1. **What are Ansible Modules.**

Modules are considered to be the units of work in Ansible

Each module is mostly standalone and can be written in a standard scripting language (example includes: Python, Perl, Ruby, Bash, etc.).

One of the guiding properties of modules is idem potency, which means that even if an operation is repeated multiple times (i.e. ; upon recovery from an outage), it will always place the system into the same state

1. For this Modules concept we are going to create new instance and type is Ubuntu machine in aws and we are making same like previous nodes set up.
2. After successfully created AWS instance with ubuntu node. Now we have to login instance now using downloaded key pair.

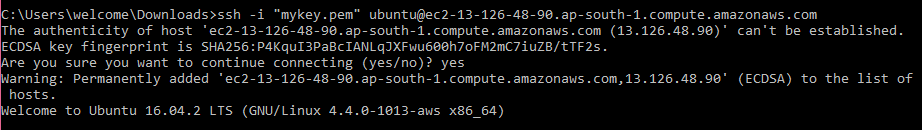
Goto command prompt in windows

Goto terminal if you are working on mac or linux

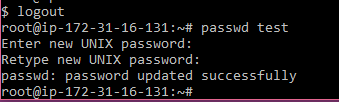
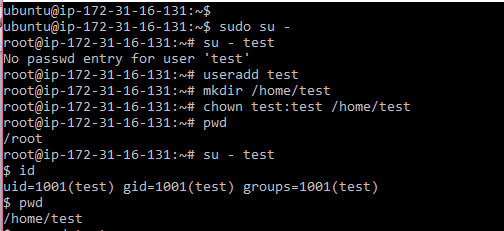
Type below commands

cd Downloads

Login with abs instance key pair like below.



Then we have create our own id like test



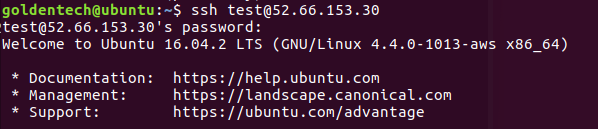
Then we have to update sshd\_config file with

Passwordauthentication = yes (default no will be there so yes replace with no value)

We have to restart the sshd\_config file.

Systemctl restart sshd

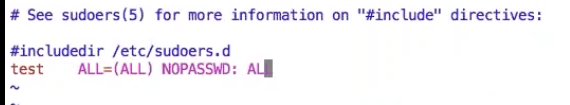
Now we are able to login from anywhere using test user.

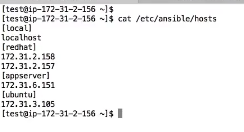


Now we have to make this test user can access to root user access. For this we have to add test user details into sudoers file.

cd /etc/

vi sudoers





We are going to discuss about set up module



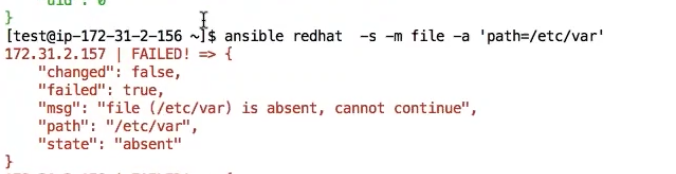
If you want to get IP addresses from our set use this command

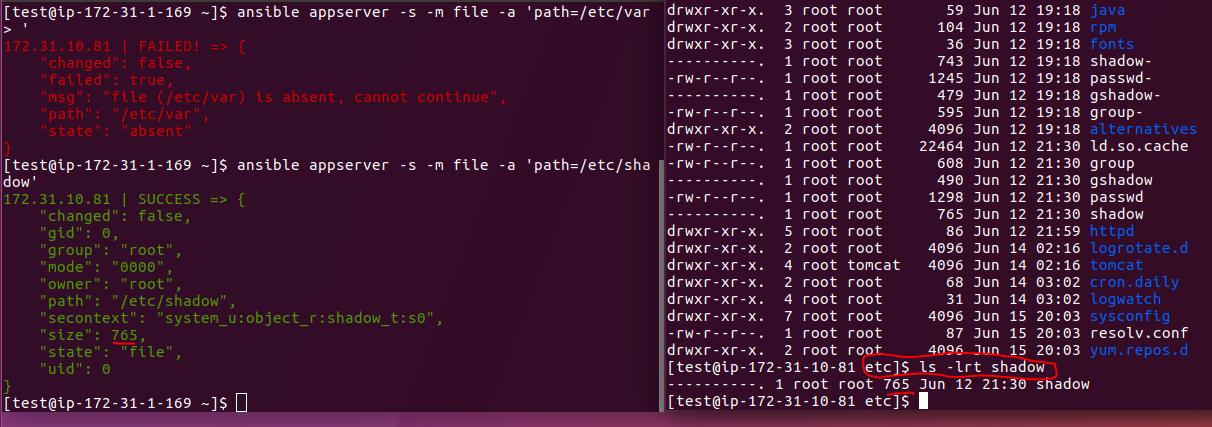
 after filter command

We are going to discuss file module

If you want to create or if you want to get any created file details we can use following commands

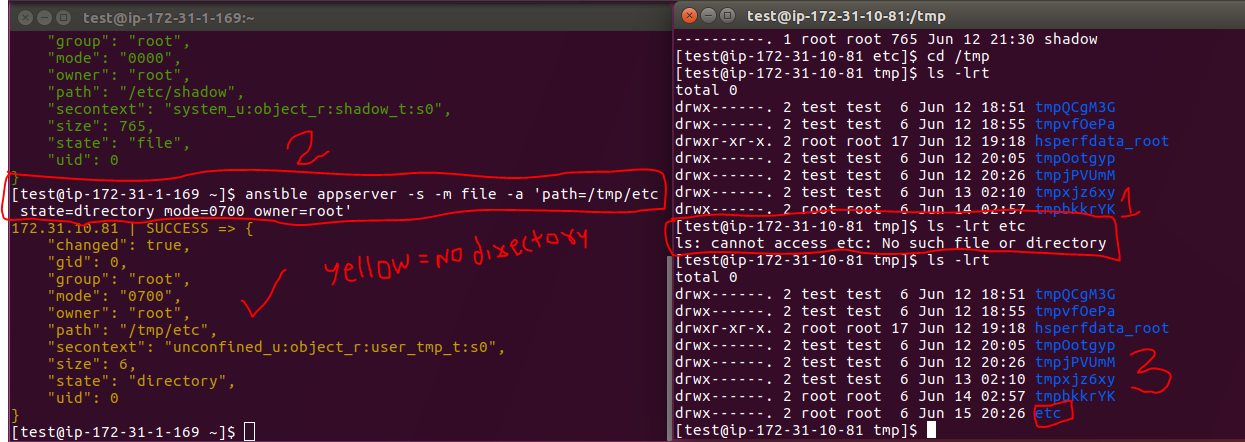




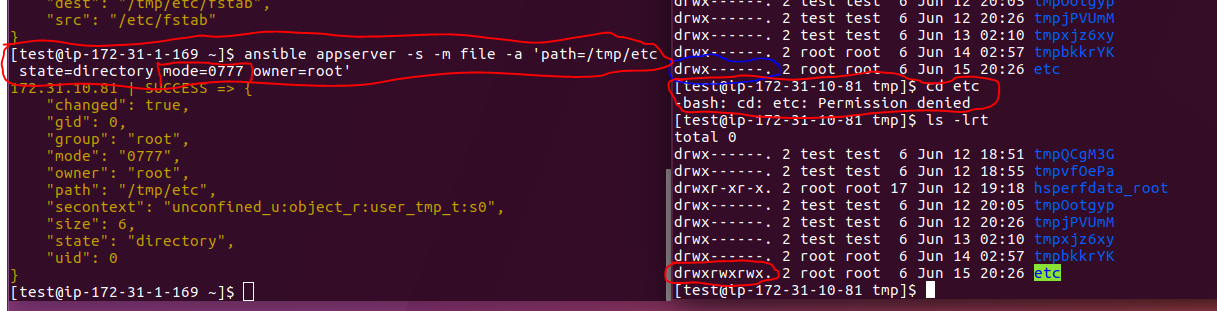


If you want to create any new directory or if you want to change any file permissions which are already existed, we have to use below command

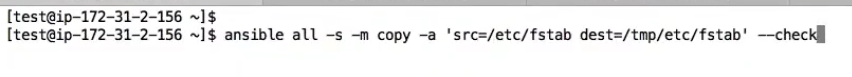


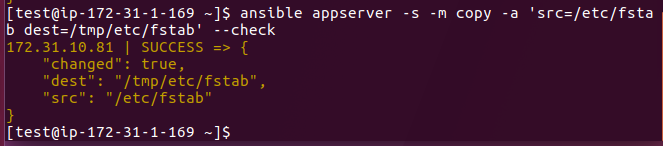


Created directory having 700 permissions so iam not able to open this folder since root is the owner of this file so again I have changed permissions to 777 so that I can open that folder and I can modify nthing there.

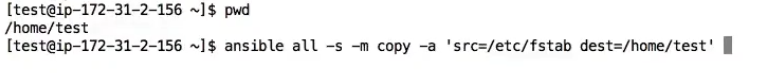


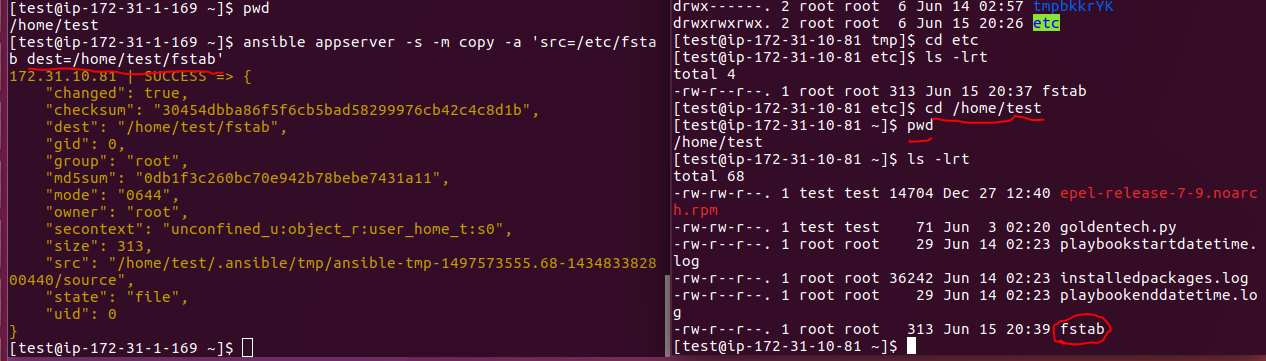
Samething we can use copy module instead of file module iinside of created directory called etc.





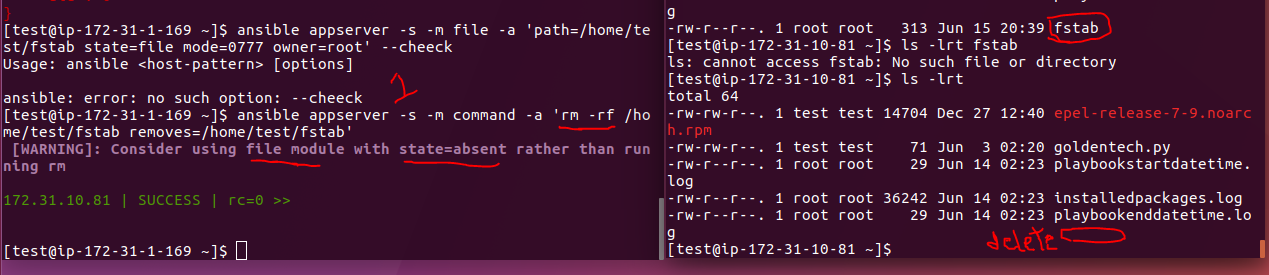






Now I want to remove the file from /home/test directory which is just created file.



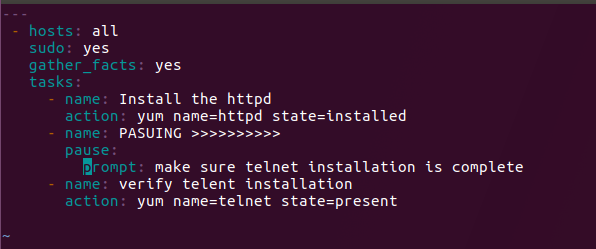


Next we are going discuss about PAUSE module :

Till now we have executed all module examples in a single command from now we are going to use in a playboks.

cd Playbooks

vim pause.yml



ansible-playbook pause.yml –check

So if any Ubuntu nodes in all environment we will get below error



That time we have to use apt-get utility package for Ubuntu node.

***ANSIBLE INSTALLATION ON UBUNTU***

The best way to get Ansible for Ubuntu is to add the project's PPA (personal package archive) to your system.

To do this effectively, we need to install the software-properties-common package, which will give us the ability to work with PPAs easily. (This package was called python-software-properties on older versions of Ubuntu.)

* sudo apt-get update
* sudo apt-get install software-properties-common

Once the package is installed, we can add the Ansible PPA by typing the following command:

* sudo apt-add-repository ppa:ansible/ansible

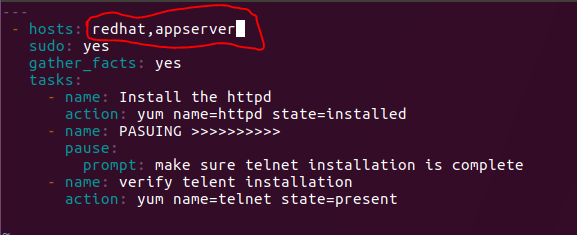
Press ENTER to accept the PPA addition.

* sudo apt-get update
* sudo apt-get install ansible

check version

ansible –version

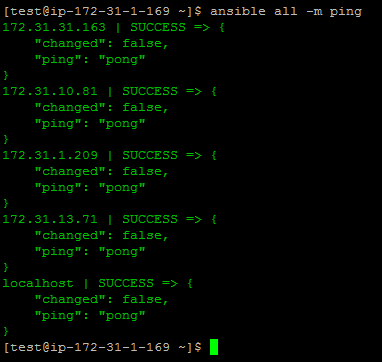
Next we can use multiple group names in Hosts header section like below,



Now we are going to discuss WAIT module:

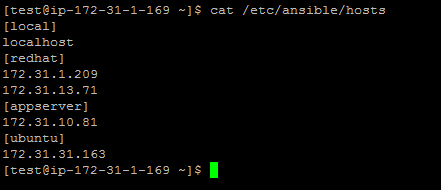
We can check all nodes are connecting with ansible or not ?

ansible all –m ping



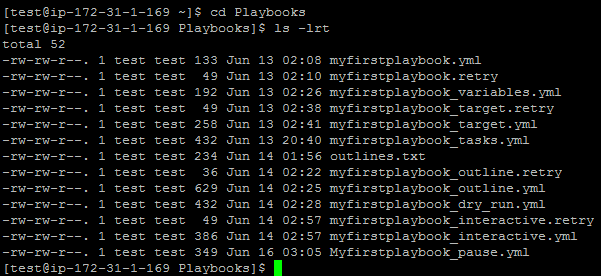
We can what are all groups we have created once

cat cd /etc/ansible/hosts

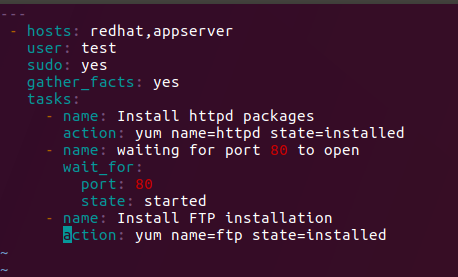


cd Playbooks

ls –lrt



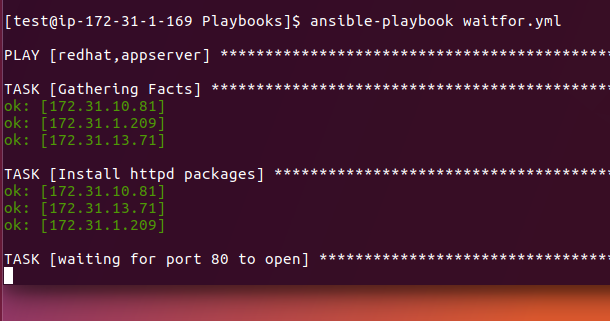
vim waitfor.yml



ansible-playbook waitfor.yml - - check

ansible-playbook waitfor.yml

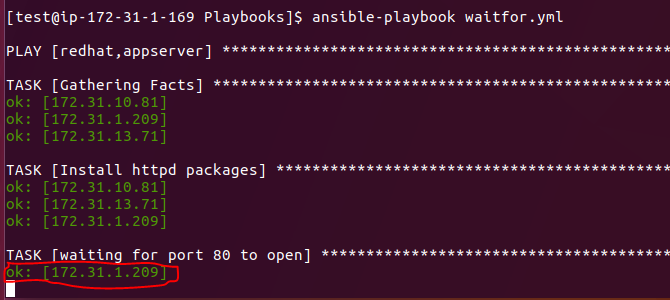
so after installing httpd it is going to wait fot port 80 should be open like below screen



That time we have to go hosts nodes mentioned in playbook and we have open the port 80 like below. Here in our example am going to redhat group nodes and opening the port like below



Go to master node now

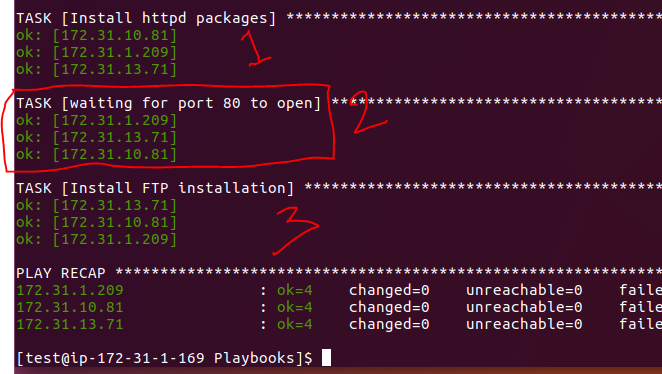


Node1 started port 80

Now it is waiting for other ports as well so we have open in all provded nodes like above.

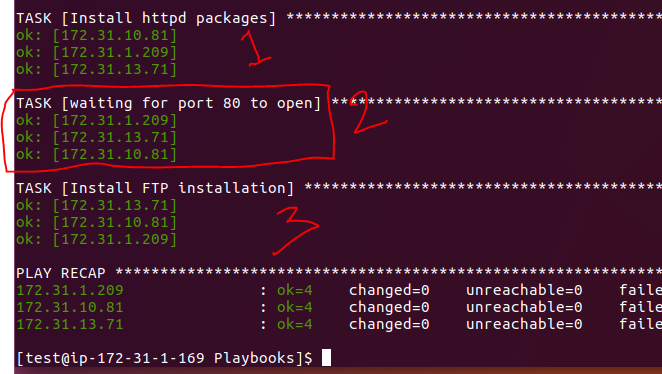
After port 80 in all nodes with this command

systemctl start httpd



If you run again this playbook no issues it wil not hold for starting port 80 since it is already started.

ansible-playbook waitfor.yml



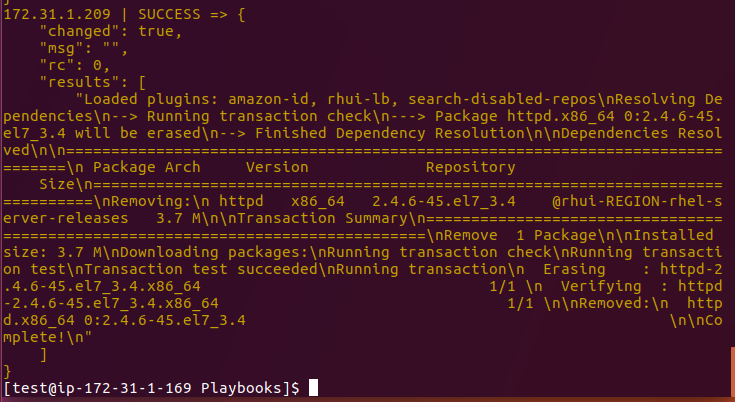
If I stop port 80 on all nodes again it will stop and wait for port 80 when you run this playbook. We can stop and test it once if you want. Using below command we can stop

sudo systemctl stop httpd

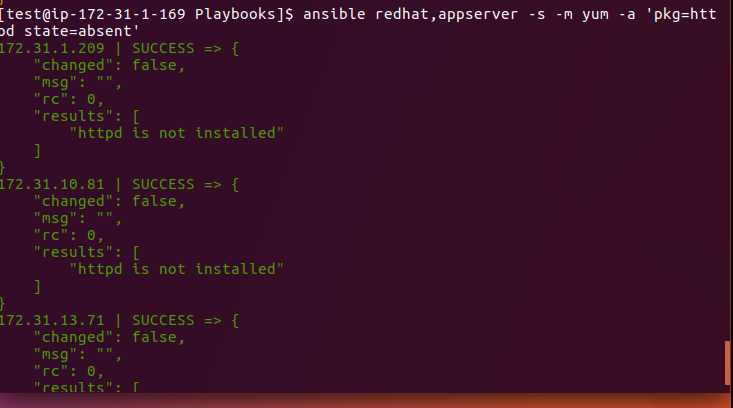
If you want to uninstall httpd again we have to pass in playbook as **state=absent**

Or elase we can write in adhoc command also like below

ansible redhat,appserver –s –m yum –a ‘pkg=httpd state=absent’



So it is removed successfully and all in yellow colour again do the same step



You will get like above screen successfully removed. So we can do via playbook and command line

We are going to discuss with YUM module now.

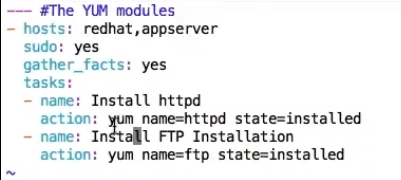
Till now we discussed yum module in command line mode now am going to use in playbook.

ansible all –m ping

copy playbook from previous one

cp –pr waitfor.yml yum.yml

vim yum.yml



You just remove wait for module and save it

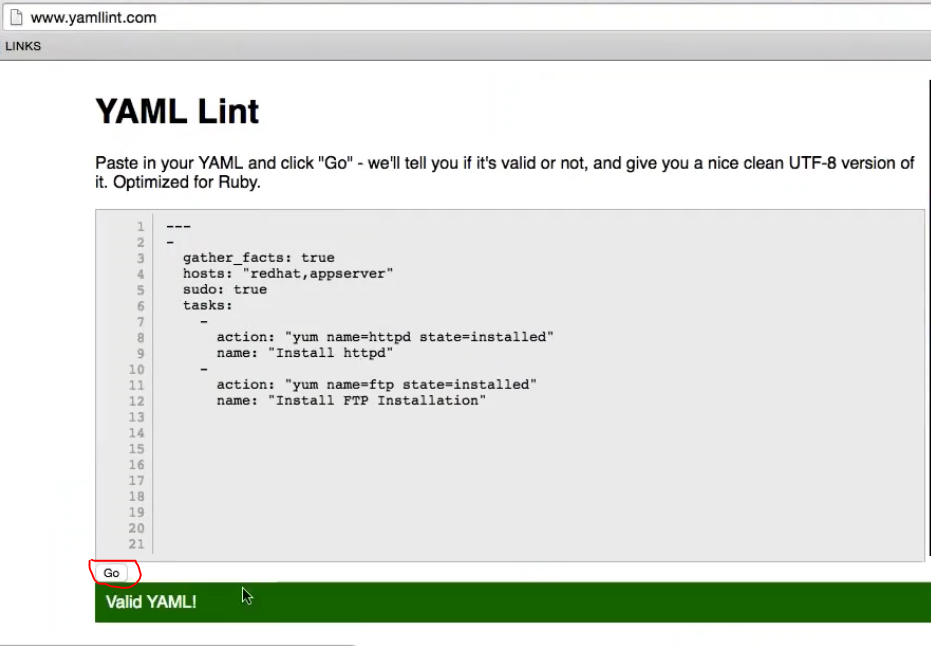
Just try to wth dry run and actual run same yum.yml playbook

Then run ansible-playbook yum.yml

So here I want to tell one more thing we can verify our file using third party tools. Just go to your favorite browser and type verify my YML file. So that you can see lot mote links.

Like [www.yamllint.com](http://www.yamllint.com)

Copy your playbook code and paste in provided link and click on go



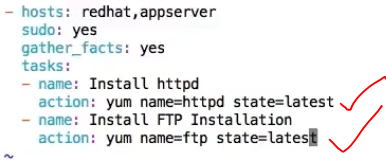
if you want to uninstall so playbook needs to modified with state=absent like below

vim yum.yml



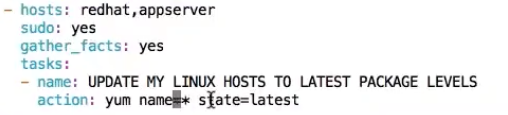
ansible-playbook yum.yml

If you want to install latest version you can use below playbook

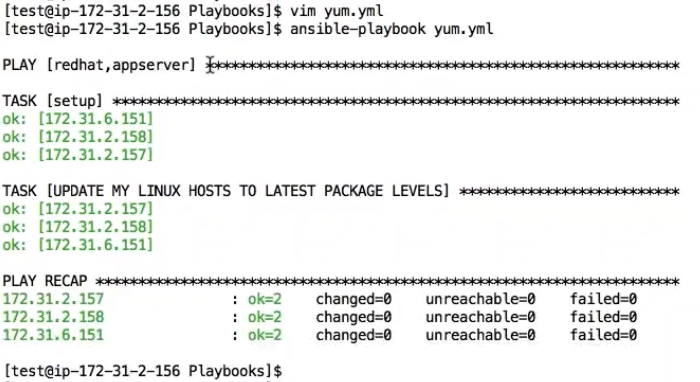


**ansible-playbook yum.yml**

In yum module f I want to update my linux packages I can use below playbook.



**Ansible-playbook yum.yml**



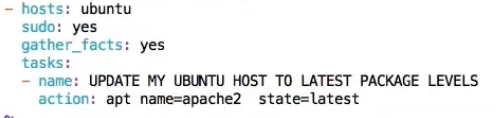
**Now am going to discuss about apt module.**

So apt module we all already know it is a utility package for Ubuntu system so same like yum for redhat.

Now we are going to use Ubuntu system for this module.

**Cp –pr yum.yml apt.yml**

**Vim apt.yml**



Same like we can write anything like ftp and we can use state=absent also here.

**ansible-playbook apt.yml**

**We are going to discuss service module now.**

**We are copying yum from yum.yml**

**Cp –pr yum.yml service.yml**

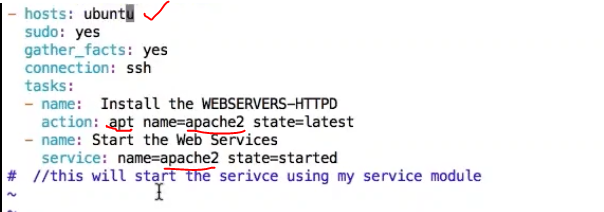
**And edit like below**



**If you want to change redhat to Ubuntu you just need to replace with yum=ant in above above playbook.**

**ansible-playbook service.yml**

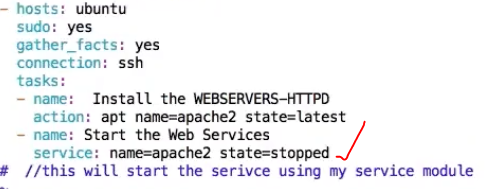
**For Ubuntu node for apache2 installation we can do this below playbook**



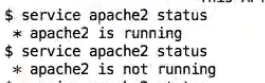
**We can check the output now in Ubuntu node wether atarted or not using below command.**



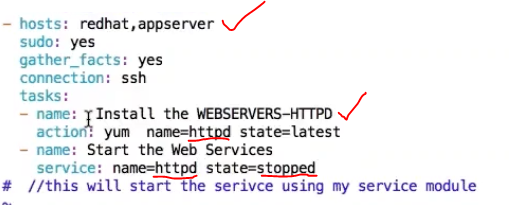
**Samethng if you want to stop the service we have to replace state=stopped like below**



**Now it is stopped.**

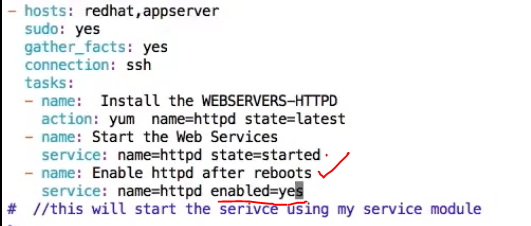


**For same thing we can do for redhat systems also like below for httpd installation,**



**So now you can go to redhat group nodes and check httpd actve or not.**

**And one more thing we can set booting options as well with playbook. Suppose whenever you booted your system automatically httpd will starts the services. We can customize our playbook like below,**

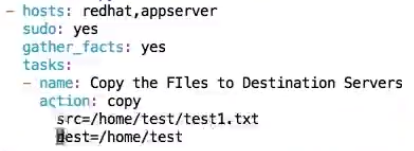


**If you run this this wll anable to httpd service should be active for all the boot ups.**

**Now we are going to discuss about copy module, copy module we already did in adhoc commands now we are going to use in playbooks.**

**cp –pr service.yml copy.yml**

**vim copy.yml**



**First we have to create test.txt file in master node then it is going to copy to mentioned nodes.**

**Cd /home/test**

**Touch text1.txt**

**Vi test1.txt**

**Enter some data into file**

**:wq**

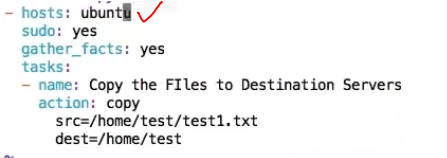
**So you are able to see fiile in master node.**

**Now you can do below command**

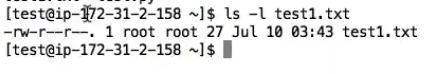
**ansible-playbook copy.yml**

**Now you can check all the destination nodes in /home/test folder for test1.txt file.**

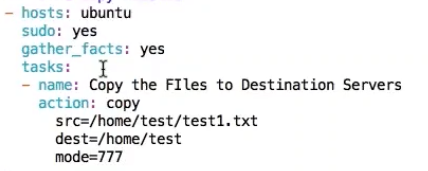
**Samething you can do for Ubuntu node as well**

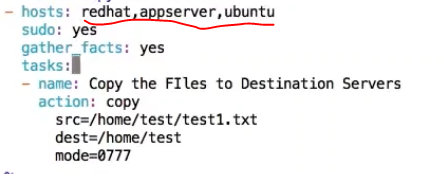


**Now we can file permissions for the created file on all nodes**



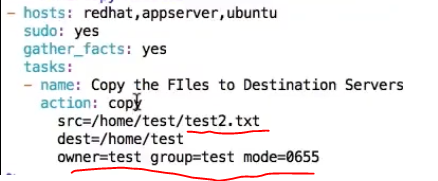
**File is having 644 mode. If you want to change mode go and edit copy.yml playbook like below**





**If you want more details to files like owner and group as well we can add in playbook like below**

**We are going to copy test2.txt file and adding few more details to it . First we have to check test2.txt file is there or not if not we have to create same like above.**



**If you want to copy one file test3.txt but if it is already there in the destination dir so that time we can make old file as backup file and we can copy new file as normal file using below playbook. First we should create test3.txt into our source location same like test1.txt.**



**And this playbook has to run 2 times and first time it is going copy test3.txt file and second it is going to take old file as backup and new file as normal file.**

**Go and verify our node systems**

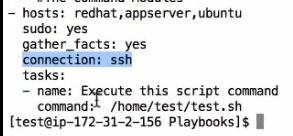


**We are going work on command module now :**

am going copy my copy.yml file to command.yml file

cp -pr copy.yml command.yml

vim command.yml



:wq!

Now you are going to mentioned path in palybook.

vi /home/test/test.sh

#!bin/bash

echo “This is my script” > /home/test/output.txt



chmod u+x /home/test/test.sh

chmod 755 /home/test/test.sh

sh /home/test/test.sh ---- so am going to see my output.txt file in my home directory

Now am going to my home where the sh file available.

cd /home/test

ls -lrt

more output.txt

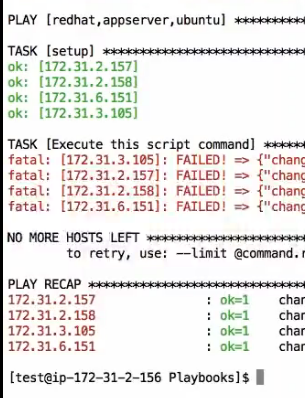
so am able to run my test.sh file and in script output.text got generated in specified path.

So this was just executed using command line in my master node now am going to run this palybook and this .sh file going to exeecute on all set up nodes.

cd Playbooks

ansible-playbook command.yml –check

ansible-playbook command.yml



This was my out put why because we dont test.sh file on my destination nodes that is why am not going to execute test.sh file on my destination.

So we can create a file on each destination or we can use copy module to execute this palybook as expected.

Go to each node and create test.sh file under /home/test location like below.

vim test.sh

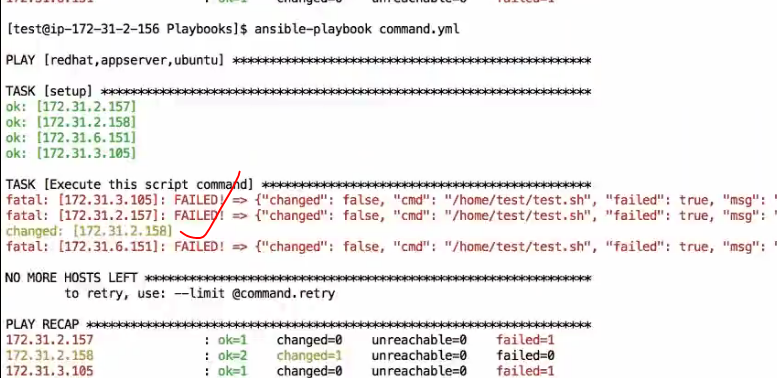
#!/bin/bash

echo “This is my generated file” > output.txt

:wq



and now go to master node and try to re run the command.yml file and I can see my yml file is going to work edited node and created output.txt.



Now am going to other nodes as well like same above

vim test.sh

#!/bin/bash

echo “This is my generated file” > output.txt

:wq

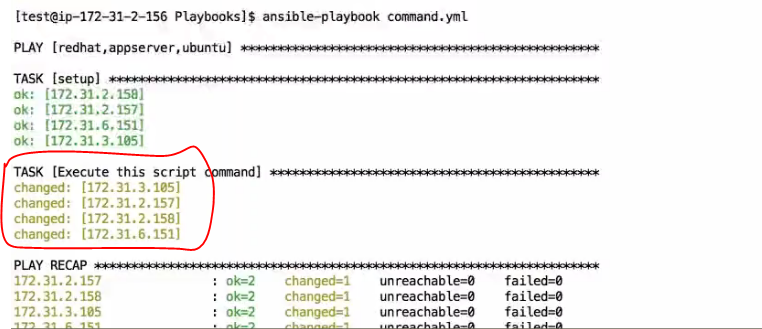


and now again go to master node and try to re run the command.yml file and I can see my yml file is going to work edited node and created output.txt.

ansible-playbook command.yml



Same thing we can go and create test.sh file on all nodes finally command.yml playbook can be executable on all nodes.



Now we are going to discuss about cron module :

CRON is schedule module where we can scheduled a job in a day or night give a time for particullar job is going to run. The Playbook is going create a cron for particular user.

CRON modules are very very important particullarly when you are going to jobs for production environment for particullar time and duration.

If you want check about cron jobs currently available or not, for this we have to use below command where ever you want to check it like maste node or any node.

crontab -l



Now we are going to our cron.yml playbook

cp -pr commandyml cron.yml

vim cron.yml

---

- hosts: all

user: test ----- here am not going write sudo because am writing cron job for test user only

gather\_facts: yes

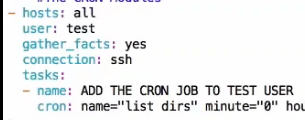
connection: ssh

tasks:

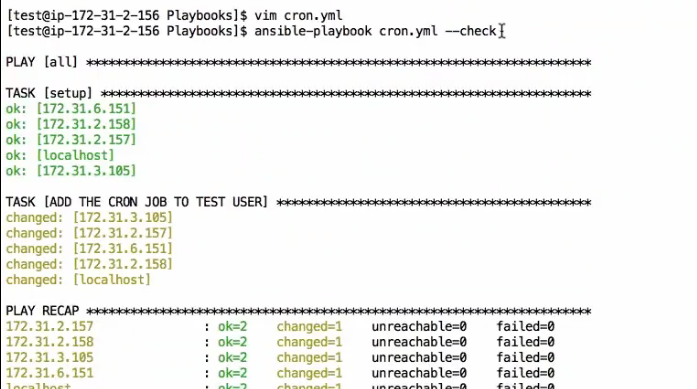
- name: Add the cron job to test user

cron: name=”list dirs” minute=”0” hour=”1” job=”ls -al /var/log > /home/test/cron.log”

:wq!



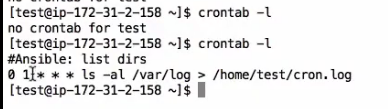
ansible-playbook cron.yml –check



ansible-palybook cron.yml

Now it is going to execute my playbook on all mentioned hosts. Now you can go each node cehck for cronjobs with below command.

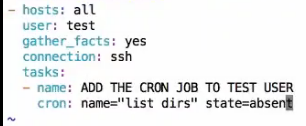
crontab -l



Now you are able to see cron job all on nodes.

If you want to remove the cron job we you can write playbook using below syntax,

state=absent and remove cron notation



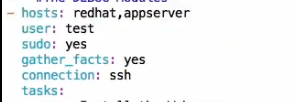
ansible-playbook cron.yml

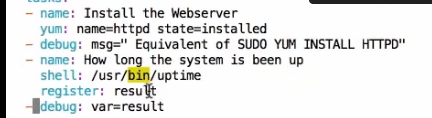
It is going to remove mentioned cron job in name section on all nodes.

We are going to discuss about DEBUG module now:

cp -pr cron.yml debug.yml

vim debug.yml





ansible-playbook debug.yml

Now we are going to discuss about Fetch Module:

It will fetch the files from remote hosts nodes to our maser node.

vim fetch.yml

type below lines of code

---

- hosts: redhat

user: test

sudo: yes

gather\_facts: yes

connection: ssh

tasks:

- name: copying the file from remote server to master server

fetch: src=/etc/passwd dest=/home/test

now we can go to on master node test user home location

cd

look for hosts ip addresses folders

cd <node ip>

cd /etc

cat passwd

so like this you can all nodes passwd file from all mentioned nodes in hosts section in playbook.

**Now we are going to discuss about user module:**

vim user.yml

type below line of code for this playbook

---

- hosts: redhat,appserver

user: test

sudo: yes

gather\_facts: yes

connection: ssh

tasks:

- name: add the user called apache

user: name=apache comment="TEst apache" group=wheel shell=/bin/bash

~

:wq

ansible-playbook user.yml –check

before going to run this play book go to each node and go to below location,

cat /etc/passwd|grep -i apache

tomcat:x:91:91:Apache Tomcat:/usr/share/tomcat:/sbin/nologin

apache:x:48:48:Apache:/usr/share/httpd:/**sbin/nologin**

after running this paly book on master

ansible-playbook user.yml

go to again hosts node and check same command

cat /etc/passwd|grep -i apache

tomcat:x:91:91:Apache Tomcat:/usr/share/tomcat:/sbin/nologin

apache:x:1002:10:TEst apache:/home/apache:/**bin/bash**

and type below command also

id -a apache

uid=1002(apache) gid=10(wheel) groups=10(wheel)

So successfully added user with bin/bash all on nodes, and we can verify all on like above 2 commands.

Now am going to remove created user with lines of playbook

vim user.yml

---

- hosts: redhat,appserver

user: test

sudo: yes

gather\_facts: yes

connection: ssh

tasks:

- name: add the user called apache

user: name=apache state=absent remove=yes

~

:wq

then we have to check on nodes again with following commands

[test@ip-172-31-10-81 ~]$ cat /etc/passwd|grep -i apache

tomcat:x:91:91:Apache Tomcat:/usr/share/tomcat:/sbin/nologin

[test@ip-172-31-10-81 ~]$ id -a apache

id: apache: no such user

[test@ip-172-31-10-81 ~]$

Now we are done with imp modules if you want to know any neede module you just go to ansible modules documentation site and just follow the syntax.