Sharded MongoDB Deployment on Cloudmesh

- By Rahul Singhania

Follow this guide to setup and install cloudmesh and then run python script to install sharded mongoDB.

1. Installing Cloudmesh on the VM
2. Key Management
3. Introduction to Cloudmesh
4. Firewall configuration for MongoDB access
5. Running python script : To Install Sharded MongoDB
6. Future Improvement of mongoDB installation script
7. Troubleshoot guide
8. Other helper scripts

1. Install Cloudmesh on the VM
   Setting up Cloudmesh on a VM is an especially convenient way during development and testing. Follow the following link to install cloudmesh on VM:

   http://cloudmesh.github.io/introduction_to_cloud_computing/cloudmesh/setup/setup_openstack.html

- Mainly you have to have an account on FutureSystems that allows use of resources and services. It allows VM creation, key management and other services.
- Make sure to keep VM image snapshot to fire new VMs when needed :
  Follow this guide to create a snapshot of image
  http://cloudmesh.github.io/introduction_to_cloud_computing/iaas/openstack.html#make-a-snapshot-of-an-instance

- VM security is maintained using ssh keys. Please look at the Key Management section for issues with key management.

2. Key Management
   - FutureSystems :
     Web portal to manage keys. This web portal contains ssh keys to allow access to openstack using nova client.

   - Openstack :
     Cloud where all the futuresystems services and resources can be accessed using nova client. As seen from the tutorial above, ssh key is created after logging into the openstack. This key is used to create new VMs.

     Command: **`nova keypair-list`**
     Above command lists all the keys present in Openstack.

- Cloudmesh :
  A library to support automation of VM creation and other tasks in cloud. Cloudmesh can be installed locally or can be installed on the VM created using openstack. *Cloudmesh has its own key, that key needs to be uploaded to the futureSystems and that key must be present in openstack as well.*

  After the cloudmesh key is uploaded to the FutureSystems account it is automatically added to the Openstack but in some cases due to duplicity it is not added and hence the key needs to be added manually.

  Follow the following steps to make sure your key is correctly uploaded to all three places:
  1. Generate ssh key for cloudmesh : Can have multiple keys in cloudmesh
  2. Upload the cloudmesh ssh key to FutureSystems
  3. Login to openstack and run command: "nova keypair-list" and make sure the same key is present in openstack, if not add it manually.

3. Introduction to Cloudmesh
   Cloudmesh is an important component to deliver a software-defined system – encompassing virtualized and bare-metal infrastructure, networks, application, systems and platform software – with a unifying goal of providing Cloud Testbeds as a Service (CTaaS). Cloudmesh federates a number of resources from academia and industry. This includes existing FutureSystems, Amazon Web Services, Azure, HP Cloud, Karlsruhe using various technologies. [1]
   [1] http://cloudmesh.github.io/introduction_to_cloud_computing/cloudmesh/overview.html

4. Firewall configuration for MongoDB access:
   All firewall gateways are managed through openstack. All the VMs get updated with the firewall rule automatically after the rule is added to openstack. To open the required port perform the following steps:
   - Login to openstack and then run following commands:
     nova secgroup-add-rule default tcp 27017 27017 0.0.0.0/0
     nova secgroup-add-rule default tcp 27018 27018 0.0.0.0/0
     nova secgroup-add-rule default tcp 27019 27019 0.0.0.0/0

   - MongoDB uses following ports 27017, 27018, 27019

5. Running python script : To Install Sharded MongoDB
   After cloudmesh is installed and Keys are setup correctly. Run the python script to install sharded mongoDB. Follow the following steps to run the python script:

- Login to the VM where your cloudmesh is running
- Clone the mongo github repo in HOME dir: (https://github.com/cloudmesh/mongo)
  cd /~
  git clone https://github.com/cloudmesh/mongo.git

- Go to bin directory and make the mongo-sharded.yaml file to configure it according to your need and requirement
  vi mongo/bin/mongo-sharded.yaml

- Run the python script : mongo-sharded.py
  cd mongo/bin
  python mongo-sharded.py

6. Future Improvement of mongoDB installation script

- Current installation of MongoDB installation is not secure. It needs to be secured through the use of automated script.

7. Troubleshoot guide
   - Restarting Cloudmesh server
     Go to home Dir : `cd ~`
     Activate virtual ENV : `source ~/ENV/bin/activate`
     `Running cloudmesh server in background: nohup fab server.start`

     fab server.start restarts the server and deleted any unused instances.

   - Useful links while working with scripting:
     Shell Commands Documentation:
     http://cloudmesh.github.io/cloudmesh/man/man.html#cluster

     Shell API tutorial
     http://cloudmesh.github.io/introduction_to_cloud_computing/cloudmesh/shell/_vm-shell.html

     Mesh API tutorial
     http://cloudmesh.github.io/introduction_to_cloud_computing/cloudmesh/api/_vm_api.html

8. Other helper scripts

- *json-dec.py* :
  Example to extract information from json formatted string

- *mongo-db-mesh.py* :
  This script shows the usage of cloudmesh mesh api. It fires 9 VM and installs mongoDB on them and sets up config and router server. It requires manual setup to finally start using the mongoDB as keys are not added automatically for login into each other which is improved using cluster command in the main script.

- *mongo*-gregor.py
  It uses cluster command to fire VMs and setup public ip for all of them. It uses yaml file (mongo-gregor.yaml) for getting config information