Hewlett Packard Enterprise

Reference guide

HPE SIMPLIVITY PERSISTENT VOLUME PLUGIN FOR DOCKER

CONTENTS

OVERVIEW

Containers are fast, but data portability is still a challenge. HPE SimpliVity can provision and port your data as fast and efficiently as Docker Containers. Now you can speed up DevOps, lift and shift applications and data, and deliver enterprise grade availability and more.

With our easy-to-use Docker Volume Plugin backed by HPE SimpliVity Hyper Converged Infrastructure, that provides data deduplication and compression benefits and helps accelerate performance and enable space savings, your stateful applications and containers will have powerful persistent storage to match.

AUDIENCE

This guide is intended for systems administrators and developers seeking information on the installation, configuration, and maintenance of the HPE SimpliVity Docker Persistent Volume Plugin. It also covers how to use plugin with basic Docker CLI commands.

SCOPE

The scope of this document is to describe the installation, configuration, and usage of the plugin. It also covers how to debug and uninstall the plugin.

The plugin currently supports Docker platform with Linux® containers only.

PLUGIN COMPONENTS

This plugin has 2 components:

1. HPE SimpliVity nodePlugin

   It needs to be installed on all worker nodes where you want to provision volumes.

2. HPE SimpliVity Container Provider

   It needs to be run as a container on one of the worker nodes in the Swarm mode.

HPE SimpliVity provides a Docker certified plugin (hub.docker.com/plugins/hpe-simplivity-docker-volume-plugin) and Container Provider (hub.docker.com//hpe-simplivity-container-provider) delivered through the Docker Store.

PLUGIN REQUIREMENTS

1. Docker Engine 18.09.01 or greater
2. If using Docker Enterprise Edition 2.x, the plugin is only supported in swarmmode
3. Red Hat® 7.4, Debian, or Ubuntu-based Linux distribution (Ubuntu 18.04/CentOS 7.6)
4. SimpliVity OmniStack 3.7.0 and above, VMware vSphere® 6.5.0 and above

---

NOTE
We have tested the plugin with the versions specified above. In addition, we have tested the plugin with VMware vSphere 6.7.0 and with SimpliVity OmniStack 4.0.0.179

HOW TO USE THIS PLUGIN

Plugin privileges

In order to create connections, attach devices and mount file systems, the plugin requires more privileges than a standard application container. These privileges are enumerated during installation. These permissions need to be granted for the plugin to operate correctly.

```
Plugin "simplivity" is requesting the following privileges:
 - network: [host]
 - mount: [/dev]
 - mount: [/usr]
 - mount: [/run/lock]
```

```
- mount: [/sys]
- mount: [/etc]
- mount: [/var/lib]
- mount: [/var/run/docker.sock]
- mount: [/sbin/iscsiadm]
- mount: [/lib/modules]
- mount: [/lib64]
- mount: [/lib64]
- allow-all-devices: [true]
- capabilities: [CAP_SYS_ADMIN CAP_SYS_MODULE CAP_MKNOD]
```

HPE SimpliVity nodePlugin installation steps

Setting up the plugin varies between Linux distributions. The following workflows have been tested using a HPE SimpliVity Docker Plugin Container provider at **10.110.4.5** with VMware vCenter® **admin** credentials:

These procedures **requires** root privileges and **prerequisites** mentioned below:

1. Install VMware Tools™ on all worker nodes.

   Ubuntu: `sudo apt-get install open-vm-tools-lts-trusty`

   Debian: `sudo apt-get install open-vm-tools`

   Red Hat 7.5+, CentOS 7.5+, Oracle Enterprise Linux 7.5+, and Fedora 28+

   ```
   yum install -y open-vm-tools
   ```

2. Add “disk.EnableUUID = “TRUE”” parameter for all worker nodes VM’s
   a. Locate the virtual machine for which you are enabling the disk UUID attribute, and power off the virtual machine.
   b. After power-off, right-click the virtual machine, and choose Edit Settings.
   c. Click VM Options tab, and select Advanced.
   d. Click Edit Configuration in Configuration Parameters.
   e. Click Add parameter.
   f. In the Key column, type disk.EnableUUID.
   g. In the Value column, type TRUE.
   h. Click OK and click Save.
   i. Power on the virtual machine

Red Hat 7.5+, CentOS 7.5+, Oracle Enterprise Linux 7.5+, and Fedora 28+

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

$ docker plugin install --disable --grant-all-permissions --alias simplivity
hpesimplivity/simplivity:1.0

$ docker plugin set simplivity SCOPE=local PLUGIN_TYPE=simplivity PROVIDER_IP=10.110.4.5
PROVIDER_USERNAME=admin PROVIDER_PASSWORD=admin LOG_LEVEL=trace INSECURE=true PROVIDER_PORT=9080

$ docker plugin enable simplivity

$ systemctl restart docker
```

Ubuntu 18.04 LTS (Ubuntu 19.10 also tested)

```
$ apt-get install -y open-iscsi multipath-tools xfsprogs

$ docker plugin install --disable --grant-all-permissions --alias simplivity
hpesimplivity/simplivity:1.0

$ docker plugin set simplivity SCOPE=local PLUGIN_TYPE=simplivity PROVIDER_IP=10.110.4.5
PROVIDER_USERNAME=admin PROVIDER_PASSWORD=admin LOG_LEVEL=trace INSECURE=true PROVIDER_PORT=9080
glibc_libs.source=/lib/x86_64-linux-gnu

$ docker plugin enable simplivity

$ systemctl restart docker
```

Debian 9.x (stable)

```
$ apt-get install -y open-iscsi multipath-tools xfsprogs

$ docker plugin install --disable --grant-all-permissions --alias simplivity
hpesimplivity/simplivity:1.0

$ docker plugin set simplivity SCOPE=local PLUGIN_TYPE=simplivity PROVIDER_IP=10.110.4.5
PROVIDER_USERNAME=admin PROVIDER_PASSWORD=admin LOG_LEVEL=trace INSECURE=true PROVIDER_PORT=9080
iscsiadm.source=/usr/bin/iscsiadm glibc_libs.source=/lib/x86_64-linux-gnu

$ docker plugin enable simplivity

$ systemctl restart docker
```

HPE SimpliVity Container Provider installation steps

To run HPE SimpliVity Container Provider as a container on one of the worker node in the cluster, we need to follow below steps:

1. First create 3 directories at below paths in the VM where you are planning to run this container.

   ```
   $ mkdir -p /simplivity/var/private/config/current/

   $ mkdir -p /simplivity/var/private/log/group/

   $ mkdir -p /simplivity/var/private/temp/
   ```

2. Now create appint-container-provider.default file to configure the HPE SimpliVity Container Provider properties.

   ```
   $ vi /simplivity/var/private/config/current/appint-container-provider.default
   ```

3. Configure datacenter, datastore, VMware ESXi™ hostname, OmniStack IP and VMware vCenter Server® IP. Below is a sample config file.

   ```
   docker.volume.datacenter=Barcelona

   docker.volume.datastore=SVT_Barcelona01

   docker.volume.hostname=omnicube304240.cloud.local

   docker.volume.omnistack.hostname=10.30.4.245

   docker.volume.vcenter.server=10.20.0.58
   ```

4. Now pull the Docker image and then run it as a container.

   ```
   $ sudo docker pull hpesimplivity/simplivity-container-provider:1.0

   $ sudo docker run -v
   /simplivity/var/private/config/current/:/simplivity/var/private/config/current/ \

   -v /simplivity/var/private/log/group/:/simplivity/var/private/log/group/ \

   -v /simplivity/var/private/temp/:/simplivity/var/private/temp \

   -p 9080:9080 -e JAVA_OPTIONS="-Djetty.port=9080" \

   -e "TZ=America/Los_Angeles" \

   hpesimplivity/simplivity-container-provider:1.0
   ```

Making changes

The `docker plugin set` command can only be used on the plugin if it is disabled. To disable the plugin, use the `docker plugin disable` command. For example:

```
$ docker plugin disable simplivity
```

Security consideration

The HPE SimpliVity credentials are visible to any user who can execute `docker plugin inspect simplivity`.

In the event of reassociating the plugin with a different HPE SimpliVity Docker PV container provider, certain procedures need to be followed:

1. Disable the plugin

   ```
   $ docker plugin disable simplivity
   ```

2. Set new parameters

   ```
   $ docker plugin set simplivity remove=true
   ```

3. Enable the plugin

   ```
   $ docker plugin enable simplivity
   ```

4. Disable the plugin

   ```
   $ docker plugin disable simplivity
   ```

5. The plugin is now ready for re-configuration

   ```
   $ docker plugin set simplivity PROVIDER_IP=<New IP address>
   PROVIDER_USERNAME=admin PROVIDER_PASSWORD=admin remove=false
   ```

NOTE

The remove=false parameter must be set if the plugin ever has been unassociated from a HPE SimpliVity Docker PV container provider.

Configuration files and options

The configuration directory for the plugin is `/etc/hpe-storage` on the host. Files in this directory are preserved between plugin upgrades.

The `/etc/hpe-storage/10.110.4.5/simplivity/local/volume-driver.json` file contains three sections, `global`, `defaults` and `overrides`. The global options are plugin runtime parameters and doesn't have any end-user configurable keys at this time.

The `defaults` map allows the Docker host administrator to set default options during volume creation. The Docker user may override these default options with their own values for a specific option.

The `overrides` map allows the Docker host administrator to enforce a certain option for every volume creation. The Docker user may not override the option and any attempt to do so will be silently ignored.

These maps are essential to discuss with the HPE SimpliVity administrator. A common pattern is that a default protection template is selected for all volumes to fulfill a certain data protection policy enforced by the business it's serving.

NOTE

`defaults` and `overrides` are dynamically read during runtime while `global` changes require a plugin restart.

Below is an example `/etc/hpe-storage/10.110.4.5/simplivity/local/volume-driver.json` outlining the above use cases:

```
{
    "global":   {
        "volumeDir": "/opt/hpe-storage-mounts/simplivity"
```

```
    },
    "defaults": {
                 "sizeInGiB":"10",
                 "limitIOPS":"-1",
                 "limitMBPS":"-1",
                 "perfPolicy": "",
                 "dedupe": "false"
                },
    "overrides":{}
}
```

For an exhaustive list of options, either refer to the HPE SimpliVity plugin documentation or use the help option from the Docker CLI:

```
$ docker volume create -d simplivity -o help

SimpliVity Docker Volume Driver: Create Help

Create or Clone a SimpliVity backed Docker Volume or Restore the SimpliVity backed Volume 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)
```

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 cgName=X             X is the name of Consistency Group (VM) which holds the one or more
                        Volumes (optional)
-o description=X        X is the text to be added to volume description (optional)
-o protectionTemplate=X X is the name of the protection template (default is Fixed Default
                        Backup Policy)
                        Protection Templates: simple, randomPolicyName, Fixed Default Backup
                  Policy
```

Clone options:

```
-o cloneOf=X            X is the name of Docker Volume to create a clone of
-o cgFlag=X             X if sets to true, indicates to delete all the other metadata files
           in case of cloning the simpliVity volume
```

```
Restore Volume options:

-o restoreOf=X          X is the name of the SimpliVity backup volume to restore

-o cgName=X             X is the name of Consistency Group (VM) which holds the one or more
            volumes

-o cgFlag=X             X if sets to true, indicates to delete all the other metadata files in
                  case of cloning the simpliVity volume
```

Uninstall

The plugin can be removed using the `docker plugin rm` command. This command will not remove the configuration directory (/etc/`hpe-storage`).

```
$ docker plugin rm simplivity
```

Debugging

The HPE SimpliVity plugin documentation covers basic debugging. That documentation applies to this plugin as well. To do this, use the docker plugin ls command.

```
$ docker plugin ls

ID              NAME              DESCRIPTION                                    ENABLED

8a4231e33dc8    simplivity:1.0    Docker Volume plugin for HPE SimpliVity         true
```

Note the ID column in the output. Use this to glob the directory to the log of the plugin.

```
less /var/lib/docker/plugins/8a4231e33dc8*/rootfs/var/log/hpe-flexvolume-plugin.log
```

Usage of the HPE SimpliVity Volume Plug-in for Docker

The following guide covers many of the options used for provisioning volumes and volume management within standalone Docker environments.

Creating a basic HPE SimpliVity volume

```
$ sudo docker volume create -d simplivity --name <vol_name>
```

HPE SimpliVity Docker Volume parameters

The HPE SimpliVity Docker Volume Plug-in supports several optional parameters that can be used during volume creation:

- **size**—specifies the desired size in GB of the volume. If size is not specified during volume creation, it defaults to 100 GB
- **mountConflictDelay**—specifies period in seconds to wait for a mounted volume to gracefully unmount from a node before it can be mounted to another. If graceful unmount doesn't happen within the specified time then a forced cleanup of the VLUN is performed so that volume can be remounted to another node
- **cgName**—name of Consistency Group(VM) which holds the one or more volumes(optional)
- **description**—text to be added to volume description (optional)
- **cloneOf**—name of Docker Volume to create a clone of
- **cgFlag**—if sets to true, indicates to delete all the other metadata files in case of cloning the HPE SimpliVity volume
- **restoreOf**—name of the HPE SimpliVity backup volume to restore
- **fsOwner**—user ID and group ID that should own root directory of file system
- **fsMode**—mode of the root directory of file system to be specified as octal number
- **protectionTemplate**—name of the protection template (default is Fixed Default Backup Policy)
- **help**—displays usage help and backend initialization status

The following is an example Docker command creating 50 GB volume:

```
$ docker volume create -d simplivity --name <vol_name> -o size=50
```

Enabling file permissions and ownership

1. To set permissions of root directory of a file system:

```
$ docker volume create -d simplivity --name <volume-name> -o fsMode=<octal-number-specified-in-chmod>
```

2. To set ownership of root directory of a file system:

```
$ docker volume create -d simplivity --name <volume-name> -o fsOwner=<UserId>:<GroupId>
```

Creating volume with backup policy to get scheduled backups

```
$ docker volume create -d simplivity --name <volume-name> -o
protectionTemplate=<backup_policy_name>
```

Deleting a volume

```
$ docker volume rm <vol_name>
```

Listing volumes

```
$ docker volume ls
```

Inspecting a volume

```
$ docker volume inspect <vol_name>
```

Mounting a volume

Use the following command to mount a volume and start a bash prompt:

```
$ docker run -it -v <vol_name>:/<mount_point>/ --volume-driver simplivity <image_name> bash
```

e.g. $ docker run -it -v devVolume:/data --volume-driver simplivity alpine /bin/sh

The image used for mounting can be any image located on hub.docker.com/ or the local filesystem. See docs.docker.com/docker-hub/official_images/ for more details.

Unmounting a volume

Exiting the bash prompt will cause the volume to unmount:

```
/ exit
```

The volume is still associated with a container at this point.

Run the following command to get the container ID associated with the volume:

```
$ sudo docker ps -a
```

Then stop the container:

```
$ sudo docker stop <container_id>
```

Next, delete the container:

```
$ sudo docker rm <container_id>
```

Finally, remove the volume:

```
$ sudo docker volume rm <vol_name>
```

Creating a clone of a volume

```
$ docker volume create -d simplivity --name <target_vol_name> -o cloneOf=<source_vol_name> -o
cgFlag=<True/False>
```

Restoring volumes

```
$ docker volume create -d simplivity --name <target_vol_name> -o restoreOf=<backup_vol_name> -o
cgName=<source_vol_name> -o cgFlag=<True/False>
```

Displaying help on usage

```
$ docker volume create -d simplivity -o help
```

Displaying available backups

```
$ docker volume inspect <source_vol_name>
```

Support and other resources

Issues and feedback

Issues pertaining to the management pack can be raised in the issues tab in the public GitHub repo.

Any feedback/enhancements to be proposed can be done in the same tab with relevant tags.

The issues tab can be found here: github.com/HewlettPackard/Docker-SimpliVity-Volume-Plugin/issues

Updates

Subsequent updates for newer versions of HPE SimpliVity Volume Plugin for Docker would be released in the public GitHub site itself and the release information can be found in the release notes link: github.com/HewlettPackard/Docker-SimpliVity-Volume-Plugin/blob/master/README.md

LEARN MORE AT

github.com/HewlettPackard/Docker-SimpliVity-Volume-Plugin/blob/master/README.md

January 2020

Hewlett Packard Enterprise