White Paper

SUSE Enterprise Storage

SUSE Enterprise Storage™ Technical Overview

Table of Contents

Storage Redefined Powered by Ceph

Network storage used to mean a file server—a large, standalone server system with a big hard drive that stored and served files in a hierarchical filesystem.

But today's world of Big Data, data centers, and on-prem and public clouds demands better scalability, better performance and less down-time. And, despite the huge demands they're placing on network storage systems, companies are not really interested in spending more money or expending additional IT resources to build complex, proprietary custom storage solutions.

If your organization is like most, you need to store more data than ever before—and to remain competitive, you need a simple, reliable, self-healing storage solution that scales efficiently and supports state-of-the-art storage technology on commodity hardware. SUSE Enterprise Storage™ offers an affordable and highly scalable software-defined storage solution tailored to the needs of your modern hybrid data center. SUSE Enterprise Storage is powered by Ceph, an advanced, open-source solution that provides object, block and file storage in a unified, scalable way.

Read on for an introduction to SUSE Enterprise Storage and learn about some of the tools you'll need to deploy and manage your SUSE Enterprise Storage environment.

Software-Defined Storage

SUSE Enterprise Storage is based on software-defined storage technology, which separates the storage logic (the rules and protocols for managing the data) from the underlying hardware. A conventional storage scenario might be a server attached directly to an external hard drive or another block storage device (see Figure 1). In this setting, it's customary to say the server "writes the data to the disk." But at a closer look, the server doesn't really know whether the device on the other end of the connection is a disk or not. The server merely transmits data across the interface using the communication standard defined for the device, and it listens for information back from the device using the same communication standard.

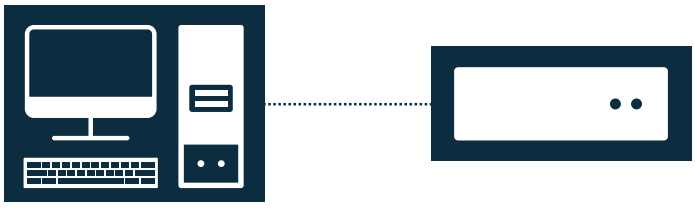

Figure 1. *A conventional file storage scenario: a server writes to an external disk.*

In fact, anything could be on the other end of the interface as long as it interacts with the system using the predefined communication format. For instance, another computer could be at the end of the interface, and it could be running a program that emulates a storage device. To take the scenario a step further, what if the thing on the other end of the interface is a whole self-healing, redundant and highly scalable storage cluster (see Figure 2). Instead of emulating a single type of storage device, the cluster would have several interfaces that let it interact with clients in a number of different ways. It could present a block storage interface to systems that expect to write to a disk, and it could also present a filesystem interface (more like a conventional file server) to clients configured to work with directories and files.

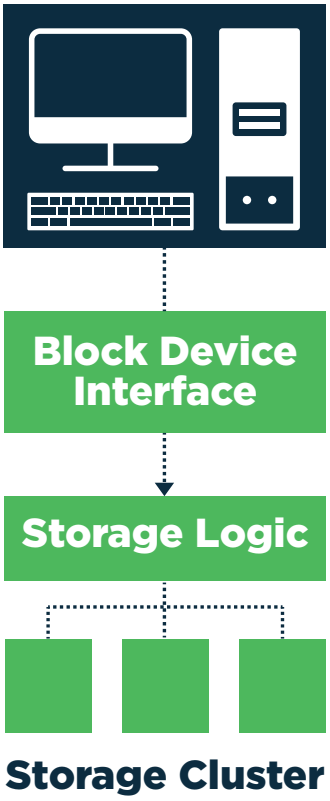

Figure 2. *Instead of writing directly to a disk drive at the other end of the interface, the server writes to a complete self-healing storage cluster. This storage scenario is the starting point for understanding the power of software-defined storage.*

Behind the scenes, the cluster provides invisible fault tolerance, so the client never has to worry about data loss, and the cluster manages the data using fast and ultra-scalable object storage techniques.

The virtual storage scenario in Figure 2 requires a collection of logic rules, program interfaces and software components to provide a complete solution for seamless integration with storage clients and physical resources. That complete solution is software-defined storage.

SUSE Enterprise Storage is a powerful and versatile storage solution that delivers all the benefits of software-defined storage with minimal configuration and minimal startup costs. SUSE Enterprise Storage and its constellation of simple management tools—including the Ceph Dashboard—let you launch and configure a complete software-defined storage cluster in a matter of hours. And because SUSE Enterprise Storage is based on Ceph technology, a leading software-defined storage solution based on open standards, you'll never have to worry about vendor lock-in or the complications of proprietary technologies. At the same time, SUSE Enterprise Storage offers the value-added testing, certification and support that comes with an enterprise-grade solution.

SUSE Enterprise Storage and Ceph

An important component of Ceph is the RADOS object store. As shown in Figure 3, RADOS supports a cluster of object storage nodes and a system for storing and retrieving data from the cluster using object storage techniques (see the box titled "Understanding Object Storage" on pages 4-5).

Ceph provides a library of components that interface with file, block and object storage systems. Ceph's diverse collection of interfaces give the system much of its versatility and power, allowing the cluster to fit in conveniently with a number of usage scenarios.

The components of the Ceph architecture distribute the data through the cluster, balancing the load and continually redistributing it to prevent failure and to optimize efficiency. If a driver or a node goes down, the system recalculates and rebalances. If another node is added, the cluster finds it and rebalances to redistribute the workload. All data is replicated to prevent data loss based on rules you set. The result is a complete, versatile, and self-healing storage system.

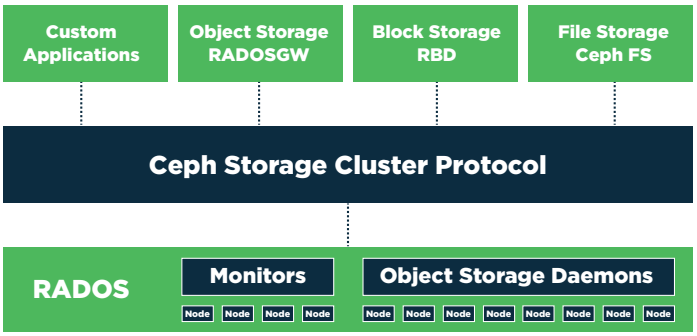

Figure 3. *The Ceph architecture at a glance.*

Interfaces

Depending on how you intend to use your SUSE Enterprise Storage cluster, you'll need to spend some time configuring the interfaces and gateways that provide access to the cluster. The most common options for interfacing with a SUSE Enterprise Storage cluster are:

- *RBD (RADOS Block Device): a distributed block device with a Linux kernel client and QEMU/KVM driver*
- *CephFS: a POSIX filesystem*
- *RADOS Gateway (RGW): A bucket-based REST gateway*
- *iSCSI: a network protocol for block storage*

The RBD interface lets the Ceph cluster act as an external block storage device (as depicted in Figure 2).

CephFS is a Linux filesystem that lets the cluster serve the familiar role as a file server. The CephFS filesystem is an ideal interface for legacy applications that require a conventional filesystem structure, and it offers a low learning curve for users accustomed to interacting with files and directories. The version of CephFS included with SUSE Enterprise Storage also provides gateways for NFS and Samba (SMB) network clients. Samba can be deployed to expose the CephFS filesystem to SMB clients such as Windows and macOS. Samba gateway deployments can be standalone or, when combined with CTDB, highly available.

The RGW object-based RESTful gateway lets the Ceph cluster act as an object store for applications and infrastructures that require object-based storage. RGW allows you to use Ceph as a storage system for the Amazon S3 and OpenStack Swift. SUSE Enterprise Storage is tooled to act as back-end storage for OpenStack cloud systems. Many SUSE customers use SUSE Enterprise Storage with OpenStack, and the SUSE support staff is well versed in the issues and solutions required for providing seamless OpenStack support.

Ceph also provides an iSCSI gateway, a multiplatform interface that lets Windows, VMware, and other iSCSI-ready clients write to the cluster as if writing to an iSCSI device. This is familiar to many who use proprietary SAN or NAS devices, and it delivers the fault tolerance and scale-out capabilities through multi-path I/O.

See the **SUSE Enterprise Storage documentation** for the Deployment Guide, the SUSE Enterprise Storage Administration Guide and more on how to configure and manage your cluster.

Fault Tolerance

The key to any storage solution is its ability to preserve data even when failures occur. Ceph offers automated fault tolerance to prevent data loss in the case of a failed node or disk. One option is Ceph's built-in system for data replication, which you can define at the time you set up your cluster.

Ceph also supports an optional fault tolerance technique known as erasure coding. Erasure coding is a flexible system that includes data striping with additional chunks of encoded redundancy information. The choice of replication or erasure coding depends on your storage needs and the characteristics of your environment. In general, replication tends to be a little faster, but erasure coding makes more efficient use of disk resources, thus reducing the overall cost per gigabyte.

The version of Ceph included with SUSE Enterprise Storage offers expanded erasure coding features, including overwrite support for erasure coded pools. Full support for erasure coding is now available with the RBD block storage, CephFS filesystem, and RGW object storage interfaces.

Understanding Object Storage

Object-based storage is a cutting-edge storage system used with some of the Internet's highest-traffic websites.

A conventional block storage environment breaks data into blocks of fixed length and uses a centralized system for storing and retrieving the blocks. To store a file, the filesystem must break the file into block-sized pieces and fit the blocks into the storage structure. When a file is opened, the filesystem retrieves the blocks one by one, reassembling them into the original file. The process of breaking up and assembling blocks slows down the process of storing and retrieving data. Perhaps more importantly, the filesystem must maintain some form of centralized table or broker with knowledge of where the chunks of data are stored on the system, and the need to look up location information in this table becomes a bottleneck that means the system does not scale well for high-traffic and large networks.

Object storage offers a higher-performing and much more scalable approach. In an object-based storage system, the size of the data can vary, thus eliminating the need to break the file into fixed-size blocks. Knowledge of where the data is stored is derived through a calculation and distributed through the network, which eliminates the need for a central broker or lookup table that could potentially form a bottleneck.

continued on next page

Understanding Object Storage *(continued)*

Ceph treats each chunk of data as an object. If the data comes from a block storage interface, an object corresponds to a single block of data. If the data comes from a filesystem interface, an object corresponds to a block of the file.

The Ceph network consists of clients, monitors and Object Storage Daemons (OSDs). The monitor nodes maintain the map of the cluster known as the CRUSH map. The CRUSH map contains a directory of storage devices and a system of rules for traversing the hierarchy when storing data (the CRUSH algorithm). A client system joining the network receives a copy of the CRUSH map.

The object storage daemons are organized into pools. The client writes data to a specific pool and uses a hash of the object name to determine which placement group within the pool will receive the data. The primary OSD for the placement group receives the data and uses additional rules within the CRUSH algorithm to store and replicate the data to the other OSDs holding copies of the placement group.

To retrieve an object from the data store, the client follows a similar process, addressing the request to the placement group derived from the CRUSH algorithm. The primary OSD within the placement group receives the request and can either serve it directly or, during reconstruction, use further information revealed in the CRUSH algorithm to discover the object location.

Note that this design allows the client to operate with only minimal knowledge of the underlying cluster. The client sees the cluster as a collection of logical entities—pools and placement groups—without directly associating an object with a specific storage location. Knowledge of specific storage locations is kept within the network itself, which allows the cluster to rebalance dynamically, adapting invisibly to changes. Physical storage is managed at the placement group level, and because the cluster includes multiple placement groups, the burden of performing CRUSH calculations is distributed evenly around the network, thus eliminating the possibility of bottlenecks. (See the box entitled "On the Disk" for more on disk storage in SUSE Enterprise Storage.)

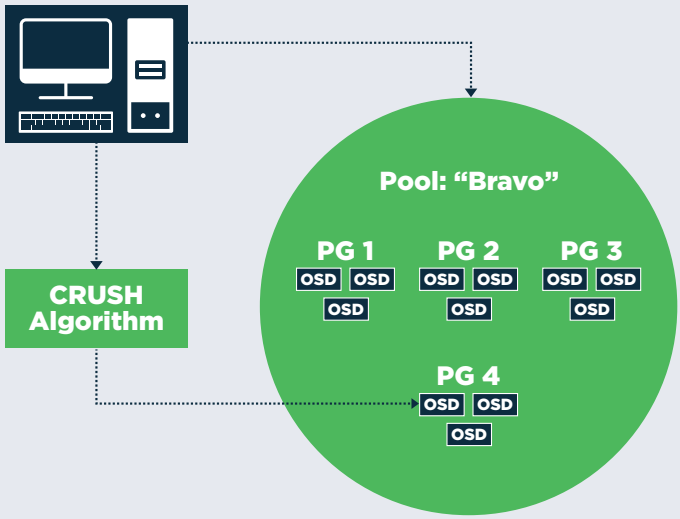

Figure 4. *The client writes to a storage pool and uses a hash algorithm defined through the CRUSH algorithm to determine a placement group within the pool.*

While the OSDs are processing requests, they are also checking on the health of the network. Each OSD regularly checks neighboring OSDs to be sure they are running and sends status information back to the monitor, which replicates the information on other monitor nodes. The monitor incorporates the status information in new versions of the CRUSH map, which it updates at regular intervals.

Once your Ceph cluster is up and running, the storage decisions, load balancing, replication and failure protection go on with little or no intervention from the system administrator. Much of the human input occurs up front when you're building the cluster and designing the structure of OSDs, monitors and gateway interface components. Some networks achieve better performance by customizing the CRUSH algorithm for the local environment. The experts at SUSE can help you install and configure your SUSE Enterprise Storage cluster for optimum performance

On the Disk

Ceph's CRUSH algorithm offers a powerful and versatile approach for scalable, inexpensive data storage. However, once the data has traversed the path from the client down through the data pools and into the placement groups, eventually, it still must be saved to a disk or other permanent storage medium.

Previous versions of Ceph relied on POSIX-compatible filesystems such as XFS, Btrfs, and Ext4 for disk storage. These general-purpose filesystems were effective, but they offered more features than Ceph really needed, which tended to slow down performance.

SUSE Enterprise Storage now comes with the new Ceph component called BlueStore, which manages the storage to disk without the extra overhead associated with a general-purpose filesystem.

The BlueStore back-end disk storage component means that SUSE Enterprise Storage is the fastest yet. Experts predict BlueStore could improve save times by up to 50 percent, depending on your hardware and network configuration.

Recommended Production Cluster Configuration

- Seven Object Storage Nodes
  - No single node exceeds ~15% of total storage
  - 10 Gb Ethernet (four physical networks bonded to multiple switches)
  - 56+ OSDs per storage cluster
  - RAID 1 OS disks for each OSD storage node
  - SSDs for journal with 6:1 ratio SSD journal to OSD
  - 1.5 GB of RAM per TB of raw OSD capacity for each Object Storage Node
  - 2 GHz per OSD for each Object Storage Node
- Dedicated physical infrastructure nodes
  - Three Ceph Monitor nodes: 4 GB RAM, 4-core processor, RAID 1 SSDs for disk
  - One SES management node: 4 GB RAM, 4-core processor, RAID 1 SSDs for disk
- Redundant physical deployment of gateway or Metadata Server nodes:
  - RADOS Gateway nodes: 32 GB RAM, 8-core processor, RAID 1 SSDs for disk
  - iSCSI Gateway nodes: 16 GB RAM, 4-core processor, RAID 1 SSDs for disk
  - Metadata Server nodes (one active/one hot standby): 32 GB RAM, 8-core processor, RAID 1 SSDs for disk

What You'll Need

It's possible to configure Ceph in a virtual environment, but for performance reasons, you're better off building your cluster on bare-metal hardware. Although a complete Ceph cluster can run on a single node, SUSE recommends at least seven object storage nodes for a production environment. See the box entitled "Recommended Production Cluster Configuration" for a summary of the recommended cluster components.

Your network should consist of a public part and a trusted internal part. Ceph clients and the rest of the network reside on the public part. Inter-cluster communication between the Ceph nodes should occur on the trusted internal network.

Getting Started

Configuring a SUSE Enterprise Storage cluster begins with installing SUSE Linux Enterprise Server (SLES) with the SUSE Enterprise Storage extensions. The deployment guide outlines some additional steps you'll need to take once the system is installed, including setting up the Network Time Protocol (NTP), configuring networking, turning off any firewalls and AppArmor, and installing an SSH server.

After you've set up and configured SLES and SUSE Enterprise Storage on all nodes, you're ready to create and deploy your Ceph cluster. The easiest way to set up a Ceph cluster is with DeepSea, a collection of scripts and components based on the Salt configuration management framework that's at the core of any SUSE Enterprise Storage deployment.

If you're using DeepSea to configure your Ceph cluster, the first step is to set up one of the cluster nodes to serve as the Salt master. This is accomplished by including the Deepsea pattern during installation, or by running the following command (as root) to install the salt-master packages on the Salt master:

```
$ sudo zypper in deepsea
```

The other cluster nodes will act as Salt minions. Install the salt-minion package on the cluster nodes, including the Salt master, using the following command:

```
$ sudo zypper in salt-minion
```

You'll need to configure the Salt minions to point to the Salt master. By default, they'll point to a host named "salt," so if your master has a different name (such as "admin"), you'll need to edit the name in /etc/salt/minion. See the **SUSE Enterprise Storage Deployment Guide** for details on completing the Salt configuration and starting the Salt service.

Once Salt is configured and deployed on all cluster nodes, you're ready to set up the Ceph cluster. DeepSea collects the configuration tasks into a series of stages. To deploy the cluster, you just need to execute the commands associated with the various stages on the Salt master, as follows:

1. To install any updates:

   ```
   $ sudo deepsea stage run ceph.stage.prep (or deepsea stage run ceph.stage.0)
   ```

2. To detect hardware and discover system information necessary for the Ceph configuration:

   ```
   $ sudo deepsea stage run ceph.stage.discovery (deepsea stage run ceph.stage.1)
   ```

Before running the next stage, you must deploy a /srv/pillar/ceph/proposals/policy.cfg file, which maps various cluster functions to the nodes in your cluster. Examples are located in

`/usr/share/doc/packages/deepsea/examples/`. The role-based example can be copied as a starting point:

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

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

# Monitoring
role-prometheus/cluster/admin*.sls
role-grafana/cluster/admin*.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/igw*.sls

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

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

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

# Storage
role-storage/cluster/data*.sls
```

3. With the `policy.cfg` in place, prepare the Ceph cluster configuration by running the following:

   ```
   $ sudo deepsea stage run ceph.stage.configure (or deepsea stage run ceph.stage.2
   ```

4. To deploy the Ceph cluster:

   ```
   $ sudo deepsea stage run ceph.stage.deploy (or deepsea stage run ceph.stage.3)
   ```

   This command will take several minutes. After the command completes, check the status with:

   ```
   $ sudo ceph -s
   ```

5. To instantiate Ceph services, such as iSCSI, the CephFS filesystem, the RADOS gateway, and the Ceph Dashboard, run the services stage:

   ```
   $ sudo deepsea stage run ceph.stage.services (or deepsea stage run ceph.stage.4)
   ```

   The goal of DeepSea is to let the user define the configuration in advance, through the use of the policy.cfg file and other configuration files, to allow for a simplified and more automated rollout. Talk to the support team at SUSE for more on configuring and deploying Ceph with DeepSea.

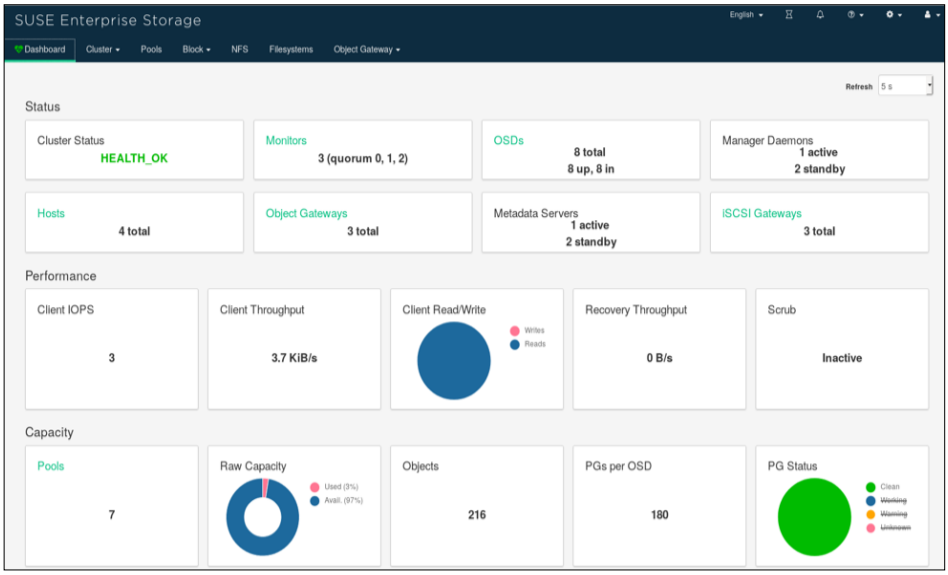

Figure 5. *The Ceph Dashboard.*

Managing

Your SUSE Enterprise Storage cluster will perform many of its replication, rebalancing and storage tasks automatically, but you'll still need a way to perform everyday management and monitoring tasks. SUSE Enterprise Storage offers several tools to help you.

Beginning with SUSE Enterprise Storage 6, users can take advantage of the Ceph Dashboard, a browser-based GUI that expands the capabilities and grew out of SUSE's work on openATTIC. Ceph Dashboard is the preferred way to manage your Ceph cluster, providing you with real-time information about your nodes, OSDs, pools, gateways and more through Grafana dashboards.

The Ceph Dashboard provides additional options for adding and managing OSDs, interfaces, and pools, and even offers a convenient interface for viewing and managing the CRUSH map.

You can also use the command line on the Salt master to manage your cluster, but the idea is to avoid logging into any individual node for routine tasks and using the Ceph Dashboard instead.

Some basic CLI commands include:

`$ sudo ceph -s` -- outputs a summary of health statistics, such as how many OSD and monitor nodes are up.

`$ sudo ceph df` -- checks the cluster's data usage and the data distribution among pools:

`$ ceph -w` -- watch ongoing events as they unfold (a sort of tail -f for ceph -s):

Conclusion

SUSE Enterprise Storage is a full-featured storage clustering solution built around Ceph software-defined storage technology. SUSE Enterprise Storage builds on the power of Ceph and adds:

- *Enterprise-grade testing and hardware certification*
- *Easy, GUI-based cluster management with the Ceph Dashboard*
- *Strong support for OpenStack and Kubernetes*
- *Expert customer service*

A collection of useful admin tools makes it easy to deploy and manage your SUSE Enterprise Storage cluster.

If you're looking for a highly scalable, self-healing, fault-tolerant storage solution that offers high performance with minimal administrative overhead, contact SUSE for more information on how you can get started with deploying a SUSE Enterprise Storage cluster.

Resources

- **The SUSE Enterprise Storage website**
- **The Deployment Guide**
- **How to deploy a SUSE Enterprise Storage test environment in about 30 minutes**
- **Managing Ceph with the Ceph Dashboard**

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

www.suse.com