**Installation of Apache, Php Through Ansible Playbook**

* Firstly, create two instances in EC2(one as ‘Ansible Control’ and another as ‘Node-1’).
* Add same user in both the VM (Virtual Machines) with same password. e.g., “shoaib”.

**Command 🡪 (sudo adduser shoaib)**

* Add the user into sudoers file below “sudo” and write “username ALL=(ALL:ALL) NOPASSWD:ALL”

**Command 🡪 (sudo visudo)**

* Change the authentication from “no” to “yes” in sshd\_config at “PASSWORD AUTHENTICATION”.

**Command 🡪 (sudo vi /etc/ssh/sshd\_config)**

* Restart the sshd service.

**Command 🡪 (sudo service sshd restart)**

* Check the status of sshd service. (It will show as “Running”)

**Command 🡪 (sudo service sshd status)**

* Run the above steps in other VM (Node-1).
* Switch the user from “ubuntu” to “shoaib” in Ansible Control.

**Command 🡪 (su shoaib)**

* Generate the ssh key in Ansible Control being in user “shoaib”.

**Command 🡪 (ssh-keygen)**

* Then copy the ssh key into the other VM(Node-1).

**Command 🡪 (ssh-copy-id username@private\_ip)**

* Now you can connect to the other VM(Node-1) without password from ‘Ansible Control’.

**Command 🡪 (ssh private\_ip)**

* Update the cache in ‘Ansible Control’.

**Command 🡪 (sudo apt update)**

* Install Ansible to execute Ansible-Playbook in other VM(Node-1).

**Command 🡪 (sudo apt install ansible)**

* Create a directory as “apache” in user(shoaib).

**Command 🡪 (mkdir apache)**

* Go inside the directory.

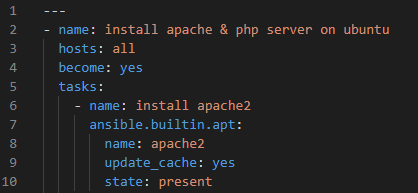
**Command 🡪 (cd apache)**

* Create a file and name it as “hosts” and paste the private\_ip of Node-1.

**Command 🡪 (sudo vi hosts)**

* Create another file and name it as “apache.yml” and paste the playbook which is written in vscode into “apache.yml” as shown below.

**Command 🡪 (sudo vi apache.yml)**



**Comment**: To write the playbook, first we need to understand the manual commands which we are going to execute to install any server/application, in this case refer here(<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-22-04>). Read and find the module in Ansible-Playbook with the required manual command, for example(<https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_module.html>).

**Note**: Don’t write the whole playbook at a time, go step-by-step while executing any command as shown in the above screenshot.

* Run a command to check the syntax saved in the “apache.yml” is right or wrong. If this shows an error means that, there is something wrong in our playbook, then we need to go into “apache.yml” and do the necessary changes and run it again and again and again, until you see the message as shown below in the screenshot.

**Command 🡪 (ansible-playbook -i hosts –syntax-check apache.yml)**

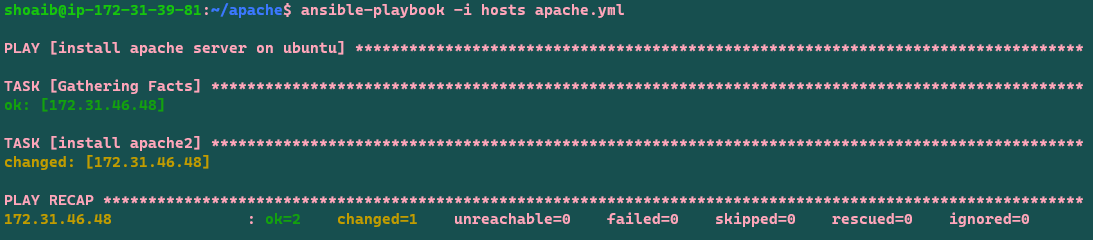


* Now it’s time to execute the “apache.yml”.

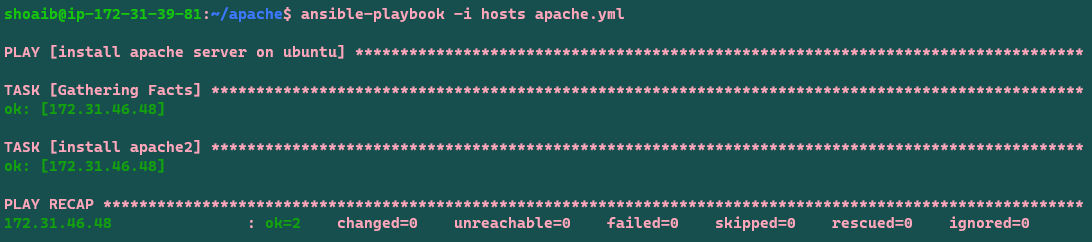
**Command 🡪 (ansible-playbook -i hosts apache.yml)**

**Comment**: “-i” means check the inventory of “hosts” file and execute “apache.yml”, it will install the same thing for all the private ip\_addresses present in the hosts’ file.

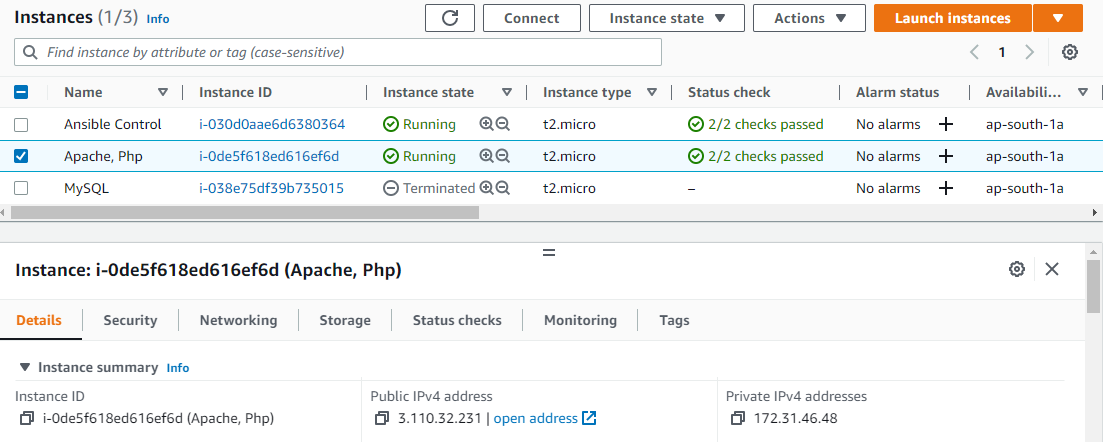
* If you see “changed=1” and it appears in yellow color means the installation of Apache2 has been done successfully.



* If you see “changed=0” and other things appear in green color means the installation of Apache2 has been done already.

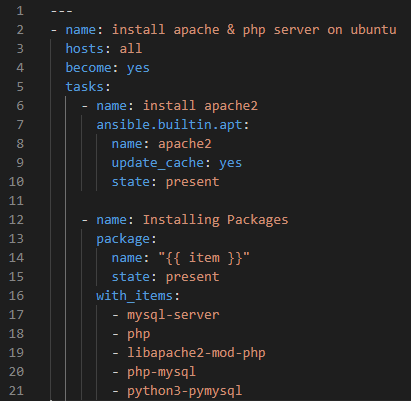


* After the above steps, cross-check that the Apache2 server is installed in Node-1 by clicking on the “open address” as shown below.



Here is the result



* ****Now go for the execution of other commands written in the playbook as shown below by again editing the file “apache.yml”.

**Command 🡪 (sudo vi apache.yml)**

**Note**: The above screenshot tells us that, we are installing list of items from the package. It installs mysql-server, php, libapache2-mod-php, php-mysql and python3-pymysql.

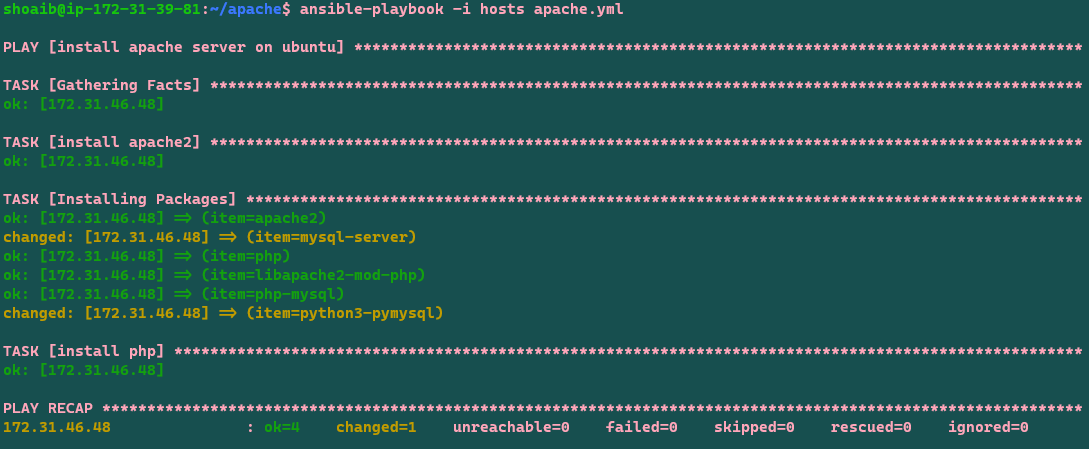
* Again, cross-check the syntax

**Command** 🡪 **(ansible-playbook -i hosts --syntax-check apache.yml)**

* Then run “apache.yml”.

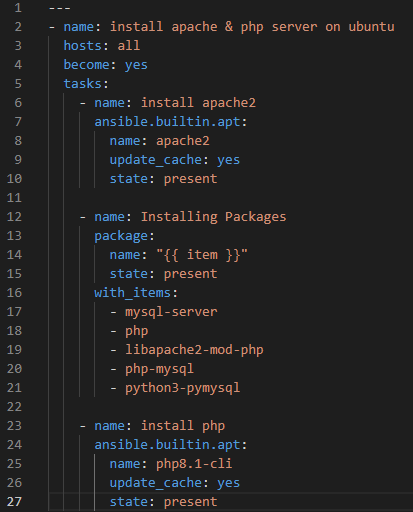
**Command** 🡪 **(ansible-playbook -i hosts apache.yml)**

* This time, it gives a result as shown below.



* Now, we need to install “php8.1-cli” which is mentioned in the document. So, we are again editing “apache.yml” and paste the playbook which is shown below into “apache.yml”.

**Command 🡪 (sudo vi apache.yml)**



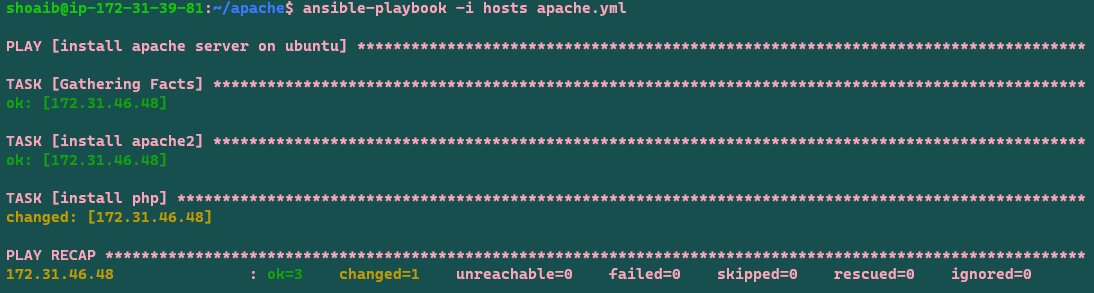
* Again, cross-check the syntax

**Command** 🡪 **(ansible-playbook -i hosts --syntax-check apache.yml)**

* Then run “apache.yml”.

**Command** 🡪 **(ansible-playbook -i hosts apache.yml)**

* This time, it gives a result as shown below.



* Now we need to login into the “Node-1” machine and create a file “info.php” inside a below location and paste “<?php

phpinfo();

?>”

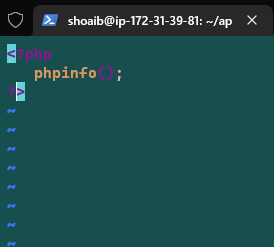
**Command** 🡪 **(sudo nano /var/www/html/info.php)**



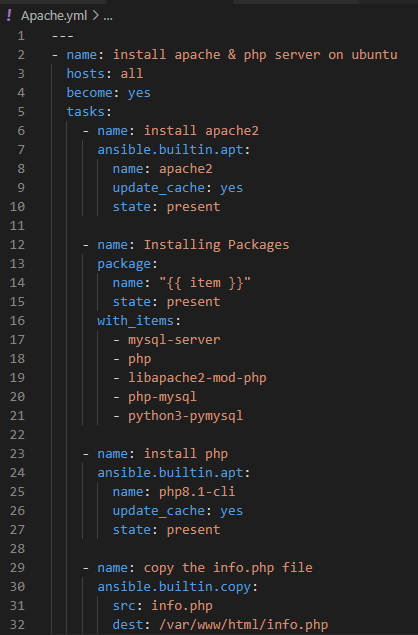
(or)

* Or else we can create a file “info.php” in apache directory in “Ansible Control” with the given syntax as shown below.

**Command** 🡪 **(sudo vi info.php)**



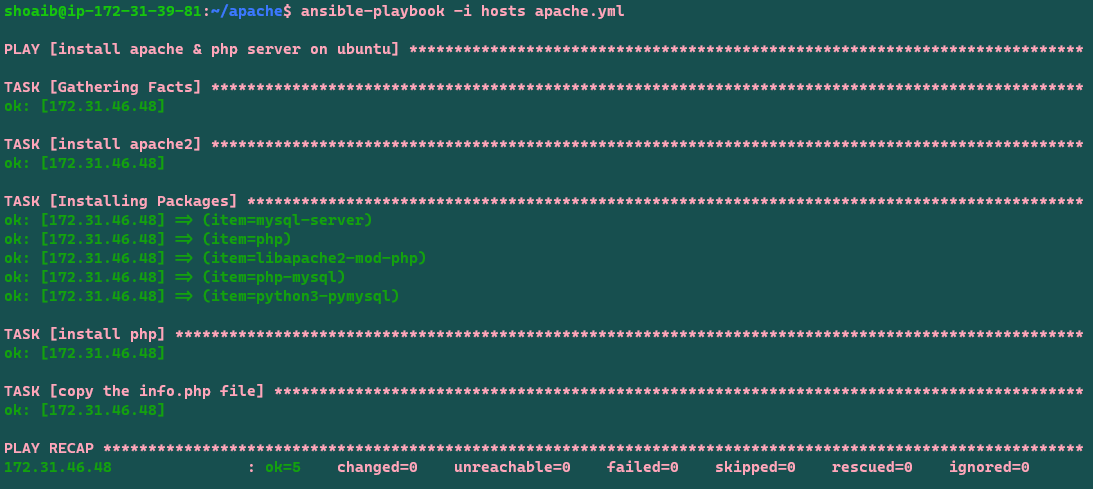
* Copy the “info.php” file into “Node-1”. So, we need to edit the playbook and continue writing the tasks as shown below. And then run the commands.



**Command** 🡪 **(ansible-playbook -i hosts --syntax-check apache.yml)**

**Command** 🡪 **(ansible-playbook -i hosts apache.yml)**

**This will be the result.**

****

* Finally, we need to go to the web server page and after the public ip\_address write “/info.php”, it takes us to PHP page.

