Network Softwarization: Technologies and Enablers
OpenStack Lab
Due Date: February 12 11:59am
Total Points: 50
Winter 2019

TABLE DES MATIERES

ENVIRONMENT SETUP

To experiment with OpenStack, we are going to use DevStack, which is a set of scripts allowing to install OpenStack within a single node or a single virtual machine.

In this lab, we provide you with a virtual machine containing a pre-installed version of OpenStack (using DevStack). To set up the DevStack virtual machine, please use the following instructions:

- Download the VirtualBox image from one of the following sources:
  https://transferts.etsmtl.ca/index.php/s/WlXIN5bud5dZ5B0
  https://www.dropbox.com/s/ggr7xtz28fpzztb/DevStackOVSFinal-W2019.ova?dl=0
- Open VirtualBox and go to File menu> Import Appliance.
- Select the downloaded image and click next.
- Set at least 4096MB of memory and 1 CPU in the Appliance Settings window and click Import.
- Here are the DevStack VM credentials:
  - There are three users: root, devstack, stack
  - They have the same password: devstack
  - You can login as devstack then switch to stack using the following command:

```
sudo su - stack
```

- OpenStack credentials and service access links:
  - The default OpenStack users are: admin and demo
  - The password for both users: secret
- **Important:** Please do not work directly on your DevStack virtual machine through VirtualBox as the VM will be very slow. It is better to keep the VM running and use a secure shell to access it. The DevStack VM that you are running is connected to your physical host through a NAT, you need to use 127.0.0.1 instead of the IP address of the DevStack VM in order to access the VM from your physical host. You can ssh your VM and access to OpenStack Horizon as follows:
  - If your physical machine is using Linux or MACOS, you can access to your DevStack VM using the following command executed from a shell terminal:

    ```
    ssh devstack@127.0.0.1 -p 8091
    ```
  - If your physical machine uses Windows, you will need to use the tool putty with the following parameters:
    - IP address: 127.0.0.1
    - Port: 8091
    - User: devstack
  - The OpenStack Horizon Web interface is available from the browser of your physical host (e.g., Firefox) via: http://127.0.0.1:8090/dashboard
- To check that your OpenStack is working properly, access to the Horizon web interface and try to launch a VM instance (using m1.nano flavor, do not ask to create a new volume when choosing the source image). If the VM becomes active, it means your OpenStack is operational.

USING OPENSTACK COMMAND LINE INTERFACE

- Access to your VM through ssh as “devstack” and then access as user "stack" using the following command: `sudo su - stack`

- In order to use the OpenStack CLI, you need first to configure the administrative account, run the following commands in your shell:

```
export OS_USERNAME=admin
export OS_PASSWORD=secret
export OS_PROJECT_NAME=admin
export OS_PROJECT_DOMAIN_ID=default
export OS_USER_DOMAIN_ID=default
export OS_IDENTITY_API_VERSION=3
export OS_AUTH_URL=http://localhost/identity
```

- To access OpenStack CLI, run the following command: `openstack`

- To see available commands, type the following command: `help`

- To check the help about a particular command, the following: `<command> help`

Some examples:

```
project help
service list
network list
flavor list
user list
image list
security group list
```

- To know more about launching an instance from the command line and accessing to it through a virtual console: click here

OPENSTACK RESTFUL APIS

In order to use OpenStack Restful APIs, you need first to request an authentication token that will allow you to access particular project and OpenStack services until the token expires (in this case an Unauthorized (401) error occurs).

- To request a token, you must first issue an authentication request with a payload of credentials to OpenStack Identity:

```
curl -i \
  -H "Content-Type: application/json" \
  -d '
{ "auth": {
    "identity": {
      "methods": ["password"],
      "password": {
        "user": {
          "name": "admin",
          "domain": { "id": "default" },
          "password": "secret"
        }
      }
    },
    "scope": {
      "project": {
        "name": "admin",
        "domain": { "id": "default" }
      }
    }
  }
}' \
  http://localhost/identity/v3/auth/tokens
```

The token is provided in the *X-Subject-Token* header in the http response.

- When you send API requests, you should include the token in the X-Auth-Token header of the http request.
- You can export `OS_TOKEN` in your shell script and use the variable OS_TOKEN in your future requests:

```
export OS_TOKEN=put the token provided by the previous command here
```

- Here are some API examples using CURL :

```
#List of users:
curl -s \
  -H "X-Auth-Token:$OS_TOKEN" \
  "http://localhost/identity/v3/users" | python -mjson.tool

#List of VMs:
curl -s  -H "X-Auth-Token: $OS_TOKEN"
"http://localhost/compute/v2.1/servers" | python -mjson.tool
```

  - For more examples: Click here
  - Full OpenStack API documentation: you can now use the OpenStack APIs to launch server instances, create images, create storage containers and objects, and complete other actions in your OpenStack cloud. The full OpenStack API documentation is available at https://developer.openstack.org/api-guide/quick-start/ . You will find documentation of all OpenStack components including the Compute API to manage VM instances and Networking API v2.0 to create and manage virtual networks.

LAB INSTRUCTIONS

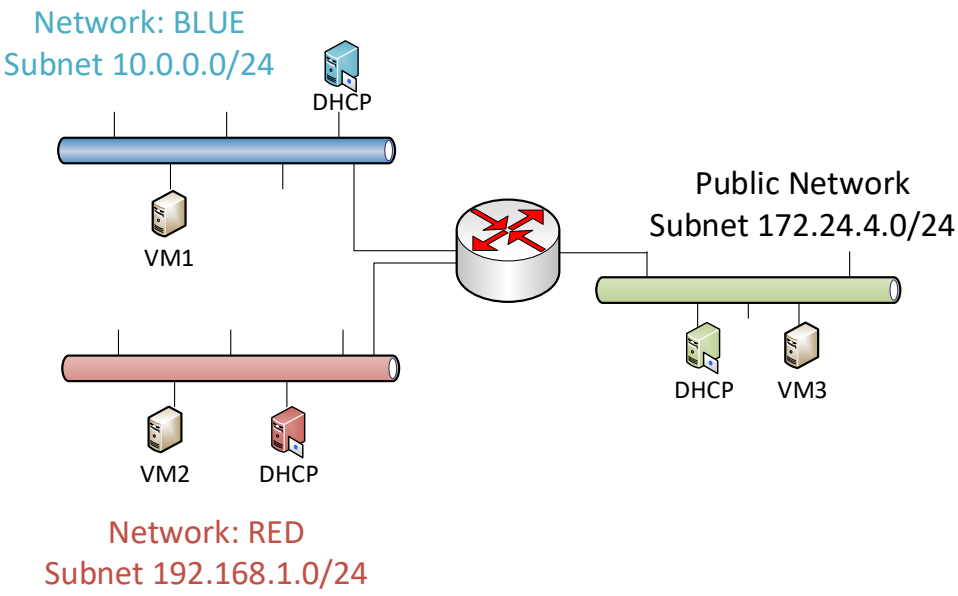
Create the virtual infrastructure provided below using OpenStack Restful APIs. The deliverables are as follows:

- Script 1: it takes as input the IP address of the DevStack VM and creates the requested topology (use a shell script and use "curl" to send http Restful APIs). The three VMs should be able to ping each other (20).

- Script 2: a script that shows the status (active or inactive) of the networks, the virtual machines and the router interfaces. The output format is json (20).

- The output of script 2 (json format). (10)

Please make sure to carefully read the FAQ Section (see below) before starting working on you scripts.



FAQ

- About Script 1:
  - Please assume that there is no public network or routers already existing in your OpenStack environment.
  - Your script should require a token and use it to run the curl commands.
  - A security group called *default* already exists in the project with basic forwarding rules (IP traffic only; no rules about ICMP are defined).
  - The project and user "admin" exist in the environment. The password is "secret"
- About the virtual machines:
  - All VMs are using *m1.nano* flavor
- Requirements for both scripts:
  - All *ids* might change from one environment to another so please make sure to not hardcode *ids*.

TROUBLESHOOTING

Check the troubleshooting online document (click here) if you face any problem with your DevStack VM or any other bug while developing your scripts. You may find there the solution for your problem.