

# Installing and configuring Oracle Linux KVM on Dell PowerFlex

June 2024

H18823.1

## Reference Architecture Guide

### Abstract

This reference architecture guide demonstrates installing and configuring Oracle Linux KVM on the Dell PowerFlex platform.

Dell Technologies Solutions

ORACLE | Partner

Dell Technologies

**Validated Design**

## Copyright

© 2021-2024 Dell Inc. or its subsidiaries. All rights reserved. Dell Technologies, Dell, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

# Contents

Executive Summary ..... 4

PowerFlex family overview ..... 6

PowerFlex deployment architectures ..... 8

PowerFlex consumption options ..... 9

Introduction to Oracle Linux KVM..... 10

Introduction to partitioning..... 18

Storage domains ..... 25

Deploying Oracle Real Application Clusters..... 33

Conclusion ..... 45

References ..... 46

## Executive Summary

Customers running large enterprise databases require a cost-effective solution to manage their IT resources such as compute, network, and storage.

Oracle Linux KVM, an open-source hypervisor, which is distributed as part of Oracle Linux, offers a virtualization layer that when combined with Oracle Linux KVM CPU pinning, can become cost optimized for Oracle deployments. Oracle Linux Virtualization Manager provides a solution for managing different resources such as the compute, network, storage, and virtual machines, in enterprise-class virtualized environments. Oracle Linux KVM and Oracle Linux Virtualization Manager are included with an Oracle Linux Premier Support subscription at no additional cost.

This reference architecture guide explains the deployment of Oracle Linux KVM with PowerFlex. In addition, it provides an example of deploying a virtualized database cluster in this environment.

## Audience

This reference architecture guide is intended for Oracle Database administrators, system administrators, architects, and technical administrators of IT environments, and anyone who is interested in installing and configuring a virtualized environment for PowerFlex using Oracle Linux KVM.

The readers of this document are expected to have basic knowledge of PowerFlex and Oracle Database technologies and should be familiar with storage, compute, networking technologies and topologies.

## Terminology

The following table provides definitions for some of the terms that are used in this document.

**Table 1. Terminology**

Term	Definition
Oracle RAC	Oracle Real Application Clusters
KVM	Kernel-based Virtual Machine
Hypervisor	The software, hardware, and firmware that creates, manages, and runs virtual machines. It can manage a physical host's hardware for this purpose.
oVirt	A free open-source distributed virtualization solution which can manage a KVM environment.
oVirt Engine	The management tool that is used to configure, customize, and monitor the KVM environment.
SDS	Storage Data Server
SDC	Storage Data Client
OS	Operating System
MG	Medium Granularity
FG	Fine Granularity

Term	Definition
PCI	Peripheral component interconnect
VDSM	Virtual Desktop and Server Manager
Volume	Used interchangeably with “device” to refer to a host device
NUMA	Non-uniform memory access – a design of computer systems where the memory access time depends on how close it is to the processor. NUMA improves the performance of servers.
Node	As well as “host”, refers to a physical machine with the KVM hypervisor
Host	As well as “node”, refers to a physical machine with the KVM hypervisor
Device	Used interchangeably with “volume” to refer to a host device

## 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:** Harsha Yadappanavar, Drew Tonnesen

**Contributors:** Kevin M Jones, Dell Technologies Solution Information Development and Design team

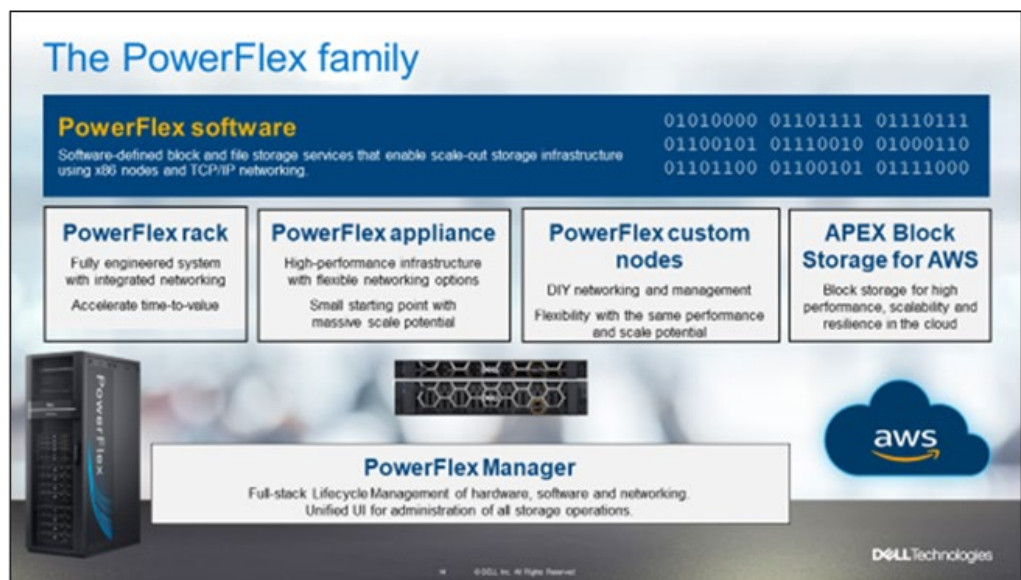
## Revisions

Date	Description
June 2021	Initial Release
December 2021	Updated with Oracle
June 2024	Oracle 21c, PowerFlex update, additional topics

## PowerFlex family overview

PowerFlex software-defined infrastructure enables broad consolidation across the data center, encompassing almost any type of workload and architecture. The software-defined architecture offers automation and programmability of the complete infrastructure and provides scalability, performance, and resiliency to enable effortless adherence to stringent workload SLAs.

The PowerFlex family provides a foundation that combines compute and high-performance storage resources in a managed unified fabric. PowerFlex comes in flexible deployment options (rack, appliance, or custom nodes and in the public cloud) that enables independent (two-layer), HCI (single-layer), or mixed architectures. PowerFlex is ideal for high-performance applications and databases, building an agile private-hybrid cloud, or consolidating resources in heterogeneous environments.



**Figure 1. PowerFlex family**

### PowerFlex software components

Software is the key differentiation in the PowerFlex offering. PowerFlex software components not only provide software-defined storage services, but also help simplify infrastructure management and orchestration. This software enables comprehensive IT Operational Management (ITOM) and Life Cycle Management (LCM). These capabilities span compute and storage infrastructure, from BIOS and Firmware to nodes, software, and networking.

### PowerFlex

PowerFlex is the software foundation of PowerFlex software-defined infrastructure. It is a scale-out block and file storage service that is designed to deliver flexibility, elasticity, and simplicity with predictable high performance and resiliency at scale.

### PowerFlex Manager

PowerFlex Manager is the software component in the PowerFlex family that enables ITOM automation and LCM capabilities for PowerFlex systems. Starting with PowerFlex 4.0, the unified PowerFlex Manager brings together three separate components from previous releases: PowerFlex Manager, the core PowerFlex UI, and the PowerFlex

gateway. The new PowerFlex UI runs in Kubernetes and embraces a modern development framework.

### PowerFlex file services

PowerFlex File Controllers, also known as File Nodes, are physical nodes that enable PowerFlex software defined File Services. They host the NAS Servers, which in turn host tenant namespaces and file systems, mapping PowerFlex volumes to the file systems presented by the NAS Servers. All major protocols are supported, such as NFS, SMB-CIFS, FTP, and NDMP. Both NFS 3 and 4 are supported. PowerFlex file service is supported from PowerFlex 4.0.

### PowerFlex CSI and CSM

An important component outside of PowerFlex that enables a flexible consumption model for Kubernetes is the PowerFlex CSI driver. It was developed as a part of the Dell Kubernetes strategy. After the CSI driver for PowerFlex is loaded into Kubernetes, it can be used to provision persistent volumes from the underlying PowerFlex storage resource. If the Kubernetes deployment is running low on PowerFlex storage resources, you can add PowerFlex storage nodes to increase the system capacity and performance.

The CSI driver connects the PowerFlex system and Kubernetes deployments. It is a storage broker agent which dynamically provisions volumes from PowerFlex through the PowerFlex API gateway to the Kubernetes cluster. Once the volume is available on PowerFlex, it is immediately mapped to the requesting pod. If a pod is destroyed or rescheduled, the CSI plug-in ensures that the volumes are remapped upon rescheduling of that pod.

Customers running Kubernetes clusters on PowerFlex use the Dell Container Storage Modules (CSM), which extend the CSI driver capabilities. These modules:

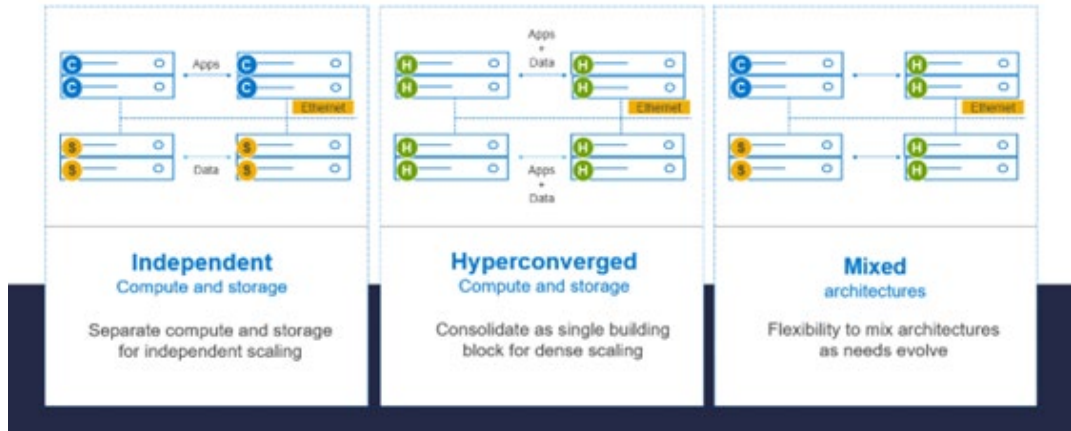
- Provide enterprise storage capabilities to Kubernetes for cloud-native stateful applications.
- Reduce management complexity, so that developers can independently use enterprise storage with ease and automate daily operations.
- Extend storage functionality and capabilities beyond using the CSI driver alone.

These modules include snapshot, observability, authorization, application mobility, and resiliency.

PowerFlex supports multiple operating systems and different deployment options for on-premises and public cloud deployment models (available in AWS). PowerFlex is validated with the leading Kubernetes distributions.

## PowerFlex deployment architectures

PowerFlex software-defined infrastructure excels in deployment flexibility. PowerFlex can be deployed in a two-layer (independent compute and storage layers), single-layer (Hyperconverged Infrastructure, or HCI), or a mixture of the two architectures (Mixed).



**Figure 2. PowerFlex architectures**

### Independent architecture

In an independent architecture or two-layer architecture, some nodes provide storage capacity for data in applications. Other separate and independent nodes provide compute resources for applications and workloads. Compute and storage resources can be scaled independently by adding nodes to the cluster while it remains active. This separation of compute and storage resources helps to minimize software licensing costs in certain situations. This architecture can be ideal for high-performance databases and application workloads.

### Hyperconverged architecture

In an HCI architecture, each node in the cluster contributes storage and compute resources simultaneously to the applications and workloads. This architecture allows you to scale your infrastructure uniformly with building blocks that add both storage and compute resources. This architecture is appropriate for data center and workload consolidation.

### Mixed architecture

A mixed architecture has a combination of both the HCI and Independent architectures. As shown previously in [Figure 2](#), there would be some storage only nodes, compute only nodes, and hyperconverged nodes as part of the same PowerFlex cluster. This architecture is desirable when working with an existing compute infrastructure and adding high-performance software-defined infrastructure. This architecture can also be a starting point for a two-layer deployment design as external workloads are migrated to PowerFlex.

## PowerFlex consumption options

<b>PowerFlex rack</b>	PowerFlex rack is a software-defined infrastructure platform that delivers flexibility, elasticity, and simplicity with predictable performance and resiliency at scale. It combines compute and high-performance storage resources in a managed unified network. This rack-based engineered system, with integrated networking, enables customers to achieve the scalability and management requirements of a modern data center.
<b>PowerFlex appliance</b>	PowerFlex appliance is a PowerEdge server which has been configured to be a node in a software-defined infrastructure deployment that runs PowerFlex software components. This offering allows customers the flexibility and savings to bring their own compatible networking.
<b>PowerFlex Custom Nodes</b>	PowerFlex Custom Nodes are validated server building-blocks that are configured for use with PowerFlex. They are available with thousands of configuration options and are available for customers who prefer to build their own environments.
<b>Dell APEX Block Storage for AWS</b>	PowerFlex software can also be deployed in the public cloud. It is available in the Amazon Marketplace as Dell APEX Block Storage for AWS (formerly known as PowerFlex cloud storage). PowerFlex on AWS offers the same on-premises benefits of high-performance, linear scalability, and high resilience as with cloud. PowerFlex also adds cloud-specific benefits, such as large volume sizes, extreme performance based on NVMe drives, and predictable scalability. With PowerFlex on AWS, you can also get higher Multi Availability Zone (Multi AZ) resiliency when PowerFlex Fault sets are distributed across multiple AWS Availability Zones. For more information, see <a href="#">Dell APEX Block Storage for AWS</a> .

## Introduction to Oracle Linux KVM

Oracle Linux Kernel-based Virtual Machine (KVM) is an open-source virtualization infrastructure for the Linux kernel that allows it to act as a hypervisor. A hypervisor, or virtual machine monitor (VMM), is a hardware, software, or firmware layer that runs virtual machines on a physical host. It does this by virtualizing the hardware, enabling one system to become many. There are two types of hypervisors: Type 1 which runs directly on the physical host; and Type 2 that runs on top of the operating system of the physical host. Oracle Linux KVM is a Type 1 hypervisor as it manages the hardware resources directly rather than Type 2 which relies on the operating system device drivers.

Oracle Linux Virtualization Manager is a server virtualization management platform which is used to configure, monitor, and manage an Oracle Linux KVM environment. Oracle Linux KVM and Oracle Linux Virtualization Manager are two separate components that can be installed in an Oracle Linux environment.

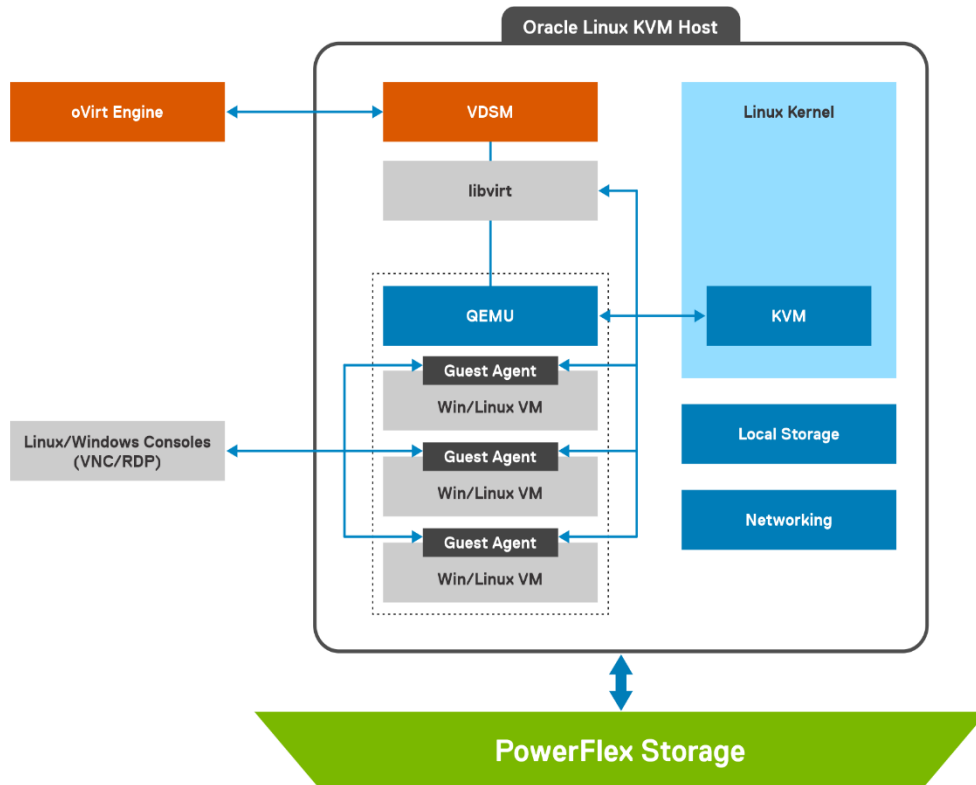
### Oracle Linux KVM architecture

Oracle Linux KVM provides a set of modules that enable the Oracle Linux kernels (the Unbreakable Enterprise Kernel and Red Hat Compatible Kernel) to be used as a hypervisor. Oracle Linux KVM consists of a loadable kernel module, `kvm.ko`, that provides the core virtualization infrastructure, and a processor-specific module, `kvm-intel.ko` or `kvm-amd.ko`.

Oracle Linux KVM runs in the host kernel space and the VMs running on Oracle Linux KVM hosts can run as individual QEMU processes in user space. QEMU stands for quick emulator, which enables KVM to become a complete hypervisor by emulating the hardware for VMs such as CPU, memory, network, and disk devices. Oracle Linux KVM allows QEMU to run code in the VM directly on the host CPU, thus allowing the VM's operating system direct access to the host's resources without modification.

The libvirt daemon runs as a service on Oracle Linux KVM hosts and provides an application programming interface (API) for managing various hypervisors, including Oracle Linux KVM. VDSM uses libvirt to manage the complete life cycle of virtual machines and their virtual devices on the host, and to collect statistics about them.

A representation of the Oracle Linux KVM architecture is shown in [Figure 3](#).



**Figure 3. Oracle Linux KVM host architecture and associated components**

Server virtualization with [Oracle Linux KVM](#) is supported on Intel VT, AMD-V, or ARM servers that are [certified](#) for Oracle Linux 8 or 9 with the Unbreakable Enterprise Kernel.<sup>1</sup> Users can configure KVM from a base Oracle Linux installation. Oracle Linux KVM includes support for Intel VT-x and VT-d hardware extensions along with Secure Encrypted Virtualization (SEV) for AMD-V enabled processors.

### Highlights of Oracle Linux KVM

- Hard partitioning support
- Zero-license cost for the hypervisor
- Support included with an [Oracle Linux Premier Support](#) subscription at no additional cost
- Easy to implement, install, manage, and configure through the Oracle Linux Virtualization Manager UI
- Supports hardware assisted virtualization
- Supports paravirtualized drivers (virtio)
- Virtual machine snapshot and cloning capabilities
- Online VM migration, memory, and storage
- VM setup from templates
- Supports PCI passthrough
- Easy memory management for guest VMs
- Fully documented

<sup>1</sup> See the Oracle [certification matrix](#) for exact UEK version support. Oracle Linux Virtualization Manager supports the management of KVM hypervisors for Intel VT and AMD-V.

## Oracle Linux Virtualization Manager

Oracle Linux Virtualization Manager is a JBoss-based Java application running as a web service that provides centralized management for server and desktop virtualization. It is based on the open source oVirt project and is referred to as the oVirt Engine. The engine runs on another Oracle Linux environment, a dedicated host or deployed as a self-hosted engine. A self-hosted engine is a virtualized environment where the engine runs inside a virtual machine on the hosts in the environment. The virtual machine for the engine is created as part of the host configuration process. The engine communicates directly with the Virtual Desktop and Server Manager (VDSM) service, running on Oracle Linux KVM hosts as a daemon, to perform tasks such as managing hosts, VMs, networks, and storage, and to create new images and templates. The engine is deployed by the engine-setup script, the summary of which is shown in [Figure 4](#).

```
[ INFO ] Stage: Setup validation

--== CONFIGURATION PREVIEW ==--

Application mode                : virt
Default SAN wipe after delete   : False
Host FQDN                      : austin245.dellhclilab.com
Update Firewall                 : False
Set up Cinderlib integration    : False
Configure local Engine database : True
Set application as default page : True
Configure Apache SSL            : True
Keycloak installation           : True
Engine database host            : localhost
Engine database port            : 5432
Engine database secured connection : False
Engine database host name validation : False
Engine database name            : engine
Engine database user name       : engine
Engine installation             : True
PKI organization                : dellhclilab.com
Set up ovirt-provider-ovn       : True
DWH installation                : True
DWH database host               : localhost
DWH database port               : 5432
DWH database secured connection : False
DWH database host name validation : False
DWH database name               : ovirt_engine_history
Configure local DWH database    : True
Grafana integration             : True
Grafana database user name      : ovirt_engine_history_grafana
Keycloak database host          : localhost
Keycloak database port          : 5432
Keycloak database secured connection : False
Keycloak database host name validation : False
Keycloak database name          : ovirt_engine_keycloak
Keycloak database user name     : ovirt_engine_keycloak
Configure local Keycloak database : True
Configure VMConsole Proxy       : True
Configure WebSocket Proxy       : True

Please confirm installation settings (OK, Cancel) [OK]:
```

**Figure 4. Engine-setup summary**

The setup script uses the host FQDN as the URL for web access, though the user can also append ovirt-engine which is the redirect:

<https://austin245.dellhclilab.com/ovirt-engine/>.

### oVirt Engine backups

After deployment of the oVirt Engine, it is important to setup regular backups. oVirt offers a simple CLI tool, `engine-backup`, that dumps the engine to a single file which can be used for various restore operations. In fact, if backups are not taken regularly, Oracle Virtualization Manager warns the user in the event log with the following message: “There

*is no full backup available, please run engine-backup to prevent data loss in case of corruption.”*

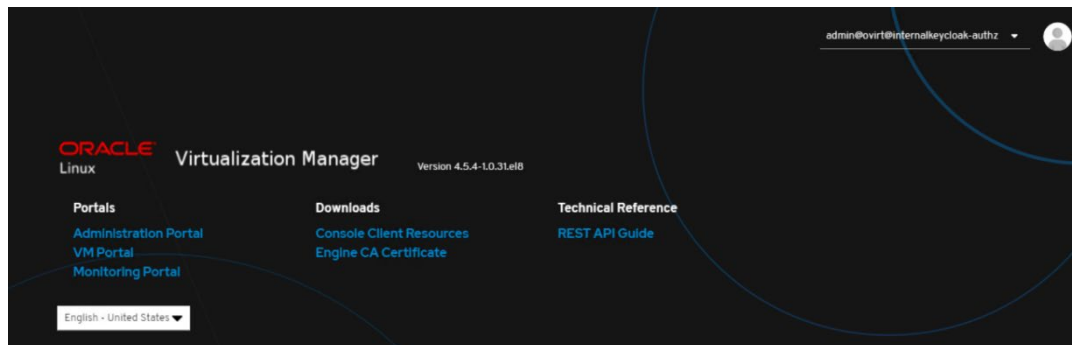
Although there are many options for backing up using the utility, including adding config files, the simplest operation is to run a full backup. It is best to save the backup file to a location off the hosted engine server. In this environment, a separate PowerFlex NFS mount, `engine_backup`, is used for this purpose.

Run the following command as root, modifying the file locations:

```
engine-backup --scope=all --mode=backup --
file=/engine_backup/engine_backup_1-18-2024 --
log=/engine_backup/engine_backup.log
```

## Portals

Beyond the basics already noted, Oracle Linux Virtualization Manager offers administrators the ability to migrate VMs through the interface, manage high availability, and even setup storage quotas or performance limitations (throttling). There are three options for UI views shown in [Figure 5](#): Administration Portal, VM Portal, and Monitoring Portal.



**Figure 5.** Oracle Linux Virtualization Manager landing page from browser

## Administration portal

After logging in to the administration portal, users will be presented with a dashboard view displaying key information about the deployment such as VM counts, hosts count, clusters, storage, and so on. It also displays the status of each entity and key performance metrics in [Figure 6](#).

This portal view is the heart of Oracle Linux Virtualization Manager as almost all activities related to the environment can be accomplished here. The user can create storage domains, clusters, VMs, add new hosts and change networking. The portal also provides an easy-to-use interface to upgrade Oracle Linux KVM hosts.

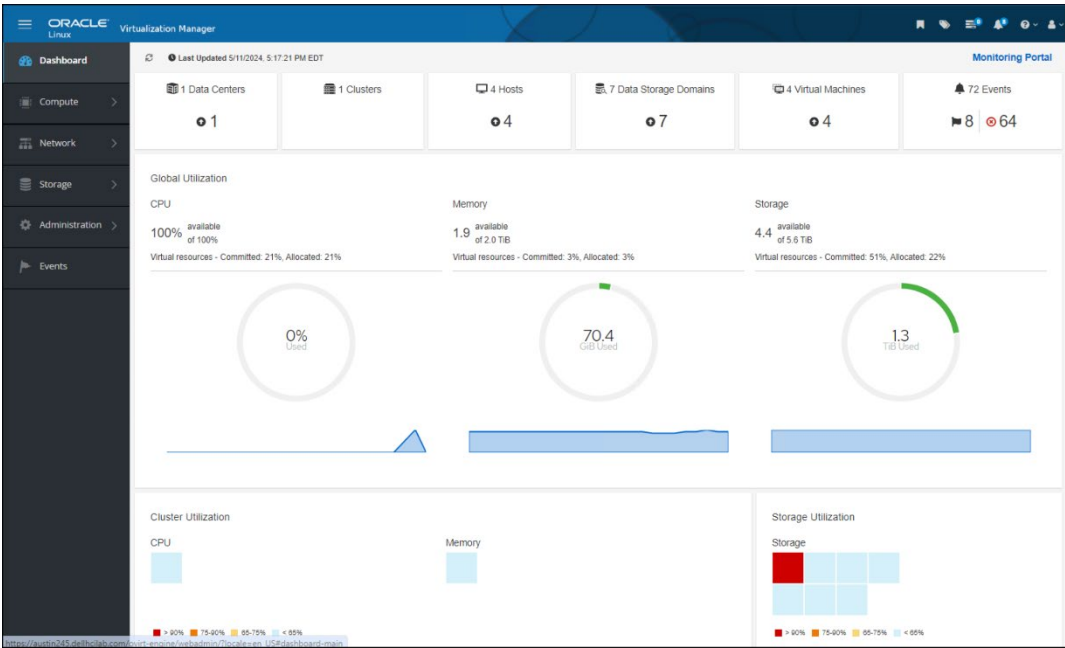


Figure 6. Display of dashboard showing details such as hosts, VMs, and storage metrics

### Upgrades

In the **Clusters** view of the **Administration** portal if new software becomes available, the interface informs the user. If the user drills down to the Hosts, each host indicates when an upgrade is available and enables a user to check for upgrades on demand. Both these views are present in Figure 7.

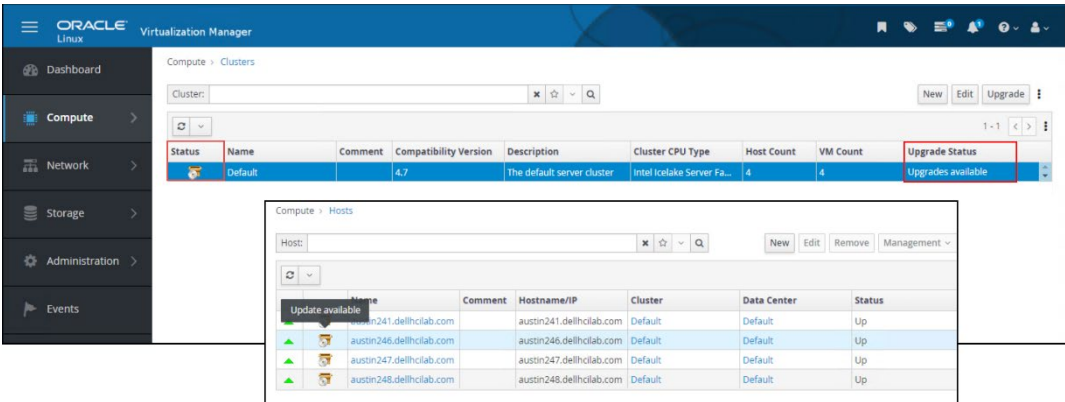
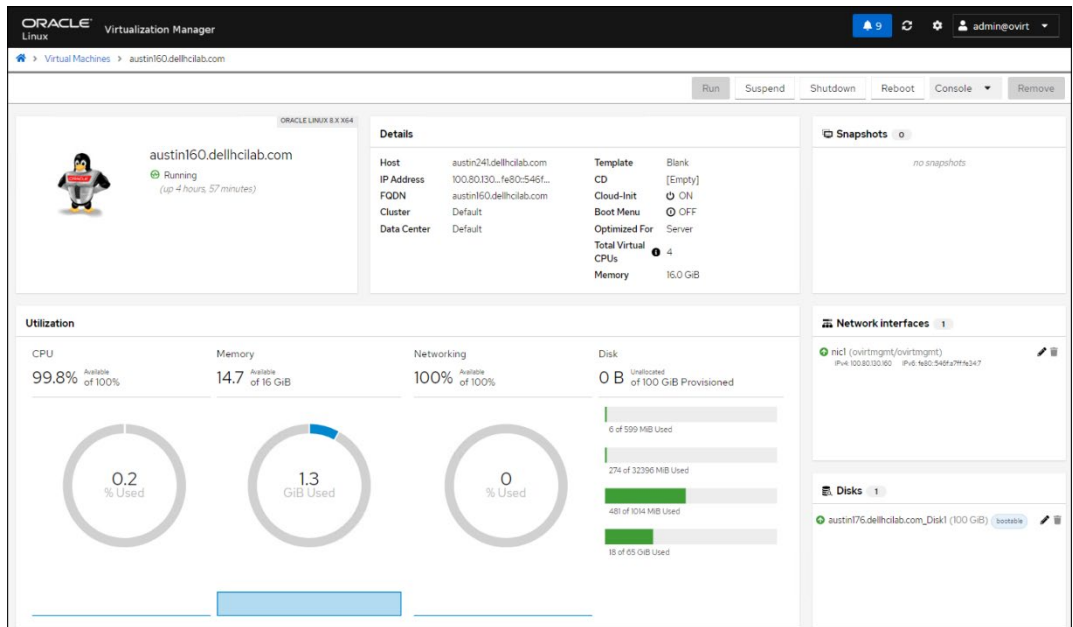


Figure 7. Cluster upgrade status

### VM portal

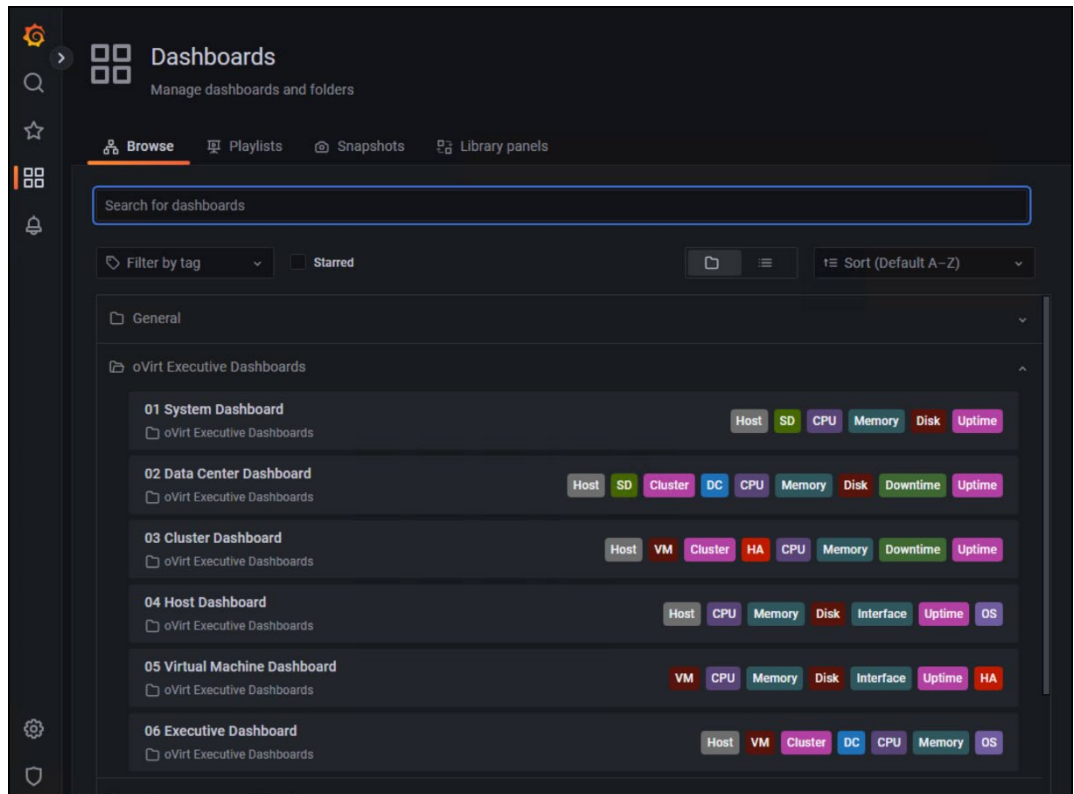
The VM Portal allows the user to drill down into each virtual machine in the environment and make changes at that level. Figure 8 shows the detail of one particular VM, austin160.dellhclab.com.



**Figure 8. VM Portal**

## Monitoring portal

The final option is the Monitoring Portal which has Grafana at its foundation. Grafana is an open-source analytics and monitoring platform that integrates with oVirt and provides dashboards, alerting, and monitoring capabilities. It is a highly customizable platform permitting users to modify and share dashboards. Grafana comes preconfigured with a selection of dashboard categories and dashboards shown in [Figure 9](#).



**Figure 9. Monitoring using Grafana**

## Oracle Linux KVM terminology

This section lists some of the important terms used in Oracle Linux KVM.

### Data Centers

The data center is a high-level logical entity for all physical and logical resources in the environment. After the Oracle Linux Virtualization Manager is installed, a default data center is created automatically. The user can initialize a data center by adding a cluster, a host, and a storage domain.

### Cluster

The cluster is a logical grouping of one or more Oracle Linux KVM hosts on which VMs can run. Oracle Linux KVM hosts in a cluster must share the same storage domains and need to have the same CPU type (either Intel or AMD). Each cluster belongs to a data center and each Oracle Linux KVM host belongs to a cluster.

Virtual machines are dynamically allocated to any Oracle Linux KVM host in the cluster and can be migrated between them. Even when one of the hosts is unavailable, the virtual machine can be started in any available host.

### Hosts

On a bare-metal server, after installation of Oracle Linux, the server can be used as the Oracle Linux KVM hypervisor in Oracle Linux Virtualization Manager and host virtual machines. All the hosts in such an environment should be Oracle Linux KVM hosts. Dell Technologies recommends running the engine on a separate Oracle Linux host, rather

than as a VM in the KVM setup. Oracle Linux Virtualization Manager can manage many Oracle Linux KVM hosts, each of which can run multiple virtual machines simultaneously. Each VM runs as individual Linux processes and threads on the Oracle Linux KVM host.

### Virtual Machine

VMs can be created new or cloned from an existing template in the virtual machine pools. A virtual machine pool is a group of on-demand virtual machines that are all clones of the same template. The template is a copy of a virtual machine that can be used to repeat the creation of a similar virtual machine.

### Storage

The virtualization manager uses the centralized storage system for virtual machine disk images and ISO files. The protocols iSCSI, FC, and NFS are all supported, along with POSIX compliant FS and GlusterFS. A data center cannot be initialized unless a storage domain is attached to it and activated. A storage domain contains complete images of templates, virtual images, virtual machine snapshots, or ISO files.

### Network

The following high-level networking is recommended:

- Oracle Linux Virtualization Manager uses bonded network interfaces, which is preferred
- Use VLANs to distinguish between traffic types
- Use 4 ports of 25 GbE for data uplink and 1 GbE networks for iDRAC traffic

The Oracle Linux Virtualization Manager host and all Oracle Linux KVM hosts must have a fully qualified domain name (FQDN).

### Guest agent

The guest agent runs inside the virtual machine and provides information about resource usage to the engine. It provides the information, notifications, and actions between the engine and the guest.

### Scalability requirements

For detailed minimum and maximum system requirements and scalability limitations, see the [Oracle documentation](#).

## Introduction to partitioning

Oracle supports both soft and hard partitioning. Partitioning allows the user to use a subset of resources for a specific purpose, licensing only the used resources such as CPU. The partitioned resource might be a physical server or even the CPU on that server. Partitioning can be advantageous when the user wants to run multiple operating systems on the same server, or to manage how CPU resources are doled out. A soft partitioning example is Oracle VM where the resources are virtualized. Hard partitioning examples include Solaris Zones and IBM's LPAR, but also CPU pinning on an Oracle Linux KVM environment. It is this latter use case that is covered next.

### Hard partitioning

Oracle Linux KVM offers a method to create hard partitioning, which is also known as CPU pinning. This involves binding vCPUs to physical CPU threads or cores, thus preventing vCPUs from being scheduled to run on physical CPUs other than the ones specified. With Oracle Linux Virtualization Manager and Oracle Linux KVM, to conform to the Oracle hard partition licensing requirement, you must bind a virtual machine to physical CPUs or cores. In order to qualify for this partitioning, a user must adhere to the following documentation: [Hard Partitioning with Oracle Linux KVM](#).

On an x86-based system, a CPU core (without hyperthreading) or a CPU thread (with hyperthreading) is presented as a physical CPU by the hypervisor or bare metal operating system. vCPUs are exposed to the guest virtual machine as CPUs. The guest schedules applications on these vCPUs, and the hypervisor schedules these vCPUs over the physical CPU cores or threads.

### Configuring hard partitioning

The host running Oracle Linux KVM should meet the requirements of the Oracle Linux KVM compute host as defined in the hard partitioning document mentioned previously. The Oracle Linux Virtualization Manager provides a utility called `olvmm-vmcontrol`, through which users can get and set the CPU/vCPU bindings for a virtual machine on Oracle Linux KVM.

CPU pinning should be configured as one of the first tasks after creation of the virtual machine. At the very least, it needs to be done prior to installation of the Oracle Database software.

The `olvmm_vmcontrol` utility can run on the host running the Oracle Linux Virtualization Manager or on a separate Oracle Linux host that has connectivity to the Oracle Linux Virtualization Manager.

### VM host affinity

Essential to CPU pinning is associating the VM in question with a particular KVM host since pinning does not traverse, or cannot be transferred, between hosts. If the user does select a server, when the VM is pinned, the server it starts on after will be the designated one. It is a best practice, however, to set the server in the UI as in [Figure 10](#).

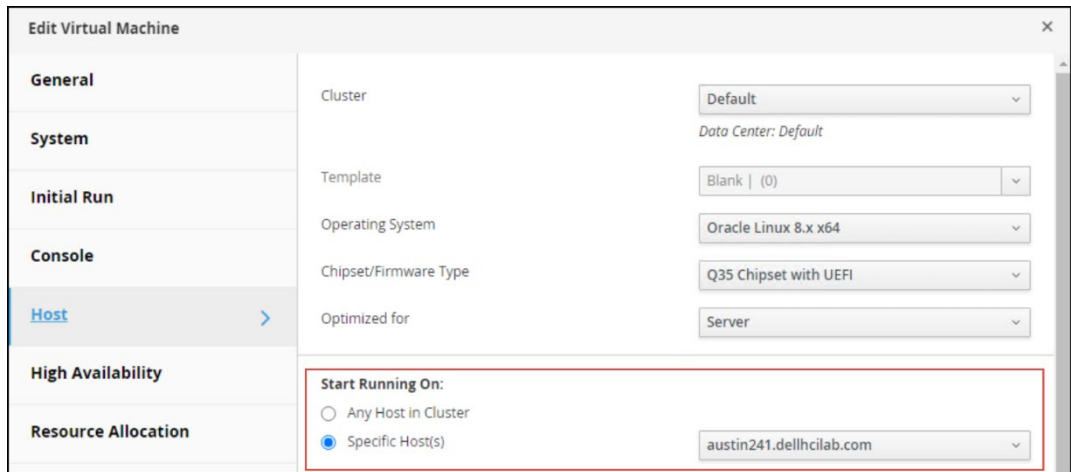


Figure 10. Setting VM host affinity

### Installation of olvm\_vmcontrol

To install the utility, users must install the required RPM which is available in the existing Oracle Linux repositories:

```
yum install olvm-vmcontrol
```

The utility is run with the same name, `olvm-vmcontrol`, and accepts three commands to manipulate CPU pinning: `getvcpu`, `setvcpu`, and `rmvcpu`.

### CPU configuration

Begin by determining the attributes of the Oracle Linux KVM CPU configuration using the command `lscpu`. Figure 11 is provided as an example below:

```
[root@austin246 ~]# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                96
On-line CPU(s) list:   0-95
Thread(s) per core:    2
Core(s) per socket:    24
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
BIOS Vendor ID:        Intel
CPU family:            6
Model:                 106
Model name:            Intel(R) Xeon(R) Gold 6336Y CPU @ 2.40GHz
BIOS Model name:       Intel(R) Xeon(R) Gold 6336Y CPU @ 2.40GHz
Stepping:              6
CPU MHz:               3600.000
CPU max MHz:           3600.0000
CPU min MHz:           800.0000
BogoMIPS:              4800.00
Virtualization:        VT-x
L1d cache:             48K
L1i cache:             32K
L2 cache:              1280K
L3 cache:              36864K
NUMA node0 CPU(s):     0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,62,64,66,68,70,72,74,76,78,80,82,84,86,88,90,92,94
NUMA node1 CPU(s):     1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,79,81,83,85,87,89,91,93,95
```

Figure 11. Oracle Linux KVM CPU configuration (flags not included)

The above output indicates:

- The server has 2 sockets with 24 cores and 2 threads per core for a total of 96 CPUs
- NUMA node0 includes even threads 0-94 and NUMA node1 includes odd threads 1-95 (each thread representing a CPU)

### CPU topology

Oracle Linux KVM includes a command line utility named `virsh` which can be performed to manage the virtual machines in the environment. There is a read-only flag that can be used without authentication and is useful in retrieving information about the status of CPU pinning. Here, in [Figure 12](#), list the running VMs on the chosen KVM host.

```
[root@austin241 ~]# virsh --readonly list
Id      Name                                     State
-----
5       austin171.dellhclilab.com              running
8       austin160.dellhclilab.com              running

[root@austin241 ~]# virsh --readonly vcpuinfo austin160.dellhclilab.com --pretty
VCPU:      0
CPU:       15
State:     running
CPU time:  30.4s
CPU Affinity: 0-95 (out of 96)

VCPU:      1
CPU:       29
State:     running
CPU time:  24.9s
CPU Affinity: 0-95 (out of 96)

VCPU:      2
CPU:       53
State:     running
CPU time:  25.3s
CPU Affinity: 0-95 (out of 96)

VCPU:      3
CPU:       83
State:     running
CPU time:  24.3s
CPU Affinity: 0-95 (out of 96)
```

**Figure 12.** List the CPU pinning of the select VM

The commands indicate that a VM named `austin160.dellhclilab.com` is configured with 4 vCPUs (0-3) and that currently all CPUs 0-95 (threads) of the KVM host are available for all vCPUs. This means that the vCPUs for this VM are not pinned to physical cores.

### Setup CPU pinning

Next, pin the CPUs to the VM. As mentioned, the `olvmm-vmcontrol` utility is utilized to configure hard partitioning of a vCPU on the VM to a physical CPU on the KVM host. Though `virsh` demonstrated pinning is not in use, the `olvmm-vmcontrol` command also offers the command `getvcpu` to see the status. Be sure to run the utility on the correct host (typically where the engine was installed), and not the KVM host as `virsh` was.

First, check if pinning is set, and then assign the 4 vCPUs of the VM to 4 of the CPUs (0,1,2,3) in [Figure 13](#).

```
[root@austin245 ~]# olvm-vmcontrol -m austin245.dellhclab.com -u admin@localhost@internalssso -c getvcpu -f -e -v
austin160.dellhclab.com
Oracle Linux Virtualization Manager VM Control Utility 4.5.4-1.1
Connected to Oracle Linux Virtualization Manager 4.5.4-1.0.31.el8
Getting vcpu pinning ...
No CPU pinning is configured
[root@austin245 ~]# olvm-vmcontrol -m austin245.dellhclab.com -u admin@localhost@internalssso -c setvcpu -s 0,1,2
,3 -f -e -v austin160.dellhclab.com
Oracle Linux Virtualization Manager VM Control Utility 4.5.4-1.1
Connected to Oracle Linux Virtualization Manager 4.5.4-1.0.31.el8
Setting vcpu pinning ...
Trying to pin virtual cpu # 0
Trying to pin virtual cpu # 1
Trying to pin virtual cpu # 2
Trying to pin virtual cpu # 3
Retrieving vcpu pinning to confirm it has been set...
No CPU pinning is configured

NOTE: if the VM is running you must now stop and then start the VM from the Oracle Linux Virtualization Manager i
n order for CPU pinning changes to take effect.
NOTE: a restart or a reboot of the VM is not sufficient to put CPU pinning changes into effect.

[root@austin245 ~]#
```

**Figure 13. Getting and setting CPU pinning**

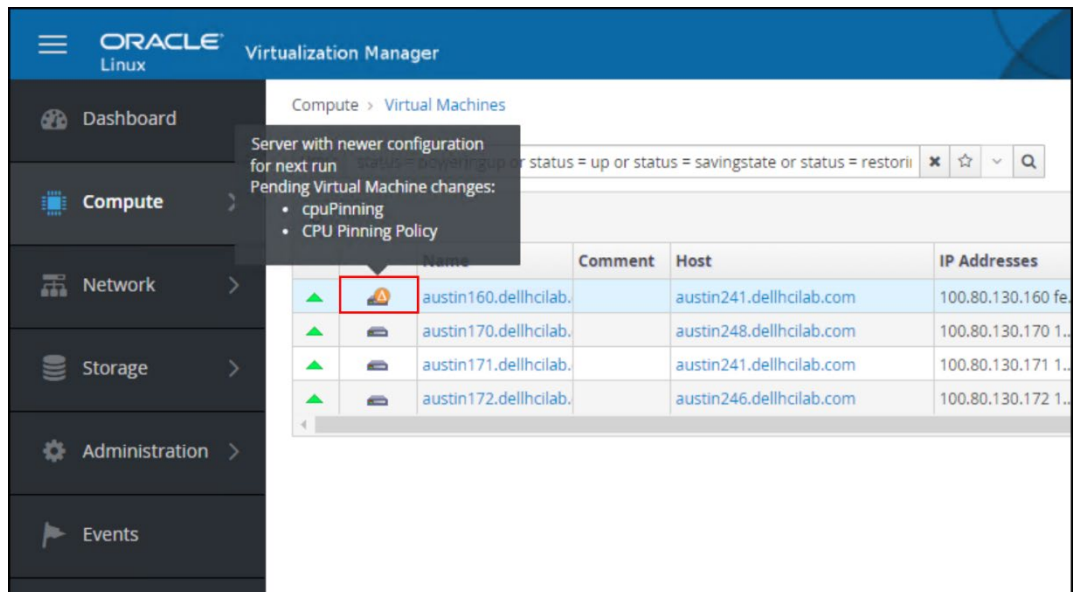
The utility at the end of pinning the CPUs first notifies the user that the pinning has not actually been done because the VM is in a running state. To take effect, the VM must be shut down and then restarted. The message is quite clear that a reboot or restart of the VM is insufficient for pinning. The VM must first come down.

---

**Note:** CPU pinning must be done prior to installing the Oracle Database software.

---

Helpfully, the UI shows an information icon which also reveals this information. An example is show in [Figure 14](#).



**Figure 14. UI icon indicating CPU pinning is pending**

After stopping and starting the VM, check the CPU pinning status first with the `olvm-vmcontrol` utility and then with the `virsh` command.

CPU pinning can be verified by the `virsh` command as follows:

The VM now has CPU affinity assigned for 4 virtual CPUs to physical CPU 0-3 in [Figure 15](#).

```
[root@austin241 ~]# virsh --readonly vcpuinfo austin160.dellhclilab.com --pretty
VCPU:      0
CPU:        0
State:      running
CPU time:   18.4s
CPU Affinity: 0-3 (out of 96)

VCPU:      1
CPU:        2
State:      running
CPU time:   10.8s
CPU Affinity: 0-3 (out of 96)

VCPU:      2
CPU:        0
State:      running
CPU time:   10.0s
CPU Affinity: 0-3 (out of 96)

VCPU:      3
CPU:        2
State:      running
CPU time:   9.6s
CPU Affinity: 0-3 (out of 96)
```

Figure 15. Post-CPU pinning

CPU pinning from UI

Users can also set CPU pinning topology from the resource allocation tab of the VM. Figure 16 shows the settings based on the example above.

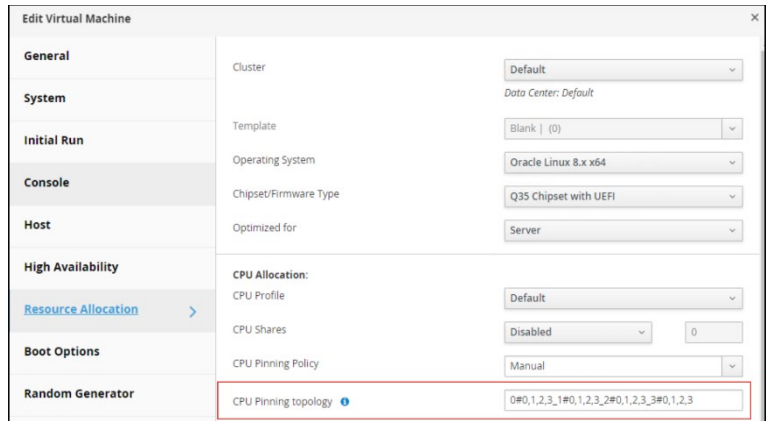


Figure 16. CPU pinning topology as displayed by VM configuration in UI

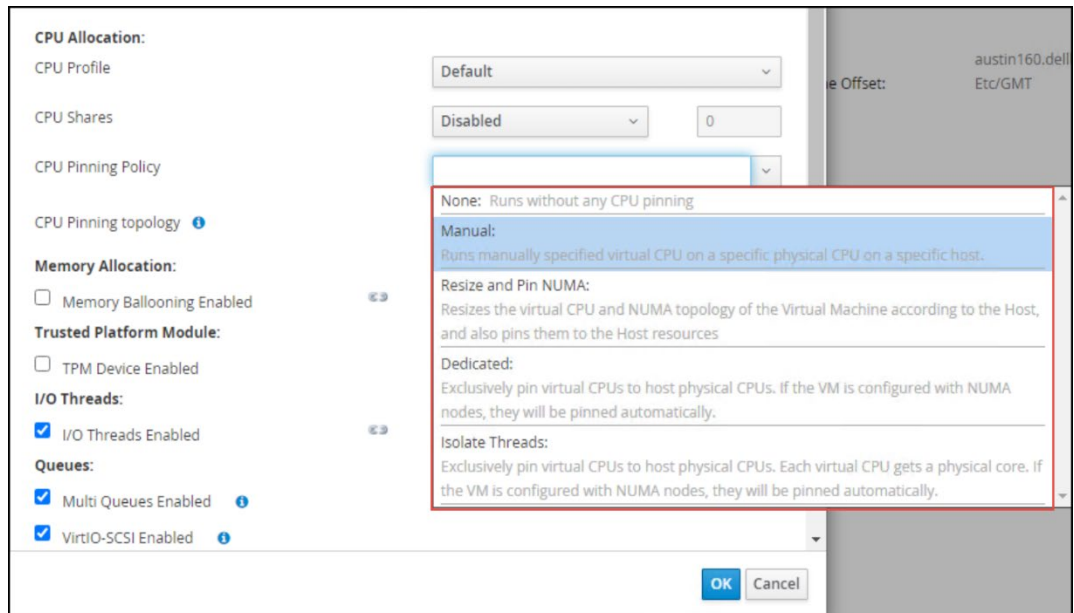
The following syntax can be used when manually setting CPU pinning:

Table 2. CPU pinning topology Format: v#p[v#p]

Example	Description
0#0	Pin vCPU 0 to pCPU 0
0#0_1#3	Pin vCPU 0 to pCPU 0 and Pin vCPU 1 to pCPU 1
1#1-4,^2	Pin vCPU 1 to pCPU set to 1 to 4, excluding 2

In addition to manual pinning, the UI offers the following automated options which are defined in [Figure 17](#):

- Resize and Pin NUMA
- Dedicated
- Isolate Threads



**Figure 17. Automated settings for CPU pinning in the UI**

For example, if Isolate Threads is selected, each virtual CPU will be tied to a unique physical CPU. Using the same VM as in the manual example, Isolate Threads was used and then the VM re-started. [Figure 18](#) shows that each vCPU was assigned a unique physical CPU:

- vCPU 0 → pCPU 0
- vCPU 1 → pCPU 4
- vCPU 2 → pCPU 8
- vCPU 3 → pCPU 12

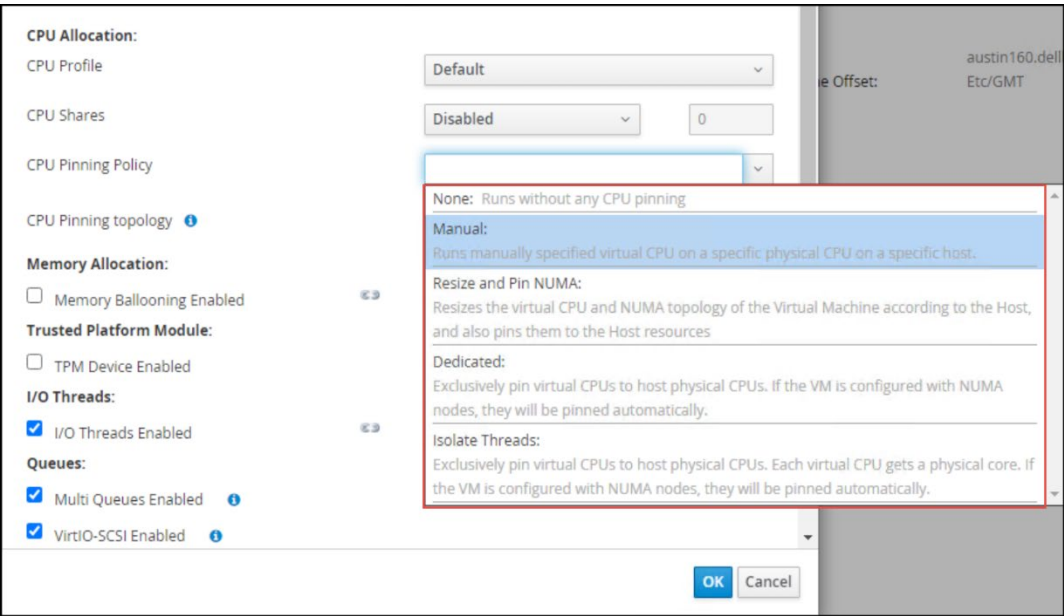


Figure 18. Isolate Threads example

## Restrictions

While CPU pinning can help ensure that a VM gets dedicated resources, there are some caveats the user needs to be aware of. While these are detailed in Oracle's pinning documentation previously referenced, one restriction is worth emphasizing. Oracle does not permit live migration of VMs that have pinned CPUs. If those VMs are on shared storage in an Oracle Linux KVM cluster, there can be no scheduling policy in Oracle Linux Virtualization Manager. A good use case, however, is Oracle Real Application Clusters (RAC) because the VMs should be deployed on their own Linux KVM host and not moved. The Oracle RAC instances are what provide HA, not the Linux KVM hosts.

**Note:** While online VM migration is not supported with CPU pinning, it is possible to move an Oracle RAC One instance between nodes so long as the nodes contain licensed Oracle Database software.

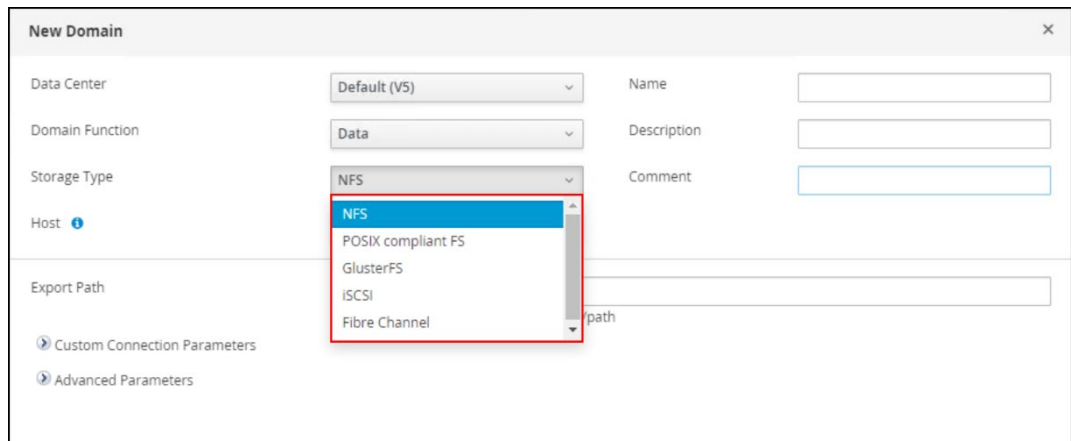
## Storage domains

Storage plays an essential role in the Oracle Linux KVM implementation. The storage dictates the type of cluster: shared or local. Shared storage is the more common cluster, enabling features such as HA and migration. For enterprise-level Oracle environments running on PowerFlex, Dell Technologies recommends creating storage domains on PowerFlex using the SDC. The process for doing so follows next.

### Storage types

When creating new shared storage domains in an Oracle Linux KVM environment, the user is limited to certain storage types. They are the following and appear in [Figure 19](#):

- NFS
- POSIX-compliant FS (global file system)
- GlusterFS
- iSCSI
- Fibre Channel



**Figure 19. Storage type options for a storage domain**

Putting aside GlusterFS which is not a storage protocol, PowerFlex supports two of these options, NFS and Fibre Channel (FC). While PowerFlex has native NFS support, it does not have FC support. Technically, however, the storage domain wizard is not looking for FC-presented devices, rather it is simply scanning for SCSI devices. So despite PowerFlex presenting volumes over the IP network, if the Oracle Linux KVM host is configured properly, Oracle Linux Virtualization Manager finds PowerFlex volumes that are mapped to the hosts through the Storage Data Client, or SDC. These volumes are regular SCSI devices, just like those presented with iSCSI or FC. Oracle Linux KVM thus treats PowerFlex devices that SDS presents as legitimate storage for the shared storage domain.

Oracle Linux KVM has specific requirements for each storage type that is used in a storage domain. The next sections cover the types that are supported with PowerFlex.

### NFS

Network File System (NFS) is a distributed file system protocol that allows users to access files over a network. NFS is a common choice for storage domains because it is simple to implement and manage. PowerFlex meets the performance and scalability requirements for Oracle Linux KVM as its NAS implementation can scale up to 16 nodes,

supports both NFS 3 and 4.1, and delivers superior performance when matched with a high performance, low latency network.

NFS is a good option in Oracle environments for shared homes and can also provide a storage domain for holding ISOs and other shared files between hosts. For NFS, the permissions and the ownership of the share must be altered so that Oracle Linux Virtualization Manager can use it in a storage domain. The available ownership and permissions on the PowerFlex NFS export are limited and do not meet the strict requirements of Oracle Linux KVM. It is necessary, therefore, to manipulate the share either on an existing Oracle Linux KVM node, or on a separate host altogether. In the following example, a Linux host was configured for the sole purpose of meeting the prerequisites of NFS storage. This example sets up the storage for an NFS mount named **ovirt\_engine**.

### Modify the NFS file system for the storage domain

Start by creating an NFS file system and mount on the PowerFlex system. This file system can be of any size but should have default access of **Read/Write, allow Root**. The summary of such a file system is shown in [Figure 20](#).

The screenshot shows a 'Create File System' window with a sidebar on the left containing four items: 'Select NAS Server', 'File System Details', 'NFS Export (Optional)', and 'Configure Access', each with a green checkmark. Below these is the 'Summary' tab, which is highlighted. The main area displays the following information:

Summary	
<b>FILE SYSTEM DETAILS</b>	
NAS Server	NAS-Production
Description	The file system that contains the oVirt Engine VM.
Size	50.0 GB
<b>NFS EXPORT</b>	
Name	engine_install
Description	Export for the oVirt Engine
Local Path	engine_install/
NFS Export Path	10.228.246.141:/engine_install
Default Access	Read/Write, allow Root
Minimum Security	Sys
Configured Host	No
Access	

At the bottom right, there are three buttons: 'Cancel', 'Back', and 'Create File System'.

**Figure 20. PowerFlex NFS file system and mount**

Now, to allow the storage domain to use the PowerFlex NFS file system, a few modifications must be made to folder ownership. Take the following steps to create a folder in the NFS file system with the correct ownership (**vdsm:kvm**) and permissions (0755) for the storage domain.

On most Linux implementations, the **kvm** group exists with the id of 36; however, the **vdsm** user will not. If another user on the box is already assigned the id of 36, use another host. If the user or group already exists with the correct uid, skip the step.

1. Log in as the root user.
2. Create the group **kvm**:  

```
groupadd kvm -g 36
```
3. Create the user **vds**m in the group **kvm**:  

```
useradd vds -u 36 -g kvm
```
4. Create a mount point.  

```
mkdir /engine_install
```
5. Mount the PowerFlex NFS share created in section xxx.  

```
mount -t nfs 10.228.246.141:/engine_install /engine_install
```
6. Create a sub-directory under the NFS share. Modify the access rights to that subfolder, changing the permissions and ownership.  

```
cd /engine_install
mkdir engine_install
chmod 0755 engine_install
chown 36:36 engine_install (or chown vds:kvm
engine_install)
```
7. Unmount the share.  

```
cd /
umount /engine_install
```

Once these steps are complete, the user can add the NFS storage domain using the export path of: 10.228.246.141:/engine\_install/engine\_install because that second directory will have the correct permissions that the wizard requires.

## FC/SDC

For Oracle databases on PowerFlex, Dell Technologies recommends using the Storage Data Client, or SDC-presented volumes. SDC is a lightweight device driver that exposes PowerFlex volumes as block devices to the host on which SDC is installed. PowerFlex presents SCSI devices through the SDC similar to FC or iSCSI. As there is no special integration in KVM for the SDC, it is necessary to use the Fibre Channel option when creating a storage domain.

Begin by installing the SDC package on each Oracle Linux KVM host following the PowerFlex deployment guide. Be sure that each KVM node has network access to the MDM IPs of the PowerFlex. This may require configuring additional IPs on the KVM hosts. Perform the following steps:

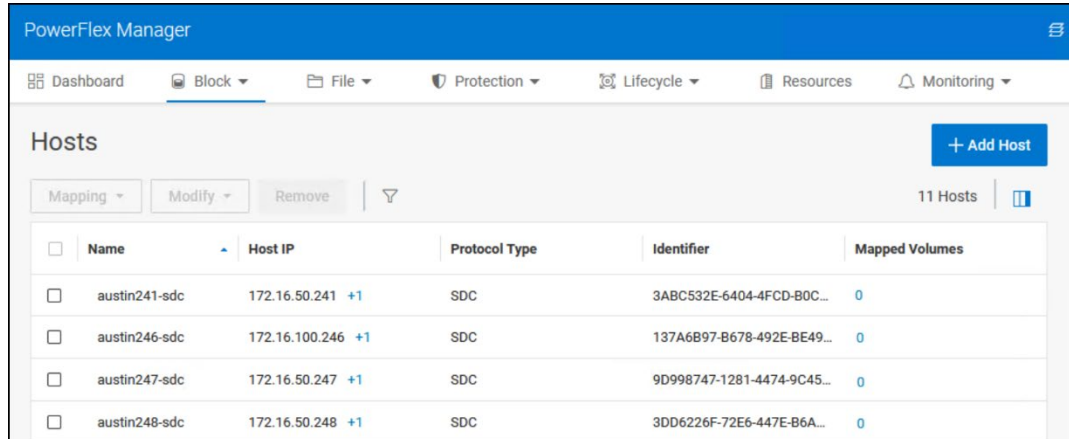
1. Copy the SDC rpm to the KVM node. In this example the non-GA lab rpm is EMC-ScaleIO-sdc-4.5-2000.137.el8.x86\_64.rpm which is for an OS based on Oracle Linux.
2. Set the environment variable for MDM\_IP so that after installation the SDC can communicate with the chosen cluster. The MDM\_IP could be the MDM IPs themselves, or virtual MDM IPs, depending on how PowerFlex is configured.

```
export MDM_IP=172.16.100.1
```

## 3. Install the rpm.

```
rpm -ivh EMC-ScaleIO-sdc-4.5-2000.137.el8.x86_64.rpm
```

The package installs and configures the scini service. No reboot is required. Upon completion, the PowerFlex cluster automatically registers the new KVM hosts. The four Oracle hosts in this configuration are shown in [Figure 21](#) in the PowerFlex UI.



<input type="checkbox"/>	Name	Host IP	Protocol Type	Identifier	Mapped Volumes
<input type="checkbox"/>	austin241-sdc	172.16.50.241 +1	SDC	3ABC532E-6404-4FCD-B0C...	0
<input type="checkbox"/>	austin246-sdc	172.16.100.246 +1	SDC	137A6B97-B678-492E-BE49...	0
<input type="checkbox"/>	austin247-sdc	172.16.50.247 +1	SDC	9D998747-1281-4474-9C45...	0
<input type="checkbox"/>	austin248-sdc	172.16.50.248 +1	SDC	3DD6226F-72E6-447E-B6A...	0

**Figure 21. KVM hosts in PowerFlex UI**

The hosts all show no mapped volumes. Prior to creating and mapping a new volume, however, an additional configuration for multipathing is required on each KVM host as detailed next.

## Multipathing

The PowerFlex does not use, nor require, Linux multipathing functionality; however, for the purposes of Fibre Channel device discovery, Oracle Linux Virtualization Manager does. The storage domain wizard requires that for a device to be recognized as FC, the multipath daemon must recognize it. Unfortunately, by default the multipath.conf file is configured to filter out most devices by exception. This includes PowerFlex devices which have the moniker `/dev/scini<x>`. In order for the multipath daemon to recognize the PowerFlex devices, the user must add an exception. A new directory and file should be created under the `/etc/multipath` directory. Using an editor like vi, add the following syntax to the configuration file:

```
mkdir -p /etc/multipath/conf.d
vi /etc/multipath/conf.d/powerflex.conf
```

Enter the following information and save.

```
blacklist_exceptions {
devnode "scini*"
```

The file entry should appear as in [Figure 22](#):

```
# You can also add exceptions to blacklist (see man(5) multipath.conf for more
# details) by adding blacklist_exceptions to your drop-in configuration file,
# e.g.
#
#   blacklist_exceptions {
#       devnode "scini*"
#   }
}
```

**Figure 22. Multipath blacklist exceptions**

After modifying the file, restart the multipath daemon so that it picks up the changes:

```
systemctl restart multipathd
```

Once complete, the user can provision PowerFlex volumes to the Oracle Linux KVM nodes for use as FC storage domains.

## Provision volume

To provision the volume, follow these steps:

1. Open the PowerFlex UI and navigate to **Block -> Volume -> + Create Volume**.
2. Enter the following information in [Figure 23](#):
  - Number of volumes
  - Volume name
  - Provisioning
  - Size
  - Select Storage Pool

**Figure 23. Volume creation**

3. Click **Create**.
4. Map the created volume. Check the box next to the newly created volume, then navigate to the menu **Mapping > Map**. Check the box next to each KVM host and click **Map** in Figure 24.

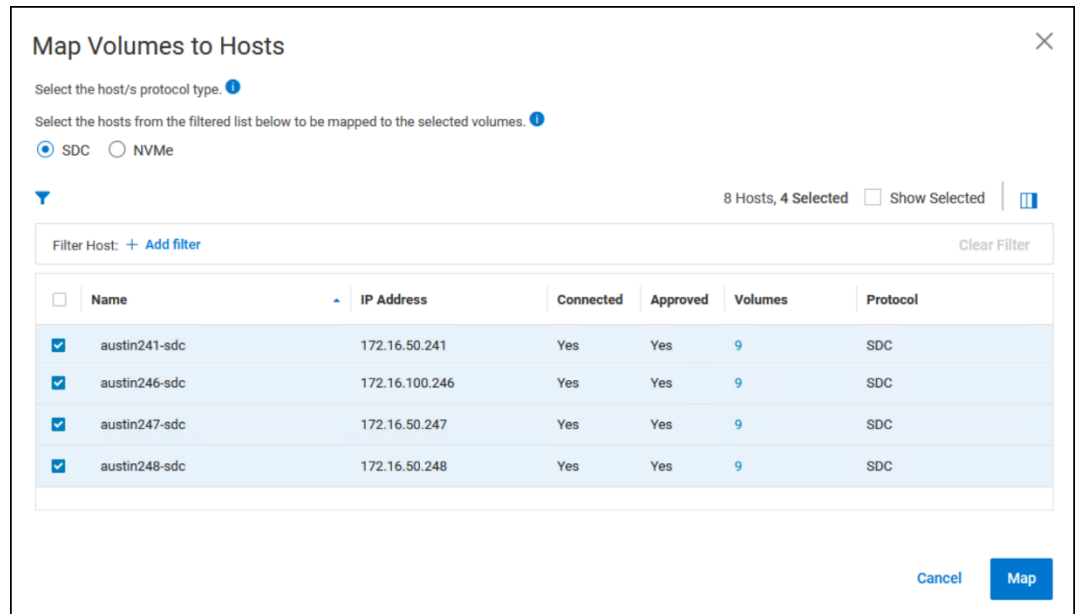


Figure 24. Map volume to host

The volume should now be available on each KVM node. To check, list the available volumes on any of the nodes by running `multipath -ll`. The 240 GB volume is highlighted in Figure 25.

```
[root@austin241 tmp]# multipath -ll
13101880e963d20f-8eabab7b00000002 dm-36 ##,##
size=504G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  - #:#:#:# scinih 251:112 active ready running
13101880e963d20f-8eabab830000000a dm-79 ##,##
size=704G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  - #:#:#:# scinii 251:128 active ready running
13101880e963d20f-8eabab840000000b dm-89 ##,##
size=240G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  - #:#:#:# scinij 251:144 active ready running
MTFDDAV480TDS_22303B9B6450 dm-3 ATA,MTFDDAV480TDS
size=447G features='1 queue_if_no_path' hwhandler='0' wp=rw
```

Figure 25. PowerFlex device using multipath command

## FC storage domain

With the volume recognized by multipath, create the FC storage domain.

1. In Oracle Linux Virtualization Manager, navigate in the left-hand menu to **Storage > Domains**. In the right-hand corner click **New Domain** to start the wizard.

- Begin by selecting the wanted **Host**. The volume must be presented to all hosts in the cluster or storage domain creation fails. Next, use the **Storage Type** drop-down box and select **Fibre Channel**. Oracle Linux Virtualization Manager immediately initiates a scan of the host and returns the available volumes as in [Figure 26](#). Any PowerFlex volumes do not display a Product ID and do not have a prefix. Enter a **Name** and **Description** (if wanted). Then click **Add** next to the correct PowerFlex volume. Leave all other **Advanced Parameters** as default unless the business requires a change. Click **OK**.

**New Domain**

Data Center: Default (V5) | Name: Oracle\_Volume\_1

Domain Function: Data | Description: Volume for an Oracle database

Storage Type: Fibre Channel | Comment:

Host: austin241.dellhcllab.com

LUN ID	Size	#path	Vendor ID	Product ID	Serial	Add
13101880e963d20f-8eabab7b00000002	1788 GiB	1				N/A
eui.000000000000000008ce38ee20cd4f701	1788 GiB	1		Dell Ent NVMe		Add
13101880e963d20f-8eabab7b00000002	504 GiB	1				N/A
eui.000000000000000008ce38ee20cd4ee01	1788 GiB	1		Dell Ent NVMe		Add
13101880e963d20f-8eabab7e00000005	2000 GiB	1				N/A
eui.000000000000000008ce38ee20cd03901	1788 GiB	1		Dell Ent NVMe		Add
eui.000000000000000008ce38ee20cd03101	1788 GiB	1		Dell Ent NVMe		Add
13101880e963d20f-8eabab7c00000003	2000 GiB	1				N/A
<b>13101880e963d20f-8eabab840000000b</b>	<b>240 GiB</b>	<b>1</b>				<b>Add</b>
eui.000000000000000008ce38ee21290bc01	1788 GiB	1		Dell Ent NVMe		Add

OK Cancel

**Figure 26. FC storage domain**

**Note:** Although it is possible to select multiple devices for use in a single storage domain, Dell Technologies does not recommend doing so. The user should rely on the PowerFlex array for performance and therefore present a single volume of the total size required.

Oracle Linux Virtualization Manager initiates the FC storage domain, locking it until completion. When done, the storage domain displays as active in [Figure 27](#).

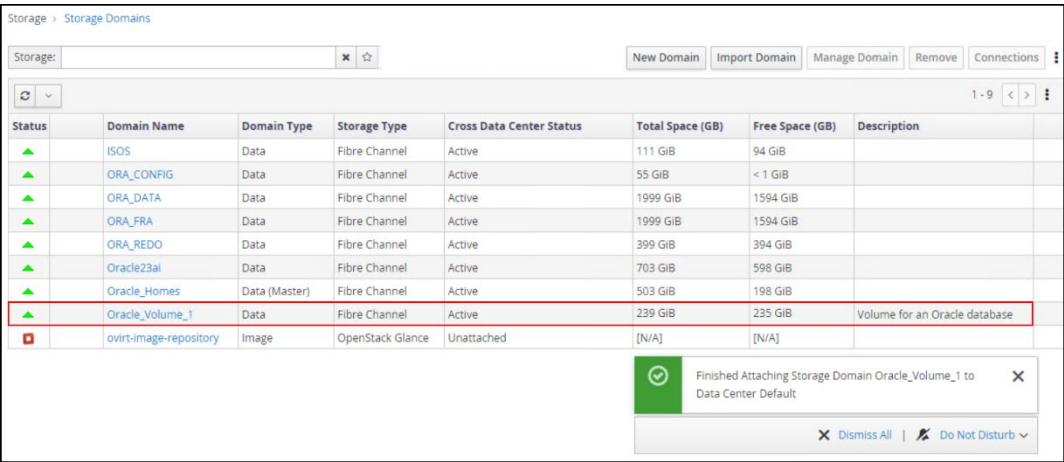


Figure 27. Active FC storage domain

# Deploying Oracle Real Application Clusters

This section provides an architecture overview and the steps to follow in setting up a 3-node Oracle Real Application Clusters (RAC) database using the Oracle Linux Virtualization Manager on a two-layer PowerFlex setup. This is provided only as an example to illustrate how PowerFlex can enable a business to run an Oracle Linux KVM environment with Oracle RAC. The sizing of the ASM disk groups and the database are completely arbitrary. Best practices are included, however, and apply to any deployment of this type in production.

## Logical architecture

The following figure shows a logical view of the 3-node setup:

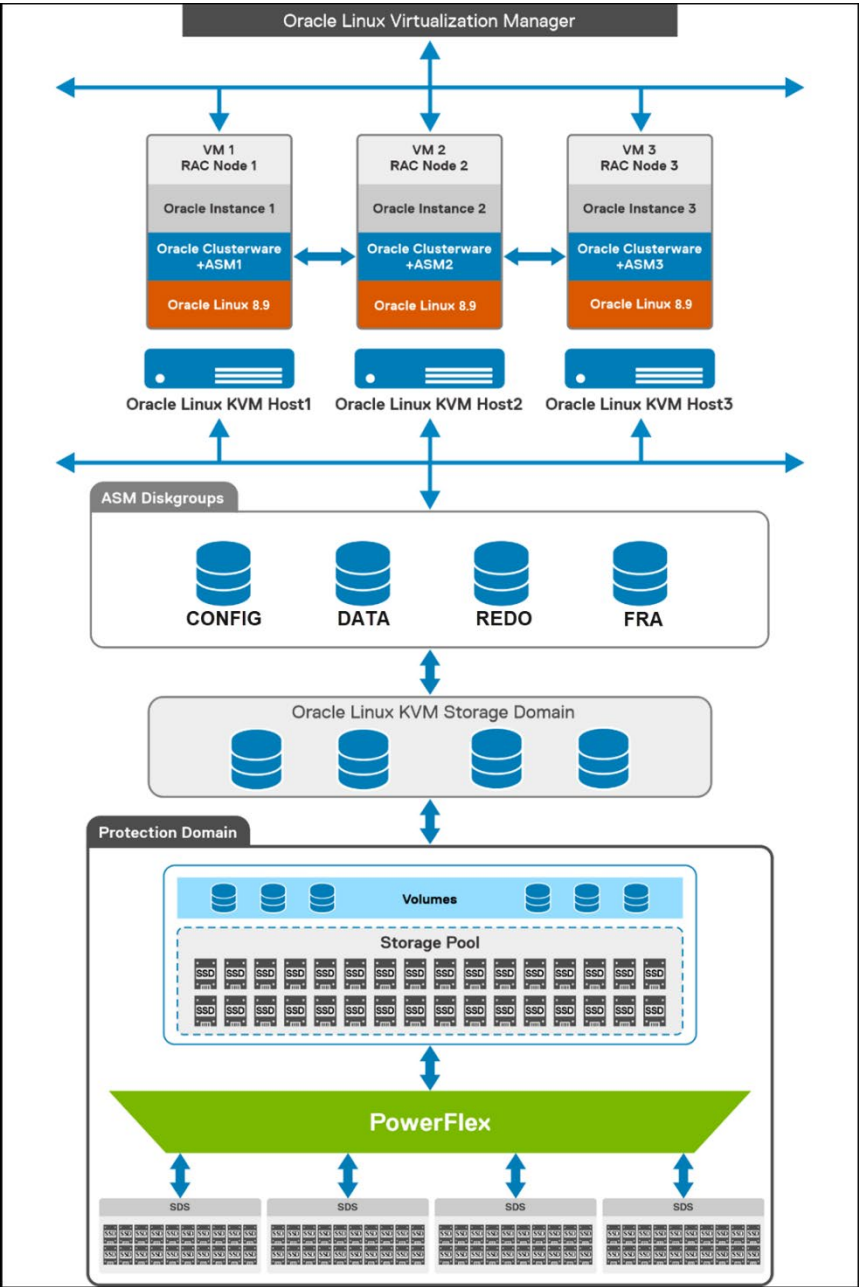


Figure 28. Logical architecture

In the two-layer PowerFlex system, the SDC is installed on the compute-only host (Oracle Linux KVM), while the MDM and SDS components are installed on backend, storage-only nodes. The SDS aggregates and serves raw local storage in each node and shares that storage as part of the PowerFlex cluster. A single Storage Pool is created using all the disks on each node within the Protection Domains, volumes are then provisioned from the Storage Pool and presented to the compute hosts, which Oracle Linux Virtualization Manager uses as storage domains. From the storage domain, respective size disks are carved out to meet the Oracle RAC ASM disk group database requirements, including volumes for data, redo logging, voting disk, and the flash recovery area. The volumes are mapped and shared between the virtual machines and then consumed by ASM to create the groups. While the Oracle Grid and database software is installed independently on each VM, the Oracle RAC database itself is built on ASM and thus made available to all the nodes.

Network architecture

The following networks and VLANs were used in the lab for this Oracle Linux KVM solution:

Table 3. PowerFlex networking details at host level

Network name	Description
Bond0 (p2p1, p3p1)	Management and VM Traffic
Bond1 (p3p2, p2p2)	PowerFlex data traffic (SDS and SDC)

Table 4. Oracle Linux KVM networking details at VM level

Network name	VLAN	Description
ovirtmgmt	105	Management Network
Privatevlan106	106	Private vlan for Oracle private interconnect
VM_Network	100	Client Oracle network

VLAN tagging

Oracle Linux Virtualization Manager supports adding multiple logical networks to physical NICs on the Oracle Linux KVM node, including those with VLAN tagging. As VLANs are an essential component of the PowerFlex architecture, the steps for adding a new logical network with VLAN tagging for the Oracle interconnect are included here.

1. Navigate to **Network -> Networks** screen in Oracle Linux Virtualization Manager and click **New** in [Figure 29](#).

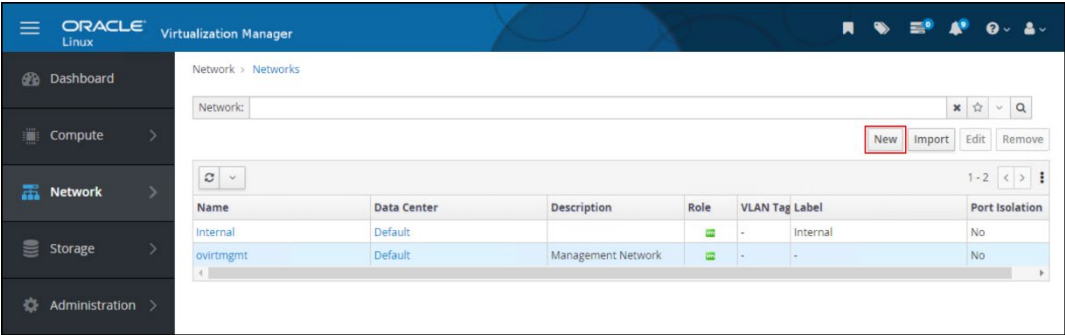


Figure 29. Logical networks

Enter the following information in Figure 30:

- Name
- Description
- Network label
- Check the box for Enable VLAN tagging and add the VLAN value

Leave the Cluster as default (it attaches automatically), and the vNIC Profiles (the name defaults to the network name).

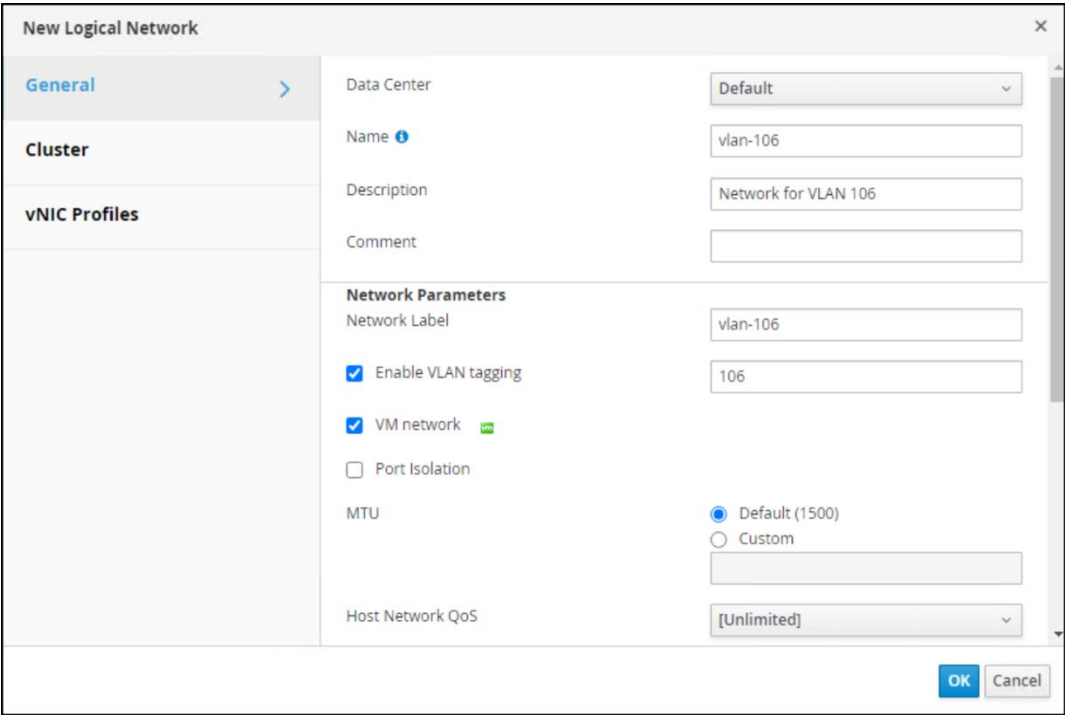


Figure 30. New logical network

- Once created, navigate to **Network -> Networks** and click the newly created hyperlink for the **vlan-106** network.
- Click the Hosts tab, highlight one of the unattached hosts, and click **Setup Host Networks** in Figure 31.

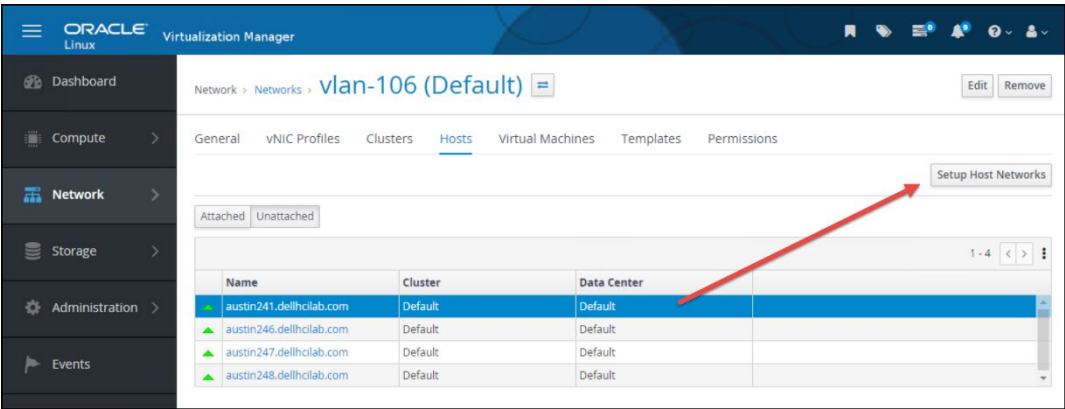


Figure 31. vlan-106 host assignment

4. The **Setup Host Networks** dialog appears. The new logical network appears on the right side. Click the network and drag it to the appropriate physical NIC as shown in Figure 32. As here, more than one logical network can be assigned to an interface.

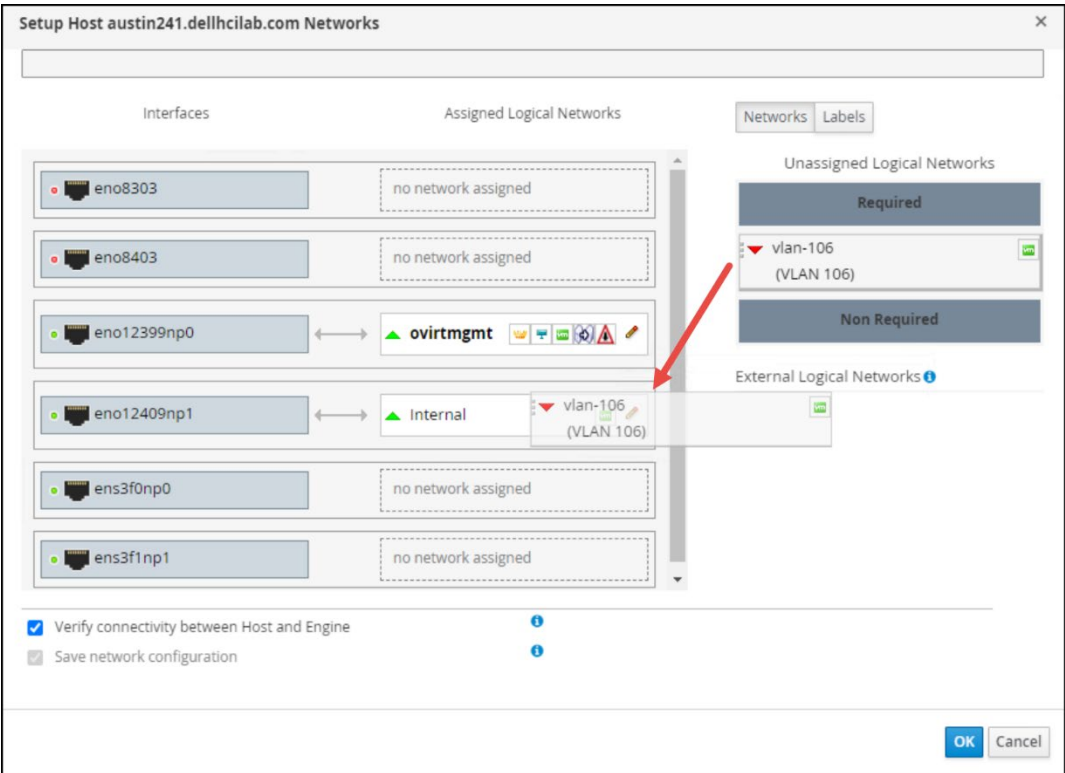


Figure 32. Assign a logical network to interface

5. Next, click the pencil icon in the corner of the logical network. This allows the user to assign an IP address (if wanted). Choose the appropriate **Boot Protocol**, add an address if required, and click **OK** in Figure 33. Oracle Linux Virtualization Manager then configures the network on the host.

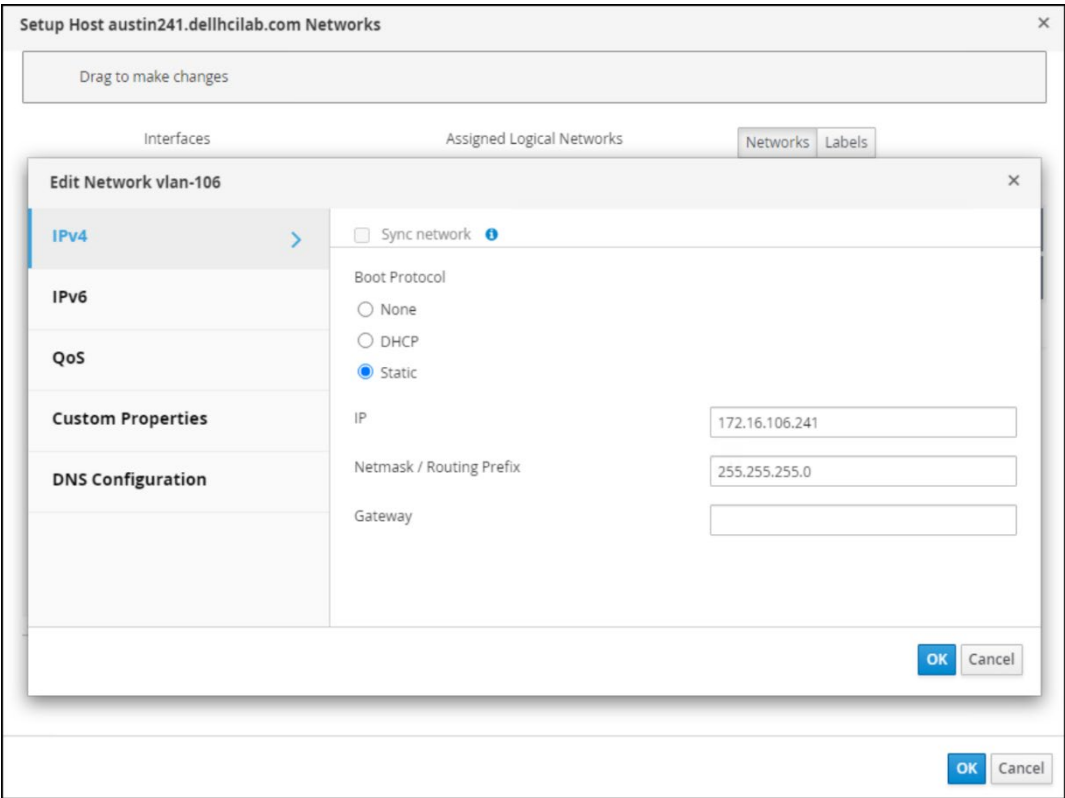


Figure 33. Assign boot protocol and IP

The logical network is created and configured in [Figure 34](#).

```
48: vlan-106: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether e8:eb:d3:84:30:17 brd ff:ff:ff:ff:ff:ff
    inet 172.16.106.241/24 brd 172.16.106.255 scope global noprefixroute vlan-106
        valid_lft forever preferred_lft forever
49: eno12409np1.106@eno12409np1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master vlan-106 state UP group default qlen 1000
    link/ether e8:eb:d3:84:30:17 brd ff:ff:ff:ff:ff:ff
```

Figure 34. IP assigned

Oracle RAC  
configuration

The following section provides details on setting up Oracle Linux KVM and installing a 3-node Oracle RAC 21c database.

Hardware and software configuration details

The following table describes the hardware and software components of the infrastructure used for the solution. Both the PowerFlex (storage-only) nodes and those used for Oracle Linux KVM (compute-only) are the same:

Table 5. Hardware and software configuration

Components	Source domain
Server model	Dell R650
Number of compute-only nodes	3
Number of storage-only nodes	4
CPU	Intel® Xeon® Gold 6336Y CPU @ 2.40 GHz

Components	Source domain
Sockets and cores	2 socket 24 cores
Hyperthreading	Enabled
Memory	512 GB per host
Storage	2 x 447.13 GB (SATA SSD) 10 x 1490.42 GB (SAS SSD)
PCIe	Mellanox ConnectX-5 EN 25 GbE SFP28 Adapter (2 ports)
NVDIMM	2 x 16 GB, 2933 MT/s NVDIMM-N DDR-4
PowerFlex	R4_6
PowerFlex Manager	Version 4.6.0
Oracle Linux Virtualization Manager	4.5.4-1.0.31.el8
Oracle Linux	Release 8 Update 9
Oracle Database version	21.3.0.0.0
VM OS - Oracle Linux	Release 8 Update 9
Number of VMs	3
VM configuration	16 vCPU, 24 GB Memory
VM nodes	austin170, austin171, austin172
Database name	orcl
Instance names	orcl1, orcl2, orcl3
ASM disk groups	CONFIG, DATA, REDO, FRA

## Host configuration

Concurrent to installing the Oracle Linux Virtualization Manager on its own host, users need to prepare the Oracle Linux KVM hosts which will also serve as the PowerFlex compute nodes.

Take the following steps to install an Oracle Linux KVM host for the Oracle RAC environment:

- Install Oracle Linux 8.9 OS on each of the compute hosts.
- Configure management networking for each host. Assign an IP address to each host.
- Configure networking to support SDC connectivity to the PowerFlex.
- Perform the following commands on each of the hosts, to prepare the host for receiving commands from oVirt Engine:

```
dnf config-manager --enable ol8_baseos_latest
dnf install oracle-ovirt-release-45-el8 -y
dnf clean all
dnf repolist
```

## Oracle Linux Virtualization Manager

To install Oracle Linux Virtualization Manager, perform the following steps:

- Create the VM and install the Oracle Linux 8.9 OS using the **Virtualization Host Base Environment**. Choosing a different base can lead to issues with the implementation. This base does not come with a UI but Gnome Desktop can be added postinstall if desired.
- Install the oVirt Engine package and install engine by performing the following commands:

```
dnf config-manager --enable ol8_baseos_latest
dnf install oracle-ovirt-release-45-el8 -y
dnf clean all
dnf repolist
dnf install ovirt-engine
```

- Perform the engine-setup to install Oracle Linux Virtualization Manager.  
`engine-setup`
- Once installation completes, the user is provided with a web URL, which is the FQDN of the host, to access the virtualization manager.

## Storage domains for ASM

The following table provides details of storage domains created from PowerFlex and mapped to the Oracle Linux KVM required for Oracle ASM disks. PowerFlex volumes must be sized in factors of 8.

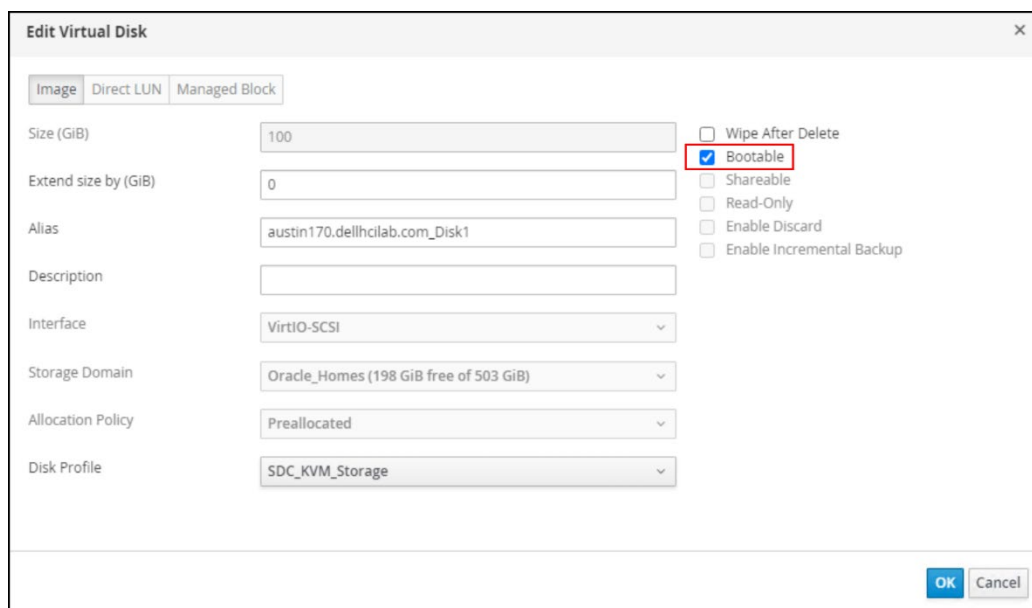
**Table 6. Storage domains used for the Oracle RAC database**

Storage domain	Size	Description
Oracle_Homes	504 GB	To be used for OS file system for VM as well as the Oracle software
ORA_CONFIG	56 GB	To be used for CONFIG ASM disk group
ORA_REDO_1	56 GB	To be used for REDO ASM disk group
ORA_REDO_2	56 GB	To be used for REDO ASM disk group
ORA_REDO_3	56 GB	To be used for REDO ASM disk group
ORA_DATA_1	504 GB	To be used for DATA ASM disk group
ORA_DATA_2	504 GB	To be used for DATA ASM disk group
ORA_DATA_3	504 GB	To be used for DATA ASM disk group
ORA_FRA_1	504 GB	To be used for FRA ASM disk group
ORA_FRA_2	504 GB	To be used for FRA ASM disk group
ORA_FRA_3	504 GB	To be used for FRA ASM disk group

## VM configuration

The following steps were used in this configuration to set up the 3-node Oracle RAC database with Oracle Linux Virtualization Manager running on PowerFlex:

- Create VMs, one VM per host. Install Oracle Linux 8.9 OS.
  - Create 3 x 100 GB virtual disks, from **Oracle\_Homes**, to be used for OS installation for the VM file system, one for each VM.
  - These disks are to be made “bootable”



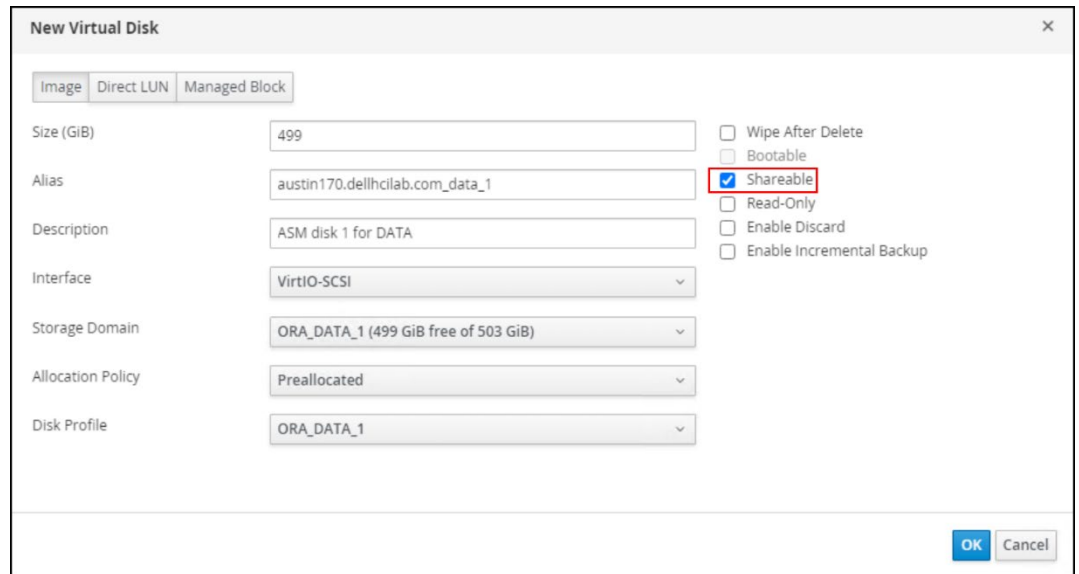
**Figure 35. Disk being made bootable for OS Installation**

- Install Oracle Linux 8.9 OS on each VM and assign IPs for each VM. The installation can be a **Base Environment** of **Server with GUI** or **Server**.
- Create the necessary disks from the storage domain required for ASM disk groups DATA, OCR, MGMT REDO and FRA.

**Table 7. ASM disks from storage domains**

ASM disk groups	Size	From storage domain
CONFIG	1 x ~50 GB	ORA_CONFIG
OCR	3 x ~50 GB	ORA_REDO_1, ORA_REDO_2, ORA_REDO_3
DATA	3 x ~500 GB	ORA_DATA_1, ORA_DATA_2, ORA_DATA_3
FRA	3 x ~500 GB	ORA_FRA_1, ORA_FRA_2, ORA_FRA_3

- Attach the ASM disks to all the VMs by making them shareable.



**Figure 36. ASM disks being made shareable for Oracle RAC database installation**

- There are 3 interfaces to choose from:
  - IDE
    - Standard interface connecting to storage devices. In terms of performance, it is slightly slower than VirtIO or VirtIO-SCSI
  - VirtIO
    - A para-virtualized driver offers increased I/O performance over emulated devices, for example IDE, by optimizing the coordination and communication between the virtual machine and the hypervisor.
  - VirtIO-SCSI
    - A newer para-virtualized SCSI controller device. This driver offers similar functionality to virtIO devices with some additional enhancements such as improved scalability, a standard command set, and SCSI device passthrough. Specifically, it supports adding hundreds of devices and the naming of those devices using the standard SCSI device naming scheme.

---

**Note:** The configuration in the lab used VirtIO-SCSI devices since it is recommended for better I/O performance.

---

- Dell Technologies recommends selecting high-performance optimization for Virtual Machines (VMs). By doing so, the VMs run with performance metrics as close to bare metal as possible. When high performance is chosen, the VM is configured with a set of automatic and recommended manual settings for maximum efficiency.
- 

**Note:** For additional information about high performance settings, see [Configuring High-Performance Virtual Machines](#).

---

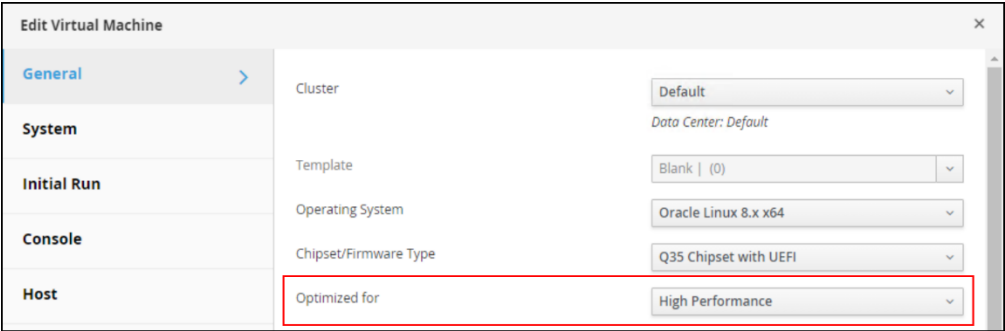


Figure 37. Virtual Machines configuration displaying high performance

- Configure additional networks, such as interconnect for Oracle RAC.

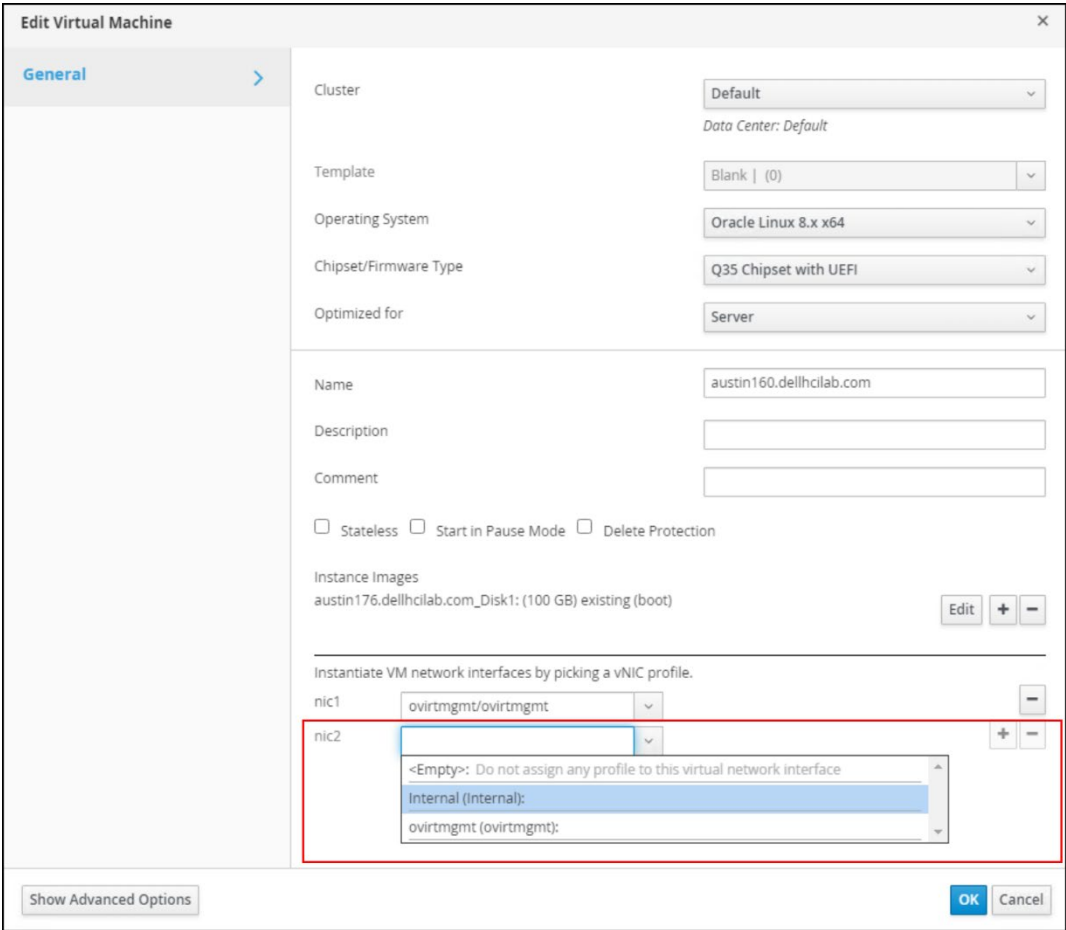


Figure 38. Additional networking for Oracle interconnect

- Disable headless mode for each VM for optimization. Users can configure a VM in headless mode when it is not necessary to access the VM using a graphical console. By disabling headless mode, the VM runs without graphical and video devices. This is useful in situations where the host has limited resources.

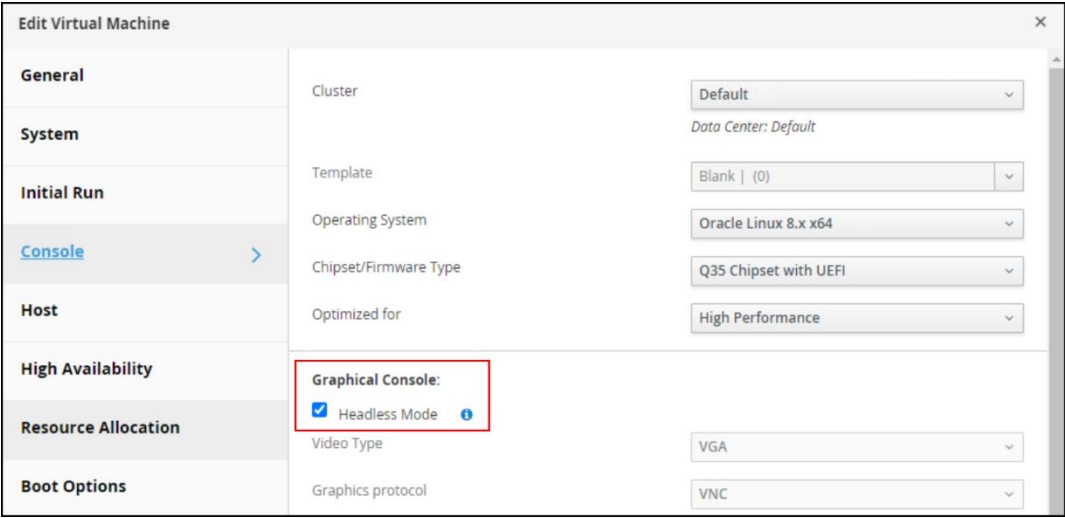


Figure 39. Disabling headless mode for VM

- Run the VM on a specific host in the cluster so that the Oracle RAC VMs are spread across hosts in the Oracle Linux KVM cluster and to adhere to CPU pinning requirements.

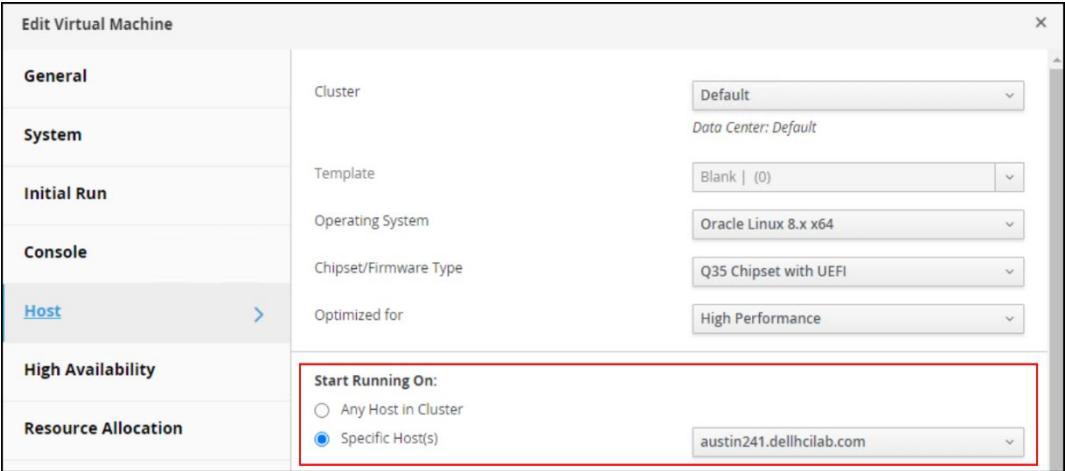


Figure 40. Selection for VM to run on specific host in the cluster

- Install Oracle Grid Infrastructure and Database 21c software and create the database.

### Best practices

The following are some best practices when running Oracle RAC on ASM with PowerFlex and Oracle Linux KVM.

- If possible, use different ASM disk groups for each database function. The groups should use external redundancy. This provides for greater flexibility.
  - DATA for data
  - REDO for redo logs
  - FRA for archive logs
  - CONFIG for voting disk

- Use multiple storage domains for each ASM disk group with a single, shared virtual disk in each that consumes the space. This makes it easier to increase or decrease ASM disk groups and provide more concurrency.
- On each VM, the shareable disks must be owned by oracle with a permission mode of 0660.
- Members of an ASM disk group should be of similar capacity. If devices are initially sized large, each capacity increment to the ASM disk group will need to be as large.
- Oracle ASM best practice is to add multiple devices together to increase the ASM disk group capacity rather than adding one device at a time. This method spreads ASM extents during rebalance to avoid hot spots. Use a device size that allows for ASM capacity increments, in which multiple devices are added to the ASM disk group together. Each device should have the same size as its original device.

## Conclusion

This reference architecture guide demonstrates the deployment of Oracle Linux KVM on a PowerFlex infrastructure. Oracle Linux KVM is an open-source hypervisor that offers many of the same enterprise-level features of its more well-known competitors, while providing excellent performance and scalability at minimal cost. For many businesses running on PowerFlex with Linux, it can fulfill their virtualization requirements, offering a flexible platform that can be customized to meet their needs.

## References

### Dell Technologies documentation

The following Dell Technologies documentation provides additional 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 Solutions Documents\*](#)
- [\*Oracle RAC and Dell PowerFlex Asynchronous Replication\*](#)

### Oracle documentation

The following Oracle documentation provides additional information:

- [\*Oracle Virtualization Documentation\*](#)
- [\*Certified Virtualization and Partitioning Technologies for Oracle Database and RAC Product Releases\*](#)
- [\*Oracle Real Application Clusters 21c Technical Architecture\*](#)
- [\*Real Application Clusters Installation Guide for Linux and UNIX\*](#)
- [\*Oracle Linux Premier Support\*](#)
- [\*Oracle Database and Oracle Real Application Cluster on Oracle Linux KVM\*](#)
- [\*Hard Partitioning with Oracle Linux KVM\*](#)

### oVirt documentation

The following documentation provides more information about the oVirt open-source project:

- [\*oVirt documentation\*](#)