



Red Hat Container Development Kit 2.0 Installation Guide

Installation Guide

Robert Krátký Chris Negus
Red Hat Developer Group Documentation
Team

Red Hat Container Development Kit2.0 Installation Guide

Installation Guide

Robert Krátký
rkratky@redhat.com

Chris Negus

Legal Notice

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.

Abstract

Guide to installing Red Hat Container Development Kit.

Table of Contents

PREFACE	3
CHAPTER 1. UNDERSTANDING RED HAT CONTAINER DEVELOPMENT KIT	4
1.1. WHERE CAN YOU RUN CDK?	4
1.2. OBTAINING AND SETTING UP CDK	4
1.3. WHAT IS INSIDE CDK	5
1.4. INSTALLATION COMPONENTS	6
CHAPTER 2. INSTALLING RED HAT CONTAINER DEVELOPMENT KIT ON MICROSOFT WINDOWS	7
2.1. PREREQUISITES	7
2.2. INSTALLING VIRTUALIZATION AND CDK COMPONENTS ON WINDOWS	7
2.3. SETTING UP CDK SOFTWARE COMPONENTS	10
2.4. STARTING CDK VAGRANT BOX IN WINDOWS	11
CHAPTER 3. INSTALLING RED HAT CONTAINER DEVELOPMENT KIT ON MAC OS X	14
3.1. PREREQUISITES	14
3.2. INSTALLING VIRTUALIZATION AND CDK COMPONENTS ON MAC OS X	14
3.3. SETTING UP CDK SOFTWARE COMPONENTS	15
3.4. STARTING THE CDK VAGRANT BOX ON MAC OS X	16
CHAPTER 4. INSTALLING RED HAT CONTAINER DEVELOPMENT KIT ON FEDORA OR RED HAT ENTERPRISE LINUX	19
4.1. PREREQUISITES	19
4.2. INSTALLING VIRTUALIZATION AND CDK COMPONENTS	19
4.3. SETTING UP CDK SOFTWARE COMPONENTS	22
4.4. STARTING THE CDK VAGRANT BOX IN RED HAT ENTERPRISE LINUX OR FEDORA	24
CHAPTER 5. USING RED HAT CONTAINER DEVELOPMENT KIT	27
5.1. USING CDK TO DEVELOP WITH OPENSIFT	27
5.2. USING CDK TO DEVELOP WITH KUBERNETES	29
5.3. USING DOCKER AND OC COMMAND-LINE TOOLS	30
5.4. USING CDK WITH DOCKER TOOLING ON ECLIPSE	30
5.5. WORKING WITH CDK VAGRANT VIRTUAL MACHINE	32
CHAPTER 6. USING VAGRANT CDK PLUGINS	34
6.1. USING THE VAGRANT-SERVICE-MANAGER PLUGIN	34
6.2. USING THE VAGRANT-REGISTRATION PLUGIN	34
6.3. USING THE VAGRANT-SSHFS PLUGIN	36
CHAPTER 7. TROUBLESHOOTING CDK PROBLEMS	38
7.1. TROUBLESHOOTING GENERAL CDK PROBLEMS	38
7.2. TROUBLESHOOTING CDK PROBLEMS WITH LIBVIRT	39
7.3. TROUBLESHOOTING CDK PROBLEMS ON WINDOWS SYSTEMS	41

PREFACE



Important

Please, note that the Red Hat Container Development Kit 2.0 product has been released as **Technical Preview**. This impacts the level of support offered by Red Hat. For additional information, please, see [Technology Preview Features Support Scope](#).

CHAPTER 1. UNDERSTANDING RED HAT CONTAINER DEVELOPMENT KIT

If you are looking to develop containerized applications for Red Hat Enterprise Linux systems, Red Hat Container Development Kit (CDK) can help you by:

- ✧ Providing a Red Hat personal container development environment you can install on your own laptop, desktop or server system.
- ✧ Including the same container development and run-time tools used to create and deploy containers for large data centers.
- ✧ Offering an easy installation method that results in virtual machines created from pre-configured Vagrant boxes and Vagrantfiles running on your local system.

This guide tells you how to install and run the Vagrant-enabled Red Hat Enterprise Linux virtual machines on your chosen system. These virtual machines are configured for Red Hat container development. The guide then describes how you can start using those virtual machines to develop Red Hat Enterprise Linux-based containers using tools such as OpenShift, Eclipse, and various command-line tools.

1.1. WHERE CAN YOU RUN CDK?

Red Hat Container Development Kit was designed to let you do your container development using the same computer on which you do your other work. CDK can be installed on the following systems:

- ✧ **Microsoft Windows:** You can use a 64-bit version of Microsoft Windows to install CDK. Windows 7 or later is required. Virtualization support must be included and turned on in the BIOS.
- ✧ **Mac OS X:** You can use an Intel-based Apple Mac to install and run CDK. The computer should have at least 4 GB of RAM and be running a recent, 64-bit version of Mac OS X, such as 10.11 (El Capitan). Virtualization support must be included and turned on in the BIOS.
- ✧ **Red Hat Enterprise Linux and Fedora:** The latest version of Red Hat Enterprise Linux 7 or Fedora is recommended for installing CDK. A 64-bit computer with at least 4 GB of RAM and virtualization support (must be turned on in the BIOS) are required.

See the CDK installation procedure for each system for more detailed hardware and software requirements.

1.2. OBTAINING AND SETTING UP CDK

Red Hat CDK software is available from the [Red Hat Customer Portal](#) to anyone with a valid Red Hat Enterprise Linux Developer Subscription or Developer Suite. Joining the [Red Hat Developers program](#) provides a path to getting the CDK. See the "Using the Red Hat CDK" chapter for details about Red Hat subscriptions and registration.

This guide provides the following instructions for installing and using the Red Hat CDK to begin developing containerized applications:

- ✧ Choosing your development system (Windows, Mac, Fedora, or Red Hat Enterprise Linux)
- ✧ Making the CDK Vagrant box and Vagrantfiles available on your system
- ✧ Starting up the Red Hat Enterprise Linux virtual machine from the Vagrant box with the selected

configuration (OpenShift or stand-alone Kubernetes)

- ✧ Accessing the Red Hat Enterprise Linux virtual machine from Vagrant and various user interfaces (OpenShift, Eclipse, or command line)
- ✧ Using special Vagrant plugins with the CDK to manage subscriptions, set up the development environment and do other activities

The rest of this book describes how to do those things.

1.3. WHAT IS INSIDE CDK

Once the Red Hat Enterprise Linux 7 virtual machine from the Red Hat CDK is running on your chosen system, you can begin exploring the contents. Some services and tools run automatically when you boot up the virtual machine while others will require some configuration. Here is a list of some of those features:

- ✧ **OpenShift:** A containerized version of OpenShift Enterprise is included in the CDK. OpenShift Enterprise provides developers with a platform for creating, provisioning, managing and scaling container-based applications for cloud environments. Once the OpenShift container is running, you can use a Web console from your browser or work from the command line with the **oc** command to develop container projects.
- ✧ **Docker:** The Docker project develops the basic container format and the **docker** command for working with containers that are included in the CDK. The CDK is configured to have the docker daemon start automatically when you boot the Red Hat Enterprise Linux virtual machine. With the **docker** command, you can build, run, start, stop, investigate and otherwise work with individual containers.
- ✧ **Kubernetes:** To orchestrate containers in what are referred to as pods, the CDK comes with all the features needed to run a Kubernetes cluster. Within the CDK, you can use CDK directly with your own container development tools or use the version of Kubernetes that is integrated with OpenShift. When started without OpenShift, Kubernetes is set up to run as an all-in-one Kubernetes master (for managing pods) and node (for running pods).

Kubernetes includes features for replicating pods (to scale up applications on the fly) and services (for interconnecting sets of containers). When you bring up the Vagrantfile for Kubernetes, the VM starts up with all required Kubernetes services running.

- ✧ **Eclipse:** Red Hat CDK includes features that let you connect to it from an instance of the Eclipse Workbench with Linux Tools/Docker Tooling plug-in. This allows CDK users to manage containers from a graphical interface on their host system.

While this document gives you the basics for installing and initially connecting to the features in your Red Hat CDK virtual machine, you should refer to other documentation for more in-depth descriptions for working with the CDK. For example:

- ✧ **CDK Getting Started Guide:** Steps you through your first experiences using CDK.
- ✧ **Container Development Guide** (in development): Provides guidance for more advanced container development. It illustrates the different ways of creating containers to run with Docker, Kubernetes, Nuclecule, Atomic and other container run-time environments.
- ✧ **Getting Started with Containers** : Describes the basics of setting up Docker and Kubernetes (all-in-one or cluster) to run containers. It also covers basic storage setup, Kubernetes troubleshooting, starting containers with systemd and running super privileged containers.

1.4. INSTALLATION COMPONENTS

1.4. INSTALLATION COMPONENTS

Regardless of which platform you choose as the workstation for using CDK, you will use Vagrant to start and manage it. There are a few things you should know about Vagrant before you get started:

- ✎ Vagrant is an open-source tool for using light-weight, portable, and consistent development environments.
- ✎ Virtual machines that are packaged for use with Vagrant are called boxes.
- ✎ CDK is delivered with two different Vagrantfiles to configure CDK software components in different ways.

While this guide contains separate chapters for installing on Microsoft Windows, Mac OS X, and Linux systems, some of the components and steps are the same for all installation types. For example, the Red Hat Developer Tools ZIP file you download includes Vagrantfiles that are used on all platforms. Likewise, Vagrant boxes you use are tied to the hypervisor (such as VirtualBox or Libvirt), rather than the operating system. The installation and use of Vagrant plugins is also the same on all operating systems.

Now that you understand the basics of what is inside CDK, choose from one of the next chapters to learn how to obtain CDK and install it on a Microsoft Windows, Mac OS X, Red Hat Enterprise Linux, or Fedora system.

CHAPTER 2. INSTALLING RED HAT CONTAINER DEVELOPMENT KIT ON MICROSOFT WINDOWS

To prepare your Microsoft Windows development system to run the CDK, the steps are:

1. Download and install the **VirtualBox** virtualization software.
2. Download and install **Vagrant**.
3. Download **Red Hat Container Tools** and Vagrant boxes for Red Hat Enterprise Linux.
4. Install additional Vagrant plugins to support Red Hat Subscription Management and other features.



Note

The version of the CDK described in this document is made to be used with VirtualBox software. It will not work if you are using other software that provides virtualization support on your Windows system. Check back later for support for other Windows virtualization environments.

2.1. PREREQUISITES

To run the CDK on a Microsoft Windows system, you will need:

- ✦ A 64-bit machine with a minimum of 4 GB of RAM running Windows 7 64-bit or later.
- ✦ A minimum of 3 GB of free disk space for virtual machine images. Note that during the setup process you will need to be able to store multiple copies of each of the virtual machine images.
- ✦ Adequate Internet connectivity to download 1—2 GB of software.
- ✦ An available Red Hat Enterprise Linux Server subscription with support or Red Hat Enterprise Linux Developer Suite. Note that Red Hat Enterprise Linux subscriptions with self-support do not have access to all of the necessary software for CDK in all environments. (See the [Using the CDK](#) chapter for more information on working with subscriptions in the CDK.)
- ✦ A text editor that allows editing text files that do not have extensions, such as **Vagrantfile**. Ideally your text editor should be flexible about line endings as you may encounter files that only have new-lines.
- ✦ The **rsync** and **ssh** command-line utilities need to be installed. The recommended source is the [Cygwin project](#). If you choose to use **rsync** and **ssh** from an alternate source, there may be some path and command line incompatibilities that are not addressed in this guide.

2.2. INSTALLING VIRTUALIZATION AND CDK COMPONENTS ON WINDOWS

Vagrant is used to run a Red Hat Enterprise Linux virtual machine with all necessary components for CDK. Virtualization is provided by installing VirtualBox.

Please note that Vagrant is strictly command-line oriented. All interaction with Vagrant is through the command line from a command prompt. Vagrant does not install any Start Menu entries or desktop shortcuts. You must launch PowerShell, **cmd.exe** or other shell to run Vagrant.

Command-line instructions use path names with a tilde, such as `~/cdk`. This is a shorthand notation for a path relative to the current user's home directory, `C:\User\<username>\cdk`. The `%USERPROFILE%` environment variable contains the Windows path to the user's home directory and is the equivalent to `~/` in Linux.

2.2.1. Additional Software Requirements for Microsoft Windows

It is recommended that you install the Cygwin environment from cygwin.com to provide the **ssh** and **rsync** tools for use by Vagrant. Please note that if you choose to use **ssh** and **rsync** from a different source, you may run into some incompatibilities in command-line arguments and path names.

An SSH client is required to access Vagrant virtual machines. While a graphical utility that provides SSH, such as PuTTY, can be used, it is preferred to have an SSH client that can be run from the command line. Vagrant manages the SSH environment, allowing you to log into your Vagrant box by simply running the **vagrant ssh** command. For this to work, **ssh** must be in the Windows path.

Synchronized folder support in Vagrant uses **rsync**. When VirtualBox shared folders are not available or **rsync** synchronization has been specified, Vagrant does not start if **rsync.exe** is not in the path.

To install Cygwin software on your Windows system, do the following:

NOTE: As an alternative to the steps below, you could install **apt-cyg** from the [apt-cyg](https://github.com/apt-cyg/apt-cyg) page on github.com. Then you could install the required packages as follows: **apt-cyg install rsync openssh**

1. Go to the [Cygwin Installation](#) page.
2. Download and install the appropriate setup executable (for example **setup-x86_64.exe**).

2.2.2. Translating Windows Paths and rsync

rsync uses POSIX style path names and cannot use Windows paths that contain drive letters and backslashes directly. Cygwin uses paths that start at **/cygdrive/drive-letter/** so, **C:\file.exe** translates to **/cygdrive/c/file.exe**. Other ports of Linux utilities use different conventions. For example, the MinGW/MSYS environment uses paths that start at **/drive-letter**, so **C:\file.exe** translates to **/c/file.exe**. Vagrant tries to detect what environment is being used and translate paths appropriately. However, there are cases where this does not work.

One way to deal with this issue is to add the following lines to the per-user Vagrantfile in the `%USERPROFILE%\ .vagrant.d\Vagrantfile` file. Note that you will most likely need to create this file.

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

If you use Cygwin's **rsync** under Windows **cmd.exe**, Vagrant is not able to tell that you are using Cygwin and supplies the wrong path to **rsync**. The workaround is to set the **VAGRANT_DETECTED_OS** variable to **cygwin**. For example:

```
$ setx VAGRANT_DETECTED_OS cygwin
```

To permanently set this variable for a bash shell, add the following line to the **.bashrc** file in the user's home directory (**~/.bashrc**):

```
export VAGRANT_DETECTED_OS=cygwin
```

An alternative is to run Vagrant inside of Cygwin's Terminal Window. This will provide a Linux-like experience with the Bash shell and a window that can be sized beyond 80 columns. In this way, you can use both Linux and Windows paths. However, to use Windows paths, you must surround those paths with single quotes. For example, **c:\User\joe** would be interpreted by the shell as **c:Userjoe** without single quotes. Surrounding **'c:\User\joe'** with single quotes, causes it to be interpreted correctly.

2.2.3. Software Download and Installation to add to Windows

1. Download and Install VirtualBox for Microsoft Windows from [virtualbox.org](https://www.virtualbox.org).

Optional: Choose a location for storing VirtualBox VM images. By default, these are stored in your home directory (**~/VirtualBox VMs**). You will need several gigabytes of space wherever you choose to store these images. To change the location start VirtualBox, Use the **VirtualBox** → **Preferences** → **General** menu, then change Default Machine Folder to the desired location. Documentation for VirtualBox can be found on the [virtualbox.org](https://www.virtualbox.org) website.



Note

The Start Menu entry for VirtualBox is **Oracle VM VirtualBox**.

Documentation for VirtualBox can be found on the [virtualbox.org](https://www.virtualbox.org) web site.

2. Download and install Vagrant from [vagrantup.com](https://www.vagrantup.com). The download page is: [Vagrant Versions](https://www.vagrantup.com). Select the folder with the latest 1.7.x release (such as 1.7.4) and download the vagrant file in .msi format (for example, **vagrant_1.7.4.msi**).



Note

For Windows, you should use the latest version of Vagrant 1.7 (for example, 1.7.4). The latest Vagrant 1.8 version does not work in this environment. Vagrant 1.7 may be required for other versions of Windows as well.

The Vagrant installer should automatically add Vagrant to the Windows path. If it does not, add **C:\HashiCorp\Vagrant\bin** to the path through the Windows Control Panel.

At this point the Vagrant installer recommends a reboot. At a minimum, you must restart your command shell to get the updated path.

3. Download the CDK software components from the [Red Hat Product Downloads web site](#). You must log in to get access to this page. If you are on the right page, you should see "Product Variant: Red Hat Container Development Kit". You need to download the following items:

- ✧ Red Hat Developer Tools
- ✧ RHEL 7.2 Vagrant box for VirtualBox

**Note**

The page also offers Vagrant box downloads formatted for other virtualization platforms, such as libvirt. You only need to download the box image that matches the virtualization you are using — VirtualBox.

The following steps assume you have saved these files in your home directory in **%USERPROFILE%\Downloads**. If you used a different directory, adjust the paths accordingly. You need several gigabytes of free space for the Vagrant box images.

2.3. SETTING UP CDK SOFTWARE COMPONENTS

1. Unzip the ZIP file you downloaded in a directory of your choice. The following commands assume you have unpacked it in your home directory, **%USERPROFILE%\cdk (C:\Users\\cdk)**.

At this point, review the included **README** files to familiarize yourself with Red Hat Container Tools and to be notified if there have been any changes since this installation guide was published. The **README** file associated with each Vagrantfile can be viewed as a plain text file. The plugin **README** files are formatted in Markdown. So you might want to use a Markdown reader for a better experience with those files.

2. Install additional Vagrant plugins for using Red Hat Vagrant boxes.

All of the remaining steps can be performed using the Windows PowerShell, command-line shell, **cmd.exe** or other command line shell.

To install additional Vagrant plugins to support several features, use the **vagrant plugin install** command. The plugins in the form of **.gem** files are included in the ZIP file.

Note that the version number of the **.gem** files may have changed. You should use the file name that matches your downloaded files.

The installation of the first plugin may take several minutes, and Vagrant may install some additional gem files as needed.

```
C:> cd %USERPROFILE%\cdk\plugins
C:> dir *.gem
C:> vagrant plugin install vagrant-registration-1.2.1.gem
C:> vagrant plugin install vagrant-service-manager.1.0.1.gem
C:> vagrant plugin install vagrant-sshfs-1.1.0.gem
```

Verify the plugins are installed by running the following command:

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



For information on using the plugins after the CDK is installed, see the "Managing Your Red Hat Enterprise Linux Vagrant Boxes" chapter.

3. Add the Red Hat Enterprise Linux Server box to Vagrant.

This is the configured virtual machine image that you downloaded in the previous step.

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

Note that if you plan on using a Vagrantfile to initialize the Vagrant box, the name you assign to the box using the **--name** parameter in the above step must correspond to the name used by the Vagrantfile to refer to the box. (The default is cdkv2.)

Verify that the box is installed:

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

The box image files will be stored in your home directory under **%USERPROFILE%\ .vagrant .d**. You will need adequate space there, approximately 2 GB.

2.4. STARTING CDK VAGRANT BOX IN WINDOWS

With the Vagrant box and Vagrantfiles in place, you are ready to bring up the selected Vagrant box. Your choices of Vagrantfiles are as follows:

- ✦ **OpenShift (rhel-ose)**: The OpenShift (rhel-ose) Vagrantfile launches a Red Hat Enterprise Linux server virtual machine (VM) with OpenShift Enterprise running on it. With OpenShift running, you can use either the Web Console from the browser on your desktop or **docker, oc**, and related commands by logging into the VM. With some additional setup, you can add other tools to access feature in the VM from your host (see [Using the CDK](#) for information on how to to access the VM from Eclipse or command line tools from your desktop).
- ✦ **Kubernetes(rhel-k8s-singlenode-setup)**: Use the Kubernetes (rhel-k8s-singlenode-setup) Vagrantfile to start a more generic CDK VM. Because OpenShift is not running, you can configure a more basic Kubernetes configuration or use docker directly.

NOTE: Instead of using one of the provided Vagrantfiles, you can copy one of the existing Vagrantfiles and modify it for your own purposes. For example, you might want to use a different IP address if it conflicts with an address on your local network.

Follow these instruction to start up a VM from the selected VM, from the proper Vagrantfile provided in the Red Hat Container Tools ZIP file.



Note

Before you start your CDK Vagrant box, consider reviewing the Vagrantfile to see if there are any settings you want to change. In particular, you might want to configure SSHFS to share a directory between your CDK virtual machine and your local system. See the description of the vagrant-sshfs plugin in the "Using the Red Hat CDK" chapter for information on how to configure this.

In order to run the following steps, you need to have **ssh.exe** and **rsync.exe** in your path from installing Cygwin (or your choice of Linux-like utilities). You can temporarily add it for the current **cmd.exe** using the following command. If you go this route, you will need to do this every time you start a **cmd.exe** shell for use with Vagrant.

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

To add this location to your PATH permanently (for example, on a Windows 10 system), right click and select start → system → advanced system settings → Environment variables. Then change the PATH variable as shown.

You also need to have edited the **%USERPROFILE%\vagrant.d\Vagrantfile** file as discussed in the section [Translating Windows Paths and rsync](#).

To use an existing Vagrantfile, go to the directory with the Vagrantfile you wish to use, and start the Red Hat Enterprise Linux CDK Vagrant box by running the **vagrant up** command from there. Two different Vagrantfiles are provided in the Red Hat Container Tools ZIP file for the following use cases.



Note

Before you start your CDK Vagrant box, consider reviewing the Vagrantfile to see if there are any settings you want to change. In particular, you might want to configure SSHFS to share a directory between your CDK virtual machine and your local system. See the description of the `vagrant-sshfs` plugin in the "Using the Red Hat CDK" chapter for information on how to configure this.

- ✦ Start the OpenShift Enterprise integration (rhel-ose)

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

- ✦ Start the Single-node Kubernetes setup (rhel-k8s-singlenode-setup)

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

At this point, if all is going well, you are asked for a username and password for Red Hat Subscription Management in order to register the system and enable downloading from Red Hat repositories. You have the choice of:

- ✦ Registering your Red Hat Enterprise Linux VM now (recommended) or
- ✦ Registering your Red Hat Enterprise Linux VM later

NOTE: See the [Using the CDK](#) chapter for information on obtaining Red Hat Enterprise Linux subscriptions, how registration works with the CDK, and details on why you need to register your system.

To register your system now, enter the username and password you use with the [Red Hat Customer Portal](#).

```
==> 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:
```

You can check whether your Vagrant box is running using the **vagrant status** command. Note that you must be in the same directory where your Vagrant file is located. For example, to check the status of the rhel-ose VM, type the following:

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

If the machine state shows as running, you are ready to start using your CDK. Refer to the following:

- ✎ The [Using the CDK](#) chapter will help you understand the different Vagrantfiles, plugins and interfaces needed to work with your CDK. It also discusses special topics, such as other ways to register your Red Hat Enterprise Linux virtual machines.
- ✎ The [Getting Started with Container Development Kit](#) guide provides information on getting started with container development.

CHAPTER 3. INSTALLING RED HAT CONTAINER DEVELOPMENT KIT ON MAC OS X

To prepare your Mac OS X development system for running the CDK, the steps are:

1. Download and install the VirtualBox virtualization software.
2. Download and install Vagrant.
3. Download Red Hat Container Tools and the Vagrant box for Red Hat Enterprise Linux.
4. Install additional Vagrant plugins to support Red Hat Subscription Management and other features.

3.1. PREREQUISITES

To run the CDK on a Mac OS X system, you need:

- ✶ An Intel-based Mac with a minimum of 4 GB of RAM running a recent, 64-bit version of Mac OS X, such as 10.11 (El Capitan) or later.
- ✶ A minimum of 3 GB of free disk space for virtual machine images. Note that during the setup process, you will need to be able to store multiple copies of each of the virtual machine images.
- ✶ Adequate Internet connectivity to download 1—2 GB of software.
- ✶ An available Red Hat Enterprise Linux Server subscription with support or Red Hat Enterprise Linux Developer Suite. Note that Red Hat Enterprise Linux subscriptions with self-support do not have access to all of the necessary software for CDK in all environments.

3.2. INSTALLING VIRTUALIZATION AND CDK COMPONENTS ON MAC OS X

Vagrant is used to run a Red Hat Enterprise Linux virtual machine with all necessary components for CDK. Virtualization is provided by installing VirtualBox.

Please note that Vagrant is strictly command-line oriented. All interaction with Vagrant is through the command line from a Terminal prompt. You should use Terminal.app from the **Applications** → **Utilities** folder or a terminal emulator of your choice for running Vagrant.

Command-line instructions use path names with a tilde, such as `~/cdk`. This is a shorthand notation for a path relative to the current user's home directory, `/Users/username/cdk`.

3.2.1. Additional Software Requirements for Mac OS X

In order to run CDK, some command-line development tools are needed on your Mac. You should install either Apple Command Line Development Tools or Apple's Xcode.

3.2.2. Installing Software and Configuring the System

1. Download and install VirtualBox for Mac OS X from [virtualbox.org](https://www.virtualbox.org).

Optional: Choose a location for storing VirtualBox VM images. By default, these are stored in your home directory (`~/VirtualBox VMs`). You will need several gigabytes of space wherever you choose to store these images. To change the location start VirtualBox, Use the **VirtualBox** → **Preferences** → **General** menu, then change **Default Machine Folder** to the desired location. Documentation for VirtualBox can be found on the virtualbox.org website.

2. Download and install Vagrant from vagrantup.com. The download page is: [Vagrant Versions](#). Select the folder with the latest 1.7.x release (such as 1.7.4) and download the vagrant file in .dmg format (for example, **vagrant_1.7.4.dmg**).

The default is to install the software in the `/opt/vagrant` directory. You may change this to whatever you prefer. The Vagrant installer places a **vagrant** symbolic link in `/usr/bin` to `/opt/vagrant/bin/vagrant`, so no adjustments to the **PATH** environment variable should be necessary.

3.3. SETTING UP CDK SOFTWARE COMPONENTS

1. Download the CDK software components from the [Red Hat Product Downloads web site](#). You must log in to get access to this page. If you are on the right page, you should see "Product Variant: Red Hat Container Development Kit". You need to download the following items:

- ✱ Red Hat Developer Tools
- ✱ RHEL 7.2 Vagrant box for VirtualBox



Note

The page also offers Vagrant box downloads formatted for other virtualization platforms, such as libvirt. You only need to download the box image that matches the virtualization you are using — VirtualBox.

The following steps assume you have saved these files in your home directory in `~/Downloads`. If you used a different directory, adjust the paths accordingly. You need several gigabytes of free space for the Vagrant box images.

All of the remaining steps need to be performed using the command-line using the Terminal Application (Terminal.app). You can find Terminal.app in the **Application** → **Utilities** Folder.

2. Unzip the ZIP file you downloaded in your home directory (the name of the zip file may change to reflect a release number). This should create the `~/cdk` subdirectory (`/Users/username/cdk`):

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

At this point, please view the included **README** files to familiarize yourself with the Red Hat Container Tools and to be notified if there have been any changes since this installation guide was published. The **README** file associated with each Vagrantfile can be viewed as a plain text file. The plugin **README** files are formatted in Markdown. So you might want to use a Markdown reader for a better experience with those files.

3. Install the **vagrant-registration**, **vagrant-service-manager**, and **vagrant-sshfs** plugins (the plugins in the form of **.gem** files are included in the ZIP file).

The installation of the first plugin may take several minutes, and Vagrant may install some additional gem files as needed.

```
$ cd ~/cdk/plugins
$ ls -l *.gem
$ vagrant plugin install ./vagrant-registration-1.2.1.gem \
    ./vagrant-service-manager-1.0.1.gem \
    ./vagrant-sshfs-1.1.0.gem
```

Verify the plugins are installed by running the following command:

```
$ vagrant plugin list
```

4. Add the Red Hat Enterprise Linux Server box to Vagrant. This is the configured virtual machine image that you downloaded in one of the [previous steps](#). You will be using this for container development and testing.

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

Note that if you plan on using a Vagrantfile to initialize the Vagrant box, the name you assign to the box using the **--name** parameter in the above step must correspond to the name used by the Vagrantfile to refer to the box. By default, this is **cdkv2**.

Verify that the box is installed:

```
$ vagrant box list
```

The box image file will be stored in your home directory under **~/ .vagrant .d**. You need adequate space there, approximately 2 GB.

3.4. STARTING THE CDK VAGRANT BOX ON MAC OS X

With the Vagrant box and Vagrantfiles in place, you are ready to bring up the selected Vagrant box. Your choices of Vagrantfiles are as follows:

- **OpenShift (rhel-ose)**: The OpenShift (rhel-ose) Vagrantfile launches a Red Hat Enterprise Linux server virtual machine (VM) with OpenShift Enterprise running on it. With OpenShift running, you can use either the Web Console from the browser on your desktop or **docker**, **oc**, and related commands by logging into the VM. With some additional setup, you can add other tools to access features in the VM from your host (see [Using the CDK](#) for information on how to access the VM from Eclipse or command line tools from your desktop).
- **Kubernetes(rhel-k8s-singlenode-setup)**: Use the Kubernetes (rhel-k8s-singlenode-setup) Vagrantfile to start a more generic CDK VM. Because OpenShift is not running, you can configure a more basic Kubernetes configuration or use docker directly.

NOTE: Instead of using one of the provided Vagrantfiles, you can copy one of the existing Vagrantfiles and modify it for your own purposes. For example, you might want to use a different IP address if it conflicts with an address on your local network.

Follow these instruction to start up a VM from the selected VM, from the proper Vagrantfile provided in the Red Hat Container Tools ZIP file.



Note

Before you start your CDK Vagrant box, consider reviewing the Vagrantfile to see if there are any settings you want to change. In particular, you might want to configure SSHFS to share a directory between your CDK virtual machine and your local system. See the description of the `vagrant-sshfs` plugin in the "Using the Red Hat CDK" chapter for information on how to configure this.

- ✧ Start the OpenShift Enterprise integration (rhel-ose)

```
$ cd ~/cdk/components/rhel/rhel-ose/
$ vagrant up
$ vagrant provision      # If OpenShift fails, repeat this command
                        until it starts
```

- ✧ Start the Single-node Kubernetes setup (rhel-k8s-singlenode-setup)

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

At this point, if all is going well, you are asked for a username and password for Red Hat Subscription Management in order to register the system and enable downloading from Red Hat repositories. You have the choice of:

- ✧ Registering your Red Hat Enterprise Linux VM now (recommended) or
- ✧ Registering your Red Hat Enterprise Linux VM later

NOTE: See the "Using the CDK" chapter for information on obtaining Red Hat Enterprise Linux subscriptions, how registration works with the CDK, and details on why you need to register your system.

To register your system now, enter the username and password you use with the [Red Hat Customer Portal](#).

```
==> 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:
```

You can check whether your Vagrant box is running using the **vagrant status** command. Note that you must be in the same directory where your Vagrant file is located. For example:

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

If the machine state shows as running, you are ready to start using your CDK. Refer to the following:

- ✦ The [Using the CDK](#) chapter will help you understand the different Vagrantfiles, plugins and interfaces needed to work with your CDK. It also discusses special topics, such as other ways to register your Red Hat Enterprise Linux virtual machines.
- ✦ The [Getting Started with Container Development Kit](#) guide provides information on getting started with container development.

CHAPTER 4. INSTALLING RED HAT CONTAINER DEVELOPMENT KIT ON FEDORA OR RED HAT ENTERPRISE LINUX

To prepare to run the CDK on a Fedora or Red Hat Enterprise Linux system, an overview of the steps are:

1. If you have not already done so, install the host operating system (in this case, we tested both Fedora 23 Workstation and Red Hat Enterprise Linux 7.2 Server)
2. Install the KVM/libvirt virtualization software
3. Install Vagrant
4. Download Red Hat Container Tools and the Vagrant box for Red Hat Enterprise Linux.
5. Install additional Vagrant plugins to support Red Hat Subscription Management and other features.
6. Bring up the Vagrant box using a Vagrantfile.

4.1. PREREQUISITES

To run the CDK on a Fedora or Red Hat Enterprise Linux system, you need:

- ✳ A 64-bit machine with a minimum of 4 GB of RAM. At least 2 GB should be reserved for the CDK, with 3-4GB reserved for the CDK being more reasonable if you plan to run multiple virtual machines. The Kubernetes Vagrantfiles defines 1 GB of disk space while the OSE Vagrantfile asks for up to 3 GB of disk space.
- ✳ A minimum of 3 GB of free disk space for virtual machine images. Note that during the setup process, you will need to be able to store multiple copies of each of the virtual machine images.
- ✳ Adequate Internet connectivity to download 1—2 GB of software.
- ✳ An available Red Hat Enterprise Linux Server subscription with support for Red Hat Enterprise Linux Developer Suite. Note that Red Hat Enterprise Linux subscriptions with self-support do not have access to all of the necessary software for CDK in all environments.

4.2. INSTALLING VIRTUALIZATION AND CDK COMPONENTS

Vagrant is used to run a Red Hat Enterprise Linux virtual machine with all necessary components for CDK included in it. Virtualization will be provided by using the native Linux Kernel-based Virtual machine (KVM) hypervisor and libvirt, an API and a set of tools for managing virtual machines.

A Vagrant plugin is installed to enable using libvirt as one of Vagrant virtualization providers. If you are not familiar with KVM and libvirt, see the [Fedora Virtualization Getting Started Guide](#).

You need root privilege to install the necessary software and perform configuration on your Fedora or Red Hat Enterprise Linux development host. Once Vagrant and libvirt are installed and correctly configured, you will complete the preparation as a regular, non-root user. You will be able to start, stop, and configure Vagrant boxes using your regular user account.

Take care to use the same user ID for running Vagrant. Since Vagrant stores configuration components, such as plugins and downloaded box images, in the user's home directory (`~/` **.vagrant.d**), if you change to a different user ID, you will need to repeat the steps that install Vagrant plugins and boxes.

Do not run Vagrant when you are logged in as the root user, to avoid creating problems with file permissions.

4.2.1. Installing Software and Configuring the Host System

The following steps need to be completed as root to install the virtualization software and the necessary configuration on your Red Hat Enterprise Linux or Fedora host system.

1. If you have not already done so, install Fedora Workstation or Red Hat Enterprise Linux (Server or Workstation) directly on hardware (or on a virtual machine that is set to act as a hypervisor).
2. Open a terminal session and use the **su -** command to gain root privileges to install Vagrant and configure the development host for running Vagrant boxes.

```
$ su -
```

3. Subscribe your host system (Red Hat Enterprise Linux only). If your host is a Red Hat Enterprise Linux system, you must subscribe that host system (using your Red Hat username and password) and enable several software repositories. (These commands are not required for Fedora systems.) For example:

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

In the above examples, replace **variant** with **server** or **workstation**, depending on which variant of Red Hat Enterprise Linux you are using. If the Red Hat Software Collections repository is not available, you might need to subscribe with a specific pool ID that includes that repository.

4. Enable the CentOS SCLO repository (Red Hat Enterprise Linux only). Because Vagrant is not officially packaged for Red Hat Enterprise Linux, it needs to be installed using a Software Collection packaged for CentOS. (This is not required for Fedora systems.) To enable the repository containing the Vagrant Software Collection, run the following command:

```
# 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. Update your system. If a new kernel is installed during the update, reboot your system before proceeding with the remaining steps. This step is different on Red Hat Enterprise Linux and Fedora.

On a Fedora system:


```
# dnf -y update
```

On a Red Hat Enterprise Linux system:

```
# yum -y update
```

6. Install and initialize virtualization software: KVM and libvirt.

On a Fedora system:

```
# dnf install @Virtualization
```

On a Red Hat Enterprise Linux system:

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

7. Launch the libvirt daemon and configure it to start at boot.

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

8. Install Vagrant and other required packages, including the **vagrant-registration** and **vagrant-libvirt** plugins:

On a Fedora system:

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

On a Red Hat Enterprise Linux system:

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

9. Allow your regular user ID to start and stop Vagrant boxes. A PolicyKit rule will be added that allows users in the **vagrant** group to control VMs through libvirt. The necessary rule is included in one of the vagrant packages you just installed. Run one of the following commands (depending on whether your host is a Red Hat Enterprise Linux or Fedora system) to add the rule on your system:

On a Fedora system:

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

On a Red Hat Enterprise Linux system (note that your version of vagrant-libvirt may be different):

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

The following is the contents of the newly created `/etc/polkit-1/rules.d/10-vagrant-libvirt.rules` rule for reference, or if you prefer to add it by hand.

```
/*
 * 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. Restart the libvirt and PolicyKit services for the changes to take effect:

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

11. Check your user name, then become root to add your user name to the **vagrant** group.

```
$ echo $USER
$ su -
Password: *****
# usermod -a -G vagrant <username>
```

Note that you need to log out and back in for the change to your group membership to take affect.

12. If you intend to use NFS for directory sharing, check that the NFS server is available by checking that the `nfs-server` and `rpcbind` services are running, and then listing NFS exports:

```
# systemctl status nfs-server
# systemctl status rpcbind
# showmount --exports
```

If you get an **RPC: timed out** error message, it indicates that Vagrant attempted to use NFS for file synchronization, but it could not establish a connection to the NFS server. If the NFS server and RPC service are running, but the NFS service is not available, try to disable the firewall to see if it is preventing access.

The list of NFS exports is empty unless you have previously used NFS and exported some directories. Vagrant automatically adds and removes the necessary exports when you bring a Vagrant box up or halt it.

13. Exit from the root shell:

```
# exit
```

That completes the list of steps that need to be performed as root.

4.3. SETTING UP CDK SOFTWARE COMPONENTS

The remaining steps should be performed under your regular non-root user ID. This should be the same user ID you added to the **vagrant** group.

1. Start a Terminal session and verify you are a member of the **vagrant** group (you may need to log in again to have the new group appear).

```
$ grep vagrant /etc/group
vagrant:x:978:joe
$ id
uid=1001(joe) gid=1001(joe) groups=1001(joe),978(vagrant)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

2. Enable the Vagrant Software Collection and verify that things are set up to run Vagrant as a regular, non-root user:

On a Red Hat Enterprise Linux system:

```
$ scl enable scl0-vagrant1 bash
$ vagrant global-status
```

On a Fedora system (Vagrant Software Collection is not required):

```
$ vagrant global-status
```

The `global-status` command should produce a message that says there are no active Vagrant environments on this computer.

3. Download the CDK software components from the [Red Hat Product Downloads web site](#). You must log in to get access to this page. If you are on the right page, you should see "Product Variant: Red Hat Container Development Kit". You need to download the following items:

- ✶ Red Hat Container Tools (**cdk-2.0*.zip** or later)
- ✶ Red Hat Enterprise Linux 7.2 Vagrant box for libvirt (**rhel-cdk-kubernetes-7.2*.x86_64.vagrant-libvirt.box** or later)



Note

The download page also offers Vagrant **.box** files formatted for other virtualization platforms, such as VirtualBox. You only need to download the **.box** image that matches the virtualization you are using — which is libvirt for Red Hat Enterprise Linux or Fedora.

The following steps assume you have saved these files in your home directory in **~/Downloads**. If you used a different directory, adjust the paths accordingly. You need several gigabytes of free space for the Vagrant box images.

4. Unzip the ZIP file you downloaded in your home directory. This should create the **~/cdk** subdirectory:

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

At this point, review the included **README** files to familiarize yourself with Red Hat Container Tools and to find out if there have been any changes since this installation guide was published. The **README** file associated with the `rhel-ose` Vagrantfile can be viewed as a plain text file. The plugin **README** files are formatted in Markdown. So you might want to use a Markdown reader for a better experience with those files.

5. Install the plugins contained in the `plugins` directory. Each plugin is in the form of `.gem` files that were unpacked from the ZIP file (note that your versions may be different):

```
$ cd ~/cdk/plugins/
$ vagrant plugin install \
  ./vagrant-registration-1.2.1.gem \
  ./vagrant-service-manager-1.0.1.gem \
  ./vagrant-sshfs-1.1.0.gem
```

6. Verify the **vagrant-libvirt**, **vagrant-registration**, **vagrant-service-manager**, and **vagrant-sshfs** plugins are properly installed by running the following command (note that the **vagrant-registration** plugin has already been installed as a separate RPM package in [Installing Software](#), but you can install plugin supplied in the ZIP file so you get the latest version):

```
$ vagrant plugin list
vagrant-libvirt (0.0.32, system)
vagrant-registration (1.2.1)
  - Version Constraint: 1.2.1
vagrant-service-manager (1.0.1)
  - Version Constraint: 1.0.1
vagrant-sshfs (1.1.0)
  - Version Constraint: 1.1.0
```

7. Add the Red Hat Enterprise Linux Server box to Vagrant. This is the configured virtual machine image that you downloaded in one of the previous steps. (Adjust the command below if you have a newer version of the Vagrant box.) You will be using this for container development and testing.

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

Note that the name you assign to the box using the `--name` parameter in the above step must correspond to the name used by the Vagrantfile to refer to the box. By default, this is **cdkv2** for the Vagrantfiles provided with the CDK.

Verify that the box is installed:

```
$ vagrant box list
```

The box image file will be stored in your home directory under `~/.vagrant.d`. You need adequate space there, approximately 3 GB.

4.4. STARTING THE CDK VAGRANT BOX IN RED HAT ENTERPRISE LINUX OR FEDORA

With the Vagrant box and Vagrantfiles in place, you are ready to bring up the selected Vagrant box. Your choices of Vagrantfiles are as follows:

- ✧ **OpenShift (rhel-ose):** The OpenShift (rhel-ose) Vagrantfile launches a Red Hat Enterprise Linux server virtual machine (VM) with OpenShift Enterprise running on it. With OpenShift running, you can use either the Web Console from the browser on your desktop or **docker**, **oc**, and related commands by logging into the VM. With some additional setup, you can add other tools to access feature in the VM from your host (see [Using the CDK](#) for information on how to access the VM from Eclipse or command line tools from your desktop).
- ✧ **Kubernetes(rhel-k8s-singlenode-setup):** Use the Kubernetes (rhel-k8s-singlenode-setup) Vagrantfile to start a more generic CDK VM. Because OpenShift is not running, you can configure a more basic Kubernetes configuration or use docker directly.

NOTE: Instead of using one of the provided Vagrantfiles, you can copy one of the existing Vagrantfiles and modify it for your own purposes. For example, you might want to use a different IP address if it conflicts with an address on your local network.

Follow these instruction to start up a VM from the selected VM, from the proper Vagrantfile provided in the Red Hat Container Tools ZIP file.



Note

Before you start your CDK Vagrant box, consider reviewing the Vagrantfile to see if there are any settings you want to change. In particular, you might want to configure SSHFS to share a directory between your CDK virtual machine and your local system. See the description of the vagrant-sshfs plugin in the "Using the Red Hat CDK" chapter for information on how to configure this.

- ✧ Start the OpenShift Enterprise integration (rhel-ose) as follows:

```
$ cd ~/cdk/components/rhel/rhel-ose/
$ vagrant up
$ vagrant provision      # If OpenShift fails, repeat this command
                        until it starts
```

- ✧ Start the Single-node Kubernetes setup (rhel-k8s-singlenode-setup) as follows:

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

At this point, if all is going well, you are asked for a username and password for Red Hat Subscription management in order to register the system and enable downloading of RPM software packages from Red Hat repositories. You have the choice of:

- ✧ Registering your Red Hat Enterprise Linux VM now (recommended) or
- ✧ Registering your Red Hat Enterprise Linux VM later

NOTE: See the [Using the CDK](#) chapter for information on obtaining Red Hat Enterprise Linux subscriptions, how registration works with the CDK, and details on why you need to register your system.

To register your system now, enter the username and password you use with the [Red Hat Customer Portal](#).

```
==> 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: *****
```

You can check whether your Vagrant box is running using the **vagrant status** command. Note that you must be in the same directory where your Vagrant file is located.

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

If the machine state shows as running, you are ready to start using your CDK. Refer to the following:

- ✎ The [Using the CDK](#) chapter will help you understand the different Vagrantfiles, plugins and interfaces needed to work with your CDK. It also discusses special topics, such as other ways to register your Red Hat Enterprise Linux virtual machines.
- ✎ The [Getting Started with Container Development Kit](#) guide provides information on getting started with container development.

CHAPTER 5. USING RED HAT CONTAINER DEVELOPMENT KIT

Once your Vagrant box is added, and one or more VMs is running, there are things you can do to begin using and managing your CDK environment. Topics associated with using your CDK are described in this chapter.

Two different Vagrantfiles are available in the Red Hat Developer Tools ZIP file that you can use to start differently configured Red Hat Enterprise Linux virtual machines from Vagrant:

rhel-ose

preconfigured instance of OpenShift Enterprise

rhel-k8s-singlenode-setup

preconfigured instance of Kubernetes

5.1. USING CDK TO DEVELOP WITH OPENSIFT

The OpenShift (**rhel-ose**) Vagrantfile is the recommended one for container development with the Red Hat Enterprise Linux CDK. When you run **vagrant up** from the rhel-ose directory, the Vagrantfile starts up a Red Hat Enterprise Linux virtual machine from the vagrant box you installed. The characteristics of that box are as follows:

- ✳ Sets up a CDK development environment with OpenShift (see <https://www.openshift.com/enterprise/whats-new.html>).
- ✳ Creates private networking to expose services from the Red Hat Enterprise Linux VM to the host system.

If you haven't started it already, to start the CDK Red Hat Enterprise Linux VM from the rhel-ose Vagrantfile and start using it, do the following:

1. Change to the rhel-ose directory and start the Red Hat Enterprise Linux VM (for Windows, use %USERPROFILE%\cdk\components\rhel\rhel-ose):

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

2. From the same directory, provision the box as follows:

NOTE: The **vagrant provision** isn't strictly required, but it can help overcome a timeout issue and gives you a way to see help messages for next steps you can take. You may have to run this several times.

```
$ vagrant provision
==> default: Running provisioner: shell...
    default: Running: inline script
...
==> default: You can now access OpenShift console on:
https://10.1.2.2:8443/console
==> default: Configured basic user: openshift-dev, Password:
devel
```

```
==> default: Configured cluster admin user: admin, Password:
admin
==> default:
==> default: To use OpenShift CLI, run:
==> default: $ vagrant ssh
==> default: $ oc login
==> default: To browse the OpenShift API documentation, follow
this link:
==> default: http://openshift3swagger-claytondev.rhcloud.com
==> default: Then enter this URL:
==> default: https://10.1.2.2:8443/swaggerapi/oapi/v1
```

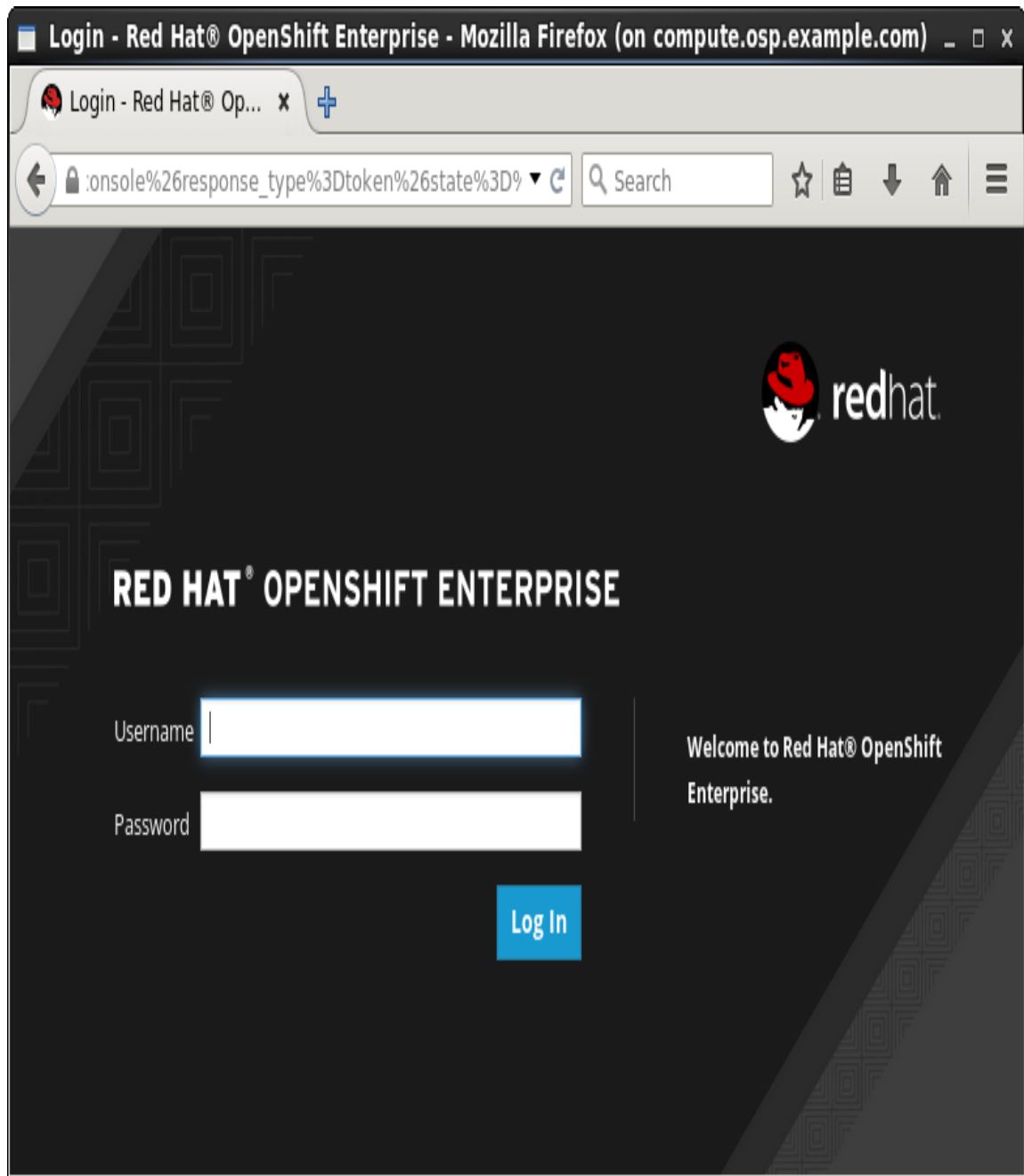
3. At the end of the provisioning, you see output describing how to access the OpenShift console or start using the **oc** command line to begin developing OpenShift projects. For more information on how to use the Red Hat Enterprise Linux VM that was created from the rhel-ose Vagrantfile, see the README.rst file in the rhel-ose directory.

The next sections step you through using the different interfaces to OpenShift.

5.1.1. Using CDK from OpenShift Web Interface

Here's one way to get started using the OpenShift Enterprise Web Console to develop containerized applications:

1. Point your Web browser to the URL shown for the OpenShift console. The default is: <https://10.1.2.2:8443/console>. Figure 5-1 shows an example of the OpenShift Enterprise login screen:



2. Login as either a basic user with the **openshift-dev** username and **devel** password or cluster admin user with the **admin** username and **admin** password.
3. To start using OpenShift, try one of the following:
 - ✱ For a walk-through of the OpenShift Web Console, refer to the [Getting Started for Developers: Web Console](#) guide
 - ✱ For help creating your first OpenShift application, refer to the [OpenShift Creating New Applications](#) guide.

NOTE: To stop the OpenShift (ose) service, you can run the **systemctl stop openshift** command.

5.2. USING CDK TO DEVELOP WITH KUBERNETES

When you run **vagrant up** from the **rhel-k8s-singlenode-setup** directory, the Vagrantfile starts a Red Hat Enterprise Linux virtual machine from the vagrant box you installed. The characteristics of that box are as follows:

- ✱ Starts up the docker service
- ✱ Starts up all the services needed to run an all-in-one Kubernetes master and node on the Red Hat Enterprise Linux VM

The resulting virtual machine is ready to start using the **kubectl** command to build Kubernetes pods, services, replication controllers and other features.

1. Change to the **rhel-docker-eclipse** directory and start the Red Hat Enterprise Linux VM (for Windows, use **%USERPROFILE%\cdk\components\rhel\rhel\rhel-k8s-singlenode-setup**):

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

2. Use **vagrant ssh** to access the Red Hat Enterprise Linux VM and check the status of kubernetes:

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

3. Begin using your Red Hat Enterprise Linux Kubernetes-enabled VM. For information on developing containerized applications to run in Kubernetes, refer to the [Launching container pods with Kubernetes](#) section of the "Get Started Orchestrating Containers with Kubernetes" guide.

5.3. USING DOCKER AND oc COMMAND-LINE TOOLS

The **docker** and **oc** commands, which can be used to control the Docker and OpenShift services respectively, are included in CDK, which allows you to run those commands directly when you are logged into the CDK virtual machine.

The **docker** and **oc** client commands can also be used on your host system to interact with the Docker and OpenShift services running inside the CDK virtual machine. For instructions on how to install and use these commands on Microsoft Windows, Mac OS X, and Linux, see the [Using the Docker Service](#) and [Using OpenShift Enterprise](#) chapters in the CDK Getting Started Guide.

5.4. USING CDK WITH DOCKER TOOLING ON ECLIPSE

You can access the Docker service running on the CDK Red Hat Enterprise Linux VM from an Eclipse IDE running on your local system. Here's how:

1. Install Eclipse. Install the Eclipse software on the system in which you are running the CDK and start up Eclipse. For example, to install and start Eclipse, do the following:

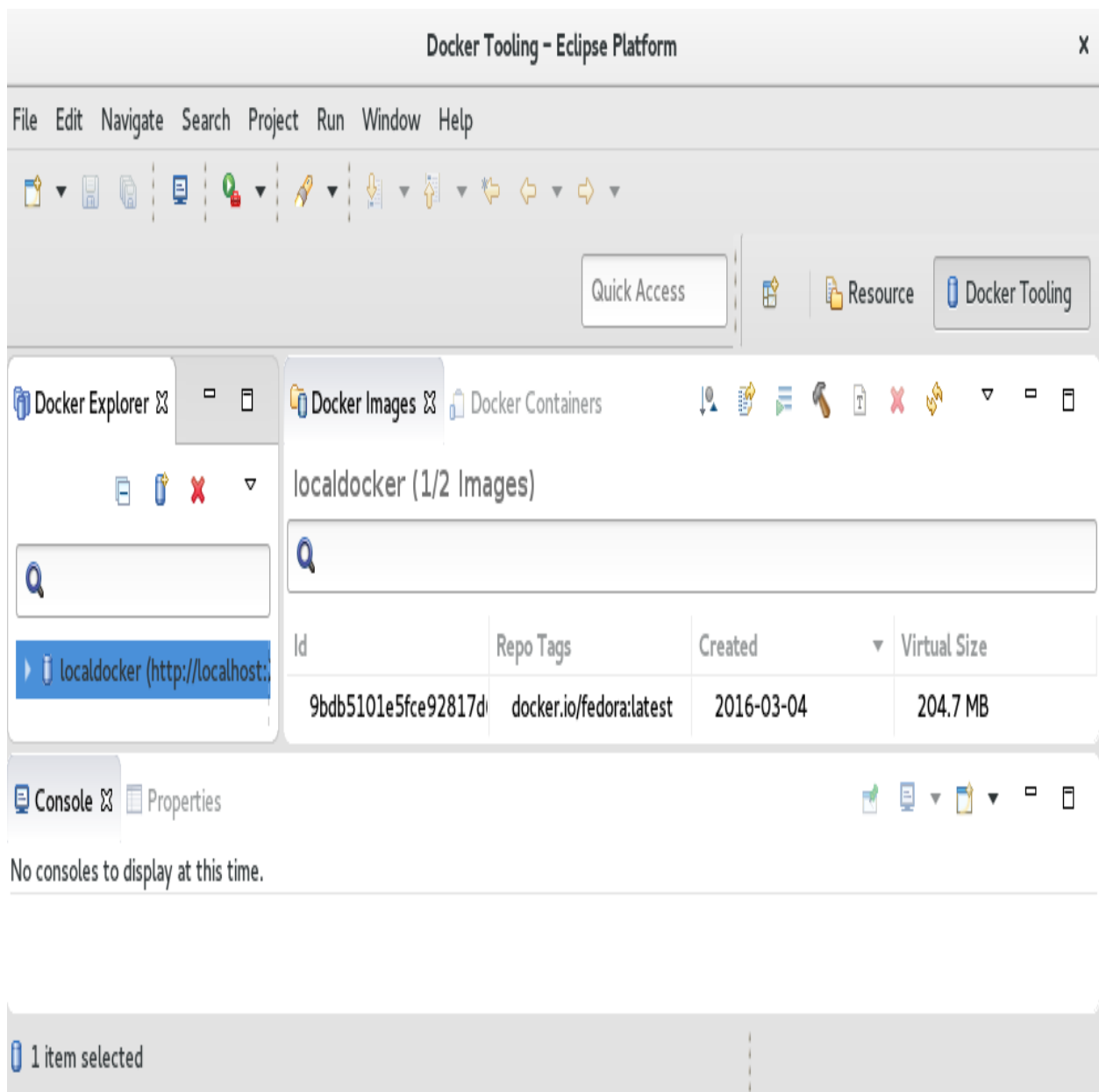
On a Fedora system:

```
$ sudo dnf install -y eclipse eclipse-linuxtools-docker
$ eval "$(vagrant service-manager env docker)"
$ eclipse &
```

On a Red Hat Enterprise Linux system:

```
$ sudo yum install -y devtoolset-4-eclipse devtoolset-4-eclipse-
linuxtools devtoolset-4-eclipse-linuxtools-docker.noarch
$ eval "$(vagrant service-manager env docker)"
$ scl enable devtoolset-4 'eclipse' &
```

2. Select a workspace. Choose the folder that will store our Eclipse work on your local desktop (such as `/home/chris/workspace`). The Eclipse Platform screen appears.
3. Open the Docker perspective. Select `Window → Perspective → Open Perspective → Other → Docker Tooling → OK`. Then select the Workbench icon. The Docker Tooling - Eclipse Platform screen should appear as shown in Figure 5-2



You can now begin using the Eclipse Docker Tooling screen to work with Docker images from inside the CDK's virtual machine.

5.5. WORKING WITH CDK VAGRANT VIRTUAL MACHINE

There are several ways you can gain access to the virtual machines you start up with your CDK. There are Vagrant commands you can use to gain shell access to a VM. Likewise, there are ways of accessing a CDK VM from Web UIs, such as OpenShift and Eclipse. The following section documents how to access your CDK VM from Vagrant.

5.5.1. Interacting with a Running Vagrant Box

To use a Vagrant box that is up and running, first change to the directory from which you started that box. For example, `cd %USERPROFILE%\cdk\components\rhel\rhel-ose` (in Windows) or `cd ~/cdk/components/rhel/rhel-ose` (in Fedora, Red Hat Enterprise Linux, or Mac OS X). From that location, you can run different **vagrant** commands to use or manage the box in different ways.

List the subcommands available to use with your **vagrant** command:

```
$ vagrant list-commands
```

Log into your Red Hat Enterprise Linux CDK Vagrant box using SSH:

```
$ vagrant ssh
```

This automatically logs you into the Red Hat Enterprise Linux virtual machine as the **vagrant** user. When you are done, exit the SSH session (type **exit**). To stop the Vagrant box, execute:

```
$ vagrant halt
```

If, at some point you want to delete the created VM and free virtualization resources, use the **vagrant destroy** command. Your Vagrantfile and the box image in the **.vagrant.d** in your home directory will remain, allowing you to recreate a fresh version of the environment with a subsequent **vagrant up** command.

```
$ vagrant destroy
```



Note

Do not delete the **.vagrant** subdirectory where Vagrant keeps its per-machine state without first using the **vagrant destroy** command to free virtualization (libvirt or Virtualbox) resources. If you no longer have the **.vagrant** subdirectory on a system using libvirt, you will need to use libvirt tools, such as **virt-manager** (GUI) or **virsh** (CLI), to manually delete the resources that were created by Vagrant before you can bring up a Vagrant box with the same name. Likewise, on a system using Virtualbox for virtualization, use the Virtualbox GUI to delete the resources you created with Vagrant.

After **vagrant destroy**, you will be able to bring the Vagrant box up again in its original, clean state.

5.5.2. Viewing all Vagrant Boxes and Directories

To view the status of all Vagrant boxes on your system and verify that your box was properly stopped, use:

```
$ vagrant global-status
```

CHAPTER 6. USING VAGRANT CDK PLUGINS

The CDK comes with several plugins that provide added features you can use with your CDK Vagrant boxes. Here are descriptions of those plugins and ways you might use them.

6.1. USING THE VAGRANT-SERVICE-MANAGER PLUGIN

With the `vagrant-service-manager` plugin, you can display and import environment variables needed to access the Docker, Kubernetes, and OpenShift features inside of your Vagrant CDK Red Hat Enterprise Linux VM from the desktop system. Here are examples of how to see and use that information:

```
$ cd $HOME/cdk/components/rhel/misc/rhel-k8s-singlenode-setup/
$ vagrant service-manager env docker
# Copying TLS certificates to
/home/chris/cdk/components/rhel/misc/rhel-k8s-singlenode-
setup/.vagrant/machines/default/libvirt/docker
# Set the following environment variables to enable access to the
# docker daemon running inside of the vagrant virtual machine:
export DOCKER_HOST=tcp://192.168.121.160:2376
export DOCKER_CERT_PATH=/home/chris/cdk/components/rhel/misc/rhel-
k8s-singlenode-setup/.vagrant/machines/default/libvirt/docker
export DOCKER_TLS_VERIFY=1
export DOCKER_MACHINE_NAME=be03562
# run following command to configure your shell:
# eval "$(vagrant service-manager env docker)"
```

In the above example, you can see that the Docker service is exposed from the IP address 192.168.121.160:2376 and TCP port 2376. A docker command or Eclipse IDE platform running on the local desktop could use that information to connect to the Docker service inside the VM. To have those values actually set inside your current shell, type the following:

```
$ cd $HOME/cdk/components/rhel/misc/rhel-k8s-singlenode-setup/
$ eval "$(vagrant service-manager env docker)"
```

To see useful information to provide manually to the `oc` command or OpenShift Web UI on the host, you can use the `vagrant-service-manager` plugin as follows:

```
$ cd $HOME/cdk/components/rhel/rhel-ose/
$ vagrant service-manager env openshift
# You can access the OpenShift console on:
https://192.168.121.122:8443/console
# To use OpenShift CLI, run: oc login https://192.168.121.122:8443
```

So you would type <https://192.168.121.122:8443/console> into the location box in your web browser to access the OpenShift Web UI from your desktop. Run **oc login <https://192.168.121.122:8443>** from the command line on your desktop to connect to the OpenShift service with the **oc** command.

6.2. USING THE VAGRANT-REGISTRATION PLUGIN

With the `vagrant-registration` plugin, you can manage the Red Hat subscriptions for your Red Hat

Enterprise Linux VMs through Vagrant. The plugin is capable of managing subscriptions with the type of subscription service that is available on the guest. However, at the moment it supports only Red Hat Enterprise Linux Subscription Manager (**subscription-manager** and related commands) and the older **rhn_register** command.

If you did not register your Red Hat Enterprise Linux VM when you started it with **vagrant up**, you should register it now. Registering your Red Hat Enterprise Linux system is highly recommended. Until you register, you cannot use the official Red Hat repositories to:

- ✧ Upgrade the software in your Red Hat Enterprise Linux virtual machine.
- ✧ Add more software packages to your Red Hat Enterprise Linux virtual machine.
- ✧ Add software packages to the Red Hat Enterprise Linux containers you build or run on that VM

Red Hat Enterprise Linux base container images are configured to have Docker use the credentials of the host system. So when you try to install packages inside of a container, the **yum** command uses the host credentials to gain access to those packages from Red Hat. Without a valid Red Hat subscription, you will not have a fully-functioning setup for building Red Hat Enterprise Linux containers.

The process of registering your CDK virtual machine with Red Hat is automated using a Vagrant plugin, **vagrant-registration**. By default, when a Vagrant box is started, you are prompted to enter your username and password for the Red Hat Customer Portal. The registration plugin automatically attaches the Vagrant box to an available subscription.

When a Red Hat Enterprise Linux VM is registered in the CDK, an identity and time-limited entitlement is created for that VM. Once it is registered, the VM does not need to be re-registered until the CDK entitlement expires. Once the time limit is up, that container loses access to the Red Hat software repositories (CDN).

You can register your CDK system with a valid Red Hat Enterprise Linux Developer Subscription or Developer Suite. Joining the [Red Hat Developers program](#) provides a path to getting the CDK. Once you register a CDK VM, you get a new entitlement that lasts for 90 days that does not come out of your pool. If you re-register the same VM, you will get a new 90 day entitlement. You can do this over and over.

There are a few things you should know about releasing a subscription:

- ✧ When you stop the Vagrant box, using either **vagrant halt** or **vagrant destroy**, the plugin automatically releases the Red Hat subscription.
- ✧ If you stop the box by some other means, such as a reboot of the host system, you may need to manually remove the subscription in order to use it on another box. Use subscription management on the [Red Hat Customer Portal Subscriptions](#) to find and delete the virtual system that is no longer being used.
- ✧ If you do not want to unregister a system when it is halted, you can set **config.registration.unregister_on_halt = false** in the selected Vagrantfile. In that case, the subscription will still be intact the next time you run **vagrant up** on that Vagrantfile.
- ✧ For more information on configuring the vagrant-registration plugin, see the [vagrant-registration GitHub](#) page.

If you do not have a Red Hat Enterprise Linux subscription, there are plans in the works to offer free entitlements to the Red Hat Enterprise Linux Developer Suite. Those entitlements will let you download and register a Red Hat Enterprise Linux host in the CDK. Keep an eye on the [Red Hat Developer's site](#) for details.

6.2.1. Automating the Registration Process (Saving Your Credentials)

It is recommended that you store your Red Hat credentials, so that you do not have to answer the prompts every time you bring up a Vagrant box. This is mandatory for complex Vagrantfiles that bring up multiple virtual machines from a single Vagrantfile.

To store your credentials, the following lines should be added to the per-user Vagrantfile. The path to that file is different for the different platforms:

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

» **Fedora, Red Hat Enterprise Linux, or Mac OS X:** ~/.vagrant.d/Vagrantfile

The configuration will be available to all boxes started under that user ID. The per-user Vagrantfile is not created automatically.

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

Alternatively, if you prefer not to store your Red Hat credential details in the file system, you can use the following configuration to retrieve them from environment variables. Remember to store your username in the **\$SUB_USERNAME** environment variable (SUB_USERNAME for Windows) and your password in the **\$SUB_PASSWORD** environment variable (SUB_PASSWORD for Windows) before starting Vagrant.

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

These settings may also be used in a specific Vagrantfile that will override the settings in the per-user **~/.vagrant.d/Vagrantfile**. In an existing Vagrantfile, there will already be a block that begins with **Vagrant.configure('2') do |config|**, so just add the two **config.registration** lines (see above) in the existing block.

For more information, see the **vagrant-registration-README.md** file in the **~/cdk/plugins** directory of the ZIP file containing the CDK software components.

6.3. USING THE VAGRANT-SSHFS PLUGIN

The vagrant-sshfs plugin allows you to mount a folder from your host system inside the CDK virtual machine for the purpose of sharing files between those two environment. The shared file system relies on the SSH File Transfer Protocol for sharing files. Although this is not the fastest or most efficient for file sharing today, the common availability of the SSH service makes this a popular way to share files between systems that don't require high volume of file copies between systems.

The vagrant-sshfs plugin works on all systems supported by the CDK (Windows, Mac, Fedora, and Red Hat Enterprise Linux). It also integrates with all CDK-supported virtualization technologies (Virtualbox and Libvirt). Here are the steps for configuring the vagrant-sshfs plugin to share files between your CDK host system and a CDK virtual machine running on that host.

1. Install the vagrant-sshfs plugin: If you haven't already done so, go to the plugin directory that you copied from the CDK zip file and install the plugin as follows:


```
$ cd ~/cdk/plugins
$ vagrant plugin install ./vagrant-sshfs
```

2. Edit Vagrantfile: Using any text editor, open the Vagrantfile you want to modify (OSE or Kubernetes) and add a `config.vm.synced_folder` entry.

Here is an example of what that entry might look like for a Red Hat Enterprise Linux or Fedora system:

```
config.vm.synced_folder "/home/joe", "/mnt/joefiles", type:
"sshfs"
```

In this example, when you run **vagrant up**, the `/mnt/joefiles` directory is created within the virtual machine and the `/home/joe` directory from the host is mount on that `/mnt/joefiles` directory inside the container. Any files you place in the `/home/joe` directory on the host will be accessible from `/mnt/joefiles` within the host.

Here is an entry for a Windows system:

```
config.vm.synced_folder 'c:\Users\chuck', "/mnt/host_home", type:
'sshfs', sshfs_opts_append: '-o umask=000 -o uid=1000 -o
gid=1000'
```

This example shares the user `chuck`'s home directory (`c:\Users\chuck`) to the `/mnt/host_home` directory on the host. The `umask` options sets the default permissions set on files created on the shared directory (wide open permission), while `uid=1000` and `gid=1000` sets the user and group IDs that will be assigned to files created in the shared directory.

3. Start the Vagrant virtual machines: Use the **vagrant up** command to start the CDK virtual machine from the Vagrantfile in the selected directory:

```
$ cd ~/cdk/components/rhel/rhel-ose
$ vagrant up
...
==> default: Rsyncing folder: /home/joe/cdk/components/rhel/rhel-
ose/ => /vagrant
==> default: Mounting SSHFS shared folders...
==> default: Mounting folder via SSHFS: /home/joe/ =>
/mnt/joefiles
==> default: Checking Mount..
Warning: Permanently added '192.168.121.125' (ECDSA) to the list
of known hosts.
==> default: Folder Successfully Mounted!
...
```

At this point, any files copied to the `/home/joe` directory on the host will be accessible from the `/mnt/joefiles` directory within the CDK virtual machines. Some reasons why this might fail include that you may have forgotten to install the plugin or the local `ssh` service may not be running.

For more details on how to configure the `vagrant-sshfs` service, refer to the [vagrant-sshfs GitHub documentation](#).

CHAPTER 7. TROUBLESHOOTING CDK PROBLEMS

If you are having trouble with the CDK in general or Vagrant in particular, refer to the information in this chapter.

7.1. TROUBLESHOOTING GENERAL CDK PROBLEMS

Q: How do I start Vagrant?

A: Vagrant is strictly command-line oriented. There are no menu entries. To run Vagrant, launch a command prompt or a terminal session on your machine. Vagrant should be automatically added to your path, so you only need to type **vagrant**.

Q: Something failed during `vagrant up` or it complained that an SSH command failed. If I try `vagrant up` again, it says the VM is already running.

A: There are a number of provisioning steps performed when Vagrant launches a virtual machine. These may come from plugins, such as the Vagrant Registration plugin for Red Hat, or from steps that are included in the Vagrantfile for provisioning the box on startup. Vagrant runs these commands by using SSH.

In many cases, errors during the provisioning step are not fatal, so the Vagrant box will still be running. Use the **vagrant status** command to see what the state of your box is. You can stop the Vagrant box with the **vagrant halt** command. If the machine is running, you should be able to log into it using the **vagrant ssh** command to examine the VM and see what went wrong.

While the box is running, you can rerun the provisioning steps by running the **vagrant provision** command.

Note that you need to be in the same directory where your Vagrantfile is. If you have lost track of where your Vagrantfile is, use the **vagrant global-status** command to list the boxes that you have started and the directory where the Vagrantfile and state is stored.

Q: The `vagrant up` command fails with error message "could not find capabilities for domaintype=kvm"

A: Any system running the CDK (Windows, Mac, or RHEL), requires not only that the computer it is running on have Virtualization Support, but also that the support be enabled in the BIOS. On a RHEL or Fedora system, if Virtualization Support is not enabled, when you run **vagrant up**, you will see a message similar to the following: "Error while creating domain: Error saving the server: Call to virDomainDefineXML failed: invalid argument: could not find capabilities for domaintype=kvm". Enabling Virtualization Support in the computer's BIOS should fix the problem.

Run the **virt-host-validate** command to check whether or not Virtualization has been enabled on the computer.

Q: Red Hat registration failed, or I entered the wrong username and password. How can I manually register the box or check my subscription status?

A: You can run the **subscription-manager** tool to log in, register or unregister the box, check your subscription status, or enable available software repositories as you would on any other Red Hat Enterprise Linux installation. However, you need to use the **sudo -i** command to be root because the root password for the Red Hat Enterprise Linux Vagrant boxes is not distributed. The **sudo** utility has been setup for the vagrant user to run any commands.

For more information, see [Red Hat Subscription Management](#).

Q: How do I free up my Red Hat subscription used by the Vagrant box?

A: The Vagrant Registration plugin automatically detaches the box from the Red Hat subscription when you shut down the box using the **vagrant halt** or **vagrant destroy** commands. If the box is not shut down by Vagrant, this does not happen. If you still have the box set up, you should be able to bring the box up again using the **vagrant up** command and then shut it down correctly with **vagrant halt** or **vagrant destroy**, which will unregister the box from Red Hat Subscription Management.

Alternatively, you can use the subscription management on the [Red Hat Customer Portal](#) to find and delete the virtual system that is no longer being used.

Q: I need to make changes as root on the Vagrant boxes, what is the root password?

A: The **sudo** utility has been set up for the **vagrant** user to run any commands as root. You can use the **sudo** command to run a single command as root, or use **sudo -i** to start an interactive root shell.

Q: Running **vagrant global-status** gives me a **permission denied** error.

A: Run the **id** command to check whether your regular user ID is a member of the **vagrant** group. If you skipped this step before, log out and log back in for the changes to take effect. Adding the user to the **vagrant** group works in conjunction with the PolicyKit rule that was added during the steps run under root to allow non-root users to work with libvirt.

7.2. TROUBLESHOOTING CDK PROBLEMS WITH LIBVIRT

Q: When I try to run Vagrant, I get an error message about connecting to libvirt. Do I need to be root to run Vagrant?

A: One of the Software Installation and Configuration steps installs a PolicyKit rule in the **/etc/polkit-1/rules.d/** directory that allows non-root users to run Vagrant and perform libvirt operations that are normally restricted to root users as long as the user is a member of the **vagrant** group. The error message is:

```
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
```

To resolve this, make sure your user ID is a member of the **vagrant** group. You can do this by running the **id** command. Review the installation steps listed above to verify that the PolicyKit rule is installed in **/etc/polkit-1/rules.d/**. Restart the PolicyKit and libvirtd services for the changes to take effect:

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

Q: The vagrant command is failing because libvirt says the resource it is trying to create already exists.

A: Normally, Vagrant takes care of all interaction with libvirt, creating and destroying resources as necessary. Running the **vagrant destroy** command should free up any resources that were allocated to an environment. However, if the Vagrant state directory, **.vagrant** is deleted, or if a Vagrant operation is interrupted, before it can clean up, it is possible that libvirt resources are left around that cannot be removed using **vagrant destroy**. If that happens, you need to use the **virsh** command-line utility or **virt-manager**, a graphical tool, to clean up.

Note that these tools must be run as root (or with the **sudo** command) to see all of the libvirt resources on the system. If you do not run them as root, it appears that there are no allocated resources.

Q: I want to install an updated Vagrant box, but when I run it I keep seeing the older box still being used

A: For the CDK with libvirt, old Vagrant boxes are not automatically removed when the new one is installed. As a result, even after adding a new box, the old one is used in place of the new box. To add a new Vagrant box for an updated CDK, you need to destroy the running boxes, delete the **.vagrant/** directory in each directory where you ran **vagrant up** and remove the image created by each box.

If you ran both the OSE and Kubernetes Vagrantfiles, here's how you could delete them completely before adding the updated box. First go to each directory to stop the running box and remove the **.vagrant/** directory.

Warning

The **rm -fr** command is a destructive command. Be absolutely sure you identify the proper directory to delete before running this command.

```
$ cd ~/cdk/components/rhel/misc/rhel-k8s-singlenode-setup
$ vagrant destroy && rm -fr .vagrant/
$ cd ~/cdk/components/rhel/rhel-ose
$ vagrant destroy && rm -fr .vagrant/
```

Next remove the images representing the Vagrant boxes in the libvirt images directory and restart libvirt.

```
$ sudo rm -fr /var/lib/libvirt/images/*_0.img && sudo systemctl
restart libvirtd
```

You can now proceed to adding the new Vagrant box.

7.3. TROUBLESHOOTING CDK PROBLEMS ON WINDOWS SYSTEMS

Q: The vagrant ssh command is not working, it can't find ssh.exe.

A: In order to use **vagrant ssh** on a Windows system, you need to have **ssh.exe** in your path. It is recommended to install Cygwin and ensure that **ssh.exe** is installed and in your path.

Q: The vagrant up command refused to run because rsync is not in the path.

A: If **rsync** is being used for Vagrant synchronized folders on a Windows system, either because it was explicitly selected, or it was the last available choice, Vagrant will not start a Vagrant box unless **rsync.exe** is available in the path. It is recommended to install Cygwin and ensure that **rsync.exe** is in your path.

Q: The vagrant up command suddenly refused to start because rsync is not in the path, it was working before without rsync.exe being available using VirtualBox Guest Additions.

A: If your Vagrant box is not configured with VirtualBox Guest Additions, then Vagrant attempts to use **rsync** for the shared folders. Using **rsync** is the supported, recommended way (VirtualBox Guest Additions may work, but is not officially supported). If **rsync** cannot be found, Vagrant generates an error and does not start.

Note that if a new kernel gets installed, you may need to rebuild VirtualBox Guest Additions on the Vagrant box. You can do this by running:

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

Q: Something failed during vagrant up or it complained that an SSH command failed. If I try vagrant up again, it says the VM is already running.

- A:** There are a number of provisioning steps performed when Vagrant launches a virtual machine. These may come from plugins, such as the Vagrant Registration plugin for Red Hat, or from steps that are included in the Vagrantfile for provisioning the box on startup. Vagrant runs these commands by using SSH.

In many cases, errors during the provisioning step are not fatal, so the Vagrant box will still be running. Use the **vagrant status** command to see what the state of your box is. You can stop the Vagrant box with the **vagrant halt** command. If the machine is running, you should be able to log into it using the **vagrant ssh** command to examine the VM and see what went wrong.

While the box is running, you can rerun the provision steps by running the **vagrant provision** command.

Note that you need to be in the same directory where your Vagrant file is. If you have lost track of where your Vagrantfile is, use the **vagrant global-status** command to list all the boxes that you have started and the directory where the Vagrantfile and state is stored.
