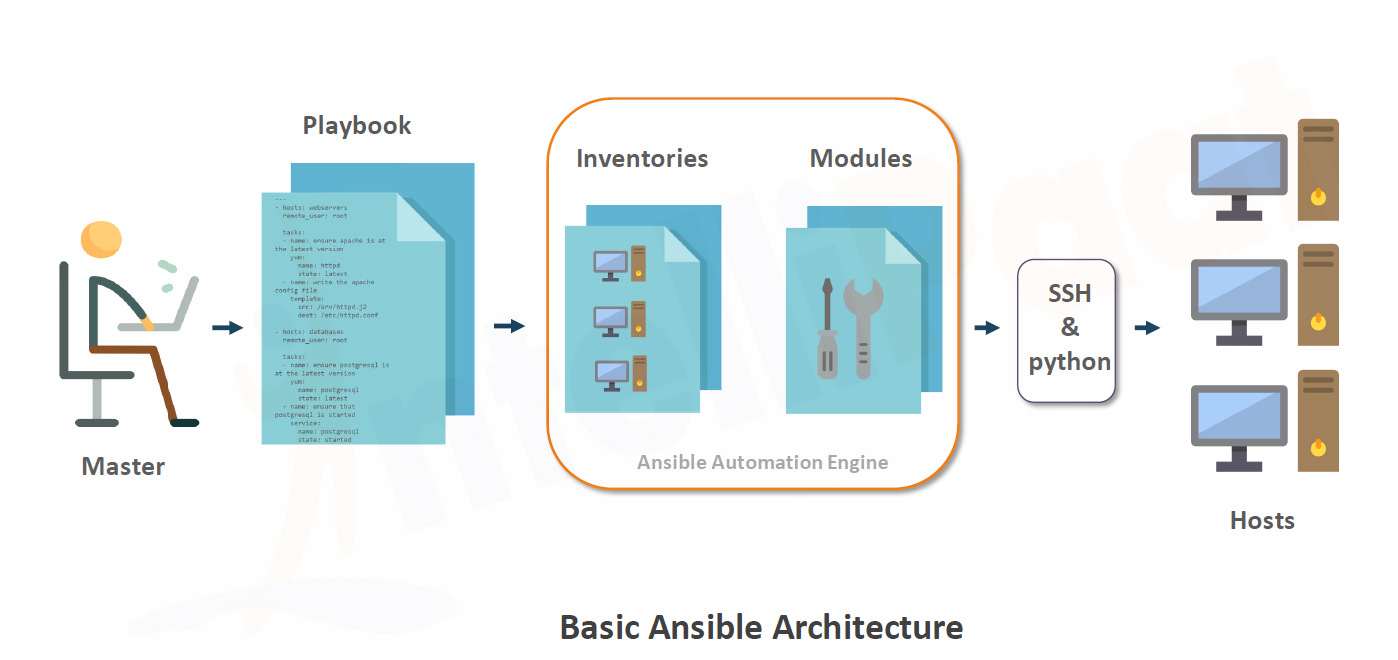
**Ansible**

* Ansible is open-source Configuration Management Tool
* How does Ansible works: With the help of Ansible Playbooks a text file which is written in a very simple language called YAML
* Place them only on Ansible server only.
* Extension of YAML file is. yaml/yml
* Not Required to install any Ansible agents on remote machines.
* On all remote machine’s python is mandatory

**Ansible Architecture**

**Ansible Architecture Master**

Master 🡪 Playbook 🡪

1. Describes the tasks to be executed.
2. Written in simple language.
3. Playbooks are like instruction manuals.

**Ansible Architecture Inventories**

Master 🡪 Playbook 🡪Inventories

1. Lists the hosts.
2. Where playbook tasks will be operated

**Ansible Architecture Modules**

Master 🡪 Playbook 🡪Inventories 🡪 Modules

1. Modules are like tools.
2. Can control system resources, like services, packages etc.
3. 500 +core Modules
4. Also allow custom modules

**Ansible Architecture Modules**

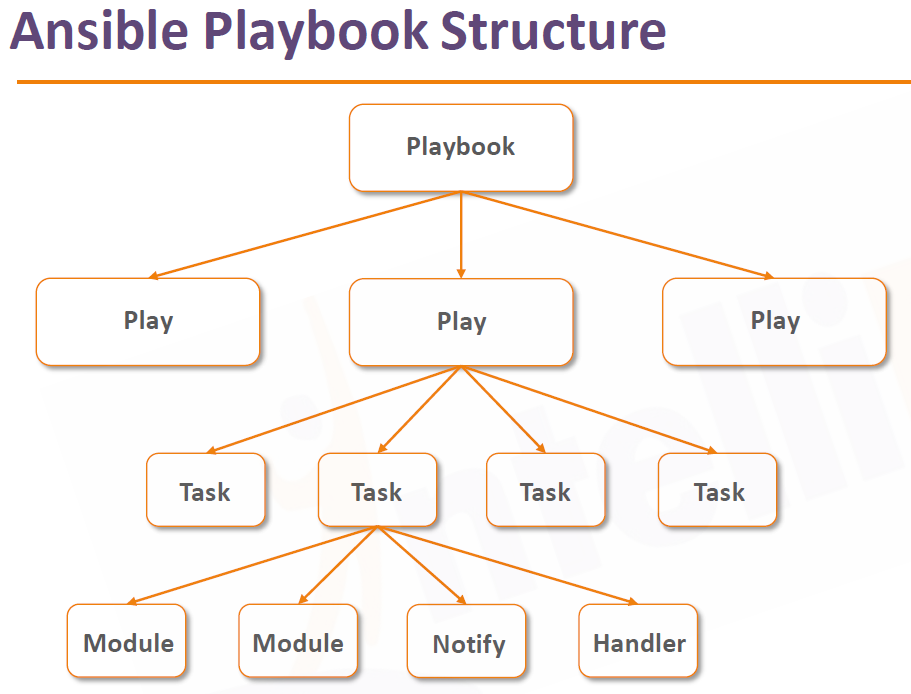
1. Master 🡪 Playbook 🡪Inventories 🡪 Modules🡪 SSH & Python 🡪 Hosts

**What is Ansible inventory?**

Inventory is a text file inside which we need to mention the list of all our server where ansible will target the automation.

**What is Ansible Playbook?**

1. An organized unit of scripts
2. Defines work for a server configuration.
3. Written in YAML.
4. Playbook has number of plays.
5. Plays contain tasks.
6. Tasks calls core or custom modules.
7. Handler gets tiggered from notify and executed at the end of only once.



**ANSIBLE INSTALLATION ON UBUNTU**

Ansible installation documentation has been divided into 3 segments.

1. A-Install Ansible
2. B-Configure the SSH Access to the Ansible Hosts
3. C-Setting Up Ansible Hosts

Change from normal user to root

$ sudo su –



To logout from the root press CTRL + D

Change to root user



Note:-

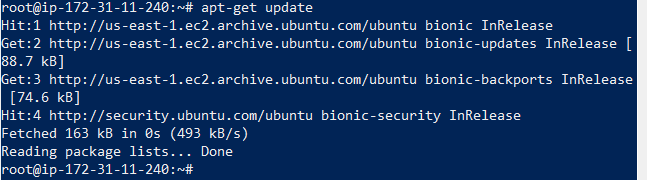
if you are root user then use directly $ apt-get install …

If you are normal user then all the commands will be $ sudo apt-get install

**A-Install Ansible**

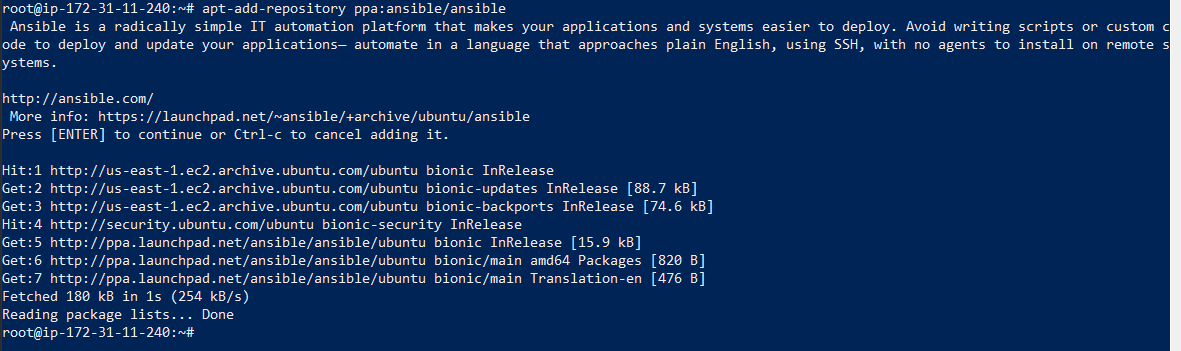
1. Update

**$ apt-get update**



1. Add the Ansible PPA (personal package archive)

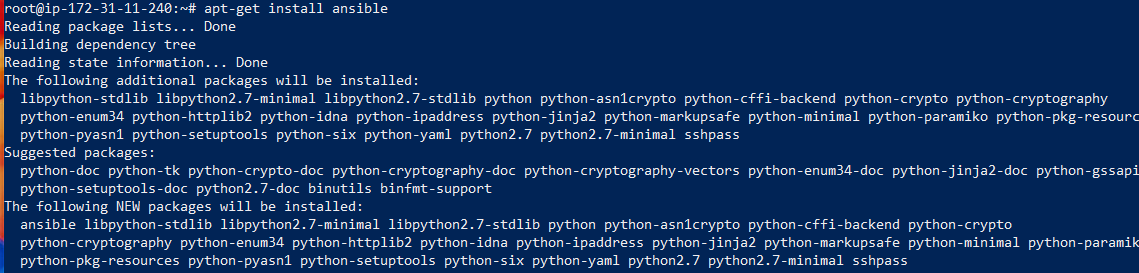
**$ apt-add-repository ppa:ansible/ansible**

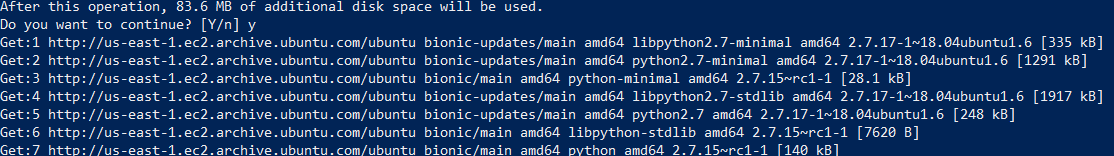


Press Enter to accept the PPA addition.

1. Install Ansible software

**$ apt-get install ansible**



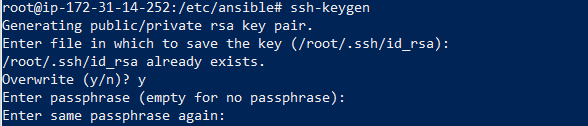


Now, your Ansible server now has all the software required to administer your hosts.

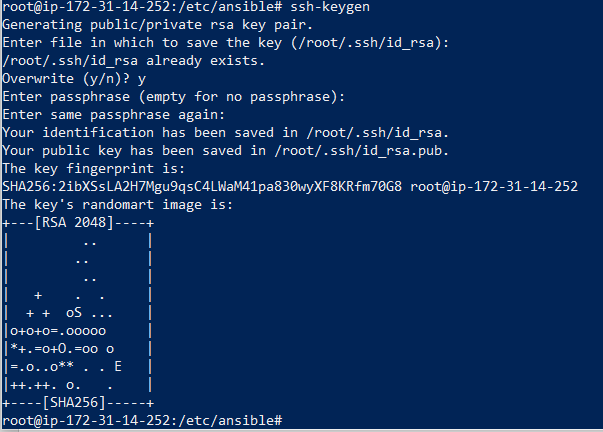
**B-Configure the SSH Access to the Ansible Hosts**

1. Create a key pair on the master terminal.

**$ ssh-keygen**



Press Enter, you will get the following output. You can optionally enter passphrase to prevent unauthorized users from logging in.

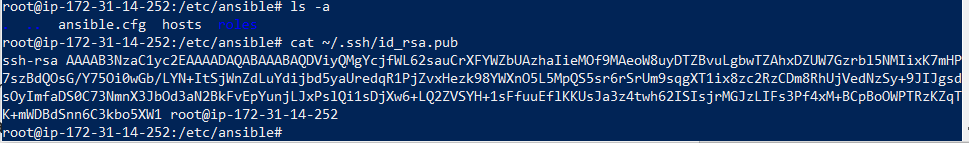


Now you have a public and private key. You can use them to authenticate from the slave terminal.

1. On the master’s terminal use the cat command.

**$Cat ~/.ssh/id\_rsa.pub**

ssh/**id\_rsa** and the public key is stored in ~/. ssh/**id\_rsa**. **pub** . The private key should only be kept on your local system and should be encrypted using a passphrase that is at least as strong as any password you would normally use



Copy the output in clipboard. Now we will open the authorized\_keys within the ~/.ssh directory and paste the copied output there.

**C-Setting Up Ansible Hosts**

Go to the client machines’ root user and open the authorized\_keys within the ~/.ssh directory.

**$sudo su –**

**$ ls -a**

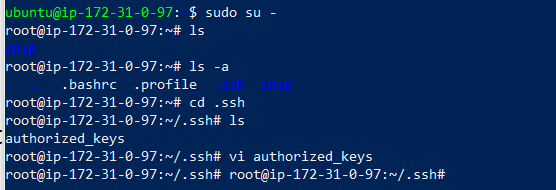
**$ ca .ssh**

**ssh** is a **DIRECTORY** and ~/. **ssh**/authorized keys is a file containing a list of public keys, one per line, allowed to access the server. The .**ssh directory** is not by default created below your home **directory**.

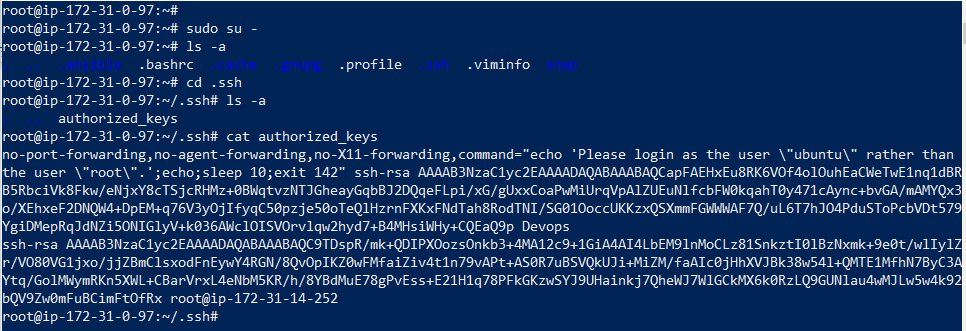
**$ ls -a**

**$ Vi authorised keys**

The **authorized\_keys** file in **SSH** specifies the **SSH** keys that can be used for logging into the user account for which the file is configured. It is a highly important configuration file, as it configures permanent access using **SSH** keys and needs proper management



**$ cat authorized\_keys**



Go to the client machine and copy the ssh/**id\_rsa** key as mentioned above in the **authorized\_keys** file

ip-172-31-0-97.ec2.internal

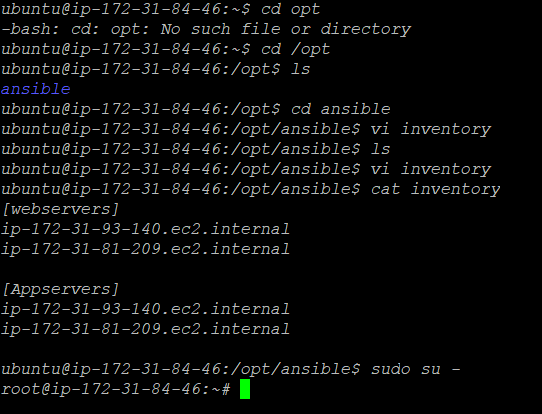
ip-172-31-9-201.ec2.internal

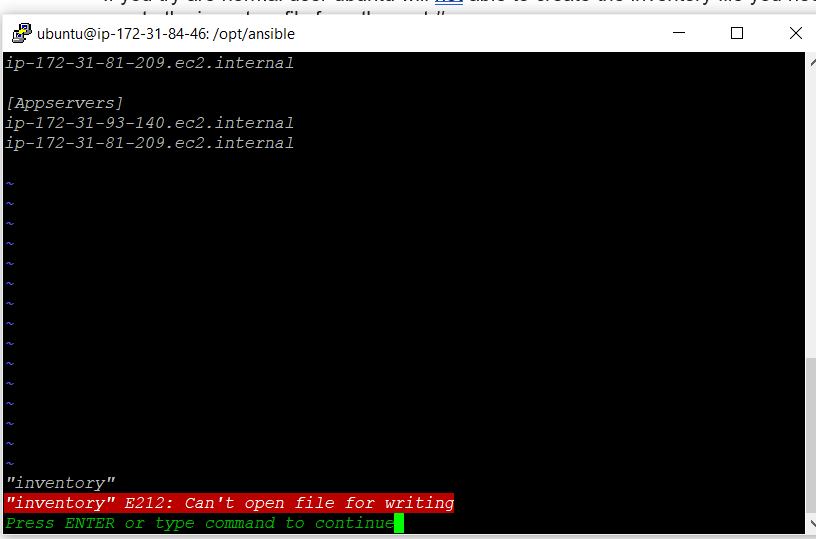
**Creating the inventory file in the Master machine**

If you try as normal user ubuntu will not able to create the inventory file you need to create the inventory file from the root #

Change the directory to opt

Create a directory ansible i.e mkdir ansible

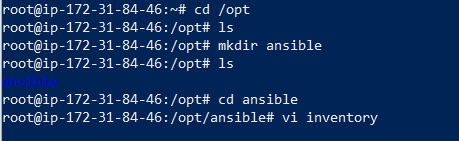




From the root #

Change the directory to opt

Create a directory ansible i.e mkdir ansible



Under ansible directory create inventory file that contains the host IP address

Vi inventory

[webservers]

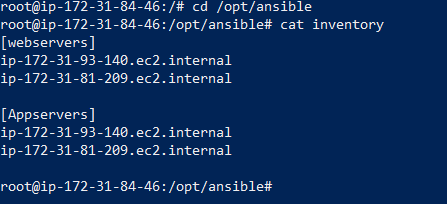
ip-172-31-0-97.ec2.internal

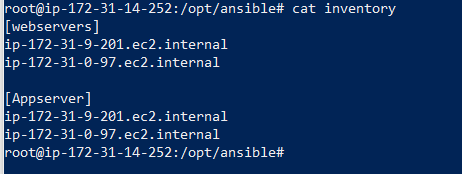
ip-172-31-9-201.ec2.internal

Note: while creating groups name don’t put “-“ will warning message use “\_”

[dev-webservers] it will give warning message.

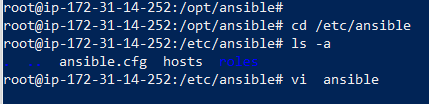
[dev\_webservers] it is correct.





Now got /etc/asnsible path do ls -a

We need to change the ansible.cfg file



Make the following changes in the ansible.cfg file

root@ip-172-31-14-252:/etc/ansible# cat ansible.cfg

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

# ===============================================

[defaults]

# some basic default values...

#inventory = /etc/ansible/hosts

change this path from default path to otp

inventory = /opt/ansible/inventory

# uncomment this to disable SSH key host checking

Uncomment this hot key and make it to false

host\_key\_checking = False

# SSH timeout

Customise the time

timeout = 5

# logging is off by default unless this path is defined

# if so defined, consider logrotate

Change the default path log\_path = /var/log/ansible.log

to opt

log\_path = /opt/ansible/ansible.log

**How to run Ansible?**

1. Ansible adhoc method:

We only can perform 1 activity.

It mostly used for testing connectivity.

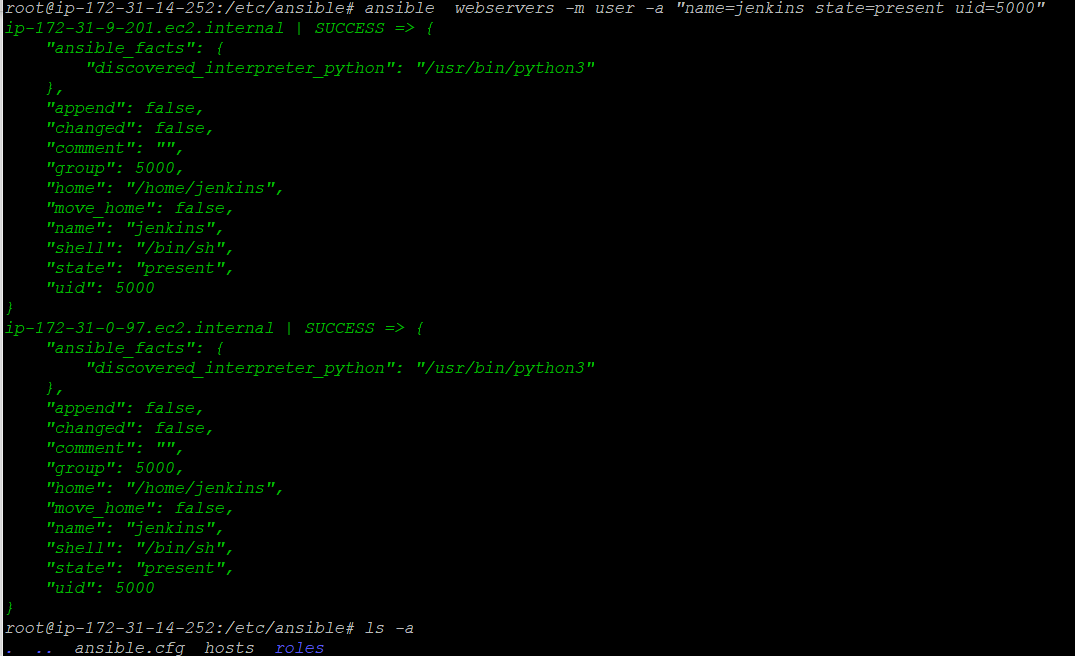
1. Ansible playbooks method
2. **Running Ansible adhoc method:**

**$ ansible <group name> -m <module name> -a <option>**

**$ ansible webservers -m user -a “name=Jinkins state=present uid=500”**



In the host system the changes are reflected that indicates the yellow color with message as changed, again run the same command it will do nothing and it changes are in green color the message as success



**C-Testing the connection**

In Ansible Server, run the following command.

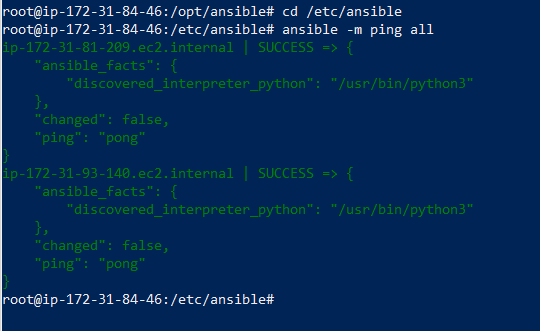
**$ ansible -m ping all**

There are 20 groups 500 servers in rare conditions we need to update some command

To run all

Build in command health check

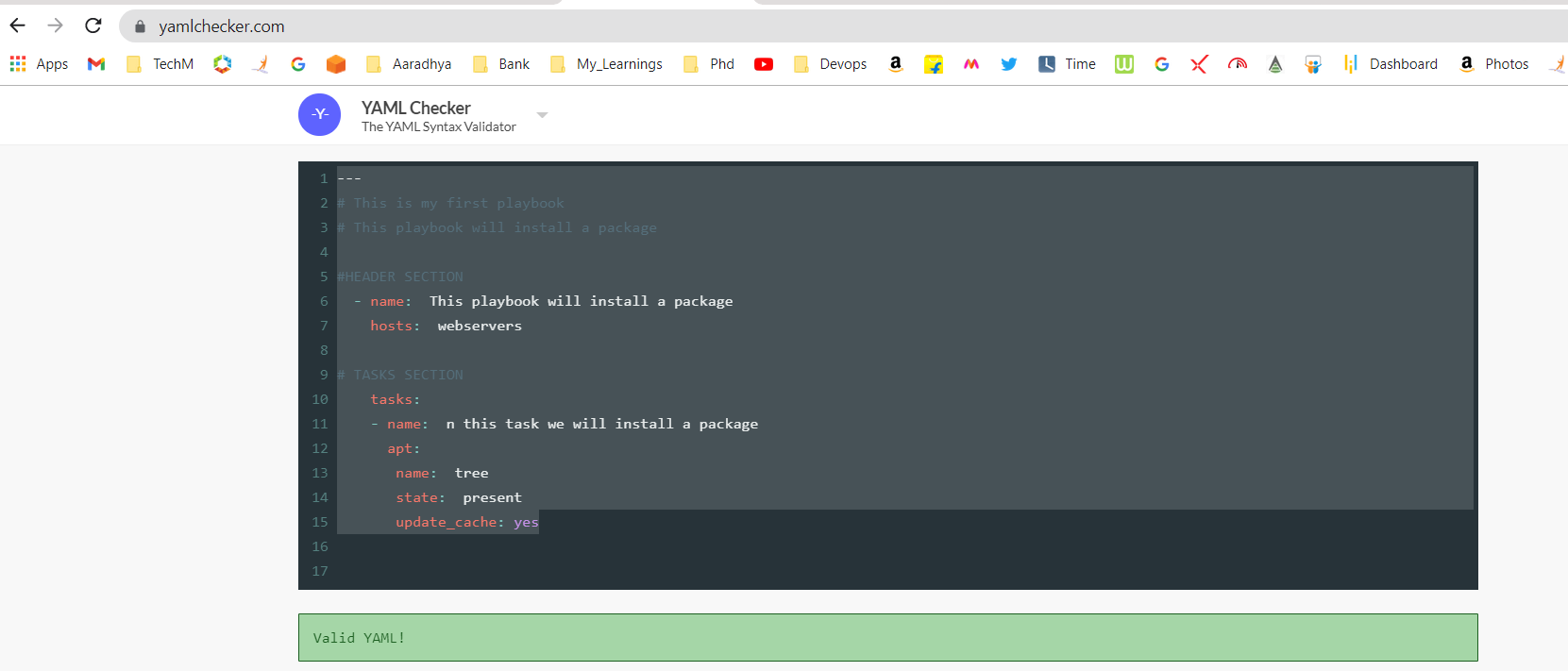
**$ ansible all -m ping**



1. **Running Ansible playbook method:**

Go to root # and change the directory to opt and create the yaml file

Creating the yaml file go to online yaml checker or validator and built the yaml file as shown below



Sample yaml file

---

# This is my first playbook

# This playbook will install a package

#HEADER SECTION

- name: This playbook will install a package

hosts: webservers

# TASKS SECTION

tasks:

- name: n this task we will install a package

apt:

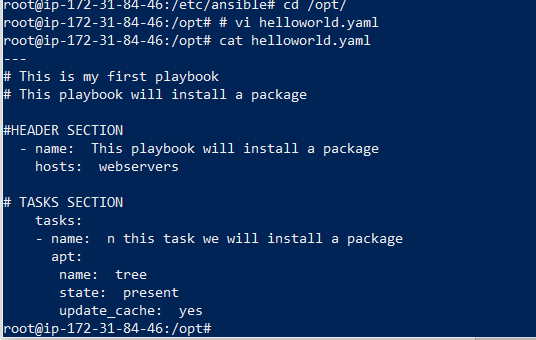
name: tree

state: present

update\_cache: yes

**$ cd /opt/**

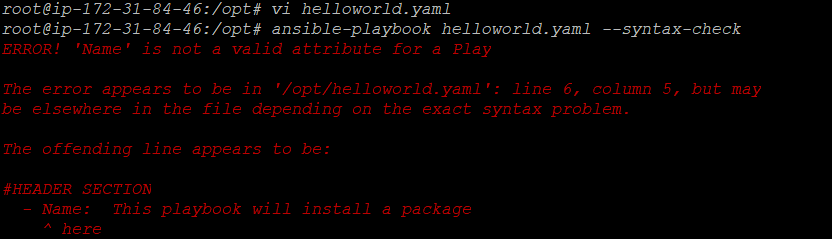
**$ vi helloworld.yaml**



Run the playbook before running the playbook check the syntax of the yaml file for any errors.

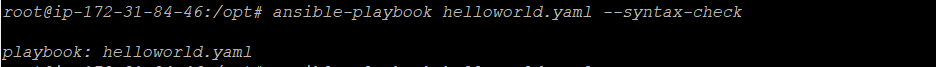
**$ ansible-playbook helloworld.yaml --syntax-check**

If any errors edit it run again



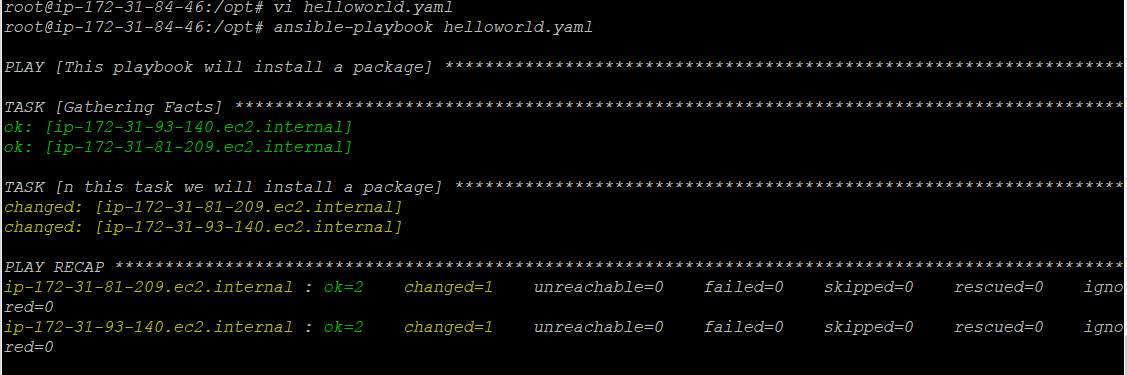
After fixing the errors again check for syntax

$ ansible-playbook helloworld.yaml --syntax-check

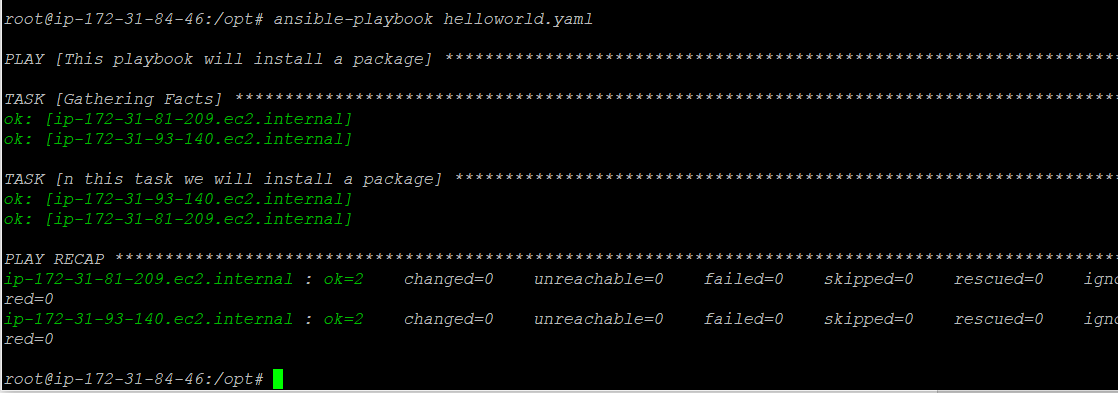


No errors now run the playbook. First time color is change to yellow

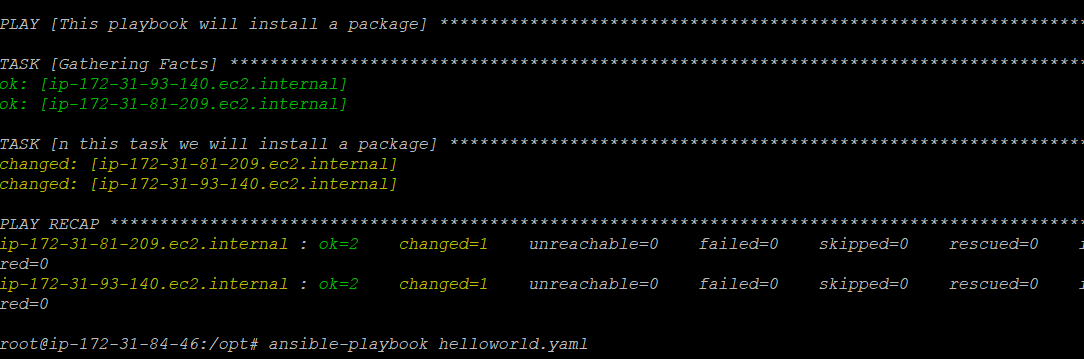
**$ ansible-playbook helloworld.yaml**



Again run the playbook. Now nothing change so the color is green



**Output of the playbook**



**---**

**# This is my first playbook**

**# Header Section**

**-name: This playbook will install the package**

**host: webservers**

**vars:**

**mypackage: vim**

**myfile: helloworld.yaml**

**# TASKS SECTION**

**tasks:**

**- name: to install a package "{{ mypackage }}"**

**apt:**

**name: "{{ mypackage }}"**

**state: present**

**- name: to copy a file name "{{ myfile }}"**

**copy:**

**scr: "/opt/{{ myfile }}"**

**dest: /opt/{{ myfile }}**

**mode: 0644**