Ansible

- Now we need to create 5 linux VMS
  - 1 for controller
  - 2 for dev
  - 2 for qa
- Now we need to login to Controller and do below

- Run sudo -i to got to root
- Install ansible using 2nd command
- Install git and python using second command
- Now run below command to add dev and qa nodes IP's to write in hosts file.
  - CMD: vi /etc/ansible/hosts
- Now we need to add dev and qa ip address in vi editor by going insert mode

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.
[dev]
172.31.35.105
172.31.44.91

[qa]
172.31.40.218
172.31.41.138

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group
```

- Save the file
- Now we need to un comment the ansible inventory and rootuser in ansible.cfg
    - vi /etc/ansible/ansible.cfg

```
# config file for ansible -- https://ansible.com/
# ===============================================

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]

# some basic default values...

inventory      = /etc/ansible/hosts
#library        = /usr/share/my_modules/
#module_utils   = /usr/share/my_module_utils/
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks          = 5
#poll_interval  = 15
sudo_user      = root
#ask_sudo_pass = True
#ask_pass      = True
#transport      = smart
#remote_port    = 22
#module_lang    = C
#module_set_locale = False
```

- we have removed the # from highlighted lines and saved
- now we have to create ansible user and set password
- Now we need to give root permission to ansible user
    - CMD: visudo

```
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
ansible   ALL=(ALL)   NOPASSWD:    ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
```

- Above we have given root permissions to ansible and saved the file.
- Now we need to set Password authenitication yes by below command
    - CMD: vim /etc/ssh/sshd_config
- Now we need to set Passwordauthentication as yes as below

```
#AuthorizedPrincipalsFile none


# For this to work you will also need host keys in /etc/ssh/ssh
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
```

- Save the file
- So for what we did commands below on controller

```
2.root@ip-172-31-61-180:~
[ec2-user@ip-172-31-61-180 ~]$ vi /etc/ansible/hosts
[ec2-user@ip-172-31-61-180 ~]$ sudo -i
[root@ip-172-31-61-180 ~]# vi /etc/ansible/hosts
[root@ip-172-31-61-180 ~]# vi /etc/ansible/ansible.cfg
[root@ip-172-31-61-180 ~]# useradd ansible
[root@ip-172-31-61-180 ~]# passwd ansible
Changing password for user ansible.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-61-180 ~]# visudo
[root@ip-172-31-61-180 ~]# vim /etc/ssh/sshd_config
[root@ip-172-31-61-180 ~]#
```

- Now we need to perform same as for few commands on node those commands are below

- We need to add ansible user and set pass
- Give sudo permission to ansible user
- Enable password authentication
- Now restart the sshd service as below

```
[root@ip-172-31-81-219 ~]# useradd ansible
[root@ip-172-31-81-219 ~]# passwd ansible
Changing password for user ansible.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-81-219 ~]# visudo
visudo: /etc/sudoers.tmp unchanged
[root@ip-172-31-81-219 ~]# visudo
[root@ip-172-31-81-219 ~]# vi /etc/ssh/sshd_config
[root@ip-172-31-81-219 ~]# restart sshd
-bash: restart: command not found
[root@ip-172-31-81-219 ~]# systemctl restart sshd
[root@ip-172-31-81-219 ~]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2023-01-29 09:00:09 UTC; 14s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 32370 (sshd)
   CGroup: /system.slice/sshd.service
           └─32370 /usr/sbin/sshd -D

Jan 29 09:00:09 ip-172-31-81-219.ec2.internal systemd[1]: Stopped OpenSSH server daemon.
Jan 29 09:00:09 ip-172-31-81-219.ec2.internal systemd[1]: Starting OpenSSH server daemon...
Jan 29 09:00:09 ip-172-31-81-219.ec2.internal sshd[32370]: Server listening on 0.0.0.0 port 22.
Jan 29 09:00:09 ip-172-31-81-219.ec2.internal sshd[32370]: Server listening on :: port 22.
Jan 29 09:00:09 ip-172-31-81-219.ec2.internal systemd[1]: Started OpenSSH server daemon.
[root@ip-172-31-81-219 ~]#
```

- Do above in all nodes

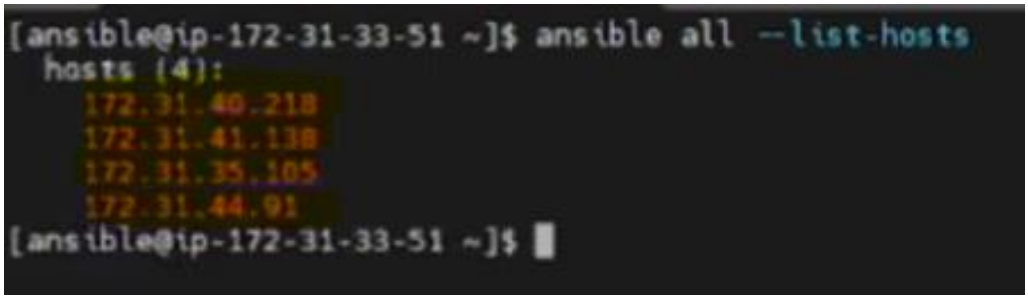
- Now login as ansible user in all servers using below command
    - CMD: su – ansible
- Now go to Ansible controller and generate ssh and copy in controller and nodes
- Go to controller run below
- Generate the ssh
    - CMD: ssh-keygen
- It will generate key like below

```
[root@ip-172-31-61-180 ~]# su - ansible
Last login: Sun Jan 29 09:04:49 UTC 2023 on pts/2
[ansible@ip-172-31-61-180 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansible/.ssh/id_rsa):
Created directory '/home/ansible/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ansible/.ssh/id_rsa.
Your public key has been saved in /home/ansible/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:h7R8S2DQ10wYzkeYhoL9+JkA4OCURuwaCkSsreH8oUM ansible@ip-172-31-61-180.ec2.internal
The key's randomart image is:
+---[RSA 2048]----+
|*+o. o..  ..Oo    |
|oO  o o.oo*.o     |
|=o.   . ++oo .    |
|= ..    o+.+ .    |
|*+      oSo+      |
|+E .     ++ .     |
|. o .      .      |
| o .              |
|  .               |
+----[SHA256]-----+
[ansible@ip-172-31-61-180 ~]$
```

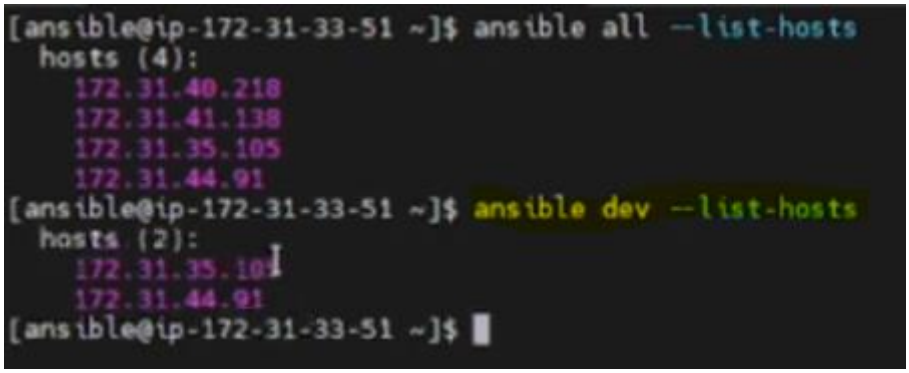
- Now copy the key in controller
    - CMD: ssh-copy-id ansible@controller ip (or) localhost
    - Or
    - CMD: ssh-copy-id <controller ip>
    - It will ask yes or no give as yes
    - And then give the password what you have given to ansible user
- Now copy key to key to node
    - CMD: ssh-copyy-id ansible@ip of node
    - Same give asnsible password
- Now do the same to copy the ssh id in all nodes by
    - CMD: ssh-copyy-id ansible@ip of node

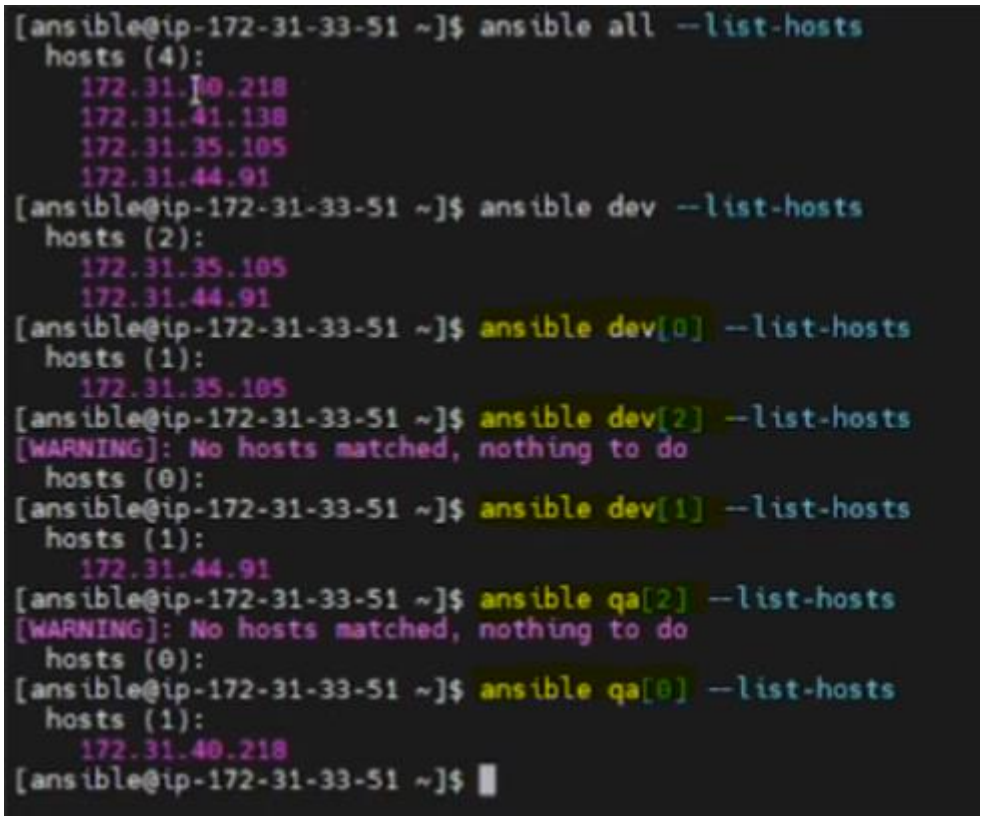
Setup has been completed

- Now to check the ansible inventory use below command it will show the nodes we have listed in hosts file above in controller
    - CMD: ansible all --list-hosts



- So now we can do the configuration on all 4 servers registered from controller
- To check node in particular environment
    - CMD: ansible dev –list-hosts





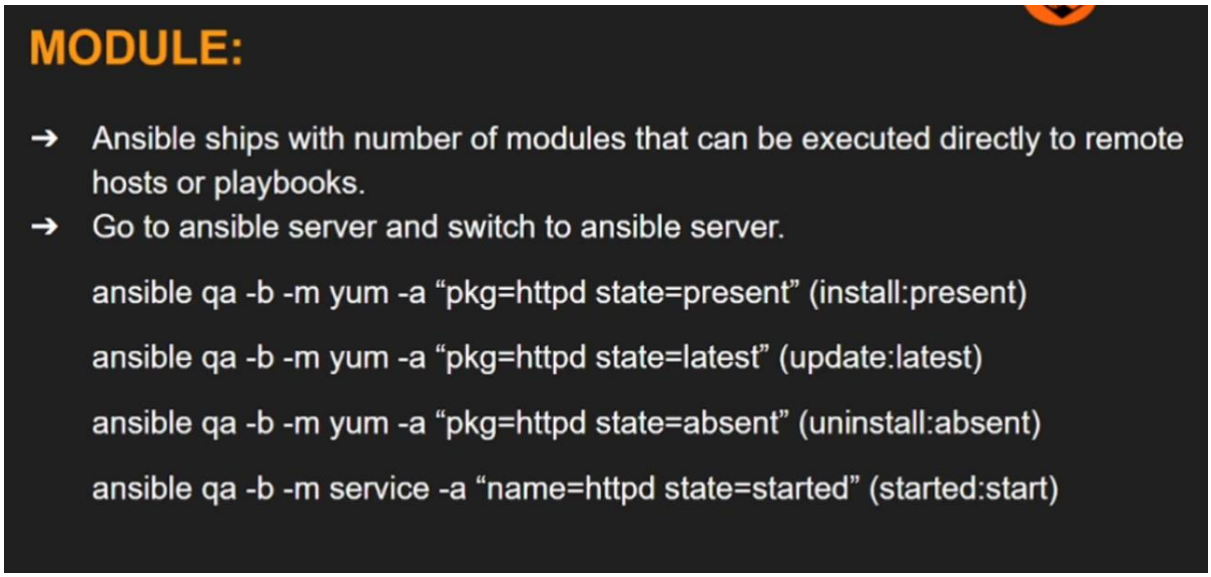
- Now we want to create file1 on all dev nodes using bleow command (below is Ad-hoc command)
- Ad-hoc commands are one time use and they can’t be reuse and this will not use in automation, this will only use an immediate and temporarily use.

- CMD: ansible dev -a “touch file1”
- It will create file1 in all dev nodes
- To check file list in all nodes
    - CMD: ansible dev -a “ls”
- Now we want to install git in all dev use below command
    - CMD: ansible dev -a “sudo yum install git -y”
- Now we want to delete the file1 from dev nodes use below commad
    - CMD: ansible dev -a “rm -rf file1”
- Now we need to install maven use below any one command
    - CMD: ansible dev -a “sudo yum install maven -y”
    - CMD: ansible dev -ba “ yum install maven -y”

Note: above we have used -ba and didn’t use sudo. -ba will work like sudo

Ansible Modules

- Ansible modules can be reused
- Ansible modules can be use like below



Ansible-Playbooks

- Ansible-playbooks are heart of the ansible
- Playbooks can use to automate configuration
- Playbooks can write in YAML language
- This will be used in DevOps

PLAYBOOKS:

- ➔ Playbooks in ansible are written in YAML Language.
- ➔ It is human readable & serialization language commonly used for configuration files.
- ➔ You can write codes consists of vars, tasks, handlers, files, templates and roles.
- ➔ Playbooks are mainly divided into sections like

Target Section

Variable Section

Task Section

YAML(Yet another markup language):

For ansible ,nearly every YAML file starts with a list

- Each item in the list is a key-value pair commonly called dictionary.
- All YAML files begin with "---" and end with "---".
- A dictionary is require a simple key form(note:space before value is must).

Name:ragu age:20

- Extension for playbook file is .yml

Go to ansible server and login as ansible and create one playbook

- ➢ Vim abc.yml

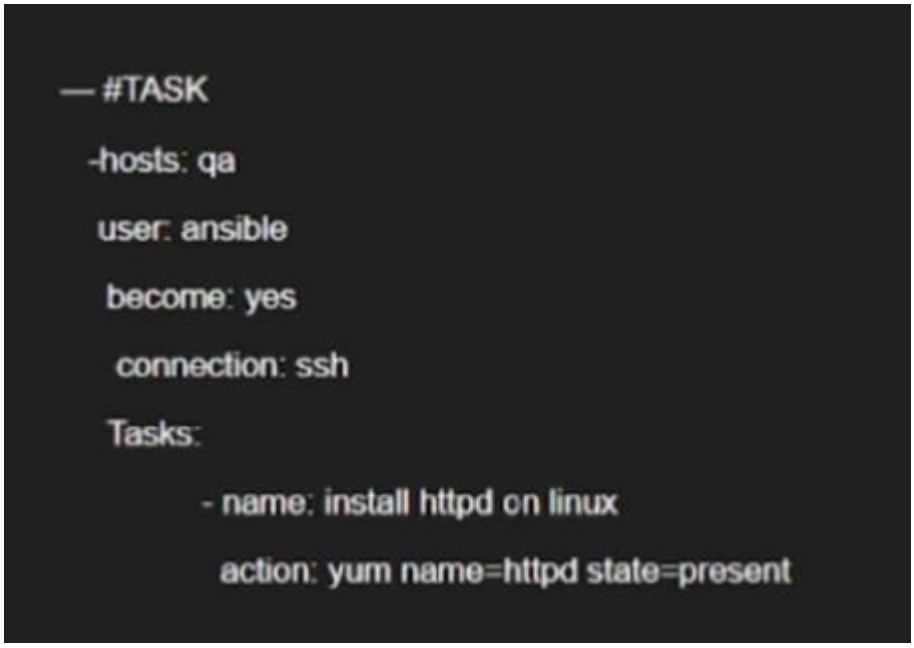
- ➢ Hosts: qa - qa:groupname

User: ansible - ansible:you are ansible user now

Become: yes -become: become sudo user

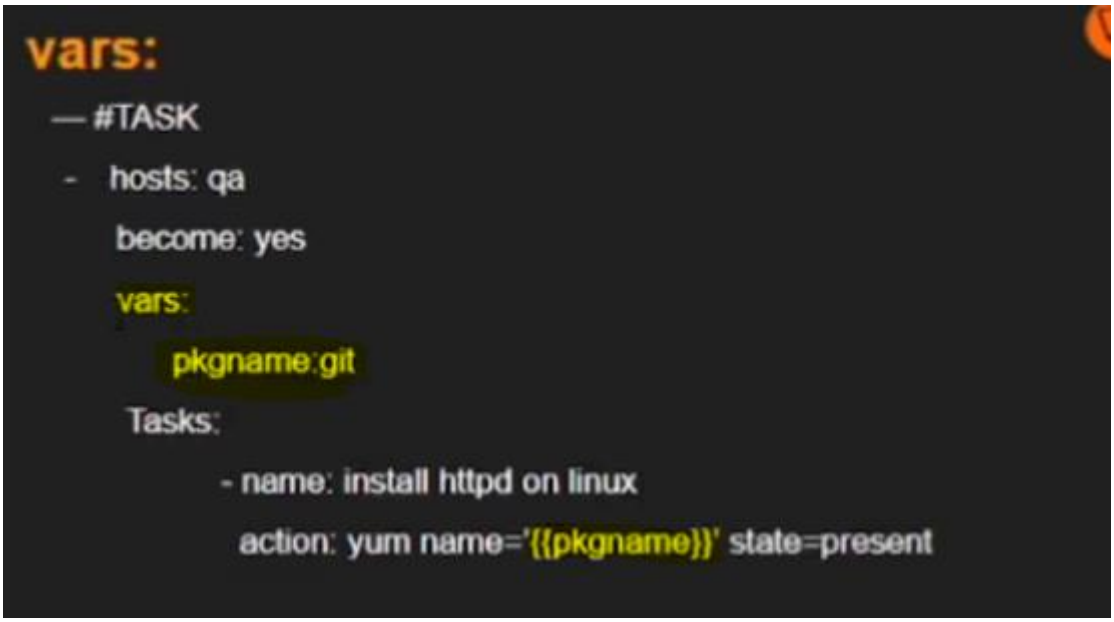
Connection: ssh

- Now we want to install httpd on all qa servers so we need to create the abc.yaml that is playbook file
    - CMD: vim abc.yaml



- Now save the file and run below command to execute the playbook
    - CMD: ansibe-playbook abc.yaml
- The above command will install the httpd

Vars in Ansible:



Note: above we haven't include the connection:ssh, by default ansible connect on ssh

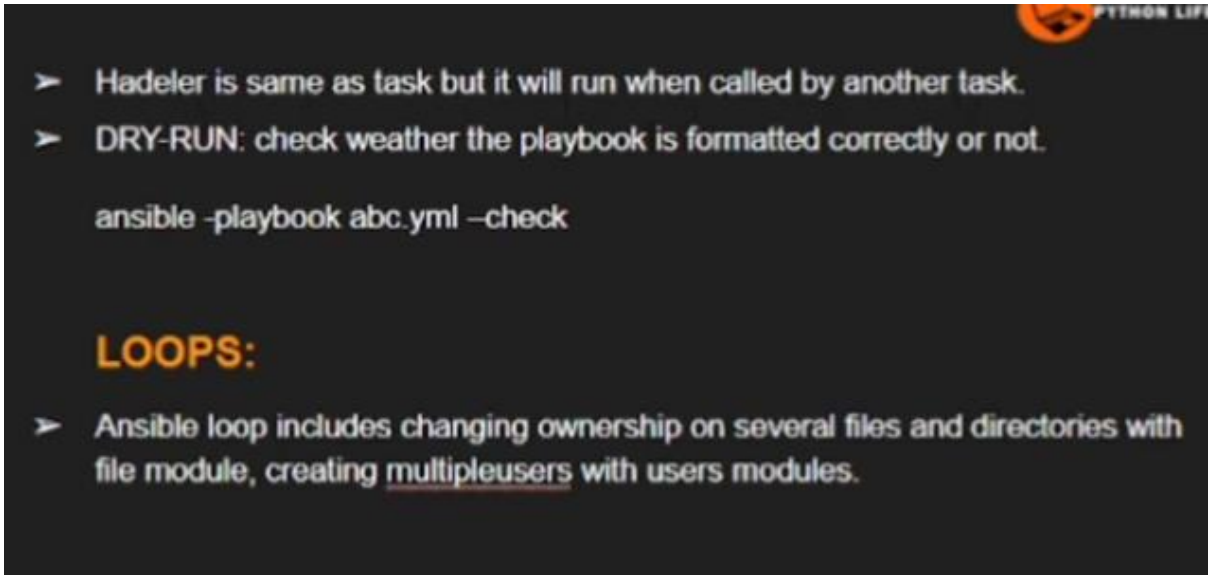
- Installing maven

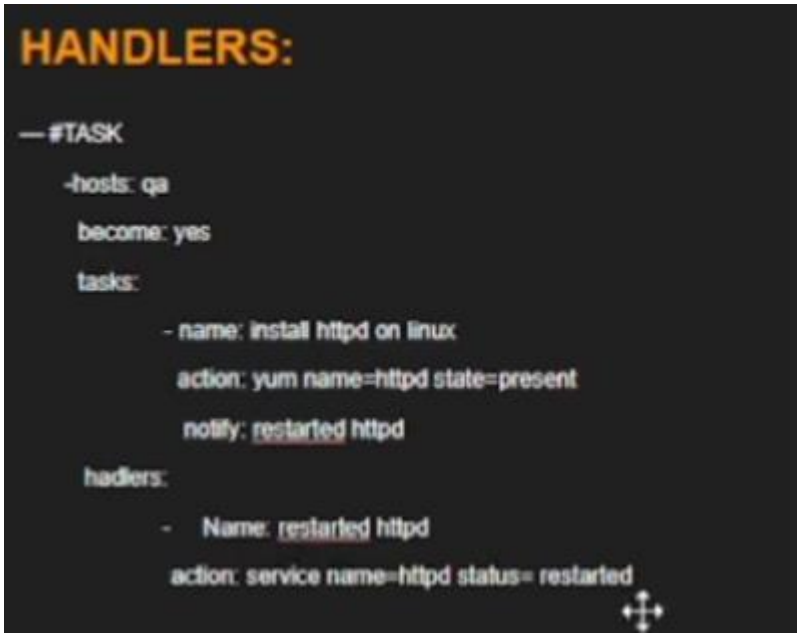
```
---
# THIS IS MY FIRST PLAYBOOK
- hosts: qa
  become: yes
  vars:
    pkgname: maven
  tasks:
    - name: INSTALLING MAVEN ON QA SERVERS
      action: yum name='{{pkgname}}' state=present
```

- Uninstalling maven

```
---
# THIS IS MY FIRST PLAYBOOK
- hosts: qa
  become: yes
  vars:
    pkgname: maven
  tasks:
    - name: INSTALLING MAVEN ON QA SERVERS
      action: yum name='{{pkgname}}' state=absent
```

Ansible Handlers:






```
--- # THIS IS MY FIRST PLAYBOOK
- hosts: qa
  become: yes
  tasks:
    - name: INSTALLING httpd ON QA SERVERS
      action: yum name=httpd state=installed
      notify: restart httpd
  handlers:
    - name: restarted httpd
      action: service name=httpd status=restarted
~
~
```

Ansible Handlers:

```
--- # THIS IS MY FIRST PLAYBOOK
- hosts: qa
  become: yes
  tasks:
    - name: adding users
      user: name='{{item}}' state=present
      with_items:
              - vikky
              - ritu
              - viyu
              - abc
```

- The above will create the multiple users once