# Method of Procedure (MOP) for Deploying a VM on OpenStack with Specific Configuration

## Document Control

\*\*Title:\*\* Deploying a VM on OpenStack with Specific Configuration

\*\*Version:\*\* 1.0

\*\*Date:\*\* 2024-06-05

\*\*Author:\*\* [Your Name]

\*\*Reviewed by:\*\* [Reviewer Name]

\*\*Approval:\*\* [Approver Name]

---

## Reference

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

- OpenStack CLI Reference: https://docs.openstack.org/python-openstackclient/latest/

---

## Prerequisites

1. \*\*OpenStack Access\*\*: Ensure you have access to an OpenStack environment with the necessary permissions to create VMs.

2. \*\*OpenStack CLI\*\*: Install the OpenStack command-line interface (CLI) on your local machine or access it through a terminal in your OpenStack environment.

3. \*\*Network Configuration\*\*: Ensure you have the necessary network details, including the specific subnet and allowed IP addresses.

---

## Objective

This MOP outlines the steps to deploy a VM on OpenStack with specific memory, CPU configuration, and a designated node name. It also includes instructions for setting a static IP address and configuring a firewall to allow only specific IPs in a designated subnet.

---

## Procedure

### Step 1: Set Up OpenStack Environment

1. \*\*Source OpenStack Credentials\*\*:

Open a terminal and source your OpenStack credentials file to authenticate your session.

```bash

source /path/to/your/openrc.sh

```

### Step 2: Create a VM with Specific Configuration

1. \*\*Define Variables\*\*: Set the following variables in your terminal:

```bash

export VM\_NAME="your\_node\_name" # Replace with your desired node name

export IMAGE\_NAME="Ubuntu 22.04" # Replace with your desired image name

export FLAVOR\_NAME="m1.small" # Replace with your desired flavor (2G RAM, 2 CPU)

export NETWORK\_NAME="your\_network" # Replace with your network name

export FLOATING\_IP="your\_floating\_ip" # Replace with your desired floating IP

```

2. \*\*Create the VM\*\*:

Use the OpenStack CLI to create the VM with the specified parameters.

```bash

openstack server create --flavor $FLAVOR\_NAME --image $IMAGE\_NAME --network $NETWORK\_NAME --key-name your\_key\_name $VM\_NAME

```

3. \*\*Assign Floating IP\*\* (if applicable):

If you need to assign a floating IP to the VM, use the following command:

```bash

openstack floating ip create public

openstack server add floating ip $VM\_NAME $FLOATING\_IP

```

### Step 3: Configure the Firewall

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

SSH into the newly created VM using the floating IP.

```bash

ssh ubuntu@$FLOATING\_IP

```

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

If UFW is not installed, install it using:

```bash

sudo apt update

sudo apt install ufw

```

3. \*\*Configure UFW\*\*:

Allow only specific IPs in the designated subnet. Replace `your\_subnet` with the actual subnet you want to allow.

```bash

sudo ufw allow from your\_subnet to any

sudo ufw enable

```

4. \*\*Verify UFW Status\*\*:

Check the status of UFW to ensure the rules are applied correctly.

```bash

sudo ufw status

```

### Step 4: Final Verification

1. \*\*Check VM Status\*\*:

Ensure the VM is running and accessible.

```bash

openstack server list

```

2. \*\*Test Connectivity\*\*:

From another machine in the allowed subnet, test connectivity to the VM using ping or SSH.

---

## Troubleshooting

- If the VM fails to launch, check the OpenStack dashboard for error messages.

- Ensure that the specified image and flavor exist in your OpenStack environment.

- Verify network configurations and ensure that the subnet allows the necessary traffic.

---

## Conclusion

This MOP provides a detailed guide for deploying a VM on OpenStack with specific configurations, including memory, CPU, and firewall settings. Follow the steps carefully to ensure a successful deployment.

---

\*\*End of Document\*\*