Technical White Paper

Dell EMC PowerProtect Data Manager: Kubernetes Role Based Access Control

Abstract

This white paper describes the minimum role and service-account requirements to integrate a Kubernetes cluster with Dell EMC PowerProtect Data Manager.

September 2021

H18572

Revisions

Date	Description
October 2020	Initial release
September 2021	Content updates

Acknowledgments

Author: Debjeet Bagchi

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Table of contents

Executive summary

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, and it facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem, which continues to increase the importance of protecting Kubernetes clusters and resources.

Dell EMC PowerProtect Data Manager solves the requirement to protect Kubernetes clusters, pods, persistent volume claims, namespaces, and other resources. This document describes how PowerProtect Data Manager protects Kubernetes resources and focuses on the role-based access control (RBAC) requirements.

To begin protecting the Kubernetes cluster, PowerProtect Data Manager must first discover the Kubernetes cluster to enable viewing the assets (namespaces) and the underlying resources that require protection. PowerProtect Data Manager creates two namespaces after discovery: PowerProtect and velero-ppdm. These namespaces contain the powerprotect-controller pod (used during backup of data or PVC) and the Velero pod (used during backup of metadata and backup of FCD PVC).

Primarily, PowerProtect Data Manager uses the following three service accounts for Kubernetes data protection:

- Service account provided to PPDM for discovery when adding Kubernetes cluster as asset-source
- Service account used by PowerProtect Controller Pod
- Service account used by Velero

As part of the discovery or integration process, these service accounts bind to the cluster administrator role and have full access to the entire cluster and its resources. The main underlying problem is that DevOps teams might not want to provide cluster administrator access to these namespaces. This issue requires addressing the intrinsic requirements for these service accounts so that a subsequent role or service account is created for PowerProtect Data Manager to use.

To restrict privileges for these service accounts, you must define RBAC roles for each service account at both the namespace level (Roles) and at the cluster level (ClusterRoles). Then, you must bind these roles to the respective service accounts using RoleBindings and ClusterRoleBindings. For more details about general RBAC rules for Kubernetes, see the Kubernetes article Using RBAC Authorization.

1 Introduction to RBAC in Kubernetes

Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within the organization. RBAC authorization uses the rbac.authorization.k8s.io API group to drive authorization decisions, allowing you to dynamically configure policies through the Kubernetes API.

Permissions that are provided during RBAC are purely additive, and there are no **deny** rules.

A Role always sets permissions within a namespace. When you create a Role, you must specify the namespace that it belongs in. The ClusterRole, by contrast, is a non-namespaced resource that grants access at the cluster level. You can use ClusterRoles for several uses:

- Define permissions on namespaced resources and be granted permissions within an individual namespace or namespaces
- Define permissions on namespaced resources and be granted permissions across all namespaces
- Define permissions on cluster-scoped resources

If you are defining a role within a namespace, use a Role. If you are defining a role cluster-wide, use a ClusterRole.

We recommended following the principle of least privilege and granting more privileges as necessary for work to proceed. Figure 1 shows a general representation of the Kubernetes RBAC resource relationship.

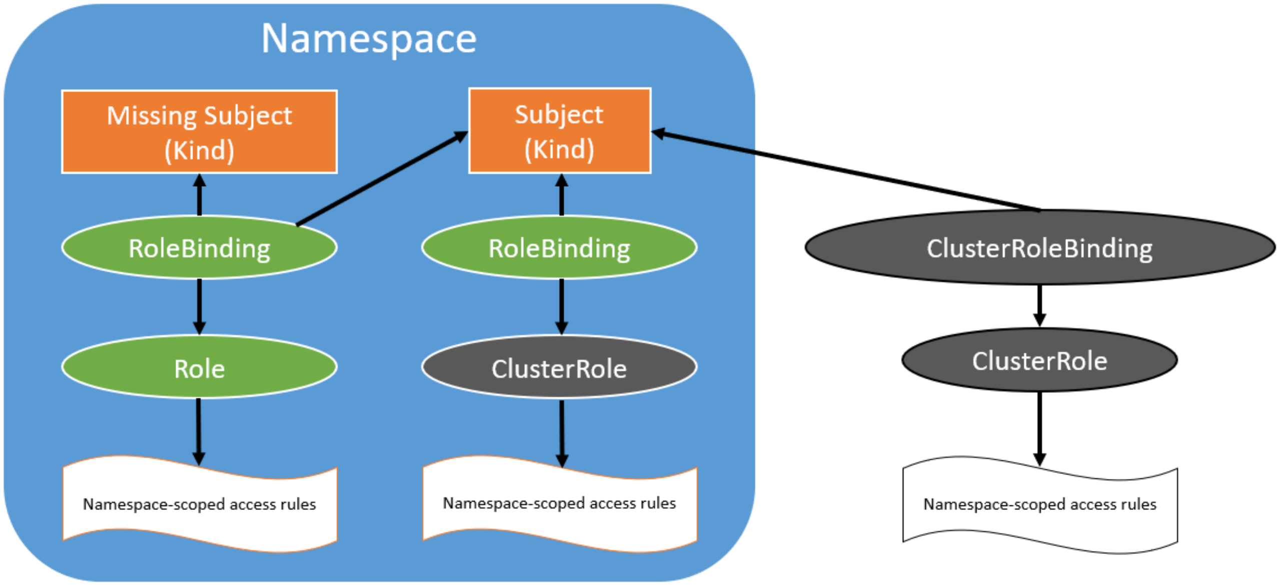


Figure 1 Kubernetes RBAC resource relationship

2 Defining permissions for service accounts

To successfully discover and integrate the Kubernetes cluster with PowerProtect Data Manager, you must create three service accounts (discovery, powerprotect-controller, and Velero).

This section discusses each account and describes how to define the RBAC permissions to minimize privileges.

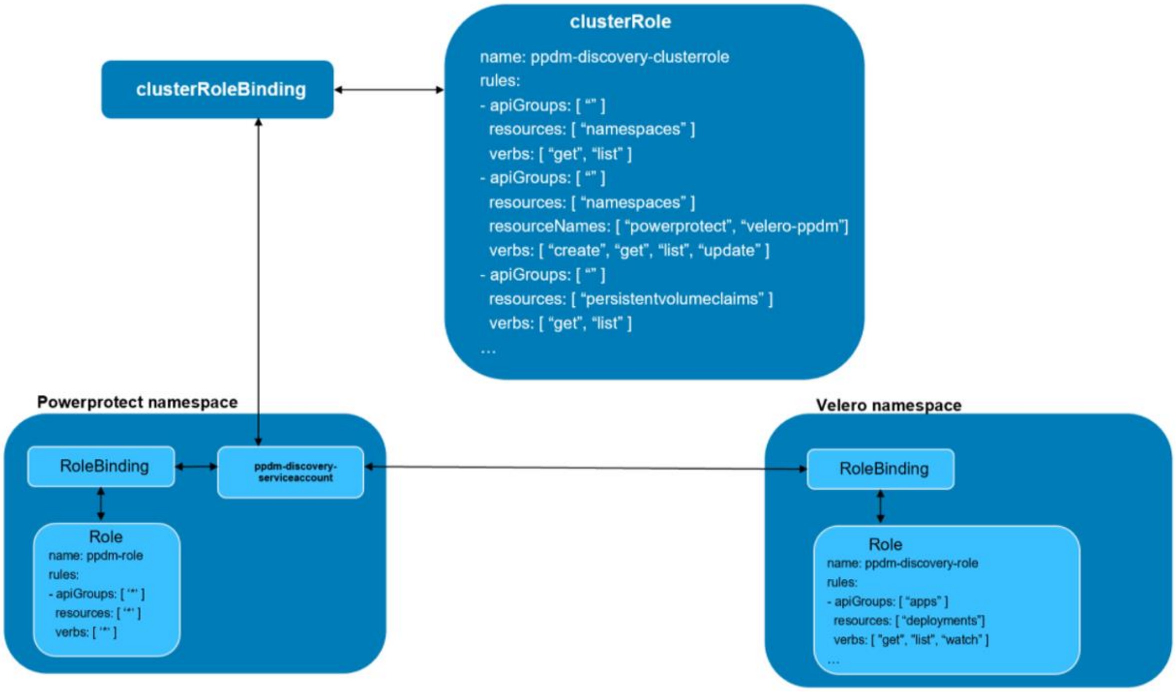
2.1 RBAC for service account provided to PPDM for discovery

The discovery service account requires permissions for the following resources at the cluster level:

- Namespaces: Read access to get and list all namespaces, and write access to create the powerprotect namespace
- ClusterRoleBindings: Read/write access to create, get, update, and patch
- StorageClasses: Read access to get and list storage classes
- PersistentVolumeClaims: Read access to get and list all PVCs across all namespaces
- CustomResoureDefinitions (CRDs): Read/write access to get, create, and update PowerProtect controller CRDs
- Full read/write access: Applied to all objects in the **powerprotect.dell.com** API group; this enables CNDM to perform CRUD operations on backup and restore jobs

Also, the discovery service account requires the following role in the Velero namespace:

- Deployments: Read access to query the status of the Velero deployment
- Pods: Read access to query the status of the Velero Pod
- PowerProtect namespace: The discovery service account is granted full permissions



```
name: ppdm-discovery-clusterrole
rules:
- apiGroups: [ “” ]
  resources: [ “namespaces” ]
  verbs: [ “get”, “list” ]
- apiGroups: [ “” ]
  resources: [ “namespaces” ]
  resourceNames: [ “powerprotect”, “velero-ppdm”]
  verbs: [ “create”, “get”, “list”, “update” ]
- apiGroups: [ “” ]
  resources: [ “persistentvolumeclaims” ]
  verbs: [ “get”, “list” ]
…
```

```
name: ppdm-role
rules:
- apiGroups: [ ‘*' ]
  resources: [ ‘*' ]
  verbs: [ ‘*' ]
```

```
name: ppdm-discovery-role
rules:
- apiGroups: [ “apps” ]
  resources: [ “deployments”]
  verbs: [ "get", "list", “watch” ]
…
```

Figure 2 RBAC for PowerProtect Data Manager discovery service account

Table 1 Minimal RBAC policy for the discovery service account for ClusterRole and Role

Role	apiGroups	resources	verbs	resourceNames	Namespace
Cluster Role	“rbac.authorization.k8s.io"	"clusterrolebindings"	1."get" 2. "list"	"powerprotect:cluster-role-binding"	
	“storage.k8s.io”	"storageclasses"	1."get" 2. "list"	N/A	
	"apiextensions.k8s.io"	"customresourcedefinitions"	1.”create” 2."get" 3. "list" 4. “update” 5. “delete” 6. “patch”	N/A	
	"powerprotect.dell.com"	‘*’	‘*’	N/A	
	“apps”	“deployments”	1."get" 2. "list" 3. “watch”	N/A	
	“” (Core API)	namespaces	1."create” 2. “get” 3. “list”	powerprotect	
		pods	1."get" 2. "list" 3. “watch”	N/A	
		persistentvolumeclaims	1."get" 2. "list"	N/A	
Role	‘*’	‘*’	‘*’	N/A	powerprotect

2.1.1 configmap for discovery service account

A configmap is created in the powerprotect namespace as part of the PowerProtect discovery phase and is explicitly mentioned in the RBAC executable yaml.

The **ppdm-custom-config-resources** offers more flexibility to the user. If the Cloud Native microservice (CNDM) locates this configmap, it regards the powerprotect namespace, powerprotect cluster-role binding, and powerprotect controller service account as read-only objects. This behavior is observed because if the powerprotect namespace is created in advance and last-applied configuration annotation does not match, the discovery proceeds.

2.2 RBAC for service account used by PowerProtect controller pod

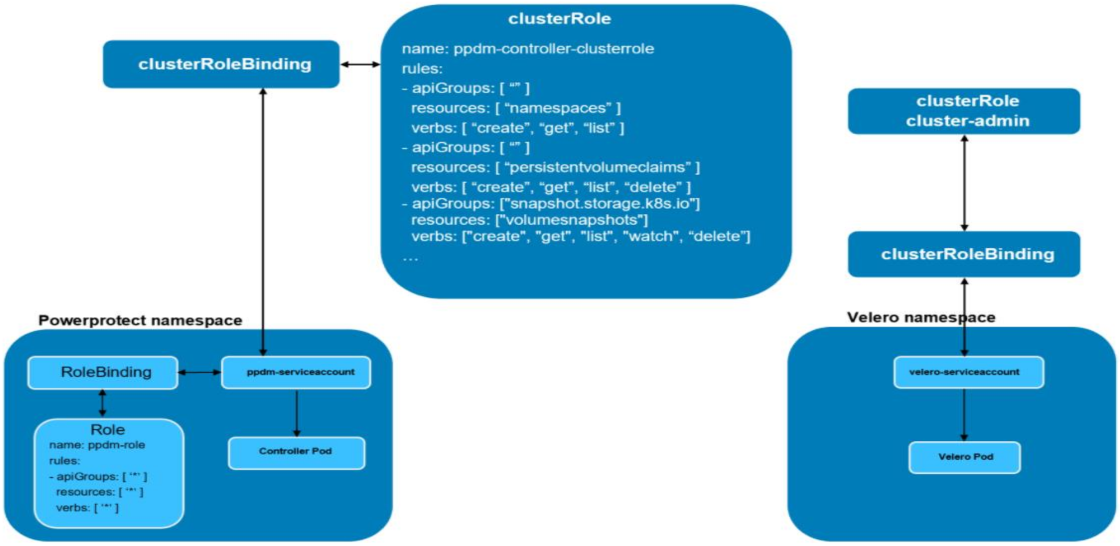
The PowerProtect controller pod runs as the **ppdm-serviceaccount** service account. This service account requires following permissions:

- At the cluster-level (ClusterRole):
  - Namespaces: Read and write access to create, get, and list all namespaces
  - ClusterRoleBindings: Read access to get and list ClusterRoles, and write access to create ClusterRole binding for Velero service account to the cluster-admin role
  - CustomResoureDefinitions (CRDs): Read/write access to get, create, and update Velero CRDs
  - StorageClasses: Read access to get and list storage classes
  - PersistentVolumes: Read access to get and list persistent volumes
  - PersistentVolumeClaims: Read/write access to create, get, list, and watch all PVCs across all namespaces
  - VolumeSnapshotClasses: Read access to get and list volume snapshot classes
  - VolumeSnapshots: Read/write access to create, get, list, watch, and delete volume snapshots
  - VolumeSnapshotContents: Read/write access to create, get, list, watch, and delete volume snapshot contents
  - Secrets: Read access to get the VMware® vSphere® secret (required for VMware Tanzu or Pacific clusters)
  - ServiceAccounts: Write access to create Velero service account in Velero namespace
  - Pods: Read/write access to create and delete (cproxy pods), or to get, list, watch, and update pods
  - Pods/exec: Write access to create pod exec hooks (required for pre or post hooks for application-consistent backups)
  - Deployments: Read/write access to get, list, and update deployments
  - Deployments/scale: Read/write access to scale deployments (required for a restore when running workloads must be scaled down to zero before their PVCs are restored)
  - ReplicaSets: Read/write access to get, list, and update replicasets
  - ReplicaSets/scale: Read/write access to scale replicasets (required for a restore when running workloads must be scaled down to zero before their PVCs are restored)
  - DaemonSets: Read/write access to get, list, and update daemonsets
  - DaemonSets/scale: Read/write access to scale daemonsets (required for a restore when running workloads must be scaled down to zero before their PVCs are restored)
  - StatefulSets: Read/write access to get, list, and update statefulsets
  - StatefulSets/scale: Read/write access to scale statefulsets (required for a restore when running workloads must be scaled down to zero before their PVCs are restored)

  - Full read/write access: Provided to all objects in the **powerprotect.dell.com** and **velero.io** API group (required for the controller to perform CRUD operations on backup storage locations, backupjobs, or restorejobs)

- PowerProtect namespace
  - Full access in the PowerProtect namespace

- Velero namespace
  - Access to create a Velero deployment and update the deployment
  - Access to create a Velero service account
  - Access to create Velero custom resources in the **velero.io** API group (Velero backup and restore objects)



```
name: ppdm-controller-clusterrole
rules:
- apiGroups: [ “” ]
  resources: [ “namespaces” ]
  verbs: [ “create”, “get”, “list” ]
- apiGroups: [ “” ]
  resources: [ “persistentvolumeclaims” ]
  verbs: [ “create”, “get”, “list”, “delete” ]
- apiGroups: ["snapshot.storage.k8s.io”]
  resources: ["volumesnapshots”]
  verbs: ["create", "get", "list", "watch", “delete”]
…
```

```
name: ppdm-role
rules:
- apiGroups: [ ‘*’ ]
  resources: [ ‘*’ ]
  verbs: [ ‘*’ ]
```

Figure 3 RBAC for PowerProtect controller servcieaccount

Table 2 Minimal RBAC policy for the PowerProtect controller service account for ClusterRole and Role

Role	apiGroups	resources	verbs	resourceNames	Namespace
Cluster Role	“rbac.authorization.k8s.io"	"clusterrolebindings"	1."get" 2. "list" 3. “create”	N/A	
		“clusterroles”	1.“bind”	“cluster-admin”	
	“storage.k8s.io”	"storageclasses"	1."get" 2. "list" 3. “watch”	N/A	

Role	apiGroups	resources	verbs	resourceNames	Namespace
	"apiextensions.k8s.io"	"customresourcedefinitions"	1. ”create” 2. "get" 3. "list" 4. “update”	N/A	
	"snapshot.storage.k8s.io"	"volumesnapshotclasses"	1."get" 2. "list" 3. “watch”	N/A	
		"volumesnapshots"	1. ”create” 2. "get" 3. "list" 4. “watch” 5. “delete”	N/A	
		"volumesnapshotcontents"	1. ”create” 2. "get" 3. "list" 4. “watch” 5. “delete”	N/A	
	"powerprotect.dell.com"	‘*’	‘*’	N/A	
	“velero.io”	‘*’	‘*’	N/A	
	“apps”	“deployments”, “deployments/scale”	1."get" 2. "list" 3. “create” 4. “update” 5. “patch” 6. “watch”	N/A	

Role	apiGroups	resources	verbs	resourceNames	Namespace
		“replicasets”, “replicasets/scale”	1. “get” 2. “list” 3. “update” 4. “watch”	N/A	
		“daemonsets”, “daemonsets/scale”	1. “get” 2. “list” 3. “update” 4. “watch”	N/A	
		“statefulsets”, “statefulsets/scale”	1. “get” 2. “list” 3. “update” 4. “watch”	N/A	
	“” (Core API)	namespaces	1."get" 2. "list" 3. “create” 4. “delete” 5. “watch”	N/A	
		pods	1."get" 2. "list" 3. “watch” 4. “create” 5. “update” 6. “delete”	N/A	
		persistentvolumeclaims	1."get" 2. "list" 3. “create”	N/A	

Role	apiGroups	resources	verbs	resourceNames	Namespace
			4. “update” 5. “watch” 6. “delete”		
		persistentvolumes	1. “get” 2. “list” 3. “update” 4. “delete”	N/A	
		pods/exec	1. “create”	N/A	
		secrets	1.“get”	N/A	
		serviceaccounts	1. “create” 2. “get”	N/A	
		replicationcontrollers replicationcontrollers /scale	1. “update” 2. “watch”	N/A	
		services	1. “create” 2. “get” 3. “list” 4. “watch” 5. “update” 6. “delete”	N/A	
		configmaps	1. “create” 2. “get” 3. “list” 4. “watch” 5. “update” 6. “delete”	N/A	

Role	apiGroups	resources	verbs	resourceNames	Namespace
	“backupdriver.cnsdp.vmware.com”	snapshots deletesnapshots deletesnapshots/status	1. “create” 2. “get” 3. “list” 4. “watch” 5. “update” 6. “delete”	N/A	
	“networking.k8s.io”	networkpolicies	1. “create” 2. “get” 3. “delete”	N/A	
	“operators.coreos.com”	‘*’	‘*’	N/A	
	“konveyor.openshift.io”	‘*’	‘*’	N/A	
Role	‘*’	‘*’	‘*’	N/A	powerprotect

Note: If there are no pods that are managed by daemonsets, statefulsets, and replicasets objects that use PVC, their respective permissions can be removed. These permissions are only additive, so if you do not have any workloads that use these objects, their permissions are not needed.

2.3 Custom role in Velero namespace for PowerProtect controller

You can restrict the permissions provided to the controller service account further by defining a role for the ppdm-serviceaccount in the Velero namespace. This role has the minimal permissions set required to install Velero and perform Velero backup and restores. You can bind the controller service account to this role using a RoleBinding in the Velero namespace.

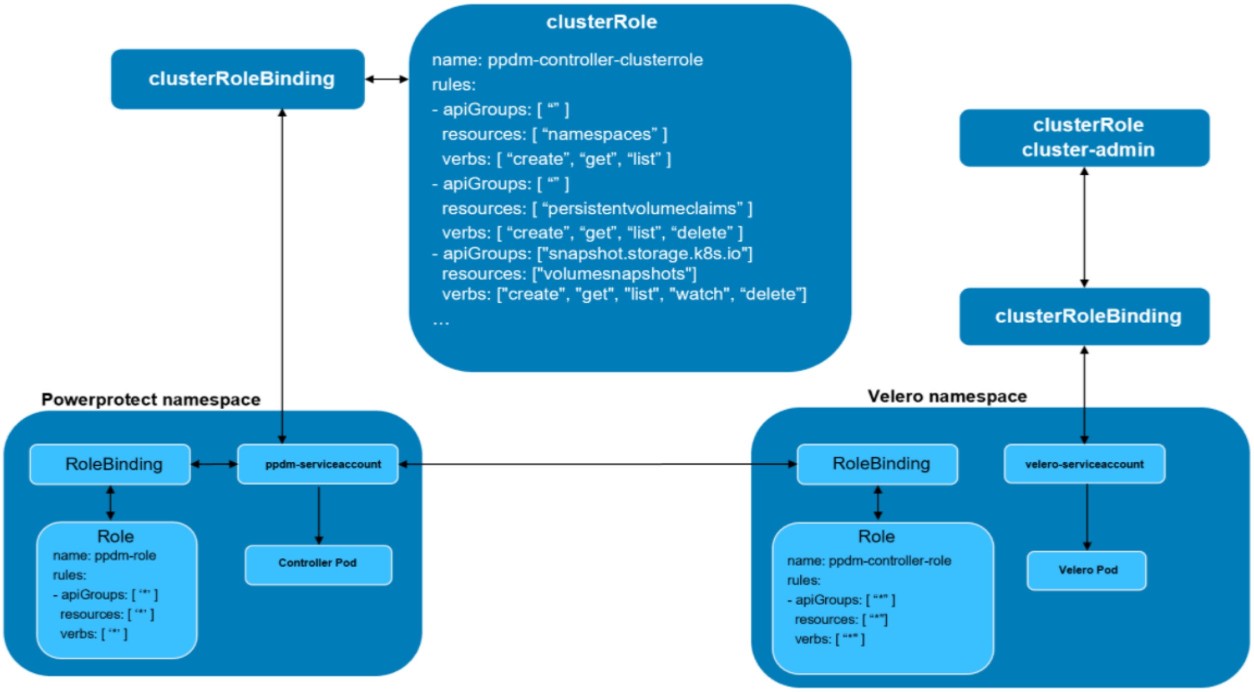


Figure 4 Custom role for velero namespace for PowerProtect controller

2.4 RBAC for service account provided to PPDM for discovery

By default, Velero runs as the cluster-admin. This behavior ensures that Velero can back up or restore all objects across all namespaces in the cluster. Velero does not provide explicit steps to restrict privileges granted to the Velero service account, but provides high-level documentation on minimizing RBAC for Velero.

2.5 Annotation requirements for integration with PowerProtect Data Manager 19.5

To use the minimum RBAC privileges for the previously mentioned service accounts with PowerProtect Data Manager 19.5, ensure that the following annotations are used:

1. PPDM discovery:

   a. kind: Namespace

   ```
   powerprotect/last-applied-configuration:
   '{"apiVersion":"v1","kind":"Namespace","metadata":{"labels":{"app.kubernet
   es.io/part-of":"powerprotect.dell.com"},"name":"powerprotect"}}'
   ```

2. PowerProtect controller:

   a. kind: Namespace

   ```
   powerprotect/last-applied-configuration:
   '{"apiVersion":"v1","kind":"Namespace","metadata":{"labels":{"app.kubernet
   es.io/part-of":"powerprotect.dell.com"},"name":"powerprotect"}}'
   ```

b. kind: ServiceAccount

```
powerprotect/last-applied-configuration:
'{"apiVersion":"v1","kind":"ServiceAccount","metadata":{"labels":{"app.kub
ernetes.io/part-of":"powerprotect.dell.com"},"name":"ppdm-
serviceaccount"}}'
```

c. kind: ClusterRoleBinding

```
powerprotect/last-applied-configuration:
'{"apiVersion":"rbac.authorization.k8s.io/v1","kind":"ClusterRoleBinding",
"metadata":{"labels":{"app.kubernetes.io/part-
of":"powerprotect.dell.com"},"name":"powerprotect:cluster-role-
binding"},"roleRef":{"apiGroup":"rbac.authorization.k8s.io","kind":"Cluste
rRole","name":"cluster-
admin"},"subjects":[{"kind":"ServiceAccount","name":"ppdm-
serviceaccount","namespace":"powerprotect"}]}'
```

3 Deployment sequence

This section describes how to use the precreated service account yaml files, and how you can use the same sequence if using a customized discovery with controller yaml files.

3.1.1 Integration with PowerProtect Data Manager

Perform the following:

1. Open the Kubernetes command-line console of the cluster control plane.
2. Run the following commands:

```
kubectl apply -f ppdm-discovery.yaml
kubectl apply -f ppdm-controller-rbac.yaml
kubectl get secrets -n powerprotect
kubectl describe secret ppdm-discovery-serviceaccount-token-xxxxx -n powerprotect
   {Record the secret key}
kubectl cluster-info {Record the Kubernetes primary/control-plane endpoint}
```

3. Go to the PowerProtect Data Manager UI and add the Kubernetes cluster as an Asset Source. Use the values for the secret key and the Kubernetes master/control-plan endpoint recorded from the previous commands.
4. Once discovery status shows **OK**, run the following commands in the Kubernetes cluster console:

```
kubectl get ns   {You should see powerprotect and velero-ppdm namespaces created}
kubectl get pods -n powerprotect     {powerprotect controller pod should be
running}
kubectl get pods -n velero-ppdm      {velero pod should be running}
```

Note: To retrieve the yaml files referenced above, log in to the PowerProtect Data Manager command-line interface using admin credentials in the path **/usr/local/brs/lib/cndm/misc/** and reference the file **rbac.tar.gz**.

A Technical support and resources

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

The Dell Technologies Info Hub provides expertise that helps to ensure customer success with Dell EMC data protection products.

A.1 References

For more information, see the following reference links:

- https://kubernetes.io/docs/reference/access-authn-authz/rbac/
- https://velero.io/docs/main/rbac/
- https://github.com/alcideio/rbac-tool