

Keystone (Image Services)

List all users

```
keystone user-list
```

List Identity Service catalog

```
keystone endpoint-list
```

Discover keystone endpoints

```
keystone discover
```

List all services in Service Catalog

```
keystone service-list
```

Create new user:

```
keystone user-create --name <user-name> --tenant-id <tenant> --pass <pass> --email <email> --enabled <true>
```

Create new tenant:

```
keystone tenant-create --name <tenant-name> --description <tenant-description> --enabled <true>
```

Nova (All things Compute)

List instances, notice status of instance

```
nova list
```

List images

```
nova image-list
```

List flavors

```
nova flavor-list
```

Boot an instance using flavor and image names (if names are unique)

```
nova boot --image <image name or id> --flavor <flavor name or id> <instance name>
```

```
nova boot --image cirros-0.3.1-x86_64-uec-ramdisk --flavor m1.tiny MyFirstInstance
```

Login to instance

```
ip netns
```

```
sudo ip netns exec <network namespace, from above command> ssh <user@server or use a key>
```

```
sudo ip netns exec qdhcp-6021a3b4-8587-4f9c-8064-0103885dfba2 ssh cirros@10.0.0.2
```

if you are on devstack, password is "cubswin:)" without the quotes

Show details of instance

```
nova show <server name or id>
```

```
nova show MyFirstInstance
```

View console log of instance

```
nova console-log MyFirstInstance
```

Pause, suspend, stop, rescue, resize, rebuild, reboot an instance

pause:

```
nova pause <name>
```

```
nova pause volumeTwoImage
```

unpause:

```
nova unpause <name>
```

```

    # suspend:
nova suspend <name>

    # unsuspend:
nova resume <name>

    # stop:
nova stop <name>

    # start:
nova start <name>

    # rescue:
nova rescue <name>

    # resize:
nova resize <name> <flavor>
nova resize my-pem-server m1.small
nova resize-confirm server1

    # rebuild:
nova rebuild <name> <image>
nova rebuild newtinny cirros-qcow2

    # reboot:
nova reboot <name>
nova reboot newtinny

    # Inject user data and files into an instance:
nova boot --user-data ./userdata.txt MyUserDataInstance
nova boot --user-data userdata.txt--image cirros-qcow2 --flavor m1.tiny MyUserDataInstance2

    #--to validate file is there, ssh into instance, go to /var/lib/cloud look for file...

# Inject a keypair into an instance and access the instance with that keypair
    # create keypair:
nova keypair-add test > test.pem
chmod 600 test.pem

    #boot:
nova boot --image cirros-0.3.1-x86_64 --flavor m1.small --key_name test my-first-server

    #--ssh:
sudo ip netns exec qdhcp-98f09f1e-64c4-4301-a897-5067ee6d544f ssh -i test.pem cirros@10.0.0.4

    # Set metadata on an instance
nova meta volumeTwoImage set newmeta='super duper data'

    # Create an instance snapshot
nova image-create volumeTwoImage snapshotOfVolumeImage
nova image-show snapshotOfVolumeImage

# Manage security groups
    # Add rules to default security group allowing ping and ssh between #instances in the default security group
nova secgroup-add-group-rule default default icmp -1 -1
nova secgroup-add-group-rule default default tcp 22 22

```

Glance (Image Service)

List images you can access

```
glance image-list
```

Delete specified image

```
glance image-delete <image>
```

Describe a specific image

```
glance image-show <image>
```

update image

```
glance image-update <image>
```

#Manage images

Kernel image:

```
glance image-create --name "cirros-threepart-kernel" --disk-format aki --container-format aki --is-public True --file ~/images/cirros-0.3.1-pre4-x86_64-vmlinuz
```

Ram image:

```
glance image-create --name "cirros-threepart-ramdisk" --disk-format ari --container-format ari --is-public True --file ~/images/cirros-0.3.1-pre4-x86_64-initrd
```

3-part image:

```
glance image-create --name "cirros-threepart" --disk-format ami --container-format ami --is-public True --property kernel_id=$KID--property ramdisk_id=$RID--file ~/images/cirros-0.3.1-pre4-x86_64-blank.img
```

Register raw image:

```
glance image-create --name "cirros-qcow2" --disk-format qcow2 --container-format bare --is-public True --file ~/images/cirros-0.3.1-pre4-x86_64-disk.img
```

Neutron (Networking)

create network

```
neutron net-create <name>
```

```
neutron net-create my-network
```

create a subnet

```
neutron subnet-create <network name> <cidr>
```

```
neutron subnet-create my-network 10.0.0.0/29
```

List network and subnet

```
neutron net-list
```

```
neutron subnet-list
```

Examine details of network and subnet

```
neutron net-show <id or name of network>
```

```
neutron subnet-show <id or name of subnet>
```

Cinder (Block Storage)

Manage volumes and volume snapshots

Create a new volume

```
cinder create <size in GB> --display-name <volume name>
```

```
cinder create 1 --display-name MyFirstVolume
```

boot an instance and attach to volume

```
nova boot --image cirros-qcow2 --flavor m1.tiny MyVolumeInstance
```

List volumes, notice status of volume

```
cinder list
```

```

    # Attach volume to instance after instance is active, and volume is available
nova volume-attach <instance-id> <volume-id> auto
nova volume-attach MyVolumeInstance /dev/vdb auto

    # login into instance, list storage devices
sudo fdisk -l

    # on the instance, make file system on volume
sudo mkfs.ext3 /dev/vdb

    # Create a mountpoint
sudo mkdir /myspace

    # Mount the volume at the mountpoint
sudo mount /dev/vdb /myspace

    # Create a file on the volume
sudo touch /myspace/helloworld.txt
sudo ls /myspace

    # Unmount the volume
sudo umount /myspace

```

Swift (Object Store)

```

    # Displays information for the account, container, or object
swift stat
swift stat <account>
swift stat <container>
swift stat <object>

    # list containers:
swift list

    # create a container
swift post mycontainer

    # upload file to a container
swift upload <container name> <file name>
swift upload mycontainer myfile.txt

    # list objects in container
swift list container

    # download object from container
swift download <container name> <file name>

    # upload with chunks, for large file:
swift upload -S <size> <container name> <file name>
swift upload -S 64 container largeFile

```