**Creating an instance of LDAP server using OpenStack:**

Step 1: Allocating Floating IP to OpenStack:

We need to navigate to the following path:

**Project** -> **Compute** -> **Access & Security** -> **Floating IPs and then select the Allocate IP To Project**

Step 2: Creating an OpenStack Image:

The image file can be used directly from the HTTP link or we can upload it directly on OpenStack cloud.

To create an image, we need to go to OpenStack web panel and navigate to

**Project** -> **Compute** -> **Images** and then select the **Create Image** button when done.

Step 3: Launch an Image Instance in OpenStack:

Now we can run the instance based on the image created in previous step.

Navigate to **Project** -> **Instances** and select **Launch Instance** button and a new window will appear.

Fill the information accordingly followed by **Next** buttons and finally click on **Launch Instance.**

Step 4: Once the instance has been started, select the right arrow from the **Create Snapshot** menu button and choose **Associate Floating IP**.

Select one of the floating IP created earlier and select the **Associate** button to make the instance reachable from our internal LAN.

Step 5: Test the network connectivity by issuing a **ping** command against the instance floating IP address from a remote computer in our LAN.

Step 6: Use the Instance **View Log** utility to obtain **Cirros** default credentials.

Step 7: By default, no DNS name servers will be allocated from the internal network DHCP server for our virtual machine. This problem leads to domain connectivity issues from instance counterpart.

To solve this issue, first stop the instance and go to **Project** -> **Network** -> **Networks** and edit the proper subnet by selecting the **Subnet Details** button.

Add the required DNS name servers, save the configuration, start and connect to the instance console to test if the new configuration has been applied by pinging a domain name.

Step 8: In case you have limited physical resources in our infrastructure and some of our instances refuse to start, edit the following line from nova configuration file and restart the machine to apply changes.

**# vi /etc/nova/nova.conf**

Change the following line to look like this:

**ram\_allocation\_ratio=3.0**

The LDAP configuration is presented in the playbook format in the last section of this document.

We can add techops\_dba group in the **/etc/security/access.conf** by issuing the following commands:

**vi /etc/security/access.conf**



Now we need to add the following line to add techops\_dba group:

**+ : techops\_dba\_access : ALL**



In the same way we need to add techops\_dba group in the /etc/sudoers file by issuing the following commands:

**visudo**



Now we need to add the following line to add techops\_dba group:

**%techops\_dba ALL=(ALL:ALL) ALL**



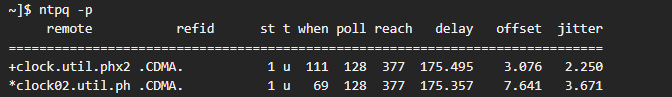
To check if **ntpd** is configured to run at system start, issue the following command:

**chkconfig --list ntpd**



To check if **ntpd** is running, issue the following command:

**ntpq -p**



The **st** column in above screenshot refers to the Stratum of Server.

We can also get a brief status report from ntpd by issuing the following command:

**ntpstat**

We can get the load average for the host by issuing the following command:

uptime

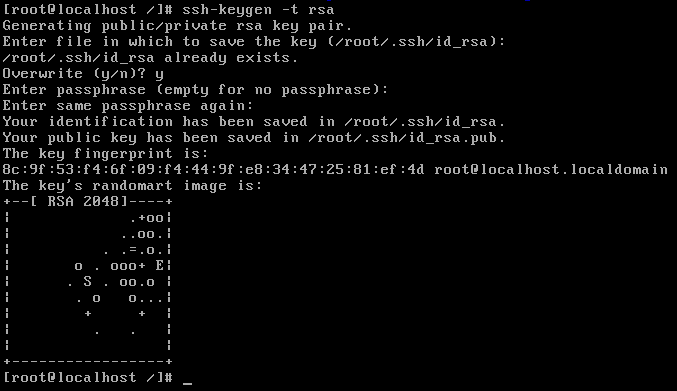


My second host name is **backup**.

For SSH connectivity we require a key pair. To establish a connection between two hosts and also indicated that only second host can access first host then for this we need to generate a key pair in the first host and we need to copy all the public and private keys to the second host i.e., backup.

The key pair can be generated in first host by issuing the following command:

**ssh-keygen -t rsa**



This command creates a public and a private key pair in the **/root/.ssh/** directory:

**Id\_rsa**

**Id\_rsa.pub**



We will send these files to the other server. Copy the files to the other server by running the following:

**scp -r /root/.ssh\* backup:/root/**



Automated testing can be configured by applying test cases accordingly.

In the first step, we need to check the correctness of LDAP server.

In the second step, we need to check whether the group techops\_dba is added to the corresponding files or not.

In the third step, we need to check the result of NTP stratum and the acceptable load average value.

In the fourth step, we need to check the SSH connectivity between two hosts and, we need to check whether the second host can access first host or not.

**The following is the Playblook:**

---

#The domain prefix for ldap

openldap\_server\_domain\_name: project.com

#This is the password for admin for openldap

openldap\_server\_rootpw: passme

#The self signed ssl parameters

openldap\_server\_country: US

openldap\_server\_state: Illinois

openldap\_server\_location: Chicago

openldap\_server\_organization: IT

openldap\_server\_enable\_ssl: true

#The ldif file

openldap\_server\_ldif: domain.ldif

- hosts: ahmed

+ sudo: true

roles:

- role: ahmed.openldap\_server

openldap\_server\_domain\_name: project.com

- name: Manage Network Time Protocol service

hosts: [ 'ahmed', 'ahmed\_service\_ntp' ]

become: True

roles:

- role: ahmed.ferm

tags: [ 'role::ferm' ]

ferm\_\_dependent\_rules:

- '{{ ntp\_\_ferm\_\_dependent\_rules }}'

- role: ahmed.ntp

tags: [ 'role::ntp' ]

- hosts: ahmed

roles:

- LDAP server

- techops\_dba group

- NTP result

- ssh connectivity

tasks:

- name: Check LDAP server

ansible\_unit\_test: True

check\_mode: yes

- name: Add techops\_dba group

ansible\_unit\_test: True

check\_mode: yes

- name: check NTP result

command: ntpstat

ansible\_unit\_test: True

check\_mode: yes

- name: Check SSH connectivity

ansible\_unit\_test: True

command: ping ahmed

check\_mode: yes