Ansible Notes

1) Ansible

   Ansible is a configuration management tool. It is pushed based mechanism tool. It is agentless also. It works with Unix to deploy configs on other hosts. To be able to run playbook on other hosts, other servers should be in the same network as ansible master node and also ssh should enabled.

2) Play

   It is program define by ansible to perform a set of tasks in sequence.

3) Playbook
   It is the yaml file where the play is written and we use the playbook to execute the program/play. One playbook can have more than one play also.

4) YAML

   Ansible supports yaml (yet another markup language) language for writing its play in playbook. It is very particular about the indentation.

5) Control Machine

   Machine on which ansible is installed and from where we can manage other machines. This machine can also be remote machine too.

6) Remote Machine

   Machines which are handled/controlled by control machine.

7) Service/Server

   A process on the machine that provides the service.

8) Task

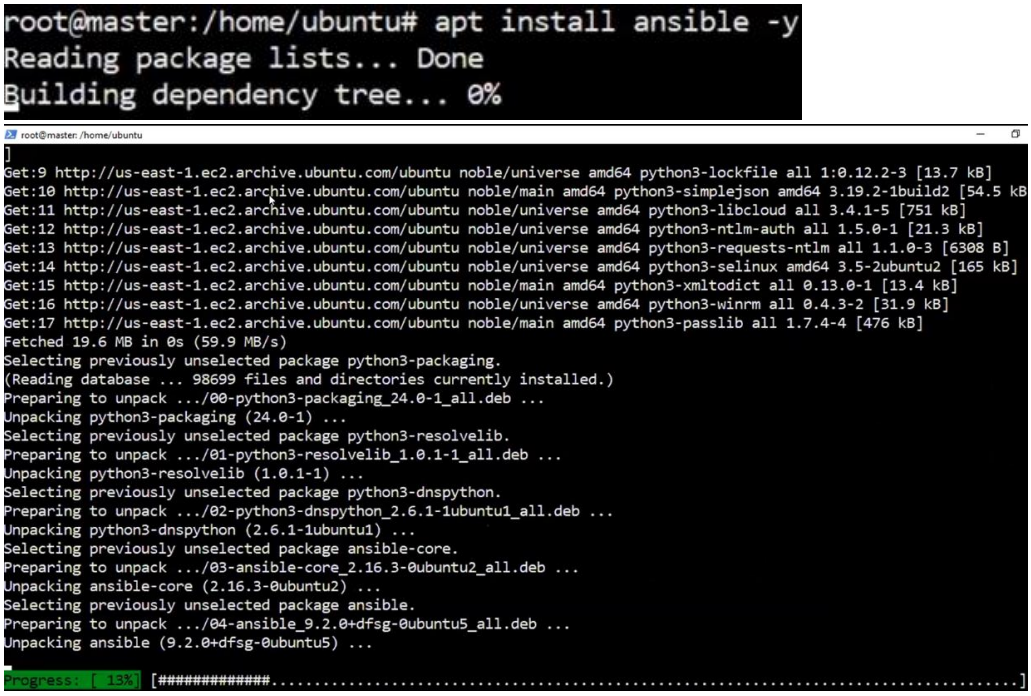
An action (run this, delete that) etc managed by ansible.

9) Modules

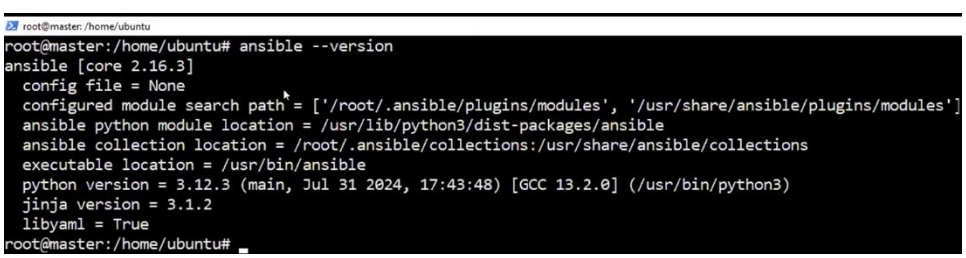
Modules are like plugin in ansible that do the actual work, they are what gets executed in each playbook task. But we can also run a single one using ansible command.

10) Ansible Installation.

apt install ansible -y (Command for Ansible installation on Ubuntu)



11) Ansible --version (Provides the version of ansible installed)



12) Ansible Directory

We need to create an ansible directory under /etc mount point

```
root@master:/home/ubuntu# cd /etc
root@master:/etc# mkdir ansible
```

13) Hosts file (Inventory File) under Ansible directory.
We need to create a hosts file under /etc/ansible (default location of hosts file) directory path. This file will be used to add hosts names which we want to run our playbooks

```
root@master:/etc/ansible# touch host
root@master:/etc/ansible# mv host hosts
root@master:/etc/ansible# ll
```

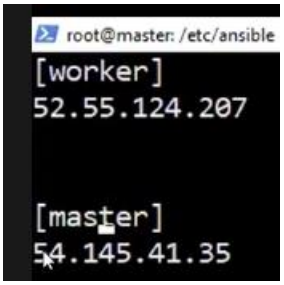
14) Enter the hostname details in hosts file.

Vi hosts

```
root@master:/etc/ansible# vi hosts
```

[Worker]
Public IP

Exit and save the file.

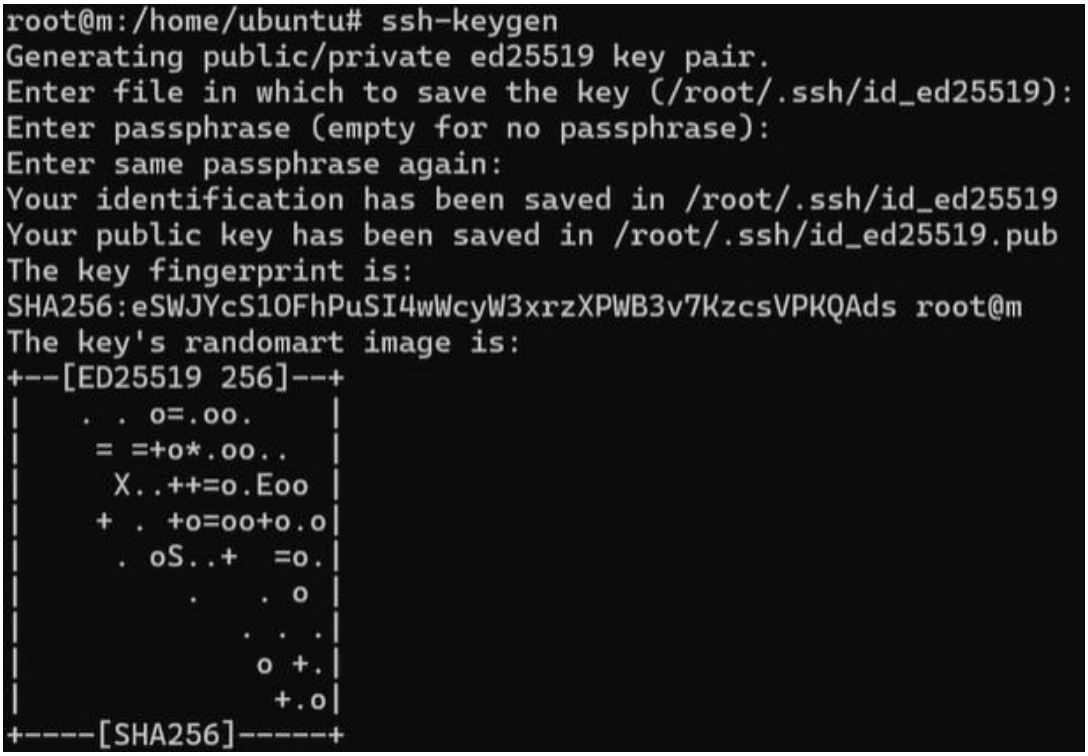


15) Password less SSH

We will setup password less ssh between master and worker so that we do not need to give password all the time. Steps to setup password less SSH.

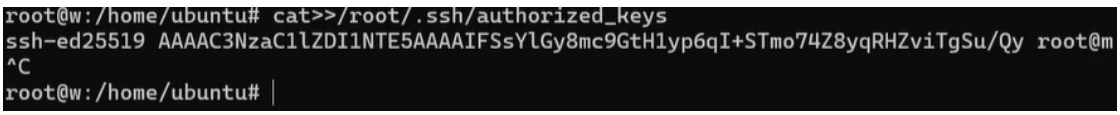
- Now setup password less ssh between master and worker by following below steps.

  On master: Type **ssh-keygen** and follow all steps as follows to generate public key and private key



  Now do **cat /root/.ssh/id_ed25519.pub** and copy all the contents inside it.

  On Worker: Do **cat >> authorized_keys** (append authorized_keys file and paste the contents copied from master server)



Now we will ssh to worker node from master using the Public IP address

```
root@m:/home/ubuntu# ssh 52.91.179.231
The authenticity of host '52.91.179.231 (52.91.179.231)' can't be established.
ED25519 key fingerprint is SHA256:HnLaE5TZWCOmsF0wdFsO9zU/XgdBeCcJ3hJrssBjjHs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

```
root@w:~#
```

If we exit the worker, we will return back to the master node.

```
root@w:~# exit
logout
Connection to 52.91.179.231 closed.
root@m:/home/ubuntu#
```

NOTE: We also need to add the public key of master in authorized_keys of master also, so that it can be setup for master also.

16) Ping

This command is use to check and verify the connectivity between ansible master and target machines.

Ansible -m ping “entry that is there within [] inside the inventory file”.

```
root@master:/home/ubuntu# ansible -m ping master
54.145.41.35 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Ansible -m ping all (This will run for all the entries inside hosts file) In our case worker and master.

```
root@master:/home/ubuntu# ansible -m ping all
52.55.124.207 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
54.145.41.35 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

17) Ansible-inventory --list -y

This command will print the list of hosts added in the file with their groups name.

```
root@master:/home/ubuntu# ansible-inventory --list -y
all:
  children:
    master:
      hosts:
        54.145.41.35: {}
    worker:
      hosts:
        52.55.124.207: {}
root@master:/home/ubuntu#
```

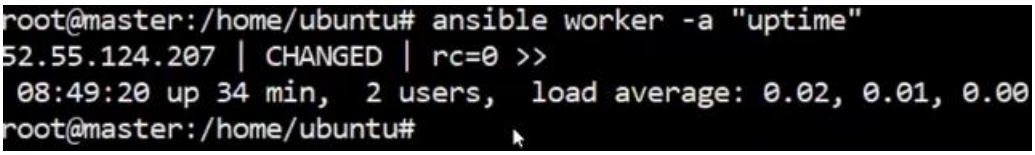
18) Ansible all -a “df -h”

This command will check the disk usage of all servers in hosts file.

```
root@master:/home/ubuntu# ansible all -a "df -h"
52.55.124.207 | CHANGED | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  2.3G  4.5G  34% /
tmpfs           479M     0  479M   0% /dev/shm
tmpfs           192M  920K  191M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda16     881M  133M  687M  17% /boot
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
tmpfs            96M   12K   96M   1% /run/user/0
54.145.41.35 | CHANGED | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  2.8G  3.9G  42% /
tmpfs           479M  148K  479M   1% /dev/shm
tmpfs           192M  920K  191M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda16     881M  133M  687M  17% /boot
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
tmpfs            96M   12K   96M   1% /run/user/0
root@master:/home/ubuntu#
```

19) Ansible worker -a “uptime”

This command will give the uptime of only servers in worker group in the hosts file.



20) Ansible-playbook <playbook.yml> [Name of playbook]

This command will run the playbook that is named as playbook.yml

21) Ansible-playbook <playbook.yml> [Name of playbook] –syntax-check

This command will check playbook for any syntax errors.