DELL Technologies

Technical Whitepaper

Dell EMC PowerProtect Data Manager protecting Kubernetes Workloads

Abstract

Kubernetes is an open-source platform to manage and orchestrate containerized workloads and services. This document describes the architecture and integration of Kubernetes workloads with Dell EMC PowerProtect Data Manager and how Kubernetes workloads are protected

May 2021

H18563.3

Revisions

Date	Description
October 2020	Initial release
January 2021	Dell EMC PowerProtect Data Manager 19.6 Updates • PostgreSQL High Availability • Cassandra Application Consistent Protection • Restore Cluster Scoped Resources
February 2021	Dell EMC PowerProtect Data Manager 19.7 Updates • VMware Tanzu Kubernetes Cluster protection
May 2021	Dell EMC PowerProtect Data Manager 19.8 Updates • Storage Class Mapping

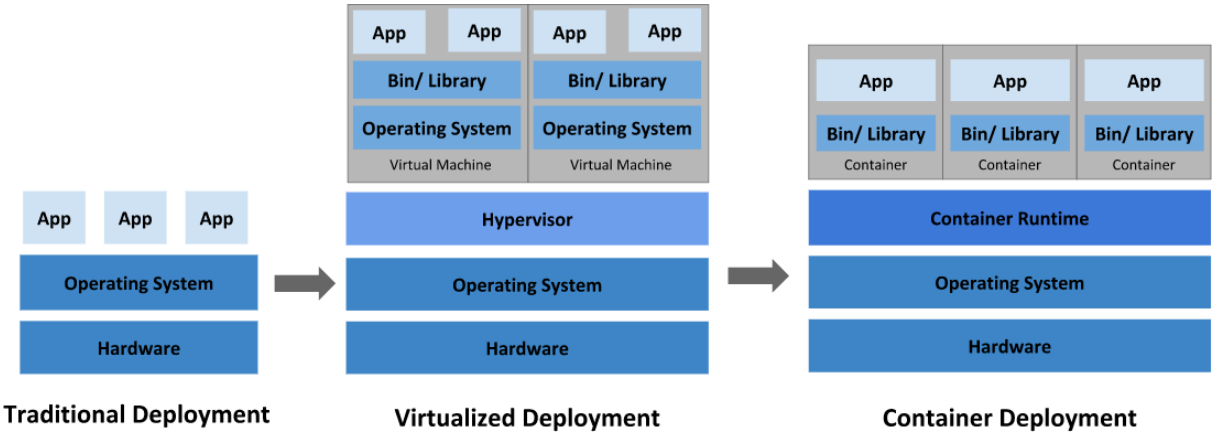
Acknowledgements

Author: Abhishek Shukla, Sr. Engineering Technologist, Data Protection Domain

Table of contents

Executive summary

Traditionally, Organizations used physical servers to run applications on. There was no way to define resource boundaries for applications in a physical server, and this caused resource allocation issues. As a solution, Virtualization was introduced. It allowed to run multiple Virtual Machines (VMs) on a single physical server's CPU. It also allowed applications to be isolated within VMs and provided a level of security.



Modern infrastructure is being transformed by Containers. Containers are similar to virtual machines but have relaxed isolation properties to share the operating system. The Container has its own filesystem, CPU, memory and process space. Agile application creation, continuous development, environmental consistency across development, application-centric management, efficient resource allocation and resource isolation are the key benefits of containers. Kubernetes is an open-source container management platform that unifies a cluster of machines into a single pool of compute resources.

With currently distributed container deployment, it is important to protect the workloads. Dell EMC PowerProtect Data Manager protects the Kubernetes workloads and ensures high availability, consistent, and reliable backup and restore for Kubernetes workload or DR situation. PowerProtect Data Manager offers centralized management, automation, multi-cloud options and advanced integration for ease and simplicity for managing workloads.

Audience

This white paper is intended for customers, partners, and others who want to understand how PowerProtect Data Manager Software helps protect Kubernetes workloads

Scope

1. Dell EMC PowerProtect Data Manager version 19.8
2. Kubernetes (version 1.6 and above)

1 Introduction

The Cloud Native definition is an architectural philosophy for designing the applications and infrastructure 'Containers' provide a way to package and run the application. To run such applications, container orchestrator is required. Kubernetes is an open-source container orchestrator for managing containerized workloads and services, that facilitate both declarative configuration and automation. It is portable, extensible, and scalable and has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

Dell EMC PowerProtect Data Manager protects existing as well as new discovered workloads. It allows IT operations and backup admins to manage Kubernetes clusters and its protection through a single management UI and define protection policies for Kubernetes workloads from Kubernetes APIs. The policy driven protection is defined by the Protection Policy mechanism. PowerProtect Data Manager discovers the namespaces, labels, and pods in the environment and can be protected by providing cluster credentials Logging, Monitoring, governance, and recovery are done through PowerProtect Data Manager

1.1 Features of Kubernetes

Kubernetes is an open-source system for automating deployment, scaling and management of containerized applications

- Kubernetes automates Linux container operations and eliminates many of the manual processes involved in deploying and scaling containerized applications
- Applications can be clustered together in group of hosts running Linux containers, and Kubernetes helps you easily and efficiently manage those clusters.
- Kubernetes is an ideal platform for hosting cloud-native applications that require rapid scaling

1.2 Capability of Dell EMC PowerProtect Data Manager for Kubernetes

PowerProtect Data Manager provides enterprise-level protection for Kubernetes

1.2.1 Efficient and Flexible

- Single Platform for Data Protection – PowerProtect Data Manager manages different workloads that is VMs, applications, and the containers through one platform.
- Protection to deduped storage allows great TCO with PowerProtect DD series.

1.2.2 Built for Kubernetes

- PowerProtect Data Manager allows flexible protection for Kubernetes clusters using the Kubernetes APIs.
- PowerProtect Data Manager discovers, monitors, and protects Kubernetes resources – namespaces, labels, pods, persistent volumes
- No need to install a backup client container for each pod for backup process
- Provides protection to controllers per node to avoid cross-node traffic.
- Application Consistency for MySQL and MongoDB databases
- Restore assets to another cluster that is connected to PowerProtect Data Manager
- Protection for AWS hosted Kubernetes clusters using PowerProtect Data Manager running on AWS and protected to PowerProtect Data Domain running on AWS

1.3 Key components of PowerProtect Data Manager

1.3.1 Cloud Native Data Manager (CNDM)

The Cloud Native Data Manager (CNDM) is in-built microservice component of PowerProtect Data Manager which communicates with the kube-apiserver of the cluster. This component is responsible for APIs for the backup and restore process.

1.3.2 PowerProtect controller

PowerProtect controller is the component which gets installed on Kubernetes cluster when the cluster gets discovered by PowerProtect Data Manager. The backup and restore controllers that manager BackupJob CR and RestoreJob CR definitions. This component is responsible for the backup and restore of Persistent Volumes.

1.3.3 VMware Velero

VMware Velero is the open-source tool which is integrated with PowerProtect Data Manager. It is in-built and does not require to be installed separately. Velero component is pushed into the Kubernetes cluster by the PowerProtect controller pod after the same is in up and running state via velero deployment object. It is responsible for the backup and restore of metadata.

1.3.4 cProxy (Containerized Proxy)

The cProxy is stateless containerized proxy which gets installed on the Kubernetes cluster when the backup and restore process initiated and gets deleted once the process is completed. It is responsible for managing. Persistent Volume snapshots (snap copies), mounting snapshots and moving the data to the target storage. It is also responsible for restoring data into Persistent Volume from target storage and making the data available for attaching to Pods. Also, agent plugin orchestrator for application aware backups.

1.4 Key Components of Kubernetes

1.4.1 Cluster

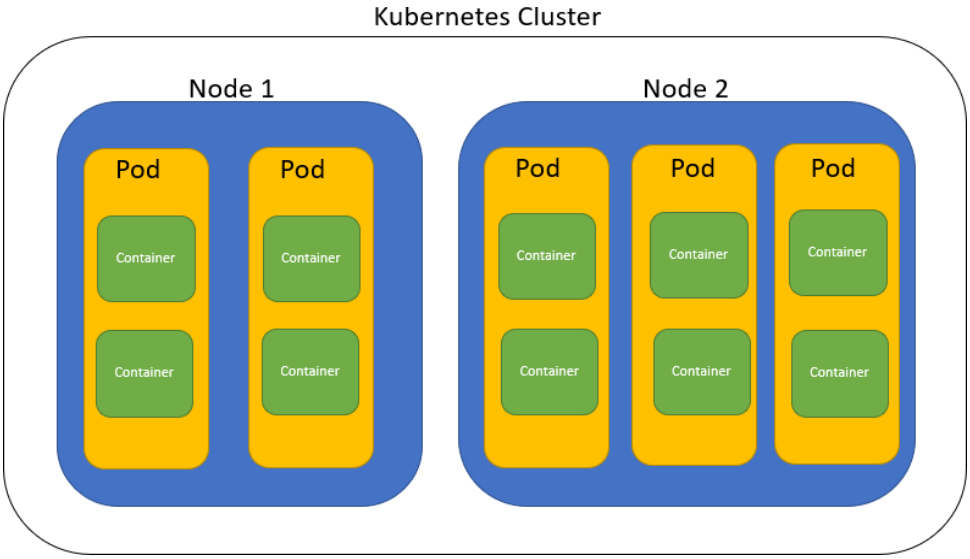
A Kubernetes Cluster is defined as group of machines called nodes that run containerized applications and has a desired state that defines which applications or workloads should be running. The cluster's desired state is defined with the Kubernetes API.

1.4.2 Node

A Node is defined for virtual or physical machine, depending on the cluster. Each node contains the services necessary to run pods and is managed by the master components. There are two kinds of Nodes: Master Node and Worker Node

1.4.3 Pods and containers

Pods operate at one level higher than individual container. Multiple containers can be encapsulated within a Pod. A Kubernetes pod is defined as a group of containers that are deployed together on the same host. The Pod is sometimes called as container when a single container is frequently deployed.



1.4.4 Kubernetes API (kube-apiserver)

The Kubernetes API server is control plane of the Kubernetes cluster that exposes the Kubernetes API. It serves as the foundation for the declarative configuration schema for the system. The `kubectl` command-line tool can be used to create, update, delete, and get API objects.

1.4.5 Persistent Volume (PV) and Persistent Volume Claim (PVC)

Persistent Volume is a storage defined for the cluster that is provisioned by an administrator or dynamically provisioned using Storage Classes (SCs). It is a resource in the cluster similar to a node. PVs are volume plugins like Volumes but have a lifecycle independent of any individual Pod that uses the PV. It captures the details of the implementation of the storage that is NFS, iSCSI, or a cloud-provider-specific storage system

A Persistent Volume Claim (PVC) is a request for storage by a user. It is like a Pod. Pods consume node resources. Similarly, PVCs consume PV resources. Pods can request specific levels of resources (CPU and Memory).

1.4.6 Container Storage Interface (CSI)

Container Storage Interface (CSI) defines a standard interface for container orchestration systems to expose arbitrary storage systems to respective container workloads. A CSI compatible volume driver is deployed on a Kubernetes cluster so that users can use the CSI volume type to attach, mount, etc. the volumes exposed by the CSI driver.

1.4.7 Storage Class

A Storage Class is described as the type of storage that is provisioned and allowed ranges for size and IOPS. When user creates a PVC, that specifies the storage class with size in GB and number of IOPS. A storage class is used to abstract the underlying storage platform.

1.4.8 Namespaces

Namespace is defined as Kubernetes object which partitions a single Kubernetes cluster into multiple virtual clusters. The Namespaces are intended for the use in environment with many users spread across multiple

teams or projects.

1.4.9 Custom Resources (CR)

A resource in Kubernetes environment is an endpoint for API that stores a collection of API objects of a certain kind and A Custom Resource (CR) is an extension of the Kubernetes API that is not necessarily available in a default Kubernetes installation. It represents a customization of a particular Kubernetes installation

2 Deployment Methods of Kubernetes

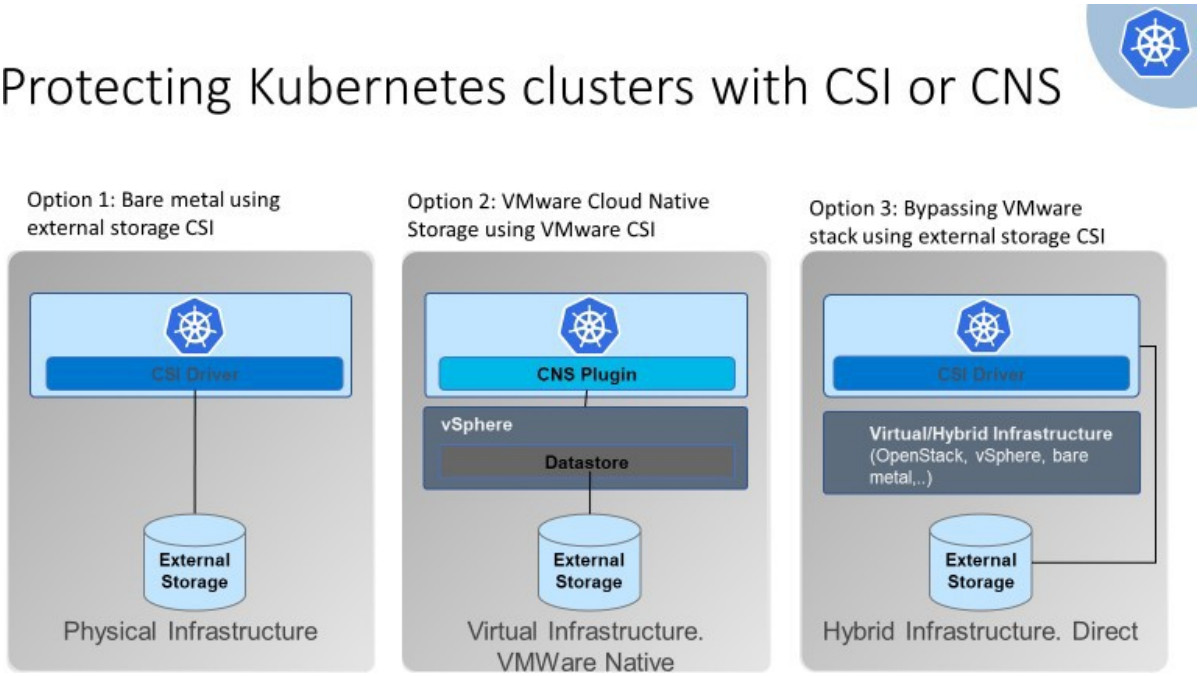
Kubernetes is an open-source container orchestrator for automating deployment, scaling, and managing containerized workloads. There are few methods to deploy Kubernetes clusters and protected accordingly. Kubernetes can be deployed on-premises or on the cloud

2.1 Kubernetes On-Premises

Kubernetes is described as a cloud-native technology. However, the cloud-native concept includes the use of on-premises infrastructure.

2.1.1 Kubernetes running on virtual environment

The virtual environments are mostly VMware based. With this method of deployment, the persistent volumes are carved with VMware vSphere managed storage (VMware vSAN or vVOLs) as First-Class Disk. A First-Class Disk (FCD) refers to Improved Virtual Disk (IVD) is one of the recent features of VMware vSphere. FCD is the independent disk which is not associated with VM. When Persistent Volumes for cloud native application is created, a virtual disk (VMDK) is attached to Kubernetes node. The disk backup of FCD is similar to that of



other VMDKs. The container orchestrates volume snapshot backup with which the volume snapshot is taken, mounted, and streamed block data to target storage. For application level backups, pre and post-hooks are used which quiesce the database, flush, and take snapshots of Persistent Volumes.

2.1.2 On-premises Kubernetes on bare metal

Bare-metal servers, such as Linux servers, run containers and Kubernetes nodes on physical servers. In this method of deployment, the Persistent Volumes are carved out from underlying storage that is SAN or NAS and connected through the storage vendor provided custom drivers or through standardized CSI drivers. The container orchestrates volume snapshot backup with which the volume snapshot is taken, mounted, and streamed block data to target storage. For application level backups, the database dump backup is taken online and streamed data to target storage.

2.1.3 Using External CSI

Container Storage Interface (CSI) us an open standard API that enables Kubernetes to expose arbitrary storage systems to containerized workloads. In this method of deployment, external CSI drivers are used to access infrastructure and external storage such as OpenStack

2.2 Kubernetes on Cloud

There are few clouds which are supported for Kubernetes. Database backups are taken online and streamed data to cloud bucket. The container orchestrates cloud snapshots. Kubernetes deployment in the cloud that is Amazon Web Services (AWS), Microsoft Azure or Google Cloud Platform (GCP) have two variants.

2.2.1 Kubernetes deployed on Infrastructure as a Service (IaaS)

In this scenario, the Linux hosts running containers are deployed on IaaS plane, e.g., AWS EC2 instance which means the underlying infrastructure is provided and similar to running on virtualized environment. In such case, user has to deploy the Kubernetes clusters either upstream or distribution on the IaaS instance of the cloud provider.

2.2.2 Kubernetes as a Service (KaaS)

Various cloud providers offer Kubernetes as a service. In such case, the cloud provider manages the Kubernetes environment completely and offers it as a service to the user. Below is the list of cloud providers and the Kubernetes as a service:

- Kubernetes on Google Cloud: Google Kubernetes Engine (GKE)
- Kubernetes on Amazon Web Services (AWS): Elastic Kubernetes Service (EKS)
- Kubernetes on Microsoft: Azure Kubernetes Service (AKS)
- Kubernetes on Rancher: Rancher Kubernetes Engine (RKE)

Additionally, Kubernetes environment can be plain vanilla upstream that is open-source version of Kubernetes, Distro (Kubernetes distribution) that is Rancher and CoreOS that automates the provisioning of Kubernetes cluster using installer script and OEM distribution that automates Kubernetes that is Red Hat OpenShift, Anthos, or Pivotal Container Service (PKS).

3 Reference Architecture

PowerProtect Data Manager protects Kubernetes workloads and ensures the data is consistent and highly available. PowerProtect Data Manager is a virtual appliance that is deployed on an ESXi host using OVA and is integrated with PowerProtect DD series (virtual edition) as protection target where backups are stored.

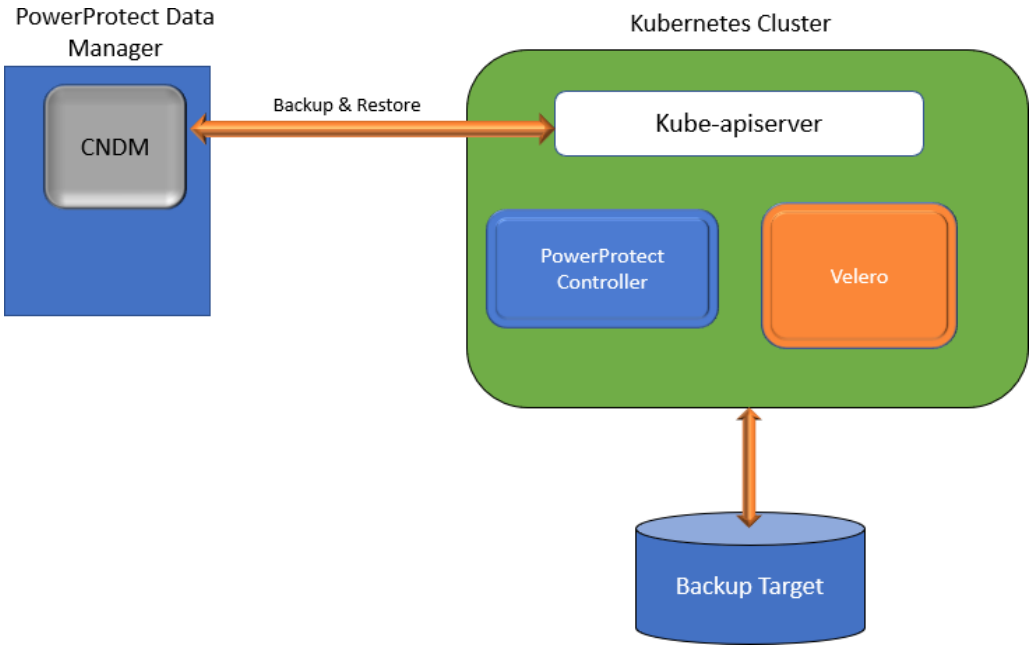
Kubernetes cluster is group of a master and worker nodes. Each worker node is managed by master(s). It handles scheduling the pods across the nodes in the cluster. PowerProtect Data Manager integrates with Kubernetes cluster through Kubernetes APIs to perform the discovery. The CNDM is the component of PowerProtect Data Manager which communicates with the kube-apiserver of the cluster.

Once the discovery of cluster completes, the cluster is added as PowerProtect Data Manager asset source and associated namespaces as assets are available to be protected. During the process of the discovery, PowerProtect Data Manager creates below two namespaces in the cluster. The data is compressed and deduplicated at the source and sent to the target storage.

1. **Velero-ppdm**: Contains a Velero pod to backup metadata and stage to the target storage in case of a BareMetal environment. It performs PVC and metadata backup in case of VMware Cloud Native Storage (CNS)
2. **PowerProtect**: Contains a PowerProtect controller pod to drive Persistent Volume Claim snapshot and backup and push the backups to target storage using intermittently spawned cProxy pods

3.1 Protection of Kubernetes clusters

PowerProtect Data Manager discovers the Kubernetes clusters using the IP address or FQDN. PowerProtect Data Manager uses the discovery service account and the token (kubeconfig file) to integrate with kube-apiserver. PowerProtect Data Manager protects two types of assets of Kubernetes cluster that is **Namespaces** and **PersistentVolumeClaims (PVCs)**. PowerProtect Controller is responsible for the backup and restore of PVCs and VMware Velero component is responsible for backup and restore of metadata and saved in the target storage.

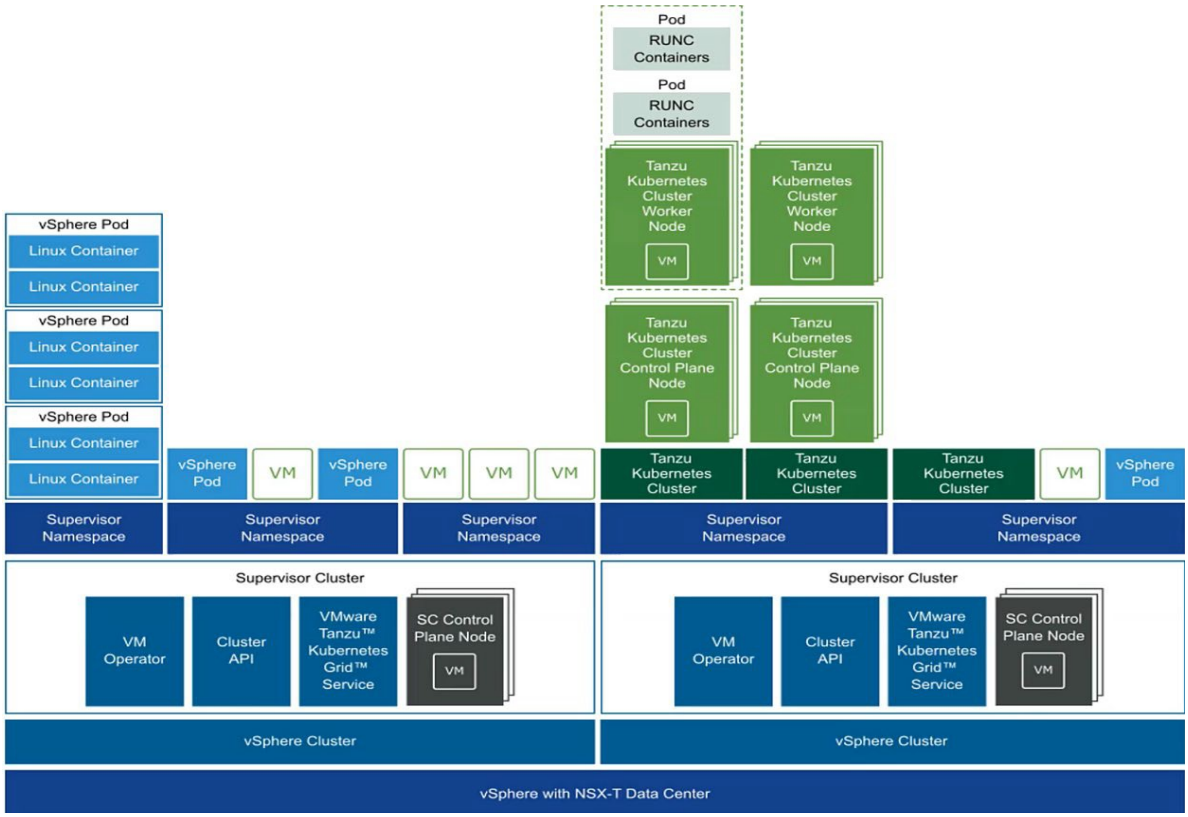


A Protection policy enables selecting a specific group of assets to be backed up. Protection Policy is created using the PowerProtect Data Manager UI for the backup schedule and retention lock. When a protection policy is created, a new storage unit (SU) is created on the protection storage as part of protection policy configuration.

In case of Kubernetes workloads, a BackupStorageLocation containing the SU information is also created on the cluster. PowerProtect Controller running in the Kubernetes cluster creates a corresponding BackupStorageLocation in the Velero namespace whenever a BackupStorageLocation is created in the PowerProtect namespace.

3.2 Protection of VMware Tanzu Kubernetes Grid (TKG)

PowerProtect Data Manager 19.7 onwards introduces ability to protect Kubernetes workloads on VMware TKG. VMware vSphere 7U1 re-architectures vSphere with native Kubernetes as its control plane. A TKG cluster is a Kubernetes cluster that runs inside the virtual machines on supervisor layer which allows to run Kubernetes with consistency. It is enabled via the TKG service for VMware vSphere and is upstream-complaint with open- source Kubernetes (Guest cluster). The Guest cluster is consistent Kubernetes cluster running on VMs and consists of its control plane VM, management plane VM, worker nodes, pods and containers.



VMware Tanzu Kubernetes Grid (Project Pacific) Architecture

For detailed whitepaper on Dell EMC PowerProtect Data Manager protecting VMware Tanzu Kubernetes Clusters: Click on this link: https://www.delltechnologies.com/resources/en-us/asset/white-papers/solutions/h18682-dell-emc-powerprotect-data-manager-protecting-vmware-tanzu-kubernetes-cluster-wp.pdf

4 Configuring PowerProtect Data Manager protecting Kubernetes Workloads

This section explains the discovery of Kubernetes cluster with PowerProtect Data Manager, Backup and Restore Configuration

4.1 Asset Discovery

The Kubernetes cluster gets registered with PowerProtect Data Manager. Following steps to be performed for discovery of assets and asset sources.

1. Log in to **PowerProtect Data Manager UI** with administrator credentials.
2. On the left pane of the PowerProtect Data Manager UI, click **Infrastructure**.
3. Select **Asset Sources,** and at the top select **Kubernetes**
4. Click **Add**
   - **Name**: Specify the name
   - **FQDN/IP**: Specify the IP address or fully qualified domain name
   - **Port:** this information is taken from `kubectl cluster-info` output from master node
   - **Host Credentials:**
     - Click on **Add**
     - Name:
     - Service Account Token:
       Run below command on master node of the Kubernetes cluster and copy the output and paste it.

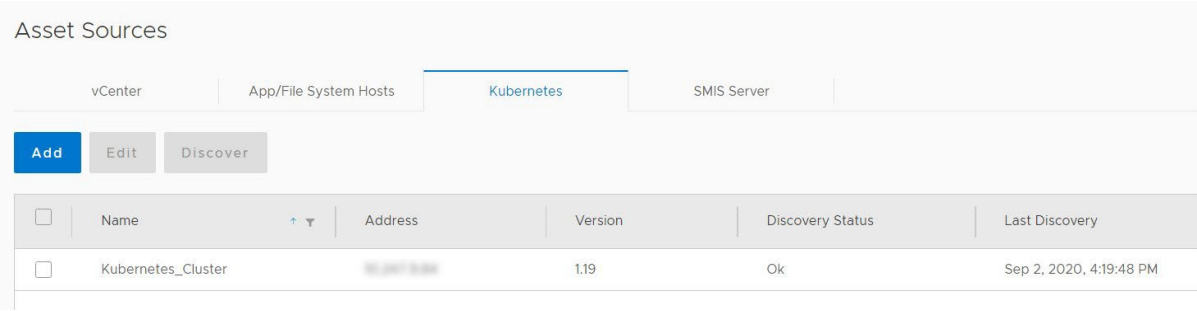
```
kubectl get secret $(kubectl get serviceaccount dashboard -o jsonpath="{.secrets[0].name}") -o jsonpath="{.data.token}" | base64 –decode
```

     - Click on **Save**
   - Click on **VERIFY** to verify the certificate

Note: Before verifying the certificate, make sure the ports on firewall are open at PowerProtect Data Manager. If not, run below command

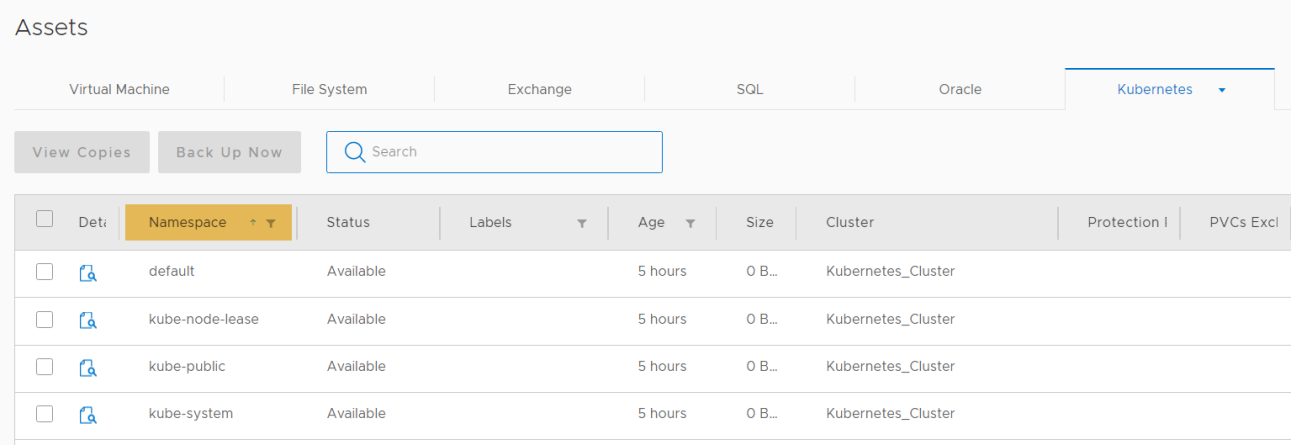
```
sudo iptables -I OUTPUT -p tcp --dport 6443 -j ACCEPT
```

   - Once verified, Click on **SAVE**



Once the discovery is completed, the Kubernetes cluster is available

5. Click **Asset** to view namespaces



Hence the discovery process is completed

4.2 Backup configuration

Backup of an asset can be taken manually as well as can be scheduled, Protection policy must be created to run the backup.

4.2.1 General Configuration

1. Log in to **PowerProtect Data Manager UI** with administrator credentials.
2. On the left pane of the PowerProtect Data Manager UI, click on **Protection** dropdown
3. Click on **Protection Policies**
4. Click on **Add** to add new policy
   a. **Type**: Specify the necessary details as
      - Name: Specify the name of the policy
      - Description: Provide any description if any
      - Type: Select **Kubernetes**
      - Click **Next**
   b. **Purpose:**
      Select the option from below according to the purpose of the backup,
      - **Crash Consistent**: Select this option to snapshot persistent volumes bound to the persistent volume claims in the namespace and back them up to the storage target **Or**
      - **Exclusion**: Select this option to exclude in this group from protection activities and protection rule assistant.

      Note: Application consistency for MYSQL and MongoDB databases are application template driven (discussed later in this section)

      - Click **Next**
   c. **Assets**
      - Select the Asset (s) to be backed up from the list or particular asset can be searched by typing in **Find More Assets** box
      - Click **Next**
   d. **Objectives**
      In this section, Backup configuration is created, Replicated, Edited and Deleted
      - Click on **Add** to add Primary Backup and fill the required details for target and schedules
        i. **Storage Name:** Select the storage from dropdown, default storage name is appeared, but you can add additional storage by clicking on **Add** and specify the storage target details

ii. **Storage Unit:** Select the specific storage unit from the dropdown list
iii. **Network Interface:** Select the network from dropdown list
iv. **Retention Lock:** Toggle On/Off button if you need to specify the retention lock
v. **SLA:** Click on dropdown to select **SLA** or to add SLA, click on **Add**
- Click on **Add** and fill the Name and select the objective (s)
- **SLA name**: Specify the name for SLA
- Check box to specify **Recovery Point Option** (minutes, hours, days, weeks, months, or years)
- Check box if any **Compliance Window** is to be mentioned
- Check box to **Verify expired copies are deleted**
- Check box to specify **Retention Time Objective** (days, weeks, months or years)
- Check box to **Verify Retention Lock is enabled for all copies**
- Click on **Save**

- **Schedules:** Click on **Add Backup** for the schedule
  - Create a backup every as Hourly, Daily, Weekly or Monthly and fill the details accordingly such as retain for, start and end time
  - Click on **Save**
- The schedule is ready, Click on **Next**

5. Verify the provided information is correct under **Summary** section. If yes, click on **Finish**
6. Click on Go to Jobs to check the status of the policy

The Protection policy triggers the backup at the scheduled time. When the protection policy is created successfully. There are options to modify the existing policy i.e. **Edit, Disable, Export** and **Back up Now**

1. **Edit**: To edit the information or to change the schedule
2. **Disable**: Backup Schedule is disabled with this option so backup would not be taken
3. **Export:** Downloadable file which contains the information about the asset protection
4. **Protect Now** This Option allows to take backup manually at ad-hoc basis.
   a. **Asset Selection**: It has further two options to select the assets i.e.
      - **All assets defined in the protection policy**
      - **Choose some of the assets defined in the policy:** This option allows to select namespaces within the cluster
   b. **Configuration:**
      - This allows to **select type of backup**

        Full: Backs up the namespace metadata and persistent volumes and creates a new full backup

        Synthetic full: Backs up namespace metadata, change blocks for persistent volumes on VMware first class disks, all other persistent volumes and creates a new full backup
      - Keep For: In days
      - Click **Next**
   c. **Summary:** Verify the information
   d. Click on **Backup**
   e. Monitor the backup job by Clicking **Go to Jobs**

4.2.2 Kubernetes MySQL Application Consistent protection

PowerProtect Data Manager has Application Consistent protection feature for MySQL and MongoDB databases. To utilize this feature, user creates a MySQL Application consistent backup using templates. PowerProtect Data Manager has in-built '**ppdmctl.tar.gz'** file available which contains the templates for MySQL and MongoDB in YAML format. This file is pushed from PowerProtect Data Manager to Kubernetes cluster root directory to enable the application consistent protection. The protection is provided only to the namespace which has asset defined in the protection policy. When the protection policy is applied with enabled MySQL template, the consistency of the backed-up data is application consistent and when the MySQL template is disabled, the consistency is crash consistent.

Kubernetes namespaces are already discovered as assets with PowerProtect Data Manager. There are the available namespaces for protection except powerprotect and velero-ppdm namespaces. To configure the application consistent protection, particular namespace is specified. In this case, test-namespace is used to demonstrate.

1. Log in to **PowerProtect Data Manager CLI** (command line interface) with administrator credentials
2. Change directory to **cndm/misc** using below command
   `cd /usr/local/brs/lib/cndm/misc`
3. Run list command to view the content which is ppdmctl.tar.gz file
   `ls -ltrh`
4. Run below command to transfer file from PowerProtect Data Manager to Kubernetes cluster (K8s-master01 is for reference for Kubernetes cluster
   `scp ppdmctl.tar.gz tme@k8s-master01`
5. Login to Kubernetes Cluster and run list command (`ls`) to list the content of the root directory and confirm 'ppdmctl.tar.gz' is available
6. Run tar command to untar the file. Once untar is completed, **ppdmctl** directory is created
   `tar -xvf ppdmctl.tar.gz`
7. Run change directory command to enter ppdmctl directory and run `ls` to list the content
   `cd ppdmctl`
   `ls`
8. Run to enter the examples directory where all the MySQL and MongoDB templates are stored
   `cd examples`
   `ls`
9. Copy mysqlapptemplatehelm.yaml file to create a template copy from the example for MySQL with command (`tmedemomysqlapptemplatestadalone.yaml` is existing template associated to the test- namespace)
   `cp mysqlapptemplatehelm.yaml tmedemomysqlapptemplatestadalone.yaml`
10. Run below command to describe the MySQL stateful set
    `kubectl describe sts mysql -n test-namespace`
11. Edit the `tmedemomysqlapptemplatestadalone.yaml` file to make the changes
    - To change the name of master pod as per the pod description, edit `selectorExpression: "mysql"` (mysql is new name)
    - If there are no slave pods are being used for MySQL, remover the `selectorTerms` section under `selectors` for slave
    - To edit the MySQL password environment variable, preHook and postHook are edited. (pod based hooks execute hook code in a new pod derived from template in a deployment configuration, preHook is to quiesce and postHook is to be unquiesce)
12. Run cd.. to come out from examples directory to ppdmctl directory
13. To create a template for MySQL application consistent protection from the edited (step11) template file in

`test-namespace`, use ppdmctl utility by running command

```
./ppdmctl template create tmemysqltemplate -type=mysql -namespace=test-
namespace -inputfile=examples/tmedemomysqlapptemplatestadalone.yaml
```

The application consistent protection is configured for specific namespace (test-namespace in this demonstration). When the backup is run manually or using protection policy, the backup copy is saved as application consistent at the target storage.

14. To disable the MySQL template, run below command

```
./ppdmctl template disable tmemysqltemplate -namespace=test-namespace
```

4.2.3 Kubernetes PostgreSQL Application Consistent Protection

PowerProtect Data Manager supports both PostgreSQL and PostgreSQL HA (High Availability). Both PostgreSQL and PostgreSQL HA configure a cluster with a primary-standby topology. The primary node has writing permission while replication is on the standby nodes which have read-only permissions. PowerProtect Data Manager has in-built '**ppdmctl.tar.gz**' file available which contains the templates for PostgreSQL in YAML format. This file is pushed to the primary node. Pgpool-II acts as proxy for PostgreSQL backend. It reduces connection overhead and used as a load balancer for PostgreSQL. Pgpool-II is responsible to spread the queries among nodes.

Application Template of PostgreSQL

```
AppLabel: "app=postgresql"
Type: "Postgresql"
AppActions:
Pod:
PreHook:  "[\"/bin/bash\",  \"-c\",  \"PGPASSWORD=$POSTGRES_PASSWORD  psql  -  U
   $POSTGRES_USER -c \\\"select pg_start_backup('ppdm-backup', true);\\\"\"]"
PostHook: "[\"/bin/bash\", \"-c\", \"PGPASSWORD=$POSTGRES_PASSWORD psql
    -U     $POSTGRES_USER -c \\\"select pg_stop_backup();\\\"\"]"
```

Application Template of PostgreSQL HA

```
AppLabel:"app.kubernetes.io/name=postgresql-
ha,app.kubernetes.io/component=postgresql"

# applabel to select pod
Type: "Postgresql"
AppActions:
Pod:
PreHook: "[\"/bin/bash\", \"-c\", \"REPMGR_PRIMARY_HOST=`echo
$REPMGR_PRIMARY_HOST | cut -f1 -d '.'`; if [ $HOSTNAME =
$REPMGR_PRIMARY_HOST ]; then PGPASSWORD=$POSTGRES_PASSWORD psql -U
$POSTGRES_USER -c \\\"select     pg_start_backup('ppdm-backup',       true);\\\";
fi\"]"
PostHook: "[\"/bin/bash\", \"-c\", \"REPMGR_PRIMARY_HOST=`echo
      $REPMGR_PRIMARY_HOST | cut -f1 -d '.'`; if [ $HOSTNAME =
```

```
$REPMGR_PRIMARY_HOST ]; then PGPASSWORD=$POSTGRES_PASSWORD psql -U
$POSTGRES_USER -c \\\"select     pg_stop_backup();\\\"; fi\"]"

Application:
Kind: StatefulSet
Selectors:
SelectorTerms: {“field” : “Name”, "selectorExpression": ".*-[1-9][0- 9]*$“ }
# Standby pods with index > 0
SelectorTerms: {“field” : “Name”, "selectorExpression": ".*-0$"} # Primary pods
with index = 0
```

4.2.4 Kubernetes Cassandra App Consistent Protection

Apache Cassandra is highly scalable, high-performance distributed database designed to handle large amounts of data with no single point of failure. It is a type of NoSQL database. Cassandra partitions data over storage nodes using consistent hashing algorithm. Each node stores multiple ranges of tokens and each range of token replicates to multiple nodes for fault-tolerance and high availability. PowerProtect Data Manager has Application Consistent protection feature for Cassandra database. PowerProtect Data Manager has in-built ‘**ppdmctl.tar.gz**’ file available which contains the templates for Cassandra in YAML format.

Application Template of Cassandra

```
AppLabel: “app=cassandra”
Type: “Cassandra”
Enable: true
AppActions:
Pod:
PreHook: "[\"/bin/bash\", \"-c\", \"nodetool flush\"]"
```

4.3 Restore Configuration

The Recovery of the assets is a manual process. The restoration of the Kubernetes namespaces includes pods, stateful sets, PVCs and other resources is done at the same cluster. With PowerProtect Data Manager, there are options to recover the Kubernetes namespaces at the **same** as well as **the alternate cluster**

4.3.1 Restore to alternate cluster

In order to restore assets to alternate cluster, the target cluster must be discovered with PowerProtect Data Manager and storage classes must be defined at source and target cluster. The same storage class must be available with source and target cluster. If the same storage class at source cluster is not available at the target cluster then a **storage-class-configmap** is configured at the target cluster providing the information on which storage class at the target cluster PVCs are bound to. Generally, with new clusters which do not share storage classes, the storage-class- configmap file to bound PVCs.

Configuration at Kubernetes cluster

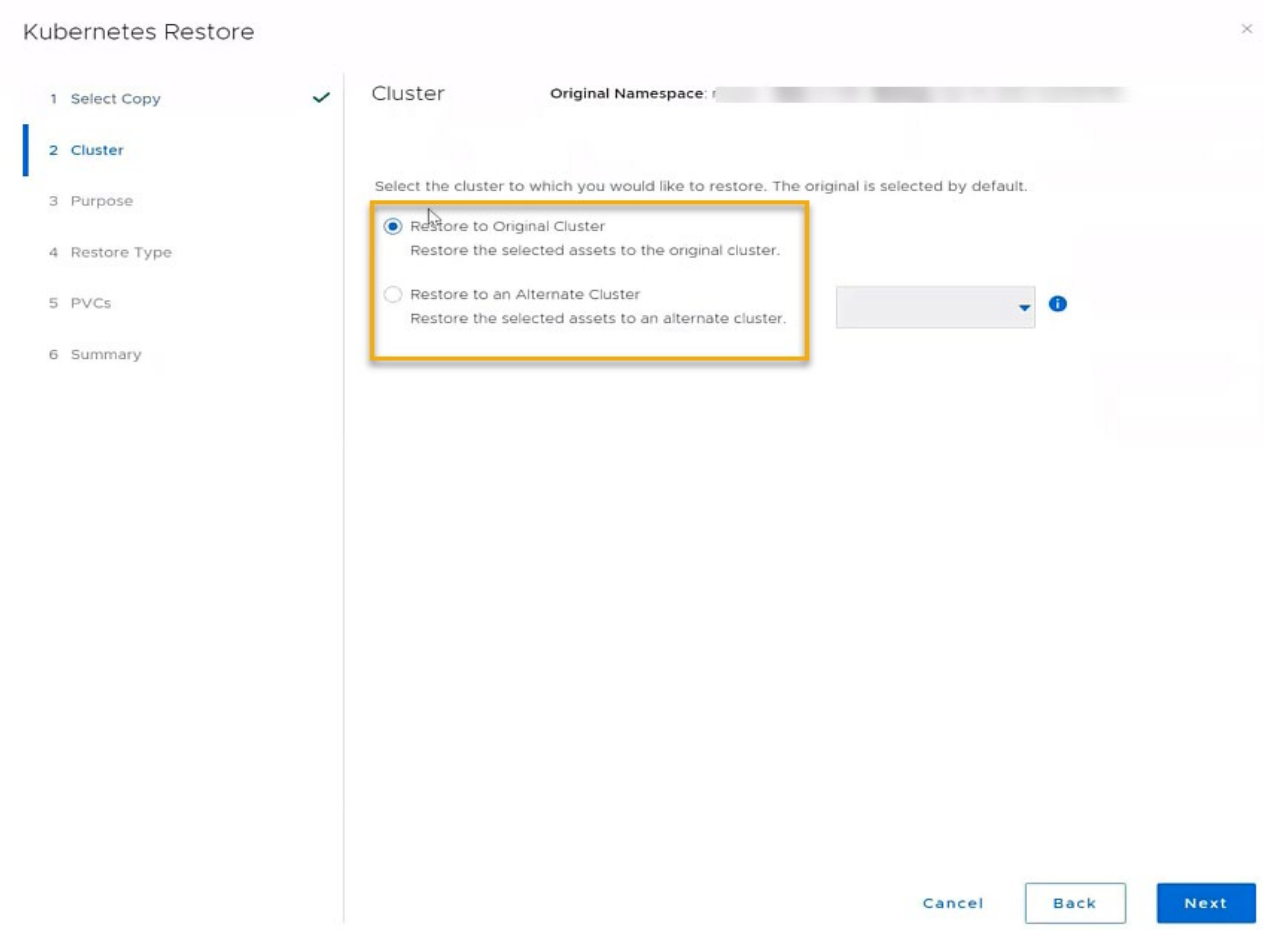
1. Login to **Kubernetes source cluster** (master01) and **Kubernetes target cluster** (master02) and run command to get the storage class information

```
Kubectl get sc
```

2. The storage class must have PVC bound to it, to verify run below command
   `Kubectl get svc,sts, pod,pvc -n test-namespace` (test-namespace is the namespace for reference)
3. Considering that the storage class at source cluster is not available at target cluster, run below command to configure storage-class-configmap at target cluster
   `Kubectl create configmap ppdm-controller-storage-class-mapping -n powerprotect`
   In case, the source and target cluster have same storage classes, then the config map is not required. To verify the config map file is created successfully, run below command at target cluster
   ```
   Kubectl get configmap -n powerprotect
   ```
4. Below command is to edit the storage class mapping file. This file provides the information about the source and target cluster while and after restoring.
   `Kubectl edit configmap ppdm-controller-storage-class-mapping -n powerprotect`
5. Run below command to view the edited configmap file
6. `Kubectl get configmap ppdm-controller-storage-class-mapping -n powerprotect -o yaml`

Above process confirms that the same storage class is available at both source and target cluster. At the time of restore, the assets can be restored at alternate cluster with the defined mapping. Below is the configuration at PowerProtect Data Manager to set up restore process at same as well as the alternate cluster.

1. Log in to **PowerProtect Data Manager** with admin credentials.
2. On the left pane of the PowerProtect Data Manager UI, click on **Recovery**
3. Click on **Assets**
4. Click on **Kubernetes** on top and select the namespaces to be restored
5. Click on **Restore**
   a. Select **Copy**:
      - Select the restore copy. The most recent copy will be used as default. To change from default copy, click on **Change Copy**
      - Click **Ok**
      - Click **Next**
   b. **Cluster**
      This provides the option to select the cluster on which assets to be restored
      - **Restore to Original Cluster**: The Assets are restored to the source cluster from which the backup is taken.
      - **Restore to Alternate Cluster**: The Assets are restored on the alternate cluster. To utilize this option, alternate cluster is added as an asset source to the managing PowerProtect Data Manager
   c. **Click** Next
   d. **Purpose:**
      Select the option what is to be restored
      - **Restore Namespace and Select PVCs:** This option restores the namespace and a subset of PVCs in the namespace. The namespace resources including pods, services, secrets, and deployments will not be overwritten during a restore. All other resources that do not currently exists in the namespace will be restored.
      - **Restore PVCs only:** This Option will restore only PVCs



e. **Restore Type**
Restore type has different options depending on the **purpose** of the restore
- If purpose is to Restore namespaces and PVCs, then options are
  - Restore to Original Namespace,
  - Restore to New Namespace and
  - Restore to an Existing Namespace
- And if purpose if to restore PVCs only, then options are
  - Restore to Original Namespace and
  - Restore to an Existing Namespace

f. **PVCs:**
Select PVCs to be restored to the namespace, options are to
- Overwrite content of existing PVCs
- Skip restores of existing PVCs

g. **Summary**
Verify the information and click on **Restore**

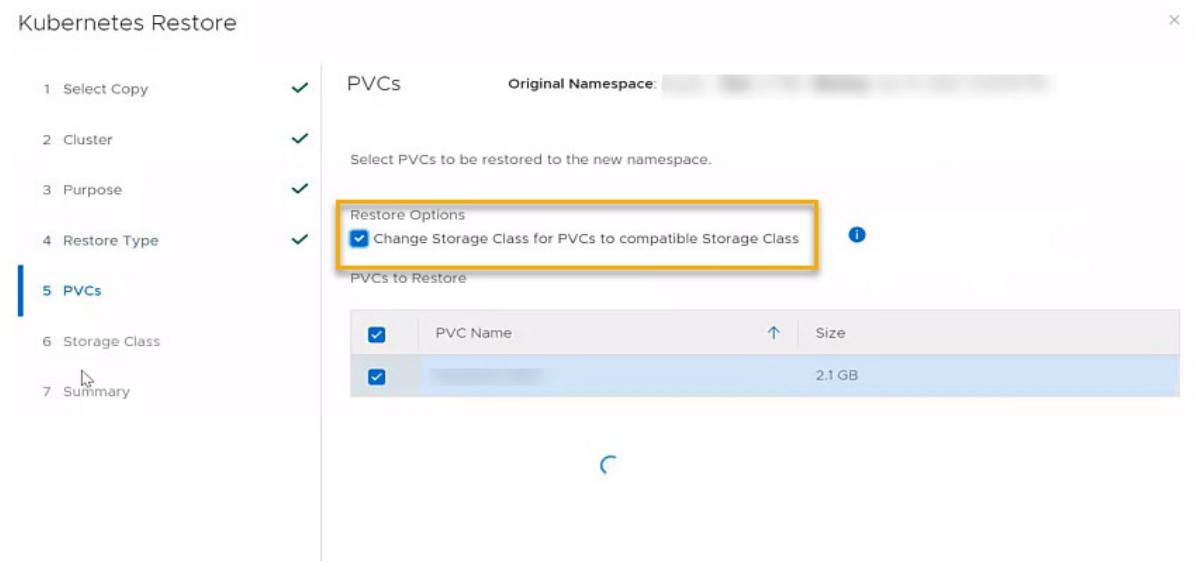
4.3.2 Restore using Storage Class Mapping

The Storage Class Mapping feature with PowerProtect Data Manager 19.8 provides capability to choose alternate storage class for PVCs with certain provisioner type while restore of persistent volumes. Storage class mapping enables restoring namespaces and PVCs from one cluster to another using different container

storage. It is also useful when the migration of data from one storage class to another storage class and from on-premises to cloud or vice-versa

1. Log in to **PowerProtect Data Manager** with admin credentials.
2. On the left pane of the PowerProtect Data Manager UI, click on **Recovery**
3. Click on **Assets**
4. Click on **Kubernetes** on top and select the namespaces to be restored
5. Click on **Restore**
   a. Select **Copy**:
      - Select the restore copy. The most recent copy will be used as default. To change from default copy, click on **Change Copy**
      - Click **Ok**
      - Click **Next**
   b. **Cluster**
      This provides the option to select the cluster on which assets to be restored
      - **Restore to Original Cluster**: The Assets are restored to the source cluster from which the backup is taken.
      - **Restore to Alternate Cluster**: The Assets are restored on the alternate cluster. To utilize this option, alternate cluster is added as an asset source to the managing PowerProtect Data Manager
   c. **Purpose**
      Select the option what is to be restored
      - **Restore Namespace and Select PVCs:** This option restores the namespace and a subset of PVCs in the namespace. The namespace resources including pods, services, secrets, and deployments will not be overwritten during a restore. All other resources that do not currently exists in the namespace will be restored.
      - **Restore PVCs only:** This Option will restore only PVCs
   d. **Restore Type**
      Restore type has different options depending on the **purpose** of the restore
      - If purpose is to Restore namespaces and PVCs, then options are
        - Restore to Original Namespace,
        - Restore to New Namespace and
        - Restore to an Existing Namespace
      - And if purpose if to restore PVCs only, then options are
        - Restore to Original Namespace and
        - Restore to an Existing Namespace
   e. **PVCs:**
      Select PVCs to be restored to the namespace, options are to
      - Overwrite content of existing PVCs
      - Skip restores of existing PVCs
      - Restore Options:
        - Checkbox to change storage class for PVCs to compatible storage class. Note that storage class will not be modified for existing PVCs being overwritten

      Note: Once the checkbox is selected, Storage Class tab is available to select the target storage



f. **Storage Class**
Change storage class for PVC's to compatible ones on the alternate cluster. Note that when changing the PVC storage class on the target Kubernetes cluster. if you select more than one PVC on the page, only the storage classes that apply to all selected PVCs are displayed. To view and select from all available storage classes, select one PVC at a time.
- Click on **Target Storage Class**
- Select one storage class from **Storage Class Name** list and click on **Save**

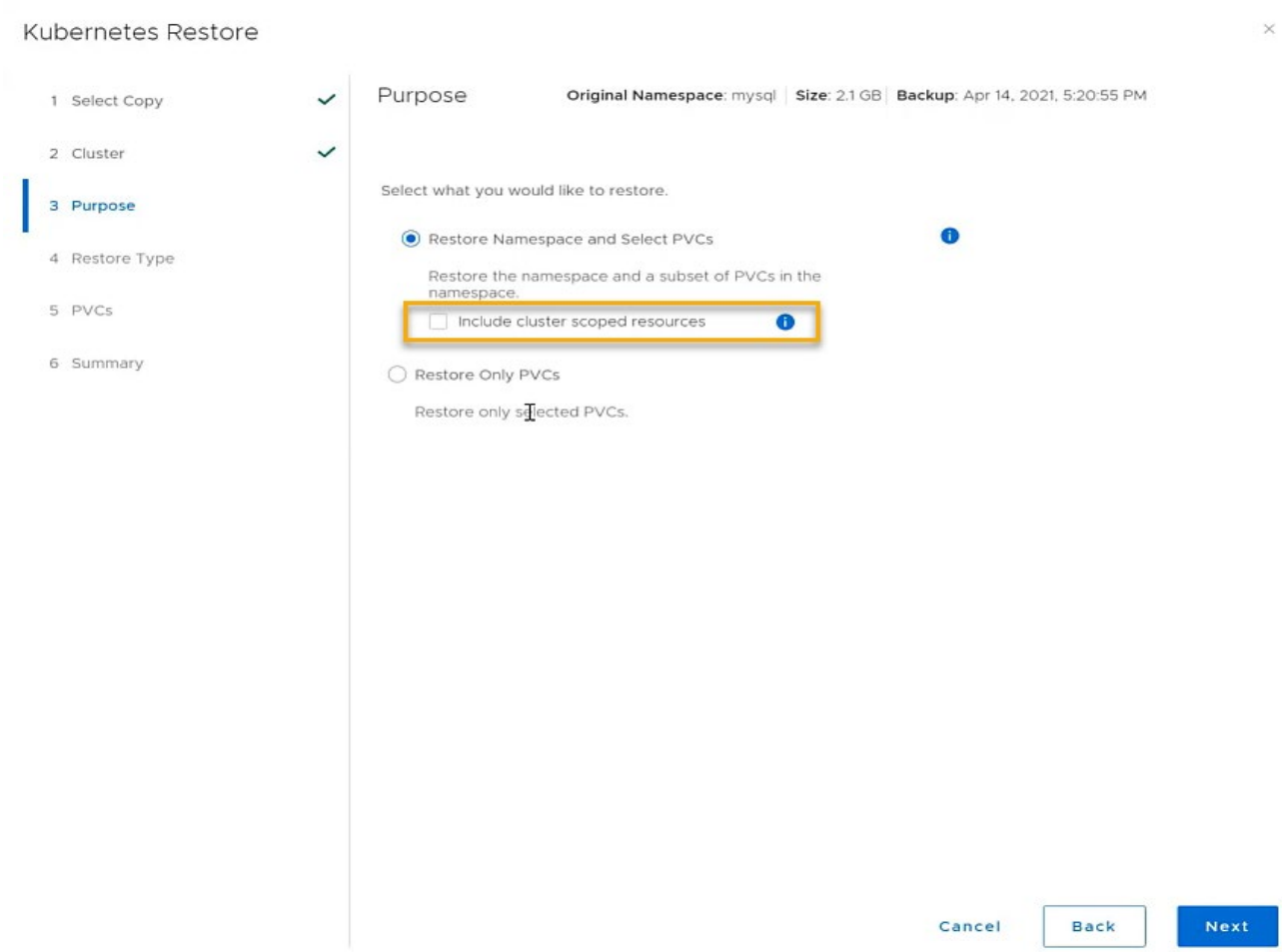
g. **Summar**
Verify the information and click on **Restore**

Note: There are limitations while using the restore with storage class mapping feature.
1. Storage class cannot be altered for existing PVCs
2. Restore from vSphere CSI storage class to other CSI storage class is not supported

4.3.3 Restore Cluster Scoped Resources

The resources that are scoped at a cluster level and not bound to any specific namespace are called Cluster scoped resources for example Cluster Roles, Cluster Role Bindings, Custom Resource Definitions (CRD). When the CRD is created, Kubernetes API server creates a new RESTful resource path for the specific version created. The CRD can be either namespaced or cluster scoped as specified in `scope` field. In this section, you will look into how you can use Kubernetes backup copies and restore the cluster scoped resources such as service accounts, cluster roles and cluster bindings. Both backup and restore of cluster scoped resources are done by Velero component. Backup of namespace associated cluster scoped resources such as cluster roles, cluster role bindings. Custom resource definitions are included as a part of each namespace backup.

Restore of cluster resources is controlled by a checkbox option in GUI "Include Cluster Scoped Restore" and run the restore process



Below are the steps to verify if the cluster role, cluster binding and CRD.

1. Login to Kubernetes cluster and run below command to view cluster role
   `Kubectl get clusterrole`
2. Run below command to view the clusterrolebinding
   `Kubectl get clusterrolebinding`
3. Run below command to view the cluser resource definitions (CRD
   `Kubectl get crd`
4. Below command to view the service account associated with the namespace
   `Kubectl get serviceaccount -n <namespace>`
5. To delete all the cluster scoped resources associated with the namespace, run below script at Kubernetes cluster
   s./`del-fcd.sh`

A Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage and data protection technical white papers and videos provide expertise that helps to ensure customer success with Dell EMC storage and data protection products.

A.1 Related resources

- Dell EMC PowerProtect Data Manager Data Sheet: https://www.dellemc.com/en-me/collaterals/unauth/data-sheets/products/data-protection/h17691-dellemc-powerprotect- software-ds.pdf
- Dell EMC PowerProtect Data Manager: https://www.delltechnologies.com/en-in/data-protection/powerprotect-data- manager.htm#scroll=off
  - Kubernetes Documentation: https://kubernetes.io/docs/home/
  - Tutorials: https://kubernetes.io/docs/tutorials/
  - CSI-driver-host-path: https://github.com/kubernetes-csi/csi-driver-host-path
- VMware Tanzu Kubernetes Grid documentation: https://docs.vmware.com/en/VMware-Tanzu-Kubernetes-Grid/1.1/tanzu-kubernetes-grid-11.pdf