Basic operation introduction

Nectar as a IaaS(Infrastructure) service provider provided a lot basic control methodologies to help user control and maintain their applications. In this project, services used could be categorized into six parts, create instances on the cloud, create volume on the cloud and attach volume to instance, volume backup and recovery from snapshots, security group design and instances, volumes and snapshots removal.

Create instance

Instances are the most important element on the cloud. Users could create different types of instances based on the memory and number of VCUPs on Nectar to meet their requirements. Some features like the location, operation systems also could be defined by the user. Before to launch an instance, security key pairs should be created to ensure security.

Create volume

Each instance is created with certain size of virtual read only memory, but sometimes for specific applications the size is not enough. For more important reason, instances on the cloud cannot ensure the stability, important data should be stored or backup in somewhere else. Volume on the cloud is the solution to tackle these problem. With volumes, data storage are scalable and more stable on the cloud. Volumes are working as external disks so it can be plugged in and out from an instance.

Volume backup and recovery

To ensure the data consistency and increase the data error tolerance ability, volumes backup is necessary in the system. On the Nectar cloud, users can create volume snapshots to archive volume data into smaller piece as data backup. Snapshots are created on different time point to record the data states. Not only the volume but also the instances can be recorded as snapshots, however the more important data are tweets compared with instance. When the volume is damaged or things going wrong with the cloud, users could use snapshots to recover data.

Security group design

Security is of importance on every cloud application. With an application connected to the internet, one of the common method to ensure network security is to set up fire walls. Users can set up and design their own security groups to decide which port and which protocol could be used within the group. So that users can defined which instance could be accessed by the internet or only by the intranet. Typically, instances used as databases and analyses nodes should only be accessed by the intranet while the web application node could be accessed by the internet.

Delete instances, volumes and snapshots

The system also need the service to delete instances, volumes and snapshots to reset the cloud. Moreover, due to the insufficient size of volume storage or instance numbers, to put the right resource into the right place some of the volume or instance should be deleted. Snapshots which is no longer needed also need to be deleted.

Implementation methods

Nectar is built based on the Openstack framework, two ways are mainly used in the cloud build processes. The first one is build the cloud with the web UI and the second one is use the APIs to control the cloud.

Nectar dash board implementation

The cloud infrastructure could be easily build on the Nectar dash board UI. Basically most of the operation could be done with the user interface. This is an lazy and quick solution and no need for much experience on programming. However the implementation process is not reusable and less of documentation make the user confused about how the infrastructure is built.

API control

The another solution is use the API provided by Nectar to build the infrastructure with programming language. A range of programming language could be used in the implementation process, users can choose anyone he or she like. This solution require the builder have some experience on programming but it is more professional and reusable. Other system user could check the scrips to understand the detail of infrastructure.

In this project both methods are used in the implementation process. In the security group design and secret key creating operation dash board is used while other operation are delivered by python programming language.

Table 1

Boto controller class design

In this project, to increase the reusability, maintainability and scalability of the scripts, a BotoController class is designed. To use this class users need to download credential file from Nectar to be able to control the cloud. To parameter are used to create a Botocontroller class which are access_key_id and secret_access_key.

Based on the system architecture design, instances for different server should have different computing power to allocate limited resource sufficiently. The type of servers could be divided into two types, the first one is database and harvester server and the other one is web application and tweets analyser server. Database and harvester server need less computing power compared with another one meanwhile database need volume to restore the data. So that the instance creation method is separated for different types of servers with different computing power and whether with volume. Moreover, even though current system does not include Spark, creation for spark instance is still created for future use. In the instance created process, tags are added into every instance to describe which type of server it is and assigned unique name for it.

Along with the instance creation, the class provide a function to export IP address of each instance into a file as inventory file to collaborate with Ansible.

Delete functions for instances, volumes and snapshots also provided in the class. A clean up function also provided in the class to delete all the elements on the cloud.

The class also provided functions to simply create snapshots and recover volume snapshots functions.

Class diagram

Python programs

Before the deployment of the system, some work have to be done on the Nectar dash board UI. As mentioned before, the access_key_id and the secret_access_key should be recorded to be able to control the cloud. User also need to create several key pair for the cloud to be used in different types of instances. To further up increase the security level, security groups also need to be predesigned for each type of server.

There are six programs to control the cloud, the first one is used to deploy the cloud, the second one is used to create instances to be able to increase scalability, the third one is used to delete an element from the cloud, the forth one is used to create a snapshot of a volume, and the last one is used to create a new volume with snapshot.

Every program is work with an specific configuration file to pass the arguments to the program.

Table 2