



SUSE® Enterprise Storage on Huawei Taishan Implementation Guide



SUSE® Enterprise Storage on Huawei Taishan Implementation Guide

Kai Liu

Publication Date: May 2 2019

SUSE LLC

10 Canal Park Drive

Suite 200

Cambridge MA 02141

USA

<https://documentation.suse.com> 

Contents

1	Introduction	1
2	Target Audience	2
3	Hardware & Software	3
4	Business Problem and Business Value	4
4.1	SUSE Enterprise Storage	4
4.2	Huawei Taishan	4
4.3	Business Problem	4
4.4	Business Value	5
5	Requirements	6
5.1	Functional Requirements	6
6	Architectural Overview	7
6.1	Solution Architecture	7
6.2	Networking Architecture	9
6.3	Network/IP Address Scheme	10
7	Component Model	12
8	Deployment	13
8.1	Network Deployment Overview	13
8.2	Hardware Recommended Actions	14
8.3	Operating System Installation	14

8.4	SUSE Enterprise Storage Installation & Configuration	15
	Software Deployment Configuration (Deepsea and Salt)	15
	• Prepare All Nodes	16
	• Prepare OSD Disks on OSD Nodes	16
	• Install and Configure Deepsea on Admin Node	17
	• Deploy Using Deepsea	18
8.5	Post-deployment Quick Tests	21
8.6	Deployment Considerations	21
9	Conclusion	23
10	References and Resources	24
A	Bill of Materials	25
B	Network Switch Configuration	27
C	policy.cfg	28
D	Performance Data	29
D.1	Sequential Writes	30
D.2	Sequential Reads	31
D.3	Random Writes	32
D.4	Random Reads	33
D.5	Backup/Recovery Simulations	34
D.6	KVM Virtual Guest Simulation	34
D.7	Database Simulations	35

1 Introduction

The objective of this guide is to present a step-by-step guide on how to implement SUSE Enterprise Storage v5.5 on the Huawei Taishan platform. It is suggested that the document be read in its entirety, along with the supplemental appendix information before attempting the process.

The deployment presented in this guide aligns with architectural best practices and will support the implementation of all currently supported protocols as identified in the SUSE Enterprise Storage documentation.

Upon completion of the steps in this document, a working SUSE Enterprise Storage v5.5 cluster will be operational as described in the [SUSE Enterprise Storage Deployment and Administration Guide](https://www.suse.com/documentation/ses-5/book_storage_admin/data/book_storage_admin.html) (https://www.suse.com/documentation/ses-5/book_storage_admin/data/book_storage_admin.html)¹.

2 Target Audience

This reference architecture is targeted at administrators who deploy software defined storage solutions within their data centers and making the different storage services accessible to their own customer base. By following this document as well as those referenced herein, the administrator should have a full view of the SUSE Enterprise Storage architecture, deployment and administrative tasks, with a specific set of recommendations for deployment of the hardware and networking platform.

3 Hardware & Software

The recommended architecture for SUSE Enterprise Storage on Huawei Taishan leverages two models of Huawei servers. The role and functionality of each type of system within the SUSE Enterprise Storage environment will be explained in more detail in the architectural overview section.

STORAGE NODES:

- Huawei 5280 (<https://www.suse.com/nbswebapp/yesBulletin.jsp?bulletinNumber=147070>) ↗

ADMIN, MONITOR, AND PROTOCOL GATEWAYS:

- Huawei 2280 (<https://www.suse.com/nbswebapp/yesBulletin.jsp?bulletinNumber=146997>) ↗

SWITCHES:

- Huawei CE6851-48S6Q-HI 10Gb

SOFTWARE:

- SUSE Enterprise Storage v5.5
- SUSE Linux Enterprise Server 12 SP3



Tip

Please note that limited use subscriptions are provided with SUSE Enterprise Storage as part of the subscription entitlement

4 Business Problem and Business Value

4.1 SUSE Enterprise Storage

SUSE Enterprise Storage delivers a highly scalable, resilient, self-healing storage system designed for large scale environments ranging from hundreds of Terabytes to Petabytes. This software defined storage product can reduce IT costs by leveraging industry standard servers to present unified storage servicing block, file, and object protocols. Having storage that can meet the current needs and requirements of the data center while supporting topologies and protocols demanded by new web-scale applications, enables administrators to support the ever-increasing storage requirements of the enterprise with ease.

4.2 Huawei Taishan

Huawei Taishan servers provide a cost effective and scalable platform for the deployment of SUSE Enterprise Storage. These platforms unlocks the full potential of the Kunpeng CPU, raising the bar of SPECint benchmark by 25%, with up to 128 cores, 32 DDR4 DIMMs, PCIe 4.0, CCIX, and 100 GE LOM.

Taishan servers are powered by 64-bit ARMv8 Hi1616 processors, each with 32 cores of 2.4 GHz frequency. They support diverse interfaces such as PCIe 3.0, 10GE, and SAS/SATA, and integrate high performance with low power consumption.

Featuring models tailored for computing, storage, or balanced needs, Taishan is perfect for demanding workloads such as big data analytics, database acceleration, high-performance computing, cloud services, and native mobile applications. Taishan servers empower data centers with the ultimate efficiency.

4.3 Business Problem

Customers of all sizes face a major storage challenge: While the overall cost per Terabyte of physical storage has gone down over the years, a data growth explosion took place driven by the need to access and leverage new data sources (ex: external sources such as social media) and the ability to "manage" new data types (ex: unstructured or object data). These ever increasing "data lakes" need different access methods: File, block, or object.

Addressing these challenges with legacy storage solutions would require either a number of specialized products (usually driven by access method) with traditional protection schemes (ex: RAID). These solutions struggle when scaling from Terabytes to Petabytes at reasonable cost and performance levels.

4.4 Business Value

This software defined storage solution enables transformation of the enterprise infrastructure by providing a unified platform where structured and unstructured data can co-exist and be accessed as file, block, or object depending on the application requirements. The combination of open-source software (Ceph) and industry standard servers reduce cost while providing the on-ramp to unlimited scalability needed to keep up with future demands.

5 Requirements


Enterprise storage systems require reliability, manageability, and serviceability. The legacy storage players have established a high threshold for each of these areas and now expect the software defined storage solutions to offer the same. Focusing on these areas helps SUSE make open source technology enterprise consumable. When combined with highly reliable and manageable hardware from Huawei, the result is a solution that meets the customer's expectation.

5.1 Functional Requirements

A SUSE Enterprise Storage solution is:

- Simple to setup and deploy, within the documented guidelines of system hardware, networking and environmental prerequisites.
- Adaptable to the physical and logical constraints needed by the business, both initially and as needed over time for performance, security, and scalability concerns.
- Resilient to changes in physical infrastructure components, caused by failure or required maintenance.
- Capable of providing optimized object and block services to client access nodes, either directly or through gateway services.

6 Architectural Overview

This architecture overview section complements the SUSE Enterprise Storage Technical Overview (https://www.suse.com/docrep/documents/1mdg7eq2kz/suse_enterprise_storage_technical_overview_wp.pdf)  document available online which presents the concepts behind software defined storage and Ceph as well as a quick start guide (non-platform specific).

6.1 Solution Architecture

SUSE Enterprise Storage provides unified block, file, and object access based on Ceph. Ceph is a distributed storage solution designed for scalability, reliability and performance. A critical component of Ceph is the RADOS object storage. RADOS enables a number of storage nodes to function together to store and retrieve data from the cluster using object storage techniques. The result is a storage solution that is abstracted from the hardware.

Ceph supports both native and traditional client access. The native clients are aware of the storage topology and communicate directly with the storage daemons over the public network, resulting in horizontally scaling performance. Non-native protocols, such as iSCSI, S3, and NFS require the use of gateways. While these gateways may be thought of as a limiting factor, the iSCSI and S3 gateways can scale horizontally using load balancing techniques.

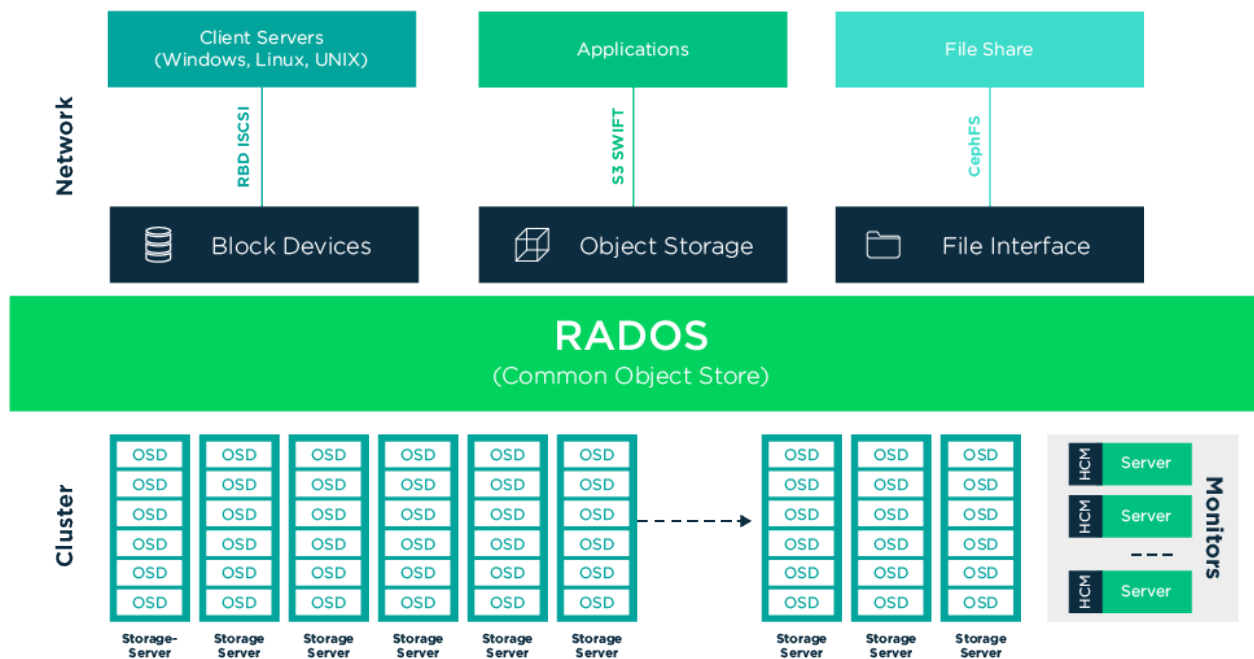


FIGURE 6.1: SES ARCHITECTURE

In addition to the required network infrastructure, the minimum SUSE Enterprise Storage cluster comprises of a minimum of one administration server (physical or virtual), four object storage device nodes (OSDs), and three monitor nodes (MONs).

SPECIFIC TO THIS IMPLEMENTATION:

- One system is deployed as the administrative host server. The administration host is the Salt master and hosts the SUSE Enterprise Storage Administration Interface, openATTIC, which is the central management system which supports the cluster.
- Three systems are deployed as monitor (MONs) nodes. Monitor nodes maintain information about the cluster health state, a map of the other monitor nodes and a CRUSH map. They also keep history of changes performed to the cluster.
- It is strongly recommended to deploy monitors and other services on dedicated nodes. However due to shortage of equipment the monitors were deployed on the OSD nodes in this specific reference setup.
- Additional servers may be deployed as iSCSI gateway nodes. iSCSI is a storage area network (SAN) protocol that allows clients (called initiators) to send SCSI command to SCSI storage devices (targets) on remote servers. This protocol is utilized for block-based connectivity to environments such as Microsoft Windows, VMware, and traditional UNIX. These systems may be scaled horizontally through client usage of multi-path technology.

- The RADOS gateway provides S3 and Swift based access methods to the cluster. These nodes are generally situated behind a load balancer infrastructure to provide redundancy and scalability. It is important to note that the load generated by the RADOS gateway can consume a significant amount of compute and memory resources making the minimum recommended configuration contain 6-8 CPU cores and 32GB of RAM.
- SUSE Enterprise Storage requires a minimum of four systems as storage nodes. The storage nodes contain individual storage devices that are each assigned an Object Storage Daemon (OSD). The OSD daemon assigned to the device stores data and manages the data replication and rebalancing processes. OSD daemons also communicate with the monitor (MON) nodes and provide them with the state of the other OSD daemons.

6.2 Networking Architecture

A software-defined solution is only as reliable as its slowest and least redundant component. This makes it important to design and implement a robust, high performance storage network infrastructure. From a network perspective for Ceph, this translates into:

- Separation of cluster (backend) and client-facing (public) network traffic. This isolates Ceph OSD daemon replication activities from Ceph clients. This may be achieved through separate physical networks or through use of VLANs.
- Redundancy and capacity in the form of bonded network interfaces connected to switches.

Figure 6.2, “Ceph Network Architecture” shows the logical layout of the traditional Ceph cluster implementation.

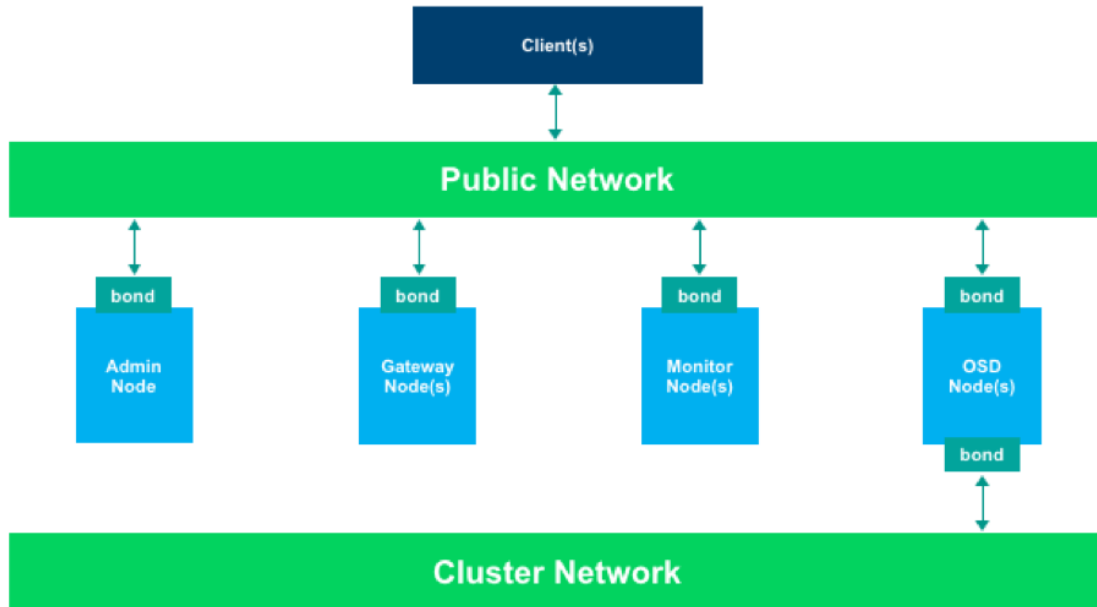


FIGURE 6.2: CEPH NETWORK ARCHITECTURE

6.3 Network/IP Address Scheme

Specific to this implementation, the following naming and addressing scheme were utilized.

TABLE 6.1: NODE ROLES AND NETWORK ADDRESSES

Role	Hostname	Public Network	Cluster Network
Admin	sesadmin.example.com	10.1.1.3	N/A
Monitor	osd1.example.com	10.1.1.4	N/A
Monitor	osd2.example.com	10.1.1.5	N/A
Monitor	osd3.example.com	10.1.1.6	N/A
OSD Node	osd1.example.com	10.1.1.4	10.2.1.4
OSD Node	osd2.example.com	10.1.1.5	10.2.1.5
OSD Node	osd3.example.com	10.1.1.6	10.2.1.6

Role	Hostname	Public Network	Cluster Network
OSD Node	osd4.example.com	10.1.1.7	10.2.1.7

7 Component Model

The preceding sections provided information on the both the overall Huawei hardware as well as an introduction to the Ceph software architecture. In this section, the focus is on the SUSE components: SUSE Linux Enterprise Server (SLES), SUSE Enterprise Storage (SES), and the Subscription Management Tool (SMT).

COMPONENT OVERVIEW (SUSE)

- SUSE Linux Enterprise Server - A world class secure, open source server operating system, equally adept at powering physical, virtual, or cloud-based mission-critical workloads. Service Pack 3 further raises the bar in helping organizations to accelerate innovation, enhance system reliability, meet tough security requirements and adapt to new technologies.
- Subscription Management Tool for SLES - Allows enterprise customers to optimize the management of SUSE Linux Enterprise (and product such as SUSE Enterprise Storage) software updates and subscription entitlements. It establishes a proxy system for SUSE Customer Center with repository and registration targets.
- SUSE Enterprise Storage - Provided as an product on top of SUSE Linux Enterprise Server, this intelligent software-defined storage solution, powered by Ceph technology with enterprise engineering and support from SUSE enables customers to transform enterprise infrastructure to reduce costs while providing unlimited scalability.

8 Deployment

This deployment section should be seen as a supplement online documentation (<https://www.suse.com/documentation/>)⁷. Specifically, the SUSE Enterprise Storage 5 Deployment Guide (https://www.suse.com/documentation/suse-enterprise-storage-5/book_storage_deployment/data/book_storage_deployment.html)⁷ as well as SUSE Linux Enterprise Server Administration Guide (https://www.suse.com/documentation/sles-12/book_sle_admin/data/book_sle_admin.html)⁷. It is assumed that a Subscription Management Tool server exists within the environment. If not, please follow the information in Subscription Management Tool (SMT) for SLES (https://www.suse.com/documentation/sles-12/book_smt/data/book_smt.html)⁷ to make one available. The emphasis is on specific design and configuration choices.

In this document we use `example.com` as the domain name for the nodes, replace it with your real domain name in your own installation.

8.1 Network Deployment Overview

The following considerations for the network configuration should be attended to:

- Ensure that all network switches are updated with consistent firmware versions.
- Configure 802.3ad for system port bonding between the switches, plus enable jumbo frames.
- Specific configuration for this deployment can be found in *Appendix B, Network Switch Configuration*
- Network IP addressing and IP ranges need proper planning. In optimal environments, a dedicated storage subnet should be used for all SUSE Enterprise Storage nodes on the primary network, with a separate, dedicated subnet for the cluster network. Depending on the size of the installation, ranges larger than /24 may be required. When planning the network, current as well as future growth should be taken into consideration.
- Setup DNS A records for all nodes. Decide on subnets and VLANs and configure the switch ports accordingly.

- Always use the same type (short or full) of hostname every where, for Salt master/minions etc.
- Ensure that you have access to a valid, reliable NTP service, as this is a critical requirement for all nodes. If not, it is recommended to use the admin node.

8.2 Hardware Recommended Actions

The following considerations for the hardware platforms should be attended to:

- Configure the boot media as RAID-1.
- Configure all data and journal devices as individual RAID-0 if RAID controllers with hardware write cache are configured.

8.3 Operating System Installation

There are several key tasks to ensure are performed correctly during the operating system installation. During the SUSE Linux Enterprise installation, be sure and register the system with an update server. Ideally, this is a local SMT server which will reduce the time required for updates to be downloaded and applied to all nodes. By updating the nodes during installation, the system will deploy with the most up-to-date packages available, helping to ensure the best experience possible.

To speed installation, on the System Role screen, it is suggested to select KVM Virtualization Host. When the Installation Settings screen is reached, select **Software** and then un-check KVM Host Server. The resulting installation is a text mode server that is an appropriate base OS for SUSE Enterprise Server.

The next item is to ensure that the operating system is installed on the correct device. Especially on OSD nodes, the system may not choose the right drive by default. The proper way to ensure the right device is being used is to select **Create Partition Setup** on the Suggested Partitioning screen. This will then display a list of devices, allowing selection of the correct boot device. Next select **Edit Proposal Settings** and unselect the **Propose Separate Home Partition** checkbox.

Do ensure that NTP is configured to point to a valid, physical NTP server. This is critical for SUSE Enterprise Storage to function properly, and failure to do so can result in an unhealthy or non-functional cluster. And keep in mind that the NTP service is not designed to be run on an virtualized environment, so make sure the NTP server been used is an physical machine or it may cause strange clock drifting problem.

8.4 SUSE Enterprise Storage Installation & Configuration

8.4.1 Software Deployment Configuration (Deepsea and Salt)

Salt, along with DeepSea, is a stack of components that help deploy and manage server infrastructure. It is very scalable, fast, and relatively easy to get running.

There are three key Salt imperatives that need to be followed:

- The Salt Master is the host that controls the entire cluster deployment. Ceph itself should NOT be running on the master as all resources should be dedicated to Salt master services. In our scenario, we used the Admin host as the Salt master.
- Salt minions are nodes controlled by Salt master. OSD, monitor, and gateway nodes are all Salt minions in this installation.
- Salt minions need to correctly resolve the Salt master's host name over the network. This can be achieved through configuring unique host names per interface (eg `osd1-cluster.example.com` and `osd1-public.example.com`) in DNS and/or local `/etc/hosts` files.

Deepsea consists of series of Salt files to automate the deployment and management of a Ceph cluster. It consolidates the administrator's decision making in a single location around cluster assignment, role assignment and profile assignment. Deepsea collects each set of tasks into a goal or stage.

The following steps, performed in order, will be used for this reference implementation:

8.4.2 Prepare All Nodes

1. Install DeepSea on the Salt master which is the Admin node:

```
zypper in salt-master
```

2. Start the salt-master service and enable:

```
systemctl enable --now salt-master.service
```

3. Install the salt-minion on all cluster nodes (including the Admin):

```
zypper in salt-minion
```

4. Configure all minions to connect to the Salt master:

Modify the entry for master in the */etc/salt/minion*

```
master: sesadmin.example.com
```

5. Restart the salt-minion service and enable it:

```
systemctl restart salt-minion.service  
systemctl enable salt-minion.service
```

6. List Salt fingerprints on all the minions:

```
salt-call --local key.finger
```

7. List all minion fingerprints on the Salt master, verify them against the fingerprints on each minions to make sure they all match. If they do, accept all Salt keys on the Salt master:

```
salt-key -F  
salt-key --list-all  
salt-key --accept-all
```

8. Verify if Salt works properly by "ping" each minions. They should all return True on success:

```
salt '*' test.ping
```

8.4.3 Prepare OSD Disks on OSD Nodes

If the OSD nodes were used in a prior installation, zap ALL the OSD disks first.

Important

This must be done on all the OSD disks that were used in a prior installation, or else the deployment will fail when activating OSDs.

Warning

Below commands should not be copied and executed on your installation blindly. The device names used below are just examples, you need to change them to match only the OSD disks in your own installation. Failed to use the correct device name may erase your OS disk or other disks that may hold valuable data.

1. Wipe the beginning of each partition:

```
for partition in /dev/sdX[0-9]*
do
    dd if=/dev/zero of=$partition bs=4096 count=1 oflag=direct
done
```

2. Wipe the partition table:

```
sgdisk -Z --clear -g /dev/sdX
```

3. Wipe the backup partition tables:

```
size=`blockdev --getsz /dev/sdX`
position=$((size/4096 - 33))
dd if=/dev/zero of=/dev/sdX bs=4M count=33 seek=$position oflag=direct
```

8.4.4 Install and Configure Deepsea on Admin Node

1. Install deepsea package on Admin node:

```
# zypper in deepsea
```

2. Check `/srv/pillar/ceph/master_minion.sls` for correctness.

3. Check `/srv/pillar/ceph/deepsea_minions.sls` file, make sure the `deepsea_minions` option targets the correct nodes. In the usual case, it can simply be put like below to match all Salt minions in the cluster:

```
deepsea_minions: '*'
```

4. Create `/srv/pillar/ceph/stack/ceph/cluster.yml` with below options:

```
cluster_network: <net/mask of cluster network>
public_network: <net/mask of public network>
time_server: <Address of NTP server, if this line is omitted admin node will be
used>
```

8.4.5 Deploy Using Deepsea

At this point Deepsea commands can be run to deploy the cluster.



Tip

Each command can be run either as:

```
salt-run state.orch ceph.stage.<stage name>
```

Or:

```
deepsea stage run ceph.stage.<stage name>
```

The latter form is preferred as it outputs real time progress.

8.4.5.1 Stage 0: Prepare

During this stage, all required updates are applied and your system may be rebooted.

```
deepsea stage run ceph.stage.0
```



Important

If the Salt master reboots during Stage 0, you need to run Stage 0 again after it boots up.

Optionally, create the `/var/lib/ceph` btrfs subvolume:

```
salt-run state.orch ceph.migrate.subvolume
```

8.4.5.2 Stage 1: Discovery

During this stage, all hardware in your cluster is detected and necessary information are collected for the Ceph configuration.

```
deepsea stage run ceph.stage.1
```



Note

Configure cluster and public network in `/srv/pillar/ceph/stack/ceph/cluster.yml` if not yet done as described in [Create cluster.yml](#).

Now a `/srv/pillar/ceph/proposals/policy.cfg` file needs to be created to instruct Deepsea on the location and configuration files to use for the different components that make up the Ceph cluster (Salt master, admin, monitor, OSD and other roles).

To do so, copy the example file to the right location then edit it to match your installation:

```
cp /usr/share/doc/packages/deepsea/examples/policy.cfg-rolebased /srv/pillar/ceph/proposals/policy.cfg
```



Tip

See [Appendix C, policy.cfg](#) for the one used when installing the cluster described in this document.

A proposal for the storage layout needs to be generated at this time. For the hardware configuration used for this work, the following command was utilized:

```
salt-run proposal.populate
```

The proposal generator will automatically use hard disks for OSD storage and NVMe SSDs for BlueStore WAL and DB storage.



Tip

On your own deployment you may need to play with the proposal generator with different arguments for several times to get what you really want.

To print the help text about the various arguments proposal command accepts:

```
salt-run proposal.help
```

To show the generated proposal on screen according to the arguments passed:

```
salt-run proposal.peek <arguments>
```

To write the proposal to the */srv/pillar/ceph/proposals* subdirectory:

```
salt-run proposal.populate <arguments> name=myprofile
```

Pass the argument name=myprofile to the command to name the storage profile. This will result in a profile-myprofile subdirectory been created to store the new proposal files.

8.4.5.3 Stage 2: Configure

During this stage necessary configuration data are prepared in particular format.

```
deepsea stage run ceph.stage.2
```



Tip

Use below command to check the attributes of each node:

```
salt '*' pillar.items
```

8.4.5.4 Stage 3: Deploy

A basic Ceph cluster with mandatory Ceph services is created.

```
deepsea stage run ceph.stage.3
```




Note

It may take quite some time for above command to finish if your cluster is large, or your Internet bandwidth is limited while you didn't register the nodes to local SMT server.

After the above command is finished successfully, check whether the cluster is up by running:

```
ceph -s
```

8.4.5.5 Stage 4: Services

Additional features of Ceph like iSCSI, Object Gateway and CephFS can be installed in this stage. Each is optional and up to your situation.

```
deepsea stage run ceph.stage.4
```

8.5 Post-deployment Quick Tests

The steps below can be used (regardless of the deployment method) to validate the overall cluster health:

```
ceph status
ceph osd pool create test 1024
rados bench -p test 300 write --no-cleanup
rados bench -p test 300 seq
```

Once the tests are complete, you can remove the test pool via:

```
ceph tell mon.* injectargs --mon-allow-pool-delete=true
ceph osd pool delete test test --yes-i-really-really-mean-it
ceph tell mon.* injectargs --mon-allow-pool-delete=false
```

8.6 Deployment Considerations

Some final considerations before deploying your own version of a SUSE Enterprise Storage cluster, based on Ceph. As previously stated, please refer to the Administration and Deployment Guide.

- With the default replication setting of 3, remember that the client-facing network will have about half or less of the traffic of the backend network. This is especially true when component failures occur or rebalancing happens on the OSD nodes. For this reason, it is important not to under provision this critical cluster and service resource.
- It is important to maintain the minimum number of monitor nodes at three. As the cluster increases in size, it is best to increment in pairs, keeping the total number of Mon nodes as an odd number. However, only very large or very distributed clusters would likely need beyond the 3 MON nodes cited in this reference implementation. For performance reasons, it is recommended to use distinct nodes for the MON roles, so that the OSD nodes can be scaled as capacity requirements dictate.
- Although in this specific implementation monitors were deployed on the OSD nodes due to shortage of equipment, ideally monitors should be deployed on dedicated nodes.
- As described in this implementation guide and the SUSE Enterprise Storage documentation, a minimum of four OSD nodes is recommended, with the default replication setting of 3. This will ensure cluster operation, even with the loss of a complete OSD node. Generally speaking, performance of the overall cluster increases as more properly configured OSD nodes are added.

9 Conclusion

The Huawei Taishan series represents a strong capacity-oriented platform. When combined with the access flexibility and reliability of SUSE Enterprise Storage and the industry leading support from Huawei, any business can feel confident in the ability to address the exponential growth in storage they are currently faced with.

10 References and Resources

SUSE Enterprise Storage Technical Overview

https://www.suse.com/media/white-paper/suse_enterprise_storage_technical_overview_wp.pdf ↗

SUSE Enterprise Storage Tech Specs

<https://www.suse.com/products/suse-enterprise-storage/#tech-specs> ↗

SUSE Enterprise Storage - Deployment Guide

https://www.suse.com/documentation/suse-enterprise-storage-5/singlehtml/book_storage_deployment/book_storage_deployment.html ↗

SUSE Enterprise Storage - Administration Guide

https://www.suse.com/documentation/suse-enterprise-storage-5/singlehtml/book_storage_admin/book_storage_admin.html ↗

SUSE Linux Enterprise Server 12 SP3 - Deployment Guide

https://www.suse.com/documentation/sles-12/singlehtml/book_sle_deployment/book_sle_deployment.html ↗

SUSE Linux Enterprise Server 12 SP3 - Administration Guide

https://www.suse.com/documentation/sles-12/singlehtml/book_sle_admin/book_sle_admin.html ↗

SUSE Linux Enterprise Server 12 SP3 - Storage Administration Guide

https://www.suse.com/documentation/sles-12/singlehtml/stor_admin/stor_admin.html ↗

Subscription Management Tool for SLES 12 SP3

https://www.suse.com/documentation/sles-12/singlehtml/book_smt/book_smt.html ↗

A Bill of Materials

Role	Qty	Component	Notes
Admin Node	1	A VM on a Huawei x86 machine	The node consists of: <ul style="list-style-type: none">• 8x vCPU cores• 64GB RAM• 100GB virtual disk for OS• 2x virtual NIC port to public and cluster network
OSD nodes. MON, MGR shared the OSD hosts	4	Huawei Taishan 5280	Each node consists of: <ul style="list-style-type: none">• 2x HiSilicon 1616• 256GB• 1x HiSilicon 1616 Integrated SAS Controller• 2x 300GB SAS HDD for OS• 34x 4TB 7.2k SATA HDD for OSD• 2x Huawei 1TB NVMe SSD for db and journal• 1x HiSilicon 1616 Embedded Network Controller - 1GbE Quad Port• 1x Dual Port Intel 82599 10Gb Ethernet adapter
Software	1	SUSE Enterprise Storage v5.5 Subscription Base configuration	Allows for 4 storage nodes and 6 infrastructure nodes. Expansion subscriptions need to be purchased if more nodes need to be added to the cluster.

Role	Qty	Component	Notes
Switches	2	Huawei CE6851-48S6Q-HI	32 Ports of 100GbE

B Network Switch Configuration

Below are the commands used to configure the Huawei CE6851-48S6Q-HI switch for 802.3ad link aggregation.

```
port link-type trunk
mode lacp-static
load-balance src-dst-ip
local-preference disable
```

In the OS bonding configuration, make sure below options are set:

```
BONDING_MODULE_OPTS='mode=802.3ad miimon=100 lacp_rate=fast xmit_hash_policy=layer3+4'
```

C policy.cfg

```
## Cluster Assignment
cluster-ceph/cluster/*.sls

## Roles
# ADMIN
role-master/cluster/sesadmin*.sls
role-admin/cluster/sesadmin*.sls

# MON
role-mon/cluster/osd[1-3].sls

# MGR (mgrs are usually colocated with mons)
role-mgr/cluster/osd[1-3]*.sls

# MDS
role-mds/cluster/osd1*.sls


# openATTIC
role-openattic/cluster/sesadmin*.sls

# COMMON
config/stack/default/global.yml
config/stack/default/ceph/cluster.yml

## Profiles
profile-default/cluster/*.sls
profile-default/stack/default/ceph/minions/*.yml
```


D Performance Data

Comprehensive performance baselines are run as part of a reference build activity. This activity yields a vast amount of information that may be used to approximate the size and performance of the solution. The only tuning applied is documented in the implementation portion of this document.

The tests are comprised of a number of Flexible I/O (fio) job files run against multiple worker nodes. The job files and testing scripts may be found for review at: <https://github.com/dmbyte/benchmaster> . This is a personal repository and no warranties are made in regard to the fitness and safety of the scripts found there.

The testing methodology involves two different types of long running tests. The types and duration of the tests have very specific purposes. There are both I/O simulation jobs and single metric jobs.

The length of the test run, in combination with the ramp-up time specified in the job file, is intended to allow the system to overrun caches. This is a worst-case scenario for a system and would indicate that it is running at or near capacity. Given that few applications can tolerate significant amounts of long tail latencies, the job files have specific latency targets assigned. These targets are intended to be in-line with expectations for the type of I/O operation being performed and set realistic expectations for the application environment.

The latency target, when combined with the latency window and latency window percentage set the minimum number of I/Os that must be within the latency target during the latency window time period. For most of the tests, the latency target is 20ms or less. The latency window is five seconds and the latency target is 99.99999%. The way that fio uses this is to ramp up the queue depth at each new latency window boundary until more than .00001% of all I/O's during a five second window are higher than 20ms. At that point, fio backs the queue depth down where the latency target is sustainable.

In the figures below, the x-axis labels indicate the block size in KiB on the top line and the data protection scheme on the bottom line. 3xRep is indicative of the Ceph standard 3 replica configuration for data protection while EC2 + 2 is Erasure Coded using the ISA plugin with $k = 2$ and $m = 2$. The Erasure Coding settings were selected to fit within the minimum cluster hardware size supported by SUSE.

These settings, along with block size, max queue depth, jobs per node, etc., are all visible in the job files found at the repository link above.

Load testing was provided by for additional Huawei x86 servers on the same 10GbE network

D.1 Sequential Writes

Sequential write I/O testing was performed across block sizes ranging from 4KiB to 4MiB.

These tests have latency targets associated. 4K is 10ms, 64K is 20ms, 1MiB is 100ms, and 4MiB is 300ms.

TABLE D.1: CEPHFS SEQUENTIAL WRITES

Data Protection	I/O Size	Write BW (MiB/s)	Write IOPS	Write Avg Latency (ms)
3xRep	4K	26	6896.3	9.2
EC2 + 2	4K	6	1759.6	36.5
3xRep	64K	99	1587.9	41.0
EC2 + 2	64K	79	1268.9	50.5
3xRep	1M	624	624.3	103.4
EC2 + 2	1M	454	454.8	139.7
3xRep	4M	977	244.3	262.2
EC2 + 2	4M	826	206.6	306.0

TABLE D.2: RBD SEQUENTIAL WRITES

Data Protection	I/O Size	Write BW (MiB/s)	Write IOPS	Write Avg Latency (ms)
3xRep	4K	34	8720.2	7.2
EC2 + 2	4K	5	1299.3	51.4
3xRep	64K	143	2297.6	27.8
EC2 + 2	64K	65	1049.3	60.4
3xRep	1M	228	228.0	69.9

Data Protection	I/O Size	Write BW (MiB/s)	Write IOPS	Write Avg Latency (ms)
EC2 + 2	1M	121	121.4	131.5
3xRep	4M	1076	269.1	233.0
EC2 + 2	4M	732	183.2	344.9

D.2 Sequential Reads

The sequential read tests were conducted across the same range of block sizes as the write testing. The latency targets are only present for 4k sequential reads, where it is set to 10ms.

TABLE D.3: CEPHFS SEQUENTIAL READS

Data Protection	I/O Size	Read BW (MiB/s)	Read IOPS	Read Avg Latency (ms)
3xRep	4K	212	54480.6	1.2
EC2 + 2	4K	31	8077.4	2.0
3xRep	64K	2369	37912.0	54.2
EC2 + 2	64K	2077	33234.7	42.5
3xRep	1M	1740	1739.4	1079.4
EC2 + 2	1M	1483	1482.3	1357.2
3xRep	4M	2475	617.9	2897.8
EC2 + 2	4M	3124	780.2	2560.1

TABLE D.4: RBD SEQUENTIAL READS

Data Protection	I/O Size	Read BW (MiB/s)	Read IOPS	Read Avg Latency (ms)
3xRep	4K	45	11548.5	1.4

Data Protection	I/O Size	Read BW (MiB/s)	Read IOPS	Read Avg Latency (ms)
EC2 + 2	4K	80	20667.4	1.5
3xRep	64K	1661	26582.4	19.3
EC2 + 2	64K	1894	30307.0	61.8
3xRep	1M	3595	3654.1	526.6
EC2 + 2	1M	611	611.5	801.9
3xRep	4M	4384	1095.1	1634.2
EC2 + 2	4M	7887	1970.8	2120.1

D.3 Random Writes

Random write tests were performed with the smaller I/O sizes of 4k and 64k. The 4k tests have a latency target of 10ms and the 64k tests have a latency target of 20ms.

TABLE D.5: CEPHFS RANDOM WRITES

Data Protection	I/O Size	Write BW (MiB/s)	Write IOPS	Write Avg Latency (ms)
3xRep	4	14	3635.6	17.6
EC2 + 2	4	4	1107.3	57.8
3xRep	64	104	1666.9	38.0
EC2 + 2	64	59	958.5	66.8

TABLE D.6: RBD RANDOM WRITES

Data Protection	I/O Size	Write BW (MiB/s)	Write IOPS	Write Avg Latency (ms)
3xRep	4	10	2606.4	24.7

Data Protection	I/O Size	Write BW (MiB/s)	Write IOPS	Write Avg Latency (ms)
EC2 + 2	4	3	952.7	67.0
3xRep	64	142	2286.6	27.8
EC2 + 2	64	50	802.8	79.7

D.4 Random Reads

The random read tests were conducted on both 4K and 64K I/O sizes with latency targets of 10ms and 20ms respectively.

TABLE D.7: CEPHFS RANDOM READS

Data Protection	I/O Size	Read BW (MiB/s)	Read IOPS	Read Avg Latency (ms)
3xRep	4K	20	5235.5	12.2
EC2 + 2	4K	9	2543.5	25.1
3xRep	64K	282	4522.5	14.1
EC2 + 2	64K	131	2111.1	30.3

TABLE D.8: RBD RANDOM READS

Data Protection	I/O Size	Read BW (MiB/s)	Read IOPS	Read Avg Latency (ms)
3xRep	4K	6	1719.7	9.3
EC2 + 2	4K	19	5007.9	12.9
3xRep	64K	75	1206.7	13.3
EC2 + 2	64K	239	3828.7	16.7

D.5 Backup/Recovery Simulations

The following test results are workload oriented.

Backup. The backup simulation test attempts to simulate the SUSE Enterprise Storage cluster being used as a disk-based backup target that is either hosting file systems on RBDs or is using CephFS. The test had a latency target of 200ms at the time of the test run. The latency target has since been removed.

TABLE D.9: BACKUP SIMULATION

Data Protection	Protocol	IO Size	Write BW (MiB/s)	Write IOPS	Write Avg Latency (ms)
3xRep	CephFS	64M	881	12.7	160976.1
EC2 + 2	CephFS	64M	982	14.3	138143.5
3xRep	RBD	64M	1038	15.2	167619.1
EC2 + 2	RBD	64M	887	12.8	182757.4

Recovery. The recovery workload is intended to simulate recovery jobs being run from SUSE Enterprise Storage. It tests both RBD and CephFS.

TABLE D.10: RECOVERY SIMULATION

Data Protection	Protocol	IO Size	Read BW (MiB/s)	Read IOPS	Read Avg Latency (ms)
3xRep	CephFS	64M	2494	37.9	52540.2
EC2 + 2	CephFS	64M	2770	42.2	54579.5
3xRep	RBD	64M	1327	20.5	23828.8
EC2 + 2	RBD	64M	2330	35.5	58883.9

D.6 KVM Virtual Guest Simulation

The kvm-krbd test roughly simulates virtual machines running. This test has a 20ms latency target and is 80% read with both reads and writes being random.

TABLE D.11: VM SIMULATION

Data Protection	Protocol	IO Size	Write BW (MiB/s)	Write IOPS	Write Avg Latency (ms)	Read BW (MiB/s)	Read IOPS	Read Avg Latency (ms)
3xRep	CephFS	4K	3	960.2	14.4	15	3856.5	13.1
EC2 + 2	CephFS	4K	1	264.9	70.7	4	1084.0	42.2
3xRep	RBD	4K	1	271.9	11.0	4	1099.9	11.8
EC2 + 2	RBD	4K	1	325.1	67.1	5	1307.9	28.6

D.7 Database Simulations

It is important to keep sight of the fact that Ceph is not designed for high performance database activity. These tests provide a baseline understanding of performance expectations should a database be deployed using SUSE Enterprise Storage.

OLTP Database Log. The database log simulation is based on documented I/O patterns from several major database vendors. The I/O profile is 80% sequential 8KB writes with a latency target of 1ms.

TABLE D.12: OLTP LOG

Data Protection	Protocol	IO Size	Write BW (MiB/s)	Write IOPS	Write Avg Latency (ms)	Read BW (MiB/s)	Read IOPS	Read Avg Latency (ms)
3xRep	CephFS	8K	33	4312.5	15.2	8	1077.8	2.3
EC2 + 2	CephFS	8K	9	1239.4	45.9	2	306.8	19.2
3xRep	RBD	8K	12	1616.0	9.6	3	406.3	1.1
EC2 + 2	RBD	8K	8	1083.8	51.7	2	270.5	19.7

OLTP Database Datafile. The OLTP Datafile simulation is set for an 80/20 mix of random reads and writes. The latency target is 10ms.

TABLE D.13: **OLTP DATA**

Data Protectio	Protocol	IO Size	Write BW (MiB/s)	Write IOPS	Write Avg Latency (ms)	Read BW (MiB/s)	Read IOPS	Read Avg Latency (ms)
3xRep	CephFS	8K	6	864.3	17.0	27	3462.6	14.1
EC2 + 2	CephFS	8K	2	307.4	64.3	9	1240.7	35.9
3xRep	RBD	8K	2	263.7	12.6	8	1064.0	11.9
EC2 + 2	RBD	8K	3	385.1	64.2	12	1553.1	19.8