HPE Solution Architecture for Backup and Recovery of Red Hat OpenShift Container Platform on HPE Synergy and HPE Nimble

Contents

Executive summary

In today's digital world, organizations are under increasing pressure to deliver applications faster while reducing costs. As these applications grow more complex, this puts stress on IT infrastructure, teams, and processes. To remain competitive, organizations must quickly adapt and developers need to be more effective, efficient, and agile. Container technology provides the right application platform to help organizations become more responsive and iterate across multiple IT environments as well as develop, deploy, and manage applications faster. But implementing a containerized environment across existing infrastructure is a complex undertaking that can require weeks or months to mobilize, particularly for enterprises. To help accelerate container application delivery, Hewlett Packard Enterprise (HPE) and Red Hat® are collaborating to optimize Red Hat OpenShift Container Platform (OCP) on HPE platforms, including HPE Synergy, the industry's first composable infrastructure, and HPE Nimble Storage.

Red Hat OpenShift Container Platform (OCP) on HPE Synergy provides an end-to-end, fully-integrated container solution that, once assembled, can be configured within hours. This eliminates the complexities associated with implementing a container platform across an enterprise data center and provides the automation of hardware and software configuration to quickly provision and deploy a containerized environment at scale. Red Hat OCP provides organizations with a reliable platform for deploying and scaling container-based applications. HPE Synergy Composable Infrastructure provides the flexibility to customers that they need to run that container platform and dynamically provision and scale applications, whether they run on virtual machines or containers, hosted on-premises, in the cloud, or as a hybrid cloud.

Containers have dramatically increased in popularity as organizations recognize the benefits with respect to both time and resource efficiency. This explosive growth of container applications overwhelms traditional data protection approaches. Applying traditional data protection strategies to containerized applications will simply not work. This document highlights the importance of protecting each component of the OCP cluster in order to restore in case of corruption or system failures. It also addresses persistent volume backup using HPE Recovery Manager Central (RMC) software with the HPE 3PAR StoreServ snapshot feature and HPE StoreOnce.

Target audience: This document is intended for systems engineers, systems administrators, architects, and installers responsible for installing and maintaining Red Hat Enterprise Linux (RHEL) and Red Hat OpenShift Container Platform on a large scale running on HPE Synergy Composable Infrastructure and HPE 3PAR StoreServ storage. The reader of this document should be familiar with RHEL, HPE Synergy Composable Infrastructure, HPE Nimble Storage, HPE StoreOnce and general backup and recovery concepts.

Solution overview

An OpenShift cluster is made up of several nodes and each node type has different roles. To protect the environment, it is very important to understand how these components fit together and the services provided by each component. A successful backup and recovery solution is highly unlikely without this understanding in place. This section provides details of each OpenShift node and what components require protection within the environment.

For the purpose of making the infrastructure highly available, there are three (3) master nodes, three (3) infrastructure nodes and three (3) etcd nodes. Worker nodes may be deployed in variable quantities. It is necessary to create a backup of the important components within the OCP cluster in order to recreate the nodes in the event of a failure. Figure 1 describes the major components involved in the deployment of Red Hat OCP.

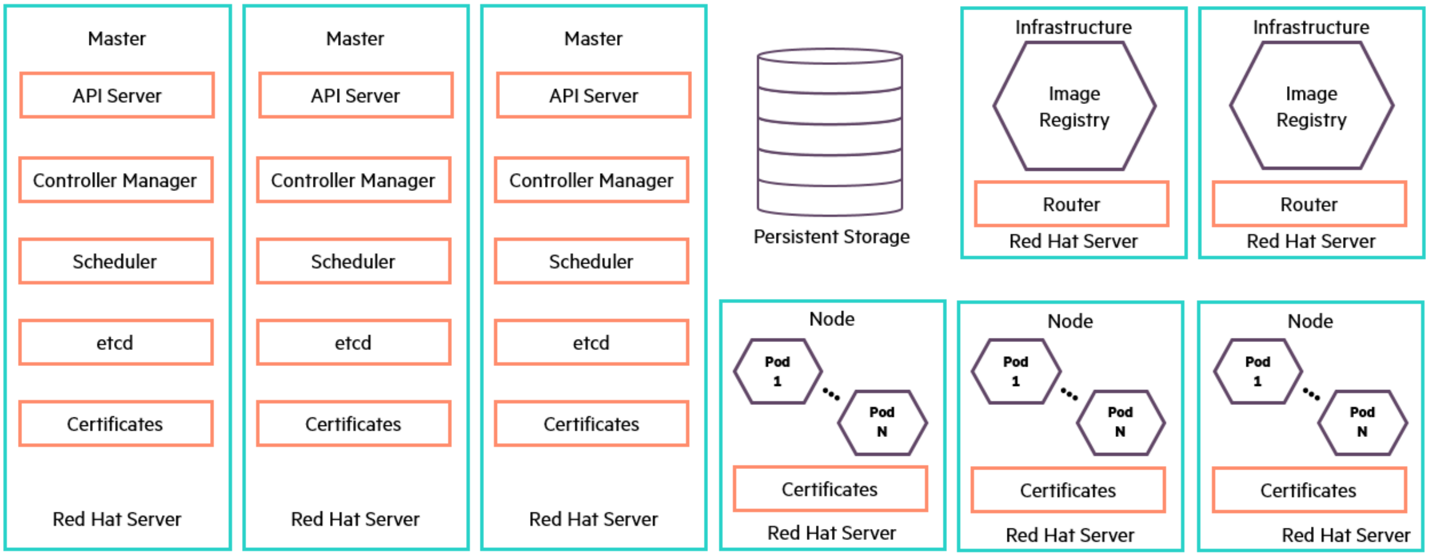


Figure 1. Solution diagram

OpenShift master node

The OpenShift master nodes are the brains of the cluster and are comprised of a set of core components including:

- API server
- Controller Manager server
- Scheduler
- etcd server

The master nodes maintain the cluster's configuration, manage nodes in the OpenShift cluster and schedule pods to run on nodes. If the OpenShift master nodes are unable to function this will not impact the end users as the container application traffic will remain functional. However, administrators and users will not be able to make any new adjustments to the OpenShift cluster.

API server

The API server provides the management entry point of the OpenShift cluster. It mediates the interactions between the OpenShift master node components via RESTful API calls. It is responsible for storing API objects into the persistent etcd store. API server high availability is built on the persistent etcd store and deploys multiple instances of API server roles on the OpenShift cluster.

Controller Manager

The Controller Manager monitors the state of the cluster through the API Server watch feature. When a state change notification is received, it makes the necessary changes attempting to move the current state towards the desired state to keep the OpenShift cluster functioning correctly. Multiple controller manager roles are configured on OpenShift master nodes to provide high availability.

Scheduler

The scheduler ensures that container applications are scheduled to run on worker nodes within the OpenShift cluster. The scheduler reads data from the pod and attempts to find a node that is a good fit based on configured policies. To ensure high availability, more than one OpenShift master node must be configured for the scheduler roles.

etcd

etcd stores the persistent master state while other components watch etcd for changes to bring themselves into the desired state. It implements the key-value stores where all of the objects in OpenShift cluster master node components are stored. The etcd store implements a distributed consensus algorithm to ensure that even if one of the storage nodes fail, there is sufficient replication to maintain data availability. etcd can be optionally configured for high availability by deploying 2n+1 etcd nodes.

Certificates

Certificates are used by the API server when securing inbound requests, authenticating users, making outbound requests, and for mutual TLS between the API server and all the other API objects in OpenShift Cluster. Certificates are copied to all the master nodes during the deployment. If more than one master host is deployed on an OpenShift cluster, the certificates are considered highly available.

OpenShift node

An OpenShift node, or worker node, provides the runtime environment for containers. Each node in an OpenShift cluster has the required services to be managed by the master. The master uses information from nodes to validate nodes with health checks. A node is ignored until it passes the health checks, and the master continues checking nodes until they are valid. Other than running pods, worker nodes contain certificates, services and authorization files. Large numbers of OpenShift nodes may be deployed in a cluster and if one node fails, it can be easily replaced without losing valuable data. However, certificates need to be deployed on the new node. Typically, certificates and authorization files are redeployed using Ansible playbooks and the Ansible Engine/Tower will hold the files. As a result, the Ansible Engine/Tower must be protected to ensure the file is highly available.

OpenShift infrastructure node

OpenShift makes use of its local Docker registry for storing Docker images. In a highly available deployment such as the one HPE has created, the infrastructure nodes are responsible for hosting these registry pods and this is the place where the local container images are stored. Registry pods are assigned with a Persistent Volume (PV) from external storage. In order to protect the data, it is recommended to take a snapshot or clone the volume or replicate it to a disaster recovery site.

Pods inside of an OpenShift cluster are only available via their IP addresses on the cluster network. An edge load balancer can be used to accept traffic from outside networks and route the traffic to pods inside the OpenShift cluster. An OpenShift administrator can deploy routers in an OpenShift cluster through infrastructure nodes. These enable routes created by developers to be used by external clients. OpenShift routers provide external hostname mapping and load balancing to services over protocols that pass distinguishing information directly to the router. The hostname must be present in the protocol in order for the router to determine where to send traffic. For high availability, an external load balancer such as F5 BIG-IP can be used along with multiple infrastructure nodes.

Persistent storage

Containers were originally designed to run stateless applications so there was no need for persistent storage. When enterprises began adopting containers and they wanted to run stateful applications persistent storage became necessary to meet the demands of the application data. HPE Nimble Storage provides persistent storage capabilities to an OpenShift cluster using plugins. Persistent storage and the data it houses need to be protected for business continuity and disaster recovery purpose. Data protection is inbuilt with HPE Nimble Storage utilizing snapshot and replication capabilities.

Solution hardware

For this solution, OCP is built on an HPE Converged Architecture 750 which offers an improved time to deployment and tested firmware recipe. The user also has the flexibility of customizing the HPE components throughout this stack as per their unique IT and workload requirements or building with individual components. Figure 2 shows the physical configuration of the racks used in this solution.

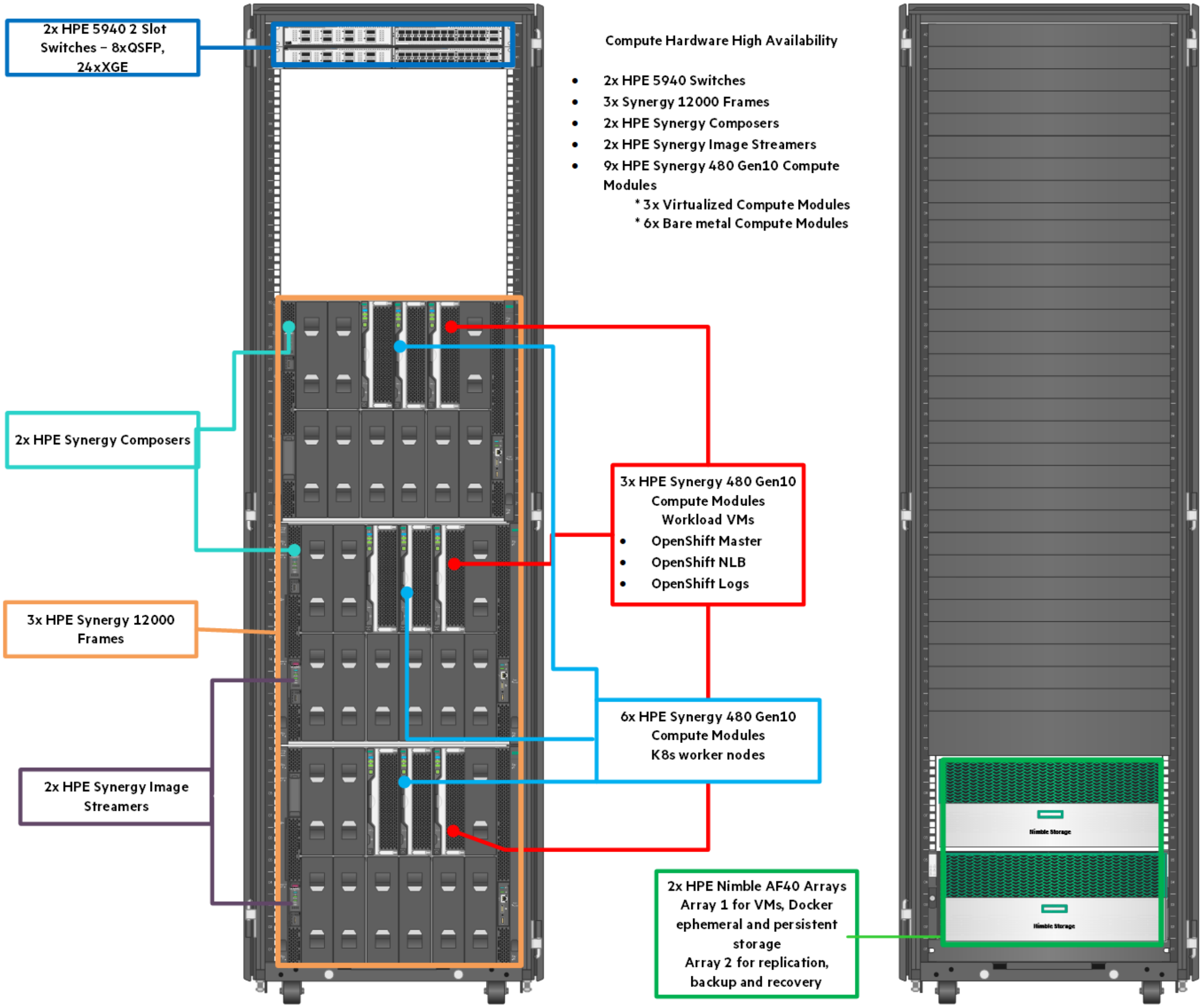

2x HPE 5940 2 Slot Switches – 8xQSFP, 24xXGE
2x HPE Synergy Composers
3x HPE Synergy 12000 Frames
2x HPE Synergy Image Streamers
Compute Hardware High Availability
• 2x HPE 5940 Switches
• 3x Synergy 12000 Frames
• 2x HPE Synergy Composers
• 2x HPE Synergy Image Streamers
• 9x HPE Synergy 480 Gen10 Compute Modules
* 3x Virtualized Compute Modules
* 6x Bare metal Compute Modules
3x HPE Synergy 480 Gen10 Compute Modules Workload VMs
• OpenShift Master
• OpenShift NLB
• OpenShift Logs
6x HPE Synergy 480 Gen10 Compute Modules K8s worker nodes
2x HPE Nimble AF40 Arrays
Array 1 for VMs, Docker ephemeral and persistent storage
Array 2 for replication, backup and recovery
Nimble Storage
Nimble Storage

Figure 2. Physical layout of the compute within the solution

For detailed configuration information including a list of components, software versions and how the solution is built, consult the Reference Configuration at https://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=a00056101enw.

Backup and recovery Considerations

A container application data protection architecture should be defined by business requirements. These requirements include factors such as the speed of recovery, the maximum permissible data loss, and data retention needs. The data protection plan must also take into consideration various regulatory requirements for data retention and restoration. Finally, different data recovery scenarios must be considered, ranging from the typical and foreseeable recovery resulting from user or application errors to disaster recovery scenarios that include the complete loss of a data center.

Small changes in data protection and recovery policies can have a significant impact on the overall architecture of Container infrastructure. It is critical to define and document standards before starting design work to avoid complicating a data protection architecture. Unnecessary features or levels of protection lead to unnecessary costs and management overhead, and an initially overlooked requirement can lead a project in the wrong direction or require last-minute design changes.

Two of the important parameters that define a backup and recovery plan are recovery point objective (RPO) and recovery time objective (RTO).

RPO refers to the maximum amount of data you can afford to lose. It is measured backward in time from a failure to the last available backup. RTO refers to the maximum downtime it takes to restore and recover an application after a disaster or corruption. It is stated in terms of time.

RPO and RTO requirements may vary based on the business needs.

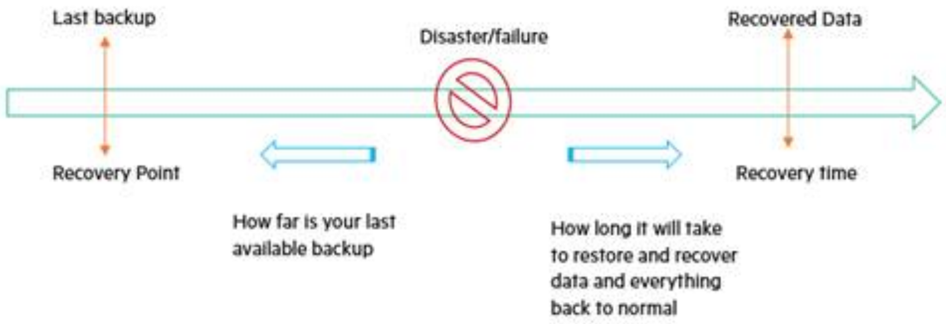

Last backup
Disaster/failure
Recovered Data
Recovery Point
Recovery time
How far is your last available backup
How long it will take to restore and recover data and everything back to normal

Figure 3. RPO/RTO description

Backup is defined as the procedure to create extra copies of data in case the original data is lost or corrupted. This helps to restore the data to a newly-created environment or to relevant components in case of a disaster or corruption.

For the purpose of this solution, HPE recommends the use of an iSCSI volume exported from HPE Nimble Storage to a master server and copy the configuration backups to this volume. The volume is protected using an HPE Nimble Storage Data Protection plan. Persistent volumes are created using the HPE Nimble Storage data protection.

Backup OpenShift Container Platform components

This section disseminates the components that need to be protected or backed up within a Red Hat OpenShift cluster. Hewlett Packard Enterprise plans to update this section with updated information over time.

A backup should be taken before any upgrade or modifications. It is recommended to perform a periodic backup to make sure you have the most recent configuration available in the event of a failure. An OCP backup involves taking backup of current state to external storage at the cluster level. This means creating individual backups of the following components:

- Master node components
- Worker node components
- Infrastructure node components
- etcd data and configurations
- Persistent storage

In Red Hat OpenShift, all of the components are treated as objects and are stored in files. This means that creating a configuration backup is the same as taking file level backups**.**

Master node

The master node is responsible for maintaining the desired state of a cluster. It is recommended to perform a master node backup before making any modifications to the OCP infrastructure. In High Availability environments, make sure to perform the backup on all master nodes.

Master node components are stored under the `/etc/origin/` and `/etc/sysconfig/` directories. Please refer to Appendix A for more details.

Worker node

The nature of worker nodes is that any specific configuration pertaining to running pods are replicated over the nodes in case of a failover, and they typically do not contain data that is necessary to run an environment.

Apart from running pods, worker nodes contain certificates that are generated during installation, services, authorization files and more. These files are stored under the `/etc/origin/` and `/etc/sysconfig/` directories. Please refer to Appendix A for more details

Infrastructure node

OpenShift uses its local registry for storing container images. The infrastructure node is responsible for hosting the registry and routers. Registry pods are deployed with a persistent volume from HPE Nimble Storage. In order to protect the data, it is recommended to take a snapshot of the volume and protect it by replicating it to a remote Nimble Storage array or to an HPE Cloud Volumes .

Registry certificates must be backed up from the `/etc/docker/certs.d` directory. Please refer to Appendix A for more details.

etcd

OpenShift uses etcd to store system configuration and state as well as metadata. etcd is a key value store for all of the object definitions. It allows nodes in the cluster to read and write data.

When backing up etcd, make sure to create a backup of the configuration and data.

etcd configurations are stored in the `/etc/etcd` directory where etcd instances are running. Unlike other configurations, etcd configurations are unique across etcd instance. etcd data can be backed up using the etcd snapshot save command or by copying the `/var/lib/etcd/member/snap/db` file to a desired location. Refer to Appendix A for more details.

Persistent storage

Containers were designed to run stateless applications. In the beginning, there was no need for persistent storage. However, the enterprises started adopting containers and they wanted to run their applications on stateful containers. Once persistent data is present, a need is created for persistent storage. Backing up and protecting this data becomes very important.

For persistent volume level backups, traditional agent-based backup software won't work natively with a container orchestrator such as Red Hat OpenShift. Backup schemes need to be consumed as a data protection service from the underlying container-aware storage infrastructure such as Nimble Storage. Refer to Appendix A for more details.

Restore OCP components

It is important to restore the OCP components in case of system failure or corruption and to ensure the nodes are in a previous working state.

Master node

Restore means recreating the components from the point in time the backup is available. In the case where a master host is corrupted or failed due to system error, reinstall the master host, copy the important configuration files and then restart the OCP services.

If restoring to a master, which is behind a highly available load balancer pool, restarting OCP service may cause downtime. Make sure to remove the master from the pool, restart the service and then add it back to the load balancer pool.

If recreating a master after the system failure, restore the backup, reboot and then add the master to the cluster. Refer to Appendix A for more details.

Worker node

In case a worker node host is corrupt or has failed due to a system error, reinstall the worker node the same way as initially installed and then copy the important configuration files back to their original locations. Once complete, restart the OCP services.

If recreating a worker node after the system failure, apply backup, reboot and then add the worker to the cluster. Refer to Appendix A for more details.

Infrastructure node
In case an infrastructure node is down due to system error, reinstall the worker, backup the certificates, and restore the persistent volume claim from the external storage. Refer to Appendix A for more details.

etcd
If the etcd configuration is corrupted or lost, restore the `/etc/etcd/etcd.conf` file from the backup and restart the service.

If the etcd data is corrupted and want to restore from the snapshot can be performed to a single etcd node. Once complete, add rest of the etcd node to the cluster. Refer to Appendix A for more details.

Persistent storage
For the persistent volume protection, use the storage capabilities within HPE Nimble Storage to take crash consistent snapshots of persistent volumes. The volumes can then be restored to a point in time from the snapshots. Refer to Appendix A for more details.

For more restore functionalities, refer to https://community.hpe.com/t5/HPE-Storage-Tech-Insiders/Data-Protection-for-Containers-Part-II-Restore/ba-p/7019117#.XQyAJ-gzZ9M.

Appendix A: Backup and Restore OpenShift node components

In order to protect the OCP components, it is recommended to take a backup of important configurations to Nimble Volume and replicate to a remote HPE Nimble Storage array.

Before initiating backup of the configuration files, a dedicated volume should be created to serve as a backup target as explained below.

1. Log in to the HPE Nimble Storage Administration web GUI and navigate to **MANAGE → DATA STORAGE** and then click **+** to create the new volume as shown in Figure A1.

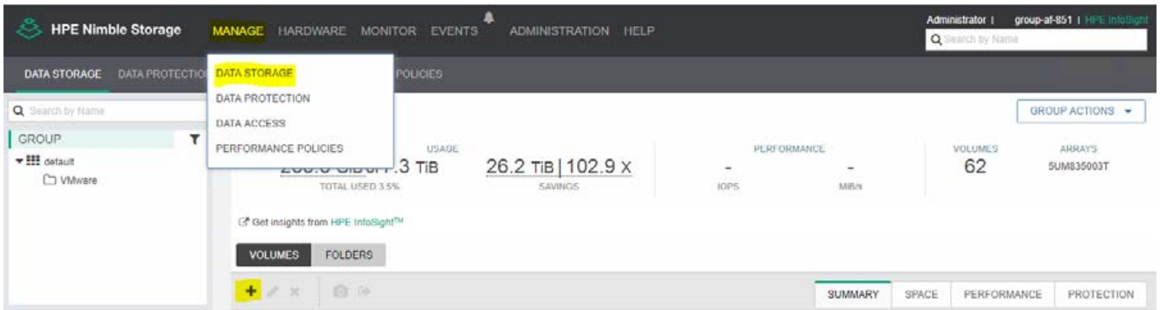
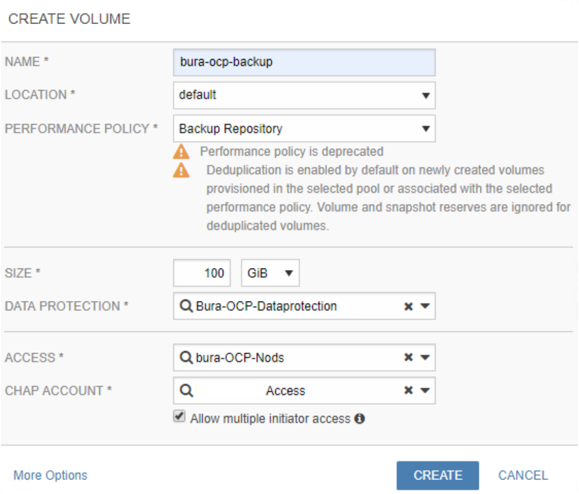


Figure A1. Creating new volume in HPE Nimble Storage

2. Give a name for the volume, select the **PERFORMANCE POLICY** for backup repository, select the **DATA PROTECTION** plan and give access to the OCP node where to mount the Volume as in Figure A2. In this solution it is mounted on the master server. It is recommended to configure the restricted access using a CHAP account.



CREATE VOLUME

NAME * bura-ocp-backup

LOCATION * default

PERFORMANCE POLICY * Backup Repository

⚠ Performance policy is deprecated

⚠ Deduplication is enabled by default on newly created volumes provisioned in the selected pool or associated with the selected performance policy. Volume and snapshot reserves are ignored for deduplicated volumes.

SIZE * 100 GiB

DATA PROTECTION * Bura-OCP-Dataprotection

ACCESS * bura-OCP-Nods

CHAP ACCOUNT * Access

☑ Allow multiple initiator access

Figure A2. HPE Nimble volume creation

3. Discover the newly created volume on the servers using the following command.

```
# iscsiadm -m discovery -t sendtargets -p <nimble discover ip>

# iscsiadm -m node -T <target iqn> -p <nimble target ip> --login
```

4. Create a filesystem and mount the filesystem on the master node.

```
# fdisk </dev/mapper/mpathx>

# mkfs.ext4 </dev/mapper/mpathx1>

# mount </dev/mapper/mpathx1> nimble_vol
```

Master node backup

Run the following commands to create a configuration backup of the important files and copy them to the previously created mount point.

```
# MYBACKUPDIR=/backup/$(hostname)/$(date +%Y%m%d)

# sudo mkdir -p ${MYBACKUPDIR}/etc/sysconfig

# sudo cp -aR /etc/origin ${MYBACKUPDIR}/etc

# sudo cp -aR /etc/sysconfig/ ${MYBACKUPDIR}/etc/sysconfig/

# sudo cp -aR /etc/sysconfig/{iptables,docker-*} ${MYBACKUPDIR}/etc/sysconfig/

# sudo cp -aR /etc/dnsmasq* /etc/cni ${MYBACKUPDIR}/etc/

# rpm -qa | sort | sudo tee $MYBACKUPDIR/packages.txt

# cp -aR /etc/docker/certs.d/  ${MYBACKUPDIR}/docker-registry-certs-$(hostname)

# sudo tar -zcvf /backup/$(hostname)-$(date +%Y%m%d).tar.gz $MYBACKUPDIR
```

Copy the tar file to the Nimble volume mount point at the master server /nimble_vol by running the following command.

```
# sudo scp -aR /backup/$(hostname)-$(date +%Y%m%d).tar.gz <user>@<master>:/nimble_vol
```

Worker node backup
Creating worker node configuration backup of all the important configuration files to above created mount point.

```
# MYBACKUPDIR=/backup/$(hostname)/$(date +%Y%m%d)

# sudo mkdir -p ${MYBACKUPDIR}/etc/sysconfig

# sudo cp -aR /etc/origin ${MYBACKUPDIR}/etc

# sudo cp -aR /etc/sysconfig/atomic-openshift-node ${MYBACKUPDIR}/etc/sysconfig/

# sudo mkdir -p ${MYBACKUPDIR}/etc/sysconfig

# sudo cp -aR /etc/sysconfig/{iptables,docker-*} ${MYBACKUPDIR}/etc/sysconfig/

# sudo cp -aR /etc/dnsmasq* /etc/cni ${MYBACKUPDIR}/etc/

# rpm -qa | sort | sudo tee $MYBACKUPDIR/packages.txt

# cp -aR /etc/docker/certs.d/  ${MYBACKUPDIR}/docker-registry-certs-$(hostname)

# sudo tar -zcvf /backup/$(hostname)-$(date +%Y%m%d).tar.gz $MYBACKUPDIR
```

Copy the tar file to the Nimble volume mounted directory at /nimble_vol.

```
# sudo scp -aR /backup/$(hostname)-$(date +%Y%m%d).tar.gz <user>@<master>:/nimble_vol
```

Run the following commands and execute the Ansible play to take the master and worker backups and move the data to the HPE Nimble Storage volume.

```
# mkdir ~/git

# cd ~/git

# git clone https://github.hpe.com/Solutions/Openshift-Synergy-RA.git

# cd /Openshift-Synergy-RA/synergy/scalable/bura

# ansible-playbook -i hosts site.yaml
```

Infrastructure node backup
OpenShift makes use of the infrastructure nodes to host the registry pods. Registry pods are assigned with a persistent volume from HPE Nimble Storage. In order to protect the volume, a storage admin has to configure a protection plan in HPE Nimble Storage in such a way that it will trigger the periodic snapshot and replicate that to a remote Nimble Storage. How to configure protection plan for a persistent volume is explained in Appendix B.

Apart from the persistent volume there are certain certificates that need to be backed up from the infrastructure nodes. In order to create a backup of these certificates execute the commands shown below.

```
# MYBACKUPDIR=/backup/$(hostname)/$(date +%Y%m%d)

# sudo mkdir -p ${MYBACKUPDIR}/etc/docker

# cp -R /etc/docker/certs.d  ${MYBACKUPDIR}/etc/docker/certs-$(date +%Y%m%d)

# sudo tar -zcvf /backup/$(hostname)-$(date +%Y%m%d).tar.gz $MYBACKUPDIR
```

Copy the tar file to the HPE Nimble Storage volume mounted at /nimble_vol using the following command.

```
# sudo scp -aR /backup/$(hostname)-$(date +%Y%m%d).tar.gz <user>@<master>:/nimble_vol
```

etcd backup

With this backup solution, etcd is running on a separate host and not as a static pod on the master. As a result, the etcd backup process is comprised of two different procedures, etcd configuration backup, including the required etcd configuration and certificates and etcd data backup

etcd configuration

The etcd configuration files to be preserved and stored in the `/etc/etcd` directory of the instances where etcd is running. This includes the etcd configuration file (`/etc/etcd/etcd.conf`) and the required certificates for cluster communication. Backup the configuration from all etcd members of the cluster using the following commands.

```
# MYBACKUPDIR=/backup/$(hostname)/$(date +%Y%m%d)

# mkdir -p ${MYBACKUPDIR}/etcd-config-$(date +%Y%m%d)

# cp -R /etc/etcd/ ${MYBACKUPDIR}/etcd-config-$(date +%Y%m%d)
```

etcd Data

1. Before backing up, ensure that the OpenShift Container Platform API service is running, connectivity with the etcd cluster (port 2379/tcp) is working and proper certificates to connect to the etcd cluster exist. To validate, run the following commands using the etcd API version V3.

```
# systemctl show etcd --property=ActiveState,SubState

# etcdctl -C https://xx.0.62.xx:2379,https://xx.0.62.xx:2379,https://xx.0.62.2xx:2379 --ca-
file=/etc/etcd/ca.crt --cert-file=/etc/etcd/peer.crt --key-file=/etc/etcd/peer.key cluster-health

# etcdctl -C https://xx.0.62.xx:2379,https://xx.0.62.xx:2379,https://xx.0.62.xx:2379 --ca-
file=/etc/etcd/ca.crt --cert-file=/etc/etcd/peer.crt --key-file=/etc/etcd/peer.key member list
```

2. Using the etcd v3 API, take a snapshot from a live member with the `etcdctl snapshot` command and save it to external storage using the following commands.

```
# MYBACKUPDIR=/backup/$(hostname)/$(date +%Y%m%d)

# mkdir -p ${MYBACKUPDIR}/etcd-data-$(date +%Y%m%d)

# ETCDCTL_API=3 etcdctl snapshot save ${MYBACKUPDIR}/etcd-data-$(date +%Y%m%d)/snapshot.db --
endpoints=https://xx.0.62.xx:2379 --cacert=/etc/etcd/ca.crt --cert=/etc/etcd/server.crt --
key=/etc/etcd/server.key
```

3. Create the etcd data backup and copy the etcd db file using the following commands.

```
# etcdctl2 backup --data-dir /var/lib/etcd --backup-dir ${MYBACKUPDIR}/etcd-data-$(date +%Y%m%d)

# tar -zcvf /backup/$(hostname)-$(date +%Y%m%d).tar.gz $MYBACKUPDIR
```

4. Copy the tar file to the HPE Nimble Storage volume mounted at /nimble_vol.

```
# scp -aR /backup/$(hostname)-$(date +%Y%m%d).tar.gz <user>@<master>:/nimble_vol
```

Restore OCP components from backup

1. To restore the master node or files, mount the backup volume from HPE Nimble Storage to the node and copy the required files to the desired location and then restart the service. The following steps illustrate restoring sample files.

```
# mount /dev/mapper/mpathx /nimble_vol

# bzip2 -d /nimble_vol/$(hostname)-$(date +%Y%m%d).tar.bz2

# tar -xvf /nimble_vol/$(hostname)-$(date +%Y%m%d).tar
# cp /nimble_vol/$(hostname)/$(date +%Y%m%d)/etc/origin/master/master-config.yaml
/etc/origin/master/master-config.yaml

# systemctl restart atomic-openshift-master-api

# systemctl restart atomic-openshift-master-controllers
```

Note
Restart the server to replace the iptables, if required.

2. To restore the worker node or any other backed up files, mount the backup volume from HPE Nimble Storage to the node and copy the required files to the desired location and restart the service. The steps below illustrate restoring sample files.

```
# mount /dev/mapper/mpathx /nimble_vol

# bzip2 -d /nimble_vol/$(hostname)-$(date +%Y%m%d).tar.bz2

# tar -xvf /nimble_vol/$(hostname)-$(date +%Y%m%d).tar

# cp /nimble_vol/$(hostname)/$(date +%Y%m%d)/etc/origin/node/node-config.yaml /etc/origin/node/node-
config.yaml

# systemctl restart atomic-openshift-node
```

Note
If restoring the running worker node, restarting services may cause downtime.

Restoring etcd

1. If an etcd node is corrupted, replace the /etc/etcd/etcd.conf file by restoring it from the backup volume mounted from Nimble Storage, and restart the service. The following steps illustrate restoring sample files.

```
# mount /dev/mapper/mpathx /nimble_vol

# gunzip /nimble_vol/$(hostname)-$(date +%Y%m%d).tar.gz

# tar -xvf /nimble_vol/$(hostname)-$(date +%Y%m%d).tar

# cp /nimble_vol/backup/$(hostname)/$(date +%Y%m%d)/etcd-config-$(date +%Y%m%d)/etcd.conf
/etc/etcd/etcd.conf

# restorecon -Rv /etc/etcd/etcd.conf

# systemctl restart etcd.service
```

2. To restore etcd API version v3 data, run the following commands.

```
# systemctl stop etcd.service

# rm -Rf /var/lib/etcd
# ETCDCTL_API=3 etcdctl snapshot restore /etcdsnap/snapshot.db --data-dir /var/lib/etcd --
endpoints=https://10.0.62.24:2379 --cacert=/etc/etcd/ca.crt --cert=/etc/etcd/server.crt --
key=/etc/etcd/server.key

# chown -R etcd.etcd /var/lib/etcd/

# restorecon -Rv /var/lib/etcd

# systemctl start etcd
```

3. Check the cluster health as mentioned above.

Restoring a Persistent Volume

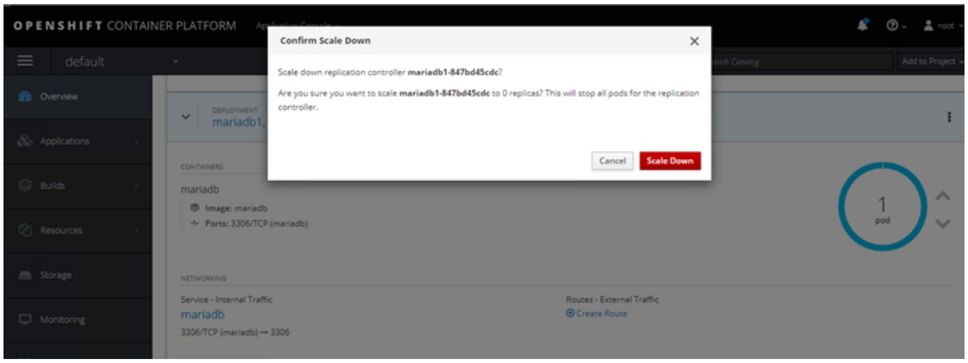
To restore a persistent volume to a point in time, navigate to the HPE Nimble Storage data storage GUI and do the following.

1. Select the volume that belongs to the particular persistent volume, and then navigate to **DATA PROTECTION** tab. Select the point in time snapshot from to restore. Before proceeding with this step, make sure the replica count is set to zero using the Deployment API object in the OCP console.

Note

This causes downtime for the application.

2. From the OCP console, navigate to the project then to Deployment Configuration and select the arrow down to scale down the replica count to 0. Accept the confirmation as in Figure A3.



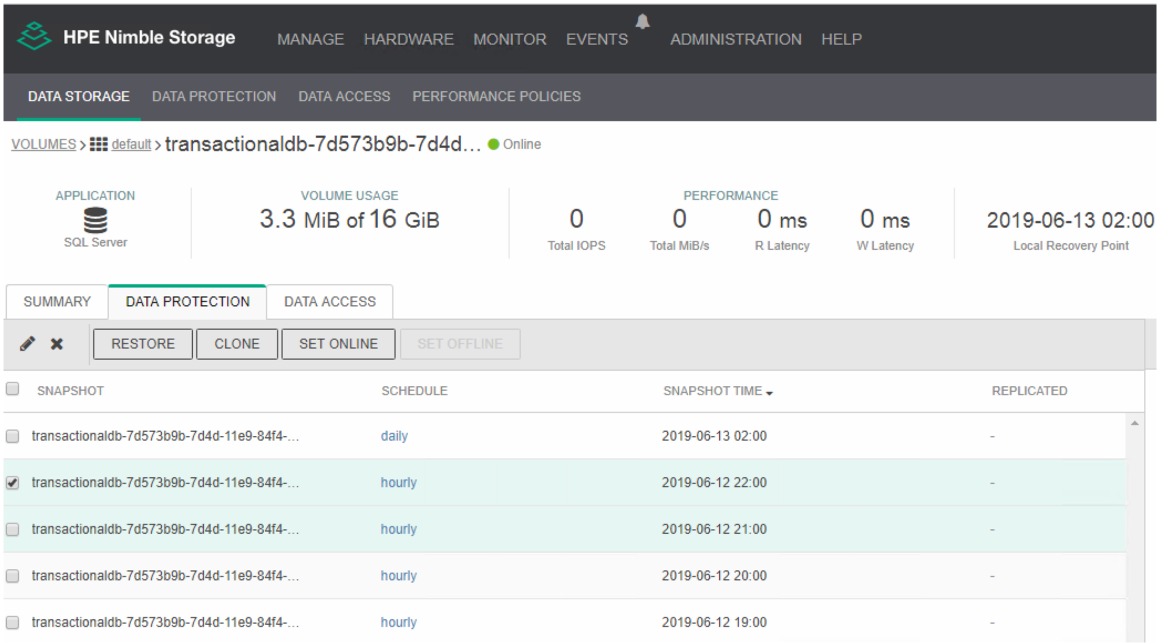
Confirm Scale Down

Scale down replication controller **mariadb1-847bd45cdc**?

Are you sure you want to scale **mariadb1-847bd45cdc** to 0 replicas? This will stop all pods for the replication controller.

Figure A3. RH-OCP console scaling down the replica set to zero

3. Locate the snapshot that will be used as in Figure A4.



VOLUMES > default > transactionaldb-7d573b9b-7d4d… Online

APPLICATION: SQL Server

VOLUME USAGE: 3.3 MiB of 16 GiB

PERFORMANCE: 0 Total IOPS; 0 Total MiB/s; 0 ms R Latency; 0 ms W Latency

2019-06-13 02:00 Local Recovery Point

	SNAPSHOT	SCHEDULE	SNAPSHOT TIME	REPLICATED
□	transactionaldb-7d573b9b-7d4d-11e9-84f4-…	daily	2019-06-13 02:00	-
☑	transactionaldb-7d573b9b-7d4d-11e9-84f4-…	hourly	2019-06-12 22:00	-
□	transactionaldb-7d573b9b-7d4d-11e9-84f4-…	hourly	2019-06-12 21:00	-
□	transactionaldb-7d573b9b-7d4d-11e9-84f4-…	hourly	2019-06-12 20:00	-
□	transactionaldb-7d573b9b-7d4d-11e9-84f4-…	hourly	2019-06-12 19:00	-

Figure A4. HPE Nimble Volume point in time restore tab

4. Proceed with the restore from the HPE Nimble Storage and raise the replica count back to 1. This will restore the data.
5. After the volume has been restored, set the replica count back to the number appropriate for the configuration.

It is possible to clone the snapshot and mount it to a different deployment and then recover the files from cloned snapshots into the running containers.

- Create a separate proxy deployment depending on the use case.
- Use `cloneOfPVC` and specify which snapshot to clone the PVC from and create a PVC.
- Attach the PVC to the proxy deployment.
- Copy files out and into containers using `cp` and `rsync.`
- Remove the volume from the proxy deployment and delete it.

For more details about how to clone the snapshot to a different proxy deployment, refer to https://community.hpe.com/t5/HPE-Storage-Tech-Insiders/Data-Protection-for-Containers-Part-II-Restore/ba-p/7019117#.XQiGbIgzY2w

Appendix B: HPE Nimble replication and protection template

For Nimble replication to work, a replication network and replication partners must be configured.

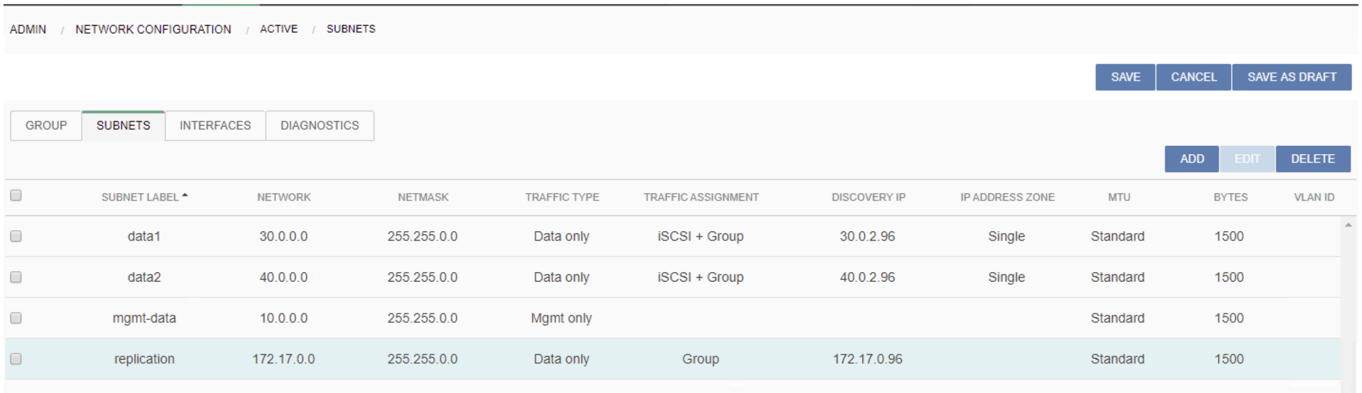
Configure the replication network

A dedicated interface is used for the replication network. The interfaces for the HPE Nimble Storage arrays are as follows:

- **eth0a:** Dedicated Management
- **eth0b:** Dedicated Replication
- **tg1a:** iSCSI Data1
- **tg1b:** iSCSI Data2

To configure replication, perform the following steps:

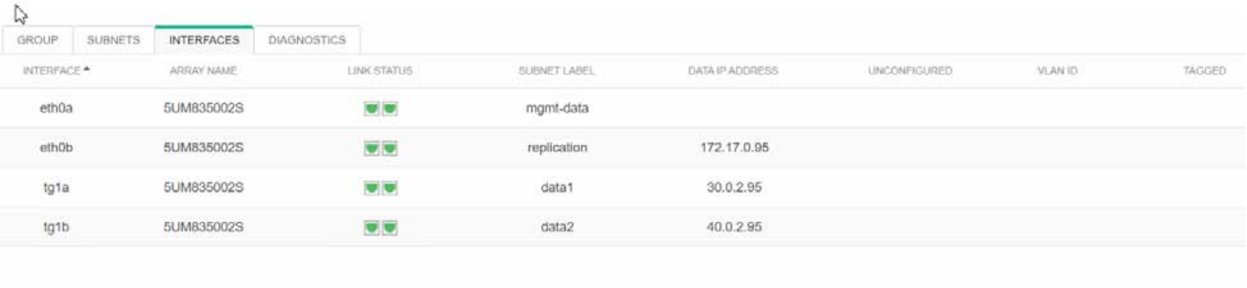
1. Log in to the HPE Nimble Storage Administration web GUI and navigate to **Administration**, then **Network**, and then select **Configure Active settings** and **subnets**.
2. Select Edit and then click **ADD** button.
3. Enter the replication network details and select the appropriate interface (eth0b) as in Figure B1.



□	SUBNET LABEL ▴	NETWORK	NETMASK	TRAFFIC TYPE	TRAFFIC ASSIGNMENT	DISCOVERY IP	IP ADDRESS ZONE	MTU	BYTES	VLAN ID
□	data1	30.0.0.0	255.255.0.0	Data only	iSCSI + Group	30.0.2.96	Single	Standard	1500	
□	data2	40.0.0.0	255.255.0.0	Data only	iSCSI + Group	40.0.2.96	Single	Standard	1500	
□	mgmt-data	10.0.0.0	255.255.0.0	Mgmt only				Standard	1500	
□	replication	172.17.0.0	255.255.0.0	Data only	Group	172.17.0.96		Standard	1500	

Figure B1. Replication network details

4. Once completed, the replication network link status displays green as in Figure B2.



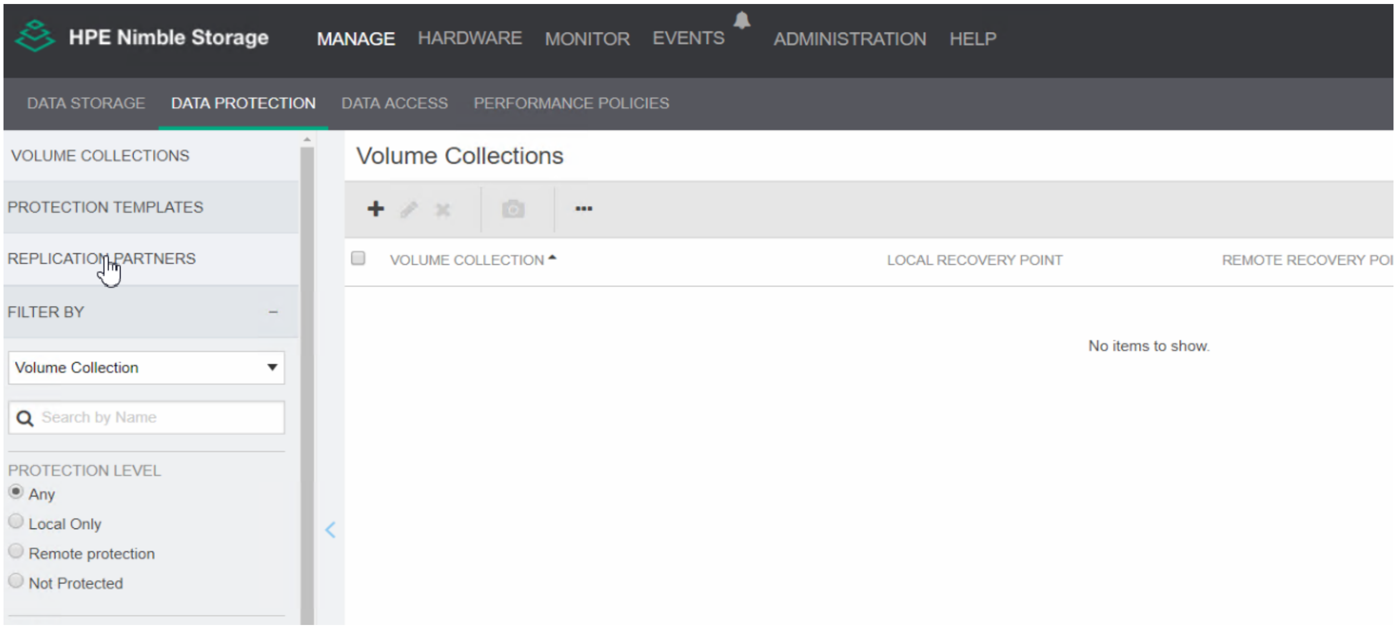
INTERFACE ▴	ARRAY NAME	LINK STATUS	SUBNET LABEL	DATA IP ADDRESS	UNCONFIGURED	VLAN ID	TAGGED
eth0a	5UM835002S	🟩🟩	mgmt-data				
eth0b	5UM835002S	🟩🟩	replication	172.17.0.95			
tg1a	5UM835002S	🟩🟩	data1	30.0.2.95			
tg1b	5UM835002S	🟩🟩	data2	40.0.2.95			

Figure B2. Successfully configured interfaces

Configure replication partners

In order to set up a replication partner, perform the following steps.

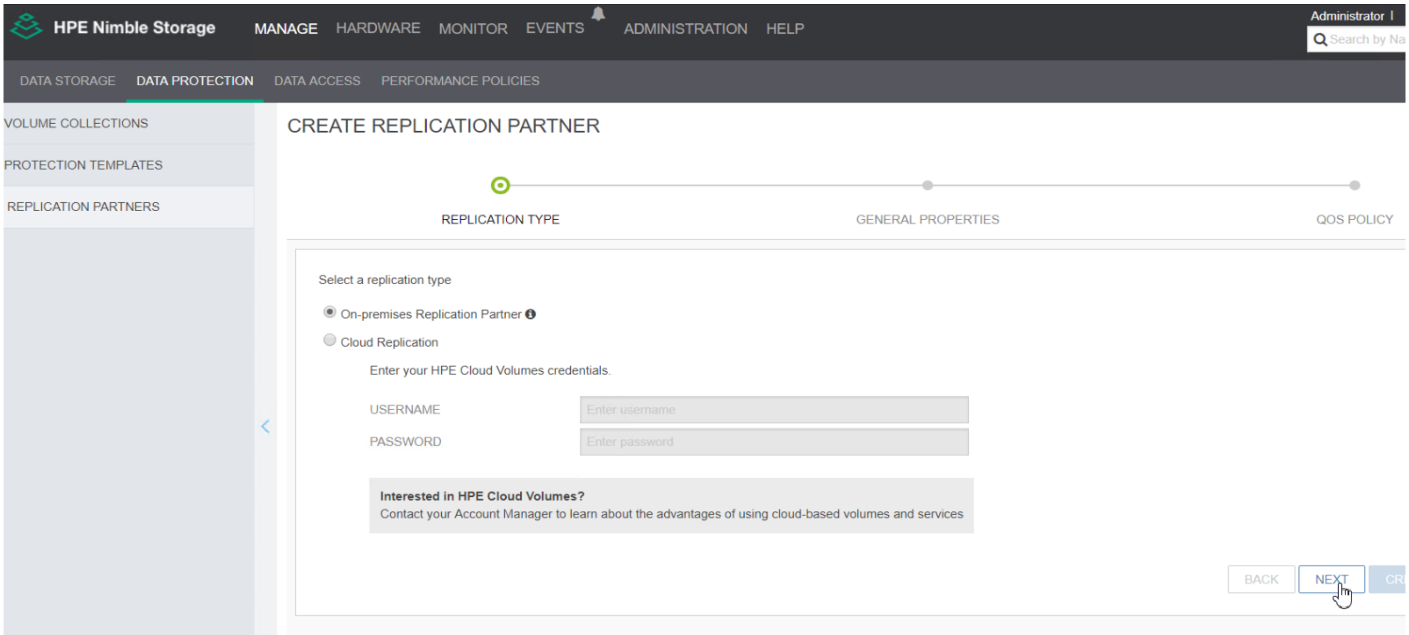
1. From the **MANAGE** tab, select **DATA PROTECTION** and then **REPLICATION PARTNERS** and then click **+** as in Figure B3.



No items to show.

Figure B3. The Data Protection screen

2. Select On-premises **REPLICATION PARTNERS** and then click **Next** as in Figure B4.



Select a replication type

● On-premises Replication Partner

○ Cloud Replication

Enter your HPE Cloud Volumes credentials.

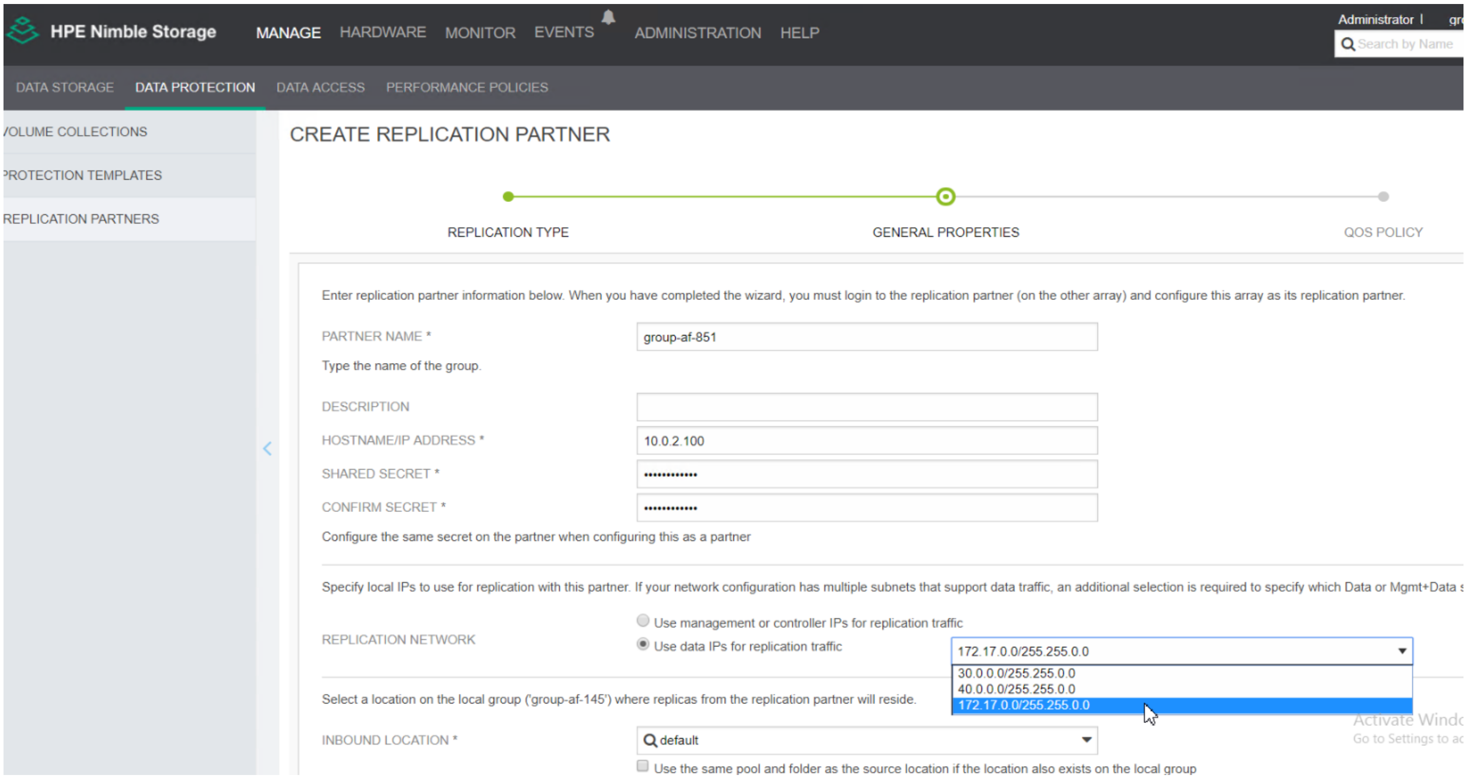
USERNAME: Enter username

PASSWORD: Enter password

Interested in HPE Cloud Volumes?
Contact your Account Manager to learn about the advantages of using cloud-based volumes and services

Figure B4. Creating the replication partner

3. Enter the downstream array and replication network details as in Figure B5.



CREATE REPLICATION PARTNER

Enter replication partner information below. When you have completed the wizard, you must login to the replication partner (on the other array) and configure this array as its replication partner.

PARTNER NAME * group-af-851

Type the name of the group.

DESCRIPTION

HOSTNAME/IP ADDRESS * 10.0.2.100

SHARED SECRET * ••••••••••••

CONFIRM SECRET * ••••••••••••

Configure the same secret on the partner when configuring this as a partner

Specify local IPs to use for replication with this partner. If your network configuration has multiple subnets that support data traffic, an additional selection is required to specify which Data or Mgmt+Data s

REPLICATION NETWORK
○ Use management or controller IPs for replication traffic
● Use data IPs for replication traffic 172.17.0.0/255.255.0.0
30.0.0.0/255.255.0.0
40.0.0.0/255.255.0.0
172.17.0.0/255.255.0.0

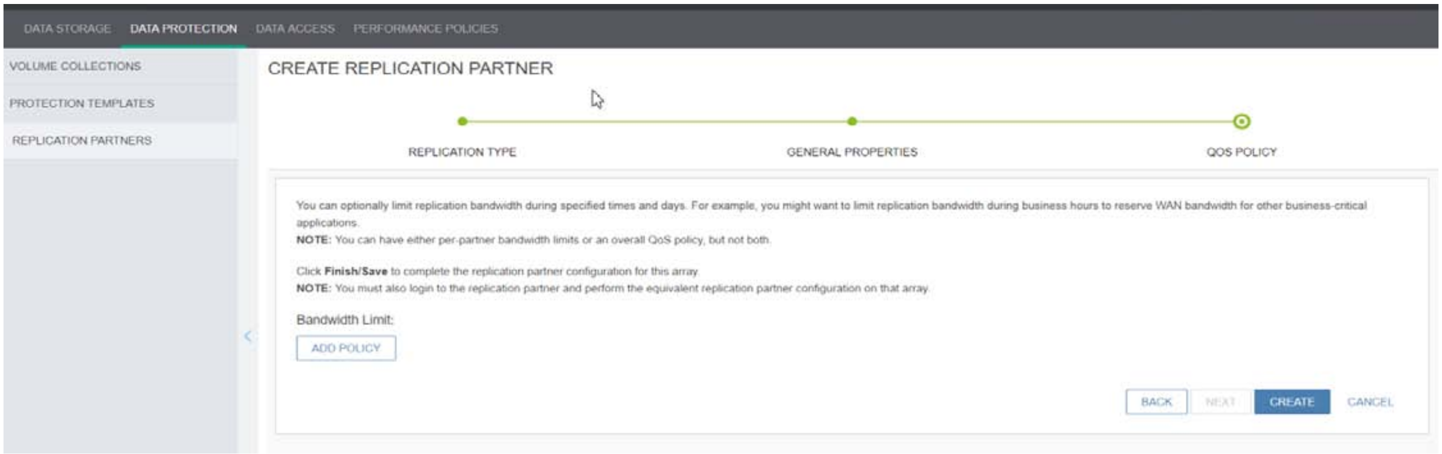
Select a location on the local group ('group-af-145') where replicas from the replication partner will reside.

INBOUND LOCATION * default

□ Use the same pool and folder as the source location if the location also exists on the local group

Figure B5. Partner configuration details

4. Click **CREATE** to set up the replication on the upstream array as in Figure B6.



CREATE REPLICATION PARTNER

You can optionally limit replication bandwidth during specified times and days. For example, you might want to limit replication bandwidth during business hours to reserve WAN bandwidth for other business-critical applications.
NOTE: You can have either per-partner bandwidth limits or an overall QoS policy, but not both.

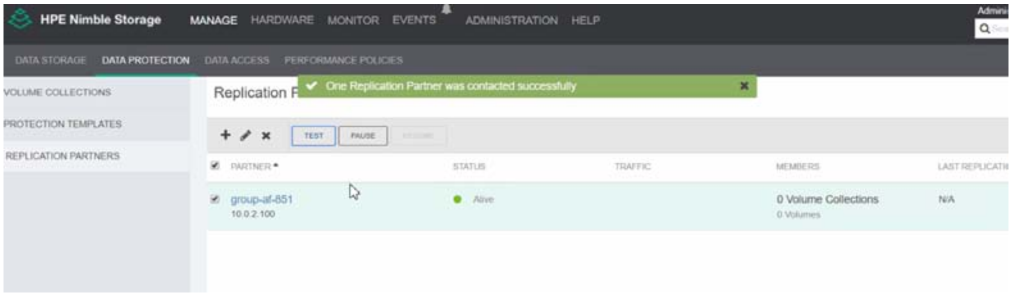
Click **Finish/Save** to complete the replication partner configuration for this array.
NOTE: You must also login to the replication partner and perform the equivalent replication partner configuration on that array.

Bandwidth Limit:

Figure B6. Final screen of the replication partner creation.

5. In order for the replication to occur between arrays, steps 1-4 should be performed on the downstream array.

6. Once the Replication Partners are set up on both the arrays, test replication by clicking the **TEST** button. Ensure that the status displays green as in Figure B7.



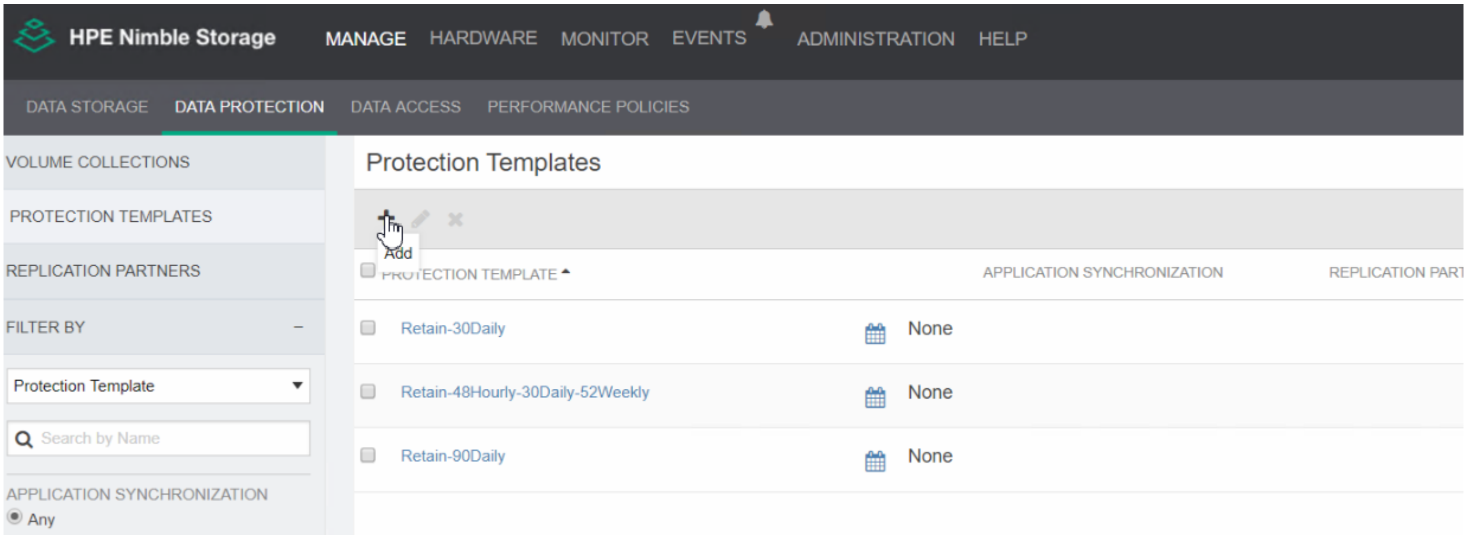
✓ One Replication Partner was contacted successfully

☑ PARTNER	STATUS	TRAFFIC	MEMBERS	LAST REPLICATI
☑ group-af-851 10.0.2.100	● Alive		0 Volume Collections 0 Volumes	N/A

Figure B7. Replication testing screen

Protection Templates

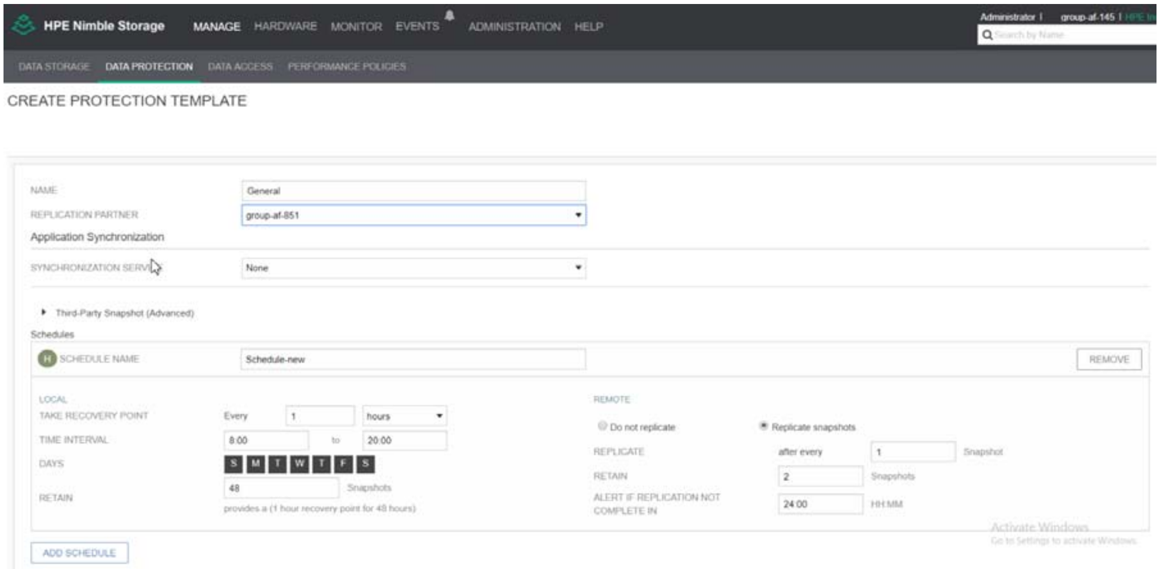
1. From the **MANAGE** tab select **DATA PROTECTION**, and then **FILTER BY Protection Templates,** and then click **+** as in Figure B8.



□ PROTECTION TEMPLATE	APPLICATION SYNCHRONIZATION	REPLICATION PART
□ Retain-30Daily	None	
□ Retain-48Hourly-30Daily-52Weekly	None	
□ Retain-90Daily	None	

Figure B8. Add a Protection Template

2. Give a name to the Protection Template and provide the requested details as in Figure B9.



CREATE PROTECTION TEMPLATE

NAME: General
REPLICATION PARTNER: group-af-851

Application Synchronization

SYNCHRONIZATION SERVICE: None

Third-Party Snapshot (Advanced)

Schedules

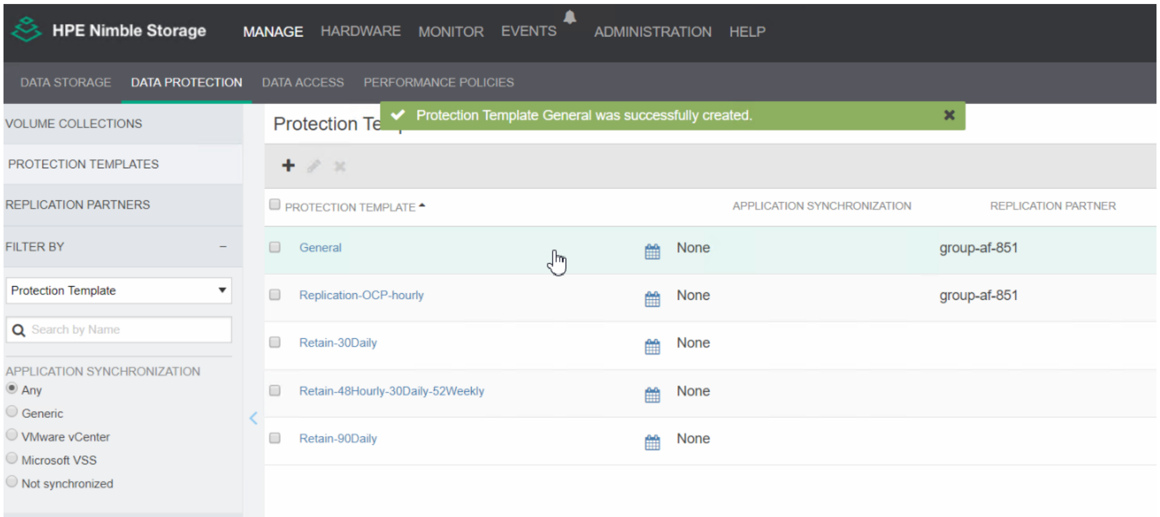
SCHEDULE NAME: Schedule-new

LOCAL
TAKE RECOVERY POINT: Every 1 hours
TIME INTERVAL: 8:00 to 20:00
DAYS: S M T W T F S
RETAIN: 48 Snapshots
provides a (1 hour recovery point for 48 hours)

REMOTE
○ Do not replicate ● Replicate snapshots
REPLICATE: after every 1 Snapshot
RETAIN: 2 Snapshots
ALERT IF REPLICATION NOT COMPLETE IN: 24:00 HH:MM

Figure B9. Protection Template details.

3. The newly created Protection Template will be listed under the PROTECTION TEMPLATE tab as in Figure B10.



✓ Protection Template General was successfully created.

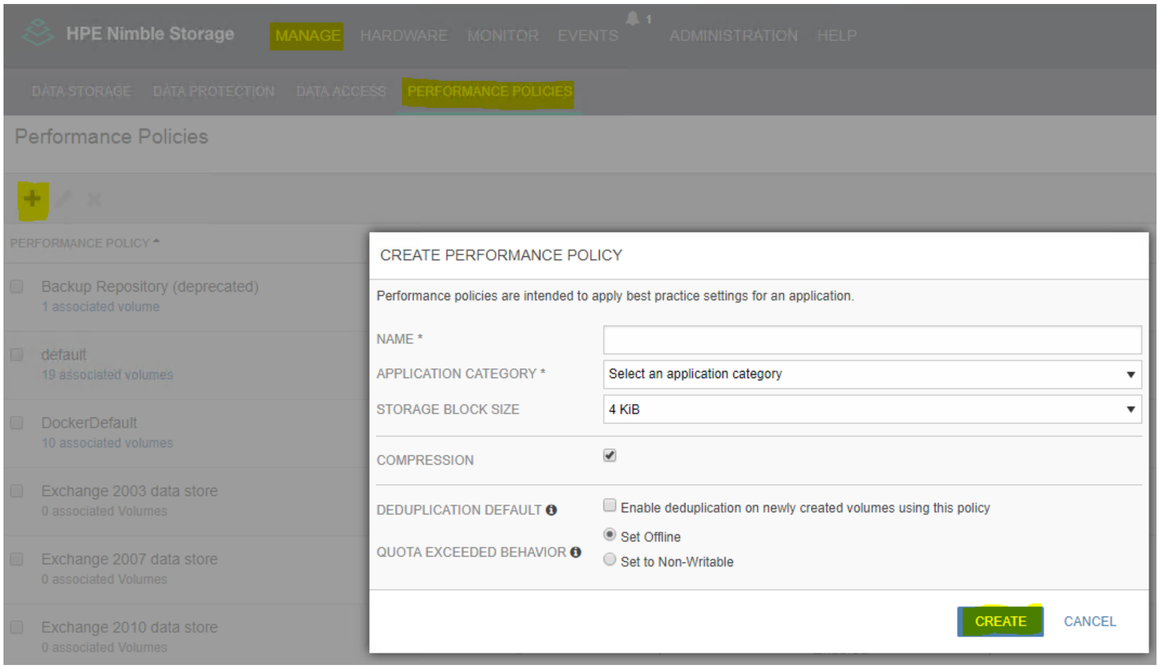
PROTECTION TEMPLATE	APPLICATION SYNCHRONIZATION	REPLICATION PARTNER
General	None	group-af-851
Replication-OCP-hourly	None	group-af-851
Retain-30Daily	None	
Retain-48Hourly-30Daily-52Weekly	None	
Retain-90Daily	None	

Figure B10. Newly created Protection Template in context

Performance Policies

While Performance Policies (PP) aren't related to data protection, they are highly relevant when creating distinguished storage classes which is part of the exercises below. NimbleOS ships a set of pre-defined Performance Policies that are refined, based on performance data gathered over the years and analyzed by our data scientists on the HPE InfoSight™. Performance Policies provides a set of defaults, such as block size, compression, deduplication and the type of behavior to adopt when the volume run out of space.

Log in to the HPE Nimble Storage Administration web GUI and navigate to **MANAGE, PERFORMANCE POLICIES** and click **+** to create the new Performance Policy as shown in Figure B11.



Backup Repository (deprecated)
1 associated volume

default
19 associated volumes

DockerDefault
10 associated volumes

Exchange 2003 data store
0 associated Volumes

Exchange 2007 data store
0 associated Volumes

Exchange 2010 data store
0 associated Volumes

CREATE PERFORMANCE POLICY

Performance policies are intended to apply best practice settings for an application.

NAME *

APPLICATION CATEGORY * Select an application category

STORAGE BLOCK SIZE 4 KiB

COMPRESSION ☑

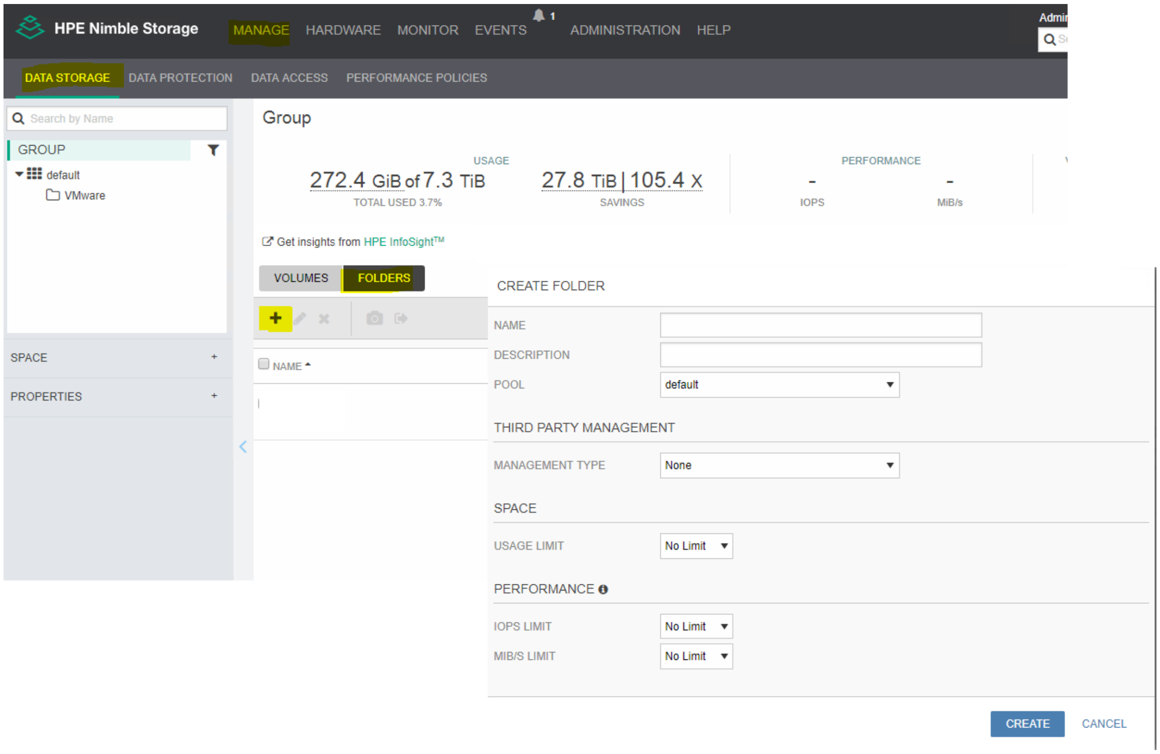
DEDUPLICATION DEFAULT ☐ Enable deduplication on newly created volumes using this policy

QUOTA EXCEEDED BEHAVIOR ● Set Offline ○ Set to Non-Writable

Figure B11. HPE Nimble Storage Performance Policies

Folders

NimbleOS provides a very simple construct to compartmentalize storage resources to better household with performance and capacity. The storage administrators create the folder and control the parameters. The folder may also be confined to a certain pool of storage in the Nimble group, such as hybrid flash or all-flash. The folder will be referenced in the StorageClass to distinguish the different data types to further help refine the characteristics needed for the application.



272.4 GiB of 7.3 TiB — TOTAL USED 3.7%

27.8 TiB | 105.4 X — SAVINGS

IOPS: -

MiB/s: -

CREATE FOLDER

NAME:

DESCRIPTION:

POOL: default

THIRD PARTY MANAGEMENT

MANAGEMENT TYPE: None

SPACE

USAGE LIMIT: No Limit

PERFORMANCE

IOPS LIMIT: No Limit

MIB/S LIMIT: No Limit

Figure B12. HPE Nimble Storage Folders

Appendix C: Container data backup using HPE Nimble Storage

Storage is a critical part of running stateful containers. When the Persistent Volume (PV) gets created dynamically, traditional agent-based backup software won't work natively with a container orchestrator in case of Red Hat OpenShift. Persistent Volume may be attached to any host at any given time in the cluster with different mount points. Also, having a backup agent hogging the mount path on the host will lead to unpredictable behavior and most certainly failed backups. Backup schemes need to be consumed as a data protection service from the underlying container aware storage infrastructure. HPE Nimble Storage provides these data protection service for Red Hat OpenShift by making use of Docker volume plugin and Kubernetes FlexVolume driver. This backup will be crash consistent. Consult specific product documentation for the procedure to take application consistent backup.

StorageClass

A StorageClass provides a way for administrators to describe the "classes" of storage they offer. The Storage class provisioner determines what volume plugin is used for provisioning Persistent Volumes. StorageClasses use provisioners that are specific to the storage platform provider to give Kubernetes access to the physical media being used.

Storage Classes allows a cluster administrator to define named classes with certain attributes, such as which provisioner to use, default Persistent Volume plugin parameters, Protection Template, performance Policy, folder and so on.

Create StorageClass resources

Create a StorageClass with the use of Protection Template, Performance Policy and a folder in Appendix B.

When the container requests for a PersistentVolumeClaim (PVC), it will use this storage class and create the Storage Volume. This enables the protection plan to schedule the snapshot and replicate the volume to a remote HPE Nimble Storage .

How to use StorageClasses

StorageClasses are the foundation of dynamic provisioning, allowing cluster administrators to define abstractions for the underlying storage platform. Users simply refer to a StorageClass by name in the PVC using the “storageClassName” parameter. Following are the examples of StorageClass and PVC used in testing.

StorageClass example:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
  name: general-sc
  annotations:
    storageclass.beta.kubernetes.io/is-default-class: ”true”
provisioner: hpe.com/nimble
parameters:
  description: ”Volume provisioned from default StorageClass”
  fsMode: ”0770”
  protectionTemplate: General
  perfPolicy: General
  folder: General
```

PVC example:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: general-pvc
  namespace: backup-plan
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: general-sc
```

Example from the Reference Configuration setup:

```
[root@buramaster01 db-templates]# oc get sc
NAME                PROVISIONER      AGE
general (default)   hpe.com/nimble   6d
transactionaldb     hpe.com/nimble   6d

[root@buramaster01 db-templates]# oc describe sc general
Name:                  general
IsDefaultClass:        Yes
Annotations:           storageclass.beta.kubernetes.io/is-default-class=true
Provisioner:           hpe.com/nimble
Parameters:            description=Volume provisioned by HPE Nimble Storage Kube Storage Controller from default
StorageClass,fsMode=0770
AllowVolumeExpansion:  <unset>
MountOptions:          <none>
ReclaimPolicy:         Delete
VolumeBindingMode:     Immediate
Events:                <none>

[root@buramaster01 db-templates]# oc get pvc
```

```
NAME            STATUS    VOLUME                                                   CAPACITY   ACCESS MODES
STORAGECLASS      AGE
default-claim   Bound     general-9818c044-7d47-11e9-84f4-566f57580060           32Gi       RWO
general           6d


[root@buramaster01 db-templates]# oc describe pvc default-claim
Name:          default-claim
Namespace:     default
StorageClass:  general
Status:        Bound
Volume:        general-9818c044-7d47-11e9-84f4-566f57580060
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed=yes
               pv.kubernetes.io/bound-by-controller=yes
               volume.beta.kubernetes.io/storage-provisioner=hpe.com/nimble
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      32Gi
Access Modes:  RWO
Events:        <none>
```

Resources and additional links

HPE Information Library, http://h17007.www1.hpe.com/us/en/enterprise/integrated-systems/info-library/index.aspx?cat=convergedsystems&subcat=cs750

HPE Reference Architectures, hpe.com/info/ra

HPE Servers, hpe.com/servers

HPE Storage, hpe.com/storage

HPE Networking, hpe.com/networking

HPE Technology Consulting Services, hpe.com/us/en/services/consulting.html

Deployment architecture, https://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=a00056101enw

To help us improve our documents, please provide feedback at hpe.com/contact/feedback.

Sign up for updates

Version 1, July 2019