# Method of Procedure (MOP) for Installing a VM in OpenStack and Configuring Firewall

## Title: OpenStack VM Installation and Firewall Configuration

### Date: 2024-06-05

### Reference

- OpenStack Documentation: https://docs.openstack.org/

- Firewall Configuration Guide: https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-20-04

### Prerequisites

- Access to an OpenStack environment with appropriate permissions to create VMs and manage security groups.

- Basic knowledge of OpenStack CLI or Horizon dashboard.

- An existing network in OpenStack to which the VM will be connected.

### Objective

This MOP outlines the steps to create a VM in OpenStack, assign a specific IP address, configure a firewall to block all traffic except for a specific subnet, and set CPU and RAM according to user specifications.

### Procedure

#### Step 1: Create a VM in OpenStack

1. \*\*Log in to OpenStack:\*\*

- Use the OpenStack CLI or Horizon dashboard to log in to your OpenStack environment.

2. \*\*Select the appropriate project:\*\*

- Ensure you are in the correct project where you want to create the VM.

3. \*\*Create a VM:\*\*

- Using the OpenStack CLI, run the following command to create a VM:

```bash

openstack server create --flavor <FLAVOR\_ID> --image <IMAGE\_ID> --network <NETWORK\_ID> --key-name <KEY\_NAME> --security-group <SECURITY\_GROUP> <VM\_NAME>

```

- Replace `<FLAVOR\_ID>`, `<IMAGE\_ID>`, `<NETWORK\_ID>`, `<KEY\_NAME>`, `<SECURITY\_GROUP>`, and `<VM\_NAME>` with your specific values.

4. \*\*Assign a specific IP address:\*\*

- To assign a specific floating IP, first allocate a floating IP:

```bash

openstack floating ip create <EXTERNAL\_NETWORK>

```

- Then associate the floating IP with your VM:

```bash

openstack floating ip set --port <PORT\_ID> <FLOATING\_IP>

```

#### Step 2: Configure Firewall

1. \*\*Access the VM:\*\*

- SSH into the VM using the floating IP assigned:

```bash

ssh -i <KEY\_PATH> <USER>@<FLOATING\_IP>

```

2. \*\*Install UFW (Uncomplicated Firewall):\*\*

- Update the package list and install UFW:

```bash

sudo apt update

sudo apt install ufw

```

3. \*\*Set default policies:\*\*

- Block all incoming traffic by default:

```bash

sudo ufw default deny incoming

```

- Allow outgoing traffic:

```bash

sudo ufw default allow outgoing

```

4. \*\*Allow specific subnet:\*\*

- Allow traffic from a specific subnet (e.g., `192.168.1.0/24`):

```bash

sudo ufw allow from 192.168.1.0/24

```

5. \*\*Enable UFW:\*\*

- Enable the firewall:

```bash

sudo ufw enable

```

6. \*\*Check UFW status:\*\*

- Verify the firewall rules:

```bash

sudo ufw status verbose

```

#### Step 3: Set CPU and RAM

1. \*\*Modify VM flavor (if necessary):\*\*

- If you need to change the CPU or RAM after the VM is created, you can either create a new flavor or resize the existing VM.

- To resize the VM, use the following command:

```bash

openstack server resize --flavor <NEW\_FLAVOR\_ID> <VM\_NAME>

```

- Confirm the resize:

```bash

openstack server resize --confirm <VM\_NAME>

```

2. \*\*Create a new flavor (if needed):\*\*

- If you want to create a new flavor, use:

```bash

openstack flavor create --ram <RAM\_MB> --disk <DISK\_GB> --vcpus <VCPUS> <FLAVOR\_NAME>

```

- Replace `<RAM\_MB>`, `<DISK\_GB>`, `<VCPUS>`, and `<FLAVOR\_NAME>` with your desired specifications.

### Conclusion

This MOP provides a comprehensive guide to creating a VM in OpenStack, assigning an IP address, configuring a firewall to restrict traffic, and setting CPU and RAM according to user requirements. Ensure to follow each step carefully and verify configurations as needed.