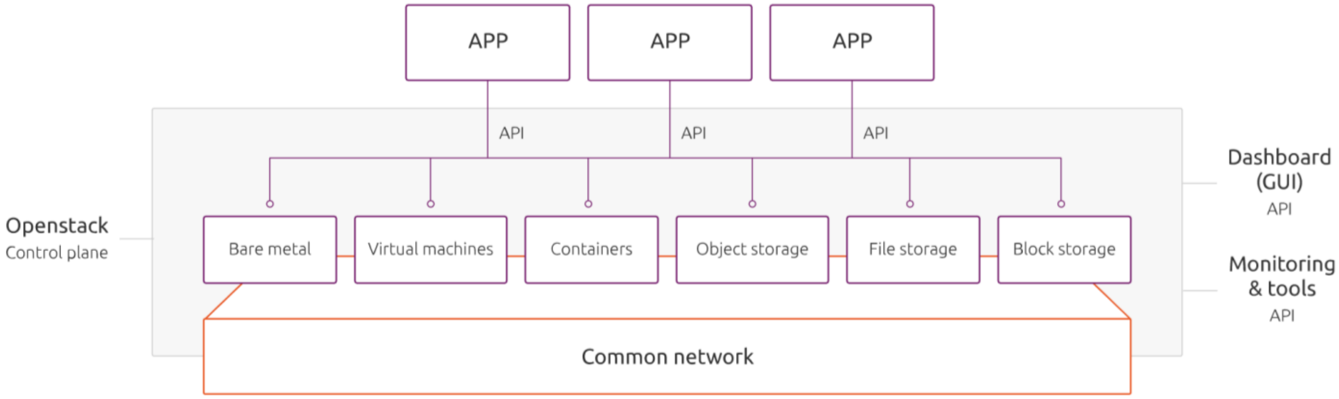
Cloud Operations

Cloud Computing Course Project, *OpenStack Engineering*

Introduction

Simply put, cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet ("the cloud") to offer faster innovation, flexible resources, and economies of scale. You *typically* pay only for cloud services you use, helping you lower your operating costs, run your infrastructure more efficiently, and scale as your business needs change. Cloud computing is a big shift from the traditional way businesses think about IT resources. Not all clouds are the same and not one type of cloud computing is right for everyone. Several different models, types, and services have evolved to help offer the right solution for your needs.

OpenStack is an open source cloud computing platform that allows businesses to control large pools of compute, storage and networking in a data center. The fact that OpenStack is open source means that anyone that chooses to use it, can access the source code, make changes, and share these with the community. One of the key benefits of this model is that the source code can then be checked by a much larger set of people than proprietary code, which is restricted to its owners. Governed by the OpenStack Foundation, there are more than 34,000 individual contributors and over 550 companies that participate in the project.



Outline

In this project, we'd like you to get familiarized with the OpenStack Lab deployment, OpenStack services and gain experience with the OpenStack GUI (Dashboard, aka. Horizon) and the CLI interfaces. By completing this project, you'll get to know some of OpenStack services a little better.

Certainly, there is a lot more to learn and this project is designed to just put you in the right mind set.

Description

Initially, we'd like you to set up a private cloud using OpenStack-Ansible All-In-One, described here. You need to deploy OpenStack on its latest stable release (currently, *Victoria*) and your deployment must include OpenStack base services plus Octavia and Heat. After a successful deployment, try to log in with the *admin* credentials (password is provided in the *keystone_auth_admin_password* field of *user_secrets.yml* file) and check *System Information* for the **status** of OpenStack services.

You own a *private* cloud and clouds are limitless in nature, but obviously that doesn't mean you have actual infinite resources! So, to protect your *Quality of Service*, make sure that each project in your cloud cannot consume more resources than the following thresholds: *2 Instances, 4 CPU Cores* and *4 GBs of RAM*.

Now, you need to create a project called **cloud** and a user with the combination of your names and set the *cloud* project as its **Primary Project** with the role *member*. Before logging out from the *admin* user, specify the custom quota for this project as follows: *6 Instances, 8 Cores of CPU* and *12 GBs of RAM*.

Our *ultimate* goal is to orchestrate the deployment of two applications in our cloud. We want to deploy them into a **non-containerized** three-node cluster with a dedicated node for the database. We will expose our apps to the outside through a load-balancer. The details are specified in the following sections.

Networks

Start off by logging in to your new user account; create a *nonshared* network called **auth-app** with the CIDR of **192.168.65.0/26**. Then, create a *router* called **cloud-router** that connects the *auth-app* network to the outside world. Note that our instances should not be accessible to the public at all. The load balancer and the instances need to be on the same network to communicate; however, the load-balancer must also serve our apps to the outside world.

General Specifications

You are free to use your choice of Linux distro for either of the apps or database nodes. If you are new to the Linux, we recommend the latest stable release of **Ubuntu Server** (Currently, *Ubuntu Server 20.04*). Each app node should have *1 CPU Core, 1 GB of RAM and 5 GBs of Storage*. The database node has the same configurations except a *Storage of 20 GBs*.
On the creation of the instances, add the OpenStack host *root* **SSH key** and **set password** for the *root* user. The root password could be used when you cannot access the instance via the network; in that case, you may log in via the console. Make sure that the SSH service on instances allows incoming requests only through SSH key.

Database

One of the two services that you are going to deploy needs a **Mongo** database. Create an instance, **install** and **secure** the Mongo database. It needs to **start on boot** and **bind** to its IP address on the *auth-app* network. For reasons you'll find out later, this instance needs to have a **static IP** address. Finally, make sure that only requests from the *auth-app* network reaches your Mongo database (*do **not** set rules inside the machine*).

Applications

The two applications that you will deploy are **Authentiq** and Basic System Info. The first is a *JWT-based* authentication service that depends on a Mongo database and the second displays a little information about the host that it is running on (it comes handy when dealing with the load-balancer). Both of these services are written in **node.js** and it is a good idea to start by installing **node** and **npm**. The Authentiq app requires some configurations which are explained thoroughly in its *ReadMe* page. You need to configure them to **start on boot** and they should **bind to 127.0.0.1 only**. Next, expose the apps **only** to the load-balancer via **domain** names by using a **reverse-proxy** software. If you are starter, we recommend **Nginx**. Publish the *System Info* app on the http://fumcloud.pro domain and the *Authentiq* app on http://api.fumcloud.pro subdomain. Set the Authentiq virtual server as **default** (figure out why this is necessary). Finally, note that we need **three** replicas of this node, configured in the same way.

Load-balancer

Create a load-balancer that **listens** on port 80 for *HTTP requests* and forwards them towards our cluster using **round-robin** algorithm with **equal weights**. The load-balancer should **monitor** the state of the cluster using the **Heartbeat API** of the *Authentiq* app.

Orchestration

So far, you have done everything manually. Now, it is time to use the OpenStack's orchestration engine, AKA **Heat**, and deploy your app into your cloud from scratch with the push of a button. Your **main task in this project** is writing a Heat template that satisfies every requirement mentioned earlier. Your template should include the creation and configuration of **everything except** the parts before the *Network* section and uploading a raw OS. It is very important to write your template **as modular as possible**. Please note that although the Heat graphical *Template Generator* is a good point to start, it has many **limitations** and therefore you should use the **OpenStack CLI** to deploy your template. To use the OpenStack CLI, prepare and *RC* file for your user and connect to the *utility container*. To learn about the **Resource Types** in a Heat templates refer to this document.

Access the Cluster

In a production environment, the IP address of the load-balancer is a *valid* IP address published on the internet and you can easily access your apps. However, in this lab deployment, the public network IP addresses are **not really public** and can only be accessed on the **OpenStack host** machine. You need to figure out a way to access your apps with their *domain names* from **your computer**. There are a lot of solutions to this requirement, but if you do not have a preference, I'd recommend **SSH Dynamic Port Forwarding** and a **static DNS record** on *OpenStack Host* machine.

Extra Credit

You may perform the following tasks to earn extra credits:

- ✓ Heat Orchestration engine supports **Horizontal Auto Scaling**. We want our app to **scale up to 5** nodes, if the **average CPU** used in the last collect time is **above 80%**, and **scale down to 3** if it has been idle **below 15%** of average CPU usage. Note that this feature requires **OpenStack Telemetry** stack (Aodh, Ceilometer and Gnocchi) to be deployed. There are a lot of tools that can create loads on your instances, but we recommend *ab* (Apache Benchmarking tool) and *stress-ng*, if you do not have a preference.
- ✓ **OpenStack Trove** is *Database as a Service* (DBaaS) that supports MongoDB. Deploy this module and create a managed database instead of configuring your own database node.
- ✓ Deploy a **near-production** 2-node OpenStack using ansible (*controller, compute*). Get started here!

Submission

Students must pair up for this project. Groups of more than two people are not allowed. Each group needs to create a private repository on GitHub and add SayidHosseini as a *Developer*. You need to use git while developing your template and commit every step of the way; each group member needs to commit using his/her own account. Bear in mind that commits will be reviewed! Your project should be ready by the end of ***Tir 25th, 1400***. There will be a virtual delivery at 4 P.M on Tir 26th, and both group members are *required* to attend. Groups willing to deliver the project are required to send me an email, including names and student numbers by the end of the Tir 24th. A schedule will be published, stating the exact time of your delivery.

Please note that the project will not be accepted after the deadline and the deadline will not be extended!

Get your Resource

Installing OpenStack with ansible has its Pros & Cons. The heavy resources required to deploy OpenStack is a discouraging point in the Cons area and has been one of our biggest challenges, since we started teaching OpenStack in the Cloud Computing course of Ferdowsi University.
Fortunately, this semester one of the well-known companies in the area of *hosting* and *infrastructure* has become the sponsor to this project and provided the required resources. This is a good starting point to bridge the gap between the university and the industry. We appreciate their gesture and hope that the time you spend on this project be as fruitful as possible.
Every group gets a coupon for a machine with *8 CPU Cores, 16 GBs of RAM and 160 GB of Storage*. Please contact me via email, specifying your *names*, *student numbers* and the *email address of your account* on their panel.

Good Luck!

SayidHosseini

in SayidHosseini

References

https://azure.microsoft.com/en-us/overview/what-is-cloud-computing
https://ubuntu.com/openstack/what-is-openstack
https://docs.openstack.org/openstack-ansible/victoria/user/aio/quickstart.html
https://docs.openstack.org/image-guide/obtain-images.html
https://docs.openstack.org/nova/victoria/user/quotas.html
https://docs.openstack.org/arch-design/design-compute/design-compute-overcommit.html
https://docs.openstack.org/security-guide/identity/tokens.html
https://docs.openstack.org/horizon/victoria/user/index.html
https://docs.openstack.org/project-deploy-guide/openstack-ansible/victoria
https://docs.openstack.org/heat/victoria/template_guide/openstack.html