policy.json` ファイルに保存され、レジストリーまたはリポジトリーの範囲、または公開鍵で署名する必要のある両方を定義します。信頼は、ユーザーがレジストリーからイメージをプルしようとするすると適用されます。

- `atomic trust show`

`/etc/containers/policy.json` ファイルの内容を表示します。

```
# atomic trust show
* (default)          accept
```

- Atomic 信頼のデフォルト

デフォルトの信頼ポリシーを管理します。 `accept` または `reject` コマンドを使用して、デフォルトの信頼ポリシーを有効または無効にします。

```
# atomic trust default reject
```

または

```
# atomic trust default accept
```

- `atomic trust add`

信頼ポリシーを更新します。公開鍵を追加するには、以下を使用します。

```
# atomic trust add --pubkeys /etc/pki/containers/foo@example.com --sigstore
https://server.example.com/foobar/sigstore/ <registry>/<image>
```

レジストリーから署名されていないすべてのイメージを受け入れるには、以下を実行します。

```
# atomuc trust add --type insecureAcceptAnything <registry>
```

- `atomic trust delete`

信頼スコープを削除します。以下に例を示します。

```
# atomic trust delete <registry>
```

イメージの署名に関する詳細は、Red Hat Enterprise Linux Atomic Host Managing Containers Guide の [Signing Container Images](#) の章を参照してください。

### 3.20. atomic update

```
atomic update <image>
```

設定されたリポジトリーからイメージの最新の更新をプルします。このイメージをベースとするコンテナが存在する場合、コンテナは古いイメージを引き続き使用します。 `--force` オプションを使用して、コンテナを削除します。出力例：

```
# atomic update rhel7/rsyslog
Using default tag: latest
00b31ffda5e9: Download complete
```

```
c4f590bbcbe3: Download complete
```

```
Status: Image is up to date for registry.access.redhat.com/rhel7/rsyslog:latest
```

### 3.21. atomic --help および man ページ

--help オプションは、atomic およびすべての atomic サブコマンドで本書で説明されているすべての atomic サブコマンドで利用できます。--help を使用して、使用メッセージと、すべての利用可能なオプションをサブコマンドに出力します。

RHEL Atomic Host には OSTree の man ページがないため、Red Hat Enterprise Atomic Tools コンテナを介して atomic および rpm-ostree の man ページにアクセスできます。以下のコマンドを使用します。

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

個々のコマンドはハイフンで区切られるため、以下の形式を使用します。

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

同じコマンドを使用して rpm-ostree の man ページにアクセスできます。