



# **Solution Validation and Use Cases**

## **NetApp Solutions**

NetApp  
May 17, 2024

# Table of Contents

- Solution Validation and Use Cases: Red Hat OpenShift with NetApp ..... 1
  - Deploy a Jenkins CI/CD Pipeline with Persistent Storage: Red Hat OpenShift with NetApp ..... 1
  - Configure Multi-tenancy on Red Hat OpenShift with NetApp ONTAP ..... 11
  - Red Hat OpenShift Virtualization with NetApp ONTAP ..... 31
  - Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp ..... 85

# Solution Validation and Use Cases: Red Hat OpenShift with NetApp

The examples provided on this page are solution validations and use cases for Red Hat OpenShift with NetApp.

- [Deploy a Jenkins CI/CD Pipeline with Persistent Storage](#)
- [Configure Multitenancy on Red Hat OpenShift with NetApp](#)
- [Red Hat OpenShift Virtualization with NetApp ONTAP](#)
- [Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp](#)

## Deploy a Jenkins CI/CD Pipeline with Persistent Storage: Red Hat OpenShift with NetApp

This section provides the steps to deploy a continuous integration/continuous delivery or deployment (CI/CD) pipeline with Jenkins to validate solution operation.

### Create the resources required for Jenkins deployment

To create the resources required for deploying the Jenkins application, complete the following steps:

1. Create a new project named Jenkins.

# Create Project

Name \*

Display Name

Description

Cancel

Create

2. In this example, we deployed Jenkins with persistent storage. To support the Jenkins build, create the PVC. Navigate to Storage > Persistent Volume Claims and click Create Persistent Volume Claim. Select the storage class that was created, make sure that the Persistent Volume Claim Name is jenkins, select the appropriate size and access mode, and then click Create.

## Create Persistent Volume Claim

[Edit YAML](#)

### Storage Class

▼

Storage class for the new claim.

### Persistent Volume Claim Name \*

A unique name for the storage claim within the project.

### Access Mode \*

Single User (RWO)  Shared Access (RWX)  Read Only (ROX)

Permissions to the mounted drive.

### Size \*

▼

Desired storage capacity.

Use label selectors to request storage

Use label selectors to define how storage is created.

## Deploy Jenkins with Persistent Storage

To deploy Jenkins with persistent storage, complete the following steps:

1. In the upper left corner, change the role from Administrator to Developer. Click +Add and select From Catalog. In the Filter by Keyword bar, search for jenkins. Select Jenkins Service with Persistent Storage.

## Developer Catalog

Add shared apps, services, or source-to-image builders to your project from the Developer Catalog. Cluster admins can install additional apps which will show up here automatical

The screenshot shows the Developer Catalog interface. On the left is a sidebar with navigation options: All Items, Languages, Databases, Middleware, CI/CD, Other, and Type. Under 'Type', there are checkboxes for 'Operator Backed (0)', 'Helm Charts (0)', 'Builder Image (0)', 'Template (4)', and 'Service Class (0)'. The main area is titled 'All Items' and contains a search box with 'jenkins' and a 'Group By: None' dropdown. Below this, four Jenkins templates are displayed as cards. Each card includes a Jenkins logo, the name 'Jenkins', the provider 'Red Hat, Inc.', and a brief description. The first two cards mention 'persistent storage', while the last two mention 'without persistent storage' and include a warning about data loss.

2. Click Instantiate Template.



### Jenkins

Provided by Red Hat, Inc.



Instantiate Template

#### Provider

Red Hat, Inc.

#### Support

[Get support](#)

#### Created At

May 26, 3:58 am

#### Description

Jenkins service, with persistent storage.

NOTE: You must have persistent volumes available in your cluster to use this template.

#### Documentation

[https://docs.okd.io/latest/using\\_images/other\\_images/jenkins.html](https://docs.okd.io/latest/using_images/other_images/jenkins.html)

3. By default, the details for the Jenkins application are populated. Based on your requirements, modify the parameters and click Create. This process creates all the required resources for supporting Jenkins on

## Instantiate Template

**Namespace \***  
jenkins

**Jenkins Service Name**  
jenkins  
The name of the OpenShift Service exposed for the Jenkins container.

**Jenkins JNLP Service Name**  
jenkins-jnlp  
The name of the service used for master/slave communication.

**Enable OAuth in Jenkins**  
true  
Whether to enable OAuth OpenShift integration. If false, the static account 'admin' will be initialized with the password 'password'.

**Memory Limit**  
1Gi  
Maximum amount of memory the container can use.

**Volume Capacity \***  
50Gi  
Volume space available for data, e.g. 512Mi, 2Gi.

**Jenkins ImageStream Namespace**  
openshift  
The OpenShift Namespace where the Jenkins ImageStream resides.

**Disable memory intensive administrative monitors**  
false  
Whether to perform memory intensive, possibly slow, synchronization with the Jenkins Update Center on start. If true, the Jenkins core update monitor and site warnings monitor are disabled.

**Jenkins ImageStreamTag**  
jenkins:2  
Name of the ImageStreamTag to be used for the Jenkins image.

**Fatal Error Log File**  
false  
When a fatal error occurs, an error log is created with information and the state obtained at the time of the fatal error.

**Allows use of Jenkins Update Center repository with invalid SSL certificate**  
false  
Whether to allow use of a Jenkins Update Center that uses invalid certificate (self-signed, unknown CA). If any value other than 'false', certificate check is bypassed. By default, certificate check is enforced.

[Create](#) [Cancel](#)

 **Jenkins**  
INSTANT-APP JENKINS  
[View documentation](#) [Get support](#)

Jenkins service, with persistent storage.

NOTE: You must have persistent volumes available in your cluster to use this template.

The following resources will be created:

- DeploymentConfig
- PersistentVolumeClaim
- RoleBinding
- Route
- Service
- ServiceAccount

4. The Jenkins pods take approximately 10 to 12 minutes to enter the Ready state.

## Pods

Create Pod Filter by name...

1 Running 0 Pending 0 Terminating 0 CrashLoopBackOff 1 Completed 0 Failed 0 Unknown

Select all filters 1 of 2 Items

Name ↑	Namespace ↓	Status ↓	Ready ↓	Owner ↓	Memory ↓	CPU ↓	
jenkins-1-c77n9	jenkins	Running	1/1	jenkins-1	-	0.004 cores	⋮

5. After the pods are instantiated, navigate to Networking > Routes. To open the Jenkins webpage, click the URL provided for the jenkins route.

## Routes

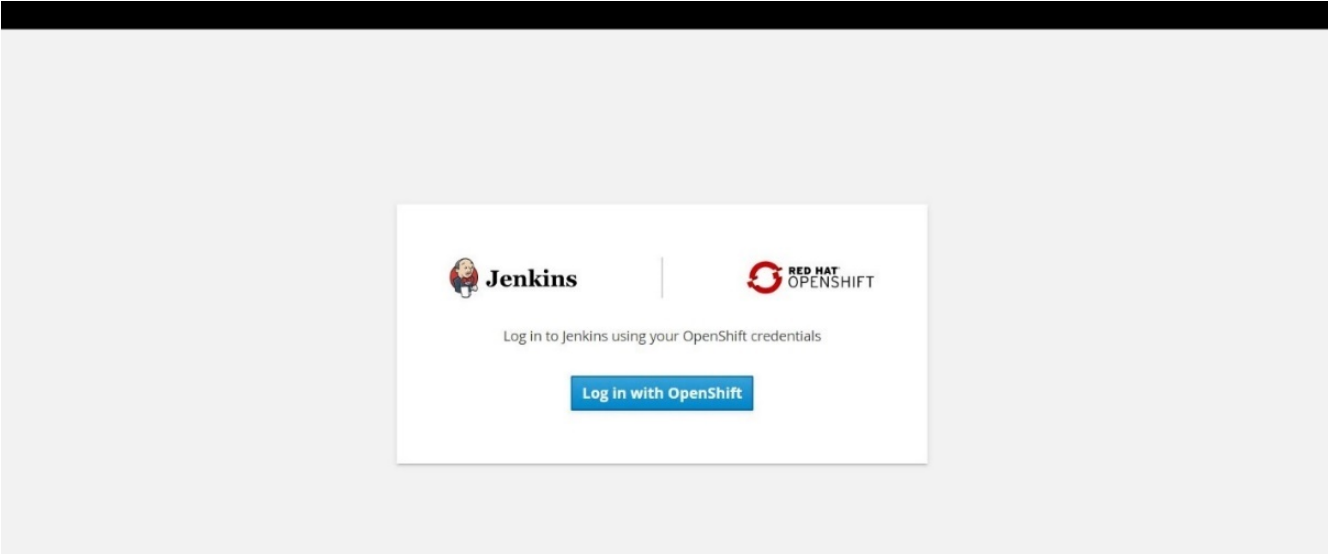
Create Route Filter by name...

1 Accepted 0 Rejected 0 Pending Select all filters 1 Item

Name ↓	Namespace ↓	Status	Location ↓	Service ↓	
jenkins	jenkins	Accepted	<a href="https://jenkins-jenkins.apps.rhv-ocp-cluster.cie.netapp.com">https://jenkins-jenkins.apps.rhv-ocp-cluster.cie.netapp.com</a>	jenkins	⋮

6. Because OpenShift OAuth was used while creating the Jenkins app, click Log in with OpenShift.





7. Authorize the Jenkins service account to access the OpenShift users.

## Authorize Access

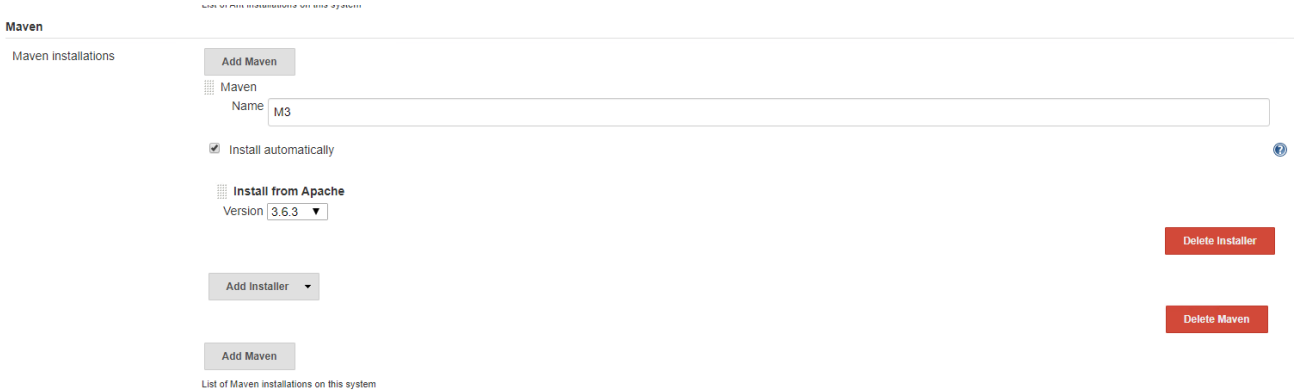
Service account `jenkins` in project `jenkins` is requesting permission to access your account (`kube:admin`)

### Requested permissions

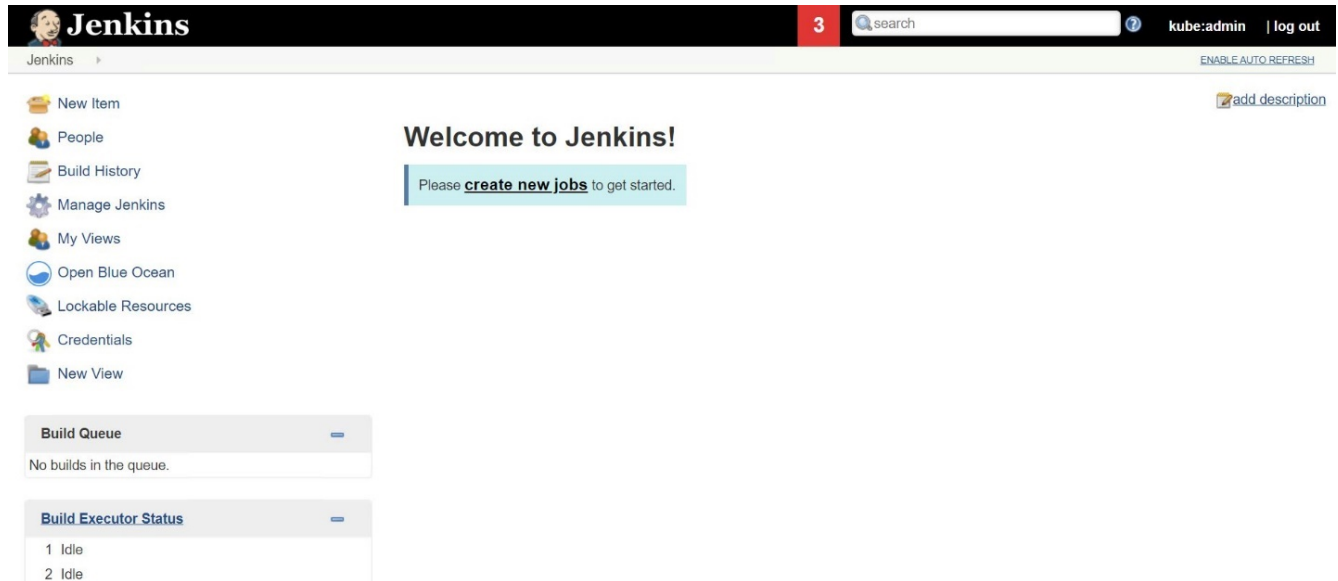
- user:info**  
Read-only access to your user information (including username, identities, and group membership)
- user:check-access**  
Read-only access to view your privileges (for example, "can I create builds?")

You will be redirected to <https://jenkins-jenkins.apps.rhv-ocp-cluster.cie.netapp.com/securityRealm/finishLogin>

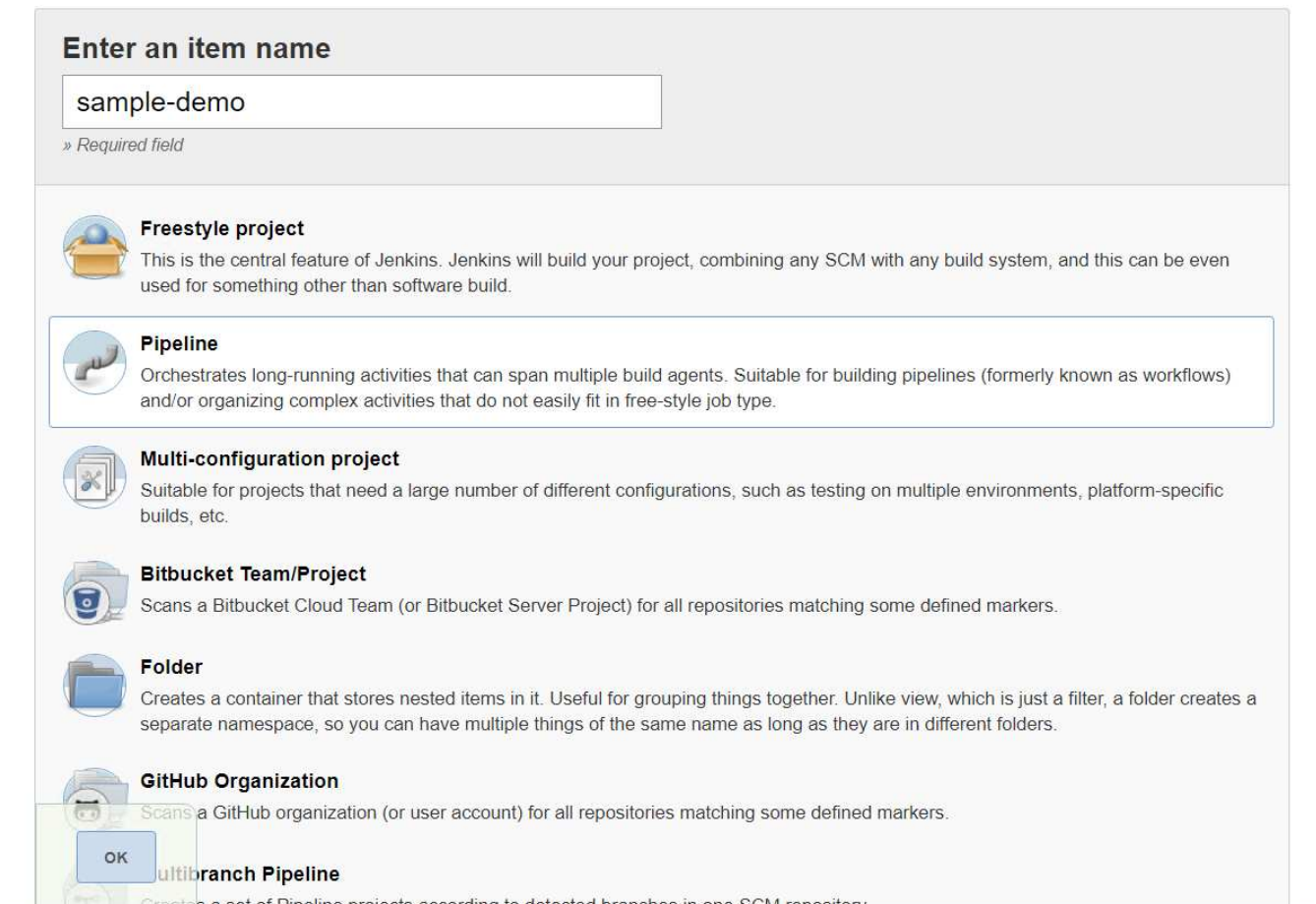
8. The Jenkins welcome page is displayed. Because we are using a Maven build, complete the Maven installation first. Navigate to Manage Jenkins > Global Tool Configuration, and then, in the Maven subhead, click Add Maven. Enter the name of your choice and make sure that the Install Automatically option is selected. Click Save.



9. You can now create a pipeline to demonstrate the CI/CD workflow. On the home page, click Create New Jobs or New Item from the left-hand menu.



10. On the Create Item page, enter the name of your choice, select Pipeline, and click Ok.



11. Select the Pipeline tab. From the Try Sample Pipeline drop-down menu, select Github + Maven. The code is automatically populated. Click Save.

General Build Triggers Advanced Project Options **Pipeline**

Advanced...

### Pipeline

Definition Pipeline script

Script

```
1 node {
2   def mvnHome
3   stage('Preparation') { // for display purposes
4     // Get some code from a GitHub repository
5     git 'https://github.com/jglick/simple-maven-project-with-tests.git'
6     // Get the Maven tool.
7     // ** NOTE: This 'M3' Maven tool must be configured
8     // **       in the global configuration.
9     mvnHome = tool 'M3'
10  }
11  stage('Build') {
12    // Run the maven build
13    withEnv(["MVN_HOME=$mvnHome"]) {
14      if (isUnix()) {
15        sh "$MVN_HOME/bin/mvn" -Dmaven.test.failure.ignore clean package'
16      } else {
17        bat("/%MVN_HOME%\bin\mvn" -Dmaven.test.failure.ignore clean package/)
```

GitHub + Maven

Use Groovy Sandbox

[Pipeline Syntax](#)

Save Apply

12. Click Build Now to trigger the development through the preparation, build, and testing phase. It can take several minutes to complete the whole build process and display the results of the build.

- Back to Dashboard
- Status
- Changes
- Build Now
- Delete Pipeline
- Configure
- Full Stage View
- Open Blue Ocean
- Rename
- Pipeline Syntax

## Pipeline sample-demo

[Last Successful Artifacts](#)

[simple-maven-project-with-tests-1.0-SNAPSHOT.jar](#) 1.71 KB [view](#)

[Recent Changes](#)

**Build History** [trend](#)

find

**#1** May 27, 2020 3:53 PM

[Atom feed for all](#) [Atom feed for failures](#)

### Stage View

Average stage times:  
(Average full run time: ~7s)

	Preparation	Build	Results
Average	2s	4s	69ms
#1	2s	4s	69ms

#1 May 27 08:53 No Changes

[Latest Test Result \(no failures\)](#)

### Permalinks

- [Last build \(#1\), 1 min 23 sec ago](#)
- [Last stable build \(#1\), 1 min 23 sec ago](#)
- [Last successful build \(#1\), 1 min 23 sec ago](#)
- [Last completed build \(#1\), 1 min 23 sec ago](#)

13. Whenever there are any code changes, the pipeline can be rebuilt to patch the new version of software enabling continuous integration and continuous delivery. Click Recent Changes to track the changes from the previous version.

- Back to Dashboard
- Status
- Changes
- Build Now
- Delete Pipeline
- Configure
- Full Stage View
- Open Blue Ocean
- Rename
- Pipeline Syntax

## Pipeline sample-demo

[Last Successful Artifacts](#)  
[simple-maven-project-with-tests-1.0-SNAPSHOT.jar](#) 1.71 KB [view](#)

[Recent Changes](#)

**Build History** [trend](#) ⇌

find

- #2 May 27, 2020 3:56 PM
- #1 May 27, 2020 3:53 PM

[Atom feed for all](#) [Atom feed for failures](#)

### Stage View

Average stage times:  
(Average full run time: ~6s)

	Preparation	Build	Results
<b>#2</b>	2s	4s	86ms
May 27 08:56 No Changes	1s	4s	104ms
<b>#1</b>	2s	4s	69ms
May 27 08:53 No Changes			

[Latest Test Result](#) (no failures)

### Permalinks

- [Last build \(#2\), 19 sec ago](#)
- [Last stable build \(#2\), 19 sec ago](#)
- [Last successful build \(#2\), 19 sec ago](#)
- [Last completed build \(#2\), 19 sec ago](#)

# Configure Multi-tenancy on Red Hat OpenShift with NetApp ONTAP

## Configuring multitenancy on Red Hat OpenShift with NetApp

Many organizations that run multiple applications or workloads on containers tend to deploy one Red Hat OpenShift cluster per application or workload. This allows them to implement strict isolation for the application or workload, optimize performance, and reduce security vulnerabilities. However, deploying a separate Red Hat OpenShift cluster for each application poses its own set of problems. It increases operational overhead having to monitor and manage each cluster on its own, increases cost owing to dedicated resources for different applications, and hinders efficient scalability.

To overcome these problems, one can consider running all the applications or workloads in a single Red Hat OpenShift cluster. But in such an architecture, resource isolation and application security vulnerabilities are one of the major challenges. Any security vulnerability in one workload could naturally spill over into another workload, thus increasing the impact zone. In addition, any abrupt uncontrolled resource utilization by one application can affect the performance of another application, because there is no resource allocation policy by default.

Therefore, organizations look out for solutions that pick up the best in both worlds, for example, by allowing them to run all their workloads in a single cluster and yet offering the benefits of a dedicated cluster for each workload.

One such effective solution is to configure multitenancy on Red Hat OpenShift. Multitenancy is an architecture that allows multiple tenants to coexist on the same cluster with proper isolation of resources, security, and so on. In this context, a tenant can be viewed as a subset of the cluster resources that are configured to be used by a particular group of users for an exclusive purpose. Configuring multitenancy on a Red Hat OpenShift cluster provides the following advantages:

- A reduction in CapEx and OpEx by allowing cluster resources to be shared
- Lower operational and management overhead
- Securing the workloads from cross-contamination of security breaches
- Protection of workloads from unexpected performance degradation due to resource contention

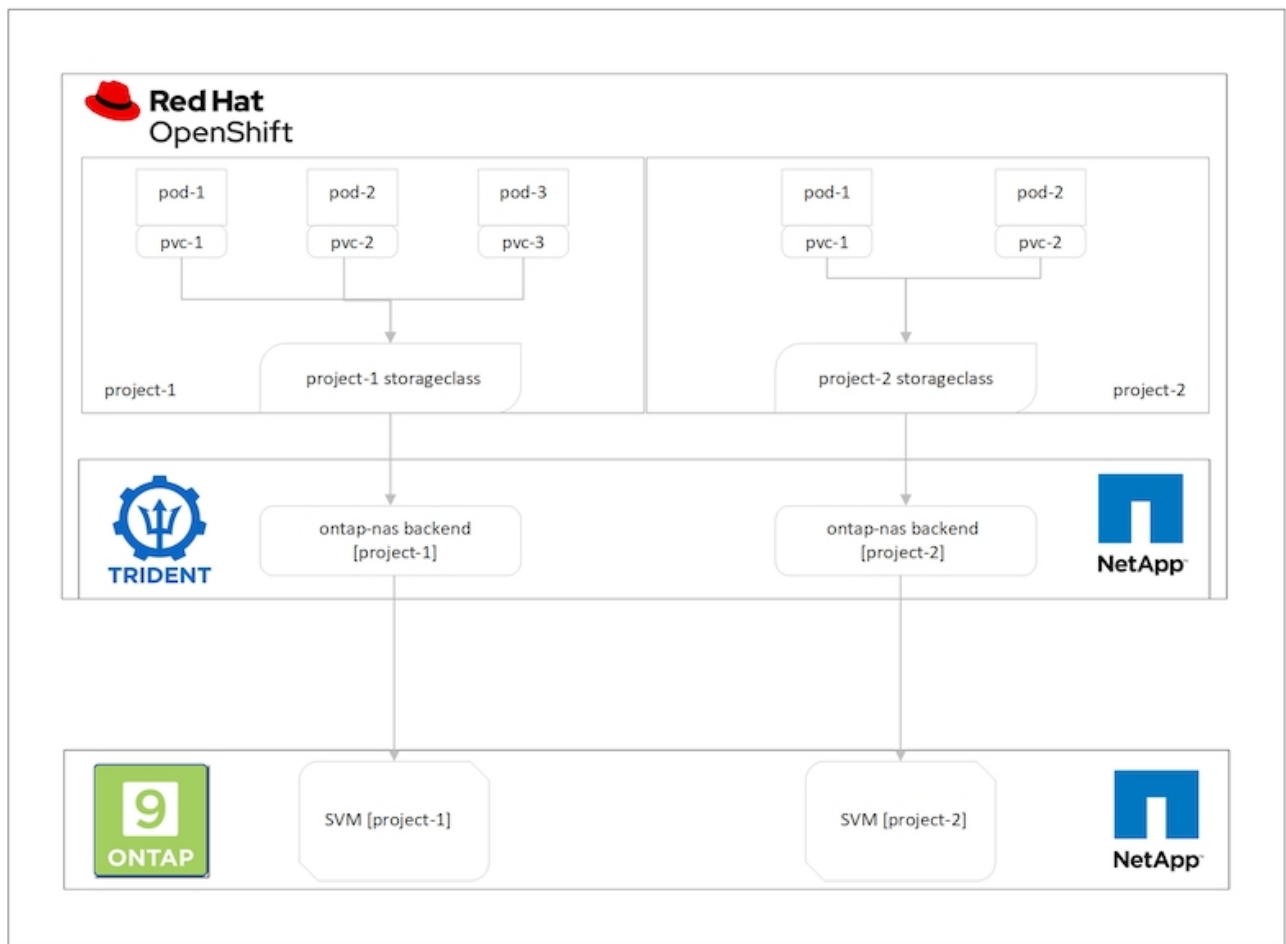
For a fully realized multitenant OpenShift cluster, quotas and restrictions must be configured for cluster resources belonging to different resource buckets: compute, storage, networking, security, and so on. Although we cover certain aspects of all the resource buckets in this solution, we focus on best practices for isolating and securing the data served or consumed by multiple workloads on the same Red Hat OpenShift cluster by configuring multitenancy on storage resources that are dynamically allocated by Astra Trident backed by NetApp ONTAP.

## Architecture

Although Red Hat OpenShift and Astra Trident backed by NetApp ONTAP do not provide isolation between workloads by default, they offer a wide range of features that can be used to configure multitenancy. To better understand designing a multitenant solution on a Red Hat OpenShift cluster with Astra Trident backed by NetApp ONTAP, let us consider an example with a set of requirements and outline the configuration around it.

Let us assume that an organization runs two of its workloads on a Red Hat OpenShift cluster as part of two projects that two different teams are working on. The data for these workloads reside on PVCs that are dynamically provisioned by Astra Trident on a NetApp ONTAP NAS backend. The organization has a requirement to design a multitenant solution for these two workloads and isolate the resources used for these projects to make sure that security and performance is maintained, primarily focused on the data that serves those applications.

The following figure depicts the multitenant solution on a Red Hat OpenShift cluster with Astra Trident backed by NetApp ONTAP.



## Technology requirements

1. NetApp ONTAP storage cluster
2. Red Hat OpenShift cluster
3. Astra Trident

## Red Hat OpenShift – Cluster resources

From the Red Hat OpenShift cluster point of view, the top-level resource to start with is the project. An OpenShift project can be viewed as a cluster resource that divides the whole OpenShift cluster into multiple virtual clusters. Therefore, isolation at project level provides a base for configuring multitenancy.

Next up is to configure RBAC in the cluster. The best practice is to have all the developers working on a single project or workload configured into a single user group in the Identity Provider (IdP). Red Hat OpenShift allows IdP integration and user group synchronization thus allowing the users and groups from the IdP to be imported into the cluster. This helps the cluster administrators to segregate access of the cluster resources dedicated to a project to a user group or groups working on that project, thereby restricting unauthorized access to any cluster resources. To learn more about IdP integration with Red Hat OpenShift, see the documentation [here](#).

## NetApp ONTAP

It is important to isolate the shared storage serving as a persistent storage provider for a Red Hat OpenShift cluster to make sure that the volumes created on the storage for each project appear to the hosts as if they are

created on separate storage. To do this, create as many SVMs (storage virtual machines) on NetApp ONTAP as there are projects or workloads, and dedicate each SVM to a workload.

## Astra Trident

After you have different SVMs for different projects created on NetApp ONTAP, you must map each SVM to a different Trident backend. The backend configuration on Trident drives the allocation of persistent storage to OpenShift cluster resources, and it requires the details of the SVM to be mapped to. This should be the protocol driver for the backend at the minimum. Optionally, it allows you to define how the volumes are provisioned on the storage and to set limits for the size of volumes or usage of aggregates and so on. Details concerning the definition of the Trident backends can be found [here](#).

## Red Hat OpenShift – storage resources

After configuring the Trident backends, the next step is to configure StorageClasses. Configure as many storage classes as there are backends, providing each storage class access to spin up volumes only on one backend. We can map the StorageClass to a particular Trident backend by using the `storagePools` parameter while defining the storage class. The details to define a storage class can be found [here](#). Thus, there is a one-to-one mapping from StorageClass to Trident backend which points back to one SVM. This ensures that all storage claims via the StorageClass assigned to that project are served by the SVM dedicated to that project only.

Because storage classes are not namespaced resources, how do we ensure that storage claims to storage class of one project by pods in another namespace or project gets rejected? The answer is to use ResourceQuotas. ResourceQuotas are objects that control the total usage of resources per project. It can limit the number as well as the total amount of resources that can be consumed by objects in the project. Almost all the resources of a project can be limited using ResourceQuotas and using this efficiently can help organizations cut cost and outages due to overprovisioning or overconsumption of resources. Refer to the documentation [here](#) for more information.

For this use case, we need to limit the pods in a particular project from claiming storage from storage classes that are not dedicated to their project. To do that, we need to limit the persistent volume claims for other storage classes by setting `<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims` to 0. In addition, a cluster administrator must ensure that the developers in a project should not have access to modify the ResourceQuotas.

## Configuration

For any multitenant solution, no user can have access to more cluster resources than is required. So, the entire set of resources that are to be configured as part of the multitenancy configuration is divided between cluster-admin, storage-admin, and developers working on each project.

The following table outlines the different tasks to be performed by different users:



Role	Tasks
<b>Cluster-admin</b>	Create projects for different applications or workloads
	Create ClusterRoles and RoleBindings for storage-admin
	Create Roles and RoleBindings for developers assigning access to specific projects
	[Optional] Configure projects to schedule pods on specific nodes
<b>Storage-admin</b>	Create SVMs on NetApp ONTAP
	Create Trident backends
	Create StorageClasses
	Create storage ResourceQuotas
<b>Developers</b>	Validate access to create or patch PVCs or pods in assigned project
	Validate access to create or patch PVCs or pods in another project
	Validate access to view or edit Projects, ResourceQuotas, and StorageClasses

## Configuration

Following are the prerequisites for Configuring Multitenancy on Red Hat OpenShift with NetApp.

### Prerequisites

- NetApp ONTAP cluster
- Red Hat OpenShift cluster
- Trident installed on the cluster
- Admin workstation with tridentctl and oc tools installed and added to \$PATH
- Admin access to ONTAP
- Cluster-admin access to OpenShift cluster
- Cluster is integrated with Identity Provider
- Identity provider is configured to efficiently distinguish between users in different teams

### Configuration: cluster-admin tasks

The following tasks are performed by the Red Hat OpenShift cluster-admin:

1. Log into Red Hat OpenShift cluster as the cluster-admin.
2. Create two projects corresponding to different projects.

```
oc create namespace project-1
oc create namespace project-2
```

### 3. Create the developer role for project-1.

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-1
  name: developer-project-1
rules:
  - verbs:
    - '*'
    apiGroups:
      - apps
      - batch
      - autoscaling
      - extensions
      - networking.k8s.io
      - policy
      - apps.openshift.io
      - build.openshift.io
      - image.openshift.io
      - ingress.operator.openshift.io
      - route.openshift.io
      - snapshot.storage.k8s.io
      - template.openshift.io
    resources:
      - '*'
  - verbs:
    - '*'
    apiGroups:
      - ''
    resources:
      - bindings
      - configmaps
      - endpoints
      - events
      - persistentvolumeclaims
      - pods
      - pods/log
      - pods/attach
      - podtemplates
      - replicationcontrollers
```

```

- services
- limitranges
- namespaces
- componentstatuses
- nodes
- verbs:
  - '*'
apiGroups:
- trident.netapp.io
resources:
- tridentnsnapshots
EOF

```



The role definition provided in this section is just an example. Developer roles must be defined based on end-user requirements.

4. Similarly, create developer roles for project-2.
5. All OpenShift and NetApp storage resources are usually managed by a storage admin. Access for storage administrators is controlled by the trident operator role that is created when Trident is installed. In addition to this, the storage admin also requires access to ResourceQuotas to control how storage is consumed.
6. Create a role for managing ResourceQuotas in all projects in the cluster to attach it to storage admin.

```

cat << EOF | oc create -f -
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: resource-quotas-role
rules:
- verbs:
  - '*'
  apiGroups:
  - ''
  resources:
  - resourcequotas
- verbs:
  - '*'
  apiGroups:
  - quota.openshift.io
  resources:
  - '*'
EOF

```

7. Make sure that the cluster is integrated with the organization's identity provider and that user groups are synchronized with cluster groups. The following example shows that the identity provider has been integrated with the cluster and synchronized with the user groups.

```
$ oc get groups
NAME                                USERS
ocp-netapp-storage-admins          ocp-netapp-storage-admin
ocp-project-1                       ocp-project-1-user
ocp-project-2                       ocp-project-2-user
```

## 8. Configure ClusterRoleBindings for storage admins.

```
cat << EOF | oc create -f -
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-trident-operator
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-operator
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-resource-quotas-cr
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: resource-quotas-role
EOF
```



For storage admins, two roles must be bound: trident-operator and resource-quotas.

## 9. Create RoleBindings for developers binding the developer-project-1 role to the corresponding group (ocp-project-1) in project-1.

```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-1
EOF
```

10. Similarly, create RoleBindings for developers binding the developer roles to the corresponding user group in project-2.

### **Configuration: Storage-admin tasks**

The following resources must be configured by a storage administrator:

1. Log into the NetApp ONTAP cluster as admin.
2. Navigate to Storage > Storage VMs and click Add. Create two SVMs, one for project-1 and the other for project-2, by providing the required details. Also create a vsadmin account to manage the SVM and its resources.

# Add Storage VM



STORAGE VM NAME

## Access Protocol

SMB/CIFS, NFS  iSCSI

Enable SMB/CIFS

Enable NFS

Allow NFS client access

Add at least one rule to allow NFS clients to access volumes in this storage VM. [?](#)

EXPORT POLICY

Default

RULES

Rule Index	Clients	Access Protocols	Read-Only R...	Read/Wr
	10.61.181.0/24	Any	Any	Any

[+ Add](#)

DEFAULT LANGUAGE [?](#)

NETWORK INTERFACE

Use multiple network interfaces when client traffic is high.

K8s-Ontap-01

IP ADDRESS

SUBNET MASK

GATEWAY

[Add optional gateway](#)

BROADCAST DOMAIN

- Log into the Red Hat OpenShift cluster as the storage administrator.
- Create the backend for project-1 and map it to the SVM dedicated to the project. NetApp recommends using the SVM's vsadmin account to connect the backend to SVM instead of using the ONTAP cluster administrator.

```

cat << EOF | tridentctl -n trident create backend -f
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "nfs_project_1",
  "managementLIF": "172.21.224.210",
  "dataLIF": "10.61.181.224",
  "svm": "project-1-svm",
  "username": "vsadmin",
  "password": "NetApp123"
}
EOF

```



We are using the ontap-nas driver for this example. Use the appropriate driver when creating the backend based on the use case.



We assume that Trident is installed in the trident project.

5. Similarly create the Trident backend for project-2 and map it to the SVM dedicated to project-2.
6. Next, create the storage classes. Create the storage class for project-1 and configure it to use the storage pools from backend dedicated to project-1 by setting the storagePools parameter.

```

cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-1-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_1:.*"
EOF

```

7. Likewise, create a storage class for project-2 and configure it to use the storage pools from backend dedicated to project-2.
8. Create a ResourceQuota to restrict resources in project-1 requesting storage from storageclasses dedicated to other projects.

```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-1-sc-rq
  namespace: project-1
spec:
  hard:
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

9. Similarly, create a ResourceQuota to restrict resources in project-2 requesting storage from storageclasses dedicated to other projects.

## Validation

To validate the multitenant architecture that was configured in the previous steps, complete the following steps:

### Validate access to create PVCs or pods in assigned project

1. Log in as ocp-project-1-user, developer in project-1.
2. Check access to create a new project.

```
oc create ns sub-project-1
```

3. Create a PVC in project-1 using the storageclass that is assigned to project-1.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```



4. Check the PV associated with the PVC.

```
oc get pv
```

5. Validate that the PV and its volume is created in an SVM dedicated to project-1 on NetApp ONTAP.

```
volume show -vserver project-1-svm
```

6. Create a pod in project-1 and mount the PVC created in previous step.

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  volumes:
    - name: test-pvc-project-1
      persistentVolumeClaim:
        claimName: test-pvc-project-1
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: test-pvc-project-1
EOF
```

7. Check if the pod is running and whether it mounted the volume.

```
oc describe pods test-pvc-pod -n project-1
```

**Validate access to create PVCs or pods in another project or use resources dedicated to another project**

1. Log in as ocp-project-1-user, developer in project-1.
2. Create a PVC in project-1 using the storageclass that is assigned to project-2.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1-sc-2
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-2-sc
EOF
```

### 3. Create a PVC in project-2.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-2-sc-1
  namespace: project-2
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```

### 4. Make sure that PVCs test-pvc-project-1-sc-2 and test-pvc-project-2-sc-1 were not created.

```
oc get pvc -n project-1
oc get pvc -n project-2
```

### 5. Create a pod in project-2.

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  containers:
  - name: test-container
    image: nginx
    ports:
    - containerPort: 80
      name: "http-server"
EOF
```

### Validate access to view and edit Projects, ResourceQuotas, and StorageClasses

1. Log in as ocp-project-1-user, developer in project-1.
2. Check access to create new projects.

```
oc create ns sub-project-1
```

3. Validate access to view projects.

```
oc get ns
```

4. Check if the user can view or edit ResourceQuotas in project-1.

```
oc get resourcequotas -n project-1
oc edit resourcequotas project-1-sc-rq -n project-1
```

5. Validate that the user has access to view the storageclasses.

```
oc get sc
```

6. Check access to describe the storageclasses.
7. Validate the user's access to edit the storageclasses.

```
oc edit sc project-1-sc
```

## Scaling: Adding more projects

In a multitenant configuration, adding new projects with storage resources requires additional configuration to make sure that multitenancy is not violated. For adding more projects in a multitenant cluster, complete the following steps:

1. Log into the NetApp ONTAP cluster as a storage admin.
2. Navigate to `Storage` → `Storage VMs` and click `Add`. Create a new SVM dedicated to project-3. Also create a `vsadmin` account to manage the SVM and its resources.

# Add Storage VM



STORAGE VM NAME

project-3-svm

## Access Protocol

SMB/CIFS, NFS

iSCSI

Enable SMB/CIFS

Enable NFS

Allow NFS client access

Add at least one rule to allow NFS clients to access volumes in this storage VM. [?](#)

EXPORT POLICY

Default

RULES

Rule Index	Clients	Access Protocols	Read-Only R...	Read/Wr
	10.61.181.0/24	Any	Any	Any

[+](#) Add

DEFAULT LANGUAGE [?](#)

c.utf\_8

NETWORK INTERFACE

Use multiple network interfaces when client traffic is high.

K8s-Ontap-01

IP ADDRESS

10.61.181.228

SUBNET MASK

24

GATEWAY

[Add optional gateway](#)

BROADCAST DOMAIN

Default-4

3. Log into the Red Hat OpenShift cluster as cluster admin.
4. Create a new project.

```
oc create ns project-3
```

5. Make sure that the user group for project-3 is created on IdP and synchronized with the OpenShift cluster.

```
oc get groups
```

## 6. Create the developer role for project-3.

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-3
  name: developer-project-3
rules:
- verbs:
  - '*'
  apiGroups:
  - apps
  - batch
  - autoscaling
  - extensions
  - networking.k8s.io
  - policy
  - apps.openshift.io
  - build.openshift.io
  - image.openshift.io
  - ingress.operator.openshift.io
  - route.openshift.io
  - snapshot.storage.k8s.io
  - template.openshift.io
  resources:
  - '*'
- verbs:
  - '*'
  apiGroups:
  - ''
  resources:
  - bindings
  - configmaps
  - endpoints
  - events
  - persistentvolumeclaims
  - pods
  - pods/log
  - pods/attach
  - podtemplates
  - replicationcontrollers
  - services
```

```

- limitranges
- namespaces
- componentstatuses
- nodes
- verbs:
  - '*'
apiGroups:
- trident.netapp.io
resources:
- trident.snapshots
EOF

```



The role definition provided in this section is just an example. The developer role must be defined based on the end-user requirements.

7. Create RoleBinding for developers in project-3 binding the developer-project-3 role to the corresponding group (ocp-project-3) in project-3.

```

cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-3-developer
  namespace: project-3
subjects:
- kind: Group
  apiGroup: rbac.authorization.k8s.io
  name: ocp-project-3
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-3
EOF

```

8. Login to the Red Hat OpenShift cluster as storage admin
9. Create a Trident backend and map it to the SVM dedicated to project-3. NetApp recommends using the SVM's vsadmin account to connect the backend to the SVM instead of using the ONTAP cluster administrator.

```

cat << EOF | tridentctl -n trident create backend -f
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "nfs_project_3",
  "managementLIF": "172.21.224.210",
  "dataLIF": "10.61.181.228",
  "svm": "project-3-svm",
  "username": "vsadmin",
  "password": "NetApp!23"
}
EOF

```



We are using the ontap-nas driver for this example. Use the appropriate driver for creating the backend based on the use-case.



We assume that Trident is installed in the trident project.

10. Create the storage class for project-3 and configure it to use the storage pools from backend dedicated to project-3.

```

cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-3-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_3:.*"
EOF

```

11. Create a ResourceQuota to restrict resources in project-3 requesting storage from storageclasses dedicated to other projects.



```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-3-sc-rq
  namespace: project-3
spec:
  hard:
    project-1-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

12. Patch the ResourceQuotas in other projects to restrict resources in those projects from accessing storage from the storageclass dedicated to project-3.

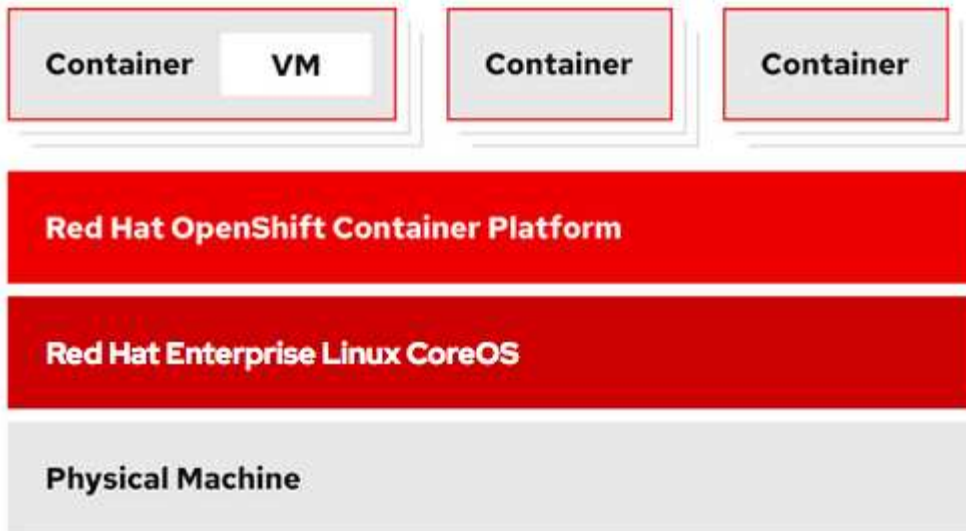
```
oc patch resourcequotas project-1-sc-rq -n project-1 --patch
'{"spec":{"hard":{"project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
oc patch resourcequotas project-2-sc-rq -n project-2 --patch
'{"spec":{"hard":{"project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
```

## Red Hat OpenShift Virtualization with NetApp ONTAP

### Red Hat OpenShift Virtualization with NetApp ONTAP

Depending on the specific use case, both containers and virtual machines (VMs) can serve as optimal platforms for different types of applications. Therefore, many organizations run some of their workloads on containers and some on VMs. Often, this leads organizations to face additional challenges by having to manage separate platforms: a hypervisor for VMs and a container orchestrator for applications.

To address this challenge, Red Hat introduced OpenShift Virtualization (formerly known as Container Native Virtualization) starting from OpenShift version 4.6. The OpenShift Virtualization feature enables you to run and manage virtual machines alongside containers on the same OpenShift Container Platform installation, providing hybrid management capability to automate deployment and management of VMs through operators. In addition to creating VMs in OpenShift, with OpenShift Virtualization, Red Hat also supports importing VMs from VMware vSphere, Red Hat Virtualization, and Red Hat OpenStack Platform deployments.



Certain features like live VM migration, VM disk cloning, VM snapshots and so on are also supported by OpenShift Virtualization with assistance from Astra Trident when backed by NetApp ONTAP. Examples of each of these workflows are discussed later in this document in their respective sections.

To learn more about Red Hat OpenShift Virtualization, see the documentation [here](#).

## Deployment for OpenShift Virtualization

### Deploy Red Hat OpenShift Virtualization with NetApp ONTAP

This section details how to deploy Red Hat OpenShift Virtualization with NetApp ONTAP.

#### Prerequisites

- A Red Hat OpenShift cluster (later than version 4.6) installed on bare-metal infrastructure with RHCOS worker nodes
- The OpenShift cluster must be installed via installer provisioned infrastructure (IPI)
- Deploy Machine Health Checks to maintain HA for VMs
- A NetApp ONTAP cluster
- Astra Trident installed on the OpenShift cluster
- A Trident backend configured with an SVM on ONTAP cluster
- A StorageClass configured on the OpenShift cluster with Astra Trident as the provisioner
- Cluster-admin access to Red Hat OpenShift cluster
- Admin access to NetApp ONTAP cluster
- An admin workstation with `tridentctl` and `oc` tools installed and added to `$PATH`

Because OpenShift Virtualization is managed by an operator installed on the OpenShift cluster, it imposes additional overhead on memory, CPU, and storage, which must be accounted for while planning the hardware requirements for the cluster. See the documentation [here](#) for more details.

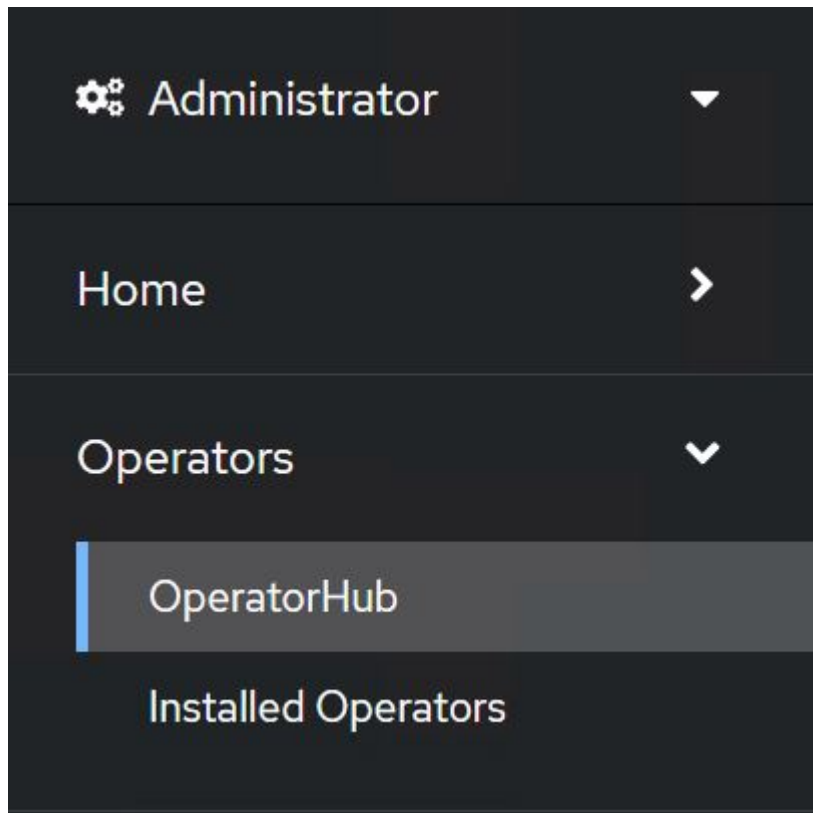
Optionally, you can also specify a subset of the OpenShift cluster nodes to host the OpenShift Virtualization operators, controllers, and VMs by configuring node placement rules. To configure node placement rules for OpenShift Virtualization, follow the documentation [here](#).

For the storage backing OpenShift Virtualization, NetApp recommends having a dedicated StorageClass that requests storage from a particular Trident backend, which in turn is backed by a dedicated SVM. This maintains a level of multitenancy with regard to the data being served for VM-based workloads on the OpenShift cluster.

### Deploy Red Hat OpenShift Virtualization with NetApp ONTAP

To install OpenShift Virtualization, complete the following steps:

1. Log into the Red Hat OpenShift bare-metal cluster with cluster-admin access.
2. Select Administrator from the Perspective drop down.
3. Navigate to Operators > OperatorHub and search for OpenShift Virtualization.



4. Select the OpenShift Virtualization tile and click Install.



Install

### Latest version

2.6.2

### Capability level

- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

### Provider type

Red Hat

### Provider

Red Hat

## Requirements

Your cluster must be installed on bare metal infrastructure with Red Hat Enterprise Linux CoreOS workers.

## Details

**OpenShift Virtualization** extends Red Hat OpenShift Container Platform, allowing you to host and manage virtualized workloads on the same platform as container-based workloads. From the OpenShift Container Platform web console, you can import a VMware virtual machine from vSphere, create new or clone existing VMs, perform live migrations between nodes, and more. You can use OpenShift Virtualization to manage both Linux and Windows VMs.

The technology behind OpenShift Virtualization is developed in the [KubeVirt](#) open source community. The KubeVirt project extends [Kubernetes](#) by adding additional virtualization resource types through [Custom Resource Definitions](#) (CRDs). Administrators can use Custom Resource Definitions to manage [VirtualMachine](#) resources alongside all other resources that Kubernetes provides.

5. On the Install Operator screen, leave all default parameters and click Install.

### Update channel \*

- 2.1
- 2.2
- 2.3
- 2.4
- stable

### Installation mode \*

- All namespaces on the cluster (default)  
This mode is not supported by this Operator
- A specific namespace on the cluster  
Operator will be available in a single Namespace only.

### Installed Namespace \*

- Operator recommended Namespace: **PR** openshift-cnv

**i** Namespace creation  
Namespace **openshift-cnv** does not exist and will be created.

- Select a Namespace

### Approval strategy \*

- Automatic
- Manual

Install Cancel

OpenShift Virtualization  
provided by Red Hat

### Provided APIs

**HC** OpenShift Virtualization Deployment **Required**

Represents the deployment of OpenShift Virtualization

6. Wait for the operator installation to complete.



**OpenShift Virtualization**  
2.6.2 provided by Red Hat



## Installing Operator

The Operator is being installed. This may take a few minutes.

[View installed Operators in Namespace openshift-cnv](#)

7. After the operator has installed, click Create HyperConverged.



**OpenShift Virtualization**  
2.6.2 provided by Red Hat



## Installed operator - operand required

The Operator has installed successfully. Create the required custom resource to be able to use this Operator.

**HC** HyperConverged **Required**

Creates and maintains an OpenShift Virtualization Deployment

[Create HyperConverged](#)

[View installed Operators in Namespace openshift-cnv](#)

8. On the Create HyperConverged screen, click Create, accepting all default parameters. This step starts the installation of OpenShift Virtualization.

**Name \***

**Labels**

**Infra** >

infra HyperConvergedConfig influences the pod configuration (currently only placement) for all the infra components needed on the virtualization enabled cluster but not necessarily directly on each node running VMs/VMIs.

**Workloads** >

workloads HyperConvergedConfig influences the pod configuration (currently only placement) of components which need to be running on a node where virtualization workloads should be able to run. Changes to Workloads HyperConvergedConfig can be applied only without existing workload.

**Bare Metal Platform**

true

BareMetalPlatform indicates whether the infrastructure is baremetal.

**Feature Gates** >

featureGates is a map of feature gate flags. Setting a flag to `true` will enable the feature. Setting `false` or removing the feature gate, disables the feature.

**Local Storage Class Name**





LocalStorageClassName the name of the local storage class.

- After all the pods move to the Running state in the openshift-cnv namespace and the OpenShift Virtualization operator is in the Succeeded state, the operator is ready to use. VMs can now be created on the OpenShift cluster.

Project: openshift-cnv ▾

## Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#) or create an Operator and ClusterServiceVersion using the [Operator SDK](#).

Name	Managed Namespaces	Status	Last updated	Provided APIs
 <b>OpenShift Virtualization</b> 2.6.2 provided by Red Hat	 openshift-cnv	 Succeeded Up to date	 May 18, 8:02 pm	<a href="#">OpenShift Virtualization Deployment</a> <a href="#">HostPathProvisioner deployment</a>

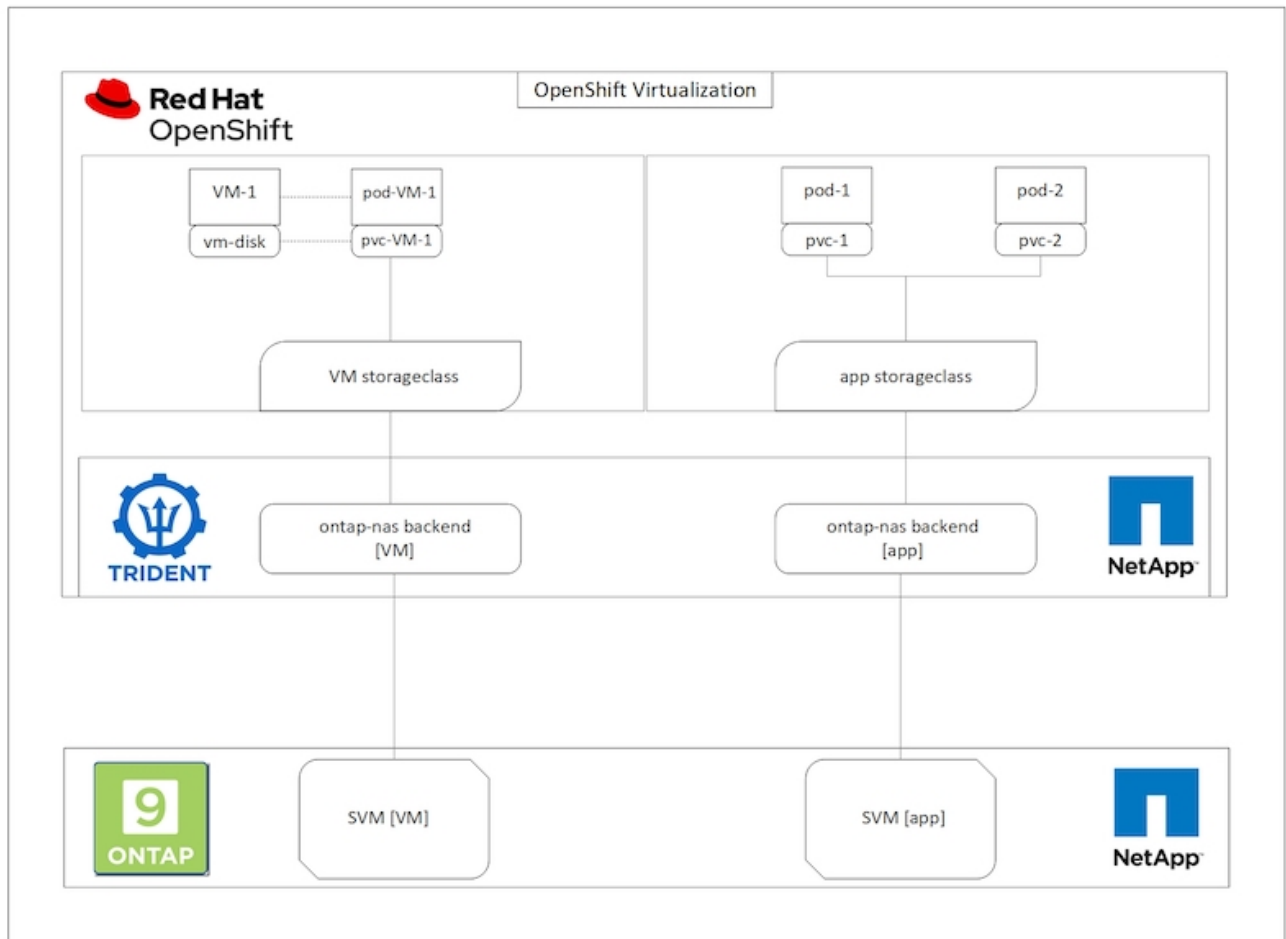
## Workflows

### Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP

This section covers the how to create a virtual machine with Red Hat OpenShift Virtualization.

## Create VM

VMs are stateful deployments that require volumes to host the operating system and data. With CNV, because the VMs are run as pods, the VMs are backed by PVs hosted on NetApp ONTAP through Trident. These volumes are attached as disks and store the entire filesystem including the boot source of the VM.



To create a virtual machine on the OpenShift cluster, complete the following steps:

1. Navigate to Workloads > Virtualization > Virtual Machines and click Create > With Wizard.
2. Select the desired the operating system and click Next.
3. If the selected operating system has no boot source configured, you must configure it. For Boot Source, select whether you want to import the OS image from an URL or from a registry and provide the corresponding details. Expand Advanced and select the Trident-backed StorageClass. Then click Next.

## Boot source

This template does not have a boot source. Provide a custom boot source for this **CentOS 8.0+ VM** virtual machine.

### Boot source type \*

Import via URL (creates PVC) ▼

### Import URL \*

<https://access.cdn.redhat.com/content/origin/files/sha256/58/588167f828001e57688ec4b9b31c11a59d532489f527488ebc89ac5e952...>

Example: For RHEL, visit the [RHEL download page](#) (requires login) and copy the download link URL of the KVM guest image

Mount this as a CD-ROM boot source [?](#)

### Persistent Volume Claim size \*

5 GiB ▼

Ensure your PVC size covers the requirements of the uncompressed image and any other space requirements. More storage can be added later.

### ▼ Advanced

#### Storage class \*

basic (default) ▼

#### Access mode \*

Single User (RWO) ▼

#### Volume mode \*

Filesystem ▼

4. If the selected operating system already has a boot source configured, the previous step can be skipped.
5. In the Review and Create pane, select the project you want to create the VM in and furnish the VM details. Make sure that the boot source is selected to be Clone and boot from CD-ROM with the appropriate PVC assigned for the selected OS.



- 1 Select template
- 2 Review and create

### Review and create

You are creating a virtual machine from the **Red Hat Enterprise Linux 8.0+** VM template.

**Project \***

**Virtual Machine Name \* ⓘ**

**Flavor \***

**Storage**      **Workload profile ⓘ**  
 40 GiB      server

**Boot source**  
 Clone and boot from CD-ROM  
 PVC rhel8

**ⓘ** A new disk has been added to support the CD-ROM boot source. Edit this disk by customizing the virtual machine.

▼ **Disk details**

rootdisk-install - Blank - 20GiB - virtio - default Storage class

Start this virtual machine after creation

6. If you wish to customize the virtual machine, click **Customize Virtual Machine** and modify the required parameters.
7. Click **Create Virtual Machine** to create the virtual machine; this spins up a corresponding pod in the background.

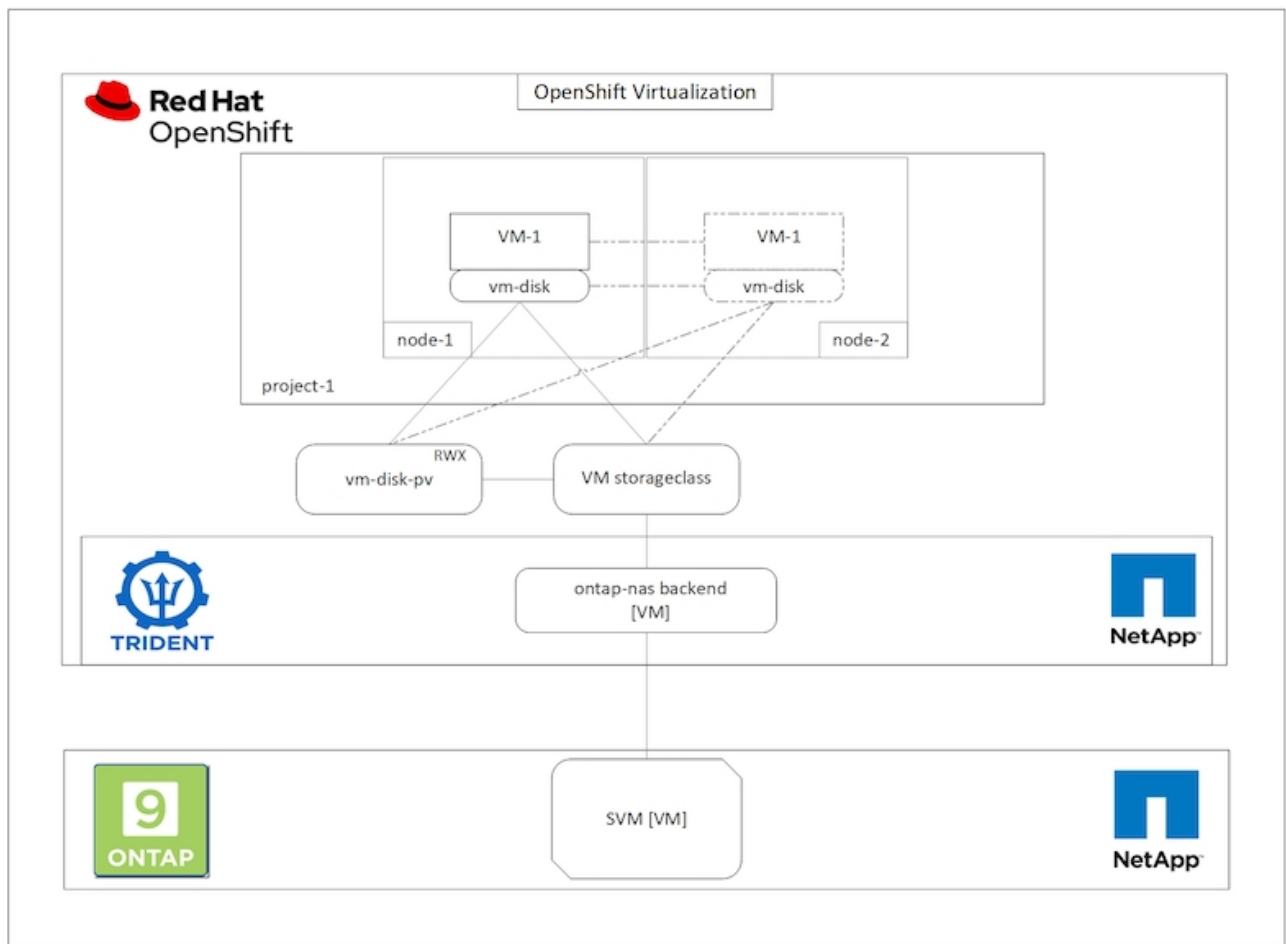
When a boot source is configured for a template or an operating system from an URL or from a registry, it creates a PVC in the `openshift-visualization-os-images` project and downloads the KVM guest image to the PVC. You must make sure that template PVCs have enough provisioned space to accommodate the KVM guest image for the corresponding OS. These PVCs are then cloned and attached as rootdisks to virtual machines when they are created using the respective templates in any project.

#### Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP

This section covers the how to migrate a virtual machine between clusters with Red Hat OpenShift Virtualization.

### VM Live Migration

Live Migration is a process of migrating a VM instance from one node to another in an OpenShift cluster with no downtime. For live migration to work in an OpenShift cluster, VMs must be bound to PVCs with shared ReadWriteMany access mode. Astra Trident backend configured with an SVM on a NetApp ONTAP cluster that is enabled for NFS protocol supports shared ReadWriteMany access for PVCs. Therefore, the VMs with PVCs that are requested from StorageClasses provisioned by Trident from NFS-enabled SVM can be migrated with no downtime.



To create a VM bound to PVCs with shared ReadWriteMany access:

1. Navigate to Workloads > Virtualization > Virtual Machines and click Create > With Wizard.
2. Select the desired the operating system and click Next. Let us assume the selected OS already had a boot source configured with it.
3. In the Review and Create pane, select the project you want to create the VM in and furnish the VM details. Make sure that the boot source is selected to be Clone and boot from CD-ROM with the appropriate PVC assigned for the selected OS.
4. Click Customize Virtual Machine and then click Storage.
5. Click the ellipsis next to rootdisk, and make sure that the storageclass provisioned using Trident is selected. Expand Advanced and select Shared Access (RWX) for Access Mode. Then click Save.

# Edit Disk

Type

Disk

Interface \*

virtio

Storage Class

basic (default)

Advanced

Volume Mode

Filesystem

Volume Mode is set by Source PVC

Access Mode

Shared Access (RWX) - Not recommended for basic storage class

**i** Access and Volume modes should follow storage feature matrix  
[Learn more](#)

Cancel Save

6. Click Review and confirm and then click Create Virtual Machine.

To manually migrate a VM to another node in the OpenShift cluster, complete the following steps.

1. Navigate to Workloads > Virtualization > Virtual Machines.

2. For the VM you wish to migrate, click the ellipsis, and then click Migrate the Virtual Machine.
3. Click Migrate when the message pops up to confirm.



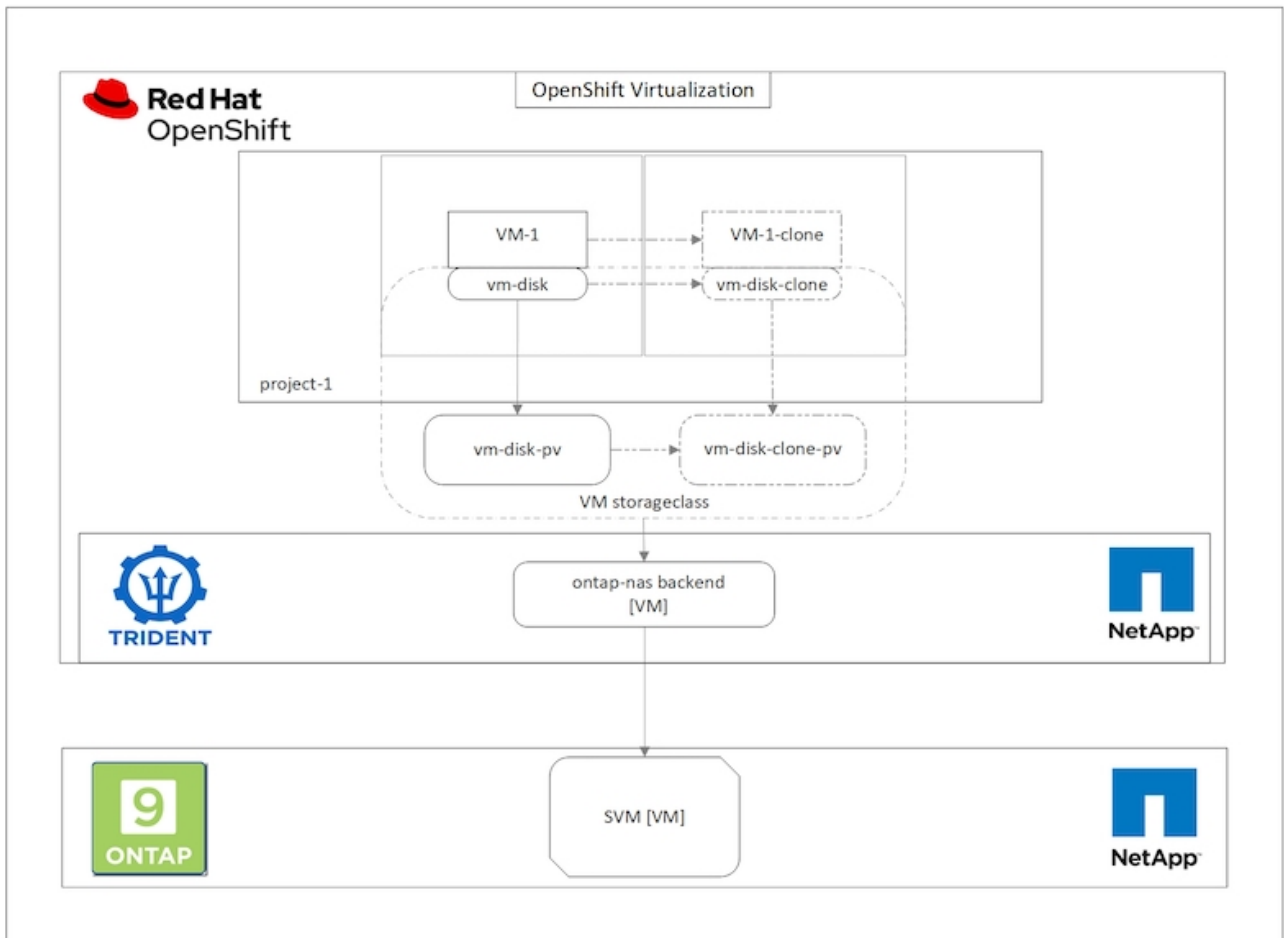
A VM instance in an OpenShift cluster automatically migrates to another node when the original node is placed into maintenance mode if the evictionStrategy is set to LiveMigrate.

### Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP

This section covers the how to clone a virtual machine with Red Hat OpenShift Virtualization.

### VM cloning

Cloning an existing VM in OpenShift is achieved with the support of Astra Trident's Volume CSI cloning feature. CSI volume cloning allows for creation of a new PVC using an existing PVC as the data source by duplicating its PV. After the new PVC is created, it functions as a separate entity and without any link to or dependency on the source PVC.



There are certain restrictions with CSI volume cloning to consider:

1. Source PVC and destination PVC must be in the same project.
2. Cloning is supported within the same storage class.

3. Cloning can be performed only when source and destination volumes use the same VolumeMode setting; for example, a block volume can only be cloned to another block volume.

VMs in an OpenShift cluster can be cloned in two ways:

1. By shutting down the source VM
2. By keeping the source VM live

### **By Shutting down the source VM**

Cloning an existing VM by shutting down the VM is a native OpenShift feature that is implemented with support from Astra Trident. Complete the following steps to clone a VM.

1. Navigate to Workloads > Virtualization > Virtual Machines and click the ellipsis next to the virtual machine you wish to clone.
2. Click Clone Virtual Machine and provide the details for the new VM.

# Clone Virtual Machine

Name \*

rhel8-short-frog-clone

Description

Namespace \*

default

Start virtual machine on clone

Configuration

Operating System

Red Hat Enterprise Linux 8.0 or higher

Flavor

Small: 1 CPU | 2 GiB Memory

Workload Profile

server

NICs

default - virtio

Disks

cloudinitdisk - cloud-init disk

rootdisk - 20Gi - basic



The VM rhel8-short-frog is still running. It will be powered off while cloning.

Cancel

Clone Virtual Machine

3. Click Clone Virtual Machine; this shuts down the source VM and initiates the creation of the clone VM.
4. After this step is completed, you can access and verify the content of the cloned VM.

## By keeping the source VM live

An existing VM can also be cloned by cloning the existing PVC of the source VM and then creating a new VM using the cloned PVC. This method does not require you to shut down the source VM. Complete the following steps to clone a VM without shutting it down.

1. Navigate to Storage > PersistentVolumeClaims and click the ellipsis next to the PVC that is attached to the source VM.
2. Click Clone PVC and furnish the details for the new PVC.

# Clone

Name \*

Access Mode \*


Single User (RWO)  Shared Access (RWX)  Read Only (ROX)

Size \*

GiB ▼

PVC details

**Namespace**

 default

**Requested capacity**

20 GiB

**Access mode**

Shared Access (RWX)

**Storage Class**

 basic

**Used capacity**

2.2 GiB

**Volume mode**

Filesystem

Cancel

Clone

3. Then click Clone. This creates a PVC for the new VM.
4. Navigate to Workloads > Virtualization > Virtual Machines and click Create > With YAML.
5. In the spec > template > spec > volumes section, attach the cloned PVC instead of the container disk. Provide all other details for the new VM according to your requirements.

```
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dvh-clone
```

6. Click Create to create the new VM.
7. After the VM is created successfully, access and verify that the new VM is a clone of the source VM.

#### **Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP**

This section covers the how to create a virtual machine from a Snapshot with Red Hat OpenShift Virtualization.

#### **Create VM from a Snapshot**

With Astra Trident and Red Hat OpenShift, users can take a snapshot of a persistent volume on Storage Classes provisioned by it. With this feature, users can take a point-in-time copy of a volume and use it to create a new volume or restore the same volume back to a previous state. This enables or supports a variety of use-cases, from rollback to clones to data restore.

For Snapshot operations in OpenShift, the resources `VolumeSnapshotClass`, `VolumeSnapshot`, and `VolumeSnapshotContent` must be defined.

- A `VolumeSnapshotContent` is the actual snapshot taken from a volume in the cluster. It is cluster-wide resource analogous to `PersistentVolume` for storage.
- A `VolumeSnapshot` is a request for creating the snapshot of a volume. It is analogous to a `PersistentVolumeClaim`.
- `VolumeSnapshotClass` lets the administrator specify different attributes for a `VolumeSnapshot`. It allows you to have different attributes for different snapshots taken from the same volume.





```
1  apiVersion: snapshot.storage.k8s.io/v1
2  kind: VolumeSnapshotClass
3  metadata:
4    name: trident-snapshot-class
5  driver: csi.trident.netapp.io
6  deletionPolicy: Delete
7
```

[Create](#) [Cancel](#) [Download](#)

3. Identify the PVC that is attached to the source VM and then create a Snapshot of that PVC. Navigate to `Storage > VolumeSnapshots` and click `Create VolumeSnapshots`.
4. Select the PVC that you want to create the Snapshot for, enter the name of the Snapshot or accept the default, and select the appropriate `VolumeSnapshotClass`. Then click `Create`.

## Create VolumeSnapshot

[Edit YAML](#)

PersistentVolumeClaim \*

PVC rhel8-short-frog-rootdisk-28dvv ▼

Name \*

rhel8-short-frog-rootdisk-28dvv-snapshot

Snapshot Class \*

VSC trident-snapshot-class ▼

[Create](#) [Cancel](#)

5. This creates the snapshot of the PVC at that point in time.

## Create a new VM from the snapshot

1. First, restore the Snapshot into a new PVC. Navigate to Storage > VolumeSnapshots, click the ellipsis next to the Snapshot that you wish to restore, and click Restore as new PVC.
2. Enter the details of the new PVC and click Restore. This creates a new PVC.

# Restore as new PVC

When restore action for snapshot **rhel8-short-frog-rootdisk-28dvv-snapshot** is finished a new crash-consistent PVC copy will be created.

Name \*

rhel8-short-frog-rootdisk-28dvv-snapshot-restore

Storage Class \*

 basic

Access Mode \*

Single User (RWO)  Shared Access (RWX)  Read Only (ROX)

Size \*

20

GiB

## VolumeSnapshot details

Created at

 May 21, 12:46 am

Namespace

 default

Status

 Ready

API version

snapshot.storage.k8s.io/v1

Size

20 GiB

3. Next, create a new VM from this PVC. Navigate to Workloads > Virtualization > Virtual Machines and click Create > With YAML.
4. In the spec > template > spec > volumes section, specify the new PVC created from Snapshot instead of

from the container disk. Provide all other details for the new VM according to your requirements.

```
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dvh-snapshot-restore
```

5. Click Create to create the new VM.
6. After the VM is created successfully, access and verify that the new VM has the same state as that of the VM whose PVC was used to create the snapshot at the time when the snapshot was created.

#### Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP

This section covers the how to migrate a virtual machine between clusters using Red Hat OpenShift Virtualization migration toolkit.

#### Migration of VM from VMware to OpenShift Virtualization using Migration Toolkit for Virtualization

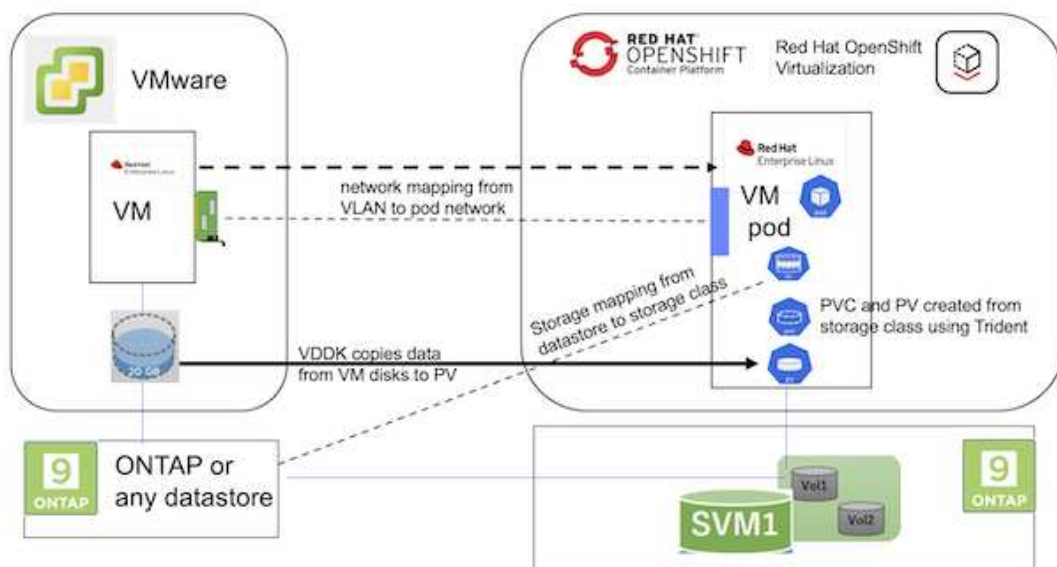
In this section, we will see how to use the Migration Toolkit for Virtualization (MTV) to migrate virtual machines from VMware to OpenShift Virtualization running on OpenShift Container platform and integrated with NetApp ONTAP storage using Astra Trident.

The following video shows a demonstration of the migration of a RHEL VM from VMware to OpenShift Virtualization using ontap-san for persistent storage.

[Using Red Hat MTV to migrate VMs to OpenShift Virtualization with NetApp ONTAP Storage](#)

The following diagram shows a high level view of the migration of a VM from VMware to Red Hat OpenShift Virtualization.

### Migration of VM from VMware to OpenShift Virtualization



## Prerequisites for the sample migration

### On VMware

- A RHEL 9 VM using rhel 9.3 with the following configurations were installed:
  - CPU: 2, Memory: 20 GB, Hard disk: 20 GB
  - user credentials: root user and an admin user credentials
- After the VM was ready, postgresql server was installed.
  - postgresql server was started and enabled to start on boot

```
systemctl start postgresql.service`  
systemctl enable postgresql.service  
The above command ensures that the server can start in the VM in  
OpenShift Virtualization after migration
```

- Added 2 databases, 1 table and 1 row in the table were added. Refer [here](#) for the instructions for installing postgresql server on RHEL and creating database and table entries.



Ensure that you start the postgresql server and enable the service to start at boot.

### On OpenShift Cluster

The following installations were completed before installing MTV:

- OpenShift Cluster 4.13.34
- [Astra Trident 23.10](#)
- Multipath on the cluster nodes enabled for iSCSI (for ontap-san storage class). See the provided yaml to create a daemon set that enables iSCSI on each node in the cluster.
- Trident backend and Storage class for ontap SAN using iSCSI. See the provided yaml files for trident backend and storage class.
- [OpenShift Virtualization](#)

To install iscsi and multipath on the OpenShift Cluster nodes use the yaml file given below

#### Preparing the cluster nodes for iSCSI

```
apiVersion: apps/v1  
kind: DaemonSet  
metadata:  
  namespace: trident  
  name: trident-iscsi-init  
  labels:  
    name: trident-iscsi-init  
spec:  
  selector:  
    matchLabels:
```

```

    name: trident-iscsi-init
template:
  metadata:
    labels:
      name: trident-iscsi-init
  spec:
    hostNetwork: true
    serviceAccount: trident-node-linux
    initContainers:
    - name: init-node
      command:
        - nsenter
        - --mount=/proc/1/ns/mnt
        - --
        - sh
        - -c
      args: ["$(STARTUP_SCRIPT)"]
      image: alpine:3.7
      env:
      - name: STARTUP_SCRIPT
        value: |
          #!/bin/bash
          sudo yum install -y lsscsi iscsi-initiator-utils sg3_utils
device-mapper-multipath
          rpm -q iscsi-initiator-utils
          sudo sed -i 's/^\(node.session.scan\).*\/\1 = manual/'
/etc/iscsi/iscsid.conf
          cat /etc/iscsi/initiatorname.iscsi
          sudo mpathconf --enable --with_multipathd y --find_multipaths
n
          sudo systemctl enable --now iscsid multipathd
          sudo systemctl enable --now iscsi
      securityContext:
        privileged: true
    hostPID: true
    containers:
    - name: wait
      image: k8s.gcr.io/pause:3.1
    hostPID: true
    hostNetwork: true
    tolerations:
    - effect: NoSchedule
      key: node-role.kubernetes.io/master
  updateStrategy:
    type: RollingUpdate

```

Use the following yaml file to create trident backend configuration for using ontap san storage

### Trident backend for iSCSI

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: <username>
  password: <password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ontap-san
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <management LIF>
  backendName: ontap-san
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-san-secret
```

Use the following yaml file to create trident storage class configuration for using ontap san storage

### Trident storage class for iSCSI

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
allowVolumeExpansion: true
```

### Install MTV

Now you can install the Migration Toolkit for virtualization (MTV). Refer to the instructions provided [here](#) for help with the installation.

The Migration Toolkit for Virtualization (MTV) user interface is integrated into the OpenShift web console. You can refer [here](#) to start using the user interface for various tasks.

## Create Source Provider

In order to migrate the RHEL VM from VMware to OpenShift Virtualization, you need to first create the source provider for VMware. Refer to the instructions [here](#) to create the source provider.

You need the following to create your VMware source provider:

- VCenter url
- VCenter Credentials
- VCenter server thumbprint
- VDDK image in a repository

Sample source provider creation:

The screenshot displays a form titled "Select provider type \*". The "vm vSphere" option is selected. Below this, several fields are filled out:

- Provider resource name \***: vmware-source (with a green checkmark and the text "Unique Kubernetes resource name identifier" below it).
- URL \***: [Redacted] (with a green checkmark and the text "URL of the vCenter SDK endpoint. Ensure the URL includes the '/sdk' path. For example: https://vCenter-host-example.com/sdk" below it).
- VDDK init image:** docker.repo.eng.netapp.com/banum/vddk:801 (with a green checkmark and the text "VDDK container image of the provider, when left empty some functionality will not be available" below it).
- Username \***: administrator@vsphere.local (with a "Text" label and the text "vSphere REST API user name." below it).
- Password \***: [Redacted] (with a green checkmark, an eye icon, and the text "vSphere REST API password credentials." below it).
- SSHA-1 fingerprint \***: [Redacted] (with a green checkmark and the text "The provider currently requires the SHA-1 fingerprint of the vCenter Server's TLS certificate in all circumstances. vSphere calls this the server's thumbprint." below it).
- Skip certificate validation**:





The Migration Toolkit for Virtualization (MTV) uses the VMware Virtual Disk Development Kit (VDDK) SDK to accelerate transferring virtual disks from VMware vSphere. Therefore, creating a VDDK image, although optional, is highly recommended. To make use of this feature, you download the VMware Virtual Disk Development Kit (VDDK), build a VDDK image, and push the VDDK image to your image registry.

Follow the instructions provided [here](#) to create and push the VDDK image to a registry accessible from the OpenShift Cluster.

## Create Destination provider

The host cluster is automatically added as the OpenShift virtualization provider is the source provider.

## Create Migration Plan

Follow the instructions provided [here](#) to create a migration plan.

While creating a plan, you need to create the following if not already created:

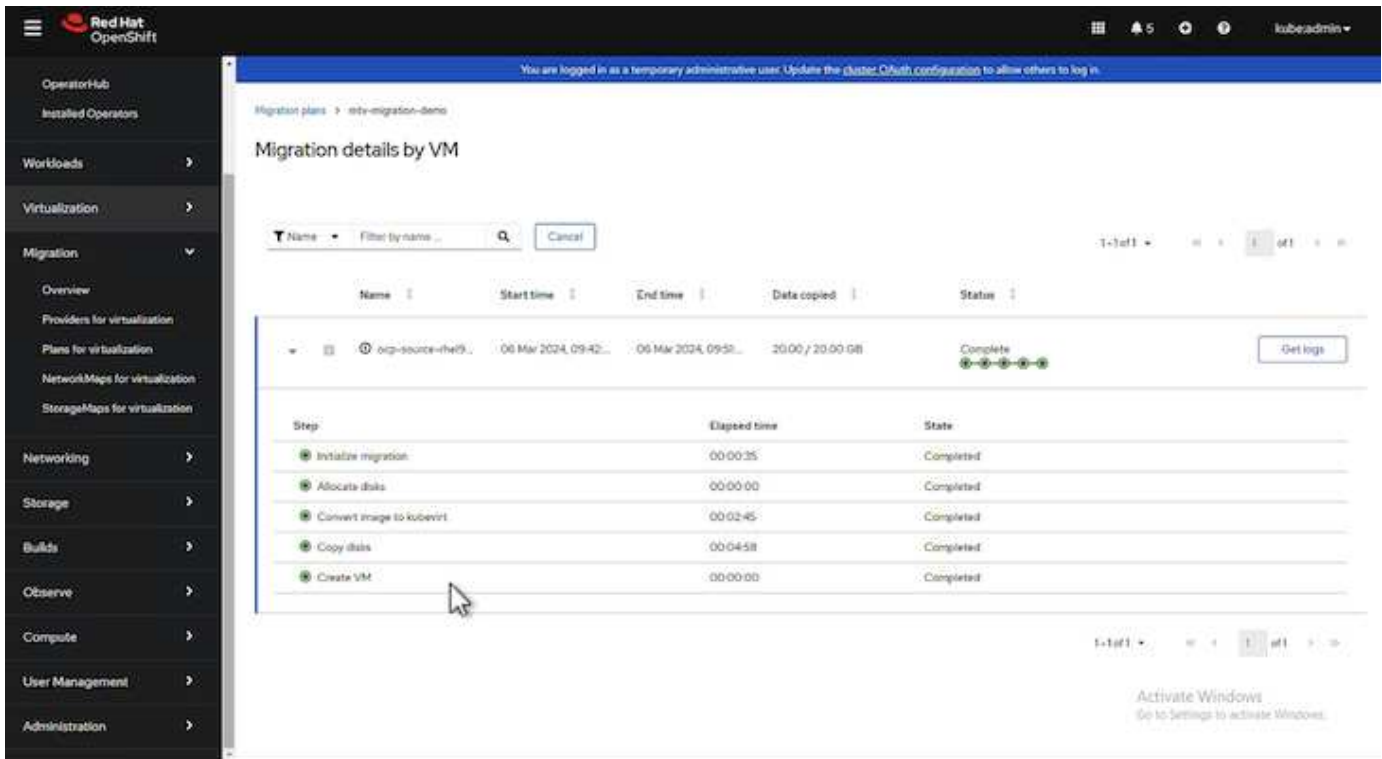
- A network mapping to map the source network to the target network.
- A storage mapping to map the source datastore to the target storage class. For this you can choose ontap-san storage class.

Once the migration plan is created, the status of the plan should show **Ready** and you should now be able to **Start** the plan.

The screenshot shows the OpenShift MTV console interface. The left sidebar contains navigation options: OperatorHub, Installed Operators, Workloads, Virtualization, Migration (selected), Overview, Providers for virtualization, Plans for virtualization (highlighted), NetworkMaps for virtualization, StorageMaps for virtualization, and Networking. The main content area displays a table of migration plans under the heading 'Plans'. The table has columns for Name, Source, Target, VMs, Status, and Description. The first row shows a plan named 'mtv-migration-demo' with a status of 'Ready' and a 'Start' button. The second row shows a plan named 'vmware-osv-migration' with a status of 'Succeeded' and a progress indicator '1 of 1 VMs migrated'. The third row shows a plan named 'vmware-osv-migration-plan1' with a status of 'Succeeded' and a progress indicator '1 of 1 VMs migrated'. The fourth row shows a plan named 'vmware-osv-migration-plan2' with a status of 'Succeeded' and a progress indicator '1 of 1 VMs migrated'. A 'Create plan' button is located in the top right corner of the table area.

Name	Source	Target	VMs	Status	Description
mtv-migration-demo	vmware	host	1	Ready	Plan for migrating VM to OpenShift Virt...
vmware-osv-migration	vmware2	host	1	Succeeded	Migrating RHEL 9 vm to OpenShift Virtu...
vmware-osv-migration-plan1	vmware2	host	1	Succeeded	
vmware-osv-migration-plan2	vmware2	host	1	Succeeded	migrating RHEL 9 vm using ONTAP NFS...

Clicking on **Start** will run through a sequence of steps to complete the migration of the VM.



When all steps are completed, you can see the migrated VMs by clicking on the **virtual machines** under **Virtualization** in the left-side navigation menu.

Instructions to access the virtual machines are provided [here](#).

You can log into the virtual machine and verify the contents of the postgresql databases. The databases, tables and the entries in the table should be the same as what was created on the source VM.

## Data Protection for OpenShift Virtualization

### Data protection for VMs in OpenShift Virtualization using OpenShift API for Data Protection (OADP)

Author: Banu Sundhar, NetApp

This section of the reference document provides details for creating backups of VMs using the OpenShift API for Data Protection (OADP) with Velero on NetApp ONTAP S3 or NetApp StorageGRID S3. The backups of Persistent Volumes(PVs) of the VM disks are created using CSI Astra Trident Snapshots.

Virtual machines in the OpenShift Virtualization environment are containerized applications that run in the worker nodes of your OpenShift Container platform. It is important to protect the VM metadata as well as the persistent disks of the VMs, so that when they are lost or corrupted, you can recover them.

The persistent disks of the OpenShift Virtualization VMs can be backed by ONTAP storage integrated to the OpenShift Cluster using [Astra Trident CSI](#). In this section we use [OpenShift API for Data Protection \(OADP\)](#) to perform backup of VMs including its data volumes to

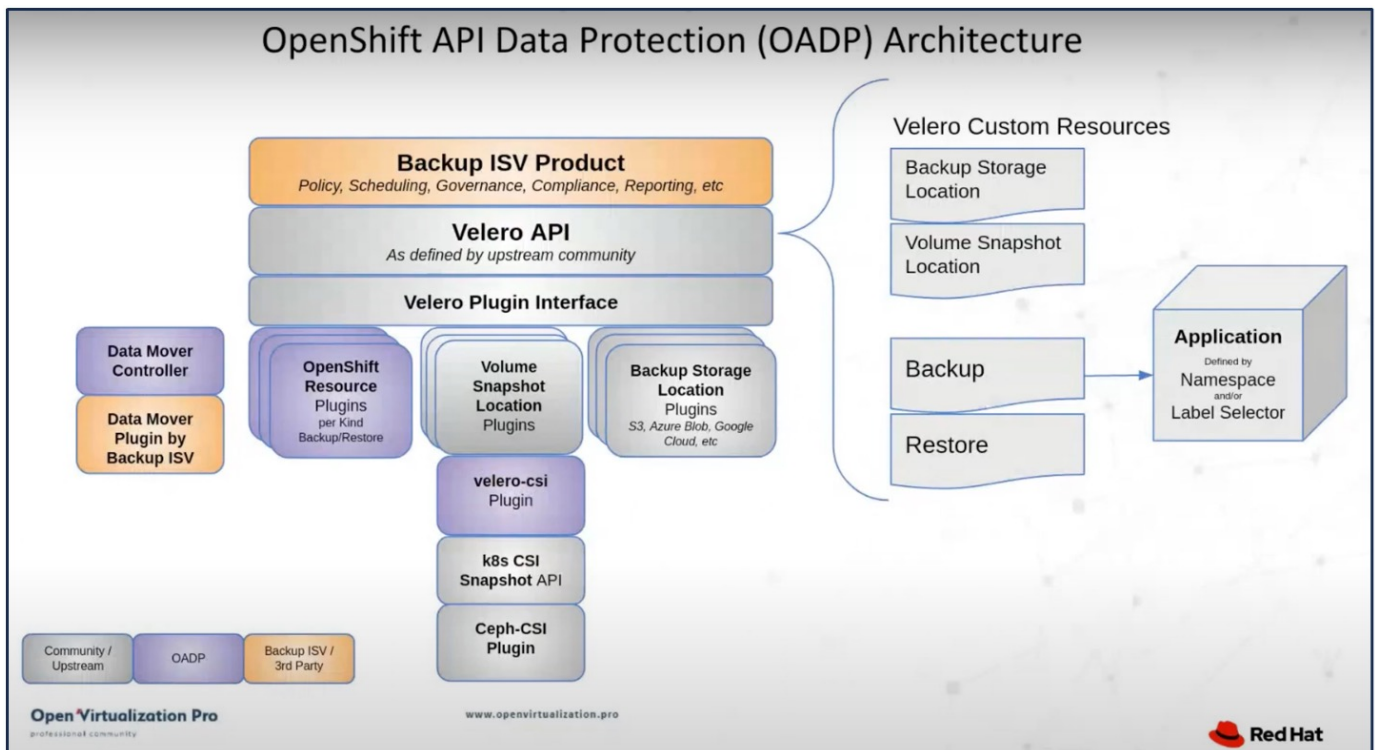
- ONTAP Object Storage
- StorageGrid

We then restore from the backup when needed.

OADP enables backup, restore, and disaster recovery of applications on an OpenShift cluster. Data that can be protected with OADP include Kubernetes resource objects, persistent volumes, and internal images.



Red Hat OpenShift has leveraged the solutions developed by the OpenSource communities for data protection. [Velero](#) is an open-source tool to safely backup and restore, perform disaster recovery, and migrate Kubernetes cluster resources and persistent volumes. To use Velero easily, OpenShift has developed the OADP operator and the Velero plugin to integrate with the CSI storage drivers. The core of the OADP APIs that are exposed are based on the Velero APIs. After installing the OADP operator and configuring it, the backup/restore operations that can be performed are based on the operations exposed by the Velero API.



OADP 1.3 is available from the operator hub of OpenShift cluster 4.12 and later. It has a built-in Data Mover that can move CSI volume snapshots to a remote object store. This provides portability and durability by moving snapshots to an object storage location during backup. The snapshots are then available for restoration after disasters.

The following are the versions of the various components used for the examples in this section

- OpenShift Cluster 4.14
- OpenShift Virtualization installed via OperatorOpenShift Virtualization Operator provided by Red Hat
- OADP Operator 1.13 provided by Red Hat
- Velero CLI 1.13 for Linux
- Astra Trident 24.02
- ONTAP 9.12

[Astra Trident CSI](#)  
[OpenShift API for Data Protection \(OADP\)](#)  
[Velero](#)

### Installation of OpenShift API for Data Protection (OADP) Operator

This section outlines the installation of OpenShift API for Data Protection (OADP) Operator.

#### Prerequisites

- A Red Hat OpenShift cluster (later than version 4.12) installed on bare-metal infrastructure with RHCOS worker nodes
- A NetApp ONTAP cluster integrated with the cluster using Astra Trident

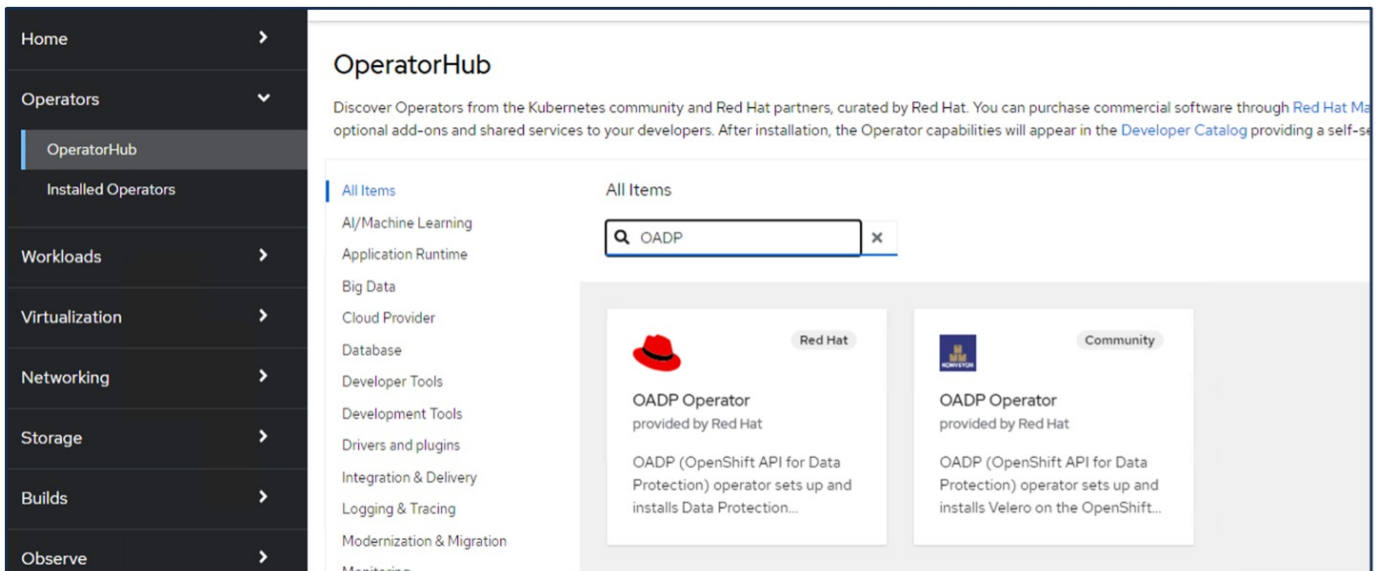
- A Trident backend configured with an SVM on ONTAP cluster
- A StorageClass configured on the OpenShift cluster with Astra Trident as the provisioner
- Trident Snapshot class created on the cluster
- Cluster-admin access to Red Hat OpenShift cluster
- Admin access to NetApp ONTAP cluster
- OpenShift Virtualization operator installed and configured
- VMs deployed in a Namespace on OpenShift Virtualization
- An admin workstation with tridentctl and oc tools installed and added to \$PATH



If you want to take a backup of a VM when it is in the Running state, then you must install the QEMU guest agent on that virtual machine. If you install the VM using an existing template, then QEMU agent is installed automatically. QEMU allows the guest agent to quiesce in-flight data in the guest OS during the snapshot process, and avoid possible data corruption. If you do not have QEMU installed, you can stop the virtual machine before taking a backup.

### Steps to install OADP Operator

1. Go to the Operator Hub of the cluster and select Red Hat OADP operator. In the Install page, use all the default selections and click install. On the next page, again use all the defaults and click Install. The OADP operator will be installed in the namespace openshift-adp.





# OADP Operator

1.3.0 provided by Red Hat

Install

## Channel

stable-1.3

OpenShift API for Data Protection (OADP) operator sets up and installs Velero on the OpenShift platform, allowing users to backup and restore applications.

## Version

1.3.0

Backup and restore Kubernetes resources and internal images, at the granularity of a namespace, using a version of Velero appropriate for the installed version of OADP.

## Capability level

- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

OADP backs up Kubernetes objects and internal images by saving them as an archive file on object storage. OADP backs up persistent volumes (PVs) by creating snapshots with the native cloud snapshot API or with the Container Storage Interface (CSI). For cloud providers that do not support snapshots, OADP backs up resources and PV data with Restic or Kopia.

- [Installing OADP for application backup and restore](#)
- [Installing OADP on a ROSA cluster and using STS, please follow the Getting Started Steps 1-3 in order to obtain the role ARN needed for using the standardized STS configuration flow via OLM](#)
- [Frequently Asked Questions](#)

## Source

Red Hat

## Provider

Red Hat

## Infrastructure features

Disconnected

Activate Windows

Project: All Projects

## Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#) Operator and ClusterServiceVersion using the [Operator SDK](#).

Name Search by name... /

Name	Namespace	Managed Namespaces	Status
<b>OpenShift Virtualization</b> 4.14.4 provided by Red Hat	<b>NS</b> openshift-cnv	<b>NS</b> openshift-cnv	✓ Succeeded Up to date
<b>OADP Operator</b> 1.3.0 provided by Red Hat	<b>NS</b> openshift-adp	<b>NS</b> openshift-adp	✓ Succeeded Up to date
<b>Package Server</b> 0.0.1-snapshot provided by	<b>NS</b> openshift-operator-lifecycle-manager	<b>NS</b> openshift-operator-lifecycle-manager	✓ Succeeded

## Prerequisites for Velero configuration with Ontap S3 details

After the installation of the operator succeeds, configure the instance of Velero.

Velero can be configured to use S3 compatible Object Storage. Configure ONTAP S3 using the procedures shown in the [Object Storage Management section of ONTAP documentation](#). You will need the following information from your ONTAP S3 configuration to integrate with Velero.

- A Logical Interface (LIF) that can be used to access S3
- User credentials to access S3 that includes the access key and the secret access key
- A bucket name in S3 for backups with access permissions for the user
- For secure access to the Object storage, TLS certificate should be installed on the Object Storage server.

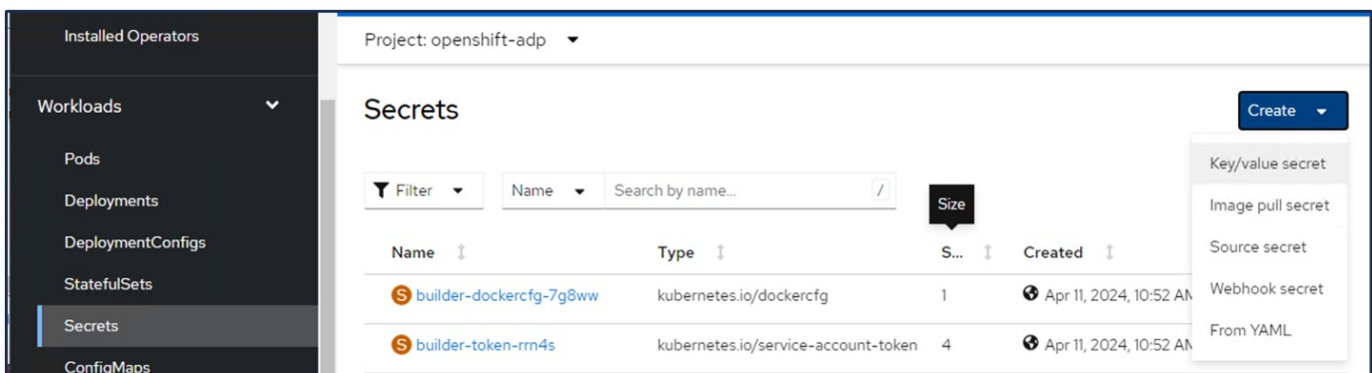
## Prerequisites for Velero configuration with StorageGrid S3 details

Velero can be configured to use S3 compatible Object Storage. You can configure StorageGrid S3 using the procedures shown in the [StorageGrid documentation](#). You will need the following information from your StorageGrid S3 configuration to integrate with Velero.

- The endpoint that can be used to access S3
- User credentials to access S3 that includes the access key and the secret access key
- A bucket name in S3 for backups with access permissions for the user
- For secure access to the Object storage, TLS certificate should be installed on the Object Storage server.

## Steps to configure Velero

- First, create a secret for an ONTAP S3 user credential or StorageGrid Tenant user credentials. This will be used to configure Velero later. You can create a secret from the CLI or from the web console. To create a secret from the web console, select Secrets, then click on Key/Value Secret. Provide the values for the credential name, key and the value as shown. Be sure to use the Access Key Id and Secret Access Key of your S3 user. Name the secret appropriately. In the sample below, a secret with ONTAP S3 user credentials named `ontap-s3-credentials` is created.



Project: openshift-adp ▾

## Create key/value secret

Key/value secrets let you inject sensitive data into your application as files or environment variables.

**Secret name \***

Unique name of the new secret.

**Key \***

**Value**

Drag and drop file with your value here or browse to upload it.

```
[default]
aws_access_key_id=<Access Key Id of S3 user>
aws_secret_access_key=<Secret Access Key of S3 user>
```

[+ Add key/value](#)

To create a secret named sg-s3-credentials from the CLI you can use the following command.

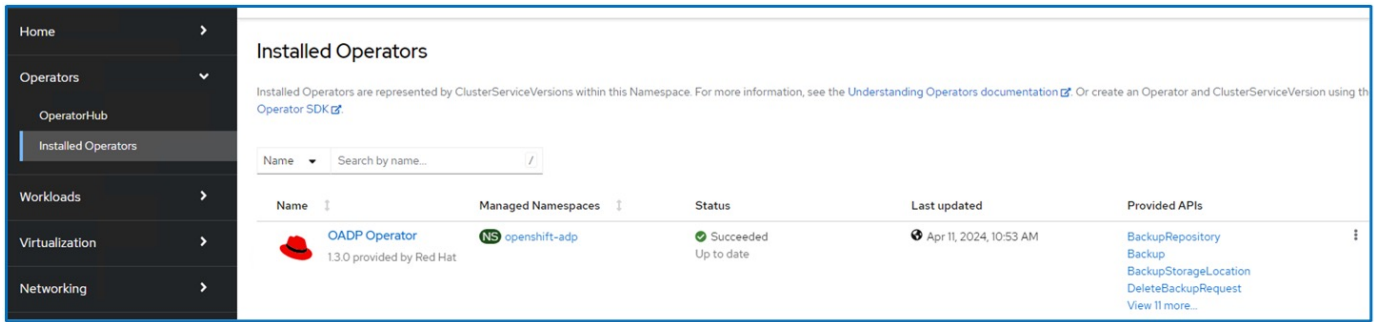
```
# oc create secret generic cloud-credentials --namespace openshift-adp --
from-file cloud=cloud-credentials.txt

credentials.txt file contains the Access Key Id and the Secret Access Key of
the S3 user in the following format:

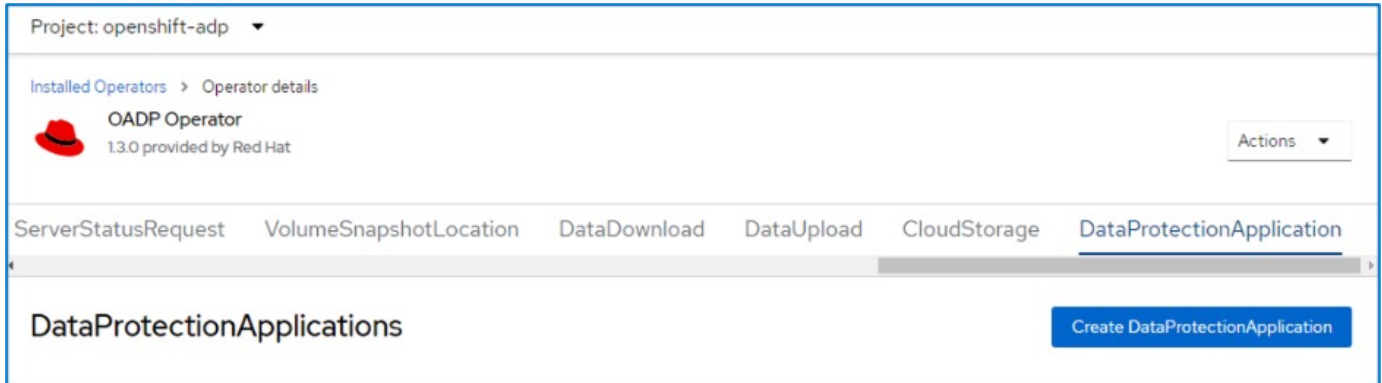
[default]
aws_access_key_id=<Access Key Id of S3 user>
aws_secret_access_key=<Secret Access Key of S3 user>
```

- Next, to configure Velero, select Installed Operators from the menu item under Operators, click on OADP operator, and then select the DataProtectionApplication tab.





Click on Create DataProtectionApplication. In the form view, provide a name for the DataProtection Application or use the default name.



Now go to the YAML view and replace the spec information as shown in the yaml file examples below.

**Sample yaml file for configuring Velero with ONTAP S3 as the backupLocation**

```

spec:
  backupLocations:
    - velero:
      config:
        insecureSkipTLSVerify: 'true' ->use this for https communication
with ONTAP S3
        profile: default
        region: us-east
        s3ForcePathStyle: 'True' ->This allows use of IP in s3URL
        s3Url: 'https://10.xx.xx.xx' ->Ensure TLS certificate for S3 is
configured
      credential:
        key: cloud
        name: ontap-s3-credentials ->previously created secret
        default: true
      objectStorage:
        bucket: velero ->Your bucket name previously created in S3 for
backups
        prefix: demobackup ->The folder that will be created in the
bucket
      provider: aws
  configuration:
    nodeAgent:
      enable: true
      uploaderType: kopia
      #default Data Mover uses Kopia to move snapshots to Object Storage
    velero:
      defaultPlugins:
        - csi ->Add this plugin
        - openshift
        - aws
        - kubevirt ->Add this plugin

```

**Sample yaml file for configuring Velero with StorageGrid S3 as the backupLocation and snapshotLocation**

```

spec:
  backupLocations:
    - velero:
      config:
        insecureSkipTLSVerify: 'true'
        profile: default
        region: us-east-1 ->region of your StorageGrid system
        s3ForcePathStyle: 'True'
        s3Url: 'https://172.21.254.25:10443' ->the IP used to access S3
      credential:
        key: cloud
        name: sg-s3-credentials ->secret created earlier
      default: true
      objectStorage:
        bucket: velero
        prefix: demobackup
      provider: aws
  configuration:
    nodeAgent:
      enable: true
      uploaderType: kopia
    velero:
      defaultPlugins:
        - csi
        - openshift
        - aws
        - kubevirt

```

The spec section in the yaml file should be configured appropriately for the following parameters similar to the example above

### backupLocations

ONTAP S3 or StorageGrid S3 (with its credentials and other information as shown in the yaml) is configured as the default BackupLocation for velero.

### snapshotLocations

If you use Container Storage Interface (CSI) snapshots, you do not need to specify a snapshot location because you will create a VolumeSnapshotClass CR to register the CSI driver. In our example, you use Astra Trident CSI and you have previously created VolumeSnapShotClass CR using the Trident CSI driver.

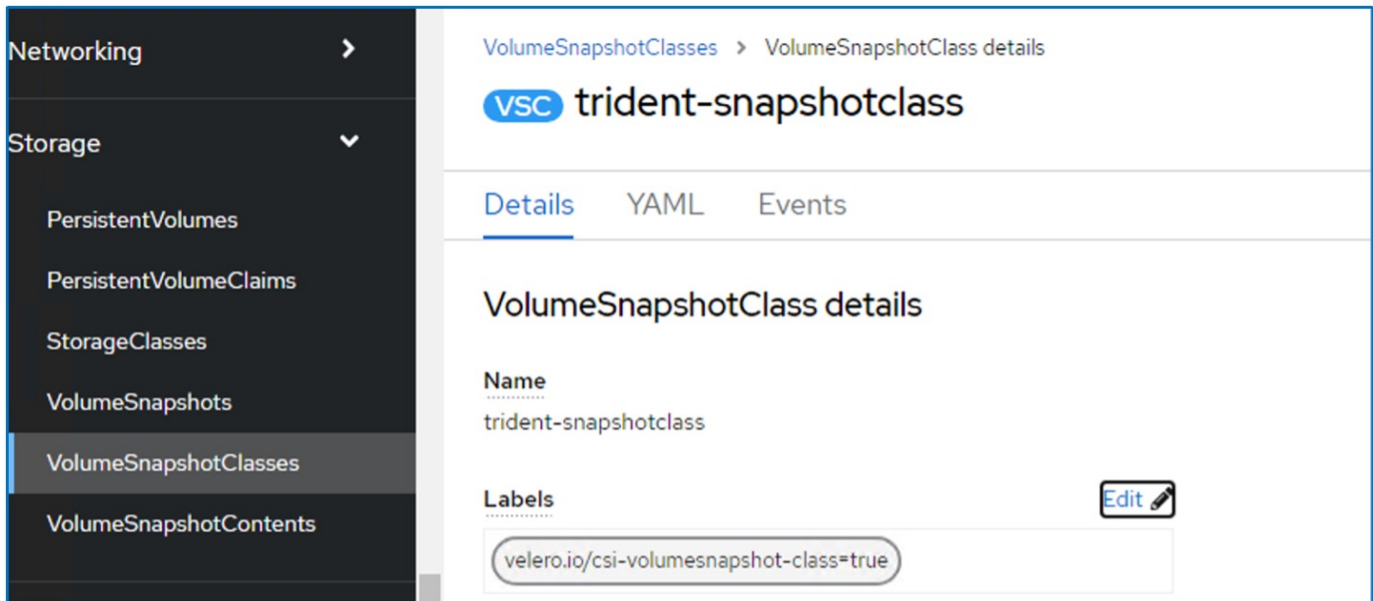
### Enable CSI plugin

Add csi to the defaultPlugins for Velero to back up persistent volumes with CSI snapshots.

The Velero CSI plugins, to backup CSI backed PVCs, will choose the VolumeSnapshotClass in the cluster that has **velero.io/csi-volumesnapshot-class** label set on it. For this

- You must have the trident VolumeSnapshotClass created.
- Edit the label of the trident-snapshotclass and set it to

`velero.io/csi-volumesnapshot-class=true` as shown below.



The screenshot displays the Kubernetes dashboard interface. On the left, a dark sidebar contains a navigation menu with 'Storage' expanded and 'VolumeSnapshotClasses' selected. The main content area shows the 'VolumeSnapshotClass details' for 'trident-snapshotclass'. The 'Name' field is 'trident-snapshotclass'. The 'Labels' field contains the label 'velero.io/csi-volumesnapshot-class=true'. There is an 'Edit' button next to the labels field.

Ensure that the snapshots can persist even if the VolumeSnapshot objects are deleted. This can be done by setting the **deletionPolicy** to Retain. If not, deleting a namespace will completely lose all PVCs ever backed up in it.

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: trident-snapshotclass
driver: csi.trident.netapp.io
deletionPolicy: Retain
```


VolumeSnapshotClasses > VolumeSnapshotClass details

**VSC** trident-snapshotclass


Details | YAML | Events

### VolumeSnapshotClass details

**Name**  
trident-snapshotclass

**Labels** Edit 

velero.io/csi-volumesnapshot-class=true



**Annotations**  
1 annotation 

**Driver**  
csi.trident.netapp.io

**Deletion policy**  
Retain

Ensure that the DataProtectionApplication is created and is in condition:Reconciled.


Installed Operators > Operator details







 **OADP Operator**  
1.3.0 provided by Red Hat Actions 

ServerStatusRequest | VolumeSnapshotLocation | DataDownload | DataUpload | CloudStorage | DataProtectionApplication

### DataProtectionApplications

Create DataProtectionApplication


Name  Search by name... /

Name 	Kind 	Status 	Labels 
 <b>velero-demo</b>	DataProtectionApplication	Condition: Reconciled	No labels 

The OADP operator will create a corresponding BackupStorageLocation. This will be used when creating a backup.

Project: openshift-adp ▾

Installed Operators > Operator details

 **OADP Operator**  
1.3.0 provided by Red Hat


Actions ▾

Repository Backup BackupStorageLocation DeleteBackupRequest DownloadRequest PodVolumeBackup PodVolumeRe

## BackupStorageLocations

Create BackupStorageLocation

Name ▾ Search by name... /

Name ↕	Kind ↕	Status ↕	Labels ↕
 <b>velero-demo-1</b>	BackupStorageLocation	Phase: Available	<ul style="list-style-type: none"> <li>app.kubernetes.io/component=bsl</li> <li>app.kubernetes.io/instance=velero-demo-1</li> <li>app.kubernetes.io/manager=oadp-oper...</li> <li>app.kubernetes.io/n...=oadp-operator-ve...</li> <li>openshift.io/oadp=True</li> <li>openshift.io/oadp-registry=True</li> </ul>

## Creating on-demand backup for VMs in OpenShift Virtualization

This section outlines how to create on-demand backup for VMs in OpenShift Virtualization.

### Steps to create a backup of a VM

To create an on-demand backup of the entire VM (VM metadata and VM disks), click on the **Backup** tab. This creates a Backup Custom Resource (CR). A sample yml is provided to create the Backup CR. Using this yml, the VM and its disks in the specified namespace will be backed up. Additional parameters can be set as shown in the [documentation](#).

A snapshot of the persistent volumes backing the disks will be created by the CSI. A backup of the VM along with the snapshot of its disks are created and stored in the backup location specified in the yml. The backup will remain in the system for 30 days as specified in the ttl.

```

apiVersion: velero.io/v1
kind: Backup
metadata:
  name: backup1
  namespace: openshift-adp
spec:
  includedNamespaces:
  - virtual-machines-demo
  snapshotVolumes: true
  storageLocation: velero-demo-1 -->this is the backupStorageLocation
  previously created
                                     when Velero is configured.
  ttl: 720h0m0s

```

Once the backup completes, its Phase will show as completed.

The screenshot shows the OpenShift console interface for the OADP Operator. The top navigation bar includes 'Project: openshift-adp', 'Installed Operators > Operator details', and an 'Actions' dropdown. Below this, there are tabs for 'Details', 'YAML', 'Subscription', 'Events', 'All instances', 'BackupRepository', 'Backup', 'BackupStorageLocation', and 'DeleteBa'. The 'Backup' tab is active, displaying a 'Backups' section with a 'Create Backup' button. A search bar is present with the text 'Search by name...'. Below the search bar is a table with columns for 'Name', 'Kind', 'Status', and 'Labels'. The table contains one entry: 'backup1' (Kind: Backup, Status: Phase: Completed, Labels: velero.io/storage-location=velero-demo-1).

You can inspect the backup in the Object storage with the help of an S3 browser application. The path of the backup shows in the configured bucket with the prefix name (velero/demobackup). You can see the contents of the backup includes the volume snapshots, logs, and other metadata of the virtual machine.



In StorageGrid, you can also use the S3 console that is available from the Tenant Manager to view the backup objects.

Path: / demobackup/ backups/ backup1/

Name	Size	Type	Last Modified	Storage Class
backup1.tar.gz	230.36 KB	GZ File	4/15/2024 10:26:29 PM	STANDARD
velero-backup.json	3.35 KB	JSON File	4/15/2024 10:26:29 PM	STANDARD
backup1-resource-list.json.gz	1.12 KB	GZ File	4/15/2024 10:26:29 PM	STANDARD
backup1-itemoperations.json.gz	600 bytes	GZ File	4/15/2024 10:26:28 PM	STANDARD
backup1-volumesnapshots.json.gz	29 bytes	GZ File	4/15/2024 10:26:28 PM	STANDARD
backup1-podvolumebackups.json.gz	29 bytes	GZ File	4/15/2024 10:26:28 PM	STANDARD
backup1-results.gz	49 bytes	GZ File	4/15/2024 10:26:28 PM	STANDARD
backup1-csi-volumesnapshotclasses.json.gz	426 bytes	GZ File	4/15/2024 10:26:28 PM	STANDARD
backup1-csi-volumesnapshotcontents.json.gz	1.43 KB	GZ File	4/15/2024 10:26:28 PM	STANDARD
backup1-csi-volumesnapshots.json.gz	1.34 KB	GZ File	4/15/2024 10:26:28 PM	STANDARD
backup1-logs.gz	13.49 KB	GZ File	4/15/2024 10:26:28 PM	STANDARD

### Creating scheduled backups for VMs in OpenShift Virtualization

To create backups on a schedule, you need to create a Schedule CR.

The schedule is simply a Cron expression allowing you to specify the time at which you want to create the backup. A sample yaml to create a Schedule CR.

```

apiVersion: velero.io/v1
kind: Schedule
metadata:
  name: <schedule>
  namespace: openshift-adp
spec:
  schedule: 0 7 * * *
  template:
    hooks: {}
    includedNamespaces:
    - <namespace>
    storageLocation: velero-demo-1
    defaultVolumesToFsBackup: true
    ttl: 720h0m0s

```

The Cron expression 0 7 \* \* \* means a backup will be created at 7:00 every day.


The namespaces to be included in the backup and the storage location for the backup are also specified. So instead of a Backup CR, Schedule CR is used to create a backup at the specified time and frequency.

Once the schedule is created, it will be Enabled.



Project: openshift-adp ▾



Installed Operators > Operator details

 **OADP Operator**  
1.3.0 provided by Red Hat

storageLocation DeleteBackupRequest DownloadRequest PodVolumeBackup PodVolumeRestore Restore **Schedule**

## Schedules


Name ▾ Search by name... /

Name	Kind	Status	Labels
 schedule1	Schedule	Phase:  Enabled	No labels

Backups will be created according to this schedule, and can be viewed from the Backup tab.

Project: openshift-adp ▾

Installed Operators > Operator details


 **OADP Operator**  
1.3.0 provided by Red Hat

Events All instances BackupRepository **Backup** BackupStorageLocation DeleteBackupRequest DownloadRequest

## Backups

[Create Backup](#)

Name ▾ Search by name... /

Name	Kind	Status	Labels
 schedule1-20240416140507	Backup	Phase: InProgress	<span>velero.io/schedule-name=schedule1</span> <span>velero.io/storage-location=velero-demo-1</span>

## Restore a VM from a backup

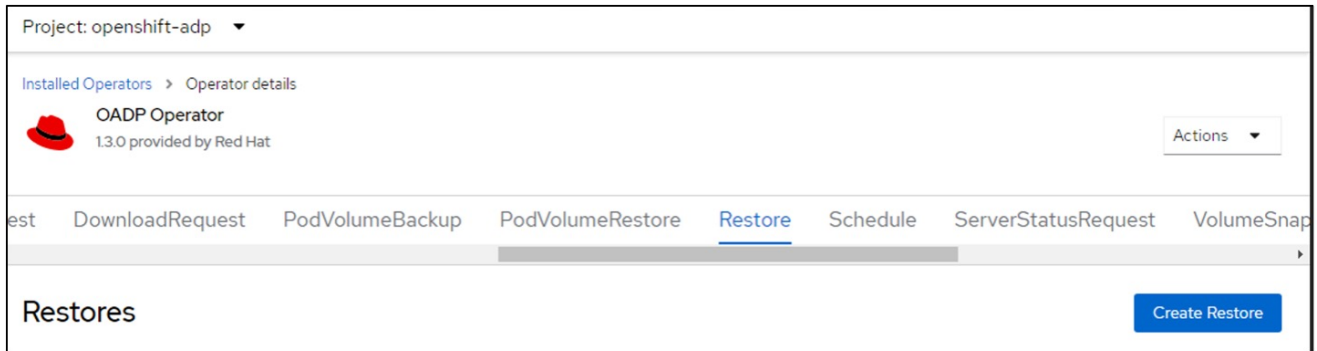
This section describes how to restore virtual machine(s) from a backup.

### Prerequisites

To restore from a backup, let us assume that the namespace where the virtual machine existed got accidentally deleted.

## Restore to the same namespace

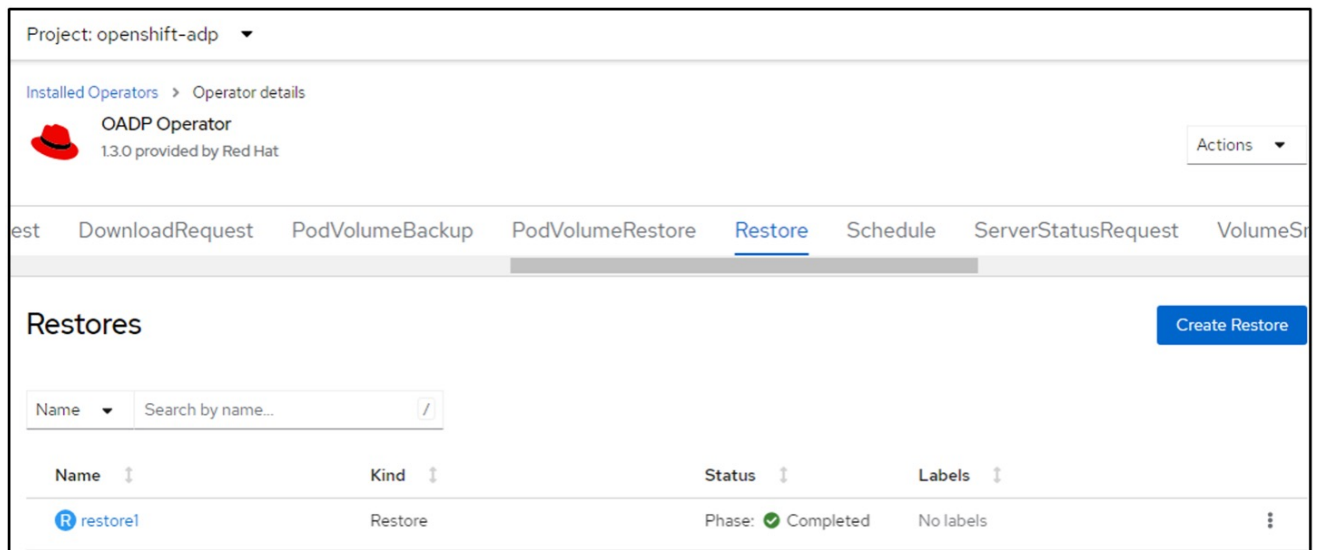
To restore from the backup that we just created, we need to create a Restore Custom Resource (CR). We need to provide it a name, provide the name of the backup that we want to restore from and set the restorePVs to true. Additional parameters can be set as shown in the [documentation](#). Click on Create button.





The screenshot shows the OADP Operator interface. At the top, it says "Project: openshift-adp". Below that, it shows "Installed Operators > Operator details" for the "OADP Operator" (13.0 provided by Red Hat). A navigation bar includes "DownloadRequest", "PodVolumeBackup", "PodVolumeRestore", "Restore" (selected), "Schedule", "ServerStatusRequest", and "VolumeSnap". Below the navigation bar, the "Restores" section is visible with a "Create Restore" button.

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: restore1
  namespace: openshift-adp
spec:
  backupName: backup1
  restorePVs: true
```

When the phase shows completed, you can see that the virtual machines have been restored to the state when the snapshot was taken. (If the backup was created when the VM was running, restoring the VM from the backup will start the restored VM and bring it to a running state). The VM is restored to the same namespace.



The screenshot shows the OADP Operator interface with the "Restores" section. A search bar is present with "Name" and "Search by name...". Below it, a table lists the restore operation:

Name	Kind	Status	Labels
 restore1	Restore	Phase:  Completed	No labels

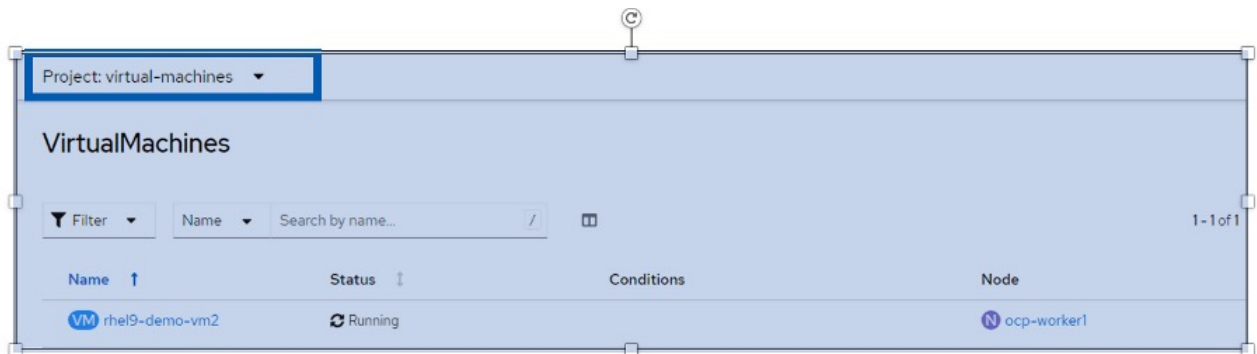
## Restore to a different namespace

To restore the VM to a different namespace, you can provide a namespaceMapping in the yaml definition of the Restore CR.




The following sample yaml file creates a Restore CR to restore a VM and its disks in the virtual-machines-demo namespace when the backup was taken to the virtual-machines namespace.

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: restore-to-different-ns
  namespace: openshift-adp
spec:
  backupName: backup
  restorePVs: true
  includedNamespaces:
  - virtual-machines-demo
  namespaceMapping:
    virtual-machines-demo: virtual-machines
```

When the phase shows completed, you can see that the virtual machines have been restored to the state when the snapshot was taken. (If the backup was created when the VM was running, restoring the VM from the backup will start the restored VM and bring it to a running state). The VM is restored to a different namespace as specified in the yaml.



The screenshot shows the OpenShift console interface for the 'virtual-machines' project. The 'VirtualMachines' section is active, displaying a table with one entry. The table has columns for Name, Status, Conditions, and Node. The entry shows a VM named 'rhe19-demo-vm2' with a status of 'Running' and is located on the 'ocp-worker1' node.

Name ↑	Status ↓	Conditions	Node
 rhe19-demo-vm2	 Running		 ocp-worker1

## Restore to a different storage class

Velero provides a generic ability to modify the resources during restore by specifying json patches. The json patches are applied to the resources before they are restored. The json patches are specified in a configmap and the configmap is referenced in the restore command. This feature enables you to restore using different storage class.

In the example below, the virtual machine, during creation uses ontap-nas as the storage class for its disks. A backup of the virtual machine named backup1 is created.

The screenshot shows the configuration page for a virtual machine named 'rhel9-demo-vm1' in the 'virtual-machines-demo' project. The 'Disks' section is active, displaying a table of disks. The 'disk1' and 'rootdisk' are both backed up from PVCs named 'rhel9-demo-vm1-disk1' and 'rhel9-demo-vm1' respectively, both using the 'ontap-nas' storage class.

Name	Source	Size	Drive	Interface	Storage class
cloudinitdisk	Other	-	Disk	virtio	-
disk1	PVC rhel9-demo-vm1-disk1	31.75 GiB	Disk	virtio	ontap-nas
rootdisk	PVC rhel9-demo-vm1	31.75 GiB	Disk	virtio	ontap-nas

The screenshot shows the backup details for the OADP Operator in the 'openshift-adp' project. A backup named 'backup1' is listed with a status of 'Completed'.

Name	Kind	Status
backup1	Backup	Phase: Completed

Simulate a loss of the VM by deleting the VM.

To restore the VM using a different storage class, for example, ontap-nas-eco storage class, you need to do the following two steps:

### Step 1

Create a config map (console) in the openshift-adp namespace as follows:

Fill in the details as shown in the screenshot:

Select namespace : openshift-adp

Name: change-storage-class-config (can be any name)

Key: change-storage-class-config.yaml:

Value:

```
version: v1
resourceModifierRules:
- conditions:
  groupResource: persistentvolumeclaims
  resourceNameRegex: "^rhel*"
  namespaces:
  - virtual-machines-demo
patches:
- operation: replace
  path: "/spec/storageClassName"
  value: "ontap-nas-eco"
```

The screenshot shows the OpenShift console interface for editing a ConfigMap. The page title is 'Edit ConfigMap' and it includes a description: 'Config maps hold key-value pairs that can be used in pods to read application configuration.' The 'Configure via' section has 'Form view' selected. The 'Name' field contains 'change-storage-class-config'. There is an 'Immutable' checkbox which is unchecked. The 'Data' section has a 'Remove key/value' button. The 'Key' field contains 'change-storage-class-config.yaml'. The 'Value' field contains the YAML content from the previous block. There is a 'Browse...' button next to the value field. At the bottom, there is an 'Add key/value' button. The left sidebar shows the navigation menu with 'ConfigMaps' selected under the 'Virtualization' section.

The resulting config map object should look like this (CLI):

```

# kubectl describe cm/change-storage-class-config -n openshift-
adp
Name:          change-storage-class-config
Namespace:     openshift-adp
Labels:        velero.io/change-storage-class=RestoreItemAction
               velero.io/plugin-config=
Annotations:   <none>

Data
====
change-storage-class-config.yaml:
----
version: v1
resourceModifierRules:
- conditions:
    groupResource: persistentvolumeclaims
    resourceNameRegex: "^rhel*"
    namespaces:
    - virtual-machines-demo
patches:
- operation: replace
  path: "/spec/storageClassName"
  value: "ontap-nas-eco"

BinaryData
====

Events:   <none>

```

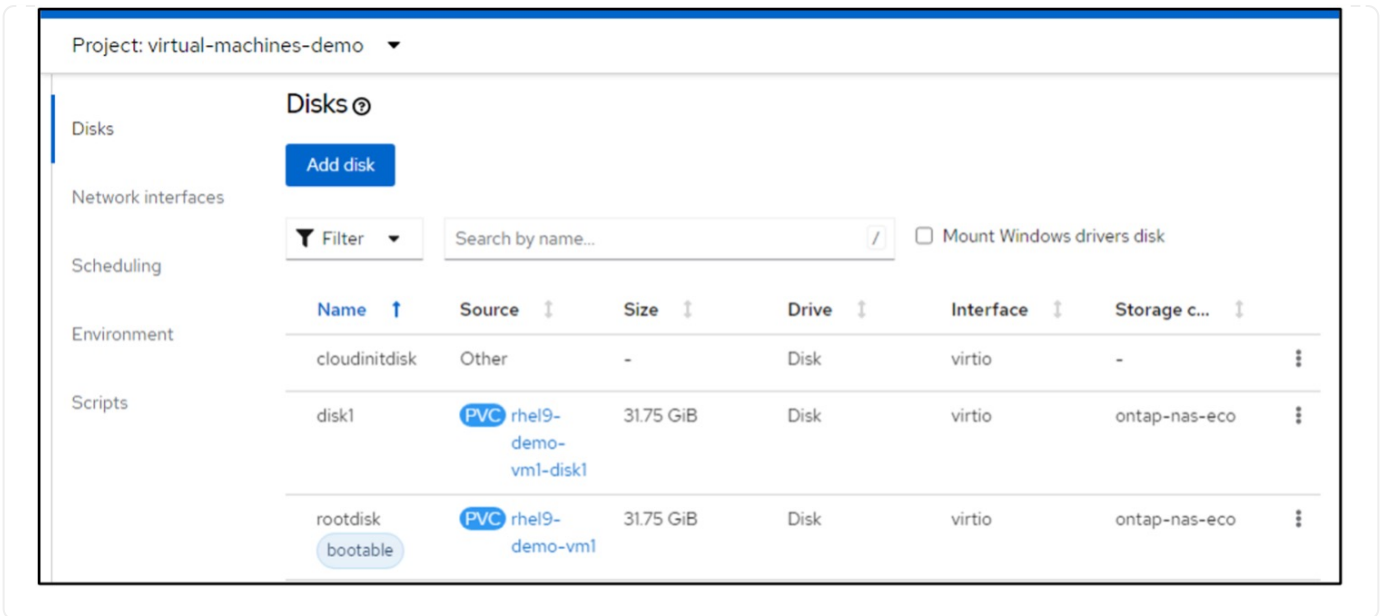
This config map will apply the resource modifier rule when the restore is created. A patch will be applied to replace the storage class name to ontap-nas-eco for all persistent volume claims starting with rhel.

## Step 2

To restore the VM use the following command from the Velero CLI:

```
#velero restore create restore1 --from-backup backup1 --resource
-modifier-configmap change-storage-class-config -n openshift-adp
```

The VM is restored in the same namespace with the disks created using the storage class ontap-nas-eco.



## Deleting backups and restores in using Velero

This section outlines how to delete backups and restores for VMs in OpenShift Virtualization using Velero.

### Deleting a backup

You can delete a Backup CR without deleting the Object Storage data by using the OC CLI tool.

```
oc delete backup <backup_CR_name> -n <velero_namespace>
```

If you want to delete the Backup CR and delete the associated object storage data, you can do so by using the Velero CLI tool.

Download the CLI as given in the instructions in the [Velero documentation](#).

Execute the following delete command using the Velero CLI

```
velero backup delete <backup_CR_name> -n <velero_namespace>
```

You can also delete the Restore CR using the Velero CLI

```
velero restore delete restore --namespace openshift-adp
```

You can use oc command as well as the UI to delete the restore CR

```
oc delete backup <backup_CR_name> -n <velero_namespace>
```

## Monitoring using Cloud Insights

### Monitoring using Cloud Insights for VMs in Red Hat OpenShift Virtualization

Author: Banu Sundhar, NetApp

This section of the reference document provides details for integrating NetApp Cloud Insights with a Red Hat OpenShift Cluster to monitor OpenShift Virtualization VMs.

NetApp Cloud Insights is a cloud infrastructure monitoring tool that gives you visibility into your complete infrastructure. With Cloud Insights, you can monitor, troubleshoot, and optimize all your resources including your public clouds and your private data centers. For more information about NetApp Cloud Insights, refer to the [Cloud Insights documentation](#).

To start using Cloud Insights, you must sign up on the NetApp BlueXP portal. For details, refer to the [Cloud Insights Onboarding](#)

Cloud Insights has several features that enable you to quickly and easily find data, troubleshoot issues, and provide insights into your environment. You can find data easily with powerful queries, you can visualize data in dashboards, and send email alerts for data thresholds you set. Refer to the [video tutorials](#) to help you understand these features.

For Cloud Insights to start collecting data you need the following

#### Data Collectors

There are 3 types of Data Collectors:

- \* Infrastructure (storage devices, network switches, compute infrastructure)
- \* Operating Systems (such as VMware or Windows)
- \* Services (such as Kafka)

Data Collectors discover information from the data sources, such as ONTAP storage device (infrastructure data collector). The information gathered is used for analysis, validation, monitoring, and troubleshooting.

#### Acquisition Unit

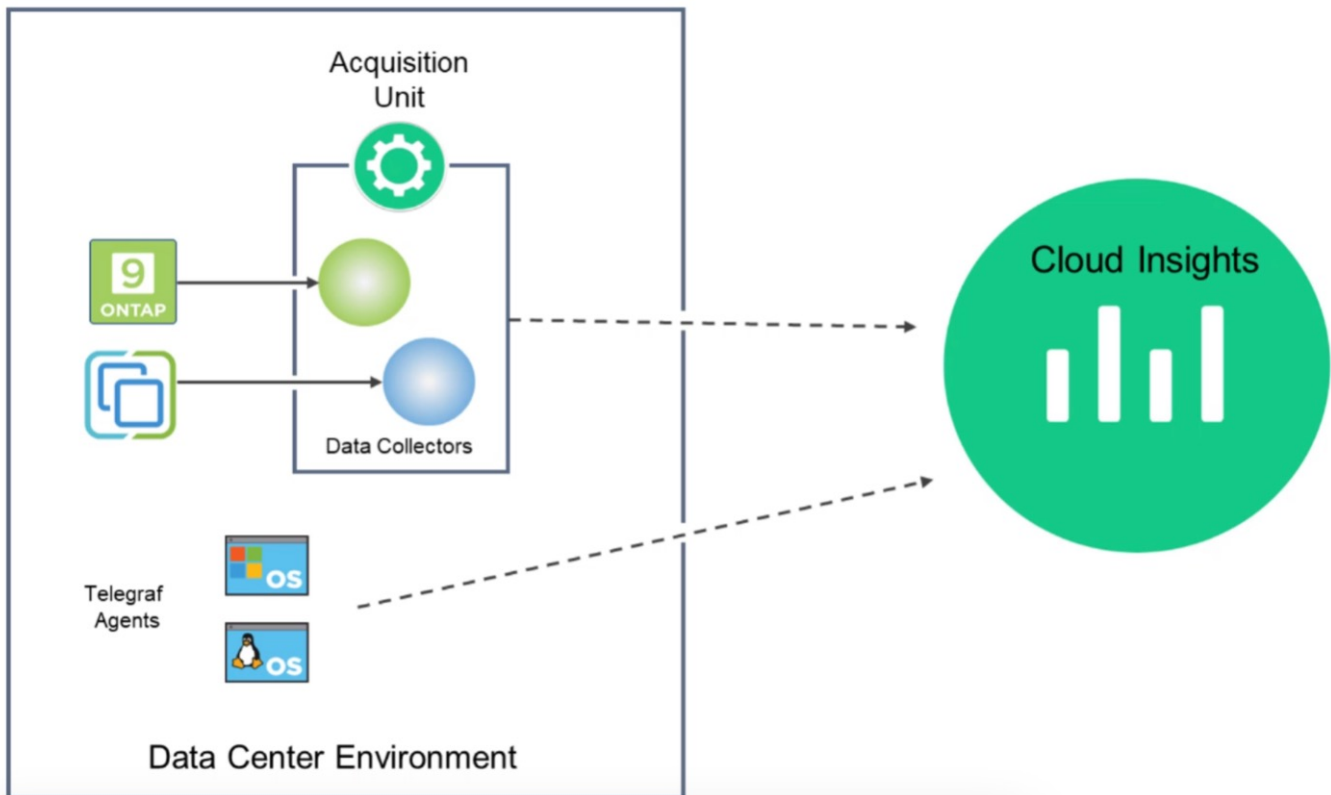
If you are using an infrastructure Data Collector, you also need an Acquisition Unit to inject data into Cloud Insights. An Acquisition Unit is a computer dedicated to hosting data collectors, typically a Virtual Machine. This computer is typically located in the same data center/VPC as the monitored items.

#### Telegraf Agents

Cloud Insights also supports Telegraf as its agent for collection of integration data. Telegraf is a plugin-driven server agent that can be used to collect and report metrics, events, and logs.

Cloud Insights Architecture





## Integration with Cloud Insights for VMs in Red Hat OpenShift Virtualization

To start collecting data for VMs in OpenShift Virtualization you will need to install:

1. A Kubernetes monitoring operator and data collector to collect Kubernetes data  
For complete instructions, refer to the [documentation](#).
2. An acquisition unit to collect data from ONTAP storage that provides persistent storage for the VM disks  
For complete instructions, refer to the [documentation](#).
3. A data collector for ONTAP  
For complete instructions, refer to the [documentation](#)

Additionally, if you are using StorageGrid for VM backups, you need a data collector for the StorageGRID as well.

## Sample Monitoring capabilities for VMs in Red Hat OpenShift Virtualization

This section discusses monitoring using Cloud Insights for VMs in Red Hat OpenShift Virtualization.

### Monitoring based on events and creating Alerts

Here is a sample where the namespace that contains a VM in OpenShift Virtualization is monitored based on events. In this example, a monitor is created based on `logs.kubernetes.event` for the specified namespace in the cluster.

The screenshot shows the 'Edit log monitor' configuration in the NetApp Observability console. The breadcrumb path is 'NetApp PCS Sandbox / Observability / Alerts / Manage Monitors / Monitor virtual-machines-demo-ns'. A yellow warning banner states: 'Filter/Advanced Query and Group by in section 1 must not be empty. If alert resolution is based on log entry, section 3 filter/advanced query also must not be empty.'

**1 Select the log to monitor**

Log Source: logs.kubernetes.event

Filter By: kubernetes\_cluster ocp-cluster4, involvedobject.namespace virtual-machines-demo

Group By: reason

27 Items found

timestamp ↓	type	source	message
04/19/2024 10:31:18 AM	logs.kubernetes.event	kubernetes_cluster:ocp-cluster4;namespace:cloudi	VirtualMachineInstance started.
04/19/2024 10:31:18 AM	logs.kubernetes.event	kubernetes_cluster:ocp-cluster4;namespace:cloudi	VirtualMachineInstance defined.

**2 Define alert behavior**

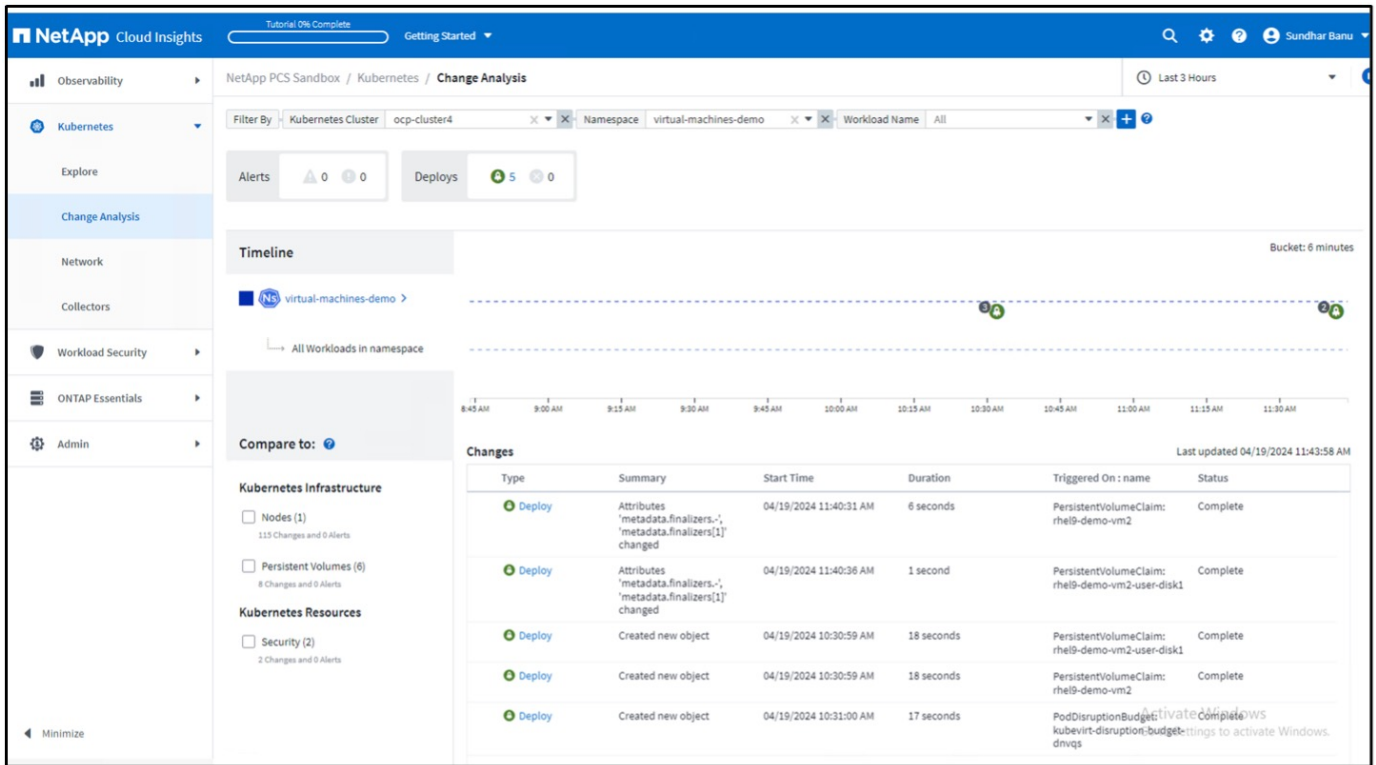
Create an alert at severity Warning when the conditions above occur 1 time

This query provides all the events for the virtual machine in the namespace. (There is only one virtual machine in the namespace). An advanced query can also be constructed to filter based on the event where the reason is “failed” or “FailedMount” These events are typically created when there is an issue in creating a PV or mounting the PV to a pod indicating issues in the dynamic provisioner for creating persistent volumes for the VM.

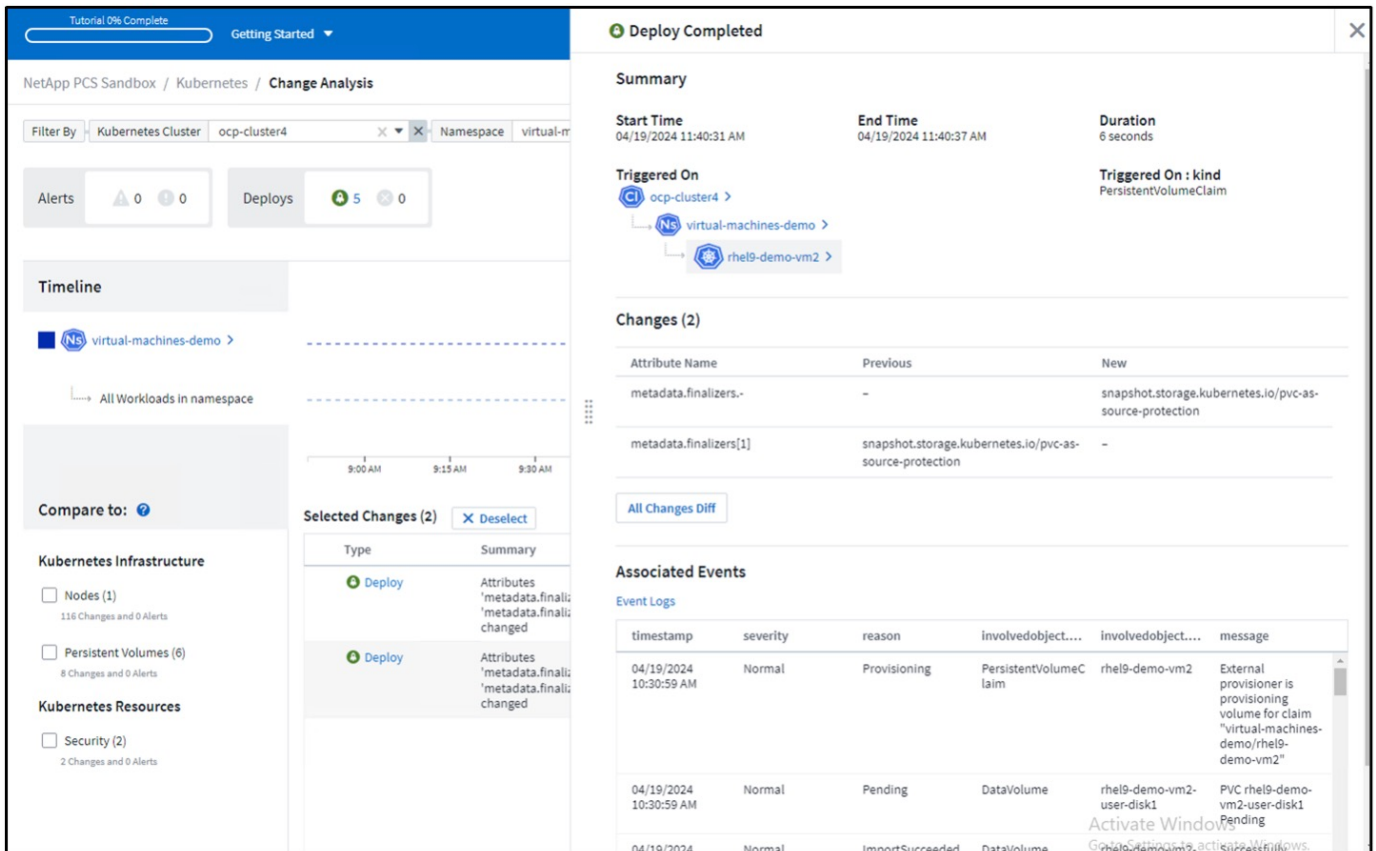
While creating the Alert Monitor as shown above, you can also configure notification to recipients. You can also provide corrective actions or additional information that can be useful to resolve the error. In the above example, additional information could be to look into the Trident backend configuration and storage class definitions for resolving the issue.

### Change Analytics

With Change Analytics, you can get a view of what changed in the state of your cluster including who made that change which can help in troubleshooting issues.



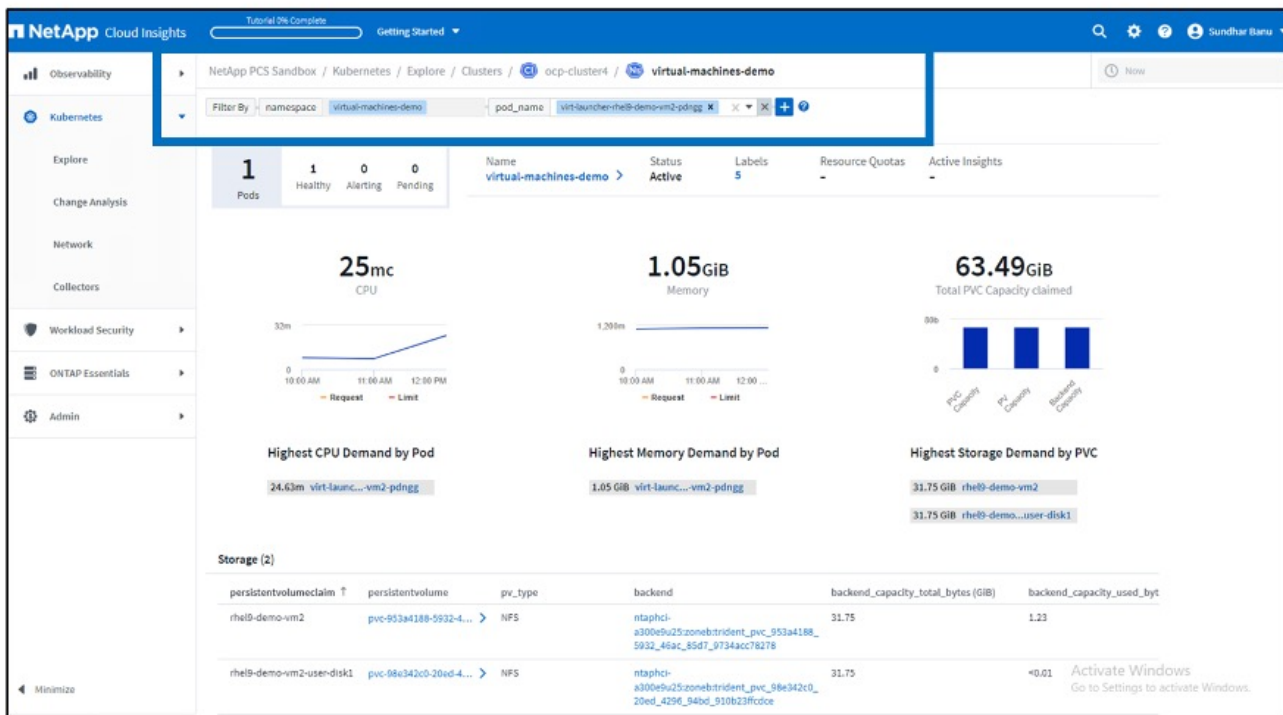
In the above example, Change Analysis is configured on the OpenShift cluster for the namespace that contains an OpenShift Virtualization VM. The dashboard shows changes against the timeline. You can drill down to see what changed and the click on All Changes Diff to see the diff of the manifests. From the manifest, you can see that a new backup of the persistent disks was created.



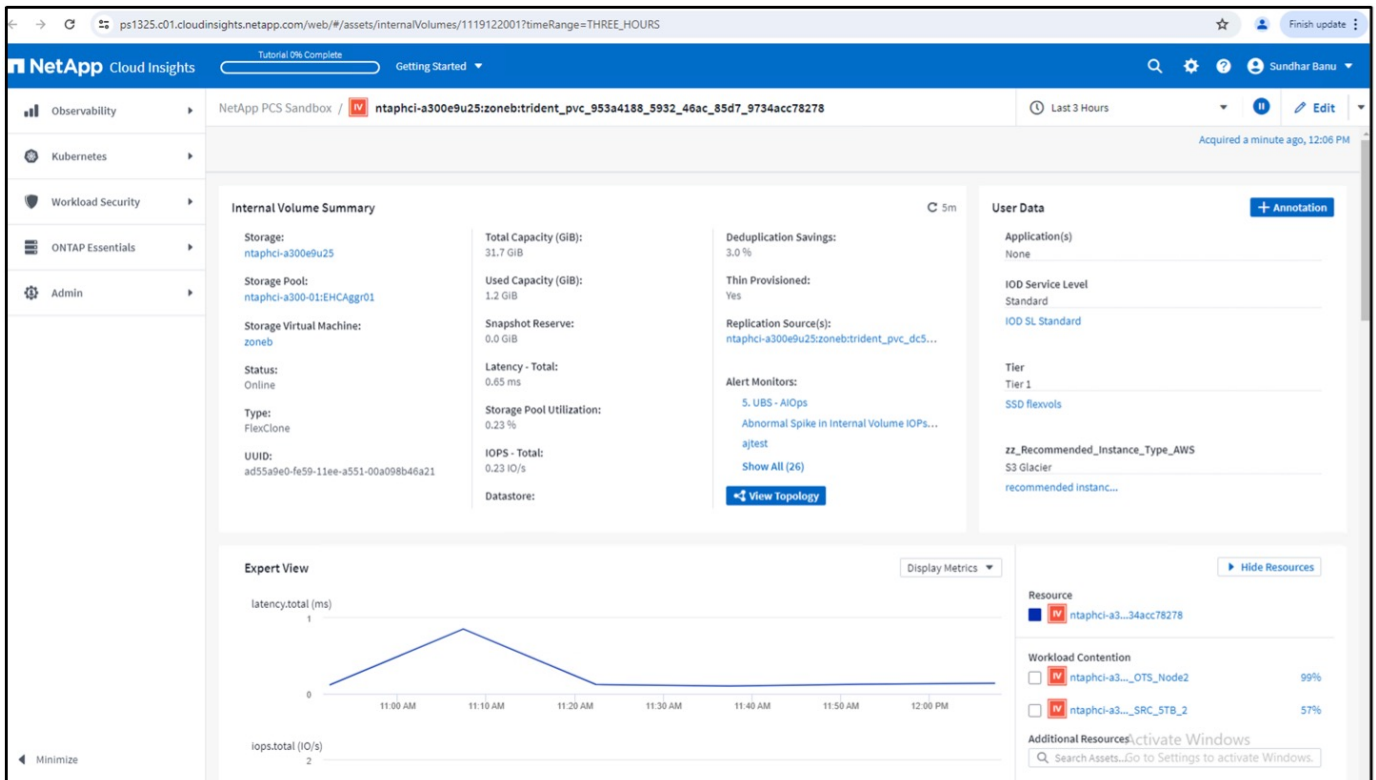
All Changes Diff			
Previous		New	
<b>Expand 45 lines ...</b>			
46	kind: DataVolume	46	kind: DataVolume
47	name: rhel9-demo-vm2	47	name: rhel9-demo-vm2
48	uid: dcf93b7a-71bc-409b-ad12-4916d05e0980	48	uid: dcf93b7a-71bc-409b-ad12-4916d05e0980
49	- resourceVersion: "8569671"	49	+ resourceVersion: "8619670"
50	uid: 953a4188-5932-46ac-85d7-9734acc78278	50	uid: 953a4188-5932-46ac-85d7-9734acc78278
51	spec:	51	spec:
52	accessModes:	52	accessModes:
<b>Expand 15 lines ...</b>			

## Backend Storage Mapping

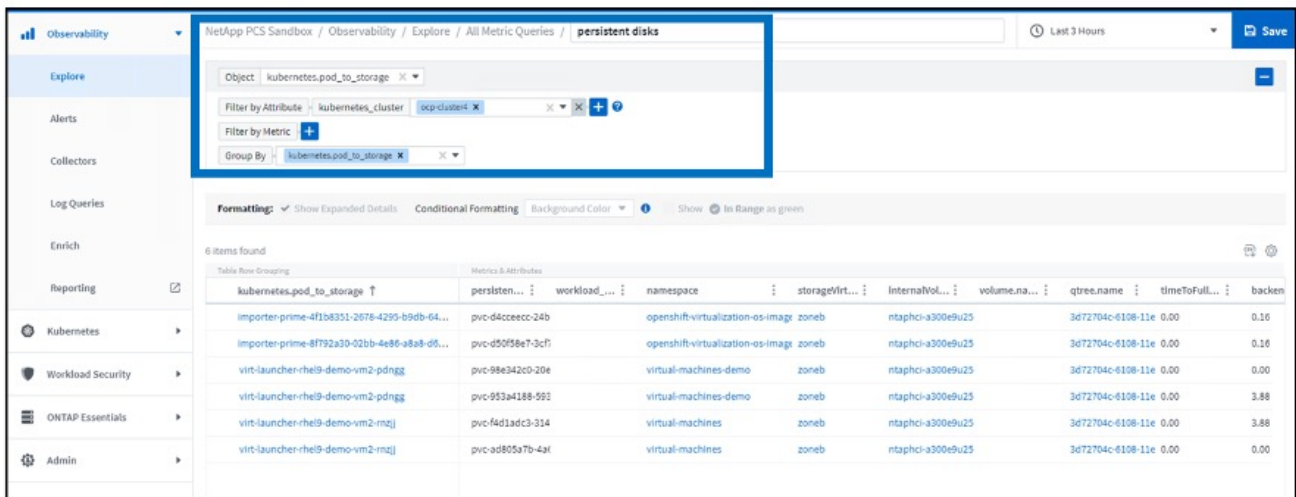
With Cloud Insights, you can easily see the backend storage of the VM disks and several statistics about the PVCs.



You can click on the links under the backend column, which will pull data directly from the backend ONTAP storage.



Another way to look at all the pod to storage mapping is creating an All Metrics query From Observability menu under Explore.



Clicking on any of the links will give you the corresponding details from ONTP storage. For example, clicking on an SVM name in the storageVirtualMachine column will pull details about the SVM from ONTAP. Clicking on an internal volume name will pull details about the volume in ONTAP.

storageVirtualMachin...	internalVolume.name	volume.na..
zation-os-image zoneb		ntaphci-a300e9u25:zoneb:trident_p
zation-os-image zoneb		ntaphci-a300e9u25:zoneb:trident_p
demo zoneb		ntaphci-a300e9u25:zoneb:trident_p
demo zoneb		ntaphci-a300e9u25:zoneb:trident_p
	zoneb	ntaphci-a300e9u25:zoneb:trident_p
	zoneb	ntaphci-a300e9u25:zoneb:trident_p

The screenshot displays the NetApp PCS Sandbox interface for a Storage Virtual Machine (SVM) in the 'zoneb' namespace. The main view is the 'Storage Virtual Machine Summary' page, which includes the following details:

- Internal Volume Link:** N/A
- Capacity (GB):** 1,874.4 GB
- Used Capacity (GB):** 107.6 GB
- IOPS - Total:** 26.21 IOPS
- Latency - Total:** 0.24 ms
- Comment:**
- URI:** 13348361-c8b0-11e6-8100-000990949423
- Alert Monitors:**

The 'Expert View' section shows two line graphs over time (from 9:45 AM to 12:15 PM):

- latency\_total (ms):** A line graph showing latency fluctuating between approximately 0.15 ms and 0.30 ms.
- io\_persec (IOPS):** A line graph showing IOPS fluctuating between approximately 20 and 40.

Additional information includes:

- Resource:** zoneb
- Top Cert Buttons:**  ntaphci-a3-43-nsvc01 (97%)
- Additional Resources:** Search Resources...

The bottom screenshot shows a more detailed view of the SVM, including:

- Total Capacity (GB):** 26,70.8
- Used Capacity (GB):** 107.0 GB
- Storage Pool:** ntaphci-a300e9u25:zoneb
- Storage Virtual Machine:** zoneb
- Status:** Online
- Type:** Flash
- URI:** 13348361-c8b0-11e6-8100-000990949423
- IOPS - Total:** 26.21 IOPS
- Latency - Total:** 0.24 ms
- Storage Pool Utilization:** 0.23%
- IOPS - Total:** 26.21 IOPS
- Latency - Total:** 0.24 ms
- Storage Pool Utilization:** 0.23%
- IOPS - Total:** 26.21 IOPS
- Latency - Total:** 0.24 ms
- Storage Pool Utilization:** 0.23%

The 'Expert View' at the bottom shows a line graph for 'latency\_total (ms)' fluctuating between 0.15 ms and 0.30 ms.

# Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

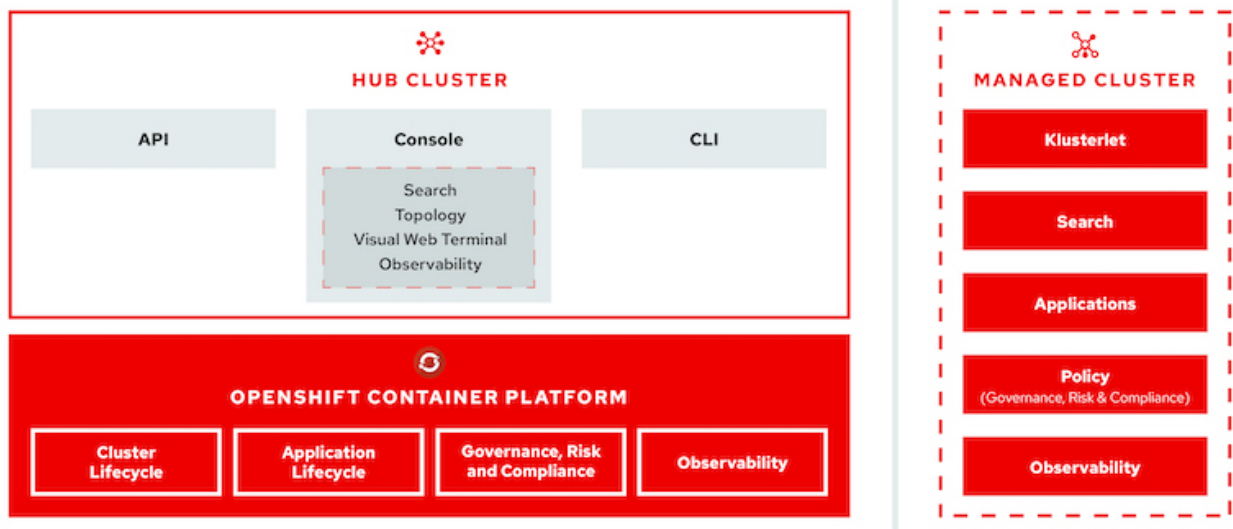
## Advanced Cluster Management for Kubernetes: Red Hat OpenShift with NetApp - Overview

As a containerized application transitions from development to production, many organizations require multiple Red Hat OpenShift clusters to support the testing and deployment of that application. In conjunction with this, organizations usually host multiple applications or workloads on OpenShift clusters. Therefore, each organization ends up managing a set of clusters, and OpenShift administrators must thus face the added challenge of managing and maintaining multiple clusters across a range of environments that span multiple on-premises data centers and public clouds. To address these challenges, Red Hat introduced Advanced Cluster Management for Kubernetes.

Red Hat Advanced Cluster Management for Kubernetes enables you to perform the following tasks:

1. Create, import, and manage multiple clusters across data centers and public clouds
2. Deploy and manage applications or workloads on multiple clusters from a single console
3. Monitor and analyze health and status of different cluster resources
4. Monitor and enforce security compliance across multiple clusters

Red Hat Advanced Cluster Management for Kubernetes is installed as an add-on to a Red Hat OpenShift cluster, and it uses this cluster as a central controller for all its operations. This cluster is known as hub cluster, and it exposes a management plane for the users to connect to Advanced Cluster Management. All the other OpenShift clusters that are either imported or created via the Advanced Cluster Management console are managed by the hub cluster and are called managed clusters. It installs an agent called Klusterlet on the managed clusters to connect them to the hub cluster and serve the requests for different activities related to cluster lifecycle management, application lifecycle management, observability, and security compliance.



For more information, see the documentation [here](#).

## Deployment

### Deploy Advanced Cluster Management for Kubernetes

This section covers advanced cluster management for Kubernetes on Red Hat OpenShift with NetApp.

#### Prerequisites

1. A Red Hat OpenShift cluster (greater than version 4.5) for the hub cluster
2. Red Hat OpenShift clusters (greater than version 4.4.3) for managed clusters
3. Cluster-admin access to the Red Hat OpenShift cluster
4. A Red Hat subscription for Advanced Cluster Management for Kubernetes

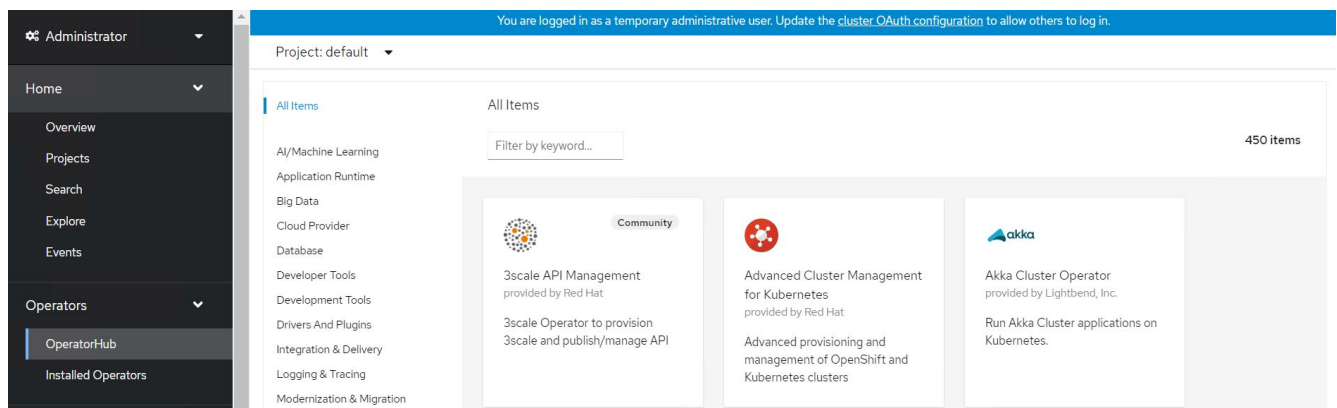
Advanced Cluster Management is an add-on on for the OpenShift cluster, so there are certain requirements and restrictions on the hardware resources based on the features used across the hub and managed clusters. You need to take these issues into account when sizing the clusters. See the documentation [here](#) for more details.

Optionally, if the hub cluster has dedicated nodes for hosting infrastructure components and you would like to install Advanced Cluster Management resources only on those nodes, you need to add tolerations and selectors to those nodes accordingly. For more details, see the documentation [here](#).

### Deploy Advanced Cluster Management for Kubernetes

To install Advanced Cluster Management for Kubernetes on an OpenShift cluster, complete the following steps:

1. Choose an OpenShift cluster as the hub cluster and log into it with cluster-admin privileges.
2. Navigate to Operators > Operators Hub and search for Advanced Cluster Management for Kubernetes.



3. Select Advanced Cluster Management for Kubernetes and click Install.





# Advanced Cluster Management for Kubernetes

2.2.3 provided by Red Hat



Install

## Latest version

2.2.3

## Capability level

- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

## Provider type

Red Hat

## Provider

Red Hat

## Infrastructure features

Disconnected

Red Hat Advanced Cluster Management for Kubernetes provides the multicluster hub, a central management console for managing multiple Kubernetes-based clusters across data centers, public clouds, and private clouds. You can use the hub to create Red Hat OpenShift Container Platform clusters on selected providers, or import existing Kubernetes-based clusters. After the clusters are managed, you can set compliance requirements to ensure that the clusters maintain the specified security requirements. You can also deploy business applications across your clusters.

Red Hat Advanced Cluster Management for Kubernetes also provides the following operators:

- **Multicluster subscriptions:** An operator that provides application management capabilities including subscribing to resources from a channel and deploying those resources on MCH-managed Kubernetes clusters based on placement rules.
- **Hive for Red Hat OpenShift:** An operator that provides APIs for provisioning and performing initial configuration of OpenShift clusters. These operators are used by the multicluster hub to provide its provisioning and application-management capabilities.

## How to Install

Use of this Red Hat product requires a licensing and subscription agreement.

4. On the Install Operator screen, provide the necessary details (NetApp recommends retaining the default parameters) and click Install.

## Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

### Update channel \*

- release-2.0
- release-2.1
- release-2.2

### Installation mode \*

- All namespaces on the cluster (default)  
This mode is not supported by this Operator
- A specific namespace on the cluster  
Operator will be available in a single Namespace only.

### Installed Namespace \*

- Operator recommended Namespace: **PR** open-cluster-management

#### **i** Namespace creation

Namespace **open-cluster-management** does not exist and will be created.

- Select a Namespace

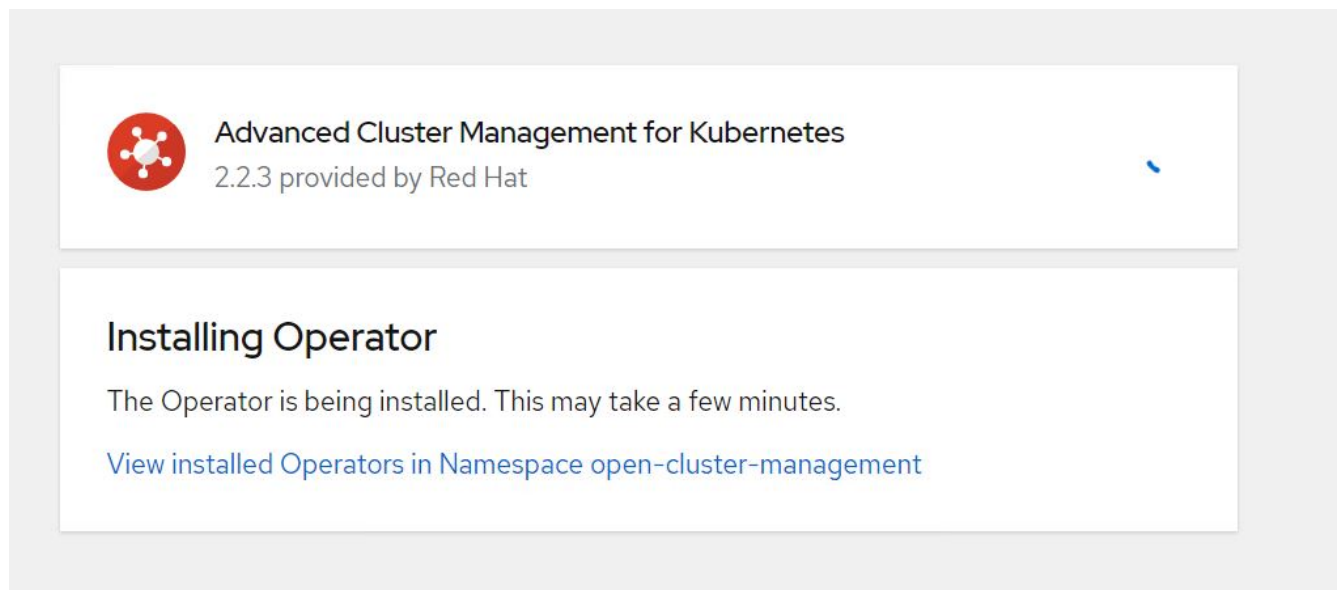
### Approval strategy \*


- Automatic
- Manual

**Install**

Cancel

5. Wait for the operator installation to complete.



 **Advanced Cluster Management for Kubernetes**  
2.2.3 provided by Red Hat

### Installing Operator

The Operator is being installed. This may take a few minutes.

[View installed Operators in Namespace open-cluster-management](#)

6. After the operator is installed, click Create MultiClusterHub.



Advanced Cluster Management for Kubernetes  
2.2.3 provided by Red Hat



## Installed operator - operand required

The Operator has installed successfully. Create the required custom resource to be able to use this Operator.

**MCH** MultiClusterHub **Required**

Advanced provisioning and management of OpenShift and Kubernetes clusters

Create MultiClusterHub

[View installed Operators in Namespace open-cluster-management](#)

7. On the Create MultiClusterHub screen, click Create after furnishing the details. This initiates the installation of a multi-cluster hub.

Project: open-cluster-management

Advanced Cluster Management for Kubernetes > Create MultiClusterHub

### Create MultiClusterHub

Create by completing the form. Default values may be provided by the Operator authors.

Configure via:  Form view  YAML view

**Note:** Some fields may not be represented in this form view. Please select "YAML view" for full control.



MultiClusterHub

provided by Red Hat

MultiClusterHub defines the configuration for an instance of the MultiCluster Hub

Name \*

multiclusterhub

Labels

app=frontend

> Advanced configuration




Create

Cancel

8. After all the pods move to the Running state in the open-cluster-management namespace and the operator moves to the Succeeded state, Advanced Cluster Management for Kubernetes is installed.


## Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#). Or create an Operator and ClusterServiceVersion using the [Operator SDK](#).

Name	Managed Namespaces	Status	Provided APIs
 <b>Advanced Cluster Management for Kubernetes</b> 2.2.3 provided by Red Hat	 open-cluster-management	 Succeeded Up to date	MultiClusterHub ClusterManager ClusterDeployment ClusterState <a href="#">View 25 more...</a>

9. It takes some time to complete the hub installation, and, after it is done, the MultiCluster hub moves to Running state.

Installed Operators > Operator details




**Advanced Cluster Management for Kubernetes**  
 2.2.3 provided by Red Hat

Actions

[Details](#)
[YAML](#)
[Subscription](#)
[Events](#)
[All instances](#)
[MultiClusterHub](#)
[ClusterManager](#)
[ClusterDeployment](#)
[ClusterSt...](#)

### MultiClusterHubs

Create MultiClusterHub

Name	Kind	Status	Labels
 multiclusterhub	MultiClusterHub	Phase:  Running	No labels




10. It creates a route in the open-cluster-management namespace. Connect to the URL in the route to access the Advanced Cluster Management console.

## Routes

Create Route

Filter Name mul

Name mul Clear all filters

Name	Status	Location	Service
 multcloud-console	 Accepted	<a href="https://multicloud-console.apps.ocp-vmware2.cie.netapp.com">https://multicloud-console.apps.ocp-vmware2.cie.netapp.com</a>	 management-ingress

## Features

### Cluster Lifecycle Management

To manage different OpenShift clusters, you can either create or import them into Advanced Cluster Management.

1. First navigate to Automate Infrastructures > Clusters.
2. To create a new OpenShift cluster, complete the following steps:
  - a. Create a provider connection: Navigate to Provider Connections and click Add a Connection, provide all the details corresponding to the selected provider type and click Add.

Select a provider and enter basic information

Provider \* ⓘ

aws Amazon Web Services

Connection name \* ⓘ

nik-hcl-aws

Namespace \* ⓘ

default

Configure your provider connection

Base DNS domain ⓘ

cie.netapp.com

AWS access key ID \* ⓘ

AKIATCFBZDOIASDSA

AWS secret access key \* ⓘ

.....

Red Hat OpenShift pull secret \* ⓘ

```
FuS3pNbktVaHplNFc2MkZsbmtBVG6N6TKtmUIZXcHcxOW9teEZwQ0lYzld3cjJobGxJeDBON0xlZE0yeGM5O0ZwZk5RR2JUanlxNnNUM2lRb0FJbUFjNCIBYlpEWVZE0HitNkxTMDZPUVpoWFRhcGwtREIDQ2RSYlJRaTlxblDL2oyQ3pVeUJfNllwcENSa2YyOUsyLWZGSFVfNA==", "email": "Nikhil.kulkarni@netapp.com"}, "registry.redhat.io":
```

SSH private key \* ⓘ

```
-----BEGIN OPENSSH PRIVATE KEY-----  
b3BlbnNzaC1rZXktZjEAAAAABG5vbmUAAAAEbasdadssadm9uZQAAAAAAAAABAAAAmWAAAAAtzc2gtZW  
QyNTUxOQAAACCLcwLgAvSIHAeP+DevIRNzaG2zkNreMIZ/UHyfOUWwAAAAAJhy/wa6xf8Gu
```

SSH public key \* ⓘ

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIltzAuAC746agdh21cB4/4N6/VE3NobbOQ2t4zVn9OfJ/RRa8A root@nik-rhel8
```

- b. To create a new cluster, navigate to Clusters and click Add a Cluster > Create a Cluster. Provide the details for the cluster and the corresponding provider and click Create.

^ Configuration


Cluster name \* ⓘ

rh-aws




---



^ Distribution

Select the type of Kubernetes distribution to use for your cluster.

 Red Hat OpenShift

Select an infrastructure provider to host your Red Hat OpenShift cluster.

 Amazon Web Services   Google Cloud  Microsoft Azure

 VMware vSphere  Bare Metal

Release image \* ⓘ

quay.io/openshift-release-dev/ocp-release:4.7.12-x86\_64

Provider connection \* ⓘ

nik-hcl-aws

[Add a connection](#)

- c. After the cluster is created, it appears in the cluster list with the status Ready.
3. To import an existing cluster, complete the following steps:
    - a. Navigate to Clusters and click Add a Cluster > Import an Existing Cluster.
    - b. Enter the name of the cluster and click Save Import and Generate Code. A command to add the existing cluster is displayed.
    - c. Click Copy Command and run the command on the cluster to be added to the hub cluster. This initiates the installation of the necessary agents on the cluster, and, after this process is complete, the cluster appears in the cluster list with status Ready.

**Name \***

ocp-vmw1

**Additional labels**

Once you click on "Save import and generate code", the information you entered will be used to generate the code and cannot be modified anymore. If you wish to change any information, you will have to delete and re-import this cluster.

Code generated successfully Import saved

**Run a command**

**1. Copy this command**

Click the button to have the command automatically copied to your clipboard.

[Copy command](#)

**2. Run this command with kubectl configured for your targeted cluster to start the import**

Log in to the existing cluster in your terminal and run the command.

[View cluster](#) [Import another](#)

4. After you create and import multiple clusters, you can monitor and manage them from a single console.

### Application lifecycle management

To create an application and manage it across a set of clusters,

1. Navigate to Manage Applications from the sidebar and click Create Application. Provide the details of the application you would like to create and click Save.

Create an application  YAML: Off

Cancel

Save

**Name\*** ⓘ

demo-app


**Namespace\*** ⓘ

default X ▾

^ **Repository location for resources**

^ **Repository types**

Select the type of repository where resources that you want to deploy are located

 Git

**URL\*** ⓘ

https://github.com/open-cluster-management/acm-hive-openshift-releases.git X ▾

**Branch** ⓘ

main X ▾

**Path** ⓘ

clusterImageSets/fast/4.7 X ▾

- After the application components are installed, the application appears in the list.

## Applications

Refresh every 15s ▾


Last update: 7:36:23 PM

Overview

Advanced configuration

Create application

Q Search

Name	Namespace	Clusters	Resource	Time window	Created
demo-app	default	Local	Git 		8 days ago <span>⋮</span>

1 - 1 of 1 ▾ << < 1 of 1 > >>

- The application can now be monitored and managed from the console.



## Governance and risk

This feature allows you to define the compliance policies for different clusters and make sure that the clusters adhere to it. You can configure the policies to either inform or remediate any deviations or violations of the rules.

1. Navigate to Governance and Risk from the sidebar.
2. To create compliance policies, click Create Policy, enter the details of the policy standards, and select the clusters that should adhere to this policy. If you want to automatically remediate the violations of this policy, select the checkbox Enforce if Supported and click Create.

# Create policy ⓘ YAML: Off

Name \*

policy-complianceoperator

Namespace \* ⓘ

default

Specifications \* ⓘ

1 x ComplianceOperator

Cluster selector ⓘ

1 x local-cluster: "true"

Standards ⓘ

1 x NIST-CSF

Categories ⓘ

1 x PR.IP Information Protection Processes and Procedures

Controls ⓘ

1 x PR.IP-1 Baseline Configuration

Enforce if supported ⓘ

Disable policy ⓘ

3. After all the required policies are configured, any policy or cluster violations can be monitored and remediated from Advanced Cluster Management.

Summary 1

Standards

NIST-CSF



No violations found

Based on the industry standards, there are no cluster or policy violations.

Policies

Cluster violations

Find policies

Policy name	Namespace	Remediation	Cluster violations	Standards	Categories	Controls	Created
policy-complianceoperator	default	inform	0/1	NIST-CSF	PR.IP Information Protection Processes and Procedures	PR.IP-1 Baseline Configuration	32 minutes ago

1 - 1 of 1

<< <

1

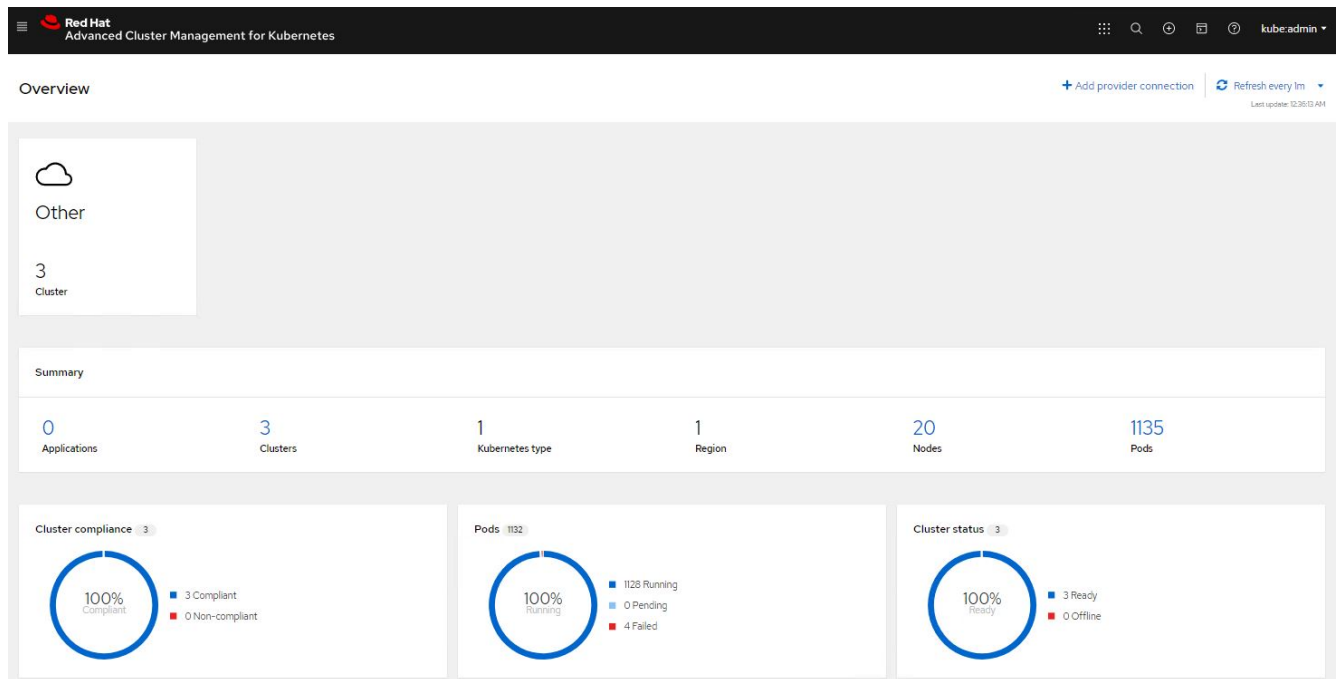
of 1

> >>

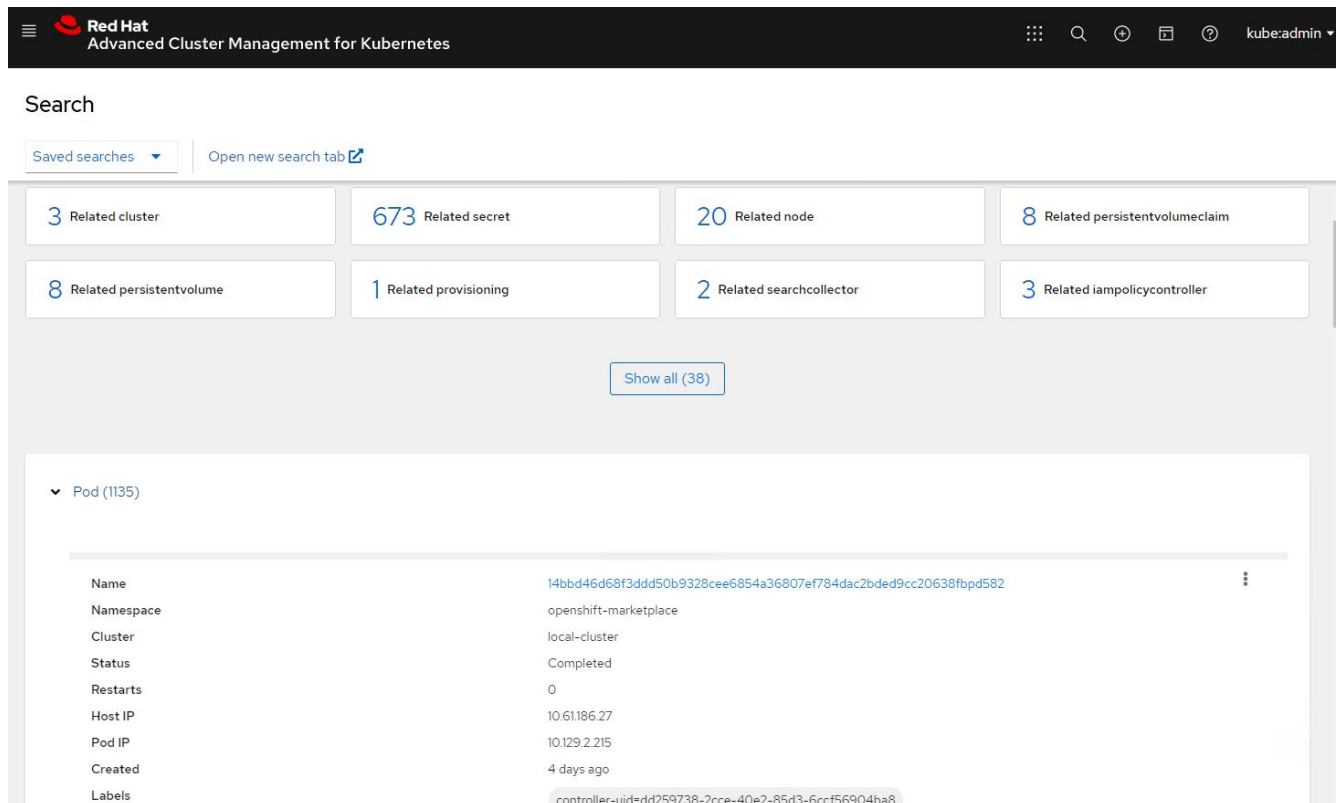
### Observability

Advanced Cluster Management for Kubernetes provides a way to monitor the nodes, pods, and applications, and workloads across all the clusters.

1. Navigate to Observe Environments > Overview.



2. All pods and workloads across all clusters are monitored and sorted based on a variety of filters. Click Pods to view the corresponding data.



3. All nodes across the clusters are monitored and analyzed based on a variety of data points. Click Nodes to get more insight into the corresponding details.

## Search

Saved searches [Open new search tab](#)

3 Related cluster | 1k Related pod | 12 Related service

[Show all \(3\)](#)

▼ Node (20)

Name ↑	Cluster ↓	Role ↓	Architecture ↓	OS image ↓	CPU ↓	Created ↓	Labels ↓
ocp-master-1.ocp-bare-metal.cie.netapp.com	ocp-bare-metal	master; worker	amd64	Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Ootpa)	48	a month ago	beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64 5 more
ocp-master-2.ocp-bare-metal.cie.netapp.com	ocp-bare-metal	master; worker	amd64	Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Ootpa)	48	a month ago	beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64 5 more
ocp-master-3.ocp-bare-metal.cie.netapp.com	ocp-bare-metal	master; worker	amd64	Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Ootpa)	48	a month ago	beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64 5 more

4. All clusters are monitored and organized based on different cluster resources and parameters. Click Clusters to view cluster details.

## Search

Saved searches [Open new search tab](#)

3k Related secret | 787 Related pod | 15 Related persistentvolumeclaim | 17 Related node | 1 Related application

15 Related persistentvolume | 1 Related searchcollector | 8 Related clusterclaim | 3 Related resourcequota | 5 Related identity

[Show all \(159\)](#)

▼ Cluster (2)

Name ↑	Available ↓	Hub accepted ↓	Joined ↓	Nodes ↓	Kubernetes version ↓	CPU ↓	Memory ↓	Console URL ↓	Labels ↓
local-cluster	True	True	True	8	v1.20.0+c8905da	84	418501Mi	<a href="#">Launch</a>	cloud=VSphere clusterID=148632d9-69d5-4ae4-98ee-8df886463c3 installer.name=multiclusterhub 4 more
ocp-vmw	True	True	True	9	v1.20.0+df9c838	28	111981Mi	<a href="#">Launch</a>	cloud=VSphere clusterID=9d76ac4e-4aae-4d45-92e8-11b6b54282fe name=ocp-vmw 1 more

### Create resources on multiple clusters

Advanced Cluster Management for Kubernetes allows users to create resources on one or more managed clusters simultaneously from the console. As an example, if you have OpenShift clusters at different sites backed with different NetApp ONTAP clusters and want to provision PVC's at both sites, you can click the (+) sign on the top bar. Then select the clusters on which you want to create the PVC, paste the resource YAML, and click Create.

# Create resource

Cancel

Create

Clusters | Select the clusters where the resource(s) will be deployed.

2 x local-cluster, ocp-vmw

Resource configuration | Enter the configuration manifest for the resource(s).

YAML

```
1 kind: PersistentVolumeClaim
2 apiVersion: v1
3 metadata:
4   name: demo-pvc
5 spec:
6   accessModes:
7     - ReadWriteOnce
8   resources:
9     requests:
10    storage: 1Gi
11   storageClassName: ocp-trident
```

## Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

## Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.