```
        caps: [mon] allow profile osd
        caps: [osd] allow *
client.admin
        key: AQCi6Dhajw7pIRAA/ECkwyipx2/raLWjgbklyA==
        caps: [mds] allow *
        caps: [mgr] allow *
        caps: [mon] allow *
        caps: [osd] allow *
...output omitted...
```

To get the details of a specific account, use the `ceph auth get` command:

```
[ceph: root@node /]# ceph auth get client.admin
exported keyring for client.admin
[client.admin]
        key = AQCi6Dhajw7pIRAA/ECkwyipx2/raLWjgbklyA==
        caps mds = "allow *"
        caps mgr = "allow *"
        caps mon = "allow *"
        caps osd = "allow *"
```

You can print the secret key:

```
[ceph: root@node /]# ceph auth print-key client.admin
AQCi6Dhajw7pIRAA/ECkwyipx2/raLWjgbklyA==
```

To export and import user accounts, run the `ceph auth export` and `ceph auth import` commands:

```
[ceph: root@node /]# ceph auth export client.operator1 > ~/operator1.export
[ceph: root@node /]# ceph auth import -i ~/operator1.export
```

Creating New User Accounts

The `ceph auth get-or-create` command creates a new user account and generates its secret key. The command prints this key to `stdout` by default, so it is common to add the `-o` option to save standard output to a key-ring file.

This example creates the `app1` user account with read and write access to all pools, and stores the key-ring file in `/etc/ceph/ceph.client.app1.keyring`:

```
[ceph: root@node /]# ceph auth get-or-create client.app1 \
 mon 'allow r' osd 'allow rw' -o /etc/ceph/ceph.client.app1.keyring
```

Authentication requires the key-ring file, so you must copy the file to all client systems that operate with this new user account.

Modifying User Capabilities

Modify the capabilities of a user account with the `ceph auth caps` command.

版权禁止外传