# Deploying Web Server on AWS through ANSIBLE!!

**Ansible (RH 294)**

**Task:**

♦Provision EC2 instance through ansible.

♦Retrieve the IP Address of instance using a dynamic inventory concept.

♦Configure the webserver through ansible!

## Amazon Web Services:

Amazon Web Services (AWS) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing web services provide a variety of basic abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet.

## Ansible:

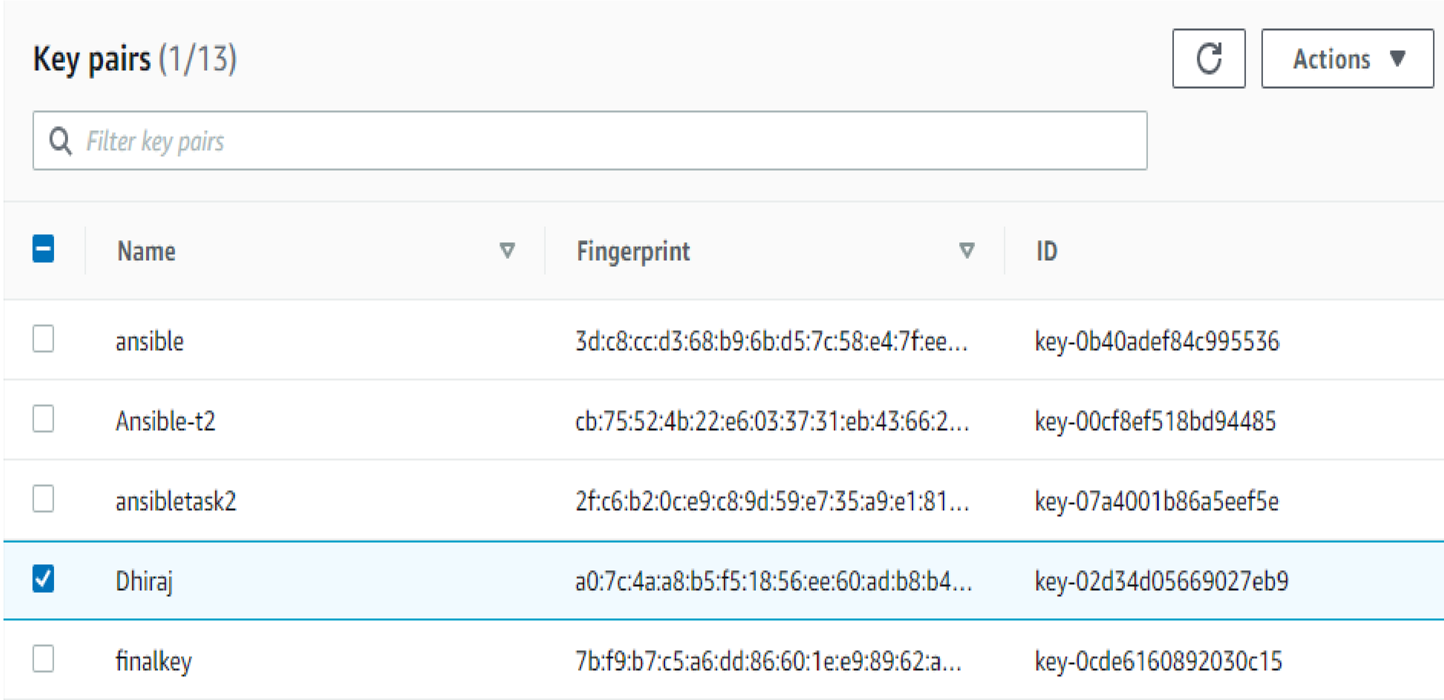
Ansible is an open-source software provisioning, configuration management, and application-deployment tool enabling infrastructure as code. It runs on many Unix-like systems and can configure both Unix-like systems as well as Microsoft Windows. It includes its own declarative language to describe system configuration. Ansible was written by Michael DeHaan and acquired by Red Hat in 2015.

.........................................................................................................................................

First of all, we have to set up an Ansible environment in our system. For doing anything on the aws using the local system with the help of ansible then you have to install boto library of python.

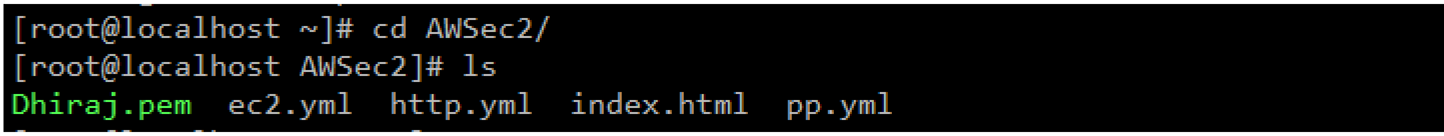
```
# pip3 install boto
# pip3 install boto3
```

Then we create a key pair named as Dhiraj.pem and download key pair. Then paste the same key pair in file contains our main ec2.yml playbook.



Key pairs (1/13)

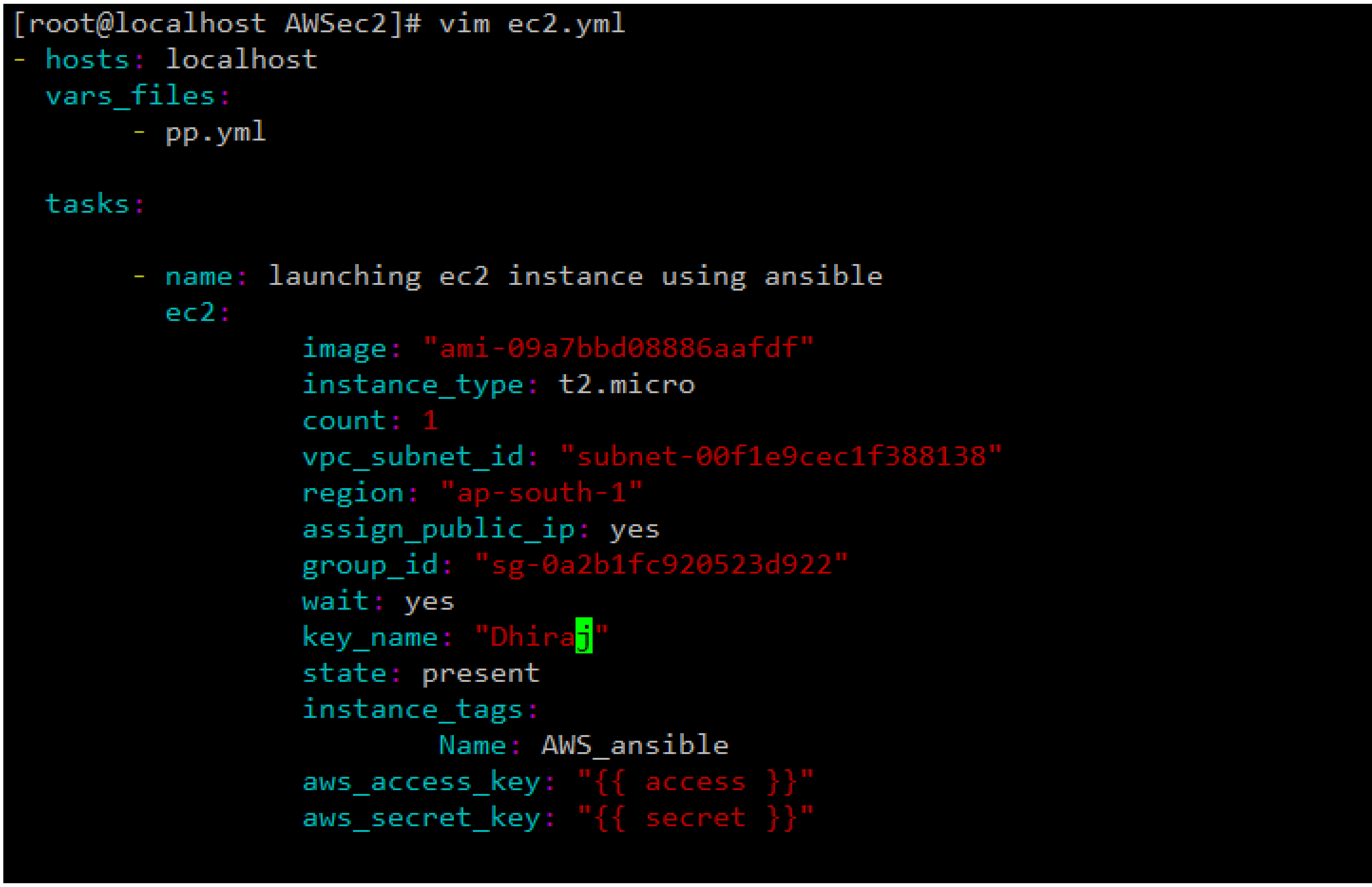
| ▬ | Name | Fingerprint | ID |
|---|---|---|---|
| □ | ansible | 3d:c8:cc:d3:68:b9:6b:d5:7c:58:e4:7f:ee... | key-0b40adef84c995536 |
| □ | Ansible-t2 | cb:75:52:4b:22:e6:03:37:31:eb:43:66:2... | key-00cf8ef518bd94485 |
| □ | ansibletask2 | 2f:c6:b2:0c:e9:c8:9d:59:e7:35:a9:e1:81... | key-07a4001b86a5eef5e |
| ☑ | Dhiraj | a0:7c:4a:a8:b5:f5:18:56:ee:60:ad:b8:b4... | key-02d34d05669027eb9 |
| □ | finalkey | 7b:f9:b7:c5:a6:dd:86:60:1e:e9:89:62:a... | key-0cde6160892030c15 |



```
[root@localhost ~]# cd AWSec2/
[root@localhost AWSec2]# ls
Dhiraj.pem  ec2.yml  http.yml  index.html  pp.yml
```

Now first we have to create an ec2 instance on AWS by writing .yml code.

ec2.yml:



```
[root@localhost AWSec2]# vim ec2.yml
- hosts: localhost
  vars_files:
       - pp.yml

  tasks:

       - name: launching ec2 instance using ansible
         ec2:
                 image: "ami-09a7bbd08886aafdf"
                 instance_type: t2.micro
                 count: 1
                 vpc_subnet_id: "subnet-00f1e9cec1f388138"
                 region: "ap-south-1"
                 assign_public_ip: yes
                 group_id: "sg-0a2b1fc920523d922"
                 wait: yes
                 key_name: "Dhiraj"
                 state: present
                 instance_tags:
                         Name: AWS_ansible
                 aws_access_key: "{{ access }}"
                 aws_secret_key: "{{ secret }}"
```

Create one more pp.yml file for store the my-access and my-secret keys.

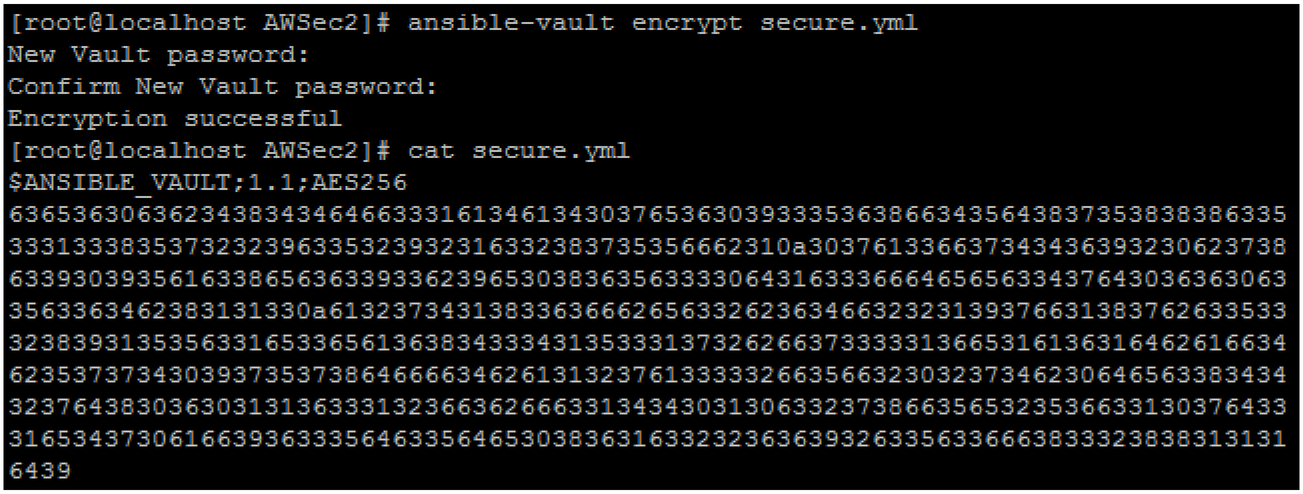
- # vim pp.yml

myaccess: "ACCESSKEY"
mysecret: "SECRETKEY"

After creating or save pp.yml file we have a vault for more secure for this we have used:

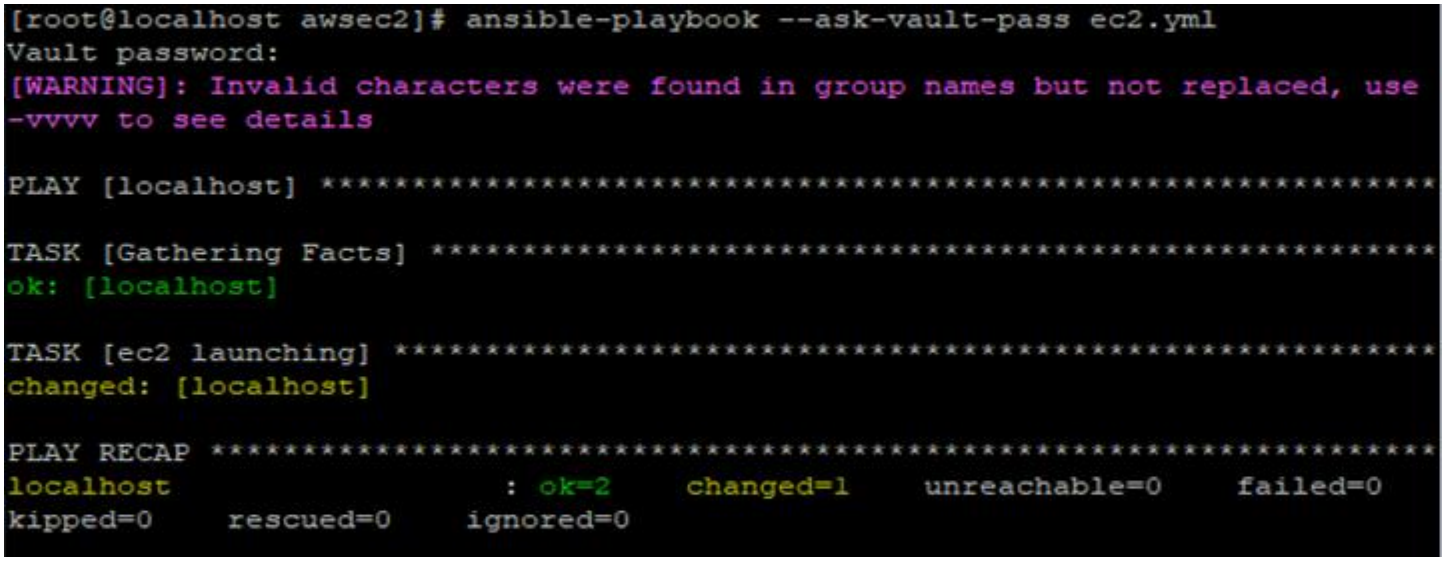
# ansible-vault encrypt pp.yml (File_Name)

For check the vault is working or not:

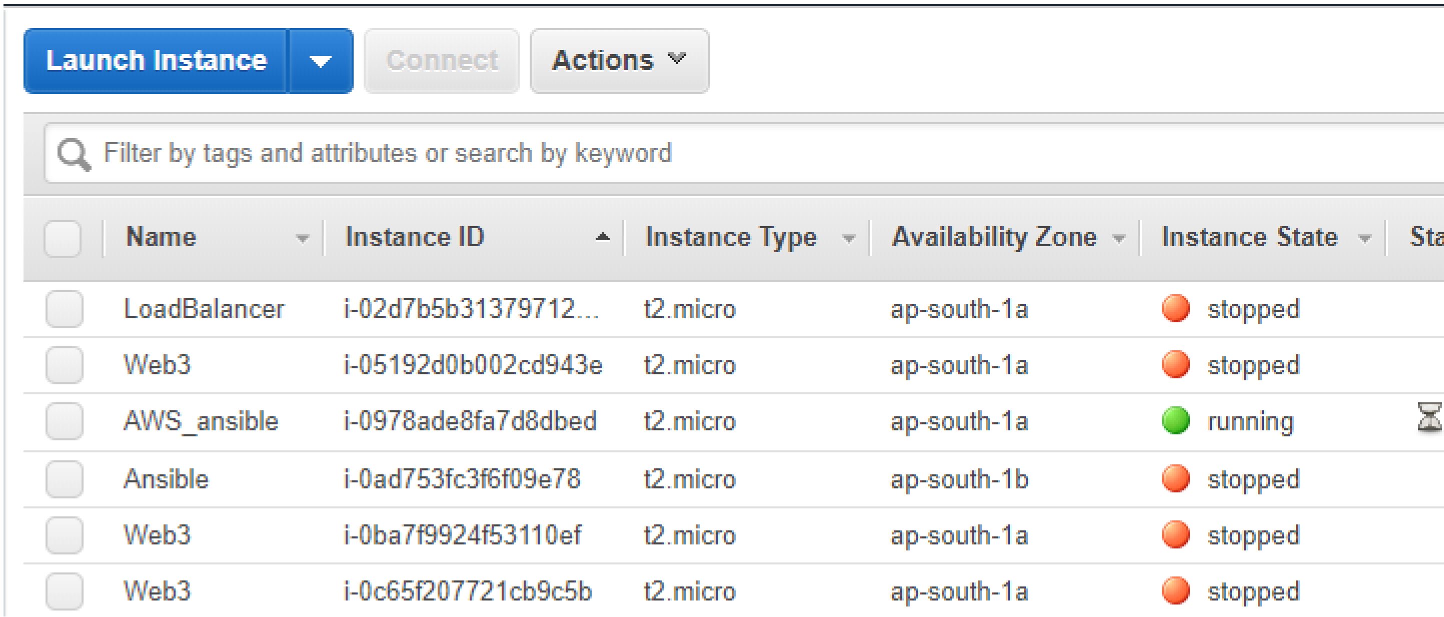


```
[root@localhost AWSec2]# ansible-vault encrypt secure.yml
New Vault password:
Confirm New Vault password:
Encryption successful
[root@localhost AWSec2]# cat secure.yml
$ANSIBLE_VAULT;1.1;AES256
63653630636234383434646633316134613430376536303933353638663435643837353838386335
3331333835373232396335323932316332383735356662310a303761336637343436393230623738
63393039356163386563633933623965303836356333306431633366646565633437643036363063
3563363462383131330a613237343138336366626563326236346632323139376631383762633533
32383931353563316533656136383433343135333137326266373333313665316136316462616634
62353737343039373537386466663462613132376133333266356632303237346230646563383434
32376438303630313136333132366362666331343430313063323738663565323536633130376433
31653437306166393633356463356465303836316332323636393263356336663833323838313131
6439
```

Now run the ansible-playbook to launch ec2 instance:



```
[root@localhost awsec2]# ansible-playbook --ask-vault-pass ec2.yml
Vault password:
[WARNING]: Invalid characters were found in group names but not replaced, use
-vvvv to see details

PLAY [localhost] ****************************************************************

TASK [Gathering Facts] **********************************************************
ok: [localhost]

TASK [ec2 launching] ************************************************************
changed: [localhost]

PLAY RECAP **********************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0
kipped=0    rescued=0    ignored=0
```



| □ | Name | Instance ID | Instance Type | Availability Zone | Instance State | Sta |
|---|---|---|---|---|---|---|
| □ | LoadBalancer | i-02d7b5b31379712... | t2.micro | ap-south-1a | ● stopped | |
| □ | Web3 | i-05192d0b002cd943e | t2.micro | ap-south-1a | ● stopped | |
| □ | AWS_ansible | i-0978ade8fa7d8dbed | t2.micro | ap-south-1a | ● running | ⌛ |
| □ | Ansible | i-0ad753fc3f6f09e78 | t2.micro | ap-south-1b | ● stopped | |
| □ | Web3 | i-0ba7f9924f53110ef | t2.micro | ap-south-1a | ● stopped | |
| □ | Web3 | i-0c65f207721cb9c5b | t2.micro | ap-south-1a | ● stopped | |

Now here we are using python code to find the IP of instance dynamically. So we download this code from GitHub in the directory /etc/ansible:

```
# wget
https://raw.githubusercontent.com/sanket3122/Ansible_Task2/master/ec2.py
```

```
# wget
https://raw.githubusercontent.com/sanket3122/Ansible_Task2/master/ec2.ini
```

Now to make this files executable run following commands:

```
# chmod +x ec2.py
```

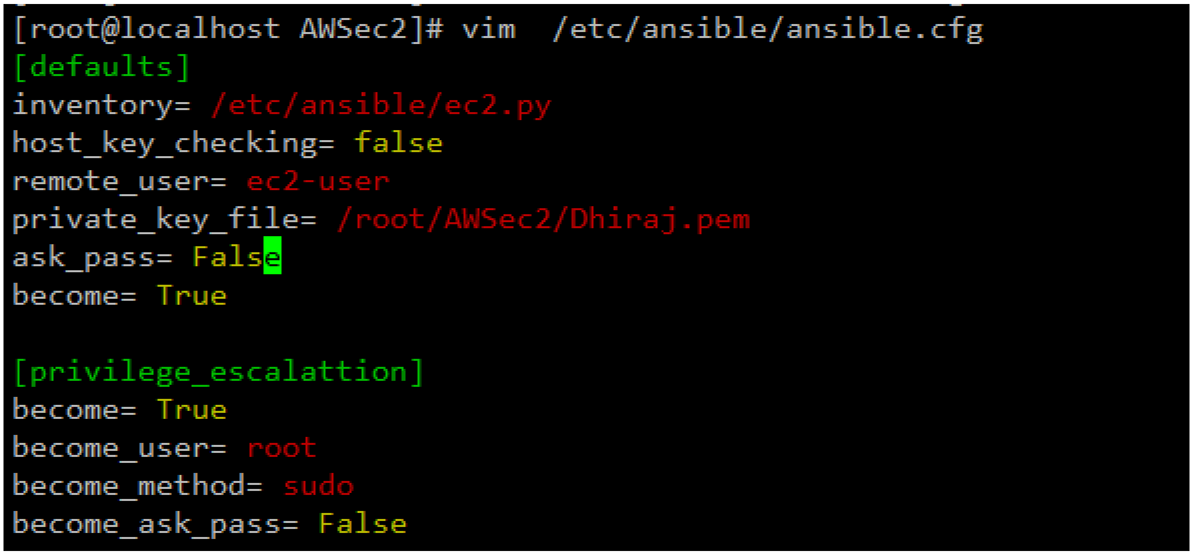
```
# chmod +x ec2.ini
```

We need to initiate them, run the following commands, and Provide your aws credentials.

```
# export EC2_INI_PATH=path_of_ec2.ini_file
# export AWS_ACCESS_KEY_ID="aws_access_key"
# export AWS_SECRET_ACCESS_KEY="aws_secret_key"
```

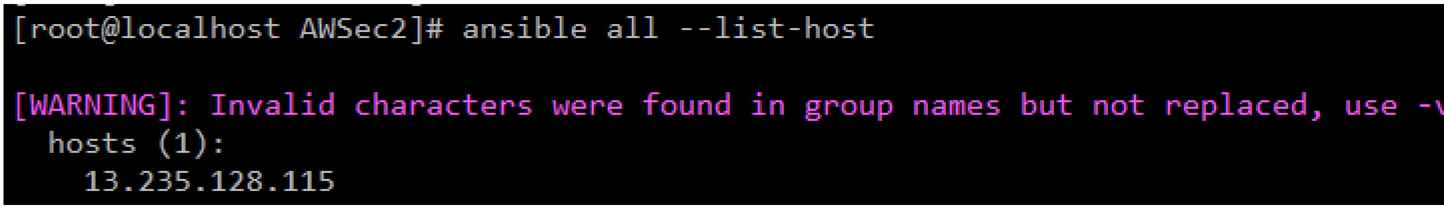
Python Script:

Python script fetches the IP of aws ec2-instance, also will play the role of dynamic inventory. We need to configure inventory and add some other details.



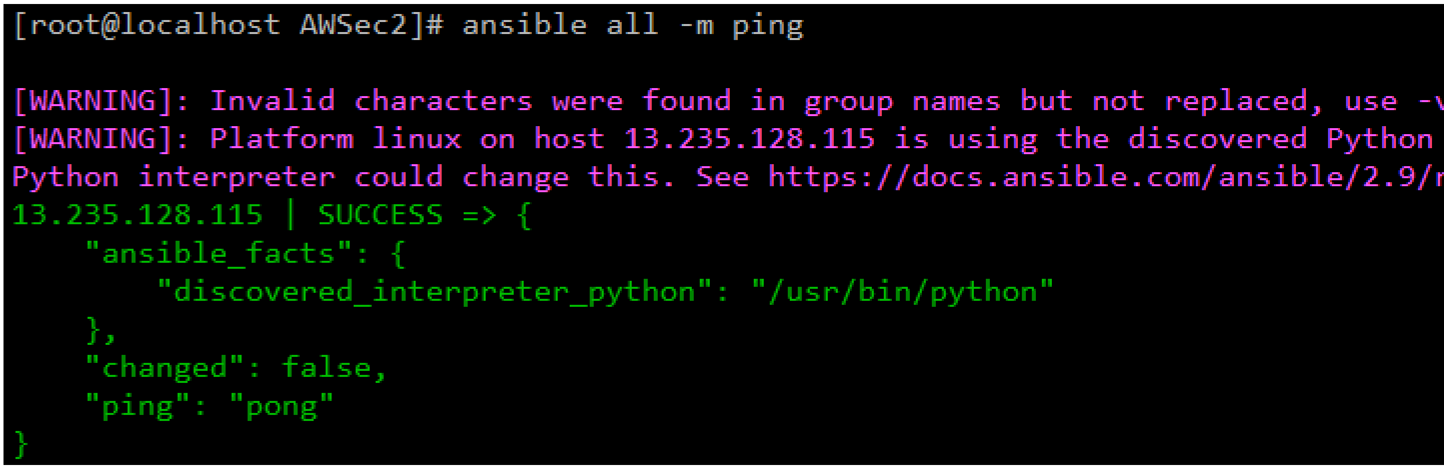
```
[root@localhost AWSec2]# vim  /etc/ansible/ansible.cfg
[defaults]
inventory= /etc/ansible/ec2.py
host_key_checking= false
remote_user= ec2-user
private_key_file= /root/AWSec2/Dhiraj.pem
ask_pass= False
become= True

[privilege_escalattion]
become= True
become_user= root
become_method= sudo
become_ask_pass= False
```

Save the inventory file and run following command, we will get ec2-instance IP:



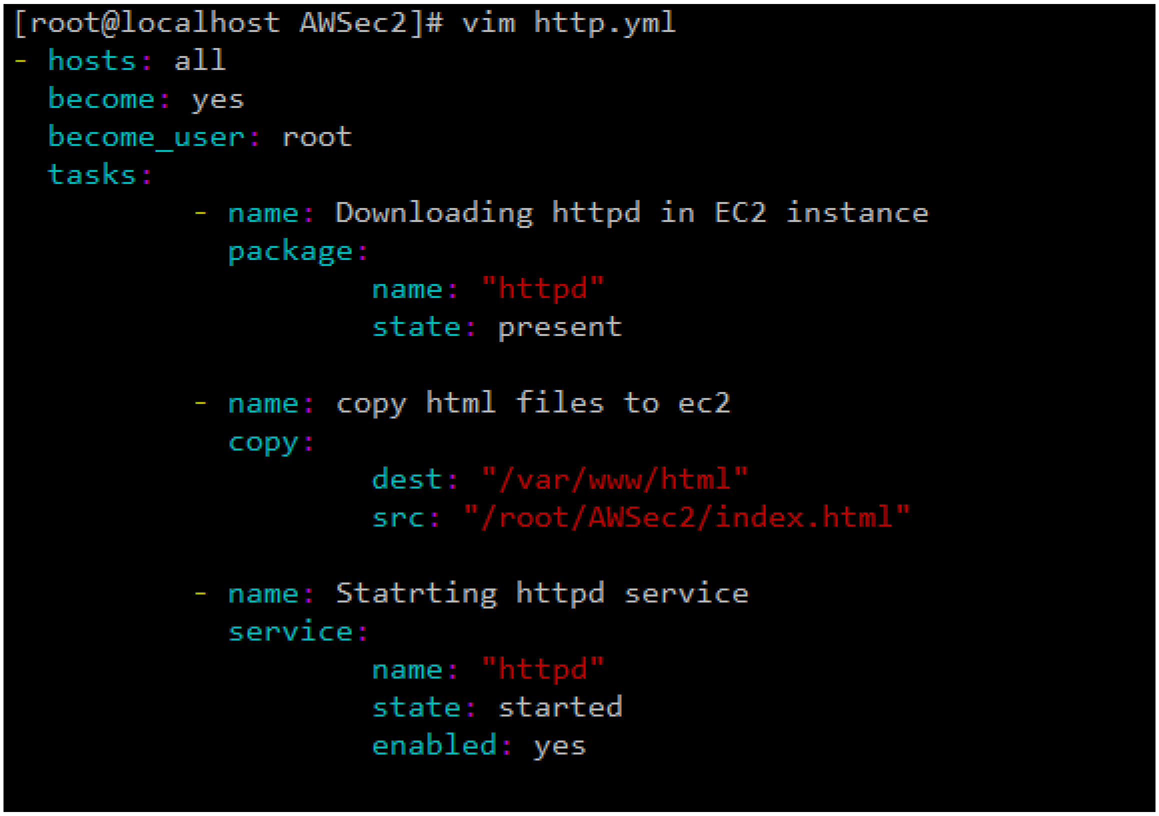
```
[root@localhost AWSec2]# ansible all --list-host

[WARNING]: Invalid characters were found in group names but not replaced, use -v
  hosts (1):
    13.235.128.115
```

Now check that IP is properly pinging or not:



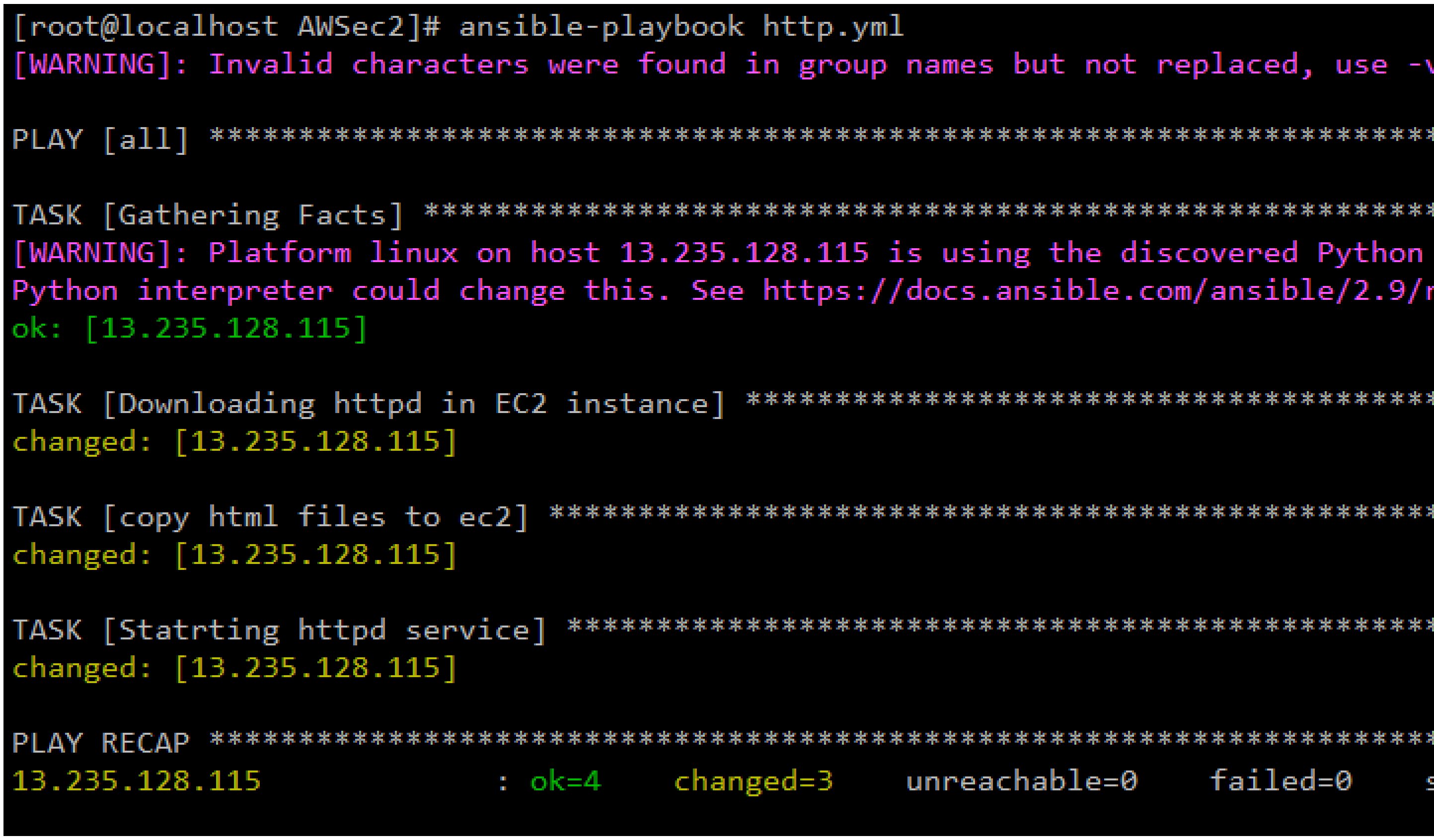
```
[root@localhost AWSec2]# ansible all -m ping

[WARNING]: Invalid characters were found in group names but not replaced, use -v
[WARNING]: Platform linux on host 13.235.128.115 is using the discovered Python
Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/r
13.235.128.115 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

Now we have to configure webserver on the ec2 instance so we have to write web.yml code.



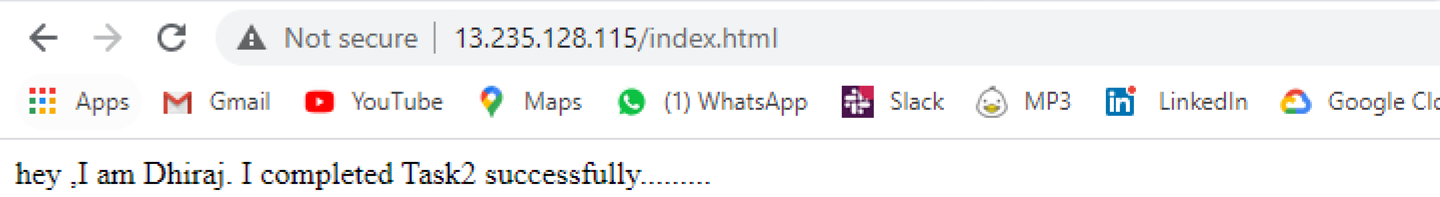
```
[root@localhost AWSec2]# vim http.yml
- hosts: all
  become: yes
  become_user: root
  tasks:
          - name: Downloading httpd in EC2 instance
            package:
                    name: "httpd"
                    state: present

          - name: copy html files to ec2
            copy:
                    dest: "/var/www/html"
                    src: "/root/AWSec2/index.html"

          - name: Statrting httpd service
            service:
                    name: "httpd"
                    state: started
                    enabled: yes
```

Save all the files and run ansible will get the IP of our running instance and configure it.



```
[root@localhost AWSec2]# ansible-playbook http.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -v

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

TASK [Gathering Facts] ***********************************************************
[WARNING]: Platform linux on host 13.235.128.115 is using the discovered Python
Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/r
ok: [13.235.128.115]

TASK [Downloading httpd in EC2 instance] *****************************************
changed: [13.235.128.115]

TASK [copy html files to ec2] ****************************************************
changed: [13.235.128.115]

TASK [Statrting httpd service] ***************************************************
changed: [13.235.128.115]

PLAY RECAP ***********************************************************************
13.235.128.115             : ok=4    changed=3    unreachable=0    failed=0    s
```

Now everything is done properly. Here our ec2-instance configure successfully, now check the web-browser.



hey ,I am Dhiraj. I completed Task2 successfully.........

***Thanks for reading!!***