**Ansible set up with AWS instances**

Step1 : we have to go amazon management console website.

Step2 : if you have credentials just sign in or else we have to sign up with new id and password.

Step3: so we are inside of aws ec2 instance dashboard.

Step4 : first we have to set up our location details with “South Asia - Mumbai”.

Step5: We have to click on EC2 instance Dashboard under Services tab.

Step6: We have to click on Launch Instance under Create instance section.

Step7: Choose AMI tab -Select Redhat Enterprose Linux.

Step8 : Choose Instance type – General purpose – t2.micro (free tier available)

Step9: Configure Instance – Number of Instance = 4

Step10 : Add Storage : 10 gb for each one instance

Step 11: Add Tags : Name –nodes

Step12 : Security : mysecurity for group name and for description.

Select ssh and source anywhere

Select http and source anywhere

Select https and source anywhere

Step13 : select Launch and Review.

Step14: Select create new pair and give keypaiir value and download it into your system.

Step15 : Then Launch finally.

Here you can change your instances names what ever you want. Like

Master, node1, node2, node3.

Now we are able to see all created 4 instances with running status.

Now we have to assign constant public ips. For that we have to create same 4 elastic IPs

Step16: Then go to click on elastic IPs tab in LHS menu bar.

Step17: We have to create 4 elastic IPs.

Step18: We have to assign one ip address to one instance name.

Step19 : If windows go to command prompt and go to key pair downloaded file location. If Ubuntu go to terminal window and change to key pair downloaded location.

Step20 : Now go to AWS instances and select first instance and click on connect.

Dailog box will get open and just copy the example ssh key and paste in windows command prompt or Ubuntu terminal window and enter. So that you will be able to login selected instance via remotely.

Step21: Same thing needs to be done for all 4 instances.

Step22 : so now you are just logined with ec2 user in all 4 instances. Now you can become root user from here using command called “ sudo su – “

So till now we are able to login all 4 instances with ec2 user with downloaded key pair.

So now we want to login all 4 instances with new user and without using downloaded keypair file.

For this we have to the following process.

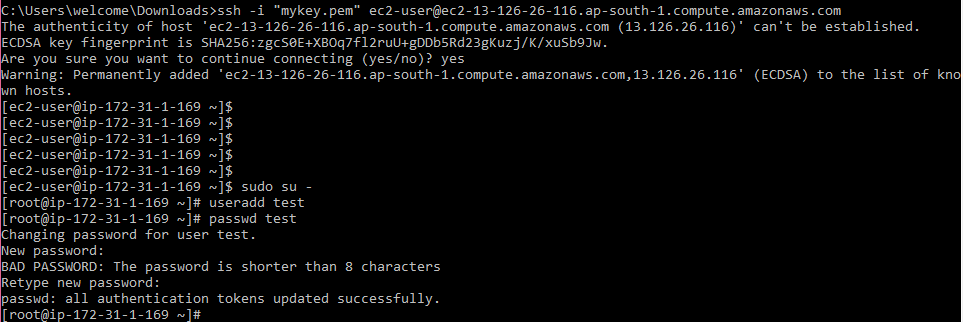
Step23 : go to each instance terminal and create one user and same for all 4 instances like below

useradd <username>

passwd <username>

enter password 2 times

you will get meassgae saying successfully details updated.



This step can be done on all 4 created instances terminal windows.

Step24: So after creating new user on all 4 instances we have to one more step for login instance from any where without pair key.

We have to go each terminal and go to below location.

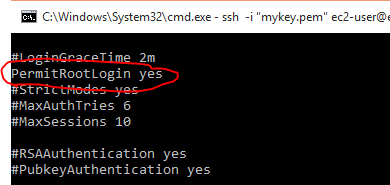
cd /etc/ssh

ls –lrt

vi sshd\_config

we have to take out “#” symbol infront of 2 variable values

1. Permitrootlogin
2. Passwordauthentication





:wq ------ to save the sshd\_config file

Then we have to restart the sshd service using below command

“systemctl restart sshd”

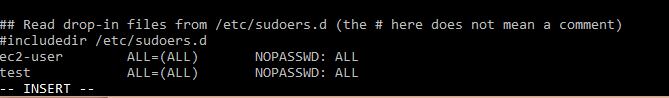
And test user has been added into sudoers file for logging to root accesss via test user like ec2 user to root.

Goto this path

cd /etc/ssh

vi sshd\_config

Add test user details under ec2 user same like ec2 user like below picture



:wq!

Step25: Above step24 needs to be repeated for all 4 instances.

Step26: Now you can go any where means from any system you are able to login these 4 instances without downloaded key pair file. By using below command.

1. open terminal window from any system.

2. type

ssh <username>@<public IP of instance>

we have to give password here. So that you are able to login now

this step can be repeated for all 4 instances to check connection for instances.

===================================================================================

We just creates all 4 instances in AWS and did set up to login user test from anywhere making some configuration. Now we are going to generate keys and sharing across the environment servers so that we can login with out password from any server, why because ansible will work on password less authentication.

We have to note down our server details like this :

Master server

Public IP:

Private IP:

Node1 server

Public IP:

Private IP:

Node2 server

Public IP:

Private P:

Node3 server

Public IP:

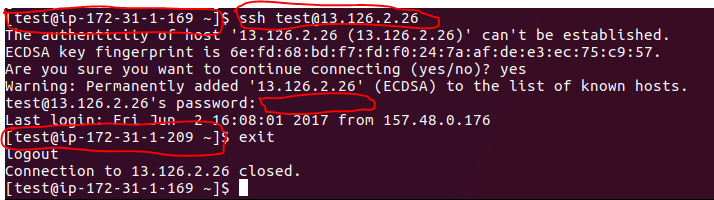
|  |
| --- |
| Private IP: |
|  |

Then we have to generate keys now

Trying to connect node1 server test user from master server test user:

ssh test@<node1 private ip>

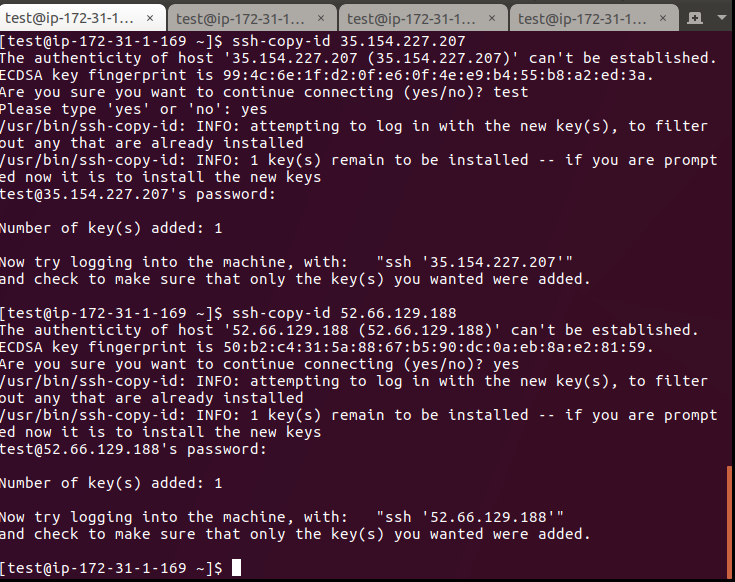
So it will ask for password to login. But it should not ask for password to login like below picture.



Type command to generate key on master server:

ssh-keygen

ssh-copy-id <private ip of all other 3 servers one by one>



it will ask pwd to copy then we can login each server without giving password like above picture

this step can be repeated for all 4 instances to check connection for instances.

uname –a

cat /etc/redhat-release -- type these commands on all 4 servers to check about os

For Ansible installation we are going to install EPEL repository why beacuase we are not able to install original ansible installation.

To check repositories available in our EC2 nstances

cd /etc/yum.repos.d/

ls –lrt

test user :

cd

sudo yum install wget

wget <http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-9.noarch.rpm>

sudo rpm –ivh epel-release-7-9.noarch.rpm

or

sudo rpm -Uvh https://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-9.noarch.rpm

cd /etc/yum.repos.d/

we can see epel repo’s here

we have to update all 4 instances

sudo yum update

---- It wil take some extra time to get update yum why because it is like new system so many updates wll get updated.

sudo yum install ansible –y

to check ansible :

ansible –version

cd /etc/ansible

ls -lrt

more ansible.config

sudo vi ansible.config

take out # sybmbol infront of log\_path

ansible –version

sudo touch /var/log/ansible.log

sudo chmod 777 /var/log/ansible.log

ansible –version

**How to configuring Ansible :**

Before anything working on config file better to take backup file

**Test user:**

cd /etc/ansible

sudo cp ansible.cfg ansible.cfg.original

**To check python dependcies all are installed or not**

sudo yum list installed | grep –i python

**To check python is working or not as expected:**

Type : python ----- to get version of the python

**To check python is working or not we are going write simple program:**

go back to home directory --- cd

vi hellow.py

#!/usr/bin/python

#Hello Golden Tech python program

print “Hello Golden Tech welcome to python session”

:wq

python hellow.py -- it will display the output of the program

Hello,   
  
Red Hat are aware of this issue and are working on a fix at the moment.  
  
There is a temporary workaround available. To do this, set 'sslverify=0' in the following files:  
  
/etc/yum.repos.d/redhat-rhui.repo  
/etc/yum.repos.d/redhat-rhui-client-config.repo  
  
Then execute "yum clean all". Once you have done this, it will work.  
  
Please note that this is a temporary fix and if you do this you should change the value back once this issue has been resolved properly.

https://forums.aws.amazon.com/thread.jspa?threadID=124278

same thing we can on other 3 nodes instances and we can verify python is working on all 4 instances as expected.

**Lecture: Overriding the Default System Ansible.cfg File :**

We have to check the path for ansible.config file

cd /etc/ansible

ls -lrt

if you want to override this file we have to change this path of confg file to do ths run following command

export ANSIBLE\_CONFIG = /home/test/config/myansible.cfg

**to check path**

set | grep –I ansible.cfg

**back to normal set up:**

export ANSIBLE\_CONFIG= /etc/ansible/ansible.cfg

**to check path**

set | grep –I ansible.cfg

**Lecture : Overriding the Default Roles Path**

cd /etc/ansible

ls –lrt

we have to check existing path we can rename it

sudo vi ansible.cfg

we can identify here roles path

roles path = /home/test/roles ----- like this we can change but not required now.

**Lecture : The HOSTS File**

Frst we have to take backup of this host file

sudo cp –pr hosts hosts.orig

First we have to any hosts set up or not using ths command

ansible all --list-hosts - it will list 0 hosts

sudo vi hosts

[local]

Localhost

[apacheweb]

Node1ip

Node2ip

[appserver]

Node3ip

:wq

ansible all –list-hosts

4 hosts should be displayed

Hosts(4):

1.

2.

3.

4.

Pwd

/etc/ansble

If you want to change the path of hosts file accessing,

Goto my home directory

Cd

Ls –lrt

Here there is no host file now

Original hosts file is available in /etc/ansible folder

ls –l /etc/ansible/hosts

bydefault ansible wil take this when it need hosts file

more /etc/ansible/ansible.cfg

here we can check in default values for inventory section path for default hosts

but we don’t want edit this file and we can say to ansible this is not my hosts file in other way.

Go to home directory ------ cd

vi hosts

[myhost]

Node3ip address

:wq

ls –l

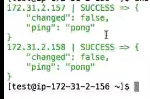
**Test the configuaration from actual config file not from my own hosts file:**

ansible all –list-hosts

it wil display all the hosts in your basic config file

ansible apacheweb –m ping

it wil be displayed to connecting ip addresses of apacheweb group and says success



Now we can check false case means not able to connect mentioned ip address in group.

For that we have to go one node system terminal and we have to stop the services and we can check

Goto one mentioned node in this group

Get ip address ---- ifconfig

sudo systemctl stop sshd

or

Switch to root

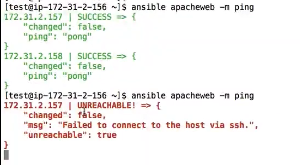
systemctl stop sshd

sudo service sshd status

Now it is inactive mode

Now go to out master server

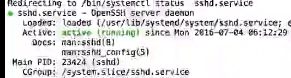
ansible apacheweb –m ping



Goto again stopped node server and restart the sshd

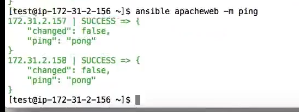
sudo systemctl start sshd

sudo service sshd status



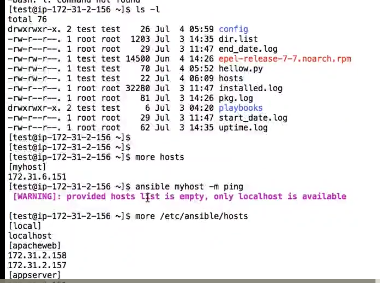
Goto Master server

ansible apacheweb –m ping



If I want to use my own hosts config file it wil give an error

ansible myhost –m ping



For connecting our own hosts config file we have to type command like this below command

ansible myhost –I hosts –m ping

we can type direct iip also

ansible <ip address> -I hosts –m ping

without password less authentication it wont work



For that we have to make password less authonticaton

Ssh-keygen

Ssh-copy-id <p address>

ansible <ip address> -I hosts –m ping



So till now we have connected individual groups in hosts file

Now we are going to connect all group names ips

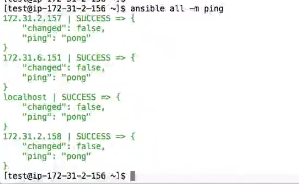
ansible all –m ping



Localhost is failing now we have to copy master key into my local system

sshd-copy-id localhost

ansible all –m ping



For own config file or override default config file

ansible myhost –I hosts –m ping

We have done our ansible environment setup with using 4 aws instances and working fine.

==================================================================================