# 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 a static IP address, install a firewall, configure it to block all traffic except for specific ports, and set the CPU and RAM according to user specifications.

## Prerequisites

- Access to an OpenStack environment with appropriate permissions to create VMs.

- OpenStack CLI or Horizon dashboard access.

- Knowledge of the desired static IP address and the ports to be allowed through the firewall.

## Procedure

### Step 1: Log in to OpenStack

1. Open a terminal or access the Horizon dashboard.

2. If using the CLI, source your OpenStack credentials:

```bash

source /path/to/your/openrc.sh

```

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

1. Create a key pair to access the VM:

```bash

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

```

### Step 3: Create a Security Group

1. Create a security group to manage firewall rules:

```bash

openstack security group create my\_security\_group

```

### Step 4: Configure Firewall Rules

1. Allow specific ports (e.g., SSH on port 22 and HTTP on port 80):

```bash

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

openstack security group rule create --proto tcp --dst-port 80 my\_security\_group

```

2. Block all other traffic by default (this is the default behavior of security groups in OpenStack).

### Step 5: Create a Network and Subnet (if not already created)

1. Create a network:

```bash

openstack network create my\_network

```

2. Create a subnet with the desired CIDR:

```bash

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

```

### Step 6: Launch the VM

1. Use the following command to launch the VM with specified CPU and RAM:

```bash

openstack server create --flavor m1.small --image my\_image --key-name my\_key --security-group my\_security\_group --network my\_network --nic net-id=my\_network --user-data my\_user\_data.sh my\_vm

```

- Replace `m1.small` with the desired flavor that meets your CPU and RAM requirements.

- Replace `my\_image` with the desired image name.

- Replace `my\_user\_data.sh` with a script if you want to run any initialization commands on boot.

### Step 7: Assign a Static IP Address

1. Allocate a floating IP (if needed) or assign a fixed IP from the subnet:

```bash

openstack server add floating ip my\_vm <floating\_ip\_address>

```

or

```bash

openstack server set --fixed-ip <fixed\_ip\_address> my\_vm

```

### Step 8: Access the VM

1. SSH into the VM using the assigned IP address:

```bash

ssh -i /path/to/your/private\_key user@<static\_ip\_address>

```

### Step 9: Install and Configure Firewall on the VM

1. Update the package list:

```bash

sudo apt update

```

2. Install UFW (Uncomplicated Firewall):

```bash

sudo apt install ufw

```

3. Set default policies to deny all incoming traffic:

```bash

sudo ufw default deny incoming

sudo ufw default allow outgoing

```

4. Allow specific ports (e.g., SSH and HTTP):

```bash

sudo ufw allow 22/tcp

sudo ufw allow 80/tcp

```

5. Enable the firewall:

```bash

sudo ufw enable

```

### Step 10: Verify Firewall Status

1. Check the status of UFW to ensure the rules are applied:

```bash

sudo ufw status

```

## Conclusion

You have successfully deployed an OpenStack VM with a static IP address, configured a firewall to block all traffic except for specified ports, and set the CPU and RAM according to your requirements. Ensure to monitor the VM and adjust firewall rules as necessary based on your application needs.