



# Red Hat Container Development Kit 2.0

## Container Development Kit インス トールガイド

---

インストールガイド

Robert Krátký                      Chris Negus  
Red Hat Developer Group Documentation  
Team



# Red Hat Container Development Kit 2.0 Container Development Kit インストールガイド

---

## インストールガイド

Robert Krátky  
rkratky@redhat.com

Chris Negus

## 法律上の通知

Copyright © 2017 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, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

Node.js ® is an official trademark of Joyent. Red Hat 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

Red Hat Container Development Kit のインストールガイド

---

## 目次

<b>第1章 RED HAT CONTAINER DEVELOPMENT KITのインストール</b> .....	<b>3</b>
1.1. RED HAT CONTAINER DEVELOPMENT KITのインストール	3
1.2. RED HAT CDK を実行する場所	3
1.3. RED HAT CDK の内容	3
1.4. CDK INSTALLATION TOOLS について	4
<b>第2章 CDK の MAC OS X へのインストール</b> .....	<b>5</b>
2.1. CDK の MAC OS X へのインストール	5
<b>第3章 CDK の MICROSOFT WINDOWS へのインストール</b> .....	<b>9</b>
3.1. CDK の MICROSOFT WINDOWS へのインストール	9
3.2. RED HAT CDK VAGRANT BOX の起動	13
<b>第4章 CDK の FEDORA または RED HAT ENTERPRISE LINUX へのインストール</b> .....	<b>15</b>
4.1. CDK の FEDORA または RED HAT ENTERPRISE LINUX へのインストール	15
<b>第5章 RHEL CDK の使用</b> .....	<b>22</b>
5.1. RHEL CDK の使用	22
<b>第6章 CDK に関連する問題のトラブルシューティング</b> .....	<b>29</b>
6.1. CDK に関連する問題のトラブルシューティング	29
6.2. CDK に関する問題のトラブルシューティング	30
6.3. WINDOWS システムの CDK のトラブルシューティング	31



# 第1章 RED HAT CONTAINER DEVELOPMENT KITのインストール

## 1.1. RED HAT CONTAINER DEVELOPMENT KITのインストール

Red Hat Enterprise Linux システムのコンテナ化されたアプリケーションの開発を検討されている場合、Red Hat Container Development Kit (CDK) は以下の点で役立ちます。

- ※ 独自のラップトップ、デスクトップまたはサーバーシステムにインストールする Red Hat コンテナ開発環境を提供する。
- ※ 大規模なデータセンター向けのコンテナの作成およびデプロイに使用されるものと同じコンテナ開発およびランタイムツールが組み込まれている。
- ※ ローカルシステムで実行される事前に設定された Vagrant box および Vagrantfile で仮想マシンを作成できる簡単なインストール方法を提供する。

本書では、お客様が選択されるシステムに、コンテナ開発用に設定された Vagrant 対応の Red Hat Enterprise Linux 仮想マシンをインストールする方法について説明します。さらに、OpenShift、Eclipse、および各種コマンドラインツールを使って Red Hat Enterprise Linux ベースのコンテナを開発する方法について説明します。

## 1.2. RED HAT CDK を実行する場所

Red Hat CDK は、ユーザーが他の作業に使用するものと同じコンピューターでコンテナ開発を行えるように設計されています。CDK をインストールするために使用できるシステムには以下が含まれます。

- ※ **Mac OS X:** Intel ベースの Apple Mac を使用して Red Hat CDK をインストールし、実行することができます。Mac には 4 GB 以上の RAM が必要で、10.10 などの最新の 64 ビットバージョンの Mac OS X (Yosemite) を実行できる必要があります。
- ※ **Microsoft Windows:** 64 ビットバージョンの Microsoft Windows を Red Hat CDK をインストールするために使用できます。Windows 7 以上が必要です。
- ※ **Red Hat Enterprise Linux:** Red Hat CDK を RHEL システムにインストールするには、Red Hat Enterprise Linux (RHEL) 7 の最新バージョンが推奨されます。4 GB 以上の RAM を搭載し、仮想化サポートのある 64 ビットのコンピューターが必要です。
- ※ **Fedora:** Red Hat CDK を Fedora にインストールするための仕様は、基本的に RHEL の仕様と同じです。この場合も 4 GB 以上の RAM 搭載の、仮想化サポートのある 64 ビットのコンピューターを用意してください。

ハードウェアおよびソフトウェアの詳細の要件については、各システムの Red Hat CDK インストール手順を確認してください。

## 1.3. RED HAT CDK の内容

RHEL CDK からの RHEL 7 仮想マシンが指定システムで実行されたら、内容を検索できます。一部のサービスおよびツールは仮想マシンの起動時に自動的に実行されますが、他のサービスおよびツールには設定が必要です。以下は、これらの一部の機能の一覧です。

- ※ **Docker:** Docker プロジェクトは、基本的なコンテナのフォーマットと、CDK に組み込まれるコンテナで使用する **docker** コマンドを開発します。CDK は、docker デーモンを RHEL 仮想

マシンの起動時に自動的に起動するように設定されます。**docker** コマンドを使用することにより、個別のコンテナの構築、実行、起動、停止、調査およびその他の方法での使用が可能になります。

- ※ **OpenShift**: コンテナ化されたバージョンの OpenShift Enterprise が CDK に組み込まれています。OpenShift コンテナが実行されると、ご使用のブラウザーから Web コンソールを使用したり、コンテナプロジェクトを開発するために **oc** コマンドを使ってコマンドラインから作業したりすることができます。
- ※ **Kubernetes**: いわゆる **pod** でコンテナのオーケストレーションを実行するために、CDK には Kubernetes クラスターを実行するために必要なすべての機能が組み込まれています。CDK では、Kubernetes は一体型の Kubernetes master (pod の管理を行う) および node (pod の実行を行う) として実行するようにセットアップされます。Kubernetes には、pod を複製し、アプリケーションをオンザフライで拡張する機能、各種サービスおよびコンテナのセットを相互に接続するための機能が含まれます。
- ※ **Eclipse**: Red Hat CDK に同梱される Eclipse Workbench には、[Linux Tools/Docker Tooling](#) プラグインが含まれます。これにより、CDK ユーザーは Web UI からコンテナを管理できます。

本書は、Red Hat CDK 仮想マシンの各種機能をインストールし、最初にこれらに接続するための基本情報を提供しますが、CDK の使用方法の詳細は、他のドキュメントを参照する必要があります。以下は参照ドキュメントの例になります。

- ※ **Container Development Kit Getting Started Guide** (作成中): Red Hat CDK をはじめて使用する際の手順を説明します。
- ※ **Container Development Guide** (作成中): 高度なコンテナ開発についての手引きです。このガイドは、Docker、Kubernetes、Nuclecule、Atomic その他のコンテナのランタイム環境で実行するコンテナを作成する各種の方法を解説します。
- ※ **Getting Started with Containers (コンテナの使用ガイド)**: コンテナを実行するために Docker および Kubernetes (一体型またはクラスター) をセットアップする基本的な方法について説明します。さらに、基本的なストレージのセットアップ、Kubernetes のトラブルシューティング、systemd によるコンテナの起動およびスーパー特権コンテナの実行についても扱います。
- ※ **RHEL Atomic Host スタートガイド**: このガイドでは、RHEL Atomic Host システムを取得し、インストールし、アップグレードし、使用方法について説明します。RHEL Atomic Host は、とくにクラウドや仮想環境でコンテナを実行するために作成された RHEL の軽量バージョンです。

## 1.4. CDK INSTALLATION TOOLS について

Red Hat CDK を使用する際のワークステーションとして選択するプラットフォームが何であれ、Vagrant を使用してこれを起動し、管理することができます。開始する前に Vagrant についての以下のいくつかの点に注意してください。

- ※ Vagrant は、軽量でポータブルで、一貫した開発環境の使用を可能にするオープンソースのツールです。
- ※ Vagrant で使用するためにパッケージ化された仮想マシンは **box** と呼ばれています。
- ※ CDK は、Vagrant box の CDK ソフトウェアコンポーネントを各種の方法で設定するために複数の異なる Vagrant ファイルと共に提供されます。

ここまでで Red Hat CDK に含まれる内容を説明したので、後続の章では Red Hat CDK を取得し、これを Microsoft Windows、Mac OS X、Red Hat Enterprise Linux、または Fedora システムにインストールする方法について説明します。



## 第2章 CDK の MAC OS X へのインストール

### 2.1. CDK の MAC OS X へのインストール

Red Hat CDK を実行するために Mac OS X 開発システムを準備するには、以下のステップを実行します。

1. VirtualBox 仮想化ソフトウェアのダウンロードおよびインストール。
2. Vagrant のダウンロードおよびインストール。
3. Red Hat Container Tools および Red Hat Enterprise Linux 用の Vagrant box のダウンロード。
4. Red Hat subscription management およびその他の機能をサポートする追加の Vagrant プラグインのインストール。

#### 2.1.1. 前提条件

Red Hat CDK を Mac OS X システムで実行するには、以下が必要です。

- ※ 10.10 (Yosemite) 以降などの最新の 64 ビットバージョンの Mac OS X を実行する最小 4 GB の RAM を搭載した Intel ベースの Mac。
- ※ 仮想マシンイメージの最小 3 GB の空きディスク領域。セットアッププロセスで、各仮想マシンイメージの複数のコピーを保存できるようにする必要があります。
- ※ 1—2 GB のソフトウェアをダウンロードするための適切なインターネット接続。
- ※ サポート付きの利用可能な Red Hat Enterprise Linux Server サブスクリプションまたは Red Hat Enterprise Linux Developer Suite。セルフサポート付きの Red Hat Enterprise Linux サブスクリプションには、すべての環境での CDK に必要なすべてのソフトウェアへのアクセスがありません。

#### 2.1.2. 仮想化および Red Hat CDK コンポーネントのインストール

Vagrant は、Red Hat CDK のすべての必要なコンポーネントと共に Red Hat Enterprise Linux 仮想マシンを実行するために使用されます。仮想化は VirtualBox をインストールして提供されます。

Vagrant はコマンドラインベースのツールであることに注意してください。Vagrant とのすべての対話は端末のプロンプトに基づいてコマンドラインで行われます。Vagrant を実行するには、**Applications** → **Utilities** フォルダから Terminal.app を使用するか、または各自で選択するターミナルエミュレーターを使用する必要があります。

コマンドラインの指示では、~/**cdk** などのチルダを含むパス名を使用します。これは、現行ユーザーのホームディレクトリーの/**Users/username/cdk** の相対パスの省略形です。

##### 2.1.2.1. Mac OS X の追加のソフトウェア要件

Red Hat CDK を実行するために、一部のコマンドライン開発ツールが Mac 上で必要になります。Apple Command Line Development Tools または Apple の Xcode のいずれかをインストールする必要があります。

##### 2.1.2.2. ソフトウェアのインストールおよびシステムの設定

1. Mac OS X の VirtualBox を [virtualbox.org](https://www.virtualbox.org) からダウンロードし、インストールします。

**オプション:** VirtualBox VM イメージを保存するための場所を選択します。デフォルトでは、これらはホームディレクトリー (~/**VirtualBox VMs**) に保存されます。これらのイメージを保存することを選択する場合には、常に数ギガバイトの領域が必要になります。VirtualBox を開始する場所を変更するには、**VirtualBox** → **Preferences** → **General** メニューを選択してから、**Default Machine Folder** を必要な場所に変更します。VirtualBox のドキュメントは [virtualbox.org](https://www.virtualbox.org) web サイトにあります。

2. [vagrantup.com](https://www.vagrantup.com) から Vagrant をダウンロードし、インストールします。ダウンロードページは、[vagrantup.com/downloads.html](https://www.vagrantup.com/downloads.html) です。デフォルトでは、ソフトウェアを **/opt/vagrant** ディレクトリーにインストールします。これは各自が選択するディレクトリーに変更することができます。

Vagrant インストーラーは **/usr/bin** の **vagrant** シンボリックリンクを **/opt/vagrant/bin/vagrant** に配置するため、**PATH** 環境変数への調整は不要になります。

### 2.1.3. Red Hat CDK ソフトウェアコンポーネントのセットアップ

1. Red Hat CDK ソフトウェアコンポーネントを [Red Hat Product Downloads web サイト](https://www.redhat.com/en/products/openshift-container-engine) からダウンロードします。以下のアイテムをダウンロードする必要があります。

- ✦ Red Hat Container Tools
- ✦ RHEL 7.2 Vagrant Box for VirtualBox



#### 注記

ダウンロードページは libvirt などの他の仮想化プラットフォーム用にフォーマットされた Vagrant **.box** ファイルも提供します。VirtualBox などの使用する仮想化に一致する **.box** イメージのみをダウンロードする必要があります。

以下の手順では、これらのファイルをホームディレクトリーの **~/Downloads** に保存していることを想定しています。別のディレクトリーを使用している場合、パスを適宜調整してください。Vagrant box イメージ用に数ギガバイトの空き領域が必要です。

残りのステップはすべてターミナルアプリケーション (Terminal.app) を使用してコマンドラインにより実行されます。**Application** → **Utilities** で Terminal.app を見つけることができます。

1. ホームディレクトリーにダウンロードした ZIP ファイルを解凍します (zip ファイルの名前はリリース番号を反映して変更される可能性があります)。これにより、**~/cdk** サブディレクトリー (**/Users/username/cdk**) が作成されます。

```
$ cd
$ unzip ~/Downloads/cdk*.zip
```

この時点で、組み込まれている **README** ファイルを確認して Red Hat Container Tools に精通し、インストールガイドの公開後の変更について確認できるようにします。各 Vagrantfile に関連付けられた **README** ファイルはテキストファイルで表示できます。プラグインの **README** ファイルはマークダウン形式で作成されています。そのため、これらの

ファイルを効果的に使用するにはマークダウンリーダーが必要になる可能性があります。

2. **vagrant-adbinfo**、**vagrant-registration**、および **landrush** プラグイン (**.gem** ファイル形式のプラグインは ZIP ファイルに組み込まれます) をインストールします。

最初のプラグインのインストールには数分の時間がかかる可能性があり、Vagrant により追加の gem ファイルが随時インストールされる場合があります。

```
$ cd ~/cdk/plugins
$ ls -l *.gem
$ vagrant plugin install vagrant-registration vagrant-adbinfo
landrush
```

以下のコマンドを実行してプラグインがインストールされていることを確認します。

```
$ vagrant plugin list
```

3. Red Hat Enterprise Linux Server box を Vagrant に追加します。これは、[直前の手順](#) のいずれかでダウンロードした設定済みの仮想マシンイメージです。コンテナの開発およびテスト用にこれを使用します。

```
$ vagrant box add --name cdkv2 ~/Downloads/rhel-cdk-kubernetes-
7.2-13.x86_64.vagrant-virtualbox.box
```

Vagrant box を初期化するために Vagrantfile を使用する予定の場合、上記のステップで **--name** パラメーターを使用して box に割り当てる名前は、box を参照するために Vagrantfile が使用する名前に対応している必要があります。デフォルトでは、この名前は **cdkv2** になります。

box がインストールされていることを確認します。

```
$ vagrant box list
```

box イメージファイルはホームディレクトリーの **~/.vagrant.d** に保存されます。約 2 GB の十分な領域が必要になります。

#### 2.1.4. Red Hat CDK Vagrant Box の起動

Vagrant box および Vagrantfile を配置したら、選択した Vagrant box を起動する準備ができます。事前に定義された Vagrantfile を使用して box を初期化するか、または単に box を初期化してから独自の Vagrant ファイルを作成することができます。

既存の Vagrantfile を使用するには、使用する Vagrantfile のあるディレクトリーに移動し、そこから **vagrant up** コマンドを実行して Red Hat Enterprise Linux CDK の Vagrant box を起動します。3 つの異なる Vagrantfile が以下のユースケース用に Red Hat Container Tools ZIP ファイルで提供されます (または独自の Vagrantfile を作成することもできます)。

- ※ Docker-Eclipse 統合 (rhel-docker-eclipse) の開始

```
$ cd ~/cdk/components/rhel/rhel-docker-eclipse/
$ vagrant up
```

- ※ 単一ノード Kubernetes セットアップ (rhel-k8s-singlenode-setup) の開始

```
$ cd ~/cdk/components/rhel/rhel-k8s-singlenode-setup/
$ vagrant up
```

- ※ OpenShift Enterprise 統合 (rhel-ose) の開始

```
$ cd ~/cdk/components/rhel/rhel-ose/
$ vagrant up
```

- ※ Vagrant box および新規の Vagrantfile の初期化 (独自に作成)

```
$ mkdir ~/mycdkv2
$ cd ~/mycdkv2
$ vagrant init cdkv2
$ vagrant up
```

この時点までに問題がないと、システムを登録し、Red Hat リポジトリからのダウンロードを有効にするために Red Hat subscription management のユーザー名とパスワードの入力が求められます。以下のオプションを選択できます。

- ※ すぐに RHEL VM を登録する (推奨)
- ※ 後で RHEL VM を登録する

**注意:** RHEL サブスクリプションの取得方法、CDK に関連する登録方法、およびシステムの登録が必要な理由の詳細は「Using the CDK (CDKの使用)」を参照してください。

すぐにシステムを登録するには、[Red Hat カスタマーポータル](#) で使用するユーザー名とパスワードを入力します。

```
==> default: Registering box with vagrant-registration...
      default: Would you like to register the system now (default: yes)?
[y|n] y
      default: Subscriber username: <username>
      default: Subscriber password:
```

**vagrant status** コマンドを使用して Vagrant box が実行中であるかどうかを確認できます。Vagrant ファイルが置かれているのと同じディレクトリーにいる必要があることに注意してください。

```
$ cd ~/cdk/components/rhel/rhel-docker-eclipse/
$ vagrant status
```

マシンの状態が実行中であると表示されたら、Red Hat CDK を使用し始めることができます。以下を参照してください。

- ※ 「Using the CDK (CDKの使用)」の章は、CDK の使用に必要な各種の Vagrantfile、プラグインおよびインターフェースを理解するのに役立ちます。さらに、RHEL 仮想マシンを登録する方法などの特定のトピックについて説明しています。
- ※ コンテナ開発を開始する方法の詳細は、『Red Hat Container Development Kit Getting Started Guide』を参照してください。

## 第3章 CDK の MICROSOFT WINDOWS へのインストール

### 3.1. CDK の MICROSOFT WINDOWS へのインストール

Red Hat CDK を実行するために Microsoft Windows 開発システムを準備するには、以下のステップを実行します。

1. **VirtualBox** 仮想化ソフトウェアのダウンロードおよびインストール。
2. **Vagrant** のダウンロードおよびインストール。
3. **Red Hat Container Tools** および Red Hat Enterprise Linux 用の Vagrant box のダウンロード。
4. Red Hat subscription management およびその他の機能をサポートする追加の Vagrant プラグインのインストール。



#### 注記

本書で説明されている CDK のバージョンは、VirtualBox ソフトウェアで使用されるように作成されています。これは、Windows システムで仮想化サポートを提供する他のソフトウェアを使用している場合には機能しません。

#### 3.1.1. 前提条件

Red Hat CDK を Microsoft Windows システムで実行するには、以下が必要です。

- ※ Windows 7 64 ビット以降を実行する最小 4 GB の RAM を搭載した 64 ビットマシン。
- ※ 仮想マシンイメージの最小 3 GB の空きディスク領域。セットアッププロセス時で、各仮想マシンイメージの複数のコピーを保存できるようにする必要があります。
- ※ 1—2 GB のソフトウェアをダウンロードするための適切なインターネット接続。
- ※ サポート付きの利用可能な Red Hat Enterprise Linux Server サブスクリプションまたは Red Hat Enterprise Linux Developer Suite。セルフサポート付きの Red Hat Enterprise Linux サブスクリプションには、すべての環境での CDK に必要なすべてのソフトウェアへのアクセスがありません。(CDK でサブスクリプションを使用する方法の詳細は、「Using the CDK (CDKの使用)」の章を参照してください)。
- ※ **Vagrantfile** などの拡張子を持たないテキストファイルの編集が可能なテキストエディター。改行のみを含むファイルが出る可能性もあるため、行末の処理に柔軟性のあるテキストエディターを使用するのが望ましいでしょう。
- ※ **rsync** および **ssh** コマンドラインユーティリティーはインストールされている必要があります。推奨されるソースは [Cygwin project](#) です。代替ソースから **rsync** および **ssh** を使用することを選択する場合、一部のパスとコマンドラインの非互換性が生じる可能性があります。これについては本書では扱いません。

#### 3.1.2. 仮想化および Red Hat CDK コンポーネントのインストール

Vagrant は、Red Hat CDK のすべての必要なコンポーネントと共に Red Hat Enterprise Linux 仮想マシンを実行するために使用されます。仮想化は VirtualBox をインストールして提供されます。

Vagrant はコマンドベースのツールであることに注意してください。Vagrant とのすべての対話は、コマンドプロンプトに基づいてコマンドラインで行われます。Vagrant は Start Menu エントリやデスクトップのショートカットをインストールしません。Vagrant を実行するためには、**cmd.exe** または他のシェルを起動する必要があります。

コマンドラインの指示は、~/cdk などのパス名を使用します。これは現行ユーザーのホームディレクトリーの **C:\User\\cdk** の相対パスの省略形です。**%USERPROFILE%** 環境変数にはユーザーのホームディレクトリーへの Windows パスが含まれ、Linux の ~/ と同等です。

### 3.1.2.1. Microsoft Windows の追加のソフトウェア要件

Vagrant で使用する **ssh** および **rsync** ツールを提供するには、[cygwin.com](http://cygwin.com) から Cygwin 環境をインストールすることをお勧めします。別のソースから **ssh** および **rsync** を使用することを選択する場合には、コマンドラインの引数およびパス名に非互換性が生じる可能性があります。

Vagrant 仮想マシンにアクセスするには SSH クライアントが必要です。PuTTY などの SH を提供するグラフィカルユーティリティを使用できますが、コマンドラインから実行できる SSH クライアントを用意することが推奨されます。Vagrant は SSH 環境を管理し、単純に **vagrant ssh** コマンドを実行して Vagrant box にログインできます。これを機能させるには、**ssh** が Windows パスになければなりません。

Vagrant の同期されたフォルダーのサポートは **rsync** を使用します。VirtualBox 共有フォルダーが利用できないか、または **rsync** 同期が指定されている場合に、**rsync.exe** がパスにないと Vagrant は開始されません。

### 3.1.2.2. Windows パスの変換および rsync

**rsync** は POSIX スタイルのパスを使用し、ドライブ文字およびバックスラッシュを直接含む Windows パスを使用することはできません。Cygwin は、**/cygdrive/drive-letter/** で開始するパスを使用するため、**C:\file.exe** は **/cygdrive/c/file.exe** に変換されます。Linux ユーティリティの他のポートでは異なる規則を使用します。たとえば、MinGW/MSYS 環境は **/drive-letter** で開始するパスを使用するため、**C:\file.exe** は **/c/file.exe** に変換されます。Vagrant は使用されている環境の検出を試行してパスを適宜変換します。ただし、これが機能しない場合もあります。

Cygwin の **rsync** を Windows **cmd.exe** で使用する場合、Vagrant はユーザーが Cygwin を使用していることを確認できず、**rsync** に正しくないパスを提供します。これを回避するには、以下の行を **%USERPROFILE%\vagrant.d\Vagrantfile** にあるユーザー別の Vagrantfile に追加します。ほとんどの場合、このファイルはユーザーが作成する必要があることに注意してください。

```
# Cygwin Rsync under CMD.EXE Workaround
ENV["VAGRANT_DETECTED_OS"] = ENV["VAGRANT_DETECTED_OS"].to_s + "
cygwin"
```

この回避策の代替法として、Cygwin のターミナルウィンドウ内で Vagrant を実行することもできます。これは Linux の bash シェルのよう使用でき、ウィンドウは 80 を超えるカラム (columns) 数に設定して拡張できます。ただし、これには Windows パスではなく、Cygwin のパスが必要になります。

### 3.1.2.3. ソフトウェアのダウンロードおよびインストール

1. Microsoft Windows の VirtualBox を [virtualbox.org](http://virtualbox.org) からダウンロードし、インストールします。

オプション: VirtualBox VM イメージを保存するための場所を選択します。デフォルトでは、これらはホームディレクトリー (~/**VirtualBox VMs**) に保存されます。これらのイメージを保存することを選択する場合は、常に数ギガバイトの領域が必要になります。VirtualBox を開始する場所を変更するには、**VirtualBox** → **Preferences** → **General** メニューを選択してから、Default Machine Folder を必要な場所に変更します。VirtualBox のドキュメントは [virtualbox.org](http://virtualbox.org) web サイトにあります。



#### 注記

VirtualBox の Start Menu エントリーは **Oracle VM VirtualBox** です。

VirtualBox のドキュメントは [virtualbox.org](http://virtualbox.org) web サイトにあります。

2. [vagrantup.com](http://vagrantup.com) から Vagrant をダウンロードし、インストールします。ダウンロードページは、[vagrantup.com/downloads.html](http://vagrantup.com/downloads.html) です。デフォルトでは、ソフトウェアを **C:\HashiCorp\Vagrant\** ディレクトリーにインストールします。これは各自が選択するディレクトリーに変更することができます。



#### 注記

Windows 7 の場合は、最新バージョンの Vagrant 1.7 (例: 1.7.4) を使用する必要があります。

Vagrant インストーラーは Vagrant を Windows パスに自動的に追加します。これが実行されない場合は、Windows のコントロールパネルから **C:\HashiCorp\Vagrant\bin** をパスに追加してください。

この時点で、Vagrant インストールは再起動することを提案します。最低でも、更新されたパスを取得するためにコマンドシェルを再起動する必要があります。

3. Red Hat CDK ソフトウェアコンポーネントを [Red Hat Product Downloads web サイト](http://Red Hat Product Downloads web サイト) からダウンロードします。以下のアイテムをダウンロードする必要があります。

- ✦ Red Hat Container Tools
- ✦ RHEL 7.2 Vagrant Box for VirtualBox



#### 注記

ダウンロードページは libvirt などの他の仮想化プラットフォーム用にフォーマットされた Vagrant **.box** ファイルも提供します。VirtualBox などの使用する仮想化に一致する **.box** イメージのみをダウンロードする必要があります。

以下のステップは、これらのファイルをホームディレクトリーの **%USERPROFILE%\Downloads** に保存していることを前提とします。別のディレクトリーを使用している場合、パスを適宜調整してください。Vagrant box イメージには数ギガバイトの空き領域が必要です。

### 3.1.3. Red Hat CDK ソフトウェアコンポーネントのセットアップ

1. ダウンロードした ZIP ファイルを選択したディレクトリーに解凍します。以下のコマンドは、これをホームディレクトリーの `%USERPROFILE%\cdk` (`C:\Users\  
<username>\cdk`) に展開していることを想定しています。

この時点で、組み込まれている **README** ファイルを確認して Red Hat Container Tools に精通し、インストールガイドの公開後の変更について確認できるようにします。各 Vagrantfile に関連付けられた **README** ファイルはテキストファイルで表示できます。プラグインの **README** ファイルはマークダウン形式で作成されます。そのため、これらのファイルを効果的に使用するにはマークダウンリーダーが必要になる可能性があります。

2. Red Hat Vagrant box を使用するために追加の Vagrant プラグインをインストールします。

残りのステップはすべて Windows コマンドラインシェルの `cmd.exe` を使用して実行される必要があります。

Red Hat subscription management をサポートするために追加の Vagrant プラグインをインストールするには、**vagrant plugin install** コマンドを使用します。`.gem` ファイル形式のプラグインは ZIP ファイルに組み込まれています。

`.gem` ファイルのバージョン番号が変更されている可能性があることに注意してください。ダウンロードされたファイルに一致するファイル名を使用する必要があります。

最初のプラグインのインストールには数分の時間がかかる可能性があり、Vagrant により追加の `gem` ファイルが随時インストールされる場合があります。

```
C:> cd %USERPROFILE%\cdk\plugins
C:> dir *.gem
C:> vagrant plugin install vagrant-registration-1.1.0.cdk.gem
C:> vagrant plugin install vagrant-adbinfom-0.1.0.gem
C:> vagrant plugin install landrush-0.18.0.cdk.gem
```

以下のコマンドを実行してプラグインがインストールされていることを確認します。

```
C:> vagrant plugin list
```

CDK のインストール後にプラグインを使用する方法の詳細は、「Managing Your Red Hat Enterprise Linux Vagrant Boxes」の章を参照してください。

3. Red Hat Enterprise Linux Server box を Vagrant に追加します。

これは直前のステップでダウンロードした設定済みの仮想マシンイメージです。

```
C:> vagrant box add --name cdkv2 %USERPROFILE%\Downloads\rhel-  
cdk-kubernetes-7.2-6.x86_64.vagrant-virtualbox.box
```

Vagrant box を初期化するために Vagrantfile を使用する予定の場合、上記のステップで `--name` パラメーターを使用して box に割り当てる名前は、box を参照するために Vagrantfile が使用する名前に対応している必要があります。

box がインストールされていることを確認します。

```
C:> vagrant box list
```

box イメージファイルはホームディレクトリーの `%USERPROFILE%\vagrant.d` に保存されます。約 2 GB の十分な領域が必要になります。



## 3.2. RED HAT CDK VAGRANT BOX の起動

Vagrant box および Vagrantfile を配置したら、選択した Vagrant box を起動する準備ができます。事前に定義された Vagrantfile を使用して box を初期化するか、または単に box を初期化してから独自の Vagrant ファイルを作成することができます。

以下のステップを実行するには、Cygwin (または選択した Linux のようなユーティリティ) をインストールするために、パスに **ssh.exe** および **rsync.exe** を含める必要があります。Cygwin のインストール時に、Cygwin の **bin** ディレクトリー (通常は **C:\cygwin64\bin**) をシステムのパスに追加するオプションがあります。これを実行しない場合は、以下のコマンドを使用して現在の **cmd.exe** 用にこのディレクトリーを一時的に追加できます。この方法を選択する場合は、Vagrant で使用する **cmd.exe** シェルを開始するたびにこれを追加する必要があります。

```
C:> PATH=%PATH%;C:\cygwin64\bin;
```

また、「[Windows パスの変換と rsync](#)」のセクションで説明されているように、**%USERPROFILE%\vagrant.d\Vagrantfile** ファイルを編集しておく必要があります。

既存の Vagrantfile を使用するには、使用する Vagrantfile のあるディレクトリーに移動し、そこから **vagrant up** コマンドを実行して Red Hat Enterprise Linux CDK の Vagrant box を起動します。3 つの異なる Vagrantfile が以下のユースケース用に Red Hat Container Tools ZIP ファイルで提供されます (または独自の Vagrantfile を作成することもできます)。

- ※ Docker-Eclipse 統合 (rhel-docker-eclipse) の開始

```
$ cd %USERPROFILE%\cdk\components\rhel\rhel-ose
$ vagrant up
```

- ※ 単一ノード Kubernetes セットアップ (rhel-k8s-singlenode-setup) の開始

```
$ cd %USERPROFILE%\cdk\components\rhel\rhel-k8s-singlenode-setup
$ vagrant up
```

- ※ OpenShift Enterprise 統合 (rhel-ose) の開始

```
$ cd %USERPROFILE%\cdk\components\rhel\rhel-ose
$ vagrant up
```

- ※ Vagrant box および新規の Vagrantfile の初期化 (独自に作成)

```
$ mkdir %USERPROFILE%\mycdkv2
$ cd %USERPROFILE%\mycdkv2
$ vagrant init cdkv2
$ vagrant up
```

この時点までに問題がないと、システムを登録し、Red Hat リポジトリーからのダウンロードを有効にするために Red Hat subscription management のユーザー名とパスワードの入力が求められます。以下のオプションを選択できます。

- ※ すぐに RHEL VM を登録する (推奨)
- ※ 後で RHEL VM を登録する

**注意:** RHEL サブスクリプションの取得方法、CDK に関連する登録方法、およびシステムの登録が必要な理由の詳細は「[Using the CDK \(CDKの使用\)](#)」を参照してください。

すぐにシステムを登録するには、[Red Hat カスタマーポータル](#) で使用するユーザー名とパスワードを入力します。

```
==> default: Registering box with vagrant-registration...
      default: Would you like to register the system now (default: yes)?
[y|n] y
      default: Subscriber username: <username>
      default: Subscriber password:
```

**vagrant status** コマンドを使用して Vagrant box が実行中であるかどうかを確認できます。Vagrant ファイルが置かれているのと同じディレクトリーにいる必要があることに注意してください。たとえば、`rhel-docker-eclipse` Vagrantfile のステータスを確認するには、以下を入力します。

```
$ cd %USERPROFILE%\cdk\components\rhel\rhel-docker-eclipse
$ vagrant status
```

マシンの状態が実行中であると表示されたら、Red Hat CDK を使用し始めることができます。以下を参照してください。

- ※ 「Using the CDK (CDKの使用)」の章は、CDK の使用に必要な各種の Vagrantfile、プラグインおよびインターフェースを理解するのに役立ちます。さらに、RHEL 仮想マシンを登録する方法などの特定のトピックについて説明しています。
- ※ コンテナ開発を開始する方法の詳細は、『Red Hat Container Development Kit Getting Started Guide』を参照してください。

## 第4章 CDK の FEDORA または RED HAT ENTERPRISE LINUX へのインストール

### 4.1. CDK の FEDORA または RED HAT ENTERPRISE LINUX へのインストール

Fedora または Red Hat Enterprise Linux システムで Red Hat CDK を実行する準備をするには、以下のステップを実行します。

1. まだ実行していない場合は、ホストのオペレーティングシステム (Fedora 23 Workstation および Red Hat Enterprise Linux 7.2 Server はテスト済み) をインストールします。
2. KVM/libvirt 仮想化ソフトウェアのインストール
3. Vagrant のインストール
4. Red Hat Container Tools および Red Hat Enterprise Linux 用の Vagrant box のダウンロード。
5. Red Hat subscription management およびその他の機能をサポートする追加の Vagrant プラグインのインストール。
6. 複数の異なる Vagrantfile ファイルのいずれかを使用して Vagrant box を起動します。

#### 4.1.1. 前提条件

Fedora または Red Hat Enterprise Linux システムで Red Hat CDK を実行する準備をするには、以下が必要です。

- ※ 最小 4 GB の RAM を搭載した 64 ビットマシン
- ※ 仮想マシンイメージの最小 3 GB の空きディスク領域。セットアッププロセスで、各仮想マシンイメージの複数のコピーを保存できるようにする必要があります。
- ※ 1—2 GB のソフトウェアをダウンロードするための適切なインターネット接続。
- ※ サポート付きの利用可能な Red Hat Enterprise Linux Server サブスクリプションまたは Red Hat Enterprise Linux Developer Suite。セルフサポート付きの Red Hat Enterprise Linux サブスクリプションには、すべての環境での CDK に必要なすべてのソフトウェアへのアクセスがありません。

#### 4.1.2. 仮想化および Red Hat CDK コンポーネントのインストール

Vagrant は Red Hat CDK の必要なすべてのコンポーネントが組み込まれた Red Hat Enterprise Linux 仮想マシンを実行するために使用されます。仮想化は、ネイティブの Linux Kernel ベースの仮想マシン (KVM) ハイパーバイザーおよび libvirt、API および仮想マシンを管理するためのツールセットを使用して提供されます。

Vagrant プラグインは、Vagrant 仮想化プロバイダーの 1 つとして libvirt の使用を可能にするためにインストールされます。KVM および libvirt に馴染みがない場合は、[Fedora Virtualization Getting Started Guide](#) を参照してください。

Fedora または Red Hat Enterprise Linux 開発ホストに必要なソフトウェアをインストールし、設定を実行するには root である必要があります。Vagrant および libvirt がインストールされ、適切に設定されると、通常の root 以外のユーザーとして準備が完了できます。通常のユーザー ID を使用して Vagrant box を開始し、停止し、設定することができます。

Vagrant を実行する際には同じユーザー ID を使用するようしてください。別のユーザー ID に変更する場合は、Vagrant はプラグインやダウンロードされた box イメージなどの設定コンポーネントをユーザーのホームディレクトリに保存するため (~/.vagrant.d)、Vagrant プラグインおよび box をインストールする手順を繰り返す必要があります。

ファイル権限に関する問題の発生を避けるため、root ユーザーでログインしている場合は Vagrant を実行しないでください。

#### 4.1.2.1. ソフトウェアのインストールおよびホストシステムの設定

仮想化ソフトウェアおよび必要な設定を Red Hat Enterprise Linux または Fedora ホストシステムにインストールするには、以下のステップを root で実行しておく必要があります。

1. これをまだ実行していない場合は、Fedora Workstation または Red Hat Enterprise Linux (Server または Workstation) をハードウェア (またはハイパーバイザーとして機能するように設定されている仮想マシン) にインストールします。
2. ターミナルセッションを開き、`su -` コマンドを使用して root 権限を取得し、Vagrant をインストールして Vagrant box を実行するための開発ホストを設定します。

```
$ su -
```

3. お使いのホストシステム (Red Hat Enterprise Linux のみ) のサブスクリプションを行います。ホストが Red Hat Enterprise Linux システムの場合、(Red Hat ユーザー名およびパスワードを使用して) そのホストシステムのサブスクリプションを行い、複数のソフトウェアリポジトリを有効にします。(これらのコマンドは Fedora システムでは不要です。) 以下は例になります。

```
# subscription-manager register --auto-attach --username=user --
password=passwd
# subscription-manager repos --enable rhel-variant-rhsc1-7-rpms
# subscription-manager repos --enable rhel-7-variant-optional-
rpms
```

上記の例では、使用している Red Hat Enterprise Linux のバリエーションに応じて、**variant** を **server** または **workstation** に置き換えます。Red Hat Software Collections リポジトリが利用できない場合は、そのリポジトリを含む特定のプール ID を使ってサブスクリプションする必要があるかもしれません。

4. CentOS SCLO リポジトリ (Red Hat Enterprise Linux のみ) を有効にします。Vagrant は Red Hat Enterprise Linux 用に正式にパッケージ化されていないため、CentOS 用にパッケージ化された Software Collection を使用してインストールされる必要があります。(これは Fedora システムでは不要です。) Vagrant Software Collection を含みリポジトリを有効にするには、以下のコマンドを実行して、次のようにリポジトリの GPG チェックを無効にします。

```
# yum-config-manager --add-repo=http://mirror.centos.org/centos-
7/7/sclo/x86_64/sclo/
# echo "gpgcheck=0" >> /etc/yum.repos.d/mirror.centos.org_centos-
7_7_sclo_x86_64_sclo_.repo
```

5. システムを更新します。更新時に新規のカーネルがインストールされる場合、システムを再起動してから残りのステップに進みます。このステップは、Red Hat Enterprise Linux と Fedora では異なります。

Fedora システム:

```
# dnf -y update
```

Red Hat Enterprise Linux システム:

```
# yum -y update
```

6. 仮想化ソフトウェア KVM および libvirt のインストールし、初期化します。

Fedora システム:

```
# dnf install @Virtualization
```

Red Hat Enterprise Linux システム:

```
# yum groupinstall -y "Virtualization Host"
```

7. libvirt デーモンを起動し、ブート時に起動するように設定します。

```
# systemctl start libvirtd
# systemctl enable libvirtd
```

8. Vagrant をインストールします。**vagrant-registration** および **vagrant-libvirt** プラグインを含め、すべての Vagrant およびその他の必要なパッケージをインストールします。

Fedora システム:

```
# dnf install vagrant vagrant-libvirt vagrant-libvirt-doc
vagrant-registration rubygem-ruby-libvirt
```

Red Hat Enterprise Linux システム:

```
# yum install sclo-vagrant1 sclo-vagrant1-vagrant-libvirt \
sclo-vagrant1-vagrant-libvirt-doc sclo-vagrant1-vagrant-
registration
```

9. 通常のユーザー ID が Vagrant box を開始し、停止できるようにします。**vagrant** グループのユーザーが libvirt で VM をコントロールできるようにするため、PolicyKit ルールを追加します。必要なルールは、インストールした vagrant パッケージのいずれかに組み込まれます。以下のコマンドのいずれか (ホストが RHEL または Fedora システムであるかによって異なる) を実行して、システム上にルールを追加します。

Fedora システム:

```
# cp /usr/share/vagrant/gems/doc/vagrant-libvirt-
0.0.30/polkit/10-vagrant-libvirt.rules /etc/polkit-1/rules.d
```

Red Hat Enterprise Linux システム:

■

```
# cp /opt/rh/sclo-
vagrant1/root/usr/share/vagrant/gems/doc/vagrant-libvirt-
0.0.30/polkit/10-vagrant-libvirt.rules /etc/polkit-1/rules.d
```

以下は、参照用の新規に作成された `/etc/polkit-1/rules.d/10-vagrant-libvirt.rules` ルールの内容であり、追加する場合は手動で行ってください。

```
/*
 * Allow users in vagrant group to manage libvirt without
 authentication.
 * Copy this file to /usr/share/polkit-1/rules.d/ to activate.
 */
polkit.addRule(function(action, subject) {
  if ((action.id == "org.libvirt.unix.manage"
    || action.id == "org.libvirt.unix.monitor")
    && subject.isInGroup("vagrant")) {
    return polkit.Result.YES;
  }
});
```

10. 変更を有効にするために、libvirt および PolicyKit サービスを再起動します。

```
# systemctl restart libvirtd
# systemctl restart polkit
```

11. ユーザー ID を **vagrant** グループに追加します。

```
# echo $USER # (If you have switched to root, $USER should be
your non-root user id)
# usermod -a -G vagrant $USER
```

グループメンバーシップへの変更を有効にするために、ログアウトしてから再びログインする必要があることに注意してください。

12. ディレクトリーを共有するために NFS を使う予定の場合、そのエクスポートを一覧表示して NFS サーバーが利用可能であることを確認してください。

```
# showmount --exports
```

NFS エクスポートの一覧は、NFS を以前に使ったことがなく、ディレクトリーをエクスポートしていない限り、空になります。Vagrant は、Vagrant box を立ち上げるか、またはこれを停止する場合に必要なエクスポートを自動的に追加し、削除します。

「RPC: timed out」のエラーメッセージが出される場合、これは Vagrant がファイルの同期のために NFS を使用することを試みましたが、NFS サーバーとの接続を設定できなかったことを示します。これを解決するには、NFS サーバーと RPC サービスが実行中であることを確認します。NFS サーバーおよび RPC サービスが実行中の場合、アクセスが妨害されているかどうかを確認するために、ファイアウォールを無効にしてみます。

13. root シェルを終了します。

```
# exit
```

これにより、root で実行する必要のある一連のステップが完了します。

### 4.1.3. Red Hat CDK ソフトウェアコンポーネントのセットアップ

残りのステップは、通常の root 以外のユーザー ID で実行する必要があります。これは、**vagrant** グループに追加したのと同じユーザー ID である必要があります。

1. ターミナルセッションを開始し、**vagrant** グループのメンバーであることを確認します。

```
$ id | egrep vagrant
```

2. Vagrant Software Collection を有効にして、Vagrant を通常の root 以外のユーザーとして実行するための設定が行われていることを確認します。

Red Hat Enterprise Linux システム:

```
$ scl enable sclo-vagrant1 bash
$ vagrant global-status
```

Fedora システム:

```
$ vagrant global-status
```

global-status コマンドにより、このコンピューターにはアクティブな Vagrant 環境がないことを示すメッセージが生成されます。

3. Red Hat CDK ソフトウェアコンポーネントを [Red Hat Product Downloads web サイト](#) からダウンロードします。以下のアイテムをダウンロードする必要があります。

✳ Red Hat Container Tools



#### 注記

ダウンロードページは VirtualBox などの他の仮想化プラットフォーム用にフォーマットされた Vagrant **.box** ファイルを提供します。libvirt など、使用している仮想化に一致する **.box** イメージのみをダウンロードする必要があります。

以下の手順では、これらのファイルをホームディレクトリーの **~/Downloads** に保存していることを想定しています。別のディレクトリーを使用している場合、パスを適宜調整してください。Vagrant box イメージ用に数ギガバイトの空き領域が必要です。

4. ホームディレクトリーにダウンロードした ZIP ファイルを解凍します。これにより、**~/cdk** サブディレクトリーが作成されるはずですが。

```
$ cd
$ unzip ~/Downloads/cdk*.zip
```

この時点で、組み込まれている **README** ファイルを確認して Red Hat Container Tools に精通し、インストールガイドの公開後の変更について確認できるようにします。各 Vagrantfile に関連付けられた **README** ファイルはテキストファイルで表示できます。プラグイン **README** ファイルはマークダウン形式で作成されます。そのため、これらのファイルを効果的に使用するにはマークダウンリーダーが必要になる可能性があります。

5. **vagrant-adbinfo** および **landrush** プラグインをインストールします (プラグインは ZIP ファイルに含まれる **.gem** ファイルの形式です)。

```
$ cd ~/cdk/plugins/
$ vagrant plugin install vagrant-adbinfo landrush
```

6. 以下のコマンドを実行して、**vagrant-libvirt**、**vagrant-registration**、**landrush** および **vagrant-adbinfo** プラグインが適切にインストールされていることを確認します (**vagrant-registration** プラグインは [Installing Software](#) の別の RPM パッケージとしてすでにインストールされているため、ZIP ファイルで提供されているプラグインを使用する必要がないことに注意してください)。

```
$ vagrant plugin list
landrush (0.18.0.cdk)
vagrant-adbinfo (0.1.0)
vagrant-libvirt (0.0.30, system)
vagrant-registration (1.1.0.cdk, system)
```

7. Red Hat Enterprise Linux Server box を Vagrant に追加します。これは、直前のステップのいずれかでダウンロードした設定済みの仮想マシンイメージです。コンテナの開発およびテスト用にこれを使用します。

```
$ vagrant box add --name cdkv2 ~/Downloads/rhel-cdk-kubernetes-7.2-13.x86_64.vagrant-libvirt.box
```

Vagrant box を初期化するために Vagrantfile を使用する予定の場合、上記のステップで **--name** パラメーターを使用して box に割り当てる名前は、box を参照するために Vagrantfile が使用する名前に対応している必要があります。デフォルトでは、この名前は **cdkv2** になります。

box がインストールされていることを確認します。

```
$ vagrant box list
```

box イメージファイルはホームディレクトリーの **~/vagrant.d** に保存されます。約 2 GB の十分な領域が必要になります。

#### 4.1.4. Red Hat CDK Vagrant Box の起動

Vagrant box および Vagrantfile を配置したら、選択した Vagrant box を起動する準備ができます。事前に定義された Vagrantfile を使用して box を初期化するか、または単に box を初期化してから独自の Vagrant ファイルを作成することができます。

既存の Vagrantfile を使用するには、使用する Vagrantfile のあるディレクトリーに移動し、そこから **vagrant up** コマンドを実行して Red Hat Enterprise Linux CDK の Vagrant box を起動します。3 つの異なる Vagrantfile が以下のユースケース用に Red Hat Container Tools ZIP ファイルで提供されます (または独自の Vagrantfile を作成することもできます)。

- ※ Docker-Eclipse 統合 (rhel-docker-eclipse) の開始

```
$ cd ~/cdk/components/rhel/rhel-docker-eclipse/
$ vagrant up
```

- ※ 単一ノード Kubernetes セットアップ (rhel-k8s-singlenode-setup) の開始

```
$ cd ~/cdk/components/rhel/rhel-k8s-singlenode-setup/
$ vagrant up
```



- ※ OpenShift Enterprise 統合 (rhel-ose) の開始

```
$ cd ~/cdk/components/rhel/rhel-ose/
$ vagrant up
```

- ※ Vagrant box および新規の Vagrantfile の初期化 (独自に作成)

```
$ mkdir ~/mycdkv2
$ cd ~/mycdkv2
$ vagrant init cdkv2
$ vagrant up
```

この時点までに問題がないと、システムを登録し、Red Hat リポジトリからのRPM ソフトウェアパッケージのダウンロードを有効にするために Red Hat subscription management のユーザー名とパスワードの入力が求められます。以下のオプションを選択できます。

- ※ すぐに RHEL VM を登録する (推奨)
- ※ 後で RHEL VM を登録する

**注意:** RHEL サブスクリプションの取得方法、CDK に関連する登録方法、およびシステムの登録が必要な理由の詳細は「Using the CDK (CDKの使用)」を参照してください。

すぐにシステムを登録するには、[Red Hat カスタマーポータル](#) で使用するユーザー名とパスワードを入力します。

```
==> default: Registering box with vagrant-registration...
      default: Would you like to register the system now (default: yes)?
[y|n] y
      default: Subscriber username: <username>
      default: Subscriber password:
```

**vagrant status** コマンドを使用して Vagrant box が実行中であるかどうかを確認できます。Vagrant ファイルが置かれているのと同じディレクトリーにいる必要があることに注意してください。

```
$ cd ~/cdk/components/rhel/rhel-docker-eclipse/
$ vagrant status
```

マシンの状態が実行中であると表示されたら、Red Hat CDK を使用し始めることができます。以下を参照してください。

- ※ 「Using the CDK (CDKの使用)」の章は、CDK の使用に必要な各種の Vagrantfile、プラグインおよびインターフェースを理解するのに役立ちます。さらに、RHEL 仮想マシンを登録する方法などの特定のトピックについて説明しています。
- ※ コンテナ開発を開始する方法の詳細は、『Red Hat Container Development Kit Getting Started Guide』を参照してください。

## 第5章 RHEL CDK の使用

### 5.1. RHEL CDK の使用

Vagrant box が追加され、1 つ以上の VM が実行中である場合、CDK 環境の使用および管理を開始するためにいくつかのタスクを実行できます。本章では、Red Hat CDK の使用に関連するトピックを扱います。

#### 5.1.1. CDK Vagrantfile の選択

CDK zip ファイルには、Vagrant から異なる方法で設定された RHEL VM を起動するために使用できる複数の異なる Vagrant ファイルがあります。現時点では、`rhel-ose`、`rhel-docker-eclipse`、および `rhel-k8s-singlenode-setup` を選択できます。

##### 5.1.1.1. OpenShift を使用した開発における `rhel-ose` の使用

`rhel-ose` ディレクトリーから **vagrant up** を実行する場合、Vagrantfile はインストールした vagrant box から RHEL 仮想マシンを起動します。その box の特徴は以下の通りです。

- ※ OpenShift を使用して CDK 開発環境をセットアップします (<https://www.openshift.com/enterprise/whats-new.html> を参照)。
- ※ RHEL VM からホストシステムにサービスを公開するためにプライベートネットワークを作成します。

`rhel-ose` Vagrantfile から CDK RHEL VM を起動し、この使用を開始するには、以下を実行します。

1. `rhel-ose` ディレクトリーに切り替え、RHEL VM を起動します (Windows の場合、`%USERPROFILE%\cdk\components\rhel\rhel-ose` を使用します)。

```
$ cd ~/cdk/components/rhel/rhel-ose/  
$ vagrant up
```

2. 同じディレクトリーから、以下のように box をプロビジョニングします。

**注意:** `vagrant provision` が必ずしも必要な訳ではありませんが、これはタイムアウトの問題を解決するのに役立ち、次のステップについてのヘルプメッセージを表示するのに使用できます。

```
$ vagrant provision  
==> default: Running provisioner: shell...  
      default: Running: inline script  
...  
==> default: [INFO] Create 'test-admin' user and 'test' project  
...  
==> default:  
==> default: You can now access OpenShift console on:  
https://10.1.2.2:8443/console  
==> default:  
==> default: To use OpenShift CLI, run:  
==> default: $ vagrant ssh  
==> default: $ oc status
```

```

==> default:
==> default: To become a cluster-admin, add '--config' to oc
commands:
==> default: $ vagrant ssh
==> default: $ oc status --
config=/var/lib/origin/openshift.local.config/master/admin.kubeco
nfig
==> default:
==> default: To browse the OpenShift API documentation, follow
this link:
==> default: http://openshift3swagger-claytondev.rhcloud.com
==> default:
==> default: Then enter this URL:
==> default: https://10.1.2.2:8443/swaggerapi/oapi/v1
==> default: .

```

3. プロビジョニングの終わりに、OpenShift コンソールにアクセスする方法について説明した出力が表示され、**oc** コマンドラインを使用して OpenShift プロジェクトの使用を開始することができます。以下は、これを開始する 1 つの方法です。

- ※ Web ブラウザーが <https://10.1.2.2:8443/console> などの OpenShift コンソールの表示される URL を指すようにします。
- ※ **test-user** として、パスワード **test-user** を使ってログインします。
- ※ OpenShift Web コンソールの手引きについては、[Getting Started for Developers: Web Console](#) ガイドを参照してください。
- ※ 初回時の OpenShift アプリケーションの作成については、[OpenShift Creating New Applications](#) ガイドを参照してください。



#### 注記

OpenShift (ose) サービスを停止するには、仮想マシンをシャットダウンする必要があります。**atomic stop** コマンドは現時点では適切に機能しません。

+

rhel-ose Vagrantfile から作成された RHEL VM を使用方法の詳細は、rhel-ose ディレクトリーの README.rst ファイルを参照してください。

#### 5.1.1.2. Eclipse を使用した開発における rhel-docker-eclipse の使用

rhel-docker-eclipse ディレクトリーから **vagrant up** を実行する場合、Vagrantfile はインストールした vagrant box から RHEL 仮想マシンを起動します。この box の特徴は以下の通りです。

- ※ docker サービスを有効にし、起動します。
- ※ VM はホストから Eclipse IDE で利用できます。

rhel-docker-eclipse Vagrantfile から CDK RHEL VM を起動し、その使用を開始するには、以下を実行します。

1. `rhel-docker-eclipse` ディレクトリーに切り替え、RHEL VM を起動します (Windows の場合、`%USERPROFILE%\cdk\components\rhel\rhel\rhel-docker-eclipse` を使用します)。

```
$ cd ~/cdk/components/rhel/rhel-docker-eclipse/
$ vagrant up
```

2. `vagrant ssh` を使用してログインし、RHEL VM のシェルを開き、`systemctl` を使用して `docker` のステータスを確認します。

```
$ vagrant ssh
[vagrant@localhost ~]$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service;
   enabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/docker.service.d
            └─flannel.conf
   Active: active (running) since Mon 2016-01-25 13:28:55 EST;
   13min ago
   Docs: http://docs.docker.com
   ...
```

`vagrant ssh` コマンドを使うと、ユーザー `vagrant` (パスワード `vagrant`) は RHEL VM に自動ログインします。`systemctl` コマンドは、`docker` サービスがアクティブで、有効にされていることを示しています。

3. `docker` コマンドの使用によるコンテナの取得、実行、停止、起動、構築またはその他の方法による使用を開始できます。直接 `docker` コマンドを使用する方法の詳細は、「[Get Started with Docker Formatted Container Images \(Docker フォーマットのコンテナイメージの使用方法\)](#)」を参照してください。

### 5.1.1.3. Kubernetes を直接使用した開発における `rhel-k8s-singlenode-setup` の使用

`rhel-k8s-singlenode-setup` ディレクトリーから `vagrant up` を実行する場合、Vagrantfile は RHEL 仮想マシンを、インストールした `vagrant box` から起動します。その `box` の特徴は以下のようになります。

- ※ `docker` サービスを起動します。
- ※ RHEL VM で一体型の Kubernetes Master および Node を実行するために必要なすべてのサービスを起動します。

結果として作成される仮想マシンでは、Kubernetes pod、サービス、レプリケーションコントローラーおよびその他の機能を構築するために `kubect1` コマンドの使用を開始できます。

1. `rhel-docker-eclipse` ディレクトリーに切り替え、RHEL VM を起動します (Windows の場合、`%USERPROFILE%\cdk\components\rhel\rhel\rhel-k8s-singlenode-setup` を使用します)。

```
$ cd ~/cdk/components/rhel/rhel-k8s-singlenode-setup/
$ vagrant up
```

2. `vagrant ssh` を使用して RHEL VM にアクセスし、`kubernetes` のステータスを確認します。

■

```
$ vagrant ssh
[vagrant@localhost ~]$ kubectl cluster-info
Kubernetes master is running at http://localhost:8080
```

3. RHEL Kubernetes 対応の VM の使用を開始します。Kubernetes で実行するコンテナ化されたアプリケーションを開発する方法の詳細は、『Get Started Orchestrating Containers with Kubernetes』ガイドの「[Launching container pods with Kubernetes](#)」セクションを参照してください。

### 5.1.2. RHEL VM へのアクセス

Red Hat CDK を使用して起動する仮想マシンにアクセスするにはいくつかの方法があります。同様に、OpenShift および Eclipse などの Web UI から CDK VM にアクセスする方法は複数あります。以下のセクションでは、Vagrant から CDK にアクセスする方法について説明します。

### 5.1.3. 実行中の Running Vagrant Box との相互作用

起動している Vagrant box を使用するには、まずその box を起動したディレクトリーに切り替えます。たとえば、`cd %USERPROFILE%\cdk\components\rhel\rhel-ose` (Windows) または `cd ~/cdk/components/rhel/rhel-ose` (Fedora、RHEL、または Mac OS X) などに移動します。この場所から複数の異なる **vagrant** コマンドを実行して、box を異なる方法で使用し、管理することができます。

**vagrant** コマンドと一緒に使用できるサブコマンドを一覧表示します。

```
$ vagrant list-commands
```

SSH を使用して Red Hat Enterprise Linux CDK Vagrant Box にログインします。

```
$ vagrant ssh
```

上記により、ユーザーは vagrant ユーザーとしての RHEL 仮想マシンに自動ログインします。これが終了したら、SSH セッションを終了します (**exit** を入力)。Vagrant box を停止するには、以下を実行します。

```
$ vagrant halt
```

システム上ですべての Vagrant box の状態を確認し、box が適切に停止されていることを確認するには、以下を使用します。

```
$ vagrant global-status
```

任意の時点で作成した VM および仮想化の空きリソースを削除する必要がある場合は **vagrant destroy** コマンドを使用します。ホームディレクトリーの **.vagrant.d** にある Vagrantfile および box イメージはそのまま残り、後続の **vagrant up** コマンドを使って新規バージョンの環境を再作成できます。

```
$ vagrant destroy
```



## 注記

Vagrant がマシンごとの状態を保持する **.vagrant** サブディレクトリーを削除する前に **vagrant destroy** コマンドを使用して仮想化 (libvirt または Virtualbox) リソースを解放します。libvirt を使用するシステムに **.vagrant** サブディレクトリーがない場合は、**virt-manager** (GUI) または **virsh** (CLI) などの libvirt ツールを使用して、Vagrant が作成したリソースを手動で削除してから同じ名前の Vagrant box を立ち上げることができます。同様に、仮想化に Virtualbox を使用しているシステムでは、Virtualbox GUI を使用して Vagrant で作成したリソースを削除します。

**vagrant destroy** の後に、Vagrant box を元の正常な状態で再び立ち上げることができます。

### 5.1.4. Vagrant CDK プラグインの使用

CDK には、CDK Vagrant box の使用に対して追加機能を提供するいくつかのプラグインが同梱されます。以下は、それらのプラグインとそれらを使用する方法についての説明です。

#### 5.1.4.1. vagrant-adbinfo プラグインの使用

vagrant-adbinfo プラグインを使用すると、Vagrant CDK RHEL VM 内の docker サービスにアクセスするために必要な環境変数を表示できます。以下は、その情報を表示し、使用方法例です。

```
$ vagrant adbinfo
# Set the following environment variables to enable access to the
# docker daemon running inside of the vagrant virtual machine:

export DOCKER_HOST=tcp://192.168.121.45:2376
export DOCKER_CERT_PATH=/home/chris/cdk/components/rhel/rhel-
ose/.vagrant/machines/default/libvirt/.docker
export DOCKER_TLS_VERIFY=1
export DOCKER_MACHINE_NAME=785b349

$ eval $(vagrant adbinfo)
```

adbinfo の詳細は、CDK プラグインディレクトリーの **vagrant-adbinfo-README.md** ファイルを参照してください。

#### 5.1.4.2. landrush プラグインの使用

landrush プラグインを使用すると、ホストシステムと仮想ゲストの両方が識別できる単一 DNS サービスをセットアップできます。landrush プラグインを使用する方法の詳細は、CDK プラグインディレクトリーの **vagrant-landrush-README.md** ファイルを参照してください。

#### 5.1.4.3. vagrant-registration プラグインの使用

vagrant-registration プラグインを使用すると、RHEL VM の Red Hat サブスクリプションを Vagrant で管理できます。このプラグインは、ゲストで利用可能なタイプのサブスクリプションサービスでサブスクリプションを管理できます。ただし、現時点でこのプラグインは RHEL Subscription Manager (**subscription-manager** および関連コマンド) およびこれ以前の **rhn\_register** コマンドのみをサポートしています。

「Red Hat Enterprise Linux Vagrant Boxe の Red Hat への登録」セクションには、CDK でサブスクリプションを管理する方法の詳細情報が記載されています。vagrant-registration プラグインの詳細は、CDK プラグインディレクトリーの **vagrant-registration-README.md** ファイルを参照してください。

### 5.1.5. Red Hat Enterprise Linux Vagrant Boxe の Red Hat への登録

**vagrant up** で RHEL VM を起動する時点で RHEL VM を登録していない場合は、すぐにこれを登録する必要があります。この際、RHEL システムの登録を強くお勧めします。登録しない場合、正式の Red Hat リポジトリを使用して以下を行うことはできません。

- ※ RHEL 仮想マシンでのソフトウェアのアップグレード。
- ※ RHEL 仮想マシンへのソフトウェアパッケージの追加。
- ※ VM に構築または実行する RHEL コンテナへのソフトウェアパッケージの追加。

RHEL ベースコンテナイメージは、ホストシステムの資格情報を Docker が使用できるように設定されます。コンテナ内にパッケージをインストールしようとする、**yum** コマンドはホストの資格情報を使用して、Red Hat のそれらのパッケージにアクセスできるようにします。有効な Red Hat サブスクリプションがないと、RHEL コンテナの構築に必要な完全に機能する設定を行うことはできません。

Red Hat に CDK 仮想マシンを登録するプロセスは、Vagrant プラグインの **vagrant-registration** を使って自動化されます。デフォルトで Vagrant box が起動すると、Red Hat カスタマーポータルของผู้ー名とパスワードを入力するようにプロンプトが出されます。登録プラグインは Vagrant box を利用可能なサブスクリプションに自動的に割り当てます。

RHEL VM が CDK に登録されると、ID および有効期限のあるエンタイトルメントが VM 用に作成されます。いったん登録されると、CDK エンタイトルメントの期限が切れるまで VM を再登録する必要はありません。この期限が過ぎると、コンテナは Red Hat ソフトウェアリポジトリ (CDN) にアクセスできなくなります。

サブスクリプションがアクティブな状態で CDK VM を登録する場合、90 日間有効な新規のエンタイトルメントを取得できます。同じ VM を再登録する場合、新たに 90 日のエンタイトルメントを取得できます。これを何度も繰り返すことができます。

**vagrant halt** または **vagrant destroy** のいずれかを使用して Vagrant box を停止する場合、このプラグインは Red Hat サブスクリプションを自動的にリリースします。ホストシステムの再起動などのその他の手段で Box を停止する場合、サブスクリプションを別の Box で使用するために手動でサブスクリプションを削除する必要があります。Red Hat カスタマーポータルで subscription management を使用し、すでに使用されなくなった仮想システムを見つけ、これを削除します。

RHEL サブスクリプションがない場合、Red Hat Enterprise Linux Developer Suite の無料のエンタイトルメントを提供する計画が検討されています。それらのエンタイトルメントにより、RHEL ホストをダウンロードし、CDK で登録することができます。これらについての更新情報を確認してください。

#### 5.1.5.1. 登録プロセスの自動化 (資格情報の保存)

Vagrant box を起動するたびにプロンプトに対応しなくてもよいように Red Hat の資格情報を保存することをお勧めします。これは、単一の Vagrantfile から複数の仮想マシンを立ち上げる複雑な Vagrantfile の場合に必須になります。

資格情報を保存するには、以下の行をユーザー別の Vagrantfile に追加してください。このファイルへのパスはプラットフォームごとに異なります。

※ **Windows:** %USERPROFILE%\.vagrant.d\Vagrantfile

※ **Fedora, RHEL, or Mac OS X:** ~/.vagrant.d/Vagrantfile

設定はそのユーザー ID で開始したすべての box で利用できます。ユーザーごとの Vagrantfile は自動作成されません。

```
Vagrant.configure('2') do |config|
  config.registration.subscriber_username = '<your Red Hat username>'
  config.registration.subscriber_password = '<your Red Hat password>'
end
```

または、Red Hat の資格情報の詳細をファイルシステムに保存しないことを選択する場合は、以下の設定を使用して、資格情報を環境変数から取得することができます。ユーザー名は **\$SUB\_USERNAME** 環境変数 (Windows の SUB\_USERNAME) に、パスワードは **\$SUB\_PASSWORD** 環境変数 (Windows の SUB\_PASSWORD) に保存してから Vagrant を開始するようにしてください。

```
Vagrant.configure('2') do |config|
  config.registration.subscriber_username = ENV['SUB_USERNAME']
  config.registration.subscriber_password = ENV['SUB_PASSWORD']
end
```

さらに、これらの設定はユーザーごとの **~/.vagrant.d/Vagrantfile** の設定を上書きする特定の Vagrantfile でも使用できます。既存の Vagrantfile では、**Vagrant.configure('2') do |config|** で開始されるブロックがすでにあるため、既存のブロックにこれらの 2 つの **config.registration** 行 (上記を参照) のみを追加します。

詳細情報は、CDK ソフトウェアコンポーネントを含む ZIP ファイルの **~/cdk/plugins** ディレクトリにある **vagrant-registration-README.md** ファイルを参照してください。



## 第6章 CDK に関連する問題のトラブルシューティング

### 6.1. CDK に関連する問題のトラブルシューティング

全般的な Red Hat CDK の問題、またはとくに Vagrant 関連の問題がある場合は、本章の該当する情報を参照してください。

#### 6.1.1. 全般的な CDK の問題のトラブルシューティング

**問：** Vagrant を開始する方法は？

**答：** Vagrant はコマンドラインベースのツールです。メニューエントリーはありません。Vagrant を実行するには、マシン上でコマンドプロンプトまたはターミナルセッションを起動します。Vagrant は自動的にパスに追加されるため、**vagrant** と入力することのみが必要になります。

**問：** **vagrant up** の実行時に障害が発生したか、または SSH コマンドの失敗が報告されました。**vagrant up** を再度実行すると、VM がすでに実行中であるというメッセージが表示されます。

**答：** Vagrant が仮想マシンを起動する場合に実行されるプロビジョニングには多数のステップがあります。これらには、Red Hat 用の Vagrant Registration プラグインなどのプラグインに関連するステップや、起動時に box をプロビジョニングするために Vagrantfile に組み込まれるステップが含まれます。Vagrant は SSH を使用してこれらのコマンドを実行します。

多くの場合、プロビジョニングステップ時のエラーは致命的ではなく、Vagrant box は依然として実行中になります。**vagrant status** コマンドを使用して box の状態を確認します。**vagrant halt** コマンドを使用すると Vagrant box を停止できます。マシンが実行中の場合、**vagrant ssh** コマンドを使用してこれにログインでき、VM を調べて問題の原因を確認することができます。

box が実行されている間に、**vagrant provision** コマンドを実行してプロビジョニングステップを再度実行します。

Vagrant ファイルが置かれているのと同じディレクトリーにいる必要があることに注意してください。Vagrantfile の場所が分からなくなった場合は、**vagrant global-status** コマンドを使用して起動した box と Vagrantfile と状態が保存されているディレクトリーを一覧表示します。

**問：** Red Hat 登録が失敗したか、正しくないユーザー名とパスワードを入力しました。どのように box を手動で登録し、サブスクリプションの状態を確認できますか？

**答：** **subscription-manager** ツールを使用すると、その他の Red Hat Enterprise Linux インストールの場合と同様に、ログインや、box の登録または登録解除、サブスクリプションのステータス確認、または利用可能なソフトウェアリポジトリーの有効化を実行できます。ただし、Red Hat Enterprise Linux Vagrant box の root パスワードは配布されないため、**sudo -i** コマンドを使用して root に切り換える必要があります。**sudo** ユーティリティーは vagrant ユーザーがすべてのコマンドを実行できるようにセットアップされています。

詳細情報は、[Red Hat Subscription Management](#) を参照してください。

**問：** Vagrant box が使用する Red Hat サブスクリプションをリリースする方法は？

**答：** Vagrant Registration プラグインは、**vagrant halt** または **vagrant destroy** コマンドを使用して box をシャットダウンする場合に box を Red Hat サブスクリプションから自動的に切り離します。box が Vagrant によってシャットダウンされない場合には、これは生じません。box が依然としてセットアップされた状態の場合は、**vagrant up** コマンドを使用して box を再び立ち上げ、**vagrant halt** または **vagrant destroy** を使用してこれを正常にシャットダウンできます。これにより、box の Red Hat subscription management からの登録解除が行われます。

または、[Red Hat カスタマーポータル](#) で subscription management を使用し、すでに使用されていない仮想システムを見つけ、これを削除します。

**問：** Vagrant box 上で root として変更を行う必要がありますが、root パスワードは何ですか？

**答：** **sudo** ユーティリティーは、**vagrant** ユーザーが root としてコマンドを実行できるようにセットアップされています。**sudo** コマンドを使用して root として単一コマンドを実行するか、または **sudo -i** を使用してインタラクティブな root シェルを起動できます。

**問：** **vagrant global-status** を実行すると、**permission denied** のエラーが出されます。

**答：** **id** コマンドを実行して、通常のユーザー ID が **vagrant** グループのメンバーであるかどうかを確認します。このステップをこれまで省略していた場合は、ログアウトしてから再びログインして変更が有効になるようにします。ユーザーの **vagrant** グループへの追加は、root 以外のユーザーが **libvirt** を使用できるように root で実行した各種ステップの実行時に追加された PolicyKit ルールで機能します。

## 6.2. CDK に関する問題のトラブルシューティング

**問：** Vagrant の実行を試行する際に、**libvirt** への接続についてのエラーメッセージが出されません。root で Vagrant を実行する必要はありますか？

**答：** ソフトウェアのインストールおよび設定手順のいずれかで、PolicyKit ルールを **/etc/polkit-1/rules.d/** ディレクトリーにインストールします。これにより、root 以外のユーザーは、そのユーザーがグループ **vagrant** グループのメンバーである限り、Vagrant を実行したり、通常は root ユーザーに制限されている **libvirt** 操作を実行したりすることができるようになります。エラーメッセージは以下のようになります。

```
Error while connecting to libvirt: Error making a connection to
libvirt URI qemu:///system:
Call to virConnectOpen failed: authentication failed: no agent is
available to authenticate
```

これを解決するには、ユーザー ID が **vagrant** グループのメンバーであることを確認します。これは、**id** コマンドを実行して確認できます。上記のインストールステップを確認し、

PolicyKit ルールが `/etc/polkit-1/rules.d/` にインストールされていることを確認します。変更を有効にするには、PolicyKit および libvirtd サービスを再起動します。

```
# systemctl restart libvirtd
# systemctl restart polkit
```

**問：** `vagrant` コマンドは、`libvirt` が作成しようとしているリソースがすでに存在していることを示しているために失敗します。

**答：** 通常、Vagrant は必要に応じてリソースの作成や破棄を行うことにより、`libvirt` とのすべての対話を管理します。`vagrant destroy` を実行すると、環境に割り当てられたすべてのリソースが解放されます。ただし、Vagrant の状態ディレクトリーの `.vagrant` がクリーンアップされる前に削除されるか、または Vagrant 操作が中断される場合に、`libvirt` リソースは `vagrant destroy` で削除できない状態で残される可能性があります。これが生じる場合は、クリーンアップにはコマンドラインユーティリティの `virsh` またはグラフィカルツールの `virt-manager` を使用する必要があります。

注意: システム上のすべての `libvirt` リソースを確認するには、これらのツールを `root` で実行する必要があります。これらを `root` で実行しない場合、リソースが割り当てられていないように表示されます。

### 6.3. WINDOWS システムの CDK のトラブルシューティング

**問：** `vagrant ssh` コマンドが機能しておらず、`ssh.exe` を見つけることができません。

**答：** Windows システムで `vagrant ssh` を使用するには、パスに `ssh.exe` を組み込む必要があります。Cygwin をインストールし、`ssh.exe` がインストールされており、パスにあることを確認します。

**問：** `rsync` がパスにないため、`vagrant up` コマンドは実行を拒否しています。

**答：** `rsync` が明示的に選択されたか、またはこれが最後に選択可能なオプションであったために Windows 上の Vagrant の同期フォルダーで使用されている場合、Vagrant は `rsync.exe` がパスで利用可能でない場合には Vagrant box を起動しません。Cygwin をインストールし、`rsync.exe` がパスにあることを確認することをお勧めします。

**問：** `rsync` がパスにないため、`vagrant up` コマンドが起動を突然拒否しました。このコマンドは `rsync.exe` が利用可能になる前は機能していました。

**答：** Vagrant box が VirtualBox Guest Addition で設定されていない場合、Vagrant は共有フォルダーに `rsync` を使用することを試みます。`rsync` が見つからない場合、Vagrant はエラーを生成し、起動しません。

新規カーネルがインストールされている場合、VirtualBox Guest Additions を Vagrant box に再構築する必要がある場合があります。これは、以下で実行できます。

```
/etc/init.d/vboxdrv reload
```

---

**問：** `vagrant up` の実行時に障害が発生したか、または SSH コマンドの失敗が報告されました。 `vagrant up` を再度実行すると、VM がすでに実行中であるというメッセージが表示されます。

**答：** Vagrant が仮想マシンを起動する場合に実行されるプロビジョニングには多数のステップがあります。これらには、Red Hat 用の Vagrant Registration プラグインなどのプラグインに関連するステップや、起動時に box をプロビジョニングするために Vagrantfile に組み込まれるステップが含まれます。Vagrant は SSH を使用してこれらのコマンドを実行します。

多くの場合、プロビジョニングステップ時のエラーは致命的ではなく、Vagrant box は依然として実行中になります。 `vagrant status` コマンドを使用して box の状態を確認します。 `vagrant halt` コマンドを使用すると Vagrant box を停止できます。マシンが実行中の場合、 `vagrant ssh` コマンドを使用してこれにログインでき、VM を調べて問題の原因を確認することができます。

box が実行されている間に、 `vagrant provision` コマンドを実行してプロビジョンステップを再度実行できます。

Vagrant ファイルが置かれているのと同じディレクトリーにいる必要があることに注意してください。Vagrantfile の場所が分からなくなった場合は、 `vagrant global-status` コマンドを使用して起動したいずれかの box と Vagrantfile および状態が保存されているディレクトリーを一覧表示します。

---