requests a ticket from a Monitor to authenticate to cluster daemons. This is similar to the Kerberos protocol, with a cephx key-ring file being comparable to a Kerberos keytab file.

A more detailed discussion of the protocol is available from the upstream Ceph project's documentation at High Availability Authentication [https://docs.ceph.com/docs/master/architecture/#high-availability-authentication].

Configuring User Authentication

Using command-line tools such as `ceph`, `rados`, and `rbd`, administrators can specify the user account and the key-ring file by using the `--id` and `--keyring` options. When not specified, commands authenticate as the `client.admin` user.

In this example, the `ceph` command authenticates as `client.operator3` to list the available pools:

```
[ceph: root@node /]# ceph --id operator3 osd lspools
1 myfirstpool
2 mysecondpool
```

Important

Do not include the `client.` prefix when using the `--id` option. The `--id` option automatically assumes that `client.` prefix. Alternatively, the `--name` option requires the `client.` prefix.

If you store the key-ring file in its default location, you do not need the `--keyring` option. The `cephadm` shell automatically mounts the key-ring from the `/etc/ceph/` directory.

Configuring User Authorization

When you create a new user account, grant cluster permissions sufficient to authorize the user's cluster tasks. Permissions within `cephx` are known as *capabilities*, and you grant them by daemon type (`mon`, `osd`, `mgr`, or `mds`.)

Use capabilities to restrict or provide access to data in a pool, a pool's namespace, or a set of pools based on application tags. Capabilities also allow the daemons in the cluster to interact with each other.

Cephx Capabilities

Within `cephx`, for each daemon type, several capabilities are available:

- `r` grants read access. Each user account should have at least read access on the Monitors to be able to retrieve the CRUSH map.
- `w` grants write access. Clients need write access to store and modify objects on OSDs. For Managers (MGRs), `w` grants the right to enable or disable modules.
- `x` grants authorization to execute extended object classes. This allows clients to perform extra operations on objects such as setting locks with `rados lock get` or listing RBD images with `rbd list`.
- `*` grants full access.
- `class-read` and `class-write` are subsets of `x`. You typically use them on RBD pools.