Configuring Load Balancer on AWS using Ansible with dynamic inventory!

Task Description:

◆ Provision EC2 instances through ansible.

◆ Retrieve the IP Address of instances using the dynamic inventory concept.

◆ Configure the web servers through the ansible role.

◆ Configure the load balancer through the ansible role.

◆ The target nodes of the load balancer should auto-update as per the status of web servers.

Load Balancer-

A load balancer serves as the single point of contact for clients. The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. This increases the availability of your application. You add one or more listeners to your load balancer.

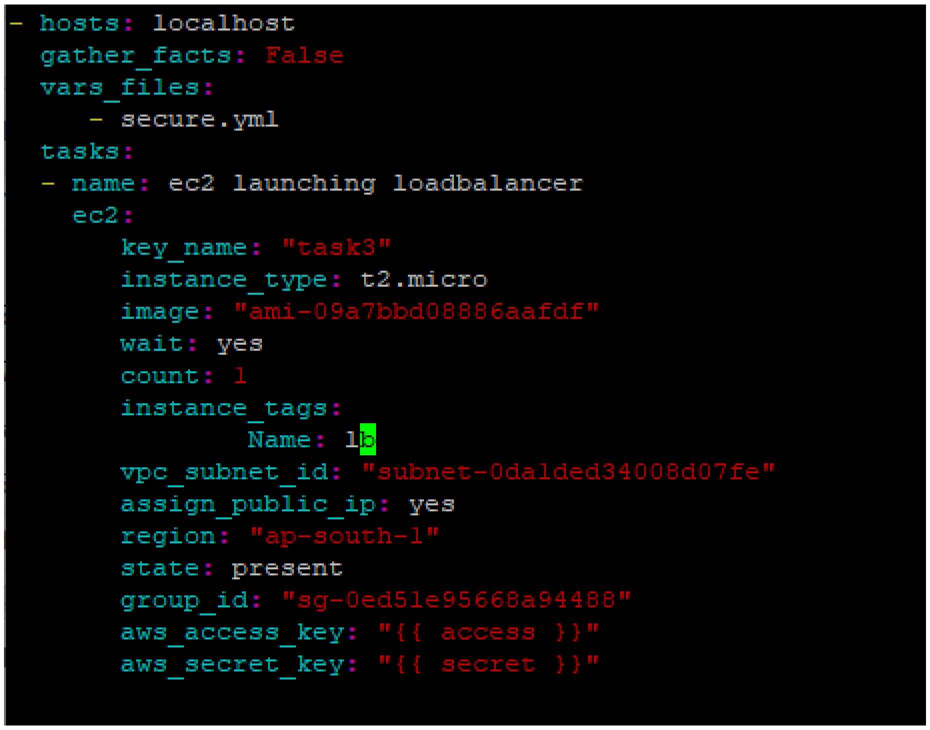
Load Balancer:-

Creating load balancer on ec2 instance:-

1) As we are launching ec2 instance through ansible we need to install boto3 module as

```
# pip3 install boto3
```

2) Now, write playbook for launching ec2 instance with tag name lb and other with web as

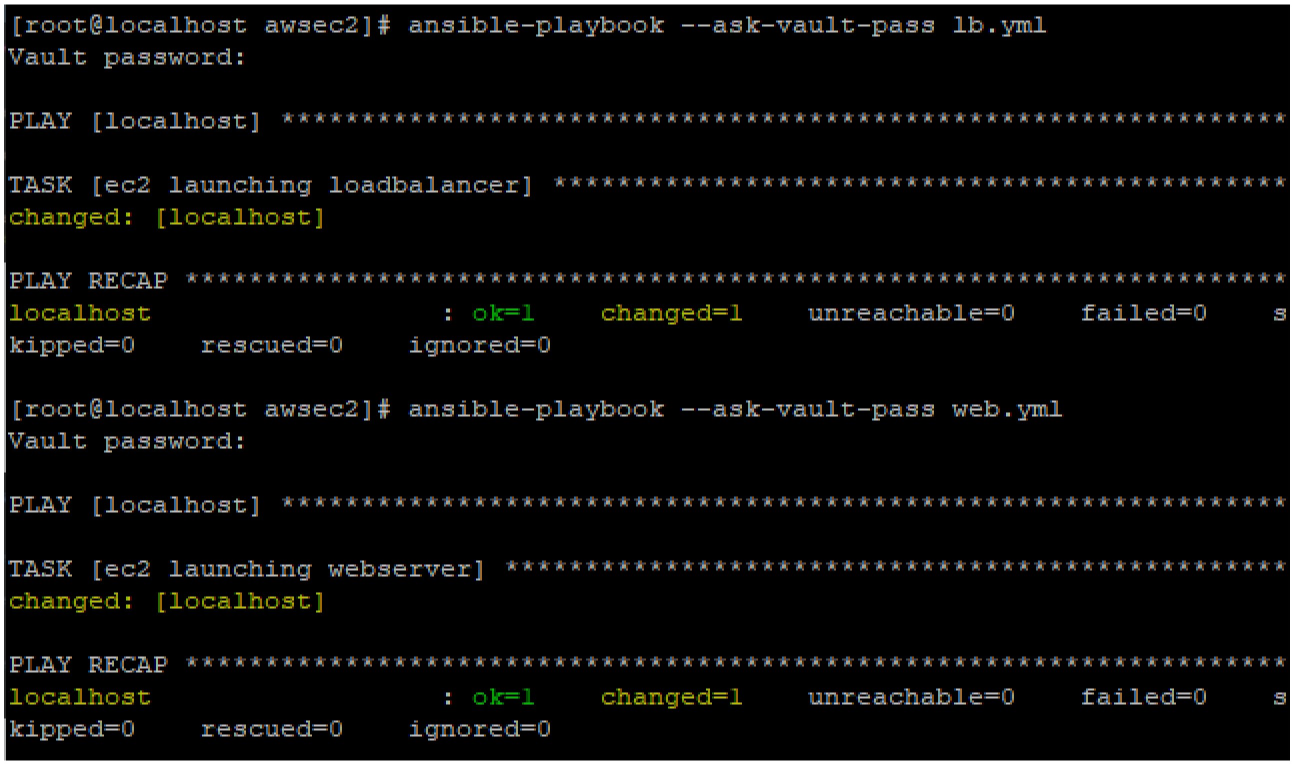


```
- hosts: localhost
  gather_facts: False
  vars_files:
      - secure.yml
  tasks:
  - name: ec2 launching loadbalancer
    ec2:
       key_name: "task3"
       instance_type: t2.micro
       image: "ami-09a7bbd08886aafdf"
       wait: yes
       count: 1
       instance_tags:
               Name: lb
       vpc_subnet_id: "subnet-0da1ded34008d07fe"
       assign_public_ip: yes
       region: "ap-south-1"
       state: present
       group_id: "sg-0ed51e95668a94488"
       aws_access_key: "{{ access }}"
       aws_secret_key: "{{ secret }}"
```

3) for secure the access key and secret key we create vault file and then put credentials in file

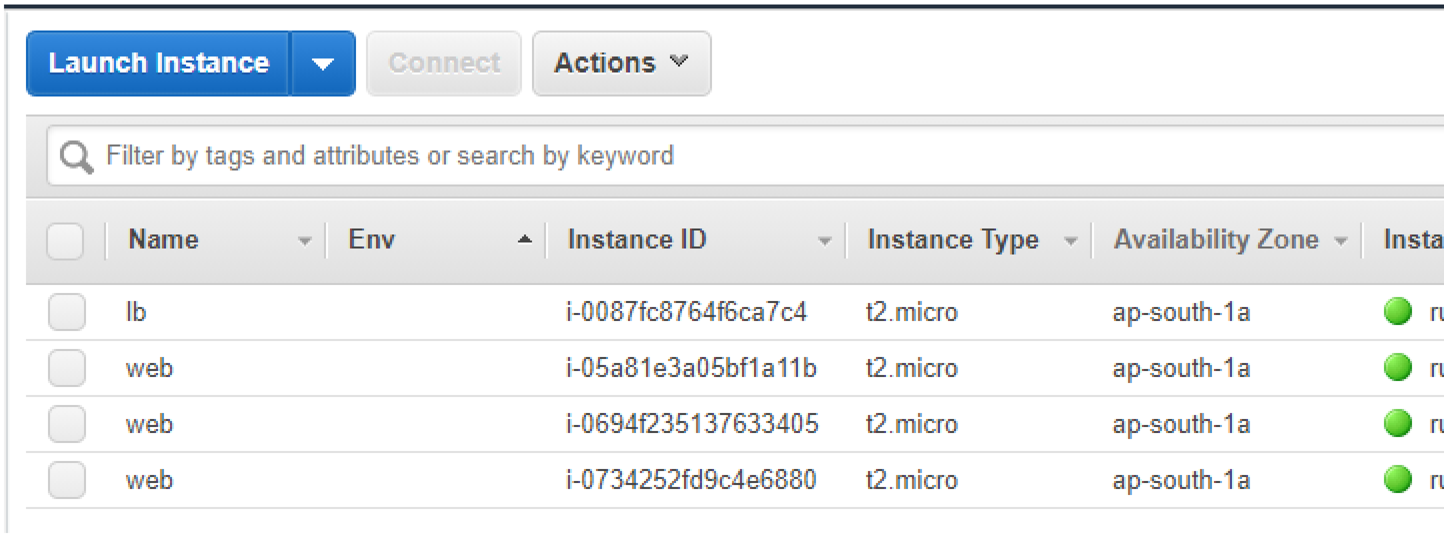
```
# ansible-vault create secure.yml
```

4) now run the both lb.yml and web.yml file to launch instances as



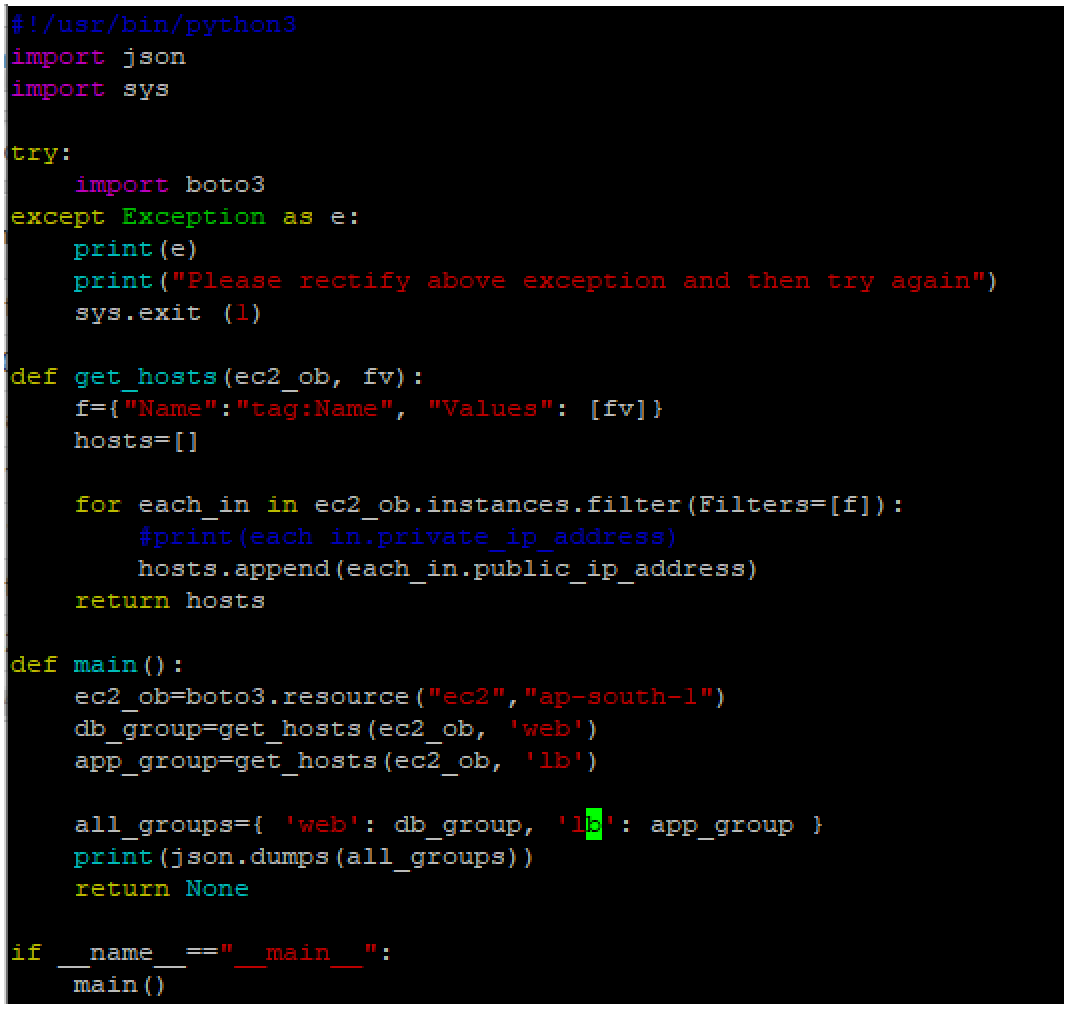
```
[root@localhost awsec2]# ansible-playbook --ask-vault-pass lb.yml
Vault password:

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

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

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

[root@localhost awsec2]# ansible-playbook --ask-vault-pass web.yml
Vault password:

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

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

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

5) After that, check on the aws instances that are successfully launched or not.



	Name	Env	Instance ID	Instance Type	Availability Zone	Insta
	lb		i-0087fc8764f6ca7c4	t2.micro	ap-south-1a	● ru
	web		i-05a81e3a05bf1a11b	t2.micro	ap-south-1a	● ru
	web		i-0694f235137633405	t2.micro	ap-south-1a	● ru
	web		i-0734252fd9c4e6880	t2.micro	ap-south-1a	● ru

6) Now we have to fetch ip of instances dynamically we use Dynamic inventory environment concept for that we need python code as



```
#!/usr/bin/python3
import json
import sys

try:
    import boto3
except Exception as e:
    print(e)
    print("Please rectify above exception and then try again")
    sys.exit (1)

def get_hosts(ec2_ob, fv):
    f={"Name":"tag:Name", "Values": [fv]}
    hosts=[]

    for each_in in ec2_ob.instances.filter(Filters=[f]):
        #print(each_in.private_ip_address)
        hosts.append(each_in.public_ip_address)
    return hosts

def main():
    ec2_ob=boto3.resource("ec2","ap-south-1")
    db_group=get_hosts(ec2_ob, 'web')
    app_group=get_hosts(ec2_ob, 'lb')

    all_groups={ 'web': db_group, 'lb': app_group }
    print(json.dumps(all_groups))
    return None

if __name__=="__main__":
    main()
```

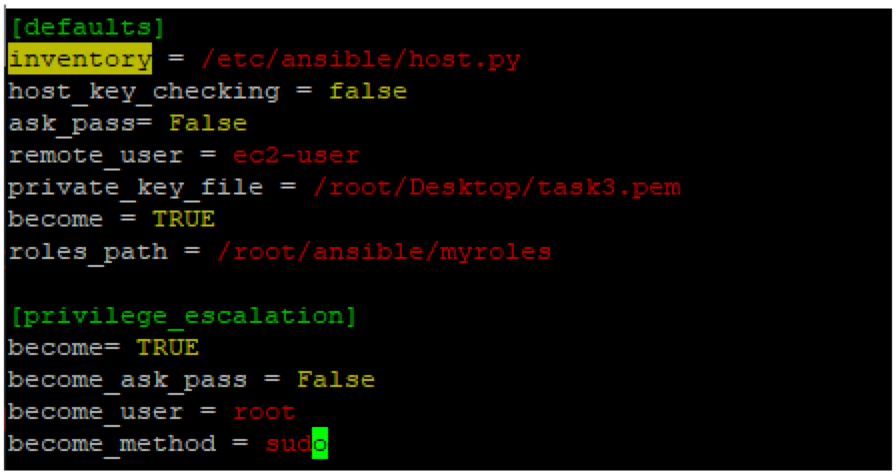
Now to make this files executable run the following command:

```
# chmod +x host.py
```

also make task3.pem file executable as

```
# chmod 700 task3.pem
```

7) Now configure inventory as

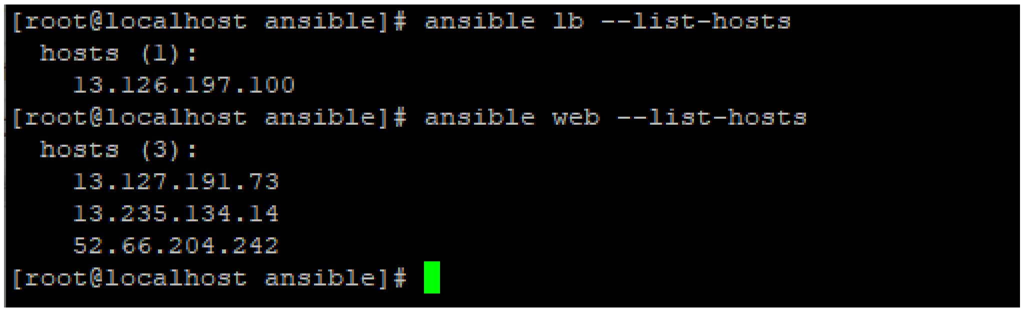


```
[defaults]
inventory = /etc/ansible/host.py
host_key_checking = false
ask_pass= False
remote_user = ec2-user
private_key_file = /root/Desktop/task3.pem
become = TRUE
roles_path = /root/ansible/myroles

[privilege_escalation]
become= TRUE
become_ask_pass = False
become_user = root
become_method = sudo
```

and export access key and secret key as:

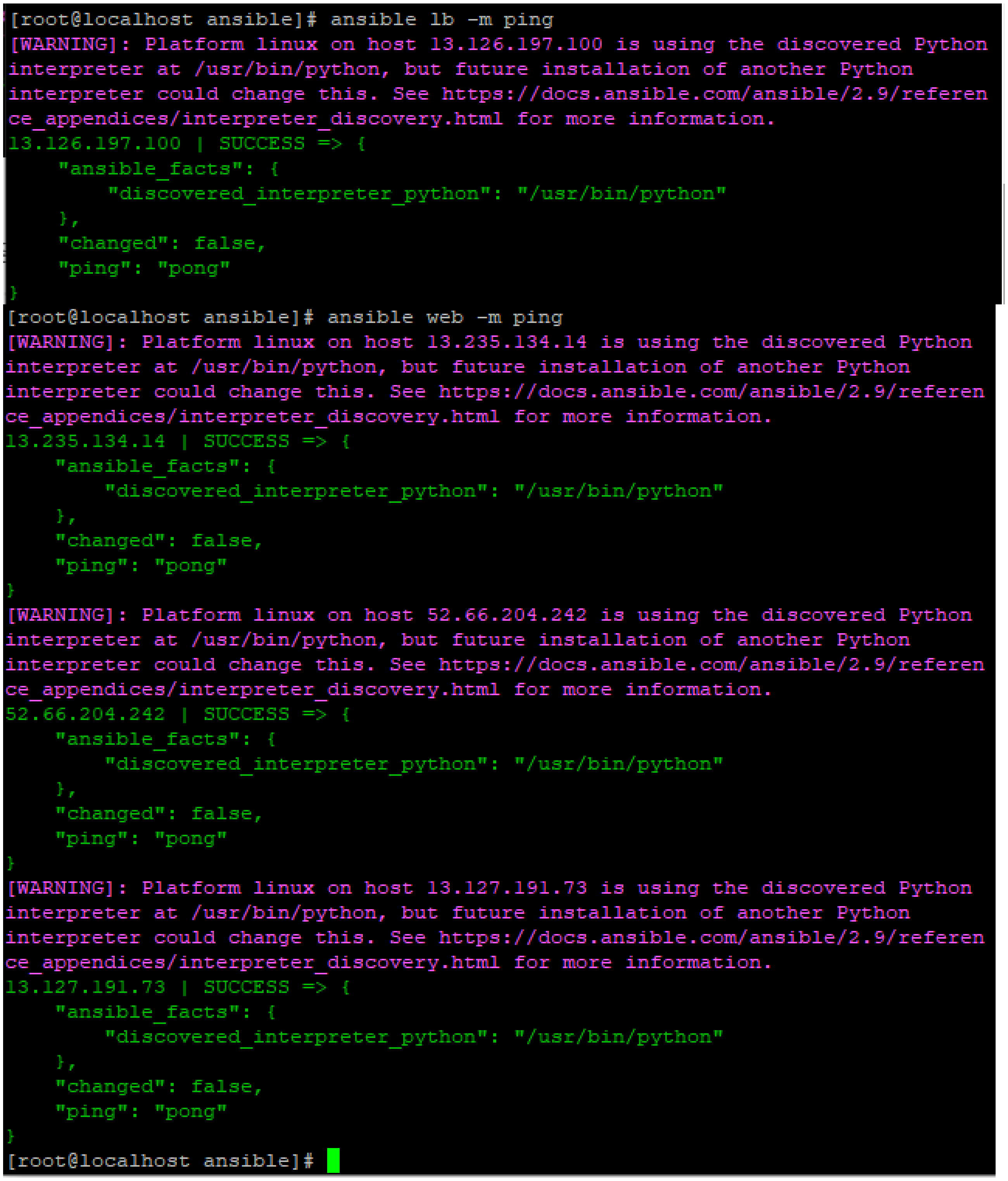
```
# export AWS_ACCESS_KEY_ID="accesskey"

# export AWS_SECRET_ACCESS_KEY="secretkey"
```

8) run the following commands, we will get ec2-instances IP. Python code fetches the IP of aws ec2-instances, also will play the role of dynamic inventory. following are the IP of web and lb instances respectively.



```
[root@localhost ansible]# ansible lb --list-hosts
  hosts (1):
    13.126.197.100
[root@localhost ansible]# ansible web --list-hosts
  hosts (3):
    13.127.191.73
    13.235.134.14
    52.66.204.242
[root@localhost ansible]#
```

9) now check ip's of loadbalancer and webserver/targetserver pinging or not



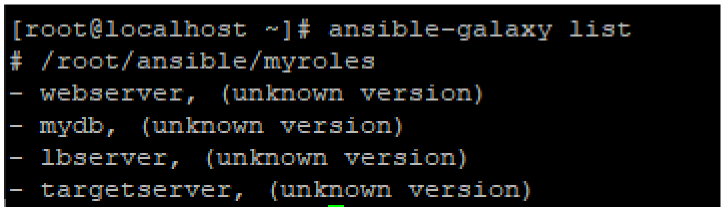
```
[root@localhost ansible]# ansible lb -m ping
[WARNING]: Platform linux on host 13.126.197.100 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
13.126.197.100 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[root@localhost ansible]# ansible web -m ping
[WARNING]: Platform linux on host 13.235.134.14 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
13.235.134.14 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 52.66.204.242 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
52.66.204.242 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 13.127.191.73 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
13.127.191.73 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[root@localhost ansible]#
```

10) Now we have to create roles for the configuration of haproxy into the Load-Balancer instance and httpd into the Web-Servers instances.

Use the following command for creating roles load balancer and webserver:

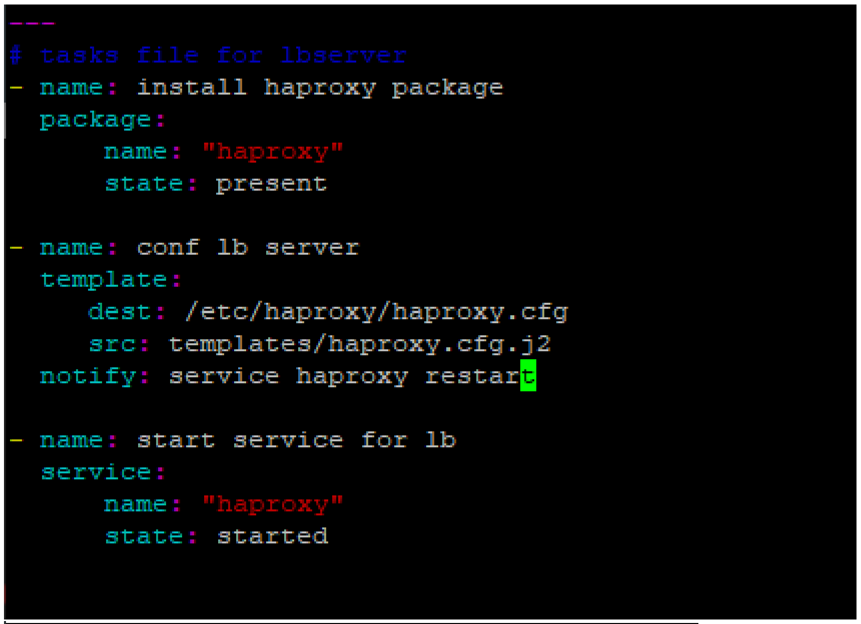
# ansible-galaxy init <rolename>

Now check Roles List as:

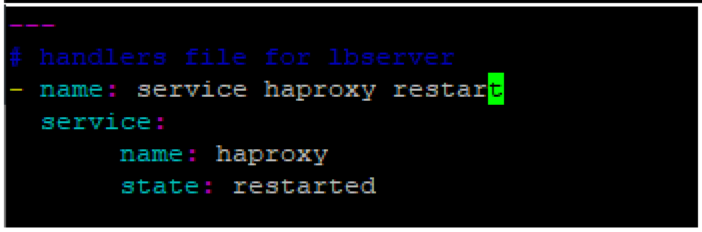


```
[root@localhost ~]# ansible-galaxy list
# /root/ansible/myroles
- webserver, (unknown version)
- mydb, (unknown version)
- lbserver, (unknown version)
- targetserver, (unknown version)
```

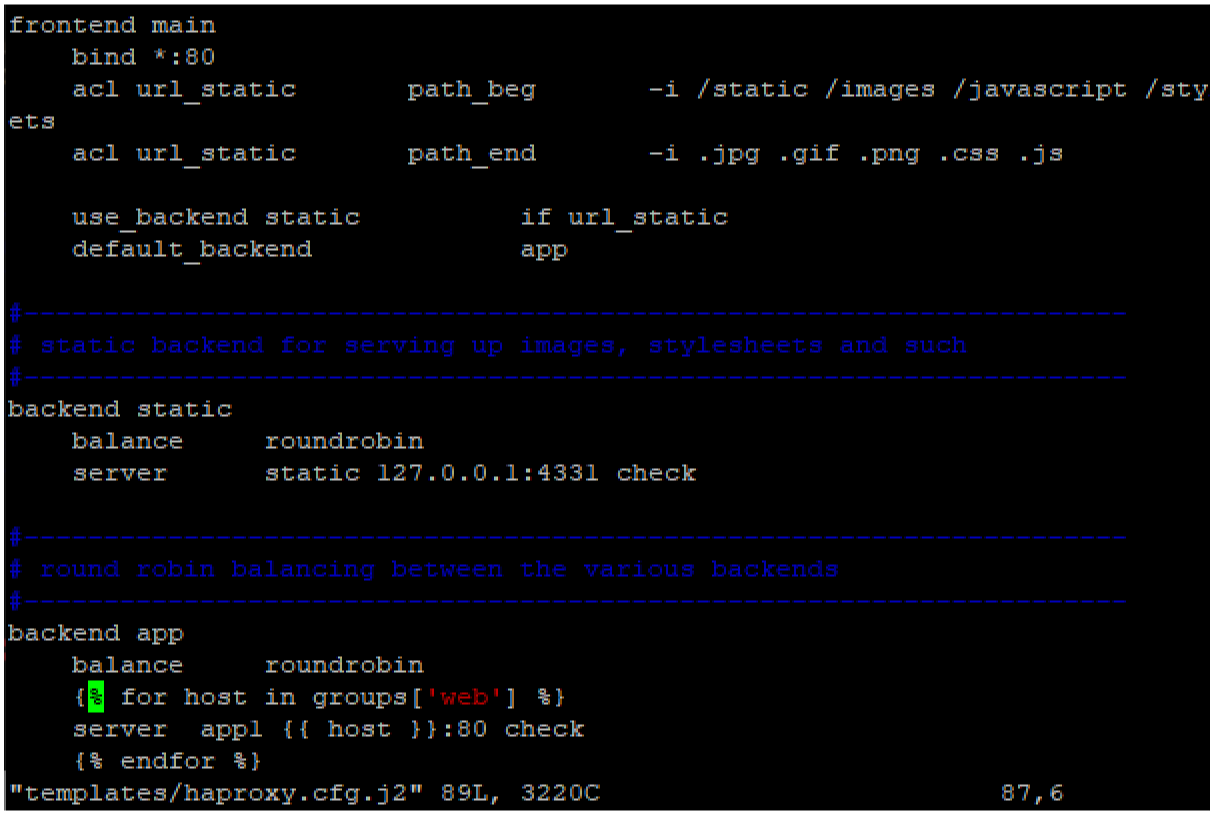
11) Configuring Haproxy service inside the load balancer(lb) role:



```
---
# tasks file for lbserver
- name: install haproxy package
  package:
      name: "haproxy"
      state: present

- name: conf lb server
  template:
     dest: /etc/haproxy/haproxy.cfg
     src: templates/haproxy.cfg.j2
  notify: service haproxy restart

- name: start service for lb
  service:
      name: "haproxy"
      state: started
```

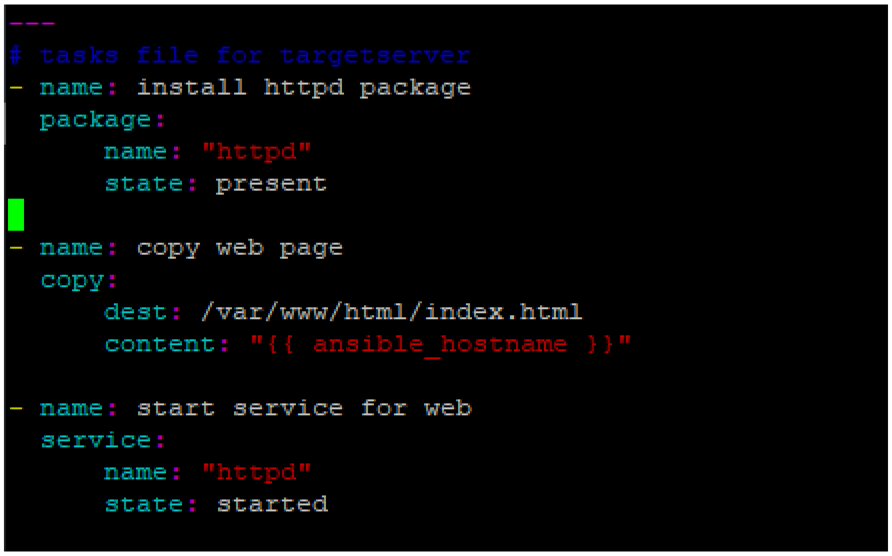


```
---
# handlers file for lbserver
- name: service haproxy restart
  service:
       name: haproxy
       state: restarted
```



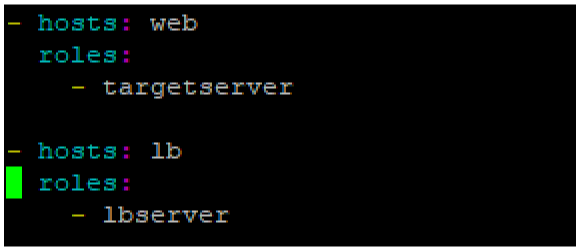
```
frontend main
    bind *:80
    acl url_static       path_beg       -i /static /images /javascript /sty
ets
    acl url_static       path_end       -i .jpg .gif .png .css .js

    use_backend static          if url_static
    default_backend             app

#---------------------------------------------------------------------
# static backend for serving up images, stylesheets and such
#---------------------------------------------------------------------
backend static
    balance     roundrobin
    server      static 127.0.0.1:4331 check

#---------------------------------------------------------------------
# round robin balancing between the various backends
#---------------------------------------------------------------------
backend app
    balance     roundrobin
    {% for host in groups['web'] %}
    server  app1 {{ host }}:80 check
    {% endfor %}
"templates/haproxy.cfg.j2" 89L, 3220C                          87,6
```

12) Configuring httpd server inside the webserver role:



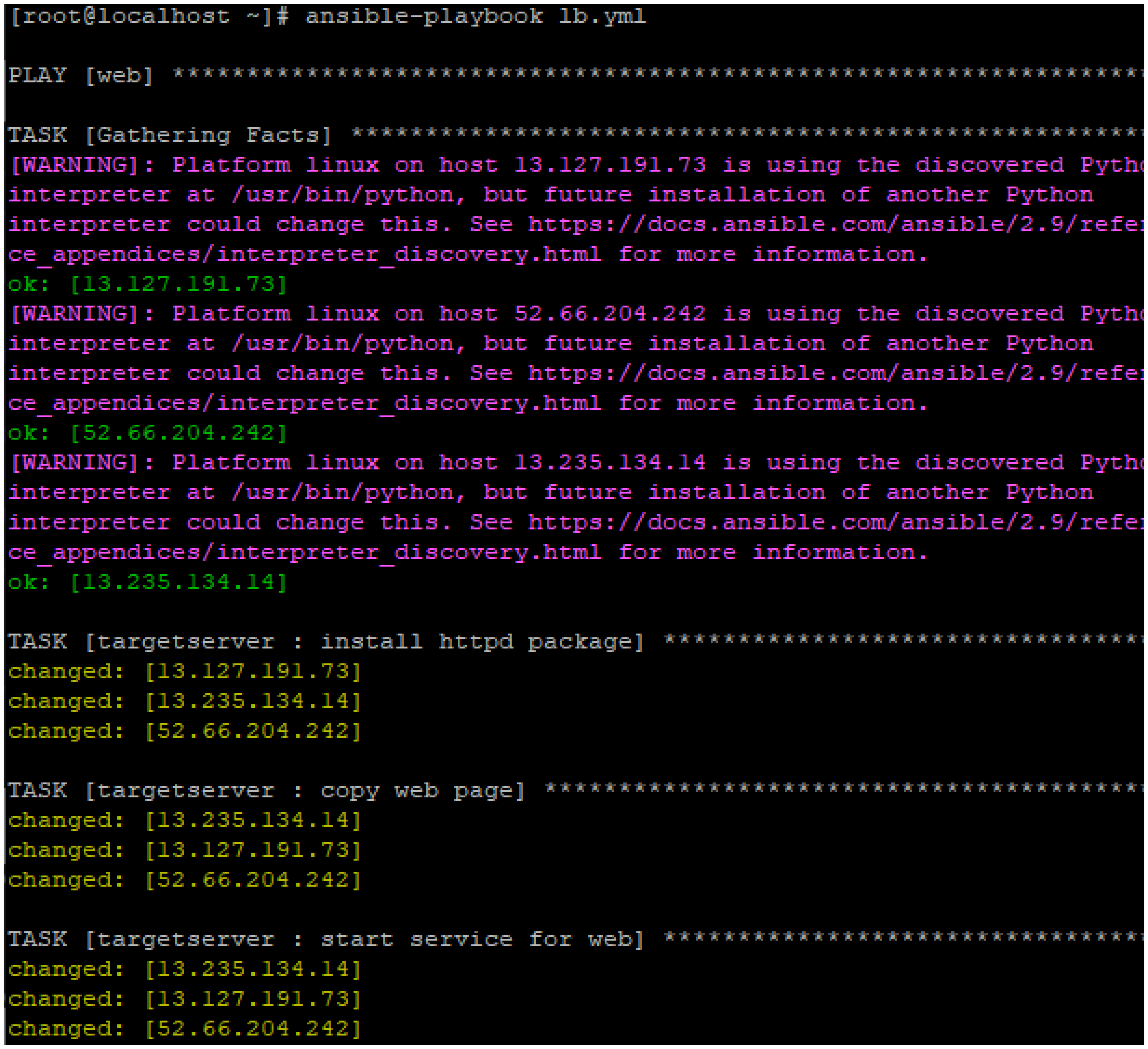
```
---
# tasks file for targetserver
- name: install httpd package
  package:
      name: "httpd"
      state: present

- name: copy web page
  copy:
      dest: /var/www/html/index.html
      content: "{{ ansible_hostname }}"

- name: start service for web
  service:
      name: "httpd"
      state: started
```

13) setup of roles.yaml:

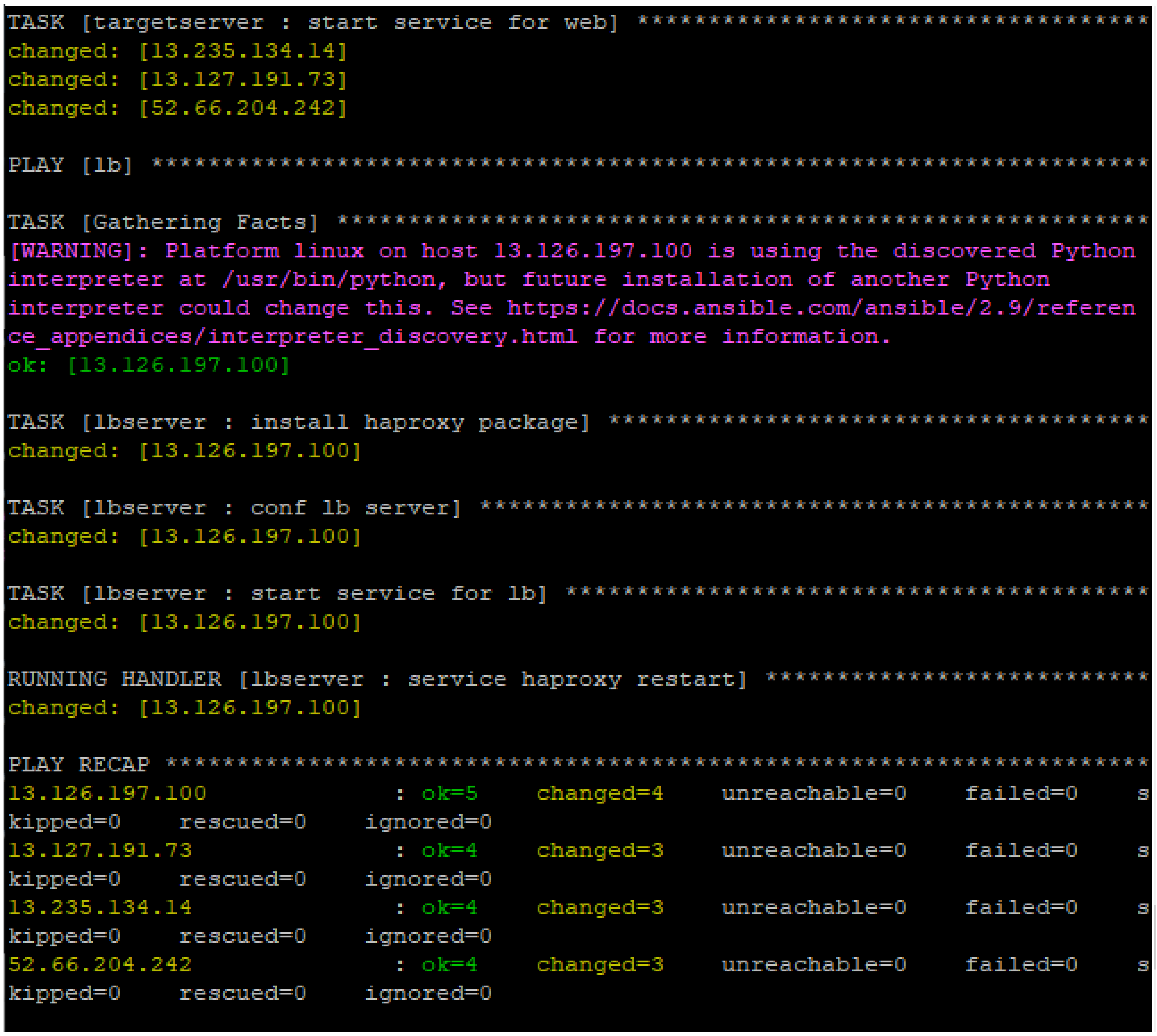


```
- hosts: web
  roles:
    - targetserver

- hosts: lb
  roles:
    - lbserver
```

14) Running ansible-playbook command:

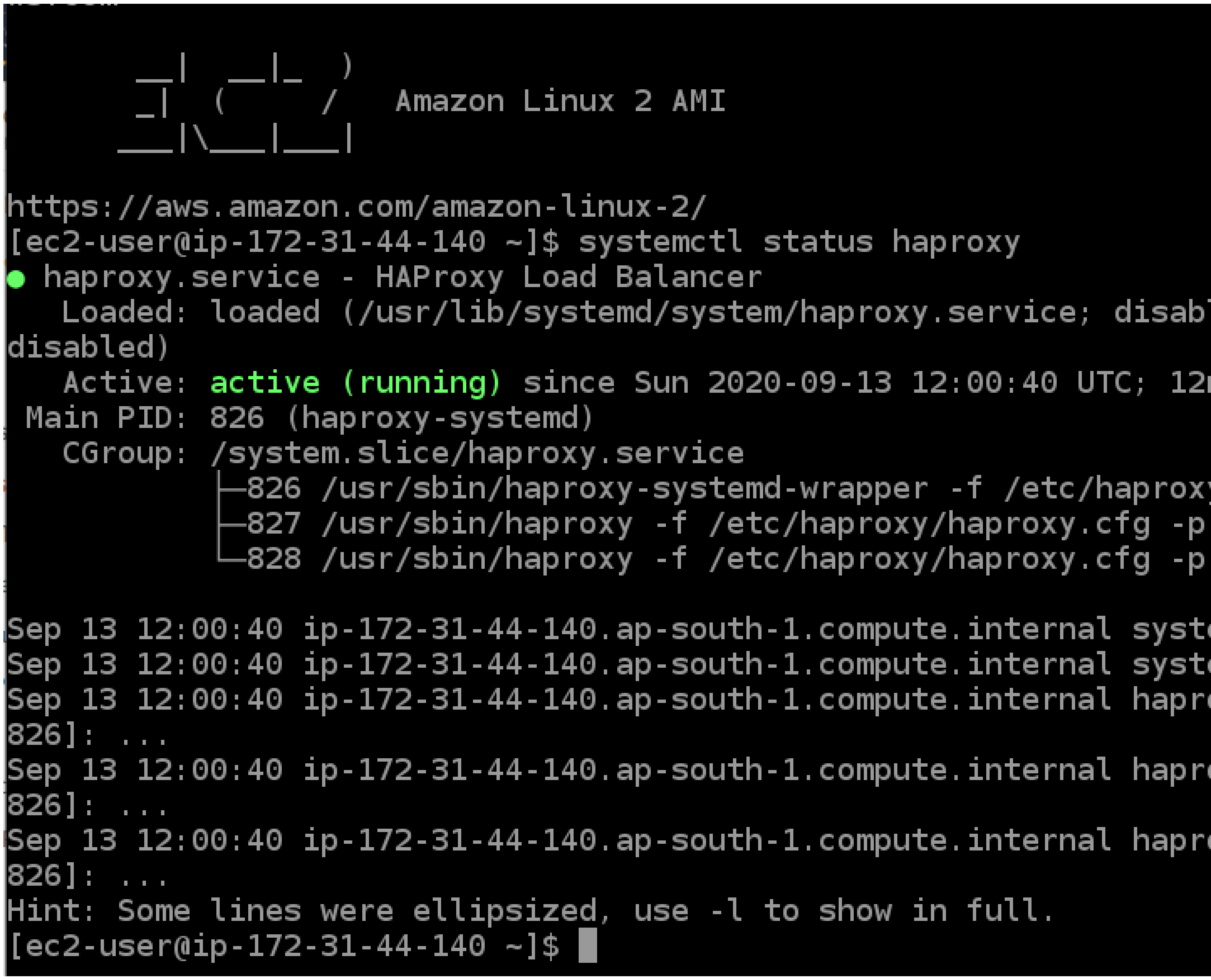


```
[root@localhost ~]# ansible-playbook lb.yml

PLAY [web] ********************************************************************

TASK [Gathering Facts] ********************************************************
[WARNING]: Platform linux on host 13.127.191.73 is using the discovered Pytho
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/refe
ce_appendices/interpreter_discovery.html for more information.
ok: [13.127.191.73]
[WARNING]: Platform linux on host 52.66.204.242 is using the discovered Pytho
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/refe
ce_appendices/interpreter_discovery.html for more information.
ok: [52.66.204.242]
[WARNING]: Platform linux on host 13.235.134.14 is using the discovered Pytho
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/refe
ce_appendices/interpreter_discovery.html for more information.
ok: [13.235.134.14]

TASK [targetserver : install httpd package] ***********************************
changed: [13.127.191.73]
changed: [13.235.134.14]
changed: [52.66.204.242]

TASK [targetserver : copy web page] *******************************************
changed: [13.235.134.14]
changed: [13.127.191.73]
changed: [52.66.204.242]

TASK [targetserver : start service for web] ***********************************
changed: [13.235.134.14]
changed: [13.127.191.73]
changed: [52.66.204.242]
```



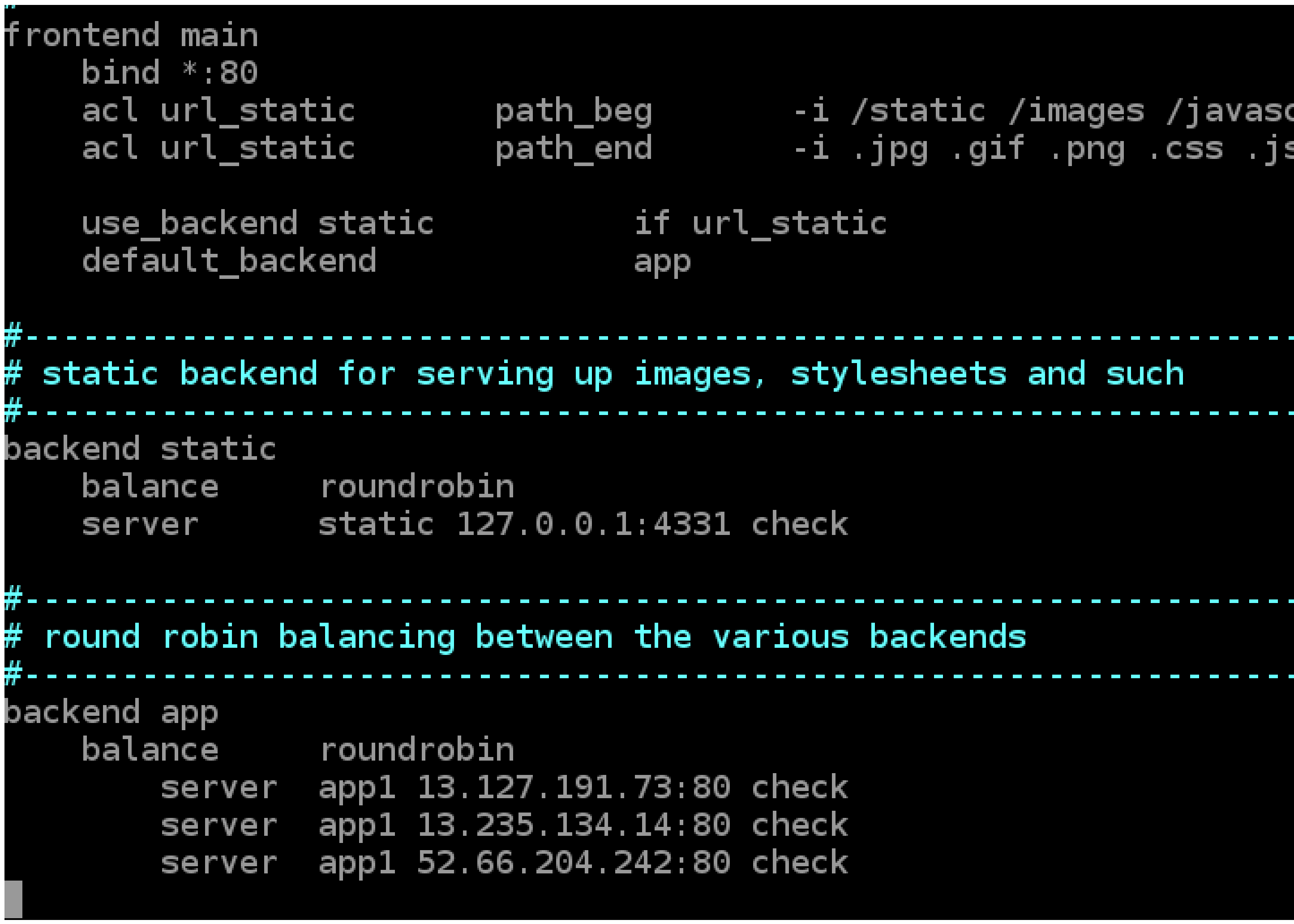
```
TASK [targetserver : start service for web] ************************************
changed: [13.235.134.14]
changed: [13.127.191.73]
changed: [52.66.204.242]

PLAY [lb] **********************************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host 13.126.197.100 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [13.126.197.100]

TASK [lbserver : install haproxy package] **************************************
changed: [13.126.197.100]

TASK [lbserver : conf lb server] ***********************************************
changed: [13.126.197.100]

TASK [lbserver : start service for lb] *****************************************
changed: [13.126.197.100]

RUNNING HANDLER [lbserver : service haproxy restart] ***************************
changed: [13.126.197.100]

PLAY RECAP *********************************************************************
13.126.197.100             : ok=5    changed=4    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
13.127.191.73              : ok=4    changed=3    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
13.235.134.14              : ok=4    changed=3    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
52.66.204.242              : ok=4    changed=3    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

15) Now check that haproxy service is started or not in the Load-Balancer instance:



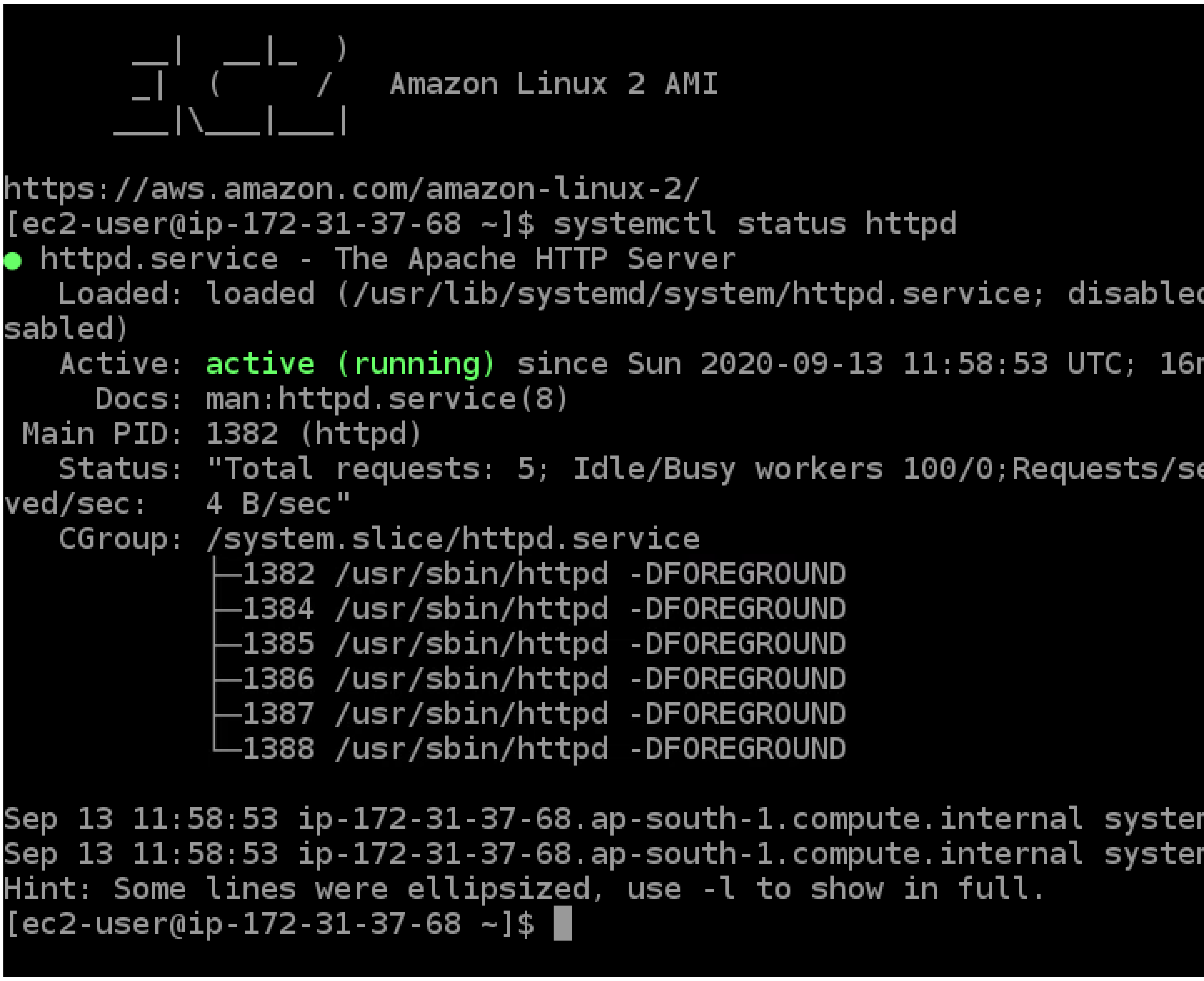
```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-44-140 ~]$ systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/usr/lib/systemd/system/haproxy.service; disab
disabled)
   Active: active (running) since Sun 2020-09-13 12:00:40 UTC; 12
 Main PID: 826 (haproxy-systemd)
   CGroup: /system.slice/haproxy.service
           ├─826 /usr/sbin/haproxy-systemd-wrapper -f /etc/haprox
           ├─827 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p
           └─828 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p

Sep 13 12:00:40 ip-172-31-44-140.ap-south-1.compute.internal syst
Sep 13 12:00:40 ip-172-31-44-140.ap-south-1.compute.internal syst
Sep 13 12:00:40 ip-172-31-44-140.ap-south-1.compute.internal hapr
826]: ...
Sep 13 12:00:40 ip-172-31-44-140.ap-south-1.compute.internal hapr
826]: ...
Sep 13 12:00:40 ip-172-31-44-140.ap-south-1.compute.internal hapr
826]: ...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-44-140 ~]$ 
```

haproxy.cfg file



```
frontend main
    bind *:80
    acl url_static       path_beg       -i /static /images /javasc
    acl url_static       path_end       -i .jpg .gif .png .css .js

    use_backend static          if url_static
    default_backend             app

#---------------------------------------------------------------------
# static backend for serving up images, stylesheets and such
#---------------------------------------------------------------------
backend static
    balance     roundrobin
    server      static 127.0.0.1:4331 check

#---------------------------------------------------------------------
# round robin balancing between the various backends
#---------------------------------------------------------------------
backend app
    balance     roundrobin
        server  app1 13.127.191.73:80 check
        server  app1 13.235.134.14:80 check
        server  app1 52.66.204.242:80 check
```

16) Now check that httpd service is started or not in the one of the Web-Balancer instance:



```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-37-68 ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disablec
sabled)
   Active: active (running) since Sun 2020-09-13 11:58:53 UTC; 16r
     Docs: man:httpd.service(8)
 Main PID: 1382 (httpd)
   Status: "Total requests: 5; Idle/Busy workers 100/0;Requests/se
ved/sec:   4 B/sec"
   CGroup: /system.slice/httpd.service
           ├─1382 /usr/sbin/httpd -DFOREGROUND
           ├─1384 /usr/sbin/httpd -DFOREGROUND
           ├─1385 /usr/sbin/httpd -DFOREGROUND
           ├─1386 /usr/sbin/httpd -DFOREGROUND
           ├─1387 /usr/sbin/httpd -DFOREGROUND
           └─1388 /usr/sbin/httpd -DFOREGROUND

Sep 13 11:58:53 ip-172-31-37-68.ap-south-1.compute.internal systen
Sep 13 11:58:53 ip-172-31-37-68.ap-south-1.compute.internal systen
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-37-68 ~]$ 
```

Thanks for reading!!