
Omnia

Release 1.6

dell/omnia

Apr 12, 2024

CONTENTS

1	Omnia: Overview	3
1.1	Architecture	4
1.2	New Features	6
1.3	Releases	6
1.4	Support Matrix	11
1.5	Network Topologies	17
1.6	Blogs about Omnia	19
1.7	What Omnia does	20
2	Quick Installation Guide	21
2.1	Running prereq.sh	22
2.2	Local repositories for the cluster	22
2.3	Installing the provision tool	38
2.4	Creating node inventory	58
2.5	Configuring the cluster	59
2.6	Installing AI tools	84
2.7	Adding new nodes	95
2.8	Re-provisioning the cluster	97
2.9	Configuring switches	99
2.10	Configuring PowerVault	108
2.11	Running HPC benchmarks on omnia clusters	111
2.12	Remove Slurm/K8s configuration from a node	118
2.13	Soft reset the cluster	119
2.14	Delete provisioned node	120
2.15	Uninstalling the provision tool	121
3	Features	123
3.1	Centralized authentication on the cluster	123
3.2	Shared and distributed storage deployment	130
3.3	GPU accelerator configuration	134
3.4	Additional utilities	135
3.5	Telemetry and visualizations	144
4	Logging	169
4.1	Log management	169
4.2	Control plane logs	170
4.3	Logs of individual containers	172
4.4	Provisioning logs	172
4.5	Telemetry logs	173
4.6	Grafana Loki	173

5	Troubleshooting	175
5.1	Known issues	175
5.2	Frequently asked questions	182
5.3	Troubleshooting guide	188
6	Security Configuration Guide	195
6.1	Preface	195
6.2	Security Quick Reference	197
6.3	Product and Subsystem Security	198
6.4	Authentication to external systems	200
6.5	Network security	200
6.6	Miscellaneous Configuration and Management Elements	204
7	Sample Files	207
7.1	inventory file	207
7.2	pxe_mapping_file.csv	208
7.3	switch_inventory	208
7.4	powervault_inventory	208
7.5	NFS Server inventory file	208
8	Limitations	209
9	Best Practices	211
10	Contributing To Omnia	213
10.1	Creating A Pull Request	213

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.

Omnia is an open source project hosted on [GitHub](#). Go to [GitHub](#) to view the source, open issues, ask questions, and participate in the project.

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

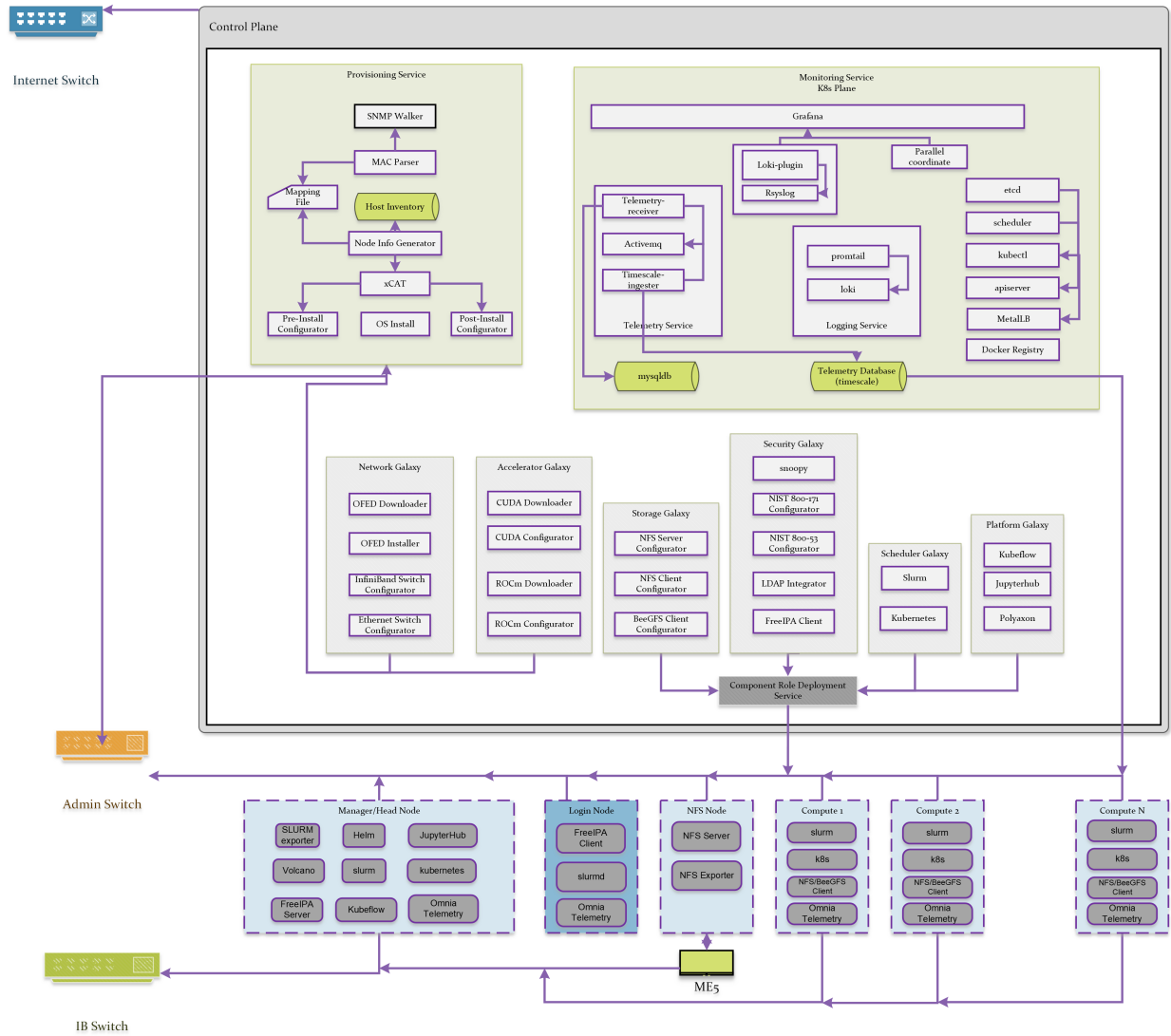


Table Of Contents

OMNIA: OVERVIEW

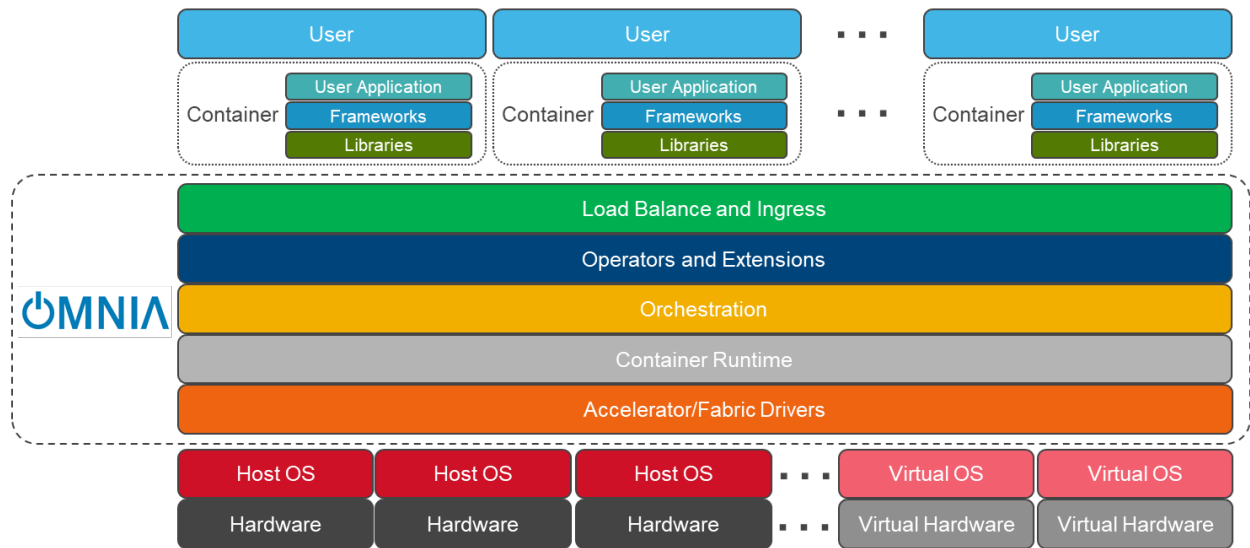
Omnia (Latin: all or everything) is a deployment tool to configure Dell PowerEdge servers running standard RPM-based Linux OS images into clusters capable of supporting HPC, AI, and data analytics workloads. It uses Slurm, Kubernetes, and other packages to manage jobs and run diverse workloads on the 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

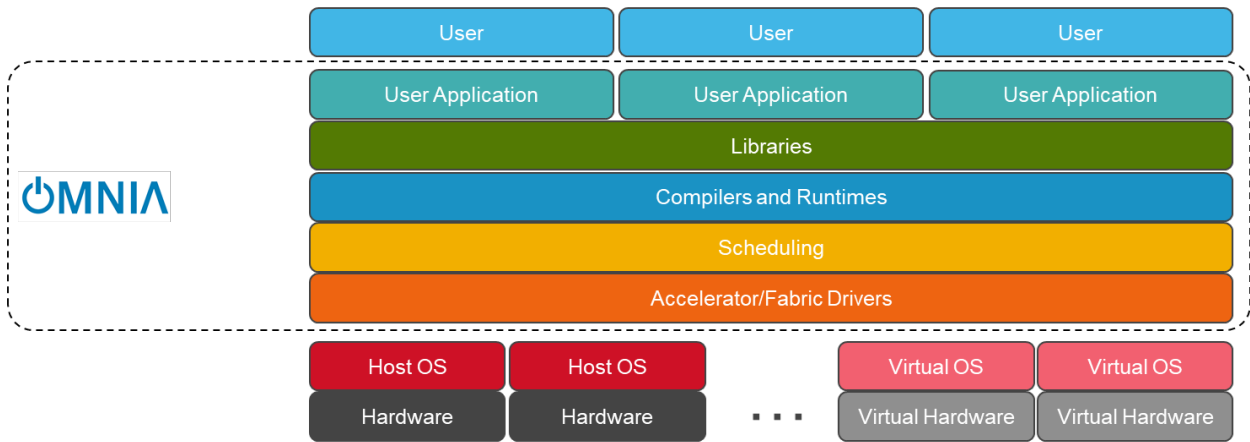


1.1.1 Omnia stack

Omnia Kubernetes stack



Omnia Slurm stack



If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

1.2 New Features

- WACO support with Ubuntu 22.04 OS:
- Local repository and registry creation for packages and container images.
- Cluster provision with Ubuntu 22.04 OS.
- AMD GPU driver and ROCm installation.
- Broadcom RoCE driver installation.
- IP configuration on the additional NICs: IPv4 support
- Kubernetes installation.
- NFS client/server configuration.
- OpenLDAP support with documented support for replication.
- AI Software Stack support including the installation of the following tools: * Jupyter notebook * Kubeflow * Kserve * Pytorch * Tensorflow * vLLM (MI210x support)
- Additional Features
- RHEL 8.8 support
- OFED Installation
- CUDA Driver installation
- Add/remove nodes to the cluster.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

1.3 Releases

1.3.1 1.6

- WACO support with Ubuntu 22.04 OS:
- Local repository and registry creation for packages and container images.
- Cluster provision with Ubuntu 22.04 OS.
- AMD GPU driver and ROCm installation.
- Broadcom RoCE driver installation.
- IP configuration on the additional NICs: IPv4 support
- Kubernetes installation.
- NFS client/server configuration.
- OpenLDAP support with documented support for replication.
- AI Software Stack support including the installation of the following tools:
 - Jupyter notebook
 - Kubeflow
 - Kserve

- Pytorch
- Tensorflow
- vLLM (MI210x support)
- Additional Features
- RHEL 8.8 support
- OFED Installation
- CUDA Driver installation
- Add / remove nodes to the cluster.

1.3.2 1.5.1

- Omnia now installs Kubernetes 1.26.

1.3.3 1.5

- **Extensive Telemetry and Monitoring** has been added to the Omnia stack, intended for consumption by customers that are using Dell systems and Omnia to provide SaaS/IaaS solutions. These include, but are not limited to:
 - CPU Utilization and status
 - GPU utilization
 - Node Count
 - Network Packet I/O
 - HDD capacity and free space
 - Memory capacity and utilization
 - Queued and Running Job Count
 - User Count
 - Cluster HW Health Checks (PCIE, NVLINK, BMC, Temps)
 - Cluster SW Health Checks (dmesg, BeeGFS, k8s nodes/pods, MySQL on control plane)
- Metrics are extracted using a combination of the following: PSUtil, Smartctl, beegfs-ctl, nvidia-smi, rocm-smi. Since groundwork is already laid, additional requests from these tools will be quicker to implement in the future.
- Telemetry and health checks can be optionally disabled.
- **Log Aggregation** via xCAT syslog:
 - Aggregated on control plane, grouping default is “severity” with others available.
 - Uses Grafana-Loki for viewing.
- Docker Registry Creation.
- Integration of aptainer for **containerized HPC benchmark execution**.
- Hardware Support: Intel E810 NIC, ConnectX-5/6 NICs.
- Omnia github now hosts a “genesis” image with this functionality baked in for initial bootup.
- Host aliasing for Scheduler and IPA authentication.

- Login and kube_control_plane access from both public and private NIC.
- Validation check enhancements:
- Rearranged to occur as early as possible.
- Isolate checks when running smaller playbooks.
- Added a [Benchmark Install Guide](#): OneAPI for Intel, MPI AOCC HPL for AMD.

1.3.4 1.4.3

- XE 9640, R760 XA, R760 XD2 are now supported as control planes or target nodes with Nvidia H100 accelerators.
- Added ability for split port configuration on NVIDIA Quantum-2-based QM9700 (Nvidia InfiniBand NDR400 switches).
- Extended password-less SSH support for multiple user configuration in a single execution.
- Input mapping files and inventory files now support commented entries for customized playbook execution.
- NFS share is now available for hosting user home directories within the cluster.

1.3.5 1.4.2

- XE9680, R760, R7625, R6615, R7615 are now supported as control planes or target nodes.
- Added ability for switch-based discovery of remote servers and PXE provisioning.
- Active RedHat subscription is no longer required on the control plane and the cluster nodes. Users can configure and use local RHEL repositories.
- IP ranges can be defined for assignment to remote nodes when discovered via the switch.

1.3.6 1.4.1

- R660, R6625 and C6620 platforms are now supported as control planes or target nodes.
- One touch provisioning now allows for OFED installation, NVIDIA CUDA-toolkit installation along with iDRAC and InfiniBand IP configuration on target nodes.
- Potential servers can now be discovered via iDRAC.
- Servers can be provisioned automatically without manual intervention for booting/PXE settings.
- Target node provisioning status can now be checked on the control plane by viewing the OmniaDB.
- Omnia clusters can be configured with password-less SSH for seamless execution of HPC jobs run by non-root users.
- Accelerator drivers can be installed on Rocky target nodes in addition to RHEL.

1.3.7 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, and FreeIPA installation in all cluster nodes
- Infiniband switch configuration and split port functionality
- Added support for Ethernet Z series switches.

1.3.8 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.

1.3.9 1.2.2

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

1.3.10 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

1.3.11 1.2.0.1

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

1.3.12 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
- cluster 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 cluster nodes with a single NIC.
- All Cluster metrics can be viewed using Grafana on the Control plane (as opposed to checking the kube_control_plane 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.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

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
16G	C6620 R660 R6625 R760 XE8640 R760xa ¹ R760xd2 XE9680

Note: Since Cloud Enclosures only support shared LOM connectivity, it is recommended that BMC or Switch-based methods of discovery are used.

AMD servers

Server Type	Server Model
14G	R6415 R7415 R7425
15G	R6515 R6525 R7515 R7525 C6525
16G	R6625 R7625 R7615 R6615

New in version 1.2: 15G servers

New in version 1.3: AMD servers

New in version 1.4.1: Intel 16G servers

¹ The R760xa supports both H100 and A100.

New in version 1.4.3: Intel: R760 XE8640 R760xa R760xd2 XE9680; AMD 16G servers

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

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

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

Switches

Switch Type	Switch Model
Mellanox InfiniBand Switches	NVIDIA MQM8700-HS2F Quantum HDR InfiniBand Switch 40 QSFP56 NVIDIA QUANTUM-2 QM9700

Switch Type	Switch Model
Dell Networking Switches	PowerSwitch S3048-ON PowerSwitch S5232F-ON PowerSwitch Z9264F-ON PowerSwitch N3248TE-ON PowerSwitch S4148

Note:

- The switches that have reached EOL might not function properly. It is recommended by Omnia to use 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.
-

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

1.4.2 Operating Systems

Red Hat Enterprise Linux

OS Version	Control Plane	Cluster Nodes
8.6	Yes	Yes
8.7 ¹	Yes	Yes
8.8	Yes	Yes

Note: Always deploy the DVD Edition of the OS on cluster nodes to access offline repos.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

Rocky

Caution: THE ROCKY LINUX OS VERSION ON THE CLUSTER WILL BE UPGRADED TO THE LATEST 8.x VERSION AVAILABLE IRRESPECTIVE OF THE PROVISION_OS_VERSION PROVIDED IN PROVISION_CONFIG.YML.

OS Version	Control Plane	Cluster Nodes
8.6	Yes	No
8.7 ¹	Yes	No
8.8	Yes	Yes

Note:

- Always deploy the DVD (Full) Edition of the OS on cluster nodes.
 - AMD ROCm driver installation is not supported by Omnia on Rocky cluster nodes.
-

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

Ubuntu

OS Version	Control Plane	Cluster Nodes
20.04 ¹	Yes	Yes
22.04	Yes	Yes

Note:

-
- ¹ This version of RHEL does not support vLLM installation via Omnia.
¹ This version of Rocky does not support vLLM installation via Omnia.
¹ This version of Ubuntu does not support vLLM and racadm installation via Omnia.

- Only the live-server version of Ubuntu for provisioning via Omnia.
 - Ubuntu does not support the use of Slurm as a clustering software. As a result, benchmarking software and FreeIPA is not supported on Ubuntu.
 - Ubuntu does not support the use of powervault storage.
-

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

1.4.3 Testing matrix

Hosts

Generation	Server Model
14G	PowerEdge C6420
14G	PowerEdge R640
14G	PowerEdge R740xd
14G	PowerEdge R740xc
14G	PowerEdge R540
14G	PowerEdge R440
15G	PowerEdge R550
15G	PowerEdge R650
15G	PowerEdge R450
15G	PowerEdge R6525
15G	PowerEdge R750xa
15G	PowerEdge R6515
15G	PowerEdge R7525
16G	PowerEdge R760
16G	PowerEdge R760xa
16G	PowerEdge R6615
16G	PowerEdge R6625
16G	PowerEdge R7615
16G	PowerEdge R7625
16G	PowerEdge C6620
16G	PowerEdge R660
16G	PowerEdge R760
16G	PowerEdge XE9640 ¹
16G	PowerEdge XE9680 ¹

¹ This model was tested using Omnia 1.4.3.

NICs

NIC
Intel® Ethernet 10G 4P X710/I350 rNDC
Intel® Ethernet Converged Network Adapter X710
Mellanox ConnectX-5 Single Port 100 GbE QSFP+
Mellanox ConnectX-5 Single Port 0 GbE QSFP
I350GbE Controller
Broadcom Adv Dual 25Gb Ethernet
Mellanox ConnectX-6 Single Port VPI HDR QSFP
Mellanox ConnectX-5 Single Port 56 GbE QSFP+
Mellanox ConnectX-6 Single Port VPI HDR 100 QSFP
Broadcom Gigabit Ethernet BCM5720
Broadcom Adv Dual 10GBASE-t Ethernet
Mellanox Network Adapter (10 Gb)
Mellanox ConnectX-5 Ex 100 GbE QSFP
Intel® Ethernet 100GbE Network Adapter E810

GPUs

GPU
Nvidia - T4
AMD - MI200

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

1.4.4 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 ComponentCon
kubeadm	Apache-2.0	“fast paths” for creating Kubernetes clusters
kubectrl	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 framework
MPI	3Clause BSD License	HPC library
spark	Apache-2.0	

OSS Title	License Name/Version #	Description
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 an
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 nod
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 and mo
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 program
mysql	GPL 2.0	MySQL is an open-source relational databas
postgresSQL	PostgreSQL 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
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 progr
snoopy	GPL 2.0	Snoopy is a small library that logs all progr
timescaledb	Apache-2.0	TimescaleDB is a time-series SQL database
Beegfs-Client	GPLv2	BeeGFS is a high-performance parallel file s
redhat subscription	Apache-2.0	Red Hat Subscription Management (RHSM)
Lmod	MIT License	Lmod is a Lua based module system that eas
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 di
CUDA Toolkit	NVIDIA License	The NVIDIA® CUDA® Toolkit provides a
MLNX-OFED	BSD License	MLNX_OFED is an NVIDIA tested and pac
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
psycpg2-binary	GNU Lesser General Public License	Psycpg is the most popular PostgreSQL da

OSS Title	License Name/Version #	Description
python.requests	Apache-2.0	Makes HTTP requests simpler and more human-friendly

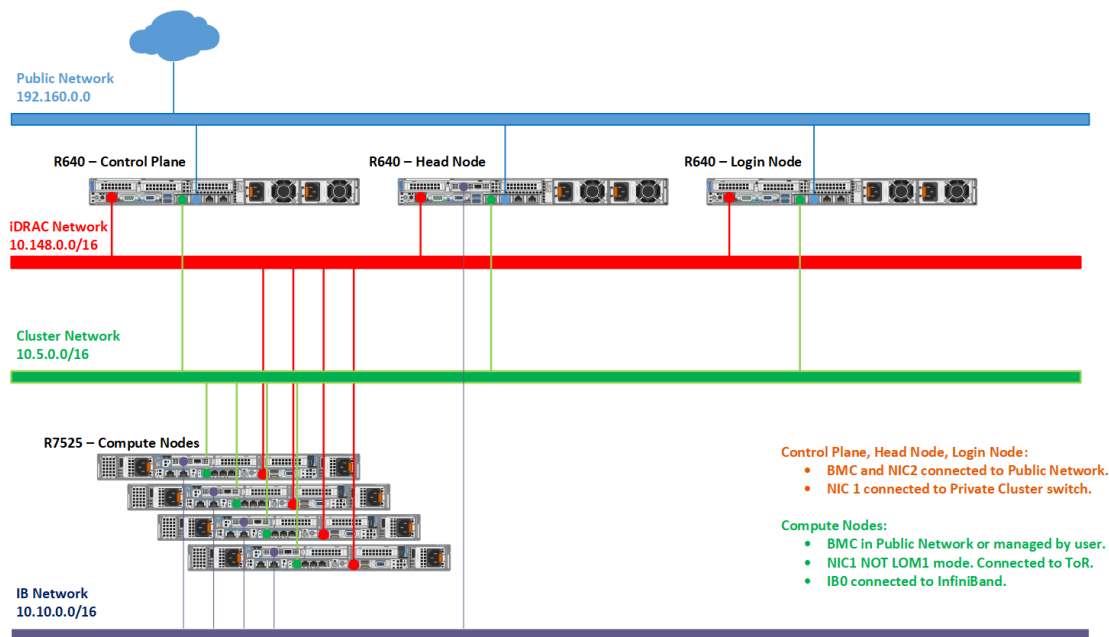
If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

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

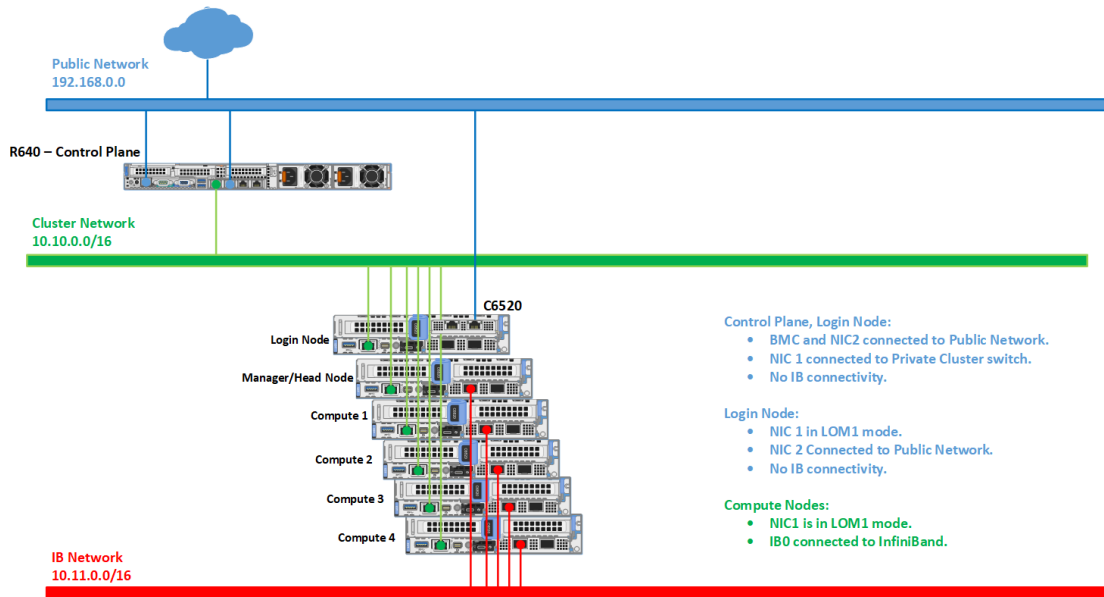
When all compute nodes rely on the control plane for public network access, the variables `primary_dns` and `secondary_dns` in `provision_config.yml` are used to indicate that the control plane is the gateway for all compute nodes to get internet access. Since all public network traffic will be routed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

- [mapping](#)

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

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.



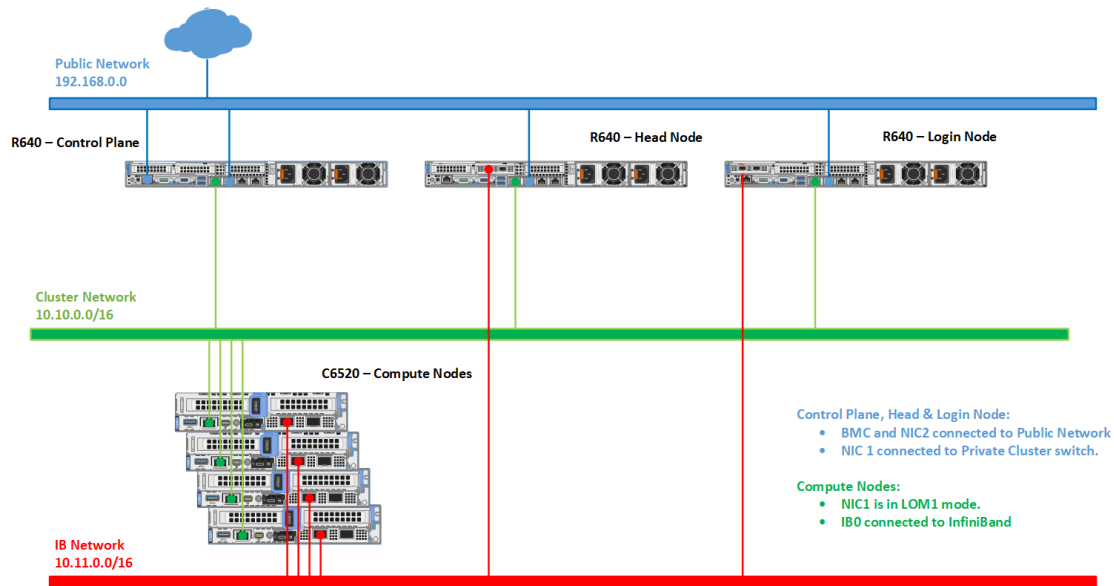
Recommended discovery mechanism

- mapping
- bmc
- switch-based

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

1.5.3 Network Topology: Hybrid setup

For an environment containing both LOM and BMC ports, the provision tool needs to be run twice to correctly manage all servers in the network.



The first time the provision tool is run (to discover the dedicated BMC ports), ensure that the following variables are set in `input/provision_config.yml`:

- `network_interface_type`: dedicated
- `discovery_mechanism`: mapping
- Leave the variables `bmc_nic_subnet`, `bmc_static_start_range` and `bmc_static_end_range` blank in `input/provision_config.yml`. Entering these variables will cause IP reassignment and can interfere with the availability of ports on your target servers.
- Do not use the `switch_based` methods to discover nodes in a Hybrid setup.

Once all the dedicated NICs are discovered, re-run the provisioning tool (to discover the shared LOM ports) with the following variables in `input/provision_config.yml`:

- `network_interface_type`: lom
- `discovery_mechanism`: bmc

To assign BMC NICs and route internet access to your target nodes, populate the values of `bmc_nic_subnet`, `bmc_static_start_range`, and `bmc_static_end_range` in `input/provision_config.yml` during the second run of the provision tool.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

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

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

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, Rocky, and Ubuntu. For a complete list of versions supported, check out the [Support Matrix](#) .

For RHEL and Rocky installations:

```
dnf install git -y
```

For Ubuntu installations:

```
apt install git -y
```

Note: Optionally, if the control plane has an Infiniband NIC installed on RHEL or Rocky, run the below command:

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

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

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

Change directory to Omnia using:

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

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

Note: The permissions on the Omnia directory are set to **0755** by default. Do not change these values.

2.1 Running prereq.sh

`prereq.sh` is used to install the software utilized by Omnia on the control plane including Python (3.9), Ansible (2.14).

```
cd omnia
./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.
 - The file `input/software_config.json` is overwritten with the default value (based on the operating system) when `prereq.sh` is executed.
-

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.2 Local repositories for the cluster

The local repository feature will help create offline repositories on the control plane which all the cluster nodes will access. `local_repo/local_repo.yml` runs with inputs from `input/software_config.json` and `input/local_repo_config.yml`:

2.2.1 Before you create local repositories

Space considerations

If all available software stacks are configured, the free space required on the control plane is as below:

- For packages: 30GB
- For images (in `/var`): 400GB
- For storing repositories (the file path should be specified in `repo_store_path` in `input/local_repo_config.yml`): 30GB.

On Ubuntu clusters

For persistent offline local repositories, (If the parameter `repo_config` in `input/software_config` is set to `always`), click [here](#) to set up the required repositories.

Note: This link explains how to build a mirror on an Ubuntu 20.04 server. Adapt the steps and scripts as required for any other version of Ubuntu.

When creating user registries

To avoid docker pull limits, provide docker credentials (`docker_username`, `docker_password`) in `input/provision_config_credentials.yml`.

Images listed in `user_registry` in `input/local_repo_config.yml` are accessed from user defined registries. To ensure that the control plane can correctly access the registry, ensure that the following naming convention is used to save the image:

```
<host>/<image name>:v<version number>
```

Therefore, for the image of calico/cni version 1.2 available on quay.io that has been pulled to a local host: server1.omnia.test, the accepted user registry name is:

```
server1.omnia.test:5001/calico/cni:v1.2
```

Omnia will not be able to configure access to any registries that do not follow this naming convention. Do not include any other extraneous information in the registry name.

There are two ways to pull images from the user registries in the form of a digest:

- Update the digest value to the listed image in the registry. All images to be pulled are listed in input/config/<os>/<version>/<software_file>.json. A sample of the listing is shown below:

```
{
  "package": "gcr.io/knative-releases/knative.dev/serving/cmd/webhook",
  "digest": ".1305209ce498caf783f39c8f3e85df..35ece6947033bf50b0b627983fd65953",
  "type": "image"
},
```

- While pushing the image to the user registry, create a tag and update the JSON file to take the tag value instead of the digest.

Note:

- Enable a repository from your RHEL subscription, run the following commands:

```
subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
```

- Enable an offline repository by creating a .repo file in /etc/yum.repos.d/. Refer the below sample content:

```
[RHEL-8-appstream]

name=Red Hat AppStream repo

baseurl=http://xx.yy.zz/pub/Distros/RedHat/RHEL8/8.6/AppStream/x86_64/os/

enabled=1

gpgcheck=0

[RHEL-8-baseos]

name=Red Hat BaseOS repo

baseurl=http://xx.yy.zz/pub/Distros/RedHat/RHEL8/8.6/BaseOS/x86_64/os/

enabled=1

gpgcheck=0
```

- Verify your changes by running:

```
yum repolist enabled
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered with an entitlement server. You can use subscription-
↪manager to register.
  repo id                                repo name
  RHEL-8-appstream-partners              Red Hat↵
↪Enterprise Linux 8.6.0 Partners (AppStream)
  RHEL-8-baseos-partners                 Red Hat↵
↪Enterprise Linux 8.6.0 Partners (BaseOS)
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.2.2 Input parameters for Local Repositories

- Input all required values in `input/software_config.json`.

Table 1: Parameters for Software Configuration

Parameter	Details
cluster_os_type string Required	<ul style="list-style-type: none">• The operating system running on the cluster (rhel, rocky, and ubuntu). Default value: ubuntu.
cluster_os_version string Required	<ul style="list-style-type: none">• The OS Version that will be provisioned on compute nodes.• For RHEL, the accepted values are 8.6, 8.7, and 8.8.• For Rocky, the accepted values are 8.6, 8.7, and 8.8.• For Ubuntu, the accepted values are 20.04, 22.04.• Default value: 22.04

continues on next page

Table 1 – continued from previous page

Parameter	Details
repo_config string Required	<ul style="list-style-type: none"> • The type of offline configuration user needs. • When the value is set to <code>always</code>, Omnia creates a local repository/registry on the Control plane hosting all the packages/images required for the cluster. • When the value is set to <code>partial</code>, Omnia creates a local repository/registry on the Control plane hosting all the packages/images except those listed in the <code>user_registry</code> in <code>input/local_repo_config.yml</code>. • When the value is set to <code>never</code>, Omnia does not create a local repository/registry. All the packages/images are directly downloaded on the cluster. <hr/> <p>Note:</p> <ul style="list-style-type: none"> • After <code>local_repo.yml</code> has run, the value of <code>repo_config</code> in <code>input/software_config.json</code> cannot be updated without running the <code>control_plane_cleanup.yml</code> script first. • Irrespective of the value of <code>repo_config</code>, all local repositories that are not available as images, debian packages, or RPMs will be downloaded and configured locally on the control plane. Additionally, AMD GPU drivers, ROCm drivers, CUDA, and OFED are downloaded by default. <hr/> <ul style="list-style-type: none"> • Accepted values: <ul style="list-style-type: none"> – <code>always</code> – <code>partial</code> <- Default – <code>never</code>
softwares JSON list Required	<ul style="list-style-type: none"> • A JSON list of required software and (optionally) the software revision. • The following software should be listed with a version in the list: BeeGFS, AMD GPU, Kubernetes, CUDA, OFED, BCM RoCE, UCX, and ROCm. • A minimum of one software should be provided in the list for <code>local_repo.yml</code> to execute correctly. <hr/> <p>Note: The accepted names for software is taken from <code>input/config/<cluster_os_type>/<cluster_os_version>/</code>.</p>

Below is a sample version of the file:

```
{
  "cluster_os_type": "ubuntu",
  "cluster_os_version": "22.04",
  "repo_config": "partial",
  "softwares": [
    {"name": "k8s", "version": "1.26.12"},
    {"name": "jupyter"},
    {"name": "openldap"},
    {"name": "kubeflow"},
    {"name": "beegfs", "version": "7.4.2"},
    {"name": "nfs"},
    {"name": "kserve"},
    {"name": "amdgpu", "version": "6.0"},
    {"name": "cuda", "version": "12.3.2"},
    {"name": "ofed", "version": "24.01-0.3.3.1"},
    {"name": "vllm"},
    {"name": "pytorch"},
    {"name": "tensorflow"},
    {"name": "bcm_roce", "version": "229.2.9.0"}
  ],
  "kserve": [
    {"name": "istio"},
    {"name": "cert_manager"},
    {"name": "knative"}
  ],
  "amdgpu": [
    {"name": "rocm", "version": "6.0" }
  ],
  "vllm": [
    {"name": "vllm_amd"},
    {"name": "vllm_nvidia"}
  ],
  "pytorch": [
    {"name": "pytorch_cpu"},
    {"name": "pytorch_amd"},
    {"name": "pytorch_nvidia"}
  ],
  "tensorflow": [
    {"name": "tensorflow_cpu"},
    {"name": "tensorflow_amd"},
    {"name": "tensorflow_nvidia"}
  ]
}
```

For a list of accepted values in softwares, go to `input/config/<operating_system>/<operating_system_version>` and view the list of JSON files available. The filenames present in this location (without the `*.json` extension) are a list of accepted software names. The repositories to be downloaded for each software are listed the corresponding JSON file. For example: For a cluster running Ubuntu 22.04, go to `input/config/ubuntu/22.04/` and view the file list:

```
amdgpu.json
bcm_roce.json
beegfs.json
cuda.json
jupyter.json
k8s.json
kserve.json
kubeflow.json
nfs.json
ofed.json
openldap.json
pytorch.json
tensorflow.json
vllm.json
```

For a list of repositories (and their types) configured for `amdgpu`, view the `amdgpu.json` file:

```
{
  "amdgpu": {
    "cluster": [
      {"package": "linux-headers-$(uname -r)", "type": "deb", "repo_name": "jammy"},
      {"package": "linux-modules-extra-$(uname -r)", "type": "deb", "repo_name": "jammy"},
      {"package": "amdgpu-dkms", "type": "deb", "repo_name": "amdgpu"}
    ],
    "rocm": {
      "cluster": [
        {"package": "rocm-hip-sdk{{ rocm_version }}*", "type": "deb", "repo_name": "rocm"}
      ]
    }
  }
}
```

Note: To configure a locally available repository that does not have a pre-defined json file, [click here](#).

- Input the required values in `input/local_repo_config.yml`.

Table 2: Parameters for Local Repository Configuration

Parameter	Details
repo_store_path string Required	<ul style="list-style-type: none"> The intended file path for offline repository data. Ensure the disk partition has enough space. Default value: "/omnia_repo"
user_repo_url JSON List Optional	<ul style="list-style-type: none"> This variable accepts the repository urls of the user which contains the packages required for the cluster. When <code>repo_config</code> is always, the given list will be configured on the control plane and packages required for cluster will be downloaded into a local repository. When <code>repo_config</code> is partial, a local repository is created on the control plane containing packages that are not part of the user's repository. When <code>repo_config</code> is never, no local repository is created and packages are downloaded on all cluster nodes. 'url' defines the baseurl for the repository. 'gpgkey' defines gpgkey for the repository. If 'gpgkey' is omitted then gpgcheck=0 is set for that repository. Sample value: - {url: "http://crb.com/CRB/x86_64/os/", gpgkey: "http://crb.com/CRB/x86_64/os/RPM-GPG-KEY"}
user_registry JSON List Optional	<ul style="list-style-type: none"> This variable accepts the registry url along with port of the user which contains the images required for cluster. When <code>repo_config</code> is always, the list given in <code>user_registry</code> will be configured on the control plane and packages required for cluster will be downloaded into a local repository. If the same repository is available in both the <code>user_repo_url</code> and the <code>user_registry</code>, the repository will be configured using the values in <code>user_registry</code>. When <code>repo_config</code> is partial, a local registry is created on the control plane containing packages that are not part of the <code>user_registry</code>. Images listed in <code>user_registry</code> are directly configured as a mirror on compute nodes. Compute nodes are expected to connect to the URLs in the <code>user_registry</code> via <code>http_proxy</code>. When <code>repo_config</code> is never, no local registry is created and packages/images are downloaded on all cluster nodes. 'host' defines the URL and path to the registry. 'cert_path' defines the absolute path where the security certificates for each registry. If this path is not provided, insecure registries are configured. Sample value:

- Input `docker_username` and `docker_password` in `input/provision_config_credentials.yml` to avoid image pullback errors.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.2.3 Configuring specific local repositories

AMDGPU ROCm

To install ROCm, include the following line under `softwares`:

```
"amdgpu": [
  {"name": "rocm", "version": "6.0" }
]
```

BCM RoCE

To install RoCE, include the following line under `softwares`:

```
{"name": "bcm_roce", "version": "229.2.9.0"}
```

For a list of repositories (and their types) configured for RoCE, view the `input/config/ubuntu/<operating_system_version>/bcm_roce.json` file. To customize your RoCE installation, update the file. URLs for different versions can be found [here](#):

```
{
  "bcm_roce": {
    "cluster": [
      {
        "package": "bcm_roce_driver-{{ bcm_roce_version }}",
        "type": "tarball",
        "url": "",
        "path": ""
      }
    ]
  }
}
```

Note:

- The RoCE driver is only supported on Ubuntu clusters.
 - The only accepted URL for the RoCE driver is from the Dell Driver website.
-

BeeGFS

To install BeeGFS, include the following line under `softwares`:

```
{"name": "beegfs"},
```

For information on deploying BeeGFS after setting up the cluster, [click here](#).

CUDA

To install CUDA, include the following line under `softwares`:

```
{"name": "cuda", "version": "12.3.2"},
```

For a list of repositories (and their types) configured for CUDA, view the `input/config/<operating_system>/<operating_system_version>/cuda.json` file. To customize your CUDA installation, update the file. URLs for different versions can be found [here](#):

For Ubuntu:

```
{
  "cuda": {
    "cluster": [
      { "package": "cuda",
        "type": "iso",
        "url": "https://developer.download.nvidia.com/compute/cuda/12.3.2/
→local_installers/cuda-repo-ubuntu2204-12-3-local_12.3.2-545.23.08-1_amd64.deb
→",
        "path": ""
      }
    ]
  }
}
```

For RHEL or Rocky:

```
{
  "cuda": {
    "cluster": [
      { "package": "cuda",
        "type": "iso",
        "url": "https://developer.download.nvidia.com/compute/cuda/12.3.2/
→local_installers/cuda-repo-rhel8-12-3-local-12.3.2_545.23.08-1.x86_64.rpm",
        "path": ""
      },
      { "package": "dkms",
        "type": "rpm",
        "repo_name": "epel"
      }
    ]
  }
}
```

- If the package version is customized, ensure that the version value is updated in `software_config.json`.
- If the target cluster runs on RHEL or Rocky, ensure the “dkms” package is included in `input/config/<operating system>/8.x/cuda.json` as illustrated above.

Custom repositories

Include the following line under `softwares`:

```
{"name": "custom"},
```

Create a `custom.json` file in the following directory: `input/config/<operating_system>/<operating_system_version>` to define the repositories. For example, For a cluster running RHEL 8.8, go to `input/config/rhel/8.8/` and create the file there. The file is a JSON list consisting of the

package name, repository type, URL (optional), and version (optional). Below is a sample version of the file:

```
{
  "custom": {
    "cluster": [
      {
        "package": "ansible==5.3.2",
        "type": "pip_module"
      },
      {
        "package": "docker-ce-24.0.4",
        "type": "rpm",
        "repo_name": "docker-ce-repo"
      },
      {
        "package": "gcc",
        "type": "rpm",
        "repo_name": "appstream"
      },
      {
        "package": "community.general",
        "type": "ansible_galaxy_collection",
        "version": "4.4.0"
      },
      {
        "package": "perl-Switch",
        "type": "rpm",
        "repo_name": "codeready-builder"
      },
      {
        "package": "prometheus-slurm-exporter",
        "type": "git",
        "url": "https://github.com/vpenso/prometheus-slurm-exporter.git",
        "version": "master"
      },
      {
        "package": "ansible.utils",
        "type": "ansible_galaxy_collection",
        "version": "2.5.2"
      },
      {
        "package": "prometheus-2.23.0.linux-amd64",
        "type": "tarball",
        "url": "https://github.com/prometheus/prometheus/releases/download/v2.
↪23.0/prometheus-2.23.0.linux-amd64.tar.gz"
      },
      {
        "package": "metallb-native",
        "type": "manifest",
        "url": "https://raw.githubusercontent.com/metallb/metallb/v0.13.4/
```

(continues on next page)

(continued from previous page)

```
↪config/manifests/metallb-native.yaml"
  },
  {
    "package": "registry.k8s.io/pause",
    "version": "3.9",
    "type": "image"
  }
]
}
```

FreeIPA

To install FreeIPA, include the following line under **softwares**:

```
{"name": "freeipa"},
```

For information on deploying FreeIPA after setting up the cluster, [click here](#).

Jupyterhub

To install Jupyterhub, include the following line under **softwares**:

```
{"name": "jupyter"},
```

For information on deploying Jupyterhub after setting up the cluster, [click here](#).

Kserve

To install Kserve, include the following line under **softwares**:

```
"kserve": [
  {"name": "istio"},
  {"name": "cert_manager"},
  {"name": "knative"}
]
```

For information on deploying Kserve after setting up the cluster, [click here](#).

Kubeflow

To install kubeflow, include the following line under **softwares**:

```
{"name": "kubeflow"},
```

For information on deploying kubeflow after setting up the cluster, [click here](#).

Kubernetes

To install Kubernetes, include the following line under **softwares**:

```
{"name": "k8s", "version": "1.26.12"},
```

Note: The version of the software provided above is the only version of the software Omnia supports.

OFED

To install OFED, include the following line under `softwares`:

```
{"name": "ofed", "version": "24.01-0.3.3.1"},
```

For a list of repositories (and their types) configured for OFED, view the `input/config/<operating_system>/<operating_system_version>/ofed.json` file. To customize your OFED installation, update the file.:

For Ubuntu:

```
{
  "ofed": {
    "cluster": [
      { "package": "ofed",
        "type": "iso",
        "url": "https://content.mellanox.com/ofed/MLNX_OFED-24.01-0.3.3.1/
↪MLNX_OFED_LINUX-24.01-0.3.3.1-ubuntu20.04-x86_64.iso",
        "path": ""
      }
    ]
  }
}
```

For RHEL or Rocky:

```
{
  "ofed": {
    "cluster": [
      { "package": "ofed",
        "type": "iso",
        "url": "https://content.mellanox.com/ofed/MLNX_OFED-24.01-0.3.3.1/MLNX_
↪OFED_LINUX-24.01-0.3.3.1-rhel8.7-x86_64.iso",
        "path": ""
      }
    ]
  }
}
```

Note: If the package version is customized, ensure that the `version` value is updated in `software_config.json`.

OpenLDAP

To install OpenLDAP, include the following line under `softwares`:

```
{"name": "openldap"},
```

Features that are part of the OpenLDAP repository are enabled by running `security.yml`

OpenMPI

To install OpenMPI, include the following line under `softwares`:

```
{"name": "openmpi", "version": "4.1.6"},
```

OpenMPI is deployed on the cluster when the above configurations are complete and `omnia.yml` is run.

Pytorch

To install PyTorch, include the following line under `softwares`:

```
{ "name": "pytorch" },

"pytorch": [
  { "name": "pytorch_cpu" },
  { "name": "pytorch_amd" },
  { "name": "pytorch_nvidia" }
],
```

For information on deploying Pytorch after setting up the cluster, [click here](#).

Secure Login Node

To secure the login node, include the following line under `softwares`:

```
{ "name": "secure_login_node" },
```

Features that are part of the `secure_login_node` repository are enabled by running `security.yml`

TensorFlow

To install TensorFlow, include the following line under `softwares`:

```
{ "name": "tensorflow" },

"tensorflow": [
  { "name": "tensorflow_cpu" },
  { "name": "tensorflow_amd" },
  { "name": "tensorflow_nvidia" }
]
```

For information on deploying TensorFlow after setting up the cluster, [click here](#).

Unified Communication X

To install UCX, include the following line under `softwares`:

```
{ "name": "ucx", "version": "1.15.0" },
```

UCX is deployed on the cluster when the `local_repo.yml` is run then `omnia.yml` is run.

vLLM

To install vLLM, include the following line under `softwares`:

```
{ "name": "vLLM" },

"vllm": [
  { "name": "vllm_amd" },
  { "name": "vllm_nvidia" }
],
```

For information on deploying vLLM after setting up the cluster, [click here](#).

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.2.4 Running local repo

The local repository feature will help create offline repositories on the control plane which all the cluster nodes will access.

Configurations made by the playbook

- A registry is created on the control plane at <Control Plane hostname>:5001.
- If `repo_config` in `local_repo_config.yml` is set to `always` or `partial`, all images present in the `input/config/<operating system>/<version>` folder will be downloaded to the control plane.
 - If the image is defined using a tag, the image will be tagged using `<control plane host-name>:5001/<image_name>:<version>` and pushed to the Omnia local registry.
 - If the image is defined using a digest, the image will be tagged using `<control plane host-name>:5001/<image_name>:omnia` and pushed to the Omnia local registry.
- When `repo_config` in `local_repo_config.yml` is set to `always`, the control plane is set as the default registry mirror.
- When `repo_config` in `local_repo_config` is set to `partial`, the `user_registry` (if defined) and the control plane are set as default registry mirrors.

To create local repositories, run the following commands:

```
cd local_repo
ansible-playbook local_repo.yml
```

Verify changes made by the playbook by running `cat /etc/containerd/certs.d/_default/hosts.toml` on compute nodes.

Note:

- View the status of packages for the current run of `local_repo.yml` in `/opt/omnia/offline/download_package_status.csv`.
- If any software packages failed to download during the execution of this script, scripts that rely on the package for their working (that is, scripts that install the software) may fail.

To fetch images from the `user_registry` or the Omnia local registry, run the below commands:

- Images defined with versions: `nerdctl pull <global_registry>/<image_name>:<tag>`
- Images defined with digests: `nerdctl pull <global_registry>/<image_name>:omnia`

Note:

- After `local_repo.yml` has run, the value of `repo_config` in `input/software_config.json` cannot be updated without running the `control_plane_cleanup.yml` script first.
- To configure additional local repositories after running `local_repo.yml`, update `software_config.json` and re-run `local_repo.yml`.
- For images coming from `gcr.io`, digests are defined as tags are not available. Omnia gives a custom tag of 'omnia' to these images. If such images need to be taken from the `user_registry`, use one of the below steps:
 - Append 'omnia' to the end of the image name while pushing images to the `user_registry`. Update the image definition in `input/config/<operating system>/<version>/<software>.json` to follow the same nomenclature.

- If a different tag is provided, update the digest value in `input/config/<operating system>/<version>/<software>.json` as per the image digest in the `user_directory`. To get the updated digest from the `user_registry`, use the below steps:
 - * Check the tag of image: `curl -k https://<user_registry>/v2/<image_name>/tags/list`
 - * Check the digest of the tag: `curl -H <headers> -k https://<user_registry>/v2/<image_name>/manifests/omnia`
-

Update local repositories

This playbook updates all local repositories configured on a provisioned cluster after local repositories have been configured.

To run the playbook:

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

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.2.5 Configuring custom repositories

Use the local repository feature to create a customized set of local repositories on the control plane for the cluster nodes to access.

1. Ensure the custom entry is included in the `software_config.json` file.

```
{
  "cluster_os_type": "ubuntu",
  "cluster_os_version": "22.04",
  "repo_config": "partial",
  "softwares": [
    {"name": "k8s", "version": "1.26.12"},
    {"name": "jupyter", "version": "3.2.0"},
    {"name": "kubeflow", "version": "1.8"},
    {"name": "openldap"},
    {"name": "beegfs", "version": "7.2.6"},
    {"name": "nfs"},
    {"name": "kserve"},
    {"name": "custom"},
    {"name": "amdgpu", "version": "6.0"},
    {"name": "cuda", "version": "12.3.2"},
    {"name": "ofed", "version": "24.01-0.3.3.1"},
    {"name": "telemetry"},
    {"name": "utils"},
    {"name": "vllm"},
    {"name": "pytorch"},
    {"name": "tensorflow"}
  ],
  "amdgpu": [
    {"name": "rocm", "version": "6.0"}
  ],
  "vllm": [
```

(continues on next page)

(continued from previous page)

```

        {"name": "vllm_amd", "version": "vllm-v0.2.4"},
        {"name": "vllm_nvidia", "version": "latest"}
    ],
    "pytorch": [
        {"name": "pytorch_cpu", "version": "latest"},
        {"name": "pytorch_amd", "version": "latest"},
        {"name": "pytorch_nvidia", "version": "23.12-py3"}
    ],
    "tensorflow": [
        {"name": "tensorflow_cpu", "version": "latest"},
        {"name": "tensorflow_amd", "version": "latest"},
        {"name": "tensorflow_nvidia", "version": "23.12-tf2-py3"}
    ]
}

```

2. Create a `custom.json` file in the following directory: `input/config/<operating_system>/<operating_system_version>` to define the repositories. For example, For a cluster running RHEL 8.8, go to `input/config/rhel/8.8/` and create the file there. The file is a JSON list consisting of the package name, repository type, URL (optional), and version (optional). Below is a sample version of the file:

```

{
  "custom": {
    "cluster": [
      {
        "package": "ansible==5.3.2",
        "type": "pip_module"
      },
      {
        "package": "docker-ce-24.0.4",
        "type": "rpm",
        "repo_name": "docker-ce-repo"
      },
      {
        "package": "gcc",
        "type": "rpm",
        "repo_name": "appstream"
      },
      {
        "package": "community.general",
        "type": "ansible_galaxy_collection",
        "version": "4.4.0"
      },
      {
        "package": "perl-Switch",
        "type": "rpm",
        "repo_name": "codeready-builder"
      },
      {
        "package": "prometheus-slurm-exporter",

```

(continues on next page)

(continued from previous page)

```

    "type": "git",
    "url": "https://github.com/vpenso/prometheus-slurm-exporter.git",
    "version": "master"
  },
  {
    "package": "ansible.utils",
    "type": "ansible_galaxy_collection",
    "version": "2.5.2"
  },
  {
    "package": "prometheus-2.23.0.linux-amd64",
    "type": "tarball",
    "url": "https://github.com/prometheus/prometheus/releases/download/v2.23.0/
↪prometheus-2.23.0.linux-amd64.tar.gz"
  },
  {
    "package": "metallb-native",
    "type": "manifest",
    "url": "https://raw.githubusercontent.com/metallb/metallb/v0.13.4/config/
↪manifests/metallb-native.yaml"
  },
  {
    "package": "registry.k8s.io/pause",
    "version": "3.9",
    "type": "image"
  }
]
}
}

```

2. Enter the parameters required in `input/local_repo_config.yml` as explained [here](#).
3. Run the following commands:

```

cd local_repo
ansible-playbook local_repo.yml

```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.3 Installing the provision tool

This playbook achieves the following tasks:

- Discovers potential cluster nodes.
- Installs Rocky, Ubuntu, or RHEL on the discovered nodes.

2.3.1 Before you run the provision tool

- (Recommended) Run `prereq.sh` to get the system ready to deploy Omnia. Alternatively, ensure that [Ansible 2.14](#) and [Python 3.9](#) are installed on the system.
- All target bare-metal servers should be reachable to the chosen control plane.
- Set the IP address of the control plane. The control plane NIC connected to remote servers (through the switch) should be configured with two IPs (BMC IP and admin IP) in a shared LOM or hybrid set up. In the case dedicated network topology, a single IP (admin IP) is required.

```
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 18:66:da:89:d4:68 brd ff:ff:ff:ff:ff:ff
    inet 10.5.255.254/16 brd 10.5.255.255 scope global noprefixroute eno1
        valid_lft forever preferred_lft forever
    inet 10.3.255.254/16 brd 10.3.255.255 scope global noprefixroute eno1
        valid_lft forever preferred_lft forever
    inet6 fe80::1a66:daff:fe89:d468/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Fig. 1: Control plane NIC IP configuration in a LOM setup

```
2: eno8303: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether b4:45:06:eb:da:4d brd ff:ff:ff:ff:ff:ff
    inet 10.5.255.254/16 brd 10.5.255.255 scope global noprefixroute eno8303
        valid_lft forever preferred_lft forever
    inet6 fe80::b645:6ff:feeb:da4d/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Fig. 2: Control plane NIC IP configuration in a dedicated setup

- Set the hostname of the control plane using the `hostname.domain.name` format.

Hostname requirements

- 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: `hostname000000x.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 cluster node to 'node000001.omnia.test'. Omnia appends 6 digits to the hostname to individually name each target node.

For example, `controlplane.omnia.test` is acceptable.

```
hostnamectl set-hostname controlplane.omnia.test
```

Note: The domain name specified for the control plane should be the same as the one specified under `domain_name` in `input/provision_config.yml`.

- To provision the bare metal servers, download one of the following ISOs to the control plane:

1. Rocky 8

2. [RHEL 8.x](#)3. [Ubuntu](#)

Note: Ensure the ISO provided has downloaded seamlessly (No corruption). Verify the SHA checksum/ download size of the ISO file before provisioning to avoid future failures.

Note the compatibility between cluster OS and control plane OS below:

Control Plane OS	Cluster Node OS	Compatibility
RHEL ¹	RHEL	Yes
RHEL ¹	Rocky	Yes
Rocky	Rocky	Yes
Ubuntu	Ubuntu	Yes
Rocky	Ubuntu	No
RHEL	Ubuntu	No
Ubuntu	RHEL	No
Ubuntu	Rocky	No

- **Ensure that all connection names under the network manager match their corresponding device names.**

To verify network connection names:

```
nmcli connection
```

To verify the device name:

```
ip link show
```

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

- When discovering nodes via a mapping file, all target nodes should be set up in PXE mode before running the playbook.

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.
 - For servers with an existing OS being discovered via BMC, ensure that the first PXE device on target nodes should be the designated active NIC for PXE booting.
-

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

¹ Ensure that control planes running RHEL have an active subscription or are configured to access local repositories. The following repositories should be enabled on the control plane: **AppStream**, **BaseOS**.

2.3.2 Discovery Mechanisms

Depending on the values provided in `input/provision_config.yml`, target nodes can be discovered in one of three ways:

mapping

Manually collect PXE NIC information for target servers and define them to Omnia (using the `pxe_mapping_file` variable in `input/provision_config.yml`) using a mapping file using the below format:

`pxe_mapping_file.csv`

```
SERVICE_TAG,HOSTNAME,ADMIN_MAC,ADMIN_IP,BMC_IP
XXXXXXXX,n1,xx:yy:zz:aa:bb:cc,10.5.0.101,10.3.0.101
XXXXXXXX,n2,aa:bb:cc:dd:ee:ff,10.5.0.102,10.3.0.102
```

Note:

- The header fields mentioned above are case sensitive.
- The service tags provided are not validated. Ensure the correct service tags are provided.
- The hostnames provided should not contain the domain name of the nodes.
- All fields mentioned in the mapping file are mandatory except `bmc_ip`.
- The MAC address provided in `pxe_mapping_file.csv` should refer to the PXE NIC on the target nodes.
- If the field `bmc_ip` is not populated, manually set the nodes to PXE mode and start provisioning. If the fields are populated and IPMI is enabled, Omnia will take care of provisioning automatically.
- Target servers should be configured to boot in PXE mode with the appropriate NIC as the first boot device.
- To assign IPs on the BMC network while discovering servers using a mapping file, target servers should be in DHCP mode or switch details should be provided.

Caution: Details provided in the mapping file are not validated. If incorrect details are passed on to the Omnia DB (this takes place when `discovery.yml` or `discovery_provision.yml` is run), delete the nodes with incorrect information using [the linked script](#). If the `bmc_ip` alone is incorrect, manually PXE boot the target server to update the database.

To continue to the next steps:

- [Provisioning the cluster](#)

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

switch_based

Pre requisites

- Set the value of `enable_switch_based` to `true` in `input/provision_config.yml`. Additionally, ensure that the variable `switch_based_details` in `input/provision_config.yml` is populated with the IP address and port details of the ToR switch.
- Switch port range where all BMC NICs are connected should be provided.
- BMC credentials should be the same across all servers and provided as input to Omnia. All BMC network details should be provided in `input/network_spec.yml`.
- SNMP v3 should be enabled on the switch and the credentials should be provided in `input/provision_config_credentials.yml`.
- Non-admin user credentials for the switch need to be provided.

Note:

- To create an SNMPv3 user on S series switches (running OS10), use the following commands:
 - To create SNMP view: `snmp-server view test_view internet included`
 - To create SNMP group: `snmp-server group testgroup 3 auth read test_view`
 - To create SNMP users: `snmp-server user authuser1 testgroup 3 auth sha authpasswd1`
- To verify the changes made, use the following commands:
 - To view the SNMP views: `show snmp view`
 - To view the SNMP groups: `show snmp group`
 - To view the SNMP users: `show snmp user`
- To save this configuration for later use, run: `copy running-configuration startup-configuration`
- For more information on SNMP on S series switch [click here](#)
- For more information on SNMP on N series switch [click here](#)

-
- IPMI over LAN needs to be enabled for the control plane.

```
racadm set iDRAC.IPMILan.Enable 1
racadm get iDRAC.IPMILan
```

- Target servers should be configured to boot in PXE mode with appropriate NIC as the first boot device.
- Set the IP address of the control plane. The control plane NIC connected to remote servers (through the switch) should be configured with two IPs (BMC IP and admin IP) in a shared LOM or hybrid set up. In the case dedicated network topology, a single IP (admin IP) is required.

```
valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 18:66:da:89:d4:68 brd ff:ff:ff:ff:ff:ff
    inet 10.5.255.254/16 brd 10.5.255.255 scope global noprefixroute eno1
        valid_lft forever preferred_lft forever
    inet 10.3.255.254/16 brd 10.3.255.255 scope global noprefixroute eno1
        valid_lft forever preferred_lft forever
    inet6 fe80::1a66:daff:fe89:d468/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Caution:

- Do not use daisy chain ports or the port used to connect to the control plane in `switch_based_details` in `input/provision_config.yml`. This can cause IP conflicts on servers attached to potential target ports.
- Omnia does not validate SNMP switch credentials, if the provision tool is run with incorrect credentials, use the clean-up script and re-run the provision tool with the correct credentials.
- If you are re-provisioning your cluster (that is, re-running the `discovery_provision.yml` playbook) after a [clean-up](#), ensure to use a different `static_range` against `bmc_network` in `input/network_spec.yml` to avoid a conflict with newly assigned servers. Alternatively, disable any OS available in the Boot Option Enable/Disable section of your BIOS settings (**BIOS Settings > Boot Settings > UEFI Boot Settings**) on all target nodes.

Note:

- If any of the target nodes have a pre-provisioned BMC IP, ensure that these IPs are not part of the `static_range` specified in `input/network_spec.yml` under the `bmc_network` to avoid any bmc IP conflicts.
- In case of a duplicate node object, duplicate BMC nodes will be deleted automatically by the **duplicate_node_cleanup** service that runs every 30 minutes. When nodes are discovered via mapping and switch details, the nodes discovered via switch details will not be deleted. Delete the node manually [using the delete node playbook](#).

To clear the configuration on Omnia provisioned switches and ports, [click here](#).

To continue to the next steps:

- [Provisioning the cluster](#)

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

BMC

For automatic provisioning of servers and discovery, the BMC method can be used.

Pre requisites

- Set the IP address of the control plane. The control plane NIC connected to remote servers (through the switch) should be configured with two IPs (BMC IP and admin IP) in a shared LOM or hybrid set up. In the case dedicated network topology, a single IP (admin IP) is required.

```
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 18:66:da:89:d4:68 brd ff:ff:ff:ff:ff:ff
    inet 10.5.255.254/16 brd 10.5.255.255 scope global noprefixroute eno1
        valid_lft forever preferred_lft forever
    inet 10.3.255.254/16 brd 10.3.255.255 scope global noprefixroute eno1
        valid_lft forever preferred_lft forever
    inet6 fe80::1a66:daff:fe89:d468/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

- To assign IPs on the BMC network while discovering servers using a BMC details, target servers should be in DHCP mode or switch details should be provided.
- BMC credentials should be the same across all servers and provided as input to Omnia in the parameters explained below.
- Target servers should be configured to boot in PXE mode with the appropriate NIC as the first boot device.

- If the `discovery_ranges` provided are outside the `bmc_subnet`, ensure the target nodes can reach the control plane.
- IPMI over LAN needs to be enabled for the BMC.

```
racadm set iDRAC.IPMITLan.Enable 1
racadm get iDRAC.IPMITLan
```

Caution: If you are re-provisioning your cluster (that is, re-running the `discovery_provision.yml` playbook) after a [clean-up](#), ensure to use a different `static_range` against `bmc_network` in `input/network_spec.yml` to avoid a conflict with newly assigned servers. Alternatively, disable any OS available in the `Boot Option Enable/Disable` section of your BIOS settings (**BIOS Settings > Boot Settings > UEFI Boot Settings**) on all target nodes.

- All target servers should be reachable from the `admin_network` specified in `input/network_spec.yml`.
- BMC network details should be provided in the `input/network_spec.yml` file.

When entering details in `input/network_spec.yml`:

- Ensure that the netmask bits for the BMC network and the admin network are the same.
- The static and dynamic ranges for the BMC network accepts multiple comma-separated ranges.
- The network gateways on both admin and BMC networks are optional.

Note: If the value of `enable_switch_based` is set to true, nodes will not be discovered via BMC irrespective of the contents in `input/network_spec.yml`.

To continue to the next steps:

- [Provisioning the cluster](#)

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

switch_based

Omnia can query known switches (by SNMPv3 username/password) for information on target node MAC IDs.

Pros

- The whole discovery process is totally automatic.
- Admin IP, BMC IP and Infiniband IP address configuration is automatic on the target nodes.
- Re-provisioning of servers will be automatic.
- PXE booting servers is supported via split ports on the switch.

Cons

- Users need to enable IPMI on target servers.
- Servers require a manual PXE boot after the first run of the provision tool.

For more information regarding switch-based discovery, [click here](#)

mapping

Manually collect PXE NIC information for target servers and manually define them to Omnia using a mapping file using the below format:

pxe_mapping_file.csv

```
SERVICE_TAG,HOSTNAME,ADMIN_MAC,ADMIN_IP,BMC_IP
XXXXXXXX,n1,xx:yy:zz:aa:bb:cc,10.5.0.101,10.3.0.101
XXXXXXXX,n2,aa:bb:cc:dd:ee:ff,10.5.0.102,10.3.0.102
```

Pros

- Easily customized if the user maintains a list of MAC addresses.

Cons

- The user needs to be aware of the MAC/IP mapping required in the network.
- Servers require a manual PXE boot if iDRAC IPs are not configured.

For more information regarding mapping files, [click here](#)

bmc

Omnia can also discover nodes via their iDRAC using IPMI.

Pros

- Discovery and provisioning of servers is automatic.
- Admin, BMC and Infiniband IP address configuration is automatic on the control plane.
- LOM architecture is supported (including cloud enclosures: C6420, C6520, C6620).

Cons

- For iDRACs that are not DHCP enabled (ie Static), users need to enable IPMI manually.

For more information regarding BMC, [click here](#)

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.3.3 Input parameters for the provision tool

Fill in all required parameters in `input/provision_config.yml`, `provision_config_credentials.yml`, `input/software_config.json`.

Caution: Do not remove or comment any lines in the `input/provision_config.yml` file.

Table 3: provision_config.yml

Parameter	Details
public_nic string Required	The nic/ethernet card that is connected to the public internet. Default values: eno2
iso_file_path string Required	Path where user has placed the iso image that needs to be provisioned on target nodes. Accepted files are Rocky8-DVD or RHEL-8.x-DVD (full OS). iso_file_path should contain the provision_os and provision_os_version values in the filename. Default values: "/home/RHEL-8.6.0-20220420.3-x86_64-dvd1.iso"
node_name string Required	<ul style="list-style-type: none"> Prefix for target node names, if dynamically allocated. Hostname = node_name + '0000x' + domain_name Hostname <= 65 characters Example: servernode00001.Omnia.test , where node_name =servernode, domain_name =Omnia.test , 00001 used by Omnia. Default values: node
domain_name string Required	<ul style="list-style-type: none"> Domain name the user intends to configure on the cluster. Hostname = node_name + '0000x' + domain_name Hostname <= 65 characters Please provide a valid domain name according to the domain name standards. Example: servernode00001.Omnia.test , where node_name=servernode, domain_name=Omnia.test , 00001 used by Omnia.
pxe_mapping_file_path string Optional	<ul style="list-style-type: none"> This variable is required to discover nodes using a mapping file. The mapping file consists of the Service tag, Admin MAC,Hostname and its respective admin IP address and/or BMC IP. Ensure that the admin IP addresses provided are within the admin_static_ranges. A sample file is provided here: examples/pxe_mapping_file.csv. The headers of the CSV are SERVICE_TAG,ADMIN_MAC,HOSTNAME,ADMIN_IP,BMC_IP.
enable_switch_based boolean ¹ Required	<ul style="list-style-type: none"> Variable indicates whether switch based discovery should be enabled to discover the nodes To enable switch based discovery, set enable_switch_based to true. If enable_switch_based is set to true,the following inputs should be provided: <ul style="list-style-type: none"> switch_based_details should be provided in provision_config.yml switch_snmp3_username and switch_snmp3_password should be provided in provision_config_credentials.

Table 4: provision_config_credentials.yml

Parameter	Details
provision_password string Required	<ul style="list-style-type: none"> • Password set for the root account of target nodes during provisioning. • Length ≥ 8 characters • Password must not contain -, ', "
postgresdb_password string Required	<ul style="list-style-type: none"> • Password set for the postgresDB on target nodes during provisioning. • Length ≥ 8 characters • Password must not contain -, ', "
bmc_username string Required	<ul style="list-style-type: none"> • The username set on target iDRACs. • Username must not contain -, ', "
bmc_password string Required	<ul style="list-style-type: none"> • The password set on target iDRACs. • The username must not contain -, ', "
switch_snmp3_username string Optional	<ul style="list-style-type: none"> • This variable is required when discovering nodes via switch details. • Non-admin SNMPv3 credentials of the PXE switch. • If multiple switches are provided, ensure the credentials are same across switches. • Username must not contain -, ', "
switch_snmp3_password string Optional	<ul style="list-style-type: none"> • This variable is required when discovering nodes via switch details. • Non-admin SNMPv3 credentials of the PXE switch. • If multiple switches are provided, ensure the credentials are same across switches. • Password must not contain -, ', "
docker_username string Optional	<ul style="list-style-type: none"> • Username for Dockerhub account used for Docker logins. • A kubernetes secret will be created and patched to the service account in default namespace. • This kubernetes secret can be used to pull images from private repositories.
docker_password string Optional	<ul style="list-style-type: none"> • Password for Dockerhub account used for Docker logins. • This value is mandatory if docker_username is provided.

¹ Boolean parameters do not need to be passed with double or single quotes.

Table 5: software_config.json

Parameter	Details
cluster_os_type string Required	<ul style="list-style-type: none"> The operating system running on the cluster (rhel, rocky, and ubuntu). Default value: ubuntu.
cluster_os_version string Required	<ul style="list-style-type: none"> The OS Version that will be provisioned on compute nodes. For RHEL, the accepted values are 8.6, 8.7, and 8.8. For Rocky, the accepted values are 8.6, 8.7, and 8.8. For Ubuntu, the accepted values are 20.04, 22.04. Default value: 22.04
repo_config string Required	<ul style="list-style-type: none"> The type of offline configuration user needs. When the value is set to always, Omnia creates a local repository/registry on the Control plane hosting all the packages/images required for the cluster. When the value is set to partial, Omnia creates a local repository/registry on the Control plane hosting all the packages/images except those listed in the <code>user_registry</code> in <code>input/local_repo_config.yml</code>. When the value is set to never, Omnia does not create a local repository/registry. All the packages/images are directly downloaded on the cluster. <hr/> Note: <ul style="list-style-type: none"> After <code>local_repo.yml</code> has run, the value of <code>repo_config</code> in <code>input/software_config.json</code> cannot be updated without running the <code>control_plane_cleanup.yml</code> script first. Irrespective of the value of <code>repo_config</code>, all local repositories that are not available as images, debian packages, or RPMs will be downloaded and configured locally on the control plane. Additionally, AMD GPU drivers, ROCm drivers, CUDA, and OFED are downloaded by default. <hr/> <ul style="list-style-type: none"> Accepted values: <ul style="list-style-type: none"> <code>always</code> <code>partial</code> <- Default <code>never</code>
softwares JSON list Required	<ul style="list-style-type: none"> A JSON list of required software and (optionally) the software revision. The following software should be listed with a version in the list: BeeGFS, AMD GPU, Kubernetes, CUDA, OFED, BCM RoCE, UCX, and ROCm. A minimum of one of the following software should be listed for <code>local_repo.yml</code> to execute correctly.

Update the `input/network_spec.yml` file for all networks available for use by the control plane.

- The following `admin_nic` details are mandatory:
 - `nic_name`: The name of the NIC on which the administrative network is accessible to the control plane.
 - `netmask_bits`: The 32-bit “mask” used to divide an IP address into subnets and specify the network’s available hosts.
 - `static_range`: The static range of IPs to be provisioned on target nodes.
 - `dynamic_range`: The dynamic range of IPs to be provisioned on target nodes.
 - `correlation_to_admin`: Boolean value used to indicate whether all other networks specified in the file (eg: `bmc_network`) should be correlated to the admin network. For eg: if a target node is assigned the IP `xx.yy.0.5` on the admin network, it will be assigned the IP `aa.bb.0.5` on the BMC network. This value is irrelevant when discovering nodes using a mapping file.
 - `admin_uncorrelated_node_start_ip`: If `correlation_to_admin` is set to true but correlated IPs are not available on non-admin networks, provide an IP within the `static_range` of the admin network that can be used to assign admin static IPs to uncorrelated nodes. If this is empty, then the first IP in the `static_range` of the admin network is taken by default. This value is irrelevant when discovering nodes using a mapping file.
 - `CIDR`: Classless or Classless Inter-Domain Routing (CIDR) addresses use variable length subnet masking (VLSM) to alter the ratio between the network and host address bits in an IP address.
 - `MTU`: Maximum transmission unit (MTU) is a measurement in bytes of the largest data packets that an Internet-connected device can accept.
 - `DNS`: A DNS server is a computer equipped with a database that stores the public IP addresses linked to the domain names of websites, enabling users to reach websites using their IP addresses.
 - `VLAN`: A 12-bit field that identifies a virtual LAN (VLAN) and specifies the VLAN that an Ethernet frame belongs to. This value is not supported on admin and bmc networks.
- If the `nic_name` is the same on both the `admin_network` and the `bmc_network`, a LOM setup is assumed.
- BMC network details are not required when target nodes are discovered using a mapping file.
- If `bmc_network` properties are provided, target nodes will be discovered using the BMC method in addition to the methods whose details are explicitly provided in `provision_config.yml`.

Caution:

- Do not assign the subnet `10.4.0.0/24` to any interfaces in the network as `nerdctl` uses it by default.
- If a DNS server is available on the network, ensure that the ranges provided in the `input/network_spec.yml` file do not include the IP ranges of the DNS server.
- All provided network ranges and nic IP addresses should be distinct with no overlap in the `input/network_spec.yml`.

A sample is provided below:

```
---
Networks:
- admin_network:
  nic_name: "eno1"
  netmask_bits: "16"
  static_range: "10.5.0.1-10.5.0.200"
```

(continues on next page)

(continued from previous page)

```
dynamic_range: "10.5.1.1-10.5.1.200"
correlation_to_admin: true
admin_uncorrelated_node_start_ip: "10.5.0.50"
network_gateway: ""
DNS: ""
MTU: "1500"

- bmc_network:
  nic_name: ""
  netmask_bits: ""
  static_range: ""
  dynamic_range: ""
  reassignment_to_static: true
  discover_ranges: ""
  network_gateway: ""
  MTU: "1500"
```

Note:

- The input/provision_config_credentials.yml file is encrypted on the first run of the provision tool:

To view the encrypted parameters:

```
ansible-vault view provision_config_credentials.yml --vault-password-file .
↪provision_vault_key
```

To edit the encrypted parameters:

```
ansible-vault edit provision_config_credentials.yml --vault-password-file .
↪provision_vault_key
```

- The strings `admin_network` and `bmc_network` in the `input/network_spec.yml` file should not be edited. Also, the properties `nic_name`, `static_range`, and `dynamic_range` cannot be edited on subsequent runs of the provision tool.
 - Netmask bits are mandatory and should be same for both the `admin_network` and `bmc_network` (ie between 1 and 32; 1 and 32 are acceptable values).
 - Ensure that the CIDR is aligned with the `netmask_bits` provided.
 - The `discover_ranges` property of the `bmc_network` can accept multiple comma-separated ranges.
 - The `VLAN` property is optional but should be between 0 and 4095 (0 and 4095 are not acceptable values).
-

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.3.4 Provisioning the cluster

Edit the `input/provision_config.yml`, `input/provision_config.yml`, and `input/network_spec.yml` files to update the required variables. A list of the variables required is available by [discovery mechanism](#).

Note: The first PXE device on target nodes should be the designated active NIC for PXE booting.

Network Settings

	Current Value
PXE Device1	Enabled ▾
PXE Device2	Disabled ▾
PXE Device3	Disabled ▾
PXE Device4	Disabled ▾
<div> <div>▾ PXE Device1 Settings</div> </div>	
	Current Value
Interface	Integrated NIC 1 Port 1 Partition 1 ▾
Protocol	IPv4 ▾
VLAN	Enabled ▾

Optional configurations managed by the provision tool

Using multiple versions of a given OS

Omnia now supports deploying different versions of the same OS. With each run of `discovery_provision.yml`, a new deployable OS image is created with a distinct type:

- Rocky: 8.6, 8.7, 8.8
- RHEL: 8.6, 8.7, 8.8
- Ubuntu: 20.04, 22.04

depending on the values provided in `input/software_config.json`.

Note: While Omnia deploys the minimal version of the OS, the multiple version feature requires that the Rocky full (DVD) version of the OS be provided.

Disk partitioning

Omnia now allows for customization of disk partitions applied to remote servers. The disk partition `desired_capacity` has to be provided in MB. Valid `mount_point` values accepted for disk partition are `/var`, `/tmp`, `/usr`, `swap`. The default partition size provided for RHEL/Rocky is `/boot`: 1024MB, `/boot/efi`: 256MB and remaining space to `/` partition. Default partition size provided for Ubuntu is `/boot`: 2148MB, `/boot/efi`: 1124MB and remaining space to `/` partition. Values are accepted in the form of JSON list such as:

```
disk_partition:
  - { mount_point: "/var", desired_capacity: "102400" }
  - { mount_point: "swap", desired_capacity: "10240" }
```

Running the provision tool

To deploy the Omnia provision tool, ensure that `input/provision_config.yml`, `input/network_spec.yml`, and `input/provision_config_credentials.yml` are updated and then run:

```
ansible-playbook discovery_provision.yml
```

`discovery_provision.yml` runs in three stages that can be called individually:

Caution: Always execute `discovery_provision.yml` within the `omnia` directory. That is, always change directories (`cd omnia`) to the path where the playbook resides before running the playbook.

Preparing the control plane

- Installs required tool packages.
- Verifies and updates firewall settings.
- Installs xCAT.
- Configures Omnia databases basis `input/network_spec.yml`.
- Creates empty inventory files in the control plane at `/opt/omnia/omnia_inventory/`. These inventory files will be filled with information of compute node service tag post provisioning based on type of CPUs and GPUs they have. The inventory files are:
 - `compute_cpu_amd`
 - `compute_cpu_intel`
 - `compute_gpu_amd`
 - `compute_gpu_nvidia`
 - `compute_servicetag_ip`

Note:

- Service tags will only be written into the inventory files after the nodes are successfully PXE booted post provisioning.
 - For a node's service tag to list in an inventory file, two conditions must be met:
 - Node status must be "booted" in DB.
 - Node's service tag information is present in DB.
 - Nodes are not removed from the inventory files even if they are physically disconnected. Ensure to run the `delete node` [playbook](#) to remove the node.
 - To regenerate an inventory file, use the `playbook omnia/utils/inventory_tagging.yml`.
-

```
cd prepare_cp
ansible-playbook prepare_cp.yml
```

Discovering the nodes

- Discovers all target servers.
- PostgreSQL database is set up with all relevant cluster information such as MAC IDs, hostname, admin IP, BMC IPs etc.
- Configures the control plane with NTP services for cluster node synchronization.

To call this playbook individually, run:

```
cd discovery
ansible-playbook discovery.yml
```

Provisioning the nodes

- The intended operating system and version is provisioned on the primary disk partition on the nodes. If a BOSS Controller card is available on the target node, the operating system is provisioned on the boss controller disks.

To call this playbook individually, run:

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

After successfully running `discovery_provision.yml`, go to [Building Clusters to setup Slurm, Kubernetes, NFS, BeeGFS and Authentication](#).

Note:

- racadm and ipmitool are installed on all target nodes except Ubuntu 20.04.
- Ansible playbooks by default run concurrently on 5 nodes. To change this, update the `forks` value in `ansible.cfg` present in the respective playbook directory.
- While the `admin_nic` on cluster nodes is configured by Omnia to be static, the public NIC IP address should be configured by user.
- If the target nodes were discovered using switch-based or mapping mechanisms, manually PXE boot the target servers after the `discovery_provision.yml` playbook is executed and the target node lists as **booted** in the `nodeinfo` table.
- All ports required for xCAT to run will be opened (For a complete list, check out the [Security Configuration Document](#)).
- After running `discovery_provision.yml`, the file `input/provision_config_credentials.yml` will be encrypted. To edit the file, use the command: `ansible-vault edit provision_config.yml --vault-password-file .provision_vault_key`
- Post execution of `discovery_provision.yml`, IPs/hostnames cannot be re-assigned by changing the mapping file. However, the addition of new nodes is supported as explained [here](#).
- Default Python is installed during provisioning on Ubuntu cluster nodes. For Ubuntu 22.04, Python 3.10 is installed. For Ubuntu 20.04, Python 3.8 is installed.

Caution:

- Once xCAT is installed, restart your SSH session to the control plane to ensure that the newly set up environment variables come into effect. If the new environment variables still do not come into effect, enable manually using:

```
source /etc/profile.d/xcat.sh
```

- To avoid breaking the passwordless SSH channel on the control plane, do not run `ssh-keygen` commands post execution of `discovery_provision.yml` to create a new key.

- **Do not delete the following directories:**

- /root/xcat
- /root/xcat-dbback
- /docker-registry
- /opt/omnia
- /var/log/omnia

- On subsequent runs of `discovery_provision.yml`, if users are unable to log into the server, refresh the ssh key manually and retry.

```
ssh-keygen -R <node IP>
```

- If a subsequent run of `discovery_provision.yml` fails, the `input/provision_config.yml` file will be unencrypted.

To create a node inventory in `/opt/omnia`, [click here](#).

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.3.5 Checking node status

Via CLI

Run `nodels all nodelist.status` for a list of nodes and their statuses.

```
omnia-node00001: installing
omnia-node00002: booted
omnia-node00003: powering-on
omnia-node00004: booted
```

Possible values of node status are powering-off, powering-on, bmcready, installing, booting, post-booting, booted, failed.

Caution:

- Once xCAT is installed, restart your SSH session to the control plane to ensure that the newly set up environment variables come into effect. This will also allow the above command to work correctly. If the new environment variables still do not come into effect, enable manually using:

```
source /etc/profile.d/xcat.sh
```

Via omniadb

1. To access the DB, run:

```
psql -U postgres
\c omniadb
```

2. To view the schema being used in the cluster: \dn
3. To view the tables in the database: \dt
4. To view the contents of the nodeinfo table: `select * from cluster.nodeinfo;`

id	service_tag	node	hostname	admin_mac	admin_ip	
↪	bmc_ip	status	discovery_mechanism	bmc_mode	switch_ip	switch_name
↪	switch_port	cpu	gpu	cpu_count	gpu_count	\$

↪	-----					
↪	-----					
1		control_plane	newcp.new.dev	00:0a:f7:dc:11:42	10.5.255.	
↪ 254	0.0.0.0					
↪						
2	xxxxxxx	node2	node2.new.dev	c4:cb:e1:b5:70:44	10.5.0.12	
↪	10.30.0.12	booted	mapping			
↪		amd		1	0	
3	xxxxxxx	node3	node3.new.dev	f4:02:70:b8:bc:2a	10.5.0.10	
↪	10.30.0.10	booted	mapping			
↪		amd	amd	2	1	
(3 rows)						

Possible values of node status are powering-off, powering-on, bmcready, installing, booting, post-booting, booted, failed.

Note:

- The `gpu_count` in the DB is only updated every time a cluster node is PXE booted.
- Nodes listed as “failed” can be diagnosed using the `/var/log/xcat/xcat.log` file on the target node. Correct any underlying issues and [re-provision the node](#).
- Information on debugging nodes stuck at ‘powering-on’, ‘bmcready’ or ‘installing’ for longer than expected is available [here](#). Correct any underlying issue on the node and [re-provision the node](#).
- A blank node status indicates that no attempt to provision has taken place. Attempt a manual PXE boot on the node to initiate provisioning.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.3.6 Configuring additional NICs on the nodes

After running `discovery_provision.yml` or `discovery_provision.yml` and the nodes boot up, additional NICs can be configured on target nodes using the `nic_update.yml` playbook.

Prerequisites

- All target nodes are provisioned and booted. [Click here to verify the status of all nodes.](#)
- The `input/network_spec.yml` file has been updated with all network information in addition to admin network and bmc network information. Below are all applicable properties of an additional network:
 - `nic_name`: The name of the NIC on which the administrative network is accessible to the control plane.
 - `netmask_bits`: The 32-bit “mask” used to divide an IP address into subnets and specify the network’s available hosts.
 - `static_range`: The static range of IPs to be provisioned on target nodes.
 - `dynamic_range`: The dynamic range of IPs to be provisioned on target nodes.
 - `correlation_to_admin`: Boolean value used to indicate whether all other networks specified in the file (eg: `bmc_network`) should be correlated to the admin network. For eg: if a target node is assigned the IP `xx.yy.0.5` on the admin network, it will be assigned the IP `aa.bb.0.5` on the BMC network. This value is irrelevant when discovering nodes using a mapping file.
 - `admin_uncorrelated_node_start_ip`: If `correlation_to_admin` is set to true but correlated IPs are not available on non-admin networks, provide an IP within the `static_range` of the admin network that can be used to assign admin static IPs to uncorrelated nodes. If this is empty, then the first IP in the `static_range` of the admin network is taken by default. This value is irrelevant when discovering nodes using a mapping file.
 - `VLAN`: A 12-bit field that identifies a virtual LAN (VLAN) and specifies the VLAN that an Ethernet frame belongs to. This property is not supported on clusters running Ubuntu.

The below properties are only applicable to additional NICs

- `CIDR`: Classless or Classless Inter-Domain Routing (CIDR) addresses use variable length subnet masking (VLSM) to alter the ratio between the network and host address bits in an IP address.
- `MTU`: Maximum transmission unit (MTU) is a measurement in bytes of the largest data packets that an Internet-connected device can accept.
- `DNS`: A DNS server is a computer equipped with a database that stores the public IP addresses linked to the domain names of websites, enabling users to reach websites using their IP addresses.

Note:

- If a CIDR value is provided, the complete subnet is used for Omnia to assign IPs and where possible, the IPs will be correlated with the assignment on the admin network.
 - If a VLAN is required, ensure that a VLAN ID is provided in the field `vlan`. This field is not supported on admin or bmc networks.
-

Below is a sample of additional NIC information in a `input/network_spec.yml` file:


```

- thor_network1:
  netmask_bits: "20"
  CIDR: "10.10.16.0"
  network_gateway: ""
  MTU: "1500"
  VLAN: ""

- thor_network2:
  netmask_bits: "20"
  static_range: "10.10.1.1-10.10.15.254"
  network_gateway: ""
  MTU: "1500"
  VLAN: "1"

```

- The input/server_spec.yml file has been updated with all NIC information of the target nodes.
 - All NICs listed in the server_spec.yml file are grouped into categories (groups for servers). The string “Categories:” should not be edited out of the input/server_spec.yml file.
 - The name of the NIC specified in the file (in this sample, ensp0, ensp0.5, and eno1) is the unique identifier of NICs in the file.
 - The property nictype indicates what kind of NIC is in use (ethernet, infiniband, or vlan). If the nictype is set to vlan, ensure to specify a primary NIC for the VLAN using the property nicdevices.
 - While new groups can be added to the server_spec.yml file on subsequent runs of the play-book, existing groups cannot be edited or deleted.

Note: The nicnetwork property should match any of the networks specified in input/network_spec.yml.

Below is a sample input/server_spec.yml file:

```

---
Categories:
- group-1:
  - network:
    - ensp0:
      nicnetwork: "thor_network1"
      nictypes: "ethernet"
    - ensp0.5:
      nicnetwork: "thor_network2"
      nictypes: "vlan"
      nicdevices: "ensp0"

- group-2:
  - network:
    - eno1:
      nicnetwork: "thor_network1"
      nictypes: "ethernet"

```

Use the below commands to assign IPs to the NICs:

```
cd nic_update
ansible-playbook nic_update -i inventory
```

Where the inventory file passed includes user-defined groups, servers associated with them, and a mapping from the groups specified and the categories in `input/server_spec.yml` under [`<group name>:vars`]. Below is a sample:

```
[waco1]
10.5.0.3

[waco1:vars]
Categories=group-1

[waco2]
10.5.0.4
10.5.0.5

[waco2:vars]
Categories=group-2
```

Based on the provided sample files, server 10.5.0.3 has been mapped to waco1 which corresponds to group-1. Therefore, the NICs `ensp0` and `ensp0.5` will be configured in an ethernet VLAN group with `ens0` as the primary device.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.4 Creating node inventory

When `discovery_provision.yml`, `prepare_cp.yml`, or `utils/inventory_tagging.yml` is run, a set of inventory files is created in `/opt/omnia/omnia_inventory/` based on the [Omnia database](#). The inventories are created based on the type of CPUs and GPUs nodes have. The inventory files are:

- `compute_cpu_amd`

```
[compute_cpu_amd]
ABCD1
```

- `compute_cpu_intel`

```
[compute_cpu_intel]
ABCD1
```

- `compute_gpu_amd`

```
[compute_gpu_amd]
ABCD2
ABCD3
```

- `compute_gpu_nvidia`

```
[compute_gpu_nvidia]
ABCD1
```

- `compute_servicetag_ip`

```
[compute_servicetag_ip]
ABCD1 ansible_host=10.5.0.2
ABCD2 ansible_host=10.5.0.3
ABCD3 ansible_host=10.5.0.4
```

Note:

- Service tags will only be written into the inventory files after the nodes are successfully PXE booted post provisioning.
- For a node's service tag to list in an inventory file, two conditions must be met:
 - Node status must be “booted” in DB.
 - Node's service tag information is present in DB.
- To regenerate all the inventory files, use the playbook `utils/inventory_tagging.yml`.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.5 Configuring the cluster

Features enabled by `omnia.yml`

- Centralized authentication: Once all the required parameters in `security_config.yml` are filled in, `omnia.yml` can be used to set up FreeIPA/LDAP.
- Slurm: Once all the required parameters in `omnia_config.yml` are filled in, `omnia.yml` can be used to set up slurm.
- 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.

2.5.1 Input parameters for the cluster

These parameters are located in `input/omnia_config.yml`, `input/security_config.yml`, `input/telemetry_config.yml` and [optional] `input/storage_config.yml`.

Caution: Do not remove or comment any lines in the `input/omnia_config.yml`, `input/security_config.yml` and [optional] `input/storage_config.yml` file.

`omnia_config.yml`

Table 6: Parameters for kubernetes setup

Variables	Details
mariadb_password string Required	<ul style="list-style-type: none"> • Password used for Slurm database. • The Length of the password should be at least 8. • The password must not contain -, , • Default value: "password"
k8s_cni string Required	<ul style="list-style-type: none"> • Kubernetes SDN network. • Required when scheduler_type: "k8s" Choices: <ul style="list-style-type: none"> – "calico" <- default – "flannel"
pod_external_ip_range string Required	<ul style="list-style-type: none"> • These addresses will be used by Loadbalancer for assigning External IPs to K8s services • Make sure the IP range is not assigned to any node in the cluster. • Sample values: "10.11.0.100-10.11.0.150" , "10.11.0.0/16"
ansible_config_file_path string Required	<ul style="list-style-type: none"> • Path to directory hosting ansible config file (ansible.cfg file) • This directory is on the host running ansible, if ansible is installed using dnf • If ansible is installed using pip, this path should be set • Default value: /etc/ansible
slurm_installation_type string Optional	<ul style="list-style-type: none"> • Indicates whether the slurm installation will support configless or nfs mode. Choices: <ul style="list-style-type: none"> – nfs_share <- default – configless
k8s_service_addresses string Optional	<ul style="list-style-type: none"> • Kubernetes internal network for services. • This network must be unused in your network infrastructure. • Default value: "10.233.0.0/18"
k8s_pod_network_cidr string Optional	<ul style="list-style-type: none"> • Kubernetes pod network CIDR for internal network. When used, it will assign IP addresses from this range to individual pods. • This network must be unused in your network infrastructure. • Default value: "10.233.64.0/18"

Table 7: Parameters for slurm setup

Variables	Details
mariadb_password string Required	<ul style="list-style-type: none"> • Password used for Slurm database. • The Length of the password should be at least 8. • The password must not contain -, , • Default value: "password"
ansible_config_file_path string Required	<ul style="list-style-type: none"> • Path to directory hosting ansible config file (ansible.cfg file) • This directory is on the host running ansible, if ansible is installed using dnf • If ansible is installed using pip, this path should be set • Default value: /etc/ansible
slurm_installation_type string Optional	<ul style="list-style-type: none"> • Indicates whether the slurm installation will support configless or nfs mode Choices: – nfs_share <- default – configless
restart_slurm_services boolean Optional	<ul style="list-style-type: none"> • Indicates whether slurm services should be restarted. • Choices * true <- Default * false

security_config.yml

Table 8: Parameters for Authentication

Parameter	Details
domain_name string Required	<ul style="list-style-type: none"> • Sets the intended domain name. • If dc=omnia,dc=test, Provide omnia.test • If dc=dell,dc=omnia,dc=com Provide dell.omnia.com • Default values: omnia.test

Table 9: Parameters for OpenLDAP configuration

Parameter	Details
ldap_connection_type string Required	For a TLS connection, provide a valid certification path. For an SSL connection, ensure port 636 is open. Choices: <ul style="list-style-type: none"> • TLS <- Default • SSL
tls_ca_certificate string Optional	File path pointing to the Certificate Authority (CA) issued certificate path. Certificate files should be saved with a .pem or .crt extension. If not provided, a self-signed certificate is generated by Omnia.
tls_certificate string Optional	File path pointing to the certificate used to authorize the LDAP server. Certificate files should be saved with a .pem or .crt extension.
tls_certificate_key string Optional	The private key that matches the LDAP certificate.
openldap_db_username string Required	The username used to manage the LDAP database. Default value: "admin"
openldap_db_password string Required	The password used to configure and manage the LDAP database. Ensure that this value is 8 characters long.
openldap_config_username string Required	The username used to configure the LDAP database. Default value: "admin"
openldap_config_password string Required	The password used to configure the LDAP database. Ensure that this value is 8 characters long.
openldap_monitor_password string Required	The password used to monitor the LDAP database. Ensure that this value is 8 characters long.
openldap_organization string Required	LDAP server is configured using organizations. They are necessary for user creation and group mapping. Default value: "omnia"
openldap_organizational_unit string Required	LDAP server is configured using organizations. They are necessary for user creation and group mapping. Default value: "People"

Table 10: Parameters for FreeIPA configuration

Parameter	Details
realm_name string Required	<ul style="list-style-type: none"> • Sets the intended kerberos realm name. • It is required for FreeIPA setups. • A realm name is often, but not always the upper case version of the name of the DNS domain over which it presides. • Default value: "OMNIA.TEST"
directory_manager_password string Required	<ul style="list-style-type: none"> • The directory server operations require an administrative user. This user is referred to as the Directory Manager and has full access to the Directory for system management tasks and will be added to the instance of directory server created for IPA. • The password must be at least 8 characters long. • The password must not contain -, , ,"
kerberos_admin_password string Required	<ul style="list-style-type: none"> • kerberos_admin_password used by IPA admin user. The IPA server requires an administrative user, named 'admin'. • The password must be at least 8 characters long. • The password must not contain -, , ,"

storage_config.yml

Table 11: Parameters for Storage

Variables	Details
nfs_client_params JSON List Required	<ul style="list-style-type: none"> This JSON list contains all parameters required to set up NFS. For a bolt-on set up where there is a pre-existing NFS export, set <code>nfs_server</code> to <code>false</code>. When <code>nfs_server</code> is set to <code>true</code>, an NFS share is created on the control plane for access by all cluster nodes. For more information on the different kinds of configuration available, click here.
beegfs_rdma_support boolean Optional	This variable is used if user has RDMA-capable network hardware (e.g., InfiniBand) Choices: <ul style="list-style-type: none"> <code>false</code> <- Default <code>true</code>
beegfs_ofed_kernel_modules_path string Optional	<ul style="list-style-type: none"> The path where separate OFED kernel modules are installed. Ensure that the path provided here exists on all target nodes. Default value: <code>"/usr/src/ofa_kernel/default/include"</code>
beegfs_mgmt_server string Required	BeeGFS management server IP. <hr/> Note: The provided IP should have an explicit BeeGFS management server running . <hr/>
beegfs_mounts string Optional	BeeGfs-client file system mount location. If <code>storage.yml</code> is being used to change the BeeGFS mounts location, set <code>beegfs_unmount_client</code> to <code>true</code>. Default value: <code>"/mnt/beegfs"</code>
beegfs_unmount_client boolean Optional	Changing this value to <code>true</code> will unmount running instance of BeeGFS client and should only be used when decommissioning BeeGFS, changing the mount location or changing the BeeGFS version. Choices: <ul style="list-style-type: none"> <code>false</code> <- Default <code>true</code>
beegfs_version_change boolean Optional	Use this variable to change the BeeGFS version on the target nodes. Choices: <ul style="list-style-type: none"> <code>false</code> <- Default <code>true</code>
ansible_config_file_path string Required	<ul style="list-style-type: none"> Path to directory hosting ansible config file (ansible.cfg file) This directory is on the host running ansible, if ansible is installed using <code>dnf</code> If ansible is installed using <code>pip</code>, this path should be set

telemetry_config.yml

Table 12: Parameters

Parameter	Details
idrac_telemetry_support boolean ¹ Required	<ul style="list-style-type: none"> Enables iDRAC telemetry support and visualizations. Values: * false <- Default * true <hr/> <p>Note: When <code>idrac_telemetry_support</code> is true, <code>mysqldb_user</code>, <code>mysqldb_password</code> and <code>mysqldb_root_password</code> become mandatory.</p> <hr/>
omnia_telemetry_support boolean ^{Page 67, 1} Required	<ul style="list-style-type: none"> Starts or stops Omnia telemetry If <code>omnia_telemetry_support</code> is true, then at least one of <code>collect_regular_metrics</code> or <code>collect_health_check_metrics</code> or <code>collect_gpu_metrics</code> should be true, to collect metrics. If <code>omnia_telemetry_support</code> is false, telemetry acquisition will be stopped. Values: * false <- Default * true
visualization_support boolean ^{Page 67, 1} Required	<ul style="list-style-type: none"> Enables visualizations. Values: * false <- Default * true <hr/> <p>Note: When <code>visualization_support</code> is true, <code>grafana_username</code> and <code>grafana_password</code> become mandatory.</p> <hr/>
appliance_k8s_pod_net_cidr string Required	<ul style="list-style-type: none"> Kubernetes pod network CIDR for appliance k8s network. Make sure this value does not overlap with any of the host networks. Default value: "192.168.0.0/16"
pod_external_ip_start_range string Required	<ul style="list-style-type: none"> The start of the range that will be used by Load-balancer for assigning IPs to K8s services in admin NIC subnet configured on the control plane. The first and second octets (x,y) are not used/validated by Omnia. These values are internally calculated based on the value of <code>admin_nic_subnet</code> in <code>input/provision_config.yml</code>. If <code>pod_external_ip_start_range</code>: "x.y.240.100" and <code>pod_external_ip_end_range</code>: "x.y.240.105" and
66	<p>Chapter 2. Quick Installation Guide</p> <ul style="list-style-type: none"> If <code>admin_nic_subnet</code> is provided in <code>provision_config.yml</code> is 10.5.0.0, <code>pod_external_ip_start_range</code> will be 10.5.240.100 and

Click here for more information on [OpenLDAP](#), [FreeIPA](#), [Telemetry](#), [BeeGFS](#) or, [NFS](#).

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.5.2 Before you build clusters

- Ensure that all cluster nodes are up and running.
- Verify that all inventory files are updated.
- If the 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 inventory file is case-sensitive. Follow the format provided in the sample file link.
- If [NFS](#) or [BeeGFS](#) are required on the cluster, run `storage.yml`.

Note:

- The inventory file accepts both IPs and FQDNs as long as they can be resolved by DNS.
 - In a multi-node setup, IP's cannot be listed as a control plane and a cluster node. That is, don't include the `kube_control_plane` IP address in the compute group. In a single node setup, the compute node and the `kube_control_plane` must be the same.
-

- Users should also ensure that all repos are available on the cluster nodes running RHEL.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.5.3 Building clusters

1. In the `input/omnia_config.yml`, `input/security_config.yml`, `input/telemetry_config.yml` and [optional] `input/storage_config.yml` files, provide the [required details](#).
2. Create an inventory file in the `omnia` folder. Check out the [sample inventory for more information](#). If a hostname is used to refer to the target nodes, ensure that the domain name is included in the entry. IP addresses are also accepted in the inventory file.

Hostname requirements

- 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: `hostname000000x.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 cluster node to 'node000001.omnia.test'. Omnia appends 6 digits to the hostname to individually name each target node.

Note:

¹ Boolean parameters do not need to be passed with double or single quotes.

- RedHat nodes that are not configured by Omnia need to have a valid subscription. To set up a subscription, [click here](#).
 - 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. `omnia.yml` is a wrapper playbook comprising of:

- i. `security.yml`: This playbook sets up centralized authentication (LDAP/FreeIPA) on the cluster. For more information, [click here](#).
- ii. `storage.yml`: This playbook sets up storage tools like [BeeGFS](#) and [NFS](#).
- iii. `scheduler.yml`: This playbook sets up job schedulers (Slurm or Kubernetes) on the cluster.
- iv. `telemetry.yml`: This playbook sets up [Omnia telemetry](#) and/or [iDRAC telemetry](#). It also installs [Grafana](#) and [Loki](#) as Kubernetes pods.

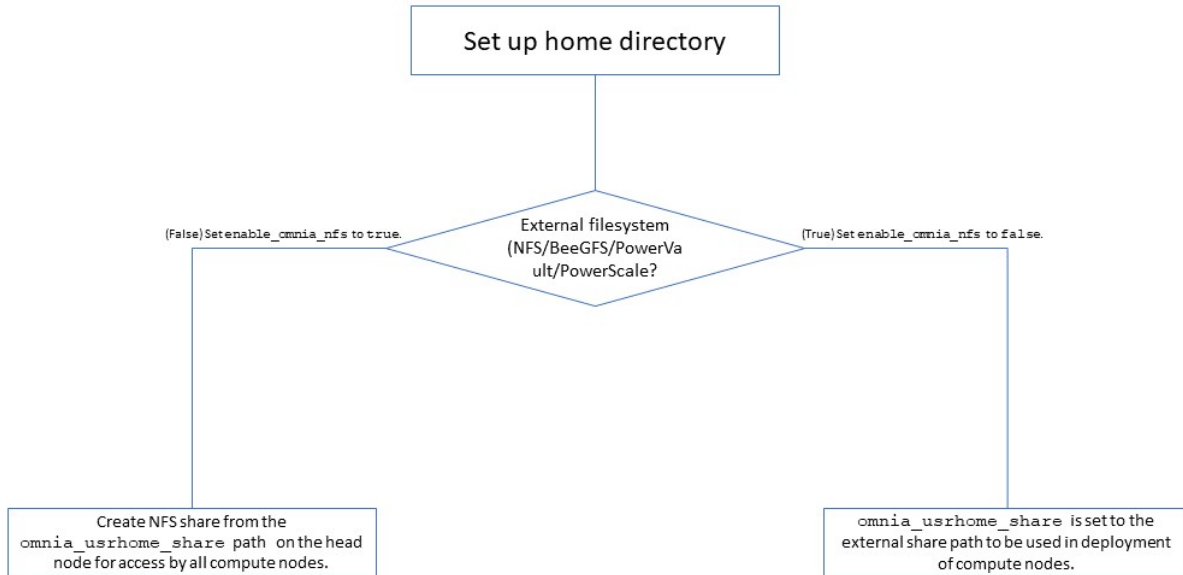
To run `omnia.yml`:

```
ansible-playbook omnia.yml -i inventory
```

Note:

- For a Kubernetes cluster installation, ensure that the inventory includes an `[etcd]` entry. etcd is a consistent and highly-available key value store used as Kubernetes' backing store for all cluster data. For more information, [click here](#).
 - 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.
 - Use the `ansible-vault view` or `edit` commands and not the `ansible-vault decrypt` or `encrypt` commands. If you have used the `ansible-vault decrypt` or `encrypt` commands, provide 644 permission to the parameter files.
-

Setting up a shared home directory



Users wanting to set up a shared home directory for the cluster can do it in one of two ways:

- **Using the head node as an NFS host:** Set `enable_omnia_nfs` (`input/omnia_config.yml`) to `true` and provide a share path which will be configured on all nodes in `omnia_usrhome_share` (`input/omnia_config.yml`). During the execution of `omnia.yml`, the NFS share will be set up for access by all cluster nodes.
- **Using an external filesystem:** Configure the external file storage using `storage.yml`. Set `enable_omnia_nfs` (`input/omnia_config.yml`) to `false` and provide the external share path in `omnia_usrhome_share` (`input/omnia_config.yml`). Run `omnia.yml` to configure access to the external share for deployments.

Slurm job based user access

To ensure security while running jobs on the cluster, users can be assigned permissions to access cluster 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>`.

Configuring UCX and OpenMPI on the cluster

If a local repository for UCX and OpenMPI has been configured on the cluster, the following configurations take place when running `omnia.yml` or `scheduler.yml`.

- UCX will be compiled and installed on the NFS share (based on the `client_share_path` provided in the `nfs_client_params` in `input/storage_config.yml`).
- If the cluster uses Slurm and UCX, OpenMPI is configured to compile with the UCX and Slurm on the NFS share (based on the `client_share_path` provided in the `nfs_client_params` in `input/storage_config.yml`).

- All corresponding compiled UCX and OpenMPI files will be saved to the <client_share_path>/compile directory on the nfs share.
- All corresponding UCX and OpenMPI executables will be saved to the <client_share_path>/benchmarks/ directory on the nfs share.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.5.4 Centralized authentication on the cluster

The security feature allows users to set up FreeIPA and LDAP to help authenticate into HPC clusters.

Configuring FreeIPA/LDAP security

Pre requisites

- Run `local_repo.yml` to create offline repositories of FreeIPA or OpenLDAP. If both were downloaded, ensure that the non-required system is removed from `input/software_config.json` before running `security.yml`. For more information, [click here](#).
- Enter the following parameters in `input/security_config.yml`.

Table 13: Parameters for Authentication

Parameter	Details
domain_name string Required	<ul style="list-style-type: none">• Sets the intended domain name.• If <code>dc=omnia,dc=test</code>, Provide <code>omnia.test</code>• If <code>dc=dell,dc=omnia,dc=com</code> Provide <code>dell.omnia.com</code> Default values: <code>omnia.test</code>

Table 14: Parameters for OpenLDAP configuration

Parameter	Details
ldap_connection_type string Required	For a TLS connection, provide a valid certification path. For an SSL connection, ensure port 636 is open. Choices: <ul style="list-style-type: none"> • TLS <- Default • SSL
tls_ca_certificate string Optional	File path pointing to the Certificate Authority (CA) issued certificate path. Certificate files should be saved with a .pem or .crt extension. If not provided, a self-signed certificate is generated by Omnia.
tls_certificate string Optional	File path pointing to the certificate used to authorize the LDAP server. Certificate files should be saved with a .pem or .crt extension.
tls_certificate_key string Optional	The private key that matches the LDAP certificate.
openldap_db_username string Required	The username used to manage the LDAP database. Default value: "admin"
openldap_db_password string Required	The password used to configure and manage the LDAP database. Ensure that this value is 8 characters long.
openldap_config_username string Required	The username used to configure the LDAP database. Default value: "admin"
openldap_config_password string Required	The password used to configure the LDAP database. Ensure that this value is 8 characters long.
openldap_monitor_password string Required	The password used to monitor the LDAP database. Ensure that this value is 8 characters long.
openldap_organization string Required	LDAP server is configured using organizations. They are necessary for user creation and group mapping. Default value: "omnia"
openldap_organizational_unit string Required	LDAP server is configured using organizations. They are necessary for user creation and group mapping. Default value: "People"

Table 15: Parameters for FreeIPA configuration

Parameter	Details
realm_name string Required	<ul style="list-style-type: none"> Sets the intended kerberos realm name. It is required for FreeIPA setups. A realm name is often, but not always the upper case version of the name of the DNS domain over which it presides. Default value: "OMNIA.TEST"
directory_manager_password string Required	<ul style="list-style-type: none"> The directory server operations require an administrative user. This user is referred to as the Directory Manager and has full access to the Directory for system management tasks and will be added to the instance of directory server created for IPA. The password must be at least 8 characters long. The password must not contain -, , ,
kerberos_admin_password string Required	<ul style="list-style-type: none"> kerberos_admin_password used by IPA admin user. The IPA server requires an administrative user, named 'admin'. The password must be at least 8 characters long. The password must not contain -, , ,

Create a new user on OpenLDAP

1. Create an LDIF file (eg: `create_user.ldif`) on the auth server containing the following information:

- DN: The distinguished name that indicates where the user will be created.
- objectClass: The object class specifies the mandatory and optional attributes that can be associated with an entry of that class. Here, the values are `inetOrgPerson`, `posixAccount`, and `shadowAccount`.
- UID: The username of the replication user.
- sn: The surname of the intended user.
- cn: The given name of the intended user.

Below is a sample file:

```
# User Creation
dn: uid=ldapuser,ou=People,dc=omnia,dc=test
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: ldapuser
sn: ldapuser
loginShell: /bin/bash
uidNumber: 2000
gidNumber: 2000
homeDirectory: /home/ldapuser
```

(continues on next page)

(continued from previous page)

```
shadowLastChange: 0
shadowMax: 0
shadowWarning: 0

# Group Creation
dn: cn=ldapuser,ou=Group,dc=omnia,dc=test
objectClass: posixGroup
cn: ldapuser
gidNumber: 2000
memberUid: ldapuser
```

Note: Avoid whitespaces when using an LDIF file for user creation. Extra spaces in the input data may be encrypted by OpenLDAP and cause access failures.

2. Run the command `ldapadd -D <enter admin binddn> -w <bind_password> -f create_user.ldif` to execute the LDIF file and create the account.
3. To set up a password for this account, use the command `ldappasswd -D <enter admin binddn> -w <bind_password> -S <user_dn>`. The value of `user_dn` is the distinguished name that indicates where the user was created. (In this example, `ldapuser,ou=People,dc=omnia,dc=test`)

Configuring login node security

Prerequisites

- Run `local_repo.yml` to create an offline repository of all utilities used to secure the login node. For more information, [click here](#).

Enter the following parameters in `input/login_node_security_config.yml`.

Variable	Details
max_failures integer Optional	The number of login failures that can take place before the account is locked out. Default values: 3
failure_reset_interval integer Optional	Period (in seconds) after which the number of failed login attempts is reset. Min value: 30; Max value: 60. Default values: 60
lockout_duration integer Optional	Period (in seconds) for which users are locked out. Min value: 5; Max value: 10. Default values: 10
session_timeout integer Optional	User sessions that have been idle for a specific period can be ended automatically. Min value: 90; Max value: 180. Default values: 180
alert_email_address string Optional	Email address used for sending alerts in case of authentication failure. When blank, authentication failure alerts are disabled. Currently, only one email ID is accepted.
user string Optional	Access control list of users. Accepted formats are <code>username@ip</code> (<code>root@1.2.3.4</code>) or username (<code>root</code>). Multiple users can be separated using whitespaces.
allow_deny string Optional	This variable decides whether users are to be allowed or denied access. Ensure that AllowUsers or DenyUsers entries on sshd configuration file are not commented. Choices: <ul style="list-style-type: none"> • <code>allow</code> <- Default • <code>deny</code>
restrict_program_support boolean Optional	This variable is used to disable services. Root access is mandatory. Choices: <ul style="list-style-type: none"> • <code>false</code> <- Default • <code>true</code>
restrict_softwares string Optional	List of services to be disabled (Comma-separated). Example: <code>'telnet,lpd,bluetooth'</code> Choices: <ul style="list-style-type: none"> • <code>telnet</code> • <code>lpd</code> • <code>bluetooth</code> • <code>rlogin</code> • <code>rexec</code>

Installing LDAP Client

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

FreeIPA installation on the NFS node

IPA services are used to provide account management and centralized authentication.

To customize your installation of FreeIPA, enter the following parameters in `input/security_config.yml`.

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 <code>input/security_config.yml</code> .
ipa_server_hostname	The hostname of the IPA server	The hostname can be found on the control plane.
domain_name	Domain name	The domain name can be found in the file <code>input/security_config.yml</code> .
ipa_server_ip	The IP address of the IPA server	The IP address can be found on the IPA server on the control plane using <code>ip a</code> . This IP address should be accessible from the NFS node.

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_ipaddress=""
```

Hostname requirements

- 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: `hostname000000x.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 cluster node to ‘node000001.omnia.test’. Omnia appends 6 digits to the hostname to individually name each target node.

Note: Use the format specified under [NFS inventory in the Sample Files](#) for inventory.

Running the security role

Run:

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

The inventory should contain `auth_server` as per the inventory file in [samplefiles](#). The inventory file is case-sensitive. Follow the format provided in the sample file link.

- Do not include the IP of the control plane or local host in the `auth_server` group in the passed inventory.
- To customize the security features on the login node, fill out the parameters in `input/login_node_security_config.yml`.
- If a subsequent run of `security.yml` fails, the `security_config.yml` file will be unencrypted.

Caution: No users will be created by Omnia.

Setting up Passwordless SSH

Once user accounts are created, admins can enable passwordless SSH for users to run HPC jobs on the cluster nodes.

Note: Once user accounts are created on the auth server, use the accounts to login to the cluster nodes to reset the password and create a corresponding home directory.

To customize your setup of passwordless ssh, input parameters in `input/passwordless_ssh_config.yml`.

Parameter	Details
user_name string Required	The list of users that requires password-less SSH. Separate the list of users using a comma. Eg: <code>user1,user2,user3</code>
authentication_type string Required	Indicates whether LDAP or FreeIPA is in use on the cluster. Choices: <ul style="list-style-type: none"> • <code>freeipa</code> • <code>ldap</code> <- Default

Use the below command to enable passwordless SSH:

```
ansible-playbook user_passwordless_ssh.yml -i inventory
```

Where inventory follows the format defined under inventory file in the provided [Sample Files](#). The inventory file is case-sensitive. Follow the format provided in the sample file link.

Caution: Do not run `ssh-keygen` commands after passwordless SSH is set up on the nodes.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.5.5 Granting Kubernetes access

Omnia grants Kubernetes node access to users defined on the kube_control_plane using the k8s_access.yml play-book.

Prerequisites

- Ensure the Kubernetes cluster is up and running.

Input parameters

- Update the variable user_name, in the input/k8s_access_config.yml file with a comma-separated list of users.

Parameter	Details
user_name String Required	<ul style="list-style-type: none"> – A comma-separated list of users to whom access must be granted. – Every user defined here must have a home directory configured on the kube_control_plane. – Sample values: user1 or user1,user2, user3.

- Verify that all intended users have a home directory (in the format /home/<user_name>) set up on the kube_control_plane.
- Job access is granted based on the values provided in resources and verbs variables in scheduler/roles/k8s_access/template/role.yml.j2. These values cannot be modified.
 - resources are a list of kubernetes objects or entities that are used to define, configure, and manage applications or infrastructure within a Kubernetes cluster. Possible values include ["pods", "services", "deployments", "jobs"].
 - verbs are a list of actions that can be taken on the resources. Possible values are ["create", "get", "list", "update", "delete"].
- The passed inventory should contain a defined kube_control_plane.

```
[auth_server]

#node12

#AI Scheduler: Kubernetes

[kube_control_plane]

# node1

[kube_node]
```

(continues on next page)

(continued from previous page)

```
# node2
# node3
# node4
# node5
# node6
```

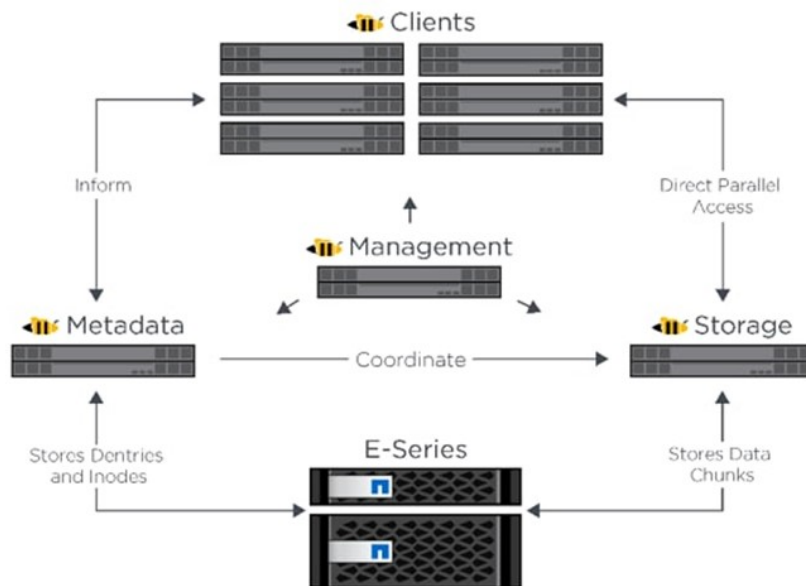
To run the playbook, use the below command:

```
cd scheduler
ansible-playbook -i inventory k8s_access.yml
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.5.6 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

- Ensure that the BeeGFS server is set up using the [linked steps](#).
- Ensure that a `connAuthFile` is configured on the server as explained [here](#)

Caution: Configuring a `connAuthFile` is now mandatory. Services will no longer start if a `connAuthFile` is not configured

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

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

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`

Note: BeeGFS services over RDMA 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 by checking our [support matrix](#).

Installing the BeeGFS client via Omnia

After the required parameters are filled in `input/storage_config.yml`, Omnia installs BeeGFS on all nodes while executing the `storage.yml` playbook.

Caution: Do not remove or comment any lines in the `input/storage_config.yml` file.

Table 16: Parameters for storage

Variables	Details
nfs_client_params JSON List Required	<ul style="list-style-type: none"> This JSON list contains all parameters required to set up NFS. For a bolt-on set up where there is a pre-existing NFS export, set <code>nfs_server</code> to <code>false</code>. When <code>nfs_server</code> is set to <code>true</code>, an NFS share is created on the control plane for access by all cluster nodes. For more information on the different kinds of configuration available, click here.
beegfs_rdma_support boolean Optional	This variable is used if user has RDMA-capable network hardware (e.g., InfiniBand) Choices: <ul style="list-style-type: none"> <code>false</code> <- Default <code>true</code>
beegfs_ofed_kernel_modules_path string Optional	<ul style="list-style-type: none"> The path where separate OFED kernel modules are installed. Ensure that the path provided here exists on all target nodes. Default value: <code>"/usr/src/ofa_kernel/default/include"</code>
beegfs_mgmt_server string Required	BeeGFS management server IP. <hr/> Note: The provided IP should have an explicit BeeGFS management server running . <hr/>
beegfs_mounts string Optional	BeeGfs-client file system mount location. If <code>storage.yml</code> is being used to change the BeeGFS mounts location, set <code>beegfs_unmount_client</code> to <code>true</code>. Default value: <code>"/mnt/beegfs"</code>
beegfs_unmount_client boolean Optional	Changing this value to <code>true</code> will unmount running instance of BeeGFS client and should only be used when decommissioning BeeGFS, changing the mount location or changing the BeeGFS version. Choices: <ul style="list-style-type: none"> <code>false</code> <- Default <code>true</code>
beegfs_version_change boolean Optional	Use this variable to change the BeeGFS version on the target nodes. Choices: <ul style="list-style-type: none"> <code>false</code> <- Default <code>true</code>
ansible_config_file_path string Required	<ul style="list-style-type: none"> Path to directory hosting ansible config file (ansible.cfg file) This directory is on the host running ansible, if ansible is installed using <code>dnf</code> If ansible is installed using <code>pip</code>, this path should be set

Note:

- BeeGFS client-server communication can take place over TCP or RDMA. If RDMA support is required, set `beegfs_rdma_support` should be set to true. Also, OFED should be installed on all cluster nodes.
 - For BeeGFS communication happening over RDMA, the `beegfs_mgmt_server` should be provided with the Infiniband IP of the management server.
 - The parameter `inventory` refers to the [inventory file](#) listing all relevant nodes.)
-

If `input/storage_config.yml` is populated before running `omnia.yml`, BeeGFS client will be set up during the run of `omnia.yml`.

If `omnia.yml` is not leveraged to set up BeeGFS, run the `storage.yml` playbook :

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

Note: Once BeeGFS is successfully set up, set `enable_omnia_nfs` (`input/omnia_config.yml`) to false and `omnia_usrhome_share` (`input/omnia_config.yml`) to an accessible share path in BeeGFS to use the path across the cluster for deployments.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@del.com.

2.5.7 NFS

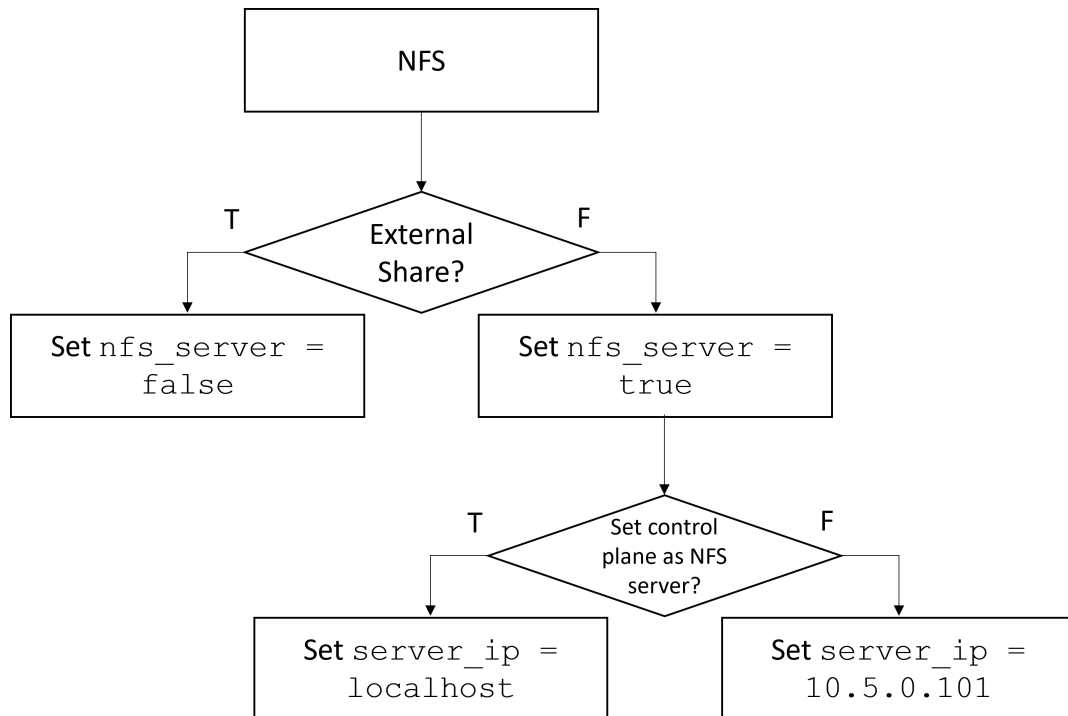
Network File System (NFS) is a networking protocol for distributed file sharing. A file system defines the way data in the form of files is stored and retrieved from storage devices, such as hard disk drives, solid-state drives and tape drives. NFS is a network file sharing protocol that defines the way files are stored and retrieved from storage devices across networks.

Note: NFS is a mandatory feature for all clusters set up by Omnia.

Pre requisites

- NFS is set up on Omnia clusters based on the inputs provided in `input/storage_config.yml`.

Parameter	Details
nfs_client_params JSON List Required	<ul style="list-style-type: none"> - This JSON list contains all parameters required to set up NFS. - For a bolt-on set up where there is a pre-existing NFS server, set <code>nfs_server</code> to <code>false</code>. - When <code>nfs_server</code> is set to <code>true</code>, an NFS share is created on a server IP in the cluster for access by all other cluster nodes. - Ensure that the value of <code>share_path</code> in <code>input/omnia_config.yml</code> matches at least one of the <code>client_share_path</code> values in the JSON list provided. - For more information on the different kinds of configuration available, click here.



- The fields listed in `nfs_client_params` are:
 - * **server_ip**: IP of the intended NFS server. To set up an NFS server on the control plane, use the value `localhost`. Use an IP address to configure access anywhere else.
 - * **server_share_path**: Folder on which the NFS server mounted.
 - * **client_share_path**: Target directory for the NFS mount on the client. If left empty, the respective `server_share_path` value will be taken for `client_share_path`.
 - * **nfs_server**: Indicates whether an external NFS server is available (`false`) or an NFS server will need to be created (`true`).
 - * **slurm_share**: Indicates that the target cluster uses Slurm.

* **k8s_share**: Indicates that the target cluster uses Kubernetes.

Note: To install any Benchmarking software like UCX or OpenMPI, at least **slurm_share** or **k8s_share** should be set to true. If both are set to true, a higher precedence is given to **slurm_share**.

To configure all cluster nodes to access a single external NFS server export, use the below sample:

```
- { server_ip: 10.5.0.101, server_share_path: "/mnt/share", client_share_path: "/"
  ↪home", client_mount_options: "nosuid,rw,sync,hard", nfs_server: true, slurm_
  ↪share: true, k8s_share: true }
```

To configure the cluster nodes to access a new NFS server on the control plane as well as an external NFS server, use the below example:

```
- { server_ip: localhost, server_share_path: "/mnt/share1", client_share_path: "/"
  ↪home", client_mount_options: "nosuid,rw,sync,hard", nfs_server: true, slurm_
  ↪share: true, k8s_share: true }
- { server_ip: 198.168.0.1, server_share_path: "/mnt/share2", client_share_path: "/"
  ↪mnt/mount2", client_mount_options: "nosuid,rw,sync,hard", nfs_server: false,
  ↪slurm_share: true, k8s_share: true }
```

To configure the cluster nodes to access new NFS server exports on the cluster nodes, use the below sample:

```
- { server_ip: 198.168.0.1, server_share_path: "/mnt/share1", client_share_path: "/"
  ↪mnt/mount1", client_mount_options: "nosuid,rw,sync,hard", nfs_server: false,
  ↪slurm_share: true, k8s_share: true }
- { server_ip: 198.168.0.2, server_share_path: "/mnt/share2", client_share_path: "/"
  ↪mnt/mount2", client_mount_options: "nosuid,rw,sync,hard", nfs_server: false,
  ↪slurm_share: true, k8s_share: true }
```

- Ensure that an NFS local repository is created by including {"name": "nfs"}, in input/software_config.json. For more information, [click here](#).
- If the intended cluster will run Slurm, set the value of slurm_installation_type in input/omnia_config.yml to nfs_share.
- If an external NFS share is used, 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.
- Omnia supports all NFS mount options. Without user input, the default mount options are no-suid,rw,sync,hard,intr.

Running the playbook

Run the storage.yml playbook :

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

Use the linked [inventory file](#) for the above playbook.

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

Caution:

- 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
```

- When `nfs_server` is false, enable the following services after configuration 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

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.6 Installing AI tools

AI (Artificial Intelligence) tools are software applications or systems that use AI technologies such as machine learning, natural language processing (NLP), computer vision, and deep learning to perform various tasks autonomously or with human interaction. These tools are designed to mimic human intelligence and can be used across different industries and domains for purposes such as automation, data analysis, decision-making, and more.

2.6.1 Setup Jupyterhub

Using Helm charts, Omnia can install Jupyterhub on Kubernetes clusters. Once Jupyterhub is deployed, log into the UI to create your own notebook servers. For more information, [click here](#).

Prerequisites

- Ensure the kubernetes cluster is setup and working.
- Ensure the inventory file includes a `kube_node` group listing all cluster nodes.
- Review the `tools/jupyter_config.yml` file to ensure that the deployment meets your requirements. If not, modify the file.
- Ensure that a local Jupyterhub repository is created using [the local repository script](#).
- Omnia deploys the `quay.io/jupyterhub/k8s-singleuser-sample:3.2.0` image irrespective of whether the intended notebooks are CPU-only, NVidia GPU, or AMD GPU. To use a custom image, modify the `omnia/tools/roles/jupyter_config.yml` file.
- Ensure that NFS has been deployed on the cluster using `storage.yml` followed by `scheduler.yml` or `omnia.yml`. Verify that the required NFS storage provisioner is deployed using the below command:

```
[root@node3 ~]# kubectl get pod -A
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
default	nfs-omnia-nfs-subdir-external-provisioner-54785fccd-9mp8z	1/1	Running	1 (12m ago)	3h24m

- Verify that the default storage class is set to nfs_client for dynamic persistent volume provisioning.

```
[root@node3 ~]# kubectl get storageclass
```

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	AGE
nfs-client (default)	cluster.local/nfs-omnia-nfs-subdir-external-provisioner	Delete	Immediate	true	17h

Deploying Jupyterhub

1. Change directories to the tools folder:

```
cd tools
```

2. Run the jupyterhub.yml playbook using:

```
ansible-playbook jupyterhub.yml -i inventory
```

Note: The default namespace for deployment is jupyterhub.

Accessing the Jupyterhub UI

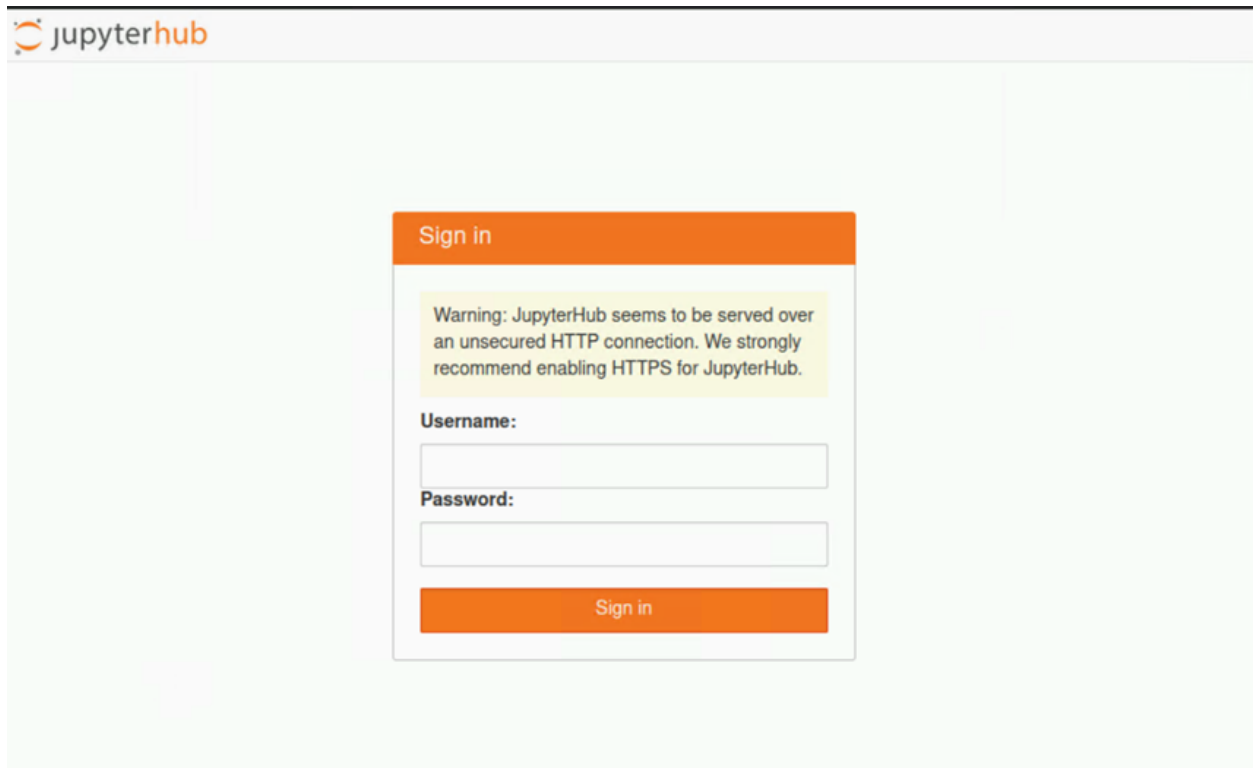
1. Verify that the Jupyterhub service is running using metallb loadbalancer.
2. Find the IP address of the Jupyterhub service using:

```
root@omnianode0000x:/usr/local# kubectl get svc -A
```

NAMESPACE	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
default	kubernetes	ClusterIP	xx.xx.xx.xx	<none>	443/TCP
jupyterhub	hub	ClusterIP	xx.xx.xx.xx	<none>	8081/TCP
jupyterhub	proxy-api	ClusterIP	xx.xx.xx.xx	<none>	8001/TCP
jupyterhub	proxy-public	LoadBalancer	xx.xx.xx.xx	xx.xx.xx.xx	80:31134/TCP

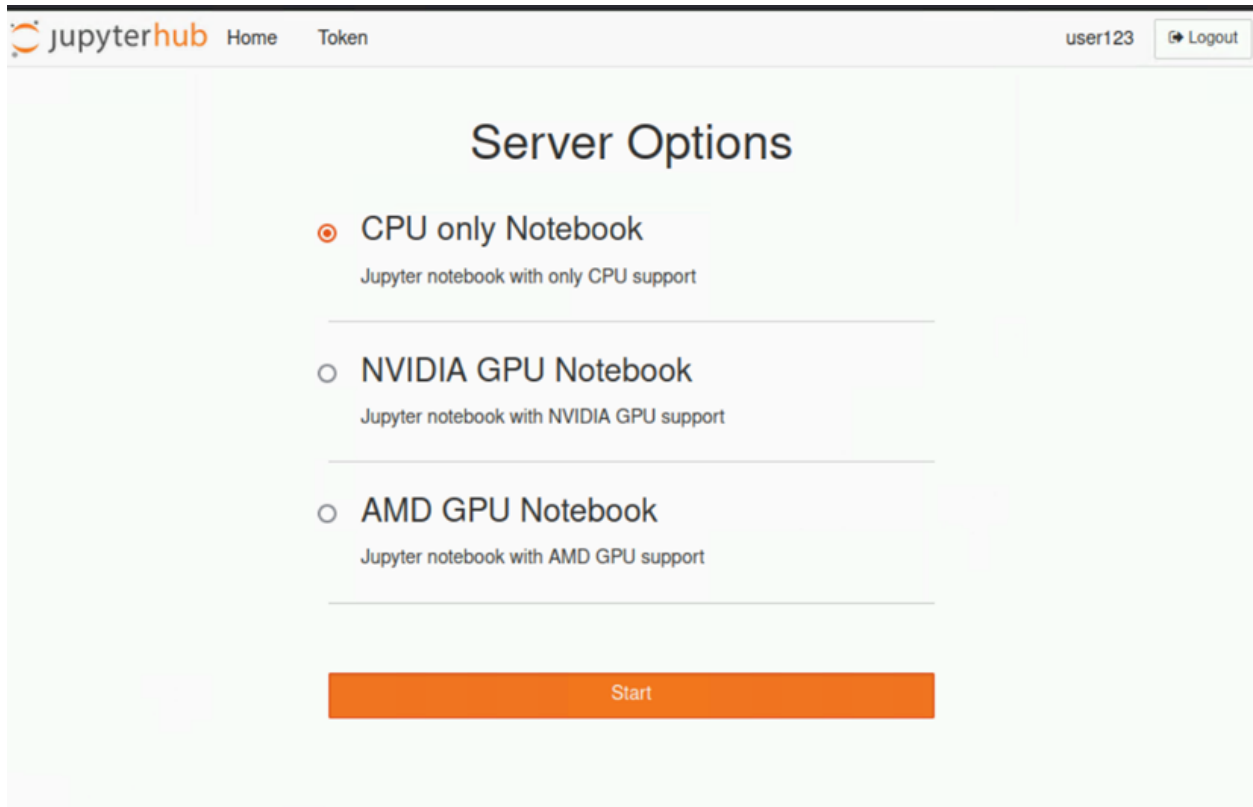
The IP address is listed against proxy-public under External IP.

3. The Jupyterhub GUI should be accessible from the control plane GUI via the external IP mentioned above. Use any browser to log in with user credentials.

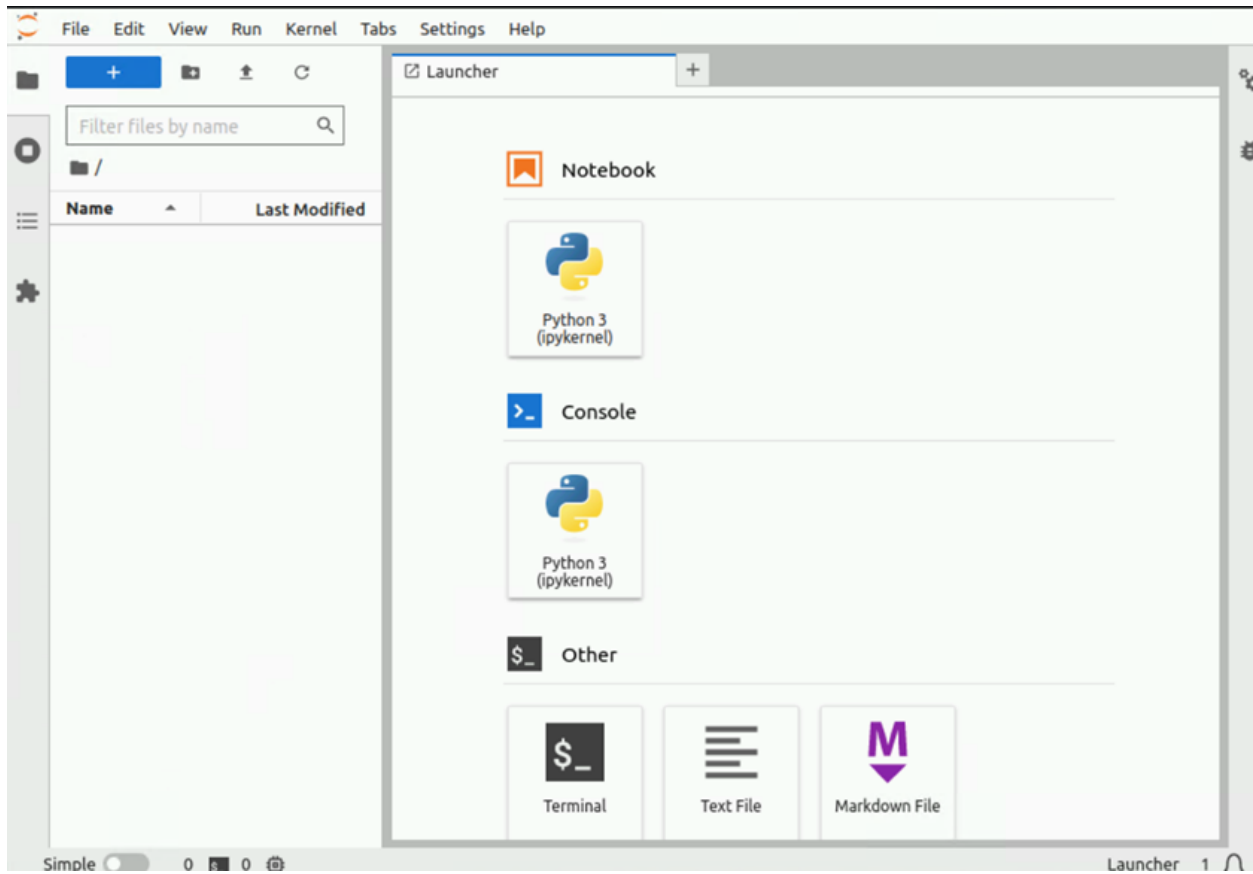


The image shows the JupyterHub sign-in interface. At the top left is the JupyterHub logo. The main content area is a light green box. In the center is a white box with an orange header that says "Sign in". Below the header is a yellow warning box with the text: "Warning: JupyterHub seems to be served over an unsecured HTTP connection. We strongly recommend enabling HTTPS for JupyterHub." Below the warning are two input fields: "Username:" and "Password:". At the bottom of the white box is an orange button labeled "Sign in".

4. Choose your preferred notebook server option and click **Start**. A pod will be created for the user. Available server options will depend on the user logging in.



The image shows the JupyterHub "Server Options" page. At the top is a navigation bar with the JupyterHub logo, "Home", "Token", "user123", and a "Logout" button. The main heading is "Server Options". Below it are three radio button options: "CPU only Notebook" (selected), "NVIDIA GPU Notebook", and "AMD GPU Notebook". Each option has a description below it: "Jupyter notebook with only CPU support", "Jupyter notebook with NVIDIA GPU support", and "Jupyter notebook with AMD GPU support" respectively. At the bottom is a large orange button labeled "Start".



Stopping the Notebook server

1. Click **File > Hub Control Plane**.
2. Select **Stop Server**.

Note: Stopping the notebook server only terminates the user pod. The users data persists and can be accessed by login in and starting the notebook server again.

Redeploy Jupyterhub with new configurations

1. Update the tools/jupyter_config.yml file with the new configuration.
2. Re-run the jupyterhub.yml playbook.

```
cd tools
ansible-playbook jupyterhub.yml -i inventory
```

Clearing Jupyterhub configuration

Clear the existing configuration by running the below command:

```
kubectl delete ns jupyterhub
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.6.2 Setup Kubeflow

Kubeflow is an open-source platform for machine learning and MLOps on Kubernetes introduced by Google.

Commands to install Kubeflow:

```
ansible-playbook tools/kubeflow.yml -i inventory
```

Note: When the Internet connectivity is unstable or slow, it may take more time to pull the images to create the Kubeflow containers. If the time limit is exceeded, the **Apply Kubeflow configurations** task may fail.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.6.3 Setup vLLM

vLLM is a fast and easy-to-use library for LLM inference and serving. It is seamlessly integrated with popular HuggingFace models. It is also compatible with OpenAI API servers and GPUs (Both NVIDIA and AMD). vLLM 0.2.4 and above supports model inferencing and serving on AMD GPUs with ROCm. At the moment AWQ quantization is not supported in ROCm, but SqueezeLLM quantization has been ported. Data types currently supported in ROCm are FP16 and BF16.

For NVidia, vLLM is a Python library that also contains pre-compiled C++ and CUDA (12.1) binaries.

With an Ansible script, deploy vLLM on both the kube_node and kube_control_node. After the deployment of vLLM, access the vllm container (AMD GPU) and import the vLLM Python package (NVIDIA GPU). For more information, [click here](#)

Note: This playbook was validated using Ubuntu 22.04 and RHEL 8.8.

Pre requisites

- Ensure nerdctl is available on all cluster nodes.
- Only AMD GPUs from the MI200s (gfx90a) are supported.
- For nodes using NVidia, ensure that the GPU has a compute capacity that is higher than 7 (Eg: V100, T4, RTX20xx, A100, L4, H100, etc).
- Ensure the kube_node, kube_control_node is setup and working. If NVidia or AMD GPU acceleration is required for the task, install the NVidia (with containerd) or AMD ROCm GPU drivers during provisioning.
- Use local_repo.yml to create an offline vLLM repository. For more information, [click here](#).

[Optional prerequisites]

- Ensure the system has enough available space. (Approximately 100GiB is required for the vLLM image. Any additional scripting will take disk capacity outside the image.)
- Ensure the passed inventory file has a kube_control_plane and kube_node_group listing all cluster nodes.
- Update the /input/software_config.json file with the correct vLLM version required. The default value is vllm-v0.2.4 for AMD container and vllm latest for NVidia.
- Omnia deploys the vLLM pip installation for NVidia GPU, or embeddedllminfo/vllm-rocm:vllm-v0.2.4 container image for AMD GPU.
- Nerdctl does not support mounting directories as devices because it is not a feature of containerd (The runtime that nerdctl uses). Individual files need to be attached while running nerdctl.

Deploying vLLM

1. Change directories to the tools folder:

```
cd tools
```

2. Run the vllm.yml playbook using:

```
ansible-playbook vllm.yml -i inventory
```

The default namespace is for deployment is vLLM.

Accessing the vLLM (AMD)

1. Verify that the vLLM image is present in the container engine images:

```
nerdctl images | grep vllm
```

2. Run the container image using modifiers to customize the run:

```
nerdctl run -it --network=host --group-add=video --ipc=host --cap-add=SYS_PTRACE --
↪security-opt seccomp=unconfined --device /dev/kfd --device /dev/dri/card0 --
↪device /dev/dri/card1 --device /dev/dri/renderD128 -v /opt/omnia:/app/model_
↪embeddedllminfo/vllm-rocm:vllm-v0.2.4
```

3. To enable an endpoint, [click here](#).

Accessing the vLLM (NVIDIA)

1. Verify that the vLLM package is installed:

```
python3.9 -c "import vllm; print(vllm.__version__)"
```

2. Use the package within a python script as demonstrated in the sample below:

```
from vllm import LLM, SamplingParams

prompts = [
    "Hello, my name is",
    "The president of the United States is",
    "The capital of France is",
    "The future of AI is",
]

sampling_params = SamplingParams(temperature=0.8, top_p=0.95)
llm = LLM(model="mistralai/Mistral-7B-v0.1")

outputs = llm.generate(prompts, sampling_params)

# Print the outputs.
for output in outputs:
    prompt = output.prompt
    generated_text = output.outputs[0].text
    print(f"Prompt: {prompt!r}, Generated text: {generated_text!r}")
```

3. To enable an endpoint, [click here](#).

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.6.4 Setup PyTorch

PyTorch is a popular open-source deep learning framework, renowned for its dynamic computation graph that enhances flexibility and ease of use, making it a preferred choice for researchers and developers. With strong community support, PyTorch facilitates seamless experimentation and rapid prototyping in the field of machine learning.

Prerequisites

- Ensure `nerdctl` is available on all cluster nodes.
- If GPUs are present on the target nodes, install NVidia CUDA (with `containerd`) or AMD Rocm drivers during provisioning. CPUs do not require any additional drivers.
- Use `local_repo.yml` to create an offline PyTorch repository. For more information, [click here](#).

[Optional prerequisites]

- Ensure the system has enough space.
- Ensure the passed inventory file includes a `kube_control_plane` and a `kube_node_group` listing all cluster nodes. [Click here](#) for a sample file.
- Nerdctl does not support mounting directories as devices because it is not a feature of `containerd` (The runtime that `nerdctl` uses). Individual files need to be attached while running `nerdctl`.

Deploying PyTorch

1. Change directories to the `tools` folder:

```
cd tools
```

2. Run the `pytorch.yml` playbook:

```
ansible-playbook pytorch.yml -i inventory
```

Accessing PyTorch (CPU)

1. Verify that the PyTorch image present in container engine images:

```
nerdctl images
```

2. Use the container image per your needs:

```
nerdctl run -it --rm pytorch/pytorch:latest
```

For more information, [click here](#).

Accessing PyTorch (AMD)

1. Verify that the PyTorch image present in container engine images:

```
nerdctl images
```

2. Use the container image per your needs:

```
nerdctl run -it --cap-add=SYS_PTRACE --security-opt seccomp=unconfined --device=/  
↪dev/kfd --device /dev/dri/card0 --device /dev/dri/card1 --device /dev/dri/card2 --  
↪device /dev/dri/renderD128 --device /dev/dri/renderD129 --group-add video --  
↪ipc=host --shm-size 8G rocm/pytorch:latest
```

For more information, [click here](#).

Accessing PyTorch (NVidia)

1. Verify that the PyTorch image present in container engine images:

```
nerdctl images
```

2. Use the container image per your needs:

```
nerdctl run --gpus all -it --rm nvcr.io/nvidia/pytorch:23.12-py3
```

For more information, [click here](#).

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.6.5 Setup TensorFlow

TensorFlow is a widely-used open-source deep learning framework, recognized for its static computation graph that optimizes performance and scalability, making it a favored choice for deploying machine learning models at scale in various industries.

With an Ansible script, deploy TensorFlow on both `kube_nodes` and the `kube_control_node`. After the deployment of TensorFlow, you gain access to the TensorFlow container.

Prerequisites

- Ensure `nerdctl` is available on all cluster nodes.
- If GPUs are present on the target nodes, install NVidia CUDA (with `containerd`) or AMD ROCm drivers during provisioning. CPUs do not require any additional drivers.
- Use `local_repo.yml` to create an offline TensorFlow repository. For more information, [click here](#).

[Optional prerequisites]

- Ensure the system has enough space.
- Ensure the passed inventory file includes a `kube_control_plane` and a `kube_node_group` listing all cluster nodes. [Click here](#) for a sample file.
- Nerdctl does not support mounting directories as devices because it is not a feature of `containerd` (The runtime that `nerdctl` uses). Individual files need to be attached while running `nerdctl`.
- Container Network Interface should be enabled with `nerdctl`.

Deploying TensorFlow

1. Change directories to the `tools` folder:

```
cd tools
```

2. Run the `tensorflow.yml` playbook:

```
ansible-playbook tensorflow.yml -i inventory
```

Accessing TensorFlow (CPU)

1. Verify that the tensorflow image present in container engine images:

```
nerdctl images
```

2. Use the container image per your needs:

```
nerdctl run -it --rm tensorflow/tensorflow
```

For more information, [click here](#).

Accessing TensorFlow (AMD)

1. Verify that the tensorflow image present in container engine images:

```
nerdctl images
```

2. Use the container image per your needs:

```
nerdctl run -it --network=host --device=/dev/kfd --device /dev/dri/card0 --device /  
↪dev/dri/card1 --device /dev/dri/card2 --device /dev/dri/renderD128 --device /dev/  
↪dri/renderD129 --ipc=host --shm-size 16G --group-add video --cap-add=SYS_PTRACE -  
↪-security-opt seccomp=unconfined rocm/tensorflow:latest
```

For more information, [click here](#).

Accessing TensorFlow (NVidia)

1. Verify that the tensorflow image present in container engine images:

```
nerdctl images
```

2. Use the container image per your needs:

```
nerdctl run --gpus all -it --rm nvcr.io/nvidia/tensorflow:23.12-tf2-py3
```

For more information, [click here](#).

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.6.6 Setup Kserve

Kserve is an open-source serving platform that simplifies the deployment, scaling, and management of machine learning models in production environments, ensuring efficient and reliable inference capabilities. For more information, [click here](#). Omnia deploys KServe (v0.11.0) on the kubernetes cluster. Once KServe is deployed, any inference service can be installed on the kubernetes cluster.

Prerequisites

- Ensure nerdctl and containerd is available on all cluster nodes.
- The cluster is deployed with Kubernetes.
- MetalLB pod is up and running to provide an external IP to istio-ingressgateway.
- The domain name on the kubernetes cluster should be **cluster.local**. The KServe inference service will not work with a custom `cluster_name` property on the kubernetes cluster.
- A local Kserve repository should be created using `local_repo.yml`. For more information, [click here](#).
- Ensure the passed inventory file includes a `kube_control_plane` and a `kube_node` listing all cluster nodes. [Click here](#) for a sample file.
- To access NVIDIA or AMD GPU acceleration in inferencing, Kubernetes NVIDIA or AMD GPU device plugins need to be installed during Kubernetes deployment. `kserve.yml` does not deploy GPU device plugins.

Deploy KServe

1. Change directories to tools.

```
cd tools
```

2. Run the kserve.yml playbook:

```
ansible-playbook kserve.yml -i inventory
```

Post deployment, the following dependencies are installed:

- Istio (version: 1.17.0)
- Certificate manager (version: 1.13.0)
- Knative (version: 1.11.0)

To verify the installation, run `kubectl get pod -A` and look for the namespaces: `cert-manager`, `istio-system`, `knative-serving`, and `kserve`.

```
root@sparknode1:/tmp# kubectl get pod -A
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
cert-manager	cert-manager-5d999567d7-mfgdk	1/1	Running	0	44h
cert-manager	cert-manager-cainjector-5d755dcf56-877dm	1/1	Running	0	44h
cert-manager	cert-manager-webhook-7f7b47c4d4-qzjst	1/1	Running	0	44h
default	model-store-pod	1/1	Running	0	43h
default	sklearn-pvc-predictor-00001-deployment-667d9f764c-clkbn	2/2	Running	0	43h
istio-system	istio-ingressgateway-79cc8bf885-lqgm7	1/1	Running	0	44h
istio-system	istiod-777dc7ffbc-b4plt	1/1	Running	0	44h
knative-serving	activator-59dff6d45c-28t2x	1/1	Running	0	44h
knative-serving	autoscaler-dbf4d8d66-4wj8f	1/1	Running	0	44h
knative-serving	controller-6bfd96676f-rdlxl	1/1	Running	0	44h
knative-serving	net-istio-controller-6ff9b86f6b-9trb8	1/1	Running	0	44h
knative-serving	net-istio-webhook-845d4d74b4-r9d8z	1/1	Running	0	44h
knative-serving	webhook-678bd64859-q4ghb	1/1	Running	0	44h
kserve	kserve-controller-manager-f9c5984c5-xz7lp	2/2	Running	0	44h

Deploy inference service

Prerequisites

- To deploy a model joblib file with PVC as model storage, [click here](#)

- Verify that the inference service is up and running using the command: `kubectl get isvc -A`:

```
root@sparknode1:/tmp# kubectl get isvc -A
```

NAMESPACE	NAME	URL	READY	PREV
↪ LATEST	PREVROLLEDOUTREVISION	LATESTREADYREVISION	AGE	
default	sklearn-pvc	http://sklearn-pvc.default.example.com	True	
↪ 100		sklearn-pvc-predictor-000001	9m18s	

- Pull the intended inference model and the corresponding runtime-specific images into the nodes.
- As part of the deployment, Omnia deploys [standard model runtimes](#). If a custom model is deployed, deploy a custom runtime first.
- To avoid problems with image to digest mapping when pulling inference runtime images, [click here](#).

Access the inference service

1. Use `kubectl get svc -A` to check the external IP of the service `istio-ingressgateway`.

```
root@sparknode1:/tmp# kubectl get svc -n istio-system
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
↪	AGE			
istio-ingressgateway	LoadBalancer	10.233.30.227	10.20.0.101	15021:32743/ ↪ TCP,80:30134/TCP,443:32241/TCP 44h
istiod	ClusterIP	10.233.18.185	<none>	15010/TCP, ↪ 15012/TCP,443/TCP,15014/TCP 44h
knative-local-gateway	ClusterIP	10.233.37.248	<none>	80/TCP 44h

2. To access inferencing from the ingressgateway with HOST header, run the below command from the `kube_control_plane` or `kube_node`:

```
curl -v -H "Host: <service url>" -H "Content-Type: application/json" "http://<istio-  
↪ ingress external IP>:<istio-ingress port>/v1/models/<model name>:predict" -d @./  
↪ iris-input.json
```

For example:

```
root@sparknode2:/tmp# curl -v -H "Host: sklearn-pvc.default.example.com" -H "Content-  
↪ Type: application/json" "http://10.20.0.101:80/v1/models/sklearn-pvc:predict" -d @./  
↪ iris-input.json
```

```
* Trying 10.20.0.101:80...
* Connected to 10.20.0.101 (10.20.0.101) port 80 (#0)
> POST /v1/models/sklearn-pvc:predict HTTP/1.1
> Host: sklearn-pvc.default.example.com
> User-Agent: curl/7.81.0
> Accept: */*
> Content-Type: application/json
> Content-Length: 76
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< content-length: 21
< content-type: application/json
< date: Sat, 16 Mar 2024 09:36:31 GMT
< server: istio-envoy
```

(continues on next page)

(continued from previous page)

```
< x-envoy-upstream-service-time: 7
<
* Connection #0 to host 10.20.0.101 left intact
{"predictions":[1,1]}
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.7 Adding new nodes

Provisioning the new node

A new node can be added using the following ways:

- If `pxe_mapping_file_path` is provided in `input/provision_config.yml`:
 - 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.

Note: When re-running `discovery_provision.yml` with a new mapping file, ensure that existing IPs from the current mapping file are not provided in the new mapping file. Any IP overlap between mapping files will result in PXE failure. This can only be resolved by running [the Clean Up script](#) followed by `discovery_provision.yml`.

- Run `discovery_provision.yml`:

```
ansible-playbook discovery_provision.yml
```

- Manually PXE boot the target servers after the `discovery_provision.yml` playbook (if `bmc_ip` is not provided in the mapping file) is executed and the target node lists as **booted** in [the nodeinfo table](#)

- When target nodes were discovered using BMC:
 - Run `discovery_provision.yml` once the node has joined the cluster using an IP that exists within the provided range.

```
ansible-playbook discovery_provision.yml
```

- When target nodes were discovered using `switch_based_details` in `input/provision_config.yml`:
 - Edit or append JSON list stored in `switch_based_details` in `input/provision_config.yml`.

Note:

- All ports residing on the same switch should be listed in the same JSON list element.
 - Ports configured via Omnia should be not be removed from `switch_based_details` in `input/provision_config.yml`.
-

- Run `discovery_provision.yml`.

```
ansible-playbook discovery_provision.yml
```

- Manually PXE boot the target servers after the `discovery_provision.yml` playbook is executed and the target node lists as **booted** in the `nodeinfo` table

Verify that the node has been provisioned successfully by [checking the Omnia nodeinfo table](#).

Adding new compute nodes to the cluster

1. Insert the new IPs in the existing inventory file following the below example.

Existing kubernetes inventory

```
[kube_control_plane]
10.5.0.101

[kube_node]
10.5.0.102
10.5.0.103

[auth_server]
10.5.0.101

[etcd]

10.5.0.110
```

Updated kubernetes inventory with the new node information

```
[kube_control_plane]
10.5.0.101

[kube_node]
10.5.0.102
10.5.0.103
10.5.0.105
10.5.0.106

[auth_server]
10.5.0.101

[etcd]

10.5.0.110
```

Existing Slurm inventory

```
[slurm_control_node]
10.5.0.101

[slurm_node]
10.5.0.102
10.5.0.103

[login]
```

(continues on next page)

(continued from previous page)

```
10.5.0.104
[auth_server]
10.5.0.101
```

Updated Slurm inventory with the new node information

```
[slurm_control_node]
10.5.0.101

[slurm_node]
10.5.0.102
10.5.0.103
10.5.0.105
10.5.0.106

[login]
10.5.0.104

[auth_server]
10.5.0.101
```

In the above examples, nodes 10.5.0.105 and 10.5.0.106 have been added to the cluster as compute nodes.

Note:

- The [etcd] group only supports an odd number of servers in the group.
 - Do not change the kube_control_plane/slurm_control_node/auth_server in the existing inventory. Simply add the new node information in the kube_node/slurm_node group.
 - When re-running omnia.yml to add a new node, ensure that the input/security_config.yml and input/omnia_config.yml are not edited between runs.
-

3. To install [security](#), [job scheduler](#) and storage tools ([NFS](#), [BeeGFS](#)) on the node, run omnia.yml:

```
ansible-playbook omnia.yml -i inventory
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.8 Re-provisioning the cluster

Pre-requisites

- Run the [delete node playbook](#). for every target node.

In the event that an existing Omnia cluster needs a different OS version or a fresh installation, the cluster can be re-provisioned.

If a re-deployment with no modifications are required

```
ansible-playbook discovery_provision.yml
```

Setting up the cluster

1. Insert the new IPs in the existing inventory file following the below example.

Existing kubernetes inventory

```
[kube_control_plane]
10.5.0.101

[kube_node]
10.5.0.102
10.5.0.103

[auth_server]
10.5.0.101

[etcd]
10.5.0.110
```

Updated kubernetes inventory with the new node information

```
[kube_control_plane]
10.5.0.101

[kube_node]
10.5.0.102
10.5.0.103
10.5.0.105
10.5.0.106

[auth_server]
10.5.0.101

[etcd]
10.5.0.110
```

Existing Slurm inventory

```
[slurm_control_node]
10.5.0.101

[slurm_node]
10.5.0.102
10.5.0.103

[login]
10.5.0.104

[auth_server]
10.5.0.101
```

Updated Slurm inventory with the new node information

```
[slurm_control_node]
10.5.0.101

[slurm_node]
```

(continues on next page)

(continued from previous page)

```
10.5.0.102
10.5.0.103
10.5.0.105
10.5.0.106
```

```
[login]
10.5.0.104
```

```
[auth_server]
10.5.0.101
```

In the above examples, nodes 10.5.0.105 and 10.5.0.106 have been added to the cluster as compute nodes.

Note:

- Do not change the kube_control_plane/slurm_control_node/auth_server in the existing inventory. Simply add the new node information in the kube_node/slurm_node group.
 - When re-running omnia.yml to add a new node, ensure that the input/security_config.yml and input/omnia_config.yml are not edited between runs.
-

3. To install security, job scheduler and storage tools (NFS, BeeGFS) on the node, run omnia.yml:

```
ansible-playbook omnia.yml -i inventory
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.9 Configuring switches

Note: Omnia supports only ethernet switches running OS10. Ethernet switches running sonic west will have be configured manually by users.

2.9.1 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)
- NDR Switches (32 ports)

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

Caution: Do not remove or comment any lines in the network/infiniband_edr_input.yml, network/infiniband_hdr_input.yml or network/infiniband_ndr_input.yml file.

Parameters	Details
enable_split_port boolean ¹ Required	Indicates whether ports are to be split. Choices: <ul style="list-style-type: none"> • false <- default • true
ib_split_ports string Optional	<ul style="list-style-type: none"> • Stores the split configuration of the ports. • For EDR and HDR switches, the accepted formats are : comma-separated (EX: “1,2”), ranges (EX: “1-10”), comma-separated ranges (EX: “1,2,3-8,9,10-12”) • For NDR switches, the accepted format is: 2/1, 2/2, 3/1 <hr/> <p>Note: The port prefix IB1 can be ignored when setting this value.</p> <hr/>
snmp_community_name string Optional	<p>The “SNMP community string” is like a user ID or password that allows access to a router’s or other device’s statistics.</p> <p>Default values: public</p>
cache_directory string Optional	Cache location used by OpenSM
log_directory string Optional	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 string Optional	<p>List of configuration lines to apply to the switch. # Example:</p> <p># mellanox_switch_config: # - Command 1 # - Command 2</p> <p>By default, the list is empty.</p>
ib 1/(1-xx) config string Optional	<p>Indicates the required state of ports 1-xx (depending on the value of 1/x).</p> <p>Default values: "no shutdown"</p>
save_changes_to_startup boolean ^{Page 100, 1} Optional	<p>Indicates whether the switch configuration is to persist across reboots.</p> <p>Choices:</p> <ul style="list-style-type: none"> • false <- default • true

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:

¹ Boolean parameters do not need to be passed with double or single quotes.

```
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:

- Currently, Omnia only supports the splitting of switch ports. Switch ports cannot be un-split using this script. For information on manually un-splitting ports, [click here](#).
 - 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.
 - The `ib_password` remains unchanged on switches that are in split-ready mode.
-

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 Infiniband switch.
-

Note:

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

- Passwords should be different from 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](#)
-

- 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/NDR

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.9.2 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.

Caution: Do not remove or comment any lines in the <code>network/ethernet_tor_input.yml</code> file.

Name	Details
os10_config string Required	Global configurations for the switch. Choices: <ul style="list-style-type: none"> • <code>interface vlan1 <- Default</code> • <code>exit</code>
breakout_value string 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. Choices: <ul style="list-style-type: none"> • <code>10g-4x <- Default</code> • <code>25g-4x</code> • <code>40g-1x</code> • <code>50g-2x</code> • <code>100g-1x</code>
snmp_trap_destination string 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 string Optional	An SNMP community string is a means of accessing statistics stored within a router or other device. Default values: <code>public</code>
ethernet 1/1/(1-52) config string Required	By default: <ul style="list-style-type: none"> • 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 <code>breakout_value</code> variable. • Update the individual interfaces of the Dell PowerSwitch S5232F-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 boolean ¹ 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. Choices: <ul style="list-style-type: none"> • <code>false <- Default</code> • <code>true</code>

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

Running the playbook:

¹ Boolean parameters do not need to be passed with double or single quotes.

```
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.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.9.3 Configuring ethernet switches (S5 series)

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

Caution: Do not remove or comment any lines in the `network/ethernet_sseries_input.yml` file.

Name	Details
os10_config string Required	Global configurations for the switch. Choices: <ul style="list-style-type: none"> • <code>interface vlan1 <- Default</code> • <code>exit</code>
breakout_value string 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. Choices: <ul style="list-style-type: none"> • <code>10g-4x <- Default</code> • <code>25g-4x</code> • <code>40g-1x</code> • <code>50g-2x</code> • <code>100g-1x</code>
snmp_trap_destination string 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 string Optional	An SNMP community string is a means of accessing statistics stored within a router or other device. Default values: <code>public</code>
ethernet 1/1/(1-34) config string Required	By default: <ul style="list-style-type: none"> • Port description is provided. • Each interface is set to “up” state. • The fanout/breakout mode for 1/1/1 to 1/1/31 is as per the value set in the <code>breakout_value</code> variable. • 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 boolean ¹ 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. Choices: <ul style="list-style-type: none"> • <code>false <- Default</code> • <code>true</code>

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

¹ Boolean parameters do not need to be passed with double or single quotes.

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.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.9.4 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.

Caution: Do not remove or comment any lines in the `network/ethernet_zseries_input.yml` file.

Name	Details
os10_config string Required	Global configurations for the switch. Choices: <ul style="list-style-type: none"> • <code>interface vlan1 <- Default</code> • <code>exit</code>
breakout_value string 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. Choices: <ul style="list-style-type: none"> • <code>10g-4x <- Default</code> • <code>25g-4x</code> • <code>40g-1x</code> • <code>50g-2x</code> • <code>100g-1x</code>
snmp_trap_destination string 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 string Optional	An SNMP community string is a means of accessing statistics stored within a router or other device. Default values: <code>public</code>
ethernet 1/1/(1-63) config string Required	By default: <ul style="list-style-type: none"> • Port description is provided. • Each interface is set to “up” state. • The fanout/breakout mode for 1/1/1 to 1/1/63 is as per the value set in the <code>breakout_value</code> variable. • 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 boolean ¹ 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. Choices: <ul style="list-style-type: none"> • <code>false <- Default</code> • <code>true</code>

- 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.

¹ Boolean parameters do not need to be passed with double or single quotes.

- 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.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.10 Configuring PowerVault

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`:

Caution: Do not remove or comment any lines in the `storage/powervault_input.yml` file.

Parameter	Details
powervault_protocol string Required	This variable indicates the network protocol used for data connectivity . Default values: sas
powervault_controller_mode string Required	This variable indicates the number of controllers available on the target powervault. Choices: <ul style="list-style-type: none"> • multi <- default • single
powervault_locale string Optional	Represents the selected language. Currently, only English is supported. Default values: English
powervault_system_name string Optional	The system name used to identify the PowerVault Storage device. The name should be less than 30 characters and must not contain spaces. Default values: Uninitialized_Name
powervault_snmp_notify_level string Required	Select the SNMP notification levels for PowerVault Storage devices. Default values: none
powervault_pool_type string Required	This variable indicates the kind of pool created on the target powervault. Choices: <ul style="list-style-type: none"> • linear <- default • virtual
powervault_raid_levels string Optional	Enter the required RAID levels and the minimum / maximum number of disks for each RAID levels. Choices: <ul style="list-style-type: none"> • raid1 <- default • raid5 • raid6 • raid10
powervault_disk_range string 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 sub-groups 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. Default values: 0.0-1
powervault_disk_group_name string Required	Specifies the disk group name Default values: omnia

2.10. Configuring PowerVault

109

powervault_volumes
string Required

Specify the volume details for powervault and NFS Server node. Multiple volumes can be defined as comma separated values. example: omnia homel, omnia

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.

Note: Once the storage is successfully set up, set `enable_omnia_nfs` (`input/omnia_config.yml`) to false and `omnia_usrhome_share` (`input/omnia_config.yml`) to an accessible share path in BeeGFS to use the path across the cluster for deployments.

Configuring NFS servers

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

Parameter	Details
powervault_ip string Optional	Mandatory field when nfs group is defined with an IP and omnia is required to configure nfs server. IP of Powervault connected to NFS Server should be provided. In a single run of omnia, only one NFS Server is configured. To configure multiple NFS Servers, add one IP in nfs group in a single run of omnia.yml and give variable values accordingly. To configure another nfs node, update variables and run <code>nfs_sas.yml</code>
powervault_volumes JSON list Required	Specify the volume details for powervault and NFS Server node For multiple volumes, list of json with volume details should be provided. <ul style="list-style-type: none"> • <code>server_share_path</code>: The path at which volume is mounted on nfs node • <code>server_export_options</code>: Default value is <code>rw,sync,no_root_squash</code> (unless specified otherwise). For a list of accepted options, click here • <code>client_shared_path</code>: The path at which volume is mounted on all nodes. This value is taken as <code>server_share_path</code> unless specified otherwise. • <code>client_mount_options</code>: Default value is <code>nosuid,rw,sync,hard,intr</code> (unless specified otherwise). For a list of accepted options, click here <p>Must specify atleast 1 volume</p> <p>Default values: `` - { name: omnia_home, server_share_path: /home/omnia_home, server_export_options: } ``</p>

Run the playbook:

```
cd storage
ansible-playbook nfs_sas.yml -i /root/inventory -e powervault_username="xxxxxx" -e
↪powervault_password="xxxxxx"
```

- Where the `inventory` refers to a list of all nodes in the format of [NFS server inventory file](#)
- To set up NFS client services, [click here](#)

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.11 Running HPC benchmarks on omnia clusters

2.11.1 Automate installation oneAPI on Intel processors for MPI jobs

This topic explains how to automatically update servers for MPI jobs.

Caution: oneAPI is not supported on Ubuntu clusters.

Pre-requisites

- `discovery_provision.yml` has been executed.
- The cluster has been set up with kubernetes.
- An Omnia **slurm** cluster has been set up by `omnia.yml` running with at least 2 nodes: 1 `slurm_control_node` and 1 `slurm_node`.
- A local repository has been set up by listing `{"name": "intel_benchmarks"}`, in `input/software_config.json` and running `local_repo.yml`. For more information, [click here](#).
- Verify that the target nodes are in the booted state. For more information, [click here](#).

To run the playbook:

```
cd benchmarks
ansible-playbook intel_benchmark.yml -i inventory
```

To execute multi-node jobs

- Make sure to have NFS shares on each node.
- Copy slurm script to NFS share and execute it from there.
- Load all the necessary modules using module load:

```
module load mpi
module load pmi/pmix-x86_64
module load mkl
```

- If the commands/batch script are to be run over TCP instead of Infiniband ports, include the below line:

```
export FI_PROVIDER=tcp
```

Job execution can now be initiated.

Note: Ensure `runme_intel64_dynamic` is downloaded before running this command.

```
srunk -N 2 /mnt/nfs_shares/appshare/mkl/2023.0.0/benchmarks/mp_linpack/runme_intel64_
↪dynamic
```

For a batch job using the same parameters, the script would be:

```
#!/bin/bash
#SBATCH --job-name=testMPI
#SBATCH --output=output.txt
#SBATCH --partition=normal
#SBATCH --nodelist=node00004.omnia.test,node00005.omnia.test

pwd; hostname; date
export FI_PROVIDER=tcp
module load pmi/pmix-x86_64
module use /opt/intel/oneapi/modulefiles
module load mkl
module load mpi

srun /mnt/appshare/benchmarks/mp_linpack/runme_intel64_dynamic
date
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.11.2 Open MPI AOCC HPL benchmark for AMD processors

This topic explains how to manually update servers for MPI jobs. To automatically install pmix and configure slurm, [click here](#).

Prerequisites

1. Provision the cluster and install slurm on all cluster nodes.
2. Dependent packages have to be installed on the cluster nodes using the following steps.:
 - i. Download the dependent packages on the control plane.:

- a. Create a package list.:

```
cd /install/post/otherpkgs/<os_version>/x86_64/custom_software/
cat openmpi.pkglist
```

- b. Enter the following contents into openmpi.pkglist:

```
custom_software/pmix-devel
custom_software/libevent-devel
```

- c. Download the packages:

```
cd packages
dnf download pmix-devel --resolve --alldeps
dnf download libevent-devel --resolve --alldeps
```

- ii. Push the packages to the cluster nodes:

- a. Update the package_list variable in the utils/os_package_update/package_update_config.yml file and save it.

```
package_list: "/install/post/otherpkgs/<os_version>/x86_64/custom_software/
↪openmpi.pkglist"
```


- b. Update the cluster nodes by running the `package_update.yml` playbook.

```
ansible-playbook package_update.yml
```

3. OpenMPI and aocc-compiler-*.tar should be installed and compiled with slurm on all cluster nodes or should be available on the NFS share.

Note:

- Omnia currently supports `pmix version2`, `pmix_v2`.
- While compiling OpenMPI, include `pmix`, `slurm`, `hwloc` and, `libevent` as shown in the below sample command:

```
./configure --prefix=/home/omnia-share/openmpi-4.1.5 --enable-mpi1-compatibility --
↪enable-orterun-prefix-by-default --with-slurm=/usr --with-pmix=/usr --with-
↪libevent=/usr --with-hwloc=/usr --with-ucx CC=clang CXX=clang++ FC=flang 2>&1 |
↪tee config.out
```

To execute multi-node jobs

1. Update the following parameters in `/etc/slurm/slurm.conf`:

```
SelectType=select/cons_tres
SelectTypeParameters=CR_Core
TaskPlugin=task/affinity,task/cgroup
```

2. Restart `slurmd.service` on all compute nodes.

```
systemctl stop slurmd
systemctl start slurmd
```

3. Once the service restarts on the compute nodes, restart `slurmctld.service` on the `kube_control_plane`.

```
systemctl stop slurmctld.service
systemctl start slurmctld.service
```

4. Job execution can now be initiated.

For a job to run on multiple nodes (10.5.0.4 and 10.5.0.5) where OpenMPI is compiled and installed on the NFS share (`/home/omnia-share/openmpi/bin/mpirun`), the job can be initiated as below: .. note:: Ensure `amd-zen-hpl-2023_07_18` is downloaded before running this command.

```
srun -N 2 --mpi=pmix_v2 -n 2 ./amd-zen-hpl-2023_07_18/xhpl
```

For a batch job using the same parameters, the script would be:

```
#!/bin/bash

#SBATCH --job-name=test

#SBATCH --output=test.log

#SBATCH --partition=normal
```

(continues on next page)

(continued from previous page)

```
#SBATCH -N 3

#SBATCH --time=10:00

#SBATCH --ntasks=2


source /home/omnia-share/setenv_AOCC.sh

export PATH=$PATH:/home/omnia-share/openmpi/bin

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/omnia-share/openmpi/lib

srun --mpi=pmix_v2 ./amd-zen-hpl-2023_07_18/xhpl
```

Alternatively, to use mpirun, the script would be:

```
#!/bin/bash

#SBATCH --job-name=test

#SBATCH --output=test.log

#SBATCH --partition=normal

#SBATCH -N 3

#SBATCH --time=10:00

#SBATCH --ntasks=2


source /home/omnia-share/setenv_AOCC.sh

export PATH=$PATH:/home/omnia-share/openmpi/bin

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/omnia-share/openmpi/lib

/home/omnia-share/openmpi/bin/mpirun --map-by ppr:1:node -np 2 --display-map --
↪oversubscribe --mca orte_keep_fqdn_hostnames 1 ./xhpl
```

Note: The above scripts are samples that can be modified as required. Ensure that `--mca orte_keep_fqdn_hostnames 1` is included in the mpirun command in sbatch scripts. Omnia maintains all hostnames in FQDN format. Failing to include `--mca orte_keep_fqdn_hostnames 1` may cause job initiation to fail.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.11.3 Installing pmix and updating slurm configuration for AMD processors

This topic explains how to automatically update AMD servers for MPI jobs. To manually install pmix and update the slurm configuration, [click here](#).

Pre-requisites

- discovery_provision.yml has been executed.
- An Omnia **slurm** cluster has been set up by omnia.yml running with at least 2 nodes: 1 slurm_control_node and 1 slurm_node.
- Verify that the target nodes are in the booted state. For more information, [click here](#).
- A local OpenMPI repository has been created. For more information, [click here](#). <../LocalRepo/openMPI.html>

To run the playbook:

```
cd benchmarks
ansible-playbook amd_benchmark.yml -i inventory
```

To execute multi-node jobs

- OpenMPI and aocc-compiler-*.tar should be installed and compiled with slurm on all cluster nodes or should be available on the NFS share.

Note:

- Omnia currently supports pmix version2, pmix_v2.
- While compiling OpenMPI, include pmix, slurm, hwloc and, libevent as shown in the below sample command:

```
./configure --prefix=/home/omnia-share/openmpi-4.1.5 --enable-mpi1-compatibility --
↪enable-orterun-prefix-by-default --with-slurm=/usr --with-pmix=/usr --with-
↪libevent=/usr --with-hwloc=/usr --with-ucx CC=clang CXX=clang++ FC=flang 2>&1 |
↪tee config.out
```

- For a job to run on multiple nodes (10.5.0.4 and 10.5.0.5) where OpenMPI is compiled and installed on the NFS share (/home/omnia-share/openmpi/bin/mpirun), the job can be initiated as below:

Note: Ensure amd-zen-hpl-2023_07_18 is downloaded before running this command.

```
srun -N 2 --mpi=pmix_v2 -n 2 ./amd-zen-hpl-2023_07_18/xhpl
```

For a batch job using the same parameters, the script would be:

```
#!/bin/bash

#SBATCH --job-name=test

#SBATCH --output=test.log

#SBATCH --partition=normal
```

(continues on next page)

(continued from previous page)

```
#SBATCH -N 3

#SBATCH --time=10:00

#SBATCH --ntasks=2


source /home/omnia-share/setenv_AOCC.sh

export PATH=$PATH:/home/omnia-share/openmpi/bin

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/omnia-share/openmpi/lib

srun --mpi=pmix_v2 ./amd-zen-hpl-2023_07_18/xhpl
```

Alternatively, to use mpirun, the script would be:

```
#!/bin/bash

#SBATCH --job-name=test

#SBATCH --output=test.log

#SBATCH --partition=normal

#SBATCH -N 3

#SBATCH --time=10:00

#SBATCH --ntasks=2


source /home/omnia-share/setenv_AOCC.sh

export PATH=$PATH:/home/omnia-share/openmpi/bin

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/omnia-share/openmpi/lib

/home/omnia-share/openmpi/bin/mpirun --map-by ppr:1:node -np 2 --display-map --
↪oversubscribe --mca orte_keep_fqdn_hostnames 1 ./xhpl
```

Note: The above scripts are samples that can be modified as required. Ensure that `--mca orte_keep_fqdn_hostnames 1` is included in the mpirun command in sbatch scripts. Omnia maintains all hostnames in FQDN format. Failing to include `--mca orte_keep_fqdn_hostnames 1` may cause job initiation to fail.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.11.4 Containerized HPC benchmark execution

Use this playbook to download docker images and pull images onto cluster nodes using `apptainer`.

1. Ensure that the cluster has been [provisioned by the provision tool](#), and the cluster has been set up using `omnia.yml`.
2. Enter the following variables in `utils/hpc_apptainer_job_execution/hpc_apptainer_job_execution_config.yml`:

Parameter	Details
hpc_apptainer_image JSON list Required	<ul style="list-style-type: none"> • Docker image details to be downloaded in to cluster nodes using apptainer to create a sif file. • Example (for single image): <pre>hpc_apptainer_image: - { image_url: "docker.io/intel/ ↪oneapi-hpckit:latest" }</pre> • Example (for multiple images): <pre>hpc_apptainer_image: - { image_url: "docker.io/intel/ ↪oneapi-hpckit:latest" } - { image_url: "docker.io/ ↪tensorflow/tensorflow:latest" }</pre> • If provided, docker credentials in <code>omnia_config.yml</code>, it will be used for downloading docker images.
hpc_apptainer_path string Required	<ul style="list-style-type: none"> • Directory to filepath for storing apptainer sif files on cluster nodes. • It is recommended to use a directory inside a shared path that is accessible to all cluster nodes. • Default value: <code>"/home/omnia-share/softwares/apptainer"</code>

To run the playbook:

```
cd utils/hpc_apptainer_job_execution
ansible-playbook hpc_apptainer_job_execution.yml -i inventory
```

Note: Use the inventory file format specified under [Sample Files](#).

HPC apptainer jobs can be initiated on a slurm cluster using the following sample command:

```
srun -N 3 --mpi=pmi2 --ntasks=4 apptainer run /home/omnia-share/softwares/apptainer/
↪oneapi-hpckit_latest.sif hostname
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.12 Remove Slurm/K8s configuration from a node

Use this playbook to remove slurm and kubernetes configuration from slurm or kubernetes worker nodes of the cluster and stop all clustering software on the worker nodes.

Note:

- All target nodes should be drained before executing the playbook. If a job is running on any target nodes, the playbook may timeout waiting for the node state to change.
 - When running `remove_node_configuration.yml`, ensure that the `input/storage_config.yml` and `input/omnia_config.yml` have not been edited since `omnia.yml` was run.
-

Configurations performed by the playbook

- Nodes specified in the `slurm_node` group or `kube_node` group in the inventory file will be removed from the slurm and kubernetes cluster respectively.
- Slurm and Kubernetes services are stopped and uninstalled. OS startup service list will be updated to disable Slurm and Kubernetes.

To run the playbook

Run the playbook using the following commands:

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

- To specify only Slurm or Kubernetes nodes while running the playbook, use the tags `slurm_node` or `kube_node`. That is:
- To remove only slurm nodes, use `ansible-playbook remove_node_configuration.yml -i inventory --tags slurm_node`.
- To remove only kubernetes nodes, use `ansible-playbook remove_node_configuration.yml -i inventory --tags kube_node`.
- Passed inventory files should exclusively contain either service tags or admin IPs. Do not provide a mix of both in a single inventory file.
- To skip confirmation while running the playbook, use `ansible-playbook remove_node_configuration.yml -i inventory --extra-vars skip_confirmation=yes` or `ansible-playbook remove_node_configuration.yml -i inventory -e skip_confirmation=yes`.

2.13 Soft reset the cluster

Use this playbook to stop all Slurm and Kubernetes services. This action will destroy the cluster.

Note:

- All target nodes should be drained before executing the playbook. If a job is running on any target nodes, the playbook may timeout waiting for the node state to change.
 - When running `reset_cluster_configuration.yml`, ensure that the `input/storage_config.yml` and `input/omnia_config.yml` have not been edited since `omnia.yml` was run.
-

Configurations performed by the playbook

- The configuration on the `kube_control_plane` or the `slurm_control_plane` will be reset.
- Slurm and Kubernetes services are stopped and removed.

To run the playbook

Run the playbook using the following commands:

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

To specify only Slurm or Kubernetes clusters while running the playbook, use the tags `slurm_cluster` or `k8s_cluster`. That is:

To reset a slurm cluster, use `ansible-playbook reset_cluster_configuration.yml -i inventory --tags slurm_cluster`. To reset a kubernetes cluster, use `ansible-playbook reset_cluster_configuration.yml -i inventory --tags k8s_cluster`.

To skip confirmation while running the playbook, use `ansible-playbook reset_cluster_configuration.yml -i inventory --extra-vars skip_confirmation=yes` or `ansible-playbook remove_node_configuration.yml -i inventory -e skip_confirmation=yes`.

The inventory file passed for `reset_cluster_configuration` should follow the below format. Passed inventory files should exclusively contain either service tags or admin IPs. Do not provide a mix of both in a single inventory file.:

For a slurm cluster

```
[slurm_control_node]
{ip or servicetag}

[slurm_node]
{ip or servicetag}
{ip or servicetag}
```

For a kubernetes cluster

```
[kube_control_plane]
{ip or servicetag}

[etcd]
{ip or servicetag}
```

(continues on next page)

(continued from previous page)

```
[kube_node]
{ip or servicetag}
{ip or servicetag}
```

2.14 Delete provisioned node

Use this playbook to remove discovered or provisioned nodes from all inventory files and Omnia database tables. No changes are made to the Slurm or Kubernetes cluster.

Note: To undo changes made by this playbook, re-run the provision tool on the target node.

Configurations performed by the playbook

- Nodes will be deleted from the Omnia DB and the xCAT node object will be deleted.
- Telemetry services will be stopped and removed.

To run the playbook

Run the playbook using the following commands:

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

To skip confirmation while running the playbook, use `ansible-playbook delete_node.yml -i inventory --extra-vars skip_confirmation=yes` or `ansible-playbook remove_node_configuration.yml -i inventory -e skip_confirmation=yes`.

The inventory file passed for `delete_node.yml` should follow one of the below formats. Passed inventory files should exclusively contain either service tags or admin IPs. Do not provide a mix of both in a single inventory file.:

```
[nodes]
{ip address}
{ip address}
```

```
[nodes]
{service tag}
{service tag}
```

Note:

- When the node is added or deleted, the autogenerated inventories: `amd_gpu`, `nvidia_gpu`, `amd_cpu`, and `intel_cpu` should be updated for the latest changes.
 - Nodes passed in the above inventory will be removed from the cluster. To reprovision the node, use the *add node script*. <addinganewnode.html>
-

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

2.15 Uninstalling the provision tool

Use this script to undo all the changes made by the provision tool. For a list of actions taken by the provision tool, [click here](#).

To run the script:

```
cd utils
ansible-playbook control_plane_cleanup.yml
```

To skip the deletion of the configured local repositories (stored in `repo_store_path` and xCAT repositories), run:

```
ansible-playbook control_plane_cleanup.yml -skip-tags downloads
```

To delete the changes made by `local_repo.yml` while retaining the `repo_store_path` folder, run:

```
ansible-playbook control_plane_cleanup.yml -tags local_repo --skip-tags downloads
```

To delete the changes made by `local_repo.yml` including the `repo_store_path` folder, run:

```
ansible-playbook control_plane_cleanup.yml -tags local_repo
```

Caution:

- When re-provisioning your cluster (that is, re-running the `discovery_provision.yml` playbook) after a clean-up, ensure to use a different `admin_nic_subnet` in `input/provision_config.yml` to avoid a conflict with newly assigned servers. Alternatively, disable any OS available in the **Boot Option Enable/Disable** section of your BIOS settings (**BIOS Settings > Boot Settings > UEFI Boot Settings**) on all target nodes.
- On subsequent runs of `discovery_provision.yml`, if users are unable to log into the server, refresh the ssh key manually and retry.

```
ssh-keygen -R <node IP>
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

FEATURES

From Omnia 1.4, all of Omnia's many features are available individually. Specific playbooks allow users to choose different features and customize their deployment journey specifically to their needs.

Below is a list of all Omnia's features:

3.1 Centralized authentication on the cluster

The security feature allows users to set up FreeIPA and LDAP to help authenticate into HPC clusters.

3.1.1 Configuring FreeIPA/LDAP security

Pre requisites

- Run `local_repo.yml` to create offline repositories of FreeIPA or OpenLDAP. If both were downloaded, ensure that the non-required system is removed from `input/software_config.json` before running `security.yml`. For more information, [click here](#).
- Enter the following parameters in `input/security_config.yml`.

Table 1: Parameters for Authentication

Parameter	Details
domain_name string Required	<ul style="list-style-type: none">• Sets the intended domain name.• If <code>dc=omnia,dc=test</code>, Provide <code>omnia.test</code>• If <code>dc=dell,dc=omnia,dc=com</code> Provide <code>dell.omnia.com</code> Default values: <code>omnia.test</code>

Table 2: Parameters for OpenLDAP configuration

Parameter	Details
ldap_connection_type string Required	For a TLS connection, provide a valid certification path. For an SSL connection, ensure port 636 is open. Choices: <ul style="list-style-type: none"> • TLS <- Default • SSL
tls_ca_certificate string Optional	File path pointing to the Certificate Authority (CA) issued certificate path. Certificate files should be saved with a .pem or .crt extension. If not provided, a self-signed certificate is generated by Omnia.
tls_certificate string Optional	File path pointing to the certificate used to authorize the LDAP server. Certificate files should be saved with a .pem or .crt extension.
tls_certificate_key string Optional	The private key that matches the LDAP certificate.
openldap_db_username string Required	The username used to manage the LDAP database. Default value: "admin"
openldap_db_password string Required	The password used to configure and manage the LDAP database. Ensure that this value is 8 characters long.
openldap_config_username string Required	The username used to configure the LDAP database. Default value: "admin"
openldap_config_password string Required	The password used to configure the LDAP database. Ensure that this value is 8 characters long.
openldap_monitor_password string Required	The password used to monitor the LDAP database. Ensure that this value is 8 characters long.
openldap_organization string Required	LDAP server is configured using organizations. They are necessary for user creation and group mapping. Default value: "omnia"
openldap_organizational_unit string Required	LDAP server is configured using organizations. They are necessary for user creation and group mapping. Default value: "People"

Table 3: Parameters for FreeIPA configuration

Parameter	Details
realm_name string Required	<ul style="list-style-type: none"> Sets the intended kerberos realm name. It is required for FreeIPA setups. A realm name is often, but not always the upper case version of the name of the DNS domain over which it presides. Default value: "OMNIA.TEST"
directory_manager_password string Required	<ul style="list-style-type: none"> The directory server operations require an administrative user. This user is referred to as the Directory Manager and has full access to the Directory for system management tasks and will be added to the instance of directory server created for IPA. The password must be at least 8 characters long. The password must not contain -, ', "
kerberos_admin_password string Required	<ul style="list-style-type: none"> kerberos_admin_password used by IPA admin user. The IPA server requires an administrative user, named 'admin'. The password must be at least 8 characters long. The password must not contain -, ', "

Create a new user on OpenLDAP

1. Create an LDIF file (eg: `create_user.ldif`) on the auth server containing the following information:

- DN: The distinguished name that indicates where the user will be created.
- objectClass: The object class specifies the mandatory and optional attributes that can be associated with an entry of that class. Here, the values are `inetOrgPerson`, `posixAccount`, and `shadowAccount`.
- UID: The username of the replication user.
- sn: The surname of the intended user.
- cn: The given name of the intended user.

Below is a sample file:

```
# User Creation
dn: uid=ldapuser,ou=People,dc=omnia,dc=test
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: ldapuser
sn: ldapuser
loginShell: /bin/bash
uidNumber: 2000
gidNumber: 2000
homeDirectory: /home/ldapuser
```

(continues on next page)

(continued from previous page)

```
shadowLastChange: 0
shadowMax: 0
shadowWarning: 0

# Group Creation
dn: cn=ldapuser,ou=Group,dc=omnia,dc=test
objectClass: posixGroup
cn: ldapuser
gidNumber: 2000
memberUid: ldapuser
```

Note: Avoid whitespaces when using an LDIF file for user creation. Extra spaces in the input data may be encrypted by OpenLDAP and cause access failures.

2. Run the command `ldapadd -D <admin database username> -w <admin database password> -f create_user.ldif` to execute the LDIF file and create the account.
3. To set up a password for this account, use the command `ldappasswd -D <admin database username> -w <admin database password> -S <user_dn>`. The value of `user_dn` is the distinguished name that indicates where the user was created. (In this example, `ldapuser,ou=People,dc=omnia,dc=test`)

3.1.2 Configuring login node security

Prerequisites

- Run `local_repo.yml` to create an offline repository of all utilities used to secure the login node. For more information, [click here](#).

Enter the following parameters in `input/login_node_security_config.yml`.

Variable	Details
max_failures integer Optional	The number of login failures that can take place before the account is locked out. Default values: 3
failure_reset_interval integer Optional	Period (in seconds) after which the number of failed login attempts is reset. Min value: 30; Max value: 60. Default values: 60
lockout_duration integer Optional	Period (in seconds) for which users are locked out. Min value: 5; Max value: 10. Default values: 10
session_timeout integer Optional	User sessions that have been idle for a specific period can be ended automatically. Min value: 90; Max value: 180. Default values: 180
alert_email_address string Optional	Email address used for sending alerts in case of authentication failure. When blank, authentication failure alerts are disabled. Currently, only one email ID is accepted.
user string Optional	Access control list of users. Accepted formats are <code>username@ip</code> (<code>root@1.2.3.4</code>) or username (<code>root</code>). Multiple users can be separated using whitespaces.
allow_deny string Optional	This variable decides whether users are to be allowed or denied access. Ensure that AllowUsers or DenyUsers entries on sshd configuration file are not commented. Choices: <ul style="list-style-type: none"> • <code>allow</code> <- Default • <code>deny</code>
restrict_program_support boolean Optional	This variable is used to disable services. Root access is mandatory. Choices: <ul style="list-style-type: none"> • <code>false</code> <- Default • <code>true</code>
restrict_softwares string Optional	List of services to be disabled (Comma-separated). Example: <code>'telnet,lpd,bluetooth'</code> Choices: <ul style="list-style-type: none"> • <code>telnet</code> • <code>lpd</code> • <code>bluetooth</code> • <code>rlogin</code> • <code>rexec</code>

3.1.3 Installing LDAP Client

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

FreeIPA installation on the NFS node

IPA services are used to provide account management and centralized authentication.

To customize your installation of FreeIPA, enter the following parameters in `input/security_config.yml`.

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

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_ipaddress=""
```

Hostname requirements

- 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: `hostname000000x.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 cluster node to ‘node000001.omnia.test’. Omnia appends 6 digits to the hostname to individually name each target node.

Note: Use the format specified under [NFS inventory in the Sample Files](#) for inventory.

Running the security role

Run:

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

The inventory should contain `auth_server` as per the inventory file in [samplefiles](#). The inventory file is case-sensitive. Follow the format provided in the sample file link.

- Do not include the IP of the control plane or local host in the `auth_server` group in the passed inventory.
- To customize the security features on the login node, fill out the parameters in `input/login_node_security_config.yml`.
- If a subsequent run of `security.yml` fails, the `security_config.yml` file will be unencrypted.

Caution: No users will be created by Omnia.

How to replicate LDAP

Once Omnia has set up an LDAP server for the cluster, external LDAP servers can be replicated onto the cluster LDAP server using the following steps.

[Optional]Create a replication user

1. Create an LDIF file (eg: `replication_user.ldif`) on the external LDAP server (source) containing the following information:
 - DN: The distinguished name that indicates where the user will be created.
 - objectClass: The object class specifies the mandatory and optional attributes that can be associated with an entry of that class. Here, the values are `simpleSecurityObject`, `account`, and `shadowAccount`.
 - UID: The username of the replication user.
 - Description: A user-defined string describing the account.
 - UserPassword: The SHA encrypted value of the intended user password. This can be obtained using `slappasswd`

Below is a sample file:

```
dn: uid=replicausser,dc=orchid,dc=cluster
objectClass: simpleSecurityObject
objectclass: account
objectClass: shadowAccount
uid: replicausser
description: Replication User
userPassword: {SSHA}BL5xdrUvHQ8GPvdvHh0/4OmKHYoXQ1IK
```

2. Run the command `ldapadd -D <enter admin binddn> -w < bind_password> -f replication_user.ldif` to execute the LDIF file and create the account.

Initiate the replication

1. Create an LDIF file (eg: `Replication.ldif`) on the auth server on the cluster (destination) containing the following information:

- Provider: The IP address of the source LDAP server. It is routed over the LDAP protocol and via port 389.
- binddn: The distinguished name of the dedicated replication user or admin user being used to authenticate the replication.
- credentials: The corresponding password of the user indicated in binddn.
- searchbase: The groups of users to be replicated.

Below is a sample file:

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcSyncRepl
olcSyncRepl: rid=001
  provider=ldap://xx.xx.xx.xx:389/
  bindmethod=simple
  binddn="uid=replicausers,dc=orchid,dc=cluster"
  credentials=sync1234
  searchbase="dc=orchid,dc=cluster"
  scope=sub
  schemachecking=on
  type=refreshAndPersist
  retry="30 5 300 3"
  interval=00:00:05:00
```

2. Run the command `ldapadd -D cn=<config_username>,cn=config -w < config_password > -f Replication.ldif` to execute the LDIF file and initiate the replication.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.2 Shared and distributed storage deployment

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

1. Enter all required parameters in `input/storage_config.yml`

Table 4: Parameters for Storage

Variables	Details
nfs_client_params JSON List Required	<ul style="list-style-type: none"> This JSON list contains all parameters required to set up NFS. For a bolt-on set up where there is a pre-existing NFS export, set <code>nfs_server</code> to <code>false</code>. When <code>nfs_server</code> is set to <code>true</code>, an NFS share is created on the control plane for access by all cluster nodes. For more information on the different kinds of configuration available, click here.
beegfs_rdma_support boolean Optional	This variable is used if user has RDMA-capable network hardware (e.g., InfiniBand) Choices: <ul style="list-style-type: none"> <code>false</code> <- Default <code>true</code>
beegfs_ofed_kernel_modules_path string Optional	<ul style="list-style-type: none"> The path where separate OFED kernel modules are installed. Ensure that the path provided here exists on all target nodes. Default value: <code>"/usr/src/ofa_kernel/default/include"</code>
beegfs_mgmt_server string Required	BeeGFS management server IP. <hr/> Note: The provided IP should have an explicit BeeGFS management server running . <hr/>
beegfs_mounts string Optional	BeeGfs-client file system mount location. If <code>storage.yml</code> is being used to change the BeeGFS mounts location, set <code>beegfs_unmount_client</code> to <code>true</code>. Default value: <code>"/mnt/beegfs"</code>
beegfs_unmount_client boolean Optional	Changing this value to <code>true</code> will unmount running instance of BeeGFS client and should only be used when decommissioning BeeGFS, changing the mount location or changing the BeeGFS version. Choices: <ul style="list-style-type: none"> <code>false</code> <- Default <code>true</code>
beegfs_version_change boolean Optional	Use this variable to change the BeeGFS version on the target nodes. Choices: <ul style="list-style-type: none"> <code>false</code> <- Default <code>true</code>
ansible_config_file_path string	<ul style="list-style-type: none"> Path to directory hosting ansible config file (<code>ansible.cfg</code> file)

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

2. Ensure that the entry `{"name": "beegfs", "version": "7.2.6"}`, is included in `input/software_config.json` and a local repository is created. For more information, [click here](#).

Installing BeeGFS Client

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

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

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

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-mgmt.conf`, `/etc/beegfs/beegfs-meta.conf` and `/etc/beegfs/beegfs-storage.conf`. Restart the services to complete the task:

```
systemctl restart beegfs-mgmt
systemctl restart beegfs-meta
systemctl restart beegfs-storage
systemctl status beegfs-mgmt
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 `no-suid,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 `kube_control_plane`, `login_node` and `compute nodes`.)

Note: If a subsequent run of `storage.yml` fails, the `storage_config.yml` file will be unencrypted.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.3 GPU accelerator configuration

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.

Ensure that CUDA and ROCm local repositories are configured using the `local_repo.yml` script.

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

Parameters	Details
amd_gpu_version string 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 cluster os version is rhel 8.5. Then the url transforms to https://repo.radeon.com/amdgpu/latest/rhel/8.5/main/x86_64/ Default values: 22.20.3
amd_rocm_version string 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. Default values: latest/main
cuda_toolkit_version string Optional	Required CUDA toolkit version. By default latest cuda is installed unless <code>cuda_toolkit_path</code> is specified. Default: latest (11.8.0). Default values: latest
cuda_toolkit_path string 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 <code>cuda_toolkit_version</code> is not latest, giving <code>cuda_toolkit_path</code> is mandatory.
cuda_stream string 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. Default values: latest-dkms

Note:

- Nodes provisioned using the Omnia provision tool do not require a RedHat subscription to run `accelerator.yml` on RHEL target nodes.
- For RHEL target nodes not provisioned by Omnia, ensure that RedHat subscription is enabled on all target nodes. Every target node will require a RedHat subscription.
- AMD ROCm driver installation is not supported by Omnia on Rocky cluster nodes.

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

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

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.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.4 Additional utilities

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

3.4.1 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/utlis
ansible-playbook install_hpc_thirdparty_packages.yml -i inventory
```

Where the inventory refers to a file listing all nodes per the format provided in [inventory file](#).

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.4.2 Updating kernels on RHEL (with subscription)

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 nodes per the format provided in [inventory file](#). The inventory file is case-sensitive. Follow the format provided in the sample file link.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.4.3 Red Hat Subscription

Required Parameters

Name	Description
redhat_subscription_method string Optional	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. Choices: <ul style="list-style-type: none"> • portal <- Default • satellite
redhat_subscription_release string Optional	RHEL release version (e.g. 8.1)
redhat_subscription_force_register boolean Optional	Register the system even if it is already registered. Choices: <ul style="list-style-type: none"> • false <- Default • true
redhat_subscription_pool_ids string Optional	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 string Optional	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 string Optional	The state of all repos in redhat_subscription_repos. Choices: <ul style="list-style-type: none"> • enabled <- Default • disabled
redhat_subscription_repos_purge boolean Optional	This parameter disables all currently enabled repositories that are not specified in redhat_subscription_repos. Only set this to true if the redhat_subscription_repos field has multiple repos. Choices: <ul style="list-style-type: none"> • false <- Default • true
redhat_subscription_server_hostname string Optional	FQDN of subscription server. Mandatory field if redhat_subscription_method is set to satellite. Default values: subscription.rhn.redhat.com
redhat_subscription_port integer Optional	Port to use when connecting to subscription server. Set 443 for Satellite or RHN. If capsule is used, set 8443. Choices: <ul style="list-style-type: none"> • 443 <- Default • 8443
3.4. Additional utilities redhat_subscription_insecure boolean Optional	Disable certificate validation. Choices: <ul style="list-style-type: none"> • false <- Default • true

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 nodes per the format provided in [inventory file](#). The inventory file is case-sensitive. Follow the format provided in the sample file link.

3.4.4 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
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.4.5 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:

```
10.5.0.102
10.5.0.103
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.4.6 FreeIPA installation on the NFS node

IPA services are used to provide account management and centralized authentication.

To customize your installation of FreeIPA, enter the following parameters in `input/security_config.yml`.

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_ipaddress=""
```

Hostname requirements

- 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: `hostname000000x.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 cluster node to 'node000001.omnia.test'. Omnia appends 6 digits to the hostname to individually name each target node.

Note: Use the format specified under [NFS inventory in the Sample Files](#) for inventory.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.4.7 Uninstalling the provision tool

Use this script to undo all the changes made by the provision tool. For a list of actions taken by the provision tool, [click here](#).

To run the script:

```
cd utils
ansible-playbook control_plane_cleanup.yml
```

To skip the deletion of the configured local repositories (stored in `repo_store_path` and xCAT repositories), run:

```
ansible-playbook control_plane_cleanup.yml -skip-tags downloads
```

To delete the changes made by `local_repo.yml` while retaining the `repo_store_path` folder, run:

```
ansible-playbook control_plane_cleanup.yml -tags local_repo --skip-tags downloads
```

To delete the changes made by `local_repo.yml` including the `repo_store_path` folder, run:

```
ansible-playbook control_plane_cleanup.yml -tags local_repo
```

Caution:

- When re-provisioning your cluster (that is, re-running the `discovery_provision.yml` playbook) after a clean-up, ensure to use a different `admin_nic_subnet` in `input/provision_config.yml` to avoid a conflict with newly assigned servers. Alternatively, disable any OS available in the Boot Option Enable/Disable section of your BIOS settings (BIOS Settings > Boot Settings > UEFI Boot Settings) on all target nodes.
- On subsequent runs of `discovery_provision.yml`, if users are unable to log into the server, refresh the ssh key manually and retry.

```
ssh-keygen -R <node IP>
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.4.8 Remove node from the cluster

Use this playbook to remove nodes from the cluster and stop all clustering software on the target nodes.

Note: All target nodes should be drained before executing the playbook. If a job is running on any target nodes, the playbook will exit.

Configurations performed by the playbook

- Remove node from Slurm and Kubernetes cluster.
- Update Slurm and Kubernetes config.
- Slurm and Kubernetes services are stopped (not uninstalled). OS startup service list will be updated to disable Slurm and Kubernetes.

To run the playbook

Run the playbook using the following commands:

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

3.4.9 Soft reset the cluster

Use this playbook to stop all Slurm and Kubernetes services. This action will destroy the cluster.

Note: All target nodes should be drained before executing the playbook. If a job is running on any target nodes, the playbook will exit.

Configurations performed by the playbook

- The Slurm or Kubernetes cluster will be reset.
- The configuration on the kube_control_plane or the slurm_control_plane will be reset.
- Slurm and Kubernetes services are stopped (not uninstalled).

To run the playbook

Run the playbook using the following commands:

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

3.4.10 Delete node from the cluster

Use this playbook to remove nodes from all inventory files and tables. No changes are made to the Slurm or Kubernetes cluster.

Note: All target nodes should be drained before executing the playbook. If a job is running on any target nodes, the playbook will exit.

Configurations performed by the playbook

- Nodes will be deleted from the Omnia DB and xCAT node object will be deleted.
- Telemetry services will be stopped.

To run the playbook

Run the playbook using the following commands:

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

Note: When the node is added or deleted, the autogenerated inventories: `amd_gpu`, `nvidia_gpu`, `amd_cpu`, and `intel_cpu` should be updated for the latest changes. Slurm partition is also needs to be updated with these changes.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.4.11 OS Package Update

To install multiple packages on target nodes in a bulk operation, the `package_update.yml` playbook can be leveraged.

Prerequisites

- All target nodes should be running RHEL or Rocky.
- Download the packages (RPMs) for the target nodes and place them in this folder: `/install/post/otherpkgs/<Provision OS.Version>/x86_64/custom_software/Packages`.

Note: Do not use ISO files for updates or package installations.

- Create a package list by creating the following text file (For packages that are not in RHEL repos, name the file `update.otherpkgs.pkglist`. For OS packages, `xxxx.pkglist`) and place in the default path. For example: `/install/post/otherpkgs/<Provision OS.Version>/x86_64/custom_software/update.otherpkgs.pkglist`:

```

custom_software/<package1>-<version1>
custom_software/<package2>-<version2>
custom_software/<package3>-<version3>

```

To customize the package update, enter the following parameters in `utils/package_update_config.yml`:

Parameter	Details
os_type string Required	The operating system in use on the target cluster nodes. Choices: <ul style="list-style-type: none"> • rhel <- Default • rocky
os_version string Required	OS version of target nodes in the cluster. Default value: 8.6
package_list string Required	<ul style="list-style-type: none"> • Location path for the package list file • For other packagelist, file name should be - (xxx.otherpkgs.pkglist) • For os packagelist, file name should be - (xxx.pkglist) • All packages in this list will be installed/updated on remote nodes Default value: <code>"/install/post/otherpkgs/rhels8.6.0/x86_64/custom_software/update.otherpkgs.pkglist"</code>
package_type string Required	<ul style="list-style-type: none"> • Indicates whether the packages to be installed are os packages (they are available in baseos or appstream) or other (they're not part of os repos, appstream or baseos). • If the package is being downloaded to /install/post/otherpkgs/<Provision OS.Version>/x86_64/custom_software/Packages/, use the value other. Choices: <ul style="list-style-type: none"> • os • other <- Default
odelist string Required	comma-separated list of all target nodes in the cluster. Default value: all
reboot_required boolean Required	Indicates whether the remote nodes listed will be re-booted. Choices: <ul style="list-style-type: none"> • true • false <- Default

To run the playbook, run the following commands:

```
cd utils
ansible-playbook package_update.yml
```

Note: At the end of the playbook, the package update status is displayed by target node. If the update status of any node is failed, use the command log (/var/log/xcat/commands.log) to debug the issue. Alternatively, verify that the node is reachable post provisioning.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.4.12 Clearing ports from Omnia

To undo the configurations made by Omnia to switch ports in the event of a reconfiguration or a clean-up, the `delete_switch_ports.yml` playbook can be utilized.

Enter the required details in `utils/provision/switch_based_deletion_config.yml`:

Parameter	Details
switch_based_details JSON List Required	<ul style="list-style-type: none"> JSON list of ports to be cleared from the Omnia DBs. Example: <pre>- { ip: 172.96.28.12, ports: '1-48, ↪49:3,50' }</pre> Example with 2 switches: <pre>- { ip: 172.96.28.12, ports: '1-48, ↪49:3,50' } - { ip: 172.96.28.14, ports: '1,2,3,5 ↪' }</pre>

To run the playbook, use the below commands:

```
cd utils/provision
ansible-playbook delete_switch_ports.yml
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.4.13 TimescaleDB utility

Telemetry metrics stored in a timescaleDB can be copied locally in a csv format. This file can be used to generate insights into key statistics in your cluster.

To customize the local copy of the timescale DB, fill out the below parameters in `utils/timescaledb_utility/timescaledb_utility_config.yml`.

Parameter	Details
column_name string Optional	<ul style="list-style-type: none"> Filters timescaleDB data by metric name and value. If this value is not provided, all metrics and their corresponding values will be stored in the file.
column_value string Optional	<ul style="list-style-type: none"> Filters timescaleDB data by metric name and value. If this value is not provided, all metrics and their corresponding values will be stored in the file.
start_time string Optional	<ul style="list-style-type: none"> Filters timescaleDB data by time of polling. If this value is not provided, all metric values collected will be stored.
stop_time string Optional	<ul style="list-style-type: none"> Filters timescaleDB data by time of polling. If this value is not provided, all metric values collected will be stored.
filename string Required	<ul style="list-style-type: none"> Target filepath where all timescaleDB will be backed up.

To initiate the backup to local file, run the following ansible playbook:

```
cd utils/timescaledb_utility
ansible-playbook timescaledb_utility.yml
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.5 Telemetry and visualizations

The telemetry feature allows the set up of Omnia telemetry (to poll values from all Omnia provisioned nodes in the cluster) and/or iDRAC telemetry (To poll values from all eligible iDRACs in the cluster). It also installs [Grafana](#) and [Loki](#) as Kubernetes pods.

To initiate telemetry support, fill out the following parameters in `input/telemetry_config.yml`:

Table 5: Parameters

Parameter	Details
idrac_telemetry_support boolean ¹ Required	<ul style="list-style-type: none"> Enables iDRAC telemetry support and visualizations. Values: <pre>* false <- Default</pre> <pre>* true</pre> <hr/> <p>Note: When <code>idrac_telemetry_support</code> is true, <code>mysqldb_user</code>, <code>mysqldb_password</code> and <code>mysqldb_root_password</code> become mandatory.</p> <hr/>
omnia_telemetry_support boolean ^{Page 146, 1} Required	<ul style="list-style-type: none"> Starts or stops Omnia telemetry If <code>omnia_telemetry_support</code> is true, then at least one of <code>collect_regular_metrics</code> or <code>collect_health_check_metrics</code> or <code>collect_gpu_metrics</code> should be true, to collect metrics. If <code>omnia_telemetry_support</code> is false, telemetry acquisition will be stopped. Values: <pre>* false <- Default</pre> <pre>* true</pre>
visualization_support boolean ^{Page 146, 1} Required	<ul style="list-style-type: none"> Enables visualizations. Values: <pre>* false <- Default</pre> <pre>* true</pre> <hr/> <p>Note: When <code>visualization_support</code> is true, <code>grafana_username</code> and <code>grafana_password</code> become mandatory.</p> <hr/>
appliance_k8s_pod_net_cidr string Required	<ul style="list-style-type: none"> Kubernetes pod network CIDR for appliance k8s network. Make sure this value does not overlap with any of the host networks. Default value: "192.168.0.0/16"
pod_external_ip_start_range string Required	<ul style="list-style-type: none"> The start of the range that will be used by Load-balancer for assigning IPs to K8s services in admin NIC subnet configured on the control plane. The first and second octets (x,y) are not used/validated by Omnia. These values are internally calculated based on the value of <code>admin_nic_subnet</code> in <code>input/provision_config.yml</code>. If <code>pod_external_ip_start_range:</code> "x.y.240.100" and <code>pod_external_ip_end_range:</code> "x.y.240.105" and
3.5. Telemetry and visualizations	<ul style="list-style-type: none"> If <code>admin_nic_subnet</code> provided in <code>provision_config.yml</code> is 10.5.0.0, <code>pod_external_ip_start_range</code> will be 10.5.240.100 and

Once you have executed `discovery_provision.yml` and has also provisioned the cluster, initiate telemetry on the cluster as part of `omnia.yml`, which configures the cluster with scheduler, storage and authentication using the below command.

```
ansible-playbook omnia.yml -i inventory
```

Optionally, you can initiate only telemetry using the below command:

```
ansible-playbook telemetry.yml -i inventory
```

Note:

- Depending on the type of telemetry initiated, include the following groups in the inventory:
 - `omnia_telemetry`: manager, compute, [optional] login
 - `idrac_telemetry`: idrac
- If you would like a local backup of the timescaleDB used to store telemetry data, [click here](#).

After initiation, new iDRACs can be added for `idrac_telemetry` acquisition by running the following commands:

```
ansible-playbook add_idrac_node.yml -i inventory
```

Modifying telemetry information

To modify how data is collected from the cluster, modify the variables in `omnia/input/telemetry_config.yml` and re-run the `telemetry.yml` playbook.

- When `omnia_telemetry_support` is set to false, Omnia Telemetry Acquisition service will be stopped on all cluster nodes provided in the passed inventory.
- When `omnia_telemetry_support` is set to true, Omnia Telemetry Acquisition service will be restarted on all cluster nodes provided in the passed inventory.
- To start or stop the collection of regular metrics, health check metrics, or GPU metrics, update the values of `collect_regular_metrics`, `collect_health_check_metrics`, or `collect_gpu_metrics`. For a list of all metrics collected, [click here](#).

Note:

- Currently, changing the `grafana_username` and `grafana_password` values is not supported via `telemetry.yml`.
- The passed inventory should have an `idrac` group, if `idrac_telemetry_support` is true.
- If `omnia_telemetry_support` is true, then the inventory should have control plane and cluster node groups (as specified in the sample files) along with optional login group.
- Rocky 8.7 is not compatible with the Kubernetes installed by `telemetry.yml` due to known issues with cri-o. For more information, [click here](#).
- If a subsequent run of `telemetry.yml` fails, the `telemetry_config.yml` file will be unencrypted.

To access the Grafana UI*Pre requisites*

¹ Boolean parameters do not need to be passed with double or single quotes.

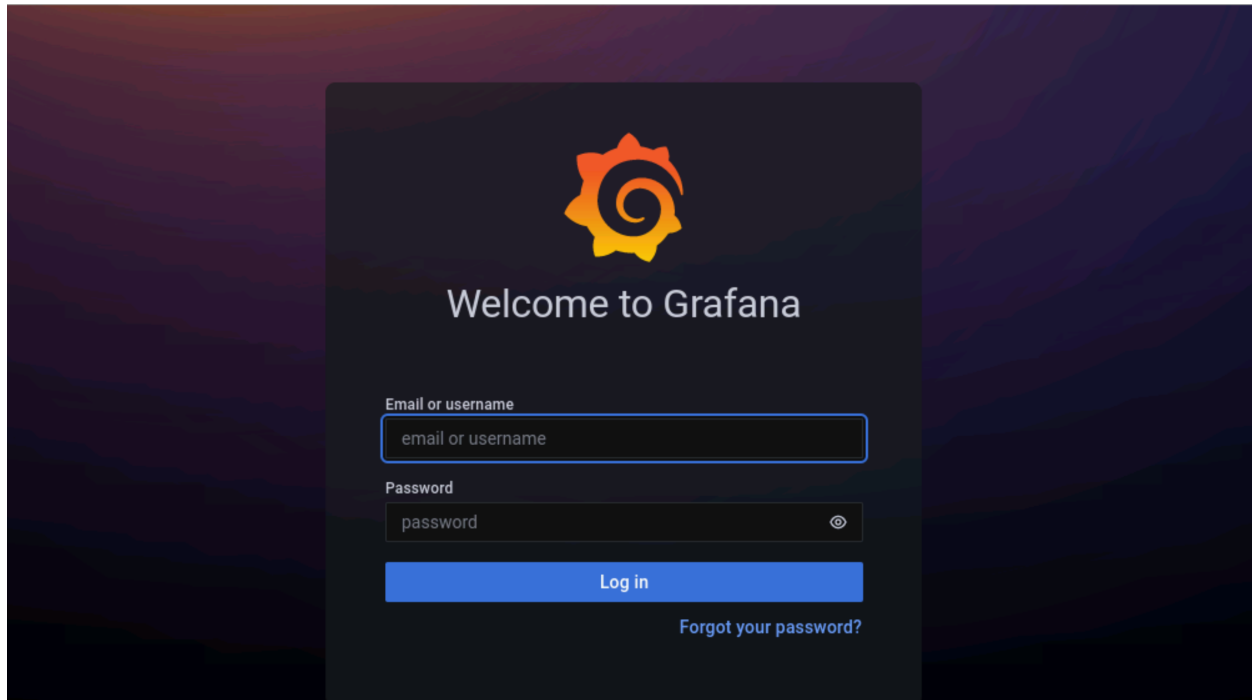
- `visualisation_support` should be set to `true` when running `telemetry.yml` or `omnia.yml`.

i. Find the IP address of the Grafana service using `kubectl get svc -n grafana`

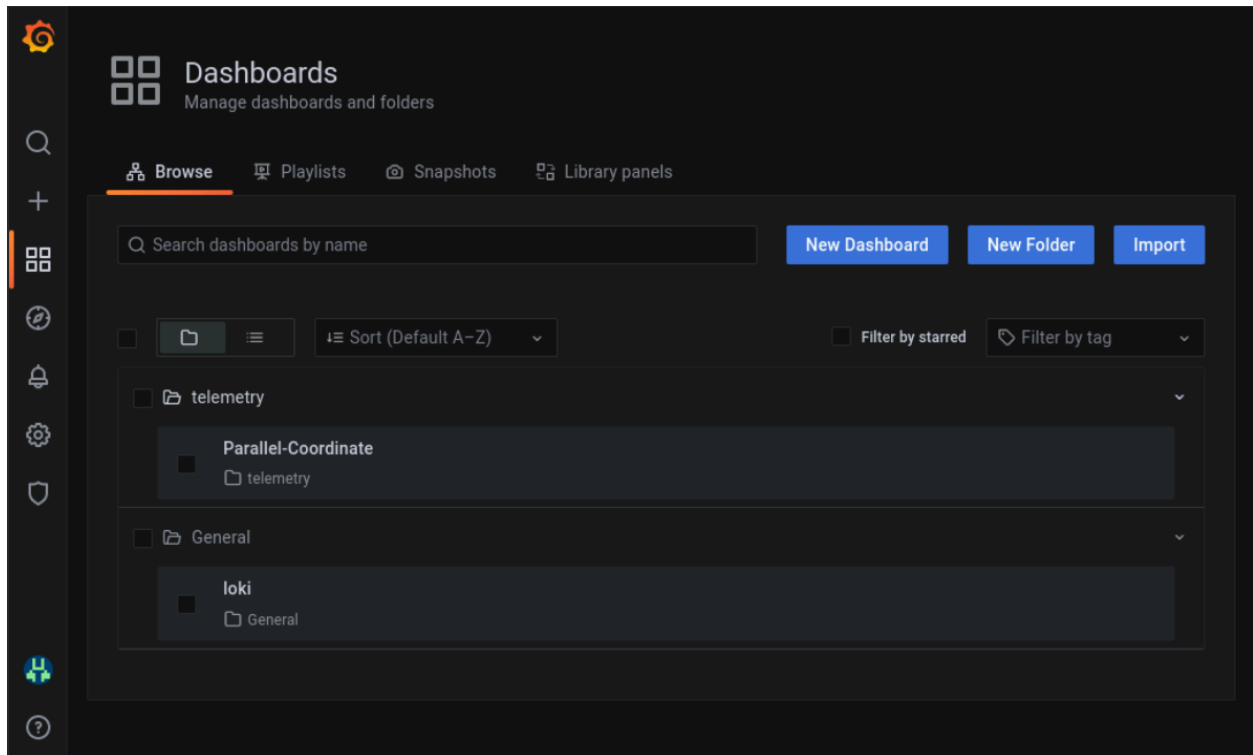
```
[root@localcp ~]# kubectl get svc -n grafana
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
grafana	LoadBalancer	10.102.225.70	10.2.240.100	5000:30590/TCP	30h
loki	ClusterIP	10.100.66.207	<none>	3100/TCP	30h

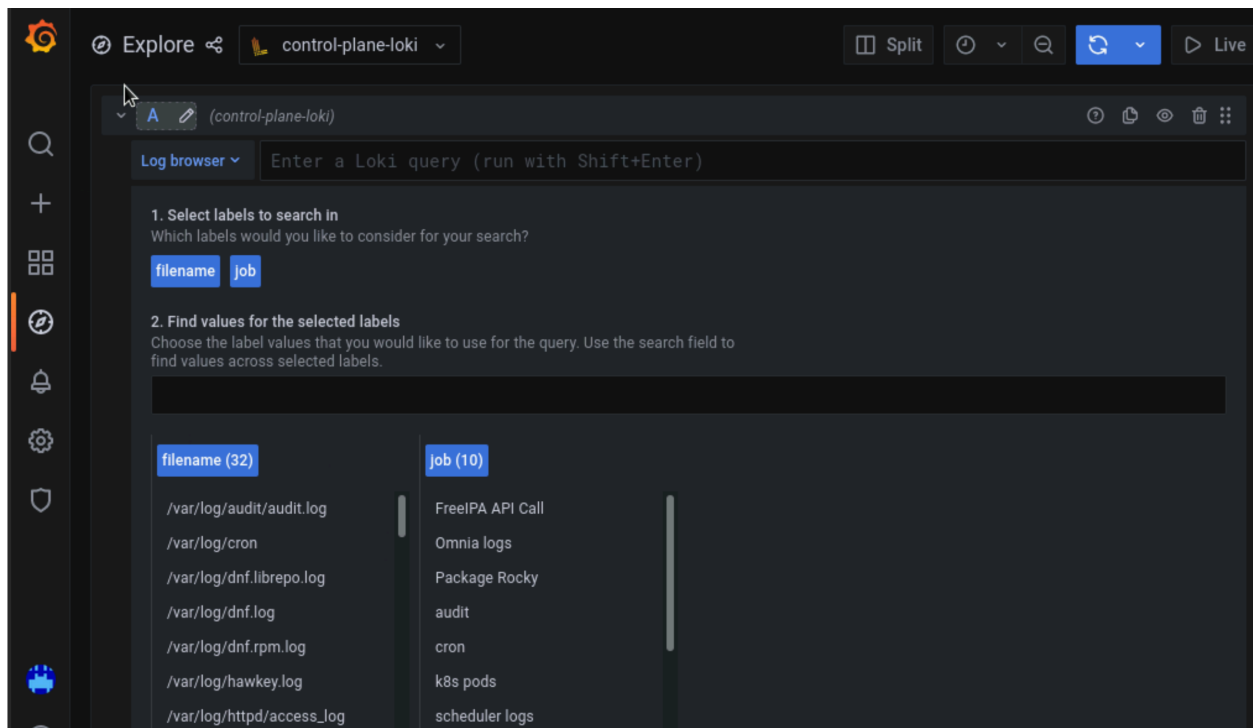
ii. Login to the Grafana UI by connecting to the cluster IP of grafana service obtained above via port 5000. That is `http://xx.xx.xx.xx:5000/login`



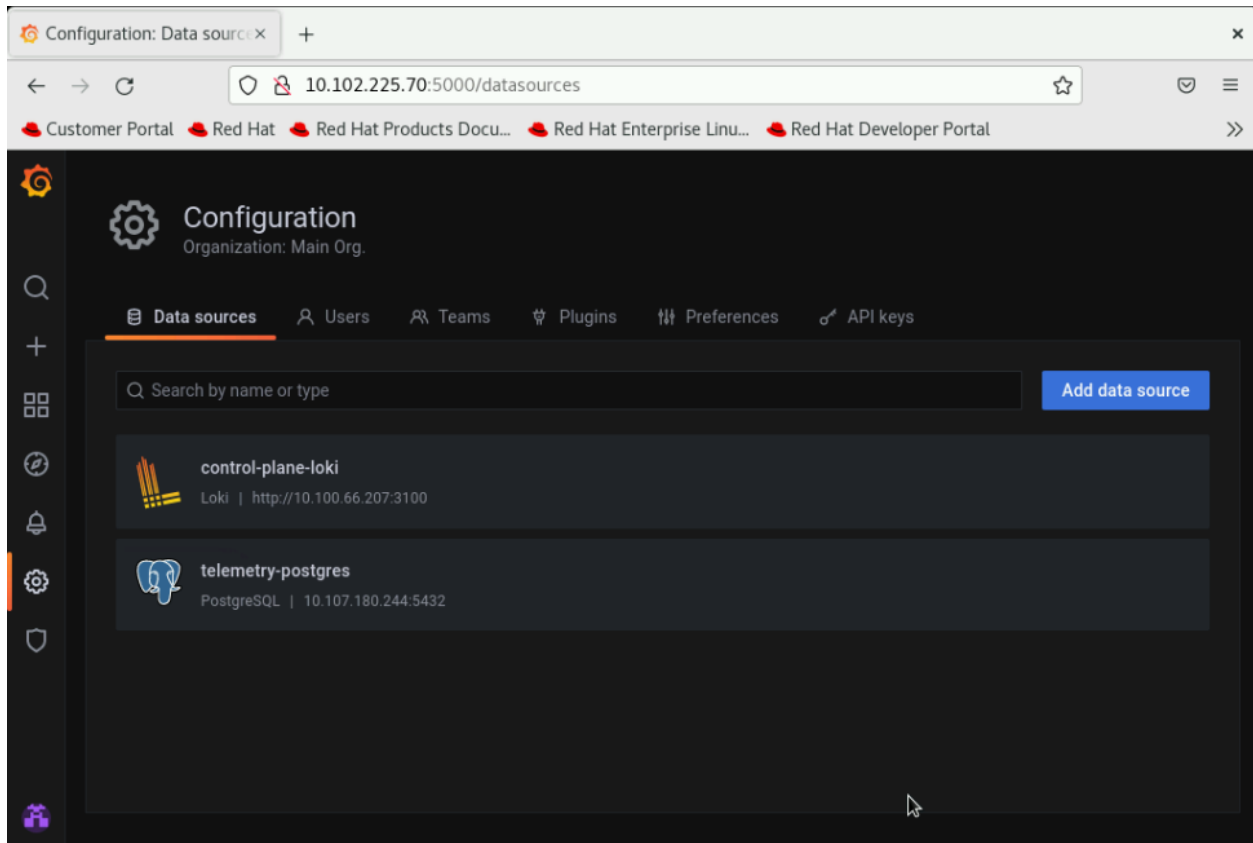
iii. Enter the `grafana_username` and `grafana_password` as mentioned in `input/telemetry_config.yml`.



Loki log collections can viewed on the explore section of the grafana UI.

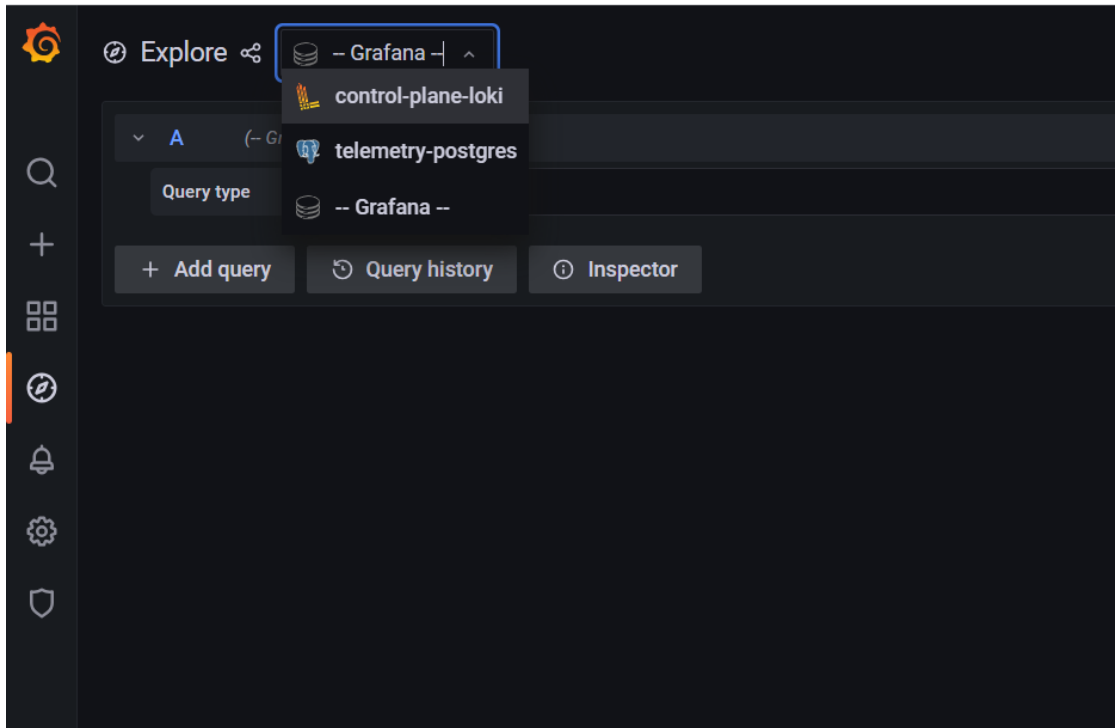


Datasources configured by Omnia can be viewed as seen below.



To use Loki for log filtering

- i. Login to the Grafana UI by connecting to the cluster IP of grafana service obtained above via port 5000. That is `http://xx.xx.xx.xx:5000/login`
- ii. In the Explore page, select **control-plane-loki**.



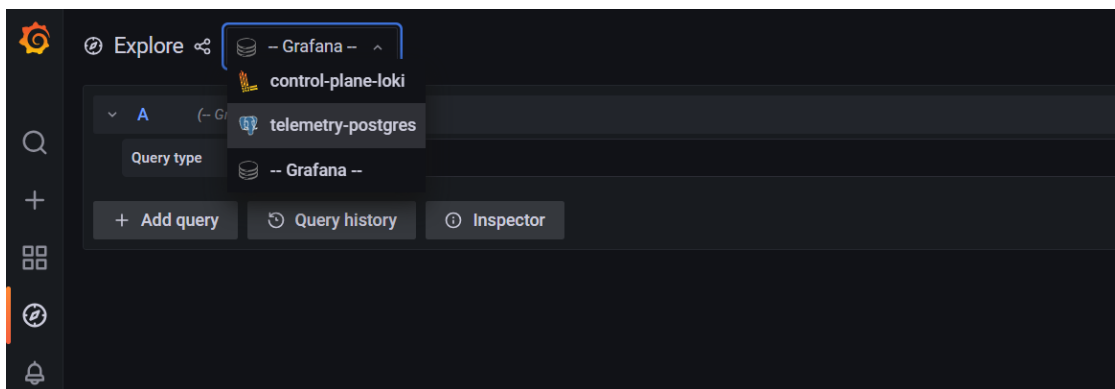
- iii. The log browser allows you to filter logs by job, node and/or user.

Example

```
(job)= "cluster deployment logs") |= "nodename"
(job="compute log messages") |= "nodename" |= "node_username"
```

To use Grafana to view telemetry data

- Login to the Grafana UI by connecting to the cluster IP of grafana service obtained above via port 5000. That is `http://xx.xx.xx.xx:5000/login`
- In the Explore page, select **telemetry-postgres**.



- iii. The query builder allows you to create SQL commands that can be used to query the `omnia_telemetry.metrics` table. Filter the data required using the following fields:

- **id**: The name of the metric.
- **context**: The type of metric being collected (Regular Metric, Health Check Metric and GPU metric).

- **label:** A combined field listing the **id** and **context** row values.
- **value:** The value of the metric at the given timestamp.
- **unit:** The unit measure of the metric (eg: Seconds, kb, percent, etc.)
- **system:** The service tag of the cluster node.
- **hostname:** The hostname of the cluster node.
- **time:** The timestamp at which the metric was polled from the cluster node.

If you are more comfortable using SQL queries over the query builder, click on **Edit SQL** to directly provide your query. Optionally, the data returned from a query can be viewed as a graph.

Visualizations

If `idrac_telemetry_support` and `visualisation_support` is set to true, Parallel Coordinate graphs can be used to view system statistics.

3.5.1 Acquiring telemetry data for iDRAC and Omnia

Using [Texas Technical University data visualization lab](#), data polled from iDRAC and Slurm can be processed to generate live graphs. These Graphs can be accessed on the Grafana UI.

Once `discovery_provision.yml` is executed and Grafana is set up, use `telemetry.yml` to initiate the Graphs. Data polled via Slurm and iDRAC is streamed into internal databases. This data is processed to create parallel coordinate graphs.

Note: This feature only works on nodes using iDRACs with a datacenter license running a minimum firmware version of 4.0.

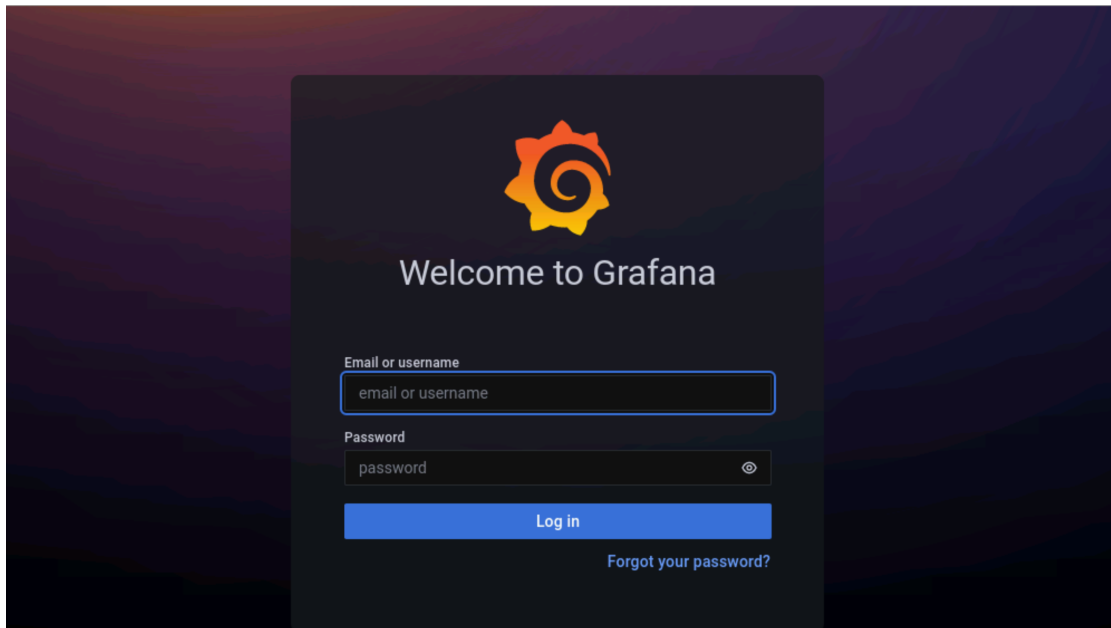
To access the grafana UI:

- i. Find the IP address of the Grafana service using `kubectl get svc -n grafana`

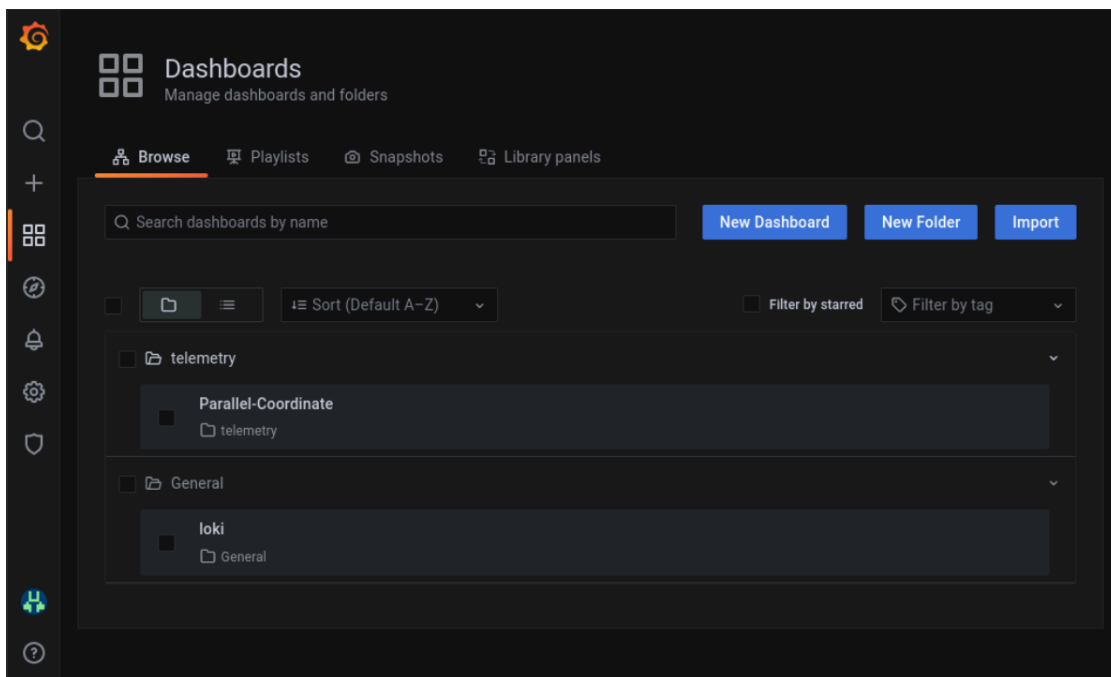
```
[root@localcp ~]# kubectl get svc -n grafana
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
grafana	LoadBalancer	10.102.225.70	10.2.240.100	5000:30590/TCP	30h
loki	ClusterIP	10.100.66.207	<none>	3100/TCP	30h

- ii. Login to the Grafana UI by connecting to the cluster IP of grafana service obtained above via port 5000. That is `http://xx.xx.xx.xx:5000/login`



- iii. Enter the `grafana_username` and `grafana_password` as mentioned in `input/telemetry_config.yml`.

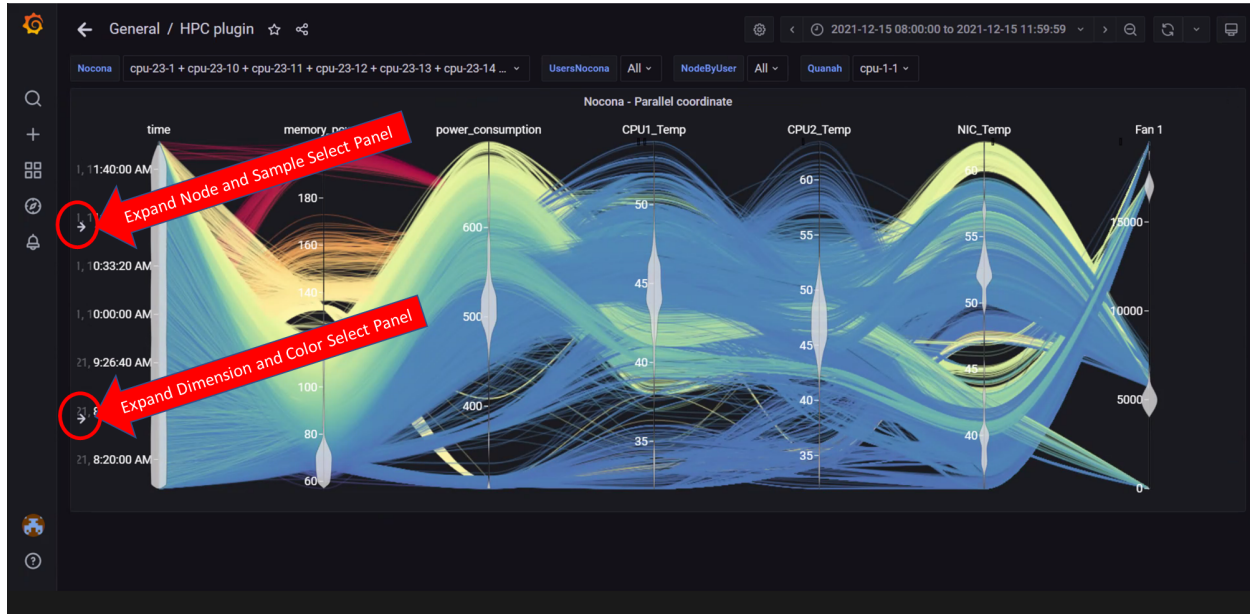


All your data in a glance:

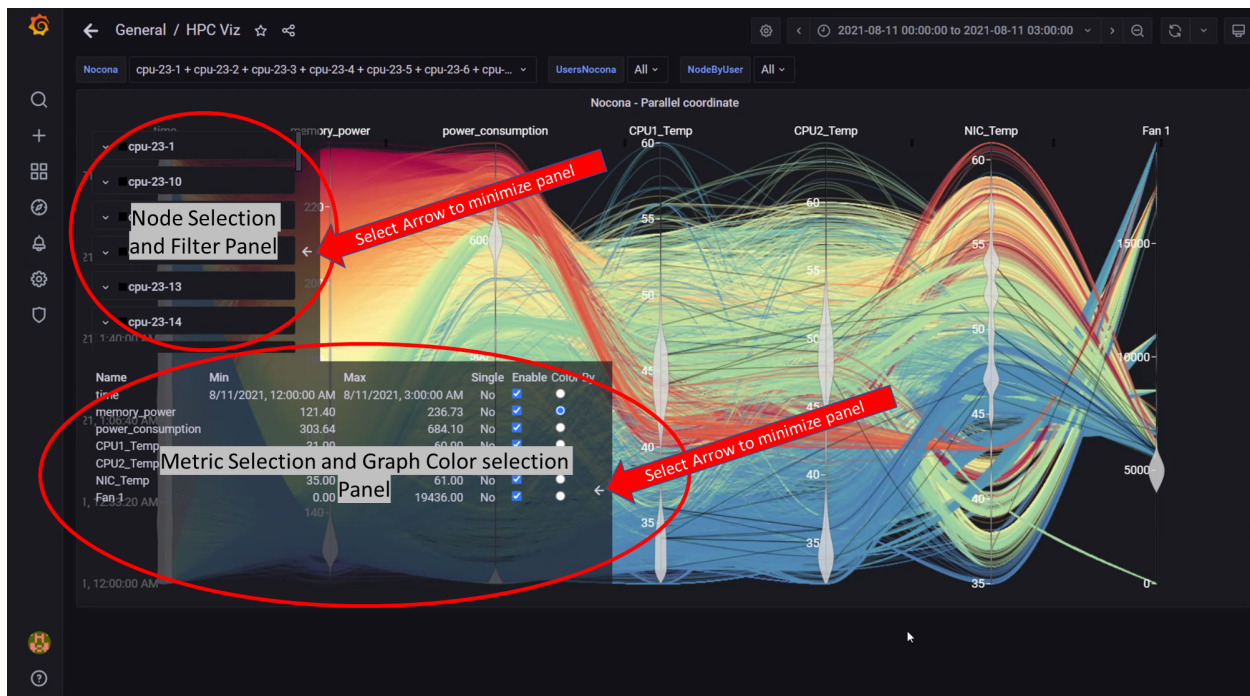
If `idrac_telemetry_support` and `visualisation_support` is set to true, Parallel Coordinate graphs can be used to view system statistics.

Parallel coordinates

Parallel coordinates are a great way to visualize multiple metric dimensions simultaneously to see trends and spot outlier activity. Metrics like CPU temp, Fan Speed, Memory Usage etc. can be added or removed as an additional vertical axis. This implementation of parallel coordinate graphing includes a display of metric value distribution in the form of a violin plot along vertical axes and the ability to interact with the graph to perform filtering. Metric range filtering on one or more axes automatically filters the node and sample list in the top left-hand panel to the nodes and samples that fit the filtering criteria.



In the above image, both left-hand panels are collapsed to allow for a better view of the graph. They can be expanded by clicking on the arrows highlighted in the picture. The expanded panels can be used to customize the graph.



In the above image, both left-hand panels are expanded and can be minimized by clicking on the minimize arrows on the right of each panel. These panels can be used to customize the graphs by:

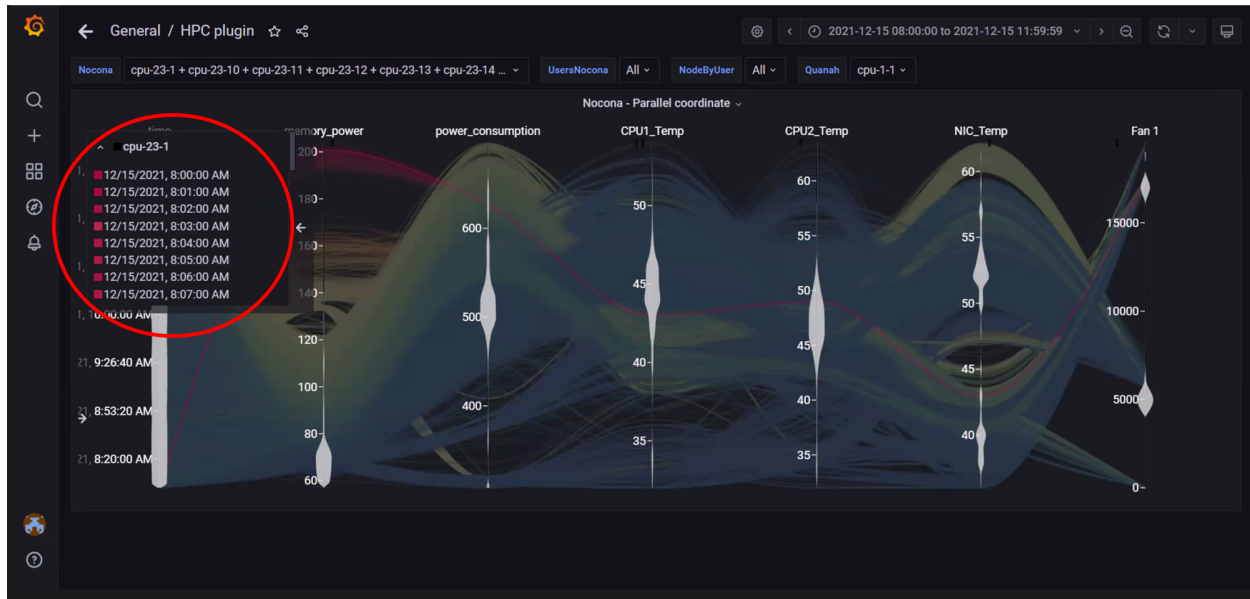
- Filtering by node and node metrics
- Assigning colors to different node metrics



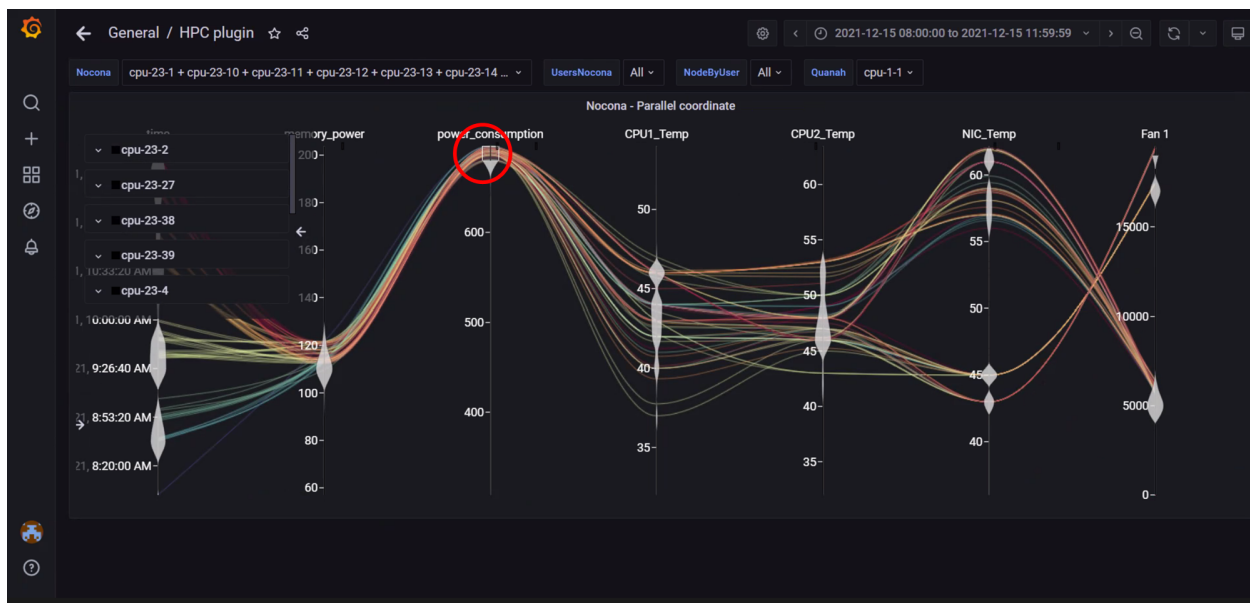
In the above image, the metric **Power Consumption** has been assigned a color to highlight the metric.



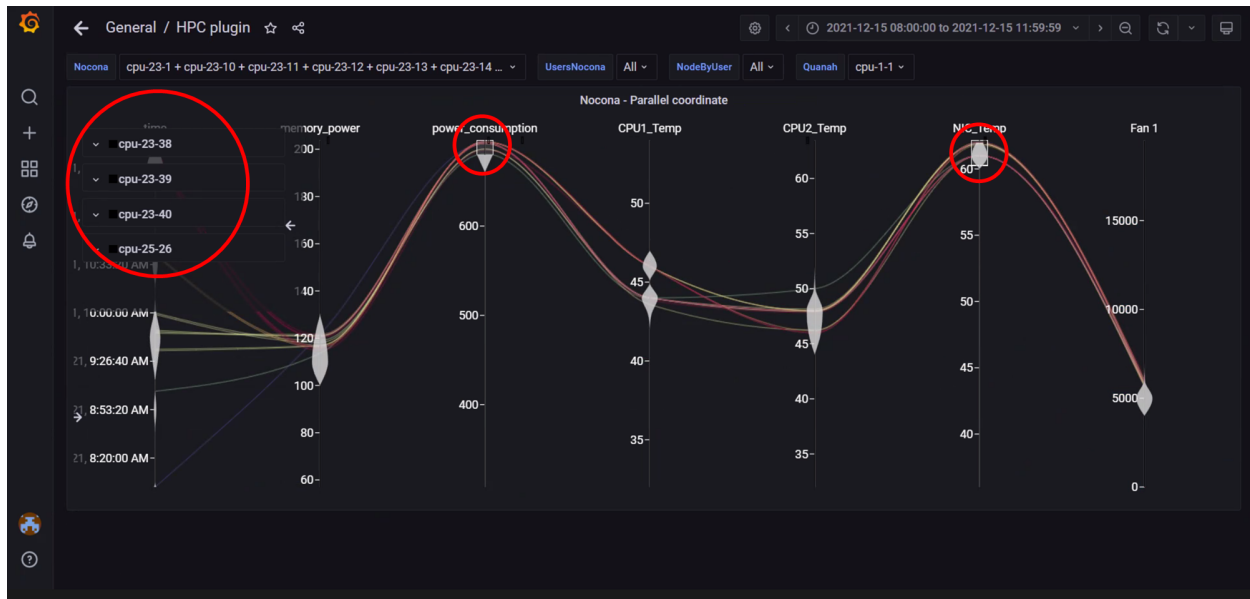
In the above image, data has been filtered by **Node** to get insights into different metrics about specific nodes.



In the above image, data for a single node has been highlighted using the top-left panel.



In the above image, metric filters were applied on **Power Consumption** by clicking on the vertical axis and dragging a filter box over the range of values required. The top left panel will display nodes and samples that fit the filter. Filters are removed by clicking on the vertical dimension axis again.



In the above image, metric filters were applied on **Power Consumption** and **NIC temperature**. Using more than one filter will result in fewer nodes and telemetry samples that meet the filtering criteria.



In the above image, the top-right panel was used to filter data by time, this can be done in 2 ways:

- In absolute yyyy-mm-dd hh:mm:ss format
- In relative time periods such as 'last 5 minutes', 'last 7 days' etc.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

Note: The timestamps used for the time metric are based on the timezone set in `input/provision_config.yml`. In the event of a mismatch between the timezone on the browser being used to access Grafana UI and the timezone in `input/provision_config.yml`, the time range being used to filter information on the Grafana UI will have to be adjusted per the timezone in `input/provision_config.yml`.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.5.2 Metrics collected

Regular metrics

Your cluster in numbers: Regular metrics include information such as CPU, memory, packets errors, drives etc.

Table 6: Regular metrics

Metric Name	Unit	Possible Values	Possible error causes
BlockedProcesses	processes	<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • This could happen if the <code>/proc/stat</code> file is inaccessible.
CPUSystem	seconds	<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • This could happen when the <code>psutil</code> library encounters errors.
CPUWait	seconds	<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • This could happen when the <code>psutil</code> library encounters errors.
ErrorsRecv		<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • This could happen when the <code>psutil</code> library encounters errors.
ErrorsSent		<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • This could happen when the <code>psutil</code> library encounters errors.
FailedJobs		<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • Slurm is not installed.
HardwareCorruptedMemory	kB	<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • This could happen if the <code>/proc/meminfo</code> file is inaccessible.
MemoryActive	bytes	<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • This could happen when the <code>psutil</code> library encounters errors.
MemoryAvailable	bytes	<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • This could happen when the <code>psutil</code> library encounters errors.
MemoryCached	bytes	<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • This could happen when the <code>psutil</code> library encounters errors.
MemoryFree	bytes	<ul style="list-style-type: none"> • Metric Value • No Data 	<ul style="list-style-type: none"> • This could happen when the <code>psutil</code> library encounters errors.

Health metrics

The health of your cluster: Health metrics include key performance indicators.

Table 7: Health metrics

Metric Name	Possible value(s)	Possible failure causes
dmesg	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • [Unknown] The dmesg command was not found on the cluster node. • [Fail] The dmesg command returned an error log message.
beegfs -beegfsstat	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • [Unknown] BeeGFS is not installed or inactive. • [Fail] The BeeGFS client service has failed or the node is not present in reachable lists of BeeGFS clients.
gpu_driver_health:gpu	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_health_nvlink:gpu ¹	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • NVLinks are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_health_pcie:gpu	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_health_pmu:gpu	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_health_power:gpu	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_health_thermal:gpu	<ul style="list-style-type: none"> • Unknown • Metric Value 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
Kubernetespodsstatus	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • Kubernetes is not installed.

GPU metrics

The GPUs of your cluster: GPU metrics include information about GPUs in the cluster

Table 8: GPU metrics

Metric Name	Unit	Possible value(s)	Potential error cause(s)
gpu_temperature:gpu	C	<ul style="list-style-type: none"> • Metric value • No data 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_utilization	percent	<ul style="list-style-type: none"> • Metric value • No data 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_utilization:average	percent	<ul style="list-style-type: none"> • Metric value • No data 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@del.com.

3.5.3 Additional metric information

Table 9: Telemetry metrics

Metric Name	Command	Comments	Aggregation Level
BlockedProcesses	grep procs_blocked /proc/stat		Node Level
CPUSystem	psutil.cpu_times(). system		Node Level
CPUWait	psutil.cpu_times(). iowait		Node Level
ErrorsRecv	psutil. net_io_counters(perni- get(interface_name). errin	Error packets received for individual network inter- faces will be populated.	Node Level

continues on next page

¹ While NVLink metrics are collected as part of our telemetry suite, NVLinks have not been tested for compatibility with Omnia.

Table 9 – continued from previous page

Metric Name	Command	Comments	Aggregation Level
ErrorsSent	psutil. net_io_counters(perni- get(interface_name). errout	Error packets sent for indi- vidual network interfaces will be populated.	Node Level
FailedJobs	sacct -P --delimiter=\t		Cluster Level
HardwareCorruptedMem- ory	grep HardwareCorrupted /proc/meminfo		Node Level
MemoryActive	psutil. virtual_memory(). active		Node Level
MemoryAvailable	psutil. virtual_memory(). available		Node Level
MemoryCached	psutil. virtual_memory(). cached		Node Level
MemoryFree	psutil. virtual_memory(). free		Node Level
MemoryInactive	psutil. virtual_memory(). inactive		Node Level
MemoryPercent	psutil. virtual_memory(). percent		Node Level
MemoryShared	psutil. virtual_memory(). shared		Node Level
MemoryTotal	psutil. virtual_memory(). total		Node Level
MemoryUsed	psutil. virtual_memory(). used		Node Level

continues on next page

Table 9 – continued from previous page

Metric Name	Command	Comments	Aggregation Level
NodesDown	<code>sinfo --format=%N\ t%P\t%a\t%C\t%t\t%D\ t%m</code>	Node is considered down if node state is any of the following: down, drained, draining, fail, failing, future, inval, maint, powered_down, powering_down, unknown, unk. Note: Node state with * in suffix will be considered as down. Example, idle* will be considered as down.	Cluster Level
NodesTotal	<code>sinfo --format=%N\ t%P\t%a\t%C\t%t\t%D\ t%m</code>		Cluster Level
NodesUp	<code>sinfo --format=%N\ t%P\t%a\t%C\t%t\t%D\ t%m</code>	Node is considered up if node state is any of the following: idle, mixed, completing. Note: Node state with * in suffix will be considered as down node. Example, idle* will be considered as down node.	Cluster Level
QueuedJobs	<code>squeue --format=%i\ t%P\t%j\t%u\t%T\t%S\ t%N</code>		Cluster Level
RunningJobs	<code>squeue --format=%i\ t%P\t%j\t%u\t%T\t%S\ t%N</code>		Cluster Level
SMARTHDATemp	<code>smartctl -a <device name></code>		Node Level

continues on next page

Table 9 – continued from previous page

Metric Name	Command	Comments	Aggregation Level
UniqueUserLogin	<code>who cut -f 1 -d " " sort -u wc -l</code>	<ul style="list-style-type: none"> Locally created users via <code>useradd</code> command are also counted in <code>UniqueUserLogin</code> count. Remote logged in LDAP users are not counted in <code>UniqueUserLogin</code> on login nodes. Remote logged in FreeIPA users are counted in <code>UniqueUserLogin</code> on login nodes. 	Login Node/ Manager Node (If Login Node is not present)
dmesg	<code>dmesg --level=err</code>		Node Level
Beegfs-beegfsstat	<code>systemctl is-active beegfs-client beegfs-ctl --nodetype=client --listnodes</code>		Node Level
gpu_driver_health:gpu	<ul style="list-style-type: none"> For NVIDIA GPU: <code>nvidia-smi --query-gpu=driver_version --format=csv, noheader</code> For AMD GPU: <code>rocm-smi --showdriverversion --csv</code> 		Node Level
gpu_health_nvlink:gpu ¹	NVIDIA: <code>nvidia-smi nvlink --status</code>		Node Level
gpu_health_pcie:gpu	<ul style="list-style-type: none"> For NVIDIA GPU: <code>nvidia-smi --query-gpu=pci_bus_id --format=csv, noheader</code> For AMD GPU: <code>rocm-smi --showbus --csv</code> 		Node Level

continues on next page

Table 9 – continued from previous page

Metric Name	Command	Comments	Aggregation Level
gpu_health_pmu:gpu	For NVIDIA GPU: nvidia-smi --query-gpu=power.management --format=csv,nounits	PMU - Power management unit	Node Level
gpu_health_power:gpu	For NVIDIA GPU: nvidia-smi --query-gpu=pci.bus_id --format=csv,nounits	Power consumption	Node Level
gpu_health_thermal:gpu	For AMD GPU: rocm-smi --showbus --csv	GPU temperature health	Node Level
Kubernetespodsstatus	sudo kubectl get pods -A -o json	Value is pass when all pods and containers are in running state, otherwise Fail.	Cluster Level
Kuberneteschildnode	sudo kubectl get nodes -o json	Value is pass when all child nodes are in Ready or Ready,SchedulingDisabled state, otherwise Fail.	Cluster Level
kubernetesnodesstatus	sudo kubectl get nodes -o json	Value is pass when all nodes are in Ready or Ready,SchedulingDisabled state, otherwise Fail.	Cluster Level
kubernetescomponentsstatus	sudo kubectl get --raw=/livez?verbose	Value is Pass when health check is passed in kubectl get --raw=/livez?verbose command , otherwise fail.	Cluster Level
Smart	smartctl -a <device name>		Node Level
gpu_temperature:gpu	<ul style="list-style-type: none"> For NVIDIA GPU: nvidia-smi --query-gpu=temperature.gpu --format=csv,nounits For AMD GPU: rocm-smi -t --csv 		Node Level

continues on next page

Table 9 – continued from previous page

Metric Name	Command	Comments	Aggregation Level
gpu_utilization:	<ul style="list-style-type: none"> For NVIDIA GPU: <code>nvidia-smi nvidia-smi --query-gpu=utilization --format=csv, nounits</code> For AMD GPU: <code>rocm-smi -u --csv</code> 		Node Level
gpu_utilization:average	<ul style="list-style-type: none"> *For NVIDIA GPU: <code>nvidia-smi nvidia-smi --query-gpu=utilization --format=csv, nounits</code> *For AMD GPU: <code>rocm-smi -u --csv</code> 	Value is average of utilization value of all GPUs	Node Level

Note: psutil (python system and process utilities) is a cross-platform library for retrieving information on running processes and system utilization (CPU, memory, network).

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

3.5.4 Timescale DB

Accessing the timescale DB

1. Check the IP of the control plane (ifconfig):

```
3: eno8403: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
↳ default qlen 1000 link/ether b4:45:06:eb:da:4e brd ff:ff:ff:ff:ff:ff
inet 198.168.0.11/24 brd 198.168.0.255 scope global dynamic noprefixroute eno8403
↳ validlft 30884289sec preferred_lft 30884289sec inet6 fe80::b645:6ff:feeb:da4e/64
↳ scope link noprefixroute validlft forever preferredlft forever
```

2. Check the external port on which timescaleDB is running (`kubectl get svc -A`):

```
[root@orchidcp xcat]# kubectl get svc -A
NAMESPACE   NAME                                     TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
calico-apiserver   calico-api                             ClusterIP   10.101.90.148 <none>        443/TCP          13h
calico-kube-controllers-metrics  calico-kube-controllers-metrics      ClusterIP   None          <none>          13h
calico-system      calico-typha                           ClusterIP   10.98.124.122 <none>        5473/TCP         13h
default           kubernetes                             ClusterIP   10.96.0.1     <none>        443/TCP         13h
grafana            grafana                                 LoadBalancer 10.104.75.196 10.5.240.100 5000:30110/TCP   13h
loki               loki                                   ClusterIP   10.98.233.254 <none>        3100/TCP         13h
kube-system        kube-dns                               ClusterIP   10.96.0.10    <none>        53/UDP,53/TCP,9153/TCP 13h
kubernetes-dashboard  dashboard-metrics-scraper             ClusterIP   10.106.196.186 <none>        8000/TCP         13h
kubernetes-dashboard  kubernetes-dashboard                 ClusterIP   10.100.31.32   <none>        443/TCP         13h
metallb-system      webhook-service                       ClusterIP   10.99.39.243   <none>        443/TCP         13h
telemetry-and-visualizations  mysqldb                               ClusterIP   10.103.156.80 <none>        3306/TCP,33060/TCP 13h
telemetry-and-visualizations  timescaledb                           LoadBalancer 10.104.33.121 10.5.240.101 5432:52160/TCP   13h
```

¹ While NVLink metrics are collected as part of our telemetry suite, NVLinks have not been tested for compatibility with Omnia.

3. Connect to DB (`psql -h <EXTERNAL-IP:of timescaledb> -p <timescaledb_port> -U <timescaledb_username> -d telemetry_metrics`)

Note: You will be prompted for the timescaledb password before being given access.

4. Query the database using SQL syntax.

Eg:

```
select * from omnia_telemetry.metrics;
select * from public.timeseries_metrics;
```

telemetry_metrics=# select * from omnia_telemetry.metrics;	id	context	label	value	unit	system	hostname	time
BlockedProcesses	Regular Metric	BlockedProcesses Regular Metric	0	processes	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
CPUSystem	Regular Metric	CPUSystem Regular Metric	183.27	seconds	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
CPUWait	Regular Metric	CPUWait Regular Metric	4.32	seconds	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
ErrorsRecv:eno1	Regular Metric	ErrorsRecv:eno1 Regular Metric	0		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
ErrorsSent:eno1	Regular Metric	ErrorsSent:eno1 Regular Metric	0		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
HardwareCorruptedMemory	Regular Metric	HardwareCorruptedMemory Regular Metric	0	kB	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
MemoryFree	Regular Metric	MemoryFree Regular Metric	69777484288	bytes	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
MemoryTotal	Regular Metric	MemoryTotal Regular Metric	66904104960	bytes	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
MemoryAvailable	Regular Metric	MemoryAvailable Regular Metric	64275017728	bytes	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
MemoryPercent	Regular Metric	MemoryPercent Regular Metric	3.9	percent	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
MemoryUsed	Regular Metric	MemoryUsed Regular Metric	1999437824	bytes	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
MemoryActive	Regular Metric	MemoryActive Regular Metric	1059233792	bytes	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
MemoryInactive	Regular Metric	MemoryInactive Regular Metric	3774611456	bytes	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
MemoryCached	Regular Metric	MemoryCached Regular Metric	4121780224	bytes	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
MemoryShared	Regular Metric	MemoryShared Regular Metric	31686656	bytes	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
SMARTDATemp:/dev/nvme0	Regular Metric	SMARTDATemp:/dev/nvme0 Regular Metric	20	C	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
SMARTDATemp:/dev/nvme1	Regular Metric	SMARTDATemp:/dev/nvme1 Regular Metric	31	C	8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
Dmesg	Health Check Metric	Dmesg Health Check Metric	Fail		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
Baeqfs client Reachable	Health Check Metric	Baeqfs client Reachable Health Check Metric	Unknown		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
smart:/dev/nvme0	Health Check Metric	smart:/dev/nvme0 Health Check Metric	Pass		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
smart:/dev/nvme1	Health Check Metric	smart:/dev/nvme1 Health Check Metric	Pass		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
gpu_health_driver	Health Check Metric	gpu_health_driver Health Check Metric	Unknown		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
gpu_health_nvlink	Health Check Metric	gpu_health_nvlink Health Check Metric	Unknown		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
gpu_health_pcie	Health Check Metric	gpu_health_pcie Health Check Metric	Unknown		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
gpu_health_pmu	Health Check Metric	gpu_health_pmu Health Check Metric	Unknown		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
gpu_health_power	Health Check Metric	gpu_health_power Health Check Metric	Unknown		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
gpu_health_temperature	Health Check Metric	gpu_health_temperature Health Check Metric	Unknown		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
gpu_temperature	GPU Metric	gpu_temperature GPU Metric	No data		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
gpu_utilization	GPU Metric	gpu_utilization GPU Metric	No data		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	
gpu_utilization:average	GPU Metric	gpu_utilization:average GPU Metric	No data		8X697C3	fallpernode00004.fall.test	2023-10-03 16:42:49+00	

images/publictimeseries.png

Data retention policy

The `omnia_telemetry.metrics` has a data retention policy that ensures data is stored for 2 months only. A cleanup job is run everyday to purge metrics older than 60 days.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

LOGGING

4.1 Log management

Use `/etc/logrotate.conf` to customize how often logs are rotated. The default settings for `logrotate.conf` are:

```
cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# use date as a suffix of the rotated file
dateext
# uncomment this if you want your log files compressed
#compress
# RPM packages drop log rotation information into this directory
include /etc/logrotate.d
# system-specific logs may be also be configured here.
```


With the above settings:

- Logs are backed up weekly.
- Data upto 4 weeks old is backed up. Any log backup older than four weeks will be deleted.

Caution: Since these logs take up `/var` space, sufficient space must be allocated to `/var` partition if it's created. If `/var` partition space fills up, control plane might crash.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

4.2 Control plane logs

All log files can be viewed using CLI. Alternatively, most log files can be viewed using the Dashboard tab () on the grafana UI.

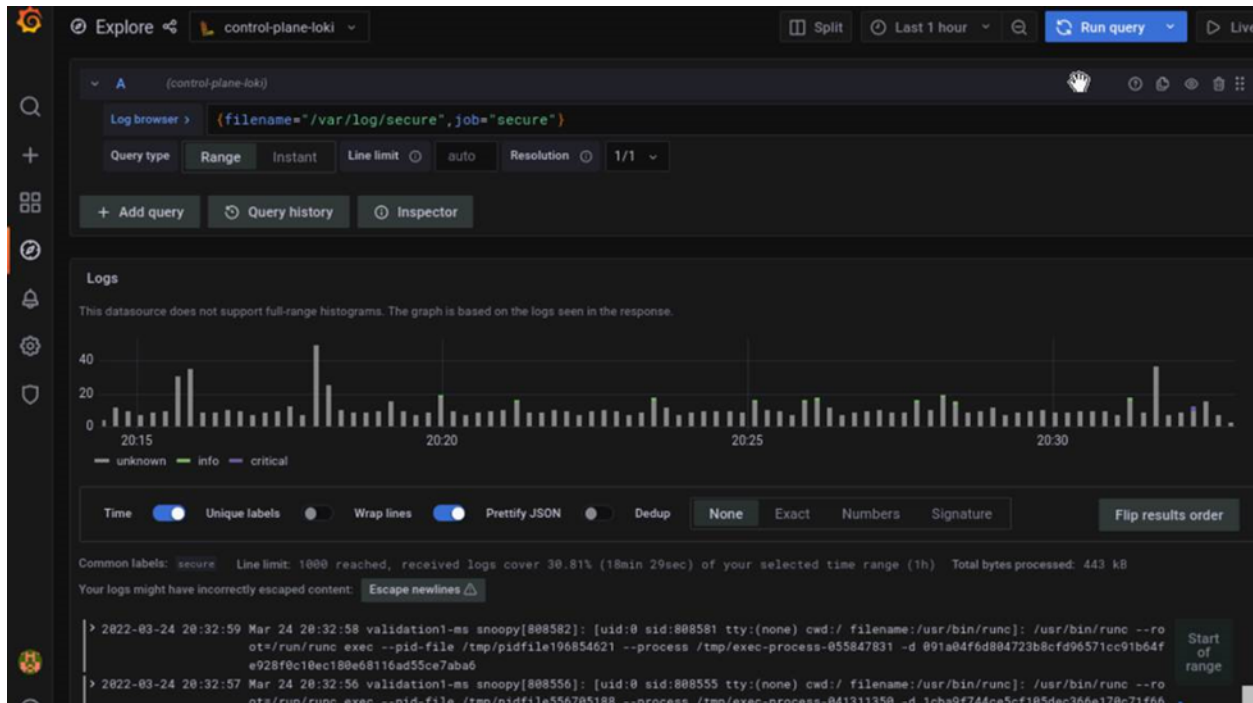
Caution: It is not recommended to delete the below log files or the directories they reside in.

Note: Log files are rotated periodically as a storage consideration. To customize how often logs are rotated, edit the `/etc/logrotate.conf` file on the node.

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

Table 1: Log files

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.
Accelerator Logs	/var/log/omnia/accel	Accelerator Log	This log is configured by Default.
Monitor Logs	/var/log/omnia/moni	Monitor Log	This log is configured by Default.
Network Logs	/var/log/omnia/netwo	Network Log	This log is configured by Default.
Platform Logs	/var/log/omnia/platfo	Platform Log	This log is configured by Default.
Provision Logs	/var/log/omnia/provi	Provision Log	This log is configured by Default.
Scheduler Logs	/var/log/omnia/sched	Scheduler Log	This log is configured by Default.
Security Logs	/var/log/omnia/secur	Security Log	This log is configured by Default.
Storage Logs	/var/log/omnia/stora	Storage Log	This log is configured by Default.
Telemetry Logs	/var/log/omnia/telem	Telemetry Log	This log is configured by Default.
Utils Logs	/var/log/omnia/utills	Utils Log	This log is configured by Default.
Cluster Utilities Logs	/var/log/omnia/utills_	Cluster Utils Log	This log is configured by Default.
syslogs	/var/log/messages	System Logging	This log is configured by Default.
Audit Logs	/var/log/audit/audit.l	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/transaction	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').
HTTPD logs	/var/log/httpd/ *	FreeIPA API Calls	This log is available when FreeIPA or 389ds is set up (ie when enable_security_support is set to 'true').
DNF logs	/var/log/dnf.log	Installation Logs	This log is configured on Rocky OS.
BeeGFS Logs	/var/log/beegfs-client.log	BeeGFS Logs	This log is configured on BeeGFS client nodes.
Compute Logs	/var/log/xcat/comput	Logs system messages from all cluster nodes.	This log is configured by Default.
Cluster deployment logs	/var/log/xcat/cluster.l	Logs deployment messages from all cluster nodes.	This log is configured by Default.



4.3 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>`

4.4 Provisioning logs

Logs pertaining to actions taken during `discovery_provision.yml` can be viewed in `/var/log/xcat/cluster.log` and `/var/log/xcat/computes.log` on the control plane.

Note: As long as a node has been added to a cluster by Omnia, deployment events taking place on the node will be updated in `/var/log/xcat/cluster.log`.

4.5 Telemetry logs

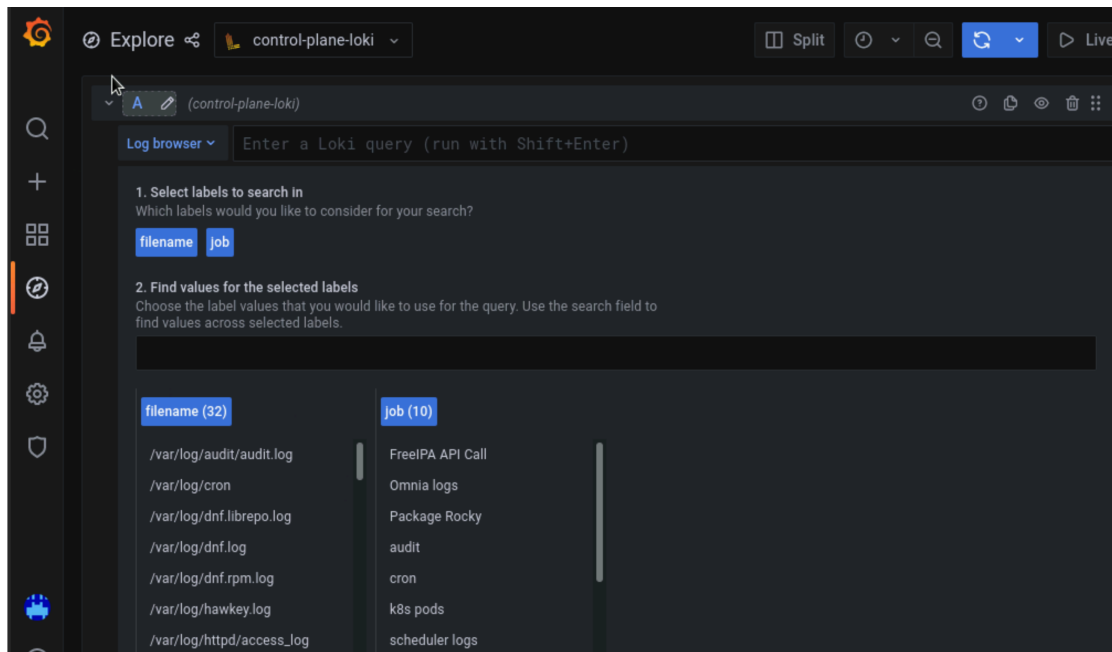
Logs pertaining to actions taken by Omnia or iDRAC telemetry can be viewed in `/var/log/messages`. Each log entry is tagged “omnia_telemetry”. Log entries typically follow this format.

```
<Date time> <Node name> omnia_telemetry[<Process ID>]: <name of file>:<name of method_
→throwing error>: <Error message>
```

4.6 Grafana Loki

After `telemetry.yml` is run, Grafana services are installed on the control plane.

- i. Get the Grafana IP using `kubectl get svc -n grafana`.
- ii. Login to the Grafana UI by connecting to the cluster IP of grafana service via port 5000. That is `http://xx.xx.xx.xx:5000/login`.
- iii. In the Explore page, select **control-plane-loki**.



- iv. The log browser allows users to filter logs by job, node, user, etc. Example

```
(job= "cluster deployment logs") |= "nodename"
(job="compute log messages") |= "nodename" |= "node_username"
```

Custom dashboards can be created as per your requirement.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

TROUBLESHOOTING

5.1 Known issues

Why doesn't my newly discovered server list a MAC ID in the cluster.nodeinfo table?

Due to internal MAC ID conflicts on the target nodes, the MAC address will be listed against the target node using this format `MAC ADDRESS 1 | MAC ADDRESS 2! *NOIP*` in the xCAT node object.

```
[root@... ]# lsdef ... | grep mac
mac=f4:02:70:b8:cc:80|f4:02:70:f1:7e:a3!*NOIP*
```

Why does the task Assign admin NIC IP fail during discovery_provision.yml with errors?

```
SK [provision_validation : Failed - Assign admin nic IP] *****
sk path: /root/shubh/omnia/provision/roles/provision_validation/tasks/assign_admin_nic_config.yml:25
tal: [localhost]: FAILED! => {"changed": false, "msg": "Failed. Please assign admin nic IP. "}

SK [provision_validation : Failed - Assign admin nic IP] *****
sk path: /root/shubh/omnia/provision/roles/provision_validation/tasks/assign_admin_nic_config.yml:37
tal: [localhost]: FAILED! => {"changed": false, "msg": "Failed. Please assign admin nic IP. "}

AY RECAP *****
calhost : ok=33 changed=4 unreachable=0 failed=1 skipped=8 rescued=1 ignored=0
```

Potential Cause: Omnia validates the admin NIC IP on the control plane. If the user has not assigned an admin NIC IP in case of dedicated network interface type, an error message is returned. There is a parsing logic that is being applied on the blank IP and hence, the error displays twice.

Resolution: Ensure a control plane IP is assigned to the admin NIC.

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

Potential Causes:

1. The server hardware does not allow for auto rebooting
2. The process of PXE booting the node has stalled.

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 [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, HTTP and JSON gateway must be enabled. To verify that they are enabled, run:

To check if HTTP is enabled: `show web`

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

To correct the issue, run:

To enable the HTTP gateway: `web http enable`

To enable the JSON gateway: `json-gw enable`

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 cluster 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 does running local_repo.yml fail with connectivity errors?

Potential Cause: The control plane was unable to reach a required online resource due to a network glitch.

Resolution: Verify all connectivity and re-run the playbook.

Why does any script that installs software fail with “The checksum for <software repository path> did not match.”?

Potential Cause: A local repository for the software was not configured by `local_repo.yml`.

Resolution:

- Delete the tarball/image/deb of the software from `<repo_path>/cluster/tarball`.
- Re-run `local_repo.yml`.
- Re-run the script to install the software.

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:

- Ensure that the `docker_username` and `docker_password` are provided in `input/provision_config_credentials.yml`.
- For a 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.](#)

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’ get 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 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 `kube_control_plane` to get the real-time k8s cluster status.
- `kubectl get pods all-namespaces` on the `kube_control_plane` to check which the pods are in the **Running** state.
- `kubectl cluster-info` on the `kube_control_plane` 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. List k8s in `input/software_config.json` and re-run `omnia.yml`.

What to do if pulling the Kserve inference model fail with “Unable to fetch image “kserve/sklearnserver:v0.11.2”: failed to resolve image to digest: Get “https://index.docker.io/v2/”: dial tcp 3.219.239.5:443: i/o timeout.”?

1. Edit the kubernetes configuration map:

```
kubectrl edit configmap -n knative-serving config-deployment
```

2. Add docker.io and index.docker.io as part of the registries-skipping-tag-resolving.

For more information, [click here](#).

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 not case-sensitive. In other words, omnia.test is identical to OMNIA.TEST and Omnia.test.

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 (For example, dvader.empire.gov)

What to do when KubeFlow pods are in ‘ImagePullBackOff’ or ‘ErrImagePull’ status after executing kube-flow.yml?

Potential Cause: Your Docker pull limit has been exceeded. For more information, [click here](#).

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

What to do when omnia.yml fails while completing the security role, and returns the following error message: ‘Error: kinit: Connection refused while getting default cache’?

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 kube_control_plane 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 cluster 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 slurm_control_node

systemctl restart slurmdbd on slurm_control_node

systemctl restart slurmd on slurm_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 have been 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 execution 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 `configure registry: Start and enable nerdctl-registry service` fail with “Job for nerdctl-registry.service failed because the control process exited with error code”?

```
TASK [configure_registry : Start and enable nerdctl-registry service] *****
Saturday 16 March 2024 08:58:48 +0000 (0:00:00.500) 0:00:47.541 *****
fatal: [localhost]: FAILED! => {"changed": false, "msg": "Unable to start service nerdctl-registry: Job for nerdctl-registry.service failed because the control process exited with error code.\nSee \"systemctl status nerdctl-registry.service\" and \"journalctl -xeu nerdctl-registry.service\" for details.\n"}

TASK [configure_registry : Failed to start nerdctl-registry service] *****
Saturday 16 March 2024 08:58:54 +0000 (0:00:05.807) 0:00:53.348 *****
fatal: [localhost]: FAILED! => {"changed": false, "msg": "Failed to initiate nerdctl-registry service."}

PLAY RECAP *****
localhost : ok=128 changed=14 unreachable=0 failed=1 skipped=72 rescued=1 ignored=0

Saturday 16 March 2024 08:58:54 +0000 (0:00:00.015) 0:00:53.364 *****
configure_registry : Extract nerdctl archive ..... 8.56s
configure_registry : Start and enable nerdctl-registry service ..... 5.81s
validation : Install Python packages ..... 2.41s
configure_registry : Docker login ..... 2.35s
configure_registry : Download nerdctl archive ..... 2.27s
validation : Install jq package ..... 2.16s
```

Potential Cause:

- The subnet 10.4.0.0/24 has been assigned to the admin, bmc, or additional network. nerdctl uses this subnet by default and cannot be assigned to any other interface in the system.
- The docker pull limit has been breached.

Resolution:

- Reassign the conflicting network to a different subnet.
- Update `input/provision_config_credentials.yml` with the `docker_username` and `docker_password`.

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)

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 using: `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 the message ‘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`.

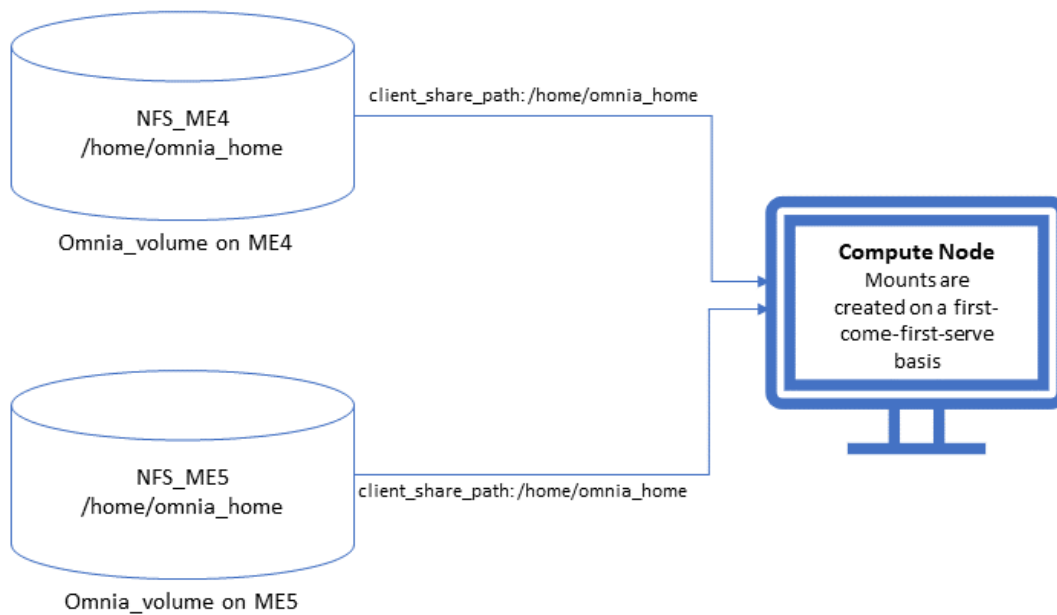
Resolution:

- Ensure that the input paths are a perfect match 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.



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-mgmt`, `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](#) 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 would FreeIPA server/client installation fail? (version 1.5 and below)

Potential Cause:

The hostnames of the auth server nodes are not configured 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, *authserver_node.omnia.test* is a valid hostname for the auth server node.

Note: To find the cause for the failure of the FreeIPA server and client installation, see *ipaserver-install.log* in the auth server.

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 kube_control_plane: `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
```

Why do Kuberneteschildnode & kubernetesnodes log as Pass in the database even if there are nodes in the Ready,Schedulingdisabled state?

Potential Cause: Omnia telemetry considers Ready,SchedulingDisabled as a Ready state of Kubernetes nodes . So, even if the `kubectl get nodes` command shows any node’s state as Ready,SchedulingDisabled, the entry in DB for Kuberneteschildnode & kubernetesnodes will be logged as Pass instead of Fail.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@del.com.

5.2 Frequently asked questions

Why is the provisioning status of the target servers stuck at ‘installing’ in cluster.nodeinfo (omniadb)?

```
omniadb=# select id,node,hostname,admin_ip,status from cluster.nodeinfo;
id | node | hostname | admin_ip | status
---+---+-----+-----+-----
2 | node1 | node1.orchid.cluster | 10.5.0.1 |
6 | control_plane | orchidcp.orchid.cluster | 10.5.255.254 |
3 | node2 | node2.orchid.cluster | 10.5.0.2 | installing
4 | node3 | node3.orchid.cluster | 10.5.0.3 | installing
5 | node4 | node4.orchid.cluster | 10.5.0.4 | installing
(5 rows)
```

```

* When reporting a bug add logs from /tmp as separate text/plain attachments
21:59:02 Running pre-installation scripts
21:59:08 Not asking for UNC because of an automated install
21:59:08 Not asking for UNC because text mode was explicitly asked for in kickstart
Starting automated install...Saving storage configuration...
Checking storage configuration...
.

=====
=====
Installation

1) [x] Language settings                2) [x] Time settings
   (English (United States))           (CST6CDT timezone)
3) [!] Installation source              4) [!] Software selection
   (Error setting up software          (Error checking software
   source)                             selection)
5) [x] Installation Destination        6) [x] Kdump
   (Custom partitioning selected)      (Kdump is enabled)
7) [x] Network configuration           8) [ ] User creation
   (Wired (eno3) connected)           (No user will be created)

Please make a selection from the above ['b' to begin installation, 'q' to quit,
'r' to refresh]:

```

Potential Causes:

- Disk partition may not have enough storage space per the requirements specified in input/provision_config (under disk_partition)
- The provided ISO may be corrupt/incomplete.
- Hardware issues (Auto reboot may fail at POST)
- A virtual disk may not have been created

Resolution:

- Add more space to the server or modify the requirements specified in input/provision_config (under disk_partition)
- Download the ISO again, verify the checksum/ download size and re-run the provision tool.
- Resolve/replace the faulty hardware and PXE boot the node.
- Create a virtual disk and PXE boot the node.

Why is the provisioning status of my target servers stuck at ‘powering-on’ in the cluster.info (omniadb)?**Potential Cause:**

- Hardware issues (Auto-reboot may fail due to hardware tests failing)
- The target node may already have an OS and the first boot PXE device is not configured correctly.

Resolution:

- Resolve/replace the faulty hardware and PXE boot the node.
- Target servers should be configured to boot in PXE mode with the appropriate NIC as the first boot device.

What to do if PXE boot fails while discovering target nodes via switch_based discovery with provisioning status stuck at ‘powering-on’ in cluster.nodeinfo (omniadb):

```

xcat.genesis.dodiscovery: Beginning echo information to discovery packet file...
xcat.genesis.dodiscovery: Discovery packet file is ready.
xcat.genesis.dodiscovery: Sending the discovery packet to xCAT (172.59.255.254:3001)...
xcat.genesis.dodiscovery: Sleeping 5 seconds...
xcat.genesis.minixcatd: The request is processing by xCAT master...
xcat.genesis.minixcatd: The request is already processed by xCAT master, but not matched.
xcat.genesis.dodiscovery: Beginning echo information to discovery packet file...
xcat.genesis.dodiscovery: Discovery packet file is ready.
xcat.genesis.dodiscovery: Sending the discovery packet to xCAT (172.59.255.254:3001)...
xcat.genesis.dodiscovery: Sleeping 5 seconds...
xcat.genesis.minixcatd: The request is processing by xCAT master...
xcat.genesis.minixcatd: The request is already processed by xCAT master, but not matched.
xcat.genesis.dodiscovery: Beginning echo information to discovery packet file...
xcat.genesis.dodiscovery: Discovery packet file is ready.
xcat.genesis.dodiscovery: Sending the discovery packet to xCAT (172.59.255.254:3001)...
xcat.genesis.dodiscovery: Sleeping 5 seconds...
xcat.genesis.minixcatd: The request is processing by xCAT master...
xcat.genesis.minixcatd: The request is already processed by xCAT master, but not matched.
xcat.genesis.dodiscovery: Beginning echo information to discovery packet file...
xcat.genesis.dodiscovery: Discovery packet file is ready.
xcat.genesis.dodiscovery: Sending the discovery packet to xCAT (172.59.255.254:3001)...
xcat.genesis.dodiscovery: Sleeping 5 seconds...
xcat.genesis.minixcatd: The request is processing by xCAT master...
xcat.genesis.minixcatd: The request is already processed by xCAT master, but not matched.
xcat.genesis.dodiscovery: Beginning echo information to discovery packet file...
xcat.genesis.dodiscovery: Discovery packet file is ready.
xcat.genesis.dodiscovery: Sending the discovery packet to xCAT (172.59.255.254:3001)...
xcat.genesis.dodiscovery: Sleeping 5 seconds...
xcat.genesis.minixcatd: The request is processing by xCAT master...
xcat.genesis.minixcatd: The request is already processed by xCAT master, but not matched.
xcat.genesis.dodiscovery: Beginning echo information to discovery packet file...
xcat.genesis.dodiscovery: Discovery packet file is ready.

```

1. Rectify any probable causes like incorrect/unavailable credentials (switch_snmp3_username and switch_snmp3_password provided in input/provision_config.yml), network glitches, having multiple NICs with the same IP address as the control plane, or incorrect switch IP/port details.
2. Run the clean up script by:

```

cd utils
ansible-playbook control_plane_cleanup.yml

```

3. Re-run the provision tool (ansible-playbook discovery_provision.yml).

What to do if playbook execution fails due to external (network, hardware etc) failure:

Re-run the playbook whose execution failed once the issue is resolved.

Why don't IPA commands work after setting up FreeIPA on the cluster?

Potential Cause:

Kerberos authentication may be missing on the target node.

Resolution:

Run `kinit admin` on the node and provide the `kerberos_admin_password` when prompted. (This password is also entered in `input/security_config.yml`.)

Why am I unable to login using LDAP credentials after successfully creating a user account?

Potential Cause:

Whitespaces in the LDIF file may have caused an encryption error. Verify whether there are any whitespaces in the file by running `cat -vet <filename>`.

Resolution:

Remove the whitespaces and re-run the LDIF file.

Why are the status and admin_mac fields not populated for specific target nodes in the cluster.nodeinfo table?

Causes:

- Nodes do not have their first PXE device set as designated active NIC for PXE booting.
- Nodes that have been discovered via multiple discovery mechanisms may list multiple times. Duplicate node entries will not list MAC addresses.

Resolution:

- Configure the first PXE device to be active for PXE booting.
- PXE boot the target node manually.
- Duplicate node objects (identified by service tag) will be deleted automatically. To manually delete node objects, use `utils/delete_node.yml`.

What to do if user login fails when accessing a cluster node:

```
[root@springcp ~]# ssh user1@10.27.0.2
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:zJq2bCTNDR+rR3zi1Kw3kHxYyPXZI50EyEvByGwh6Eg.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /root/.ssh/known_hosts:20
Password authentication is disabled to avoid man-in-the-middle attacks.
Keyboard-interactive authentication is disabled to avoid man-in-the-middle attacks.
user1@10.27.0.2: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

Potential Cause:

- ssh key on the control plane may be outdated.

Resolution:

- Refresh the key using `ssh-keygen -R <hostname/server IP>`.
- Retry login.

Why does the ‘Fail if LDAP home directory exists’ task fail during `user_passwordless_ssh.yml`?

```
TASK [passwordless_ssh_ldap : Initialize username] *****
ok: [10.5.0.7]

TASK [passwordless_ssh_ldap : Check ldap home directory exists - user1] *****
ok: [10.5.0.7]

TASK [passwordless_ssh_ldap : Fail if ldap home directory not exists - user1] *****
fatal: [10.5.0.7]: FAILED! => {"changed": false, "msg": "Failed. ldap home directory /home/omnia-share//user1 not present for the user user1 in ldap server. Login to one of the nodes other than ldap server as the user so that home directory is created. Make sure ldap home directory /home/omnia-share/ mounted as nfs share in ldap server."}

PLAY RECAP *****
10.5.0.1      : ok=22  changed=4    unreachable=0    failed=0    skipped=15    rescued=0     ignored=0
10.5.0.2      : ok=14  changed=4    unreachable=0    failed=0    skipped=2     rescued=0     ignored=0
10.5.0.7      : ok=8   changed=0    unreachable=0    failed=1    skipped=1     rescued=0     ignored=0
localhost    : ok=16  changed=2    unreachable=0    failed=0    skipped=6     rescued=0     ignored=0
```

Potential Cause: The required NFS share is not set up on the control plane.

Resolution:

If `enable_omnia_nfs` is true in `input/omnia_config.yml`, follow the below steps to configure an NFS share on your LDAP server:

- From the `kube_control_plane`:
 1. Add the LDAP server IP address to `/etc/exports`.
 2. Run `exportfs -ra` to enable the NFS configuration.
- From the LDAP server:

1. Add the required fstab entries in /etc/fstab (The corresponding entry will be available on the compute nodes in /etc/fstab)
2. Mount the NFS share using mount manager_ip: /home/omnia-share /home/omnia-share

Why does the ‘Import SCP from a local path’ task fail during idrac.yml?

```
TASK [provision_idrac : Import SCP from a local path and wait for this job to get completed] ***
FAILED - RETRYING: [165.29.0.106]: Import SCP from a local path and wait for this job to get completed (5 retries left).
FAILED - RETRYING: [165.29.0.106]: Import SCP from a local path and wait for this job to get completed (4 retries left).
FAILED - RETRYING: [165.29.0.106]: Import SCP from a local path and wait for this job to get completed (3 retries left).
FAILED - RETRYING: [165.29.0.106]: Import SCP from a local path and wait for this job to get completed (2 retries left).
FAILED - RETRYING: [165.29.0.106]: Import SCP from a local path and wait for this job to get completed (1 retries left).
fatal: [165.29.0.106]: FAILED! => {"attempts": 5, "changed": false, "msg": "Failed to import scp.", "scp_status": {"Data": {"Message": "returned status code doesn't match with the expected success code", "Status": "Failed", "StatusCode": 503}, "Message": "none", "Status": "Failed", "StatusCode": 503, "error": {"error": {"@Message.ExtendedInfo": [{"Message": "A job operation is already running. Retry the operation after the existing job is completed.", "MessageArgs": [], "MessageArgs@odata.count": 0, "MessageId": "IDRAC.2.3.RAC0679", "RelatedProperties": [], "RelatedProperties@odata.count": 0, "Resolution": "Wait until the running job is completed or delete the scheduled job and retry the operation.", "Severity": "Warning"}]}}, "code": "Base.1.7.GeneralError", "message": "A general error has occurred. See ExtendedInfo for more information"}}, "file": "/root/omnia/control_plane/roles/provision_idrac/files/idrac_scp.xml" "retval": true}}
```

Potential Cause: The target server may be stalled during the booting process.

Resolution: Bring the target node up and re-run the script.

Why is the node status stuck at ‘powering-on’ or ‘powering-off’ after a control plane reboot?

Potential Cause: The nodes were powering off or powering on during the control plane reboot/shutdown.

Resolution: In the case of a planned shutdown, ensure that the control plane is shut down after the compute nodes. When powering back up, the control plane should be powered on and xCAT services resumed before bringing up the compute nodes. In short, have the control plane as the first node up and the last node down.

For more information, [click here](#)

Why do subscription errors occur on RHEL control planes when rhel_repo_local_path (in input/provision_config.yml) is not provided and control plane does not have an active subscription?

```
TASK [provision_validation : Subscription is not enabled] *****
task path: /root/omnia/provision/roles/provision_validation/tasks/validate_repo_path.yml:79
fatal: [localhost]: FAILED! => {"changed": false, "msg": "Failed. RedHat subscription not active. Activate RedHat subscription or provide repos details in rhel_repo_path variable in provision_config.yml"}
```

For many of Omnia’s features to work, RHEL control planes need access to the following repositories:

1. AppStream
2. BaseOS

This can only be achieved using local repos specified in rhel_repo_local_path (input/provision_config.yml).

Note: To enable the repositories, run the following commands:

```
subscription-manager repos --enable=codeready-builder-for-rhel-8-x86_64-rpms
subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
```

Verify your changes by running:

```
yum repolist enabled
```

Why does the task: Initiate reposync of AppStream, BaseOS and CRB fail?

Potential Cause: The `repo_url`, `repo_name` or `repo` provided in `rhel_repo_local_path` (`input/provision_config.yml`) may not have been valid.

Omnia does not validate the input of `rhel_repo_local_path`.

Resolution: Ensure the correct values are passed before re-running `discovery_provision.yml`.

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 `discovery_provision.yml`.
2. Using the switch IP:
 - Run `discovery_provision.yml` once the switch has discovered the potential new node.

Why does the task: ‘BeeGFS: Rebuilding BeeGFS client module’ fail?

[illegible]

Potential Cause: BeeGFS version 7.3.0 is in use.

Resolution: Use BeeGFS client version 7.3.1 when setting up BeeGFS on the cluster.

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.

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.

Why does the task: ‘Orchestrator: Deploy MetalLB IP Address pool’ fail?

```
TASK [orchestrator : Deploy MetalLB IP Address Pool] *****
task path: /root/test/devel-omnia/telemetry/roles/orchestrator/tasks/configure_metallb.yml:46
fatal: [localhost]: FAILED! => {"changed": true, "cmd": ["kubectl", "apply", "-f", "/var/lib/ipaddresspool.yaml"], "delta": "0:00:01.263525", "end": "2023-09-10 00:55:38.006362", "msg": "non-zero return code", "rc": 1, "start": "2023-09-10 00:55:36.742837", "stderr": "Error from server (InternalError): error when creating \"/var/lib/ipaddresspool.yaml\": Internal error occurred: failed calling webhook \"ipaddresspoolvalidationwebhook.metallb.io\": Post \"https://webhook-service.metallb-system.svc:443/validate-metallb-io-v1beta1-ipaddresspool?timeout=10s\": dial tcp 10.98.250.10:443: connect: connection refused\", \"stderr_lines\": [\"Error from server (InternalError): error when creating \"/var/lib/ipaddresspool.yaml\": Internal error occurred: failed calling webhook \"ipaddresspoolvalidationwebhook.metallb.io\": Post \"https://webhook-service.metallb-system.svc:443/validate-metallb-io-v1beta1-ipaddresspool?timeout=10s\": dial tcp 10.98.250.10:443: connect: connection refused\"], \"stdout\": \"\", \"stdout_lines\": []}
```

Potential Cause: /var partition is full (potentially due to images not being cleared after intel-oneapi images docker images are used to execute benchmarks on the cluster using aptainer support) .

Resolution: Clear the /var partition and retry telemetry.yml.

Why does the task: [Telemetry]: TASK [grafana : Wait for grafana pod to come to ready state] fail with a timeout error?

Potential Cause: Docker pull limit exceeded.

Resolution: Manually input the username and password to your docker account on the control plane.

Is provisioning servers using BOSS controller supported by Omnia?

From Omnia 1.2.1, provisioning a server using BOSS controller is supported.

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](#) .

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

5.3 Troubleshooting guide

5.3.1 Troubleshooting Kubeadm

For a complete guide to troubleshooting kubeadm, [click here](#).

5.3.2 Connecting to internal databases

- **TimescaleDB**

- Start a bash session within the timescaledb pod: `kubectl exec -it pod/timescaledb-0 -n telemetry-and-visualizations -- /bin/bash`
- Connect to psql: `psql -U <postgres_username>`
- Connect to database: `\c telemetry_metrics`

- **MySQL DB**

- Start a bash session within the mysqldb pod: `kubectl exec -it pod/mysqldb-0 -n telemetry-and-visualizations -- /bin/bash`
- Connect to mysql: `mysql -U <mysqldb_username> -p <mysqldb_password>`
- Connect to database: `USE idrac_telemetrysource_services_db`

5.3.3 Checking and updating encrypted parameters

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

```
cd input/
```

2. To view the encrypted parameters:

```
ansible-vault view provision_config_credentials.yml --vault-password-file .
↪provision_vault_key
```

3. To edit the encrypted parameters:

```
ansible-vault edit provision_config_credentials.yml --vault-password-file .
↪provision_vault_key
```

5.3.4 Checking pod status on the control plane

- Use this command to get a list of all available pods: `kubectl get pods -A`
- Check the status of any specific pod by running: `kubectl describe pod <pod name> -n <namespace name>`

5.3.5 Using telemetry information to diagnose node issues

Table 1: Regular telemetry metrics

Metric Name	Unit	Possible Values	Possible error causes
BlockedProcesses	processes	<ul style="list-style-type: none"> Metric Value No Data 	<ul style="list-style-type: none"> This could happen if the <code>/proc/stat</code> file is inaccessible.
CPUSystem	seconds	<ul style="list-style-type: none"> Metric Value No Data 	<ul style="list-style-type: none"> This could happen when the <code>psutil</code> library encounters errors.
CPUWait	seconds	<ul style="list-style-type: none"> Metric Value No Data 	<ul style="list-style-type: none"> This could happen when the <code>psutil</code> library encounters errors.
ErrorsRecv		<ul style="list-style-type: none"> Metric Value No Data 	<ul style="list-style-type: none"> This could happen when the <code>psutil</code> library encounters errors.
ErrorsSent		<ul style="list-style-type: none"> Metric Value No Data 	<ul style="list-style-type: none"> This could happen when the <code>psutil</code> library encounters errors.
FailedJobs		<ul style="list-style-type: none"> Metric Value No Data 	<ul style="list-style-type: none"> Slurm is not installed.
HardwareCorruptedMemory	kB	<ul style="list-style-type: none"> Metric Value No Data 	<ul style="list-style-type: none"> This could happen if the <code>/proc/meminfo</code> file is inaccessible.
MemoryActive	bytes	<ul style="list-style-type: none"> Metric Value No Data 	<ul style="list-style-type: none"> This could happen when the <code>psutil</code> library encounters errors.
MemoryAvailable	bytes	<ul style="list-style-type: none"> Metric Value No Data 	<ul style="list-style-type: none"> This could happen when the <code>psutil</code> library encounters errors.
MemoryCached	bytes	<ul style="list-style-type: none"> Metric Value No Data 	<ul style="list-style-type: none"> This could happen when the <code>psutil</code> library encounters errors.

5.3. Troubleshooting guide

Table 2: Health telemetry metrics

Metric Name	Possible value(s)	Possible failure causes
dmesg	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • [Unknown] The dmesg command was not found on the cluster node. • [Fail] The dmesg command returned an error log message.
beegfs -beegfsstat	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • [Unknown] BeeGFS is not installed or inactive. • [Fail] The BeeGFS client service has failed or the node is not present in reachable lists of BeeGFS clients.
gpu_driver_health:gpu	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_health_nvlink:gpu ¹	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • NVLinks are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_health_pcie:gpu	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_health_pmu:gpu	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_health_power:gpu	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_health_thermal:gpu	<ul style="list-style-type: none"> • Unknown • Metric Value 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
Kubernetespodsstatus	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • Kubernetes is not installed.
192		Chapter 5. Troubleshooting
Kuberneteschildnode	<ul style="list-style-type: none"> • Unknown • Fail • Pass 	<ul style="list-style-type: none"> • Kubernetes is not installed.

Table 3: GPU telemetry metrics

Metric Name	Unit	Possible value(s)	Potential error cause(s)
gpu_temperature:gpu	C	<ul style="list-style-type: none"> • Metric value • No data 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_utilization	percent	<ul style="list-style-type: none"> • Metric value • No data 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.
gpu_utilization:average	percent	<ul style="list-style-type: none"> • Metric value • No data 	<ul style="list-style-type: none"> • AMD/NVIDIA accelerators are not present. • GPU drivers are not installed including Rocm and CUDA.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

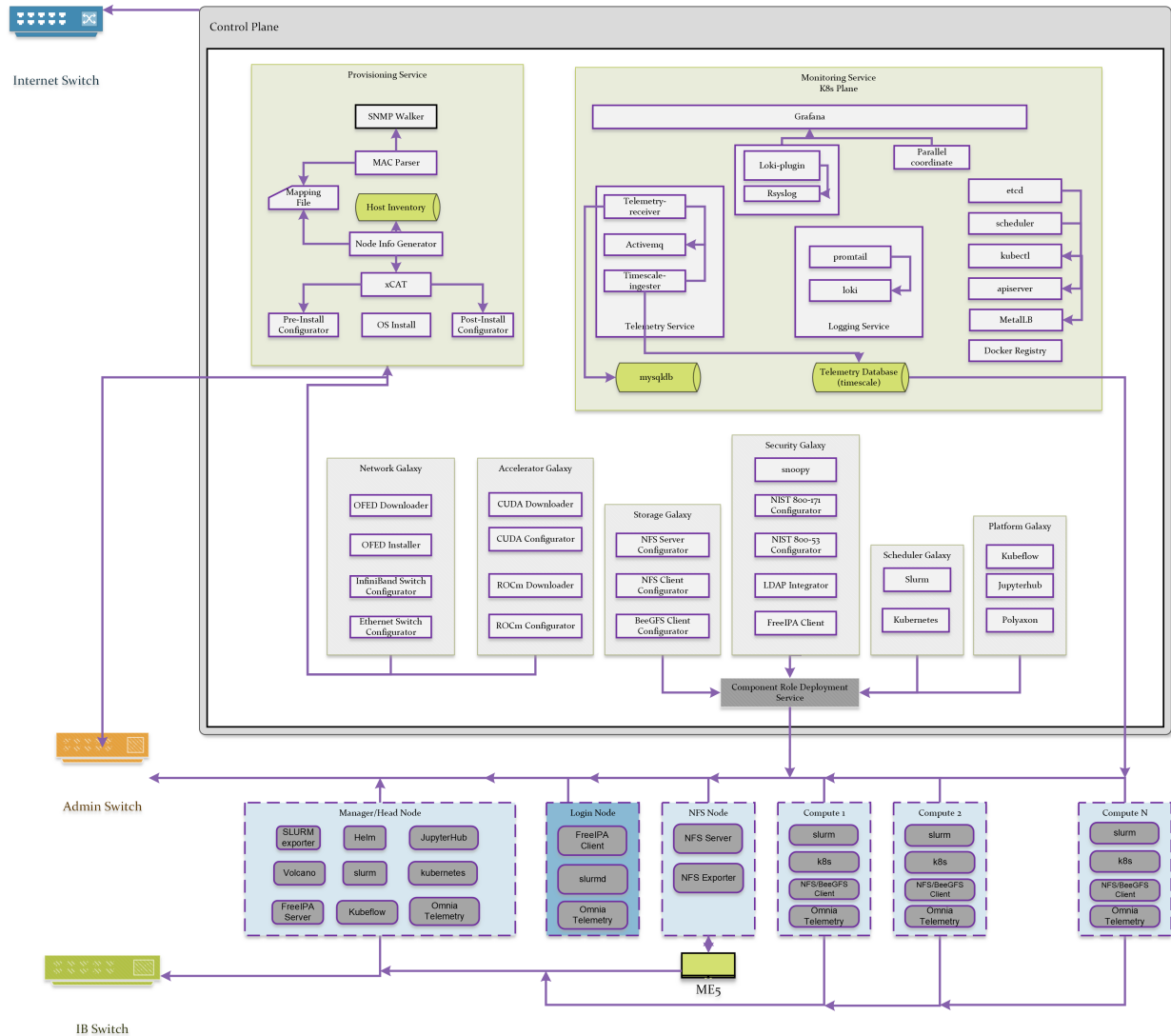
If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

¹ This metric is collected from the kube_control_plane if a login node is absent.

SECURITY CONFIGURATION GUIDE

6.1 Preface

The security configuration guide of Omnia provides Dell customers an overview and understanding of the security features supported by Omnia. As part of an effort to improve its product lines, Dell periodically releases revisions of its software and hardware. The product release notes provide the most up-to-date information about product features. Contact your Dell technical support professional if a product does not function properly or does not function as described in this document. This document was accurate at publication time. To ensure that you are using the latest version of this document, go to [Omnia: Docs](#).



6.1.1 Legal disclaimers

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS-IS.” DELL MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. In no event shall Dell Technologies, its affiliates or suppliers, be liable for any damages whatsoever arising from or related to the information contained herein or actions that you decide to take based thereon, including any direct, indirect, incidental, consequential, loss of business profits or special damages, even if Dell Technologies, its affiliates or suppliers have been advised of the possibility of such damages. The Security Configuration Guide intends to be a reference. The guidance is provided based on a diverse set of installed systems and may not represent the actual risk/guidance to your local installation and individual environment. It is recommended that all users determine the applicability of this information to their individual environments and take appropriate actions. All aspects of this Security Configuration Guide are subject to change without notice and on a case-by-case basis. Your use of the information contained in this document or materials linked herein is at your own risk. Dell reserves the right to change or update this document in its sole discretion and without notice at any time.

6.1.2 Scope of the document

This document covers the security features supported by Omnia 1.4.

6.1.3 Document references

In addition to this guide, more information on Omnia can be found using the below links:

- [Omnia: Read Me](#)
- [Omnia: Quick Installation Guide](#)

6.1.4 Reporting security vulnerabilities

Dell takes reports of potential security vulnerabilities in our products very seriously. If you discover a security vulnerability, you are encouraged to report it to Dell immediately. For the latest instructions on how to report a security issue to Dell, see the [Dell Vulnerability Response Policy](#) on the Dell.com site.

Follow Dell Security on these sites:

- dell.com/security
- dell.com/support

To provide feedback on this solution, email us at support@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

6.2 Security Quick Reference

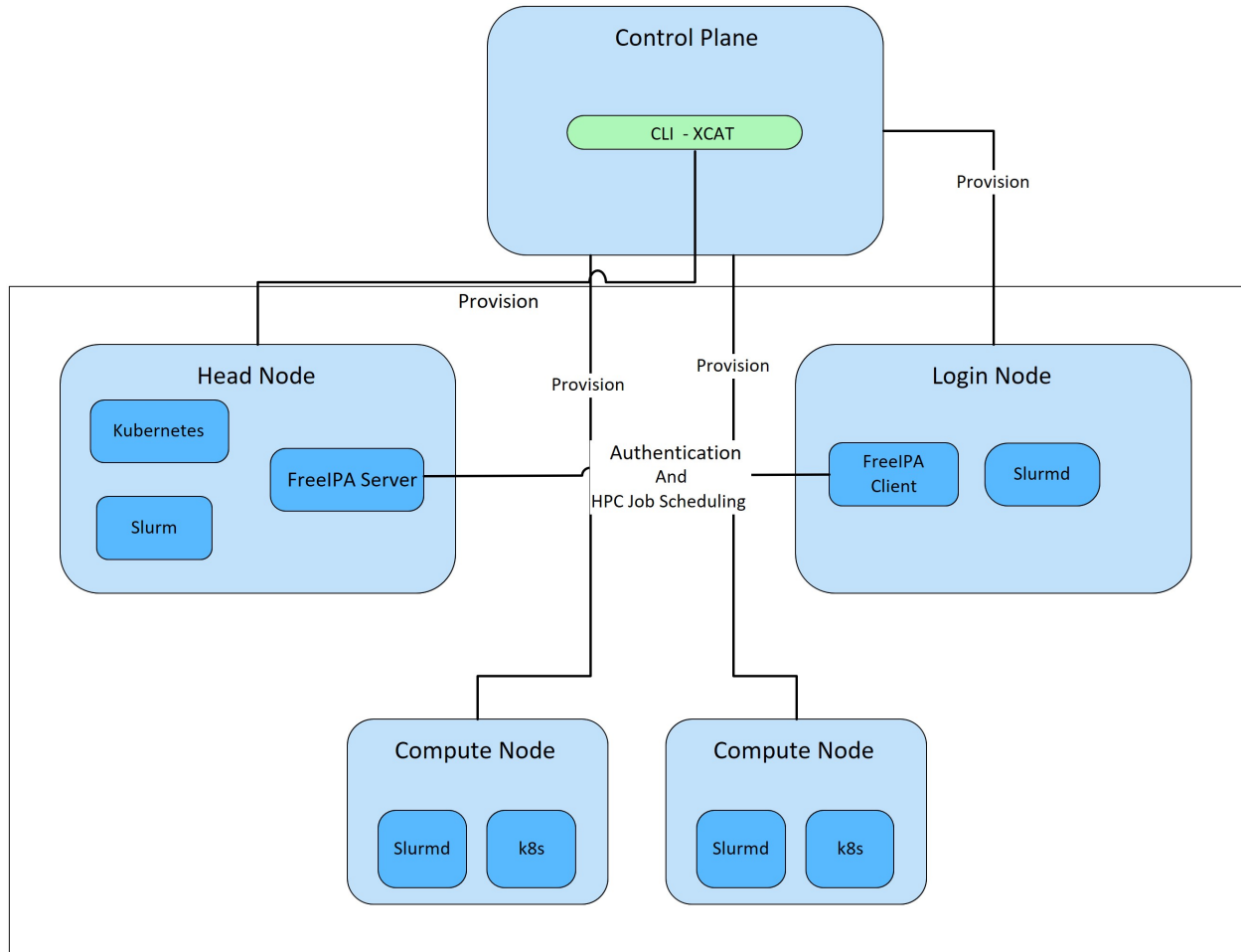
6.2.1 Security profiles

Omnia requires root privileges during installation because it provisions the operating system on bare metal servers.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

6.3 Product and Subsystem Security

6.3.1 Security controls map



Omnia performs bare metal configuration to enable AI/HPC workloads. It uses Ansible playbooks to perform installations and configurations. iDRAC is supported for provisioning bare metal servers. Omnia installs xCAT to enable provisioning of clusters via PXE in different ways:

- Mapping file **[optional]**: To dictate IP address/MAC mapping, a host mapping file can be provided.
- BMC discovery **[optional]**: To discover the cluster via BMC (iDRAC), IPMI must be enabled on remote servers. Discovery happens over IPMI. For security best practices when using this method, [click here!](#)
- Switch **[default]**: To discovery the cluster by routing communication through particular switch ports over SNMPv3, non-admin switch credentials must be provided.

Note: IPMI is not required on the control plane. However, compute nodes (iDRACs in the cluster/private network) require IPMI to be enabled for BMC discovery.

Omnia can be installed via CLI only. Slurm and Kubernetes are deployed and configured on the cluster. FreeIPA or LDAP is installed for providing authentication.

To perform these configurations and installations, a secure SSH channel is established between the management node and the following entities:

- kube_control_plane
- Compute Nodes
- Login Node

6.3.2 Authentication

Omnia does not have its own authentication mechanism because bare metal installations and configurations take place using root privileges. Post the execution of Omnia, third-party tools are responsible for authentication to the respective tool.

6.3.3 Cluster authentication tool

In order to enable authentication to the cluster, Omnia installs FreeIPA: an open source tool providing integrated identity and authentication for Linux/UNIX networked environments. As part of the HPC cluster, the login node is responsible for configuring users and managing a limited number of administrative tasks. Access to the manager/head node is restricted to administrators with the root password. For authentication on the manager and compute nodes exclusively, LDAP can also be installed by Omnia on the client.

Note: Omnia does not configure LDAP users or groups.

6.3.4 Authentication types and setup

Key-Based authentication

Use of SSH authorized_keys

A password-less channel is created between the management station and compute nodes using SSH authorized keys. This is explained in the *Security Controls Map*.

6.3.5 Login security settings

The following credentials have to be entered to enable different tools on the management station:

1. iDRAC (Username/ Password)
2. Ethernet Switch (Username/ Password)
3. Infiniband Switch (Username/ Password)
4. PowerVault ME4/ME5 (Username/ Password)
5. Provisioning OS (Password)
6. SNMPv3 PXE switch (Non-admin username/ password)

Similarly, passwords for the following tools have to be provided in `input/omnia_config.yml` and `input/provision_config_credentials.yml` to configure the cluster:

1. maria_db (Password)

2. DockerHub (Username/ Password)

For setting up authentication on the cluster, the following credentials have to be provided in `input/security_config.yml`:

1. FreeIPA (directory_manager_password, ipa_admin_password)
2. LDAP (ldap_bind_username, ldap_bind_password)

Once Omnia is invoked, these files are validated and encrypted using Ansible Vault. They are hidden from external visibility and access.

6.4 Authentication to external systems

Third party software installed by Omnia are responsible for supporting and maintaining manufactured-unique or installation-unique secrets.

6.4.1 Configuring remote connections

When setting up BeeGFS client services on the cluster, a connection authentication file is used to maintain the security of the communications between server and client.

1. Generate the connection authentication file (connAuth) and use it to set up BeeGFS meta, server and storage services.
2. Copy the connAuth file to the control plane and note the filepath.
3. Populate the value of `beegfs_secret_storage_filepath` in `input/storage_config.yml` with the filepath from the previous step.

Omnia will configure the BeeGFS clients on the cluster using the provided file. BeeGFS is responsible for maintaining and securing connAuthFile. For more information, [click here](#).

6.5 Network security

Omnia configures the firewall as required by the third-party tools to enhance security by restricting inbound and outbound traffic to the TCP and UDP ports.

6.5.1 Network exposure

Omnia uses port 22 for SSH connections, same as Ansible.

6.5.2 Firewall settings

Omnia configures the following ports for use by third-party tools installed by Omnia.

Kubernetes ports requirements

Port	Number	Layer 4	Protocol Purpose	Type of Node
6443	TCP	Kubernetes API	server	Manager
2379-2380	TCP	etcd server	client	API Manager
10251	TCP	Kube-scheduler	Manager	
10252	TCP	Kube-controller manager	Manager	
10250	TCP	Kubelet API	Compute	
30000-32767	TCP	Nodeport services	Compute	
5473	TCP	Calico services	Manager/Compute	
179	TCP	Calico services	Manager/Compute	
4789	UDP	Calico services	Manager/Compute	
8285	UDP	Flannel services	Manager/Compute	
8472	UDP	Flannel services	Manager/Compute	

Slurm port requirements

Port	Number	Layer 4	Protocol	Node
6817	TCP/UDP	Slurmctld Port	Manager	
6818	TCP/UDP	Slurmd Port	Compute	
6819	TCP/UDP	Slurmdbd Port	Manager	

BeeGFS port requirements

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

xCAT port requirements

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

continues on next page

Table 1 – continued from previous page

Port number	Protocol	Service Name
68	udp	bootpc
69	tcp	tftp-tcp
69	udp	tftp-udp
80	tcp	www-tcp
80	udp	www-udp
88	tcp	kerberos
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](#).

FreeIPA port requirements

Port Number	Layer 4	Purpose	Node
80	TCP	HTTP/HTTPS	Manager/ Login_Node
443	TCP	HTTP/HTTPS	Manager/ Login_Node
389	TCP	LDAP/LDAPS	Manager/ Login_Node
636	TCP	LDAP/LDAPS	Manager/ Login_Node
88	TCP/UDP	Kerberos	Manager/ Login_Node
464	TCP/UDP	Kerberos	Manager/ Login_Node
53	TCP/UDP	DNS	Manager/ Login_Node
7389	TCP	Dogtag's LDAP server	Manager/ Login_Node
123	UDP	NTP	Manager/ Login_Node

Note: To avoid security vulnerabilities, protocols can be restricted on the network using the parameters

`restrict_program_support` and `restrict_softwares` in `input/login_node_security_config.yml`. However, certain protocols are essential to Omnia's functioning and cannot be disabled. These protocols are: `ftp`, `smbd`, `nmbd`, `automount`, `portmap`.

6.5.3 Data security

Omnia does not store data. The passwords Omnia accepts as input to configure the third party tools are validated and then encrypted using Ansible Vault. Run `yum update --security` routinely on the control plane for the latest security updates.

For more information on the passwords used by Omnia, see *Login Security Settings*.

6.5.4 Auditing and logging

Omnia creates a log file at `/var/log/omnia` on the management station. The events during the installation of Omnia are captured as logs. For different roles called by Omnia, separate log files are created as listed below:

- `monitor.log`
- `network.log`
- `provision.log`
- `scheduler.log`
- `security.log`
- `storage.log`
- `utils.log`

Additionally, an aggregate of the events taking place during storage, scheduler and network role installation called `omnia.log` is created in `/var/log`.

There are separate logs generated by the third party tools installed by Omnia.

6.5.5 Logs

A sample of the `omnia.log` is provided below:

```
2021-02-15 15:17:36,877 p=2778 u=omnia n=ansible | [WARNING]: provided hosts
list is empty, only localhost is available. Note that the implicit localhost does not
match 'all'
2021-02-15 15:17:37,396 p=2778 u=omnia n=ansible | PLAY [Executing omnia roles]
*****
2021-02-15 15:17:37,454 p=2778 u=omnia n=ansible | TASK [Gathering Facts]
*****
*
2021-02-15 15:17:38,856 p=2778 u=omnia n=ansible | ok: [localhost]
2021-02-15 15:17:38,885 p=2778 u=omnia n=ansible | TASK [common : Mount Path]
*****
2021-02-15 15:17:38,969 p=2778 u=omnia n=ansible | ok: [localhost]
```

These logs are intended to enable debugging.

Note: The Omnia product recommends that product users apply masking rules on personal identifiable information (PII) in the logs before sending to external monitoring applications or sources.

6.5.6 Logging format

Every log message begins with a timestamp and also carries information on the invoking play and task.

The format is described in the following table.

Field	Format	Sample Value
Timestamp	yyyy-mm-dd h:m:s	2/15/2021 15:17
Process Id	p=xxxx	p=2778
User	u=xxxx	u=omnia
Name of the process executing	n=xxxx	n=ansible
The task being executed/ invoked	PLAY/TASK	PLAY [Executing omnia roles] TASK [Gathering Facts]
Error	fatal: [hostname]: Error Message	fatal: [localhost]: FAILED! => {"msg": "lookup_plugin.lines"}
Warning	[WARNING]: warning message	[WARNING]: provided hosts list is empty

6.5.7 Network vulnerability scanning

Omnia performs network security scans on all modules of the product. Omnia additionally performs Blackduck scans on the open source softwares, which are installed by Omnia at runtime. However, Omnia is not responsible for the third-party software installed using Omnia. Review all third party software before using Omnia to install it.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

6.6 Miscellaneous Configuration and Management Elements

6.6.1 Licensing

Omnia is licensed under the Apache License 2.0. A permissive license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

6.6.2 Protect authenticity

Every GitHub push requires a sign-off and a moderator is required to approve pull requests. All contributions have to be certified using the Developer Certificate of Origin (DCO):

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.

6.6.3 Ansible security

For the security guidelines of Ansible modules, go to [Developing Modules Best Practices: Module Security](#).

6.6.4 Ansible vault

Ansible vault enables encryption of variables and files to protect sensitive content such as passwords or keys rather than leaving it visible as plaintext in playbooks or roles. Please refer [Ansible Vault guidelines](#) for more information.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

SAMPLE FILES

7.1 inventory file

Caution: All the file contents mentioned below are case sensitive.

```
#Batch Scheduler: Slurm

[slurm_control_node]

10.5.1.101

[slurm_node]

10.5.1.103

10.5.1.104

[login]

10.5.1.105


#General Cluster Storage

[auth_server]

10.5.1.106


#AI Scheduler: Kubernetes

[kube_control_plane]

10.5.1.101

[etcd]

10.5.1.101
```

(continues on next page)

(continued from previous page)

```
[kube_node]
```

```
10.5.1.102
```

```
10.5.1.103
```

```
10.5.1.104
```

```
10.5.1.105
```

```
10.5.1.106
```

Note: The auth_server is common to both slurm and kubernetes clusters.

7.2 pxe_mapping_file.csv

```
SERVICE_TAG,HOSTNAME,ADMIN_MAC,ADMIN_IP,BMC_IP
XXXXXXX,n1,xx:yy:zz:aa:bb:cc,10.5.0.101,10.3.0.101
XXXXXXX,n2,aa:bb:cc:dd:ee:ff,10.5.0.102,10.3.0.102
```

7.3 switch_inventory

```
10.3.0.101
10.3.0.102
```

7.4 powervault_inventory

```
10.3.0.105
```

7.5 NFS Server inventory file

```
#General Cluster Storage
#NFS node
[nfs]
#node10
```

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

LIMITATIONS

- Omnia supports adding only 1000 nodes when discovered via BMC.
- 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.
- 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.
- Only one storage instance (Powervault) is currently supported in the HPC cluster.
- Omnia supports only basic telemetry configurations. Altering the time intervals for telemetry data collection is not supported.
- Slurm cluster metrics will only be fetched from clusters configured by Omnia.
- All iDRACs must have the same username and password.
- Currently, Omnia only supports the splitting of switch ports. Switch ports cannot be un-split using the [switch configuration script](#).
- The IP subnet 10.4.0.0 cannot be used for any networks on the Omnia cluster as it is reserved for Nerdctl.
- Installation of vLLM and racadam via Omnia is not supported on Ubuntu 20.04.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

BEST PRACTICES

- Ensure that PowerCap policy is disabled and the BIOS system profile is set to 'Performance' on the Control Plane.
- Always execute playbooks within the directory they reside in. That is, always change directories (`cd`) to the path where the playbook resides before running the playbook.
- Ensure that there is at least 50% (~50GB) free space on the Control Plane root partition before running Omnia. To maintain the free space required, place any ISO files used in the `/home` directory.
- 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.
- Ensure that the `/var` partition has adequate space to complete commands and store images.
- Run `yum update --security` routinely on the control plane for the latest security updates.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

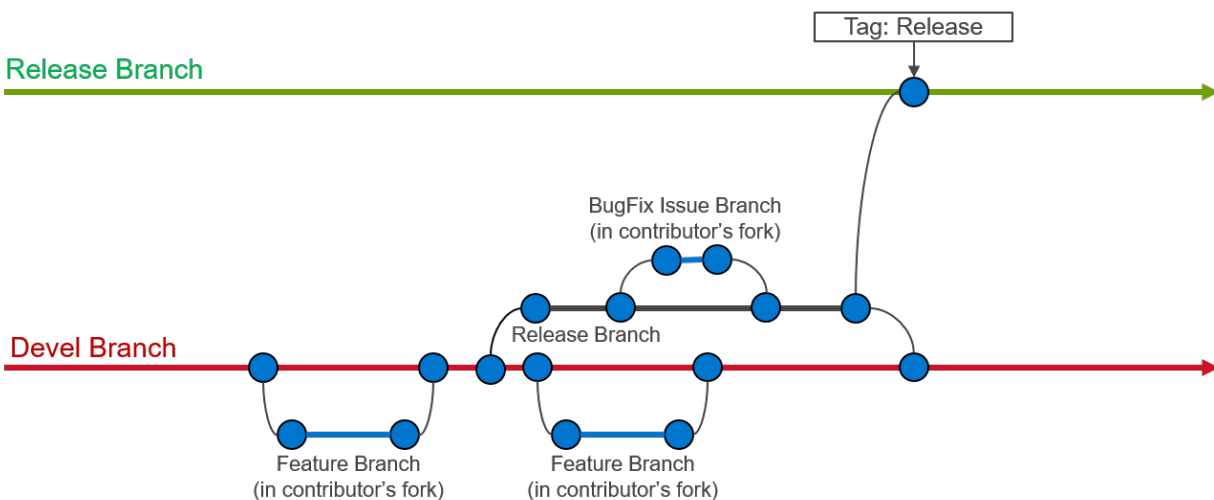
CONTRIBUTING TO OMNIA

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.

10.1 Creating A Pull Request

Contributions to Omnia are made through [Pull Requests \(PRs\)](#).



10.1.1 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?

10.1.2 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.

10.1.3 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`.

10.1.4 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](#)

Caution: 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 dell/omnia
git push --force origin <pr-branch-name> #origin is your fork of the repository (e.g.,
↪ <github_user_name>/omnia.git)
```

10.1.5 PR description

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

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

10.1.6 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.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.

If you have any feedback about Omnia documentation, please reach out at omnia.readme@dell.com.