



OpenStack Labs

Lab 09: Customizing Instances

Contents

Introduction	2
Objectives	3
Lab Settings.....	4
1 Creating Customized Instances	5
2 Verify Customized Instances	24

Introduction

In this lab, you will use the `cloud-init` utility to customize OpenStack instances.

Objectives

- Customize an instance with `cloud-init`.
- Verify instance customization.

Lab Settings

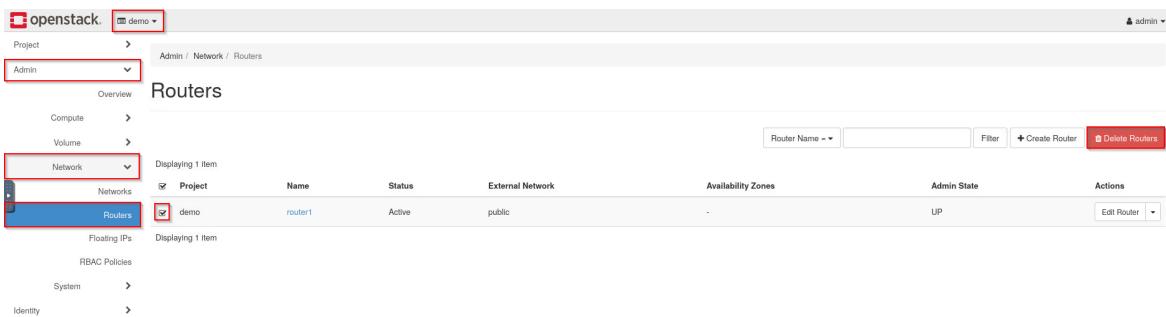
The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account	Password
workstation	ens3: 192.168.1.21 ens4: 172.25.250.21	ubuntu	ubuntu
devstack	ens3: 192.168.1.20 ens4: 172.25.250.20	ubuntu	ubuntu

1 Creating Customized Instances

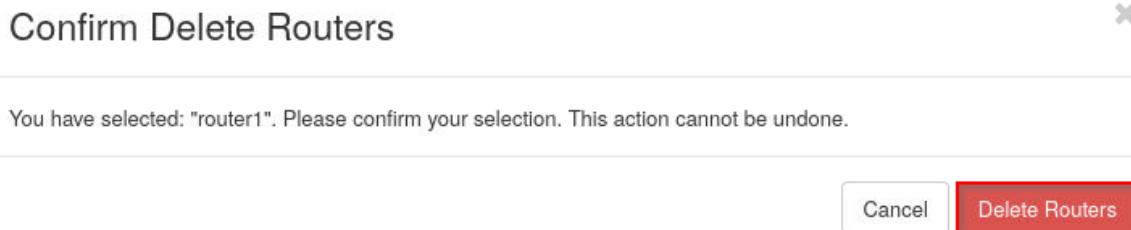
In this task, you will customize two instances using cloud-init capabilities and features. You will log into the first instance to confirm cloud-init is up and running.

1. Open the web browser and navigate to **192.168.1.20**. Log into the dashboard as **admin** with the password **secret**.
2. Switch to the **demo** project. Navigate to **Admin > Network > Routers**. Check the box in the same row as **router1**, then click **Delete Routers**.

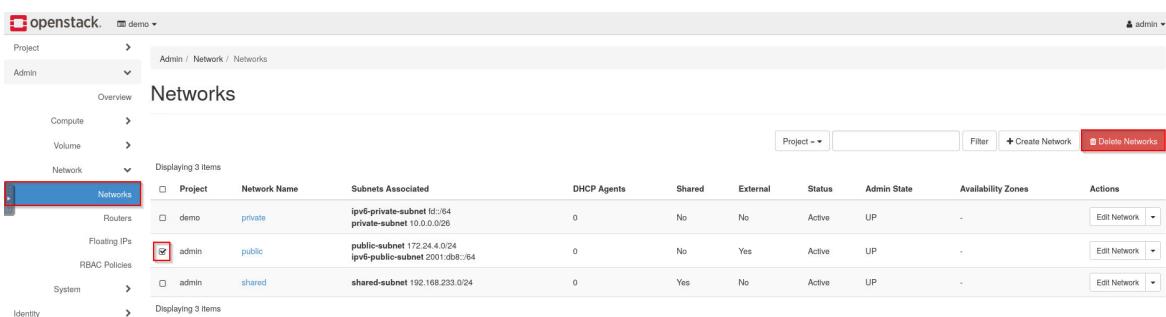


Project	Name	Status	External Network	Availability Zones	Admin State	Actions
<input checked="" type="checkbox"/> demo	router1	Active	public	-	UP	Edit Router

3. In the *Confirm Delete Routers* dialog box that pops up, click **Delete Routers**.



4. Now, navigate to **Networks**. Check the box in the same row as **public**, then click **Delete Networks**.



Project	Network Name	Subnets Associated	DHCP Agents	Shared	External	Status	Admin State	Availability Zones	Actions
<input type="checkbox"/> demo	private	ipv6-private-subnet fd::/64 private-subnet 10.0.0.0/26	0	No	No	Active	UP	-	Edit Network
<input checked="" type="checkbox"/> admin	public	public-subnet 172.24.4.0/24 ipv6-public-subnet 2001:db8::/64	0	No	Yes	Active	UP	-	Edit Network
<input type="checkbox"/> admin	shared	shared-subnet 192.168.233.0/24	0	Yes	No	Active	UP	-	Edit Network

5. In the *Confirm Delete Networks* dialog box that pops up, click **Delete Networks**.

Confirm Delete Networks



You have selected: "public". Please confirm your selection. This action cannot be undone.

[Cancel](#)

[Delete Networks](#)

6. Leave the web browser open and open a terminal window. Source the keystone credentials for the **admin** user.

```
ubuntu@workstation:~$ source ~/keystonerc-admin
```

```
ubuntu@workstation:~$ source ~/keystonerc-admin
ubuntu@workstation:~$ █
```

7. A few resources need to be created to help with customizing the instances. First, create an external network named **external**. Set the network type to **flat** and the physical network to **public**. Set the network as shared and external.

```
ubuntu@workstation:~$ openstack network create external \
> --external --share \
> --provider-network-type flat \
> --provider-physical-network public
```

```
ubuntu@workstation:~$ openstack network create external \
> --external --share \
> --provider-network-type flat \
> --provider-physical-network public
+-----+-----+
| Field | Value |
+-----+-----+
| admin_state_up | UP |
| availability_zone_hints | |
| availability_zones | |
| created_at | 2023-11-28T22:04:20Z |
| description | |
| dns_domain | None |
| id | 9413be91-6d77-4bd3-a206-d4b0d51e107a |
| ipv4_address_scope | None |
| ipv6_address_scope | None |
| is_default | False |
| is_vlan_transparent | None |
| mtu | 1500 |
| name | external |
| port_security_enabled | True |
| project_id | c524eaead1f74d4f9141f71b280e0237 |
| provider:network_type | flat |
| provider:physical_network | public |
| provider:segmentation_id | None |
| qos_policy_id | None |
| revision_number | 1 |
| router:external | External |
| segments | None |
| shared | True |
| status | ACTIVE |
| subnets | |
| tags | |
| updated_at | 2023-11-28T22:04:20Z |
+-----+-----+
ubuntu@workstation:~$ █
```

Tip

When typing the command, make sure there is a space between **external** and the **** character, and press **Enter** to get the **>** and continue typing the rest of the command.

8. Create a subnet named **subext** in the **external** network. Give the subnet a range of **172.25.250.60** to **172.25.250.80**. Disable DHCP services for the subnet and use the address **172.25.250.254** as the gateway as well as the DNS name server.

```
ubuntu@workstation:~$ openstack subnet create \
> --subnet-range 172.25.250.0/24 \
> --no-dhcp \
> --gateway 172.25.250.254 \
> --dns-nameserver 172.25.250.254 \
> --allocation-pool start=172.25.250.60,end=172.25.250.80 \
> --network external \
> subext
```

```
ubuntu@workstation:~$ openstack subnet create \
> --subnet-range 172.25.250.0/24 \
> --no-dhcp \
> --gateway 172.25.250.254 \
> --dns-nameserver 172.25.250.254 \
> --allocation-pool start=172.25.250.60,end=172.25.250.80 \
> --network external \
> subext
+-----+-----+
| Field | Value
+-----+-----+
| allocation_pools | 172.25.250.60-172.25.250.80
| cidr | 172.25.250.0/24
| created_at | 2023-11-28T22:06:04Z
| description | 
| dns_nameservers | 172.25.250.254
| enable_dhcp | False
| gateway_ip | 172.25.250.254
| host_routes | 
| id | 1fc4acf4-704b-4a61-bbf3-cc8db40497ec
| ip_version | 4
| ipv6_address_mode | None
| ipv6_ra_mode | None
| name | subext
| network_id | 9413be91-6d77-4bd3-a206-d4b0d51e107a
| project_id | c524eaead1f74d4f9141f71b280e0237
| revision_number | 0
| segment_id | None
| service_types | 
| subnetpool_id | None
| tags | 
| updated_at | 2023-11-28T22:06:04Z
+-----+-----+
ubuntu@workstation:~$
```

9. From the floating IP pool in the **external** network, create a floating IP.

```
ubuntu@workstation:~$ openstack floating ip create external
```

```
ubuntu@workstation:~$ openstack floating ip create external
+-----+-----+
| Field | Value |
+-----+-----+
| created_at | 2023-11-28T22:06:29Z |
| description | None |
| fixed_ip_address | 172.25.250.78 |
| floating_ip_address | 9413be91-6d77-4bd3-a206-d4b0d51e107a |
| id | dea3c2c7-fd39-4d9a-9835-2444cf2dfa7e |
| name | 172.25.250.78 |
| port_id | None |
| project_id | c524eaead1f74d4f9141f71b280e0237 |
| qos_policy_id | None |
| revision_number | 0 |
| router_id | None |
| status | DOWN |
| subnet_id | None |
| updated_at | 2023-11-28T22:06:29Z |
+-----+
ubuntu@workstation:~$
```

10. Create a router named **exercise-router**.

```
ubuntu@workstation:~$ openstack router create exercise-router
```

```
ubuntu@workstation:~$ openstack router create exercise-router
+-----+-----+
| Field | Value |
+-----+-----+
| admin_state_up | UP |
| availability_zone_hints | |
| availability_zones | |
| created_at | 2023-11-29T19:33:59Z |
| description | |
| distributed | False |
| external_gateway_info | None |
| flavor_id | None |
| ha | False |
| id | c8ed3b3a-f70c-418e-a00f-1a8f8dca6161 |
| name | exercise-router |
| project_id | c524eaead1f74d4f9141f71b280e0237 |
| revision_number | 1 |
| routes | |
| status | ACTIVE |
| tags | |
| updated_at | 2023-11-29T19:33:59Z |
+-----+
ubuntu@workstation:~$
```

11. Connect the router to the **shared-subnet** subnet.

```
ubuntu@workstation:~$ openstack router add subnet \
> exercise-router shared-subnet
```

```
ubuntu@workstation:~$ openstack router add subnet \
> exercise-router shared-subnet
ubuntu@workstation:~$ █
```

12. Set the **external** network as the gateway for the router.

```
ubuntu@workstation:~$ openstack router set \
> --external-gateway external \
> exercise-router
```

```
ubuntu@workstation:~$ openstack router set \
> --external-gateway external \
> exercise-router
ubuntu@workstation:~$ █
```

13. Create the key pair **dev-keypair** and save the private key to the file **~/Downloads/dev-keypair.pem**.

```
ubuntu@workstation:~$ openstack keypair create \
> dev-keypair > ~/Downloads/dev-keypair.pem
```

```
ubuntu@workstation:~$ openstack keypair create \
> dev-keypair > ~/Downloads/dev-keypair.pem
ubuntu@workstation:~$ █
```

14. the **chmod** command with a mode of **600** to make it so that the **ubuntu** user has read/write permissions on the file, and groups and other users have no permissions to the file.

```
ubuntu@workstation:~$ chmod 600 ~/Downloads/dev-keypair.pem
```

```
ubuntu@workstation:~$ chmod 600 ~/Downloads/dev-keypair.pem
ubuntu@workstation:~$ █
```

15. Create the **dev-secgroup** security group.

```
ubuntu@workstation:~$ openstack security group \
> create dev-secgroup
```

```
ubuntu@workstation:~$ openstack security group \
> create dev-secgroup
+-----+
| Field      | Value
+-----+
| created_at | 2023-11-28T22:07:57Z
| description | dev-secgroup
| id          | bf6127bb-edab-44c8-a470-20cc9c3c4f6c
| name        | dev-secgroup
| project_id  | c524eaead1f74d4f9141f71b280e0237
| revision_number | 1
| rules       | [ { "created_at": "2023-11-28T22:07:57Z", "direction": "egress", "ether_type": "IPv6", "id": "93666585-0b16-4dee-9afc-797bd77935a9", "standard_attr_id": "61", "updated_at": "2023-11-28T22:07:57Z" }, { "created_at": "2023-11-28T22:07:57Z", "direction": "egress", "ether_type": "IPv4", "id": "ccad6958-5c4d-476e-a90b-26d06e68de50", "standard_attr_id": "62", "updated_at": "2023-11-28T22:07:57Z" } ]
| updated_at  | 2023-11-28T22:07:57Z
+-----+
ubuntu@workstation:~$
```

16. Add a security rule in the **dev-secgroup** security group to allow remote ICMP traffic.

```
ubuntu@workstation:~$ openstack security group \
> rule create \
> --protocol icmp \
> dev-secgroup
```

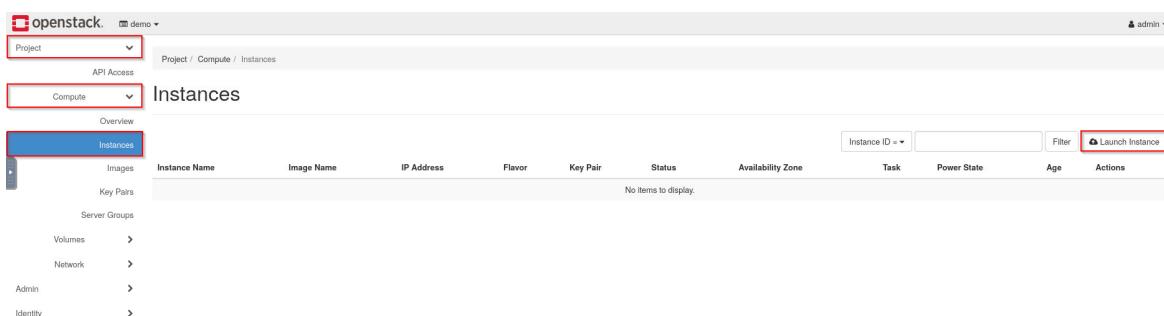
```
ubuntu@workstation:~$ openstack security group \
> rule create \
> --protocol icmp \
> dev-secgroup
+-----+
| Field      | Value
+-----+
| created_at | 2023-11-28T22:09:51Z
| description |
| direction   | ingress
| ether_type  | IPv4
| id          | d38364cb-c1ba-4710-8561-3243de310569
| name        | None
| port_range_max | None
| port_range_min | None
| project_id  | c524eaead1f74d4f9141f71b280e0237
| protocol    | icmp
| remote_group_id | None
| remote_ip_prefix | 0.0.0.0/0
| revision_number | 0
| security_group_id | bf6127bb-edab-44c8-a470-20cc9c3c4f6c
| updated_at   | 2023-11-28T22:09:51Z
+-----+
ubuntu@workstation:~$
```

17. Add another security rule to allow remote connection using SSH on the default port 22.

```
ubuntu@workstation:~$ openstack security group \
> rule create \
> --protocol tcp \
> --dst-port 22 \
> dev-secgroup
```

```
ubuntu@workstation:~$ openstack security group \
> rule create \
> --protocol tcp \
> --dst-port 22 \
> dev-secgroup
+-----+-----+
| Field | Value
+-----+-----+
| created_at | 2023-11-28T22:10:34Z
| description | ingress
| direction | ingress
| ether_type | IPv4
| id | ae97b392-7583-4f00-8cde-c544fd4b8195
| name | None
| port_range_max | 22
| port_range_min | 22
| project_id | c524eaead1f74d4f9141f71b280e0237
| protocol | tcp
| remote_group_id | None
| remote_ip_prefix | 0.0.0.0/0
| revision_number | 0
| security_group_id | bf6127bb-edab-44c8-a470-20cc9c3c4f6c
| updated_at | 2023-11-28T22:10:34Z
+-----+
ubuntu@workstation:~$
```

18. Now that the necessary resources have been created, focus back to the web browser. Navigate to **Project > Compute > Instances**, then click **Launch Instance**.



19. In the *Details* tab, enter **instance1** in the *Instance Name* field and click **Next**.

Launch Instance

Details

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Source *	Project Name	Total Instances (10 Max)
Flavor *	demo	10%
Networks *	Instance Name *	0 Current Usage 1 Added 9 Remaining
Network Ports	instance1	
Security Groups	Description	
Key Pair	Availability Zone	
Configuration	nova	
Server Groups	Count *	
Scheduler Hints	1	
Metadata		

Cancel **Next >** **Launch Instance**

20. In the **Source** tab, make sure **Image** is selected in the *Select Boot Source* dropdown and click **No** under *Create New Volume*. Select the **ubuntu** image by clicking the \uparrow symbol in the same row. Click **Next**.

Launch Instance

Details	Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.				
Source *	Select Boot Source <input type="button" value="Image"/> <input type="button" value="Volume"/> <input type="button" value="Snapshot"/> <div style="float: right;">Create New Volume <input type="button" value="Yes"/> <input checked="" type="button" value="No"/></div>				
Flavor *	<input type="button" value="Image"/>				
Networks *	Allocated Displaying 0 items				
Network Ports	Name	Updated	Size	Format	Visibility
Security Groups	Select an item from Available items below				
Key Pair	Displaying 0 items				
Configuration	Available <small>2</small> <input type="button" value="Select one"/> <div style="margin-top: 10px;"> <input type="button" value="Q"/> Click here for filters or full text search. <input type="button" value="X"/> </div>				
Server Groups					
Scheduler Hints	Displaying 2 items				
Metadata	Name	Updated	Size	Format	Visibility
	» cirros-0.6.2-x86_64-disk	11/8/23 9:23 PM	20.44 MB	QCOW2	Public <input type="button" value="↑"/>
	» ubuntu	11/8/23 10:23 PM	642.75 MB	QCOW2	Public <input checked="" type="button" value="↑"/>
	Displaying 2 items				
<input type="button" value="Cancel"/>		<input type="button" value="Back"/> <input checked="" type="button" value="Next"/>		<input type="button" value="Launch Instance"/>	

Stop

Before proceeding to the next step, confirm that **ubuntu** appears underneath the *Allocated* section.

21. In the *Flavor* tab, click the ↑ symbol in the same row as **m1.small**. Click **Next**.

Launch Instance

Allocated

Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Details

Source

Flavor *

Networks *

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Available 12

Select one

Click here for filters or full text search.

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public	
m1.nano	1	128 MB	1 GB	1 GB	0 GB	Yes	↑
m1.micro	1	192 MB	1 GB	1 GB	0 GB	Yes	↑
cirros256	1	256 MB	1 GB	1 GB	0 GB	Yes	↑
m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes	↑
ds512M	1	512 MB	5 GB	5 GB	0 GB	Yes	↑
ds1G	1	1 GB	10 GB	10 GB	0 GB	Yes	↑
m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes	↑
ds2G	2	2 GB	10 GB	10 GB	0 GB	Yes	↑
m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes	↑
ds4G	4	4 GB	20 GB	20 GB	0 GB	Yes	↑
m1.large	4	8 GB	80 GB	80 GB	0 GB	Yes	↑
m1.xlarge	8	16 GB	160 GB	160 GB	0 GB	Yes	↑

Displaying 12 items

Cancel

< Back

Next >

Launch Instance

Stop

Before proceeding to the next step, confirm that **m1.small** appears underneath the *Allocated* section.

22. In the *Networks* tab, click the ↑ symbol in the same row as **shared**. Click **Next**.

Launch Instance

[Details](#)

[Source](#)

[Flavor](#)

Networks * Selected

[Network Ports](#)

[Security Groups](#)

[Key Pair](#)

[Configuration](#)

[Server Groups](#)

[Scheduler Hints](#)

[Metadata](#)

Allocated

Displaying 0 items

Network	Subnets Associated	Shared	Admin State	Status
Select one or more networks from the available networks below.				

Available 3 Select one or more

Click here for filters or full text search.

Network	Subnets Associated	Shared	Admin State	Status
external	subext	Yes	Up	Active
shared	shared-subnet	Yes	Up	Active
private	ipv6-private-subnet private-subnet	No	Up	Active

Displaying 3 items

[Cancel](#) [Next >](#) [Launch Instance](#)

Stop

Before proceeding to the next step, confirm that **shared** appears underneath the *Allocated* section.

23. In the *Network Ports* tab, click **Next**.

Launch Instance

Details

Ports provide extra communication channels to your instances. You can select ports instead of networks or a mix of both.

Allocated

Displaying 0 items

Name	IP	Admin State	Status
Select one or more ports from the available ports below.			

Network Ports

Displaying 0 items

Available 0

Select one or more

Click here for filters or full text search.

Key Pair

Configuration

Displaying 0 items

Server Groups

Name	IP	Admin State	Status
No items to display.			

Scheduler Hints

Displaying 0 items

Metadata

Displaying 0 items

< Back **Next >** **Launch Instance**

24. In the *Security Groups* tab, click the ↓ symbol in the same row as **default**, and click the ↑ symbol in the same row as **dev-secgroup**. Click **Next**.

Launch Instance

Details

Select the security groups to launch the instance in.

Allocated 1

Displaying 1 item

Name	Description
default	Default security group

Available 1

Select one or more

Click here for filters or full text search.

Key Pair

Configuration

Displaying 1 item

Server Groups

Name	Description
dev-secgroup	dev-secgroup

Scheduler Hints

Displaying 1 item

Metadata

Displaying 1 item

< Back **Next >** **Launch Instance**

Stop

Before proceeding to the next step, confirm that only **dev-secgroup** appears underneath the *Allocated* section.

25. In the *Key Pair* tab, ensure that the key pair **dev-keypair** has been selected and is underneath the *Allocated* section. Click **Next**.

Launch Instance

Key Pair

Name	Type	Fingerprint
dev-keypair	ssh	bc:c6:93:d6:a8:71:08:bc:9d:e1:74:6e:e8:8f:b5:2b

Available 0 Select one

No items to display.

< Back **Next >** Launch Instance

26. In the *Configuration* tab, populate the **Customization Script** field with the content below. Once finished, click **Launch Instance**.

```
#!/bin/bash
echo 'Hello, world!' > /root/hello.txt
```

Launch Instance

Details

You can customize your instance after it has launched using the options available here. "Customization Script" is analogous to "User Data" in other systems.

Source

Load Customization Script from a file
Browse... No file selected.

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration Configuration

Disk Partition

Automatic

Configuration Drive

Metadata

Cancel **Back** **Next >** **Launch Instance**

Tip

A customization script can be used to perform many commands automatically upon instance creation, such as installing packages, configuring a host name, etc. The simple script above is just an example.

- Once the status for **instance1** is **Active**, attach a floating IP address to it. Select **Associate Floating IP** from the dropdown menu next to **Create Snapshot** in the row for the instance.

Project / Compute / Instances

Instances

Instance ID	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
<input type="checkbox"/> instance1	ubuntu	192.168.233.5	m1.small	dev-keypair	Active	■	nova	None	Running	0 minutes	Create Snapshot Associate Floating IP Attach Interface Detach Interface Edit Instance

- Select any one of the IP addresses from the **IP Address** dropdown and select **instance1: 192.168.233.XYZ** as the **Port to be associated**. Click **Associate**.

Manage Floating IP Associations

IP Address *

+

Select the IP address you wish to associate with the selected instance or port.

Port to be associated *

Cancel
Associate

29. To verify that the customization script worked, first click on **instance1** under the *Instance Name* column, then navigate to the *Console* tab if you are not directed there automatically. Log into the instance as **root** with the password **secret**.

```
Connected to QEMU (Instance-00000009)

Ubuntu 22.04.3 LTS instance1 tty1
instance1 login: root
Password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Thu Nov 30 17:42:55 UTC 2023

 System load:  0.0927734375      Processes:          81
 Usage of /:   15.0% of 9.51GB   Users logged in:     0
 Memory usage: 17%                IPv4 address for ens3: 192.168.233.250
 Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed Nov 29 22:05:32 UTC 2023 on tty1
root@instance1:~#
```

30. Check **/var/log/cloud-init.log** to confirm that **cloud-init** ran. Use the tail command to print the last 10 lines of the log.

```
root@instance1:~# tail /var/log/cloud-init.log
```

```
Connected to QEMU (instance-00000009)

root@instance1:~# sudo tail /var/log/cloud-init.log
sudo: unable to resolve host instance1: Temporary failure in name resolution
2023-11-30 17:32:30,843 - util.py[DEBUG]: Writing to /var/lib/cloud/instance/boot-finished - wb: [644] 70 bytes
2023-11-30 17:32:30,852 - handlers.py[DEBUG]: finish: modules-final/config-final_message: SUCCESS: config-final_message ran successfully
2023-11-30 17:32:30,853 - main.py[DEBUG]: Ran 11 modules with 0 failures
2023-11-30 17:32:30,860 - atomic_helper.py[DEBUG]: Atomically writing to file /var/lib/cloud/data/status.json (via temporary file /var/lib/cloud/data/tmppufile6z_) - w: [644] 590 bytes/chars
2023-11-30 17:32:30,867 - atomic_helper.py[DEBUG]: Atomically writing to file /var/lib/cloud/data/result.json (via temporary file /var/lib/cloud/data/tmpwymssadmy) - w: [644] 87 bytes/chars
2023-11-30 17:32:30,870 - util.py[DEBUG]: Creating symbolic link from '/run/cloud-init/result.json' => '../../../../../var/lib/cloud/data/result.json'
2023-11-30 17:32:30,872 - util.py[DEBUG]: Reading from /proc/uptime (quiet=False)
2023-11-30 17:32:30,875 - util.py[DEBUG]: Read 12 bytes from /proc/uptime
2023-11-30 17:32:30,877 - util.py[DEBUG]: cloud-init mode 'modules' took 3.172 seconds (3.17)
2023-11-30 17:32:30,878 - handlers.py[DEBUG]: finish: modules-final: SUCCESS: running modules for final
root@instance1:~#
```

31. Ensure that the **/root/hello.txt** file exists and has the correct content.

```
root@instance1:~# cat /root/hello.txt
```

```
Connected to QEMU (instance-0000000b)

root@instance1:~# cat /root/hello.txt
Hello, world!
root@instance1:~#
```

32. Log out of the *Horizon Dashboard* and close the web browser.

33. Focus back on the terminal and delete **instance1**.

```
ubuntu@workstation:~$ openstack server delete instance1
```

```
ubuntu@workstation:~$ openstack server delete instance1
ubuntu@workstation:~$ █
```

34. Another instance will be created and customized using the *OpenStack Unified CLI*. First, create a **user-data** script that will be attached to the instance at creation. Create a script called **~/hello** that matches the content shown below. Press **CTRL+X**, then **Y** to accept the file changes. Press **Enter** to confirm and exit back to the terminal.

```
ubuntu@workstation: nano ~/hello
```

```
#!/bin/bash
echo 'Hello, world!' > /root/hello.txt
```

GNU nano 2.9.3	hello	Modified
<code>#!/bin/bash</code>		
<code>echo 'Hello, world!' > /root/hello.txt</code>		

35. Launch an instance using the **user-data** option with the previously created script to perform the customization. Use the **ubuntu** image, the **m1.small** flavor, the **shared** network, the **dev-secgroup** security group, and the **dev-keypair** key pair.

```
ubuntu@workstation: openstack server create \
> --image ubuntu \
> --flavor m1.small \
> --nic net-id=shared \
> --security-group dev-secgroup \
> --key-name dev-keypair \
> --user-data ~/hello \
> --wait instance2
```

```
ubuntu@workstation:~$ openstack server create \
> --image ubuntu \
> --flavor m1.small \
> --nic net-id=shared \
> --security-group dev-secgroup \
> --key-name dev-keypair \
> --user-data ~/hello \
> --wait instance2

+-----+-----+
| Field | Value |
+-----+-----+
| OS-DCF:diskConfig | MANUAL |
| OS-EXT-AZ:availability_zone | nova |
| OS-EXT-SRV-ATTR:host | devstack |
| OS-EXT-SRV-ATTR:hypervisor_hostname | devstack |
| OS-EXT-SRV-ATTR:instance_name | instance-00000003 |
| OS-EXT-STS:power_state | Running |
| OS-EXT-STS:task_state | None |
| OS-EXT-STS:vm_state | active |
| OS-SRV-USG:launched_at | 2024-01-14T19:29:56.000000 |
| OS-SRV-USG:terminated_at | None |
| accessIPv4 | |
| accessIPv6 | |
| addresses | shared=192.168.233.151 |
| adminPass | ysFowX27h4Qi |
| config_drive | |
| created | 2024-01-14T19:29:53Z |
| flavor | m1.small (2) |
| hostId | bc6f228f7747fc0d2d35bd9a38bc7c3c031264c7b383a1ba7ddbf81d |
| id | a63d5880-9189-49c1-96b6-71162969fb07 |
| image | ubuntu (b98a6f63-3af5-49fa-8811-e12816862cbd) |
| key_name | dev-keypair |
| name | instance2 |
| progress | 0 |
| project_id | a96f441f305f48d4a2be02606b29faa8 |
| properties | |
| security_groups | name='dev-secgroup' |
| status | ACTIVE |
| updated | 2024-01-14T19:29:57Z |
| user_id | 2c41850bd89b421eb3aab59b1b4aec8 |
| volumes_attached | |
+-----+-----+
ubuntu@workstation:~$
```

36. Verify that the status of the **instance2** instance is **ACTIVE**.

```
ubuntu@workstation: openstack server list
```

```
ubuntu@workstation:~$ openstack server list
+-----+-----+-----+-----+-----+
| ID | Name | Status | Networks | Image | Flavor |
+-----+-----+-----+-----+-----+
| d178095d-49e4-4e05-9729-0e320cfef309 | instance2 | ACTIVE | shared=192.168.233.47 | ubuntu | m1.small |
+-----+-----+-----+-----+-----+
ubuntu@workstation:~$
```

37. Generate another floating IP address to assign to this instance. Take note of the IP address generated, which is listed in the *name* row in the output from the below command.

```
ubuntu@workstation:~$ openstack floating ip create external
```

```
ubuntu@workstation:~$ openstack floating ip create external
+-----+-----+
| Field | Value |
+-----+-----+
| created_at | 2023-11-29T22:29:11Z |
| description | |
| fixed_ip_address | None |
| floating_ip_address | 172.25.250.63 |
| floating_network_id | 9413be91-6d77-4bd3-a206-d4b0d51e107a |
| id | 61840770-30bb-4bed-83c2-9cb95c7a831f |
| name | 172.25.250.63 |
| port_id | None |
| project_id | c524eaead1f74d4f9141f71b280e0237 |
| qos_policy_id | None |
| revision_number | 0 |
| router_id | None |
| status | DOWN |
| subnet_id | None |
| updated_at | 2023-11-29T22:29:11Z |
+-----+
ubuntu@workstation:~$
```

38. Assign the floating IP generated from the last step to **instance2**.

```
ubuntu@workstation:~$ openstack server add floating ip \
instance2 172.25.250.63
```

```
ubuntu@workstation:~$ openstack server add floating ip \
> instance2 172.25.250.63
ubuntu@workstation:~$
```

Note

The actual value of your floating IP address may be different.

2 Verify Customized Instances

In this task, you will verify that cloud-init has correctly customized the two instances created in the previous section.

1. If a terminal window is not already open, open one and source the admin credentials from the `~/keystonerc-admin` file.
2. Determine the floating IP address associated with **instance2**. Remember that the floating IP address is in the **172.25.250.0/24** subnet.

```
ubuntu@workstation: openstack server show instance2 \
> | grep address
```

```
ubuntu@workstation:~$ openstack server show instance2 \
> | grep address
| addresses
| shared=192.168.233.92, 172.25.250.63
ubuntu@workstation:~$ █
```

Note

The floating IP addresses in your output may differ from these examples.

3. Use the `scp` command to copy the `~/Downloads/dev-keypair.pem` file to the **devstack** machine. You will be prompted to enter the password for **ubuntu@192.168.1.20**. The password can be found on the EZSetup page for the lab by clicking on the **devstack** machine, then

```
ubuntu@workstation:~$ scp ~/Downloads/dev-keypair.pem \
> ubuntu@192.168.1.20:~/dev-keypair.pem
```

```
ubuntu@workstation:~$ scp ~/Downloads/dev-keypair.pem \
> ubuntu@192.168.1.20:~/dev-keypair.pem
ubuntu@192.168.1.20's password:
dev-keypair.pem
100% 1680      2.1MB/s  00:00
ubuntu@workstation:~$ █
```

4. SSH into the **devstack** machine. The password is the same as the last step.

```
ubuntu@workstation:~$ ssh 192.168.1.20
```

```
ubuntu@workstation:~$ ssh 192.168.1.20
ubuntu@192.168.1.20's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-1046-kvm x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.

 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

11 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Wed Nov 29 21:37:36 2023 from 192.168.1.254
ubuntu@devstack:~$ █
```

5. SSH into **instance2** using the **dev-keypair** private key.

```
ubuntu@devstack:~$ ssh -i ~/dev-keypair.pem 172.25.250.63
```

```
ubuntu@devstack:~$ ssh -i ~/dev-keypair.pem 172.25.250.63
The authenticity of host '172.25.250.63 (172.25.250.63)' can't be established.
ED25519 key fingerprint is SHA256:V1eXKEIxCWeUrek+yK7Xm8Dz6DFphsZJKeMGB01j798.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.25.250.63' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-91-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information disabled due to load higher than 1.0

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@instance2:~$
```

Note

It may take several minutes for the instance to fully boot and be available for an SSH connection.

- Check **/var/log/cloud-init.log** to confirm that the **cloud-init** script ran.

```
ubuntu@instance2:~$ sudo tail /var/log/cloud-init.log
```

```
ubuntu@instance2:~$ sudo tail /var/log/cloud-init.log
sudo: unable to resolve host instance2: Temporary failure in name resolution
2024-01-14 19:36:22,239 - util.py[DEBUG]: Writing to /var/lib/cloud/instance/boot-finished - wb: [644] 70 bytes
2024-01-14 19:36:22,252 - handlers.py[DEBUG]: finish: modules-final/config-final_message: SUCCESS: config-final_message ran successfully
2024-01-14 19:36:22,253 - main.py[DEBUG]: Ran 11 modules with 0 failures
2024-01-14 19:36:22,261 - atomic_helper.py[DEBUG]: Atomically writing to file /var/lib/cloud/data/status.json (via temporary file /var/lib/cloud/data/tmponx5ch9u) - w: [644] 591 bytes/chars
2024-01-14 19:36:22,267 - atomic_helper.py[DEBUG]: Atomically writing to file /var/lib/cloud/data/result.json (via temporary file /var/lib/cloud/data/tmp9qkr_8a5) - w: [644] 87 bytes/chars
2024-01-14 19:36:22,270 - util.py[DEBUG]: Creating symbolic link from '/run/cloud-init/result.json' => '.../var/lib/cloud/data/result.json'
2024-01-14 19:36:22,272 - util.py[DEBUG]: Reading from /proc/uptime (quiet=False)
2024-01-14 19:36:22,274 - util.py[DEBUG]: Read 13 bytes from /proc/uptime
2024-01-14 19:36:22,276 - util.py[DEBUG]: cloud-init mode 'modules' took 4.037 seconds (4.03)
2024-01-14 19:36:22,277 - handlers.py[DEBUG]: finish: modules-final: SUCCESS: running modules for final
ubuntu@instance2:~$
```

- Ensure that the **/root/hello.txt** file exists and has the correct content.

```
ubuntu@instance2:~# cat /root/hello.txt
```

```
ubuntu@instance2:~$ sudo cat /root/hello.txt
sudo: unable to resolve host instance2: Temporary failure in name resolution
Hello, world!
ubuntu@instance2:~$ █
```

8. The lab is now complete.