SUSE Rancher and RKE Kubernetes cluster using CSI Driver on DELL EMC PowerFlex

September 2021

H18899

White Paper

Abstract

This white paper describes the deployment of a SUSE Rancher Kubernetes Cluster on the Dell EMC PowerFlex family and the integration of PowerFlex CSI driver 1.4 for persistent volume, for customers requiring an on-premises container platform solution. This white paper also explains how to protect the above Kubernetes workloads with Dell EMC PowerProtect Data Manager.

Dell Technologies Solutions

PowerFlex Engineering

Validated

Copyright

Contents

Executive Summary

Container technologies enable development teams to quickly provision isolated applications. Customers who want to boost their productivity and reduce the time to value, can use containers with the departments that are focused on software development.

Kubernetes orchestration provides capabilities such as auto scaling, security, and management of containerized applications. A persistent and stable data store is required to run containerized applications within a Kubernetes cluster, that can survive the lifetime of a pod or the node it is running on.

SUSE Rancher is a Kubernetes management platform that simplifies the cluster installation and operations, whether they are on-premises, in the cloud, or at the edge, giving the DevOps team freedom to build and run containerized applications anywhere.

The PowerFlex family offers key value propositions for traditional and cloud-native production workloads, deployment flexibility, linear scalability, predictable high performance, and enterprise-grade resilience.

Dell EMC PowerProtect Data Manager enables users to protect, manage, and recover data in on-premises, virtualized, or cloud deployments. The PowerProtect Data Manager platform provides centralized governance that helps mitigate risk and assures compliance of SLAs and SLOs through protection workflows. PowerProtect Data Manager is used to discover, protect, and restore production workloads in Kubernetes environments and protects production and development, or test workloads to ensure that the data is easy to backup and restore.

PowerProtect Data Manager enhances the protection by sending the data directly to the Dell EMC PowerProtect DD series appliance to gain benefits from unmatched efficiency, deduplication, performance, and scalability. Together with PowerProtect Data Manager, PowerProtect DD series appliance is the ultimate protection storage appliance.

This paper discusses how to deploy a Kubernetes cluster using SUSE Rancher Kubernetes Engine on a PowerFlex cluster and the use of the PowerFlex CSI 1.4 driver to provision persistent storage for Kubernetes managing containerized applications. This white paper also describes how to protect the workloads that are deployed in a SUSE Rancher managed Kubernetes cluster with PowerProtect Data Manager and DD series appliance.

Introduction

Revisions

Date	Description
April 2020	Initial Release
June 2020	Updated VxFlex to PowerFlex as per new rebranding guidelines
June 2021	Leveraging SUSE Enterprise Linux RKE nodes
September 2021	Updated data protection for Rancher Kubernetes cluster using Dell EMC PowerProtect Data Manager

We value your feedback

Dell Technologies and the authors of this document welcome your feedback on the solution and the solution documentation. Contact the Dell Technologies Solutions team by email or provide your comments by completing our documentation survey.

Author: Sanjay Puttaswamy, Raghavendra Biligiri, Praphul Krottapalli, Suraram Anil Kumar, and Vinod Kumaresan.

Contributors from SUSE Team: Gerson Guevara

Note: For links to additional documentation for this solution, see Dell Technologies Solutions Info Hub for PowerFlex.

Objective

This white paper gives you an overview of the creation of a Kubernetes cluster using SUSE Rancher Kubernetes Engine and managing the cluster using SUSE Rancher as the container orchestration layer on Dell EMC PowerFlex platform to meet the performance, scalability, resiliency, and availability requirements. This white paper also describes how to leverage the Dell EMC PowerFlex CSI driver to dynamically provision persistent volumes within the SUSE Rancher managed Kubernetes cluster and show how the PowerProtect Data Manager can protect the workloads in a SUSE Rancher managed Kubernetes cluster.

Audience

This white paper is intended for sales engineers, field consultants, IT administrators, customers, and anyone else interested in configuring and deploying a Kubernetes cluster using RKE with the PowerFlex CSI driver to dynamically provision persistent volumes in a SUSE Rancher managed Kubernetes cluster. This white paper is also intended for customers, partners, and anyone who wants to understand how PowerProtect Data Manager protects the Kubernetes workloads that are deployed in a SUSE Rancher managed Kubernetes cluster.

The intended audience of this white paper must have a working knowledge of containers, Kubernetes, PowerFlex, and Data Protection.

Terminology

Table 1. Terminology

Term	Definition
CA	Certificate Authority
CNS	Cloud Native Storage
CSI	Container Storage Interface

Term	Definition
DD	Data Domain
DNS	Domain Name System
DDVE	PowerProtect DD Virtual Edition
FQDN	Fully Qualified Domain Name
MDM	Meta Data Manager
OVA	Open Virtualization Appliance
PV	Persistent Volume
PVC	Persistent Volume Claim
RKE	SUSE Rancher Kubernetes Engine
RMT	Repository Mirroring Tool
SAN	Subject Alternative Name
SDC	Storage Data Client for PowerFlex
SDS	Storage Data Server for PowerFlex
SLES	SUSE Linux Enterprise Server
SSD	Solid-State Disk
TLS	Transport Layer Security
VLAN	Virtual Local Area Network
VM	Virtual Machine

PowerFlex product overview

PowerFlex family

PowerFlex is a software-defined infrastructure platform that is built to reduce operational and infrastructure complexity. PowerFlex empowers organizations to move faster by delivering flexibility, elasticity, and simplicity with extraordinary predictable performance for mission-critical workloads, while also providing resiliency at scale. The PowerFlex family provides a foundation that combines compute and high-performance storage resources in a managed, unified fabric.

PowerFlex delivers transformational value

- Delivers stringent SLAs effortlessly

  PowerFlex leverages software to unlock the full potential of rapid advances in industry-standard hardware and deliver extreme SLA outcomes. PowerFlex aggregates resources across a broad set of nodes, unlocking massive input, output, and throughput performance while minimizing the latency. Its self-balancing architecture eliminates any hotspots and ensures consistency and simplicity over time. You can scale the system while linearly scaling performance from a minimum of four nodes to thousands of nodes, on-demand and without any disruption. And with its self-healing architecture, PowerFlex can handle outages, upgrades, and maintenance without downtime resulting in 99.9999 percent availability.

- Flexible and dynamic architecture

  PowerFlex delivers transformational agility that enables organizations to rapidly respond to changing business needs. PowerFlex offers flexibility to mix and match the storage, compute, and hyperconverged nodes in a dynamic deployment, allowing you to scale storage and compute resources together or independently, one node at a time as per your requirements.

- Shared platform for heterogeneous workloads

  The platform can support a broad range of operating environments simultaneously such as, bare-metal operating systems, hypervisors, and container platforms with a unified underlying infrastructure platform and management. It can also support heterogeneous workloads with varying requirements on a flexible shared infrastructure platform and modernize your application architectures on your schedule.

- Extensive automation for predictability and simpler workflows

  PowerFlex offers full-stack IT Operations Management (ITOM) and Life Cycle Management (LCM) with PowerFlex Manager. It provides extensive automation capabilities with PowerFlex Manager REST APIs and custom Ansible modules to integrate with your infrastructure, application, and DevOps workflows. PowerFlex Manager enables automated deployments and expansions with minimal firsthand time for the IT team, letting them focus on other strategic initiatives.

- Broad ecosystem of workload solutions

  PowerFlex is designed to deliver superior outcomes at any scale for the most-demanding mission-critical environments. It is optimized for a wide range of validated workload solutions ranging from traditional relational databases and modern cloud-native NoSQL databases to throughput-intensive analytics workloads.

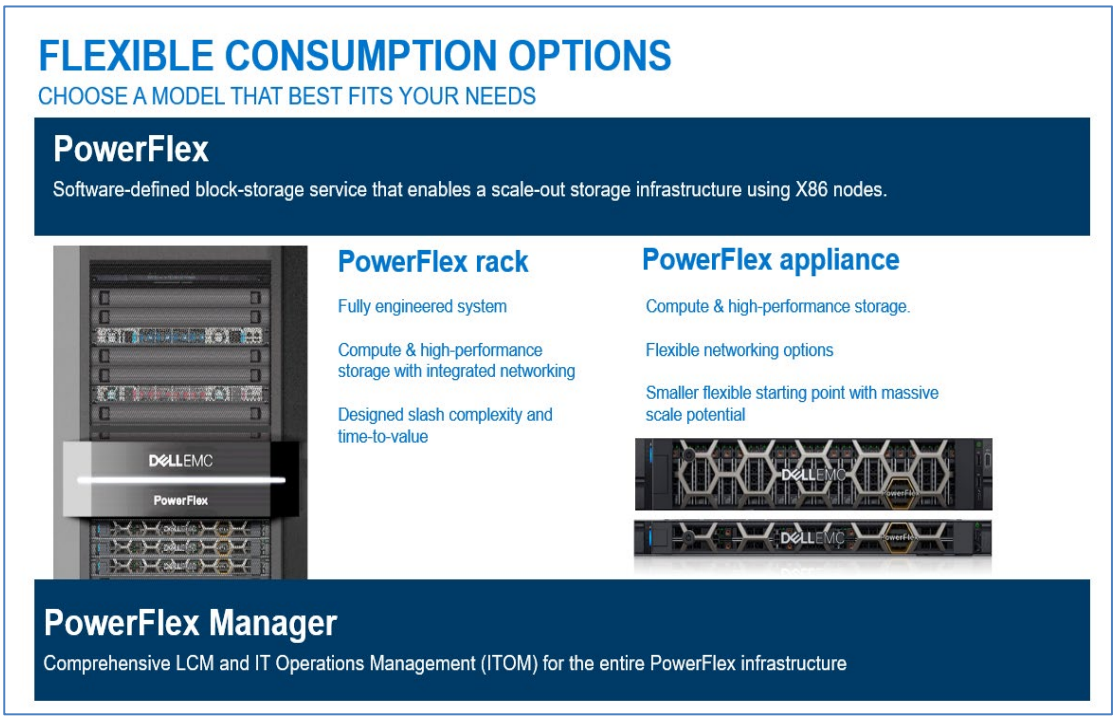


Figure 1. PowerFlex family

PowerFlex software components

Software is the key factor of success in the PowerFlex offering. PowerFlex software components provide software-defined storage services. The software components help to simplify the infrastructure management and orchestration with comprehensive IT Operations Management (ITOM) and life cycle management (LCM) capabilities that span compute and storage infrastructure, from BIOS and firmware to nodes, software, and networking.

PowerFlex

PowerFlex is the software foundation of the PowerFlex platform. It delivers high performance, highly resilient block storage service that can scale to thousands of nodes.

PowerFlex Manager

PowerFlex Manager is the software component in PowerFlex family that enables ITOM automation and LCM capabilities while enabling flexible APIs and extensive automation.

PowerFlex consumption options

PowerFlex is available in multiple consumption options to help customers meet their project and data center requirements. PowerFlex appliance and PowerFlex rack provide customers the flexibility to choose a deployment option to meet their exact requirements.

PowerFlex rack

PowerFlex rack is a fully engineered system, with integrated networking that enables the customers to simplify deployments and accelerate time to value.

PowerFlex appliance

PowerFlex appliance allows customers the flexibility and savings to bring their own compatible networking. It offers customers a smaller starting point of four nodes, while enabling them to use their existing network infrastructure.

With PowerFlex, the customers deploy to match their initial needs and easily expand with massive scale potential, without having to compromise on performance and resiliency.

Flexible consumption-based billing options

PowerFlex is available through APEX custom solutions by the APEX Flex on Demand and APEX Datacenter Utility for customers looking to adopt consumption-based OpEx models.

APEX Flex on Demand

APEX Flex on Demand allows you to pay for technology as you use it and provides immediate access to buffer capacity. Your payment adjusts to match your usage.

APEX Datacenter Utility

APEX Datacenter Utility provides the leading product portfolio from Dell that is coupled with your choice of professional services and support to manage your data center and its operations. A simple and single invoice provides monthly payments that are based on a predictable rate and vary based on your usage.

SUSE Rancher for Kubernetes

SUSE Rancher is an enterprise computing platform to run Kubernetes on-premises, in the cloud, and at the edge. It addresses the operational and security challenges of managing multiple Kubernetes clusters anywhere. SUSE Rancher also provides IT operators and development teams with integrated tools for building, deploying, and running cloud-native workloads.

SUSE Rancher deploys production-grade Kubernetes clusters from data center to cloud and edge and unites them with centralized authentication, access control, and observability. SUSE Rancher lets you streamline cluster deployment on bare metal, edge devices, private clouds, public clouds, and vSphere and secures them using global security policies. Use Helm or SUSE Rancher App Catalog to deploy and manage applications across any or all these environments, ensuring multicluster consistency with a single deployment.

How SUSE Rancher delivers production-grade Kubernetes at scale

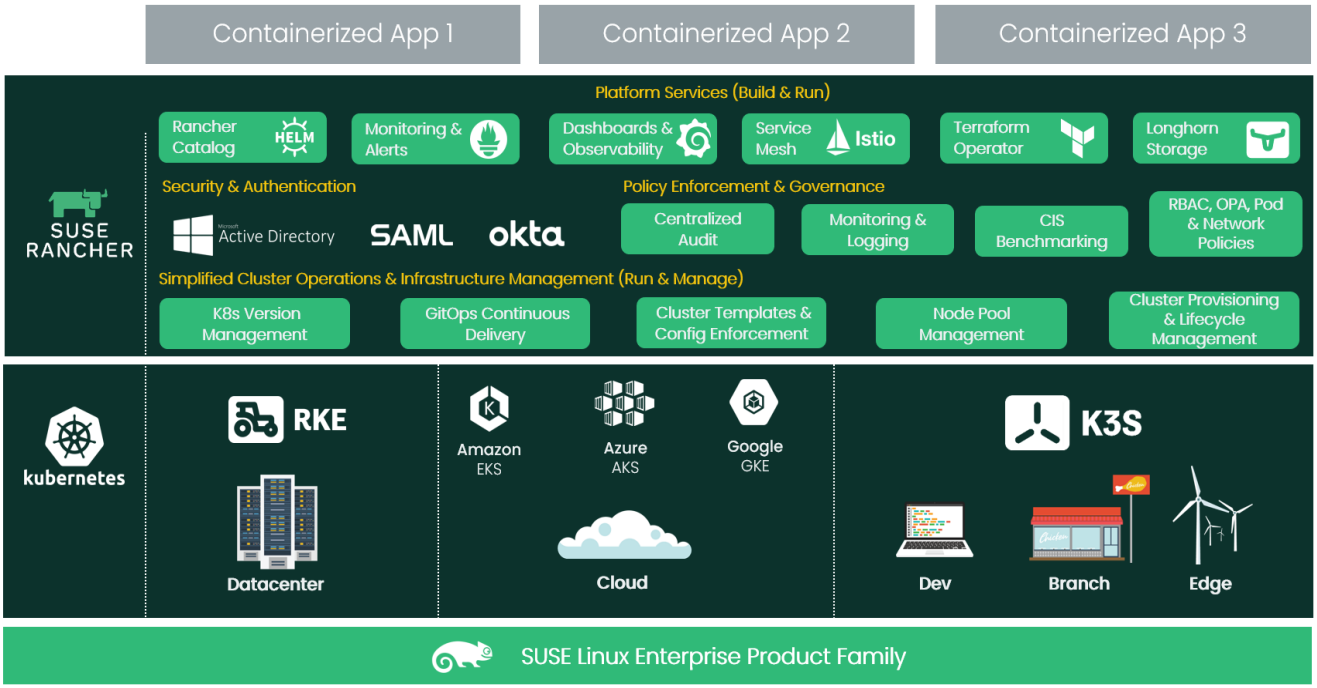


Figure 2. SUSE Rancher procedure for production quality Kubernetes at scale

Certified Kubernetes distributions

SUSE Rancher supports any certified Kubernetes distribution. For on-premises workloads, SUSE Rancher offers the RKE, a CNCF-certified Kubernetes distribution that runs entirely within docker containers. It works on bare-metal and virtualized servers. RKE solves the problem of installation complexity, a common issue in the Kubernetes community. With RKE, the installation and operation of Kubernetes is both simplified and easily automated and it is entirely independent of the operating system.

Simplified cluster operations

SUSE Rancher provides simple and consistent cluster operations including provisioning, version management, visibility and diagnostics, monitoring and alerting, and centralized audit.

Security, policy, and user management

SUSE Rancher lets you automate processes and applies a consistent set of user access and security policies to all your clusters, no matter where they are running.

Shared tools and services

SUSE Rancher provides a rich catalog of services for building, deploying, and scaling containerized applications, including application packaging, continuous integration or continuous deployment, logging, monitoring, and service mesh.

Solution architecture

This section describes how an RKE cluster is deployed on a two-layer PowerFlex cluster. You can deploy this solution on any of the PowerFlex family products. In this solution, the RKE cluster is deployed in a two-layer configuration using PowerFlex compute-only nodes that are deployed with the VMware ESXi hypervisor and dedicated PowerFlex storage-only nodes to provide the required storage capacity.

Note: The solution is validated in engineering lab using two-layer PowerFlex system with VMware vSphere environment, but the reference architecture and the best practices that are demonstrated in this white paper are applicable to both virtualized and bare metal configurations and other PowerFlex family products.

Logical architecture

The following diagram shows logical architecture of the RKE cluster that is deployed on a two-layer PowerFlex cluster setup with four storage-only nodes and three compute-only nodes.

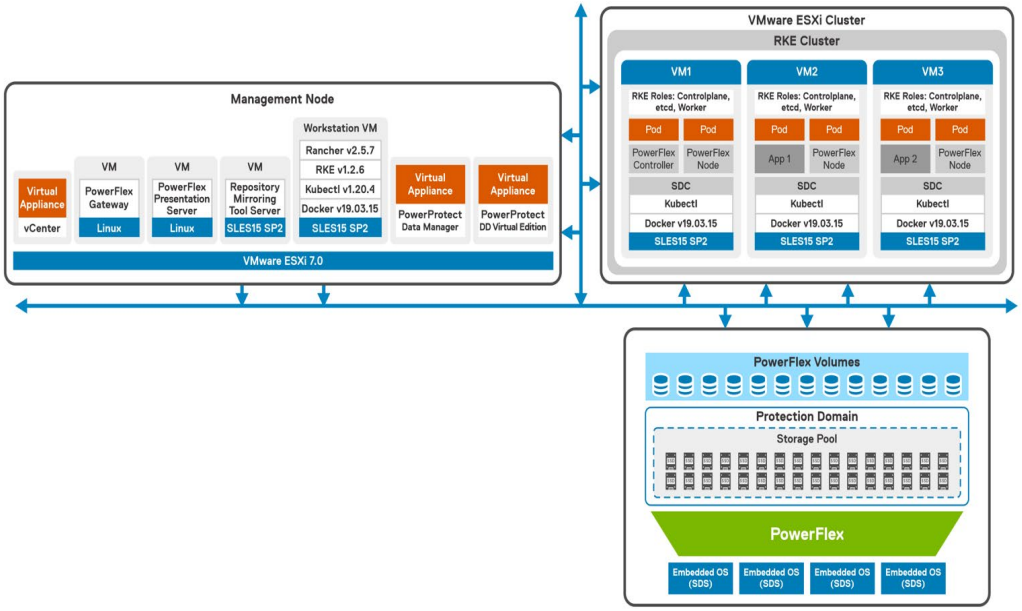


Figure 3. Logical architecture of RKE cluster

In this example, each storage-only node includes two Intel Xeon Scalable 12-core processors, 224 GB RAM, and eight 1.92 TB SSDs. From the PowerFlex standpoint, the embedded operating system storage-only nodes run the SDS component of PowerFlex to provide storage capacity. A single protection domain is created from these four SDS systems. A single storage pool is created within this protection domain from which persistent storage volumes can be provisioned for attachment to Kubernetes pods. The PowerFlex SDC component is installed into the VMware ESXi 7.x hypervisor running on the three compute-only nodes, this provides access to volumes created within the storage pool.

In this solution, each of the virtual machines are installed with SLES15 SP2 operating system. The RKE cluster is deployed on VMs on a VMware vSphere 7.0 cluster. For SUSE Rancher to work, the minimum hardware requirement to run Kubernetes node components is one CPU and 1 GB of memory. Considering the CPU and memory, it is recommended to host the different roles of the Kubernetes cluster such as etcd , control plane, and workers on different nodes, so that they can scale independently from one another. In this solution, each virtual machine is configured with 2 virtual CPUs, 8 GB RAM, and thick provisioned disks on the PowerFlex backend storage.

For more information about configuration of PowerFlex nodes, check the Configuration details.

The management node hosts the vCenter appliance, PowerFlex Gateway, PowerFlex Presentation server, Repository Mirroring Tool (RMT) server, Linux workstation for RKE, PowerProtect Data Manager, and DDVE. The RMT server and Linux workstation are VMs configured with SLES15 SP2 operating system. The RMT server acts as a proxy server to SUSE customer center with repositories. It helps the customers with SUSE Linux Enterprise software updates and subscription entitlements. For more information about RMT server and its configuration, see Repository Mirroring Tool Guide.

Network architecture

The following diagram shows the logical layout of PowerFlex rack access and aggregation with management aggregation architecture:

Note: There is an additional 1 Gb link from the PowerFlex controller nodes to the out-of-band management switch.

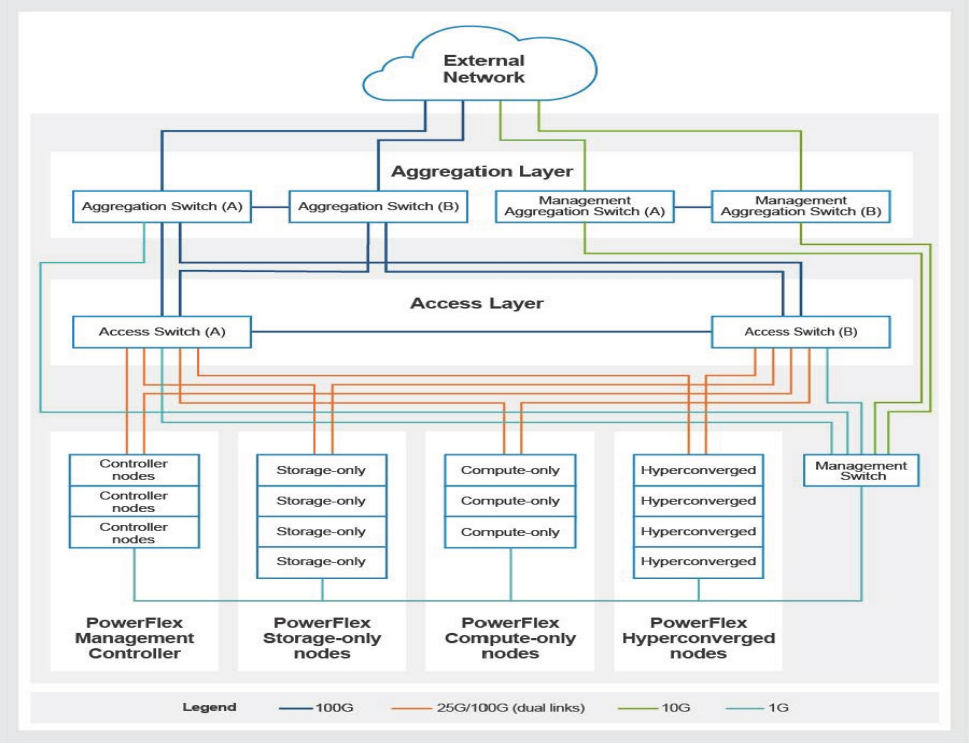


Figure 4. Logical layout of PowerFlex rack access and aggregation

Installation of the SUSE Rancher Kubernetes cluster

This section includes the Kubernetes installation using RKE and the configuration of PowerFlex CSI driver on a two-layer PowerFlex cluster.

SUSE Rancher supports Kubernetes clusters running on Ubuntu, CentOS, Oracle Linux, SLES, and RedHat Enterprise Linux. For information about the supported Operating Systems, Docker, and SUSE Rancher versions, see SUSE Rancher - All Supported Versions.

At the time of validating this solution, the latest version of SUSE Rancher v2.5.7 and RKE version v1.2.6 along with Kubernetes v1.20.4 and docker v19.03.15 for SLES15 SP2 were used. A working DNS or Fully Qualified Domain Name (FQDN) must be configured properly for all the nodes.

Requirements

The requirements for building a SUSE Rancher Kubernetes cluster using RKE are as follows:

Table 2. Requirements for RKE cluster on PowerFlex family

Name	Version	Description	Reference
SUSE Rancher server	2.5.7	SUSE Rancher server is used from Workstation VM.	https://releases.rancher.com/server-charts/stable
SUSE Rancher Kubernetes Engine (RKE)	1.2.6	RKE is used from Workstation VM.	https://github.com/rancher/rke/releases/latest
Kubectl	1.20.4	Kubectl to interact with Kubernetes cluster.	https://kubernetes.io/docs/tasks/tools/install-kubectl/
Docker	19.03.15	Docker is installed on each SLES node.	#SUSEConnect -p sle-module-containers/15.2/x86_64 #zypper install docker
SLES15 SP2 nodes	SLES15 SP2	Ensure that the nodes are accessed using SSH and the required ports must be opened before the cluster installation.	https://rancher.com/docs/rke/latest/en/os/#ports
PowerFlex CSI	1.4	PowerFlex CSI is used from the Workstation VM.	https://github.com/dell/csi-vxflexos

Steps to Install the docker on the Linux workstation and the Kubernetes nodes

Note: All servers must have SUSE SLES license and must be able to connect to SUSE customer enter or a local RMT server. To update the servers to latest updates, run the zypper update.

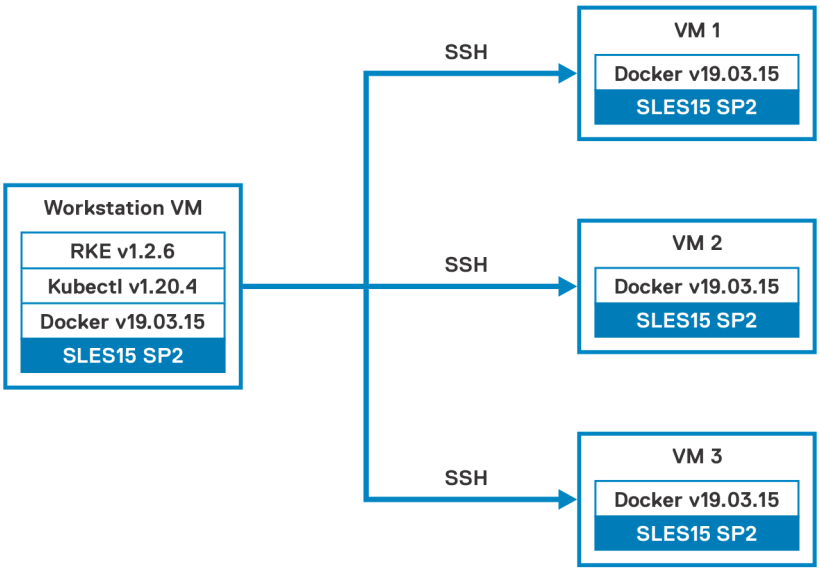
1. Run the following command to activate the containers module:

```
$ SUSEConnect -p sle-module-containers/15.2/x86_64
```

2. Run the following commands to Install the docker, enable and start the docker service:

```
$ zypper install docker
$ systemctl enable docker
$ systemctl start docker
$ systemctl status docker
```

The RKE binary is available for Windows and Linux operating systems. In this solution, RKE is run from a Linux workstation VM. RKE connects to the nodes using SSH key pairs.



Note: Make sure that the SSH login that is used for node access is a member of the docker group on the node.

3. Run the following command to create a Linux user account on every node:

```
$ useradd -m -G docker <user_name>
$ su - <user name>
$ mkdir $HOME/.ssh
$ chmod 600 $HOME/.ssh
$ touch $HOME/.ssh/authorized keys
```

4. Run the following command to test the docker socket access:

```
$ docker version
```

5. Run the following command from the Workstation VM where RKE binary exists to create an SSH key pair:

```
$ ssh-keygen
```

The following files are created after SSH key pairing:

```
$HOME/.ssh/id_rsa (SSH private key, keep this secure)
$HOME/.ssh/id rsa.pub (SSH public key)
```

6. Run the following command to copy the SSH public key to the Kubernetes nodes to provide access to the nodes, using the created SSH key pair:

```
$ cat .ssh/id rsa.pub | ssh node1 "cat >>
.ssh/authorized_keys"
```

7. Run the following command to test the SSH connectivity from the Workstation VM, replacing the 'hostname' with each of the Kubernetes nodes IP or hostname:

```
$ ssh -i $HOME/.ssh/id_rsa <user_name>@<hostname> docker
version
```

Installation of the SUSE Rancher server

SUSE Rancher recommends installing SUSE Rancher server on a Kubernetes cluster. In this white paper, a single-node Kubernetes cluster is created on the Linux workstation VM and then SUSE Rancher server is installed using the Helm package manager for Kubernetes. For high-availability, it is recommended to have a three-node Kubernetes cluster. Ensure that you are aware of the networking ports needed for the Kubernetes cluster to work. For more information, see Port Requirements.

Steps to install a single node Kubernetes cluster

1. Run the following command to download and install Helm:

```
Option1
$ curl
https://raw.githubusercontent.com/helm/helm/master/scripts
/get-helm-3 | bash
Option2
$ curl -sfL https://get.helm.sh/helm-v3.5.3-linux-
amd64.tar.gz -o helm.tgz
$ tar xf helm.tgz
$ mv linux-amd64/helm /usr/local/bin/
$ chmod +x usr/local/bin/helm
$ helm version
```

2. Run the following command to download and install kubectl.

```
$ curl -LO https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kube
ctl
$ chmod +x kubectl
$ mv kubectl /usr/local/bin/
$ kubectl version
```

3. Run the following command to download the latest RKE installer applicable to the operating system being used and keep the RKE binary in a `rke` user home directory `(/home/rke)`:

```
$ curl -LO
https://github.com/rancher/rke/releases/download/v1.2.6/rk
e_linux-amd64
```

4. Run the following command to move the `rke_linux-amd64` binary as `rke` and make the RKE binary executable and confirm:

```
$ mv
rke linux-
amd64 rke
$ chmod +x
rke
$ rke --
version
```

5. In order to deploy the cluster, you must create a configuration file, 'cluster.yml.' Create the `cluster.yml` file by running "`./rke config`" and answer the questions. This file contains all information that is required to build the Kubernetes cluster, such as node connection information and roles like controlplane, etcd, and worker to apply to each node. Setup as many nodes as needed, in this example, it runs as a single node:

```
$ rke config --name cluster.yml
[+] Cluster Level SSH Private Key Path [~/.ssh/id rsa]:
[+] Number of Hosts [1]:
[+] SSH Address of host (1) [none]: 192.168.153.111
[+] SSH Port of host (1) [22]:
[+] SSH Private Key Path of host (192.168.153.111) [none]:
[-] You have entered empty SSH key path, trying fetch from
SSH key parameter
[+] SSH Private Key of host (192.168.153.111) [none]:
[-] You have entered empty SSH key, defaulting to cluster
level SSH key: ~/.ssh/id rsa
[+] SSH User of host (192.168.153.111) [ubuntu]: tux
[+] Is host (192.168.153.111) a Control Plane host (y/n)?
[y]:
[+] Is host (192.168.153.111) a Worker host (y/n)? [n]: y
[+] Is host (192.168.153.111) an etcd host (y/n)? [n]: y
[+] Override Hostname of host (192.168.153.111) [none]:
[+] Internal IP of host (192.168.153.111) [none]:
[+] Docker socket path on host (192.168.153.111)
[/var/run/docker.sock]:
[+] Network Plugin Type (flannel, calico, weave, canal,
aci) [canal]:
[+] Authentication Strategy [x509]:
[+] Authorization Mode (rbac, none) [rbac]:
[+] Kubernetes Docker image [rancher/hyperkube:v1.20.4-
rancher1]:
[+] Cluster domain [cluster.local]:
[+] Service Cluster IP Range [10.43.0.0/16]:
[+] Enable PodSecurityPolicy [n]:
[+] Cluster Network CIDR [10.42.0.0/16]:
[+] Cluster DNS Service IP [10.43.0.10]:
[+] Add addon manifest URLs or YAML files [no]:
$
```

6. Run the following command to build the RKE cluster:

```
$ rke up
INFO[0000] Running RKE version: v1.2.6
INFO[0000] Initiating Kubernetes cluster
INFO[0000] [dialer] Setup tunnel for host
[192.168.153.111]
.
.
.
INFO[0861] Finished building Kubernetes cluster
successfully
```

7. Run the following command to configure the kubectl config file:

```
$ ls
bin  cluster.rkestate  cluster.yml
kube config cluster.yml public html  rke
$ mkdir .kube
$ cp kube_config_cluster.yml /$HOME/.kube/config
$ chmod 700 /home/tux/.kube/config
```

8. Check that the nodes are in a ready state after a successful Kubernetes cluster creation. The following output shows that all the nodes have both master and worker roles installed:

```
$ kubectl get nodes
NAME              STATUS   ROLES                      AGE
VERSION
192.168.153.111   Ready    controlplane,etcd,worker   48m
v1.20.4
```

Steps to install SUSE Rancher server on the Kubernetes cluster

1. Run the following command to add the Helm chart repository that contains charts to install Rancher:

```
$ helm repo add rancher-latest
https://releases.rancher.com/server-charts/latest
```

2. Run the following command to create a namespace for SUSE Rancher as cattle-system:

```
$ kubectl create ns cattle-system
```

3. Run the following command to create and apply a namespace for certificate manager as cert-manager:

```
$ kubectl create ns cert-manager
$ kubectl apply -f https://github.com/jetstack/cert-
manager/releases/download/v1.2.0/cert-manager.crds.yaml
```

4. Run the following command to add the Jetstack repo to helm. Jetstack cert-manager helps with management and issue of TLS certificates from various issuing sources:

```
$ helm repo add jetstack https://charts.jetstack.io
```

5. Run the following command to add the Rancher stable repo to helm and update helm:

```
$ helm repo add rancher-stable
https://releases.rancher.com/server-charts/stable
$ helm repo update
```

6. The cert-manager is a Kubernetes add-on to automate the management and issue of TLS certificates from various issuing sources. SUSE Rancher relies on cert-manager to issue certificates generated by SUSE Rancher CA or to request the encrypted certificates. Run the following command to use helm to install the cert-manager:

```
$ helm install cert-manager jetstack/cert-manager -n cert-
manager --version v1.2.0 –wait
```

7. Run the following command to check the cert-manager namespace for running pods to verify that it is deployed correctly:

```
$ kubectl get pods -n cert-manager
NAME                                       READY   STATUS
RESTARTS   AGE
cert-manager-75cf57777c-ztw9f              1/1     Running
0          2m2s
cert-manager-cainjector-f54c57bf8-wkc2z    1/1     Running
0          2m2s
cert-manager-webhook-76794c6967-84gb6      1/1     Running
0          2m2s
```

8. Run the following command to create a configuration file (`rancher-values.yaml`) for SUSE Rancher server, specifying the hostname and other details. In the following example, `rancher sles15sp2` is the hostname:

```
$ cat << EOF > rancher-values.yaml
hostname: rancher sles15sp2.testlab.com
replicas: 1
EOF
```

9. For more information about configurable options, see Rancher Helm Chart Options.

10. Run the following command to install SUSE Rancher with Helm:

```
$ helm install rancher rancher-stable/rancher -n cattle-
system --version v2.5.7 -f rancher-values.yaml
NAME:   rancher
LAST DEPLOYED: Tue Mar 16 11:05:11 2021
NAMESPACE: cattle-system
STATUS: DEPLOYED
..
..
NOTES:
Rancher Server has been installed.

NOTE: Rancher may take several minutes to fully
initialize. Please standby while Certificates are being
issued and Ingress comes up.

Check out Rancher docs at
https://rancher.com/docs/rancher/v2.x/en/

Browse to https://rancher sles15sp2.testlab.com

Happy Containering!
$
```

11. Run the following command to check the cattle-system namespace for running pods to verify that the namespace is deployed correctly:

```
$ kubectl get pods --namespace cattle-system
NAME                               READY   STATUS
RESTARTS   AGE
rancher-7f4df87477-mfcxc           1/1     Running   1
36d
rancher-webhook-b5b7b76c4-r9nwn    1/1     Running   1
36d
```

Result: Rancher is up and running.

12. Go to https://rancherssles15sp2.testlab.com to access the functional SUSE Rancher server.

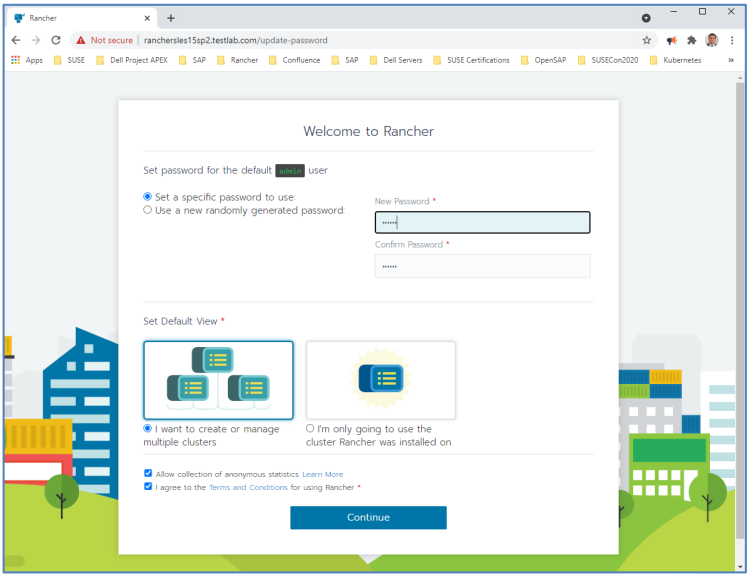
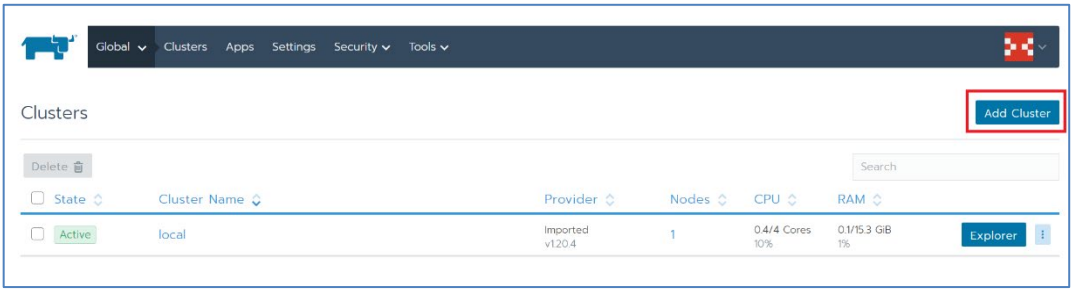


Figure 5. SUSE Rancher server

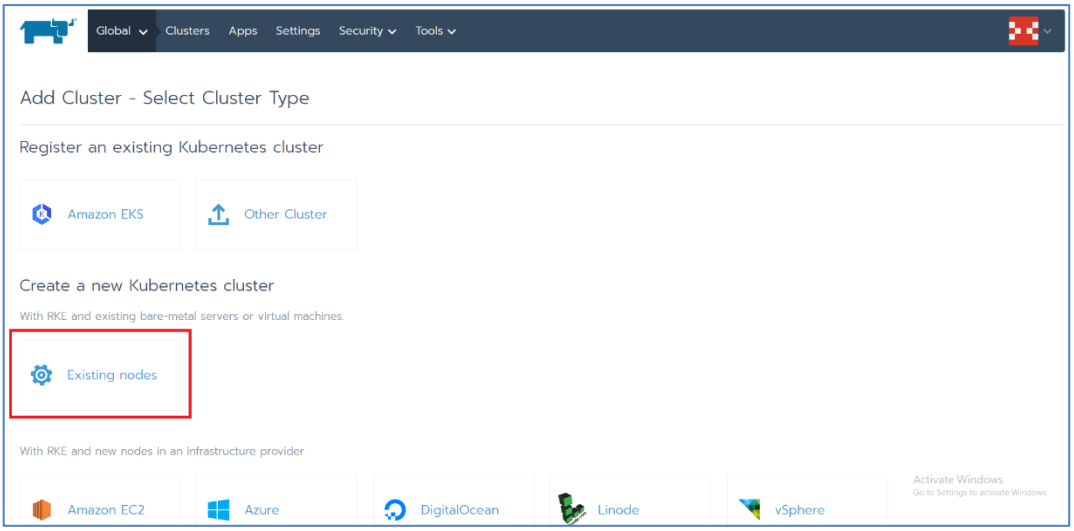
Steps to install Kubernetes workload cluster using SUSE Rancher UI

Perform the following steps to deploy an RKE Kubernetes cluster using the SUSE Rancher Server UI:

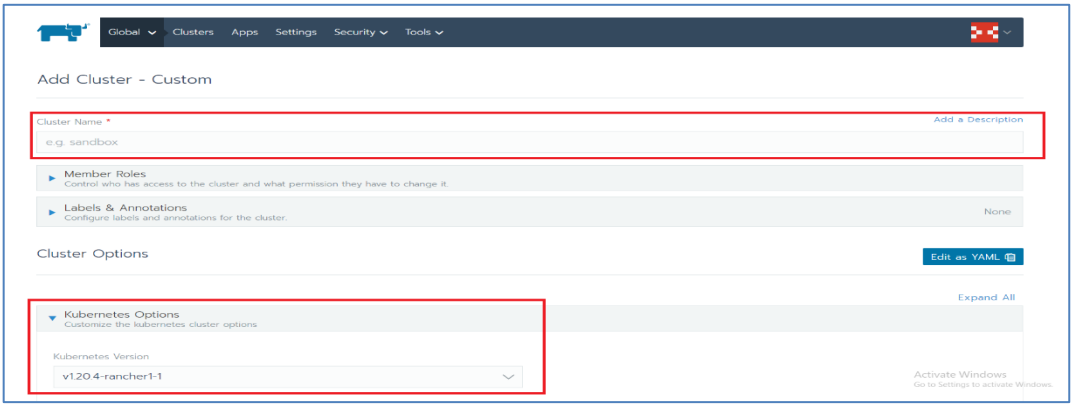
1. Log in to the Rancher Server from the browser.
2. Click **Add Cluster** to create a cluster



3. In the Cluster type, click **Existing nodes**.

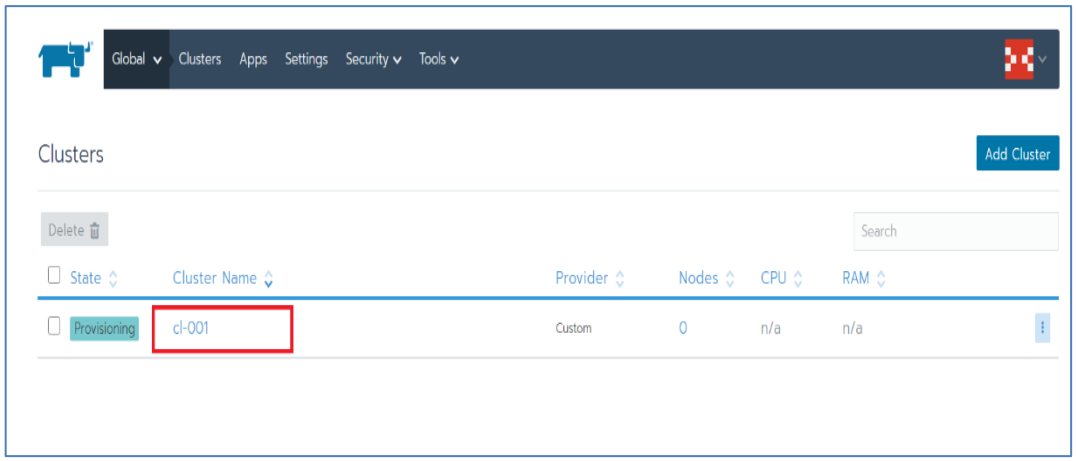


4. Provide a name for the cluster and select the required Kubernetes version and click **Next**.

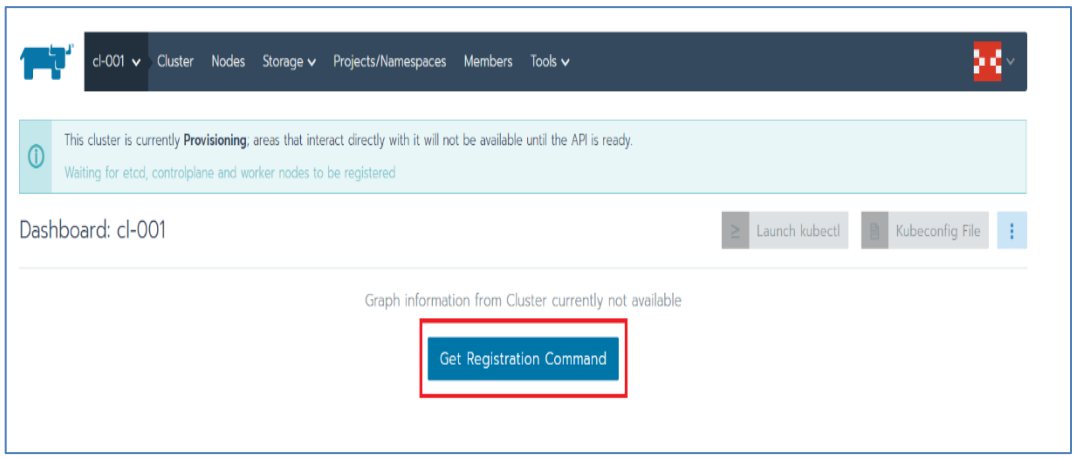


The new cluster is registered and is displayed in the **Cluster** tab.

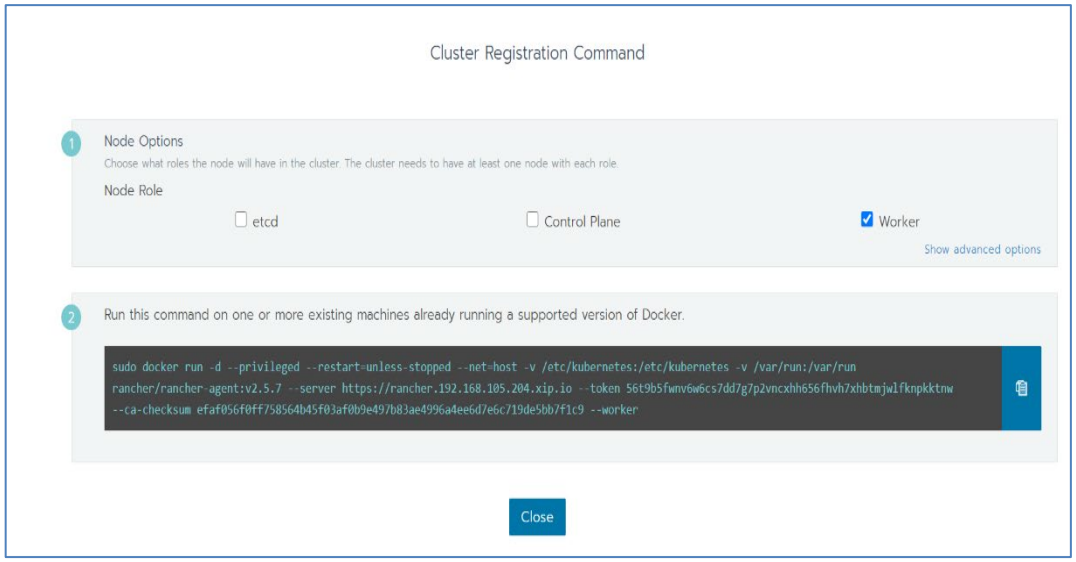
5. Click on the newly created cluster name.



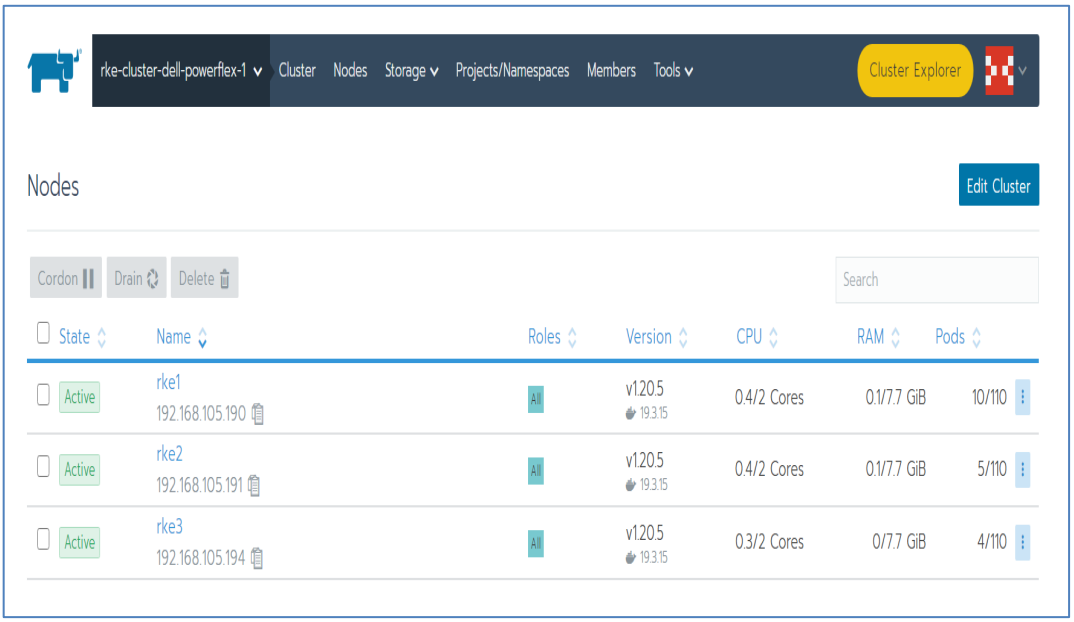
6. Click **Get Registration Command**.



7. Select the required node role and copy the command.



8. Connect to the Kubernetes node using SSH as a predefined user, for example: Root. Paste the command that is copied in the previous step and press **Enter**. The node is registered with the cluster. Repeat this step for each Kubernetes node with the required node role. Once all the nodes are registered, it is displayed under the **Nodes** tab of the cluster.



PowerFlex Container Storage Interface driver

Overview

The CSI driver for PowerFlex is a plug-in that is installed in the Kubernetes to provide persistent storage using PowerFlex storage system. The CSI driver for PowerFlex and Kubernetes communicate using the CSI protocol. The CSI driver for PowerFlex supports PV capabilities, Dynamic and Static PV provisioning, and Snapshot capabilities.

The Helm chart installs the CSI driver for Dell EMC PowerFlex using a shell script. This script installs the CSI driver container image along with the required Kubernetes sidecar containers.

The controller section of the Helm chart installs the following components in a single deployment in the namespace vxflexos:

- CSI driver for Dell EMC PowerFlex.
- Kubernetes Provisioner, which provisions the volumes.
- Kubernetes Attacher, which attaches the volumes to the containers.
- Kubernetes Snapshotter, which provides snapshot support.
- Kubernetes External Resizer, which resizes the volume.

The node section of the Helm chart installs the following components in a daemon set in the namespace vxflexos:

- CSI driver for Dell EMC PowerFlex.
- Kubernetes Node Registrar, which handles the driver registration.

Prerequisites

Before you install the CSI Driver for PowerFlex, verify that the following requirements are met:

- Install the Kubernetes v1.18 +.
- Configure the docker for bi-directional mount (MountFlags=shared).
- Install Helm 3.
- Enable Zero Padding on PowerFlex.
- Install PowerFlex SDC on all Kubernetes nodes.
- Create a Kubernetes secret for PowerFlex credentials.
- Obtain the Helm values for options like PowerFlex system name or ID, default gateway, and MDM IP addresses, and default storage pool.
- Ensure that the volume snapshot requirements are met.

For more information, see GitHub.

Steps to install CSI drives for PowerFlex

1. Run the following command to download the installation source files from GitHub:

```
$ git clone https://github.com/dell/csi-vxflexos
```

2. Run the following command to create the namespace called **vxflexos**:

```
$ kubectl create namespace vxflexos
```

3. Collect information from the PowerFlex SDC by running the **get_vxflexos_info.sh** script.
4. Copy the **csi-vxflexos/values.yaml** into a file called **myvalues.yaml** in the same directory as the **csi-install.sh** script.
5. Edit **myvalues.yaml** to set the parameters like file system types, volume name prefix, and controller count, for the installation.
6. Create a config.json for driver configuration. This file contains information like the PowerFlex system IP details and credentials.
7. Run the following **sh csi-install.sh** command to proceed with the installation:

```
$ sh csi-install.sh --namespace vxflexos –values
myvalues.yaml
```

8. Run the following commands to check the **vxflexos** namespace for running pods to verify it is deployed correctly:

```
$ kubectl get pods -n vxflexos
NAME                                   READY   STATUS
RESTARTS   AGE
vxflexos-controller-5855b6969f-jbbpq   5/5     Running   0
15d
vxflexos-node-6gnlc                    2/2     Running   0
15d
vxflexos-node-vswl2                    2/2     Running   0
15d
vxflexos-node-zr2r4                    2/2     Running   0
15d
$
```

For more information about CSI driver installation, see GitHub.

9. Create the storage classes through the Rancher graphical UI. For more information about storage class creation, see Storage Classes.
10. Ensure that the PowerFlex CSI driver is running on a Rancher Kubernetes cluster and the PowerFlex Storage classes are listed in the SUSE Rancher menu for any application that is launched in the SUSE Rancher Kubernetes cluster.

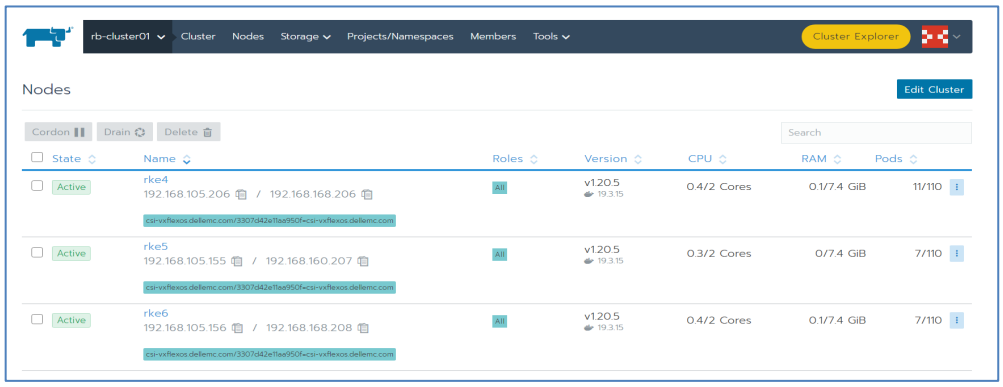


Figure 6. SUSE Rancher menu

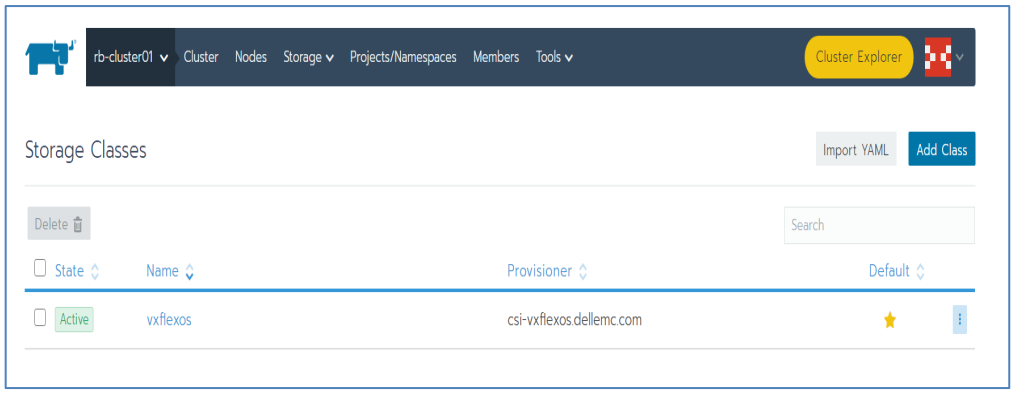


Figure 7. Cluster explorer

11. Test the deployment workflow of a simple pod on PowerFlex storage, For more information, see Test PowerFlex CSI Driver.

12. Verify that the persistent volume is created in PowerFlex cluster using the Power Flex graphical user interface.

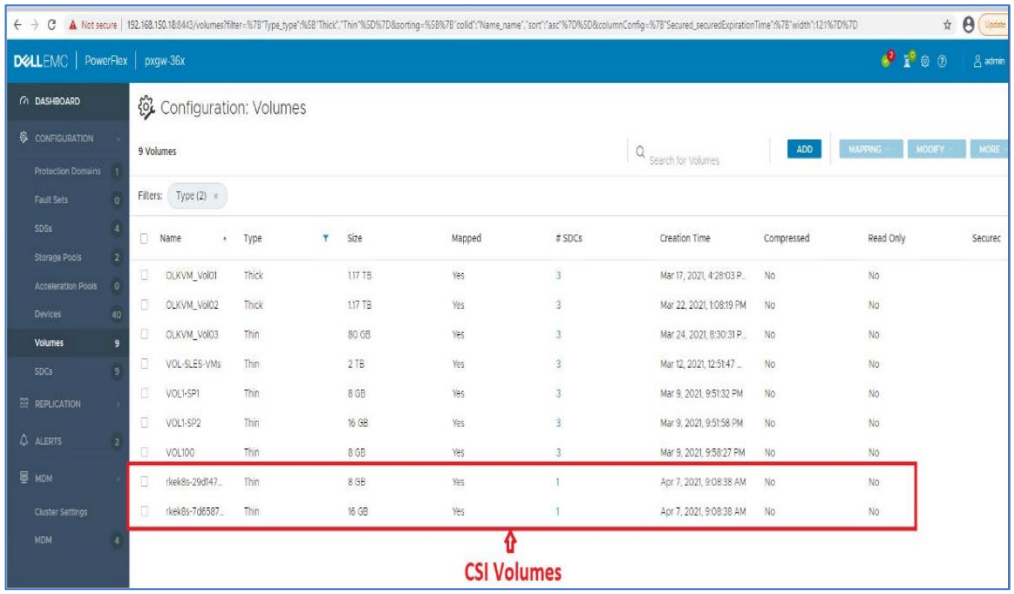


Figure 8. PowerFlex graphical user interphase

13. Verify the following output from SUSE Rancher UI and confirm that PowerFlex Persistent Volumes are used by the Kubernetes pods running under the test namespace:

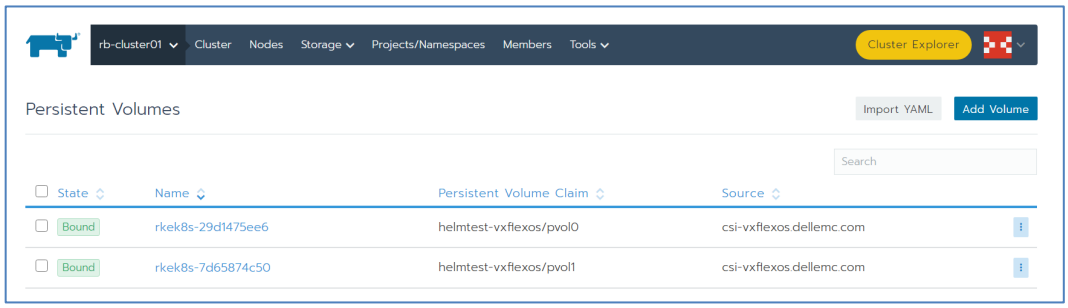


Figure 9. Kubernetes pods running under the test namespace

Rancher Kubernetes Cluster Data Protection using PowerProtect Data Manager

PowerProtect Data Manager protects Kubernetes workloads and ensures that the data is protected and recoverable. PowerProtect Data Manager is deployed using an Open Virtualization Appliance (OVA) or a machine image and is integrated with PowerProtect DD series appliances as protection storage for backups. For more information about PowerProtect Data Manager deployment methods, see the Dell EMC PowerProtect Data Manager Deployment Guide.

PowerProtect Data Manager can be integrated with SUSE Rancher managed Kubernetes cluster through Kubernetes APIs to discover protectable resources such as namespaces and PVCs. 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.

The following high-level architecture diagram shows the data protection for SUSE Rancher-managed RKE downstream single node and downstream cluster with PowerProtect Data Manager:

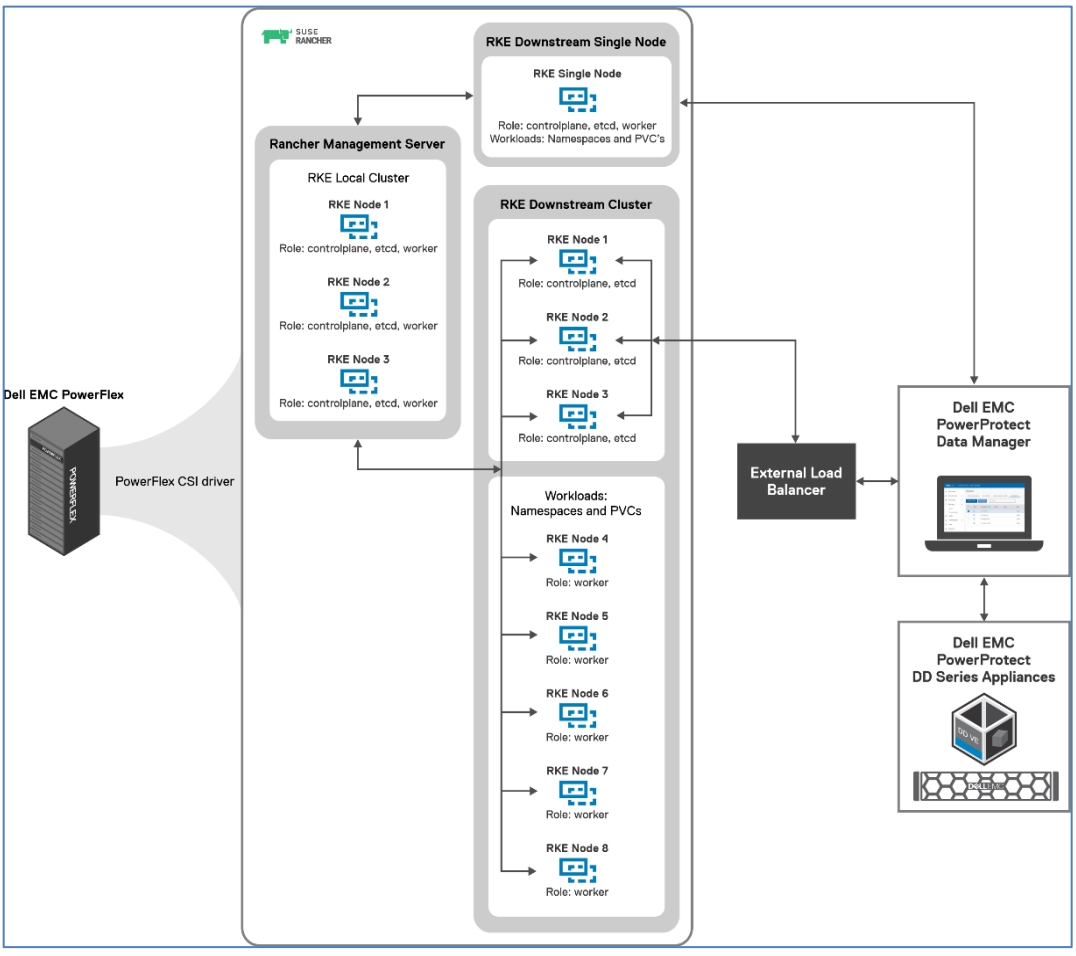


Figure 10. High-level RKE downstream single node and downstream cluster data protection overview with PowerProtect Data Manager

Once the Kubernetes cluster is added as an asset source in PowerProtect Data Manager and the discovery is complete, the associated namespaces are available as assets for protection. PowerProtect Data Manager protects the following two types of Kubernetes cluster assets - Namespaces and PersistentVolumeClaims (PVCs)**.**

During the discovery process, PowerProtect Data Manager creates the following namespaces in the cluster:

- **Velero-ppdm**: This namespace contains a Velero pod to backup metadata and stage to target storage in bare-metal environments. It performs PVC snapshot and metadata backup for a VMware cloud native storage.
- **PowerProtect**: This namespace contains a PowerProtect controller pod to drive persistent volume claim snapshot and backup, and send the backups to the target storage using dynamically deployed cProxy pods.

Kubernetes uses persistent volumes to store persisted application data. Persistent volumes are created on an external storage and then attached to a particular pod using PVCs. PVCs are included along with other namespaces in PowerProtect Data Manager backup and recovery operations.

Integrating SUSE Rancher managed Kubernetes cluster with PowerProtect Data Manager

PowerProtect Data Manager integrates with SUSE Rancher managed Kubernetes cluster for data protection in the following ways:

- Directly connecting to the RKE downstream single node with controlplane and etcd roles.
- Through a load balancer, when there are multiple RKE nodes with controlplane and etcd roles in an RKE downstream cluster.

The following diagram shows the SUSE Rancher-managed RKE downstream cluster with three RKE nodes, RKE – Node 1, RKE – Node 2, and RKE – Node 3:

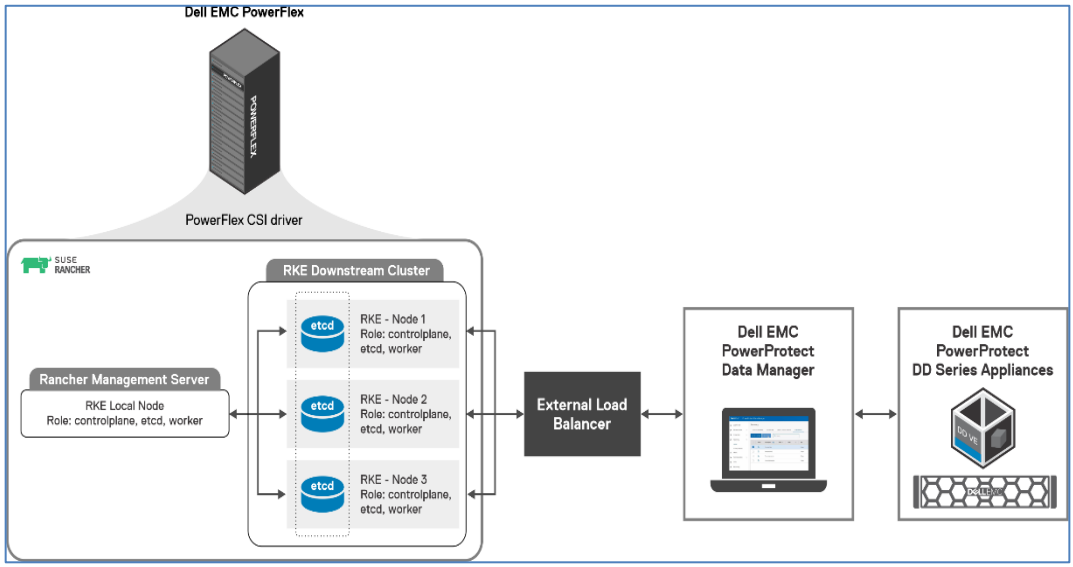


Figure 11. High-level integration overview of RKE downstream cluster with PowerProtect Data Manager

Each RKE node holds all roles such as controlplane, etcd, and worker that is managed by the Rancher management server. PowerFlex is the default storage class for the Kubernetes cluster workloads that are integrated through the PowerFlex CSI driver. An external load balancer is configured as the front-end cluster endpoint for the RKE nodes. The PowerProtect Data Manager accesses the integration and discovery of the RKE downstream cluster assets for data protection using the load balancer Virtual IP.

Note: Figure 11 is the high-level integration architecture for this white paper. Use Figure 10 as a best practice reference architecture.

RKE supports x509 authentication strategy, and also a list of SANs can be defined to add to the Kubernetes API Server PKI certificates. The optional load balancer configuration is done when there are multiple RKE nodes available with controlplane and etcd in the RKE downstream cluster. For example, you can connect to a Kubernetes cluster API server through a load balancer instead of a single RKE node.

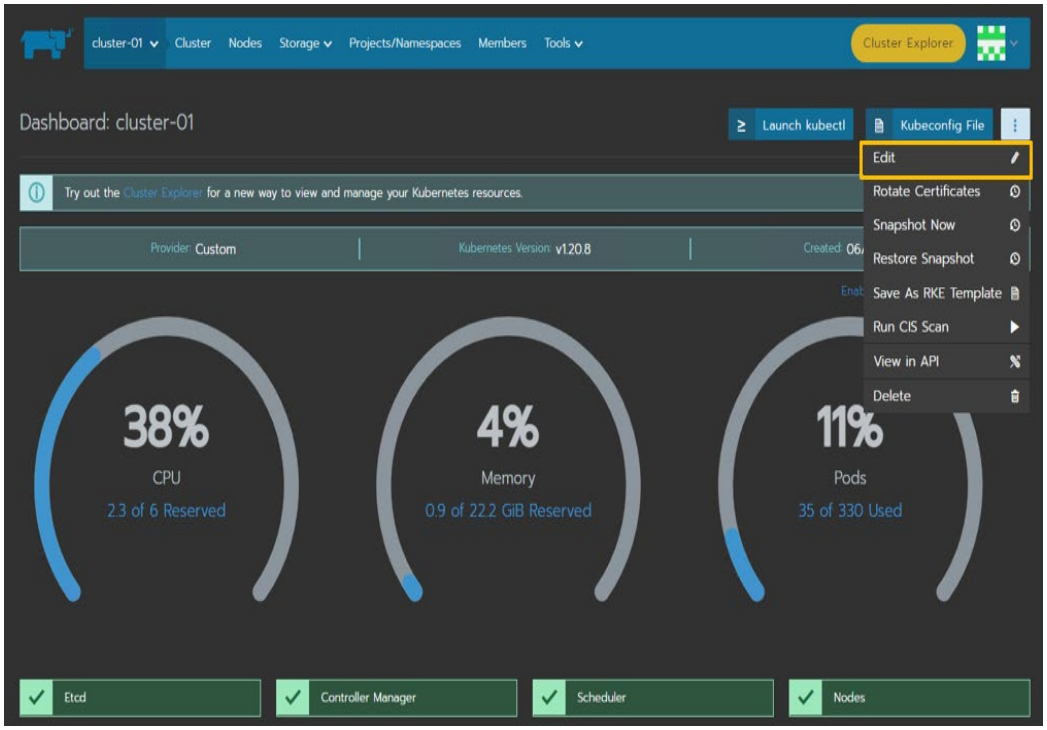
Pre-requisites

Before integrating the RKE downstream cluster with PowerProtect Data Manager, ensure that the following requirements are configured.

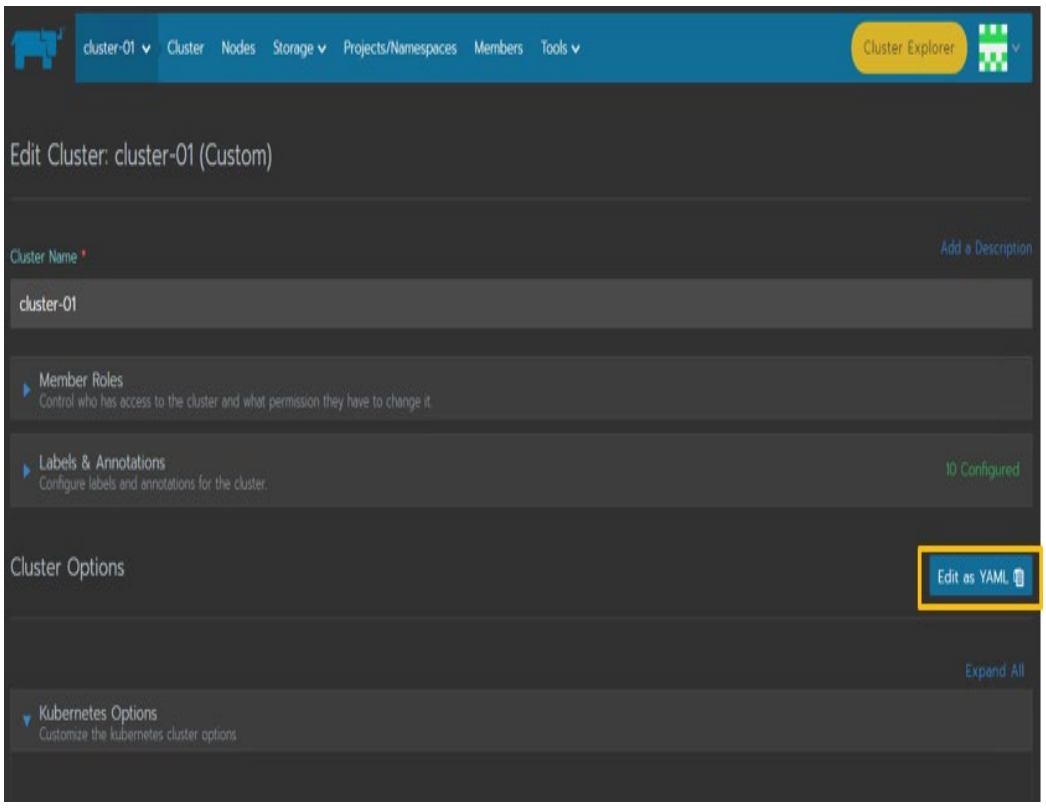
Load balancer authentication

Perform the following steps to add the load balancer details to the RKE downstream cluster:

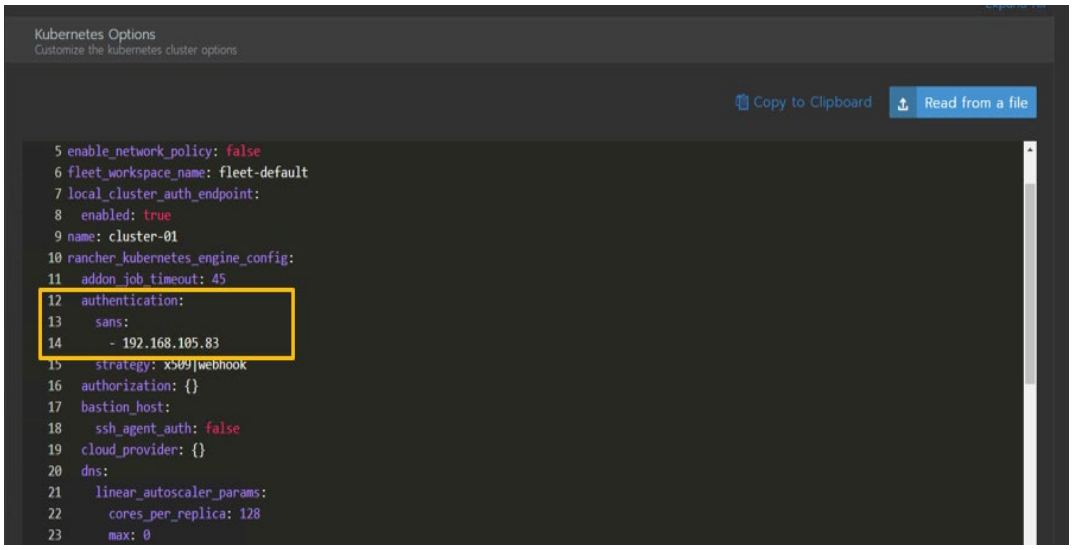
1. From the custom cluster dashboard, click **Edit** to edit the cluster configuration.



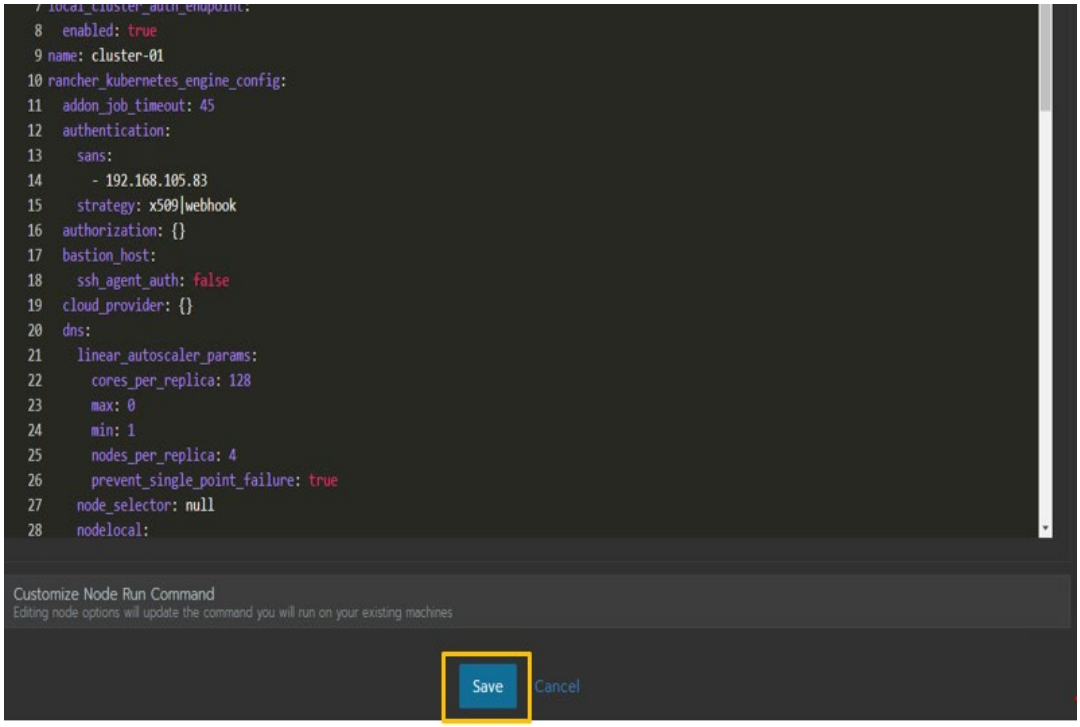
2. Click **Edit as YAML**.



3. Enter the load balancer IP or FQDN below the authentication section.



4. Click **Save** to save the configuration file. The cluster takes a few minutes to update the configuration.



Service account token for the Rancher Kubernetes cluster

The service account must have the following privileges:

- Get, Create, Update, and List for CustomResourceDefinitions.
- Get, Create, and Update ClusterRoleBinding for 'cluster-admin' role.
- Create and Update for the PowerProtect namespace.
- Get, List, Create, Update, Delete, and List.
- Get, List, Create, Update, and Delete for all kinds of resources inside the PowerProtect namespace.
- Get, List, and Watch for all the namespaces in the cluster and PV, PVC, storageclass, deployments, and pods in all the namespaces.

Note: The admin-user service account in the kube system namespace contains all the privileges. You can use the token of this account or an existing similar service account. Alternatively, you can create a service account that is bound to a cluster role that contains these privileges and provide the token of this service account.

- Run the following commands to create a service account for registering the Kubernetes cluster with PowerProtect Data Manager:

```
kubectl create serviceaccount dashboard -n default
kubectl create clusterrolebinding dashboard-admin -n
default --clusterrole=cluster-admin --
serviceaccount=default:dashboard
```

- Run the following commands to obtain the service account token:

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

Enabling the asset source

The Kubernetes asset source must be enabled on PowerProtect Data Manager UI before adding and registering the asset source for the protection of assets.

Perform the following steps to enable the asset source:

1. From the PowerProtect Data Manager UI, go to **Infrastructure → Asset Sources**, and click **+** icon to reveal the **New Asset Source** tab.
2. In the **New Asset Source** window, click **Enable Source** on the **Kubernetes** section.

Adding RKE downstream cluster with PowerProtect Data Manager as an asset source

You can use the PowerProtect Data Manager to protect the Kubernetes environment by adding an RKE downstream cluster that is managed by the SUSE Rancher as an asset source and discovering namespaces as assets for data protection operations.

A Kubernetes cluster can be added as an asset source in PowerProtect Data Manager to protect the namespaces and PVCs within the cluster.

Specifying the Kubernetes cluster attributes

- **Name**: Kubernetes cluster name to display in asset sources list.
- **Address**: FQDN or the IP address of the Kubernetes API server or Load Balancer.
- **Port**: Port to use for communication when not using the default port, 443

Note: The use of any port other than 443 or 6443 requires opening the port on PowerProtect Data Manager.

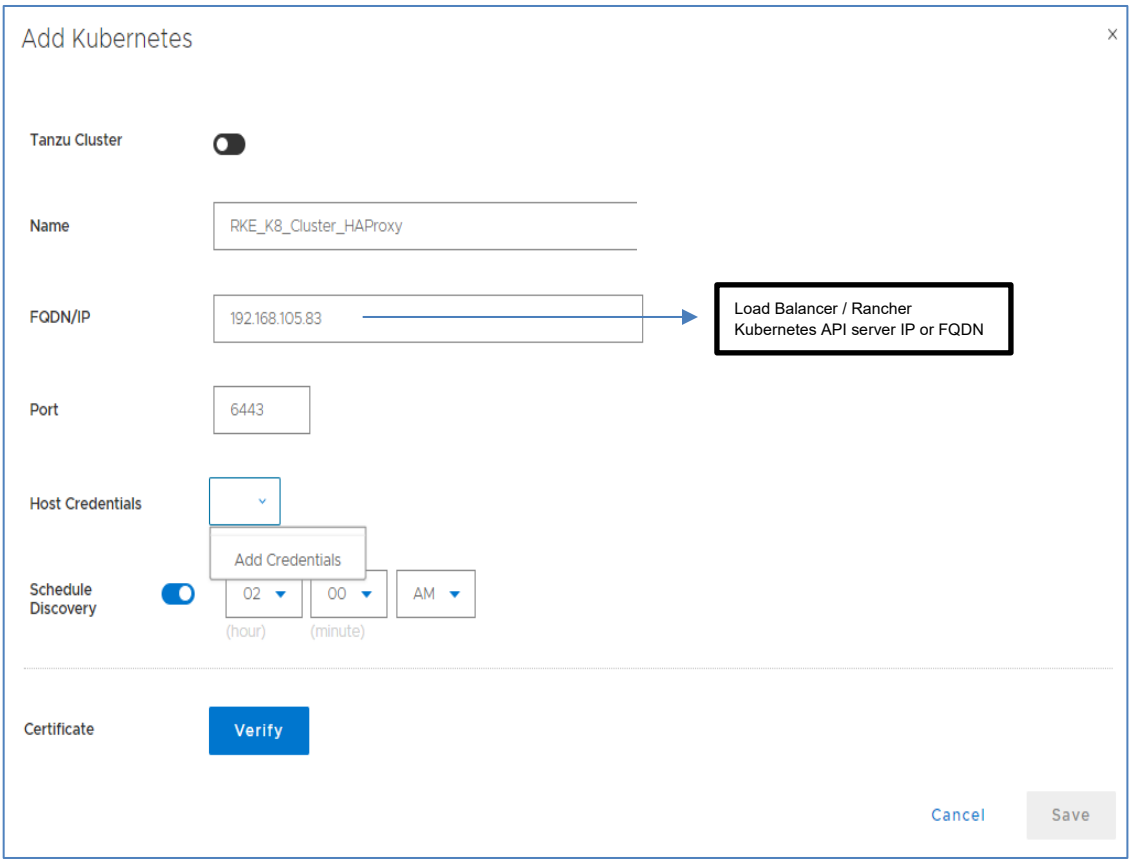
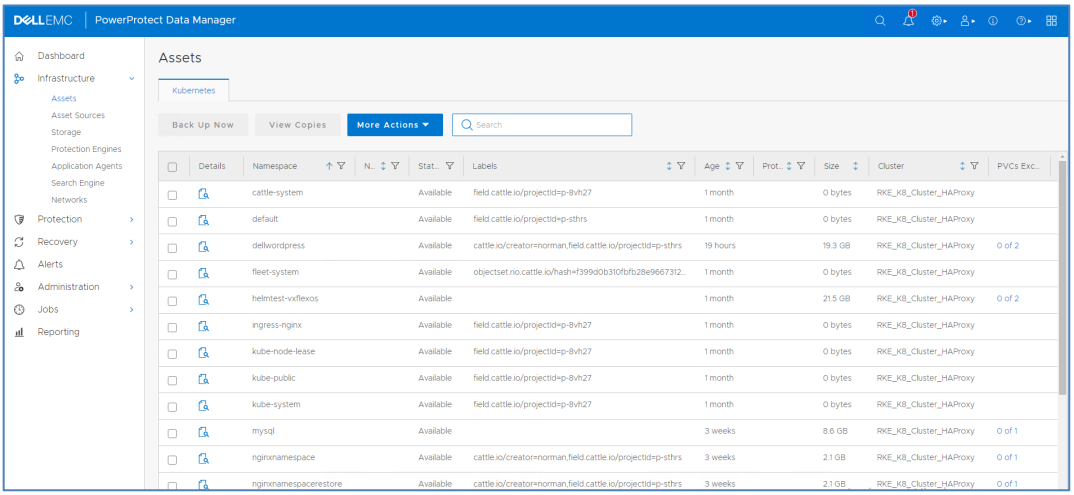


Figure 12. Kubernetes cluster attributes

When you add an RKE downstream cluster, the cluster information appears in the table, and automatic discovery of the namespace assets is initiated.

Discovering the Rancher Kubernetes cluster in PowerProtect Data Manager

Once the discovery is successful, the Kubernetes workloads (namespace and PVCs) appear as assets in the **Kubernetes** tab in the **Assets** window.

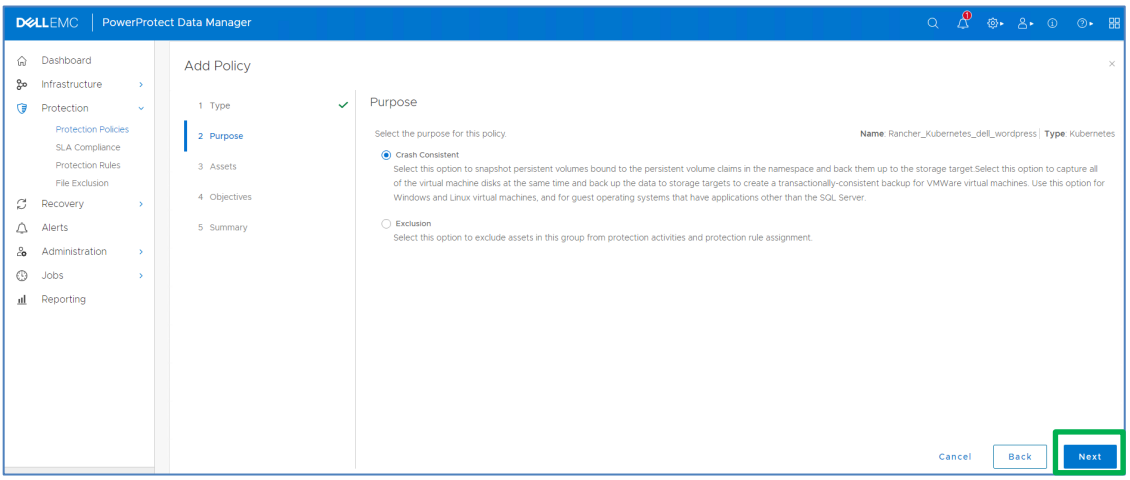


Protecting the namespaces and PVCs on the Rancher Kubernetes cluster

Protection policies define a set of objectives that apply to specific periods of time. These objectives drive configuration, active protection, and copy-data-management operations that satisfy the business requirements for the specified data. Each plan type has its own set of user objectives. Users with the system admin role can create protection policies.

PowerProtect Data Manager provides the following options when creating a Kubernetes cluster protection policy:

- **Crash Consistent**: Select this type for point-in-time backup of namespaces.
- **Exclusion**: Select this type if there are assets within the protection policy to exclude from data protection operations.



Also, the admin can select namespaces and associated PVCs statically or dynamically for inclusion or exclusion in protection policies, along with schedules, retention, and other protection operations.

For more information about creating a protection policy, see Dell EMC PowerProtect Data Manager Administration and User Guide.

Performing backup of namespaces and PVCs on Rancher Kubernetes cluster

Apart from the scheduled backup option, PowerProtect Data Manager supports the option to perform manual backups. With PowerFlex CSI, only full backup type is supported for Kubernetes protection.

For more information about performing backup and recovery of Kubernetes workloads using PowerProtect Data Manager, see Dell EMC PowerProtect Data Manager Administration and User Guide.

For example, a sample WordPress application is deployed on the Rancher Kubernetes cluster with the namespace **dellwordpress**.

The namespace is composed of two pods - WordPress application and mariadb pod as shown in the following sample:

```
rancher-ui:~ # kubectl get all -n dellwordpress
NAME                                READY   STATUS    RESTARTS   AGE
pod/wordpress-7dc5f64d5f-pq6cr      1/1     Running   0          18d
pod/wordpress-mariadb-0             1/1     Running   0          18d

NAME                        TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)                      AGE
service/wordpress           NodePort    10.43.135.14   <none>        80:31556/TCP,443:30969/TCP   18d
service/wordpress-mariadb   ClusterIP   10.43.129.68   <none>        3306/TCP                     18d

NAME                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/wordpress   1/1     1            1           18d

NAME                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/wordpress-7dc5f64d5f   1         1         1       18d

NAME                                 READY   AGE
statefulset.apps/wordpress-mariadb   1/1     18d
```

Figure 13. WordPress application pods in dellwordpress namespace

The following Rancher UI shows the WordPress application in the namespace **dellwordpress** that is accessed using the nodeports:

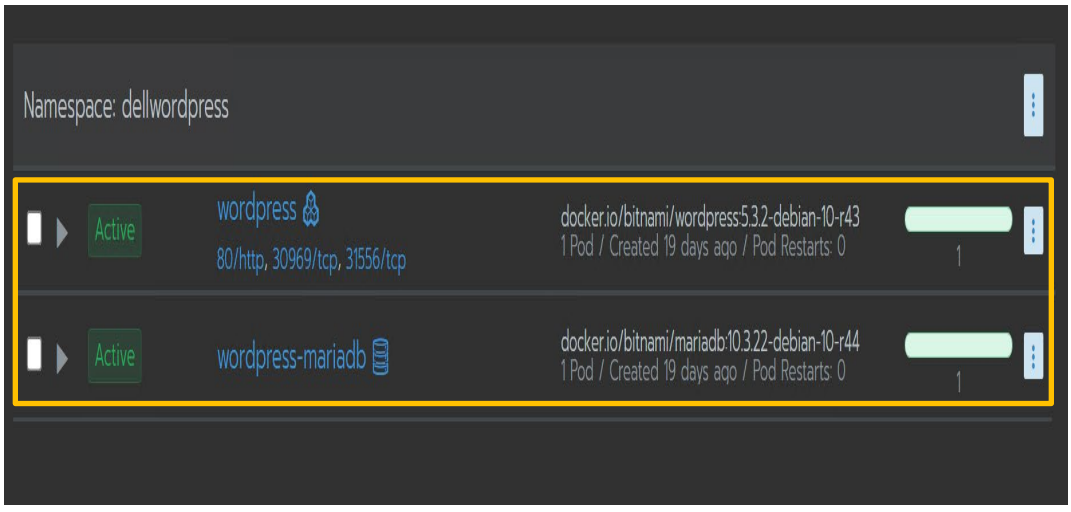


Figure 14. WordPress application pods in dellwordpress namespace from Rancher UI

The following sample shows a post that is published in the WordPress application that can be accessed with the nodeport **31556:**

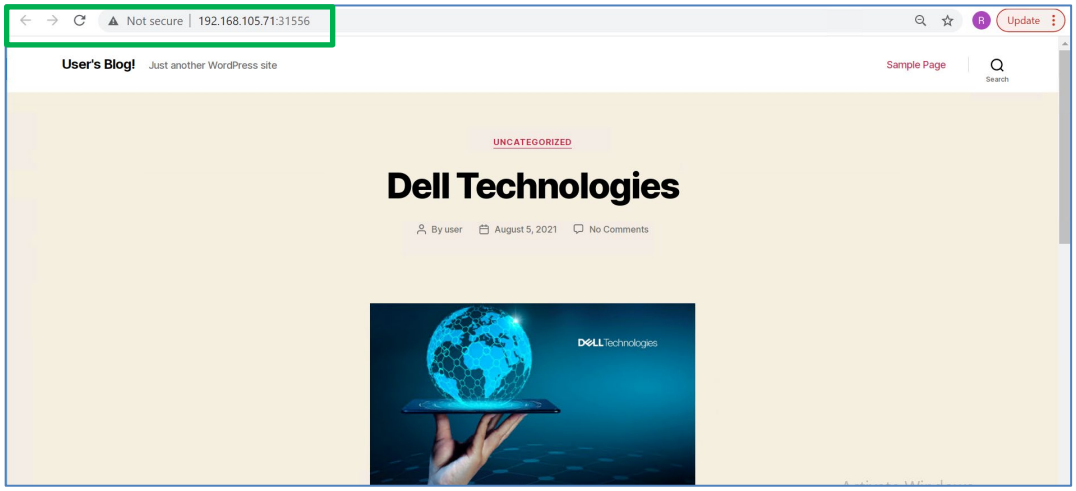


Figure 15. Published post in WordPress

The following image shows the protection job in progress for the **dellwordpress** namespace and respective PVCs:

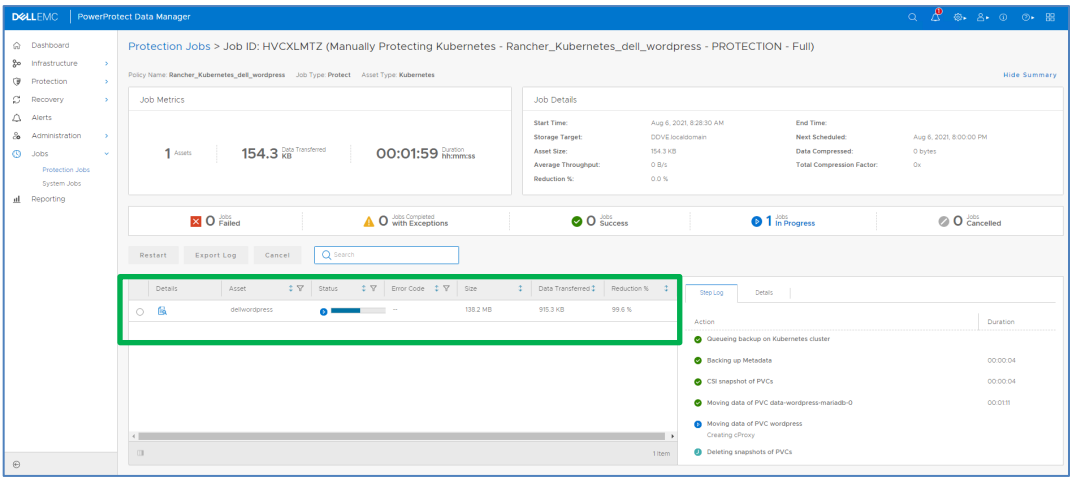


Figure 16. PowerProtect Data Manager protection job in progress

The following image shows the protection job that is completed successfully for the **dellwordpress** namespace:

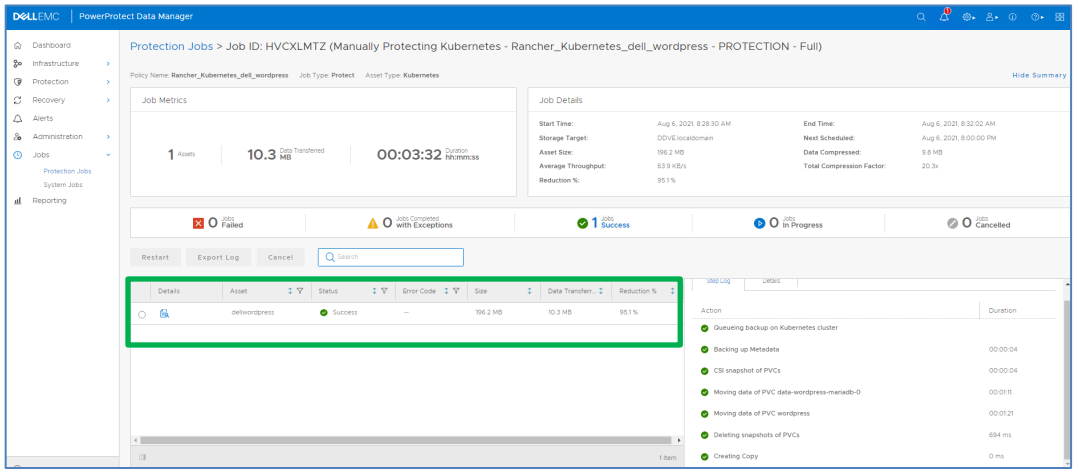


Figure 17. PowerProtect Data Manager protection job complete

The following image shows the protection copy available for the namespace **dellwordpress**:

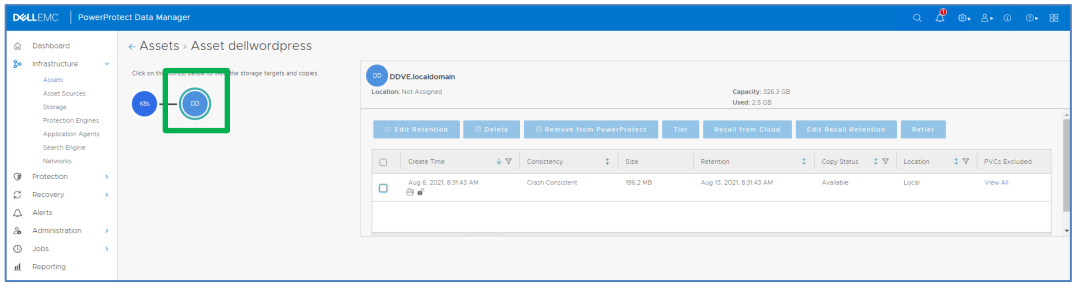


Figure 18. PowerProtect Data Manager asset window displaying the available protection copy

Restoring the namespaces and PVCs to Rancher Kubernetes cluster

After protection under the Kubernetes cluster protection policy, restore of namespace and PVCs can be done from individual namespace backups.

Use the following recovery options:

- **Restore to original namespace**: Restore to the original namespace on the original cluster.
- **Restore to new namespace**: Create a namespace and restore to this location on the original cluster or a different cluster.
- **Restore to existing namespace**: Restore to an existing namespace in the original cluster or a different cluster.

In the following sample restore, the PVCs that are backed up within the namespace **dellwordpress** are restored to the new namespace **dellwordpressrestore:**

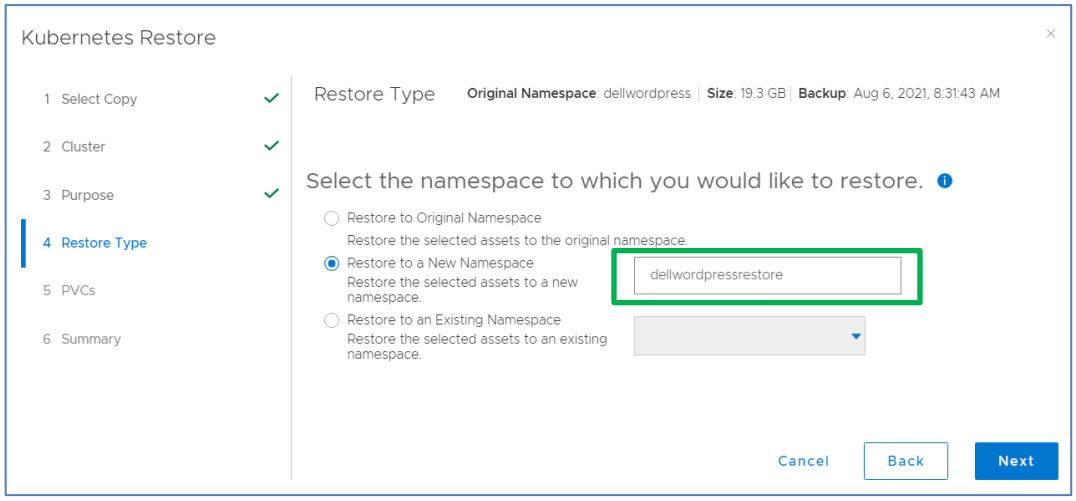


Figure 19. Restore type window

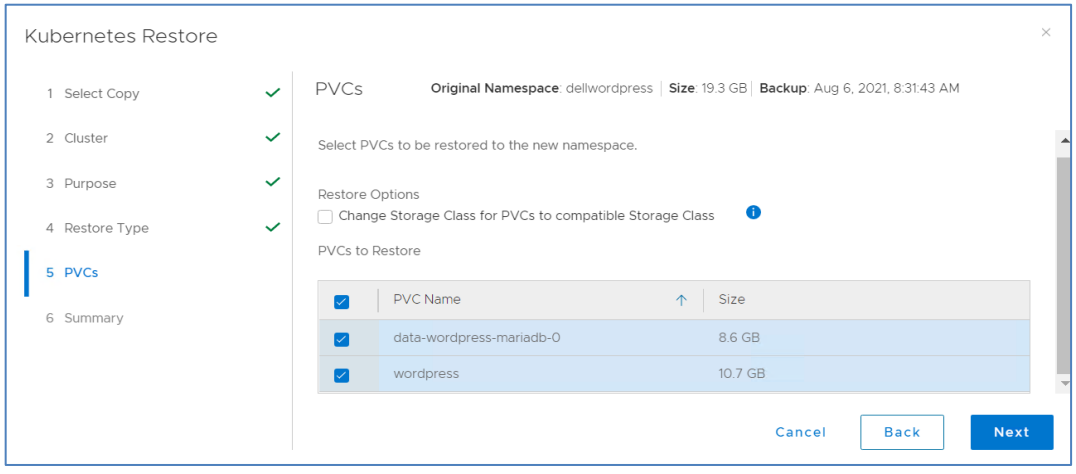


Figure 20. Restore to new namespace recovery

The following image shows that the restore job is successful:

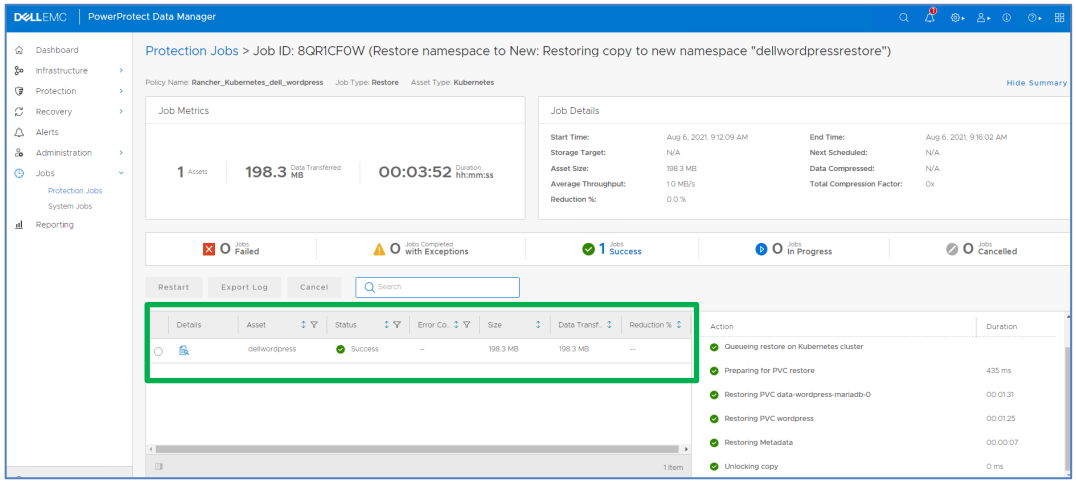


Figure 21. PowerProtect Data Manager protection restore job window

The following sample shows the new namespace **dellwordpressrestore** created successfully during the restore:

```
rancher-ui:~ # kubectl get ns
NAME                      STATUS   AGE
cattle-system             Active   55d
default                   Active   55d
dellwordpress             Active   18d
dellwordpressrestore      Active   11m
fleet-system              Active   55d
helmtest-vxflexos         Active   53d
ingress-nginx             Active   55d
kube-node-lease           Active   55d
kube-public               Active   55d
kube-system               Active   55d
mysql                     Active   43d
nginxnamespace            Active   42d
nginxnamespacerestore     Active   42d
nginxnsrestorehaproxy     Active   20d
powerprotect              Active   48d
security-scan             Active   55d
velero-ppdm               Active   48d
vxflexos                  Active   53d
```

The following sample shows the WordPress application pods are successfully restored to the new namespace **dellwordpressrestore.** The restored WordPress application is accessed with the nodeport **31640** as shown in the following sample**:**

```
rancher-ui:~ # kubectl get all -n dellwordpressrestore
NAME                                READY   STATUS    RESTARTS   AGE
pod/wordpress-7dc5f64d5f-pq6cr      1/1     Running   0          6m2s
pod/wordpress-mariadb-0             1/1     Running   0          6m2s

NAME                        TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)
service/wordpress           NodePort    10.43.20.72     <none>        80:31640/TCP,443:32563/TCP
service/wordpress-mariadb   ClusterIP   10.43.152.108   <none>        3306/TCP

NAME                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/wordpress   1/1     1            1           6m2s

NAME                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/wordpress-7dc5f64d5f   1         1         1       6m2s

NAME                                 READY   AGE
statefulset.apps/wordpress-mariadb   1/1     6m1s
```

The following image shows the restored WordPress application in the namespace **dellwordpressrestore** in the Rancher management UI:

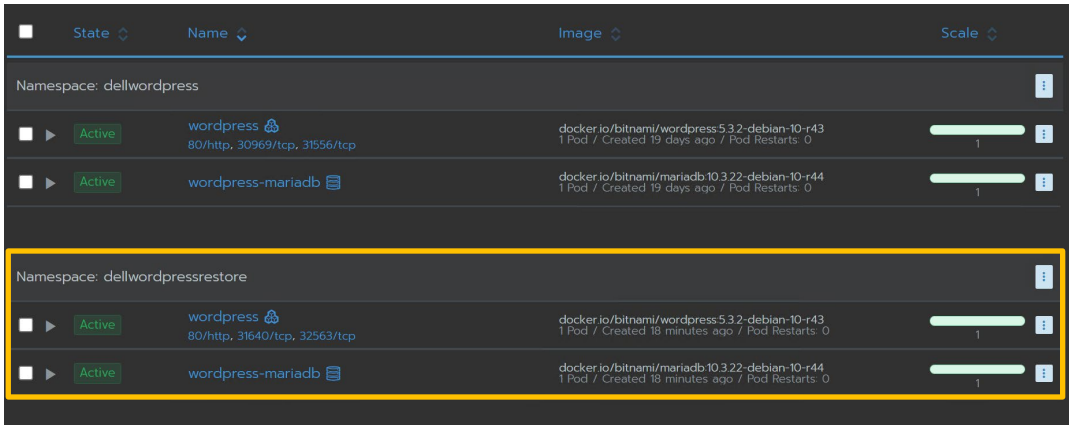


Figure 22. WordPress application pods in dellwordpressrestore namespace from Rancher UI

The following image shows the same WordPress application along with the post that is published with the namespace **dellwordpress**, on the restored namespace **dellwordpressrestore**, that is accessed with the nodeport **31640:**

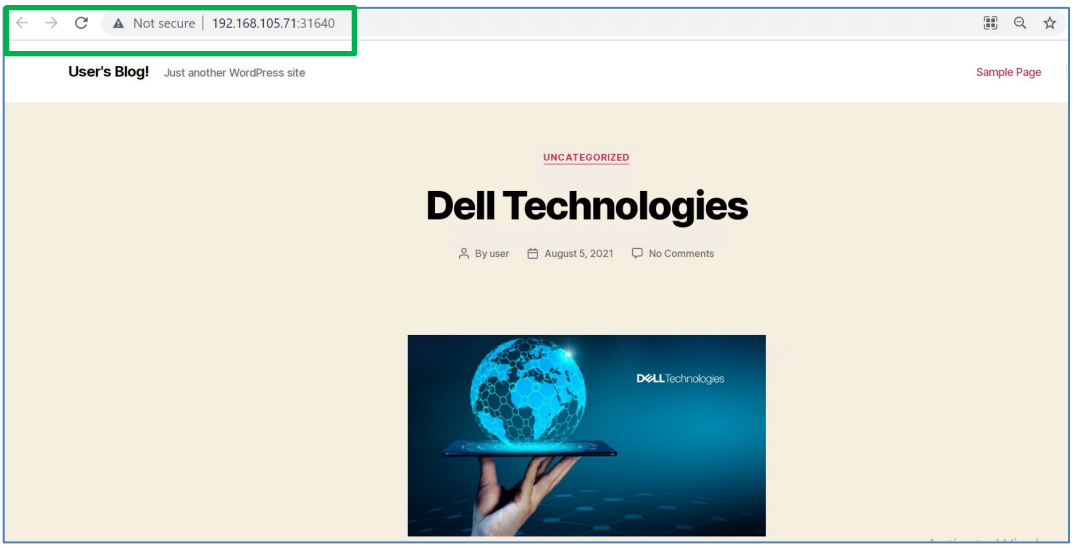


Figure 23. WordPress post available on the restored namespace

This confirms the successful backup and recovery of namespaces and PVCs that are available in the RKE downstream cluster using PowerProtect Data Manager.

For more information about performing backup and recovery of Kubernetes workloads using PowerProtect Data Manager, see Dell EMC PowerProtect Data Manager Administration and User Guide.

Conclusion

The Kubernetes eco sphere is growing rapidly providing more stability, security, and automatic service discovery. Dell Technologies is helping to streamline deployment and configuration operations, include Kubernetes cluster setup on PowerEdge, dynamically provision persistent storage using SUSE Rancher and CSI with PowerFlex, and easily integrate data protection with PowerProtect Data Manager. This empowers Kubernetes admins to deploy environments at a fast pace for developers and end users. These solutions ensure that the SUSE Rancher managed Kubernetes downstream cluster workloads are available, consistent, durable, and recoverable.

Configuration details

Storage-only and compute-only nodes

The following tables provide the configuration details of a Dell EMC PowerEdge R640 server storage-only and compute-only nodes:

Table 3. Storage-only nodes

Hardware	Configuration
CPU Cores	2 x Intel(R) Xeon(R) Gold 6126 CPU @ 2.60 GHz
Memory	14 x 16 GB
NIC	2 x Mellanox ConnectX-4 LX 25 GbE SFP Adapter
	1 x Intel(R) Ethernet 10G 4P X710/I350 rNDC
Storage	BOSS S1 Controller 2 x 240 GB SATA SSD
	Dell HBA330 controller 8 x 1.9 TB SAS SSD
Operating system	Embedded operating system
PowerFlex	3.5.1.2

Note: In a Dell EMC PowerEdge R640 server, available storage on storage nodes is the only difference between storage and compute nodes. SSDs in the storage nodes are used to form a storage pool and volumes that are used by the compute nodes.

Table 4. Compute-only nodes

Hardware	Configuration
CPU Cores	2 x Intel(R) Xeon(R) Gold 6248 CPU @ 2.50 GHz
Memory	24 x 32 GB
NIC	2 x Mellanox ConnectX-4 LX 25GbE SFP Adapter
	1 x Intel(R) Ethernet 10G 4P X710/I350 rNDC
Storage	BOSS S1 Controller 2 x 240 GB SATA SSD

Hardware	Configuration
	Dell HBA730 controller 2 x 900 GB SAS SSD
Hypervisor	ESXi 7.0
PowerFlex	3.5.1.2

SUSE Rancher Kubernetes cluster

The following table provides the configuration details of a Rancher management cluster:

Table 5. Rancher management cluster details

Components	Items	Details
Hardware	Virtualized Hardware	vSphere 7.0
	CPU	2 vCPU
	RAM	8 GB
	Hard Disk	50 GB
	NIC	VMware Virtual NIC
Software	Operating System	SLES15 SP2
	Container Runtime	Docker 19.03.15
	Rancher	v2.5.7
	Rancher Kubernetes	v1.20.4
	RKE	v1.2.6
	PowerFlex CSI driver	v1.4

References

Dell Technologies documentation

The following Dell Technologies documentation provides additional and relevant information. Access to these documents depends on your login credentials. If you do not have access to a document, contact your Dell Technologies representative.

- PowerFlex Overview — Video
- PowerFlex Specification Sheet
- PowerFlex Solutions Document — InfoHub
- PowerFlex Networking Best Practices and Design Considerations
- PowerFlex - Storage definitions
- Dell EMC CSI Drivers - Dell CSI Git Repo

Rancher Documentation

The following Rancher documentation provides additional and relevant information:

- Rancher Product Overview
- Rancher Installation
- Rancher Support Matrix
- Rancher – Kubernetes authentication
- Backup and Restore in Rancher
- Backing up Rancher Installed on an RKE Kubernetes Cluster

Dell EMC PowerProtect Documentation

The following Dell EMC PowerProtect documentation provides additional and relevant information:

- Dell EMC PowerProtect Data Manager Administration and User Guide
- PowerProtect Data Manager Compatibility Matrix
- Dell EMC PowerProtect Data Manager Deployment Guide
- Dell EMC PowerProtect DD Series Appliances