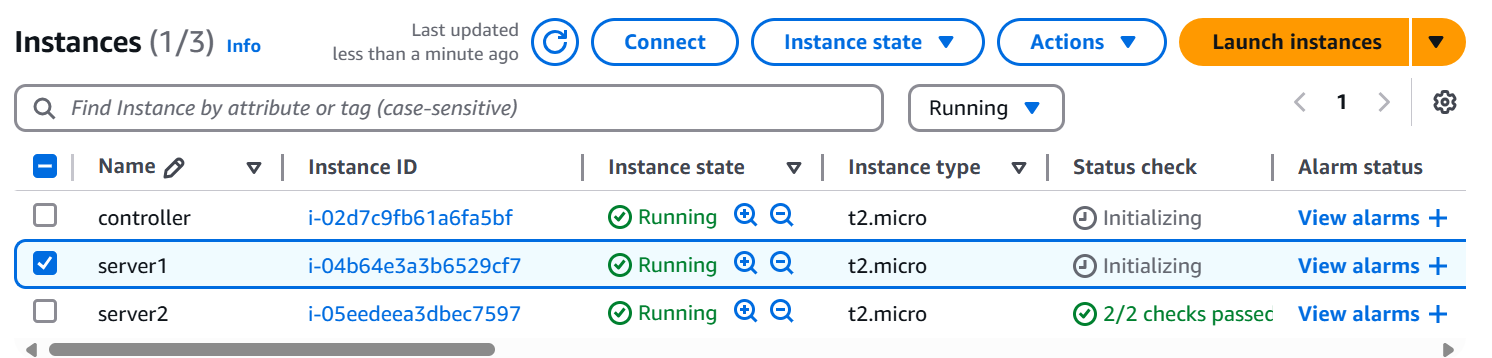
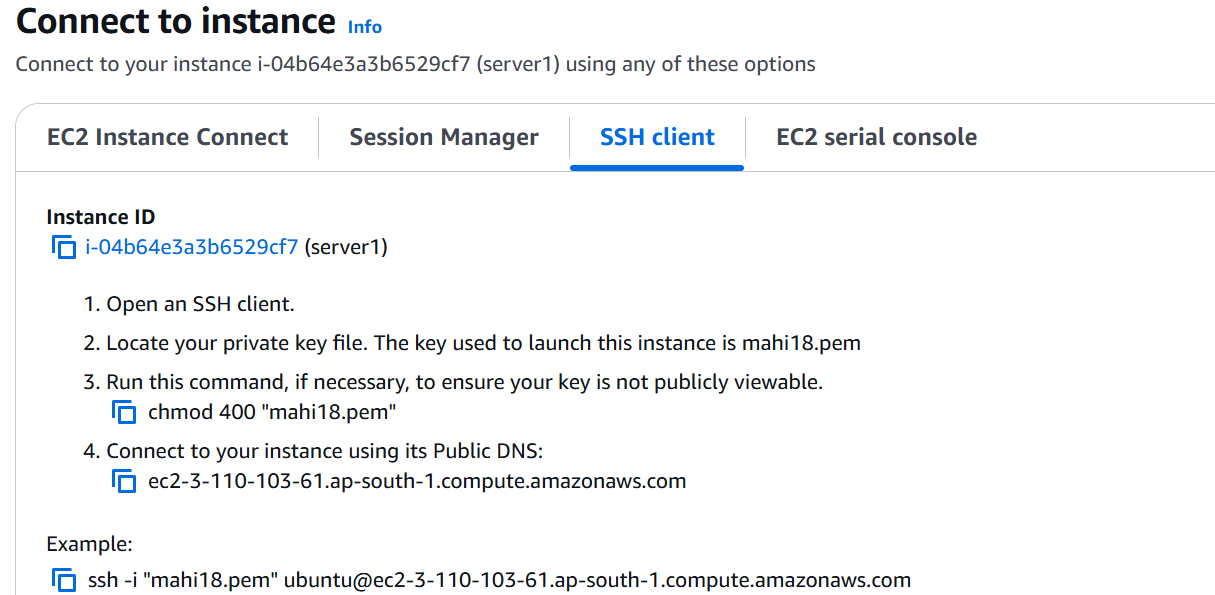
Create 3 ubuntu instances

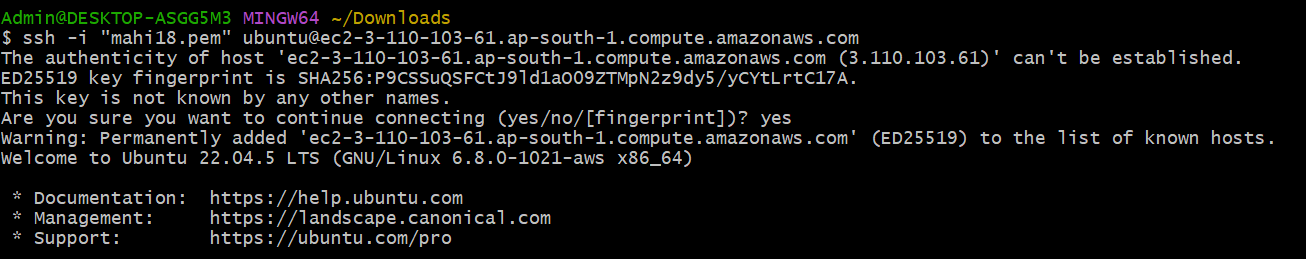


Connect to server1

Ubuntu machine default comes with python 3

But we need pytho 2



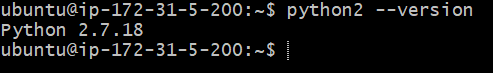


Update apt repository

Sudo apt-get update

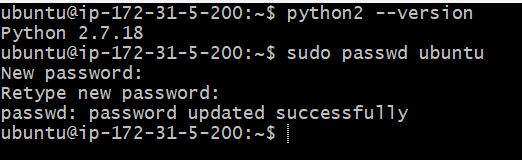
Install python2

Sudo apt-get install -y python2.7 python-pip



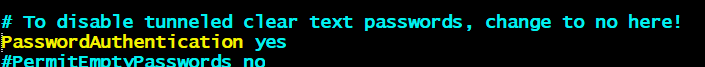
Establish password less connection between controller and managing nodes(server1 and server2)

Sudo passwd ubuntu (give password as ubuntu)



Sudo vim /etc/ssh/sshd\_config



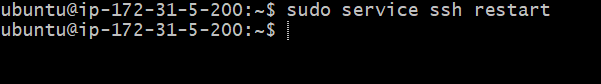


Change password authentication to yes

Save and quit

Restart the service

Sudo service ssh restart



Exit

Same procerdure to follow server 2

Now connected to controller

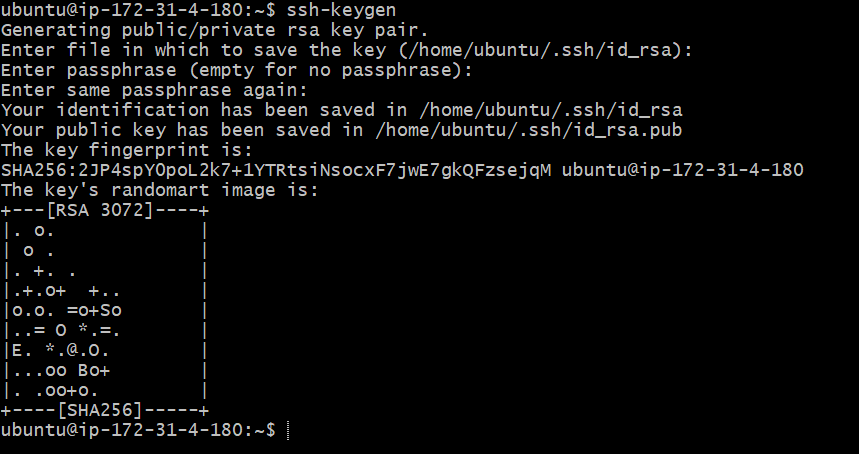
Update the repository

Sudo apt-get update

Install python 2

Generate the keys

Ssh-keygen



Controller machine open

vim .ssh/id\_rsa.pub

ssh-rsa  ubuntu@ip-172-31-4-180

Exit

Connect to server1

Vim .ssh/authorized\_keys --> paste the above key

Exit

Connect to controller machine

Ssh -v ubuntu@server1 ip address

Same procedure for server 2 (copy paste the keys)

Connect to controller

Install the ansible

Sudo apt-get install software-properties-common

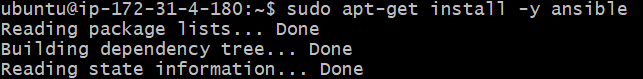


Sudo apt-add-repository ppa:ansible/ansible



Sudo apt-get update

Sudo apt-get install -y ansible



Write the ip adresses of managing nodes in the inventory file

Cd /etc/ansible

$ls

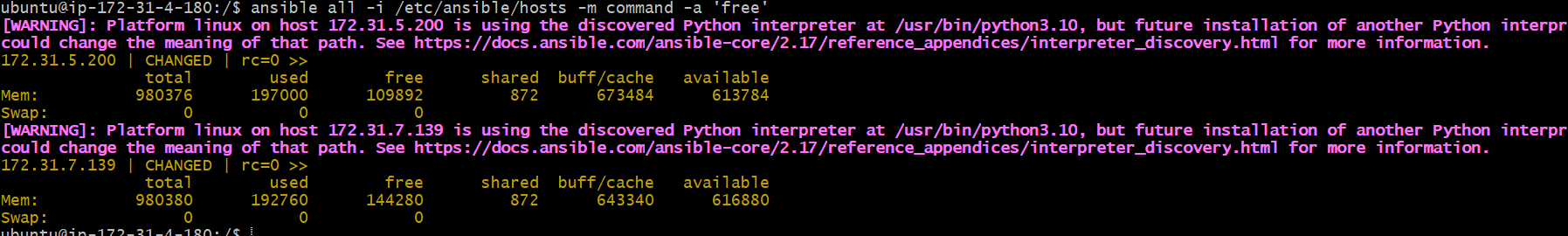
Sudo vim hosts (copy paste server1 and 2 ip addresses)

Adhoc commands: it works based on modules

Command module:

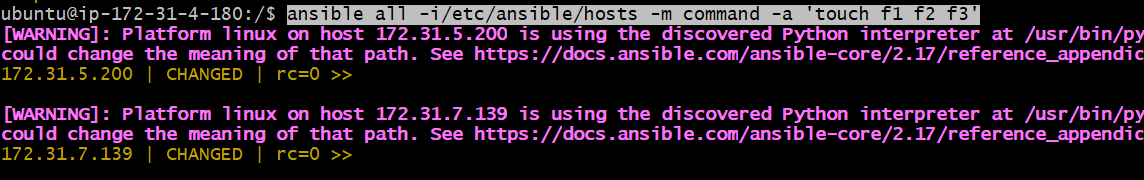
Find out the freespace in the managing nodes

ansible all -i /etc/ansible/hosts -m command -a 'free'



Create files in server1 and 2 from controller

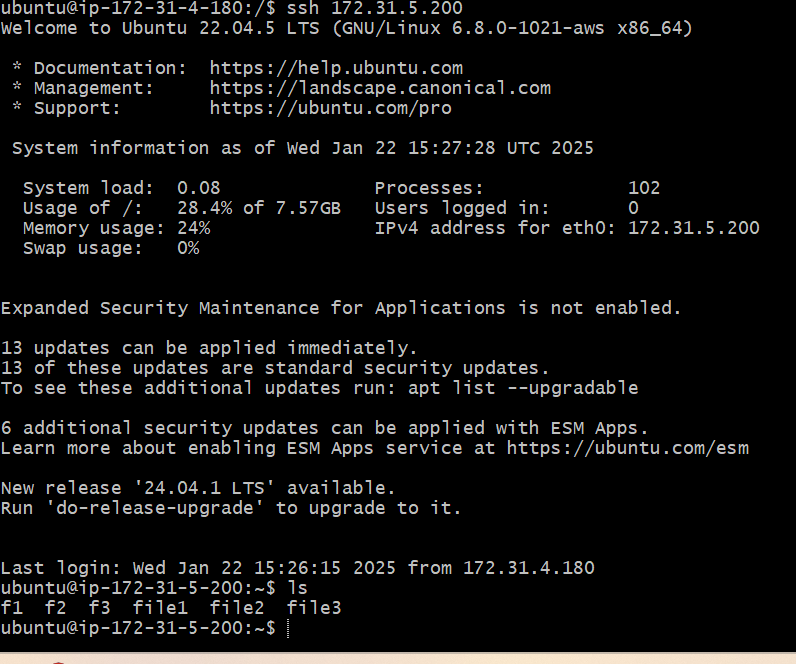
ansible all -i/etc/ansible/hosts -m command -a 'touch f1 f2 f3'



To check files are created or not

Ssh 172.31.5.200

$ls

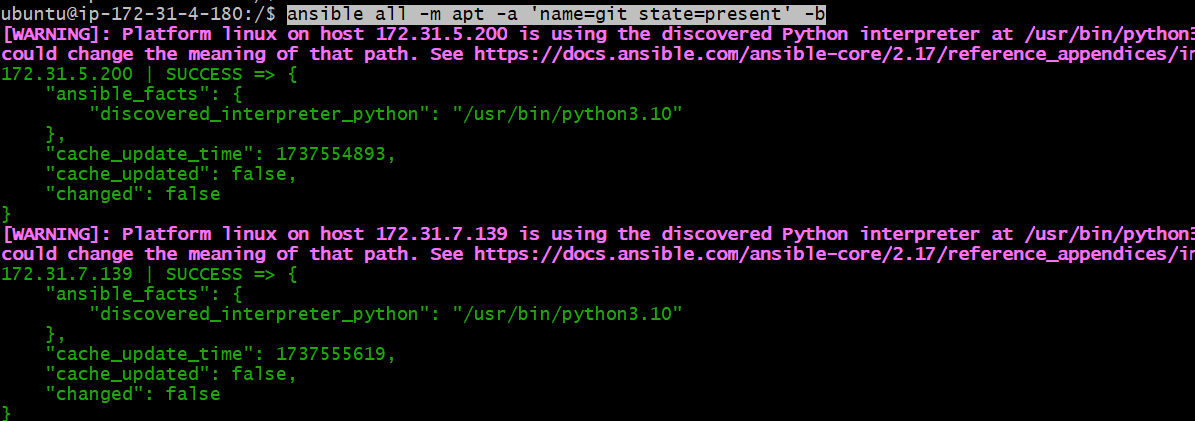


Apt: to install/uninstall any software

Go to controller

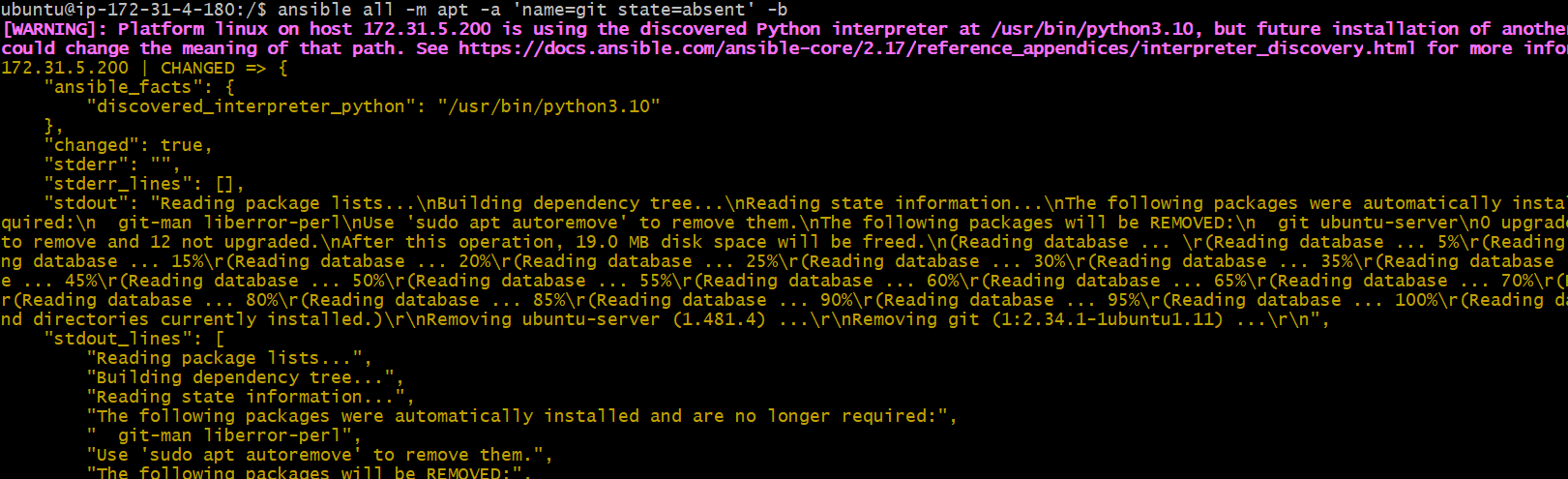
Install git in all server

ansible all -m apt -a 'name=git state=present' -b



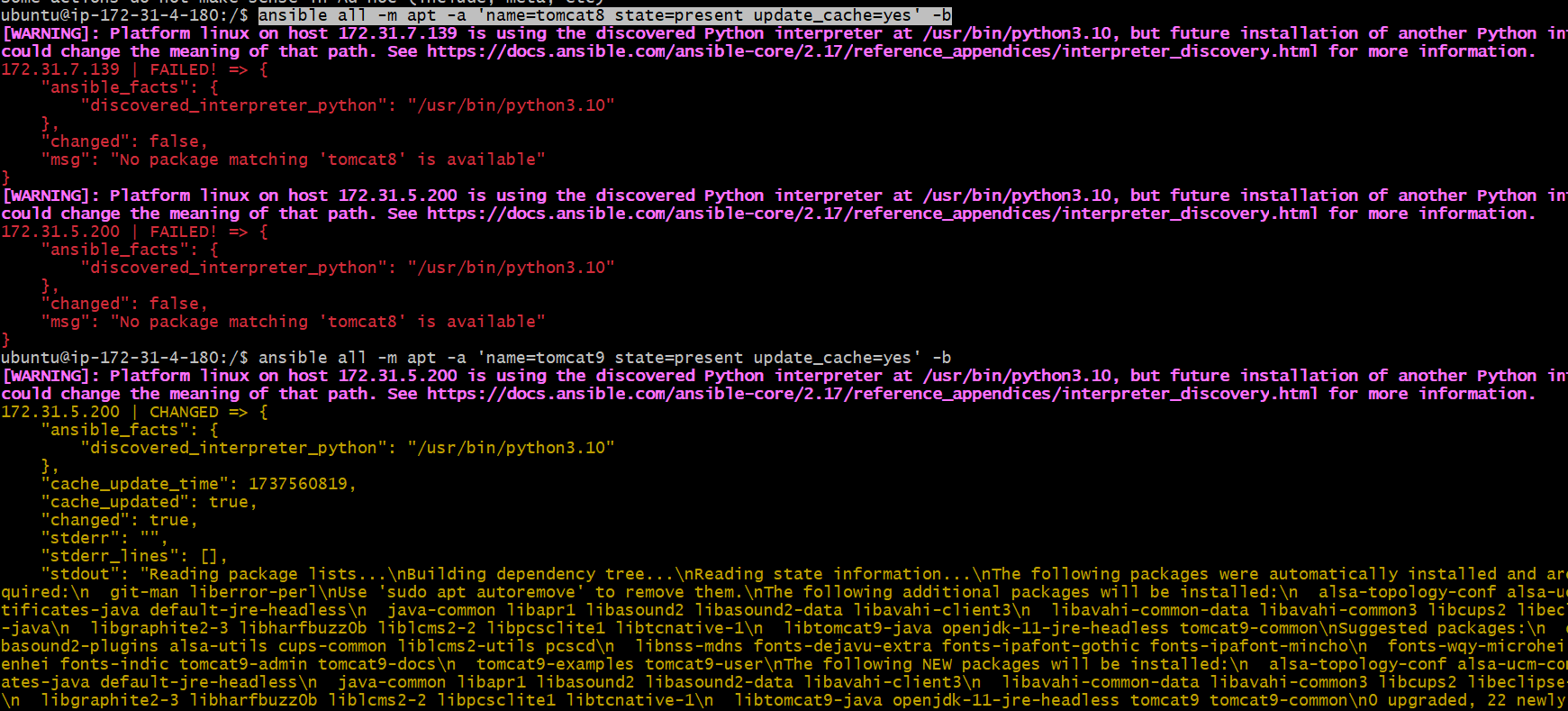
Git uninstall in all servers

ansible all -m apt -a 'name=git state=absent' -b



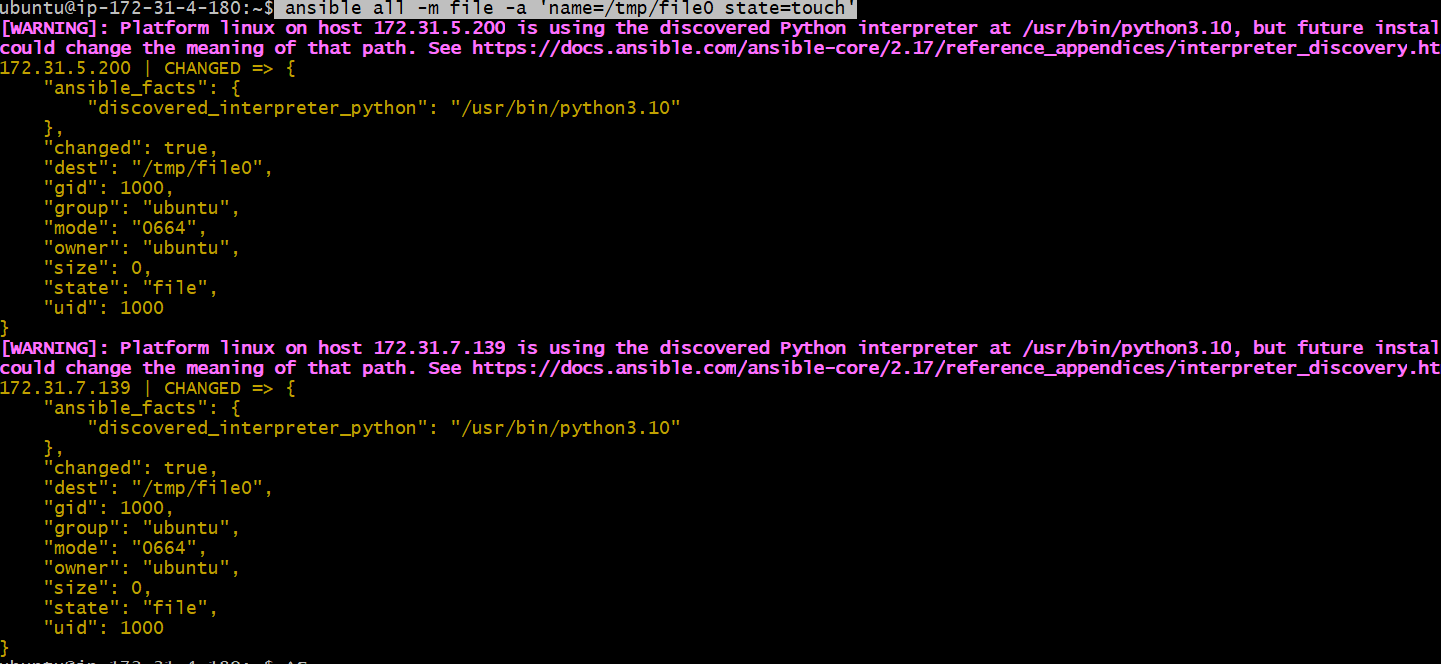
Install tomcat

ansible all -m apt -a 'name=tomcat8 state=present update\_cache=yes' -b (use tomcat9 )

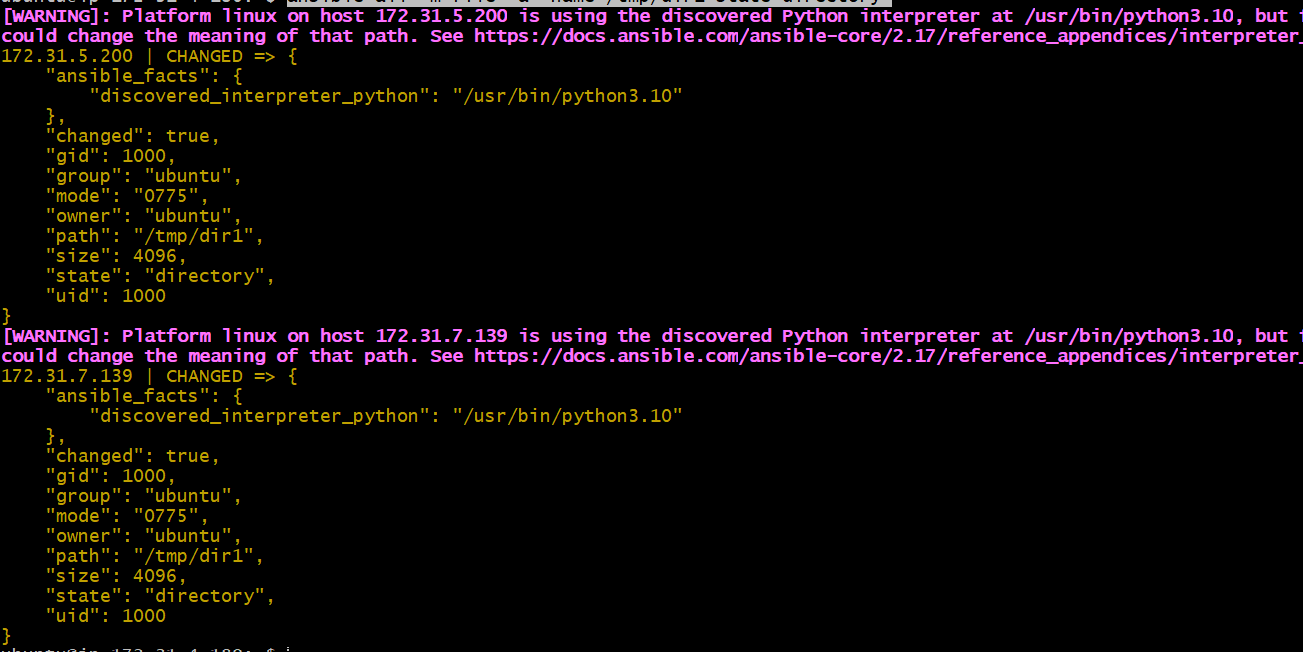


FILE: it is used to create a files and folders in managing nodes

ansible all -m file -a 'name=/tmp/file0 state=touch'



ansible all -m file -a 'name=/tmp/dir1 state=directory'

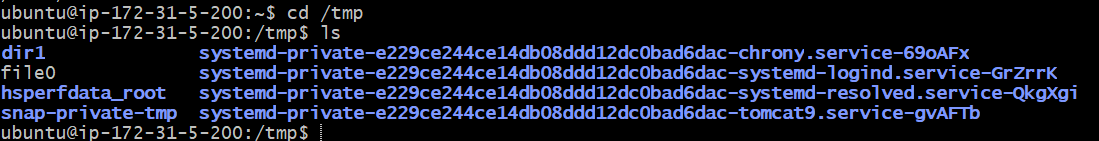


To check

$ssh privateip of server1

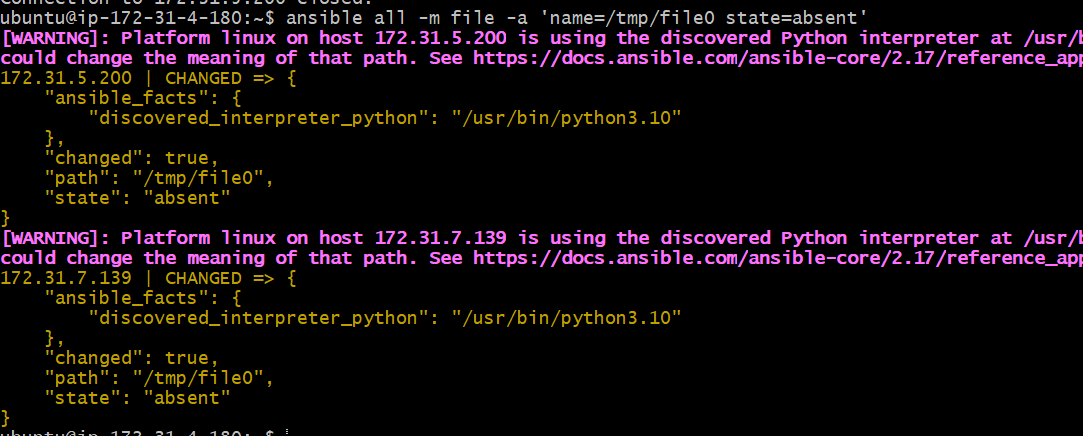
Cd /tmp

Ls



Remove the file or directory

Ansible all -m file -a ‘name=/tmp/file0 state=absent’

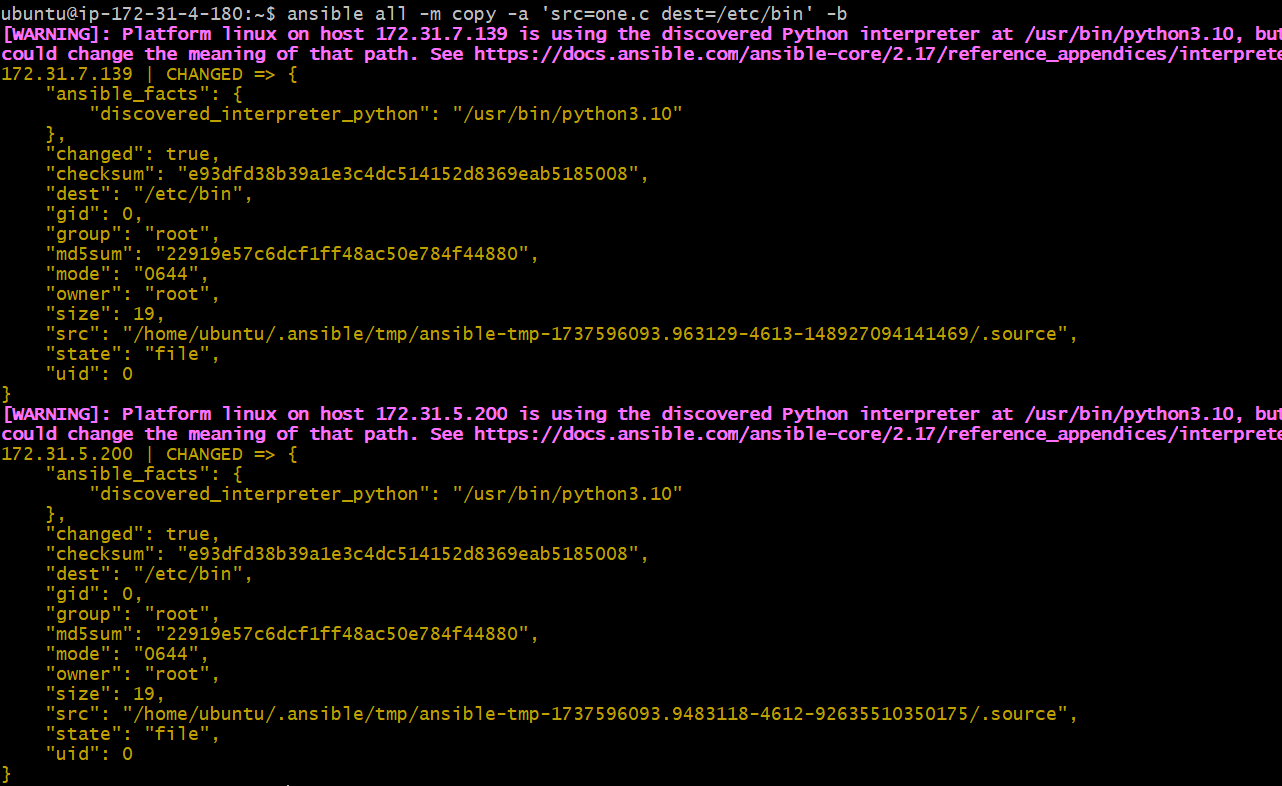


Copy: used to copy the file from controller to managing nodes

Change the permissions of file

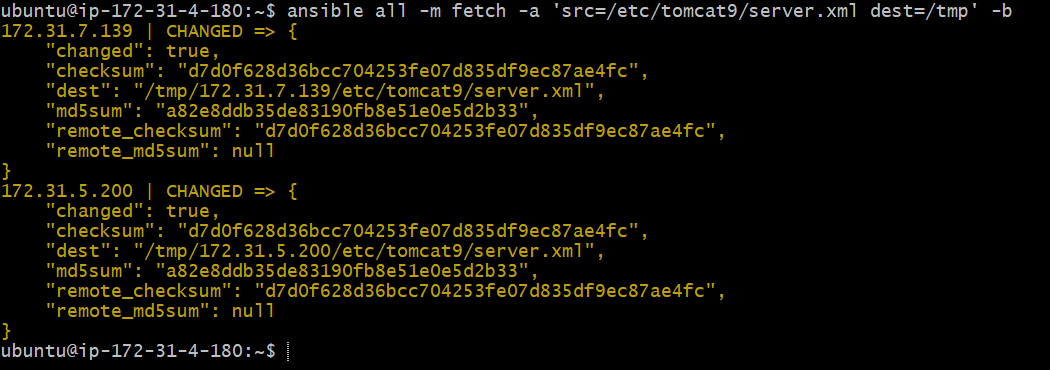
Ansible all -m copy -a ‘src=one.c dest=/etc/’ -b

Ansible all -m copy -a ‘src=one.c dest=/etc mode=777’ b

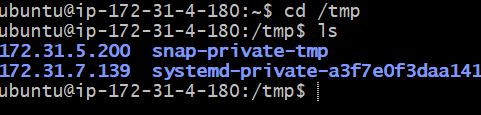


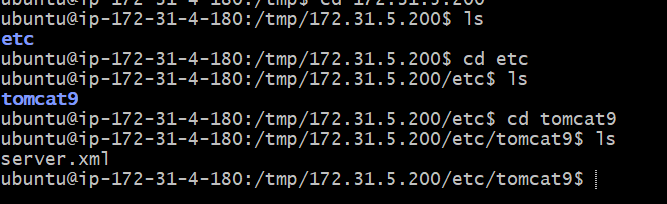
Fetch: used to copy the file from manager to controller

ansible all -i /path/to/hosts -m fetch -a 'src=/etc/tomcat9/server.xml dest=/tmp' -b



To check server.xml file but here we cant see but we see the two ip addresses





Git:

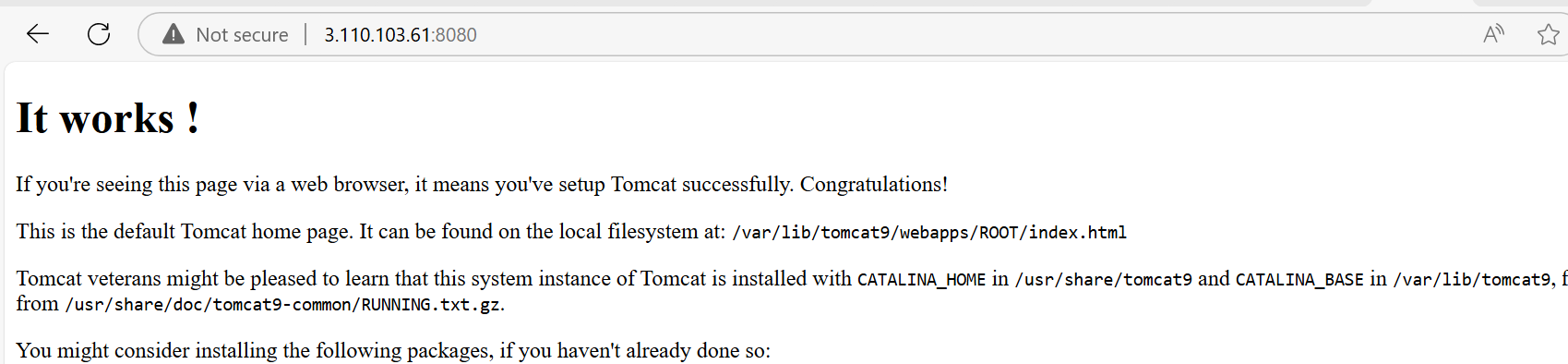
Ansible all -m git -a ‘rep=https://github.com/.git dest=/tmp/mygit’ -b

Service:

Used to start and stop and restart the services

Ansible all -m service -a ‘name=tomcat9 state=stopped’ -b

Started and restarted



Replace: is used to change the port no of servers

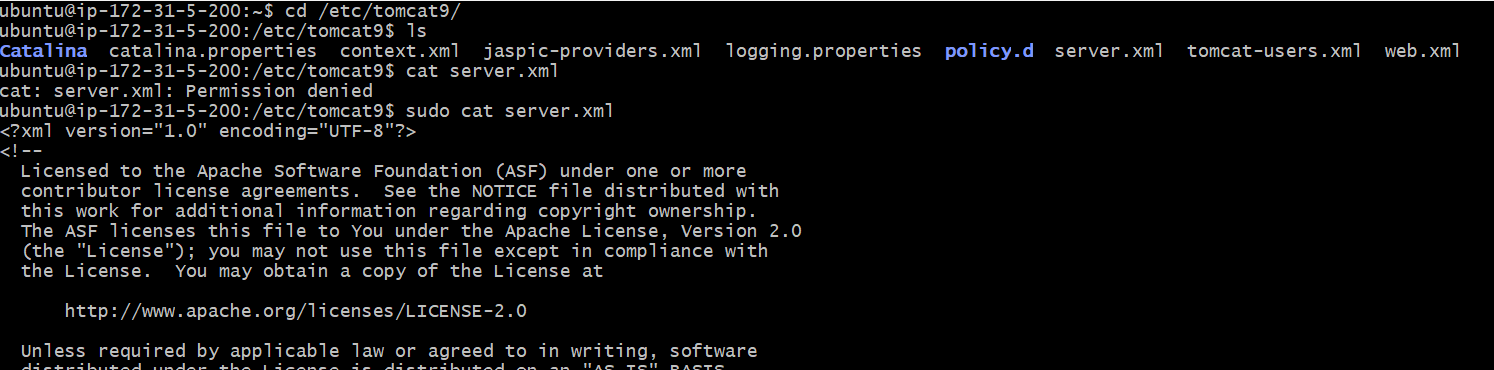
From controller connected to managing node

$ssh ip address of server1

$cd /etc/tomcat9/

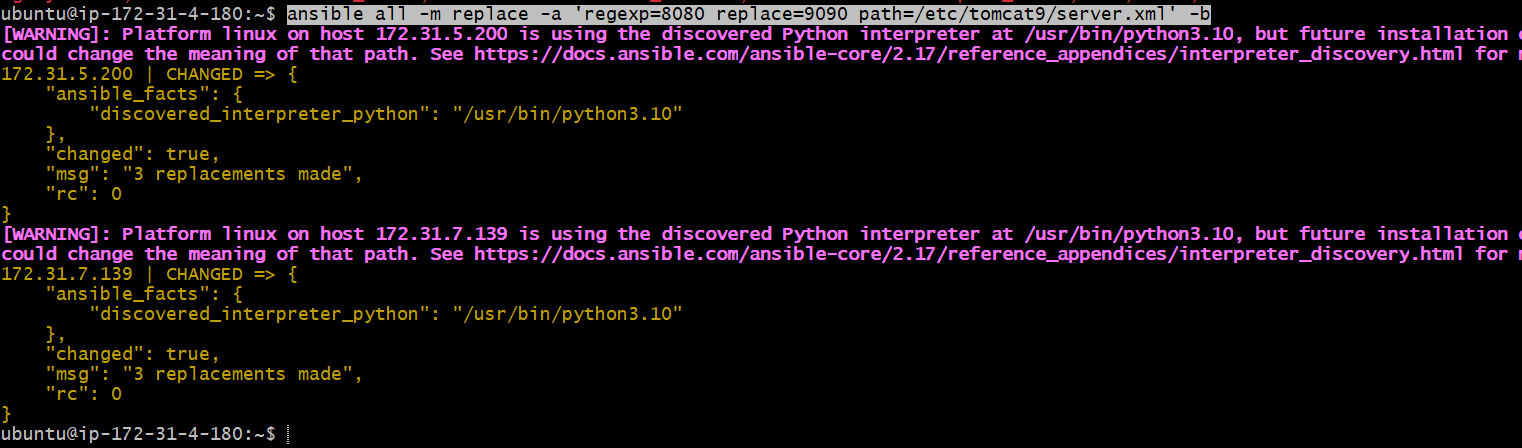
$ls

$sudo cat server.xml



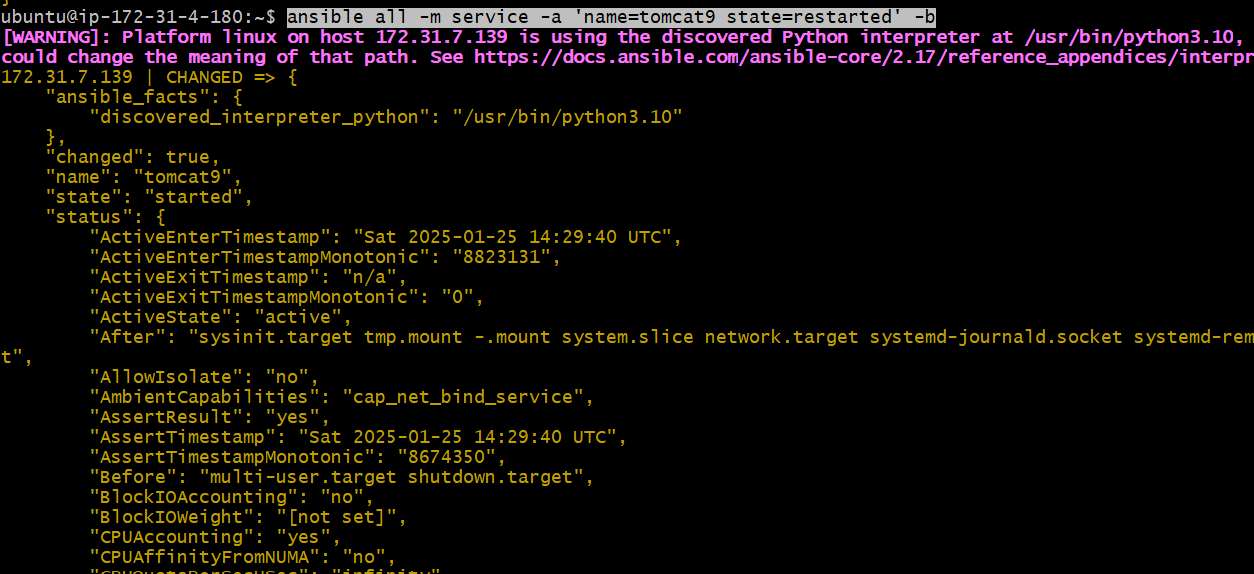
U can see the portno on above server.xml file in line no 74

ansible all -m replace -a 'regexp=8080 replace=9090 path=/etc/tomcat9/server.xml' -b

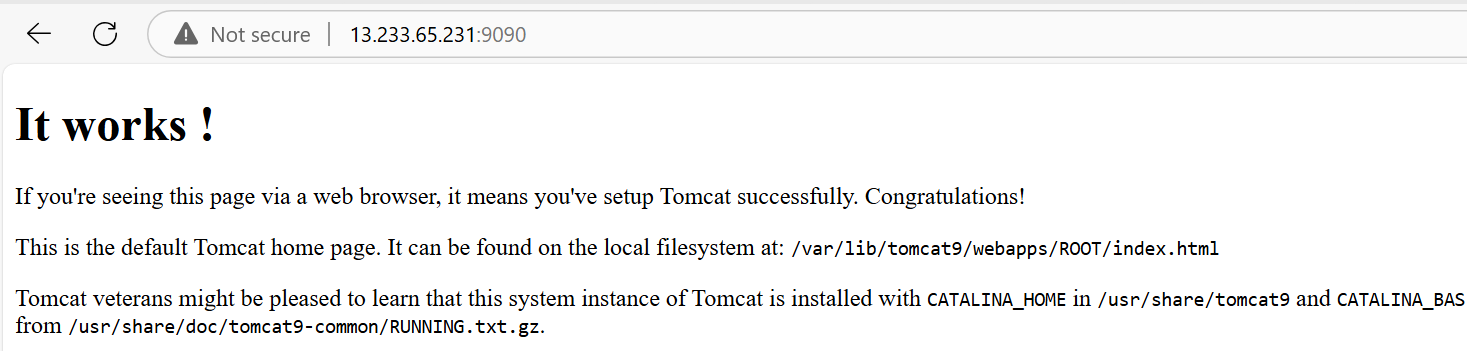


We need to restart the service

ansible all -m service -a 'name=tomcat9 state=restarted' -b

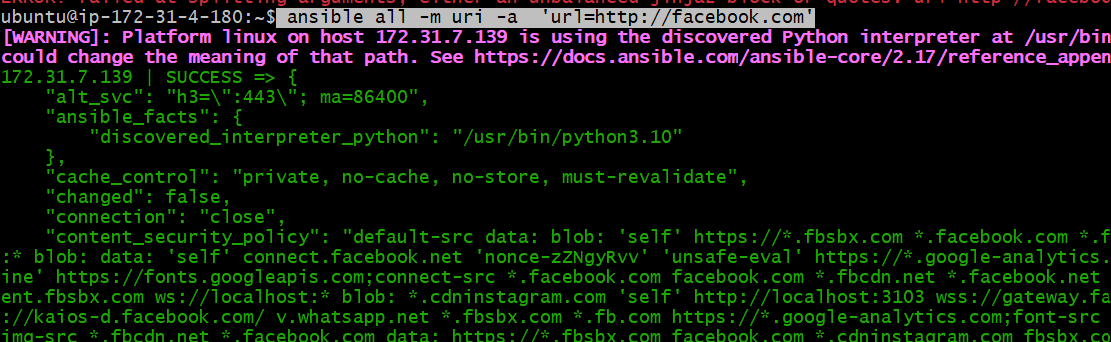


To check take the server1 publicip:9090



Uri: I want to check facebook is reachable or not in all managing nodes

$ ansible all -m uri -a 'url=http://facebook.com'



Status code =200 means output green color

Invalid url get status cod -1

Palybook:

Adhoc ommands are capable of working only on one module.

Playbook is combination of plays

Each play is designed to do some activity on the managing nodes

Main advantage is reusability .

Playbooks are created using yaml file

Example:

Playbook to configure tomcat9.

First uninstall tomact9

ansible all -m apt -a 'name=tomcat9 state=absent purge=yes' -b

$vim playbook1.yml

**---**

**- name:** Configure tomcat9

**hosts:** all

**tasks:**

**- name:** install tomcat9

**apt:**

**name:** tomcat9

**state:** present

**- name:** change the port no

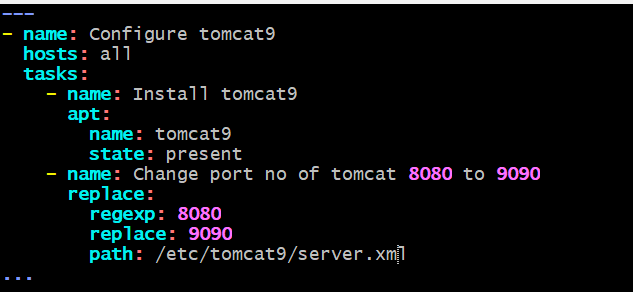
**replace:**

**regexp:** **9090**

**replace:** **7070**

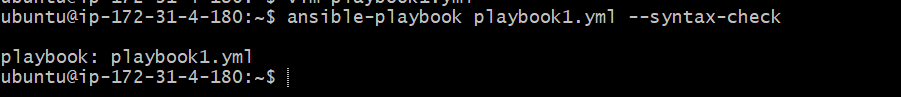
**path:** /etc/tomcat9/server.xml

**...**



Check syntax command

ansible-playbook playbook1.yml --syntax-check



Run command

