Red Hat Container Development Kit 2.1
Release Notes and Known Issues

Highlighted features and identified problems in Container Development Kit 2.1

Robert Krátký   Red Hat Developer Group Documentation Team

Red Hat Container Development Kit 2.1 Release Notes and Known Issues

Highlighted features and identified problems in Container Development Kit 2.1

Robert Krátký
rkratky@redhat.com

Legal Notice

Abstract

This document lists and briefly describes new and improved features of Red Hat Container Development Kit 2.1. It also contains information about potential problems you may encounter while using the software. Where possible, workarounds are described for identified issues.

Table of Contents

CHAPTER 1. INTRODUCING RED HAT CONTAINER DEVELOPMENT KIT

Red Hat Container Development Kit is a platform for developing containerized applications:

» Container Development Kit provides a personal Container Development Environment you can install on your own laptop, desktop, or server system. The Container Development Environment is provided in the form of a Red Hat Enterprise Linux virtual machine. The Container Development Environment itself can also be installed in a virtual machine.

» Container Development Kit includes the same container-development and run-time tools used to create and deploy containers for large data centers.

» Container Development Kit offers an easy installation method that results in virtual machines created from pre-configured Vagrant boxes and Vagrantfiles running on your local system.

1.1. UNDERSTANDING CONTAINER DEVELOPMENT KIT DOCUMENTATION

» The Red Hat Container Development Kit 2.1 Release Notes and Known Issues contains information about the current release of the product as well as a list of known problems that users may encounter when using it.

» The Red Hat Container Development Kit 2.1 Installation Guide guide contains instructions for installing the Container Development Environment provided by Red Hat Container Development Kit on your chosen system.

» The Container Development Kit Getting Started Guide contains instructions on how to start using the Container Development Environment to develop Red Hat Enterprise Linux-based containers using tools and services such as **OpenShift Enterprise**, **Docker**, **Eclipse**, and various command-line tools.

CHAPTER 2. RELEASE NOTES

This section documents the most important features and bug fixes in the Red Hat Container Development Kit 2.1 product.

2.1. COMPONENT VERSIONS

Red Hat Container Development Kit 2.1 is shipped with the following versions of the main components:

Table 2.1. Container Development Kit, Component Versions

Component	Version
Docker	1.9.1
Docker API	1.21
Kubernetes	1.2.0
OpenShift Enterprise	3.2

2.2. VAGRANT COMPATIBILITY MATRIX

The Container Development Environment provided by Container Development Kit is delivered in the form of a virtual machine that is administered by **Vagrant**. Not all versions of **Vagrant** work correctly with the Container Development Environment on all supported platforms. The following table shows version requirements for **Vagrant** based on the platform and hypervisor used.

Table 2.2. Vagrant Compatibility Matrix

	Vagrant version	
	1.7.4	1.8.1
Microsoft Windows with VirtualBox	✓	✓

Microsoft Windows with Hyper-V	✘	✓
Mac OS X with VirtualBox	✓	✓
Red Hat Enterprise Linux with libvirt	✓	✘

2.3. NEW FEATURES

This section highlights some of the new features offered by Red Hat Container Development Kit 2.1.

2.3.1. OpenShift Enterprise 3.2

OpenShift Enterprise provided by the Container Development Environment has been upgraded to version 3.2. See the OpenShift Enterprise 3.2 Release Notes for information about new features in this version.

Red Hat Container Development Kit 2.1 offers a simple way to set up the **OpenShift Enterprise** Platform as a Service (PaaS), which extends the functionality of the **Docker** service and the **Kubernetes** container orchestration tool to provide a powerful and easy-to-use platform for building, deploying, and orchestrating multi-container applications and services.

2.3.2. Hyper-V Support Added

A Container Development Kit Vagrant box is now available for the **Hyper-V** hypervisor on Microsoft Windows. Users can now set up Container Development Kit using the **Hyper-V** provider in Vagrant. **Hyper-V** is the native hypervisor in Microsoft Windows. Note that **Hyper-V** support is provided as *Technology Preview* in Red Hat Container Development Kit 2.1.

2.3.3. Reduced Size of Vagrant Boxes

In Red Hat Container Development Kit 2.1, the size of the Vagrant Boxes has been reduced by approximately 150 MB compared to Container Development Kit 2.0.

2.3.4. Persistent Volumes for Storing Data between Restarts

Users can now use persistent volumes in order to preserve data between restarts of pods, OpenShift, or even the whole virtual machine that provides the Container Development Environment.

2.3.5. OpenShift Registry Available on Host System

Users can now use the internal container-image registry provided by OpenShift as a stand-alone registry for docker-formatted container images on the host workstation.

2.4. BUG FIXES

This section highlights some of the important bugs fixed in Red Hat Container Development Kit 2.1.

2.4.1. `vagrant-service-manager` Plugin Reports Status of Services

Previously, the **`vagrant-service-manager`** plugin did report the status of services running within the Container Development Environment properly. With this update, a new command, **`status`**, has been added, which reports the status of required services. To use the new command, run:

```
$ vagrant service-manager status [service]
```

Substitute **`[service]`** for the service you wish to learn about: **`docker`**, **`kubernetes`**, or **`openshift`**. When used without any argument, the command lists all services and their status.

2.4.2. `vagrant-vbguest` Plugin Installs VirtualBox Guest Additions

Previously, a bug in the **`vagrant-registration`** plugin prevented the automatic installation of **VirtualBox Guest Additions**. The bug has been fixed, and the **`vagrant-vbguest`** plugin can now be used together with the **`vagrant-registration`** plugin to install **VirtualBox Guest Additions** in the Container Development Kit virtual machine.

2.4.3. Provisioning Runs with Every `vagrant up`

Previously, provisioning of Vagrant boxes ran only once — during the first launch of a Vagrant box using the **`vagrant up`** command after the last **`vagrant destroy`**, unless the **`--provision`** flag was used. With the latest Container Development Kit Vagrantfiles, provisioning runs with every **`vagrant up`**.

CHAPTER 3. KNOWN ISSUES

This section describes issues that users of Red Hat Container Development Kit 2.1 may encounter, as well as possible workarounds for these issues.

3.1. GENERAL ISSUES

Issues affecting all supported platforms.

3.1.1. Vagrantfiles That Set Up SSHFS Cannot Be Used with JBoss Developer Studio

Due to a known issue that causes **Red Hat JBoss Developer Studio** to freeze upon start up when it attempts to launch Container Development Kit with the configuration that sets up automatic directory sharing through SSHFS, the use of the `vagrant-sshfs` plugin is disabled by default in Red Hat Container Development Kit 2.1.

This problem does not concern any other uses than when Container Development Kit is launched from the graphical environment of Red Hat JBoss Developer Studio.

To use SSHFS when you launch Container Development Kit from the command line, you need to use the Vagrant files that are provided in the Red Hat Container Tools ZIP in the following subdirectory: `cdk/components/rhel/misc/shared_folder/`.

3.1.2. No support for network proxies

Internal network proxies, which are used for connecting to the Internet in some networking configurations, are not supported by Container Development Kit.

3.1.3. DNS service provided by `xip.io` only works with an Internet connection

DNS addressing provided by the external **xip.io** service is unavailable without a working Internet connection.

3.2. ISSUES ON MICROSOFT WINDOWS

This section describes issues that only affect users using Container Development Kit on a Microsoft Windows host system.

3.2.1. Hyper-V Vagrant Box Cannot Be Removed

Using the `vagrant box remove cdkv2` command fails to remove **Hyper-V** Vagrant boxes. To work around this issue when you wish to use a different **Hyper-V** box, you need to forcefully overwrite the existing box:

```
$ vagrant box add cdkv2 new-hyper-v-box.img
```

3.2.2. `vagrant-registration` Plugin on Cygwin Bash Does Not Hide Password

When the **vagrant-registration** plugin is used in interactive mode in the Cygwin Bash terminal, the entered password is visible in the standard input. The following warning will be printed:

```
Error! Your console doesn't support hiding input. We'll ask for
input again below, but we WILL NOT be able to hide input. If this
is a problem for you, ctrl-C to exit and fix your stdin.
```

3.3. ISSUES ON RED HAT ENTERPRISE LINUX

This section describes issues that only affect users using Container Development Kit on a Red Hat Enterprise Linux host system.

3.3.1. The `docker` command Cannot Communicate with Container Development Kit

The **/usr/bin/docker** wrapper on Red Hat Enterprise Linux 7 overwrites the **DOCKER_CERT_PATH** environment variable, which means that the Docker client does not work properly with the Docker service provided by the Container Development Environment. To work around this issue, use directly the **docker-current** command.

3.3.2. SSHFS Does Not Work when Vagrant Used under Root User

If the **vagrant up** command is run under the **root** user, the **vagrant-sshfs** plugin fails to synchronize the user's home directory (**/root**) because the **/root/** directory in the Container Development Environment is not empty, and thus **SSHFS** refuses to use it. To work around this issue use Vagrant as a regular, non-root user.

3.3.3. libvirt Fails to Unlink Virtual Machine Image

On Red Hat Enterprise Linux 7.2, the **vagrant destroy** command does not properly delete the virtually machine storage image. To work around this **libvirt** bug, you need to change the ownership of the image file and perform several steps using the **virsh** command.

Deleting libvirt Images

Substitute ***vagrant_box_image.img*** with the **libvirt** image file name used on your system in all steps of this procedure. Use the **virsh vol-list default** command to list all images.

1. Change the ownership of the image file:

   ```
   # chown root:root /var/lib/libvirt/images/vagrant_box_image.img
   ```

2. Output volume information:

   ```
   # virsh vol-dumpxml vagrant_box_image.img --pool default
   ```

3. Delete the image:

   This can be done in two alternative ways. Use the second one if the first one does not work.

   - Use the **vol-delete** command with **virsh**:

```
# virsh vol-delete vagrant_box_image.img --pool default
```

» Alternatively, delete the file directly and refresh the volume pool:

```
# rm /var/lib/libvirt/images/vagrant_box_image.img
# virsh pool-refresh default
```

4. Restartd the **libvirtd** daemon:

```
# systemctl restart libvirtd
```

CHAPTER 4. ADDITIONAL RESOURCES

≫ See the Red Hat Container Development Kit 2.1 Installation Guide for installation instructions for all supported platforms (Linux, Mac OS X, Windows).

≫ See the Troubleshooting chapter of the Red Hat Container Development Kit 2.1 Installation Guide for additional information about uninstallation issues.

≫ See the Container Development Kit Getting Started Guide for an overview of Container Development Kit features and for an introduction into the use of the **Docker** service and the **OpenShift Enterprise** platform.