OpenShift Container Platform 3.11

Container-native Virtualization Install

Container-native Virtualization Installation

Last Updated: 2019-08-02

OpenShift Container Platform 3.11 Container-native Virtualization Install

Container-native Virtualization Installation

Legal Notice

Abstract

Learn to install CNV

Table of Contents

CHAPTER 1. INSTALLING CONTAINER-NATIVE VIRTUALIZATION

1.1. PRODUCT OVERVIEW

1.1.1. Introduction to Container-native Virtualization

Container-native Virtualization is an add-on to OpenShift Container Platform that allows virtual machine workloads to run and be managed alongside container workloads. You can create virtual machines from disk images imported using the containerized data importer (CDI) controller, or from scratch within OpenShift Container Platform.

Container-native Virtualization introduces two new objects to OpenShift Container Platform:

- **Virtual Machine**: The virtual machine in OpenShift Container Platform

- **Virtual Machine Instance**: A running instance of the virtual machine

With the Container-native Virtualization add-on, virtual machines run in pods and have the same network and storage capabilities as standard pods.

Existing virtual machine disks are imported into persistent volumes (PVs), which are made accessible to Container-native Virtualization virtual machines using persistent volume claims (PVCs). In OpenShift Container Platform, the virtual machine object can be modified or replaced as needed, without affecting the persistent data stored on the PV.

> **IMPORTANT**
>
> Container-native Virtualization is currently a Technology Preview feature. For details about Red Hat support for Container-native Virtualization, see the Container-native Virtualization - Technology Preview Support Policy.
>
> Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat does not recommend to use them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.
>
> For more information on Red Hat Technology Preview features support scope, see https://access.redhat.com/support/offerings/techpreview/.

1.2. PREREQUISITES

Container-native Virtualization requires an existing OpenShift Container Platform cluster with the following configuration considerations:

1.2.1. Node configuration

See the OpenShift Container Platform Installing Clusters Guide for planning considerations for different cluster configurations.

Binary builds and MiniShift are not supported with Container-native Virtualization.

1.2.2. Admission control webhooks

Container-native Virtualization implements an admission controller as a webhook so that Container-native Virtualization-specific creation requests are forwarded to the webhook for validation. Registering webhooks must be enabled during installation of the OpenShift Container Platform cluster.

To register the admission controller webhook, add the following under the **[OSEv3:vars]** section in your Ansible inventory file during OpenShift Container Platform deployment:

> openshift_master_admission_plugin_config={"ValidatingAdmissionWebhook":{"configuration":{"kind":
> "DefaultAdmissionConfig","apiVersion": "v1","disable": false}},"MutatingAdmissionWebhook":
> {"configuration":{"kind": "DefaultAdmissionConfig","apiVersion": "v1","disable": false}}}

1.2.3. CRI-O runtime

CRI-O is the required container runtime for use with Container-native Virtualization.

See the OpenShift Container Platform 3.11 CRI-O Runtime Documentation for more information on using CRI-O.

1.2.4. Storage

Container-native Virtualization supports local volumes, block volumes, and Red Hat OpenShift Container Storage as storage backends.

1.2.4.1. Local volumes

Local volumes are PVs that represent locally-mounted file systems. See the OpenShift Container Platform Configuring Clusters Guide for more information.

1.2.4.2. Block volumes

Container-native Virtualization supports the use of block volume PVCs. In order to use block volumes, the OpenShift Container Platform cluster must be configured with the **BlockVolume** feature gate enabled. See the OpenShift Container Platform Architecture Guide for more information.

> IMPORTANT
>
> Local and block volumes both have limited support in OpenShift Container Platform 3.11 because they are currently Technology Preview. This may change in a future release.
>
> Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat does not recommend to use them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.
>
> For more information on Red Hat Technology Preview features support scope, see https://access.redhat.com/support/offerings/techpreview/.

1.2.4.3. Red Hat OpenShift Container Storage

Red Hat OpenShift Container Storage uses Red Hat Gluster Storage to provide persistent storage and dynamic provisioning. It can be used containerized within OpenShift Container Platform (converged mode) and non-containerized on its own nodes (independent mode).

See the OpenShift Container Storage Product Documentation or the OpenShift Container Platform Installing Clusters Guide for more information.

> **IMPORTANT**
>
> Requires Red Hat OpenShift Container Storage version 3.11.1 or later. Earlier versions of Red Hat OpenShift Container Storage do not support CRI-O, the required container runtime for Container-native Virtualization.

1.2.5. Metrics

Metrics are not required, but are a recommended addition to your OpenShift Container Platform cluster because they provide additional information about Container-native Virtualization resources.

See the OpenShift Container Platform Installing Clusters Guide for comprehensive information on deploying metrics in your cluster.

1.3. INSTALLING CONTAINER-NATIVE VIRTUALIZATION

1.3.1. Enabling Container-native Virtualization repository

You must enable the **rhel-7-server-cnv-1.4-tech-preview-rpms** repository for the master to install the Container-native Virtualization packages.

Prerequisites

- Register the host and attach the OpenShift Container Platform subscription.

Procedure

- Enable the repository:

```
$ subscription-manager repos --enable=rhel-7-server-cnv-1.4-tech-preview-rpms
```

1.3.2. Installing virtctl client utility

The **virtctl** client utility is used to manage the state of the virtual machine, forward ports from the virtual machine pod to the node, and open console access to the virtual machine.

Procedure

1. Install the **kubevirt-virtctl** package:

   ```
   $ yum install kubevirt-virtctl
   ```

> **NOTE**
>
> The **virtctl** utility is also available for download from the Red Hat Network.

1.3.3. Installing Container-native Virtualization to OpenShift Container Platform

The **kubevirt-ansible** RPM contains the latest automation for deploying Container-native Virtualization. This procedure installs all Container-native Virtualization components to your OpenShift Container Platform cluster.

This procedure installs the following components:

- Container-native Virtualization core components (KubeVirt)

- Containerized data importer (CDI) controller

- Multus, Open vSwitch (OVS), and SR-IOV container network interface plug-ins

- Updated Container-native Virtualization web console

Prerequisites

- A running OpenShift Container Platform 3.11 cluster

- User with **cluster-admin** privileges

- **rhel-7-server-cnv-1.4-tech-preview-rpms** must be enabled

- Ansible inventory file

> **NOTE**
>
> See the Reference section of this guide for an example inventory file that can be modified to match your configuration.

Procedure

1. Install the **kubevirt-ansible** RPM and its dependencies:

   ```
   $ yum install kubevirt-ansible
   ```

2. Log into the OpenShift Container Platform cluster as an admin user:

   ```
   $ oc login -u system:admin
   ```

3. Change directories to **/usr/share/ansible/kubevirt-ansible**:

   ```
   $ cd /usr/share/ansible/kubevirt-ansible
   ```

4. Launch Container-native Virtualization:

   > **NOTE**
   >
   > To deploy Container-native Virtualization from a custom repository, add **-e registry_url=registry.example.com** to the **ansible-playbook** command below. To set a local repository tag, add **-e cnv_repo_tag=local-repo-tag-for-cnv** to the command.

```
$ ansible-playbook -i <inventory_file> -e @vars/cnv.yml playbooks/kubevirt.yml \
-e apb_action=provision
```

5. Verify the installation by navigating to the web console at **kubevirt-web-ui.your.app.subdomain.host.com**. Log in by using your OpenShift Container Platform credentials.

1.4. UNINSTALLING CONTAINER-NATIVE VIRTUALIZATION

1.4.1. Uninstalling Container-native Virtualization

You can uninstall Container-native Virtualization with the same **ansible-playbook** command you used for deployment if you change the **apb_action** parameter value to **deprovision**.

This procedure uninstalls the following components:

- Container-native Virtualization core components (KubeVirt)

- Containerized data importer (CDI) controller

- Multus, Open vSwitch (OVS), and SR-IOV container network interface plug-ins

- Container-native Virtualization web console

Prerequisites

- Container-native Virtualization 1.4

Procedure

1. Log into the OpenShift Container Platform cluster as an admin user:

   ```
   $ oc login -u system:admin
   ```

2. Change directories to **/usr/share/ansible/kubevirt-ansible**:

   ```
   $ cd /usr/share/ansible/kubevirt-ansible
   ```

3. Uninstall Container-native Virtualization:

   ```
   $ ansible-playbook -i <inventory_file> -e @vars/cnv.yml playbooks/kubevirt.yml \
   -e apb_action=deprovision
   ```

4. Remove Container-native Virtualization packages:

   ```
   $ yum remove kubevirt-ansible kubevirt-virtctl
   ```

5. Disable the Container-native Virtualization repository:

   ```
   $ subscription-manager repos --disable=rhel-7-server-cnv-1.4-tech-preview-rpms
   ```

6. To verify the uninstallation, check to ensure that no KubeVirt pods remain:

```
$ oc get pods --all-namespaces
```

1.5. REFERENCE

1.5.1. OpenShift Container Platform example inventory file

You can use this example to see how to modify your own Ansible inventory file to match your cluster configuration.

> **NOTE**
>
> In this example, the cluster has a single master that is also an infra node, and there are two separate compute nodes.

```
[OSEv3:children]
masters
nodes
etcd

[OSEv3:vars]
openshift_deployment_type=openshift-enterprise
ansible_ssh_user=root
ansible_service_broker_registry_whitelist=['.*-apb$']
ansible_service_broker_local_registry_whitelist=['.*-apb$']

# Enable admission controller webhooks
openshift_master_admission_plugin_config={"ValidatingAdmissionWebhook":{"configuration":{"kind":
"DefaultAdmissionConfig","apiVersion": "v1","disable": false}},"MutatingAdmissionWebhook":
{"configuration":{"kind": "DefaultAdmissionConfig","apiVersion": "v1","disable": false}}}

# CRI-O
openshift_use_crio=true

# Provide your credentials to consume the redhat.io registry
oreg_auth_user=$rhnuser
oreg_auth_password='$rhnpassword'

# Host groups
[masters]
master.example.com

[etcd]
master.example.com

[nodes]
master.example.com openshift_node_group_name='node-config-master-infra-crio'
node1.example.com openshift_node_group_name='node-config-compute-crio'
node2.example.com openshift_node_group_name='node-config-compute-crio'
```