HPE Nimble Storage Integration Guide for Red Hat OpenShift and OpenShift Origin

Contents

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Intel®, Itanium®, Pentium®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the United States and other countries.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated. Java® and Oracle® are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Publication Date

Monday July 15, 2019 12:01:26

Document ID

voz1522429958863

Support

All documentation and knowledge base articles are available on HPE InfoSight at *https://infosight.hpe.com*. To register for HPE InfoSight, click the *Create Account* link on the main page.

Email: *support@nimblestorage.com*

For all other general support contact information, go to *https://www.nimblestorage.com/customer-support/*.

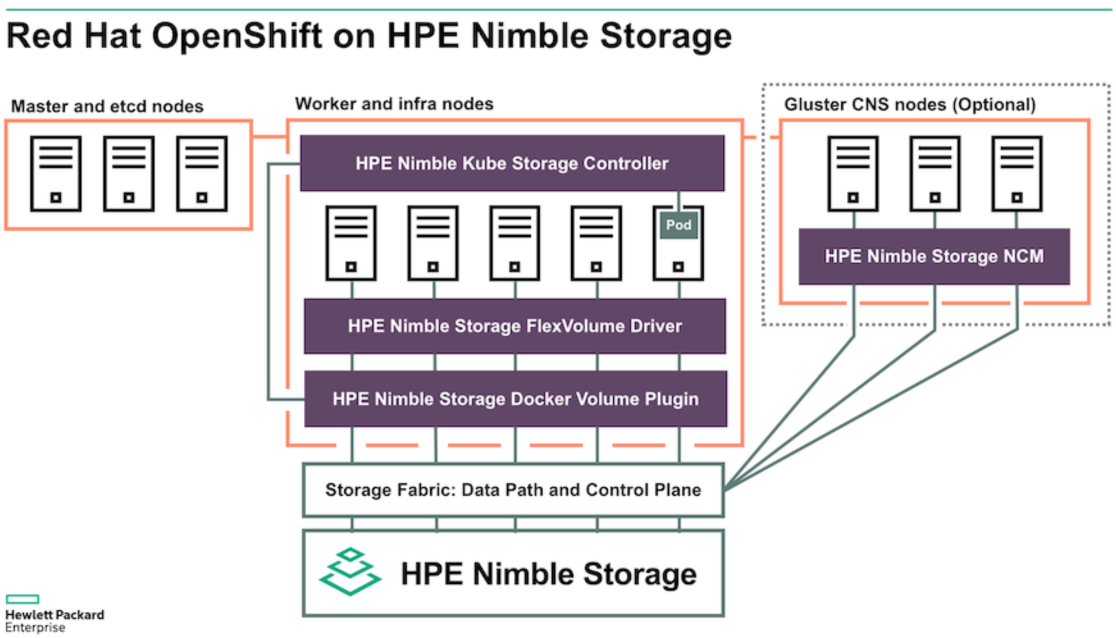
Overview

This guide describes the detailed steps for integrating HPE Nimble Storage arrays with the Red Hat® OpenShift® platform to enable users to provision Kubernetes® PersistentVolumeClaims (PVCs) from a defined StorageClass set up by an OpenShift administrator. The HPE Nimble Kube Storage Controller provisions PersistentVolumes (PVs) based on the StorageClass parameters and the size specified in the PVC.

Architecture

The following architecture diagram illustrates a potential deployment scenario with OpenShift using HPE Nimble Storage arrays.

Figure 1: Example deployment of OpenShift on HPE Nimble Storage



Prerequisites

Each of the following sections details the requirements dictated by each component that would comprise a supported environment.

Supported versions of NimbleOS

Hewlett Packard Enterprise always recommends using the latest GA (Generally Available) release of NimbleOS for production workloads:

- NimbleOS version 5.0.7.200 or later

Supported host operating systems

The HPE Nimble Kube Storage Controller requires the HPE Nimble Storage Linux Toolkit to be installed on each of the nodes in the OpenShift cluster that is schedulable for workloads. The following list correlates OpenShift requirements with HPE Nimble Storage Linux Toolkit requirements:

- Red Hat Enterprise Linux® 7.3, 7.4, 7.5, or 7.6 for Red Hat OpenShift
- CentOS 7.3, 7.4, 7.5, or 7.6 for OpenShift Origin

Red Hat CoreOS, Project Atomic, and Fedora are not currently supported.

Supported versions of NLT

Currently supported versions of the HPE Nimble Storage Linux Toolkit:

- NLT-2.5.1

Note Although there are newer major versions of NLT available, such as NLT-3.0, the 2.5.x line of releases is the only supported version and integration for Red Hat OpenShift and OpenShift Origin 3.x.

Supported versions of OpenShift

Tested and validated versions of Red Hat OpenShift and OpenShift Origin:

- Red Hat OpenShift 3.5
- Red Hat OpenShift 3.6
- Red Hat OpenShift 3.7
- Red Hat OpenShift 3.9
- Red Hat OpenShift 3.10
- Red Hat OpenShift 3.11
- OpenShift Origin (Origin Kubernetes Distribution) 3.5 to latest

From OpenShift version 3.9, Red Hat introduced CRI-O into OpenShift, a Kubernetes optimized container runtime. HPE Nimble Storage supports both Docker and CRI-O because the implementation is container runtime agnostic. CRI-O is an opt-in container runtime that must be selected at OpenShift installation.

Note OpenShift Origin is supported on a best-effort basis because it is a community project.

Install HPE Nimble Storage Linux Toolkit

You can add the HPE Nimble Storage OpenShift integration to an OpenShift cluster at any time in the lifecycle of the cluster. The following steps are a simplified version of the installation procedure of the HPE Nimble Storage Linux Toolkit (NLT), optimized for OpenShift. For a comprehensive guide, see the *Linux Integration Guide* on *HPE InfoSight*.

OpenShift 3.7 and earlier versions require NLT to be installed on every node in the cluster. For OpenShift 3.9 and later versions, only the compute nodes require NLT to be installed.

Procedure

1. Install the required host utilities from the host OS official repositories:

   ```
   $ sudo yum install -y device-mapper-multipath iscsi-initiator-utils sg3_utils
   ```

   Note For Fibre Channel hosts, the `iscsi-initiator-utils` package is not required.

2. Register for an account on *HPE InfoSight*, or log in if an account has already been established.
3. Click **Resources** > **Software Downloads**.
4. In the Integration Kits pane, click **HPE Nimble Storage Linux Toolkit (NLT)**.

   Note Use the **Anonymous Download Form** if automated installation is required directly from the OpenShift host.

5. From the HPE Nimble Storage Linux Toolkit (NLT) page, click **NLT Installer** under **Current Version**. When prompted, save the installer to your local host.
6. Transfer the file from the download location to `/tmp` on the OpenShift host.
7. Make sure the file is executable and run the installer:

   ```
   $ sudo /tmp/nlt_installer_<VERSION> --docker --flexvolume --silent-mode
   ```

   Note For Fibre Channel hosts, add `--disable-iscsi` to the list of options.

8. Accept the EULA. For unattended installations, add the `--accept-eula` parameter for a noninteractive installer.
9. Add an array group to NLT to enable the driver to provision volumes:

   ```
   $ sudo nltadm --group --add --ip-address <mgmt IP address> --username <user>
   ```

10. Validate communication with the HPE Nimble Storage array:

    ```
    $ sudo nltadm --group --verify --ip-address <management IP address>
    ```

11. Verify that the `Docker-Driver` has a status of `RUNNING`:

    ```
    $ sudo nltadm --status
    ```

12. Verify that the host can provision volumes:

    ```
    $ sudo docker volume create -d nimble mytestvol
    $ sudo docker volume inspect mytestvol
    $ sudo docker volume rm mytestvol
    ```

13. Repeat steps 6 through 12 for each node in the OpenShift cluster that requires NLT, as described before step 1.

For information about using Red Hat Ansible® software to automate the installation of the HPE Nimble Storage Linux Toolkit, see the *Ansinimble role on Ansible Galaxy*.

Note If automated installation is used for multiple nodes, there is a known NimbleOS 5.0.4 issue: NLT must be added to OpenShift hosts in a single-threaded process and in sequence to prevent a threading problem with the trust store. For more information, see the *NLT Release Notes* on *HPE InfoSight*.

Red Hat Container-Native Storage considerations

HPE Nimble Storage arrays can coexist with Red Hat Container-Native Storage (CNS). Red Hat CNS is backed by Red Hat Gluster Storage, and it is a great complement to HPE Nimble Storage because Red Hat CNS provides RWX (ReadWriteMany) access mode in addition to the RWO (ReadWriteOnce) provided by HPE Nimble Storage.

Note Red Hat CNS is required with HPE Nimble Storage only when Pods across OpenShift nodes require RWX access mode. Multiple containers within the same Pod on the same host require only RWO access mode, which HPE Nimble Storage provides natively.

To optimize the coexistence of Red Hat CNS and HPE Nimble Storage, consider the following points during OpenShift deployment:

- HPE does not support the use of Red Hat CNS to provide anything except the filesystem component in coexistence with HPE Nimble Storage.
- If any Red Hat CNS component is selected during OpenShift deployment, the `/etc/multipath.conf` file is overwritten with an incompatible version of HPE Nimble Storage. HPE recommends installing NLT after OpenShift installation is complete. Also, be vigilant during errata updates because the `/etc/multipath.conf` might be overwritten and might require an NLT reinstallation.
- The HPE Nimble Storage Connection Manager (NCM) component must be selected during NLT installation by providing the `--ncm` option to the NLT installer.
- To enable you to take full advantage of the joint benefits of Red Hat CNS and HPE Nimble Storage, the block devices that are consumed by Red Hat CNS must be provided by HPE Nimble Storage.

Note Red Hat CNS and HPE Nimble Storage implementations provide very different data services. Explaining the differences in features and capabilities between the two (besides the access mode) is beyond the scope of this guide.

The following excerpt from an `openshift-ansible` inventory file provides an example of the parameters needed for a successful deployment of Red Hat CNS on HPE Nimble Storage arrays:

```
[OSEv3:children]
glusterfs

[OSEv3:vars]
openshift_storage_glusterfs_block_deploy=false
openshift_storage_glusterfs_s3_deploy=false
openshift_storage_glusterfs_storageclass=true
openshift_storage_glusterfs_storageclass_default=false

[glusterfs]
tme-lnx1-ocp-prod.lab.nimblestorage.com
glusterfs_devices='["/dev/nimblestorage/brick1-25624d77ee9bdd77f6c9ce9004d4e9802"]'
tme-lnx2-ocp-prod.lab.nimblestorage.com
glusterfs_devices='["/dev/nimblestorage/brick2-2cb44ca5e877eb7bf6c9ce9004d4e9802"]'
tme-lnx3-ocp-prod.lab.nimblestorage.com
glusterfs_devices='["/dev/nimblestorage/brick3-23809ef88f27d21496c9ce9004d4e9802"]'
```

The `/dev/nimblestorage` devices are created for any HPE Nimble Storage volume that is exposed to the host by the udev rules provided by NCM. It is possible to use the HPE Nimble Storage Ansible role to create and map these block devices to the OpenShift hosts by using the `volume create` and `volume map` tasks. Ansinimble, the HPE Nimble Storage Ansible role, can be found on *Ansible Galaxy*.

Node isolation considerations

To accommodate a few common failure scenarios, it is important to tweak the compute node behavior when a hung task is detected. If a compute node becomes isolated from the cluster, its workloads are rescheduled elsewhere. When you use persistent storage from HPE Nimble Storage for stateful applications, it is imperative to understand the fencing mechanisms that exist to protect the data in the event of a compute node failure or a network partition.

Persistent Volumes from HPE Nimble Storage can be accessed from only one compute node at a time. This rule exists because the volumes are formatted with XFS (or `ext4` in some cases), which is not a clustered filesystem. It can be corrupted if the same volume is mounted on multiple compute nodes. The container orchestrator is not affected by the underlying intricacies of SAN storage or single-host access filesystems. When a mount request is received from a container orchestrator, the requested volume's access control records (ACRs) are stripped and replaced with the initator information from the location where the request originated. This process makes the data accessible from only one compute node at a time. HPE fully supports allowing the container orchestrator to reuse the mount point for many containers on the same compute node. However, you should understand the application to ensure that it supports a same-filesystem, multi-writer paradigm.

Volume fencing might cause inconveniences if a compute node becomes isolated. When the compute node later rejoins the cluster, it might be after the workloads that were previously running on the compute node have been rescheduled and are now unable to perform any I/O, including unmounting filesystems. In this situation, I/O operations would hang indefinitely and might under certain circumstances cause problems for future workloads because, in most cases, the compute node would look healthy from the container orchestrator's perspective.

HPE strongly recommends that you configure the compute node to panic if it detects a hung task. Take this precaution to ensure that the compute node comes up healthy without lingering I/O or residual data paths to unmapped devices.

The following kernel parameters control the compute node behavior when a hung task is detected:

```
# Reset after these many seconds after a panic
kernel.panic = 5

# Consider hung tasks reason enough to panic
kernel.hung_task_panic = 1

# Wait these many seconds before declaring a hung task
kernel.hung_task_timeout_secs = 150
```

Add these parameters to an `/etc/sysctl.d/99-hung_task_timeout.conf` file on all compute nodes in the cluster by using persistent storage, and then reboot the node.

Container storage

HPE highly recommends that you relocate `/var` to a separate filesystem that is provided by HPE Nimble Storage. Because of the many dependencies of binds and overlays into `/var`, HPE does not recommend relocating data postinstallation. Relocate `/var` before installing OpenShift.

The topics of creating, mapping, and discovering a LUN from an HPE Nimble Storage array are beyond the scope of this integration guide. For assistance, consult with a storage administrator.

The Red Hat OpenShift documentation provides system requirements that include `/var` sizing recommendations.

Note HPE recommends that you relocate the entire `/var` filesystem to an HPE Nimble Storage device, not just the `/var/lib/containers` (CRI-O) or `/var/lib/docker` (Docker) components.

Prepare OpenShift

OpenShift 3.7 and earlier versions

The HPE Nimble Kube Storage Controller is dependent on the `hostPath` plugin and the ability to run the container as `root`, which is accomplished by adding the default user in the kube-system `namespace` to the privileged Security Context Constraint:

```
$ oadm policy add-scc-to-user privileged \
system:serviceaccount:kube-system:default
```

For OpenShift based on Kubernetes versions earlier than 1.8, the HPE Nimble Kube Storage Controller must run on a master. It is therefore important to have a labelling and tainting scheme in place and to enable scheduling on the masters:

```
$ oc label node master.example.com node-role.kubernetes.io/master="true"
$ oc adm taint nodes master.example.com \
node-role.kubernetes.io/master=:NoSchedule
$ oadm manage-node --schedulable master.example.com
```

Note The labeling and tainting scheme allow pods to run on the masters, provided that the pods have the proper tolerations set.

OpenShift 3.9 and later versions

The HPE Nimble Kube Storage Controller is dependent on the `hostPath` plugin and the ability to run as `root`. For OpenShift based on Kubernetes 1.8 and later versions, the HPE Nimble Kube Storage Controller relies on its own service account with a scoped role using RBAC. It can also run on any node in the cluster. The service account requires access to the `hostPath` plugin and the ability to run the container as `root`:

```
$ oc adm policy add-scc-to-user privileged \
system:serviceaccount:kube-system:doryd
```

For fresh installations of OpenShift 3.9 or later versions, the kube-system `namespace` is affected by a default `nodeSelector`, and labeling the worker nodes might be required:

```
$ oc label node workernode.example.com node-role.kubernetes.io/compute=true
```


Deploy HPE Nimble Kube Storage Controller

For the steps that are specific to OpenShift, a user with the `system:admin` role must be logged in on a master node. Run the following command to verify the role and credentials:

```
$ oc whoami -c
default/yourmasternodename.example.com:8443/system:admin
```

Deployment and example specification files are hosted on GitHub in the Container Examples repository for HPE Nimble Storage. The rest of the guide assumes that the Container Examples repository has been checked out and the current working directory matches the OpenShift version being installed to:

```
$ git clone https://github.com/nimblestorage/container-examples
$ cd container-examples/NLT/OpenShift/VERSION
```

In place of `VERSION` in the second line, substitute the string that corresponds to the particular OpenShift version being installed; for example, `ocp35` for Red Hat OpenShift 3.5 or `org37` for OpenShift Origin 3.7.

Use the following command to deploy the HPE Nimble Kube Storage Controller:

```
$ oc create -f dep-kube-storage-controller.yaml
```

Pull from Red Hat Container Catalog

A Red Hat certified image of the HPE Nimble Kube Storage Controller is available in the Red Hat Container Catalog (RHCC). Pulling images from RHCC requires authentication and a Red Hat login obtained from the *Red Hat Customer Portal*. For an explanation of how to allow pods to reference images from secured registries, see the *OpenShift documentation*.

Patch the HPE Nimble Kube Storage Controller deployment to change from the default Docker Hub image to the RHCC image (after setting up RHCC authentication):

```
$ oc -n kube-system set image deploy/kube-storage-controller-doryd \
  kube-storage-controller=nimble/kube-storage-controller:2.4.1
deployment.apps "kube-storage-controller-doryd" image updated
```

Note The only difference in the image name is the repository portion of the name. Any image pull error, such as `ImagePullBackOff`, that the pod might experience is due to authentication or connectivity issues with RHCC.

Workaround for OpenShift version 3.6

For Red Hat OpenShift and OpenShift Origin based on Kubernetes version 1.6, the FlexVolume driver must be tuned to accommodate a certain behavior.

Copy the configuration file and restart the node service:

```
$ sudo cp nimble.json \
/usr/libexec/kubernetes/kubelet-plugins/volume/exec/hpe.com~nimble/nimble.json
```

On Red Hat OpenShift, use the following command to restart the node service:

```
$ sudo systemctl restart atomic-openshift-node
```

On OpenShift Origin, use the following command to restart the node service:

```
$ sudo systemctl restart origin-node
```

Note The FlexVolume driver must be tuned on all nodes in the cluster.

OpenShift 3.7 and earlier versions

Dynamic discovery of FlexVolume drivers was introduced in Kubernetes version 1.8, but it was not included in OpenShift until version 3.9. Therefore, after installing the FlexVolume driver with NLT and deploying the HPE Nimble Kube Storge Controller, it is necessary to restart the OpenShift node service:

```
$ sudo systemctl restart atomic-openshift-node
```

Note The OpenShift node service must be restarted on any OpenShift host that is subject to attaching persistent storage.

Validate installation of the HPE Nimble Kube Storage Controller

After the image has been pulled and the deployment has been scheduled, run the folowing command and compare the output:

```
$ oc get deploy --namespace kube-system
NAME                          DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
kube-storage-controller-doryd 1       1       1          1         3s
```

SELinux and OpenShift Origin

If you run OpenShift Origin on anything besides Red Hat Enterprise Linux and use SELinux, the HPE Nimble Kube Storage Controller might experience `permission denied` issues while trying to communicate with the HPE Nimble Storage Docker Volume plugin through `/run/docker/plugins/nimble.sock`. There are multiple ways to get around this problem, but the safest practice is to create a separate SELinux policy package. The type of enforcement is included in the repository.

Build and load the module as follows:

```
$ checkmodule -M -m -o dory.mod dory.te
$ semodule_package -o dory.pp -m dory.mod
$ sudo semodule -i dory.pp
```

Note A separate SELinux policy package must be created on every node that the HPE Nimble Kube Storage Controller is allowed to run on.

Optional steps

In single-tenant build environments that are used for development and testing, it might be practical to allow end users direct access to the FlexVolume interface to enable fine-grained control of storage resources. This degree of control is easiest to achieve by editing the restricted SCC manually and adding `flexVolume` at the end of the list of volumes:

```
$ oc edit scc restricted
```


After the change is applied, end users can create inline FlexVolumes within a Pod specification. For an example, see `pod-inline-mariadb.yaml` in the checked-out repository.

Note Users cannot delete volumes while interacting with the FlexVolume driver inline because volumes are not managed by the HPE Nimble Kube Storage Controller.

OpenShift Container Registry

HPE recommends that you run the Red Hat OpenShift Container Registry (OCR) on a persistent volume backed by HPE Nimble Storage using a StorageClass with a protection template. For an explanation of how to create a StorageClass, see *Example usage* on page 17.

The OCR requires no special care during OpenShift installation because the built-in registry is deployed by default.

Note The following instructions are provided for OpenShift 3.10 and later versions. For earlier versions of OpenShift, these steps are provided as guidance.

Procedure

1 Make sure that only one replica of the ReplicationController is running.

OCR is deployed as a ReplicationController with the help of a DeploymentConfig object. HPE Nimble Storage supports the use of only a single replica.

```
$ oc patch -n default deploymentconfig/docker-registry \
  -p '{"spec":{"replicas": 1}}'
deploymentconfig.apps.openshift.io "docker-registry" patched
```

2 Set the `mountConflictDelay` parameter.

The repository provides an example StorageClass that is optimized for the registry. The most important parameter in the StorageClass is the `mountConflictDelay` parameter. It protects your data from Kubernetes because the ReplicationController does not support the Recreate update strategy. Setting a long `mountConflictDelay` timeout ensures that the volume properly is unmounted if the registry is updated or relocated. If the node that hosts the registry crashes, the full timeout is honored before the volume is mounted at the rescheduled location.

```
$ oc create -f sc-registry.yml
storageclass.storage.k8s.io "registry" created
```

3 Modify the PVC to fit the requirements of the deployment.

The example creates a 100GiB persistent volume.

```
$ oc create -f pvc-registry.yml
persistentvolumeclaim "registry" created
```

4 Use the following command to attach the new PVC to the registry:

```
$ oc volume deploymentconfigs/docker-registry --add --name=registry-storage
\
  -t pvc --claim-name=registry --overwrite
deploymentconfig "docker-registry" updated
```

5 Monitor the reconfiguration:

```
$ oc get replicationcontroller -w
```

6 Verify that a new revision of the docker-registry is running using the newly created PVC.

Documentation Feedback

Example usage

The following examples are provided to validate the installation; this is not an exhaustive list of examples.

There are a couple of YAML files in the repository to create a StorageClass, PVCs, and an example Deployment. Pay close attention to the different API versions and annotations that are needed for the different versions of OpenShift.

Create an application optimized StorageClass:

```
$ oc create -f sc-transactionaldb.yaml
storageclass "transactionaldb" created
```

Create a PVC from the StorageClass:

```
$ oc create -f pvc-mariadb.yaml
persistentvolumeclaim "mariadb-claim" created
```

Create a Deployment with a PVC reference:

```
$ oc create -f dep-mariadb.yaml
secret "mariadb" created
deployment "mariadb" created
service "mariadb" created
```

Create a default StorageClass for “classless” PVCs:

```
$ oc create -f sc-default.yaml
storageclass "general" created
```

Create a PVC without a StorageClass:

```
$ oc create -f pvc-default.yaml
persistentvolumeclaim "default-claim" created
```

Observe the created resources:

```
$ oc get deploy,storageclass,pvc,pv
NAME             DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
deploy/mariadb   1         1         1            1           45s
NAME                TYPE
general (default)   hpe.com/nimble
transactionaldb     hpe.com/nimble
NAME                STATUS    VOLUME          CAPACITY   ACCESSMODES   AGE
pvc/default-claim   Bound     general         32Gi       RWO           12s
pvc/mariadb-claim   Bound     transactionaldb 16Gi       RWO           1m
NAME               CAPACITY ACCESS RECLAIMPOLICY STATUS CLAIM         AGE
pv/general         32Gi     RWO    Delete        Bound  default-claim 9s
pv/transactionaldb 16Gi     RWO    Delete        Bound  mariadb-claim 1m
```


Supported StorageClass parameters

The HPE Nimble Kube Storage Controller and the FlexVolume driver support all of the parameters that the underlying HPE Nimble Storage Docker Volume plugin supports. The following list shows NLT-2.3.1 parameters:

```
$ docker volume create -d nimble -o help
Nimble Storage Docker Volume Driver: Create Help
Create or Clone a Nimble Storage backed Docker Volume or Import an existing
 Nimble Volume or Clone of a Snapshot into Docker.

Universal options:
-o mountConflictDelay=X X is the number of seconds to delay a mount request
                        when there is a conflict (default is 0)
-o description=X        X is the text to set on volume description (optional)


Create options:
-o sizeInGiB=X  X is the size of volume specified in GiB
-o size=X       X is the size of volume specified in GiB (short form of
                sizeInGiB)
-o fsOwner=X    X is the user id and group id that should own the root
                directory of the filesystem, in the form of [userId:groupId]
-o fsMode=X     X is 1 to 4 octal digits that represent the file mode to be
                applied to the root directory of the filesystem
-o perfPolicy=X X is the name of the performance policy (optional)
                Performance Policies: Exchange 2003 data store,
                Exchange 2007 data store, Exchange log, SQL Server,
                SharePoint, Exchange 2010 data store, DockerDefault,
                SQL Server Logs, SQL Server 2012, Oracle OLTP,
                Windows File Server, Other Workloads, Postgres, MariaDB,
                General, Backup Repository, Veeam Backup Repository
-o pool=X       X is the name of pool in which to place the volume (optional)
-o folder=X     X is the folder in which to place the volume (optional)
-o encryption   indicates that the volume should be encrypted (optional,
                dedupe and encryption are mutually exclusive)
-o thick        indicates that the volume should be thick provisioned
                (optional, dedupe and thick are mutually exclusive)
-o dedupe       indicates that the volume should be deduplicated
-o limitIOPS=X  X is the IOPS limit of the volume. IOPS limit should be in
                range [256, 4294967294] or -1 for unlimited.
-o limitMBPS=X  X is the MB/s throughput limit for this volume. If both
                limitIOPS and limitMBPS are specified, limitMBPS must not be
                hit before limitIOPS
-o destroyOnRm  flags the Nimble volume (including snapshots) backing this
                volume should be destroyed when this volume is deleted

Data Protection Options:
-o protectionTemplate=X X is the name of the protection template (optional)
                        Protection Templates: Retain-30Daily, General,
                        Retain-48Hourly-30Daily-52Weekly, Retain-90Daily,
                        Retain384-5min

Clone options:
-o cloneOf=X       X is the name of Docker Volume to create a clone of
-o snapshot=X      X is the name of the snapshot to base the clone on
                   (optional, if missing, a new snapshot is created)
-o createSnapshot  indicates that a new snapshot of the volume should be
                   taken and used for the clone (optional)
-o destroyOnRm     indicates that the Nimble volume (including snapshots)
                   backing this volume should be destroyed when this volume
                   is deleted
```

```
-o destroyOnDetach indicates that the Nimble volume (including snapshots)
                   backing this volume should be destroyed when this volume
                   is unmounted or detached

Import Volume options:
-o importVol=X X is the name of the Nimble Volume to import
-o pool=X      X is the name of the pool in which the volume to be
               imported resides (optional)
-o folder=X    X is the name of the folder in which the volume to be imported

               resides (optional)
-o forceImport forces the import of the volume.  Note that overwrites
               application metadata (optional)
-o restore     restores the volume to the last snapshot taken on the
               volume (optional)
-o snapshot=X  X is the name of the snapshot which the volume will be
               restored to, only used with -o restore (optional)

Import Clone of Snapshot options:
-o importVolAsClone=X  X is the name of the Nimble Volume and Nimble Snapshot

                       to clone and import
-o snapshot=X          X is the name of the Nimble snapshot to clone and
                       import (optional, if missing, will use the most
                       recent snapshot)
 -o createSnapshot     indicates that a new snapshot of the volume should be
                       taken and used for the clone (optional)
-o pool=X              X is the name of the pool in which the volume to be
                       imported resides (optional)
-o folder=X            X is the name of the folder in which the volume to be
                       imported resides (optional)
-o destroyOnRm         indicates that the Nimble volume (including snapshots)

                       backing this volume should be destroyed when this
                       volume is deleted
-o destroyOnDetach     indicates that the Nimble volume (including snapshots)

                       backing this volume should be destroyed when this
                       volume is unmounted or detached
```

Note If the **docker** command is unavailable, you can communicate directly with the UNIX® socket of the HPE Nimble Storage Docker Volume plugin to retrieve the help output:

```
$ sudo curl -s -d '{ "Name": "no-op", "Opts": {"help": ""}}' \
--unix-socket /run/docker/plugins/nimble.sock http:/VolumeDriver.Create \
| sed -e 's#\\n#\n#g'
```

Security considerations

Red Hat OpenShift and OpenShift Origin ship with a restricted Security Context Constraint (SCC). By default, end users are not allowed to run containers as specific users; instead, users are assigned UIDs from a pool. This arrangement causes problems with the default behavior of freshly formatted XFS volumes, and it also causes "permission denied" issues. A workaround is to include the `fsMode` parameters with every StorageClass and set it to be group writable; for example, `0770`.

It is also possible to allow a user to run containerized processes as the container image designated user. For more information, see the *OpenShift Administration Guide*.

Diagnostics

Three main components are involved in getting storage to a container:

- The HPE Nimble Kube Storage Controller
- The HPE Nimble FlexVolume driver
- The HPE Nimble Storage Docker Volume plugin

Each component is responsible for a different phase of the process. If any component causes a problem, it might be useful to collect the logs and send them to HPE Nimble Storage support.

To view logs of the HPE Nimble Kube Storage Controller, run the following command:

```
$ oc logs deploy/kube-storage-controller-doryd --namespace=kube-system
```

To find the log file location of the HPE Nimble FlexVolume driver, run the following command:

```
$ tail /var/log/dory.log
```

The HPE Nimble Storage Docker Volume plugin listens to API calls from both of these components. For log file location and basic steps to troubleshoot the plugin, see the *Linux Integration Guide*.

Version history

Version	Release Date	Description
1.2.1	July 2019	Prerequisite and supported version changes.
1.2.0	January 2019	Support for RHEL/CentOS 7.6. Support for OpenShift 3.11. Added Node Isolation Considerations and Container Storage notes.
1.1.0	September 2018	Support for OpenShift 3.10, CRI-O introduction, FC installation notes, Red Hat CNS considerations, OpenShift Container Registry integration. Red Hat Container Catalog.
1.0.1	June 2018	Included NLT Kubernetes optimized installation instructions.
1.0.0	May 2018	Initial release