What is Ansible

- Ansible is IT configuration management tool.
- Ansible is work on push based configuration management tool.
- Ansible is an open-source IT configuration management tool, deployment and orchestration tools.
- It aims to provide large productivity gains to a wide variety of automation challenges.

Ansible History:-

- Michel dehaan developed ansible and the ansible project began in february 2012.
- Redhat accquired the ansibe tool in 2015.
- Ansible is available for RHEL, Debain, centos, oracle, Linux.
- Can use this tool whether your servers are in on-premises or in the cloud.
- It tuns your code into infrastructure that is your computing environment has some of the same attributes as your application.

Advantage or disadvantage of asnsible server:-

Advantage :-

- Ansible is free to use by everyone.
- Ansible is very consistent and light weight and no constraints regarding the os or underlaying hardware are present.
- It is very secure due to its agentless capabilites and open ssh security features.
- Ansible does not need any special system administrator skills to install and use it.
- Work on push machanism

disadvantage :-

- Insufficent user interface through ansible tower is GUI but it still in deployment stages.
- Can not acheive full automation by ansible.
- New to the market, there fore limited support and document is available.

Terms use in Ansible:-

Ansible server:- The machine where ansible is installed and from which all tasks and playbooks will be run**.**

Modules:- Basically, a module is a commands or set of similar commands ment to be executed an the client side.

Tasks:- A task is a section the consists of a single procedure to be completed.

Role:- A way of organising tasks and related files to be later called in a playbook.

Fact:- Information fetched from the client system from the global variables with the gather-facts operation.

Inventory:- File containing data about the ansible client servers.

play:- execution of a playbooks

Handler:- Task which is called only if a notifiers present

Notifier:- Section attributed to a task which calls a handler if the output is changed.

Playbooks:- It consist code in YMAL format which describes tasks to be executed.

Host:- Nodes, which are automated by ansible.

Ansible Demo:- Ansible inventry, HOST pattern

- Go to aws account ----- create 3 ec2 instance in same az take a access of all machine via putty or mobaxtrem.

sudo su

yum update -y

To install ansible server:-

wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

To check :-

ls

o/p:-epel-release-latest-7.noarch.rpm

Install epel repository:-

yum install epel-release-latest-7.noarch.rpm

Update epel repository:-

yum update -y

Install all individual packages inside the repository:-

yum install git python python-devel python-pip openssl ansible -y

After that we need to go to hosts file inside ansible server and paste "private ip" of each nodes like node 1, node2.

vi /etc/ansible/hosts

Now this hosts file is only working after updating ansible.cfg file.

vi /etc/ansible/ansible.cfg

Uncomment both line

 # inventry=/etc/ansible/hosts

 # sudo-user=root

eg:-

inventry=/etc/ansible/hosts

sudo-user=root

After create one user in all the three instances.

adduser ansible

passwd ansible

ansible (*******)

Now the user root to ansible user

su - ansible

this ansible user don't have sudo privileged right now if you want to give sudo priviledge to ansible user.

Exit from the ansible user

sudo su

visudo

Now go inside the file.

root ALL=(ALL) ALL

ansible ALL=(ALL) NOPASSWD:ALL

Now do this thing in other nodes also now go to ansible server and try to install httpd package as a ansible user.

[ansible ip] sudo yum install httpd -y

Now establish connection between server and node go to ansible server.

service sshd restart

[ansible ip] ssh (node privateip)

[ansible ip] ssh 172.31.42.34

o/p:- permission denied

Now we have to do some changes in ssh-config file go to ansible server.

vi /etc/ssh/sshd-config

do some changes & and saved the file.

do this work in node1 or node2 also.

su - ansible

[ansible ip] ssh 172.123.67.89

Now it ask for password enter the password after that you will be inside node1.

LAB:-

Create a 3 instances & login with putty all of one.

Go inside the ansible server, login node1 , node2 also

sudo su

Update Server Packages:-

yum update -y

Install ansible server :-

wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

Verify epel repository:-

ls

o/p:- epel-release-latest-7.noarch.rpm

Install epel repository:-

yum install epel-release-latest-7.noarch.rpm

Update epel repository:-

Update epel repository:-

yum update -y

Install all individual packages inside the repository:-

yum install git python python-devel python-pip openssl ansible -y

Verify Version of Ansible

ansible --version

o/p:- ansible 2.9.27

Go inside the host file path /etc/ansible /hosts

vi /etc/ansible/hosts

Ex 1:- (Here we can create a group) demo

[demo]

172.31.91.115

172.31.86.54

:wq

Note:- Inside the demo group we have to define private ip of node

Here create a another group also as per your requirement.

[developers] (here you can define your developer node details like private ip).

After that we need do some changes inside config file

vi /etc/ansible/ansible.cfg

Here we need to uncomment some line

inventry=/etc/ansible/hosts

sudo_user=root

:wq

Now will create one user:-

To create user:-

adduser ansible

To set passwd:-

passwd ansible

ansible (******)

Note:- Node1 or node2 also we have to create one user

come to node1 machine

sudo su

Now will create one user:-

To create user:-

adduser ansible

To set passwd:-

passwd ansible

ansible (******)

Come to the node2 machine

Now will create one user:-

sudo su

To create user:-

adduser ansible

To set passwd:-

passwd ansible

ansible (******)

Now come to the ansible server and login with ansible user.

sudo su

su - ansible

Create some file here.

touch file file2

ls

Now try to install some packages by ansible user.

[ansible ip] yum install httpd -y

o/p:- you need to be root to perform this cmd.

[ansible ip] sudo yum install httpd -y

password

o/p:- ansible is not in the sudoers file.

for that exit from the ansible users via exit cmd.

exit

We have to assigned sudo permission to ansible users.

sudo su

visudo

root ALL=(ALL) ALL

ansible ALL=(ALL) NOPASSWD:ALL

:wq

Note:- same thing we need to do in node 1 node2.

come to node1 machine

We have to assigned sudo permission to ansible users.

sudo su

visudo

root ALL=(ALL) ALL

ansible ALL=(ALL) NOPASSWD:ALL

:wq

come to node2 machine

We have to assigned sudo permission to ansible users.

sudo su

visudo

root ALL=(ALL) ALL

ansible ALL=(ALL) NOPASSWD:ALL

:wq

Now come to the ansible server and login with ansbile users

sudo su

To switch user:-

su - ansible

Now try to install some packages by ansible user.

[ansible ip] yum install httpd -y

o/p:- error

[ansible ip] sudo yum install httpd -y

o/p:- successfully installed.

Note:- Try to install on node1 or node2 as well as.

After that verify we are able to connect with node or not for that come to ansible server login with ansible users:-

su - ansible

try to take ssh of nod1 or node2

ssh 172.178.90.9

o/p:- permission denied

exit from the ansible users.

For ssh we need to do some changes via root users:-

sudo su

go inside the /etc/ssh/sshd_config

vi /etc/ssh/sshd_config

here we need to do some changes

#permitRootlogin yes:-uncomment line no 38

#password authentication yes:- uncomment line no 61

passwordAuthentication no:- comment line no 63

eg:-

permitRootlogin yes line no 38

password authentication yes line no 61

#passwordAuthentication no line no 63

:wq

To restart sshd:-

service sshd restart

Samething we need to node1 node2

Come to node1

sudo su

go inside the /etc/ssh/sshd_config

vi /etc/ssh/sshd_config

here we need to do some changes

#permitRootlogin yes:- we have to uncomment

#password authentication yes:- we have to uncomment

passwordAuthentication no:- we have to comment

eg:-

permitRootlogin yes

password authentication yes

#passwordAuthentication no

:wq

To restart sshd:-

service sshd restart

Come to node2

sudo su

go inside the /etc/ssh/sshd_config

vi /etc/ssh/sshd_config

here we need to do some changes

#permitRootlogin yes:- we have to uncomment

#password authentication yes:- we have to uncomment

passwordAuthentication no:- we have to comment

eg:-

permitRootlogin yes

password authentication yes

#passwordAuthentication no

:wq

To restart sshd:-

service sshd restart

Come to the ansible server login with ansible user

su - ansible

try to take ssh of node1

ssh 172.168.21.240

password:-

Here we have a remote access of node1 via ansible users

create some file

touch file1 file 4

to check file are create or not in node1 machine.

go to node1 and check the file.

Take a ssh node2 as well as.

ssh 179.168.21.240

While taking ssh asking password all the time how to get assess without password:-

For that we need to create ssh-key or we have to share that key with all nodes

This task we have done via ansible user because kegen is work on **trust relationship** method

like:-

root to root

user to user

Come to the ansbile server login as a ansible user

To jenerate ssh-key:-

su - ansible

[ansible ip] ssh-keygen

ls -a

o/p:- .ssh

cd .ssh

ls

o/p:- id_rsa id_rsa_pub

Now i need to copy public key in both the machine node1, node2

[ansible ip]ssh-copy-id username@privateip

[ansible ip]ssh-copy-id ansible@179.168.21.240

Ask one time password:- one time password ansbile

[ansible ip]ssh-copy-id ansible@180.168.21.240

Ask one time password:- one time password ansbile

HOST pattern:-

- **"all" pattern refers to all the machine in an inventry**
- **ansible all --list-hosts**
- **ansible <groupname> --list-hosts**
- **ansible <groupname> --list-hosts**
- group[0]:- picks first machine of group
- group[1]:- picks second machine of group
- group[-1]:- picks last machine of group
- group[1:4]:- picks 2-5 machine of group
- group[2:5]:- picks 3-6 machine of group

Two group

demo [1:2]: devops [2:10]

- eg:- **ansible demo [1:2]: devops [2:10] --list-hosts**

To check all hosts:-

ansible all --list-hosts

To check group hosts details:-

ansible <groupname> --list-hosts

ansible demo --list-hosts

To check first machine of group:-

ansible demo[0] --list-hosts

To check last machine of group

ansible demo[-1] --list-hosts

To check second machine of group

ansible demo[1] --list-hosts

To check 2-5 machine of group

ansible demo[1:4] --list-hosts

For more details visit:- https://docs.ansible.com/