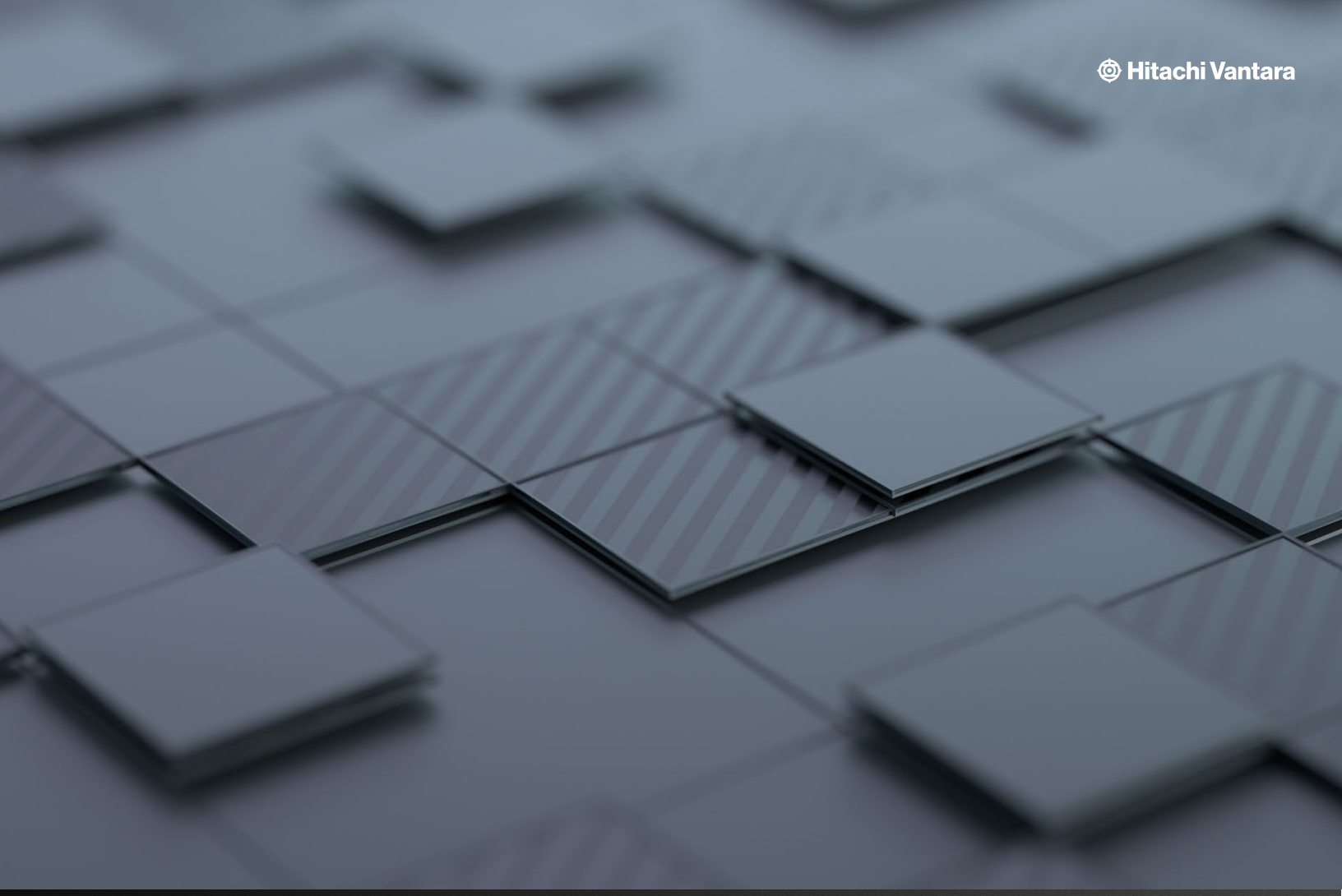


Building Mobile and Resilient Containerized Applications in a Hybrid Multi-Cloud Environment

Using Hitachi Cloud Connect for Equinix

Hitachi Vantara
January 2024

Hitachi Vantara

Table of Contents

Notices and Disclaimer

About This Guide

This reference architecture documents how to set up backup and restore operations between near-cloud Red Hat® OpenShift cluster and AWS and Azure clusters using Kasten K10 Multi-Cluster Manager and Hitachi Storage Plug-in for Containers (HSPC). Additionally, the document includes test procedures to validate the resiliency of the solution, which you can leverage for your own proof-of-concept before deploying the solution.

Intended Audience

This document is intended for Hitachi Vantara staff and IT professionals of Hitachi Vantara customers and partners who are responsible for planning and deploying such solutions.

Document Revisions

Revision Number	Date	Author	Details
v1.0	January 2024	Hitachi Vantara LLC	Initial Release

References

- Azure Red Hat OpenShift v4.11
- Hitachi Storage Administration
- Hitachi Storage Plug-in for Containers Quick Reference Guide v3.12.0
- Red Hat OpenShift Container Platform installation on AWS v4.12
- Red Hat OpenShift Container Platform installation on vSphere v4.12
- Veeam Kasten K10 Guide

Comments

Send any comments on this document to GPSE-Docs-Feedback@hitachivantara.com. Include the document title, including the revision level, and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara Corporation.

Thank you.

Executive Summary

This reference architecture documents the process of cloud-based migration of a containerized application with the Kubernetes volume snapshot function using Hitachi Storage Plug-in for Containers (HSPC) and Kasten K10 Multi-Cluster Manager by Veeam when a Hitachi Virtual Storage Platform 5200 (VSP 5200) storage system is used as the storage backend. HSPC leverages Thin Image (TI) point-in-time snapshots that are instantaneous and space efficient.

Using MySQL stateful application as an example, this document describes how to use HSPC for backup and restore, disaster recovery, and data mobility. In addition, it includes some real-world use cases. The environment used for this validation includes two Red Hat OpenShift clusters, one at the near-cloud VMware environment, one in Amazon Web Services (AWS), and one Azure-managed Red Hat OpenShift Cluster.

For all clusters, storage is provided from a VSP 5200 storage system located at the near-cloud data center. Keeping the application data in a centralized location has a number of benefits including costs, performance, and security. The near-cloud data center is a colocation operated by Equinix. This solution bridges the cloud divide and ensures availability of data across all clusters.

The Equinix colocation was selected because it offered high-speed and low latency connections to the major hyperscalers, such as AWS and Azure. Hitachi Vantara collaborated with Equinix to offer a near-cloud hybrid offering called **Hitachi Cloud Connect for Equinix**.

This offering allows clients to locate Hitachi products such as the VSP storage systems at Equinix International Business Exchange™ (IBX) data centers worldwide. In addition, there is an option for clients to procure this solution through one agreement and invoice, greatly simplifying and accelerating their time to market. By using Equinix IBX data centers and Equinix Fabric™ to interconnect sources of data to applications, organizations can locate their data residing on VSP storage systems next to clouds to leverage hybrid- or multi-cloud capabilities while still maintaining physical control of the data.

If you want to discuss hosting these types of solutions at Equinix, contact your Hitachi Vantara sales team. For more information, visit the Hitachi Cloud Connect for Equinix webpage at: https://hitachivantara.com/en-us/products/storage/flash-storage/cloud-connect-for-equinix.html.

Introduction

Red Hat OpenShift is a hybrid-cloud application platform that leverages the power of Kubernetes and combines reliable and proven services to make the process of developing, modernizing, deploying, running, and managing applications more streamlined. OpenShift ensures a uniform user experience whether applications are deployed on public-cloud, on-premises, hybrid-cloud, or edge architecture. Azure Red Hat OpenShift (ARO) delivers on-demand, fully managed OpenShift clusters with high availability, co-managed and operated in partnership with Microsoft and Red Hat.

The installation program of OpenShift Container Platform offers flexibility to deploy on a wide range of platforms. You can deploy OpenShift Container Platform on bare metal, AWS, Azure, GCP, VMware vSphere, and so on.

You can install OpenShift Container Platform using either installer-provisioned (IPI) or user-provisioned infrastructure (UPI) methods. In this reference architecture, Red Hat OpenShift clusters in near-cloud VMware and on AWS were deployed using the IPI method.

Hitachi Storage Plug-in for Containers is a software solution comprising of libraries, settings, and commands that enable you to create a container for running stateful applications. The software enables stateful applications to persist and maintain data after the life cycle of the container has ended. HSPC provides persistent volumes (PV) backed by Hitachi storage systems.

Kasten K10 is an enterprise-grade robust data management platform by Veeam that helps organizations to back up and restore container-based applications on Kubernetes/OpenShift. The capabilities include automating and orchestrating data backup, recovery, disaster recovery, and application mobility across multiple Kubernetes clusters and cloud environments. Kasten K10 offers support for a variety of Kubernetes distributions, as well as public and private cloud providers and storage solutions.

The environment used for this validation includes a Red Hat OpenShift cluster at the near-cloud data center, a Red Hat OpenShift cluster in AWS, and an ARO cluster in Azure. All clusters share the same VSP 5200 storage system located in the near-cloud data center for persistent volume requirement for stateful applications. Keeping the data at the near-cloud location ensures data availability to any cloud vendor at close proximity and avoids cloud locking. The near-cloud data center is a colocation operated by Equinix.

To summarize, our hybrid cloud environment consists of the following three domains. The relationship across the domains is shown in *Figure 1*.

- A near-cloud Equinix colocation data center (named SV5), located in San Jose, California.
- A cloud hosted by AWS in Northern California.
- A cloud hosted by Azure in California.

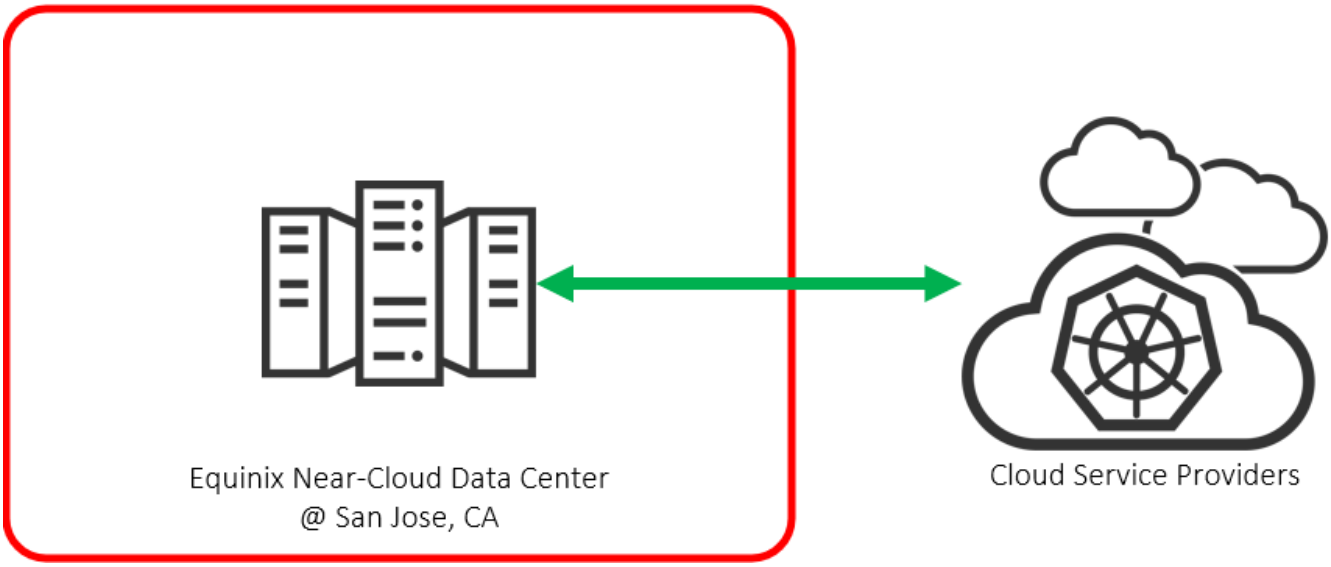


Figure 1: *Hybrid Cloud Environment*

Note: The information shared here is specific to our requirements. It can be used as a guideline or a starting point; however, you can conduct a proof-of-concept in a non-production, isolated test environment matching your production environment before implementing this solution.

Solution Overview

HSPC integrates the OpenShift Container Platform with the Hitachi storage system by using the Container Storage Interface (CSI). Integrating backup software Kasten K10 with HSPC enables protection from data loss and on demand application mobility in the OpenShift Container Platform by using the Hitachi storage system functions (such as Thin Image snapshots and ShadowImage clones).

In addition, implementing HSPC enables the availability of high-performance and high-reliability persistent volumes.

Benefits

The following lists the benefits of the business continuity solution using Red Hat OpenShift Cluster, Hitachi storage system, and Kasten K10 Multi-Cluster:

- Allows business to resume operations quickly when a disaster brings down a cluster environment.
- On-demand application mobility: Provides the flexibility to quickly snap data copies in multiple environments for on-demand analytics, data mining, disaster recovery testing, development testing, and similar use cases.
- The backup and restore operations of Kubernetes clusters in a hybrid cloud environment can be centralized with a single pane of glass UI provided by Kasten K10 Multi-Cluster manager.
- Recover from ransomware attacks: Granular, schedule-based snapshots with immutability (using the Data Retention Utility) enables the administrator to recover from a point-in-time snapshot before the attack.
- A substantial reduction in cloud egress costs can be achieved by sharing the same near-cloud storage between AWS and Azure cluster.
- Closes the gap between cloud environments and guarantees data availability across different clusters.

Key Components

The following lists the major components of the solution. For specifications, see the Hardware and Software section.

- Red Hat OpenShift Container Platform: This solution involved two Red Hat OpenShift Clusters. The first cluster consisted of three Control Plane nodes and two Worker nodes that were configured in the VMware environment at the near-cloud Equinix data center. The second cluster consisted of three Control Plane nodes and two Worker nodes deployed in AWS. Some of the key components of Red Hat OpenShift Container Platform are:
  - OpenShift Control Plane node: Runs services required for controlling the OpenShift Container Platform cluster and manages node workloads.
  - OpenShift Worker node: Worker nodes are part of the Kubernetes clusters which are responsible for running the containers and applications. Worker nodes have two main components, the Kubelet Service and the Kube-proxy Service.
  - Namespace: Provides the scope of namespace to divide cluster resources among users.
  - Persistent Volume and Persistent Volume Claim (PVC): A part of the storage of the cluster that is statically provided by the cluster administrator or dynamically provided by using the “StorageClass” object.
- Azure Red Hat OpenShift: This solution involved one ARO cluster. This cluster consisted of three Control Plane nodes and three Worker nodes. Microsoft ARO is a fully managed Red Hat OpenShift cluster, jointly operated and monitored by Microsoft and Red Hat.
- HSPC: A CSI plugin from Hitachi used to provision persistent volume from Hitachi storage systems to Red Hat OpenShift or Kubernetes cluster to preserve and maintain data after the container life cycle ends.
  - CSI-controller: Mainly incorporates the CSI controller service for storage operation. This service is deployed as “Deployment” and is run only on the control plane.
  - CSI-node: Mainly incorporates the CSI node service that manages volumes in each node. This service is deployed as “DaemonSet”. This component is required for all nodes.
- Veeam Kasten K10 Multi-Cluster Manager: Kasten K10 provides a user-friendly data management platform to perform backup or restore, disaster recovery, and mobility of containerized applications. The K10 Multi-Cluster manager provides a platform for K10 operations across multiple OpenShift clusters in a hybrid-cloud environment.
- VSP Storage Systems: A VSP 5200 storage system was used for persistent volume in Red Hat OpenShift clusters deployed in near-cloud, AWS, and Azure for stateful applications.

- Network Switches: Cisco Nexus 9000 Series switch was used to connect to AWS Direct Connect and Azure ExpressRoute. The following accessories are required for establishing a WAN between the near-cloud data center and the clouds:
  - 10/25Gbase-LR-S Optics: Long Range transceivers required to connect long distances.
  - Single-Mode Fiber Cables: Required for long-distance communications.
- Equinix Fabric: Connected equipment at the Equinix near-cloud data center to AWS cloud and Azure cloud.
- AWS Cloud: Equipment at Equinix was connected to AWS cloud using a 10 Gbps Direct Connect link. On AWS, a Virtual Private Cloud was created in the region us-west-1. Some of the key services used in AWS cloud are EC2, S3, Route53, Classic load balancer, and Network load balancer.
- Azure Cloud: Equipment at Equinix was connected to Azure cloud using a 10 Gbps ExpressRoute link. On Azure, a Virtual Network was created in the region West US. Some of the key services used in Azure cloud are ARO, virtual machines, and load balancer.

Validation

This section describes the method, test environment, hardware and software, and test scenarios used in the validation.

Validation Method

This solution consists of the following test cases.

Test case 1 involves setting up the environment, which includes three clusters - two Red Hat OpenShift clusters in the near-cloud center and AWS cloud and one Azure RedHat OpenShift cluster.

To validate test case 2, a persistent volume was allocated from the VSP 5200 storage system located in near-cloud to deploy a stateful MySQL application in the ARO cluster.

Test case 3 involves taking backup of the stateful application in the ARO cluster and restoring in the target cluster in AWS. Before performing the backup operation, fresh data was inserted into the MySQL application in the ARO cluster. A snapshot of the persistent volume was created with Kubernetes Volume Snapshot function with HSPC. In the target cluster, a PVC of the snapshot volume was created and used as a source to create a clone volume. The stateful MySQL application was restored using the clone volume in the target cluster in AWS.

To validate test case 4, fresh data was inserted into the MySQL application in ARO cluster, and after restoring the backup, the database records were verified at the AWS location to ensure data consistency. The Kasten K10 Multi-Cluster user interface was used to perform this test case. A Global Location Profile was created with AWS S3 bucket as the storage provider, followed by creating Global Policies to automate the workflows for managing data (such as snapshot and restore). The subsequent step was to add Distributions, which defines the clusters where K10 resources must be allocated. Finally, snapshot and restore operations were carried out using the Global Policies.

Test case 5 shows how business continuity can be performed if a ransomware attack corrupts the application data. To validate this test case, a stateful MySQL application was used and the Hitachi Data Retention Utility (DRU) feature was set on the snapshot volume to restrict read and write. If a ransomware attack corrupts the application data, the data can be restored from the snapshot. You can perform the recovery process in either of the Red Hat OpenShift Container Platform cluster in AWS or in ARO. The process involves creating a PVC of the DRU-enabled snapshot, creating a snap-on-snap copy of that PVC, and then restoring the stateful MySQL application using the cloned PVC in the target cluster.

High Level Diagram

Figure 2 shows the test environment used to run the validation.

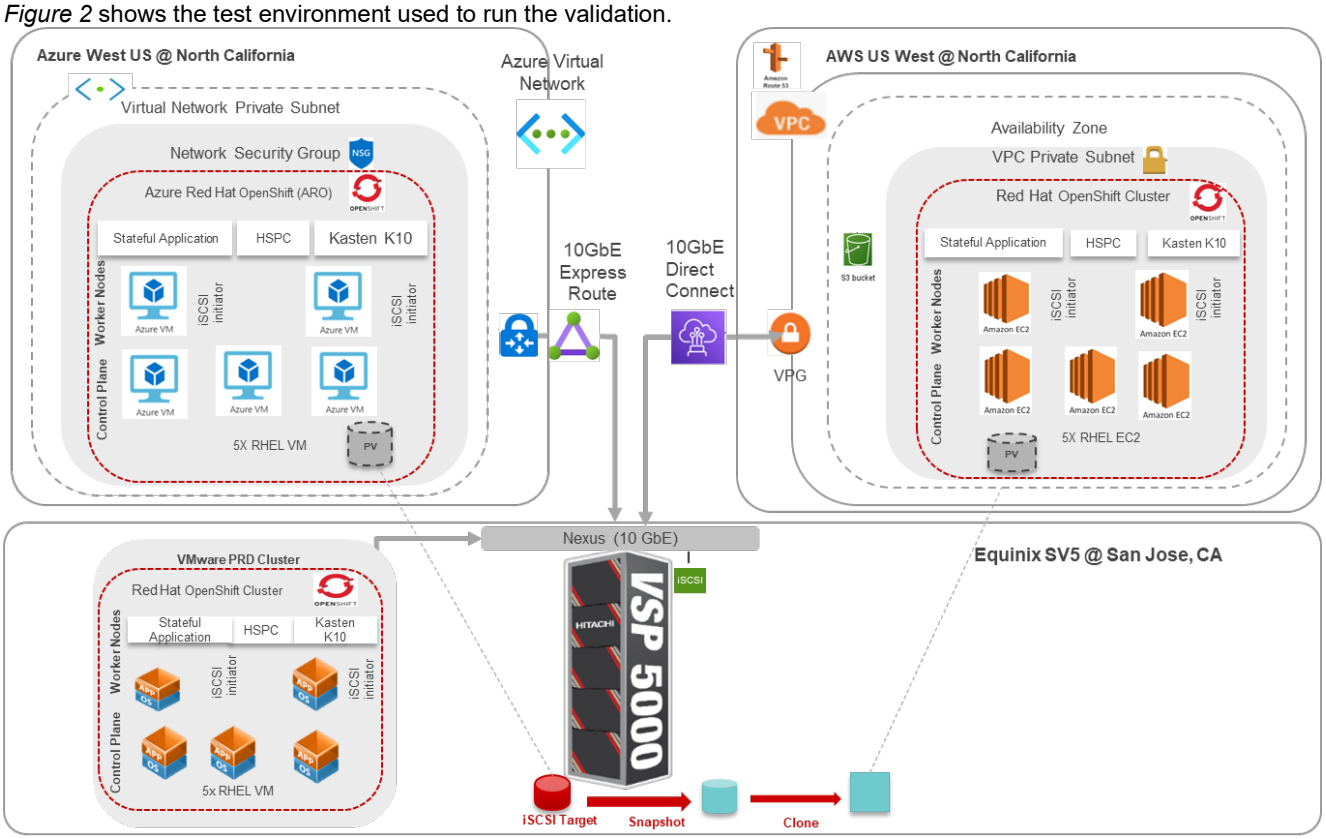


Figure 2: *Test Environment*

Hardware and Software

Table 1 provides the hardware specifications for the equipment used in this validation.

	Item	Description	Version	Function
Equinix Near-Cloud Data Center	**Hitachi VSP 5200**	1 TB cache (2) 20-core MPUs (4) RAID6 6D+2P parity groups (1) 10 GbE iSCSI port	SVOS RF 9.8.6 90-09-01-00/01	Storage system used to store application data.
	Hitachi Advanced Server DS220	(2) 18-core Intel Xeon Gold 6140 @ 2.3 GHz 128 GB cache (1) Intel Ethernet Network Adapter XXV710	BMC 4.70.06 BIOS S5BH3B22.H00	4-node VMware vSphere cluster used to deploy 5-nodes near-cloud Red Hat OpenShift cluster.
	Cisco Nexus C93180YC-FX	(48) 1/10/25-Gbps fiber ports (6) 40/100-Gbps QSFP28 ports	NXOS 9.3(4)	Network switch at the near-cloud data center servicing AWS Direct Connect.
AWS	**Amazon EC2**	(4) Intel Xeon Platinum 8000 series processor, 16 GB RAM	Instance type: t3.xlarge AMI Name: rhcos-412.86.202306132230-0-x86_64 AMI ID: ami-03260f4b6e0166045	5-nodes Red Hat OpenShift cluster in the cloud.
	Amazon S3	(1) S3 Standard bucket	N/A	Storing Kasten K10 Multi-Cluster Global Location profile.
Azure	**Azure Virtual Machine**	Master Nodes: (8) Intel Xeon CPU E5-2673 v3 @ 2.40GHz processor, 32 GB RAM Worker Nodes: (4) Intel Xeon CPU E5-2673 v3 @ 2.40GHz processor, 16 GB RAM	Master Nodes: Standard D8s v3 Worker Nodes: Standard D4s v3	6-nodes Azure Red Hat OpenShift cluster.

Table 1*: Hardware Components*

Table 2 provides the software specifications used in this validation.

Item	Version	Function
VMware vSphere	7.0 U2 (17867351)	Hypervisor operating system.
VMware vCenter Server Appliance	7.0 U3 (18700403)	Management interface for vSphere cluster.
Red Hat OpenShift	4.11.25 4.12.25	Red Hat OpenShift cluster deployed in near-cloud. Red Hat OpenShift cluster deployed in AWS.
Azure Red Hat OpenShift	4.11.26	ARO Cluster deployed in Azure.
Hitachi Storage Plug-in for Containers	3.12	HSPC plugin integrates Kubernetes or OpenShift with Hitachi storage systems using Container Storage Interface.
Kasten K10 Multi-Cluster	6.0.2	Kasten K10 Multi-Cluster is a data management platform from Veeam which provides backup operation, disaster recovery, and application mobility for OpenShift applications across multiple clusters.

Item	Version	Function
MySQL	5.7.41	A stateful database application used to validate data consistency.

Table 2*: Software Components*

Test Scenarios

Table 3 lists the test scenarios performed in the validation.

#	Description	Success Criteria
1	Prepare the environment: 1. Deploy two Red Hat OpenShift clusters. One in a VMware environment in near-cloud and another in AWS. 2. Deploy one Azure Red Hat OpenShift cluster in Azure. 3. Define storage, network, and iSCSI connections. 4. Use Dynamic Provisioning pool to provision persistent volume for stateful application in Azure and AWS. 5. Deploy HSPC in both clusters. 6. Deploy Kasten K10 and K10 Multi-Cluster in all clusters. 7. Discover ARO and Red Hat OpenShift Cluster in AWS from Kasten K10 Multi-Cluster Manager deployed in near-cloud.	Environment is set up as per specifications.
2	Deploy a stateful application in the Red Hat OpenShift Container Platform clusters. This test case is performed in Azure. The persistent volume is provisioned in the ARO cluster from the Hitachi VSP 5200 storage system located in near-cloud. 1. Define the storage class for the VSP 5200 storage system with the required settings. 2. Deploy MySQL database as a stateful application on the ARO cluster with persistent volume claim. 3. Create a new table and ingest new records.	Persistent volume from the VSP storage system can be provisioned to the ARO cluster. Stateful application can be deployed successfully.
3	Migrate a stateful application across OpenShift clusters using HSPC (this test case is performed manually instead of Kasten K10): 1. Ingest data into MySQL application in the ARO cluster. 2. Create a Kubernetes volume snapshot. 3. Create PV and PVC of the snapshot volume. 4. Create a clone PVC using the PVC created in step 3 as the source PVC. 5. Use the clone as a volume source to deploy MySQL stateful application in the Red Hat OpenShift Container Platform cluster on AWS. 6. Verify whether the ingested data is visible to the target MySQL environment.	Verify that the snapshot created in the ARO cluster can be manually restored in the Red Hat OpenShift Container Platform cluster in AWS.
4	Migrate a stateful application across OpenShift clusters using Kasten K10 Multi-Cluster: 1. Ingest data into MySQL application in Azure. 2. Create an S3 bucket in AWS. 3. Create a global location profile using this bucket. 4. Create a global snapshot policy. 5. Create a global distribution for snapshot policy and add the cluster. 6. Run the snapshot policy for the MySQL application to take the backup. 7. Create a global import policy for restore. 8. Create a global distribution for import policy and add the cluster. 9. Run the policy to restore the application in the target cluster. 10. Verify whether the MySQL application is being restored and the ingested data is visible to the target MySQL environment.	Verify that the backup taken in the ARO cluster can be restored in the Red Hat OpenShift Container Platform cluster in AWS using Kasten K10 Multi-Cluster.
5	Recover from a ransomware attack: This test case is performed manually instead of Kasten K10. The Data Retention Utility feature is set on the snapshot volume to protect the backup from any write operations and define the data retention term for the protected volumes. 1. Ingest data into MySQL application in Azure. 2. Create a Kubernetes volume snapshot.	Revert to clean stateful MySQL application from snapshot data with DRU.

#	Description	Success Criteria
	3. Set DRU attribute in the snapshot volume using Command Control Interface. 4. Assume that the application is affected by ransomware in Azure and must restore the data from the snapshot taken in step 2. 5. Create a PVC using the snapshot volume created in step 2. 6. Create a Kubernetes volume snapshot (snap-on-snap) of the PVC created in step 5. This creates a cascaded snapshot volume. 7. Create PVC of the cascaded snapshot (snap-on-snap) volume. 8. Create a clone PVC using the PVC created in step 7 as the source PVC. 9. Use the clone PVC as a volume source to deploy MySQL stateful application in the Red Hat OpenShift Container Platform cluster in AWS. 10. Verify whether the ingested data is visible to the target MySQL environment.	

Table 3*: Test Scenarios*

Guidelines and Recommendations

This section describes the lessons learned from this validation, along with guidelines and recommendations.

- While installing a Red Hat OpenShift cluster in a private environment (for example, in an existing Amazon Virtual Private Cloud with a specific AWS Identify and Access Management user), use "*CredentialMode*" to set as "Manual" in the install-config.yaml file. The default mode is "Mint", which assumes that you have administrative privileges.
- While running the OpenShift installation, install-config.yaml file is used by the installer. You must keep a backup of this file. If the installation fails and must be re-run, copy the OpenShift installer and install-config.yaml to a new directory and then run from there. You must not re-use the same directory, or else X.509 certificate error occurs.
- Prepare a separate node outside the cluster for cluster deployment and install OpenShift CLI (oc) command to interact with OpenShift Container Platform for administration.
- While migrating an application using Kasten K10 across clusters, a location profile is mandatory. Without the location profile, import policy would not generate, and restoration is not possible to other clusters. However, to restore an application in the same cluster, a location profile is not required.
- While building a POD with persistent volume, HSPC automatically performs a series of tasks such as provisioning the volume, creating an iSCSI target (or FC host group), attaching the volume to it, discovering the volume on the target node, and then attaching the volume as a block device or creating a file system on it.
- In Kubernetes environment, a "VolumeSnapshot" object cannot be attached to a POD because it is not a persistent volume. To access the snapshot data, create a clone volume and then attach the clone volume to a POD.
- Retention time cannot be reduced while DRU setting is active on a volume.
- ARO does not allow scaling the cluster workers to zero or attempt a cluster shutdown. Deallocating or powering down any virtual machine in the cluster resource group is not supported.

Validation Results

This section shows the steps and screenshots for each test scenario.

Test 1: Prepare the Environment

This test case describes the configuration of the components used in the validation.

The test environment consists of three clusters: two multi-node Red Hat OpenShift clusters deployed using IPI method in near-cloud VMware environment and in AWS, and one Azure Red Hat cluster deployed in Azure. You must configure the following components for validation of test cases:

- Configure physical LAN and iSCSI connections for OpenShift clusters.
- Establish connections among three clusters.
- Provision DP pool to be used for persistent volume from the VSP 5200 storage system.
- Deploy two Red Hat OpenShift clusters, one in near-cloud VMware environment and another in AWS.
- Deploy one Azure Red Hat cluster in Azure.
- Install HSPC.
- Deploy Kasten K10 Multi-Cluster.

For steps to configure the following components, see https://www.hitachivantara.com/en-us/pdf/architecture-guide/business-continuity-containerized-applications-in-hybrid-cloud-environment.pdf.

- Deploy Red Hat OpenShift clusters in near-cloud VMware environment and AWS.
- Install HSPC.
- Deploy Kasten K10 and Kasten K10 Multi-Cluster.
- Access Kasten K10 dashboard.
- Discover AWS Red Hat OpenShift cluster and ARO cluster from Kasten K10 Multi-Cluster UI in near-cloud as a secondary cluster.

Deploy Azure Red Hat OpenShift Cluster

Azure Red Hat OpenShift is a fully managed Red Hat OpenShift service in Azure.

Prerequisites

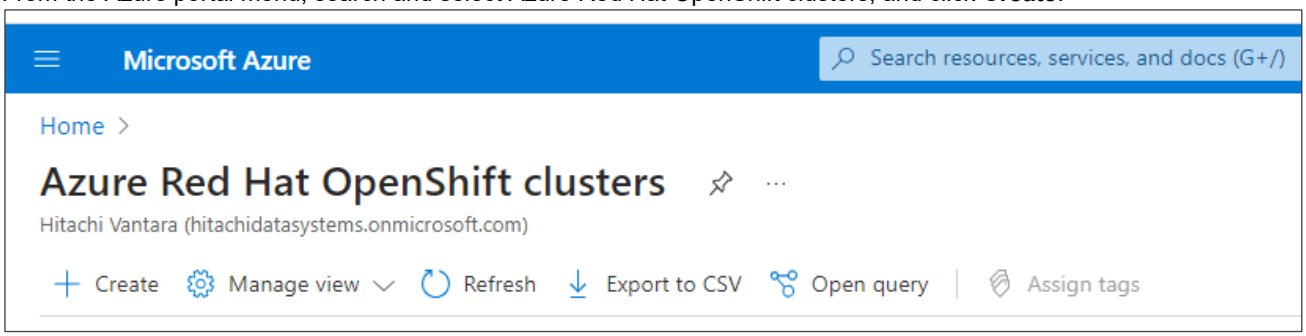
Note that the following prerequisites are outside the scope of this document, so we do not describe them in detail. For more information, see https://learn.microsoft.com/en-us/azure/openshift/quickstart-portal.

- Access to the Azure portal.
- Create a service principal.
- Create a resource group.
- Create a virtual network with two empty subnets.
- Obtain a Red Hat pull secret (optional).
- Install OpenShift CLI (oc) on the admin node to interact with OpenShift Container Platform from a command-line interface.

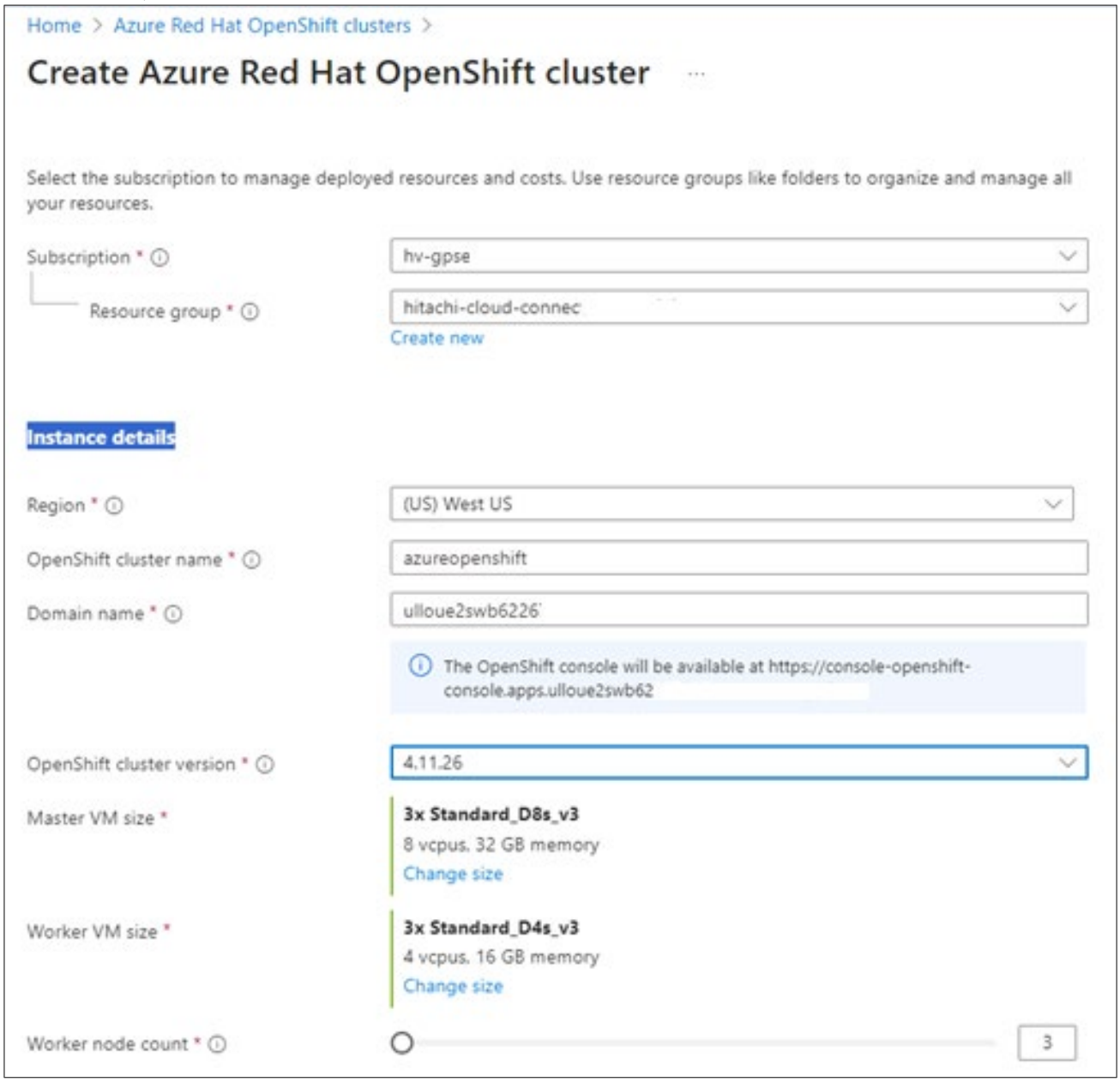
Create Azure Red Hat OpenShift Cluster

To deploy the ARO cluster from the Azure portal menu, complete the following steps:

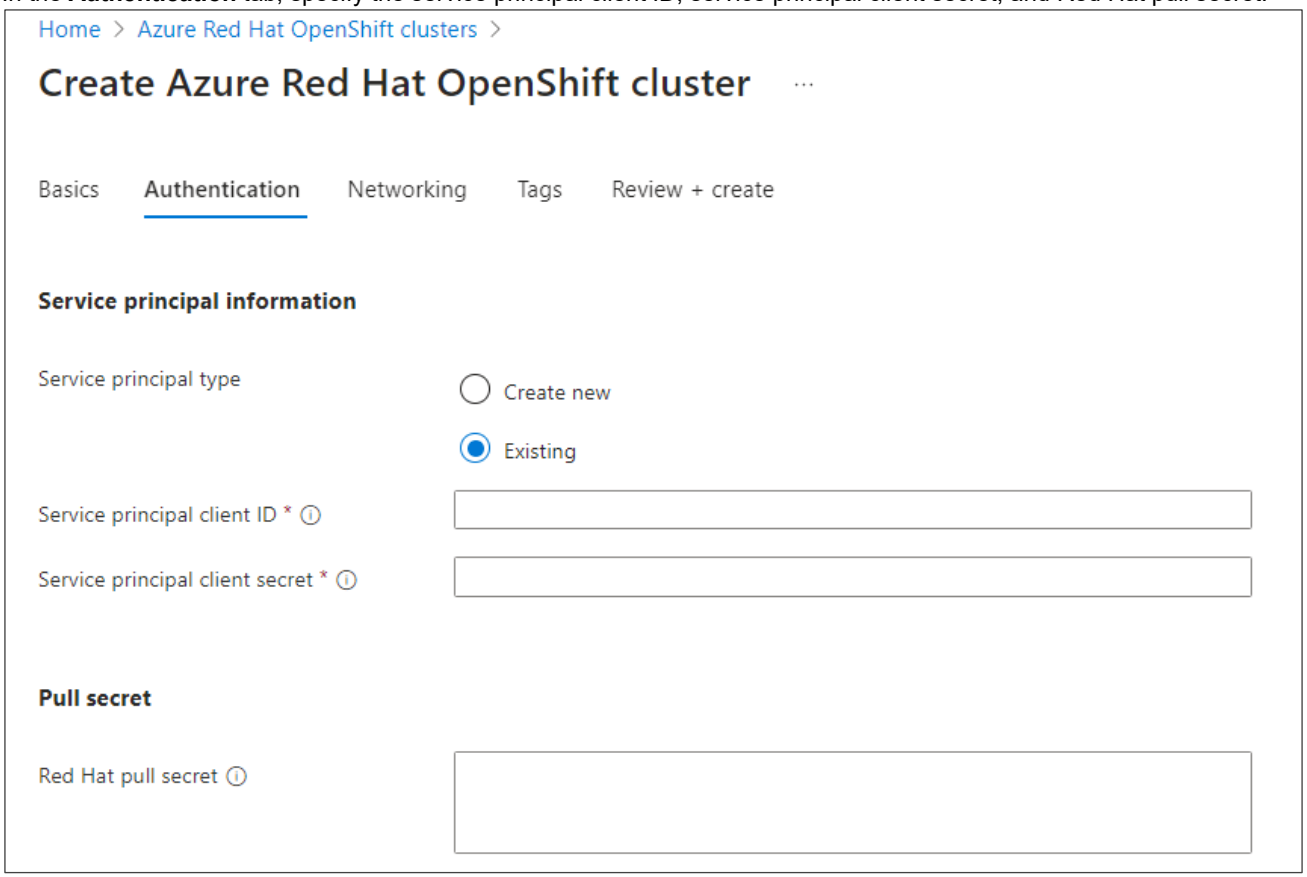
1. From the Azure portal menu, search and select Azure Red Hat OpenShift clusters, and click **Create**.



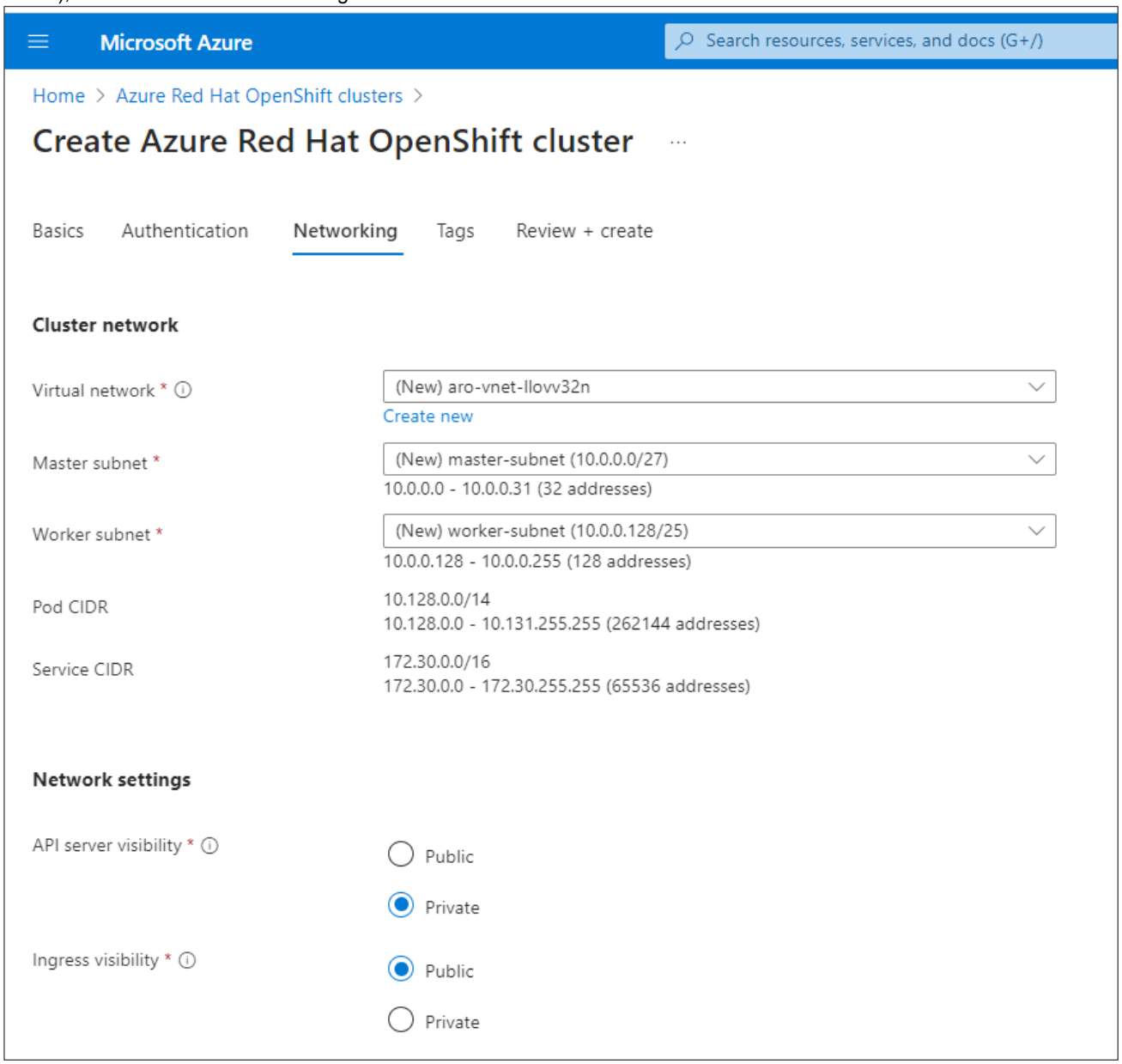
2. In the **Basics** tab, specify the project and instance details such as region, cluster name, domain name, master VM size, worker VM size, and worker node count.



3. In the **Authentication** tab, specify the service principal client ID, service principal client secret, and Red Hat pull secret.

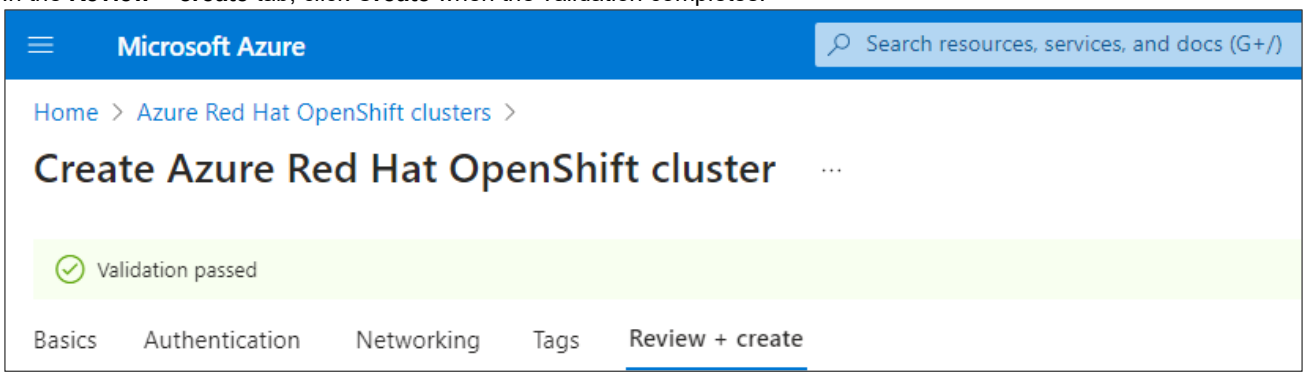


4. In the **Networking** tab, specify the virtual network name, two empty subnets (one for the control plane and one for worker nodes), and select the network settings.

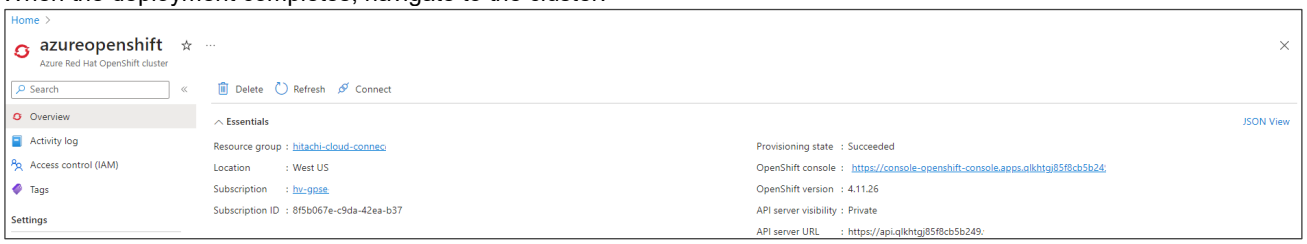


5. In the **Tags** tab, add tags to organize resources.

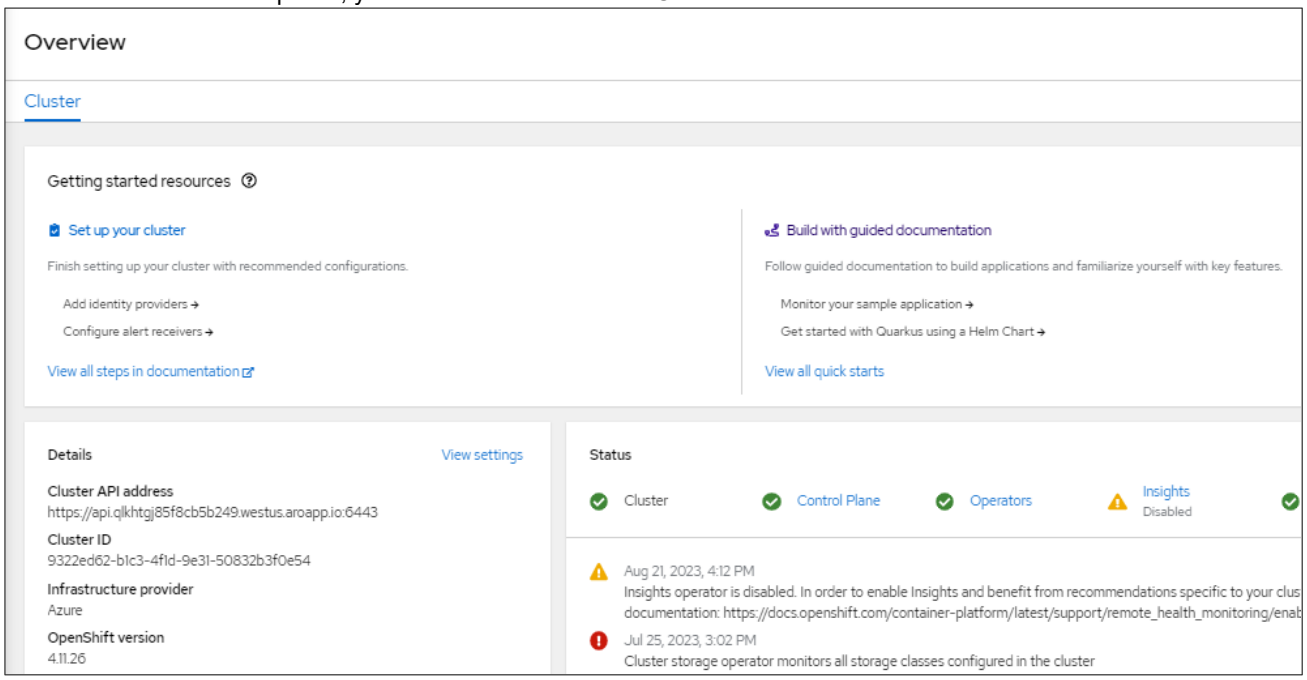
6. In the **Review + create** tab, click **Create** when the validation completes.



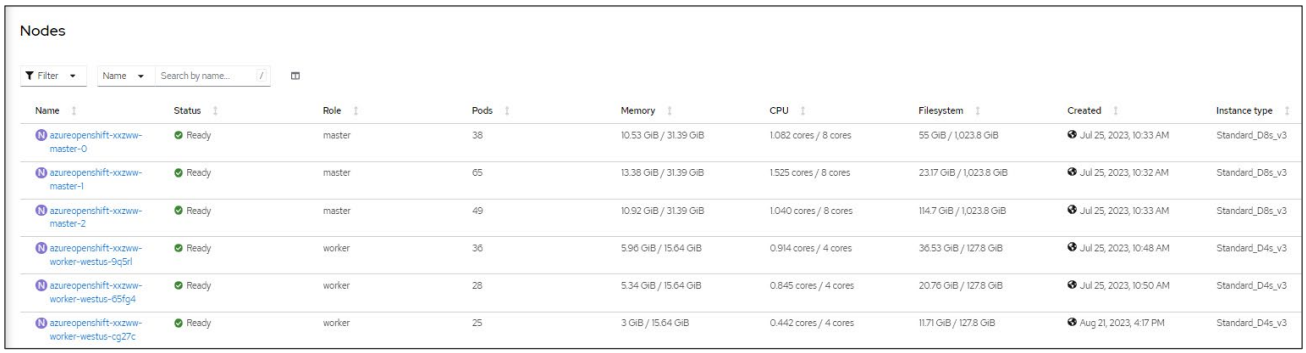
7. When the deployment completes, navigate to the cluster.



8. When the installation completes, you can access the console URL:



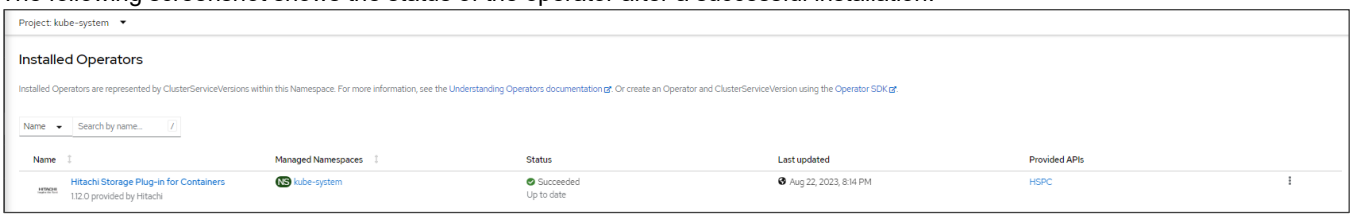
Status of the master and worker nodes of the cluster:



Install Hitachi Storage Plug-in for Containers

Installing HSPC in the Azure Red Hat OpenShift Cluster is similar to the section Install HSPC in Near-Cloud Cluster.

The following screenshot shows the status of the operator after a successful installation.



From the console, navigate to **Workloads**, click **Pods**, and ensure that the status of the operator pod is running.

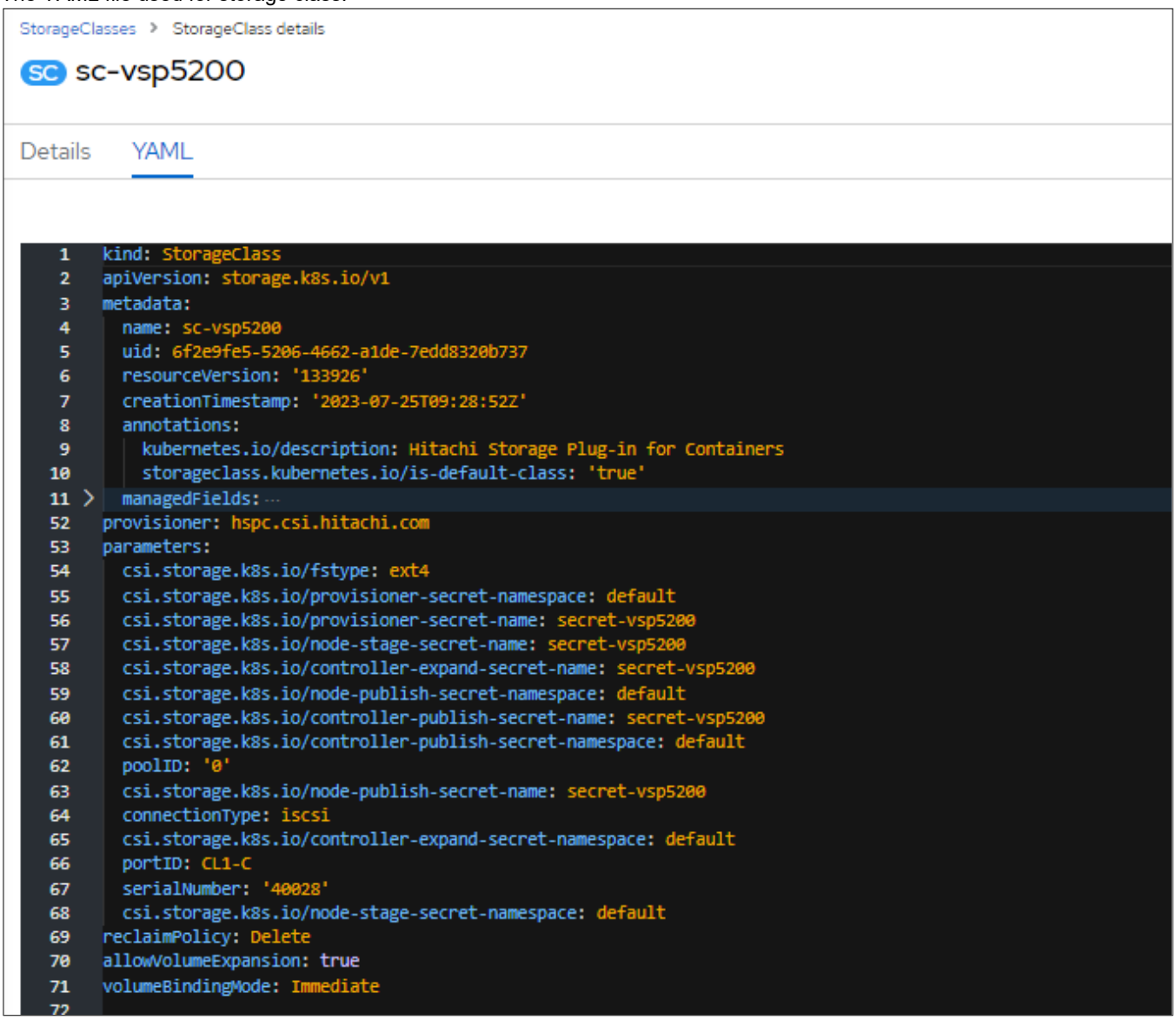
Verify that the status is Ready.

```
# oc get hspc -n kube-system
NAME    READY    AGE
hspc    true     6m02s
```

Create StorageClass and Volume SnapshotClass

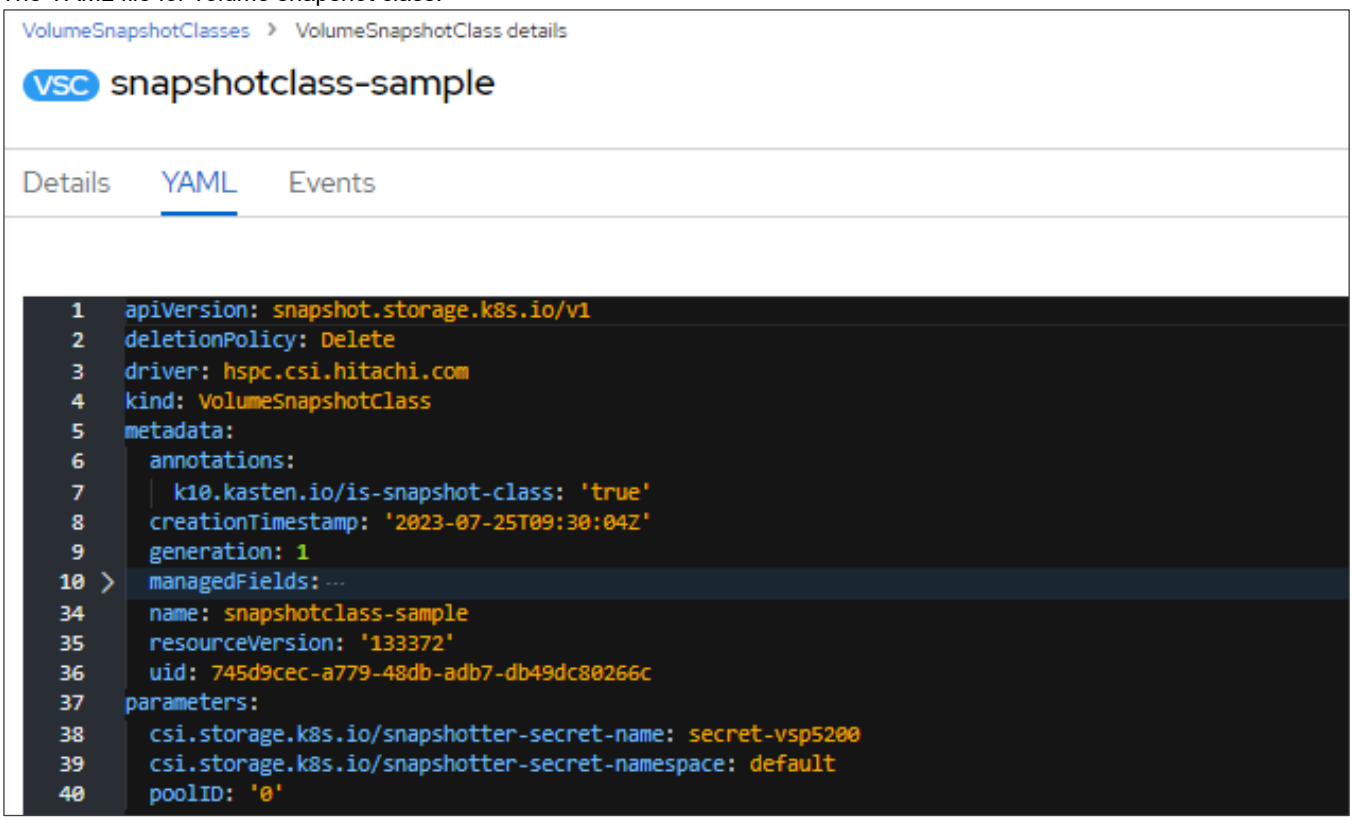
After installing HSPC, you must create storage class to provision persistent volume from the VSP 5200 storage system. Additionally, a volume snapshot class is required to take point in time snapshot. The following screenshots show the status of storage class and volume snapshot class.

The YAML file used for storage class:



Status of the StorageClasses:

The YAML file for volume snapshot class:

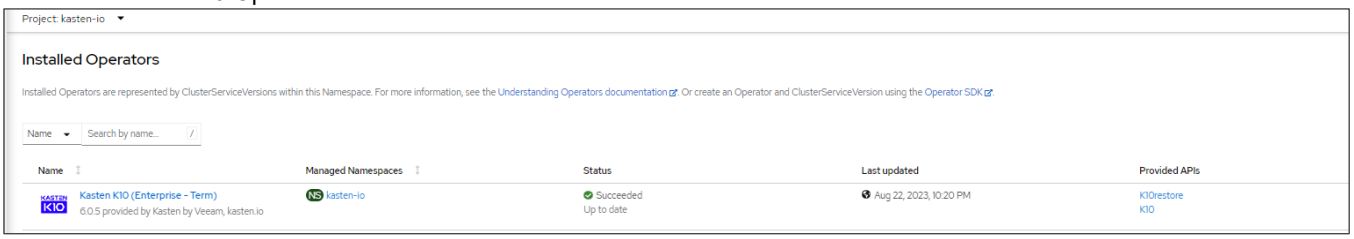


Status of the VolumeSnapshotClasses:

Install Kasten K10 in Azure Red Hat OpenShift Cluster

Deploying Kasten K10 in the ARO cluster is similar to deploying it in the near-cloud OpenShift cluster. The following screenshots show the status of Kasten K10.

Status of Kasten K10 Operator:



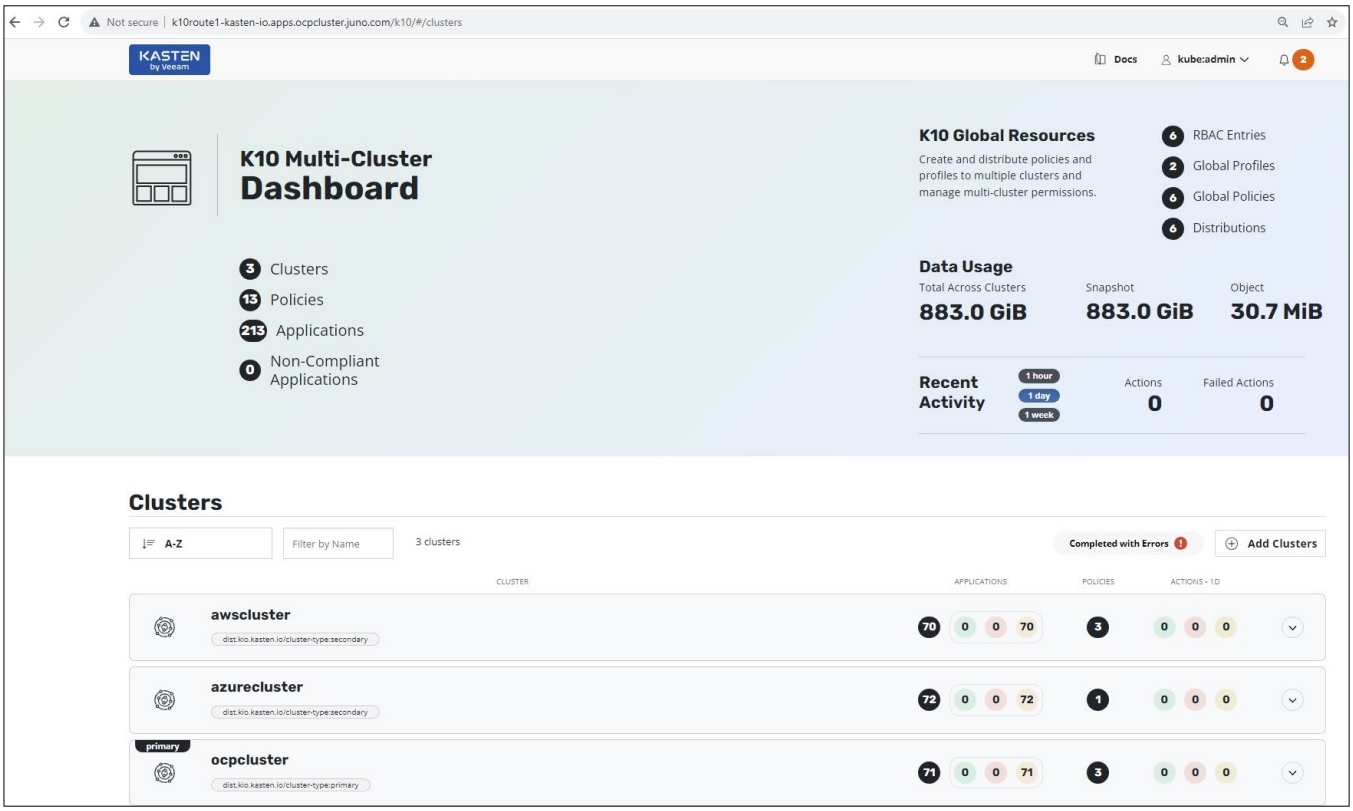
Status of Kasten K10:

Prepare Kasten K10 Multi-Cluster in Azure Red Hat OpenShift Cluster

Download and untar the Kasten K10 Multi-Cluster tool from the URL: https://github.com/kastenhq/external-tools/releases.

Discover Azure RedHat OpenShift Cluster as the secondary cluster from Kasten K10 Multi-Cluster UI in near-cloud.

Status of the discovered cluster “azurecluster” in Kasten K10 dashboard in near-cloud:

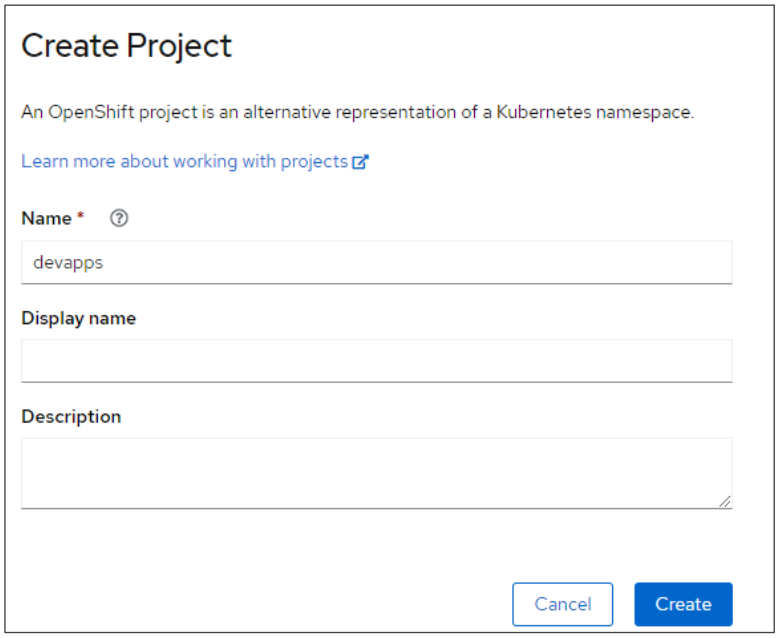


Test 2: Deploy a Stateful Application in Azure Red Hat OpenShift Cluster

This test case describes the process of deploying MySQL stateful application in Azure Red Hat OpenShift cluster using persistent volume from the near-cloud VSP 5200 storage system. HSPC enables the application to use a persistent volume from the VSP 5200 storage system.

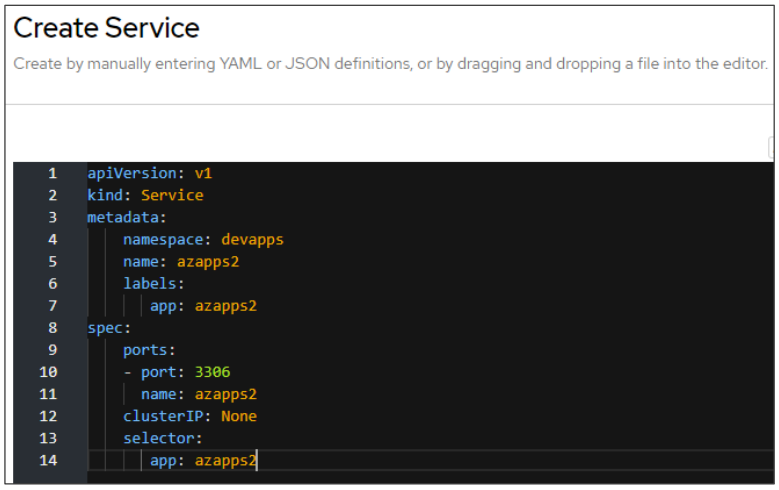
1. Deploy a stateful MySQL application.

   a. Create a project (Kubernetes namspace) for the MySQL application. From the Red Hat OpenShift console, navigate to **Home**, click **Projects**, and then click **Create Project**. In the Create Project menu, enter a project name and click **Create**.

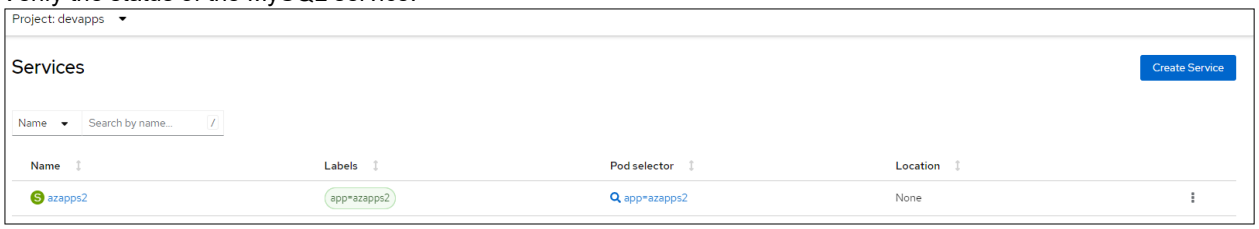


   Status of the project.

   b. Create a MySQL service. From the Red Hat OpenShift console, navigate to **Networking**, click **Services**, and then click **Create Service**. In the Create Service menu, populate the YAML file with the required information and click **Create**.



c. Verify the status of the MySQL service:



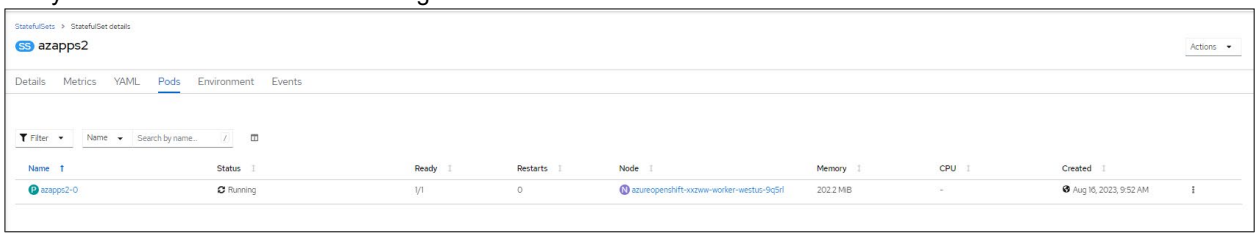
d. Create a MySQL statefulset application. From the Red Hat OpenShift console, navigate to **Workloads**, click **StatefulSet**, and then click **Create StatefulSet**. In the Create StatefulSet menu, populate the YAML file with the required information and click **Create**.

Create StatefulSet

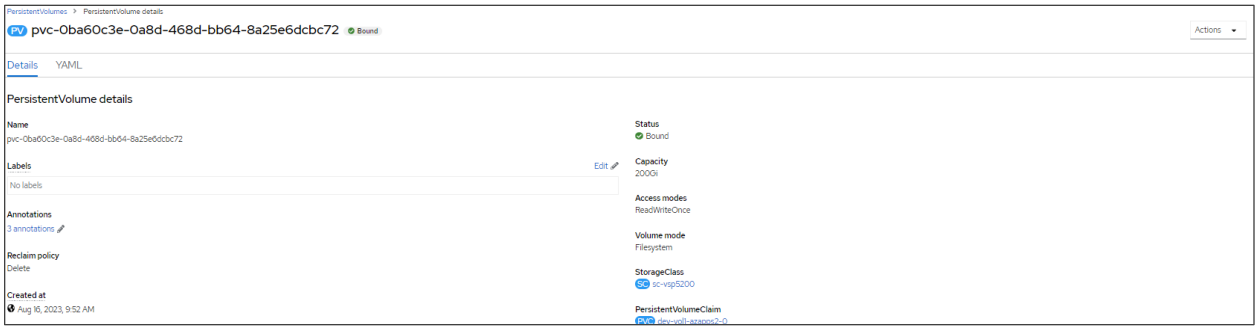
Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor.

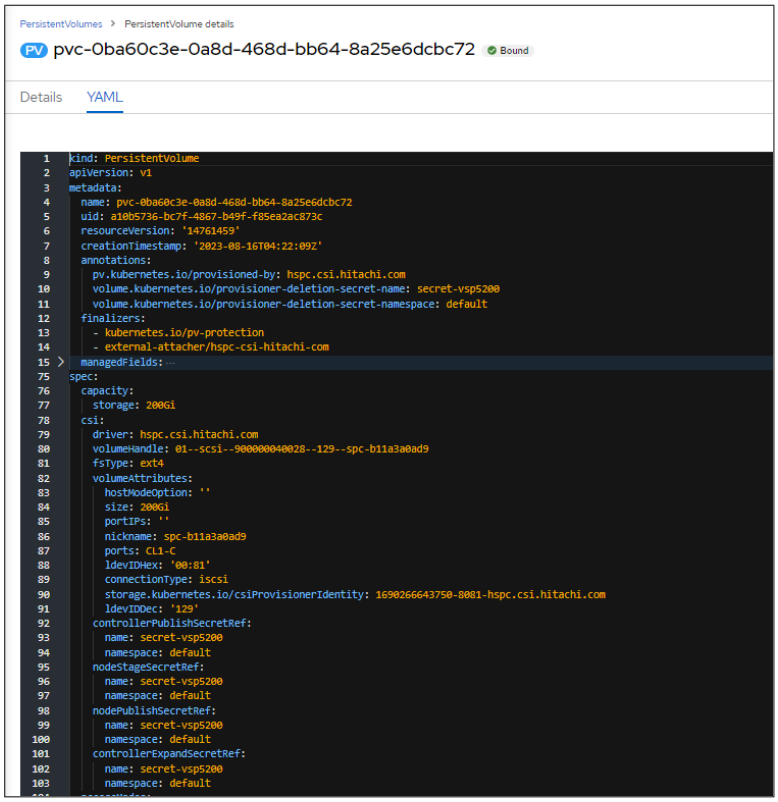
```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
    namespace: devapps
    name: azapps2
spec:
  selector:
    matchLabels:
      app: azapps2
  serviceName: "azapps2"
  podManagementPolicy: Parallel
  replicas: 1
  template:
    metadata:
      labels:
        app: azapps2
    spec:
      terminationGracePeriodSeconds: 30
      containers:
      - name: azapps2
        image: mysql:5.7
        args:
          - "--ignore-db-dir=lost+found"
        env:
        - name: MYSQL_ROOT_PASSWORD
          value: pass123
        - name: MYSQL_DATABASE
          value: devmysqldb1
        - name: MYSQL_USER
          value: admin
        - name: MYSQL_PASSWORD
          value: secret
        ports:
        - containerPort: 3306
          name: mysql
        volumeMounts:
        - name: dev-vol1
          mountPath: /var/lib/mysql
  volumeClaimTemplates:
  - metadata:
      name: dev-vol1
    spec:
      storageClassName: sc-vsp5200
      accessModes: [ "ReadWriteOnce" ]
      resources:
        requests:
          storage: 200Gi
```

e. Verify whether the StatefulSet is running.



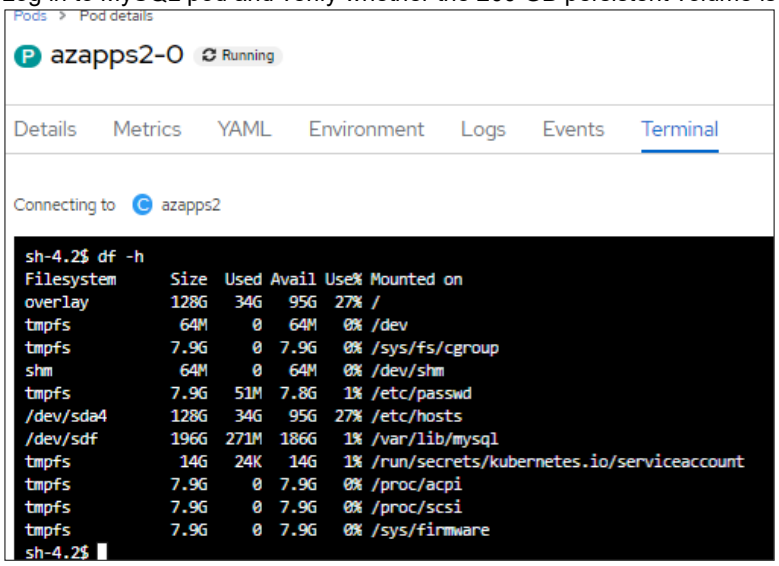
f. Verify whether the PVC is created from the VSP 5200 storage system. Using storage class dynamically provisions a persistent volume in the VSP 5200 storage system. The following screenshots show the status of the PVC and PV created.





2. Access the MySQL application.

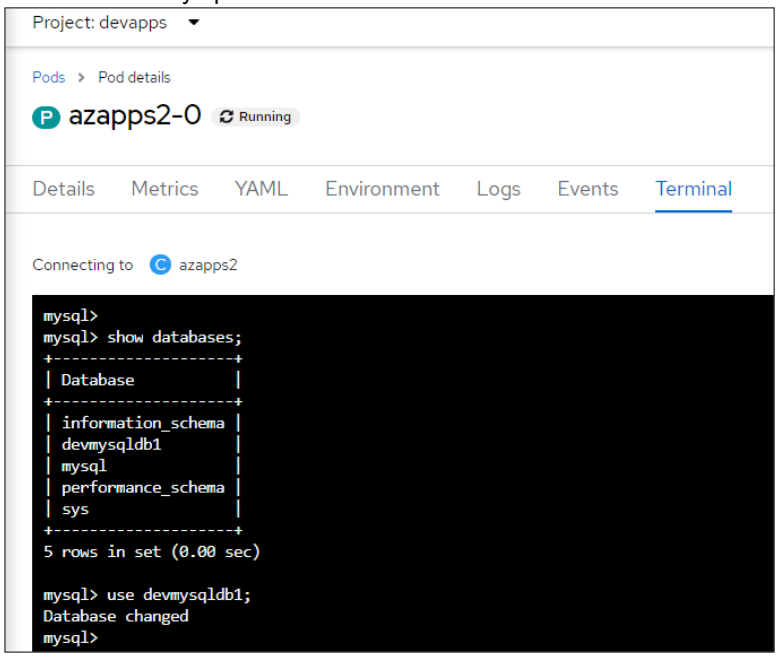
a. Log in to MySQL pod and verify whether the 200 GB persistent volume is created and mounted in /var/lib/mysql.



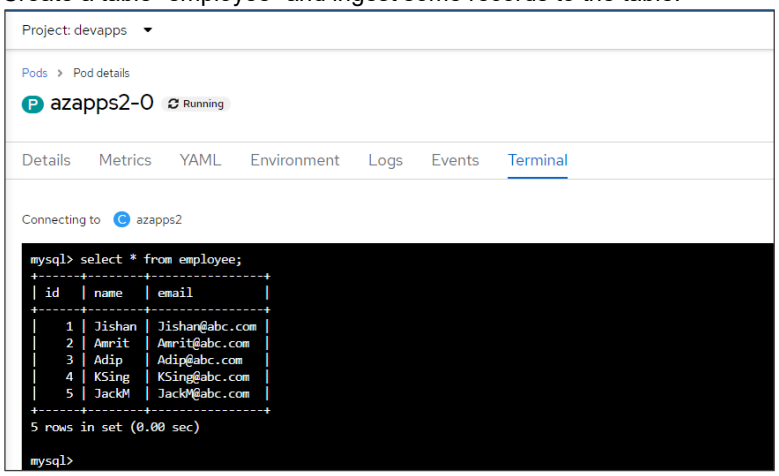
b. Log in to MySQL database using `mysql -u root -p`.

c. Verify whether the “devmysqldb1” database is created.

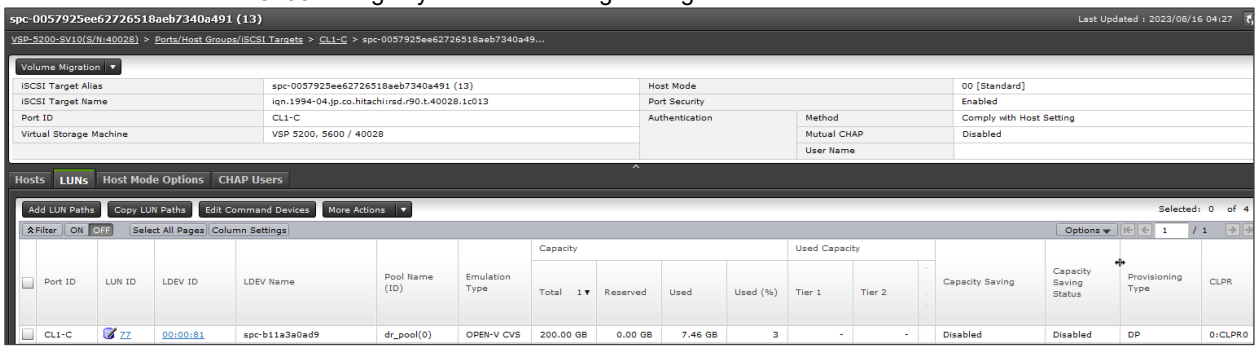
d. Select the “devmysqldb1” database.



e. Create a table “employee” and ingest some records to the table.



f. HSPC automatically creates an iSCSI target on port CL1-C of the storage system. Verify whether the 200 GB volume was created in the VSP 5200 storage system from Storage Navigator.



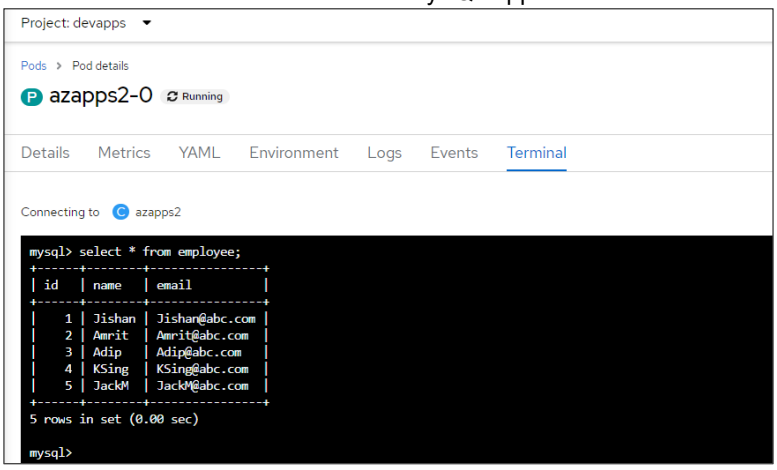
Test 3: Manually Migrate Stateful Applications Across OpenShift Clusters

This test case describes the process of migrating a stateful application from OpenShift cluster in Azure to AWS using Kubernetes commands and HSPC. The VSP 5200 storage system provides the persistent volume required for stateful MySQL applications in both clusters.

Snapshot Operation

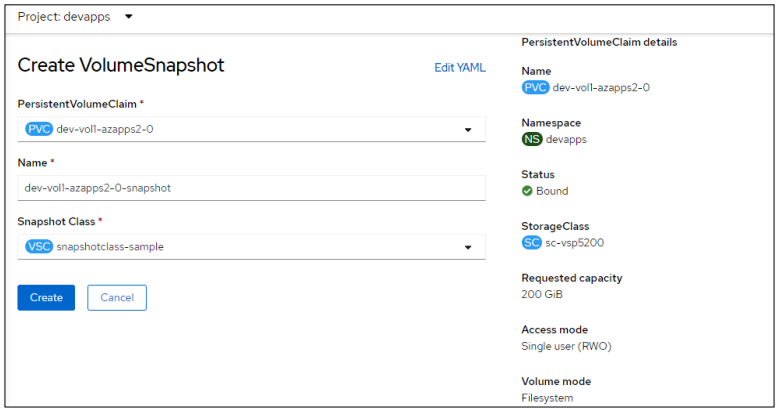
To perform snapshot operation of an application in Azure Red Hat OpenShift Cluster, complete the following steps:

To create a MySQL application with a persistent volume of 200 GB from the VSP 5200 storage system and ingest data to the database, see the section Test 2: Deploy a Stateful Application in Azure Red Hat OpenShift Cluster. The following screenshot shows that the data is available in the MySQL application.

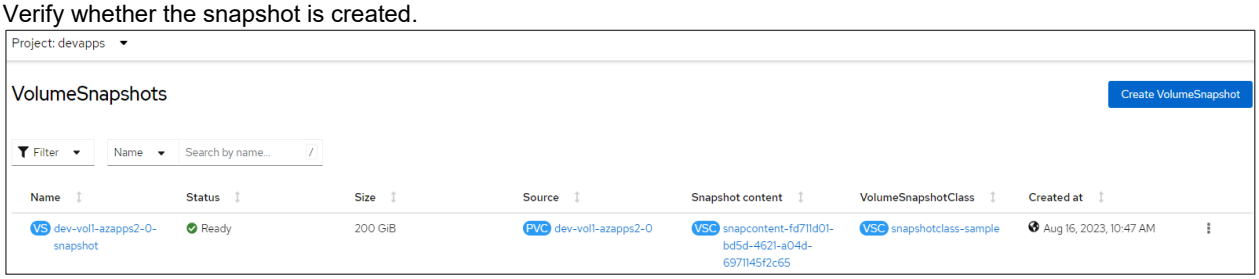


1. Create a snapshot.

   a. Create a snapshot of the persistent volume created for the application. From the Red Hat OpenShift console, navigate to **Storage**, click **VolumeSnapshots**, and then click **Create VolumeSnapshot**. In the Create VolumeSnapshot menu, enter the required information such as PVC, snapshot name, snapshot class, and click **Create**.



   b. Verify whether the snapshot is created.

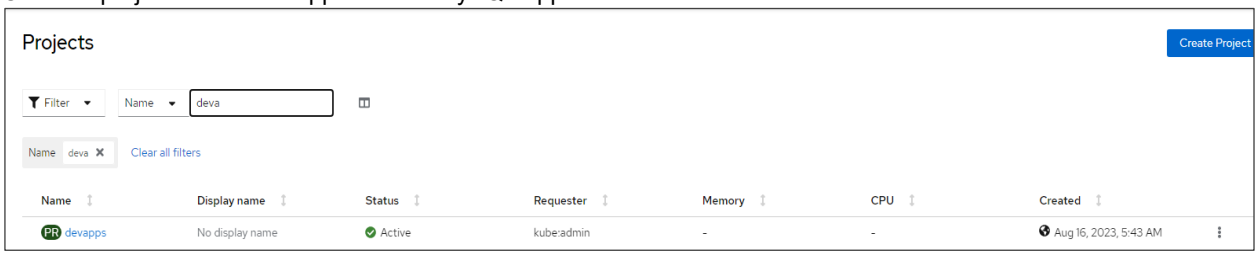


c. In Storage Navigator, verify whether the snapshot volume 00:00:89 is created successfully.

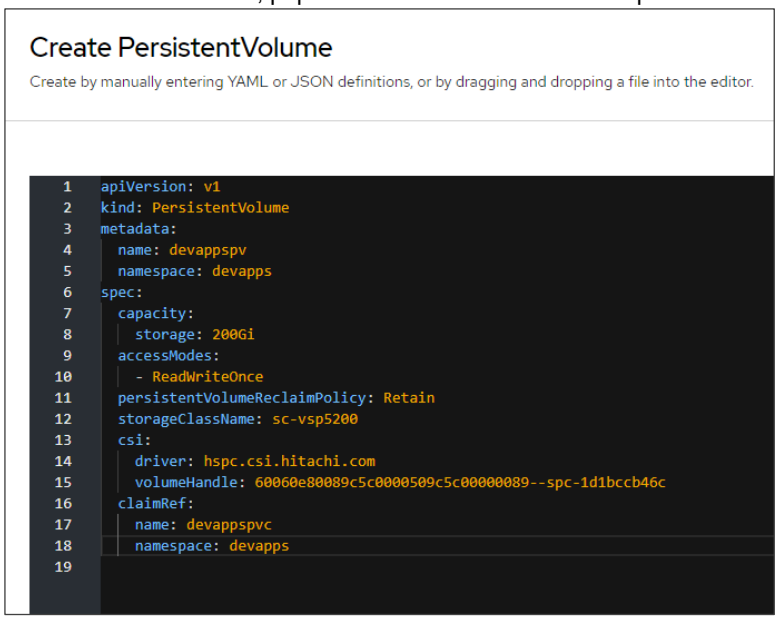
Restore Operation

To restore an application in an OpenShift cluster in AWS, complete the following steps:

1. Identify the volume handle string for the snapshot secondary volume 00:00:89. Volume handle string for this volume is "60060e80089c5c0000509c5c0000**0089--spc-1d1bccb46c**".
2. Create a PV and PVC using the volume 00:00:89 with the pre-defined volume handle string.
   a. Create a project called "devapps" for the MySQL application.

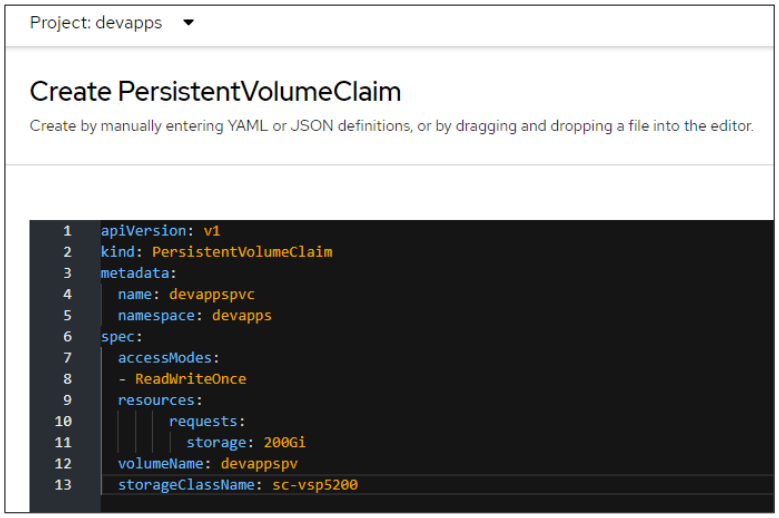


   b. Create a manifest file for PV using the volume handle string. This way, storage class does not dynamically create a new volume. Instead, it uses the existing volume to preserve the snapshot data. From the Red Hat OpenShift console, navigate to **Storage**, click **PersistantVolumes**, and then click **Create PersistentVolume**. In the Create PersistantVolume menu, populate the YAML file with the required information and click **Create**.



   c. Verify whether the PV is created.

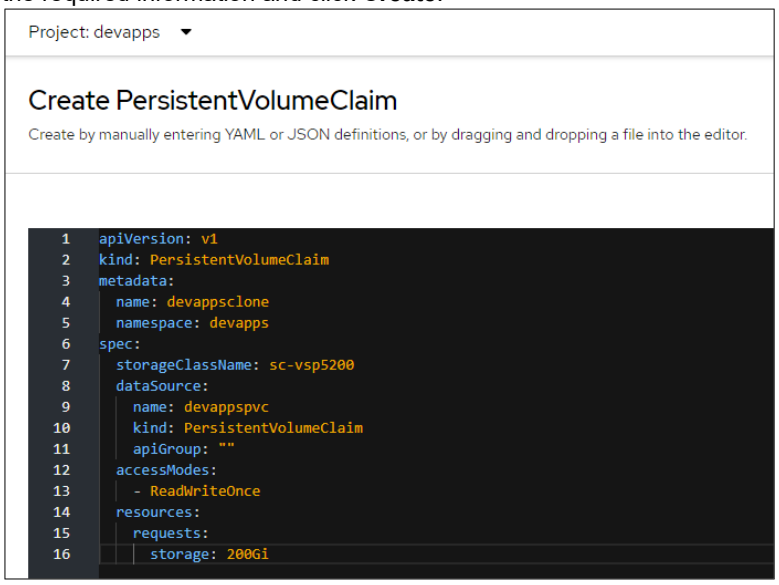
d. Create a manifest file for PVC using the PV created in step 2b. From the Red Hat OpenShift console, navigate to **Storage**, click **PersistantVolumeClaims**, and then click **Create PersistentVolumeClaim**. In the Create PersistantVolume menu, populate the YAML file with the required information and click **Create**.



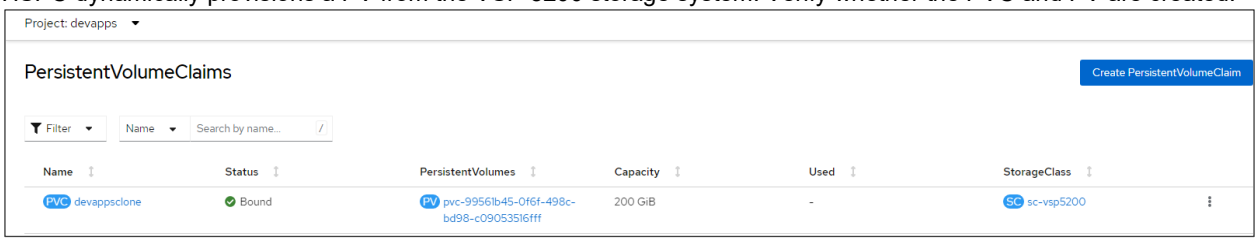
e. Verify whether the PVC is created.

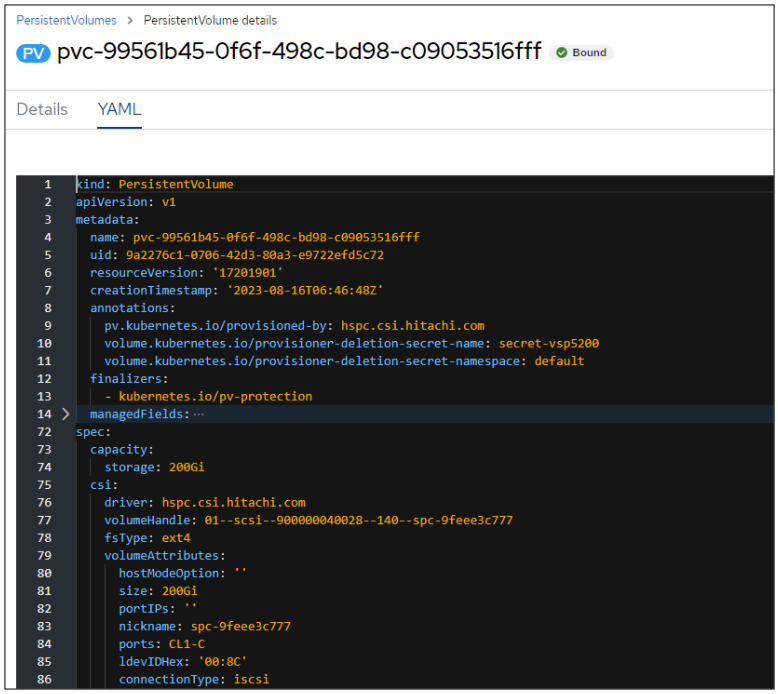
3. Create a clone PVC using the “devappspvc” PVC as data source.

a. Create a manifest file. From the Red Hat OpenShift console, navigate to **Storage**, click **PersistantVolumeClaims**, and then click **Create PersistentVolumeClaim**. In the Create PersistantVolume menu, populate the YAML file with the required information and click **Create**.

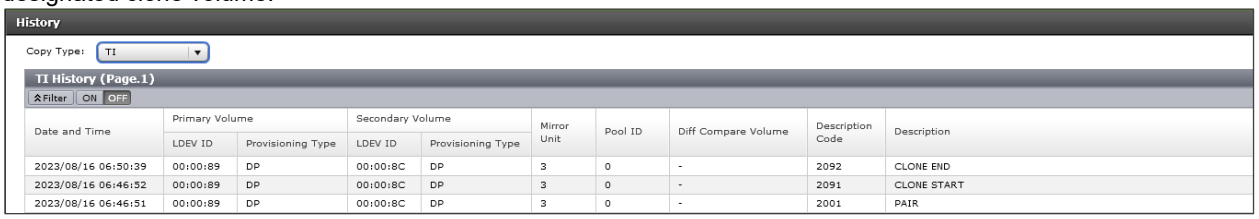


b. HSPC dynamically provisions a PV from the VSP 5200 storage system. Verify whether the PVC and PV are created.





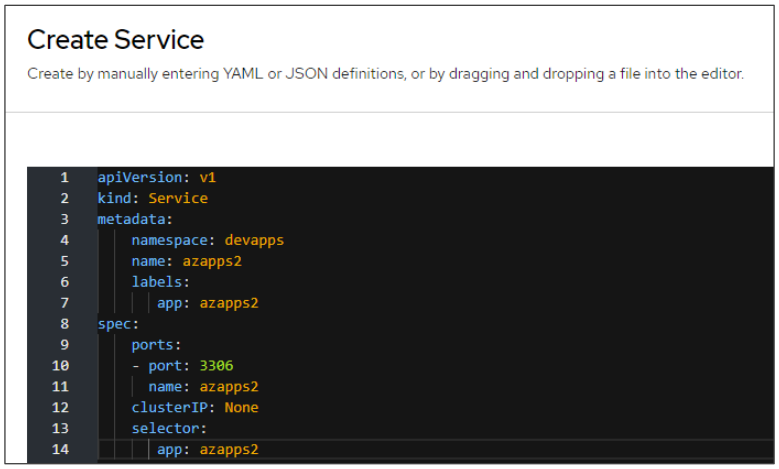
The dynamically created PV is the Thin Image clone volume. In the following screenshot, volume 00:00:8C is the designated clone volume.



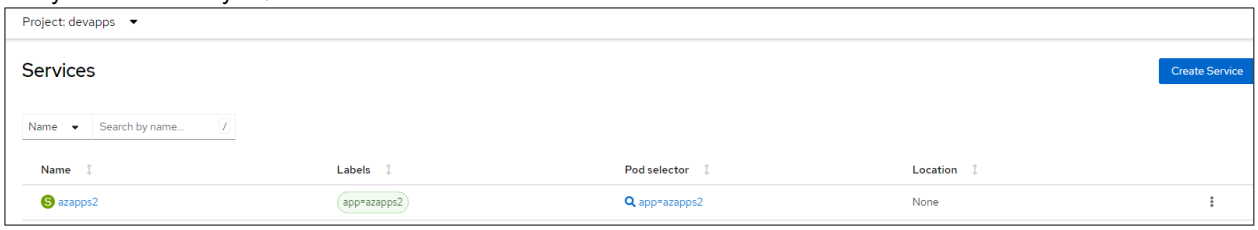
Date and Time	Primary Volume		Secondary Volume		Mirror Unit	Pool ID	Diff Compare Volume	Description Code	Description
	LDEV ID	Provisioning Type	LDEV ID	Provisioning Type					
2023/08/16 06:50:39	00:00:89	DP	00:00:8C	DP	3	0	-	2092	CLONE END
2023/08/16 06:46:52	00:00:89	DP	00:00:8C	DP	3	0	-	2091	CLONE START
2023/08/16 06:46:51	00:00:89	DP	00:00:8C	DP	3	0	-	2001	PAIR

4. Restore the MySQL application in the AWS cluster.

a. Create a MySQL service. From the Red Hat OpenShift console, navigate to **Networking**, click **Services**, and then click **Create Service**. In the Create Service menu, populate the YAML file with the required information and then click **Create**.



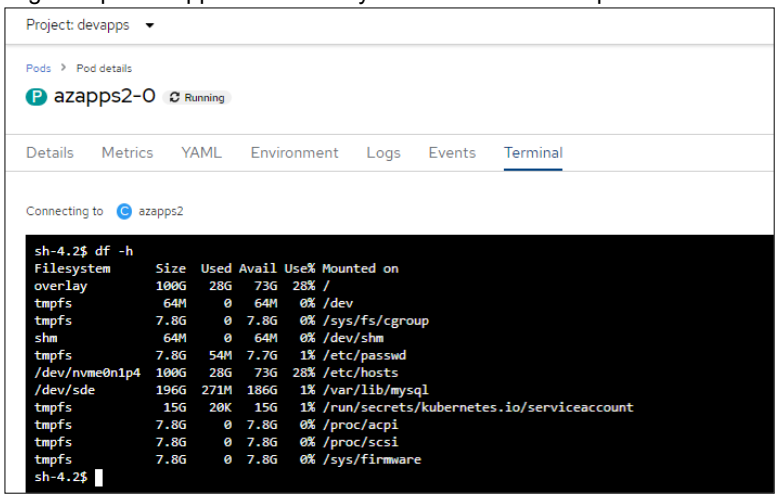
b. Verify whether the MySQL service is created.



c. Create a MySQL statefulset application. From the Red Hat OpenShift console, navigate to **Workloads**, click **StatefulSet**, and then click **Create StatefulSet**. In the Create StatefulSet menu, populate the YAML file with the required information and click **Create**. In the volume section, use the “devappsclone” claim created in step 3. This ensures that the MySQL application uses the clone PVC for persistent data.

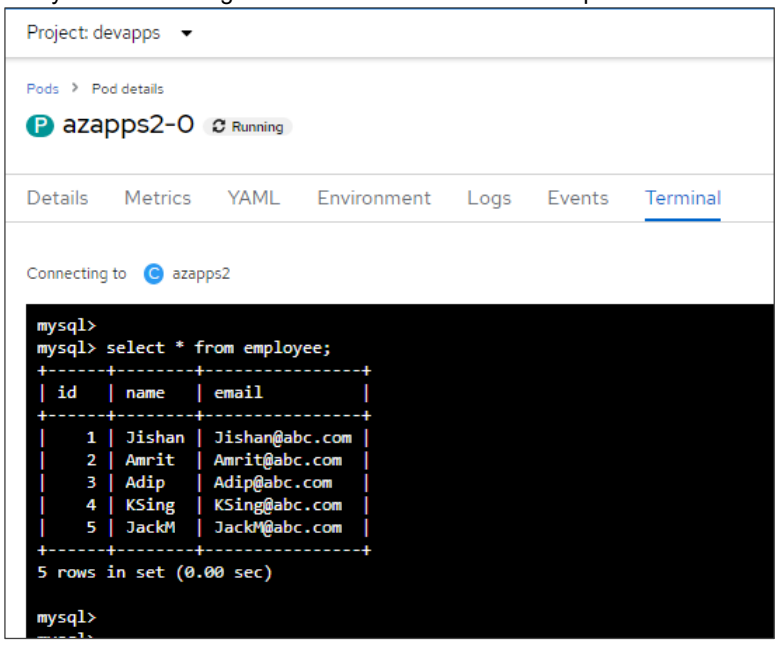
d. Verify whether the StatefulSet is running.

e. Log in to pod azapps2-0 and verify whether the 200 GB persistent volume is mounted on /var/lib/mysql.

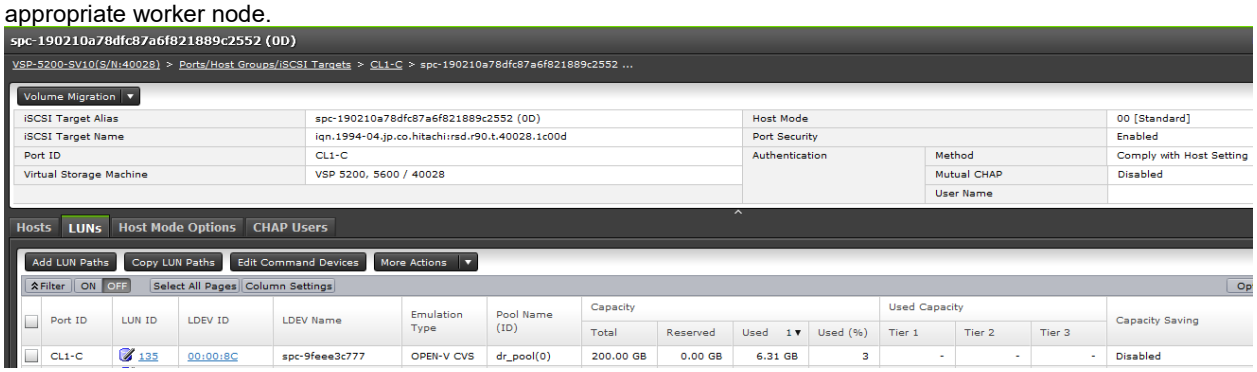


f. Log in to MySQL and verify whether the “devmysqldb1” database is available.

g. Verify whether the ingested data from Azure RedHat OpenShift cluster is available here.



HSPC automatically creates an iSCSI target on port CL1-C of the storage system and assigns the volume to the appropriate worker node.



Test 4: Migrate a Stateful Application Across OpenShift Clusters Using Kasten K10 Multi-Cluster

This test case describes the process of migrating a stateful application by performing backup and restore operations between two OpenShift clusters using Kasten K10 Multi-Cluster Global policy and HSPC. In this environment, backup and restore operations can be performed among three clusters residing on near-cloud, AWS, and Azure. To demonstrate this, we captured the snapshot of a stateful MySQL application running on Azure Red Hat OpenShift cluster and then restored it on a Red Hat OpenShift cluster in AWS. The entire process was performed from the Kasten K10 Multi-Cluster UI. The VSP 5200 storage system serves the persistent volumes required for stateful MySQL application in both clusters.

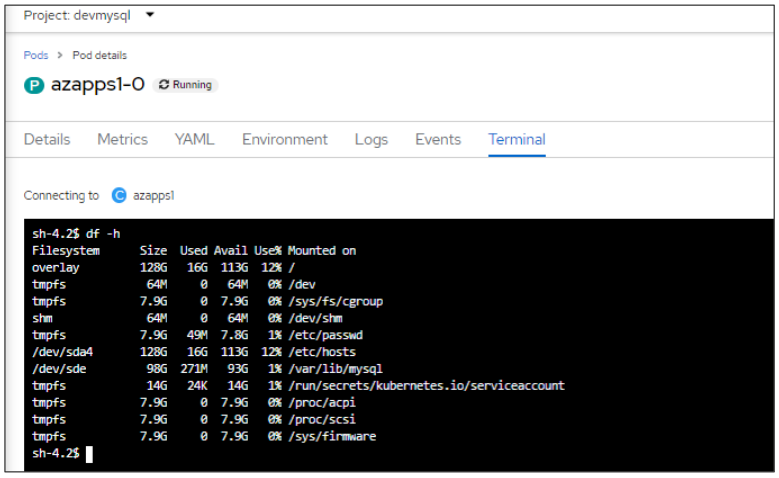
Snapshot Operation

Before performing snapshot operation of an application using Kasten K10, create the following:

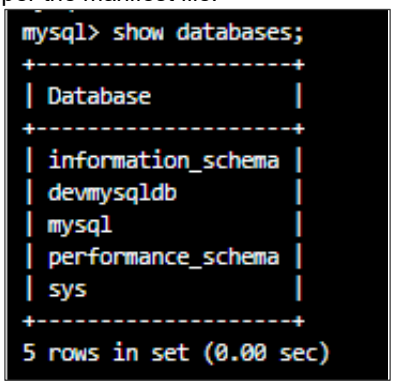
- Global Location Profile: Profiles define credentials and locations required to move the data in and out of the cluster. In this scenario, an Amazon S3 bucket is used.
- Global Policy: Policies are used to automate your data management workflows. To achieve this, they combine actions you want to take (such as snapshot), a frequency or schedule for how often you want to take that action, and a label-based selection criteria for the resources you want to manage.
- Distribution: Distributions define which K10 resources belong to which clusters.

To perform snapshot operation of an application using Kasten K10, complete the following steps:

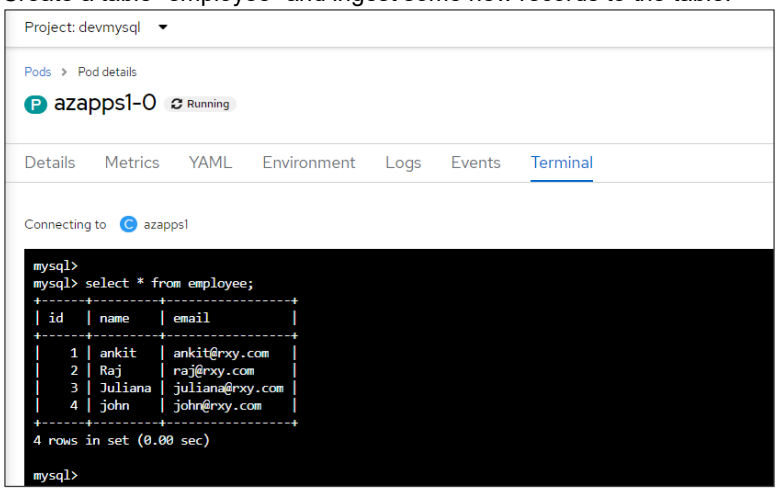
1. Create a new project “devmysql” and deploy a stateful MySQL application with a 100 GB persistent volume from the VSP 5200 storage system, as shown in the section Test 2: Deploy a Stateful Application in Azure Red Hat OpenShift Cluster.
2. Access the stateful MySQL application.
   a. Log in to pod MySQL and verify whether the 100 GB persistent volume is mounted in “/var/lib/mysql”, as per the manifest file.



   b. Log in to MySQL database using “**`mysql -u root -p`**” and verify whether the “devmysqldb” database is created as per the manifest file.



c. Create a table “employee” and ingest some new records to the table.



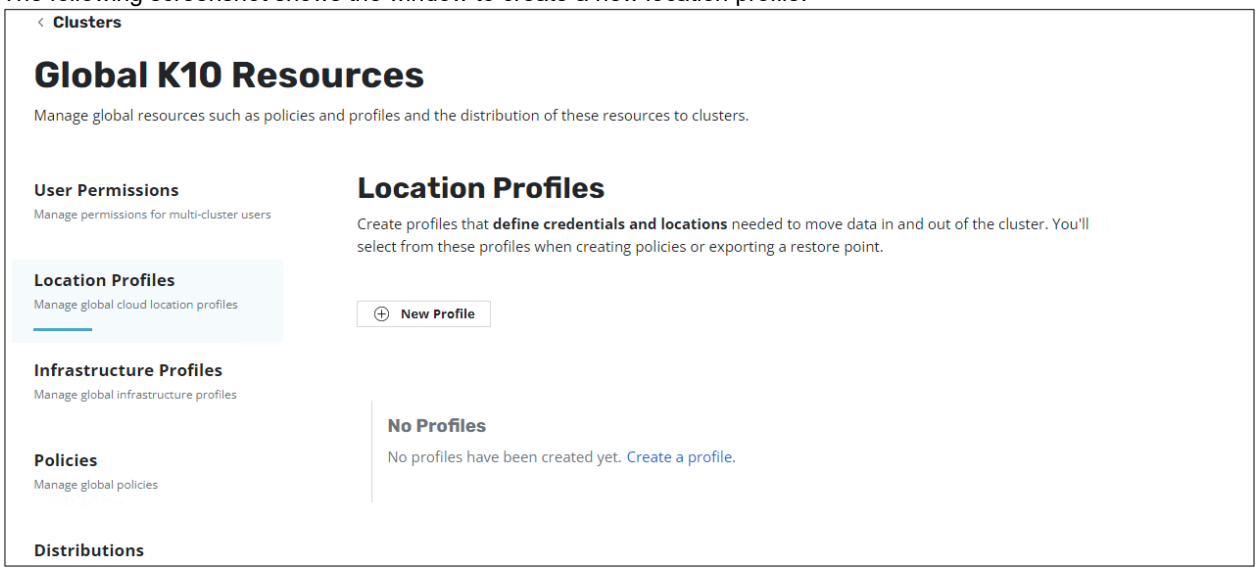
3. Verify that the application created in step 1 is registered in the Kasten K10 UI.
4. Navigate to the Kasten K10 Multi-Cluster Dashboard and verify the available clusters and registered applications.

5. Navigate to **Clusters**, select **azurecluster**, and view the registered applications. Kasten K10 registers detected namespaces as an application. The following screenshot shows that the namespace “devmysql" is detected as an application.

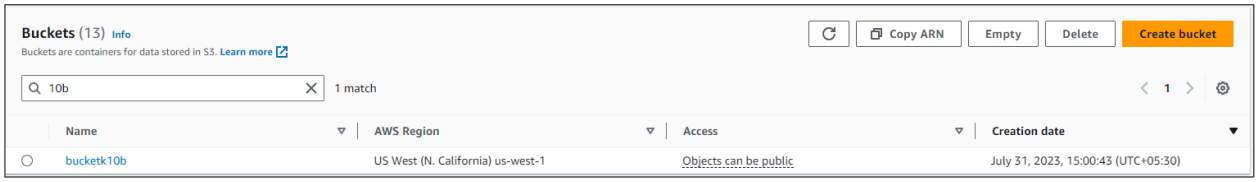
6. Create a Global Location Profile.

a. In the K10 Global Resources section of the K10 Multi-Cluster Dashboard, click **Global Profile**.

The following screenshot shows the window to create a new location profile:

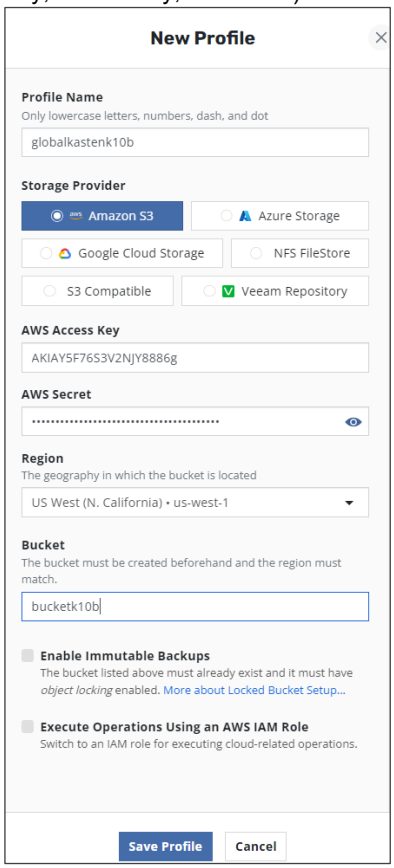


b. In this scenario, an Amazon S3 bucket named bucketk10b is created and designated as the destination for Global Location Profile. For instructions to create an S3 bucket, see the Amazon S3 User Guide.

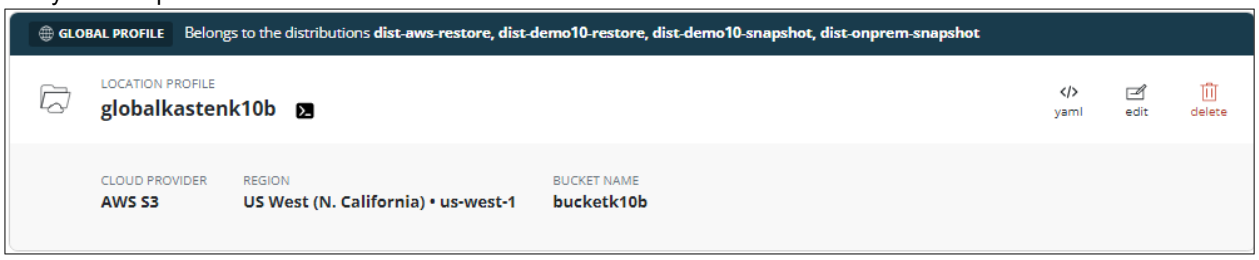


c. In the Global K10 Resources window, click **New Profile**.

d. Enter the required information (such as Profile Name, Storage Provider, AWS region, Bucket Name, AWS Access Key, Secret Key, and so on) and click **Save Profile**.



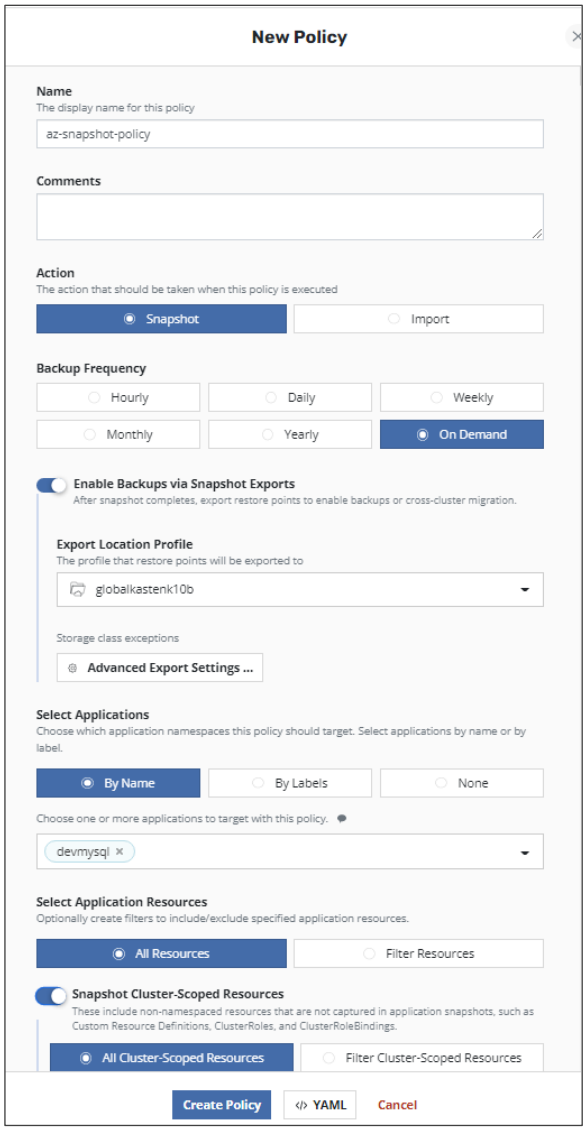
e. Verify that the profile is created.



7. Create a Global Snapshot Policy.

   a. From the K10 Global Resources page, click **Global Polices** and then click **New Policy**.

   b. Enter the snapshot related information (such as Policy Name, Backup Frequency, target application, application resources, and so on). Select **Enable Backup via Snapshot Exports**, select the location profile that you created, and click **Create Policy**. This is required to generate an import policy while restoring the application.

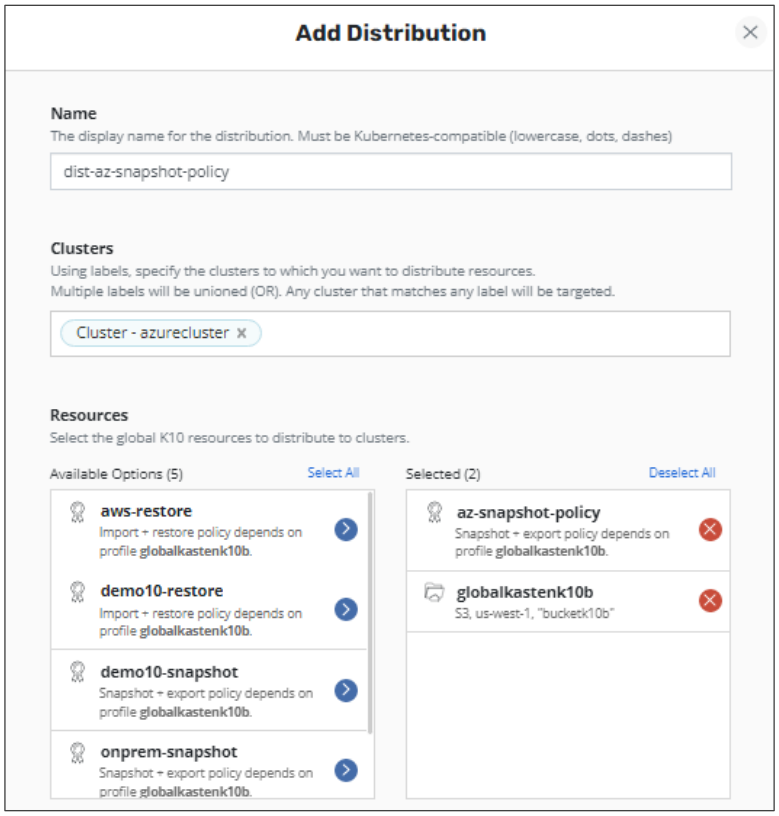


c. Verify that the policy is created.

8. Create a distribution.

   a. From the K10 Global Resources page, click **Distributions** and then click **New Distribution**.

   b. In the Add Distribution window, enter the required information (such as Distribution Name), specify the Azure cluster, specify the two resources created (Global Location Profile and Global Snapshot Policy), and then click **Add Distribution**.



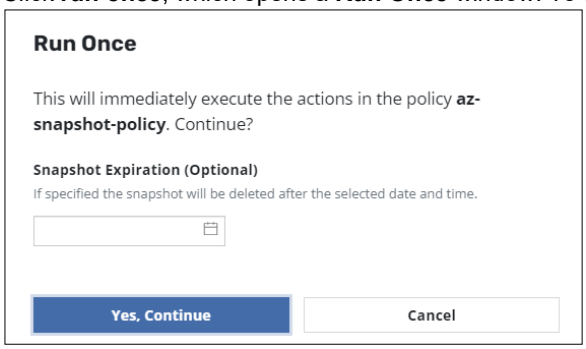
c. Verify that the distribution is added.

9. Collect a snapshot of the registered application using the Global Snapshot Policy.

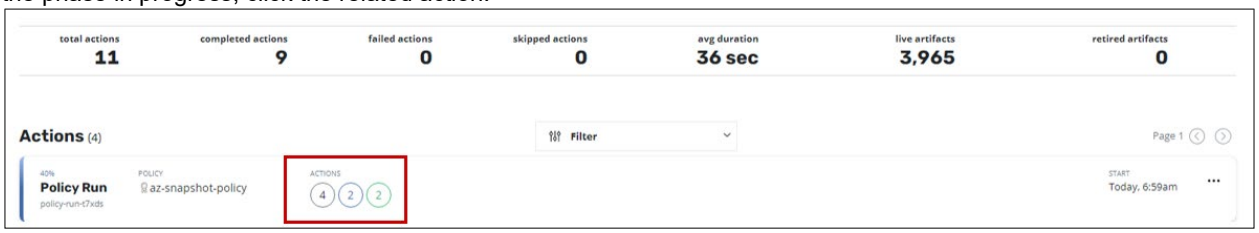
a. From the K10 Multi-Cluster Dashboard, click Cluster "**azurecluster**" and then click **Policies**.

Verify that the Global snapshot on-demand policy created in the Global Policy section is available under **Polices**.

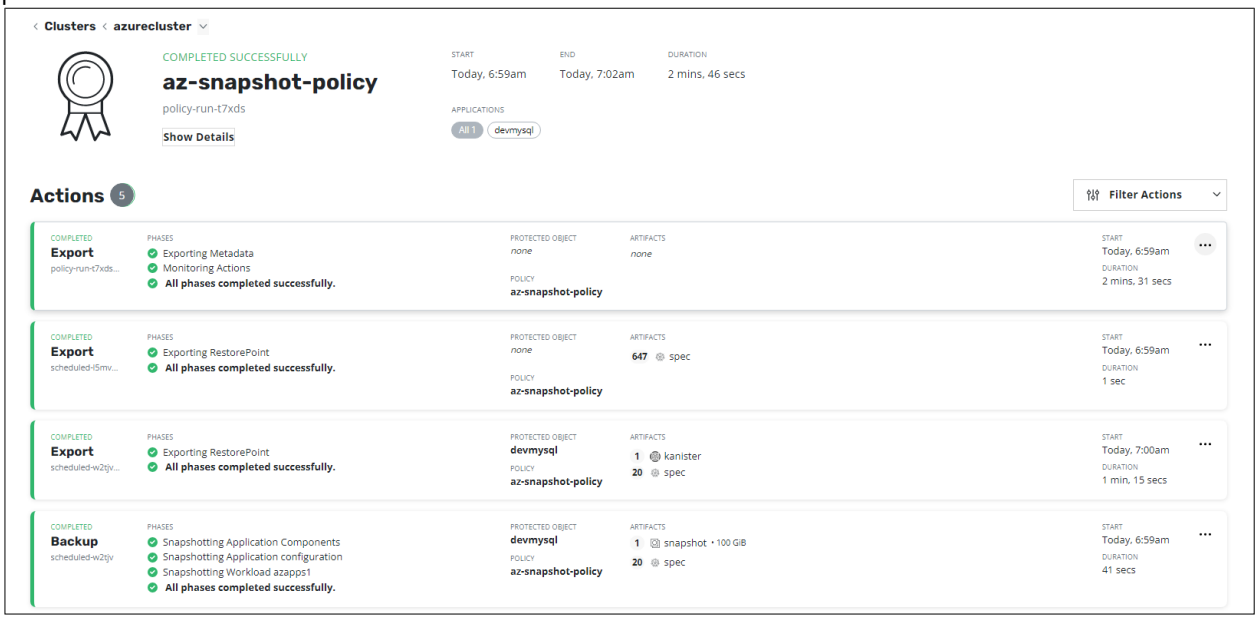
b. Click **run once**, which opens a **Run Once** window. To start the snapshot, click **Yes, continue**.



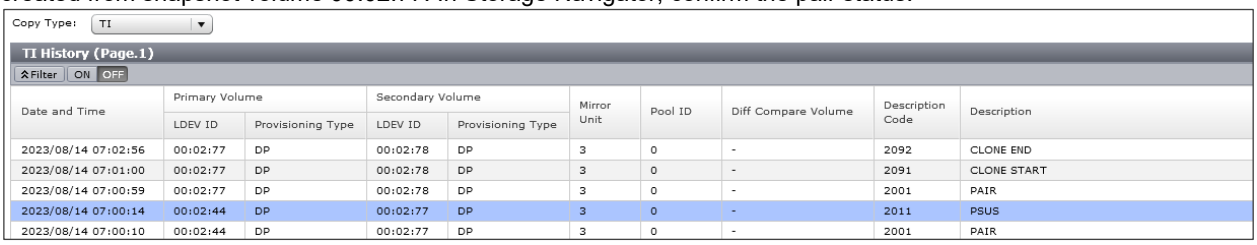
c. Open the Kasten K10 Multi-Cluster dashboard and check the status of the policy in the **Actions** window. To check the phase in progress, click the related action.



d. Verify that the phase has changed to Completed Successfully. Click the relevant action to confirm that no error is present.



e. Integrating Kasten K10 with HSPC creates a Thin Image snapshot and splits the pairs. A clone volume 00:02:78 was created from snapshot volume 00:02:77. In Storage Navigator, confirm the pair status.



Copy Type: TI

TI History (Page.1)

Filter ON OFF

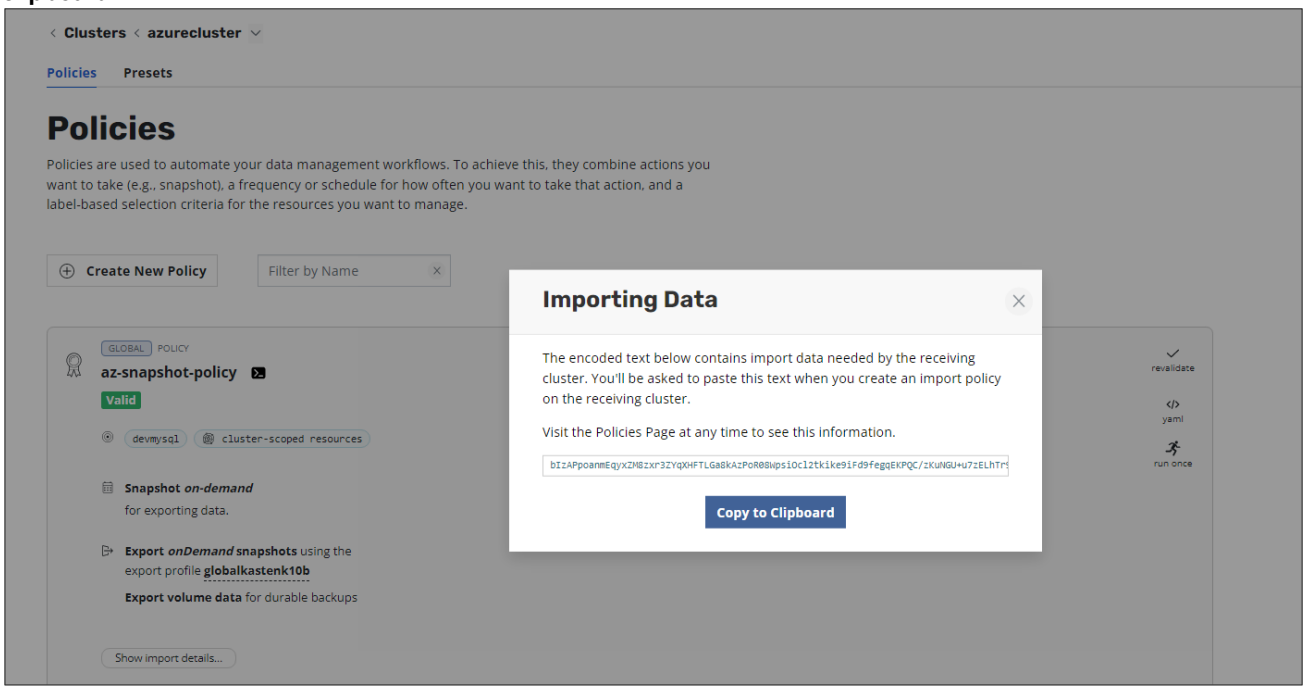
Date and Time	Primary Volume		Secondary Volume		Mirror Unit	Pool ID	Diff Compare Volume	Description Code	Description
	LDEV ID	Provisioning Type	LDEV ID	Provisioning Type					
2023/08/14 07:02:56	00:02:77	DP	00:02:78	DP	3	0	-	2092	CLONE END
2023/08/14 07:01:00	00:02:77	DP	00:02:78	DP	3	0	-	2091	CLONE START
2023/08/14 07:00:59	00:02:77	DP	00:02:78	DP	3	0	-	2001	PAIR
2023/08/14 07:00:14	00:02:44	DP	00:02:77	DP	3	0	-	2011	PSUS
2023/08/14 07:00:10	00:02:44	DP	00:02:77	DP	3	0	-	2001	PAIR

Restore Operation

You can restore an application from snapshot across clusters from the Kasten K10 Multi-Cluster. Restore operation consists of the following high-level steps:

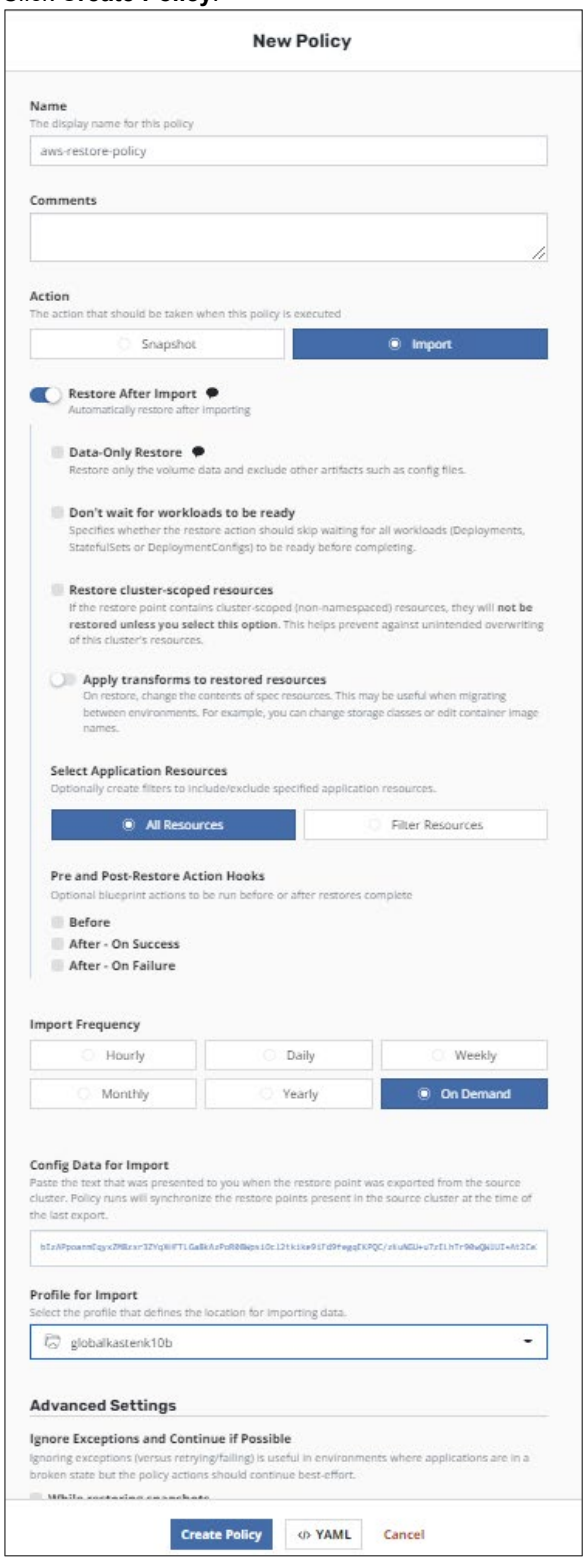
- Copying the Import data.
- Creating a Restore policy.
- Implementing the Restore Policy.

1. To copy the Import data, from the K10 Multi-Cluster Dashboard, click the production Cluster "**azurecluster**" and then click **Policies**.
2. In the Polices window, select the Global Snapshot Policy created earlier, click **Show Import data**, and then click **Copy to clipboard**.

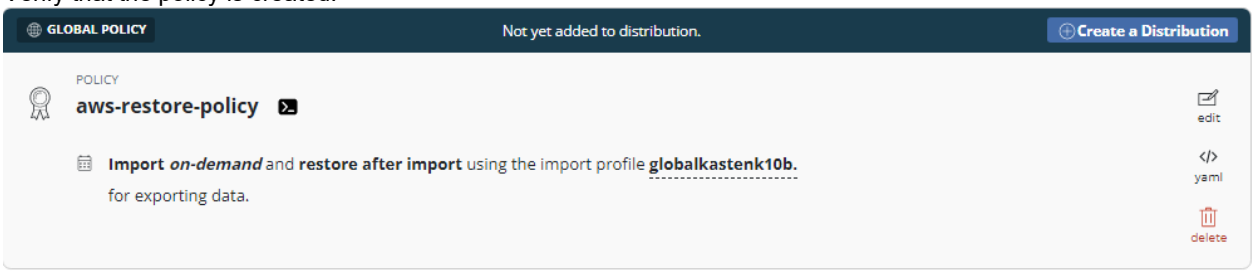


3. Create a restore policy.
   a. From the K10 Global Resources page, click **Polices** and then click **New Policy**.
   b. In the New Policy window, enter a Policy Name and select Import Frequency as **On Demand**.
   c. In Config Data for Import section, paste the import policy copied in step 2.
   d. Select **Restore after Import** and select the Global Location in Profile for Import.

e. Click **Create Policy**.



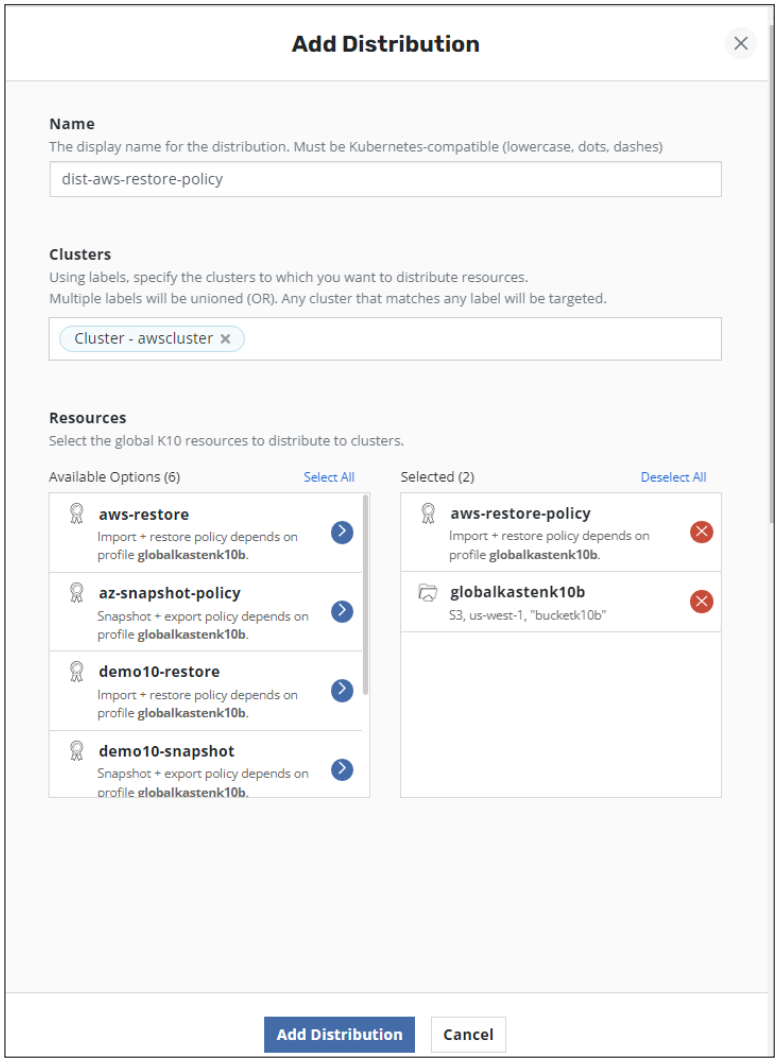
f. Verify that the policy is created.



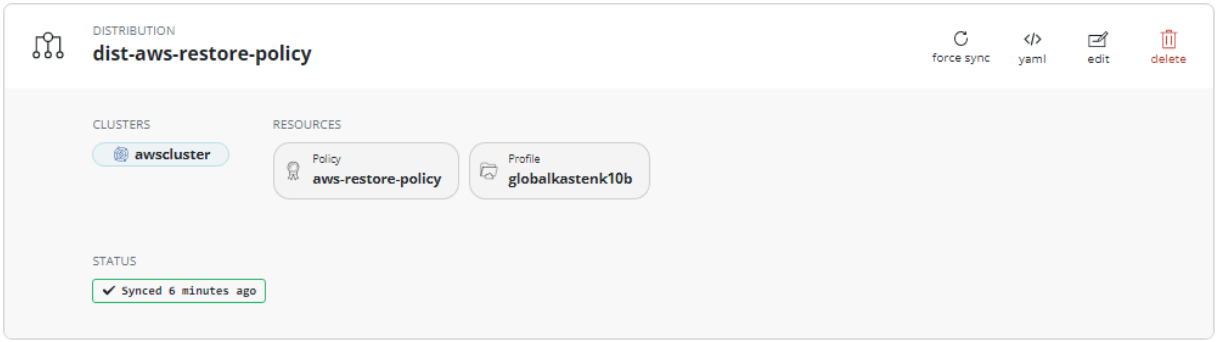
4. Create a Distribution.

a. From the K10 Global Resources page, click **Distributions** and then click **New Distribution**.

b. In the Add Distribution window, enter the name, specify the AWS cluster, select the restore policy and location profile in Resources, and then click **Add Distribution**.



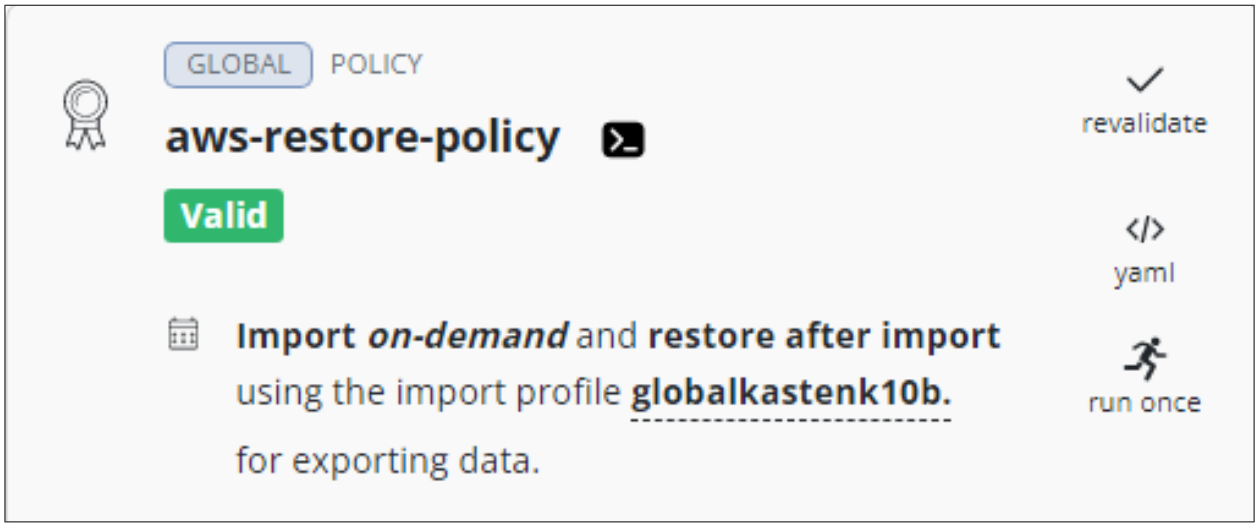
c. Verify that the distribution is added.



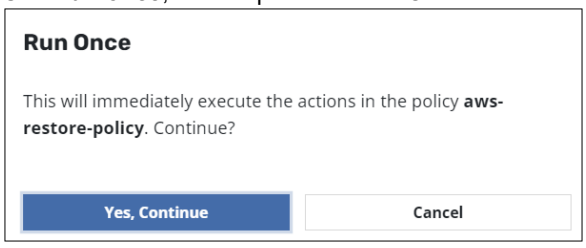
5. Run the restore operation.

a. From the K10 Multi-Cluster Dashboard, click the secondary cluster "**awscluster**" and then click **Policies**.

b. Verify whether the Global restore on-demand policy created in the Create Restore Policy section is available under **Polices**.

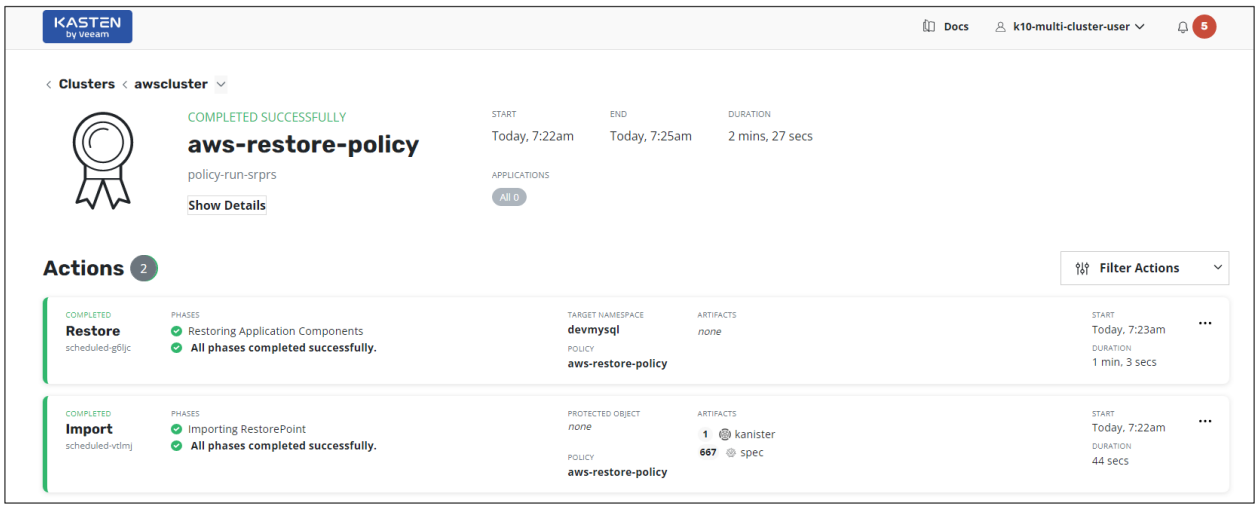


c. Click **run once**, which opens the Run Once window. To start the restore, click **Yes, continue**.



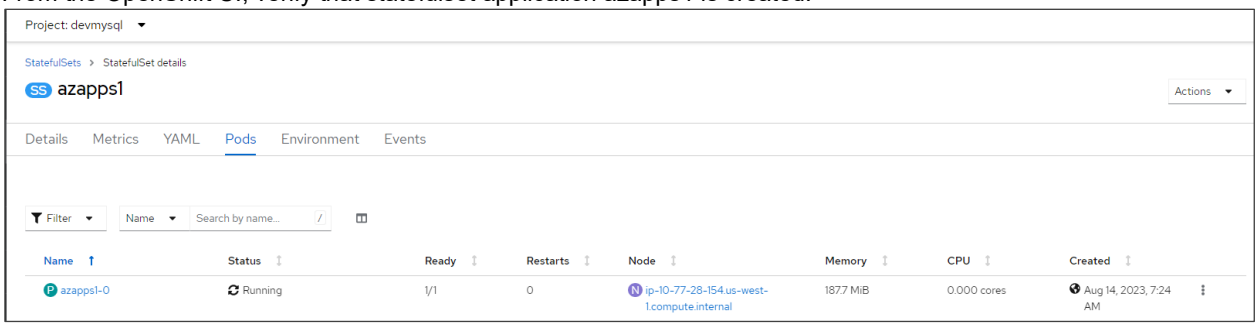
d. Open the K10 Multi-Cluster Dashboard and check the status of the policy in the **Actions** window. To check the phase in progress, click the related action.

e. Verify that the phase has changed to Completed Successfully. To confirm that no error is present, click the relevant action.



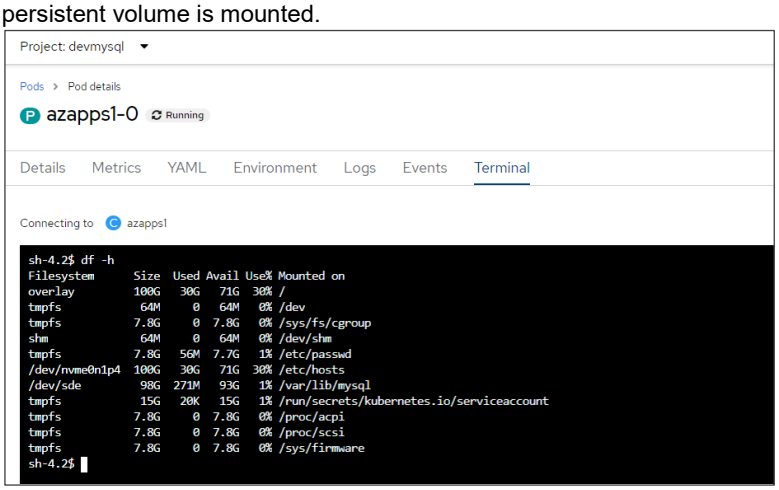
f. From the K10 Multi-Cluster Dashboard, navigate to the cluster “awscluster” and verify that application “devmysql” is restored.

g. From the OpenShift UI, verify that statefulset application azapps1 is created.



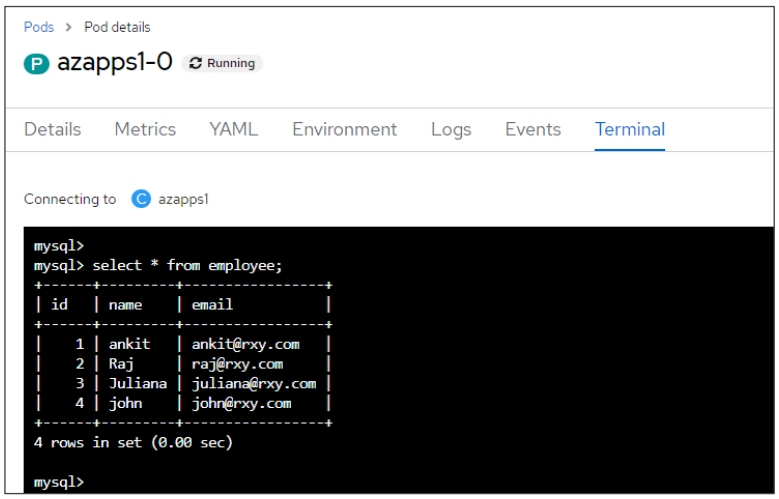
6. Verify that the data is available.

a. Log in to the pod azapps1-0 in devmysql namespace in the awscluster cluster and verify whether the 100 GB persistent volume is mounted.



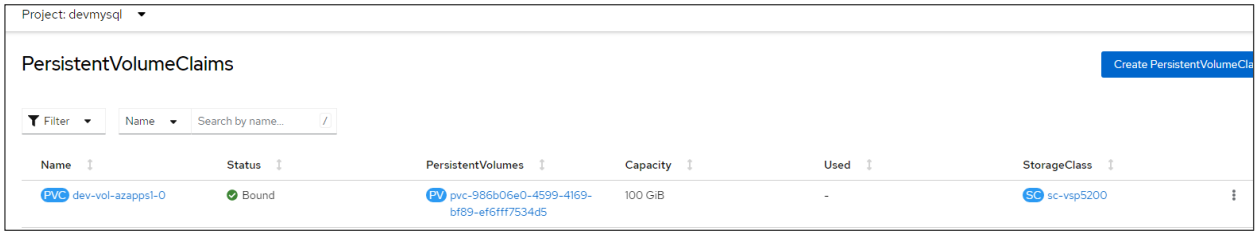
b. Log in to MySQL and verify whether database devmysqldb and employee table is available.

c. Verify whether the ingested data in Azure cluster “azurecluster” (as shown in ARO cluster section) is available in “awscluster” cluster in AWS.

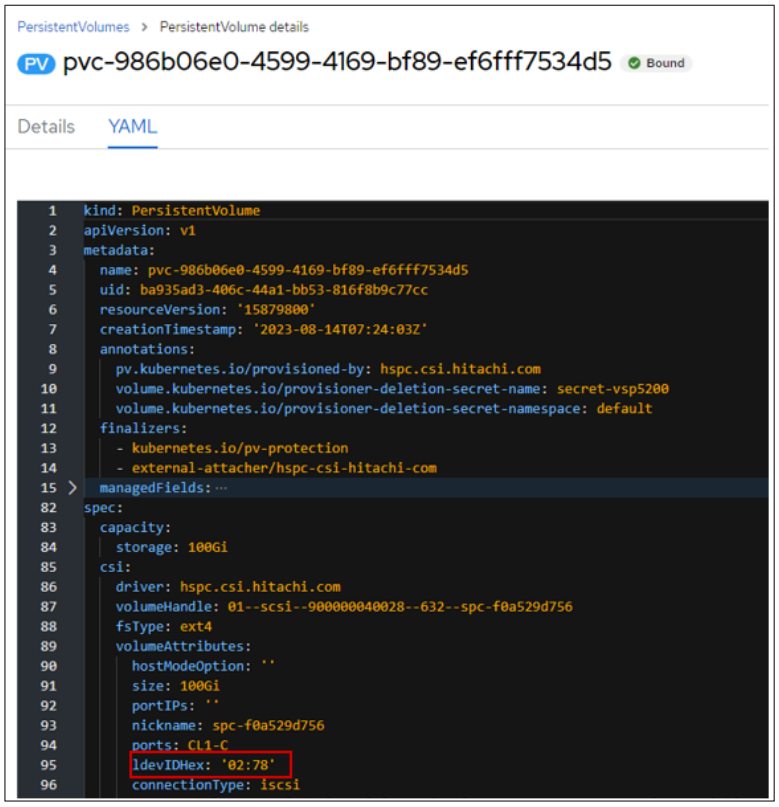


The following screenshots show the status of PVC and PV created during restoration.

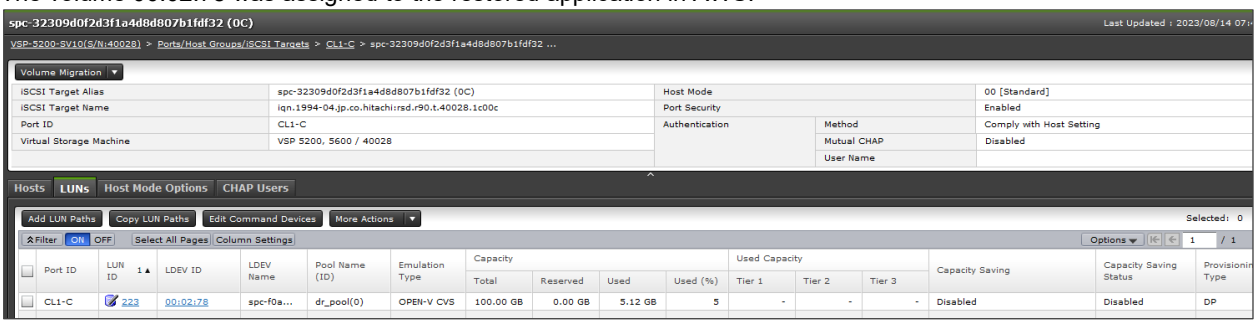
Persistent volume claim:



Persistent volume:



The volume 00:02:78 was assigned to the restored application in AWS.



Test 5: Recover from a Ransomware Attack

This test case demonstrates how a VSP snapshot combined with immutability feature from Data Retention Utility program product can be used to recover a stateful application affected by a ransomware attack. For this test case, volume snapshot of the persistent volume used in the application in Azure Red Hat OpenShift cluster has already been taken and the DRU write-disable attribute is set on the snapshot volume.

Assume that the application is affected by a ransomware attack and we must restore clean data from the snapshot. This recovery process can be carried out either in Azure or in AWS.

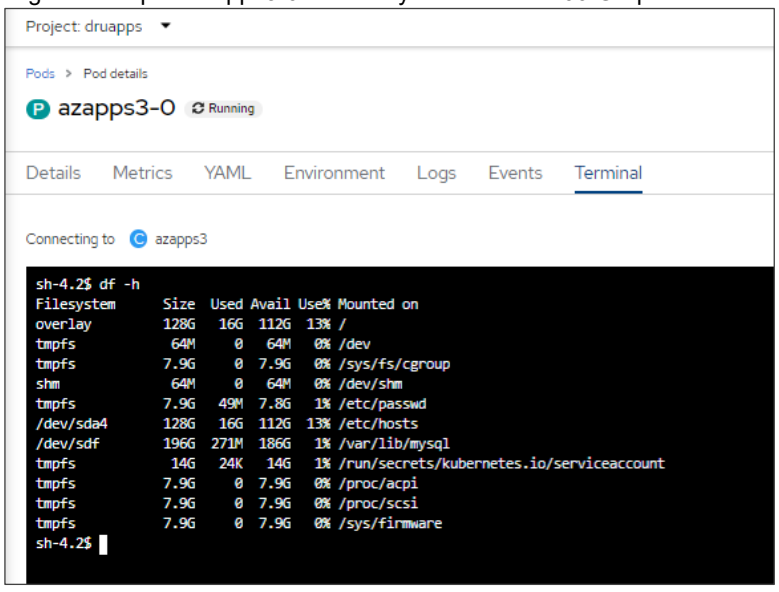
Recovering from a ransomware attack consists of the following high-level steps:

- Creating PVC with the snapshot volume (where the DRU attribute Write Disabled is set).
- Creating a cascaded snapshot of this volume because write is disabled.
- Using the cascaded snapshot (snap-on-snap) to recover the application data in any cluster.
- Creating a clone PVC and using that PVC as data volume to restore the MySQL application because snapshot volumes must not be directly used in a POD.
- Verifying that the data ingested from the ARO cluster is available.

Snapshot Operation

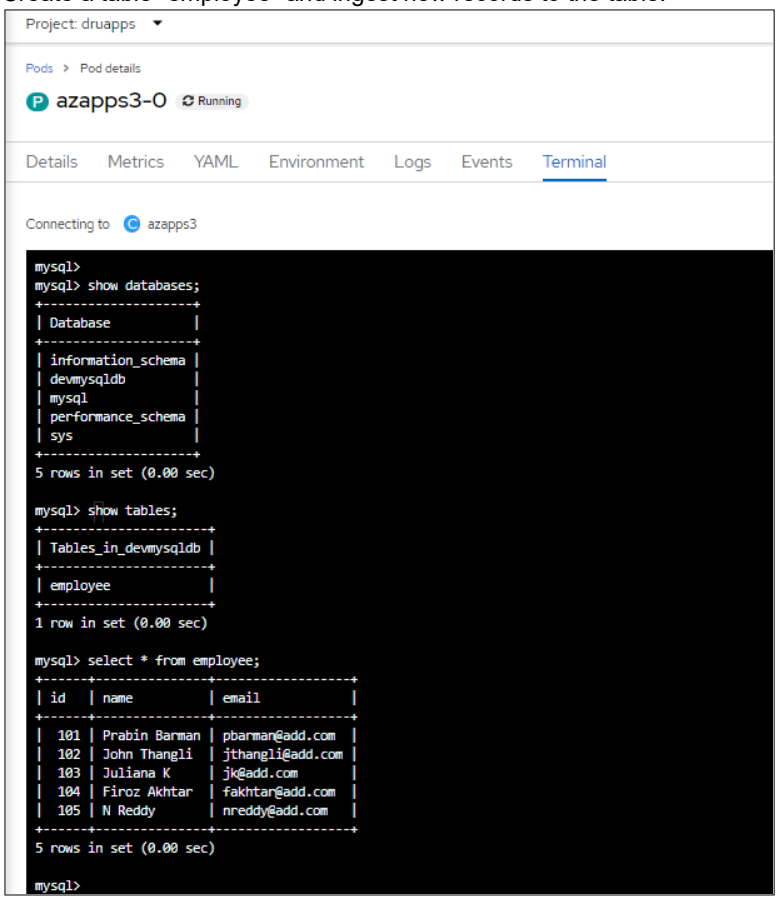
Complete the following steps in Azure Red Hat OpenShift Cluster:

1. Create a new project “druapps” and deploy a stateful MySQL application with a persistent volume of 200 GB from the VSP 5200 storage system, as shown in the section Test 2: Deploy a Stateful Application in Azure Red Hat OpenShift Cluster.
2. Access the stateful MySQL application.
   a. Log in to the pod azapps3-0 and verify whether the 200 GB persistent volume is mounted on “/var/lib/mysql”.



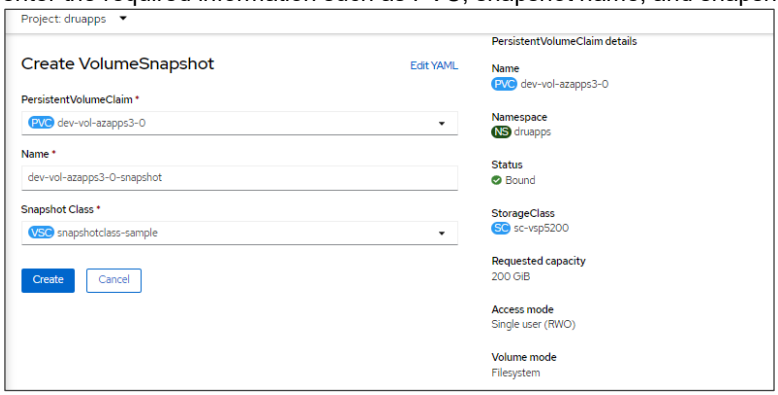
   b. Log in to MySQL and verify whether the database “devmysqldb” is created.

c. Create a table “employee” and ingest new records to the table.



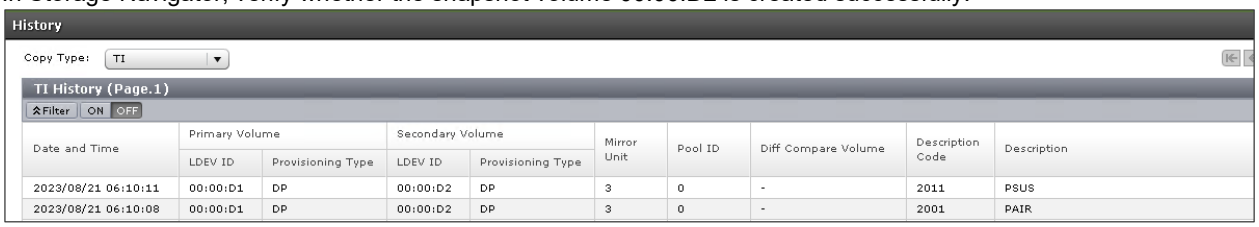
3. Create a snapshot.

a. Create a snapshot of the PVC used in the MySQL application. From the Red Hat OpenShift console, navigate to **Storage**, click **VolumeSnapshots**, and then click **Create VolumeSnapshot**. In the Create VolumeSnapshot menu, enter the required information such as PVC, snapshot name, and snapshot class, and then click **Create**.



b. Verify whether the snapshot is created.

c. In Storage Navigator, verify whether the snapshot volume 00:00:D2 is created successfully.



Date and Time	Primary Volume		Secondary Volume		Mirror Unit	Pool ID	Diff Compare Volume	Description Code	Description
	LDEV ID	Provisioning Type	LDEV ID	Provisioning Type					
2023/08/21 06:10:11	00:00:D1	DP	00:00:D2	DP	3	0	-	2011	PSUS
2023/08/21 06:10:08	00:00:D1	DP	00:00:D2	DP	3	0	-	2001	PAIR

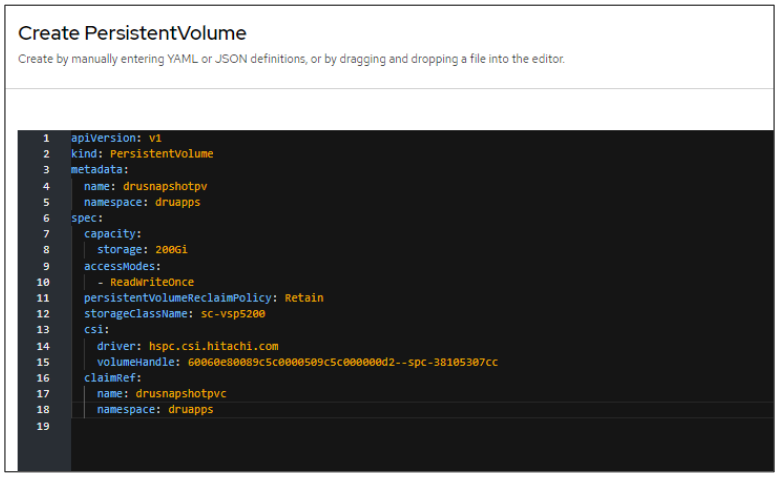
4. Set DRU write-disabled attribute to snapshot volume 00:00:D2 (LDEV# 210 in decimal format).

```
[root@linuxnfscl2 etc]# raidvchkdsp -g grp0 -fd -v gflag -Il
Group    PairVol  Device_File      Seq# LDEV# GI-C-R-W-S  PI-C-R-W-S  R-Time
grp0     pair0    Unknown        540028   210  E E E E E   E E E E E       0
[root@linuxnfscl2 etc]#
[root@linuxnfscl2 etc]# raidvchkset -g grp0 -vg wtd 5 -Il
[root@linuxnfscl2 etc]# raidvchkdsp -g grp0 -fd -v gflag -Il
Group    PairVol  Device_File      Seq# LDEV# GI-C-R-W-S  PI-C-R-W-S  R-Time
grp0     pair0    Unknown        540028   210  E E E D E   E E E D E       5
[root@linuxnfscl2 etc]#
```

Restore Operation

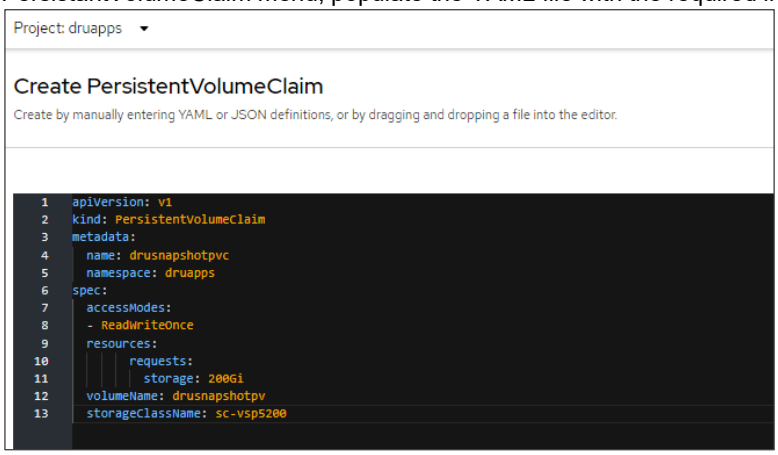
This section shows the restoration procedure when an application in the ARO cluster is affected by ransomware. Restore operation is performed in Red Hat OpenShift cluster in AWS.

1. Create a project “druapps” in OpenShift cluster in AWS.
2. Create a PV for snapshot volume 00:00:D2.
   a. Identify the volume handle string for the snapshot volume 00:00:D2. The volume handle string for this volume is “60060e80089c5c0000509c5c0000**00d2--spc-38105307cc**”.
   b. In the string, the volume ID is “00d2” and the volume name is “spc-38105307cc”. The volume name is automatically assigned by HSPC.
   c. Create a manifest file for PV using the volume handle string. This ensures that storage class does not dynamically create a new volume; instead, it uses the existing volume for preserving the snapshot data.
   d. From the Red Hat OpenShift console, navigate to **Storage**, click **PersistantVolumes**, and then click **Create PersistentVolume**. In the Create PersistantVolume menu, populate the YAML file with the required information and click **Create**.

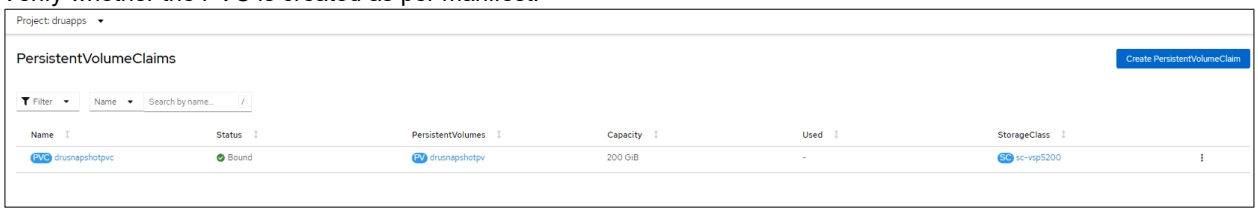


e. Verify whether the PV is created as per manifest.

f. Create a manifest file for PVC using the PV which you created. From the Red Hat OpenShift console, navigate to **Storage**, click **PersistantVolumeClaims**, and then click **Create PersistentVolumeClaim**. In the Create PersistantVolumeClaim menu, populate the YAML file with the required information and click **Create**.

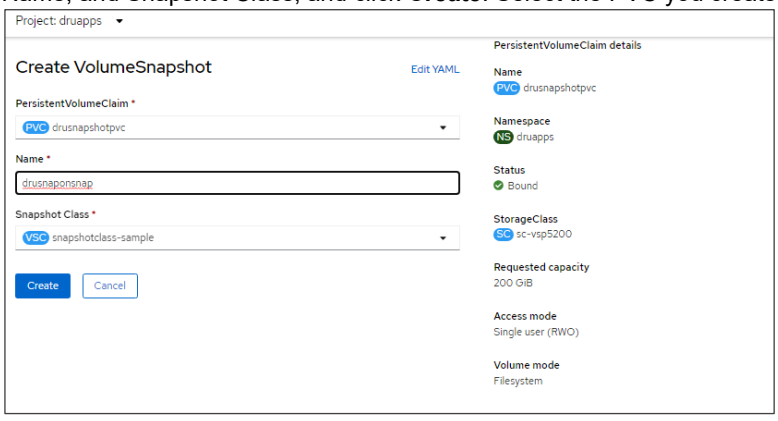


g. Verify whether the PVC is created as per manifest.

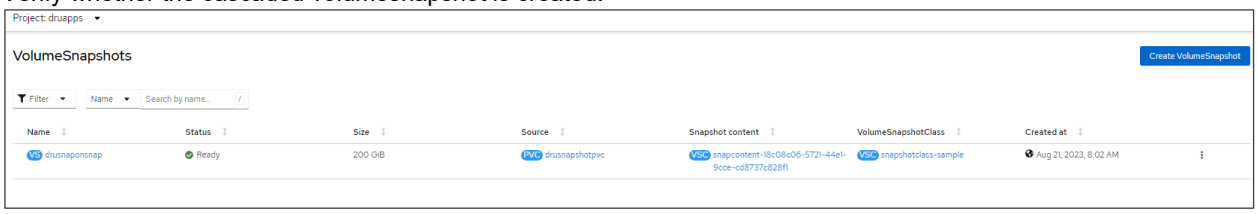


3. Create a cascaded snapshot from the volume 00:00:D2.

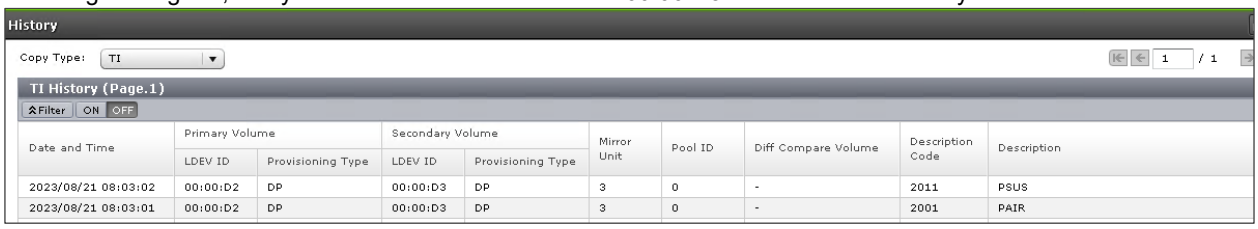
a. From the Red Hat OpenShift console, navigate to **Storage**, click **VolumeSnapshots**, and then click **Create VolumeSnapshot**. In the Create VolumeSnapshot menu, enter the required information such as PVC, Snapshot Name, and Snapshot Class, and click **Create**. Select the PVC you created in step 2.



b. Verify whether the cascaded volumesnapshot is created.



c. In Storage Navigator, verify whether the cascaded volume 00:00:D3 is created successfully.



Date and Time	Primary Volume		Secondary Volume		Mirror Unit	Pool ID	Diff Compare Volume	Description Code	Description
	LDEV ID	Provisioning Type	LDEV ID	Provisioning Type					
2023/08/21 08:03:02	00:00:D2	DP	00:00:D3	DP	3	0	-	2011	PSUS
2023/08/21 08:03:01	00:00:D2	DP	00:00:D3	DP	3	0	-	2001	PAIR

The DRU attribute is not set on the new cascaded snapshot volume 00:00:D3.

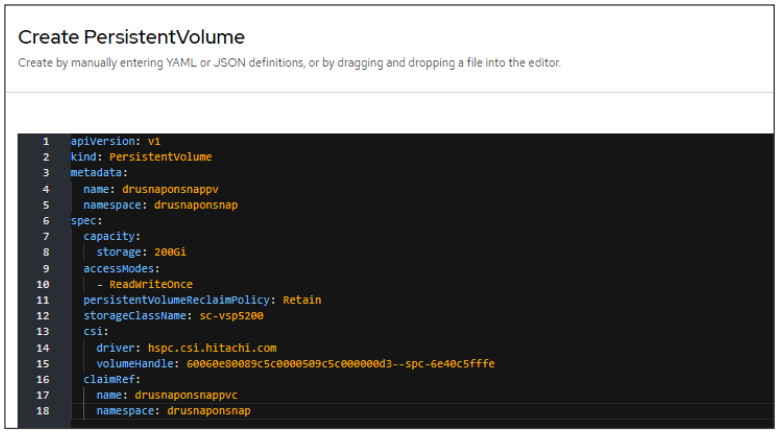
```
[root@linuxnfscl2 etc]# raidvchkdsp -g grp1 -fd -v gflag -Il
Group    PairVol  Device_File      Seq# LDEV# GI-C-R-W-S  PI-C-R-W-S  R-Time
grp1     pair0    Unknown        540028   211  E E E E E   E E E E E       0
[root@linuxnfscl2 etc]#
```

4. Use the snap-on-snap volume to create a clone volume to restore the application.

a. Create a project “drusnaponsnap”.

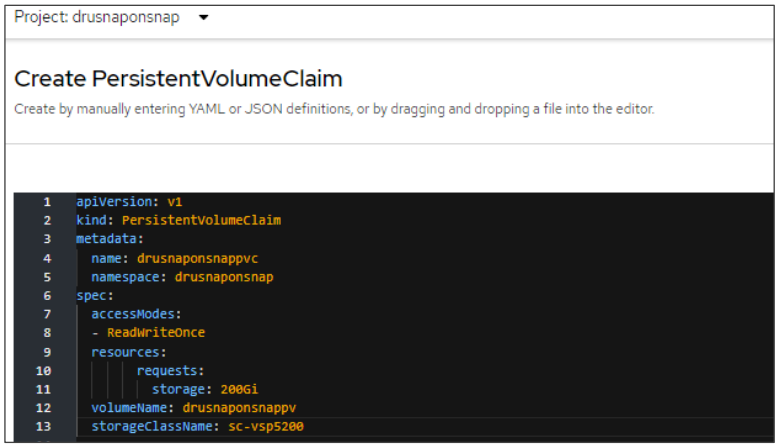
b. Create a manifest file for PV using the volume handle string of snap-on-snap volume 00:00:D3. This way, storage class does not dynamically create a new volume; instead, it uses the existing volume for preserving the snapshot data.

c. From the Red Hat OpenShift console, navigate to **Storage**, click **PersistantVolumes**, and then click **Create PersistentVolume**. In the Create PersistantVolume menu, populate the YAML file with the required information and click **Create**.



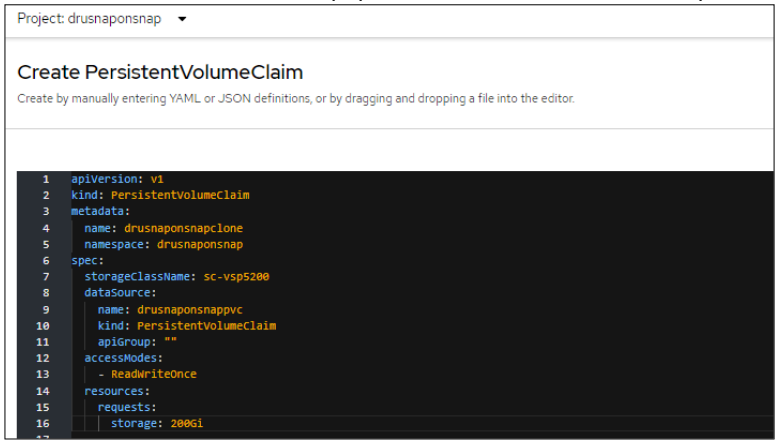
d. Verify whether the PV is created as per manifest.

e. Create a manifest file for PVC using the PV created in step 4. From the Red Hat OpenShift console, navigate to **Storage**, click **PersistantVolumeClaims**, and then click **Create PersistentVolumeClaim**. In the Create PersistantVolume menu, populate the YAML file with the required information and click **Create**.

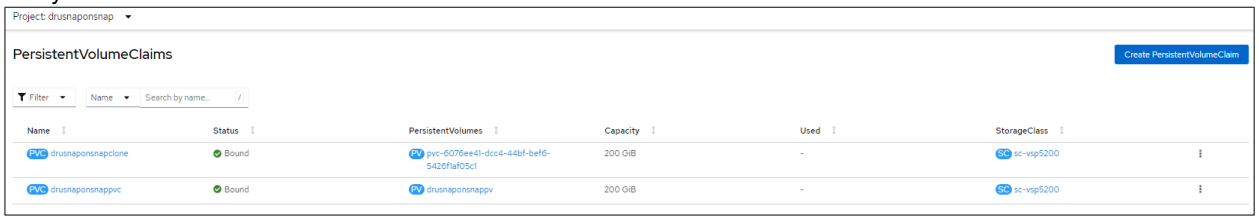


f. Verify whether the PVC is created as per manifest.

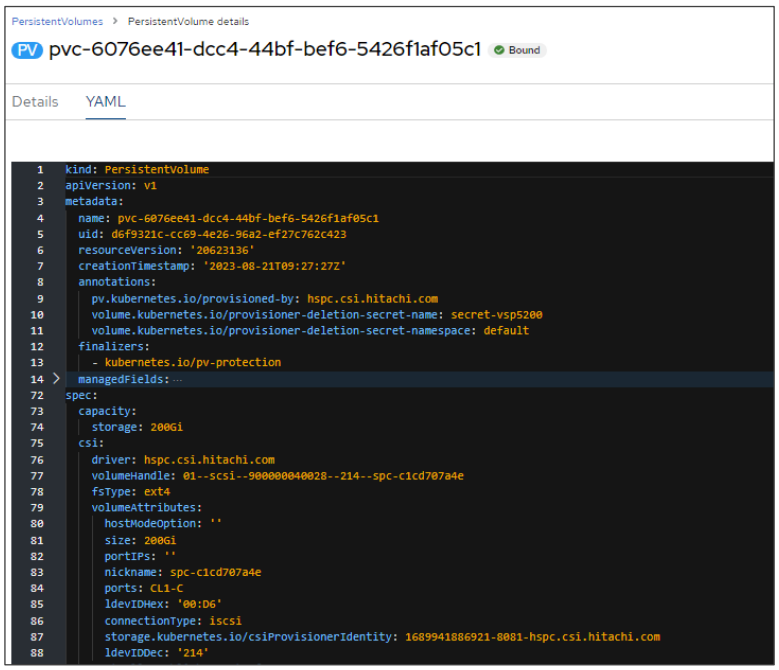
g. Create a clone PVC using the snapshot PVC created in step 4e as dataSource. From the Red Hat OpenShift console, navigate to **Storage**, click **PersistantVolumeClaims**, and then click **Create PersistentVolumeClaim**. In the Create PersistantVolumeClaim menu, populate the YAML file with the required information and click **Create**.



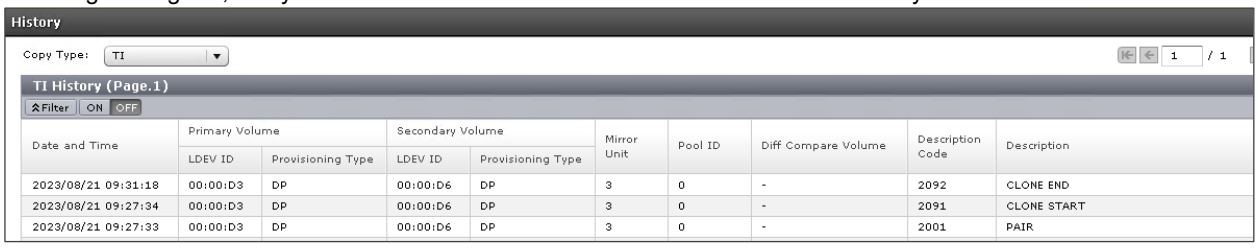
h. Verify whether the clone PVC is created.



i. Creating the PVC also creates a persistent volume. Verify the YAML of this persistent volume. In this scenario, the volume ID is 00:D6.



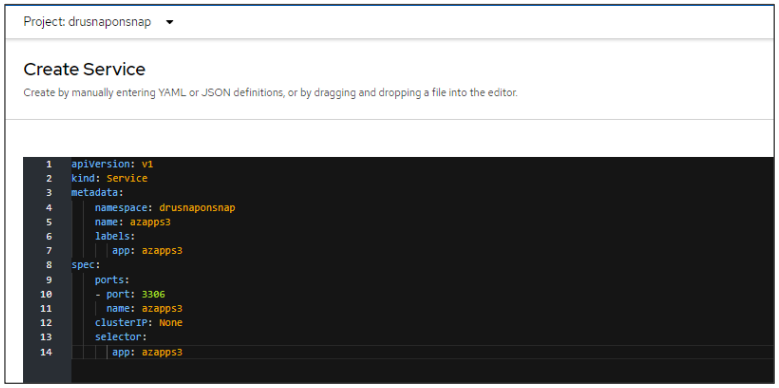
j. In Storage Navigator, verify whether the clone volume 00:00:D6 is created successfully.



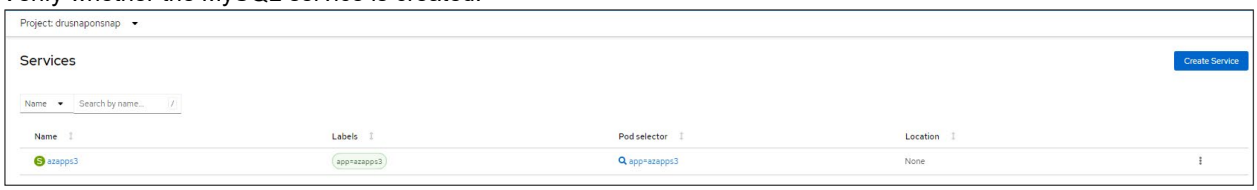
Date and Time	Primary Volume		Secondary Volume		Mirror Unit	Pool ID	Diff Compare Volume	Description Code	Description
	LDEV ID	Provisioning Type	LDEV ID	Provisioning Type					
2023/08/21 09:31:18	00:00:D3	DP	00:00:D6	DP	3	0	-	2092	CLONE END
2023/08/21 09:27:34	00:00:D3	DP	00:00:D6	DP	3	0	-	2091	CLONE START
2023/08/21 09:27:33	00:00:D3	DP	00:00:D6	DP	3	0	-	2001	PAIR

5. Restore the MySQL application in AWS using the clone PVC.

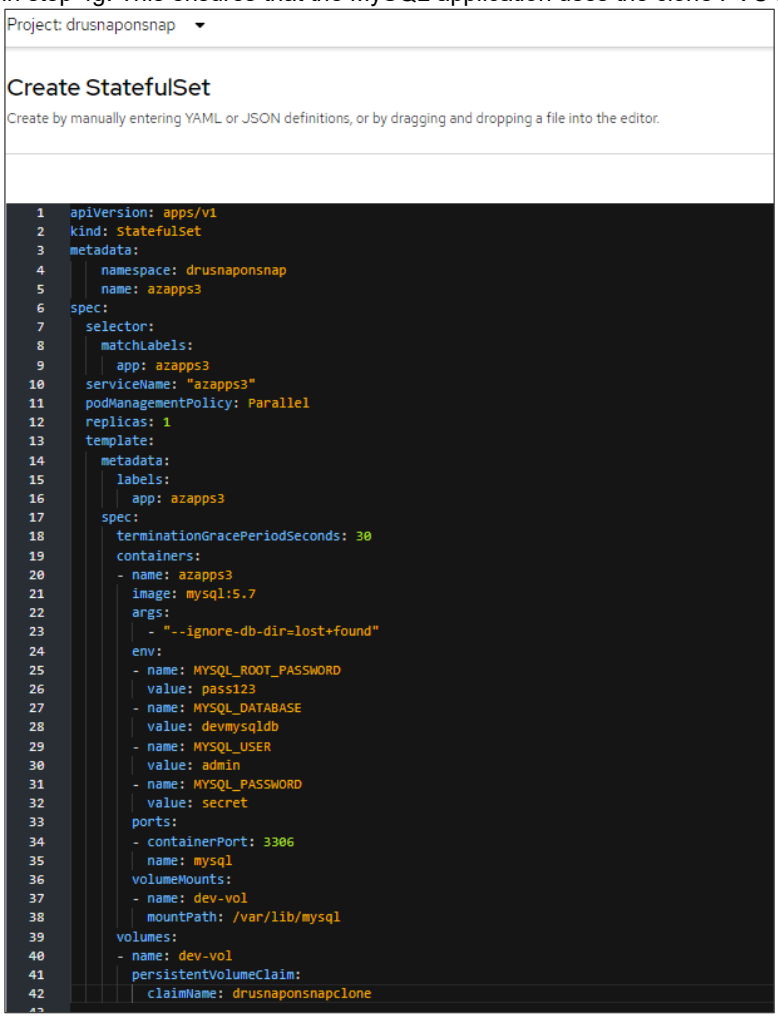
a. Create a MySQL service. From the Red Hat OpenShift console, navigate to **Networking**, click **Services**, and then click **Create Service**. In the Create Service menu, populate the YAML file with the required information and click **Create**.



b. Verify whether the MySQL service is created.

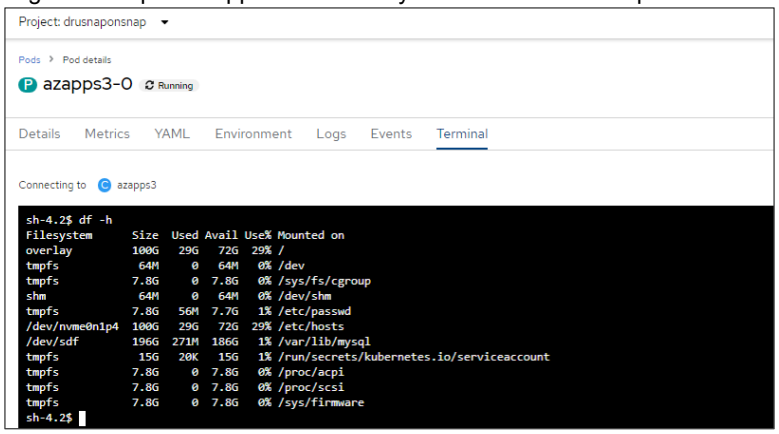


c. Create a MySQL statefulset application. From the Red Hat OpenShift console, navigate to **Workloads**, click **StatefulSets**, and then click **Create StatefulSet**. In the Create StatefulSet menu, populate the YAML file with the required information and click **Create**. In the volume section, use the claim “drusnaponsnapclone” which you created in step 4g. This ensures that the MySQL application uses the clone PVC for persistent data.

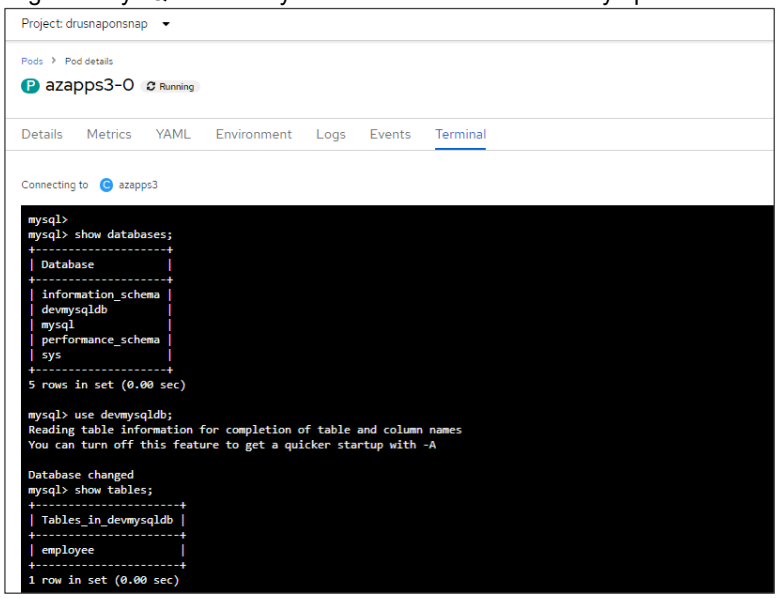


d. Verify whether the statefulset is running.

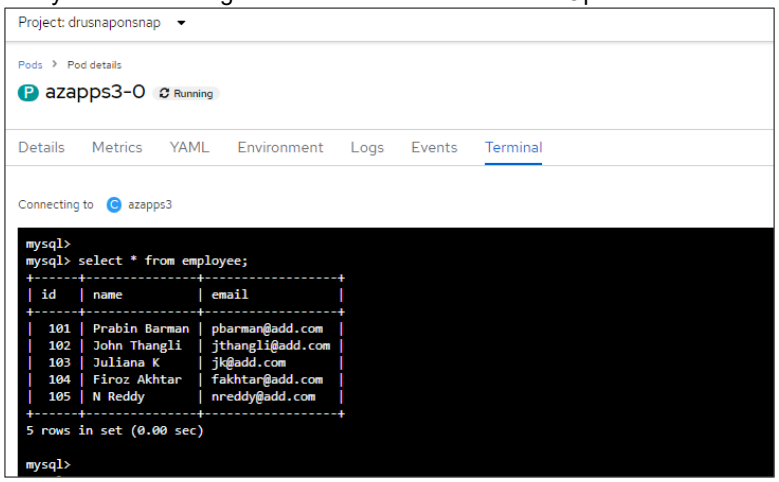
e. Log in to the pod azapps3-0 and verify whether the 200 GB persistent volume is mounted on “/var/lib/mysql”.



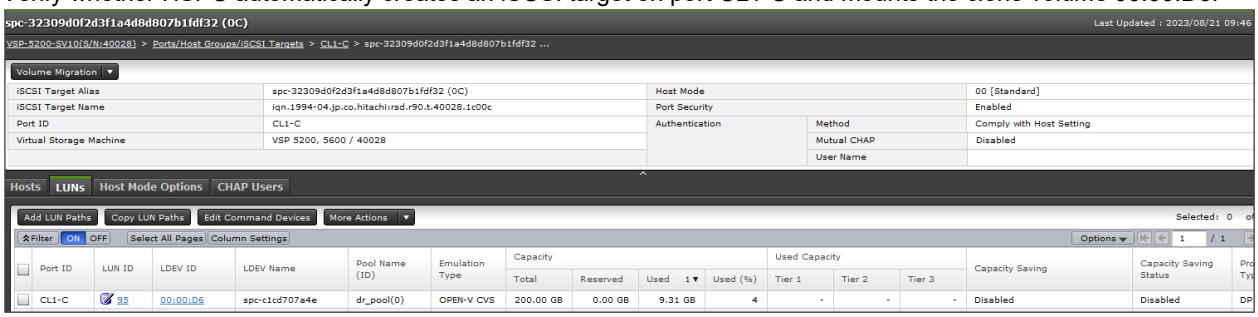
f. Log in to MySQL and verify whether the database “devmysqldb” is available.



g. Verify whether the ingested data from Azure RedHat OpenShift cluster is available here.



h. Verify whether HSPC automatically creates an iSCSI target on port CL1-C and mounts the clone volume 00:00:D6.



i. Delete the snap-on-snap PV (drusnaponsnappv) and PVC (drusnaponsnappvc) created in step 4c and step 4e.