# Method of Procedure (MOP) for Deploying an OpenStack VM

## Objective

This MOP outlines the steps to deploy a virtual machine (VM) on a specific OpenStack node, assign it a static IP address, install a firewall, and configure it to allow only specific subnet IPs. Additionally, it will cover how to set the CPU and RAM specifications for the VM.

## Prerequisites

- Access to an OpenStack environment with administrative privileges.

- OpenStack CLI installed and configured on your local machine or access to the OpenStack dashboard.

- Knowledge of the specific node where the VM will be deployed.

- The desired static IP address and subnet information.

## Steps

### 1. Log in to OpenStack

Log in to your OpenStack environment using the OpenStack CLI or the dashboard.

#### CLI Login

```bash

source /path/to/your/openrc.sh

```

### 2. Create a Key Pair (if not already created)

If you do not have a key pair for SSH access, create one.

#### CLI Command

```bash

openstack keypair create --public-key /path/to/your/public\_key.pub my\_key

```

### 3. Create a Security Group

Create a security group to define the firewall rules.

#### CLI Command

```bash

openstack security group create my\_security\_group

```

### 4. Add Rules to the Security Group

Add rules to allow SSH and ICMP (ping) traffic from a specific subnet.

#### CLI Command

```bash

# Allow SSH (port 22)

openstack security group rule create --proto tcp --dst-port 22 my\_security\_group

# Allow ICMP (ping)

openstack security group rule create --proto icmp my\_security\_group

# Allow traffic from a specific subnet (e.g., 192.168.1.0/24)

openstack security group rule create --proto tcp --dst-port 80 --remote-ip 192.168.1.0/24 my\_security\_group

```

### 5. Create a Flavor

Create a flavor that specifies the desired CPU and RAM for the VM.

#### CLI Command

```bash

openstack flavor create --ram 2048 --disk 20 --vcpus 2 my\_flavor

```

- Adjust `--ram`, `--disk`, and `--vcpus` as per your requirements.

### 6. Create a Network and Subnet (if not already created)

If you do not have a network and subnet, create them.

#### CLI Command

```bash

# Create a network

openstack network create my\_network

# Create a subnet

openstack subnet create --network my\_network --subnet-range 192.168.1.0/24 my\_subnet

```

### 7. Create a VM Instance

Deploy the VM instance on the specific node with the desired static IP.

#### CLI Command

```bash

openstack server create --flavor my\_flavor --image <image\_id> --key-name my\_key --security-group my\_security\_group --network my\_network --nic net-id=<network\_id>,v4-fixed-ip=192.168.1.10 my\_instance

```

- Replace `<image\_id>` with the ID of the image you want to use.

- Replace `<network\_id>` with the ID of the network you created.

### 8. Verify the VM Creation

Check if the VM is created successfully.

#### CLI Command

```bash

openstack server list

```

### 9. Access the VM

SSH into the VM using the assigned static IP.

#### CLI Command

```bash

ssh -i /path/to/your/private\_key.pem ubuntu@192.168.1.10

```

### 10. Install and Configure Firewall

Once logged into the VM, install a firewall (e.g., UFW) and configure it.

#### CLI Commands

```bash

# Update package list

sudo apt update

# Install UFW

sudo apt install ufw

# Allow SSH

sudo ufw allow ssh

# Allow traffic from specific subnet

sudo ufw allow from 192.168.1.0/24

# Enable UFW

sudo ufw enable

```

### 11. Verify Firewall Status

Check the status of the firewall to ensure the rules are applied.

#### CLI Command

```bash

sudo ufw status

```

## Conclusion

You have successfully deployed an OpenStack VM with a static IP, configured a firewall to allow only specific subnet traffic, and set the desired CPU and RAM specifications. Ensure to monitor the VM and adjust configurations as necessary.