Managing Shared File Storage

Objectives

After completing this section, you should be able to configure CephFS, including snapshots, replication, memory management, and client access.

CephFS Administration

Use the following commands to manage CephFS file systems:

Action	Command
Create a file system.	ceph fs new *fs-name* *meta-pool* *data-pool*
List existing file systems.	ceph fs ls
Remove a file system.	ceph fs rm fs-name [--yes-i-really-mean-it]
Force MDS to fail status.	ceph mds fail *gid/name/role*
Declare an MDS to be repaired, triggering a failback.	ceph mds repaired role

CephFS provides tools to inspect and repair MDS journals (`cephfs-journal-tool`) or MDS tables (`cephfs-table-tool`), and to inspect and rebuild metadata (`cephfs-data-scan`).

Mapping a File to an Object

For troubleshooting, it is useful to determine which OSDs store a file's objects. Directories or zero-length files might have any associated objects in a data pool.

This example retrieves object mapping information for a file within Ceph:

- Retrieve the inode number for the file.

```
[ceph: root@server /]# stat -c %i filepath
1099511627776
```

- Convert the inode number to hexadecimal. Use the %x formatting output of the `printf` command.

```
[ceph: root@server /]# printf '%x\n' 1099511627776
10000000000
```

This example combines these first two steps:

```
[ceph: root@server /]# printf '%x\n' $(stat -c %i filepath)
```