# Lab CC-OSTK - Using OpenStack Heat

## Introduction and Prerequisites

This laboratory is to:

- Explore various capabilities of OpenStack's orchestration engine Heat
- You will recreate various virtual resources similar to what was done in CC-IAAS lab but using various Heat resource supported by the engine
- You will study the automation of scaling of your virtual machine based on specific rules governing the load on the VM

In this lab you will explore OpenStack Heat. You will orchestrate a virtual application by utilizing various resources supported by the HOT specification and the orchestration engine.

The following resources and tools are required for this laboratory session:

- Any modern web browser
- Modern text editor such as Sublime Text that are capable of editing YAML files
  - http://docs.ansible.com/ansible/latest/YAMLSyntax.html
- Any modern SSH client application
  - Windows users: https://www.ssh.com/ssh/putty/windows/puttygen
    - Putty: https://www.ssh.com/ssh/putty/download
  - Mac / Unix / Linux users: `ssh` and `ssh-keygen` commands are needed
    - Openssl packages provide these
- OpenStack Horizon dashboard: https://ned.cloudlab.zhaw.ch
- OpenStack account details. Please contact the lab assistant in case you already have not received your access credentials.
- OpenStack Heat guides:
  - https://docs.openstack.org/heat/latest/getting_started/create_a_stack.html
  - https://docs.openstack.org/heat/latest/template_guide/index.html
  - https://docs.openstack.org/heat/latest/template_guide/hot_spec.html#resources-section

The code for this lab is provided in a single zip file called `CCP1-OSTK-Lab.zip`.

## Time

The entire session will take 90 minutes.

# Task 1 – Setting-up & Launching VM via Heat

In this task you will use a VM with OpenStack command line tool preinstalled as the working environment for this lab. You will further create simple heat template files (which are YAML documents) and execute these templates using the command line tool.

## Subtask 1.1 – Prepare Your Lab Environment

- Launch a VM using instance snapshot '**`ubuntu-16.04-openstackclient`**'
  - Use **`m1.small`** flavor while launching the VM
- Assign a floating IP
- SSH into this VM using the assigned floating IP
- Verify that openstack client is indeed installed
- Download your openstack credentials file
  - Log into your account in the horizon dashboard
  - Navigate to: Project → API Access → **Download OpenStack RC File v3.0**
- Upload your credentials file into the VM
  - You can use **`sftp`** or **`scp`**[1] commands to transfer the file into the VM (see the links at the end of this document if you want to learn more about sftp or scp)
- Upload the `CCP1-OSTK-Lab.zip` file into the VM
  - Unzip the file.
- Test the CLI tool in a terminal
  - Source the credentials file: **`source <file-name>-openrc.sh`**
    - Replace *`<file-name>-openrc.sh`* with the actual filename you saved
  - Try these commands
    - **`openstack stack list`**
    - **`openstack server list`**

## Subtask 1.2 – Creating a Simple VM via Heat

In this task you will create a basic stack containing just a single VM and discover the associated floating IP allocated at runtime.

- Make sure your openstack account has a public key registered.
- Analyse the `st-12.yaml` file within `CCP1-OSTK-Lab` directory.
- Replace `YOUR-KEY-NAME-HERE` and `YOUR-NETWORK-NAME-HERE` with actual values depending on how you created these in your OpenStack account. Replace `FLOATING_NETWORK_NAME` with either `public1` or `public2`. **IMPORTANT:** Note that `YOUR-NETWORK-NAME-HERE` is either `default_internal` or `default_internal_1` - wherever you acquire your floating IP.
- Use **`openstack stack`** operations to create your stack, retrieve details and finally delete your stack
  - Use **`openstack stack --help`** to find more
  - You will be using these commands:
    - `openstack stack create`
    - `openstack stack show`
    - `openstack stack delete`

[1] To transfer to your home directory: `scp -i YOUR_PRIV_KEY ubuntu@YOUR_FIP:~`

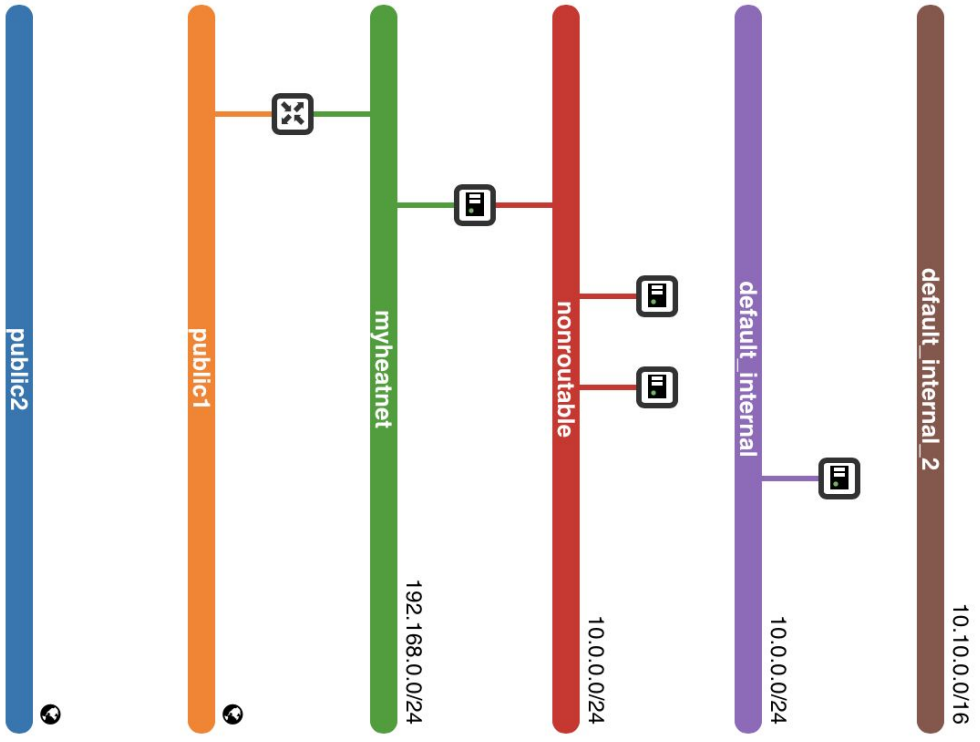
    - **Do not forget** to provide all necessary arguments that these above commands require.
- Is there another way to create a Heat stack?
- Discuss what you observe with your partner and/or lab assistants.

## Subtask 1.3 - Advance Virtual Environment Creation

In this task we will assume that you start from a completely empty environment, that is - there exists no virtual network already created and no key pair exists in your project.

- Start from the base template file '`st-13.yaml`'
- Study and analyze the template
- Discuss how it creates a new key pair, a new network, subnet, router and then a virtual machine.
- Replace `FLOATING_NETWORK_NAME` with either `public1` or `public2`. **IMPORTANT:** Note that `YOUR-NETWORK-NAME-HERE` should be connected (via router) to the same public network (`FLOATING_NETWORK_NAME`) where you acquire your floating IP.
- What does this template output?

Using this template (`st-13.yaml`) as a starting point and **extend** it to create the same setup as shown in the figure below.




- You must have **one** VM called frontend which is publicly reachable
- Two VMs as backend nodes that are **only** accessible through frontend VM
- You can name your virtual networks differently as long as the structure resembles the diagram below.
- Similarly you can choose reasonable subnet range for your networks.
    - All subnets must have an IPv4 private address range
- Ensure that **either** `public1` **or** `public2` is used consistently when making declarations about public networking

- *HINT*: If you need to enforce a particular order in creation of resources, use **`depends_on`** feature.
- Test whether you can SSH into your frontend VM. Can you ping the internal VMs? If you cannot SSH suggest a solution via Heat.

# Task 2 - Studying Auto-Scaling

Heat support **AutoScalingGroup** which allows trigger based rules to scale out or scale in. This task explores this functionality.

## Subtask 2.1 – Understand the Template File with Built-in Autoscaling

In this task you will analyze the sample heat template with autoscaling enabled, and will execute the stack.

- Change into `st-21` directory.
- Analyze the three template files in it, discuss the various elements with your partner or lab assistants.
    - Describe how the auto scaling logic is encoded in the template?

# Task 3 - Cleanup - Stop the Bills!

**IMPORTANT**: At the end of the lab session:

- **Delete** all VMs, volumes, security group rules that were created by your team
- **Release** all floating IPs and other resources back to the central pool for others to use

# Additional Documentation

Heat documentation and useful resources can be found on the following pages:

- SFTP: https://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server
- SCP: https://www.tecmint.com/scp-commands-examples/