



Red Hat Container Development Kit 3.0 Installation Guide

Guide to installing Red Hat Container Development Kit

Brian Brock
Red Hat Developer Group Documentation
Team

Robert Krátký

Chris Negus

Red Hat Container Development Kit 3.0 Installation Guide

Guide to installing Red Hat Container Development Kit

Brian Brock
bbrock@redhat.com

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

This guide contains instructions on how to install Red Hat Container Development Kit on Microsoft Windows, macOS, and Red Hat Enterprise Linux. Information and hands-on guidance is provided for obtaining all prerequisites for each of the supported platforms. You will also learn how to launch the Container Development Environment provided by Red Hat Container Development Kit, so that you can start to develop containerized applications.

Table of Contents

CHAPTER 1. WHAT IS RED HAT CONTAINER DEVELOPMENT KIT?	3
1.1. RELATION TO MINISHIFT DOCUMENTATION	3
1.2. WHAT IS IN CONTAINER DEVELOPMENT KIT	3
1.3. WHERE YOU CAN RUN CONTAINER DEVELOPMENT KIT	4
1.4. OBTAINING AND SETTING UP CONTAINER DEVELOPMENT KIT	4
1.5. INSTALLATION COMPONENTS	5
1.6. ADDITIONAL RESOURCES	5
CHAPTER 2. CONTAINER DEVELOPMENT KIT TECHNOLOGIES	6
CHAPTER 3. INSTALLING DOCKER MACHINE DRIVERS	7
3.1. OVERVIEW	7
3.2. KVM DRIVER	7
3.3. ON RED HAT ENTERPRISE LINUX	7
3.4. XHYVE DRIVER	7
3.5. HOMEBREW INSTALL	7
3.5.1. Manual Installation	8
CHAPTER 4. INSTALLATION PREREQUISITES	10
4.1. HARDWARE REQUIREMENTS	10
4.2. HYPERVISOR REQUIREMENTS	10
CHAPTER 5. INSTALLING RED HAT CONTAINER DEVELOPMENT KIT	11
5.1. INSTALLING ON MICROSOFT WINDOWS AND MACOS	11
5.2. INSTALLING ON RED HAT ENTERPRISE LINUX	11
5.2.1. Configuring Container Development Kit on Red Hat Enterprise Linux	12
5.3. REGISTER THE RED HAT ENTERPRISE LINUX VIRTUAL MACHINE	13
5.4. SKIP REGISTRATION	13
CHAPTER 6. UNINSTALLING CONTAINER DEVELOPMENT KIT	14
CHAPTER 7. TROUBLESHOOTING	15
7.1. SPECIAL CHARACTERS CAUSE PASSWORDS TO FAIL	15
7.2. UNDEFINING VIRSH SNAPSHOTS FAIL	15
7.3. KVM: ERROR CREATING NEW HOST: DIAL TCP: MISSING ADDRESS	16
7.4. KVM: FAILED TO CONNECT SOCKET TO /VAR/RUN/LIBVIRT/VIRTLOGD-SOCK	16
7.5. KVM: DOMAIN 'MINISHIFT' ALREADY EXISTS...	16
7.6. XHYVE: COULD NOT CREATE VMNET INTERFACE	17
7.7. VIRTUALBOX: ERROR MACHINE DOES NOT EXIST	17

CHAPTER 1. WHAT IS RED HAT CONTAINER DEVELOPMENT KIT?

The **Red Hat Container Development Kit** is a collection of technologies which allows users to develop container-based applications. Those applications reside in a **Docker** container, and typically several Docker containers are grouped together in a virtual machine (VM). This arrangement mirrors the environment which the application runs in on OpenShift.

Container Development Kit can run on **Red Hat Enterprise Linux**, **macOS**, and **Microsoft Windows**. Installation procedures are slightly different between platforms, but generally Container Development Kit runs the same once installed.

This guide describes the installation steps, from preparing the system to launching Container Development Kit for the first time. See the [Container Development Kit Getting Started Guide](#) to get an idea of how to use Container Development Kit after installing.

1.1. RELATION TO MINISHIFT DOCUMENTATION

This document is based on the documentation for Minishift that can be found at minishift.io, and considers that documentation to be the upstream documentation. However, this documentation has a focus on functionality in Red Hat Container Development Kit. Some upstream features that are not specifically relevant to Container Development Kit users has been modified or removed.

1.2. WHAT IS IN CONTAINER DEVELOPMENT KIT

Once the Container Development Environment from Container Development Kit is running on your system, you can begin exploring the contents. Some services and tools run automatically when you boot the virtual machine while others require configuration. Here is a list of some of those features:

- ✦ **OpenShift Container Platform:** A containerized version of OpenShift Container Platform is included in the Container Development Environment. OpenShift Container Platform 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 Container Development Environment. The Environment is configured to have the **docker** daemon start automatically when you boot the virtual machine. With the **docker** command, you can build, run, start, stop, investigate, and otherwise work with individual containers.
- ✦ **Linking to Eclipse:** The Container Development Environment lets you connect to it from the Eclipse IDE with **Linux Tools** and the **Docker Tooling** plug-in. This allows Container Development Kit users to manage containers from a graphical interface on their host system.

**Note**

Red Hat Container Development Kit 3.0 now uses minishift. If you used Container Development Kit releases prior to 3.0, you may already be familiar with running **Vagrant**. Minishift creates kvm boxes directly to provide similar functionality, without Vagrant, as well as additional functionality that is more appropriate for the new environment.

1.3. WHERE YOU CAN RUN CONTAINER DEVELOPMENT KIT

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. A hypervisor is required in each case, and the Red Hat Enterprise Linux virtual machine will use that hypervisor.

**Note**

A hypervisor is software that runs on your desktop operating system that allows the Red Hat Enterprise Linux virtual machine to run. This Red Hat Enterprise Linux virtual machine is required to use Red Hat Container Development Kit.

Container Development Kit can be installed on the following systems:

- ✦ **Microsoft Windows:** You can use a 64-bit version of Microsoft Windows to install Container Development Kit. Windows 7 or later is required. You can choose to use either VirtualBox or Hyper-V as your hypervisor on Microsoft Windows.
- ✦ **macOS:** You can use an Intel-based Apple Mac to install and run Container Development Kit. The computer should have at least 4 GB of RAM and be running a recent, 64-bit version of macOS, such as 10.11 (El Capitan). You can use either VirtualBox or xhyve as your hypervisor on macOS.
- ✦ **Red Hat Enterprise Linux:** The latest version of Red Hat Enterprise Linux 7 is recommended for installing Container Development Kit. A 64-bit computer with at least 4 GB of RAM is required. You can use either VirtualBox or KVM as your hypervisor on Red Hat Enterprise Linux.

See the Container Development Kit installation procedure for each system for more detailed hardware and software requirements.

**Note**

Virtualization support must be included and enabled in the computer's BIOS, regardless of operating system.

1.4. OBTAINING AND SETTING UP CONTAINER DEVELOPMENT KIT

Container Development Kit is available from the [Red Hat Customer Portal](#) to anyone with a valid Red Hat Enterprise Linux Developer Subscription. Joining the [Red Hat Developers program](#) also provides a path to getting Container Development Kit.

This guide provides the following instructions for installing Container Development Kit to begin developing containerized applications:

- ✦ Choosing your development system (Microsoft Windows, macOS, or Red Hat Enterprise Linux).
- ✦ Making Minishift available to users.
- ✦ Initial configuration of minishift.
- ✦ Starting the Container Development Environment from minishift.

1.5. INSTALLATION COMPONENTS

Regardless of which platform you choose as the workstation for using Container Development Kit, you will use Minishift to start and manage it.

Minishift includes a virtual machine, with Red Hat Enterprise Linux installed on that virtual machine.

- ✦ Minishift is a single-user version of OpenShift, with additional functionality needed by users developing on one system.
- ✦ Docker is an open source tool for manipulating containers in linux.

Now that you understand the basics of what is inside Container Development Kit, choose from one of the next chapters to learn how to obtain Container Development Kit and install it on a Microsoft Windows, macOS, or Red Hat Enterprise Linux system.

1.6. ADDITIONAL RESOURCES

- ✦ [Container Development Kit Getting Started Guide](#): Steps you through your first experiences using Container Development Kit.
- ✦ [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.

CHAPTER 2. CONTAINER DEVELOPMENT KIT TECHNOLOGIES

Container Development Kit is comprised of several technologies:

- ✦ **Red Hat Enterprise Linux** is the operating system installed on a VM. You will control the Minishift cluster through that installation, rather than the operating system installed on your host system (desktop).
- ✦ The **Kernel Virtual Machine (KVM)** technology within the linux kernel. A Virtual Machine, usually called a "VM" or "box", is a fully virtualized system running an operating system. The kernel of a KVM box is independent of the kernel on the host system. In Container Development Kit, a KVM box running Red Hat Enterprise Linux serves as the host for a cluster of containers.
- ✦ **Docker** runs a copy of an operating system in a container, except for the kernel. As a result, it can directly access hardware on the host system. It provides a higher level of security and independence from operating system changes, and helps with maintaining the same configuration across a cluster.
- ✦ **OpenShift** is a system for managing docker containers on Red Hat Enterprise Linux virtual machines. A VM containing containerized apps, managed by OpenShift, is called a "pod". OpenShift deploys pods to host systems in a cluster as needed, often running more than one pod on each computer system.
- ✦ **Minishift** is a tool that helps you run OpenShift locally by launching a single-node OpenShift cluster inside a virtual machine. With Minishift you can try out OpenShift or develop with it, day-to-day, on your local machine. Minishift forms the core of Container Development Kit and also controls the Container Development Kit installation and configuration.
- ✦ **Eclipse** is the Integrated Design Environment (IDE) used in the **Red Hat JBoss Developer Studio**. Container Development Kit adds functionality to the IDE to develop and manipulate a cluster of containers.

CHAPTER 3. INSTALLING DOCKER MACHINE DRIVERS

3.1. OVERVIEW

Minishift uses Docker Machine and its driver plugin architecture to provide a consistent way to manage the OpenShift VM. Minishift embeds VirtualBox and VMware Fusion drivers so no additional steps are required to use them.

However, other drivers require manual installation steps that are described in the following sections.

3.2. KVM DRIVER

Minishift is currently tested against **docker-machine-driver-kvm** version 0.7.0.

Install and execute the KVM binary as follows:

```
~]$ sudo curl -L https://github.com/dhiltgen/docker-machine-kvm/releases/download/v0.7.0/docker-machine-driver-kvm -o /usr/local/bin/docker-machine-driver-kvm
~]$ sudo chmod +x /usr/local/bin/docker-machine-driver-kvm
```

For more information, see the GitHub documentation of the [docker machine KVM driver](#).

3.3. ON RED HAT ENTERPRISE LINUX

1. Install **libvirt** and **qemu-kvm** on your system:

```
~]$ sudo yum install libvirt qemu-kvm
```

1. Add yourself to the **libvirt** group so that you do not need to use **sudo**:

```
~]$ sudo usermod -a -G libvirt <username>
```

1. Update your current session to apply the group change:

```
~]$ newgrp libvirt
```

3.4. XHYVE DRIVER

Minishift is currently tested against **docker-machine-driver-xhyve**.

3.5. HOMEBREW INSTALL

You can verify the installed version of the xhyve driver on your system before you install.

```
~]$ brew info --installed docker-machine-driver-xhyve
docker-machine-driver-xhyve: stable 0.3.1 (bottled), HEAD
```

```
Docker Machine driver for xhyve
https://github.com/zchee/docker-machine-driver-xhyve
/usr/local/Cellar/docker-machine-driver-xhyve/0.3.1 (3 files, 13.2M) *
  Poured from bottle on 2016-12-20 at 17:44:35
From: https://github.com/Homebrew/homebrew-
core/blob/master/Formula/docker-machine-driver-xhyve.rb
```

To install the latest version of the driver with brew:

```
~]$ brew install docker-machine-driver-xhyve

# docker-machine-driver-xhyve need root owner and uid
~]$ sudo chown root:wheel $(brew --prefix)/opt/docker-machine-driver-
xhyve/bin/docker-machine-driver-xhyve
~]$ sudo chmod u+s $(brew --prefix)/opt/docker-machine-driver-
xhyve/bin/docker-machine-driver-xhyve
```

For more information, see the GitHub documentation for the [xhyve driver](#).

3.5.1. Manual Installation

To manually install the xhyve driver, you need to download and install the **docker-machine-driver-xhyve** binary and place it in a directory which is on your *PATH*. `/usr/local/bin` is most likely a good choice, since it is the default installation directory for Docker Machine binaries.

The following steps explain the installation of the **docker-machine-driver-xhyve** binary to the `/usr/local/bin/` directory:

1. Download the **docker-machine-driver-xhyve** binary using:

```
~]$ sudo curl -L https://github.com/zchee/docker-machine-driver-
xhyve/releases/download/v0.3.1/docker-machine-driver-xhyve -o
/usr/local/bin/docker-machine-driver-xhyve
```

2. Enable **root** access for the **docker-machine-driver-xhyve** binary and add it to the default wheel group:

```
~]$ sudo chown root:wheel /usr/local/bin/docker-machine-driver-
xhyve
```

3. Set owner User ID (SUID) for the binary as follows:

```
~]$ sudo chmod u+s /usr/local/bin/docker-machine-driver-xhyve
```



Note

The downloaded **docker-machine-driver-xhyve** binaries are compiled against a specific version of OS X. It is possible that the driver fails to work after an OS X version upgrade. In this case you can try to compile the driver from source:

```
~]$ go get -u -d github.com/zchee/docker-machine-driver-xhyve
~]$ cd $GOPATH/src/github.com/zchee/docker-machine-driver-xhyve

# Install docker-machine-driver-xhyve binary into
/usr/local/bin
~]$ make install

# docker-machine-driver-xhyve need root owner and uid
~]$ sudo chown root:wheel /usr/local/bin/docker-machine-driver-
xhyve
~]$ sudo chmod u+s /usr/local/bin/docker-machine-driver-xhyve
```

For more information, refer to the [xhyve driver](#) documentation on GitHub.

CHAPTER 4. INSTALLATION PREREQUISITES

Make sure that your computer is ready for installing Container Development Kit.

4.1. HARDWARE REQUIREMENTS

Check that the following requirements are met:

- ✦ BIOS support. The host computer system must have hardware virtualization enabled in the BIOS.
- ✦ A Red Hat Developer account.

4.2. HYPERVISOR REQUIREMENTS

Container Development Kit requires a hypervisor to start the virtual machine on which the OpenShift cluster is provisioned. Make sure that the hypervisor of your choice is installed and enabled on your system before you start Container Development Kit.

Depending on your host OS, you have the choice of the following hypervisors:

macOS

- ✦ [xhyve](#) (default)
- ✦ [VirtualBox](#)

Red Hat Enterprise Linux

- ✦ [KVM](#) (default)
- ✦ [VirtualBox](#)

Microsoft Windows

- ✦ [Hyper-V](#) (default)



Note

To be able to use Hyper-V, ensure that, after you [enable Hyper-V](#), you also [add a Virtual Switch](#) using the Hyper-V Manager. Make sure that you pair the virtual switch with a *network card (wired or wireless) that is connected to the network*.

- ✦ [VirtualBox](#). Use **VirtualBox 5.1.2** or later.

CHAPTER 5. INSTALLING RED HAT CONTAINER DEVELOPMENT KIT



Note

The installation process has been greatly simplified in Red Hat Container Development Kit 3.0, compared with earlier 2.x releases.

5.1. INSTALLING ON MICROSOFT WINDOWS AND MACOS

Container Development Kit will be installed in the user's personal directory. Virtualized boxes and configuration files will be stored under that directory.

Complete the following steps as a normal user (not **root**), to install Container Development Kit:

1. Download the Red Hat Development Suite Installer from developers.redhat.com.
2. Run the installation program for Red Hat Development Suite that you downloaded.
3. Select Red Hat Container Development Kit components from the Red Hat Development Suite installer menu.

This will automatically select hypervisor software for installation. The default hypervisor software for Microsoft Windows and macOS is VirtualBox, however you may also use Hyper-V on Microsoft Windows and xhyve on macOS.

5.2. INSTALLING ON RED HAT ENTERPRISE LINUX

Red Hat Container Development Kit installation on Red Hat Enterprise Linux uses the native RPM technology provided by Red Hat Enterprise Linux. Installing Container Development Kit requires several steps:

1. As **root**, create the `/etc/yum/repos.d/rh-devsuite.repo` file and add the following content:

```
[rh-devsuite]
name=Red Hat Developemnt Suite RPMs
baseurl=file:///opt/rh-devsuite-repo/
gpgkey=https://www.redhat.com/security/data/a5787476.txt
enabled=1
gpgcheck=1
```

2. To install Red Hat Container Development Kit, run the following command:

```
~]$ sudo yum install cdk-minishift
```

3. To setup Red Hat Container Development Kit, first create a symbolic link for minishift:

```
~]$ sudo ln -s /var/lib/cdk-minishift-3.0.0/minishift
/usr/bin/minishift
```

4. Install required CDK components, as a normal user:

```
~]$ minishift setup-cdk --force --default-vm-driver="kvm"
```

Container Development Kit will install a KVM box in your home directory. You can find that installation under `$HOME/.minishift` as the user running Container Development Kit.

5. Create a symbolic link to provide the `oc` utility:

```
~]$ sudo ln -s /home/$(whoami)/.minishift/cache/oc/v3.5.5.8/oc
/usr/bin/oc
```

Note

`oc` must be in your `PATH` environment variable to execute Container Development Kit commands from the command line.

5.2.1. Configuring Container Development Kit on Red Hat Enterprise Linux

The `minishift start` command creates an OpenShift instance using the [cluster up](#) approach.

For this purpose it copies the `oc` binary onto your host. You will find it under `~/.minishift/cache/oc/3.5/oc`, provided you use Minishift's default version of OpenShift.

You can add this binary to your `PATH` using `minishift oc-env` which displays the command you need to type into your shell. The output of `oc-env` will differ depending on OS and shell type.

```
~]$ minishift oc-env
export PATH="/Users/john/.minishift/cache/oc/v1.5.0:$PATH"
# Run this command to configure your shell:
# eval $(minishift oc-env)
```

Note

Users who have already installed the executables `minishift` or `oc` need to ensure that the new executables that minishift provides do not conflict with those already installed. This is handled by default with `oc-env`, by putting the new Minishift executables in `PATH` first. To access the previous version, run `which oc` or `which minishift` and use the output to specify the absolute filename including directory.

As part of the `minishift start` command there is also a `minishift oc context` created. This context contains the configuration to communicate with your OpenShift cluster.

Minishift activates this context automatically, but if you need to switch back to it after for example, logging into another OpenShift instance, you can run:

```
~]$ oc config use-context minishift
```

For an introduction to `oc` usage, refer to the [Get Started with the CLI](#) topic in the OpenShift documentation.

5.3. REGISTER THE RED HAT ENTERPRISE LINUX VIRTUAL MACHINE

Start Container Development Kit for the first time. You will need to set the environment variables **MINISHIFT_USERNAME** and **MINISHIFT_PASSWORD** to activate the Red Hat Enterprise Linux subscription used by your Minishift cluster. This will register the virtual machine that runs Red Hat Enterprise Linux, until it is released at shutdown. For further assistance in taking your first steps with Container Development Kit, see the [Container Development Kit Getting Started Guide](#).

```
~]$ export MINISHIFT_USERNAME=<Red_Hat_username>
~]$ export MINISHIFT_PASSWORD=<Red_Hat_password>
~]$ minishift start
```

Instead of the environment variables **MINISHIFT_USERNAME** and **MINISHIFT_PASSWORD**, you may also specify the username and password on the command line:

```
~]$ minishift --username=<Red Hat username> --password=<Red Hat
password> start
```

The **start** command should download any required OpenShift functionality, start OpenShift services, and print some helpful information about using the command line and web interfaces.

At this point, you should be able to use the tools **minishift** and **oc** to manage your Container Development Kit cluster.

5.4. SKIP REGISTRATION

If you want to work with the virtual machine offline, you must register it to activate a Red Hat Enterprise Linux first. Container Development Kit also provides the ability for the VM to remain registered, so that you can skip that step. Skipping registration is convenient for working offline.

First, login with **minishift** while online:

```
~]$ minishift --username=<Red Hat username> --password=<Red Hat
password> start
```

Next, stop minishift with a **--skip-registration** option. This shuts down the VM without releasing the Red Hat Enterprise Linux subscription.

```
~]$ minishift stop --skip-unregistration
```

Disconnect or log out as needed. While you are offline, use the **--skip-registration** option to skip the attempt to register the Red Hat Enterprise Linux VM at boot. The Red Hat Enterprise Linux subscription is still active on the virtual machine, so you do not need to specify your username and password.

```
~]$ minishift start --skip-registration
```

CHAPTER 6. UNINSTALLING CONTAINER DEVELOPMENT KIT

1. Delete the Minishift VM and any VM-specific files.

```
~]$ minishift delete
```

This command deletes everything in the **MINISHIFT_HOME/.minishift/machines/minishift** directory. Other cached data and the [persistent configuration](#) are not removed.

2. To completely uninstall Minishift, delete everything in the **MINISHIFT_HOME** directory (default **~/.minishift**) and **~/.kube**:

```
~]$ rm -rf ~/.minishift
~]$ rm -rf ~/.kube
```

3. With your hypervisor management tool, confirm that there are no remaining artifacts related to the Minishift VM. For example, if you use KVM, you need to run the **virsh** command.

CHAPTER 7. TROUBLESHOOTING

This section contains solutions to common problems that you might encounter while using Minishift.

7.1. SPECIAL CHARACTERS CAUSE PASSWORDS TO FAIL

Depending on your operating system and shell environment, certain special characters can trigger variable interpolation and therefore cause passwords to fail.

Workaround: When creating and entering passwords, wrap the string with single quotes in the following format: '**<password>**'

7.2. UNDEFINING VIRSH SNAPSHOTS FAIL

If you use **virsh** on KVM/libvirt to create snapshots in your development workflow, and then use **minishift delete** to delete the snapshots along with the VM, you might encounter the following error:

```
~]$ minishift delete
Deleting the {mshift} VM...
Error deleting the VM: [Code-55] [Domain-10] Requested operation is
not valid: cannot delete inactive domain with 4 snapshots
```

Cause: The snapshots are stored in `~/minishift/machines`, but the definitions are stored in `var/lib/libvirt/qemu/snapshot/minishift`.

Workaround: To delete the snapshots you need to perform the following steps.

1. Delete the definitions.

```
$ sudo virsh snapshot-delete --metadata minishift <snapshot-name>
```

2. Undefine the Minishift domain.

```
~]$ sudo virsh undefine minishift
```

You can now run **minishift delete** to delete the VM and restart Minishift.

Note

In case the above steps do not resolve the issue, you can also use the following command to delete the snapshots:

```
~]$ rm -rf ~/.minishift/machines
```

It is recommended to avoid using metadata when you create snapshots. To make sure of this, you can specify the `--no-metadata` flag. For example:

```
~]$ sudo virsh snapshot-create-as --domain vm1 overlay1 --diskspec
vda,file=/export/overlay1.qcow2 --disk-only --atomic --no-metadata
```

■

7.3. KVM: ERROR CREATING NEW HOST: DIAL TCP: MISSING ADDRESS

- ✎ The problem is likely that the **libvirtd** service is not running. You can check this with the following command:

```
~]$ systemctl status libvirtd
```

If **libvirtd** is not running, start it and enable it to start on boot:

```
~]$ systemctl start libvirtd
~]$ systemctl enable libvirtd
```

- ✎ It is also possible that you are running an unsupported version of libvirt, leading to the same error. Ensure that you're using at least version 1.3. Older operating systems (like the unsupported Fedora 23) may use an earlier version of libvirt that is incompatible with Red Hat Container Development Kit 3.0

7.4. KVM: FAILED TO CONNECT SOCKET TO /VAR/RUN/LIBVIRT/VIRTLOGD-SOCK

The problem is likely that the **virtlogd** service is not running. You can check this with the following command:

```
~]$ systemctl status virtlogd
```

If **virtlogd** is not running, start it and enable it to start on boot:

```
~]$ systemctl start virtlogd
~]$ systemctl enable virtlogd
```

7.5. KVM: DOMAIN 'MINISHIFT' ALREADY EXISTS...

If you try **minishift start** and hit the above error, ensure that you use **minishift delete** to delete the VMs created earlier by you. However if this fails and you wish to completely clean up Minishift and start fresh do the following:

1. Check if any existing Minishift VM are running:

```
~]$ sudo virsh list --all
```

2. If any Minishift VM is running, stop it:

```
~]$ sudo virsh destroy minishift
```

3. Delete the VM:

```
~]$ sudo virsh undefine minishift
```

4. Delete the `.minishift/machines` directory using:

```
~]$ rm -rf ~/.minishift/machines
```

7.6. XHYVE: COULD NOT CREATE VMNET INTERFACE

The problem is likely that the xhyve driver is not able to clean up `vmnet` when a VM is removed. `vmnet.framework` determines the IP address based on the following files:

- » `/var/db/dhcpd_leases`
- » `/Library/Preferences/SystemConfiguration/com.apple.vmnet.plist`

Reset the `minishift`-specific IP database, make sure to remove the `minishift` entry section from the `dhcpd_leases` file, and reboot your system.

```
{
  ip_address=192.168.64.2
  hw_address=1,2:51:8:22:87:a6
  identifier=1,2:51:8:22:87:a6
  lease=0x585e6e70
  name=minishift
}
```



Note

You can completely reset the IP database by removing the files manually but this is very **risky**.

7.7. VIRTUALBOX: ERROR MACHINE DOES NOT EXIST

If you use Windows, make sure that you set the `--vm-driver virtualbox` flag in the `minishift start` command. Alternatively, the problem might be an outdated version of VirtualBox.

To avoid this issue, it is recommended to use VirtualBox 5.1.12 or later.