Omnia

Release 1.4

Cassey Goveas

Mar 30, 2023

CONTENTS

1 Omnia: Overview 3
- 1.1 Architecture . . . 4
- 1.2 New Features . . . 5
- 1.3 Releases . . . 6
- 1.4 Support Matrix . . . 9
- 1.5 Network Topologies . . . 12
- 1.6 Blogs about Omnia . . . 16
- 1.7 What Omnia does . . . 16

2 Quick Installation Guide 17
- 2.1 Running prereq.sh . . . 19
- 2.2 Installing The Provision Tool . . . 19
- 2.3 Building Clusters . . . 24
- 2.4 Configuring Switches . . . 32
- 2.5 Configuring Storage . . . 40

3 Roles 43
- 3.1 Provision . . . 43
- 3.2 Network . . . 48
- 3.3 Scheduler . . . 48
- 3.4 Storage . . . 57
- 3.5 Accelerator . . . 61
- 3.6 Monitor . . . 62
- 3.7 Utils . . . 62

4 Troubleshooting 67
- 4.1 Known Issues . . . 67
- 4.2 Frequently Asked Questions . . . 75
- 4.3 Troubleshooting Guide . . . 76

5 Security Configuration Guide 79
- 5.1 Ports used by Omnia . . . 79
- 5.2 Authentication . . . 80

6 Sample Files 83
- 6.1 inventory file . . . 83
- 6.2 pxe_mapping_file.csv . . . 83
- 6.3 switch_inventory . . . 83
- 6.4 powervault_inventory . . . 83
- 6.5 NFS Server inventory file . . . 84

7 Limitations **85**

8 Best Practices **87**

9 Contributing To Omnia **89**

9.1 Creating A Pull Request 89

Ansible playbook-based deployment of Slurm and Kubernetes on servers running an RPM-based Linux OS.

Omnia (Latin: all or everything) is a deployment tool to turn servers with RPM-based Linux images into functioning Slurm/Kubernetes clusters.

Licensing

Omnia is made available under the Apache 2.0 license.

Note: Omnia playbooks are licensed under the Apache 2.0 license. Once an end-user initiates Omnia, that end-user will enable deployment of other open source software that is licensed separately by their respective developer communities. For a comprehensive list of software and their licenses, click here . Dell (or any other contributors) shall have no liability regarding and no responsibility to provide support for an end-users use of any open source software and end-users are encouraged to ensure that they are complying with all such licenses. Omnia is provided "as is" without any warranty, express or implied. Dell (or any other contributors) shall have no liability for any direct, indirect, incidental, punitive, special, or consequential damages for an end-users use of Omnia.

For a better understanding of what Omnia does, check out our docs!

Omnia Community Members

DELL Technologies

UNIVERSITÀ DI PISA

Arizona State University

VIZIAS

TEXAS TECH UNIVERSITY

LIQID

Table Of Contents

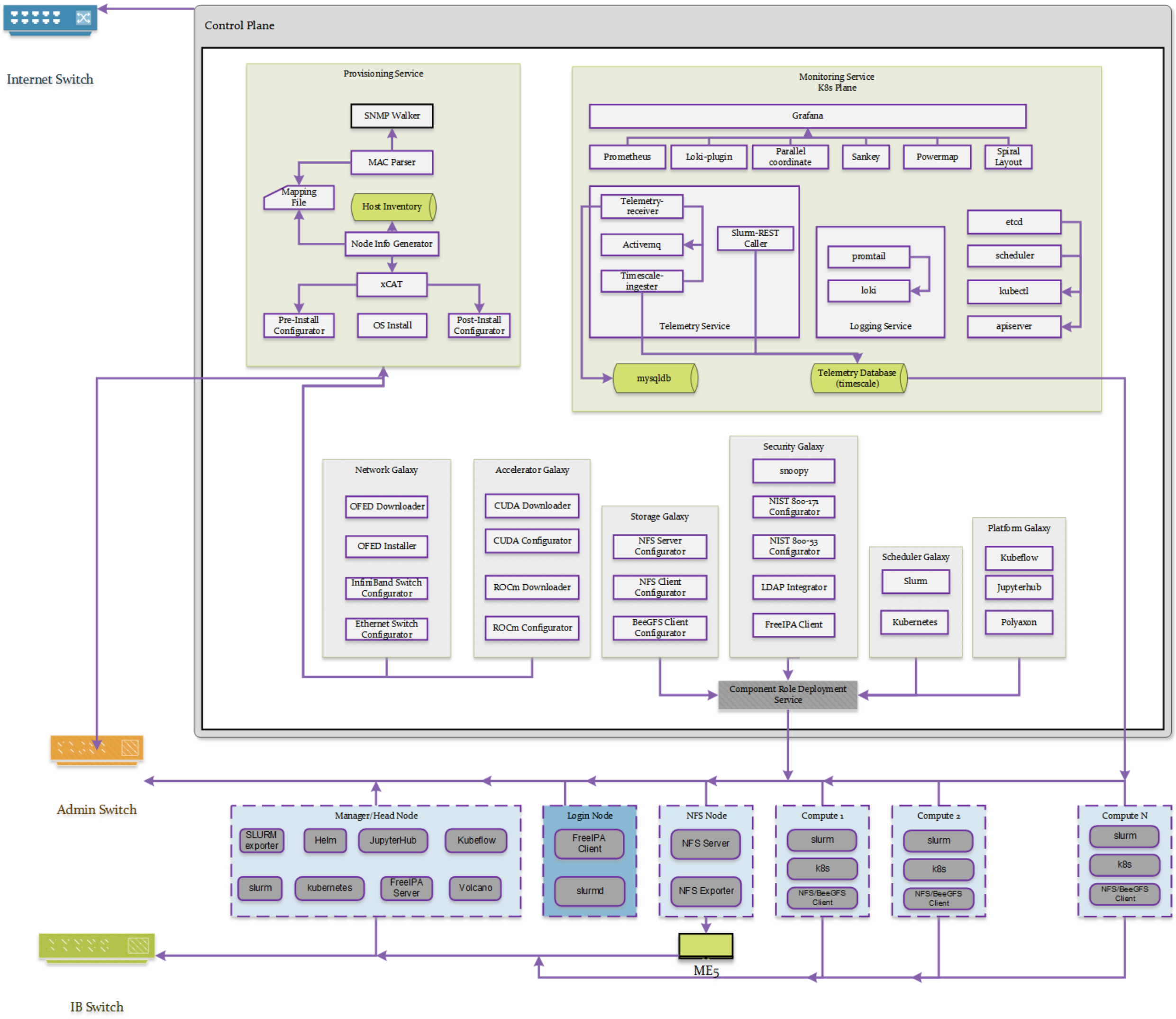
CHAPTER

ONE

OMNIA: OVERVIEW

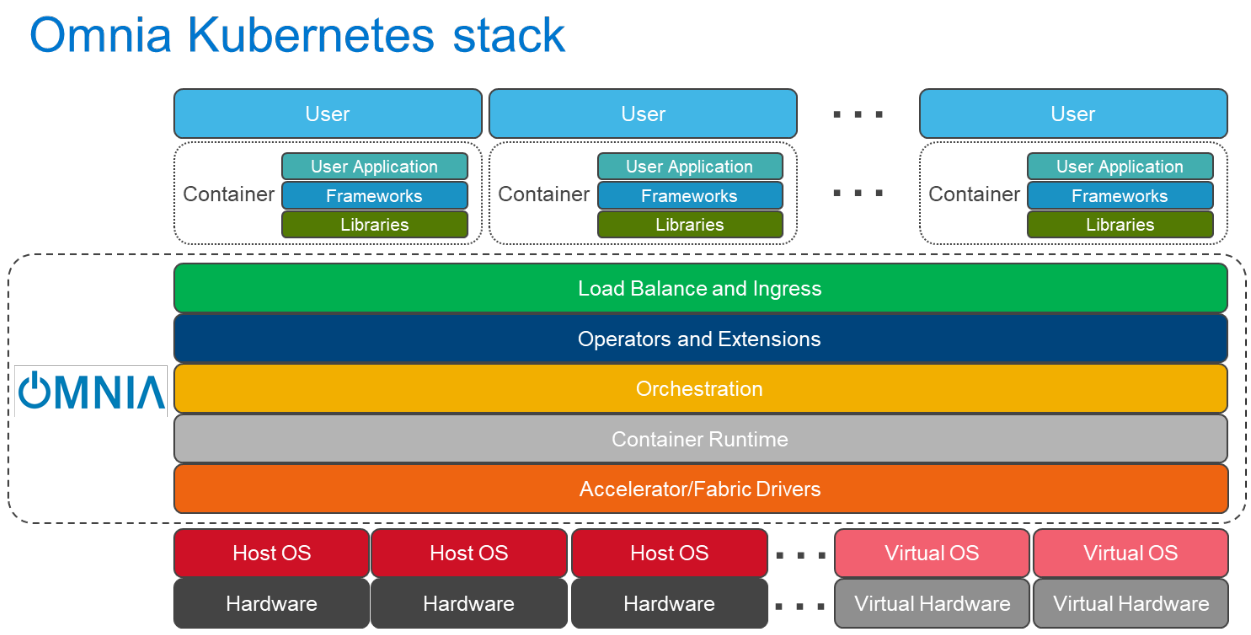
Omnia (Latin: all or everything) is a deployment tool to configure Dell PowerEdge servers running standard RPM-based Linux OS images into clusters capable of supporting HPC, AI, and data analytics workloads. It uses Slurm, Kubernetes, and other packages to manage jobs and run diverse workloads on the same converged solution. It is a collection of Ansible playbooks, is open source, and is constantly being extended to enable comprehensive workloads.

1.1 Architecture

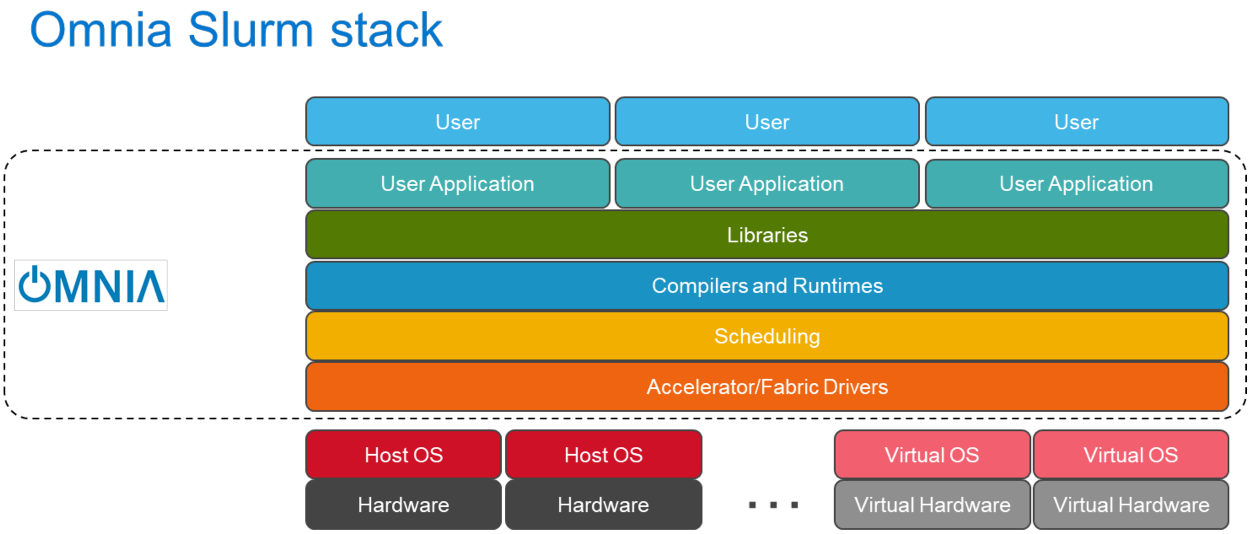


1.1.1 Omnia stack

Kubernetes



Slurm



1.2 New Features

1.2.1 Control Plane

- Omnia Prerequisites Installation
- Provision Tool - xCAT installation
- Node Discovery using Switch IP Address
- Provisioning of remote nodes using

  - Mapping File
  - Auto discovery of nodes from Switch IP
- Database Update of remote nodes info that includes,
  - Host or Admin IP
  - iDRAC IP
  - InfiniBand IP
  - Hostname
- Inventory creation on Control Plane

1.2.2 Cluster

- iDRAC and InfiniBand IP Assignment on remote nodes (nodes in the cluster)
- Installation and Configuration of:
  - NVIDIA Accelerator and CUDA Toolkit
  - AMD Accelerator and ROCm
  - OFED
  - LDAP Client

1.2.3 Device Support

- InfiniBand Switch Configuration with port split functionality
- Ethernet Z-Series Switch Configuration with port split functionality

1.3 Releases

1.3.1 1.4

- Provisioning of remote nodes through PXE boot by providing TOR switch IP
- Provisioning of remote nodes through PXE boot by providing mapping file
- PXE provisioning of remote nodes through admin NIC or shared LOM NIC
- Database update of mac address, hostname and admin IP
- Optional monitoring support(Grafana installation) on control plane
- OFED installation on the remote nodes
- CUDA installation on the remote nodes
- AMD accelerator and ROCm support on the remote nodes
- Omnia playbook execution with Kubernetes, Slurm & FreeIPA installation in all compute nodes
- Infiniband switch configuration and split port functionality
- Added support for Ethernet Z series switches.

1.3.2 1.3

- CLI support for all Omnia playbooks (AWX GUI is now optional/deprecated).
- Automated discovery and configuration of all devices (including PowerVault, InfiniBand, and ethernet switches) in shared LOM configuration.
- Job based user access with Slurm.
- AMD server support (R6415, R7415, R7425, R6515, R6525, R7515, R7525, C6525).
- PowerVault ME5 series support (ME5012, ME5024, ME5084).
- PowerVault ME4 and ME5 SAS Controller configuration and NFS server, client configuration.
- NFS bolt-on support.
- BeeGFS bolt-on support.
- Lua and Lmod installation on manager and compute nodes running RedHat 8.x, Rocky 8.x and Leap 15.3.
- Automated setup of FreeIPA client on all nodes.
- Automate configuration of PXE device settings (active NIC) on iDRAC.

1.3.3 1.2.2

- Bugfix patch release to address AWX Inventory not being updated.

1.3.4 1.2.1

- HPC cluster formation using shared LOM network
- Supporting PXE boot on shared LOM network as well as high speed Ethernet or InfiniBand path.
- Support for BOSS Control Card
- Support for RHEL 8.x with ability to activate the subscription
- Ability to upgrade Kernel on RHEL
- Bolt-on Support for BeeGFS

1.3.5 1.2.0.1

- Bugfix patch release which address the broken cobbler container issue.
- Rocky 8.6 Support

1.3.6 1.2

- Omnia supports Rocky 8.5 full OS on the Control Plane
- Omnia supports ansible version 2.12 (ansible-core) with python 3.6 support
- All packages required to enable the HPC/AI cluster are deployed as a pod on control plane
- Omnia now installs Grafana as a single pane of glass to view logs, metrics and telemetry visualization
- Compute node provisioning can be done via PXE and iDRAC
- Omnia supports multiple operating systems on the cluster including support for Rocky 8.5 and OpenSUSE Leap 15.3
- Omnia can deploy compute nodes with a single NIC.
- All Cluster metrics can be viewed using Grafana on the Control plane (as opposed to checking the manager node on each cluster)
- AWX node inventory now displays service tags with the relevant operating system.
- Omnia adheres to most of the requirements of NIST 800-53 and NIST 800-171 guidelines on the control plane and login node.
- Omnia has extended the FreeIPA feature to provide authentication and authorization on Rocky Nodes.
- Omnia uses [389ds}(https://directory.fedoraproject.org/) to provide authentication and authorization on Leap Nodes.
- Email Alerts have been added in case of login failures.
- Administrator can restrict users or hosts from accessing the control plane and login node over SSH.
- Malicious or unwanted network software access can be restricted by the administrator.
- Admins can restrict the idle time allowed in an ssh session.
- Omnia installs apparmor to restrict program access on leap nodes.
- Security on audit log access is provided.
- Program execution on the control plane and login node is logged using snoopy tool.
- User activity on the control plane and login node is monitored using psacct/acct tools installed by Omnia
- Omnia fetches key performance indicators from iDRACs present in the cluster
- Omnia also supports fetching performance indicators on the nodes in the cluster when SLURM jobs are running.
- The telemetry data is plotted on Grafana to provide better visualization capabilities.
- Four visualization plugins are supported to provide and analyze iDRAC and Slurm data.
  - Parallel Coordinate
  - Spiral
  - Sankey
  - Stream-net (aka. Power Map)
- In addition to the above features, changes have been made to enhance the performance of Omnia.

1.4 Support Matrix

1.4.1 Hardware Supported by Omnia

Servers

PowerEdge servers

Server Type	Server Model
14G	C4140 C6420 R240 R340 R440 R540 R640 R740 R740xd R740xd2 R840 R940 R940xa
15G	C6520 R650 R750 R750xa

AMD servers

Server Type	Server Model
14G	R6415 R7415 R7425
15G	R6515 R6525 R7515 R7525 C6525

New in version 1.2: 15G servers

New in version 1.3: AMD servers

Storage

Powervault Storage

Storage Type	Storage Model
ME4	ME4084 ME4024 ME4012
ME5	ME5012 ME5024 ME5084

New in version 1.3: PowerVault ME5 storage support

BOSS Controller Cards

BOSS Controller Model	Drive Type
T2GFX	EC, 5300, SSD, 6GBPS SATA, M.2, 512E, ISE,240GB
M7F5D	EC, S4520, SSD, 6GBPS SATA, M.2, 512E, ISE, 480GB

New in version 1.2.1: BOSS controller cards

Switches

Switch Type	Switch Model
Mellanox InfiniBand Switches	NVIDIA MQM8700-HS2F Quantum HDR InfiniBand Switch 40 QSFP56

Switch Type	Switch Model
Dell Networking Switches	PowerSwitch S3048-ON PowerSwitch S5232F-ON PowerSwitch Z9264F-ON

Note:

- The switches that have reached EOL might not function properly. It is recommended by Omnia to use the switch models mentioned in support matrix.
- Omnia requires that OS10 be installed on ethernet switches.
- Omnia requires that MLNX-OS be installed on Infiniband switches.

1.4.2 Operating Systems

Red Hat Enterprise Linux

OS Version	Control Plane	Compute Nodes
8.1	No	Yes
8.2	No	Yes
8.3	No	Yes
8.4	Yes	Yes
8.5	Yes	Yes
8.6	Yes	Yes

Note:

- Always deploy the DVD Edition of the OS on compute nodes to access offline repos.
- While Omnia may work with RHEL 8.4 and above, all Omnia testing was done with RHEL 8.4 on the control plane. All minor versions of RHEL 8 are supported on the compute nodes.

Rocky

OS Version	Control Plane	Compute Nodes
8.4	Yes	Yes
8.5	Yes	Yes
8.6	Yes	Yes

Note: Always deploy the DVD Edition of the OS on Compute Nodes

1.4.3 Software Installed by Omnia

OSS Title	License Name/Version #	Description
Slurm Workload manager	GNU General Public License	HPC Workload Manager
Kubernetes Controllers	Apache-2.0	HPC Workload Manager
MariaDB	GPL 2.0	Relational database used by Slurm
Docker CE	Apache-2.0	Docker Service
NVidia container runtime	Apache-2.0	Nvidia container runtime library
Python-pip	MIT License	Python Package
kubelet	Apache-2.0	Provides external, versioned ComponentCor
kubeadm	Apache-2.0	“fast paths” for creating Kubernetes clusters
kubectl	Apache-2.0	Command line tool for Kubernetes
jupyterhub	BSD-3Clause New or Revised License	Multi-user hub
kfctl	Apache-2.0	CLI for deploying and managing Kubeflow
kubeflow	Apache-2.0	Cloud Native platform for machine learning
helm	Apache-2.0	Kubernetes Package Manager
tensorflow	Apache-2.0	Machine Learning framework
horovod	Apache-2.0	Distributed deep learning training framewor
MPI	3Clause BSD License	HPC library
spark	Apache-2.0	
coreDNS	Apache-2.0	DNS server that chains plugins
cni	Apache-2.0	Networking for Linux containers
dellemc.openmanage	GNU-General Public License v3.0	OpenManage Ansible Modules simplifies ar
dellemc.os10	GNU-General Public License v3.0	It provides networking hardware abstraction
community.general ansible	GNU-General Public License v3.0	The collection is a part of the Ansible packa
redis	BSD-3-Clause License	In-memory database
cri-o	Apache-2.0	CRI-O is an implementation of the Kuberne
buildah	Apache-2.0	Tool to build and run containers
OpenSM	GNU General Public License 2	
omsdk	Apache-2.0	Dell EMC OpenManage Python SDK (OMS
freeipa	GNU General Public License v3	Authentication system used on the login noc
bind-dyndb-ldap	GNU General Public License v2	LDAP driver for BIND9. It allows you to re
slurm-exporter	GNU General Public License v3	Prometheus collector and exporter for metri
prometheus	Apache-2.0	Open-source monitoring system with a dime
singularity	BSD License	Container platform. It allows you to create a
loki	GNU AFFERO GENERAL PUBLIC LICENSE v3.0	Loki is a log aggregation system designed to
promtail	Apache-2.0	Promtail is an agent which ships the content
Kube prometheus stack	Apache-2.0	Kube Prometheus Stack is a collection of Ku
mailx	MIT License	mailx is a Unix utility program for sending a
xorriso	GPL 3.0	xorriso copies file objects from POSIX com
openshift	Apache-2.0	On-premises platform as a service built arou
grafana	GNU AFFERO GENERAL PUBLIC LICENSE	Grafana is the open source analytics & mon
kubernetes.core	GPL 3.0	Performs CRUD operations on K8s objects
community.grafana	GPL 3.0	Technical Support for open source grafana.
activemq	Apache-2.0	Most popular multi protocol, message broke
golang	BSD-3-Clause License	Go is a statically typed, compiled programn
mysql	GPL 2.0	MySQL is an open-source relational databas
postgresSQL	PostgresSQL License	PostgreSQL, also known as Postgres, is a fre
idrac-telemetry-reference tools	Apache-2.0	Reference toolset for PowerEdge telemetry r
nsfcac/grafana-plugin	MIT License	Machine Learning Framework

OSS Title	License Name/Version #	Description
jansson	MIT License	C library for encoding, decoding and manip
libjwt	Mozilla Public License-2.0 License	JWT C Library
389-ds	GPL	LDAP server used for authentication, access
apparmor	GNU General Public License	Controls access based on paths of the progra
snoopy	GPL 2.0	Snoopy is a small library that logs all progra
timescaledb	Apache-2.0	TimescaleDB is a time-series SQL database
Beegfs-Client	GPLv2	BeeGFS is a high-performance parallel file
redhat subscription	Apache-2.0	Red Hat Subscription Management (RHSM
Lmod	MIT License	Lmod is a Lua based module system that ea
Lua	MIT License	Lua is a lightweight, high-level, multi-parad
ansible posix	GNU General Public License	Ansible Collection targeting POSIX and PO
xCAT	Eclipse Public License 1.0	Provisioning tool that also creates custom d
CUDA Toolkit	NVIDIA License	The NVIDIA® CUDA® Toolkit provides a
MLNX-OFED	BSD License	MLNX_OFED is an NVIDIA tested and pa
ansible pylibssh	LGPL 2.1	Python bindings to client functionality of lib
perl-DBD-Pg	GNU General Public License v3	DBD::Pg - PostgreSQL database driver for t
ansible.utils ansible collection	GPL 3.0	Ansible Collection with utilities to ease the
pandas	BSD-3-Clause License	pandas is a fast, powerful, flexible and easy
python3-netaddr	BSD License	A Python library for representing and manip
psycopg2-binary	GNU Lesser General Public License	Psycopg is the most popular PostgreSQL da
python.requests	Apache-2.0	Makes HTTP requests simpler and more hu

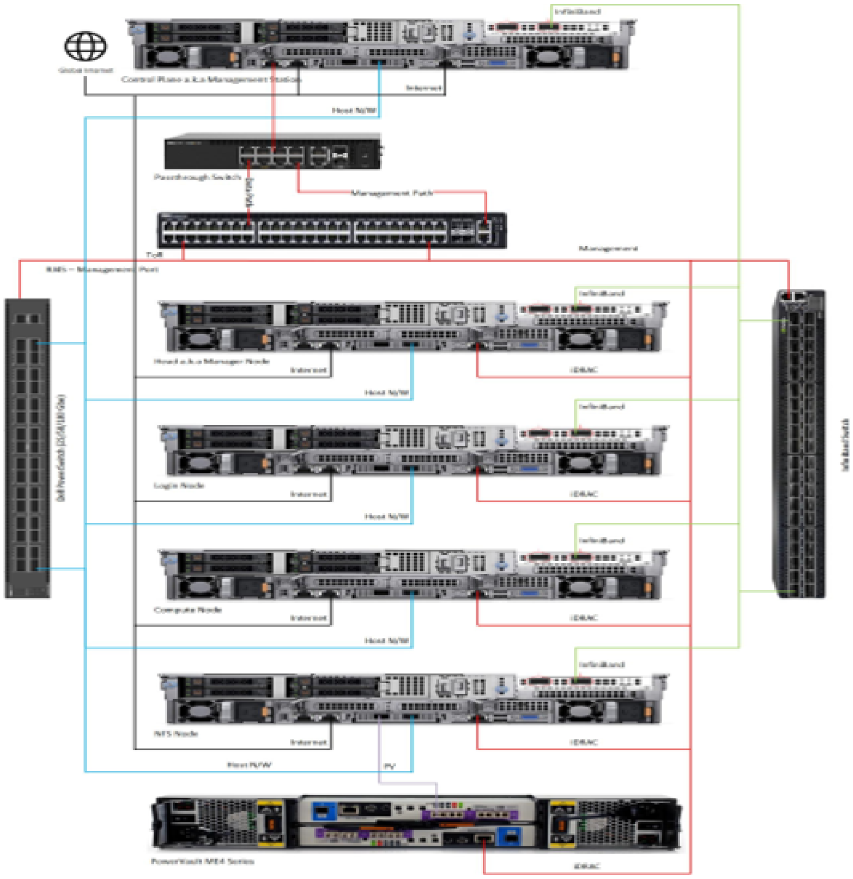
1.5 Network Topologies

1.5.1 Network Topology: Dedicated Setup

Depending on internet access for host nodes, there are two ways to achieve a dedicated NIC setup:

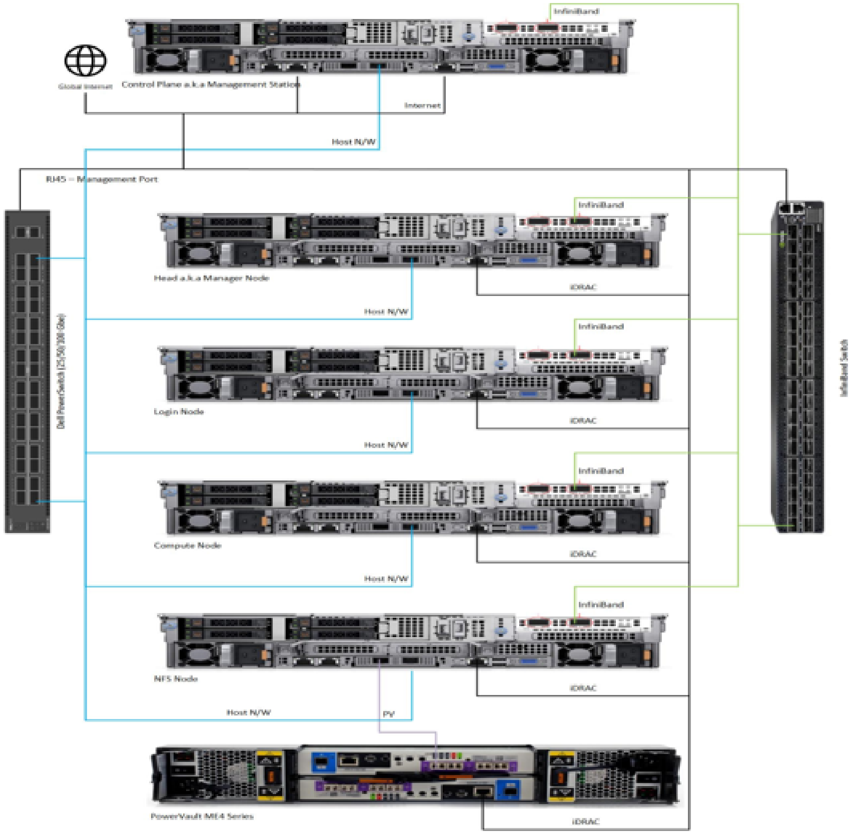
1. Dedicated Setup with dedicated public NIC on compute nodes

When all compute nodes have their own public network access, `primary_dns` and `secondary_dns` in `provision_config.yml` become optional variables as the control plane is not required to be a gateway to the network. The network design would follow the below diagram:



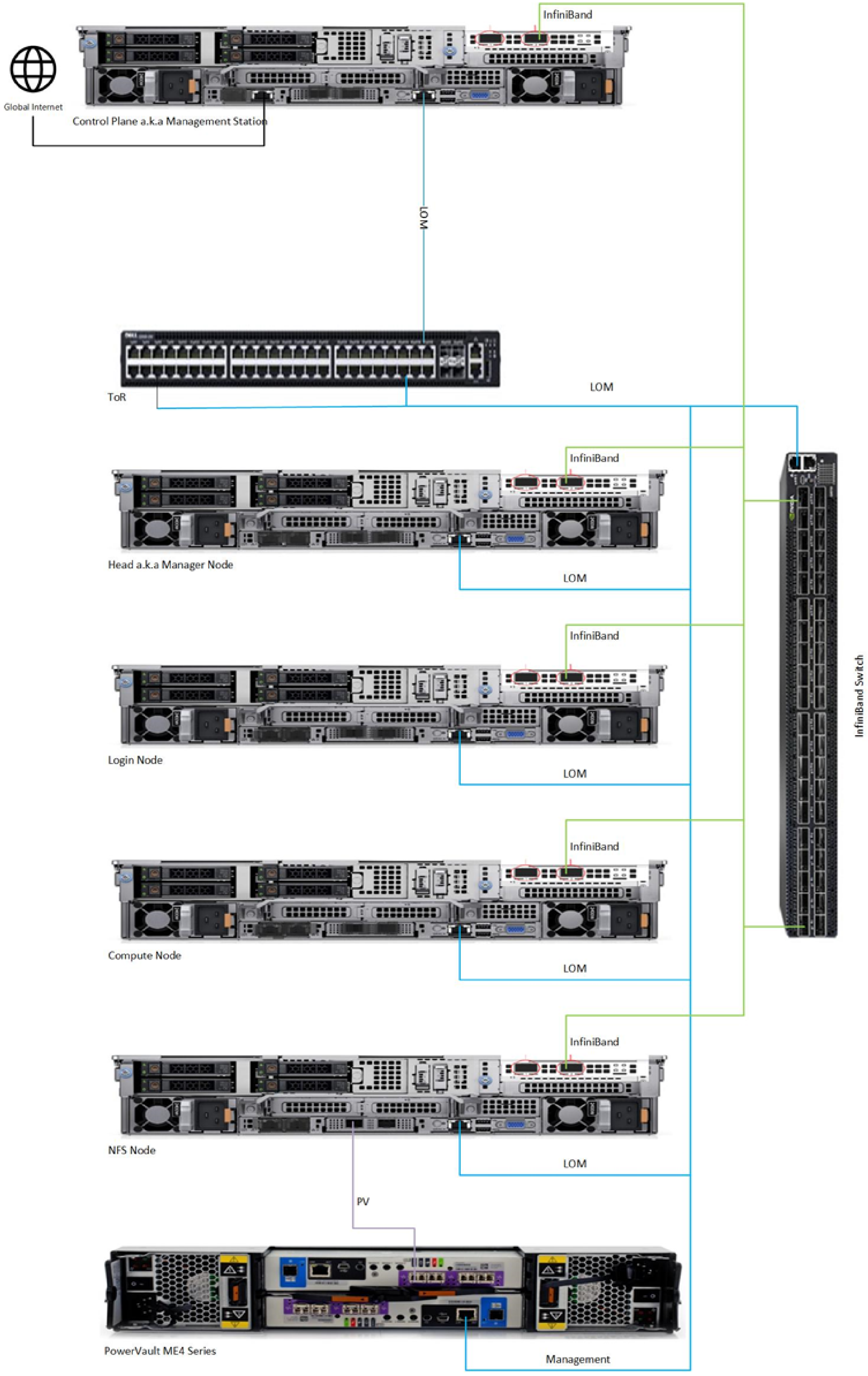
2. Dedicated Setup with single NIC on compute nodes

When all compute nodes rely on the control plane for public network access, the variables `primary_dns` and `secondary_dns` in `provision_config.yml` are used to indicate that the control plane is the gateway for all compute nodes to get internet access. Since all public network traffic will be routed through the control plane, the user may have to take precautions to avoid bottlenecks in such a set-up.



1.5.2 Network Topology: LOM Setup

A LOM port could be shared with the host operating system production traffic. Also, LOM ports can be dedicated to server management. For example, with a four-port LOM adapter, LOM ports one and two could be used for production data while three and four could be used for iDRAC, VNC, RDP, or other operating system-based management data.


InfiniBand
Global Internet
Control Plane a.k.a Management Station
LOM
ToR
LOM
InfiniBand
Head a.k.a Manager Node
LOM
InfiniBand
Login Node
LOM
InfiniBand
Compute Node
LOM
InfiniBand
NFS Node
LOM
PV
PowerVault ME4 Series
Management
InfiniBand Switch

1.6 Blogs about Omnia

- Introduction to Omnia
- Taming the Accelerator Cambrian Explosion with Omnia
- Containerized HPC Workloads Made Easy with Omnia and Singularity
- Solution Overview: Dell Omnia Software
- Solution Brief: Omnia Software

1.7 What Omnia does

Omnia can deploy and configure devices, and build clusters that use Slurm or Kubernetes (or both) for workload management. Omnia will install software from a variety of sources, including:

- Helm repositories
- Source code repositories

CHAPTER

TWO

QUICK INSTALLATION GUIDE

Choose a server outside your intended cluster to function as your control plane.

The control plane needs to be internet-capable with Github and a full OS installed.

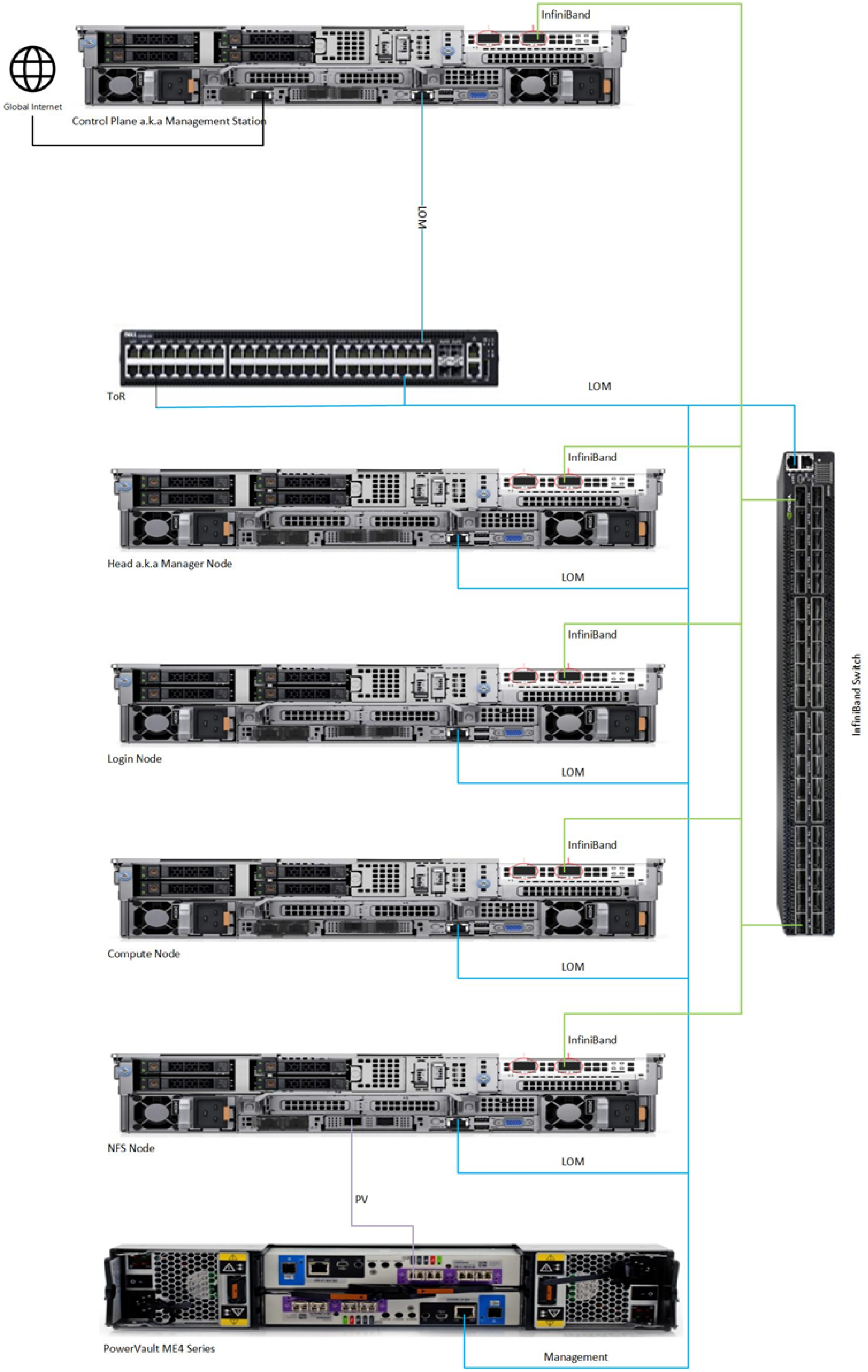
Note: Omnia can be run on control planes running RHEL and Rocky. For a complete list of versions supported, check out the Support Matrix .

```
dnf install git -y
```

If the control plane has an Infiniband NIC installed, run the below

```
yum groupinstall "Infiniband Support" -y
```

Use the image below to set up your network:


InfiniBand
Global Internet
Control Plane a.k.a Management Station
LOM
ToR
LOM
InfiniBand
Head a.k.a Manager Node
LOM
InfiniBand Switch
InfiniBand
Login Node
LOM
InfiniBand
Compute Node
LOM
InfiniBand
NFS Node
LOM
PV
PowerVault ME4 Series
Management

Once the Omnia repository has been cloned on to the control plane:

```
git clone https://github.com/dellhpc/omnia.git
```

Change directory to Omnia using:

```
cd omnia
sh prereq.sh
```

Run the script `prereq.sh` to verify the system is ready for Omnia deployment.

2.1 Running prereq.sh

`prereq.sh` is used to install the software utilized by Omnia on the control plane including Python (3.8), Ansible (2.12.9).

```
cd omnia
sh prereq.sh
```

Note:

- If SELinux is not disabled, it will be disabled by the script and the user will be prompted to reboot the control plane.

2.2 Installing The Provision Tool

2.2.1 Input Parameters for Provision Tool

Fill in all provision-specific parameters in `input/provision_config.yml`

Name	Default, Accepted Values	Required?	Additional Information
public_nic	eno2	required	The NIC/ethernet card that is connected to the public internet.
admin_nic	eno1	required	The NIC/ethernet card that is used for shared LAN over Management (LOM) capability.
admin_nic_subnet	172.29.0.0	required	The intended subnet for shared LOM capability. Note that since the last 16 bits/2 octets of IPv4 are dynamic, please ensure that the parameter value is set to x.x.0.0.
pxe_nic	eno1	required	This NIC used to obtain routing information.
pxe_nic_start_range	172.29.0.100	required	The start of the DHCP range used to assign IPv4 addresses. When the PXE range and BMC subnet are provided, corresponding NICs will be assigned IPs with the same 3rd and 4th octets. Ensure that these ranges contain enough IPs to be double the number of iDRACs present in the cluster.
pxe_nic_end_range	172.29.0.200	required	The end of the DHCP range used to assign IPv4 addresses. When the PXE range and BMC subnet are provided, corresponding NICs will be assigned IPs with the same 3rd and 4th octets. Ensure that these ranges contain enough IPs to be double the number of iDRACs present in the cluster.
ib_nic_subnet		optional	If provided, Omnia will assign static IPs to IB NICs on the compute nodes within the provided subnet. Note that since the last 16 bits/2 octets of IPv4 are dynamic, please ensure that the parameter value is set to x.x.0.0. When the PXE range and BMC subnet are provided, corresponding NICs will be assigned IPs with the same 3rd and 4th octets. IB nics should be prefixed ib.
bmc_nic_subnet		optional	If provided, Omnia will

2.2.2 Before You Run The Provision Tool

- (Recommended) Run `prereq.sh` to get the system ready to deploy Omnia. Alternatively, ensure that Ansible 2.12.9 and Python 3.8 are installed on the system. SELinux should also be disabled.
- To provision the bare metal servers, download one of the following ISOs for deployment:
  1. Rocky 8
  2. RHEL 8.x
- To dictate IP address/MAC mapping, a host mapping file can be provided. If the mapping file is not provided and the variable is left blank, a default mapping file will be created by querying the switch. Use the pxe_mapping_file.csv to create your own mapping file.
- Ensure that all connection names under the network manager match their corresponding device names.

```
nmcli connection
```

In the event of a mismatch, edit the file `/etc/sysconfig/network-scripts/ifcfg-<nic name>` using vi editor.

- All target hosts should be set up in PXE mode before running the playbook.
- If RHEL is in use on the control plane, enable RedHat subscription. Not only does Omnia not enable RedHat subscription on the control plane, package installation may fail if RedHat subscription is disabled.
- Users should also ensure that all repos are available on the RHEL control plane.
- Ensure that the `pxe_nic` and `public_nic` are in the firewalld zone: public.

Note:

- After configuration and installation of the cluster, changing the control plane is not supported. If you need to change the control plane, you must redeploy the entire cluster.
- If there are errors while executing any of the Ansible playbook commands, then re-run the playbook.

2.2.3 Running The Provision Tool

1. Edit the `input/provision_config.yml` file to update the required variables.

Warning: The IP address *192.168.25.x* is used for PowerVault Storage communications. Therefore, do not use this IP address for other configurations.

2. To deploy the Omnia provision tool, run the following command

```
cd provision
ansible-playbook provision.yml
```

3. By running `provision.yml`, the following configurations take place:

   i. All compute nodes in cluster will be enabled for PXE boot with osimage mentioned in `provision_config.yml`.

   ii. A PostgreSQL database is set up with all relevant cluster information such as MAC IDs, hostname, admin IP, infiniband IPs, BMC IPs etc.

      To access the DB, run:

```
psql -U postgres

\c omniadb
```

To view the schema being used in the cluster: `\dn`

To view the tables in the database: `\dt`

To view the contents of the `nodeinfo` table: `select * from cluster.nodeinfo`

```
id | servicetag |     admin_mac     |         hostname         |   admin_
↪ip    | bmc_ip | ib_ip

----+------------+-------------------+--------------------------+--------
↪------+-------+-------


1 |            | 00:c0:ff:43:f9:44 | node00001.winter.cluster | 172.29.1.
↪253 |        |
2 |            | 70:b5:e8:d1:84:22 | node00002.winter.cluster | 172.29.1.
↪254 |        |
3 |            | b8:ca:3a:71:25:5c | node00003.winter.cluster | 172.29.1.
↪255 |        |
4 |            | 8c:47:be:c7:6f:c1 | node00004.winter.cluster | 172.29.2.
↪0   |        |
5 |            | 8c:47:be:c7:6f:c2 | node00005.winter.cluster | 172.29.2.
↪1   |        |
6 |            | b0:26:28:5b:80:18 | node00006.winter.cluster | 172.29.2.
↪2   |        |
7 |            | b0:7b:25:de:71:de | node00007.winter.cluster | 172.29.2.
↪3   |        |
8 |            | b0:7b:25:ee:32:fc | node00008.winter.cluster | 172.29.2.
↪4   |        |
9 |            | d0:8e:79:ba:6a:58 | node00009.winter.cluster | 172.29.2.
↪5   |        |
10|            | d0:8e:79:ba:6a:5e | node00010.winter.cluster | 172.29.2.
↪6   |        |
```

iii. Offline repositories will be created based on the OS being deployed across the cluster.

Once the playbook execution is complete, ensure that PXE boot and RAID configurations are set up on remote nodes. Users are then expected to reboot target servers to provision the OS.

Note:

- If the cluster does not have access to the internet, AppStream will not function. To provide internet access through the control plane (via the PXE network NIC), update `primary_dns` and `secondary_dns` in `provision_config.yml` and run `provision.yml`
- All ports required for xCAT to run will be opened (For a complete list, check out the Security Configuration Document).
- After running `provision.yml`, the file `input/provision_config.yml` will be encrypted. To edit file, use the command: `ansible-vault edit provision_config.yml --vault-password-file .provision_vault_key`
- To re-provision target servers `provision.yml` can be re-run. Alternatively, use the following steps:

- Use `lsdef -t osimage | grep install-compute` to get a list of all valid OS profiles.
- Use `nodeset all osimage=<selected OS image from previous command>` to provision the OS on the target server.
- PXE boot the target server to bring up the OS.

Warning: Once xCAT is installed, restart your SSH session to the control plane to ensure that the newly set up environment variables come into effect.

Adding a new node

A new node can be added using one of two ways:

1. Using a mapping file:
   - Update the existing mapping file by appending the new entry (without the disrupting the older entries) or provide a new mapping file by pointing `pxe_mapping_file_path` in `provision_config.yml` to the new location.
   - Run `provision.yml`.
2. Using the switch IP:
   - Run `provision.yml` once the switch has discovered the potential new node.

2.2.4 After Running the Provision Tool

Once the **servers are provisioned**, run the post provision script to:

- Configure iDRAC IP or BMC IP if `bmc_nic_subnet` is provided in `input/provision_config.yml`.
- Configure Infiniband static IPs on remote nodes if `ib_nic_subnet` is provided in `input/provision_config.yml`.
- Set hostname for the remote nodes.
- Invoke `network.yml` and `accelerator.yml` to install OFED, CUDA toolkit and ROCm drivers.
- Create `node_inventory` in `/opt/omnia` listing provisioned nodes.

```
cat /opt/omnia/node_inventory
172.29.0.100 service_tag=XXXXXXX operating_system=RedHat
172.29.0.101 service_tag=XXXXXXX operating_system=RedHat
172.29.0.102 service_tag=XXXXXXX operating_system=Rocky
172.29.0.103 service_tag=XXXXXXX operating_system=Rocky
```

Note: Before post provision script, verify redhat subscription is enabled using the `rhsm_subscription.yml` playbook in utils only if OFED or GPU accelerators are to be installed.

To run the script, use the below command::

```
ansible-playbook post_provision.yml
```

2.3 Building Clusters

2.3.1 Input Parameters for the Cluster

These parameters is located in `input/omnia_config.yml`

Parameter Name	Default Value	Additional Information
mariadb_password	password	Password used to access the Slurm database.Required Length: 8 characters. The password must not contain -,, ‘,”
k8s_version	1.19.3	Kubernetes VersionAccepted Values: “1.16.7” or “1.19.3”
k8s_cni	calico	CNI type used by Kubernetes.Accepted values: calico, flannel
k8s_pod_network_cidr	10.244.0.0/16	Kubernetes pod network CIDR
docker_username		Username to login to Docker. A kubernetes secret will be created and patched to the service account in default namespace.This value is optional but suggested to avoid docker pull limit issues
docker_password		Password to login to DockerThis value is mandatory if a docker_username is provided
ansible_config_file_path	/etc/ansible	Path where the ansible.cfg file can be found.If dnf is used, the default value is valid. If pip is used, the variable must be set manually
login_node_required	true	Boolean indicating whether the login node is required or not
ldap_required	false	Boolean indicating whether ldap client is required or not
ldap_server_ip		LDAP server IP. Required if `ldap_required` is true.
ldap_connection_type	TLS	For a TLS connection, provide a valid certification path. For an SSL connection, ensure port 636 is open.
ldap_ca_cert_path	/etc/openldap/certs/omnialdap.pem	This variable accepts Server Certificate Path. Make sure certificate is present in the path provided. The certificate should have .pem or .crt extension. This variable is mandatory if connection type is TLS.
user_home_dir	/home	This variable accepts the user home directory path for ldap configuration. If nfs mount is created for user home, make sure you provide the LDAP users mount home directory path.
ldap_bind_username	admin	If LDAP server is configured with bind dn then bind dn user to be provided. If this value is not provided (when bind is configured in server) then ldap authentication fails.
ldap_bind_password		If LDAP server is configured with bind dn then bind dn password to be provided. If this value is not provided (when bind is configured in server) then ldap authentication fails.
domain_name	omnia.test	Sets the intended domain name
realm_name	OMNIA.TEST	Sets the intended realm name
directory_manager_password		Password authenticating admin level access to the Directory for system management tasks. It will be added to the instance of directory server created for IPA.Required Length: 8 characters. The password must not contain -,, ‘,”
kerberos_admin_password		“admin” user password for the IPA server on RockyOS.
enable_secure_login_node	false	Boolean value deciding whether security features are enabled on the Login Node.
powervault_ip		IP of the powervault connected to the NFS server. Mandatory field when nfs_node group is defined with an IP and omnia is required to configure nfs server.

Note: When `ldap_required` is true, `login_node_required` and `freeipa_required` have to be false.

2.3.2 Before You Build Clusters

- Verify that all inventory files are updated.
- If the target cluster requires more than 10 kubernetes nodes, use a docker enterprise account to avoid docker pull limits.
- Verify that all nodes are assigned a group. Use the inventory as a reference.
  - The manager group should have exactly 1 manager node.
  - The compute group should have at least 1 node.
  - The login_node group is optional. If present, it should have exactly 1 node.
  - Users should also ensure that all repos are available on the target nodes running RHEL.

Note: The inventory file accepts both IPs and FQDNs as long as they can be resolved by DNS.

- For RedHat clusters, ensure that RedHat subscription is enabled on all target nodes.

Features enabled by omnia.yml

- Slurm: Once all the required parameters in omnia_config.yml are filled in, `omnia.yml` can be used to set up slurm.
- LDAP client support: The manager and compute nodes will have LDAP installed but the login node will be excluded.
- FreeIPA support
- Login Node (Additionally secure login node)
- Kubernetes: Once all the required parameters in omnia_config.yml are filled in, `omnia.yml` can be used to set up kubernetes.
- BeeGFS bolt on installation
- NFS bolt on support

2.3.3 Building Clusters

1. In the `input/omnia_config.yml` file, provide the required details.

Note: Without the login node, Slurm jobs can be scheduled only through the manager node.

2. Create an inventory file in the *omnia* folder. Add login node IP address under the manager node IP address under the *[manager]* group, compute node IP addresses under the *[compute]* group, and Login node IP under the *[login_node]* group,. Check out the sample inventory for more information.

Note:

- Omnia checks for red hat subscription being enabled on RedHat nodes as a pre-requisite. Not having Red Hat subscription enabled on the manager node will cause `omnia.yml` to fail. If compute nodes do not have Red Hat subscription enabled, `omnia.yml` will skip the node entirely.
- Omnia creates a log file which is available at: `/var/log/omnia.log`.
- If only Slurm is being installed on the cluster, docker credentials are not required.

3. To run `omnia.yml`:

   ansible-playbook omnia.yml -i inventory

Note: To visualize the cluster (Slurm/Kubernetes) metrics on Grafana (On the control plane) during the run of `omnia.yml`, add the parameters `grafana_username` and `grafana_password` (That is `ansible-playbook omnia.yml -i inventory -e grafana_username="" -e grafana_password=""`). Alternatively, Grafana is not installed by `omnia.yml` if it's not available on the Control Plane.

Using Skip Tags

Using skip tags, the scheduler running on the cluster can be set to Slurm or Kubernetes while running the `omnia.yml` playbook. This choice can be made depending on the expected HPC/AI workloads.

- Kubernetes: `ansible-playbook omnia.yml -i inventory --skip-tags "kubernetes"` (To set Slurm as the scheduler)
- Slurm: `ansible-playbook omnia.yml -i inventory --skip-tags "slurm"` (To set Kubernetes as the scheduler)

Note:

- If you want to view or edit the `omnia_config.yml` file, run the following command:
  - `ansible-vault view omnia_config.yml --vault-password-file .omnia_vault_key` – To view the file.
  - `ansible-vault edit omnia_config.yml --vault-password-file .omnia_vault_key` – To edit the file.
- It is suggested that you use the ansible-vault view or edit commands and that you do not use the ansible-vault decrypt or encrypt commands. If you have used the ansible-vault decrypt or encrypt commands, provide 644 permission to `omnia_config.yml`.

Kubernetes Roles

As part of setting up Kubernetes roles, `omnia.yml` handles the following tasks on the manager and compute nodes:

- Docker is installed.
- Kubernetes is installed.
- Helm package manager is installed.
- All required services are started (Such as kubelet).
- Different operators are configured via Helm.
- Prometheus is installed.

Slurm Roles

As part of setting up Slurm roles, `omnia.yml` handles the following tasks on the manager and compute nodes:

- Slurm is installed.
- All required services are started (Such as slurmd, slurmctld, slurmdbd).
- Prometheus is installed to visualize slurm metrics.
- Lua and Lmod are installed as slurm modules.
- Slurm restd is set up.

Login node

If a login node is available and mentioned in the inventory file, the following tasks are executed:

- Slurmd is installed.
- All required configurations are made to `slurm.conf` file to enable a slurm login node.
- FreeIPA (the default authentication system on the login node) is installed to provide centralized authentication

Hostname requirements

- In the `examples` folder, a **mapping_host_file.csv** template is provided which can be used for DHCP configuration. The header in the template file must not be deleted before saving the file. It is recommended to provide this optional file as it allows IP assignments provided by Omnia to be persistent across control plane reboots.
- The Hostname should not contain the following characters: , (comma), . (period) or _ (underscore). However, the **domain name** is allowed commas and periods.
- The Hostname cannot start or end with a hyphen (-).
- No upper case characters are allowed in the hostname.
- The hostname cannot start with a number.
- The hostname and the domain name (that is: `hostname00000x.domain.xxx`) cumulatively cannot exceed 64 characters. For example, if the `node_name` provided in `input/provision_config.yml` is 'node', and the `domain_name` provided is 'omnia.test', Omnia will set the hostname of a target compute node to 'node00001.omnia.test'. Omnia appends 6 digits to the hostname to individually name each target node.

Note:

- To enable the login node, ensure that `login_node_required` in `input/omnia_config.yml` is set to true.
- To enable security features on the login node, ensure that `enable_secure_login_node` in `input/omnia_config.yml` is set to true.
- To customize the security features on the login node, fill out the parameters in `input/omnia_security_config.yml`.

Warning: No users/groups will be created by Omnia.

Slurm job based user access

To ensure security while running jobs on the cluster, users can be assigned permissions to access compute nodes only while their jobs are running. To enable the feature:

```
cd scheduler
ansible-playbook job_based_user_access.yml -i inventory
```

Note:

- The inventory queried in the above command is to be created by the user prior to running `omnia.yml` as `scheduler.yml` is invoked by `omnia.yml`
- Only users added to the 'slurm' group can execute slurm jobs. To add users to the group, use the command: `usermod -a -G slurm <username>`.

Installing LDAP Client

Manager and compute nodes will have LDAP client installed and configured if `ldap_required` is set to true. The login node does not have LDAP client installed.

Warning: No users/groups will be created by Omnia.

2.3.4 FreeIPA installation on the NFS node

IPA services are used to provide account management and centralized authentication. To set up IPA services for the NFS node in the target cluster, run the following command from the `utils/cluster` folder on the control plane:

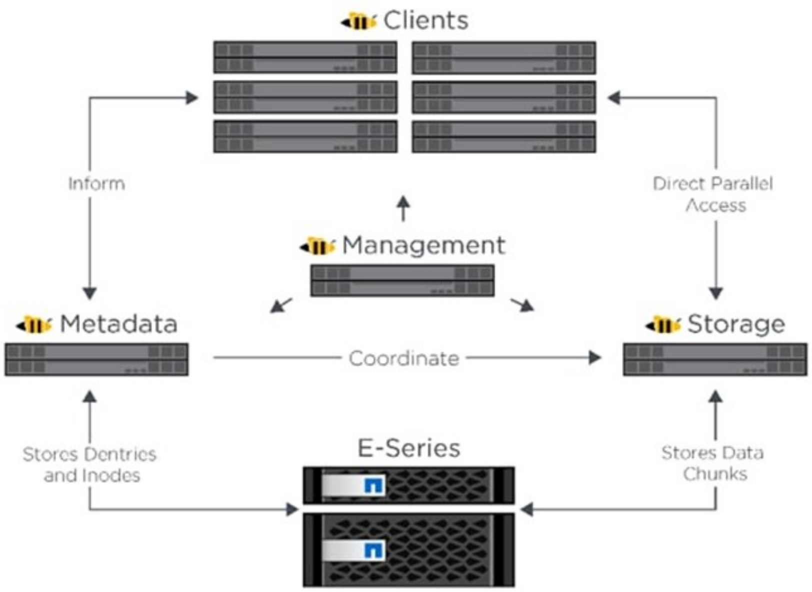
```
cd utils/cluster
ansible-playbook install_ipa_client.yml -i inventory -e kerberos_admin_password="" -e ipa_server_hostname="" -e domain_name="" -e ipa_server_ipadress=""
```

Input Parameter	Definition	Variable value
kerberos_admin_password	"admin" user password for the IPA server on RockyOS and RedHat.	The password can be found in the file `input/omnia_config.yml` .
ipa_server_hostname	The hostname of the IPA server	The hostname can be found on the manager node.
domain_name	Domain name	The domain name can be found in the file `input/omnia_config.yml`.
ipa_server_ipadress	The IP address of the IPA server	The IP address can be found on the IPA server on the manager node using the `ip a` command. This IP address should be accessible from the NFS node.

Use the format specified under NFS inventory in the Sample Files for inventory.

2.3.5 BeeGFS Bolt On

BeeGFS is a hardware-independent POSIX parallel file system (a.k.a. Software-defined Parallel Storage) developed with a strong focus on performance and designed for ease of use, simple installation, and management.



Pre Requisites before installing BeeGFS client

- If the user intends to use BeeGFS, ensure that a BeeGFS cluster has been set up with beegfs-mgmtd, beegfs-meta, beegfs-storage services running.

  Ensure that the following ports are open for TCP and UDP connectivity:

Port	Service
8008	Management service (beegfs-mgmtd)
8003	Storage service (beegfs-storage)
8004	Client service (beegfs-client)
8005	Metadata service (beegfs-meta)
8006	Helper service (beegfs-helperd)

To open the ports required, use the following steps:

1. `firewall-cmd --permanent --zone=public --add-port=<port number>/tcp`
2. `firewall-cmd --permanent --zone=public --add-port=<port number>/udp`
3. `firewall-cmd --reload`
4. `systemctl status firewalld`

- Ensure that the nodes in the inventory have been assigned **only** these roles: manager and compute.

Note:

- If the BeeGFS server (MGMTD, Meta, or storage) is running BeeGFS version 7.3.1 or higher, the security feature on the server should be disabled. Change the value of `connDisableAuthentication` to `true` in /etc/beegfs/beegfs-mgmtd.conf, /etc/beegfs/beegfs-meta.conf and /etc/beegfs/beegfs-storage.conf. Restart the services to complete the task:

```
systemctl restart beegfs-mgmtd
systemctl restart beegfs-meta
systemctl restart beegfs-storage
systemctl status beegfs-mgmtd
```

(continues on next page)

(continued from previous page)

```
systemctl status beegfs-meta
systemctl status beegfs-storage
```

Note: BeeGFS with OFED capability is only supported on RHEL 8.3 and above due to limitations on BeeGFS. When setting up your cluster with RDMA support, check the BeeGFS documentation to provide appropriate values in `input/storage_config.yml`.

- If the cluster runs Rocky, ensure that versions running are compatible:

Rocky OS version	BeeGFS version
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.3.2
Rocky Linux 8.5: no OFED, OFED 5.5	7.3.2
Rocky Linux 8.6: no OFED, OFED 5.6	7.3.2
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.3.1
Rocky Linux 8.5: no OFED, OFED 5.5	7.3.1
Rocky Linux 8.6: no OFED, OFED 5.6	7.3.1
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.3.0
Rocky Linux 8.5: no OFED, OFED 5.5	7.3.0
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.2.8
Rocky Linux 8.5: no OFED, OFED 5.5	7.2.8
Rocky Linux 8.6: no OFED, OFED 5.6	7.2.8
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.2.7
Rocky Linux 8.5: no OFED, OFED 5.5	7.2.7
Rocky Linux 8.6: no OFED, OFED 5.6	7.2.7
Rocky Linux 8.5: no OFED, OFED 5.5	7.2.6
Rocky Linux 8.6: no OFED, OFED 5.6	7.2.6
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.2.5
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.2.4

Installing the BeeGFS client via Omnia

After the required parameters are filled in `input/storage_config.yml`, Omnia installs BeeGFS on manager and compute nodes while executing the `omnia.yml` playbook.

```
.. note::
```

- BeeGFS client-server communication can take place through TCP or RDMA. If RDMA support is required, set `beegfs_rdma_support` should be set to true. Also, OFED should be installed on all target nodes.
- For BeeGFS communication happening over RDMA, the `beegfs_mgmt_server` should be provided with the Infiniband IP of the management server.

2.3.6 NFS Bolt On

- Ensure that an external NFS server is running. NFS clients are mounted using the external NFS server's IP.
- Fill out the `nfs_client_params` variable in the `input/storage_config.yml` file in JSON format using the samples provided below.
- This role runs on manager, compute and login nodes.
- Make sure that `/etc/exports` on the NFS server is populated with the same paths listed as `server_share_path` in the `nfs_client_params` in `input/storage_config.yml`.
- Post configuration, enable the following services (using this command: `firewall-cmd --permanent --add-service=<service name>`) and then reload the firewall (using this command: `firewall-cmd --reload`).
  - nfs
  - rpc-bind
  - mountd
- Omnia supports all NFS mount options. Without user input, the default mount options are nosuid,rw,sync,hard,intr. For a list of mount options, click here.
- The fields listed in `nfs_client_params` are:
  - server_ip: IP of NFS server
  - server_share_path: Folder on which NFS server mounted
  - client_share_path: Target directory for the NFS mount on the client. If left empty, respective `server_share_path value` will be taken for `client_share_path`.
  - client_mount_options: The mount options when mounting the NFS export on the client. Default value: nosuid,rw,sync,hard,intr.
- There are 3 ways to configure the feature:
  1. **Single NFS node** : A single NFS filesystem is mounted from a single NFS server. The value of `nfs_client_params` would be:

     ```
     - { server_ip: 172.10.0.101, server_share_path: "/mnt/share", client_share_
     ↪path: "/mnt/client", client_mount_options: "nosuid,rw,sync,hard,intr" }
     ```

  2. **Multiple Mount NFS Filesystem**: Multiple filesystems are mounted from a single NFS server. The value of `nfs_client_params` would be:

     ```
     - { server_ip: 172.10.0.101, server_share_path: "/mnt/share1", client_share_
     ↪path: "/mnt/client1", client_mount_options: "nosuid,rw,sync,hard,intr" }
     - { server_ip: 172.10.0.101, server_share_path: "/mnt/share2", client_share_
     ↪path: "/mnt/client2", client_mount_options: "nosuid,rw,sync,hard,intr" }
     ```

  3. **Multiple NFS Filesystems**: Multiple filesystems are mounted from multiple NFS servers. The value of `nfs_client_params` would be:

     ```
     - { server_ip: 172.10.0.101, server_share_path: "/mnt/server1", client_share_
     ↪path: "/mnt/client1", client_mount_options: "nosuid,rw,sync,hard,intr" }
     - { server_ip: 172.10.0.102, server_share_path: "/mnt/server2", client_share_
     ↪path: "/mnt/client2", client_mount_options: "nosuid,rw,sync,hard,intr" }
     - { server_ip: 172.10.0.103, server_share_path: "/mnt/server3", client_share_
     ↪path: "/mnt/client3", client_mount_options: "nosuid,rw,sync,hard,intr" }
     ```

Warning: After an NFS client is configured, if the NFS server is rebooted, the client may not be able to reach the server. In those cases, restart the NFS services on the server using the below command:

```
systemctl disable nfs-server
systemctl enable nfs-server
systemctl restart nfs-server
```

2.4 Configuring Switches

2.4.1 Configuring Infiniband Switches

Depending on the number of ports available on your Infiniband switch, they can be classified into:

- EDR Switches (36 ports)
- HDR Switches (40 ports)

Input the configuration variables into the `network/infiniband_edr_input.yml` or `network/infiniband_hdr_input.yml` as appropriate:

Name	Default, Accepted values	Required?	Purpose
enable_split_port	false, true	required	Indicates whether ports are to be split
ib_split_ports		optional	Stores the split configuration of the ports. Accepted formats are comma-separated (EX: “1,2”), ranges (EX: “1-10”), comma-separated ranges (EX: “1,2,3-8,9,10-12”)
snmp_trap_destination		optional	The IP address of the SNMP Server where the event trap will be sent. If this variable is left blank, SNMP will be disabled.
snmp_community_name	public		The “SNMP community string” is like a user ID or password that allows access to a router’s or other device’s statistics.
cache_directory			Cache location used by OpenSM
log_directory			The directory where temporary files of opensm are stored. Can be set to the default directory or enter a directory path to store temporary files.
mellanox_switch_config		optional	**List of configuration lines to apply to the switch.** **# Example:** **# mellanox_switch_config:** # - Command 1 # - Command 2 By default, the list is empty.
ib 1/(1-xx) config	“no shutdown”		Indicates the required state of ports 1-xx (depending on the value of 1/x)
save_changes_to_startup	false, true		Indicates whether the switch configuration is to persist across reboots

Before you run the playbook

Before running `network/infiniband_switch_config.yml`, ensure that SSL Secure Cookies are disabled. Also, HTTP and JSON Gateway need to be enabled on your switch. This can be verified by running:

```
show web (To check if SSL Secure Cookies is disabled and HTTP is enabled)
show json-gw (To check if JSON Gateway is enabled)
```

In case any of these services are not in the state required, run:

```
no web https ssl secure-cookie enable (To disable SSL Secure Cookies)
web http enable (To enable the HTTP gateway)
json-gw enable (To enable the JSON gateway)
```

When connecting to a new or factory reset switch, the configuration wizard requests to execute an initial configuration:

(Recommended) If the user enters 'no', they still have to provide the admin and monitor passwords.

If the user enters 'yes', they will also be prompted to enter the hostname for the switch, DHCP details, IPv6 details, etc.

Note:

- When initializing a factory reset switch, the user needs to ensure DHCP is enabled and an IPv6 address is not assigned.
- All ports intended for splitting need to be connected to the network before running the playbook.

Running the playbook

If `enable_split_port` is **true**, run:

```
cd network
 ansible-playbook infiniband_switch_config.yml -i inventory -e ib_username="" -e ib_
↪password="" -e ib_admin_password="" -e ib_monitor_password=""  -e ib_default_password="
↪" -e ib_switch_type=""
```

If `enable_split_port` is **false**, run:

```
cd network
ansible-playbook infiniband_switch_config.yml -i inventory -e ib_username="" -e ib_
↪password=""  -e ib_switch_type=""
```

- Where `ib_username` is the username used to authenticate into the switch.
- Where `ib_password` is the password used to authenticate into the switch.
- Where `ib_admin_password` is the intended password to authenticate into the switch after `infiniband_switch_config.yml` has run.
- Where `ib_monitor_password` is the mandatory password required while running the initial configuration wizard on the Inifiniband switch.
- Where `ib_default_password` is the password used to authenticate into factory reset/fresh-install switches.
- Where `ib_switch_type` refers to the model of the switch: HDR/EDR

Note:

- `ib_admin_password` and `ib_monitor_password` have the following constraints:
  - Passwords should contain 8-64 characters.
  - Passwords should be different than username.

  - Passwords should be different than 5 previous passwords.
  - Passwords should contain at least one of each: Lowercase, uppercase and digits.
- The inventory file should be a list of IPs separated by newlines. Check out the `switch_inventory` section in Sample Files

2.4.2 Configuring Ethernet Switches (S3 and S4 series)

- Edit the `network/ethernet_tor_input.yml` file for all S3* and S4* PowerSwitches such as S3048-ON, S4048T-ON, S4112F-ON, S4048-ON, S4048T-ON, S4112F-ON, S4112T-ON, and S4128F-ON.

Name	Default, accepted values	Required?	Purpose
os10_config	“interface vlan1” “exit”	required	Global configurations for the switch.
snmp_trap_destination		optional	The trap destination IP address is the IP address of the SNMP Server where the trap will be sent. Ensure that the SNMP IP is valid.
snmp_community_string	public	optional	An SNMP community string is a means of accessing statistics stored within a router or other device.
ethernet 1/1/(1-52) config	By default: Port description is provided. Each interface is set to “up” state. The fanout/breakout mode for 1/1/1 to 1/1/52 is as per the value set in the breakout_value variable.	required	By default, all ports are brought up in admin UP state
			Update the individual interfaces of the Dell PowerSwitch SS3048-ON. The interfaces are from ethernet 1/1/1 to ethernet 1/1/52. By default, the breakout mode is set for 1/1/1 to 1/1/52. Note: The playbooks will fail if any invalid configurations are entered.
save_changes_to_startup	false	required	Change it to “true” only when you are certain that the updated configurations and commands are valid. WARNING: When set to “true”, the startup configuration file is updated. If incorrect configurations or commands are entered, the Ethernet switches may not operate as expected.

- When initializing a factory reset switch, the user needs to ensure DHCP is enabled and an IPv6 address is not assigned.

Running the playbook:

```
cd network

ansible-playbook ethernet_switch_config.yml -i inventory -e ethernet_switch_username=”” -e ethernet_switch_password=””
```

- Where `ethernet_switch_username` is the username used to authenticate into the switch.
- The inventory file should be a list of IPs separated by newlines. Check out the switch_inventory section in Sample Files
- Where `ethernet_switch_password` is the password used to authenticate into the switch.

2.4.3 Configuring Ethernet Switches (S5 series)

- Edit the `network/ethernet_sseries_input.yml` file for all S5* PowerSwitches such as S5232F-ON.

Name	Default, accepted values	Required?	Purpose
os10_config	- "interface vlan1" - "exit"	required	Global configurations for the switch.
breakout_value	**10g-4x**, 25g-4x, 40g-1x, 50g-2x, 100g-1x	required	By default, all ports are configured in the 10g-4x breakout mode in which a QSFP28 or QSFP+ port is split into four 10G interfaces. For more information about the breakout modes, see Configure breakout mode.
snmp_trap_destination		optional	The trap destination IP address is the IP address of the SNMP Server where the trap will be sent. Ensure that the SNMP IP is valid.
snmp_community_string	public	optional	An SNMP community string is a means of accessing statistics stored within a router or other device.
ethernet 1/1/(1-34) config	**By default:** Port description is provided. Each interface is set to "up" state. The fanout/breakout mode for 1/1/1 to 1/1/34 is as per the value set in the breakout_value variable.	required	By default, all ports are brought up in admin UP state **Update the individual interfaces of the Dell PowerSwitch S5232F-ON.** The interfaces are from ethernet 1/1/1 to ethernet 1/1/34. By default, the breakout mode is set for 1/1/1 to 1/1/34. Note: The playbooks will fail if any invalid configurations are entered.
save_changes_to_startup	false	required	**Change it to "true" only when you are certain that the updated configurations and commands are valid.** WARNING: When set to "true", the startup configuration file is updated. If incorrect configurations or

- When initializing a factory reset switch, the user needs to ensure DHCP is enabled and an IPv6 address is not assigned.

Note: The `breakout_value` of a port can only be changed after un-splitting the port.

Running the playbook:

```
cd network

ansible-playbook ethernet_switch_config.yml -i inventory -e ethernet_switch_username=”” -
↪e ethernet_switch_password=””
```

- Where `ethernet_switch_username` is the username used to authenticate into the switch.
- The inventory file should be a list of IPs separated by newlines. Check out the switch_inventory section in Sample Files
- Where `ethernet_switch_password` is the password used to authenticate into the switch.

2.4.4 Configuring Ethernet Switches (Z series)

- Edit the `network/ethernet_zseries_input.yml` file for all Z series PowerSwitches such as Z9332F-ON, Z9262-ON and Z9264F-ON. The default configuration is written for Z9264F-ON.

Name	Default, accepted values	Required?	Purpose
os10_config	- "interface vlan1" - "exit"	required	Global configurations for the switch.
breakout_value	**10g-4x**, 25g-4x, 40g-1x, 100g-1x	required	By default, all ports are configured in the 10g-4x breakout mode in which a QSFP28 or QSFP+ port is split into four 10G interfaces. For more information about the breakout modes, see Configure breakout mode.
snmp_trap_destination		optional	The trap destination IP address is the IP address of the SNMP Server where the trap will be sent. Ensure that the SNMP IP is valid.
snmp_community_string	public	optional	An SNMP community string is a means of accessing statistics stored within a router or other device.
ethernet 1/1/(1-63) config	**By default:** Port description is provided. Each interface is set to "up" state. The fanout/breakout mode for 1/1/1 to 1/1/61 is as per the value set in the breakout_value variable.	required	By default, all ports are brought up in admin UP state **Update the individual interfaces of the Dell PowerSwitch S5232F-ON.** The interfaces are from ethernet 1/1/1 to ethernet 1/1/63. By default, the breakout mode is set for 1/1/1 to 1/1/63. Note: The playbooks will fail if any invalid configurations are entered.
save_changes_to_startup	false	required	**Change it to "true" only when you are certain that the updated configurations and commands are valid.** WARNING: When set to "true", the startup configuration file is updated. If incorrect configurations or

- When initializing a factory reset switch, the user needs to ensure DHCP is enabled and an IPv6 address is not assigned.
- The 65th port on a Z series switch cannot be split.
  - Only odd ports support breakouts on Z9264F-ON. For more information, click here.

Note: The `breakout_value` of a port can only be changed after un-splitting the port.

Running the playbook:

```
cd network

ansible-playbook ethernet_switch_config.yml -i inventory -e ethernet_switch_username=”” -e ethernet_switch_password=””
```

- Where `ethernet_switch_username` is the username used to authenticate into the switch.
- The inventory file should be a list of IPs separated by newlines. Check out the switch_inventory section in Sample Files
- Where `ethernet_switch_password` is the password used to authenticate into the switch.

2.5 Configuring Storage

Configuring Powervault Storage

To configure powervault ME4 and ME5 storage arrays, follow the below steps:

Fill out all required parameters in `storage/powervault_input.yml`:

Parameter	Default, Accepted values	Required?	Additional information
powervault_protocol	**sas**	Required	This variable indicates the network protocol used for data connectivity
powervault_controller_mode	**multi**, single	Required	This variable indicates the number of controllers available on the target powervault.
powervault_locale	**English**	Optional	Represents the selected language. Currently, only English is supported.
powervault_system_name	Uninitialized_Name	Optional	The system name used to identify the PowerVault Storage device. The name should be less than 30 characters and must not contain spaces.
powervault_snmp_notify_level	none	Required	Select the SNMP notification levels for PowerVault Storage devices.
powervault_pool_type	**linear**, virtual	Required	This variable indicates the kind of pool created on the target powervault.
powervault_raid_levels	**raid1**, raid5, raid6, raid10	Optional	Enter the required RAID levels and the minimum and maximum number of disks for each RAID levels.
powervault_disk_range	**0.1-1**	Required	Enter the range of disks in the format enclosure-number.disk-range,enclosure-number.disk-range. For example, to select disks 3 to 12 in enclosure 1 and to select disks 5 to 23 in enclosure 2, you must enter 1.3-12, 2.5-23. A RAID 10 or 50 disk group with disks in subgroups are separated by colons (with no spaces). RAID-10 example:1.1-2:1.3-4:1.7,1.10 Note: Ensure that the entered disk location is empty and the Usage column lists the range as AVAIL. The disk range specified must be of the same vendor and they must have the same description.
powervault_disk_group_name	omnia	Required	Specifies the disk group name
powervault_volumes	omnia_home	Required	Specify the volume details for powervault and NFS Server node. Multiple volumes can be defined as comma seperated values. example: omnia_home1, omnia_home2.
powervault_volume_size	100GB	Required	Enter the volume size in the format: SizeGB.
powervault_pool	**a**, A, B, b	Required	Enter the pool for the volume.
powervault_disk_partition_size		Optional	Specify the disk partition size as a percentage of available disk space.
powervault_server_nic		Optional	Enter the NIC of the server to which the PowerVault Storage is connected. Make sure nfs server also has 3 nics (for internet, OS provision and powervault connection). The nic should be specified based on the provisioned OS on nfs server.
snmp_trap_destination		Optional	The trap destination IP address is the IP address of the SNMP Server where the trap will be sent. If this variable is left blank, SNMP will be disabled. Omnia will not validate this IP.
snmp_community_name	pub-	Op-	The SNMP community string used to access statistics, MAC addresses and IPs stored within

Run the playbook:

```
cd storage
ansible-playbook powervault.yml -i inventory -e powervault_username="" -e powervault_
↪password=""
```

- Where the `inventory` refers to a list of all nodes separated by a newline.
- `powervault_username` and `powervault_password` are the credentials used to administrate the array.

Configuring NFS servers

To configure an NFS server, enter the following parameters in `storage/nfs_server_input.yml`

Run the playbook:

```
cd storage
ansible-playbook nfs_sas.yml -i inventory
```

- Where the `inventory` refers to a list of all nodes in the format of NFS server inventory file

CHAPTER

THREE

ROLES

From Omnia 1.4, all of Omnia's many features are available via collections. Collections allow users to to choose different features and customize their deployment journey individual to their needs. Alternatively, all features can be invoked using the two top level scripts:

1. provision.yml
2. Omnia.yml

Below is a list of all Omnia's roles:

3.1 Provision

3.1.1 Input Parameters for Provision Tool

Fill in all provision-specific parameters in `input/provision_config.yml`

Name	Default, Accepted Values	Required?	Additional Information
public_nic	eno2	required	The NIC/ethernet card that is connected to the public internet.
admin_nic	eno1	required	The NIC/ethernet card that is used for shared LAN over Management (LOM) capability.
admin_nic_subnet	172.29.0.0	required	The intended subnet for shared LOM capability. Note that since the last 16 bits/2 octets of IPv4 are dynamic, please ensure that the parameter value is set to x.x.0.0.
pxe_nic	eno1	required	This NIC used to obtain routing information.
pxe_nic_start_range	172.29.0.100	required	The start of the DHCP range used to assign IPv4 addresses. When the PXE range and BMC subnet are provided, corresponding NICs will be assigned IPs with the same 3rd and 4th octets. Ensure that these ranges contain enough IPs to be double the number of iDRACs present in the cluster.
pxe_nic_end_range	172.29.0.200	required	The end of the DHCP range used to assign IPv4 addresses. When the PXE range and BMC subnet are provided, corresponding NICs will be assigned IPs with the same 3rd and 4th octets. Ensure that these ranges contain enough IPs to be double the number of iDRACs present in the cluster.
ib_nic_subnet		optional	If provided, Omnia will assign static IPs to IB NICs on the compute nodes within the provided subnet. Note that since the last 16 bits/2 octets of IPv4 are dynamic, please ensure that the parameter value is set to x.x.0.0. When the PXE range and BMC subnet are provided, corresponding NICs will be assigned IPs with the same 3rd and 4th octets. IB nics should be prefixed ib.
bmc_nic_subnet		optional	If provided, Omnia will

3.1.2 Before You Run The Provision Tool

- (Recommended) Run `prereq.sh` to get the system ready to deploy Omnia. Alternatively, ensure that Ansible 2.12.9 and Python 3.8 are installed on the system. SELinux should also be disabled.
- To provision the bare metal servers, download one of the following ISOs for deployment:
  1. Rocky 8
  2. RHEL 8.x
- To dictate IP address/MAC mapping, a host mapping file can be provided. If the mapping file is not provided and the variable is left blank, a default mapping file will be created by querying the switch. Use the pxe_mapping_file.csv to create your own mapping file.
- Ensure that all connection names under the network manager match their corresponding device names.

```
nmcli connection
```

In the event of a mismatch, edit the file `/etc/sysconfig/network-scripts/ifcfg-<nic name>` using vi editor.

- All target hosts should be set up in PXE mode before running the playbook.
- If RHEL is in use on the control plane, enable RedHat subscription. Not only does Omnia not enable RedHat subscription on the control plane, package installation may fail if RedHat subscription is disabled.
- Users should also ensure that all repos are available on the RHEL control plane.
- Ensure that the `pxe_nic` and `public_nic` are in the firewalld zone: public.

Note:

- After configuration and installation of the cluster, changing the control plane is not supported. If you need to change the control plane, you must redeploy the entire cluster.
- If there are errors while executing any of the Ansible playbook commands, then re-run the playbook.

3.1.3 Running The Provision Tool

1. Edit the `input/provision_config.yml` file to update the required variables.

Warning: The IP address *192.168.25.x* is used for PowerVault Storage communications. Therefore, do not use this IP address for other configurations.

2. To deploy the Omnia provision tool, run the following command

```
cd provision
ansible-playbook provision.yml
```

3. By running `provision.yml`, the following configurations take place:

   i. All compute nodes in cluster will be enabled for PXE boot with osimage mentioned in `provision_config.yml`.

   ii. A PostgreSQL database is set up with all relevant cluster information such as MAC IDs, hostname, admin IP, infiniband IPs, BMC IPs etc.

      To access the DB, run:

```
psql -U postgres

\c omniadb
```

To view the schema being used in the cluster: `\dn`

To view the tables in the database: `\dt`

To view the contents of the `nodeinfo` table: `select * from cluster.nodeinfo`

```
id | servicetag |     admin_mac     |         hostname         |   admin_
↪ip    | bmc_ip | ib_ip

----+------------+-------------------+--------------------------+--------
↪------+-------+-------


1 |            | 00:c0:ff:43:f9:44 | node00001.winter.cluster | 172.29.1.
↪253 |        |
2 |            | 70:b5:e8:d1:84:22 | node00002.winter.cluster | 172.29.1.
↪254 |        |
3 |            | b8:ca:3a:71:25:5c | node00003.winter.cluster | 172.29.1.
↪255 |        |
4 |            | 8c:47:be:c7:6f:c1 | node00004.winter.cluster | 172.29.2.
↪0   |        |
5 |            | 8c:47:be:c7:6f:c2 | node00005.winter.cluster | 172.29.2.
↪1   |        |
6 |            | b0:26:28:5b:80:18 | node00006.winter.cluster | 172.29.2.
↪2   |        |
7 |            | b0:7b:25:de:71:de | node00007.winter.cluster | 172.29.2.
↪3   |        |
8 |            | b0:7b:25:ee:32:fc | node00008.winter.cluster | 172.29.2.
↪4   |        |
9 |            | d0:8e:79:ba:6a:58 | node00009.winter.cluster | 172.29.2.
↪5   |        |
10|            | d0:8e:79:ba:6a:5e | node00010.winter.cluster | 172.29.2.
↪6   |        |
```

iii. Offline repositories will be created based on the OS being deployed across the cluster.

Once the playbook execution is complete, ensure that PXE boot and RAID configurations are set up on remote nodes. Users are then expected to reboot target servers to provision the OS.

Note:

- If the cluster does not have access to the internet, AppStream will not function. To provide internet access through the control plane (via the PXE network NIC), update `primary_dns` and `secondary_dns` in `provision_config.yml` and run `provision.yml`
- All ports required for xCAT to run will be opened (For a complete list, check out the Security Configuration Document).
- After running `provision.yml`, the file `input/provision_config.yml` will be encrypted. To edit file, use the command: `ansible-vault edit provision_config.yml --vault-password-file .provision_vault_key`
- To re-provision target servers `provision.yml` can be re-run. Alternatively, use the following steps:

- Use `lsdef -t osimage | grep install-compute` to get a list of all valid OS profiles.
- Use `nodeset all osimage=<selected OS image from previous command>` to provision the OS on the target server.
- PXE boot the target server to bring up the OS.

Warning: Once xCAT is installed, restart your SSH session to the control plane to ensure that the newly set up environment variables come into effect.

Adding a new node

A new node can be added using one of two ways:

1. Using a mapping file:
   - Update the existing mapping file by appending the new entry (without the disrupting the older entries) or provide a new mapping file by pointing `pxe_mapping_file_path` in `provision_config.yml` to the new location.
   - Run `provision.yml`.
2. Using the switch IP:
   - Run `provision.yml` once the switch has discovered the potential new node.

3.1.4 After Running the Provision Tool

Once the **servers are provisioned**, run the post provision script to:

- Configure iDRAC IP or BMC IP if `bmc_nic_subnet` is provided in `input/provision_config.yml`.
- Configure Infiniband static IPs on remote nodes if `ib_nic_subnet` is provided in `input/provision_config.yml`.
- Set hostname for the remote nodes.
- Invoke `network.yml` and `accelerator.yml` to install OFED, CUDA toolkit and ROCm drivers.
- Create `node_inventory` in `/opt/omnia` listing provisioned nodes.

```
cat /opt/omnia/node_inventory
172.29.0.100 service_tag=XXXXXXX operating_system=RedHat
172.29.0.101 service_tag=XXXXXXX operating_system=RedHat
172.29.0.102 service_tag=XXXXXXX operating_system=Rocky
172.29.0.103 service_tag=XXXXXXX operating_system=Rocky
```

Note: Before post provision script, verify redhat subscription is enabled using the `rhsm_subscription.yml` playbook in utils only if OFED or GPU accelerators are to be installed.

To run the script, use the below command::

```
ansible-playbook post_provision.yml
```

3.2 Network

In your HPC cluster, connect the Mellanox InfiniBand switches using the Fat-Tree topology. In the fat-tree topology, switches in layer 1 are connected through the switches in the upper layer, i.e., layer 2. And, all the compute nodes in the cluster, such as PowerEdge servers and PowerVault storage devices, are connected to switches in layer 1. With this topology in place, we ensure that a 1x1 communication path is established between the compute nodes. For more information on the fat-tree topology, see Designing an HPC cluster with Mellanox infiniband-solutions.

Note:

- From Omnia 1.4, the Subnet Manager runs on the target Infiniband switches and not the control plane.
- The post-provision script calls `network.yml` to install OFED drivers.

Omnia uses the server-based Subnet Manager (SM). SM runs in a Kubernetes namespace on the control plane. To enable the SM, Omnia configures the required parameters in the `opensm.conf` file. Based on the requirement, the parameters can be edited.

Some of the network features Omnia offers are:

1. Mellanox OFED
2. Infiniband switch configuration

To install OFED drivers, enter all required parameters in `input/network_config.yml`:

Name	Default, accepted values	Required?	Purpose
mlnx_ofed_offline_path		optional	Absolute path to local copy of .tgz file containing mlnx_ofed package. The package can be downloaded from https://network.nvidia.com/products/infiniband-drivers/linux/mlnx_ofed/.
mlnx_ofed_version	5.4-2.4.1.3	optional	Indicates the version of mlnx_ofed to be downloaded. If `mlnx_ofed_offline_path` is not given, declaring this variable is mandatory.
mlnx_ofed_add_kernel_support	optional	required	Indicates whether the kernel needs to be upgraded to be compatible with mlnx_ofed.

To run the script:

```
cd network
ansible-playbook network.yml
```

3.3 Scheduler

3.3.1 Input Parameters for the Cluster

These parameters is located in `input/omnia_config.yml`

Parameter Name	Default Value	Additional Information
mariadb_password	password	Password used to access the Slurm database.Required Length: 8 characters. The password must not contain -,, ‘,”
k8s_version	1.19.3	Kubernetes VersionAccepted Values: “1.16.7” or “1.19.3”
k8s_cni	calico	CNI type used by Kubernetes.Accepted values: calico, flannel
k8s_pod_network_cidr	10.244.0.0/16	Kubernetes pod network CIDR
docker_username		Username to login to Docker. A kubernetes secret will be created and patched to the service account in default namespace.This value is optional but suggested to avoid docker pull limit issues
docker_password		Password to login to DockerThis value is mandatory if a docker_username is provided
ansible_config_file_path	/etc/ansible	Path where the ansible.cfg file can be found.If dnf is used, the default value is valid. If pip is used, the variable must be set manually
login_node_required	true	Boolean indicating whether the login node is required or not
ldap_required	false	Boolean indicating whether ldap client is required or not
ldap_server_ip		LDAP server IP. Required if `ldap_required` is true.
ldap_connection_type	TLS	For a TLS connection, provide a valid certification path. For an SSL connection, ensure port 636 is open.
ldap_ca_cert_path	/etc/openldap/certs/omnialdap.pem	This variable accepts Server Certificate Path. Make sure certificate is present in the path provided. The certificate should have .pem or .crt extension. This variable is mandatory if connection type is TLS.
user_home_dir	/home	This variable accepts the user home directory path for ldap configuration. If nfs mount is created for user home, make sure you provide the LDAP users mount home directory path.
ldap_bind_username	admin	If LDAP server is configured with bind dn then bind dn user to be provided. If this value is not provided (when bind is configured in server) then ldap authentication fails.
ldap_bind_password		If LDAP server is configured with bind dn then bind dn password to be provided. If this value is not provided (when bind is configured in server) then ldap authentication fails.
domain_name	omnia.test	Sets the intended domain name
realm_name	OMNIA.TEST	Sets the intended realm name
directory_manager_password		Password authenticating admin level access to the Directory for system management tasks. It will be added to the instance of directory server created for IPA.Required Length: 8 characters. The password must not contain -,, ‘,”
kerberos_admin_password		“admin” user password for the IPA server on RockyOS.
enable_secure_login_node	false	Boolean value deciding whether security features are enabled on the Login Node.
powervault_ip		IP of the powervault connected to the NFS server. Mandatory field when nfs_node group is defined with an IP and omnia is required to configure nfs server.

Note: When `ldap_required` is true, `login_node_required` and `freeipa_required` have to be false.

3.3.2 Before You Build Clusters

- Verify that all inventory files are updated.
- If the target cluster requires more than 10 kubernetes nodes, use a docker enterprise account to avoid docker pull limits.
- Verify that all nodes are assigned a group. Use the inventory as a reference.
  - The manager group should have exactly 1 manager node.
  - The compute group should have at least 1 node.
  - The login_node group is optional. If present, it should have exactly 1 node.
  - Users should also ensure that all repos are available on the target nodes running RHEL.

Note: The inventory file accepts both IPs and FQDNs as long as they can be resolved by DNS.

- For RedHat clusters, ensure that RedHat subscription is enabled on all target nodes.

Features enabled by omnia.yml

- Slurm: Once all the required parameters in omnia_config.yml are filled in, `omnia.yml` can be used to set up slurm.
- LDAP client support: The manager and compute nodes will have LDAP installed but the login node will be excluded.
- FreeIPA support
- Login Node (Additionally secure login node)
- Kubernetes: Once all the required parameters in omnia_config.yml are filled in, `omnia.yml` can be used to set up kubernetes.
- BeeGFS bolt on installation
- NFS bolt on support

3.3.3 Building Clusters

1. In the `input/omnia_config.yml` file, provide the required details.

Note: Without the login node, Slurm jobs can be scheduled only through the manager node.

2. Create an inventory file in the *omnia* folder. Add login node IP address under the manager node IP address under the *[manager]* group, compute node IP addresses under the *[compute]* group, and Login node IP under the *[login_node]* group,. Check out the sample inventory for more information.

Note:

- Omnia checks for red hat subscription being enabled on RedHat nodes as a pre-requisite. Not having Red Hat subscription enabled on the manager node will cause `omnia.yml` to fail. If compute nodes do not have Red Hat subscription enabled, `omnia.yml` will skip the node entirely.
- Omnia creates a log file which is available at: `/var/log/omnia.log`.
- If only Slurm is being installed on the cluster, docker credentials are not required.

3. To run `omnia.yml`:

   ansible-playbook omnia.yml -i inventory

Note: To visualize the cluster (Slurm/Kubernetes) metrics on Grafana (On the control plane) during the run of `omnia.yml`, add the parameters `grafana_username` and `grafana_password` (That is `ansible-playbook omnia.yml -i inventory -e grafana_username="" -e grafana_password=""`). Alternatively, Grafana is not installed by `omnia.yml` if it's not available on the Control Plane.

Using Skip Tags

Using skip tags, the scheduler running on the cluster can be set to Slurm or Kubernetes while running the `omnia.yml` playbook. This choice can be made depending on the expected HPC/AI workloads.

- Kubernetes: `ansible-playbook omnia.yml -i inventory --skip-tags "kubernetes"` (To set Slurm as the scheduler)
- Slurm: `ansible-playbook omnia.yml -i inventory --skip-tags "slurm"` (To set Kubernetes as the scheduler)

Note:

- If you want to view or edit the `omnia_config.yml` file, run the following command:
  - `ansible-vault view omnia_config.yml --vault-password-file .omnia_vault_key` – To view the file.
  - `ansible-vault edit omnia_config.yml --vault-password-file .omnia_vault_key` – To edit the file.
- It is suggested that you use the ansible-vault view or edit commands and that you do not use the ansible-vault decrypt or encrypt commands. If you have used the ansible-vault decrypt or encrypt commands, provide 644 permission to `omnia_config.yml`.

Kubernetes Roles

As part of setting up Kubernetes roles, `omnia.yml` handles the following tasks on the manager and compute nodes:

- Docker is installed.
- Kubernetes is installed.
- Helm package manager is installed.
- All required services are started (Such as kubelet).
- Different operators are configured via Helm.
- Prometheus is installed.

Slurm Roles

As part of setting up Slurm roles, `omnia.yml` handles the following tasks on the manager and compute nodes:

- Slurm is installed.
- All required services are started (Such as slurmd, slurmctld, slurmdbd).
- Prometheus is installed to visualize slurm metrics.
- Lua and Lmod are installed as slurm modules.

- Slurm restd is set up.

Login node

If a login node is available and mentioned in the inventory file, the following tasks are executed:

- Slurmd is installed.
- All required configurations are made to `slurm.conf` file to enable a slurm login node.
- FreeIPA (the default authentication system on the login node) is installed to provide centralized authentication

Hostname requirements

- In the `examples` folder, a **mapping_host_file.csv** template is provided which can be used for DHCP configuration. The header in the template file must not be deleted before saving the file. It is recommended to provide this optional file as it allows IP assignments provided by Omnia to be persistent across control plane reboots.
- The Hostname should not contain the following characters: , (comma), . (period) or _ (underscore). However, the **domain name** is allowed commas and periods.
- The Hostname cannot start or end with a hyphen (-).
- No upper case characters are allowed in the hostname.
- The hostname cannot start with a number.
- The hostname and the domain name (that is: `hostname00000x.domain.xxx`) cumulatively cannot exceed 64 characters. For example, if the `node_name` provided in `input/provision_config.yml` is 'node', and the `domain_name` provided is 'omnia.test', Omnia will set the hostname of a target compute node to 'node00001.omnia.test'. Omnia appends 6 digits to the hostname to individually name each target node.

Note:

- To enable the login node, ensure that `login_node_required` in `input/omnia_config.yml` is set to true.
- To enable security features on the login node, ensure that `enable_secure_login_node` in `input/omnia_config.yml` is set to true.
- To customize the security features on the login node, fill out the parameters in `input/omnia_security_config.yml`.

Warning: No users/groups will be created by Omnia.

Slurm job based user access

To ensure security while running jobs on the cluster, users can be assigned permissions to access compute nodes only while their jobs are running. To enable the feature:

```
cd scheduler
ansible-playbook job_based_user_access.yml -i inventory
```

Note:

- The inventory queried in the above command is to be created by the user prior to running `omnia.yml` as `scheduler.yml` is invoked by `omnia.yml`

- Only users added to the ‘slurm’ group can execute slurm jobs. To add users to the group, use the command: `usermod -a -G slurm <username>`.

Installing LDAP Client

Manager and compute nodes will have LDAP client installed and configured if `ldap_required` is set to true. The login node does not have LDAP client installed.

Warning: No users/groups will be created by Omnia.

3.3.4 FreeIPA installation on the NFS node

IPA services are used to provide account management and centralized authentication. To set up IPA services for the NFS node in the target cluster, run the following command from the `utils/cluster` folder on the control plane:

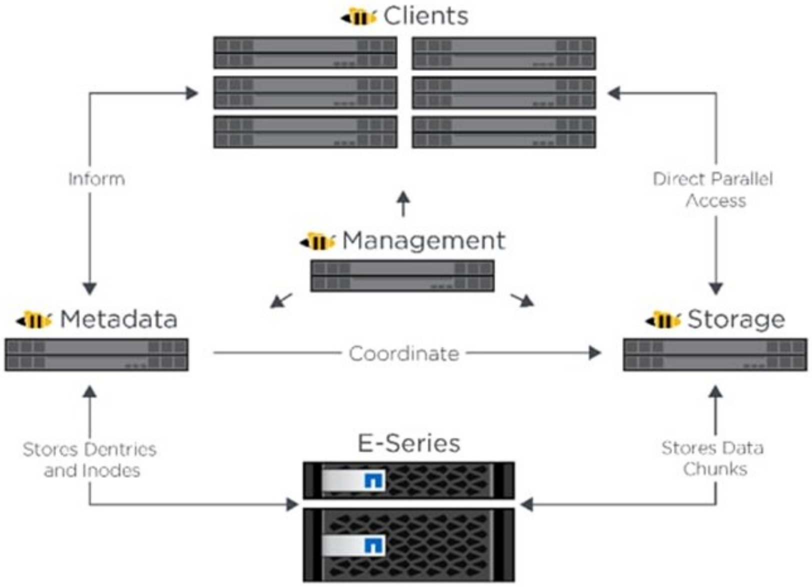
```
cd utils/cluster
ansible-playbook install_ipa_client.yml -i inventory -e kerberos_admin_password="" -e ipa_server_hostname="" -e domain_name="" -e ipa_server_ipadress=""
```

Input Parameter	Definition	Variable value
kerberos_admin_password	“admin” user password for the IPA server on RockyOS and RedHat.	The password can be found in the file `input/omnia_config.yml` .
ipa_server_hostname	The hostname of the IPA server	The hostname can be found on the manager node.
domain_name	Domain name	The domain name can be found in the file `input/omnia_config.yml`.
ipa_server_ipadress	The IP address of the IPA server	The IP address can be found on the IPA server on the manager node using the `ip a` command. This IP address should be accessible from the NFS node.

Use the format specified under NFS inventory in the Sample Files for inventory.

3.3.5 BeeGFS Bolt On

BeeGFS is a hardware-independent POSIX parallel file system (a.k.a. Software-defined Parallel Storage) developed with a strong focus on performance and designed for ease of use, simple installation, and management.



Pre Requisites before installing BeeGFS client

- If the user intends to use BeeGFS, ensure that a BeeGFS cluster has been set up with beegfs-mgmtd, beegfs-meta, beegfs-storage services running.

  Ensure that the following ports are open for TCP and UDP connectivity:

Port	Service
8008	Management service (beegfs-mgmtd)
8003	Storage service (beegfs-storage)
8004	Client service (beegfs-client)
8005	Metadata service (beegfs-meta)
8006	Helper service (beegfs-helperd)

To open the ports required, use the following steps:

1. `firewall-cmd --permanent --zone=public --add-port=<port number>/tcp`
2. `firewall-cmd --permanent --zone=public --add-port=<port number>/udp`
3. `firewall-cmd --reload`
4. `systemctl status firewalld`

- Ensure that the nodes in the inventory have been assigned **only** these roles: manager and compute.

Note:

- If the BeeGFS server (MGMTD, Meta, or storage) is running BeeGFS version 7.3.1 or higher, the security feature on the server should be disabled. Change the value of `connDisableAuthentication` to `true` in /etc/beegfs/beegfs-mgmtd.conf, /etc/beegfs/beegfs-meta.conf and /etc/beegfs/beegfs-storage.conf. Restart the services to complete the task:

```
systemctl restart beegfs-mgmtd
systemctl restart beegfs-meta
systemctl restart beegfs-storage
systemctl status beegfs-mgmtd
systemctl status beegfs-meta
systemctl status beegfs-storage
```

Note: BeeGFS with OFED capability is only supported on RHEL 8.3 and above due to limitations on BeeGFS. When setting up your cluster with RDMA support, check the BeeGFS documentation to provide appropriate values in `input/storage_config.yml`.

- If the cluster runs Rocky, ensure that versions running are compatible:

Rocky OS version	BeeGFS version
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.3.2
Rocky Linux 8.5: no OFED, OFED 5.5	7.3.2
Rocky Linux 8.6: no OFED, OFED 5.6	7.3.2
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.3.1
Rocky Linux 8.5: no OFED, OFED 5.5	7.3.1
Rocky Linux 8.6: no OFED, OFED 5.6	7.3.1
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.3.0
Rocky Linux 8.5: no OFED, OFED 5.5	7.3.0
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.2.8
Rocky Linux 8.5: no OFED, OFED 5.5	7.2.8
Rocky Linux 8.6: no OFED, OFED 5.6	7.2.8
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.2.7
Rocky Linux 8.5: no OFED, OFED 5.5	7.2.7
Rocky Linux 8.6: no OFED, OFED 5.6	7.2.7
Rocky Linux 8.5: no OFED, OFED 5.5	7.2.6
Rocky Linux 8.6: no OFED, OFED 5.6	7.2.6
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.2.5
Rocky Linux 8.4: no OFED, OFED 5.3, 5.4	7.2.4

Installing the BeeGFS client via Omnia

After the required parameters are filled in `input/storage_config.yml`, Omnia installs BeeGFS on manager and compute nodes while executing the `omnia.yml` playbook.

```
.. note::
```

- BeeGFS client-server communication can take place through TCP or RDMA. If RDMA support is required, set `beegfs_rdma_support` should be set to true. Also, OFED should be installed on all target nodes.
- For BeeGFS communication happening over RDMA, the `beegfs_mgmt_server` should be provided with the Infiniband IP of the management server.

3.3.6 NFS Bolt On

- Ensure that an external NFS server is running. NFS clients are mounted using the external NFS server's IP.
- Fill out the `nfs_client_params` variable in the `input/storage_config.yml` file in JSON format using the samples provided below.
- This role runs on manager, compute and login nodes.
- Make sure that `/etc/exports` on the NFS server is populated with the same paths listed as `server_share_path` in the `nfs_client_params` in `input/storage_config.yml`.
- Post configuration, enable the following services (using this command: `firewall-cmd --permanent --add-service=<service name>`) and then reload the firewall (using this command: `firewall-cmd --reload`).
  - nfs
  - rpc-bind
  - mountd
- Omnia supports all NFS mount options. Without user input, the default mount options are nosuid,rw,sync,hard,intr. For a list of mount options, click here.
- The fields listed in `nfs_client_params` are:
  - server_ip: IP of NFS server
  - server_share_path: Folder on which NFS server mounted
  - client_share_path: Target directory for the NFS mount on the client. If left empty, respective `server_share_path value` will be taken for `client_share_path`.
  - client_mount_options: The mount options when mounting the NFS export on the client. Default value: nosuid,rw,sync,hard,intr.
- There are 3 ways to configure the feature:
  1. **Single NFS node** : A single NFS filesystem is mounted from a single NFS server. The value of `nfs_client_params` would be:

     ```
     - { server_ip: 172.10.0.101, server_share_path: "/mnt/share", client_share_
     ↪path: "/mnt/client", client_mount_options: "nosuid,rw,sync,hard,intr" }
     ```

  2. **Multiple Mount NFS Filesystem**: Multiple filesystems are mounted from a single NFS server. The value of `nfs_client_params` would be:

     ```
     - { server_ip: 172.10.0.101, server_share_path: "/mnt/share1", client_share_
     ↪path: "/mnt/client1", client_mount_options: "nosuid,rw,sync,hard,intr" }
     - { server_ip: 172.10.0.101, server_share_path: "/mnt/share2", client_share_
     ↪path: "/mnt/client2", client_mount_options: "nosuid,rw,sync,hard,intr" }
     ```

  3. **Multiple NFS Filesystems**: Multiple filesystems are mounted from multiple NFS servers. The value of `nfs_client_params` would be:

     ```
     - { server_ip: 172.10.0.101, server_share_path: "/mnt/server1", client_share_
     ↪path: "/mnt/client1", client_mount_options: "nosuid,rw,sync,hard,intr" }
     - { server_ip: 172.10.0.102, server_share_path: "/mnt/server2", client_share_
     ↪path: "/mnt/client2", client_mount_options: "nosuid,rw,sync,hard,intr" }
     - { server_ip: 172.10.0.103, server_share_path: "/mnt/server3", client_share_
     ↪path: "/mnt/client3", client_mount_options: "nosuid,rw,sync,hard,intr" }
     ```

Warning: After an NFS client is configured, if the NFS server is rebooted, the client may not be able to reach the server. In those cases, restart the NFS services on the server using the below command:

```
systemctl disable nfs-server
systemctl enable nfs-server
systemctl restart nfs-server
```

3.4 Storage

The storage role allows users to configure PowerVault Storage devices, BeeGFS and NFS services on the cluster.

First, enter all required parameters in `input/storage_config.yml`

Name	Default, accepted values	Required?	Purpose
beegfs_support	false, true	Optional	This variable is used to install beegfs-client on compute and manager nodes
beegfs_rdma_support	false, true	Optional	This variable is used if user has RDMA-capable network hardware (e.g., InfiniBand)
beegfs_ofed_kernel_modules_path	“/usr/src/ofa_kernel/default/include”	Optional	The path where separate OFED kernel modules are installed.
beegfs_mgmt_server		Required	BeeGFS management server IP
beegfs_mounts	“/mnt/beegfs”	Optional	Beegfs-client file system mount location. If `storage_yml` is being used to change the BeeGFS mounts location, set beegfs_unmount_client to true
beegfs_unmount_client	false, true	Optional	Changing this value to true will unmount running instance of BeeGFS client and should only be used when decommisioning BeeGFS, changing the mount location or changing the BeeGFS version.
beegfs_client_version	7.2.6	Optional	Beegfs client version needed on compute and manager nodes.
beegfs_version_change	false, true	Optional	Use this variable to change the BeeGFS version on the target nodes.
nfs_client_params	{ server_ip: , server_share_path: , client_share_path: , client_mount_options: }	Optional	**If NFS client services are to be deployed, enter the configuration required here in JSON format. If left blank, no NFS configuration takes place. Possible values include:** 1. Single NFS file system: A single filesystem from a single NFS server is mounted. Sample value: - { server_ip: xx.xx.xx.xx, server_share_path: “/mnt/share”, client_share_path: “/mnt/client”, client_mount_options:

Note: If `storage.yml` is run with the `input/storage_config.yml` filled out, BeeGFS and NFS client will be set up.

Installing BeeGFS Client

- If the user intends to use BeeGFS, ensure that a BeeGFS cluster has been set up with beegfs-mgmtd, beegfs-meta, beegfs-storage services running.

  Ensure that the following ports are open for TCP and UDP connectivity:

Port	Service
8008	Management service (beegfs-mgmtd)
8003	Storage service (beegfs-storage)
8004	Client service (beegfs-client)
8005	Metadata service (beegfs-meta)
8006	Helper service (beegfs-helperd)

To open the ports required, use the following steps:

1. `firewall-cmd --permanent --zone=public --add-port=<port number>/tcp`
2. `firewall-cmd --permanent --zone=public --add-port=<port number>/udp`
3. `firewall-cmd --reload`
4. `systemctl status firewalld`

- Ensure that the nodes in the inventory have been assigned **only** these roles: manager and compute.

Note:

- When working with RHEL, ensure that the BeeGFS configuration is supported using the link here.
- If the BeeGFS server (MGMTD, Meta, or storage) is running BeeGFS version 7.3.1 or higher, the security feature on the server should be disabled. Change the value of `connDisableAuthentication` to `true` in /etc/beegfs/beegfs-mgmtd.conf, /etc/beegfs/beegfs-meta.conf and /etc/beegfs/beegfs-storage.conf. Restart the services to complete the task:

```
systemctl restart beegfs-mgmtd
systemctl restart beegfs-meta
systemctl restart beegfs-storage
systemctl status beegfs-mgmtd
systemctl status beegfs-meta
systemctl status beegfs-storage
```

NFS bolt-on

- Ensure that an external NFS server is running. NFS clients are mounted using the external NFS server's IP.
- Fill out the `nfs_client_params` variable in the `storage_config.yml` file in JSON format using the samples provided above.
- This role runs on manager, compute and login nodes.
- Make sure that `/etc/exports` on the NFS server is populated with the same paths listed as `server_share_path` in the `nfs_client_params` in `omnia_config.yml`.

- Post configuration, enable the following services (using this command: `firewall-cmd --permanent --add-service=<service name>`) and then reload the firewall (using this command: `firewall-cmd --reload`).
  - nfs
  - rpc-bind
  - mountd
- Omnia supports all NFS mount options. Without user input, the default mount options are nosuid,rw,sync,hard,intr. For a list of mount options, click here.
- The fields listed in `nfs_client_params` are:
  - server_ip: IP of NFS server
  - server_share_path: Folder on which NFS server mounted
  - client_share_path: Target directory for the NFS mount on the client. If left empty, respective `server_share_path value` will be taken for `client_share_path`.
  - client_mount_options: The mount options when mounting the NFS export on the client. Default value: nosuid,rw,sync,hard,intr.
- There are 3 ways to configure the feature:
  1. **Single NFS node** : A single NFS filesystem is mounted from a single NFS server. The value of `nfs_client_params` would be:

     ```
     - { server_ip: xx.xx.xx.xx, server_share_path: "/mnt/share", client_share_path:
     ↪"/mnt/client", client_mount_options: "nosuid,rw,sync,hard,intr" }
     ```

  2. **Multiple Mount NFS Filesystem**: Multiple filesystems are mounted from a single NFS server. The value of `nfs_client_params` would be:

     ```
     - { server_ip: xx.xx.xx.xx, server_share_path: "/mnt/server1", client_share_
     ↪path: "/mnt/client1", client_mount_options: "nosuid,rw,sync,hard,intr" }
     - { server_ip: xx.xx.xx.xx, server_share_path: "/mnt/server2", client_share_
     ↪path: "/mnt/client2", client_mount_options: "nosuid,rw,sync,hard,intr" }
     ```

  3. **Multiple NFS Filesystems**: Multiple filesystems are mounted from multiple NFS servers. The value of `nfs_client_params` would be:

     ```
     - { server_ip: xx.xx.xx.xx, server_share_path: "/mnt/server1", client_share_
     ↪path: "/mnt/client1", client_mount_options: "nosuid,rw,sync,hard,intr" }
     - { server_ip: yy.yy.yy.yy, server_share_path: "/mnt/server2", client_share_
     ↪path: "/mnt/client2", client_mount_options: "nosuid,rw,sync,hard,intr" }
     - { server_ip: zz.zz.zz.zz, server_share_path: "/mnt/server3", client_share_
     ↪path: "/mnt/client3", client_mount_options: "nosuid,rw,sync,hard,intr" }
     ```

To run the playbook:

```
cd omnia/storage
ansible-playbook storage.yml -i inventory
```

(Where inventory refers to the inventory file listing manager, login_node and compute nodes.)

3.5 Accelerator

The accelerator role allows users to set up the AMD ROCm platform or the CUDA Nvidia toolkit. These tools allow users to unlock the potential of installed GPUs.

Enter all required parameters in `input/accelerator_config.yml`.

Name	Default, Accepted Values	Required?	Information
amd_gpu_version	22.20.3	optional	This variable accepts the amd gpu version for the RHEL specific OS version. Verify if the version provided is present in the repo for the OS version on your node. Verify the url for the compatible version: https://repo.radeon.com/amdgpu/ . If 'latest' is provided in the variable and the compute os version is rhel 8.5. Then the url transforms to https://repo.radeon.com/amdgpu/latest/rhel/8.5/main/x86_64/
amd_rocm_version	latest/main	optional	Required AMD ROCm driver version. Make sure the subscription is enabled for rocm installation because rocm packages are present in code ready builder repo for RHEL. If 'latest' is provided in the variable, the url transforms to https://repo.radeon.com/rocm/centos8/latest/main/. Only single instance is supported by Omnia.
cuda_toolkit_version	latest	optional	Required CUDA toolkit version. By default latest cuda is installed unless cuda_toolkit_path is specified. Default: latest (11.8.0).
cuda_toolkit_path		optional	If the latest cuda toolkit is not required, provide an offline copy of the toolkit installer in the path specified. (Take an RPM copy of the toolkit from here). If `cuda_toolkit_version` is not latest, giving `cuda_toolkit_path` is mandatory.
cuda_stream	latest-dkms	optional	A stream in CUDA is a sequence of operations that execute on the device in the order in which they are issued by the host code.

Note:

- For target nodes running RedHat, ensure that redhat subscription is enabled before running `accelerator.yml`
- The post-provision script calls `accelerator.yml` to install CUDA and ROCm drivers.

To install all the latest GPU drivers and toolkits, run:

```
cd accelerator
ansible-playbook accelerator.yml -i inventory
```

(where inventory consists of manager, compute and login nodes)

The following configurations take place when running `accelerator.yml`

i. Servers with AMD GPUs are identified and the latest GPU drivers and ROCm platforms are downloaded and installed.

ii. Servers with NVIDIA GPUs are identified and the specified CUDA toolkit is downloaded and installed.

iii. For the rare servers with both NVIDIA and AMD GPUs installed, all the above mentioned download-ables are installed to the server.

iv. Servers with neither GPU are skipped.

3.6 Monitor

The monitor role sets up Grafana , Prometheus and Loki as Kubernetes pods.

Setting Up Monitoring

1. To set up monitoring, enter all required variables in `monitor/monitor_config.yml`.

Name	Default, Accepted Values	Required?	Additional Information
docker_username		optional	Username for Dockerhub account. This will be used for Docker login and a kubernetes secret will be created and patched to service account in default namespace. This kubernetes secret can be used to pull images from private repositories.
docker_password		optional	Password for Dockerhub account. This field is mandatory if `docker_username` is provided.
appliance_k8s_pod_net_cidr	192.168.0.0/16	required	Kubernetes pod network CIDR for appliance k8s network. Make sure this value does not overlap with any of the host networks.
grafana_username		required	The username for grafana UI. The length of username should be at least 5 characters. The username must not contain -,, ‘,”
grafana_password		required	Password used for grafana UI. The length of the password should be at least 5 characters. The password must not contain -,, ‘,”. Do not use “admin” in this field.
mount_location	/opt/omnia/telemetry	required	The path where the Grafana persistent volume will be mounted. If telemetry is set up, all telemetry related files will also be stored and both timescale and mysql databases will be mounted to this location. ‘/’ is mandatory at the end of the path.

Note: After running `monitor.yml`, the file `input/monitor_config.yml` will be encrypted. To edit the file, use `ansible-vault edit monitor_config.yml --vault-password-file .monitor_vault_key`.

2. Run the playbook using the following command:

```
cd monitor
ansible-playbook monitor.yml
```

3.7 Utils

The Utilities role allows users to set up certain tasks such as

3.7.1 Extra Packages for Enterprise Linux (EPEL)

This script is used to install the following packages:

1. PDSH
2. PDSH RCMD SSH
3. clustershell

To run the script:

```
cd omnia/utils
ansible-playbook install_hpc_thirdparty_packages.yml -i inventory
```

Where the inventory refers to a file listing all manager and compute nodes per the format provided in inventory file.

3.7.2 Updating Kernels on RHEL

Pre-requisites

1. Subscription should be available on nodes
2. Kernels to be upgraded should be available. To verify the status of the kernels, use `yum list kernel`
3. The input kernel revision cannot be a RHEL 7.x supported kernel version. e.g. "3.10.0-54.0.1" to "3.10.0-1160".
4. Input needs to be passed during execution of the playbook.

Executing the Kernel Upgrade:

Via CLI:

```
cd omnia/utils
ansible-playbook kernel_upgrade.yml -i inventory -e rhsm_kernel_version=x.xx.x-xxxx
```

Where the inventory refers to a file listing all manager and compute nodes per the format provided in inventory file.

3.7.3 Red Hat Subscription

Required Parameters

Variable	Default, Choices	Description
redhat_subscription_method	**portal**, satellite	Method to use for activation of subscription management. If Satellite, the role will determine the Satellite Server version (5 or 6) and take the appropriate registration actions.
redhat_subscription_release		RHEL release version (e.g. 8.1)
redhat_subscription_force_register	**false**, true	Register the system even if it is already registered.
redhat_subscription_pool_ids		Specify subscription pool IDs to consume.A pool ID may be specified as a string - just the pool ID (ex. 0123456789abcdef0123456789abcdef) or as a dict with the pool ID as the key, and a quantity as the value. If the quantity is provided, it is used to consume multiple entitlements from a pool (the pool must support this).
redhat_subscription_repos		The list of repositories to enable or disable.When providing multiple values, a YAML list or a comma separated list are accepted.
redhat_subscription_repos_state	**enabled**, disabled	The state of all repos in redhat_subscription_repos.
redhat_subscription_repos_purge	**false**, true	This parameter disables all currently enabled repositories that are not not specified in redhat_subscription_repos. Only set this to true if the redhat_subscription_repos field has multiple repos.
redhat_subscription_server_hostname	subscription.rhn.redhat.com	FQDN of subscription serverMandatory field if redhat_subscription_method is set to satellite
redhat_subscription_port	**443**, 8443	Port to use when connecting to subscription server.Set 443 for Satellite or RHN. If capsule is used, set 8443.
redhat_subscription_insecure	**false**, true	Disable certificate validation.
redhat_subscription_ssl_verify_depth	**3**	Sets the number of certificates which should be used to verify the servers identity.This is an advanced control which can be used to secure on premise installations.
redhat_subscription_proxy_proto	**http**, https	Set this to a non-blank value if subscription-manager should use a reverse proxy to access the subscription service. This sets the protocol for the reverse proxy.
redhat_subscription_proxy_hostname		Set this to a non-blank value if subscription-manager should use a reverse proxy to access the subscription service.
redhat_subscription_proxy_port		Set this to a non-blank value if subscription-manager should use a reverse proxy to access the subscription service. This sets the username for the reverse proxy.
redhat_subscription_proxy_user		Set this to a non-blank value if subscription-manager should use a reverse proxy to access the subscription service. This sets the username for the reverse proxy.
redhat_subscription_proxy_password		Set this to a non-blank value if subscription-manager should use a reverse proxy to access the subscription service. This sets the password for the reverse proxy.
redhat_subscription_baseurl	https://cdn.redhat.com	This setting is the prefix for all content which is managed by the subscription service.This should be the hostname for the Red Hat CDN, the local Satellite or Capsule depending on your deployment. This field is mandatory if redhat_subscription_method is set to satellite
redhat_subscription_manage_repos	**true**, false	Set this to true if subscription manager should manage a yum repos file. If set, it will manage the file /etc/yum.repos.d/redhat.repo .If set to false, the subscription is only used for tracking purposes, not content. The /etc/yum.repos.d/redhat.repo file will either be purged or deleted.
redhat_subscription_full_refresh_on_yum	**false**, true	Set to true if the /etc/yum.repos.d/redhat.repo should be updated with every server command. This will make yum less efficient, but can ensure that the most recent data is brought down from the subscription service.
redhat_subscription_report_package_profile	**true**, false	Set to true if rhsmcertd should report the system's current package profile to the subscription service. This report helps the subscription service provide better errata notifications.
redhat_subscription_cert_check_interval	**240**	The number of minutes between runs of the rhsmcertd daemon.
redhat_subscription_auto_attach_interval	**1440**	The number of minutes between attempts to run auto-attach on this consumer.

Before running omnia.yml, it is mandatory that red hat subscription be set up on compute nodes running RHEL.

- To set up Red hat subscription, fill in the rhsm_config.yml file. Once it's filled in, run the template using Ansible.
- The flow of the playbook will be determined by the value of redhat_subscription_method in rhsm_config.yml.
  - If redhat_subscription_method is set to portal, pass the values username and password. For CLI, run the command:

    ```
    cd utils
    ansible-playbook rhsm_subscription.yml -i inventory -e redhat_subscription_
    ↪username= "<username>" -e redhat_subscription_password="<password>"
    ```

  - If redhat_subscription_method is set to satellite, pass the values organizational identifier and activation key. For CLI, run the command:

    ```
    cd utils
    ansible-playbook rhsm_subscription.yml -i inventory -e redhat_subscription_
    ↪activation_key= "<activation-key>" -e redhat_subscription_org_id ="<org-id>"
    ```

Where the inventory refers to a file listing all manager and compute nodes per the format provided in inventory file.

3.7.4 Red Hat Unsubscription

To disable subscription on RHEL nodes, the red_hat_unregister_template has to be called:

```
cd utils
ansible_playbook rhsm_unregister.yml -i inventory
```

3.7.5 Set PXE NICs to Static

Use the below playbook to optionally set all PXE NICs on provisioned nodes to 'static'.

To run the playbook:

```
cd utils
ansible-playbook configure_pxe_static.yml -i inventory
```

Where inventory refers to a list of IPs separated by newlines:

```
xxx.xxx.xxx.xxx
yyy.yyy.yyy.yyy
```

3.7.6 FreeIPA installation on the NFS node

IPA services are used to provide account management and centralized authentication. To set up IPA services for the NFS node in the target cluster, run the following command from the `utils/cluster` folder on the control plane:

```
cd utils/cluster
ansible-playbook install_ipa_client.yml -i inventory -e kerberos_admin_password="" -e ipa_server_hostname="" -e domain_name="" -e ipa_server_ipadress=""
```

Input Parameter	Definition	Variable value
kerberos_admin_password	"admin" user password for the IPA server on RockyOS and RedHat.	The password can be found in the file `input/omnia_config.yml` .
ipa_server_hostname	The hostname of the IPA server	The hostname can be found on the manager node.
domain_name	Domain name	The domain name can be found in the file `input/omnia_config.yml`.
ipa_server_ipadress	The IP address of the IPA server	The IP address can be found on the IPA server on the manager node using the `ip a` command. This IP address should be accessible from the NFS node.

Use the format specified under NFS inventory in the Sample Files for inventory.

CHAPTER

FOUR

TROUBLESHOOTING

4.1 Known Issues

Why are some target servers not reachable after running PXE booting them?

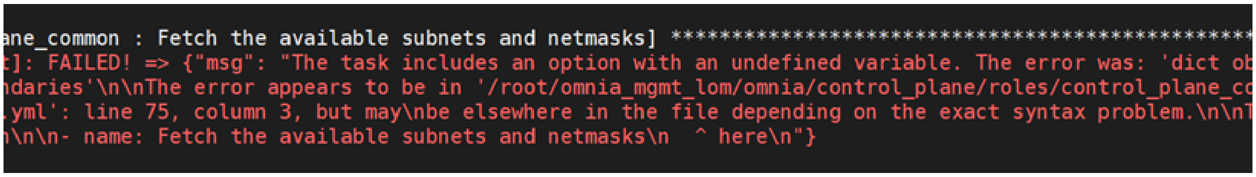
Potential Causes:

1. The server hardware does not allow for auto rebooting
2. PXE booting is hung on the node

Resolution:

1. Login to the iDRAC console to check if the server is stuck in boot errors (F1 prompt message). If true, clear the hardware error or disable POST (PowerOn Self Test).
2. Hard-reboot the server to bring up the server and verify that the boot process runs smoothly. (If it gets stuck again, disable PXE and try provisioning the server via iDRAC.)

Why does the task 'Provision: Fetch the available subnets and netmasks' fail with 'no ipv4_secondaries present'?



```
ane_common : Fetch the available subnets and netmasks] ************************************************
t]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'dict ob
ndaries'\n\nThe error appears to be in '/root/omnia_mgmt_lom/omnia/control_plane/roles/control_plane_co
.yml': line 75, column 3, but may\nbe elsewhere in the file depending on the exact syntax problem.\n\nT
n\n\n- name: Fetch the available subnets and netmasks\n  ^ here\n"}
```

Potential Cause: If a shared LOM environment is in use, the management network/host network NIC may only have one IP assigned to it.

Resolution: Ensure that the NIC used for host and data connections has 2 IPs assigned to it.

Why does provisioning RHEL 8.3 fail on some nodes with "dasbus.error.DBusError: 'NoneType' object has no attribute 'set_property'"?

This error is known to RHEL and is being addressed here. Red Hat has offered a user intervention here. Omnia recommends that in the event of this failure, any OS other than RHEL 8.3.

Why is the Infiniband NIC down after provisioning the server?

1. For servers running Rocky, enable the Infiniband NIC manually, use `ifup <InfiniBand NIC>`.
2. If your server is running LeapOS, ensure the following pre-requisites are met before manually bringing up the interface:
   1. The following repositories have to be installed:
      - Leap OSS Repo

- Leap Non OSS Repo

2. Run: `zypper install -n rdma-core librdmacm1 libibmad5 libibumad3 infiniband-diags` to install IB NIC drivers. (If the drivers do not install smoothly, reboot the server to apply the required changes)
3. Run: `service network status` to verify that `wicked.service` is running.
4. Verify that the ifcfg-< InfiniBand NIC > file is present in `/etc/sysconfig/network`.
5. Once all the above pre-requisites are met, bring up the interface manually using `ifup <InfiniBand NIC>`.

Alternatively, run `network.yml` or `post_provision.yml` (Only if the nodes are provisioned using Omnia) to activate the NIC.

Why does the Task [infiniband_switch_config : Authentication failure response] fail with the message 'Status code was -1 and not [302]: Request failed: <urlopen error [Errno 111] Connection refused>' on Infiniband Switches when running infiniband_switch_config.yml?

To configure a new Infiniband Switch, it is required that HTTP and JSON gateway be enabled. To verify that they are enabled, run:

`show web` (To check if HTTP is enabled)

`show json-gw` (To check if JSON Gateway is enabled)

To correct the issue, run:

`web http enable` (To enable the HTTP gateway)

`json-gw enable` (To enable the JSON gateway)

While configuring xCAT, why does the ``provision.yml`` fail during the Run import command?

Cause:

- The mounted .iso file is corrupt.

  Resolution:

1. Go to **var** -> **log** -> **xCAT** -> **xCAT.log** to view the error.
2. If the error message is **repo verification failed**, the .iso file is not mounted properly.
3. Verify that the downloaded .iso file is valid and correct.
4. Delete the Cobbler container using `docker rm -f cobbler` and rerun `provision.yml`.

Why does PXE boot fail with tftp timeout or service timeout errors?

Potential Causes:

- RAID is configured on the server.
- Two or more servers in the same network have xCAT services running.
- The target compute node does not have a configured PXE device with an active NIC.

  Resolution:

1. Create a Non-RAID or virtual disk on the server.
2. Check if other systems except for the control plane have xcatd running. If yes, then stop the xCAT service using the following commands: `systemctl stop xcatd`.
3. On the server, go to `BIOS Setup -> Network Settings -> PXE Device`. For each listed device (typically 4), configure an active NIC under `PXE device settings`

Why do Kubernetes Pods show “ImagePullBack” or “ErrPullImage” errors in their status?

Potential Cause:

- The errors occur when the Docker pull limit is exceeded.

Resolution:

- For `omnia.yml` and `provision.yml` : Provide the docker username and password for the Docker Hub account in the *omnia_config.yml* file and execute the playbook.
- For HPC cluster, during `omnia.yml execution`, a kubernetes secret ‘dockerregcred’ will be created in default namespace and patched to service account. User needs to patch this secret in their respective namespace while deploying custom applications and use the secret as imagePullSecrets in yaml file to avoid ErrImagePull. [Click here for more info](https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/)

Note: If the playbook is already executed and the pods are in **ImagePullBack** state, then run `kubeadm reset -f` in all the nodes before re-executing the playbook with the docker credentials.

Why does the task ‘Gather facts from all the nodes’ stuck when re-running `**`omnia.yml``?

Potential Cause: Corrupted entries in the `/root/.ansible/cp/` folder. For more information on this issue, check this out!

Resolution: Clear the directory `/root/.ansible/cp/` using the following commands:

```
cd /root/.ansible/cp/

rm -rf *
```

Alternatively, run the task manually:

```
cd omnia/utils/cluster
ansible-playbook gather_facts_resolution.yml
```

What to do after a reboot if kubectl commands return: ``The connection to the server head_node_ip:port was refused - did you specify the right host or port?``

On the control plane or the manager node, run the following commands:

```
swapoff -a

systemctl restart kubelet
```

What to do if the nodes in a Kubernetes cluster reboot:

Wait for 15 minutes after the Kubernetes cluster reboots. Next, verify the status of the cluster using the following commands:

- `kubectl get nodes` on the manager node to get the real-time k8s cluster status.
- `kubectl get pods all-namespaces` on the manager node to check which the pods are in the **Running** state.
- `kubectl cluster-info` on the manager node to verify that both the k8s master and kubeDNS are in the **Running** state.

What to do when the Kubernetes services are not in the Running state:

1. Run `kubectl get pods all-namespaces` to verify that all pods are in the **Running** state.

2. If the pods are not in the **Running** state, delete the pods using the command:`kubectl delete pods <name of pod>`
3. Run the corresponding playbook that was used to install Kubernetes: `omnia.yml`, `jupyterhub.yml`, or `kubeflow.yml`.

Why do Kubernetes Pods stop communicating with the servers when the DNS servers are not responding?

Potential Cause: The host network is faulty causing DNS to be unresponsive

Resolution:

1. In your Kubernetes cluster, run `kubeadm reset -f` on all the nodes.
2. On the management node, edit the `omnia_config.yml` file to change the Kubernetes Pod Network CIDR. The suggested IP range is 192.168.0.0/16. Ensure that the IP provided is not in use on your host network.
3. Execute omnia.yml and skip slurm `ansible-playbook omnia.yml --skip-tags slurm`

Why does pulling images to create the Kubeflow timeout causing the 'Apply Kubeflow Configuration' task to fail?

Potential Cause: Unstable or slow Internet connectivity.

Resolution:

1. Complete the PXE booting/format the OS on the manager and compute nodes.
2. In the omnia_config.yml file, change the k8s_cni variable value from `calico` to `flannel`.
3. Run the Kubernetes and Kubeflow playbooks.

Why does the 'Initialize Kubeadm' task fail with 'nnode.Registration.name: Invalid value: "<Host name>"'?

Potential Cause: The control_plane playbook does not support hostnames with an underscore in it such as 'mgmt_station'.

As defined in RFC 822, the only legal characters are the following: 1. Alphanumeric (a-z and 0-9): Both uppercase and lowercase letters are acceptable, and the hostname is case-insensitive. In other words, dvader.empire.gov is identical to DVADER.EMPIRE.GOV and Dvader.Empire.Gov.

2. Hyphen (-): Neither the first nor the last character in a hostname field should be a hyphen.
3. Period (.): The period should be used only to delimit fields in a hostname (e.g., dvader.empire.gov)

What to do when Kubeflow pods are in 'ImagePullBackOff' or 'ErrImagePull' status after executing kubeflow.yml:

Potential Cause: Your Docker pull limit has been exceeded. For more information, click [here](https://www.docker.com/increase-rate-limits)

1. Delete Kubeflow deployment by executing the following command in manager node: `kfctl delete -V -f /root/k8s/omnia-kubeflow/kfctl_k8s_istio.v1.0.2.yaml`
2. Re-execute `kubeflow.yml` after 8-9 hours

What to do when omnia.yml fail with 'Error: kinit: Connection refused while getting default ccache' while completing the security role?

1. Start the sssd-kcm.socket: `systemctl start sssd-kcm.socket`
2. Re-run `omnia.yml`

What to do when Slurm services do not start automatically after the cluster reboots:

- Manually restart the slurmd services on the manager node by running the following commands:

```
systemctl restart slurmdbd
systemctl restart slurmctld
systemctl restart prometheus-slurm-exporter
```

- Run `systemctl status slurmd` to manually restart the following service on all the compute nodes.

Why do Slurm services fail?

Potential Cause: The `slurm.conf` is not configured properly.

Recommended Actions:

1. Run the following commands:

```
slurmdbd -Dvvv
slurmctld -Dvvv
```

2. Refer the `/var/lib/log/slurmctld.log` file for more information.

What causes the "Ports are Unavailable" error?

Potential Cause: Slurm database connection fails.

Recommended Actions:

1. Run the following commands::

```
slurmdbd -Dvvv
slurmctld -Dvvv
```

2. Refer the `/var/lib/log/slurmctld.log` file.
3. Check the output of `netstat -antp | grep LISTEN` for PIDs in the listening state.
4. If PIDs are in the **Listening** state, kill the processes of that specific port.
5. Restart all Slurm services:

```
slurmctl restart slurmctld on manager node

systemctl restart slurmdbd on manager node

systemctl restart slurmd on compute node
```

Why does the task 'nfs_client: Mount NFS client' fail with ``Failed to mount NFS client. Make sure NFS Server is running on IP xx.xx.xx.xx``?

Potential Cause:

- The required services for NFS may not be running:
  - nfs
  - rpc-bind
  - mountd

Resolution:

- Enable the required services using `firewall-cmd --permanent --add-service=<service name>` and then reload the firewall using `firewall-cmd --reload`.

What to do when omnia.yml fails with nfs-server.service might not be running on NFS Server. Please check or start services``?

Potential Cause: nfs-server.service is not running on the target node.

Resolution: Use the following commands to bring up the service:

```
systemctl start nfs-server.service

systemctl enable nfs-server.service
```

Why does the task 'Install Packages' fail on the NFS node with the message: ``Failure in talking to yum: Cannot find a valid baseurl for repo: base/7/x86_64.``

Potential Cause:

There are connections missing on the NFS node.

Resolution:

Ensure that there are 3 NICs being used on the NFS node:

1. For provisioning the OS
2. For connecting to the internet (Management purposes)
3. For connecting to PowerVault (Data Connection)

Why do pods and images appear to get deleted automatically?

Potential Cause:

Lack of space in the root partition (/) causes Linux to clear files automatically (Use `df -h` to diagnose the issue).

Resolution:

- Delete large, unused files to clear the root partition (Use the command `find / -xdev -size +5M | xargs ls -lh | sort -n -k5` to identify these files). Before running `monitor.yml`, it is recommended to have a minimum of 50% free space in the root partition.
- Once the partition is cleared, run `kubeadm reset -f`
- Re-run `monitor.yml`

What to do when the JupyterHub or Prometheus UI is not accessible:

Run the command `kubectl get pods namespace default` to ensure **nfs-client** pod and all Prometheus server pods are in the **Running** state.

What to do if PowerVault throws the error: ``Error: The specified disk is not available. - Unavailable disk (0.x) in disk range '0.x-x'``:

1. Verify that the disk in question is not part of any pool: `show disks`
2. If the disk is part of a pool, remove it and try again.

Why does PowerVault throw the error: ``You cannot create a linear disk group when a virtual disk group exists on the system.``?

At any given time only one type of disk group can be created on the system. That is, all disk groups on the system have to exclusively be linear or virtual. To fix the issue, either delete the existing disk group or change the type of pool you are creating.

Why does the task 'nfs_client: Mount NFS client' fail with ``No route to host``?

Potential Cause:

- There's a mismatch in the share path listed in `/etc/exports` and in `omnia_config.yml` under `nfs_client_params`.

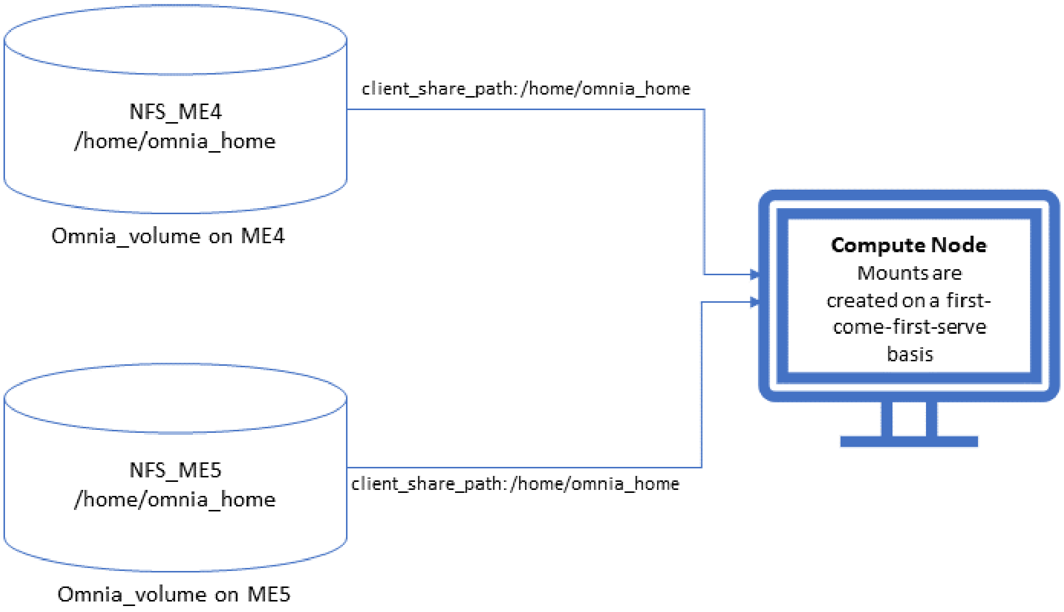
Resolution:

- Ensure that the input paths are a perfect match down to the character to avoid any errors.

Why is my NFS mount not visible on the client?

Potential Cause: The directory being used by the client as a mount point is already in use by a different NFS export.

Resolution: Verify that the directory being used as a mount point is empty by using `cd <client share path> | ls` or `mount | grep <client share path>`. If empty, re-run the playbook.


NFS_ME4
/home/omnia_home
Omnia_volume on ME4
client_share_path: /home/omnia_home
Compute Node
Mounts are created on a first-come-first-serve basis
NFS_ME5
/home/omnia_home
client_share_path: /home/omnia_home
Omnia_volume on ME5

Why does the ``BeeGFS-client`` service fail?

Potential Causes:

1. SELINUX may be enabled. (use `sestatus` to diagnose the issue)
2. Ports 8008, 8003, 8004, 8005 and 8006 may be closed. (use `systemctl status beegfs-mgmtd, systemctl status beegfs-meta, systemctl status beegfs-storage` to diagnose the issue)
3. The BeeGFS set up may be incompatible with RHEL.

Resolution:

1. If SELinux is enabled, update the file `/etc/sysconfig/selinux` and reboot the server.
2. Open all ports required by BeeGFS: 8008, 8003, 8004, 8005 and 8006
3. Check the [support matrix for RHEL or Rocky](../Support_Matrix/Software/Operating_Systems) to verify your set-up.
4. For further insight into the issue, check out `/var/log/beegfs-client.log` on nodes where the BeeGFS client is running.

Why does the task 'security: Authenticate as admin' fail?

Potential Cause: The required services are not running on the node. Verify the service status using::

```
systemctl status sssd-kcm.socket

systemctl status sssd.service
```

Resolution:

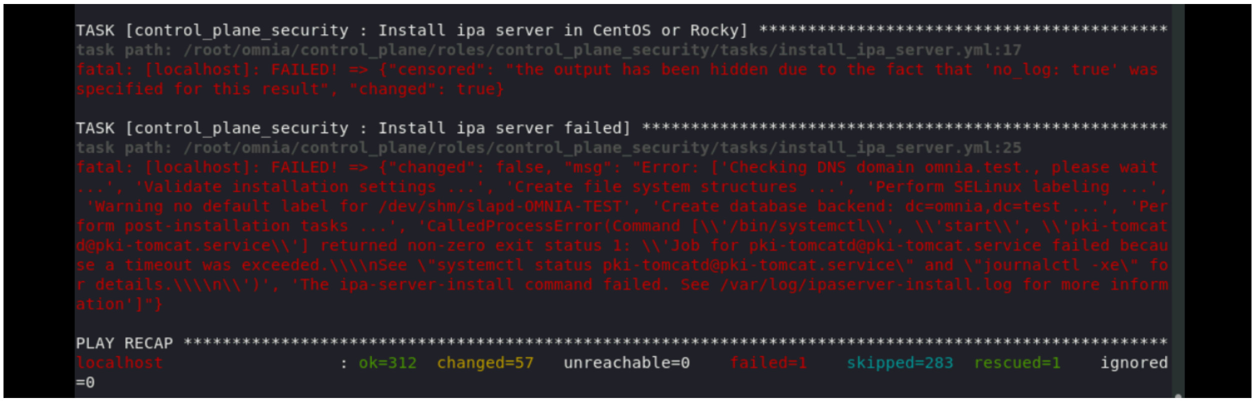
- Restart the services using::

  ```
  systemctl start sssd-kcm.socket
  systemctl start sssd.service
  ```

- Re-run `omnia.yml` using:

  ```
  ansible-playbook omnia.yml
  ```

Why does installing FreeIPA fail on RHEL servers?



```
TASK [control_plane_security : Install ipa server in CentOS or Rocky] *******************************************
task path: /root/omnia/control_plane/roles/control_plane_security/tasks/install_ipa_server.yml:17
fatal: [localhost]: FAILED! => {"censored": "the output has been hidden due to the fact that 'no_log: true' was
specified for this result", "changed": true}

TASK [control_plane_security : Install ipa server failed] *******************************************************
task path: /root/omnia/control_plane/roles/control_plane_security/tasks/install_ipa_server.yml:25
fatal: [localhost]: FAILED! => {"changed": false, "msg": "Error: ['Checking DNS domain omnia.test., please wait
...', 'Validate installation settings ...', 'Create file system structures ...', 'Perform SELinux labeling ...',
 'Warning no default label for /dev/shm/slapd-OMNIA-TEST', 'Create database backend: dc=omnia,dc=test ...', 'Per
form post-installation tasks ...', 'CalledProcessError(Command [\\'/bin/systemctl\\', \\'start\\', \\'pki-tomcat
d@pki-tomcat.service\\'] returned non-zero exit status 1: \\'Job for pki-tomcatd@pki-tomcat.service failed becau
se a timeout was exceeded.\\\\nSee \"systemctl status pki-tomcatd@pki-tomcat.service\" and \"journalctl -xe\" fo
r details.\\\\n\\')', 'The ipa-server-install command failed. See /var/log/ipaserver-install.log for more inform
ation']"}

PLAY RECAP ******************************************************************************************************
localhost                  : ok=312  changed=57   unreachable=0    failed=1    skipped=283  rescued=1    ignored
=0
```

Potential Causes: Required repositories may not be enabled by your red hat subscription.

Resolution: Enable all required repositories via your red hat subscription.

Why would FreeIPA server/client installation fail?

Potential Cause:

The hostnames of the manager and login nodes are not set in the correct format.

Resolution:

If you have enabled the option to install the login node in the cluster, set the hostnames of the nodes in the format: *hostname.domainname*. For example, *manager.omnia.test* is a valid hostname for the login node. **Note**: To find the cause for the failure of the FreeIPA server and client installation, see *ipaserver-install.log* in the manager node or */var/log/ipaclient-install.log* in the login node.

Why does FreeIPA installation fail on the control plane when the public NIC provided is static?

Potential Cause: The network config file for the public NIC on the control plane does not define any DNS entries.

Resolution: Ensure the fields `DNS1` and `DNS2` are updated appropriately in the file `/etc/sysconfig/network-scripts/ifcfg-<NIC name>`.

What to do when JupyterHub pods are in 'ImagePullBackOff' or 'ErrImagePull' status after executing jupyterhub.yml:

Potential Cause: Your Docker pull limit has been exceeded. For more information, click here.

1. Delete Jupyterhub deployment by executing the following command in manager node: `helm delete jupyterhub -n jupyterhub`
2. Re-execute `jupyterhub.yml` after 8-9 hours.

What to do if NFS clients are unable to access the share after an NFS server reboot?

Reboot the NFS server (external to the cluster) to bring up the services again:

```
systemctl disable nfs-server
systemctl enable nfs-server
systemctl restart nfs-server
```

4.2 Frequently Asked Questions

How to add a new node for provisioning?

1. Using a mapping file:
   - Update the existing mapping file by appending the new entry (without the disrupting the older entries) or provide a new mapping file by pointing `pxe_mapping_file_path` in `provision_config.yml` to the new location.
   - Run `provision.yml`.
2. Using the switch IP:
   - Run `provision.yml` once the switch has discovered the potential new node.

Why does splitting an ethernet Z series port fail with "Failed. Either port already split with different breakout value or port is not available on ethernet switch"?

Potential Cause:

1. The port is already split.
2. It is an even-numbered port.

Resolution:

> Changing the `breakout_value` on a split port is currently not supported. Ensure the port is un-split before assigning a new `breakout_value`.

How to enable DHCP routing on Compute Nodes:

To enable routing, update the `primary_dns` and `secondary_dns` in `provision_config.yml` with the appropriate IPs (hostnames are currently not supported). For compute nodes that are not directly connected to the internet (ie only host network is configured), this configuration allows for internet connectivity.

What to do if the LC is not ready:

- Verify that the LC is in a ready state for all servers: `racadm getremoteservicesstatus`
- PXE boot the target server.

Is Disabling 2FA supported by Omnia?

- Disabling 2FA is not supported by Omnia and must be manually disabled.

Is provisioning servers using BOSS controller supported by Omnia?

Provisioning server using BOSS controller is now supported by Omnia 1.2.1.

How to re-launch services after a control-plane reboot while running provision.yml

After a reboot of the control plane while running `provision.yml`, to bring up `xcatd` services, please run the below commands:

```
systemctl restart postgresql.service

systemctl restart xcatd.service
```

How to re-provision a server once it's been set up by xCAT

- Use `lsdef -t osimage | grep install-compute` to get a list of all valid OS profiles.
- Use `nodeset all osimage=<selected OS image from previous command>` to provision the OS on the target server.
- PXE boot the target server to bring up the OS.

How many IPs are required within the PXE NIC range?

Ensure that the number of IPs available between `pxe_nic_start_range` and `pxe_nic_end_range` is double the number of iDRACs available to account for potential stale entries in the mapping DB.

What are the licenses required when deploying a cluster through Omnia?

While Omnia playbooks are licensed by Apache 2.0, Omnia deploys multiple softwares that are licensed separately by their respective developer communities. For a comprehensive list of software and their licenses, click here .

4.3 Troubleshooting Guide

4.3.1 Control plane logs

All log files can be viewed via the Dashboard tab (⊞). The Default Dashboard displays `omnia.log` and `syslog`. Custom dashboards can be created per user requirements.

Below is a list of all logs available to Loki and can be accessed on the dashboard:

Name	Location	Purpose	Additional Information
Omnia Logs	/var/log/omnia.log	Omnia Log	This log is configured by Default. This log can be used to track all changes made by all playbooks in the `omnia` directory.
Omnia Control Plane	/var/log/omnia_control_plane.log	Control plane Log	This log is configured by Default. This log can be used to track all changes made by all playbooks in the `omnia/control_plane` directory.
Omnia Telemetry	/var/log/omnia/omnia_telemetry.log	Telemetry Log	This log is configured by Default. This log can be used to track all changes made by all playbooks in the `omnia/telemetry` directory.
Omnia Tools	/var/log/omnia/omnia_tools.log	Tools Log	This log is configured by Default. This log can be used to track all changes made by all playbooks in the `omnia/tools` directory.
Omnia Platforms	/var/log/omnia/omnia_platforms.log	Platforms Log	This log is configured by Default. This log can be used to track all changes made by all playbooks in the `omnia/platforms` directory.
Omnia Control Plane Tools	/var/log/omnia/omnia_control_plane_tools.log	Control Plane tools logs	This log is configured by Default. This log can be used to track all changes made by all playbooks in the `omnia/control_plane/tools` directory.
Node Info CLI log	/var/log/omnia/collect_node_info/collect_node_info_yyyy-mm-dd-HHMMSS.log	CLI Log	This log is configured when AWX is disabled. This log can be used to track scheduled and unscheduled node inventory jobs initiated by CLI.
Device Info CLI log	/var/log/omnia/collect_device_info/collect_device_info_yyyy-mm-dd-HHMMSS.log	CLI Log	This log is configured when AWX is disabled. This log can be used to track scheduled and unscheduled device inventory jobs initiated by CLI.
iDRAC CLI log	/var/log/omnia/idrac/idrac-yyyy-mm-dd-HHMMSS.log	CLI Log	This log is configured when AWX is disabled. This log can be used to track iDRAC jobs initiated by CLI.
Infiniband CLI log	//var/log/omnia/infiniband/infiniband-yyyy-mm-dd-HHMMSS.log	CLI Log	This log is configured when AWX is disabled. This log can be used to track Infiniband jobs initiated by CLI.
Ethernet CLI log	/var/log/omnia/ethernet/ethernet-yyyy-mm-dd-HHMMSS.log	CLI Log	This log is configured when AWX is disabled. This log can be used to track Ethernet jobs initiated by CLI.
Powervault CLI log	/var/log/omnia/powervault/powervault-yyyy-mm-dd-HHMMSS.log	CLI Log	This log is configured when AWX is disabled. This log can be used to track Powervault jobs initiated by CLI.
syslogs	/var/log/messages	System Logging	This log is configured by Default
Audit Logs	/var/log/audit/audit.log	All Login Attempts	This log is configured by Default
CRON logs	/var/log/cron	CRON Job Logging	This log is configured by Default
Pods logs	/var/log/pods/ * / * / * log	k8s pods	This log is configured by Default
Access Logs	/var/log/dirsrv/slapd-<Realm Name>/access	Directory Server Utilization	This log is available when FreeIPA or 389ds is set up (ie when enable_security_support is set to 'true')
Error Log	/var/log/dirsrv/slapd-<Realm Name>/errors	Directory Server Errors	This log is available when FreeIPA or 389ds is set up (ie when enable_security_support is set to 'true')
CA Transaction Log	/var/log/pki/pki-tomcat/ca/transactions	FreeIPA PKI Transactions	This log is available when FreeIPA or 389ds is set up (ie when enable_security_support is set to 'true')
KRB5KDC	/var/log/krb5kdc.log	KDC Utilization	This log is available when FreeIPA or 389ds is set up (ie when enable_security_support is set to 'true')
Secure logs	/var/log/secure	Login Error Codes	This log is available when FreeIPA or 389ds is set up (ie when enable_security_support is set to 'true')

4.3.2 Logs of individual containers

1. A list of namespaces and their corresponding pods can be obtained using: `kubectl get pods -A`
2. Get a list of containers for the pod in question using: `kubectl get pods <pod_name> -o jsonpath='{.spec.containers[*].name}'`
3. Once you have the namespace, pod and container names, run the below command to get the required logs: `kubectl logs pod <pod_name> -n <namespace> -c <container_name>`

4.3.3 Connecting to internal databases

- **TimescaleDB**
  - Go inside the pod: `kubectl exec -it pod/timescaledb-0 -n telemetry-and-visualizations -- /bin/bash`
  - Connect to psql: `psql -U <postgres_username>`
  - Connect to database: < `timescaledb_name` >
- **MySQL DB**
  - Go inside the pod: `kubectl exec -it pod/mysqldb-n telemetry-and-visualizations -- /bin/bash`
  - Connect to psql: `psql -U <mysqldb_username> -p <mysqldb_password>`
  - Connect to database: `USE <mysqldb_name>`

4.3.4 Checking and updating encrypted parameters

1. Move to the filepath where the parameters are saved (as an example, we will be using `provision_config.yml`):

   `cd input/`

2. To view the encrypted parameters:

```
``ansible-vault view provision_config.yml --vault-password-file .provision_vault_key``
```

   3. To edit the encrypted parameters:

   `ansible-vault edit provision_config.yml --vault-password-file .provision_vault_key`

4.3.5 Checking pod status on the control plane

- Select the pod you need to troubleshoot from the output of `kubectl get pods -A`
- Check the status of the pod by running `kubectl describe pod <pod name> -n <namespace name>`

CHAPTER

FIVE

SECURITY CONFIGURATION GUIDE

5.1 Ports used by Omnia

5.1.1 Ports Used By BeeGFS

Port	Service
8008	Management service (beegfs-mgmtd)
8003	Storage service (beegfs-storage)
8004	Client service (beegfs-client)
8005	Metadata service (beegfs-meta)
8006	Helper service (beegfs-helperd)

5.1.2 Ports Used by xCAT

Port number	Protocol	Service Name
3001	tcp	xcatdport
3001	udp	xcatdport
3002	tcp	xcatiport
3002	udp	xcatiport
3003(default)	tcp	xcatlport
7	udp	echo-udp
22	tcp	ssh-tcp
22	udp	ssh-udp
873	tcp	rsync
873	udp	rsync
53	tcp	domain-tcp
53	udp	domain-udp
67	udp	bootps
67	tcp	dhcp
68	tcp	dhcpc
68	udp	bootpc
69	tcp	tftp-tcp
69	udp	tftp-udp
80	tcp	www-tcp
80	udp	www-udp
88	tcp	kerberos

continues on next page

Table 1 – continued from previous page

Port number	Protocol	Service Name
88	udp	kerberos
111	udp	sunrpc-udp
443	udp	HTTPS
443	tcp	HTTPS
514	tcp	shell
514	tcp	rsyslogd
514	udp	rsyslogd
544	tcp	kshell
657	tcp	rmc-tcp
657	udp	rmc-udp
782	tcp	conserver
1058	tcp	nim
2049	tcp	nfsd-tcp
2049	udp	nfsd-udp
4011	tcp	pxe
300	tcp	awk
623	tcp	ipmi
623	udp	ipmi
161	tcp	snmp
161	udp	snmp
162	tcp	snmptrap
162	udp	snmptrap
5432	tcp	postgresDB

Note: For more information, check out the xCAT website.

5.2 Authentication

5.2.1 FreeIPA on the NFS Node

IPA services are used to provide account management and centralized authentication. To set up IPA services for the NFS node in the target cluster, run the following command from the `utils/cluster` folder on the control plane:

```
cd utils/cluster
ansible-playbook install_ipa_client.yml -i inventory -e kerberos_admin_password="" -e␣
↪ipa_server_hostname="" -e domain_name="" -e ipa_server_ipadress=""
```

Input Parameter	Definition	Variable value
kerberos_admin_password	“admin” user password for the IPA server on RockyOS and RedHat.	The password can be found in the file `input/omnia_config.yml` .
ipa_server_hostname	The hostname of the IPA server	The hostname can be found on the manager node.
domain_name	Domain name	The domain name can be found in the file `input/omnia_config.yml`.
ipa_server_ipadress	The IP address of the IPA server	The IP address can be found on the IPA server on the manager node using the `ip a` command. This IP address should be accessible from the NFS node.

Use the format specified under NFS inventory in the Sample Files for inventory.

5.2.2 LDAP authentication

LDAP, the Lightweight Directory Access Protocol, is a mature, flexible, and well supported standards-based mechanism for interacting with directory servers.

Manager and compute nodes will have LDAP client installed and configured if `ldap_required` is set to true. The login node does not have LDAP client installed.

Warning: No users/groups will be created by Omnia.

5.2.3 Slurm job based user access

To ensure security while running jobs on the cluster, users can be assigned permissions to access compute nodes only while their jobs are running. To enable the feature:

```
cd scheduler
ansible-playbook job_based_user_access.yml -i inventory
```

Note:

- The inventory queried in the above command is to be created by the user prior to running `omnia.yml` as `scheduler.yml` is invoked by `omnia.yml`
- Only users added to the ‘slurm’ group can execute slurm jobs. To add users to the group, use the command: `usermod -a -G slurm <username>`.

CHAPTER

SIX

SAMPLE FILES

6.1 inventory file

```
[manager]
172.29.0.101

[compute]
172.29.0.103

[login_node]
172.29.0.102
```

6.2 pxe_mapping_file.csv

```
MAC,Hostname,IP

xx:yy:zz:aa:bb:cc,server,172.29.0.101

aa:bb:cc:dd:ee:ff,server2, 172.29.0.102
```

6.3 switch_inventory

```
172.19.0.101
172.19.0.102
```

6.4 powervault_inventory

```
172.19.0.105
```

6.5 NFS Server inventory file

```
[nfs_node]
172.29.0.104
```

CHAPTER

SEVEN

LIMITATIONS

- Once `provision.yml` is used to configure devices, it is recommended to avoid rebooting the control plane.
- Removal of Slurm and Kubernetes component roles are not supported. However, skip tags can be provided at the start of installation to select the component roles.
- After installing the Omnia control plane, changing the manager node is not supported. If you need to change the manager node, you must redeploy the entire cluster.
- Dell Technologies provides support to the Dell-developed modules of Omnia. All the other third-party tools deployed by Omnia are outside the support scope.
- To change the Kubernetes single node cluster to a multi-node cluster or change a multi-node cluster to a single node cluster, you must either redeploy the entire cluster or run `kubeadm reset -f` on all the nodes of the cluster. You then need to run the `omnia.yml` file and skip the installation of Slurm using the skip tags.
- In a single node cluster, the login node and Slurm functionalities are not applicable. However, Omnia installs FreeIPA Server and Slurm on the single node.
- To change the Kubernetes version from 1.16 to 1.19 or 1.19 to 1.16, you must redeploy the entire cluster.
- The Kubernetes pods will not be able to access the Internet or start when firewalld is enabled on the node. This is a limitation in Kubernetes. So, the firewalld daemon will be disabled on all the nodes as part of omnia.yml execution.
- Only one storage instance (Powervault) is currently supported in the HPC cluster.
- Cobbler web support has been discontinued from Omnia 1.2 onwards.
- Omnia supports only basic telemetry configurations. Changing data fetching time intervals for telemetry is not supported.
- Slurm cluster metrics will only be fetched from clusters configured by Omnia.
- All iDRACs must have the same username and password.
- OpenSUSE Leap 15.3 is not supported on the Control Plane.
- Slurm Telemetry is supported only on a single cluster.
- Omnia might contain some unused MACs since LOM switch have both iDRAC MACs as well as ethernet MACs, PXE NIC ranges should contain IPs that are double the iDRACs present.
- FreeIPA authentication is not supported on the control plane.

CHAPTER

EIGHT

BEST PRACTICES

- Ensure that PowerCap policy is disabled and the BIOS system profile is set to ‘Performance’ on the Control Plane.
- Ensure that there is at least 50% (~35%)free space on the Control Plane before running Omnia.
- Disable SElinux on the Control Plane.
- Use a PXE mapping file even when using DHCP configuration to ensure that IP assignments remain persistent across Control Plane reboots.
- Avoid rebooting the Control Plane as much as possible to ensure that all network configuration does not get disturbed.
- Review the prerequisites before running Omnia Scripts.
- Ensure that the firefox version being used on the control plane is the latest available. This can be achieved using `dnf update firefox -y`
- It is recommended to configure devices using Omnia playbooks for better interoperability and ease of access.

CHAPTER

NINE

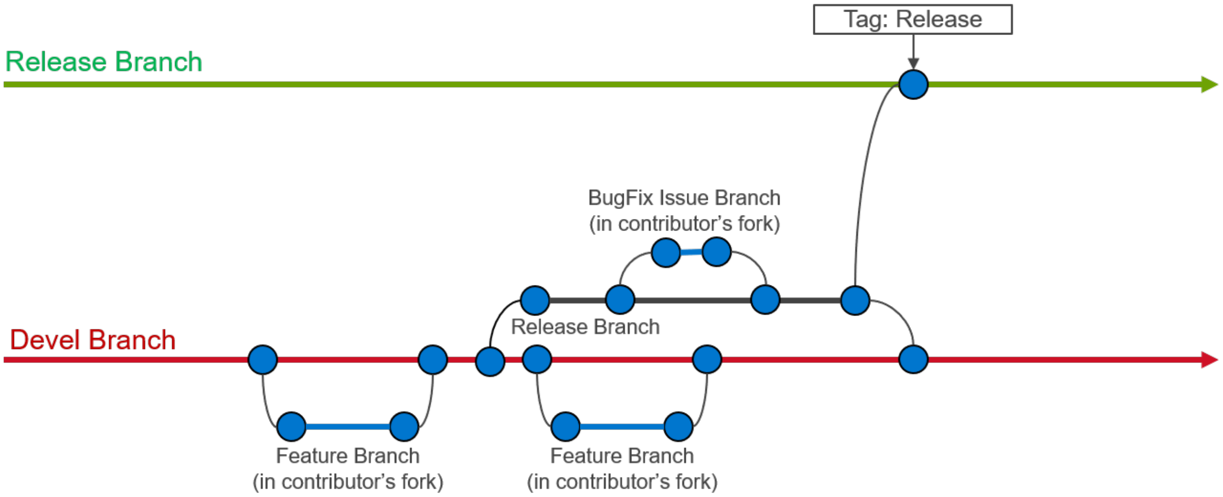
CONTRIBUTING TO OMNIA

We encourage everyone to help us improve Omnia by contributing to the project. Contributions can be as small as documentation updates or adding example use cases, to adding commenting and properly styling code segments all the way up to full feature contributions. We ask that contributors follow our established guidelines for contributing to the project.

This document will evolve as the project matures. Please be sure to regularly refer back in order to stay in-line with contribution guidelines.

9.1 Creating A Pull Request

Contributions to Omnia are made through Pull Requests (PRs). To make a pull request against Omnia, use the following steps.



9.1.1 Create an issue

Create an issue and describe what you are trying to solve. It does not matter whether it is a new feature, a bug fix, or an improvement. All pull requests must be associated to an issue. When creating an issue, be sure to use the appropriate issue template (bug fix or feature request) and complete all of the required fields. If your issue does not fit in either a bug fix or feature request, then create a blank issue and be sure to including the following information:

- **Problem description:** Describe what you believe needs to be addressed
- **Problem location:** In which file and at what line does this issue occur?
- **Suggested resolution:** How do you intend to resolve the problem?

9.1.2 Fork the repository

All work on Omnia should be done in a fork of the repository. Only maintainers are allowed to commit directly to the project repository.

9.1.3 Issue branch

Create a new branch on your fork of the repository. All contributions should be branched from devel.:

```
git checkout devel
git checkout -b <new-branch-name>
```

Branch name: The branch name should be based on the issue you are addressing. Use the following pattern to create your new branch name: `issue-xxxx`, e.g., `issue-1023`.

9.1.4 Commit changes

- It is important to commit your changes to the issue branch. Commit messages should be descriptive of the changes being made.
- All commits to Omnia need to be signed with the Developer Certificate of Origin (DCO) in order to certify that the contributor has permission to contribute the code. In order to sign commits, use either the `--signoff` or `-s` option to `git commit`:

  ```
  git commit --signoff
  git commit -s
  ```

Make sure you have your user name and e-mail set. The `--signoff | -s` option will use the configured user name and e-mail, so it is important to configure it before the first time you commit. Check the following references:

- Setting up your github user name
- Setting up your e-mail address

Warning: When preparing a pull request it is important to stay up-to-date with the project repository. We recommend that you rebase against the upstream repo frequently.

```
git pull --rebase upstream devel #upstream is dellhpc/omnia
git push --force origin <pr-branch-name> #origin is your fork of the repository (e.g.,
↪ <github_user_name>/omnia.git)
```

9.1.5 PR description

Be sure to fully describe the pull request. Ideally, your PR description will contain:

1. A description of the main point (i.e., why was this PR made?),
2. Linking text to the related issue (i.e., This PR closes issue #<issue_number>),
3. How the changes solves the problem
4. How to verify that the changes work correctly.

9.1.6 Developer Certificate of Origin

```
Developer Certificate of Origin
Version 1.1

Copyright (C) 2004, 2006 The Linux Foundation and its contributors.
1 Letterman Drive
Suite D4700
San Francisco, CA, 94129

Everyone is permitted to copy and distribute verbatim copies of this
license document, but changing it is not allowed.


Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I
    have the right to submit it under the open source license
    indicated in the file; or

(b) The contribution is based upon previous work that, to the best
    of my knowledge, is covered under an appropriate open source
    license and I have the right under that license to submit that
    work with modifications, whether created in whole or in part
    by me, under the same open source license (unless I am
    permitted to submit under a different license), as indicated
    in the file; or

(c) The contribution was provided directly to me by some other
    person who certified (a), (b) or (c) and I have not modified
    it.

(d) I understand and agree that this project and the contribution
    are public and that a record of the contribution (including all
    personal information I submit with it, including my sign-off) is
    maintained indefinitely and may be redistributed consistent with
    this project or the open source license(s) involved.
```