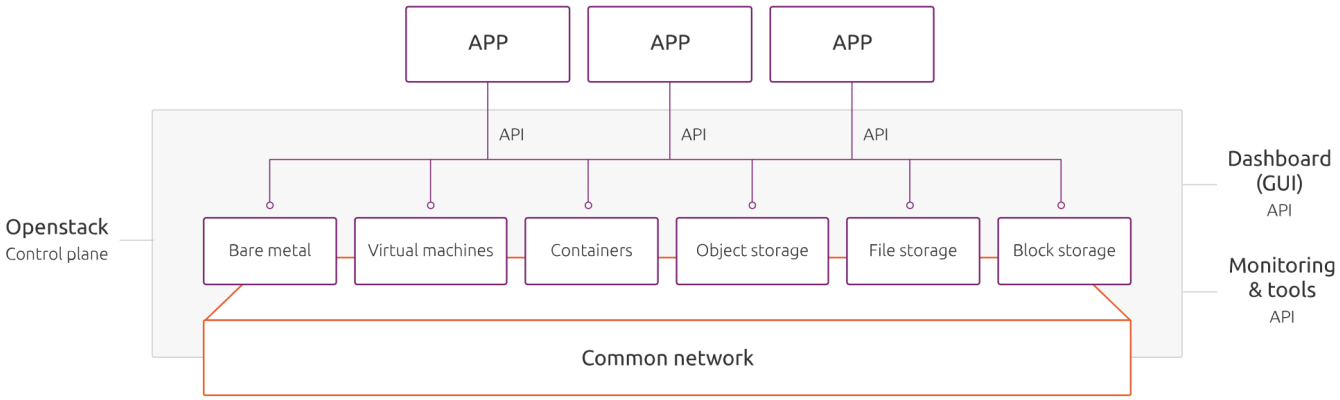
Cloud Administration

Cloud Computing Course Project, *OpenStack Admin*
Submission Deadline: *Tir 25th, 1399*

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 and 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.

Cloud Administration

Cloud Computing Course Project, *OpenStack Admin*
Submission Deadline: *Tir 25th, 1399*

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, *Ussuri*) and your deployment must include OpenStack base services plus telemetry (*Aodh, Gnocchi, Ceilometer*).

On Graphical User Interface

After a successful deployment, you need to login with the *admin* user and the password that is provided in the *user_secrets.yml* file (*keystone_auth_admin_password*). Then, find an *Ubuntu 20.04* cloud image (you can also make one) and upload it to your private cloud using the Horizon Dashboard with the following conditions:

- Name it "Ubuntu 20.04 LTS"
- Attach a metadata key-value pair {"os_type": "Linux Ubuntu"}
- Make it public and accessible to every user in the cloud
- Make sure that no instance with less than 512 MB of RAM can be created from this image
- Make sure that no instance with less than 3 GB of Storage can be created from this image

Although you now own a *private* cloud and clouds are limitless in nature, that doesn't mean you have actual limitless resources! So, to protect your *Quality of Service*, you need to make sure that each project in your cloud cannot consume more resources than the following thresholds: *5 Instances, 10 Cores of CPU* and *5 GB of RAM*.

Now, go ahead and create a project called **CloudPro**. After that create a user with the combination of your family names and set *CloudPro* as its *Primary Project* with the role *member*. As a final step before logging out from the *admin* user, specify the dedicated quota for this project as follows: *3 Instances, 4 Cores of CPU* and *4 GB of RAM*.

After you logged in with your user account, create two networks called *prv-10* and *prv-20* with CIDR of **10.10.10.0/24** and **20.20.20.0/24**, respectively. Then, create a *router* called **CloudPro_Router** that connects these two networks to the *outside world*. Finally, create 2 instances with the following attributes:

MyUbuntu

- **Source:** Ubuntu 20.04 LTS
- **VCPUs:** 2
- **RAM:** 2048 MB
- **Disk:** 5 GB
- **Network:** prv-10 (10.10.10.10)

MyCirrOS

- **Source:** CirrOS 0.5.1
- **VCPUs:** 1
- **RAM:** 64 MB
- **Disk:** 1 GB
- **Network:** prv-20 (20.20.20.20)

Once *MyUbuntu* instance is up, login through the **console** and install *docker* and *docker-compose* on it. Note that your instances should be accessible from every where on *ICMP* protocol (ping). However, the *MyUbuntu* instance should only be available via *SSH* to the OpenStack machine; similarly, the *MyCirrOS* instance should have *SSH* enabled only to the *MyUbuntu* instance. Your instances should only allow *key-based authentication* for *SSH*.
Finally, after you are all done setting up, create a snapshot from your instances.

On Command Line Interface

What you have done so far were mainly concentrated on the usage of OpenStack Dashboard. In spite of the fact that getting started on the *Horizon* is a good beginning point, it is not enough to move forward. This is because **not** all features of OpenStack and their detailed configurations are included in the graphical user interface. There are several features that are **only** available through OpenStack APIs and OpenStack CLI tools.
To this end, you are required to write a *Bash Script* that could do the following tasks by using the OpenStack CLI tools:

- Display **version**, **command format** and **developer information**
- Accept the location of *openrc* file and save it for future use as **authentication**
- Display **proper errors** to incorrect usage of commands
- When asked, display a list of user's instances and accept user's choice to **delete** an instance
- When asked, walk the user through the process of **creating** an instance by accepting user's choice on the name and the password of the instance, list of available flavors, images, networks, keypairs and security groups (multiple security groups should be selectable)

Note that your script needs to have the ability to be run from anywhere inside the **utility container**.

For your final task in this section, you are required to modify two of the OpenStack configurations:
Normally, clouds use a technique called Over Provisioning to increase utilization of infrastructure and provide service to more users; Of course, this comes with the cost of reducing the performance of instances. OpenStack has built-in support for this technique and overprovisions CPU and RAM with the ratio of 16:1 and 1.5:1 by default, respectively. Obviously, it is not a good idea to overprovision your hard disk; since it carries the risk of system crash. However, for the sake of experiment, we want you to enable overprovisioning for hard disk with the ratio of 2:1.

You may have noticed that sometimes you are logged out automatically. This is because OpenStack uses a Token-based authentication and the token you own gets expired; as a consequence, you are redirected to the login page. Let's fix that! Make sure tokens stay valid for **24 hours**.

Extra Credit

You may perform the following tasks to earn extra credit:

- ✓ **CloudKitty** is a *Rating-as-a-Service* project for OpenStack. Clearly, billing on a *Pay-as-you-Go* basis is a fundamental requirement of business clouds. *CloudKitty* provides **cost** of resources based on OpenStack telemetry services and predefined resource-cost Hash Maps. Fortunately, The OpenStack-Ansible project supports the CloudKitty ansible role. However, the *playbooks* are not included in the project by default. Deploy CloudKitty using OpenStack-Ansible.
- ✓ Deploy a **near-production** 2-node OpenStack using ansible (*controller, compute*). Get started here!

Submission

Students can pair up for this project. Groups of more than 2 people are not allowed. Each group needs to create a private repository on GitLab and add *SayidHosseini* as a Developer; Your script(s) need to be developed using git and each group member needs to commit codes using his/her own GitLab account. Bear in mind that commits will be reviewed! Your script(s) should be ready by the end of ***Tir 25th, 1399***. There will be a virtual delivery on Tir 26th, 1399 at 12 P.M, and both group members are *required* to attend. Groups willing to deliver the project need to *send me an email*, including names and student IDs by the end of the Tir 25th. 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 will not be extended!

Good Luck!

SayidHosseini

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/ussuri/user/aio/quickstart.html
- https://docs.openstack.org/image-guide/obtain-images.html
- https://docs.openstack.org/nova/ussuri/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/ussuri/user/index.html
- https://docs.openstack.org/openstack-ansible-os_cloudkitty/ussuri/
- https://docs.openstack.org/project-deploy-guide/openstack-ansible/ussuri/