SUSE® Enterprise Storage v5 Implementation Guide

Microsoft Azure

Written by:
David Byte, SUSE

Guide

www.suse.com

Implementation Guide

SUSE Enterprise Storage

SUSE

Introduction

The purpose of this document is to guide the reader in implementing SUSE® Enterprise Storage on the Azure cloud platform. The goal is to show architectural best practices for building this Ceph-based cluster, which can later support the implementation of any of the currently supported connection protocols.

Before attempting the process, we recommend that you read the document in its entirety, including the supplemental information in the Appendix.

Upon completion of the steps in this document, a working SUSE Enterprise Storage (v5) will be operational as described in the *SUSE Enterprise Storage 5 Deployment Guide*.

Target Audience

This reference architecture is targeted at administrators and consultants who deploy software-defined storage solutions within the Azure environment and make the different storage services accessible to their internal 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 and a specific set of recommendations for deployment of the storage platform.

Configuration

SUSE Enterprise Storage has been tested and validated in the Azure environment. With Azure, it is possible to select image sizes that have specific CPU, RAM and performance characteristics and thus carefully match requirements to the resources being consumed. For the testing performed, all work was done with DSx_v2 image types that were located in the US/East region. All roles were assigned to DSx_v2 images to ensure consistent performance. The role/functionality of each SUSE Enterprise Storage component will be explained in more detail in the Architectural Overview section.

CEPH OSD ROLE

- *Azure Standard_DS5_v2*
- *4 nodes with 10 standard drives each*
- *Enhanced network adapter*

ADMIN, ISCSI GW, & MONITOR NODES

- *Azure Standard_DS4_v2*

MDS & RGW NODES

- *Azure Standard_DS5_v2*

NETWORKING INFRASTRUCTURE

The enhanced networking available with the Standard_DSx_v2 image size is appropriate for higher throughput. A single network with no logical segmentation may be used.

SOFTWARE

- *SUSE Enterprise Storage 5*

Note: The SUSE Enterprise Storage subscription includes a limited-use [for SUSE Enterprise Storage] entitlement for SUSE Linux Enterprise Server.

Business Problem and Business Value

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.

Business Problem

Customers moving services to Infrastructure as a Service (IaaS) providers can encounter challenges when trying to fulfill SLAs for something as simple as backup and recovery. This is compounded when the IaaS provider doesn't offer a service that provides exactly what is needed to service the business requirements. High performance backup is one of the key examples.

Business Value

SUSE Enterprise Storage provides software that enables customers to aggregate multiple resources from Azure together and provide the storage performance required. This deployment model also allows for a closer replication of the on-premises environment, enabling an easier migration to the cloud from a process perspective.

Requirements

Enterprise storage systems require reliability, availability and serviceability, which together are known as RAS. The legacy 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 instances within Azure, the result is a solution that meets the customer's expectation for enterprise-level service.

Functional Requirements

A SUSE Enterprise Storage solution is:

- *Simple to set up and deploy, within the documented guidelines of system hardware, networking and environmental prerequisites.*
- *Adaptable to constraints needed by the business, both initially and as needed over time for performance, security and scalability concerns.*
- *Capable of providing optimized object and block services to client access nodes, either directly or through gateway services.*

Architectural Overview

This section complements the *SUSE Enterprise Storage Technical Overview* document available online, which presents the concepts behind software-defined storage and Ceph.

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 its underlying object storage. The result is a storage solution that is abstracted from the hardware. For purposes of this document, we use the terms nodes and virtual machines interchangeably.

Ceph supports both native and traditional client access. The native clients are aware of the storage topology and communicate directly with the storage daemons, resulting in horizontally scaling performance. Non-native protocols, such as ISCSI, S3 and NFS, require the use of gateways. While these gateways might be thought of as a limiting factor due to inherent protocol limitations, the ISCSI and S3 gateways can scale horizontally using load balancing techniques. In this example, we will deploy both iSCSI and Object gateways.

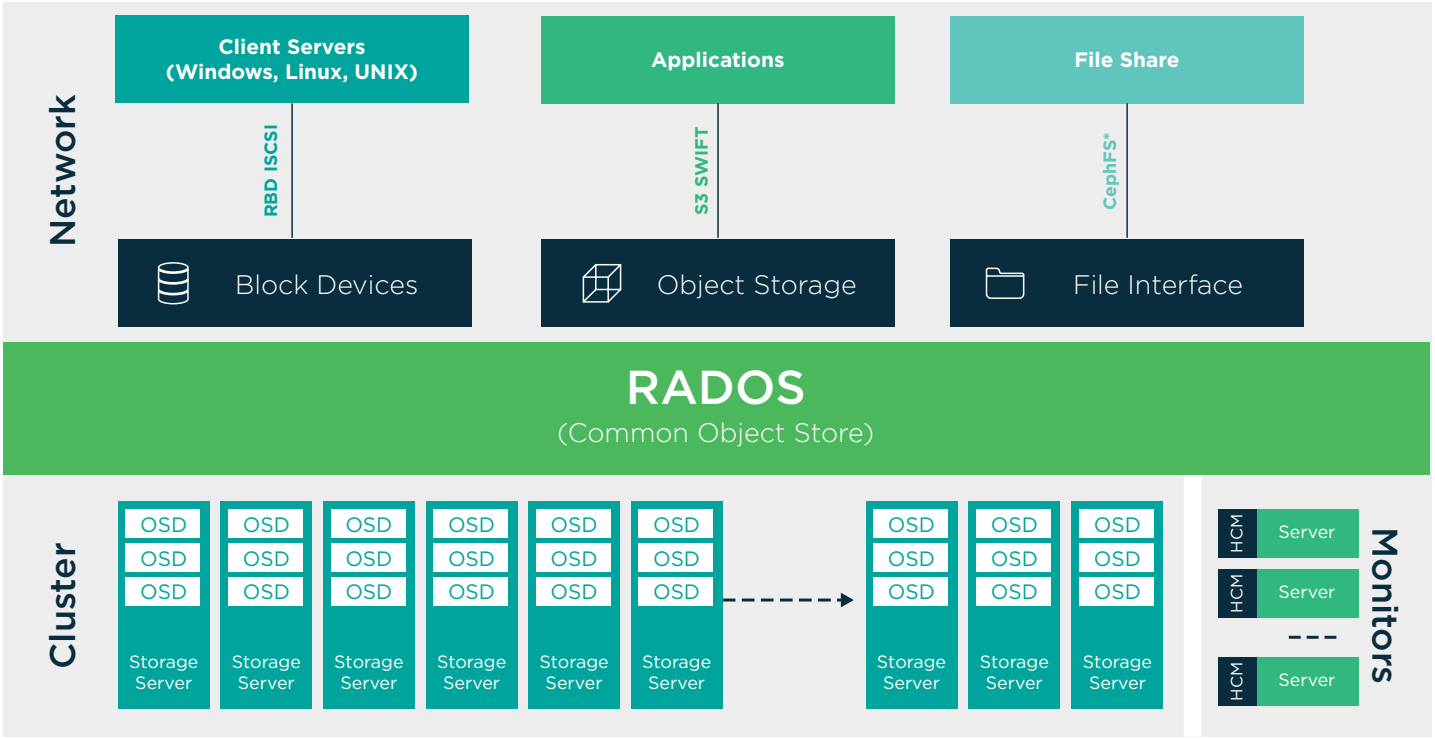


Figure 1. *Ceph architecture diagram*

In addition to the recommended 'Enhanced' network interfaces, the minimum SUSE Enterprise Storage cluster is comprised of one administration server (physical or virtual), four object storage device nodes (OSDs) and three monitor nodes (MONs). There are also HA implementations of the Management and Orchestration node, which are covered separately in this guide. The following recommendations are specific to our SUSE Enterprise Storage on Azure implementation:

- *A single Standard_DS4_v2 instance size was deployed for the admin node function. This node provides the Salt master, DNS server, NTP server, SMT server and a location to run the OpenATTIC web user interface for managing and monitoring the cluster.*
- *Monitor and Manager functions are combined and provided by the Standard_DS4_v2 instance size.*
- *The RADOS gateways should be deployed in their own instance(s) or separated from the OSD and MDS nodes. 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. It is recommended that you use the Standard_DS5_v2 or larger for this role.*
- *The MDS nodes provide metadata services when the distributed file system (CephFS) is deployed. This role consumes more RAM resources as the number of files and directories grows. This role is also capable of multiple active-active servers that subdivide the directory structure.*

It is advisable to have at least two MDS nodes in the cluster. It is also recommended to use the Standard_DS5_v2 instance size for this role. This role should be deployed independently of other roles.

- *The storage nodes are of the Standard_DS5_v2 instance size. Microsoft provides clear guidance on the performance capability of these instances (***https://docs.microsoft.com/en-us/azure/virtual-machines/linux/sizes***), which allows for a relatively simple estimation of the nodes required to achieve a particular performance requirement. The storage nodes contain individual storage devices that are each assigned an Object Storage Daemon (OSD). The OSD daemon assigned to the OSD 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. Azure disk devices leverage locally replicated storage (LRS) within Azure, meaning that there are three replicas of the storage behind each presented device. This enables a reduction in the data protection requirement employed at the software pool layer. For example, Erasure Coded 3+1.*

Networking Architecture

A software-defined solution is 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 general network perspective for Ceph on physical hardware, this translates into:

- *Separation of cluster (backend) and client-facing network traffic, which isolates Ceph OSD daemon replication activities from Ceph client to storage cluster access.*
- *Redundancy and capacity in the form of bonded network interfaces connected to switches.*

Figure 2 shows the logical layout of the traditional physical Ceph cluster implementation.

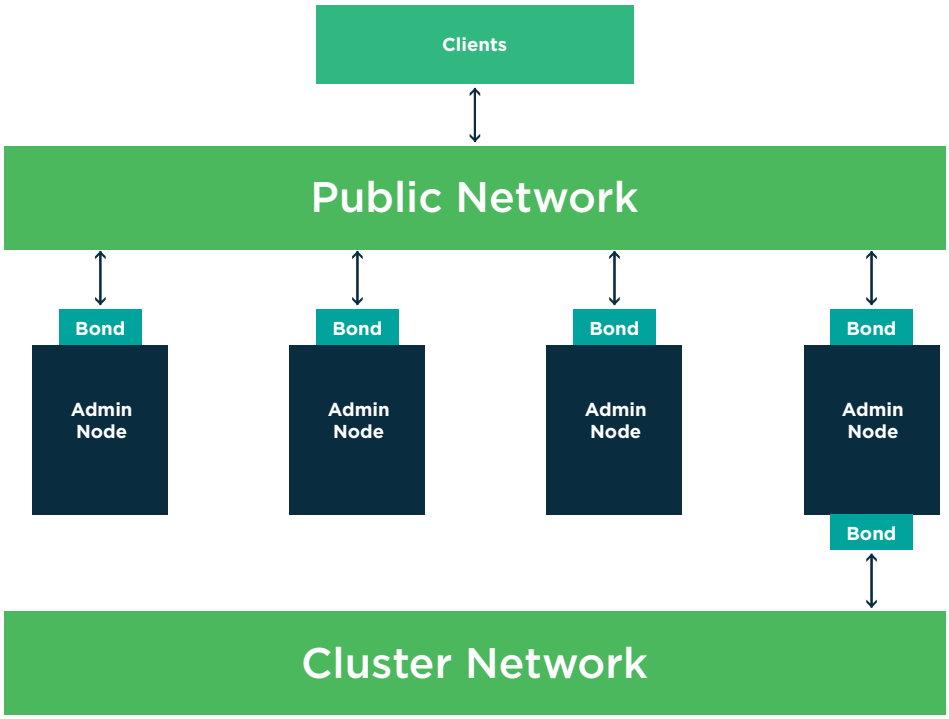


Figure 2. *Sample networking diagram for Ceph cluster*

However, in the Azure environment, bandwidth is throttled per instance. Therefore, the only advantage of logical segmentation is privacy. In the Azure test environment, a single network segment using a single 'Enhanced' network interface within a private IP range was utilized for each node. In this configuration, all instances are able to reach each other by name and IP within a single flat subnet.

Network/IP Address Scheme

192.168.101.0/28

Reference /etc/hosts file in Appendix D.

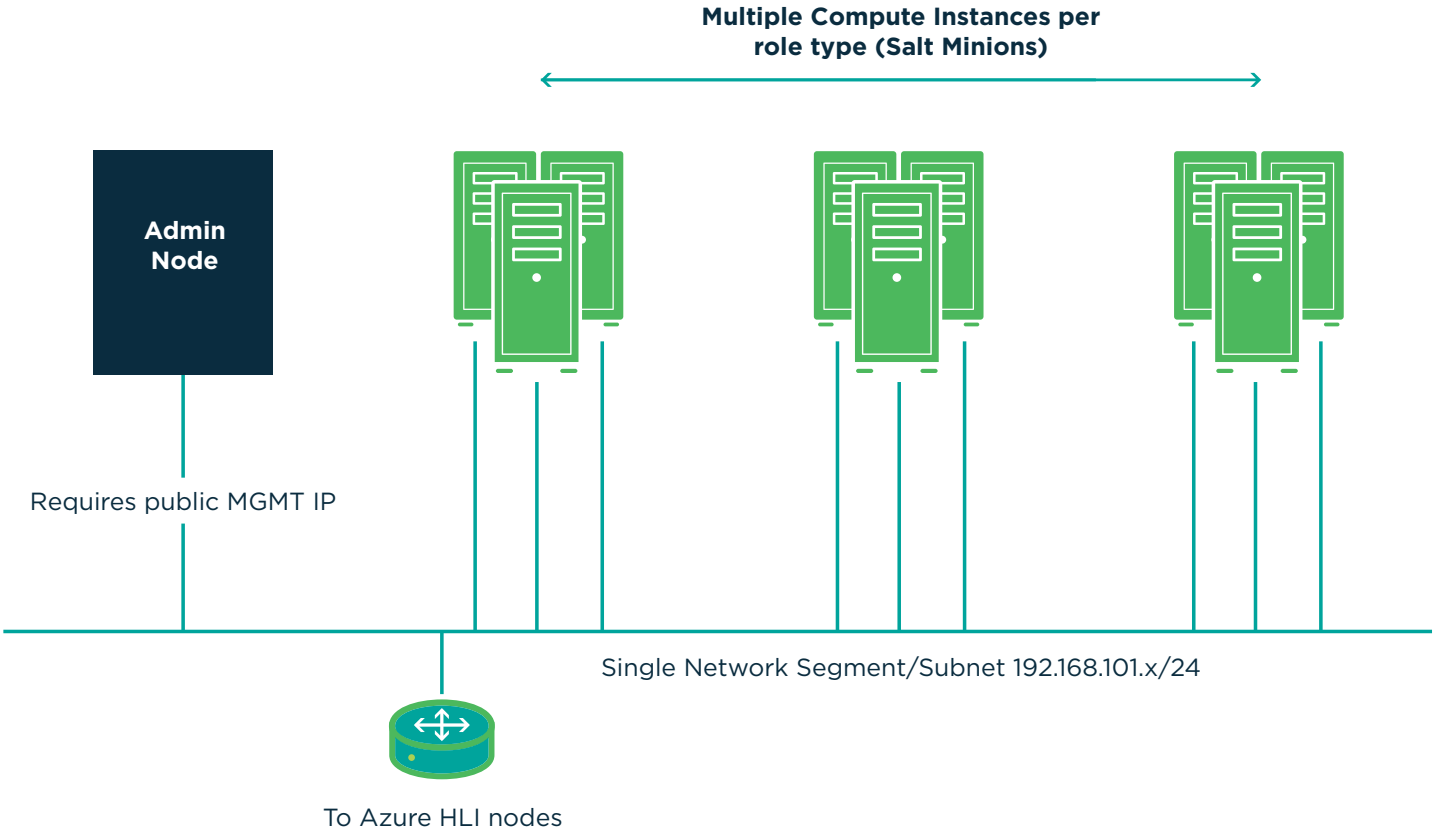


Figure 3. *Sample networking diagram for Ceph cluster with single client and cluster network*

Component Model

The preceding sections provided significant details on the overall Azure environment, 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 SLES12 SP3—Enables enterprise customers to optimize the management of SUSE Linux Enterprise (and extensions such as SUSE Enterprise Storage) software updates and subscription entitlements. It establishes a proxy system for SUSE Customer Center with repository and registration targets.

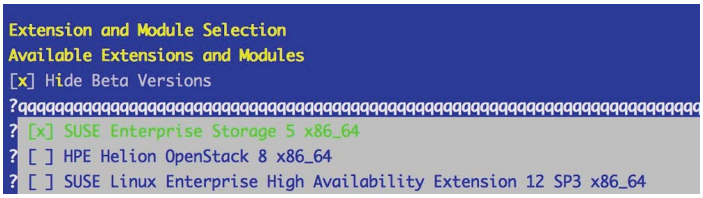
SUSE Enterprise Storage—Provided as an extension on top of SUSE Linux Enterprise Server, this intelligent software-defined storage solution (powered by Ceph technology with enterprise engineering and support from SUSE) enables customers to transform their enterprise infrastructure to reduce costs while providing unlimited scalability. This most recent release brings a new underlying storage technology, Bluestore, to the

product. Bluestore significantly reduces long tail latencies and significantly improves performance for some use cases. SUSE Enterprise Storage 5 also brings the distributed file system CephFS to production with multiple meta-data server support, allowing for broad usage of this highly-performant, scale-out technology across many use cases.

1. Only the Admin node has public internet access.
   a. The Admin node is registered with SUSE Customer Center (SCC) and has the SMT server service installed. All other nodes register with this SMT server to get Linux and Storage packages.
   b. The Admin nodes provide all other services to other nodes, including DNS and NTP.
   c. The other nodes have no internet access.
      i. Optionally, the Admin node could be turned into a NAT gateway.
2. All nodes are created with an additional public IP (the Azure Portal default). This is the easiest configuration.
   a. The Admin node functions as a Salt Master and hosts OpenAttic.
   b. All nodes receive a public IP address and, while they are able to leverage Azure DNS and access external Stratum 1 Time servers, hosting these services locally on the Admin node is still recommended.
   c. Although all nodes can be registered directly to SUSE Customer Center for updates and packages, a local or regional SMT server is recommended.

Deployment

This section should be considered a supplement to the online documentation, specifically the *SUSE Enterprise Storage 5 Deployment Guide* and the *SUSE Linux Enterprise Storage Administration Guide*. It is assumed that a Subscription Management Tool server exists within the environment or that all nodes have registered with the public SCC service. If not, please follow the information in Subscription Management Tool (SMT) for SLES 12 SP3 to make SMT available. This will reduce the total network traffic into the cloud. After launching "yast2" from the command line, choose "Software" > "Add System Extensions or Modules," then choose "SUSE Enterprise Storage 5 x86_64," and complete the installation.



The following packages are required on all nodes to leverage this guide:

Network Deployment Overview

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

- *Network IP addressing and IP ranges need proper planning. In physical environments, a single storage subnet should be used for all SUSE EnterÐÐÐprise Storage nodes on the primary network, with a separate, single subnet for the cluster network. However, within Azure, a single flat network is appropriate. Depending on the size of the installation, subnet mask ranges larger than 24 bits might be required. For a small cluster with a growth ceiling of 1 admin node, 2 gateways, 2 MDS and 4 OSD's (example herein), a 28-bit mask would be sufficient. When planning the network, current as well as future growth should be taken into consideration. Security Groups, Routing Tables and Public IP's are configured such that all nodes are manageable and can access the desired resources.*
- *SSH and Salt. It is advisable to deploy this first node using the public key corresponding to your local admin workstation's private key, unless you leverage a key management system or prefer password authentication. It is also recommended that you then enable password authentication and a root password on each node. Additionally, generate an SSH key pair on the Admin node and copy the public key to itself and to the other Salt minion nodes, using the ssh-copy-id command. This will ease cluster member administration from the Admin node.*

Most administration will be done via Salt, using its own authentication keys.

- *Set up DNS A records for all nodes or rely entirely on the /etc/hosts file. Azure DNS is not recommended at this time, due to complex naming conventions and additions to the /etc/hosts and /etc/resolv.conf files. These names would need to match the salt configuration files. The Salt master can be leveraged as a DNS server. The DNS server role can be installed via YAST. It is advised that all minions point to the salt master, which in turn points to an Azure DNS server service via resolv.conf.*
- *Ensure that you have access to a valid, reliable NTP service. This is a critical requirement for all nodes. Do not rely on the internal cloud services for time synchronization, which only provide time update during instance boot. The Salt master can be leveraged as an NTP server or all nodes can be pointed to a list of public Stratum 1 NTP servers. Alternatively, if using Public IP's for all nodes, each node can point to the same Stratum 1 server list. Reference the SUSE Linux Enterprise Server 12 Administration Guide for more information on these topics.*

Operating System Deployment and Configuration

The operating system deployment for all nodes is quite simple in the Azure environment.

- *The OS image will be "SLES 12 SP3 BYOS."*
- *If using SMT, register the system against the SMT server.*
- *When prompted for extensions, select SUSE Enterprise Storage 5.*
- *On the Suggested Partitioning selection, select Edit Proposal Settings and uncheck Propose Separate Home Partition.*
- *After installation is complete, run zypper up to ensure that all current updates are applied. Reboot and repeat zypper up.*
- *Configure proper private IP, hostname, name and time resolution throughout.*
- *All nodes should resolve similar to the following example:*

```
salt:~ # hostname
 salt
salt:~ # hostname -f
 salt.ses
```

- *All nodes should be able to ping each other using both short name and FQDN.*

SW Deployment Configuration (DeepSea and Salt)

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

Three key Salt imperatives should be followed. These are described in detail in Section 4 (Deploying with DeepSea and Salt).

1. The Salt master is the host that controls the entire cluster deployment. It also pushes changes and cluster member or role updates and hosts the OpenAttic Web UI. Salt minions are client nodes controlled by a Salt master. OSD, monitor and gateway nodes are all Salt minions in this installation. In this scenario, the Salt master node also runs the Salt minion service.
2. Salt minions need to correctly resolve the Salt master's host name over the network. If separate network interfaces are employed for this purpose, this can be achieved by way of configuring unique host names per interface (osd1-cluster.ses.east.azure.com and osd1-public.ses.east.azure.com) in DNS and/or local /etc/hosts files.
3. DeepSea consists of a 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:

- *Install the salt-master packages on the admin node:*

```
- zypper in salt-master
```

- *Start the salt-master service and enable:*

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

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

```
- zypper in salt-minion
```

- *Configure all minions (including the file on the Salt master) to connect to the Salt master: Modify the entry for master in* `/etc/salt/minion` *as below:*
  - `master: salt`
- *Start the* salt-minion *service and enable on all nodes:*
  - `systemctl start salt-minion.service`
  - `systemctl enable salt-minion.service`
- *Clear all non-OS drives on the OSD nodes, reset the labels and reboot the nodes:*
  - `dd if=/dev/zero of=/dev/sda bs=1M count=1024 oflag=direct`
  - `zypper install gptfdisk`
  - `sgdisk -Z --clear -g /dev/sda`
  - `reboot`
- *Accept and list all salt keys on the Salt master:* `salt-key --accept-all` *and verify their acceptance:*
  - `salt-key --list-all`
  - `salt-key --accept-all`
  - `salt-key --list-all` (Note that all keys are accepted.)
- *At this point Salt should resolve all nodes as illustrated:*
  - `salt:~ # salt '*' test.ping`

```
salt:
  True
osd2:
  True
mon3:
  True
osd3:
  True
osd4:
  True
osd1:
  True
....
```

- *Install DeepSea on the Salt master which is the Admin node:*
  - `zypper in DeepSea`
- *At this point, you can deploy and configure the cluster:*
  - Prepare the cluster:
    - `deepsea stage run ceph.stage.prep`
  - Run the discover stage to collect data from all minions and create configuration elements:
    - `deepsea stage run ceph.stage.discovery`
  - A `/srv/pillar/ceph/proposals/policy.cfg` file needs to be created to instruct Salt on the location and configuration files to use for the different components that make up the Ceph cluster (Salt master, admin, monitor and OSDs).
    - *See Appendix B for the* `policy.cfg` *file used in the installation.*
  - Next, proceed with the configuration stage to parse the policy.cfg file and merge the included files into their final form.
    - `deepsea stage run ceph.stage.configure`
  - The next step manages the actual deployment. Deploy monitors and ODS daemons first before considering gateways:
    - `deepsea stage run ceph.stage.deploy` *(Note: The command can take some time to complete, depending on the size of the cluster).*
  - Check for successful completion via:
    - `ceph –s`

If "ceph -s" does not produce a status message, then the policy.cfg, minion, resolv.conf, or host files throughout the cluster might not be correctly populated or formatted. The deployment stage relies heavily on correct name referencing. Azure-provided DNS throughout the cluster is not recommended for this initial configuration.

- Finally, deploy the services (gateways [iSCSI, RADOS] and openATTIC, to name a few):
  - `deepsea stage run ceph.stage.services`

Post-Deployment Quick Test

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

```
ceph status (ceph -s)
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
```

You should now be able to connect to OpenAttic via the web UI, as illustrated in the SUSE Enterprise Storage Deployment guide.

Pool Considerations

Given that the Azure standard disks are replicated multiple times behind the scenes, it is possible to use 2x replication or erasure coding with a lower number of coding chunks (e.g., 7 data chunks and 2 EC chunks for a 9-OSD cluster). Your reliability expectations will be set by compliance or other business requirements. Please read all relevant Azure SLA's when designing your pools.

Appendix A: Bill of Materials

Component / System

SUSE Product 11 adds a host of new capabilities to your IT arsenal, including the following:

Role	Qty	Component	Notes
Storage	4	Azure Standard_ DS5_v2	Each node has: 10 additional standard drives of 4TB
GW, MDS	4	Azure Standard_ DS5_v2	
Admin, Mon, Mgr	4	Standard_DS4_v2	
Software—OS	1	SUSE Enterprise Server 12 sp3 or later	Base OS license
Software—SES	1	SUSE Enterprise Storage Subscription Base configuration	Allows for 4 storage nodes and 8 infra-structure nodes

Appendix B: Policy.cfg

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

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

# MON
role-mon/cluster/mon*.sls

# MGR (mgrs are usually colocated with mons)
role-mgr/cluster/mon*.sls

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

# IGW
role-igw/cluster/gw*.sls

# RGW
role-rgw/cluster/gw*.sls

# NFS
#role-ganesha/cluster/ganesha*.sls

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

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

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

Appendix C: Resources

SUSE Enterprise Storage Technical Overview
www.suse.com/docrep/documents/1mdg7eq2kz/suse_enterprise_storage_technical_overview_wp.pdf

SUSE Enterprise Storage 5 release notes
www.suse.com/releasenotes/x86_64/SUSE-Enterprise-Storage/5/

SUSE Enterprise Storage v5—Administration Guide
www.suse.com/documentation/suse-enterprise-storage-5/book_storage_admin/data/book_storage_admin.html

SUSE DeepSea Wiki
https://github.com/suse/deepsea/wiki

SUSE Linux Enterprise Server 12 SP3—Administration Guide
www.suse.com/documentation/sles-12/book_sle_admin/data/book_sle_admin.html

Subscription Management Tool for SLES 12 SP3
www.suse.com/documentation/sles-12/book_smt/data/book_smt.html

Appendix D: Hosts File

An issue currently exists in which DeepSea wait times result in a timeout error within the DeepSea execution model, due to IPv6 entries in the hosts file. This can be overcome by adding the local host name to the line `::1*`

```
Example:
::1     localhost salt ipv6-localhost ipv6-loopback
```

It is advised not to edit the IPv6 configuration. The IPv6 errors are a nuisance, but do not produce a fatal error. Editing the IPv6 lines might result name resolution failure and DeepSea execution failure.

Below is a sample functional hosts file:

```
> cat /etc/hosts
127.0.0.1    localhost
# special IPv6 addresses
::1     localhost ipv6-localhost ipv6-loopback
fe00::0     ipv6-localnet
ff00::0     ipv6-mcastprefix
ff02::1     ipv6-allnodes
ff02::2     ipv6-allrouters
ff02::3     ipv6-allhosts
192.168.101.2     salt.ses salt
192.168.101.3     mon1.ses mon1
192.168.101.4     mon2.ses mon2
192.168.101.5     mon3.ses mon3
192.168.101.6     osd1.ses osd1
192.168.101.7     osd2.ses osd2
192.168.101.8     osd3.ses osd3
192.168.101.9     osd4.ses osd4
192.168.101.10    gw1.ses gw1
192.168.101.11    gw2.ses gw2
192.168.101.12    mds1.ses mds1
192.168.101.13    mds2.ses mds2
```

Additional contact information and office locations:
www.suse.com

www.suse.com