



LAB 4

Infrastructure-as-a-Service with DevStack (OpenStack)

Full name: Tran Dang Khoa

Student ID: B2014926

- Note: screenshots need to be clear and good-looking; submissions must be in PDF format.

1. OpenStack overview

- Please take a visit to the OpenStack [home page](#)
- OpenStack installation on Ubuntu [guide](#)
- DevStack installation [guide](#)

2. Setup DevStack on VirtualBox

Deploying OpenStack could be challenging for beginners. DevStack is a set of scripts and utilities to quickly bring up a complete OpenStack environment based on the latest versions of everything from git master. It is used interactively as a development environment and as the basis for much of the OpenStack project's functional testing.

The source of DevStack is available at <https://opendev.org/openstack/devstack>.

2.1. Ubuntu server installation

- Create a virtual machine using VirtualBox/VMWare.
 - Name: DevStack; Type: Linux; Version: Ubuntu 22.04 (64bit)
 - Memory: 8G; Processors: 4 CPUs (add more if possible)
 - Hard disk: 50G
 - Network: create 2 adapters for the VM
 - Both are attached to a bridged adapter;
 - Advanced → Promiscuous mod: Allow All
 - Enable PAE/NX and Enable Nested VT-x/AMD-v (Setting/System/Processor). If the option is grey out, enter the following command

```
$ VBoxManage modifyvm "vm name" --nested-hw-virt on
```
- Install [Ubuntu server 22.04 LTS](#) as the OS of the VM. (take a screenshot after finishing the installation)

```
khoab2014926@devstack:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:81:55:01 brd ff:ff:ff:ff:ff:ff
    inet 172.22.4.89/23 metric 100 brd 172.22.5.255 scope global dynamic enp0s3
        valid_lft 86224sec preferred_lft 86224sec
    inet6 fe80::a00:27ff:fe81:5501/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:72:65:6f brd ff:ff:ff:ff:ff:ff
    inet 172.22.5.118/23 metric 100 brd 172.22.5.255 scope global dynamic enp0s8
        valid_lft 14224sec preferred_lft 14224sec
    inet6 fe80::a00:27ff:fe72:656f/64 scope link
        valid_lft forever preferred_lft forever
khoab2014926@devstack:~$
```

- SSH to the VM and update its OS:

```
$ sudo apt update && sudo apt upgrade -y
```

```
khoab2014926@devstack:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for khoab2014926:
Hit:1 http://vn.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://vn.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://vn.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
17 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

2.2. DevStack installation

2.2.1. Add Stack User

- DevStack should be run as a non-root user with sudo enabled. We should create a separate stack user to run DevStack with

```
$ sudo useradd -s /bin/bash -d /opt/stack -m stack
```

- Ensure the home directory for the stack user has executable permission for all

```
$ sudo chmod +x /opt/stack
```

- Since this user will be making many changes to your system, it should have sudo privileges:

```
$ echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee
/etc/sudoers.d/stack
$ sudo -u stack -i
```

2.2.2. Download DevStack

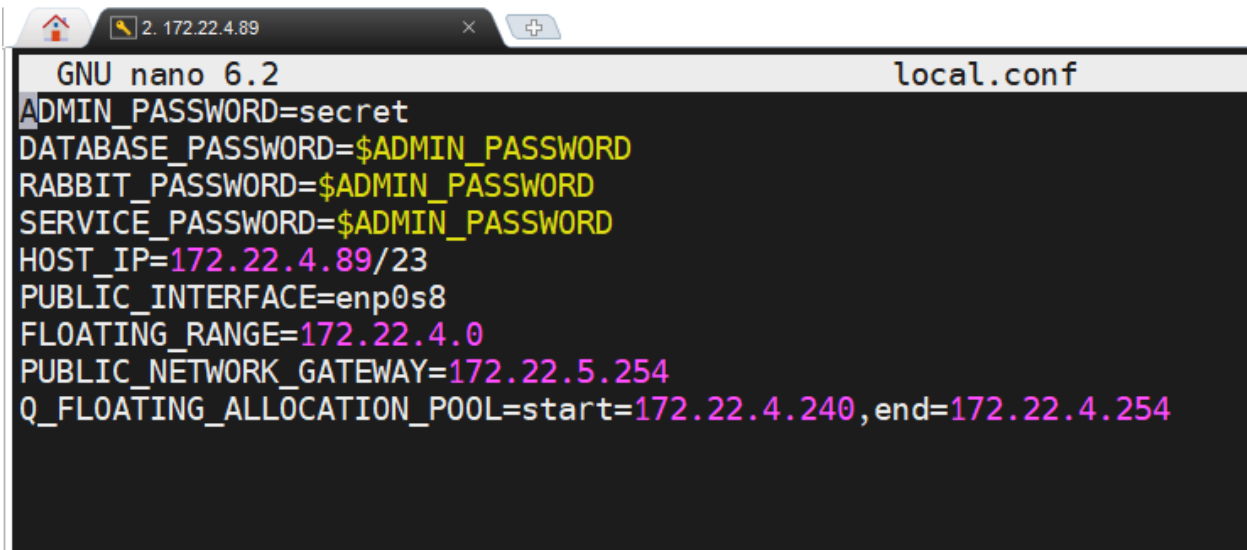
- The DevStack repo contains a script that installs OpenStack and templates for configuration files.

```
$ git clone https://opendev.org/openstack/devstack
$ cd devstack
```

2.2.3. Create a local.conf

- Create a local.conf file with four passwords preset at the root of the DevStack git repo.

```
$ nano local.conf
[[local|localrc]]
ADMIN_PASSWORD=secret
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
HOST_IP=<DevStack IP>
PUBLIC_INTERFACE=enp0s8
FLOATING_RANGE=<Network Address>
PUBLIC_NETWORK_GATEWAY=<Gateway Address>
Q_FLOATING_ALLOCATION_POOL=start=<Start of floating
IP>,end=<End of floating IP>
```



The screenshot shows a terminal window with a browser-like address bar at the top displaying '2. 172.22.4.89'. The terminal title is 'GNU nano 6.2' and the file being edited is 'local.conf'. The content of the file is as follows:

```
ADMIN_PASSWORD=secret
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
HOST_IP=172.22.4.89/23
PUBLIC_INTERFACE=enp0s8
FLOATING_RANGE=172.22.4.0
PUBLIC_NETWORK_GATEWAY=172.22.5.254
Q_FLOATING_ALLOCATION_POOL=start=172.22.4.240,end=172.22.4.254
```

2.2.4. Start the install

```
$ ./stack.sh
```

- This will take 15 - 30 minutes, depending on your internet connection speed. Many git trees and packages will be installed during this process. DevStack will have installed Keystone, Glance, Nova, Placement, Cinder, Neutron, and Horizon

- Take a snapshot of the Devstack VM **right after** the installation finishes.
- You can access Horizon to experience the web interface to OpenStack and manage VMs, networks, volumes, and images from http://YOUR_VM_IP.

3. Using OpenStack services via Horizon

- Horizon provides a dashboard for using OpenStack. On your computer, use this URL to access the GUI:
http://YOUR_VM_IP
- Log on to OpenStack using the credential “admin/secret”
(take a screenshot)

3.1. Creating Projects and Users

Using Horizon dashboard to create:

- Create a project:
 - Name: <CloudComputing_Your student ID>
- Create a user:
 - User Name: <Your student ID>
 - Password: <Your password>
 - Primary Project: <CloudComputing_Your student ID>
 - Role: admin
- Log out from OpenStack then log on again with the user <Your student ID>
(take a screenshot)

3.2. Creating Images

- Download the [CirrOS image](#), then create an image
 - Image Name: CirrOS
 - Format: QCOW2
- Download the [Ubuntu 22.04 cloud image](#), then create the second image
 - Image Name: Ubuntu 22.04
 - Format: QCOW2

3.3. Create Networks, Routers, Security Groups, and Floating IP allocation

- Create a Network:
 - Network Name: “VM_Network”
 - Subnet Name: “VM_Network_Subnet”
 - Network Address: 10.0.1.0/24
 - Gateway IP: 10.0.1.1
 - Enable DHCP
 - Allocation Pools: 10.0.1.100,10.0.1.254
 - DNS Name Servers: 8.8.8.8
- Create a router to connect VM_Network to Public_Network
 - Router Name: VM_Network_Router

- External Network: public
- On the router, choose the “Add interface” function to connect “VM_Network” to the router
(take a screenshot of network topology)
- Create a security group
 - Name: VM_Security_Group
 - Add Ingress Rule that allows: ICMP, SSH, HTTP, HTTPS
- Allocate 3 Floating IPs for the project <CloudComputing_Your student ID>
(take a screenshot)

3.4. Create instances (VMs):

3.4.1. CirrOS VM

- Instance name: <Your student ID>_Cirros
- Source: CirrOS; Select boot source: image; Create new Volume: No
- Flavor: cirros256
- Security group: VM_Security_Group
- Networks: VM_Network
- Access the VM using the web console. Then log in to the VM using “cirros/gocubsgo” credentials, and execute the “ping google.com -c 3 ; uname -a” command.
(take a screenshot)

3.4.2. Ubuntu VM

- Instance name: <Your student ID>_Ubuntu
- Source: Ubuntu 22.04; Select boot source: image; Create new Volume: No
- Flavor: ds512M
- Security group: VM_Security_Group
- Networks: VM_Network
- Key pair: Create a new key pair,
 - Key Pair Name: <Your student ID>_Ubuntu_Key
 - Key Type: SSH Key
 - Copy the private key to the file <Your student ID>_Ubuntu_Key.pem
- Configuration/Customization Script:

```
#cloud-config
password: mypasswd
chpasswd: { expire: False }
ssh_pwauth: True
```
- Launch your VM, then assign it a floating IP.
- SSH to your VM using an SSH client (MobaXterm)
 - Remote host: VM floating IP
 - Username: ubuntu

- Using the <Your student ID>_Ubuntu_Key.pem file as private key file
- Execute `"sudo ping google.com -c 3 ; uname -a"` command.
(take a screenshot)
- Execute `"sudo ping <Cirros VM IP>"` command.
(take a screenshot)

4. Using OpenStack services via CLI and REST API

- SSH to the DevStack VM
- Install the OpenStack client
 - \$ `sudo apt update`
 - \$ `sudo apt install python3-openstackclient -y`
 - On Horizon dashboard -> API Access -> Download OpenStack RC File, download OpenStack RC file (CloudComputing_<Your student ID>-openrc.sh)
- Copy the file to the computer from which you want to run OpenStack commands, then source the file
 - \$ `. CloudComputing_<Your student ID>-openrc>.sh`
- Now we can use [CLIs](#) to manage OpenStack. For example, list all users:
 - \$ `openstack user list`(take a screenshot)
- We can also manage OpenStack by using its [REST API](#).

---END---