



OpenStack Labs

Lab 12: Deploying an FTP Server

Contents

Introduction	2
Objectives	3
Lab Settings.....	4
1 Launch an FTP Server Instance.....	5

Introduction

In this lab, you will practice and demonstrate the knowledge and skills you acquired throughout the course by deploying an FTP server through OpenStack.

Objectives

- Launch an instance in your OpenStack environment and customize the instance to run an FTP server.
- Access the FTP server from the workstation to confirm the configuration.

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: 192.168.1.23 ens4: 172.25.250.23	ubuntu	ubuntu
devstack	ens3: 192.168.1.22 ens4: 172.25.250.22	ubuntu	ubuntu

1 Launch an FTP Server Instance

In this task, you will deploy an FTP server in your environment. The architecture will be comprised of an external network and an internal network, a new privileged user and a non-privileged user, and a set of new security rules to allow FTP access to the instance. A floating IP will be associated with the instance to permit external connectivity.

1. Log into the **workstation** machine as **ubuntu** with the password **ubuntu**.
2. Open a terminal window and source the `~/keystonerc-admin` keystone credentials file.

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

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

3. Create the **prod** project.

```
ubuntu@workstation:~$ openstack project create prod \
> --domain default
```

```
ubuntu@workstation:~$ openstack project create prod \
> --domain default
+-----+-----+
| Field      | Value                                |
+-----+-----+
| description |                                     |
| domain_id   | default                             |
| enabled     | True                                |
| id          | a72b27c5a34245bbbd1c873a5fd420d4   |
| is_domain   | False                              |
| name        | prod                                |
| options     | {}                                  |
| parent_id   | default                             |
| tags        | []                                  |
+-----+-----+
ubuntu@workstation:~$
```

Tip

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

4. Create a user named **superuser** with the password **secret** to the **prod** project.

```
ubuntu@workstation:~$ openstack user create \  
> --project prod \  
> --password secret \  
> --email ubuntu@workstation.lab.example.com \  
> superuser
```

```
ubuntu@workstation:~$ openstack user create \  
> --project prod \  
> --password secret \  
> --email ubuntu@workstation.lab.example.com \  
> superuser  
+-----+-----+  
| Field | Value |  
+-----+-----+  
| default_project_id | a72b27c5a34245bbbd1c873a5fd420d4 |  
| domain_id | default |  
| email | ubuntu@workstation.lab.example.com |  
| enabled | True |  
| id | 18548b0493494ebba479a1de1b3e264 |  
| name | superuser |  
| options | {} |  
| password_expires_at | None |  
+-----+-----+  
ubuntu@workstation:~$
```

5. Assign the **admin** role to the user **superuser**.

```
ubuntu@workstation:~$ openstack role add \  
> --project prod \  
> --user superuser \  
> admin
```

```
ubuntu@workstation:~$ openstack role add \  
> --project prod \  
> --user superuser \  
> admin  
ubuntu@workstation:~$
```

6. Copy the keystone credentials file **~/keystonerc-admin** to **~/keystonerc-superuser**.

```
ubuntu@workstation~$ cp ~/keystonerc-admin ~/keystonerc-superuser
```

```
ubuntu@workstation:~$ cp ~/keystonerc-admin ~/keystonerc-superuser
ubuntu@workstation:~$
```

- Use nano to edit the `~/keystonerc-superuser` file. Change the `OS_USERNAME` to **superuser**, and change the `OS_TENANT_NAME` to **prod**. The file should match the contents shown below. Press **CTRL+X** to exit the file, then press **Y** and then **ENTER** to save the changes to the file.

```
ubuntu@workstation:~$ nano ~/keystonerc-superuser
```

```
GNU nano 2.9.3 /home/ubuntu/keystonerc-superuser Modified
unset OS_SERVICE_TOKEN
unset OS_TENANT_ID
unset OS_TENANT_NAME
export OS_USERNAME=superuser
export OS_PASSWORD=secret
export OS_AUTH_URL=http://192.168.1.22/identity
export OS_REGION_NAME=RegionOne
export OS_PROJECT_NAME=prod
export OS_INTERFACE=public
export OS_IDENTITY_API_VERSION=3
```

- Now, create a non-privileged user called **cloud-lab** with the password **secret**.

```
ubuntu@workstation:~$ openstack user create \
> --project prod \
> --password secret
> --email ubuntu@workstation.lab.example.com \
> cloud-lab
```

```
ubuntu@workstation:~$ openstack user create \
> --project prod \
> --password secret \
> --email ubuntu@workstation.lab.example.com \
> cloud-lab
+-----+-----+
| Field | Value |
+-----+-----+
| default_project_id | a72b27c5a34245bbbd1c873a5fd420d4 |
| domain_id | default |
| email | ubuntu@workstation.lab.example.com |
| enabled | True |
| id | 8588c114770741b6ae3cb0d3f6e85c00 |
| name | cloud-lab |
| options | {} |
| password_expires_at | None |
+-----+-----+
ubuntu@workstation:~$
```


- Assign **cloud-lab** the **member** role in the **prod** project so that it can perform actions in that project.

```
ubuntu@workstation:~$ openstack role add \  
> --project prod \  
> --user cloud-lab \  
> member
```

```
ubuntu@workstation:~$ openstack role add \  
> --project prod \  
> --user cloud-lab \  
> member  
ubuntu@workstation:~$
```

- Copy the keystone credentials file **~/keystonerc-superuser** to **~/keystonerc-cloud-lab**.

```
ubuntu@workstation:~$ cp ~/keystonerc-superuser ~/keystonerc-cloud-lab
```

```
ubuntu@workstation:~$ cp ~/keystonerc-superuser ~/keystonerc-cloud-lab  
ubuntu@workstation:~$
```

- Use **nano** to edit the **~/keystonerc-cloud-lab** file. Change the **OS_USERNAME** to **cloud-lab**. The file should match the contents shown below. Press **CTRL+X** to exit the file, then press **Y** and then **ENTER** to save the changes to the file.

```
ubuntu@workstation:~$ nano ~/keystonerc-cloud-lab
```

GNU nano 2.9.3	/home/ubuntu/keystonerc-cloud-lab	Modified
<pre>unset OS_SERVICE_TOKEN unset OS_TENANT_ID unset OS_TENANT_NAME export OS_USERNAME=cloud-lab export OS_PASSWORD=secret export OS_AUTH_URL=http://192.168.1.22/identity export OS_REGION_NAME=RegionOne export OS_PROJECT_NAME=prod export OS_INTERFACE=public export OS_IDENTITY_API_VERSION=3 █</pre>		

- Now, source the **keystonerc-superuser** keystone file to begin working with admin privileges in the **prod** project.

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

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

13. Before making an external network for the project, the existing one must be deleted. Before the existing external network can be deleted, the router needs to be deleted, which requires first deleting its interfaces. First, show the details of the router **router1** to find the interfaces to delete.

```
ubuntu@workstation:~$ openstack router show router1
```

```
ubuntu@workstation:~$ openstack router show router1
+-----+
| Field | Value |
+-----+-----+
| admin_state_up | UP |
| availability_zone_hints | |
| availability_zones | |
| created_at | 2023-12-22T04:11:00Z |
| description | |
| distributed | False |
| external_gateway_info | None |
| flavor_id | None |
| ha | False |
| id | 95624933-06a9-4f6e-bcf8-1177e31c5542 |
| interfaces_info | [{"subnet_id": "4c8a24fd-0428-4ca0-87ba-ad060722f8e7", "ip_address": "10.0.0.1", "port_id": "5290b8a9-4f26-415f-b134-459cb139a906"}, {"subnet_id": "467647b8-2d65-43bf-925b-18a692bba47f", "ip_address": "fd01:7620::ee07:1", "port_id": "5c97ebfb-d998-413e-998b-f375171b363f"}] |
| name | router1 |
| project_id | c50851c6559442df92e8e0799376a84f |
| revision_number | 9 |
| routes | |
| status | ACTIVE |
| tags | |
| updated_at | 2024-01-04T17:46:23Z |
+-----+-----+
ubuntu@workstation:~$
```

14. Next, delete the two interfaces on the router using the port_id values from the output of the previous step.

```
ubuntu@workstation:~$ openstack router remove port \
> router1 \
> 5290b8a9-4f26-415f-b134-459cb139a906
ubuntu@workstation:~$ openstack router remove port \
> router1 \
> 5c97ebfb-d998-413e-998b-f375171b363f
```

```
ubuntu@workstation:~$ openstack router remove port \
> router1 \
> 5290b8a9-4f26-415f-b134-459cb139a906
ubuntu@workstation:~$ openstack router remove port \
> router1 \
> 5c97ebfb-d998-413e-998b-f375171b363f
ubuntu@workstation:~$
```

Note

The actual IDs may differ from this example.

15. Now, **router1** can be deleted.

```
ubuntu@workstation:~$ openstack router delete router1
```

```
ubuntu@workstation:~$ openstack router delete router1
ubuntu@workstation:~$
```

16. Finally, delete the existing external network named **public**.

```
ubuntu@workstation:~$ openstack network delete public
```

```
ubuntu@workstation:~$ openstack network delete public
ubuntu@workstation:~$
```

17. Create an external, shared network called **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	2024-01-04T20:13:06Z
description	
dns_domain	None
id	9d09170a-73c8-419c-b8b4-db0bcf418ac8
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	c50851c6559442df92e0e0799376a84f
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	2024-01-04T20:13:06Z

```
ubuntu@workstation:~$
```

18. Create the **external_subnet** subnet in the **172.25.250.0/24** range. Make the floating IP allocation pool range from **172.25.250.25** to **172.25.250.30**, and allow DHCP. Set both the gateway and DNS nameserver addresses to **172.25.250.254**.

```
ubuntu@workstation:~$ openstack subnet create \
> --subnet-range 172.25.250.0/24 \
> --allocation-pool start=172.25.250.25,end=172.25.250.30 \
> --dhcp --network external \
> --gateway 172.25.250.254 \
> --dns-nameserver 172.25.250.254 external_subnet
```

```
ubuntu@workstation:~$ openstack subnet create \  
> --subnet-range 172.25.250.0/24 \  
> --allocation-pool start=172.25.250.25,end=172.25.250.30 \  
> --dhcp --network external \  
> --gateway 172.25.250.254 \  
> --dns-nameserver 172.25.250.254 external_subnet  
+-----+  
| Field | Value |  
+-----+  
| allocation_pools | 172.25.250.25-172.25.250.30 |  
| cidr | 172.25.250.0/24 |  
| created_at | 2024-01-04T20:18:22Z |  
| description | |  
| dns_nameservers | 172.25.250.254 |  
| enable_dhcp | True |  
| gateway_ip | 172.25.250.254 |  
| host_routes | |  
| id | a7caff1b-19a8-4eb1-bc81-35c878fa43f5 |  
| ip_version | 4 |  
| ipv6_address_mode | None |  
| ipv6_ra_mode | None |  
| name | external_subnet |  
| network_id | 9d09170a-73c8-419c-b8b4-db0bcf418ac8 |  
| project_id | a72b27c5a34245bbbd1c873a5fd420d4 |  
| revision_number | 0 |  
| segment_id | None |  
| service_types | |  
| subnetpool_id | None |  
| tags | |  
| updated_at | 2024-01-04T20:18:22Z |  
+-----+  
ubuntu@workstation:~$
```

19. Source the `/keystonerc-cloud-lab` keystone credentials file.

```
ubuntu@workstation:~$ source ~/keystonerc-cloud-lab
```

```
ubuntu@workstation:~$ source ~/keystonerc-cloud-lab  
ubuntu@workstation:~$
```

20. Create an internal network called **net1**.

```
ubuntu@workstation:~$ openstack network create net1
```

```
ubuntu@workstation:~$ openstack network create net1
```

Field	Value
admin_state_up	UP
availability_zone_hints	
availability_zones	
created_at	2024-01-04T20:20:38Z
description	
dns_domain	None
id	050946d2-1cc0-4403-9f26-8e8513dfd41d
ipv4_address_scope	None
ipv6_address_scope	None
is_default	False
is_vlan_transparent	None
mtu	1442
name	net1
port_security_enabled	True
project_id	a72b27c5a34245bbbd1c873a5fd420d4
provider:network_type	geneve
provider:physical_network	None
provider:segmentation_id	38038
qos_policy_id	None
revision_number	1
router:external	Internal
segments	None
shared	False
status	ACTIVE
subnets	
tags	
updated_at	2024-01-04T20:20:39Z

```
ubuntu@workstation:~$
```

21. Create a subnet for **net1** called **subnet1** in the **192.168.0.0/24** range. Allow DHCP on the subnet.

```
ubuntu@workstation:~$ openstack subnet create \
> --subnet-range 192.168.0.0/24 \
> --network net1 subnet1
```

```
ubuntu@workstation:~$ openstack subnet create \
> --subnet-range 192.168.0.0/24 \
> --network net1 subnet1
```

Field	Value
allocation_pools	192.168.0.2-192.168.0.254
cidr	192.168.0.0/24
created_at	2024-01-04T20:21:21Z
description	
dns_nameservers	
enable_dhcp	True
gateway_ip	192.168.0.1
host_routes	
id	6b44acd8-1ef7-4db6-9c3d-4bdc21179612
ip_version	4
ipv6_address_mode	None
ipv6_ra_mode	None
name	subnet1
network_id	050946d2-1cc0-4403-9f26-8e8513dfd41d
project_id	a72b27c5a34245bbbd1c873a5fd420d4
revision_number	0
segment_id	None
service_types	
subnetpool_id	None
tags	
updated_at	2024-01-04T20:21:21Z

```
ubuntu@workstation:~$
```

22. Create a router named **router1** so that the internal and external networks can be connected.

```
ubuntu@workstation:~$ openstack router create router1
```

```
ubuntu@workstation:~$ openstack router create router1
```

Field	Value
admin_state_up	UP
availability_zone_hints	
availability_zones	
created_at	2024-01-04T20:21:53Z
description	
distributed	False
external_gateway_info	None
flavor_id	None
ha	False
id	e6109151-8725-431f-90e5-a8a947679489
name	router1
project_id	a72b27c5a34245bbbd1c873a5fd420d4
revision_number	1
routes	
status	ACTIVE
tags	
updated_at	2024-01-04T20:21:53Z

```
ubuntu@workstation:~$
```

23. Add a port to the router for the internal network.

```
ubuntu@workstation:~$ openstack router add subnet router1 subnet1
```

```
ubuntu@workstation:~$ openstack router add subnet router1 subnet1
ubuntu@workstation:~$
```

24. Set the external network as the gateway for the router.

```
ubuntu@workstation:~$ openstack router set \
> --external-gateway external \
> router1
```

```
ubuntu@workstation:~$ openstack router set \
> --external-gateway external \
> router1
ubuntu@workstation:~$
```

25. Allocate a floating IP address from the **external** network for the **prod** project.

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



```
ubuntu@workstation:~$ openstack floating ip create external
+-----+-----+
| Field                | Value                                |
+-----+-----+
| created_at           | 2024-01-04T20:48:35Z                |
| description          |                                       |
| fixed_ip_address     | None                                 |
| floating_ip_address  | 172.25.250.30                        |
| floating_network_id  | 9d09170a-73c8-419c-b8b4-db0bcf418ac8 |
| id                   | f60c1e3a-cb0e-45ed-9e3f-8c64508594fd |
| name                 | 172.25.250.30                        |
| port_id              | None                                 |
| project_id           | a72b27c5a34245bbbd1c873a5fd420d4    |
| qos_policy_id        | None                                 |
| revision_number      | 0                                    |
| router_id            | None                                 |
| status               | DOWN                                 |
| subnet_id            | None                                 |
| updated_at           | 2024-01-04T20:48:35Z                |
+-----+-----+
ubuntu@workstation:~$
```

26. Generate a key pair for the **cloud-lab** user named **key1**.

```
ubuntu@workstation:~$ openstack keypair create \
> key1 > ~/Downloads/key1.pem
```

```
ubuntu@workstation:~$ openstack keypair create \
> key1 > ~/Downloads/key1.pem
ubuntu@workstation:~$
```

27. Change the permissions of the key pair file so that only the **ubuntu** user has read and write permissions.

```
ubuntu@workstation:~$ chmod 600 ~/Downloads/key1.pem
```

```
ubuntu@workstation:~$ chmod 600 ~/Downloads/key1.pem
ubuntu@workstation:~$
```

28. Create a security group named **sg1** for the **prod** project.

```
ubuntu@workstation:~$ openstack security group create \
> --description "SSH, ICMP, and FTP" sg1
```

```
ubuntu@workstation:~$ openstack security group create \
> --description "SSH, ICMP, and FTP" sg1
+-----+
| Field | Value |
+-----+
| created_at | 2024-01-04T21:33:42Z |
| description | SSH, ICMP, and FTP |
| id | c38c7982-bf57-4f91-b008-2da8310bce31 |
| name | sg1 |
| project_id | a72b27c5a34245bbbd1c873a5fd420d4 |
| revision_number | 1 |
| rules | created_at='2024-01-04T21:33:42Z', direction='egress', ethertype='IPv4', id='63fa510b-8aaa-464d-8aef-ab2af44275e3', standard_attr_id='71', updated_at='2024-01-04T21:33:42Z' |
| updated_at | 2024-01-04T21:33:42Z |
+-----+
ubuntu@workstation:~$
```

29. Create a security group rule to allow **SSH** traffic from any IP address. SSH uses the TCP protocol on port 22 by default.

```
ubuntu@workstation:~$ openstack security group \
> rule create \
> --proto tcp --remote-ip 0.0.0.0/0 --dst-port 22:22 sg1
```

```
ubuntu@workstation:~$ openstack security group \
> rule create \
> --proto tcp --remote-ip 0.0.0.0/0 --dst-port 22:22 sg1
+-----+
| Field | Value |
+-----+
| created_at | 2024-01-04T21:40:39Z |
| description | |
| direction | ingress |
| ether_type | IPv4 |
| id | 42de396a-9317-42d9-be09-8a2476a69f85 |
| name | None |
| port_range_max | 22 |
| port_range_min | 22 |
| project_id | a72b27c5a34245bbbd1c873a5fd420d4 |
| protocol | tcp |
| remote_group_id | None |
| remote_ip_prefix | 0.0.0.0/0 |
| revision_number | 0 |
| security_group_id | 47533cb6-6c1f-4a0c-85a1-fb040708a79a |
| updated_at | 2024-01-04T21:40:39Z |
+-----+
ubuntu@workstation:~$
```

30. Create a security group rule to allow **ICMP** traffic from any IP address.

```
ubuntu@workstation:~$ openstack security group \
> rule create \
> --proto icmp --remote-ip 0.0.0.0/0 sg1
```

```
ubuntu@workstation:~$ openstack security group \
> rule create \
> --proto icmp --remote-ip 0.0.0.0/0 sg1
```

Field	Value
created_at	2024-01-04T21:50:30Z
description	
direction	ingress
ether_type	IPv4
id	7508f181-399d-4ccc-aa4f-27ad4ffcd203
name	None
port_range_max	None
port_range_min	None
project_id	a72b27c5a34245bbbd1c873a5fd420d4
protocol	icmp
remote_group_id	None
remote_ip_prefix	0.0.0.0/0
revision_number	0
security_group_id	47533cb6-6c1f-4a0c-85a1-fb040708a79a
updated_at	2024-01-04T21:50:30Z

```
ubuntu@workstation:~$
```

31. Create a security group rule to allow **FTP** traffic from any IP address. FTP uses the TCP protocol on port 20 (data channel) and port 21 (control channel).

```
ubuntu@workstation:~$ openstack security group \
> rule create \
> --proto tcp --remote-ip 0.0.0.0/0 --dst-port 20:21 sg1
```

```
ubuntu@workstation:~$ openstack security group \
> rule create \
> --proto tcp --remote-ip 0.0.0.0/0 --dst-port 20:21 sg1
+-----+
| Field          | Value                                     |
+-----+-----+
| created_at     | 2024-01-04T21:52:04Z                    |
| description    |                                           |
| direction      | ingress                                  |
| ether_type     | IPv4                                     |
| id             | b4125b68-e5a9-4778-bc4d-6dea1e3b3bd6    |
| name           | None                                     |
| port_range_max | 21                                       |
| port_range_min | 20                                       |
| project_id     | a72b27c5a34245bbbd1c873a5fd420d4       |
| protocol       | tcp                                     |
| remote_group_id| None                                     |
| remote_ip_prefix| 0.0.0.0/0                               |
| revision_number| 0                                       |
| security_group_id| 47533cb6-6c1f-4a0c-85a1-fb040708a79a  |
| updated_at     | 2024-01-04T21:52:04Z                    |
+-----+-----+
ubuntu@workstation:~$
```

32. The FTP server instance is almost ready to be launched. First, use nano to create a file named `script` in the home directory. Be sure it has the correct indentation and matches the contents shown below. Press **CTRL+X** to exit the file, then press **Y** and then **ENTER** to save the changes to the file.

```
ubuntu@workstation:~$ nano ~/script
```

```
GNU nano 2.9.3 /home/ubuntu/script Modified
#cloud-config
runcmd:
- echo "This instance has been customized by cloud-init" > /etc/motd

```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Note

This cloud-init script writes to the “message of the day” file, and its contents will be displayed upon a successful login.

33. Create an instance named **ftp_server** using **net1** for the internal network, **m1.small** as the flavor, and **ubuntu** as the image.

```
ubuntu@workstation:~$ openstack server create \  
> --image ubuntu \  
> --flavor m1.small \  
> --security-group sg1 \  
> --user-data script \  
> --key-name key1 \  
> --nic net-id=net1 \  
> --wait ftp_server
```

```
ubuntu@workstation:~$ openstack server create \
> --image ubuntu \
> --flavor m1.small \
> --security-group sgl \
> --user-data script \
> --key-name key1 \
> --nic net-id=net1 \
> --wait ftp_server
```

Field	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	nova
OS-EXT-STS:power_state	Running
OS-EXT-STS:task_state	None
OS-EXT-STS:vm_state	active
OS-SRV-USG:launched_at	2024-01-04T22:13:34.000000
OS-SRV-USG:terminated_at	None
accessIPv4	
accessIPv6	
addresses	net1=192.168.0.167
adminPass	WkwTRL595gZh
config_drive	
created	2024-01-04T22:13:30Z
flavor	m1.small (2)
hostId	db26c44769e798a3b00d88da852fc5895221d8f8f0fcec687eccf35a
id	8652f17e-dc34-4672-bd79-5b8ca98764f1
image	ubuntu (b6e959dd-7ad1-409f-966c-9c34eee29b36)
key_name	key1
name	ftp_server
progress	0
project_id	a72b27c5a34245bbbd1c873a5fd420d4
properties	
security_groups	name='sg1'
status	ACTIVE
updated	2024-01-04T22:13:34Z
user_id	8588c114770741b6ae3cb0d3f6e85c00
volumes_attached	

```
ubuntu@workstation:~$
```

34. Ensure that the instance state is **ACTIVE**.

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

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

ID	Name	Status	Networks	Image	Flavor
8652f17e-dc34-4672-bd79-5b8ca98764f1	ftp_server	ACTIVE	net1=192.168.0.167	ubuntu	m1.small

```
ubuntu@workstation:~$
```

35. When the instance state is **ACTIVE**, list the floating IP addresses available.

```
ubuntu@workstation:~$ openstack floating ip list
```

```
ubuntu@workstation:~$ openstack floating ip list
```

ID	Floating IP Address	Fixed IP Address	Port	Floating Network	Project
f60c1e3a-cb0e-45ed-9e3f-8c64508594fd	172.25.250.30	None	None	9d09170a-73c8-419c-b8b4-db0bcf418ac8	a72b27c5a34245bbbd1c873a5fd420d4

```
ubuntu@workstation:~$
```

36. Associate an open floating IP address to the instance.

```
ubuntu@workstation:~$ openstack server add \  
> floating ip ftp_server 172.25.250.30
```

```
ubuntu@workstation:~$ openstack server add \  
> floating ip ftp_server 172.25.250.30  
ubuntu@workstation:~$
```

Note

When associating the floating IP, make sure to use the IP address that appears for you in the previous step as it may differ from this example.

37. SSH into the **ftp_server** instance.

```
ubuntu@workstation:~$ ssh -i ~/Downloads/key1.pem root@172.25.250.30
```

Note

The IP address may differ slightly from this example. Make sure to use the floating IP address that you created.

38. Verify that the **vsftpd** package is installed.

```
root@ftp-server:~# apt show vsftpd
```

39. Use **nano** to edit the **vsftpd** configuration file and uncomment the variable **anon_upload_enable** and the variable **anon_mkdir_write**, then append the variable **allow_writeable_chroot**.

```
root@ftp-server:~# nano /etc/vsftpd/vsftpd.conf
```

40. Change the ownership of the **/var/ftp/pub/** directory.

```
root@ftp-server:~# sudo chown -R ftp. /var/ftp/pub/
```

41. Exit from the **ftp_server** instance.

```
root@ftp-server:~# exit
```

42. From **workstation**, create a text file named **test_file.txt** containing the string "This is my file."

```
ubuntu@workstation:~$ echo "This is my file" > test_file.txt
```

43. Open an FTP session to the FTP server and upload the **test_file.txt** file. Log out when done. Use **anonymous** as the user and when prompted for the password, press the **Enter** key for no password input. Follow the instructions from the example and summary below.

```
ubuntu@workstation:~$ ftp 172.25.250.30
ftp> passive
ftp> dir
ftp> cd pub
ftp> put test_file.txt test_file.txt
ftp> exit
```

Note

The IP address may differ slightly from this example. Make sure to use the floating IP address that you created.

44. SSH into the **ftp_server** instance.

```
ubuntu@workstation:~$ ssh -i ~/Downloads/key1.pem cloud-user@172.25.250.30
```

45. Verify the file uploaded successfully.

```
ubuntu@workstation:~$ sudo ls /var/ftp/pub
```

46. Exit from the **ftp_server** instance.

```
ubuntu@workstation:~$ exit
```

47. The lab is now complete.