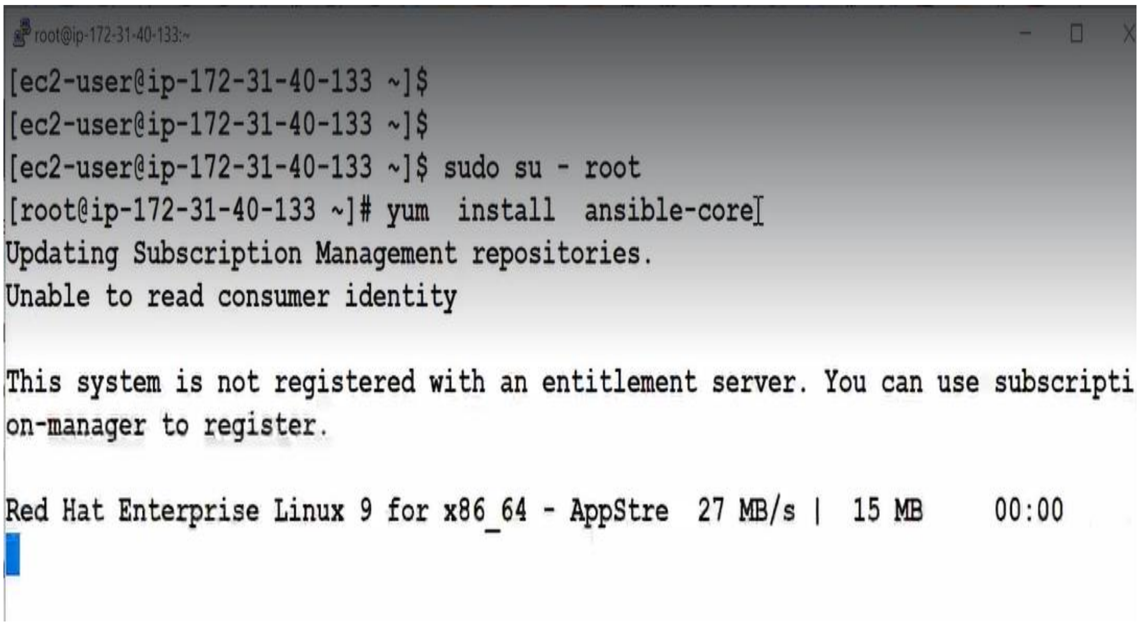
Ansible

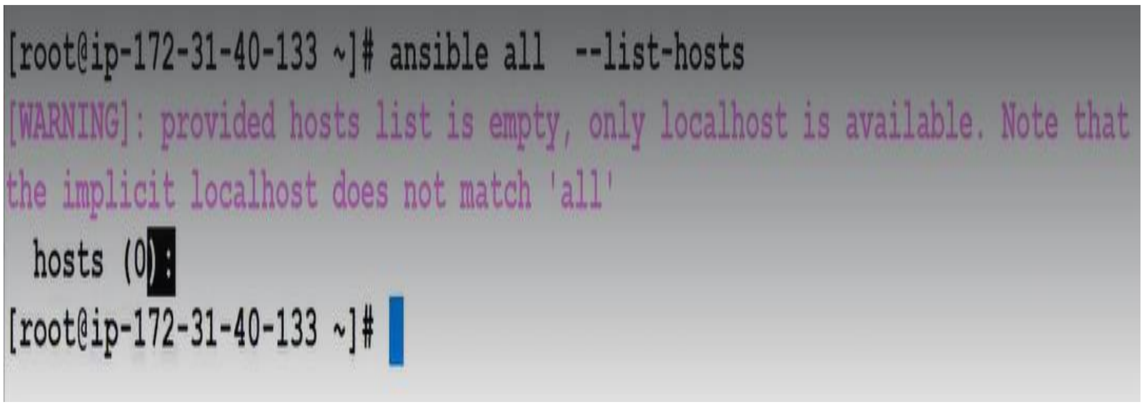
Session 2 – 3rd Dec 2022 Summary

- The node on which ansible is installed, to perform configuration management on multiple nodes by a **code** remotely is called as **Manager or Controller node.** The nodes that are managed are called as **Managed or Target node.**
- The minimum requirement for the manager node to connect to managed node is the **network connectivity,** behind networking there is a way called **Protocol.** For example if the managed node is Linux, the protocol used is ssh.
- For the ansible on controller node to connect and manage the target node, we have to give information about the IP Address, username, password and protocol name used of the target node that has to be managed. These information are called as **inventory.**
- For this, we can use three systems- one as controller node and other two as target node.
- Using Putty, Login to root user of the controller node using the command "**sudo su – root**"



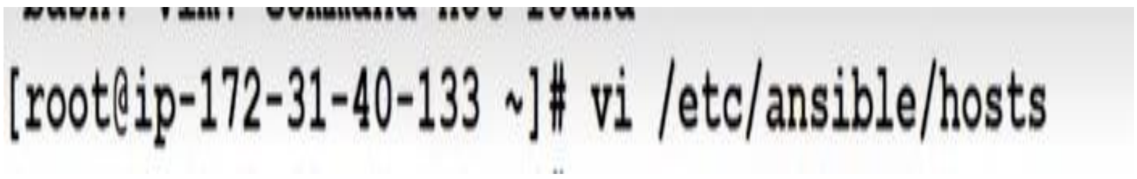
```
root@ip-172-31-40-133:~
[ec2-user@ip-172-31-40-133 ~]$
[ec2-user@ip-172-31-40-133 ~]$
[ec2-user@ip-172-31-40-133 ~]$ sudo su - root
[root@ip-172-31-40-133 ~]# yum  install  ansible-core
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscripti
on-manager to register.

Red Hat Enterprise Linux 9 for x86_64 - AppStre  27 MB/s |  15 MB     00:00
```

- The command to list all the host in the inventory “**ansible all - - list-hosts**”

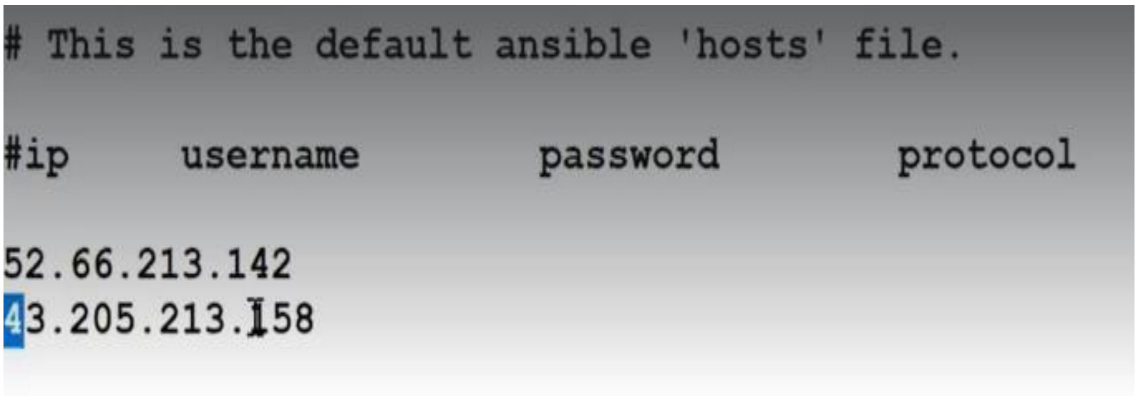


```
[root@ip-172-31-40-133 ~]# ansible all  --list-hosts
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'
  hosts (0):
[root@ip-172-31-40-133 ~]#
```

- To create inventory, open a file “/etc/ansible/hosts”, here we specify only the IP Address of the managed node

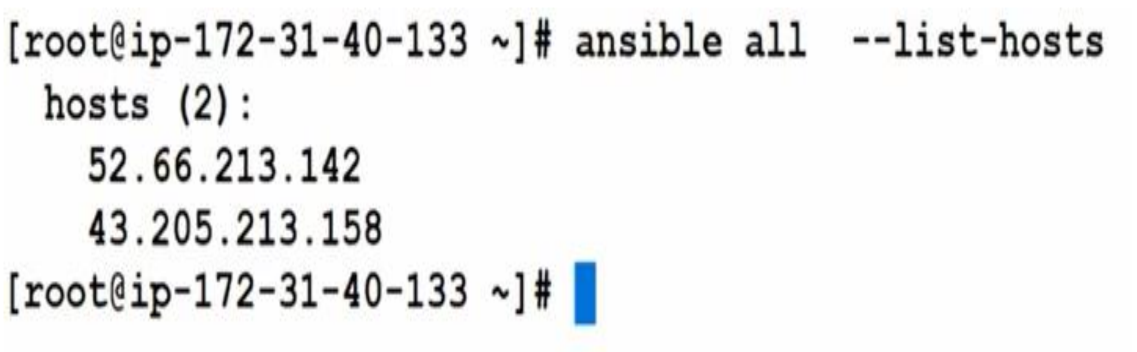


```
[root@ip-172-31-40-133 ~]# vi /etc/ansible/hosts
```



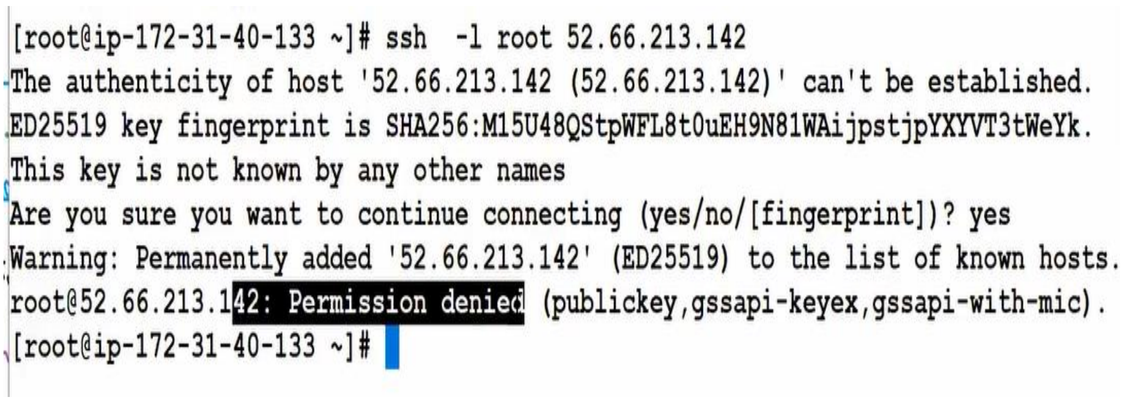
```
# This is the default ansible 'hosts' file.

#ip     username        password        protocol

52.66.213.142
43.205.213.158
```

- The command to list all the host in the inventory “**ansible all - - list-hosts**”



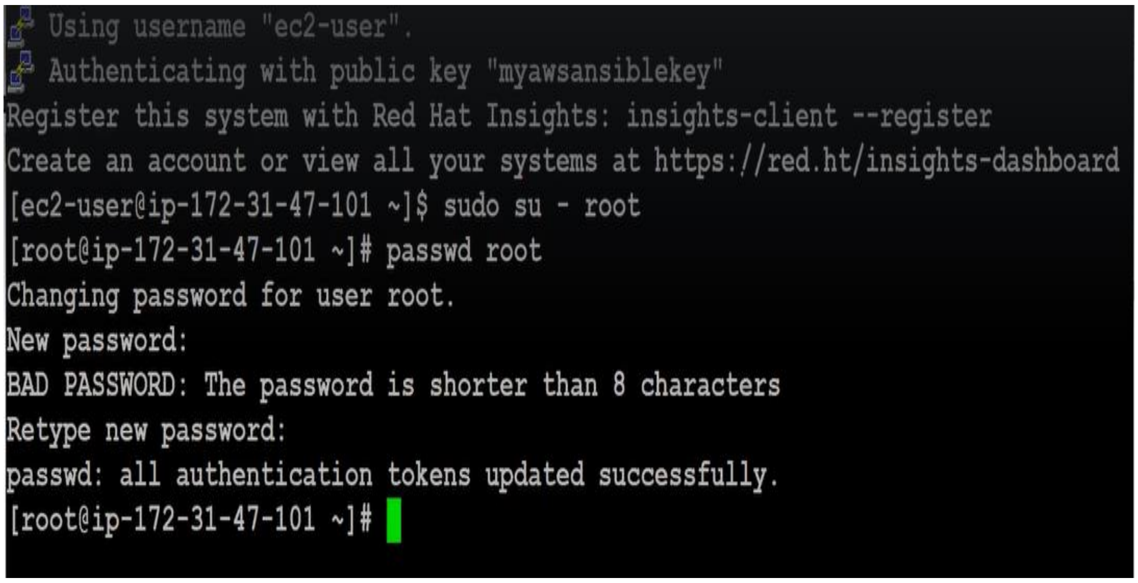
```
[root@ip-172-31-40-133 ~]# ansible all  --list-hosts
  hosts (2):
    52.66.213.142
    43.205.213.158
[root@ip-172-31-40-133 ~]#
```

➢ For configuration management you have to login as “root” user. From controller node we can check manually that we are able to perform remote login to managed node with IP Address “52.66.213.142” using the command “**ssh –l root 52.66.213.142**”, but we see here “permission denied”, since in AWS the “root” account is **blocked** for security reasons.



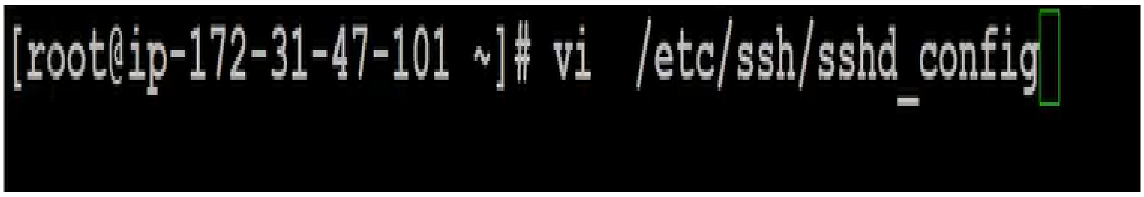
```
[root@ip-172-31-40-133 ~]# ssh  -l root 52.66.213.142
The authenticity of host '52.66.213.142 (52.66.213.142)' can't be established.
ED25519 key fingerprint is SHA256:M15U48QStpWFL8t0uEH9N81WAijpstjpYXYVT3tWeYk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.66.213.142' (ED25519) to the list of known hosts.
root@52.66.213.142: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[root@ip-172-31-40-133 ~]#
```

➢ For this, we have to **enable** “root” account. First thing using Putty, Login as root user to one of the managed node and set root password



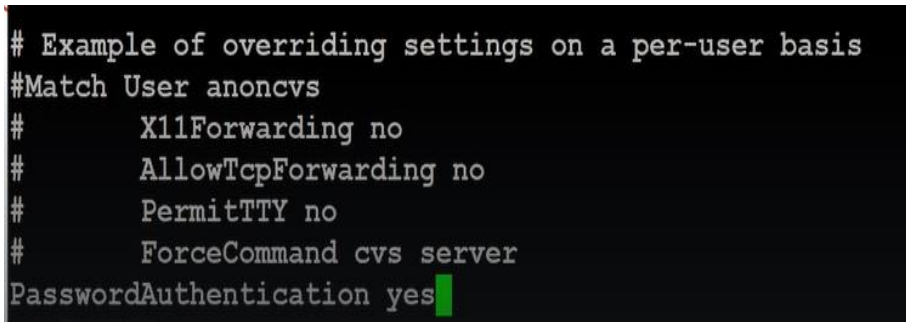
```
Using username "ec2-user".
Authenticating with public key "myawsansiblekey"
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-47-101 ~]$ sudo su - root
[root@ip-172-31-47-101 ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-47-101 ~]#
```

➢ Second thing, open the main configuration file of ssh “/etc/ssh/sshd_config”



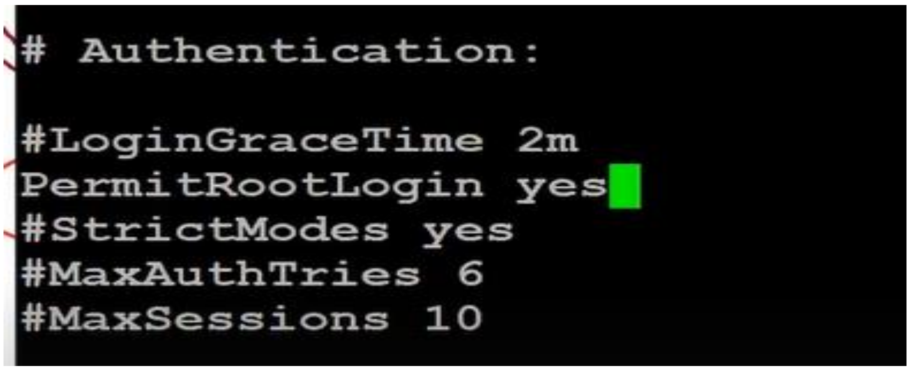
```
[root@ip-172-31-47-101 ~]# vi  /etc/ssh/sshd_config
```

- Set “PasswordAuthentication yes”



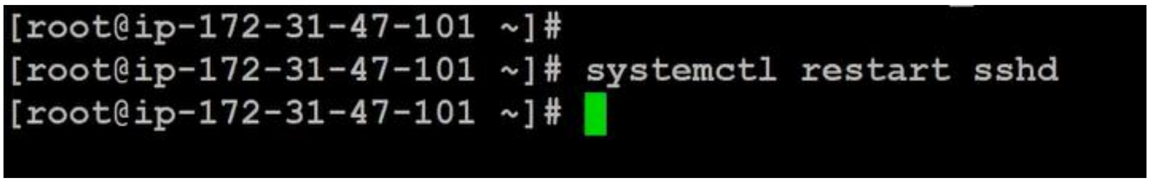
```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
```

- Set “PermitRootLogin yes”



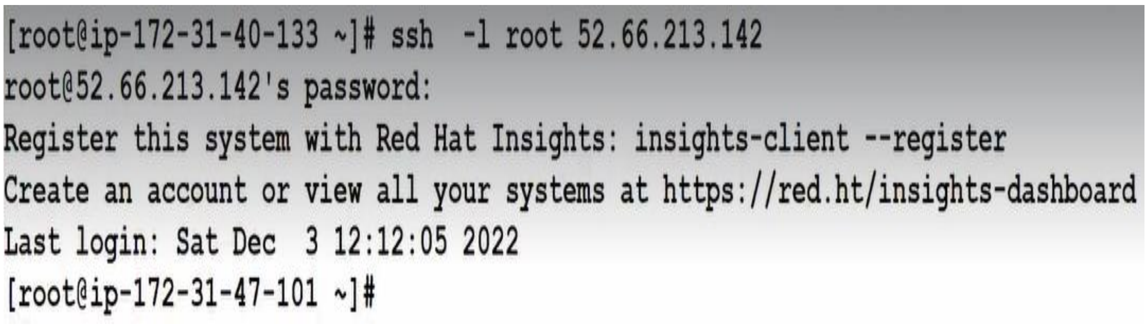
```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

➢ After making changes in the main configuration file, restart the service “sshd”



```
[root@ip-172-31-47-101 ~]#
[root@ip-172-31-47-101 ~]# systemctl restart sshd
[root@ip-172-31-47-101 ~]#
```

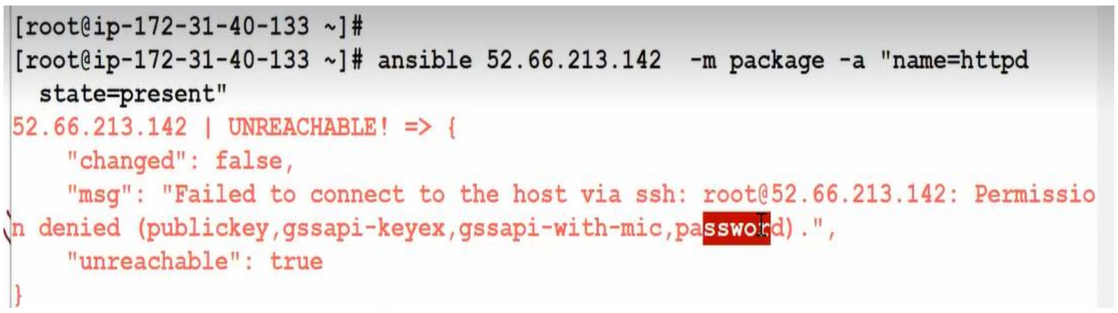
➢ Now we are able to perform remote login to managed node with IP Address “52.66.213.142” using the command “**ssh –l root 52.66.213.142**”



```
[root@ip-172-31-40-133 ~]# ssh  -l root 52.66.213.142
root@52.66.213.142's password:
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Sat Dec  3 12:12:05 2022
[root@ip-172-31-47-101 ~]#
```

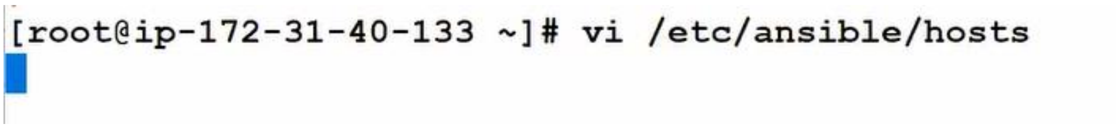
➢ The same steps can be performed on other managed node

- The package management can be done using “ansible adhoc commands”. The keyword used for package management is “**package**” and for this we have **modules** (- **m**). To specify the name of a package we have a keyword **name**. To specify either to install software (**present**) or remove a software (**absent**) we have a “**state**” keyword, the **name** and **state** keyword are put inside a **double quotes,** these are called as **attributes (-a)**
- Here the package management is done on the managed node “52.66.213.142”,we see here that the default protocol used is “ssh” and the user is “root” but the permission denied because we have to specify the root “password” in the inventory file

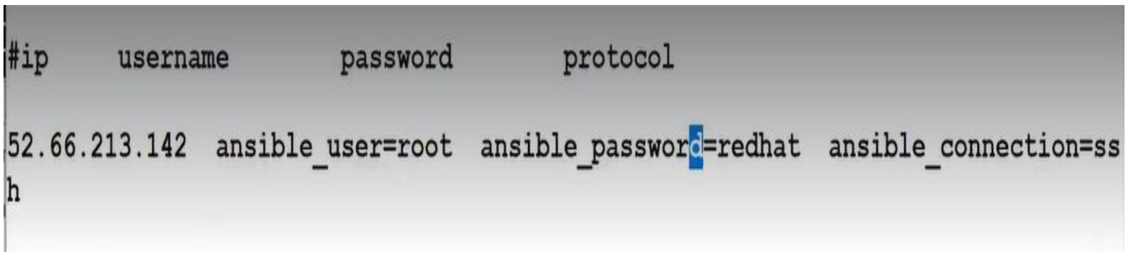


```
[root@ip-172-31-40-133 ~]#
[root@ip-172-31-40-133 ~]# ansible 52.66.213.142  -m package -a "name=httpd
  state=present"
52.66.213.142 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: root@52.66.213.142: Permissio
n denied (publickey,gssapi-keyex,gssapi-with-mic,password).",
    "unreachable": true
}
```

- For this open a file “/etc/ansible/hosts”, here we specify the information of the managed node along with the keywords.

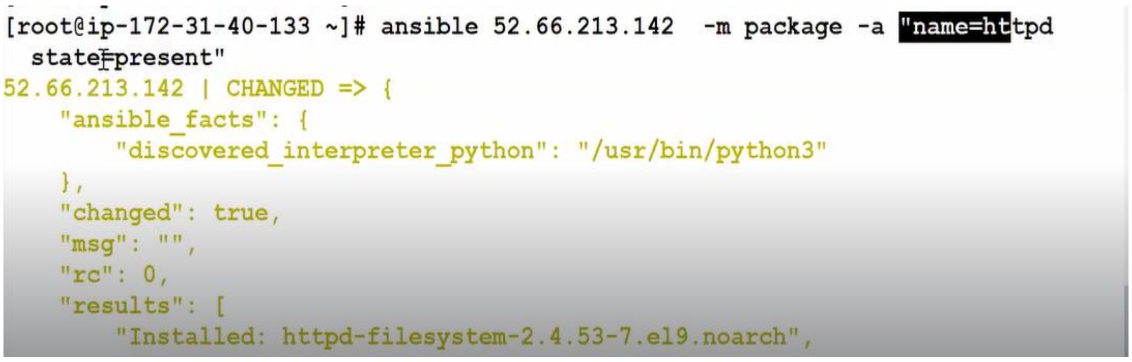


```
[root@ip-172-31-40-133 ~]# vi /etc/ansible/hosts
```



```
#ip      username        password        protocol

52.66.213.142  ansible_user=root  ansible_password=redhat  ansible_connection=ss
h
```

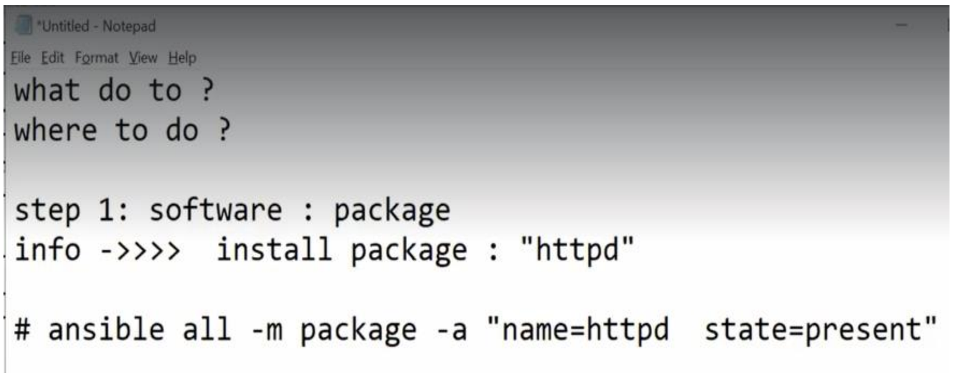
- Now we run the same command, the package “httpd” is installed



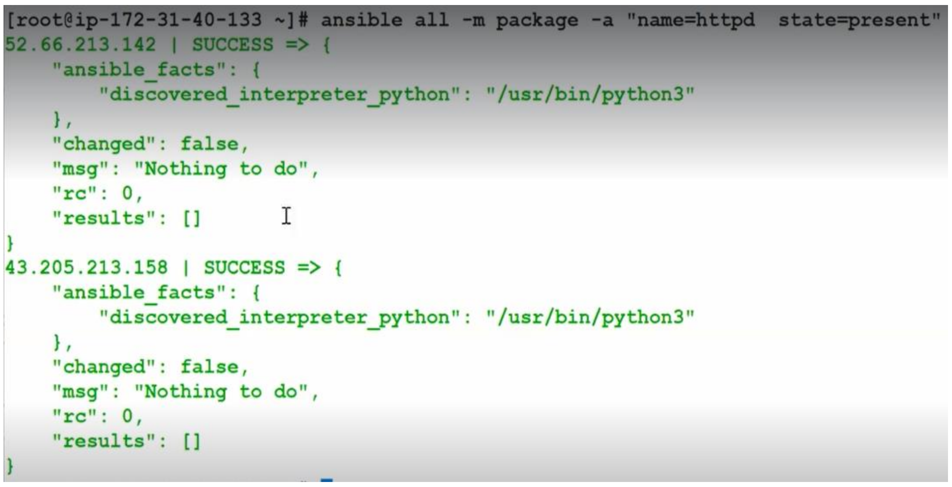
```
[root@ip-172-31-40-133 ~]# ansible 52.66.213.142  -m package -a "name=httpd
  state=present"
52.66.213.142 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        "Installed: httpd-filesystem-2.4.53-7.el9.noarch",
```

- The manual way of configuration management is by using **ansible adhoc commands**, for automation we have to write a code in a program file called **Playbook**
- To configure Apache HTTPD Webserver, these are the three steps to follow. First we can have ansible adhoc commands, then converting these command into automation is simpler

  - Step 1:- to install the package “httpd” on all managed nodes

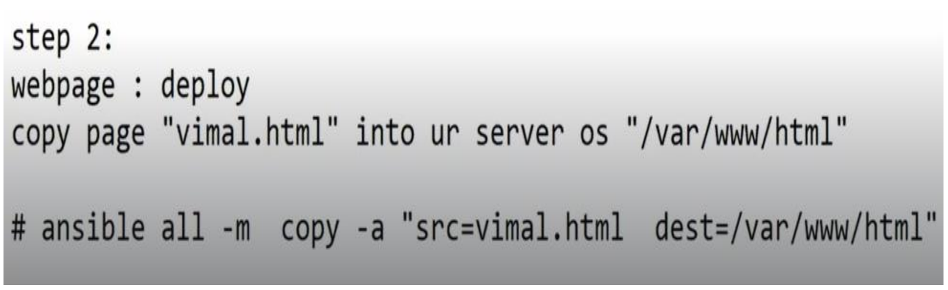


```
what do to ?
where to do ?

step 1: software : package
info ->>>>  install package : "httpd"

# ansible all -m package -a "name=httpd  state=present"
```

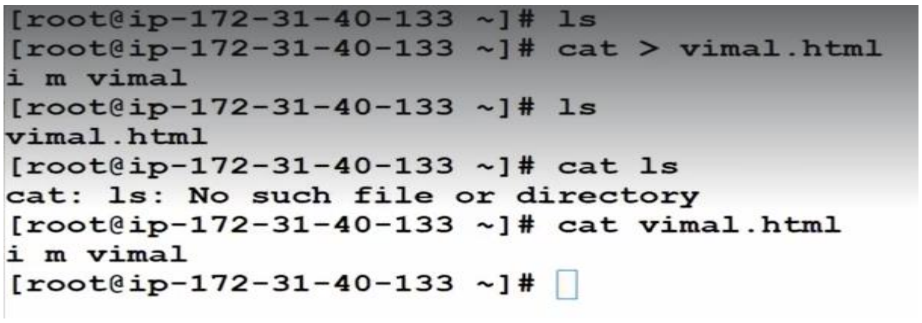


```
[root@ip-172-31-40-133 ~]# ansible all -m package -a "name=httpd  state=present"
52.66.213.142 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "msg": "Nothing to do",
    "rc": 0,
    "results": []
}
43.205.213.158 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "msg": "Nothing to do",
    "rc": 0,
    "results": []
}
```

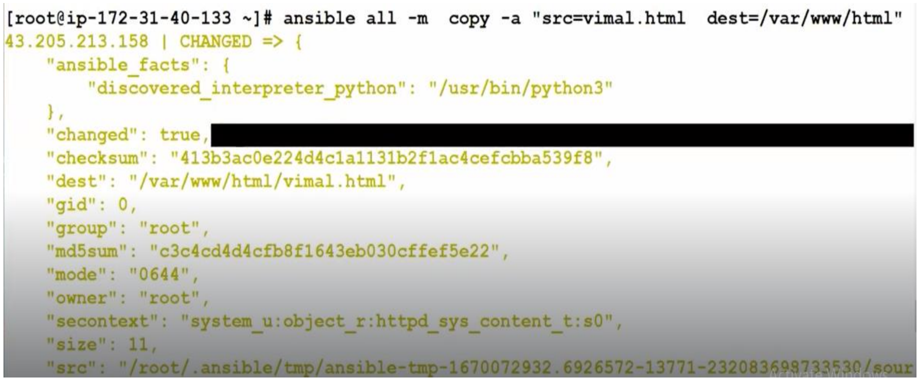
  - Step 2:- Create webpages on the controller node and copy these to all the managed nodes



```
step 2:
webpage : deploy
copy page "vimal.html" into ur server os "/var/www/html"

# ansible all -m  copy -a "src=vimal.html  dest=/var/www/html"
```

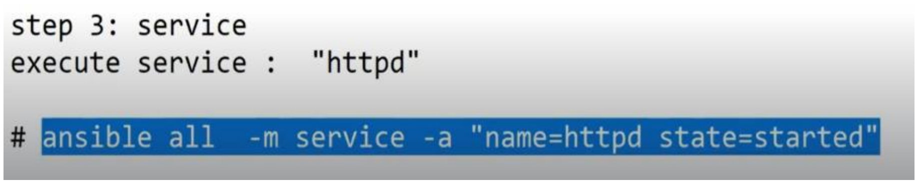


```
[root@ip-172-31-40-133 ~]# ls
[root@ip-172-31-40-133 ~]# cat > vimal.html
i m vimal
[root@ip-172-31-40-133 ~]# ls
vimal.html
[root@ip-172-31-40-133 ~]# cat ls
cat: ls: No such file or directory
[root@ip-172-31-40-133 ~]# cat vimal.html
i m vimal
[root@ip-172-31-40-133 ~]#
```

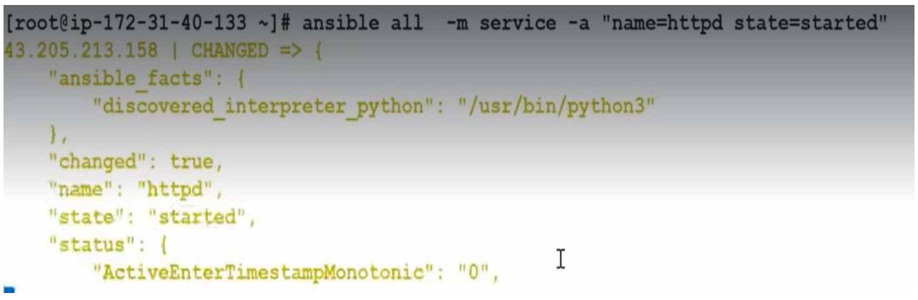


```
[root@ip-172-31-40-133 ~]# ansible all -m  copy -a "src=vimal.html  dest=/var/www/html"
43.205.213.158 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,[redacted]
    "checksum": "413b3ac0e224d4c1a1131b2f1ac4cefcbba539f8",
    "dest": "/var/www/html/vimal.html",
    "gid": 0,
    "group": "root",
    "md5sum": "c3c4cd4d4cfb8f1643eb030cffef5e22",
    "mode": "0644",
    "owner": "root",
    "secontext": "system_u:object_r:httpd_sys_content_t:s0",
    "size": 11,
    "src": "/root/.ansible/tmp/ansible-tmp-1670072932.6926572-13771-232083698733530/sour
```

- Step 3:- to start the "httpd" service

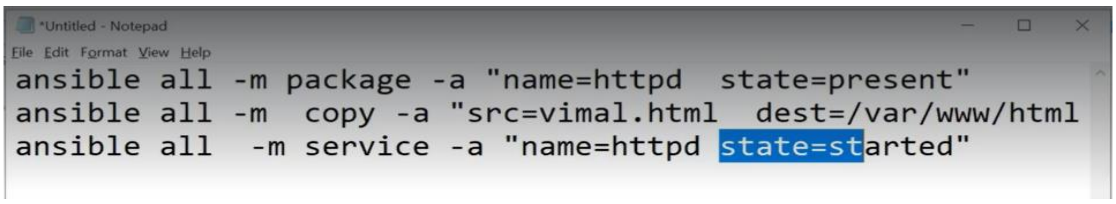


```
step 3: service
execute service :   "httpd"

# ansible all  -m service -a "name=httpd state=started"
```



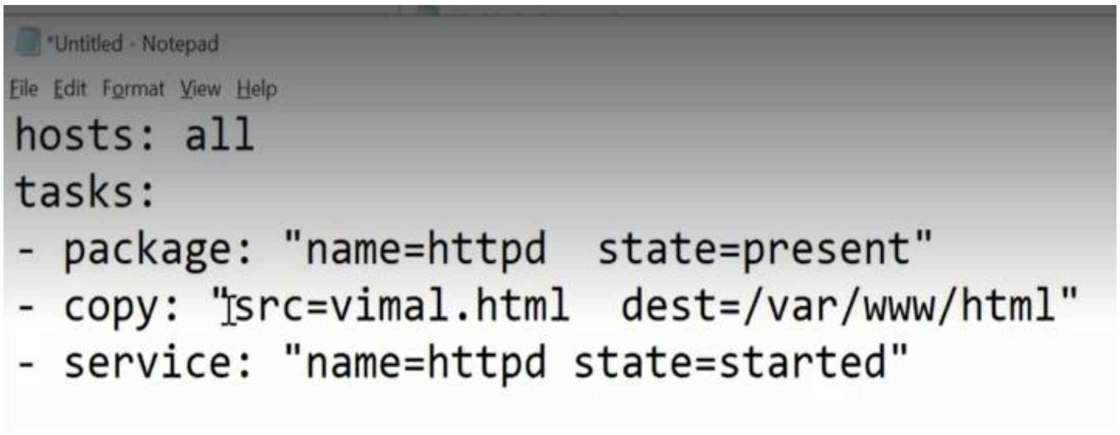
```
[root@ip-172-31-40-133 ~]# ansible all  -m service -a "name=httpd state=started"
43.205.213.158 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "name": "httpd",
    "state": "started",
    "status": {
        "ActiveEnterTimestampMonotonic": "0",
```

- For automation we have to write a code in a program file called **Playbook**
- The ansible use a syntax from YAML(yet another markup language)
- The ansible adhoc commands from the three steps can be converted into a playbook.



```
ansible all -m package -a "name=httpd  state=present"
ansible all -m  copy -a "src=vimal.html  dest=/var/www/html
ansible all  -m service -a "name=httpd state=started"
```

- For the playbook we do not require the command “ansible”. The **value** “all” is used once and we can use **keyword** “host” that means all modules are run on all the managed nodes. The keyword and value are separated by colon “**:**”
- Every module like package, copy, service are doing one task, so these can be put under “task” keyword. The list of tasks are started with hyphen “-“. The keyword “package” and the attributes are separated by colon “:”

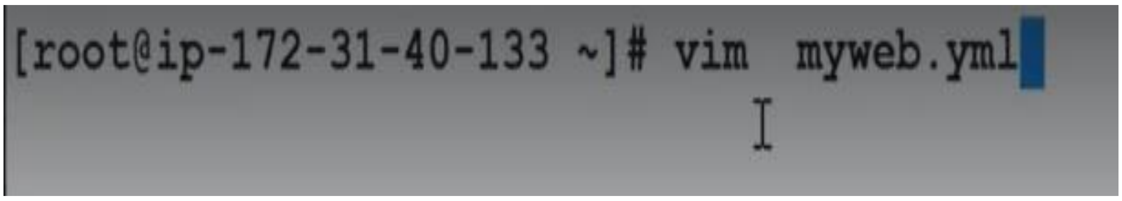


*Untitled - Notepad

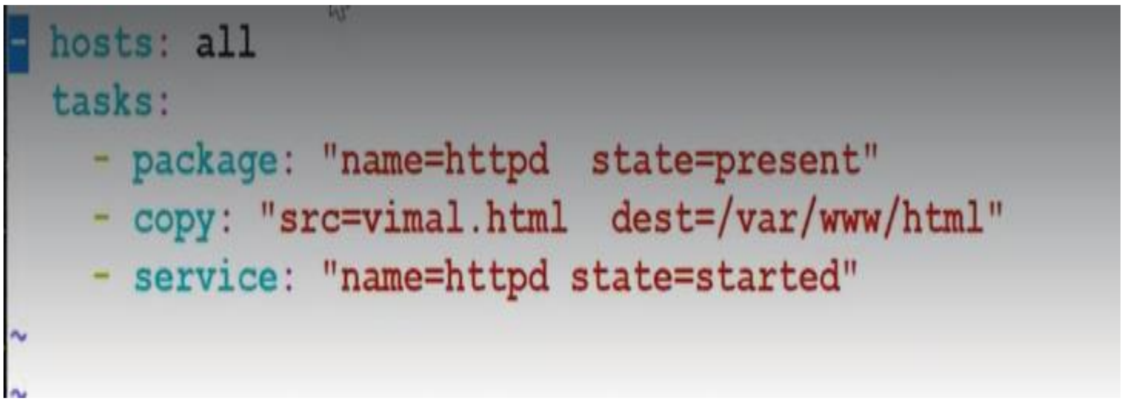
File Edit Format View Help

```
hosts: all
tasks:
- package: "name=httpd  state=present"
- copy: "src=vimal.html  dest=/var/www/html"
- service: "name=httpd state=started"
```

- The command to create a playbook file “**vim myweb.yml**”

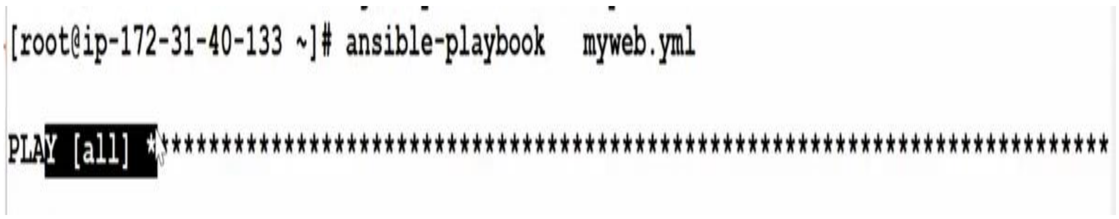


```
[root@ip-172-31-40-133 ~]# vim  myweb.yml
```

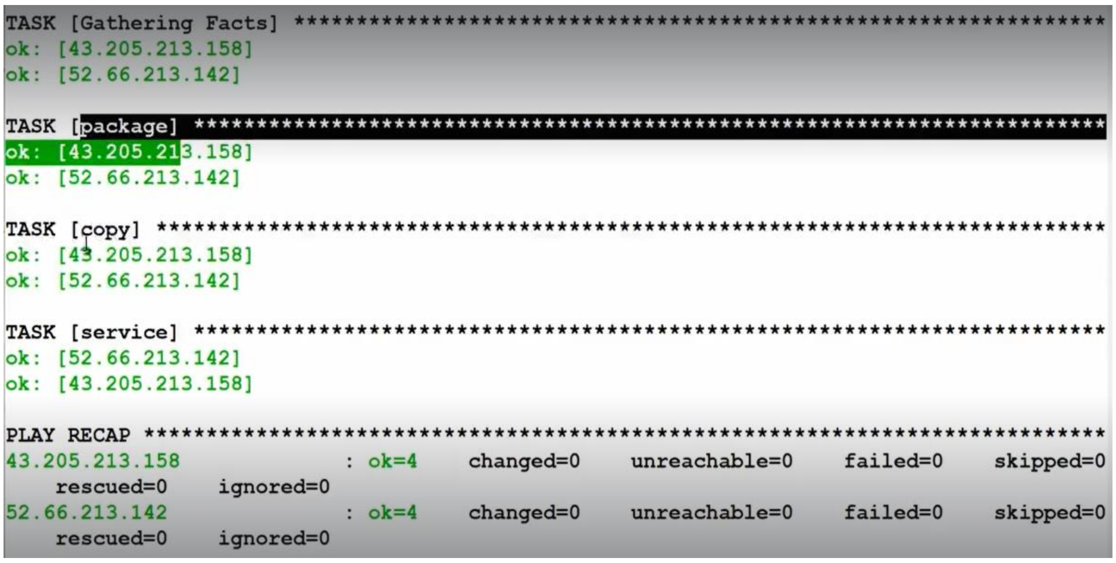


```
- hosts: all
  tasks:
    - package: "name=httpd  state=present"
    - copy: "src=vimal.html  dest=/var/www/html"
    - service: "name=httpd state=started"
~
~
```

- The command to run the playbook “**ansible-playbook myweb.yml**”



```
[root@ip-172-31-40-133 ~]# ansible-playbook   myweb.yml

PLAY [all] ***************************************************************************
```



```
TASK [Gathering Facts] *****************************************************************
ok: [43.205.213.158]
ok: [52.66.213.142]

TASK [package] *************************************************************************
ok: [43.205.213.158]
ok: [52.66.213.142]

TASK [copy] ****************************************************************************
ok: [43.205.213.158]
ok: [52.66.213.142]

TASK [service] *************************************************************************
ok: [52.66.213.142]
ok: [43.205.213.158]

PLAY RECAP *****************************************************************************
43.205.213.158             : ok=4    changed=0    unreachable=0    failed=0    skipped=0
    rescued=0    ignored=0
52.66.213.142              : ok=4    changed=0    unreachable=0    failed=0    skipped=0
    rescued=0    ignored=0
```

Important Links –

Hash13 link for Session and extra sessions recordings – https://learning.hash13.com/

Ansible Documentation- Copy Module - https://docs.ansible.com/ansible/latest/collections/ansible/builtin/copy_module.html

Ansible Documentation- Package Module - https://docs.ansible.com/ansible/latest/collections/ansible/builtin/package_module.html

Ansible Documentation- Service Module – https://docs.ansible.com/ansible/latest/collections/ansible/builtin/service_module.html