Ansible:

* Is simple open-source IT engine automates: app. deployment, intra service orchestration, cloud provisioning and many other IT tools.
* Is easy to deploy coz it does not use any agents or custom security infrastructure.
* Uses playbook to describe automation jobs, and playbook uses very simple language i.e. YAML
* Is designed for multi-tier deployment.
* Uses the hosts file where one can group the hosts & can control the actions on a specific group in the playbooks.

We use ansible coz

-it automates and simplifies repetitive, complex. And tedious operations.

-Is open-source, saves time as well as human efforts and is easy to implement.

-Ansible architecture is simple and effective, works by connecting to your nodes & pushing small programs to them.

-Ansible is **puh-based** architecture & doesn’t need any agents running opn the client nodes.

-It **works over SSH** and doesn’t require any daemons, special servers, or libraries to work. A text editor and a command line tool are usually enough to get your work done.

-Ansible infrastructure is described in a text file (INI) and then all the information about the desired state of these machines are organized in playbooks.

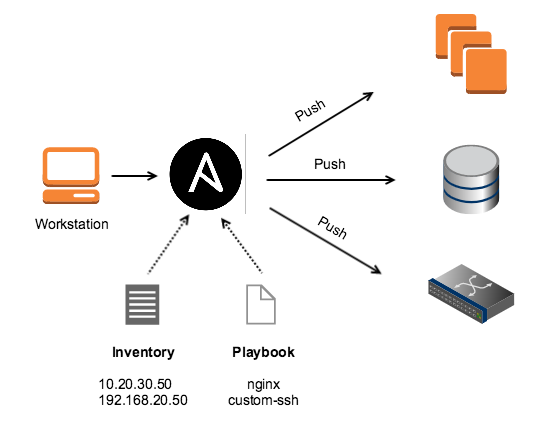
Advantages:

* **Agentless:** doesn’t require any additional software on your server nodes. This helps keep installation clean.
* **Simple:** uses a simple syntax written in YAML called **playbooks. YAML** is a human readable data serialization language, doesn’t require coding skills.
* **Powerful and flexible:** It enables us to model even the most complex IT workflows.
* **Efficient:** No extra software on your servers means more resources for your application.
* **Open-source:** one of the powerful DevOps tools which is open-source.
* **Secure:** Ansible uses SSH connection which is encrypted and secure.
* **Easy of Use:** One can configure and manage complex infrastructure solutions very easily using ansible.

**Push Based Vs Pull Based:**

-Tool like chef & puppet are pull architecture based.

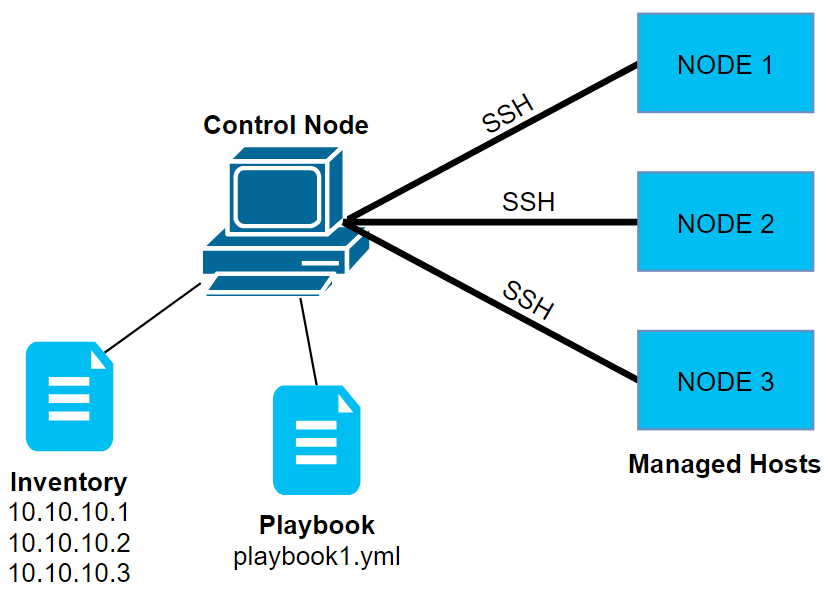
-Agents on target servers(nodes) periodically checks for config. Info. from the central server.



Ansible has push-based architecture & doesn’t need agents running on the client nodes.

Central server pushes config. Info. On the target servers(nodes).

U can control when the changes are implemented on target servers.



**Controller machine:** ansible is installed, responsible for running provisioning on the servers you r managing.

**Inventory:** An initialization file that contains info. Abt. The servers u r managing. These are the lists of nodes or hosts containing their databases, servers, IP addresses, etc.

**Playbook:** Describes the tasks that need to be executed. They are simple code files written in YAML format and can be used to declare configurations, automating tasks, etc. Playbooks are a bunch of commands which can perform multiple tasks and each playbook are in YAML file format.

**Task:** a block that defines a single procedure to be executed.

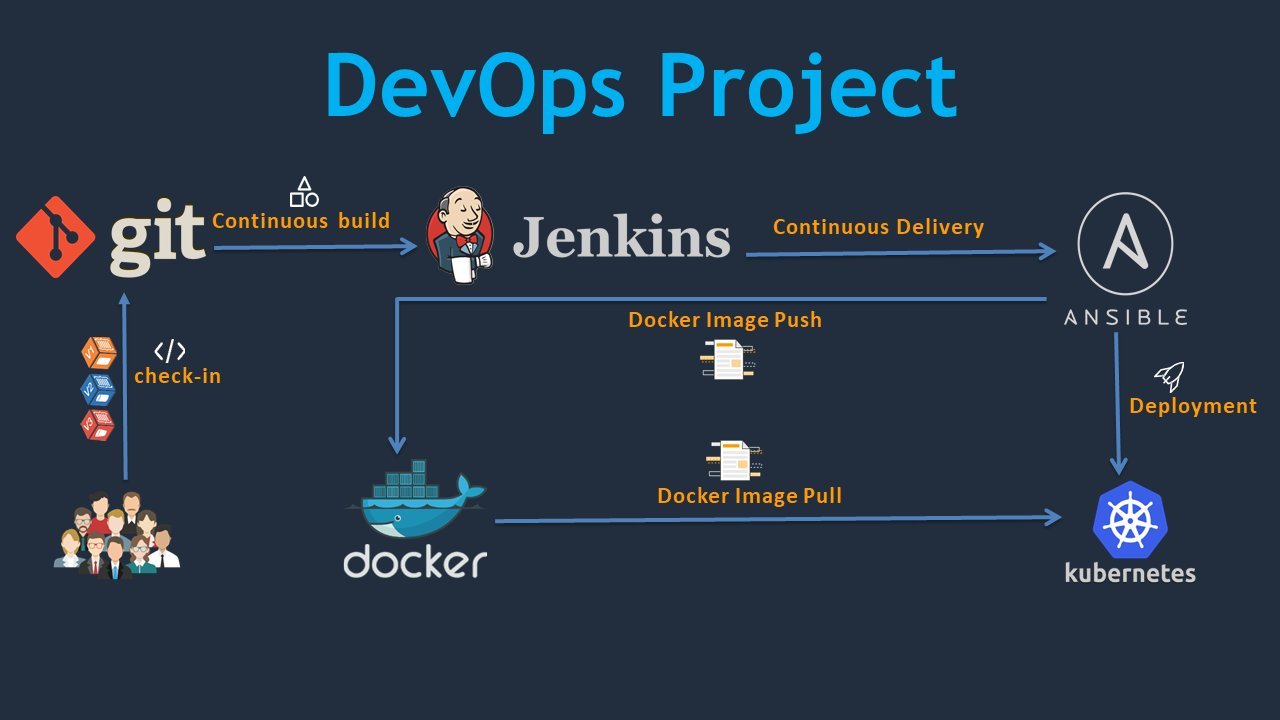
**Module:** Ansible works effectively by connecting nodes and pushing out scripts called "Ansible modules". It helps to manage packages, system resources, files, libraries, etc.

**Role:** pre-defined way for organizing playbooks and other files in order to facilitate sharing and reusing portions of provisioning.

**Play:** A provisioning executed from start to finish is called a play. (execution of playbook)

**Facts:** Gloabal variables containing info. Abt. System like network interfaces or operating system.

**Hnadlers:** used to trigger service status changes like restarting or stopping service.



**Installation of ansible in rhel8:**

Ip ad

nmcli device delete virbr0

nmcli device

nmcli device connect enp0s3

nmcli device

nmcli connection add type etherne ifname enp0s3 con-name ansible ip4 192.168.122.10/24

nmcli connection up ansible

ip ad (in the similar way do above cmds in nodes also)

ansible –version

Settings 🡪 network 🡪 adapter 1 🡪 bridged adapter 🡪 advanced : allow all.(in all machines)

If we wish to delete ansible the cmd is :

nmcli con delete ansible

nmcli device

hostnamectl set-hostname **ansible**.linux.com(similar way in nodes)

{vim /etc/yum.repos.d/server.repo

[hhh]

name=jhdjhd

baseurl=file:///mnt

enabled=1

gpgcheck=0} (in all machines)

lsblk 🡪 to find harddisk name

attach rhel file

mount /dev/sr0 /mnt

ls /mnt

nmcli con up enp0s3 🡪 to start n/w conn. In our system

ping [www.google.com](http://www.google.com)

app. 🡪 firefox 🡪 search install ansible on redhat7 🡪 itzgeek website.

Copy paste centOs cmd (yum install -y https://)

yum install ansible

ping 192.168.122.100

ping 192.168.122.50

{vim /etc/hosts

192.168.122.10 ansible.linux.com

192.168.122.50 node1.linux.com

192.168.122.100 node2.linux.com}

send /etc/hosts file to clients

scp /etc/hosts 192.168.122.50:/etc/hosts

scp /etc/hosts 192.168.122.100:/etc/hosts

ping node1.linux.com

ping node2.linux.com

**Nodes machine:**

**useradd Kathy**

**passwd Kathy**

Ansible machine

ssh [kathy@node1.linux.com](mailto:kathy@node1.linux.com)

exit

ssh [kathy@node2.linux.com](mailto:kathy@node2.linux.com)

exit

ssh-keygen

ssh-copy-id [kathy@node1.linux.com](mailto:kathy@node1.linux.com)

**Nodes machine:**

**Vim /etc/ssh/sshd\_config**

**line no.43 remove # from public key**

**line no.65 add # infront of passwd authen.**

**systemctl restart sshd**

**usermod -G wheel Kathy**

**vim /etc/sudoers**

**line no.110 remove #**

ansible

cd /etc/ansible

pwd

ls

touch inventory

{vim inventory

[cloud]

node1.linux.com

node2.linux.com}

{vim ansible.cfg

inventory = ./inventory  
remote user = kathy  
ask\_pass = false  
[privilage\_escalation]  
become = true  
become\_method = sudo  
become\_user = root  
become\_ask\_pass = false}

ansible cloud --list-hosts(lists hosts in ansible)

ansible all -m ping(to chk conn b/w ansible & node machines)

x(creates)

ansible cloud -m user -a ‘name=bob uid=4000 state=**absent**’(deletes)

**cat /etc/passwd**

ansible cloud -m command -a ‘id kathy’

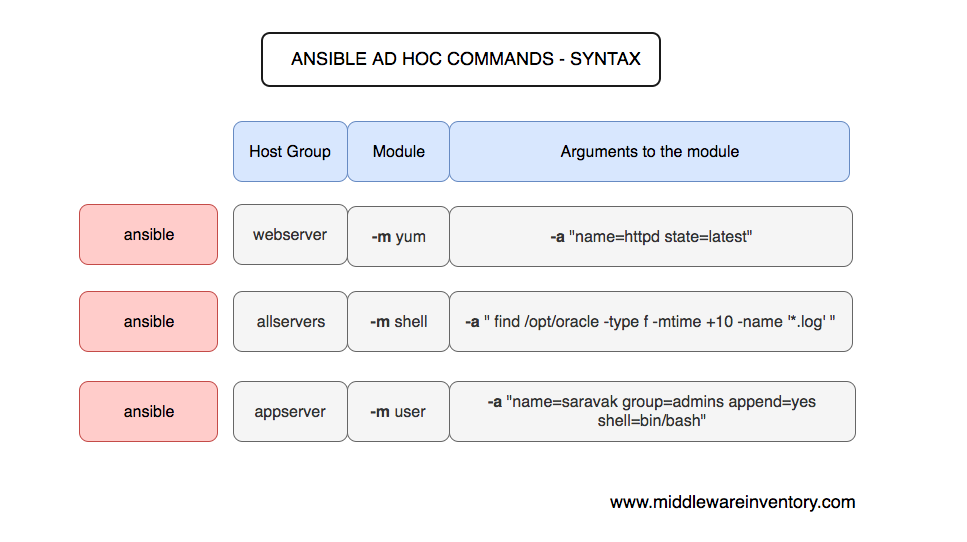
**vim /kathy.txt**

**hello welcome to linux world**

**I am Kathy how are you**

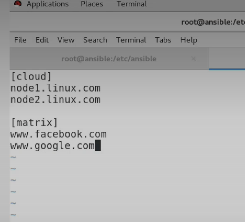
**pwd**

ansible node1.linux.com -m command -a ‘cat /root/kathy.txt’

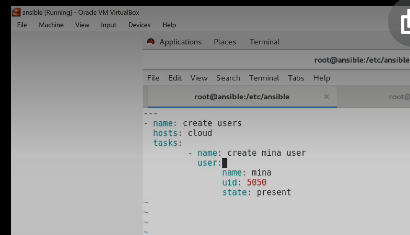


cd /etc/ansible/

vim inventory



vim playbook1.yml

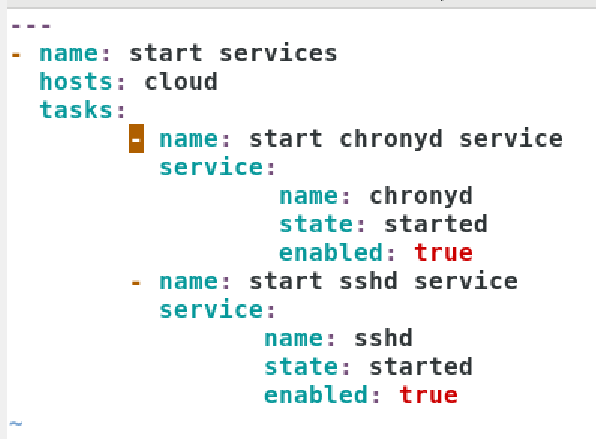


ansible-playbook playbook1.yml

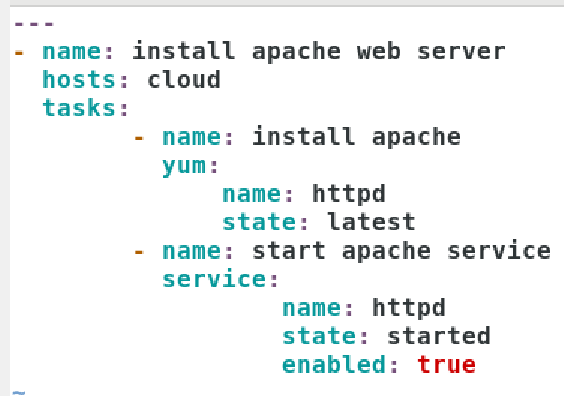
**systemctl start sshd**

**systemctl status sshd**

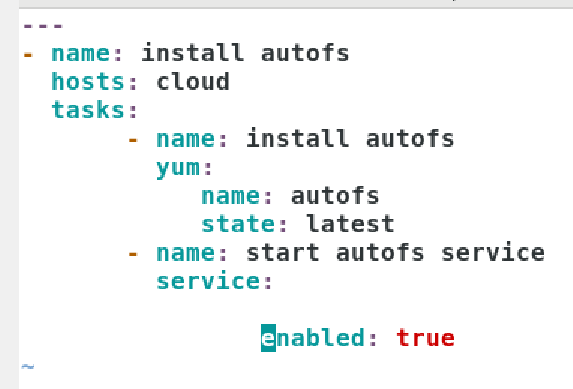
vim playbook2.yml



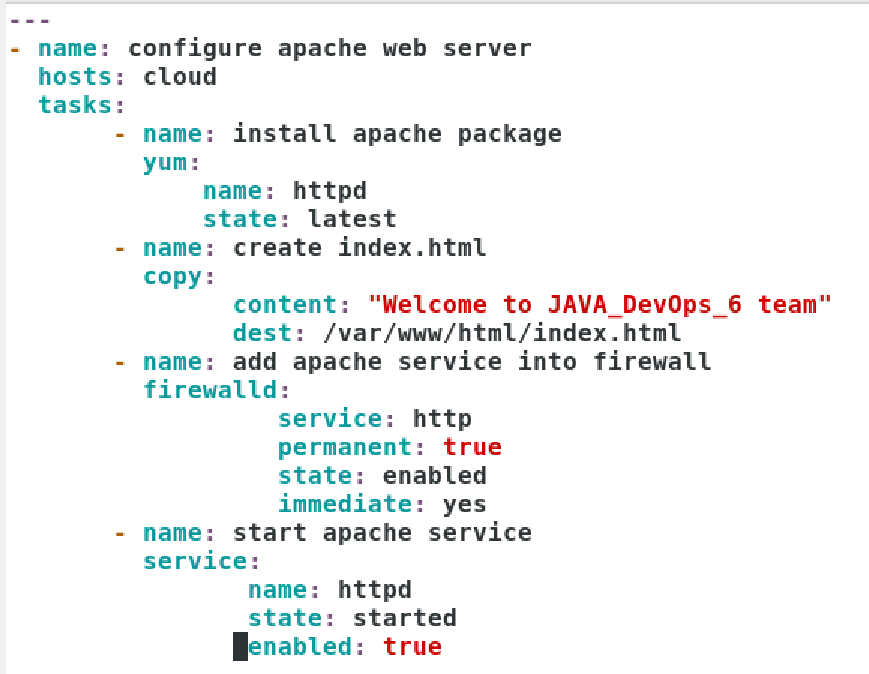
vim playbook3.yml



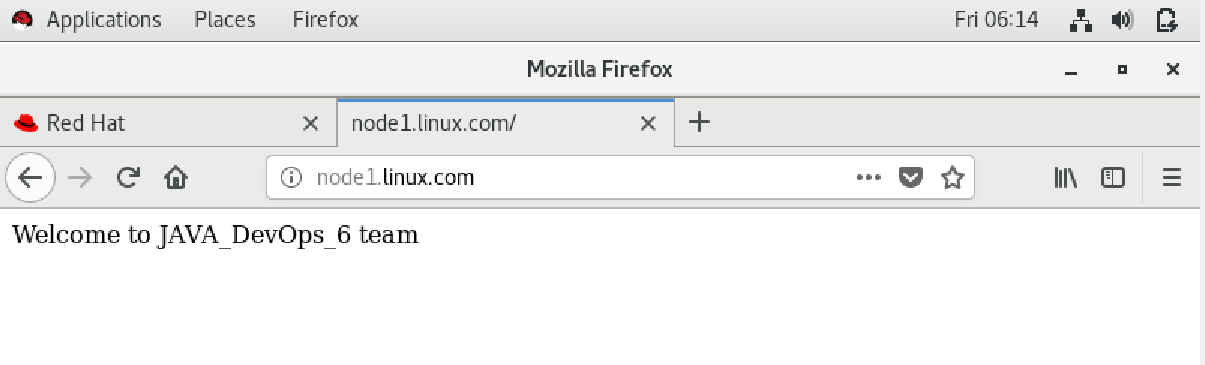
vim playbook4.yml



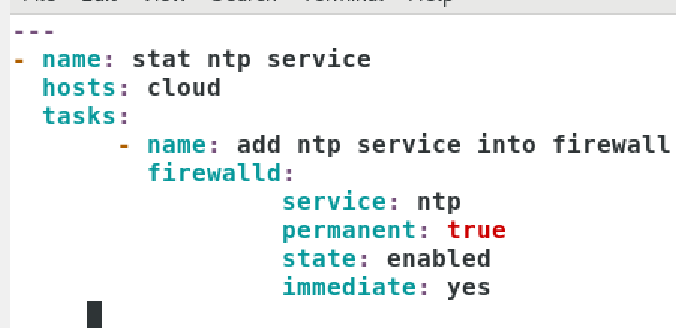
vim apachewebserver.yml



Firefox🡪 browse node1.linux.com



vim ntpwebserver.yml



Ansible First playbook writing

All playbooks are written in YAML == “Yet Another Markup Language”

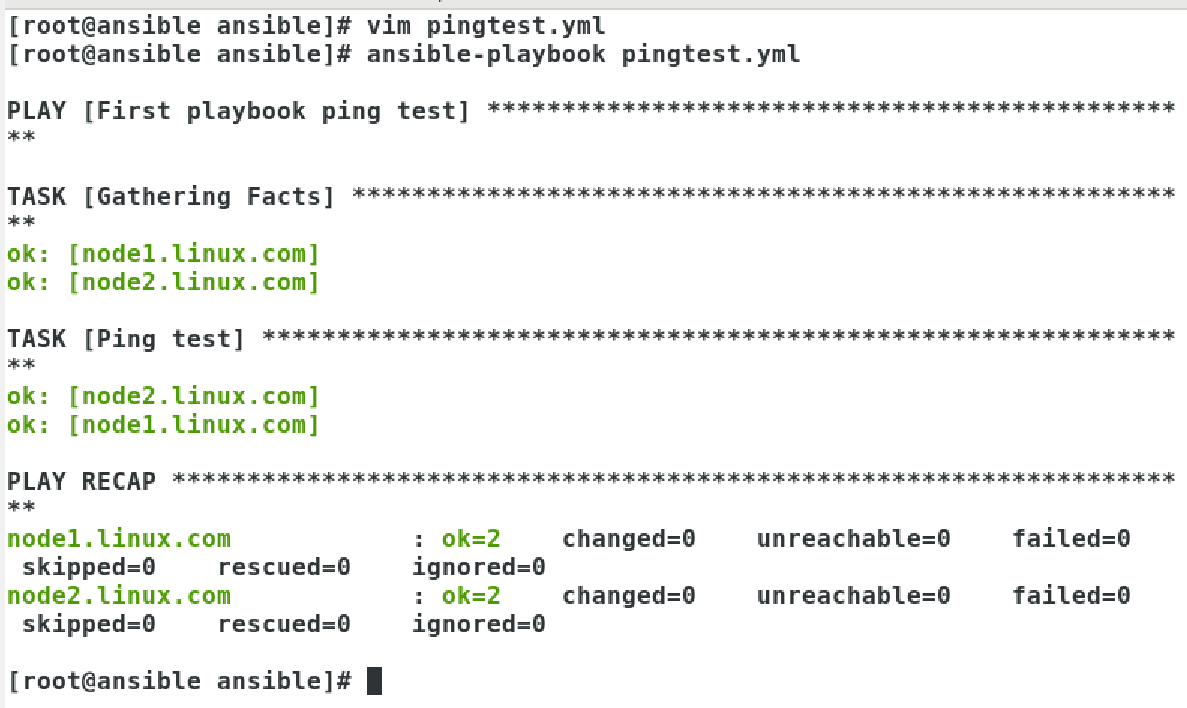
Playbook begin with ---

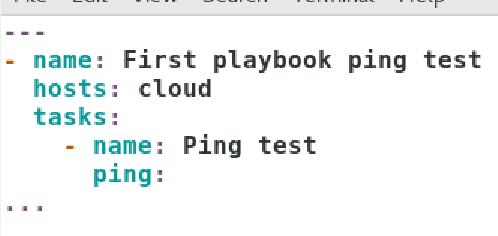
Comments begin with # hash

Members of lists begin with –

Playbook ends with …

Key value pairs <key> : <value>





Ansible Ad-Hoc Commands

Syntax:

Ansible <Group/server Name> -m <module> -a <arguments> -u <user name> --become

-B ‘SECONDS’

-k (ask password)

-T ‘TIMEOUT’

-a ‘MODULE\_ARGS’

-b (become)

-I (inventory file)

-m Module

-v or -vvv or -vvvv ‘debugging’

ansible cloud -a ‘uptime’

ansible cloud -m shell -a “service sshd status”

ansible cloud -m shell -a ‘systemctl status sshd’

ansible cloud -m command -a ‘df -h’

ansible cloud -m copy ^C

ls

touch testingansib

vim testingansib

This is an testing copy file from ansible

ansible cloud -m copy -a “src=/root/testingansib dest=/home/lakshmi”

ansible cloud -m yum -a “name=wget state=present” (install)

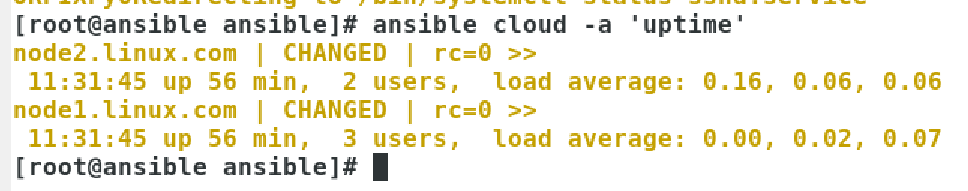
ansible cloud -m yum -a “name=wget state=absent” (uninstall)

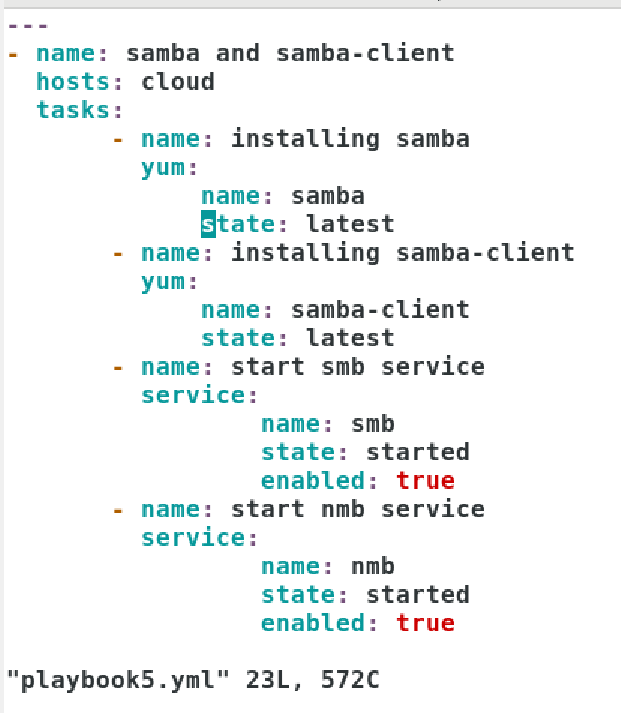
ansible cloud -m user -a “name=henry password=redhat”

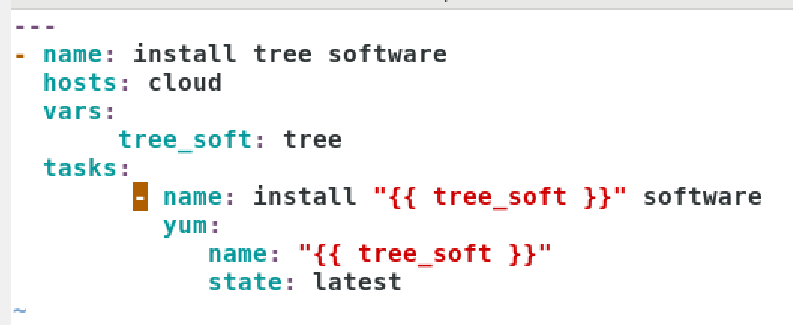
**cat /etc/passwd | grep henry**

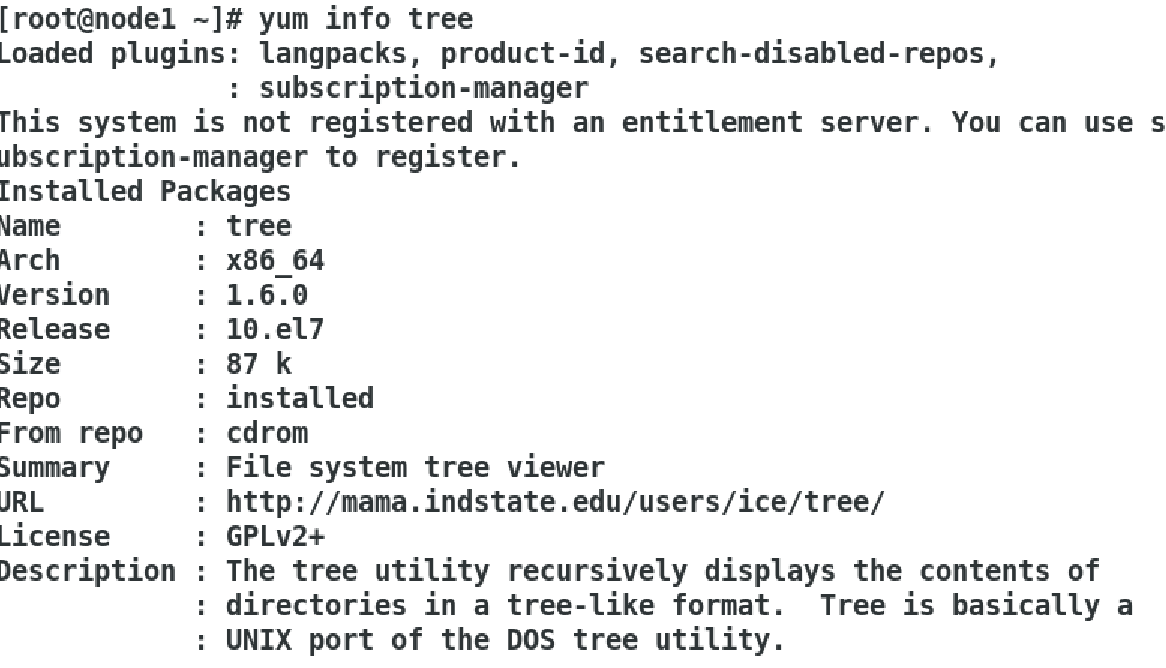
ansible cloud -m service -a “name=sshd state=started”

ansible cloud -m shell -a “/sbin/service sshd status”

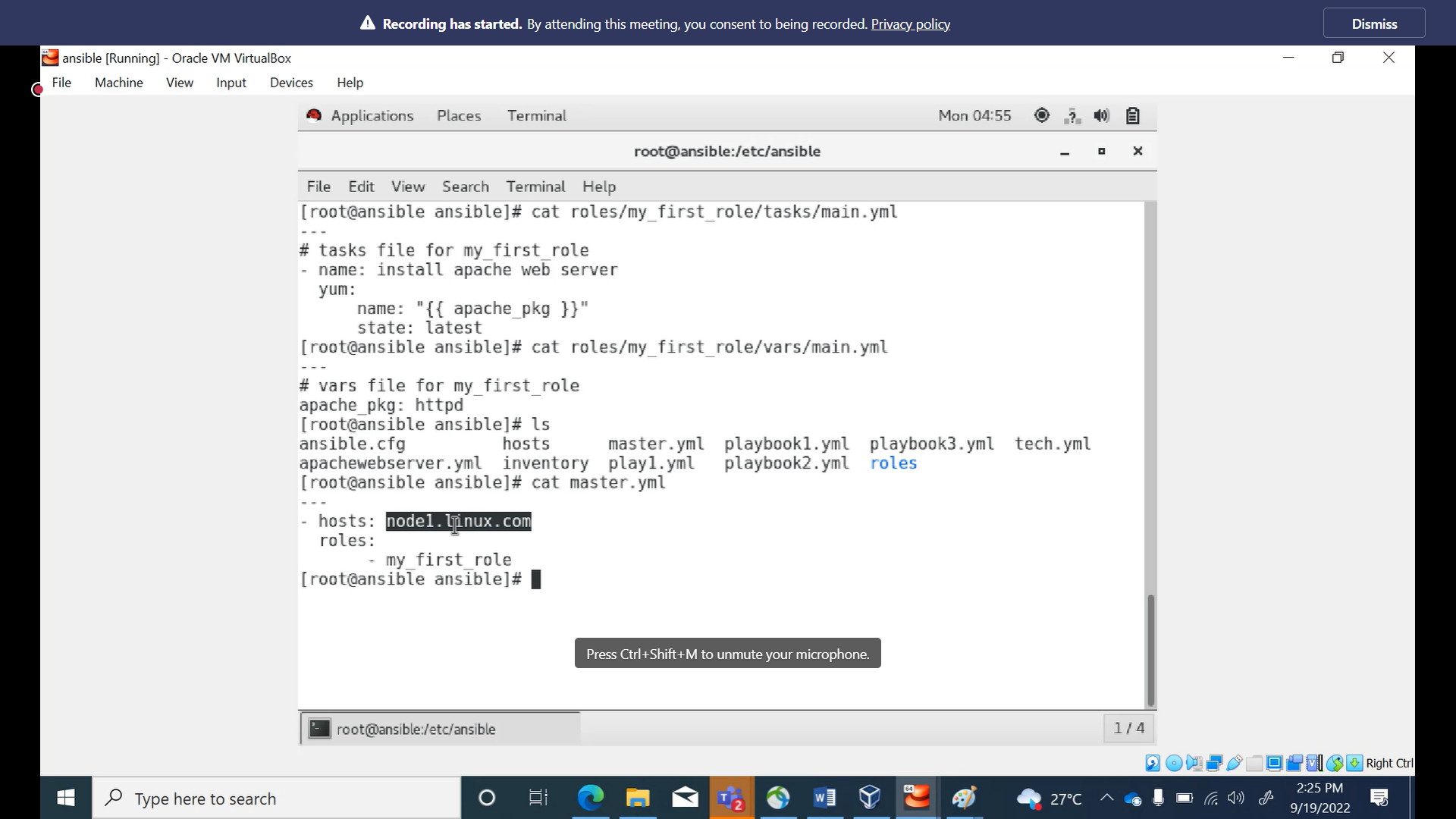




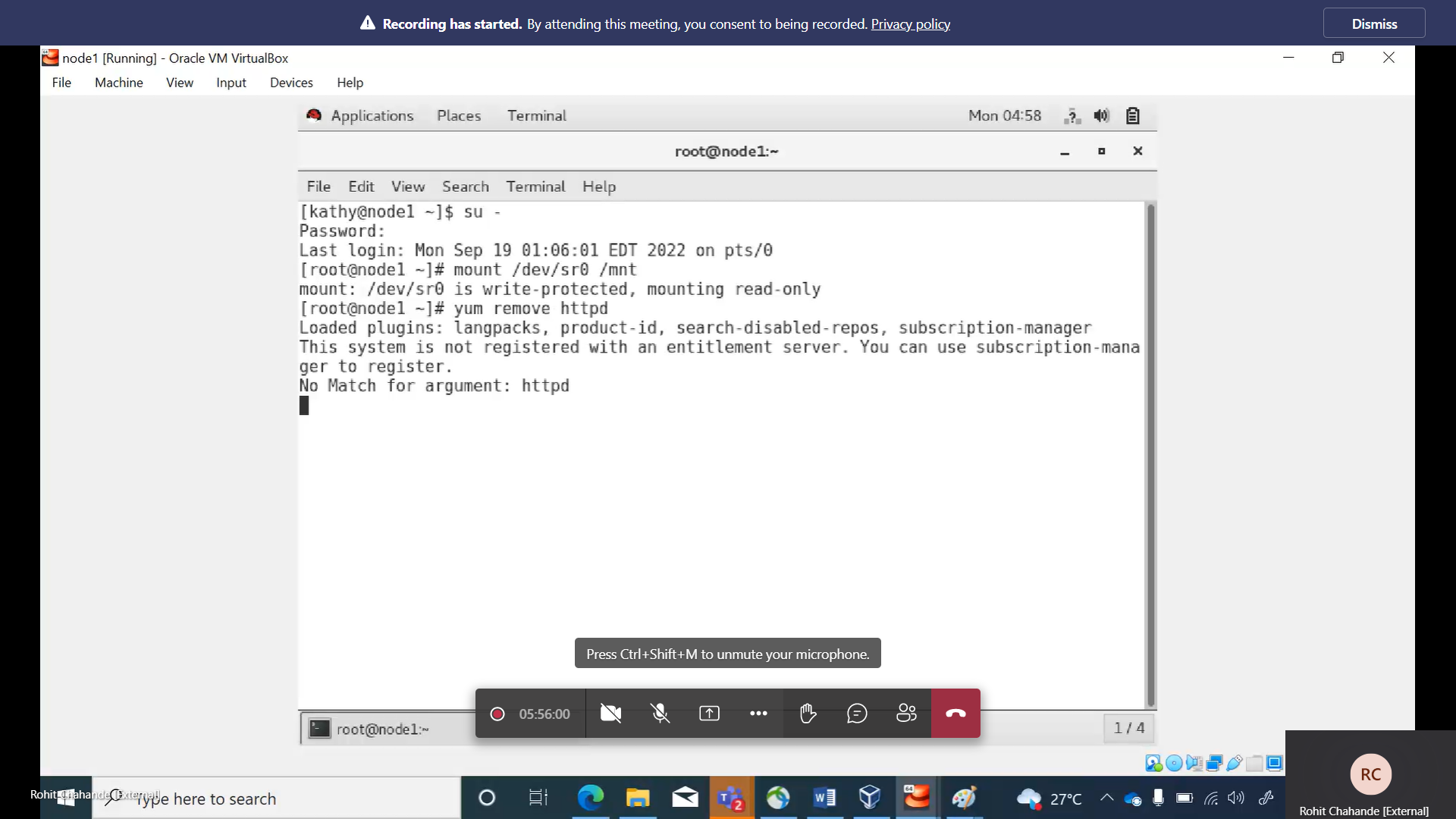
vim tech.yml

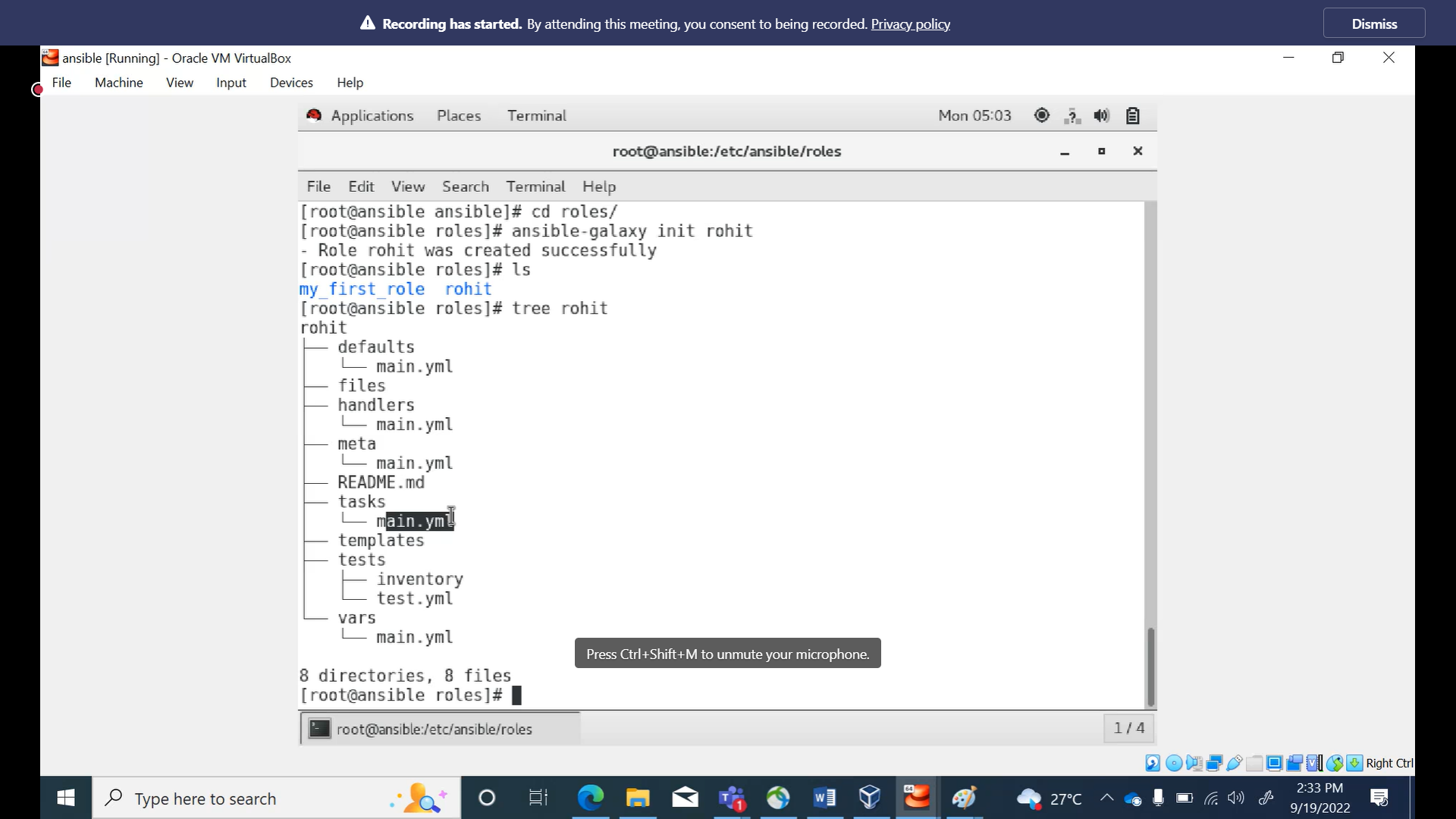


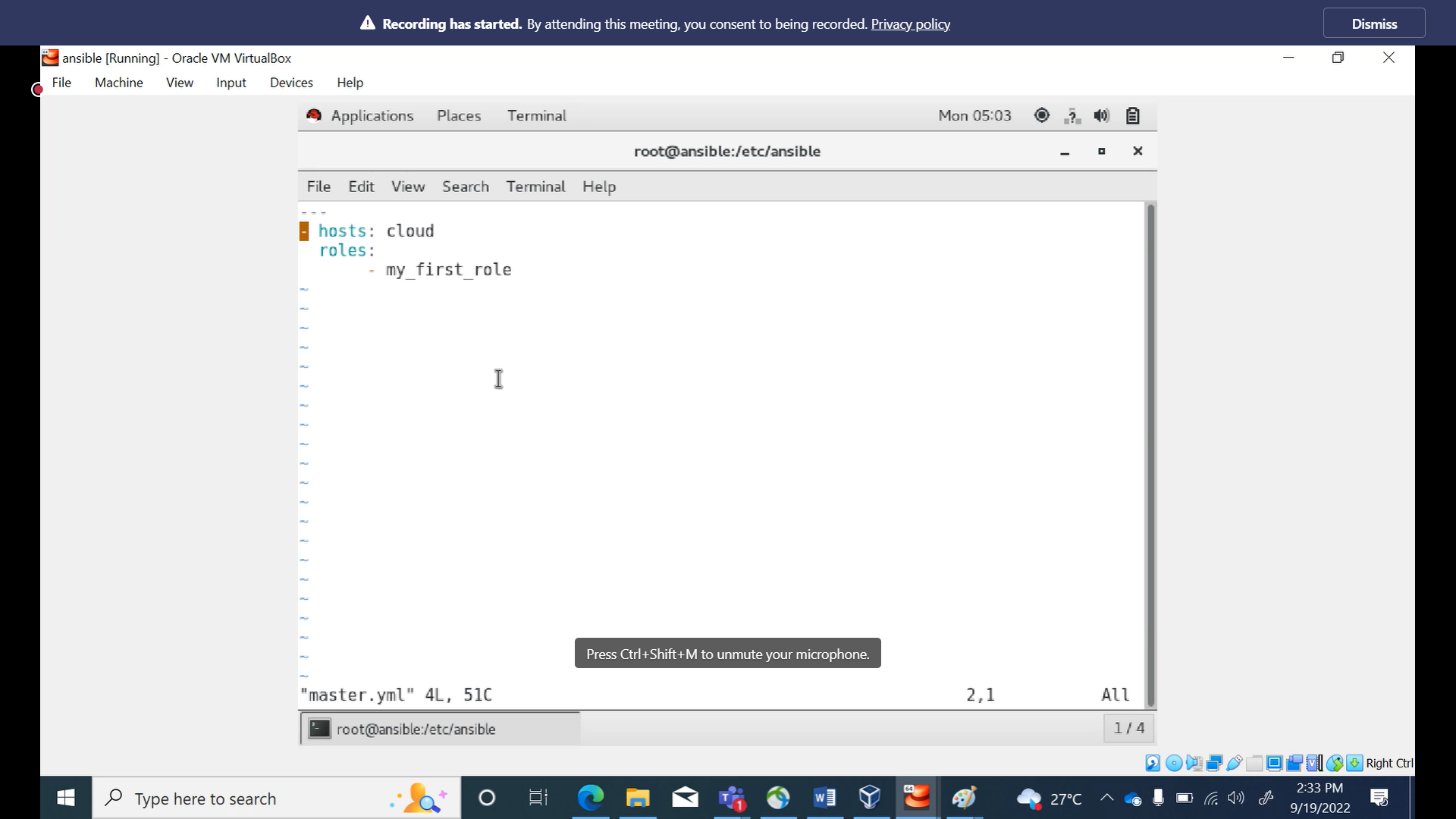
run **yum info tree** in nodes machine

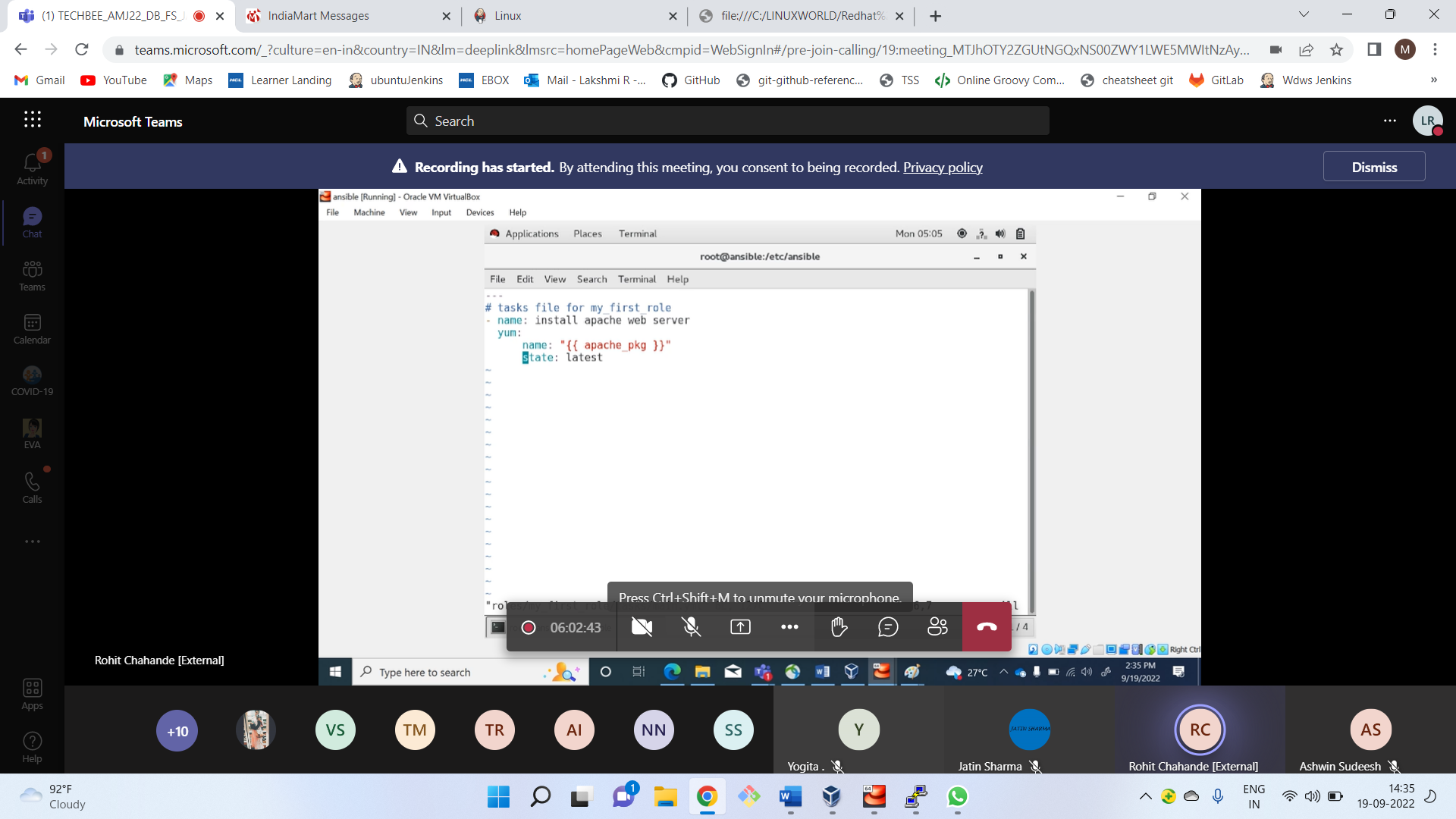


In nodes









Ansible Roles

Playbook have templates, variables, group vars and different roles in complex playbook.