TASK 4:

Task Description 👨🏻💻

- Launch an AWS instances with the help of ansible
- Retrieve the public IP which is allocated to the launched instance.
- With the help of the retrieved Public IP configure the load balancing setup using haproxy

CREATING A LOADBALANCER USING ANSIBLE

Step 1: Prerequisites

- To install boto3 module
- Setting up the dynamic inventory using python code
- chmod +x filename.yml
- Creating multiple roles loadbalancer and session14

```
[root@controllernode mypy]# pip3 list | grep boto
DEPRECATION: The default format will switch to columns in the future. You can use --format=(lega
cy|columns) (or define a format=(legacy|columns) in your pip.conf under the [list] section) to d
isable this warning.
boto (2.49.0)
boto3 (1.14.39)
botocore (1.17.39)
```

Step 2: Launching an AWS Instance using ansible

```
ansible@controllernode:~
[ansible@controllernode ~]$ ansible -m ping all
[WARNING]: Platform linux on host ansible-os is using the discovered Python interpreter at /usr/bin/python, but
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ansible-os | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host ansible-os2 is using the discovered Python interpreter at /usr/bin/python, but
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ansible-os2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host ansible-os1 is using the discovered Python interpreter at /usr/bin/python, but
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ansible-os1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```


```
[root@controllernode aws_ansible]# cat secure.yml
$ANSIBLE_VAULT;1.1;AES256
63376365393439623765656435396237386635346532383336383864323634353635303538346637
39623763336264623539343730396537343863646530631 0a3030663265353338376534316164
63363537353036646232316264353162333365373664343333323364333762616664636639383763
30393735363063563 0a33626464383937643566383462363539633261633739373561636464613 9
61396561393630326566333534323761643264393336303066373862626634326534626230663833
63636365386661626235616461393233366633656438353562313364643164656135313333343332
30316365356539326262303736333635343539326437303361666336636565436313836306333662
31376630383135646533303339393861356364656434633864616636323136383832393634633263
3363
```

```
[root@controllernode aws_ansible]# cat ec2.yml
- hosts: localhost
  vars_files:
          - secure.yml
  tasks:
#   - prog -> aws_client -> ec2_user
  - name: launching os using ansible
    ec2:
        region: "ap-south-1"
        key_name: "mykey1111.pem"
        instance_type: "t2.micro"
        image: "ami-0ebc1ac48dfd14136"
        wait: yes
        count: 4
        vpc_subnet_id: "subnet-d7ead0bf"
        assign_public_ip: yes
        group_id: "sg-022d22ccade57d756"
        state: present
        aws_access_key: "{{ access_key }}"
        aws_secret_key: "{{ secret_key }}"
```

- Creating an task4.yml file
- Storing access_key and secret_key in a separate vault file
- Using ansible-playbook --ask-vault-pass task4.yml

```
[root@controllernode aws_ansible]# ansible-vault encrypt secure.yml
New Vault password:
Confirm New Vault password:
Encryption successful
```

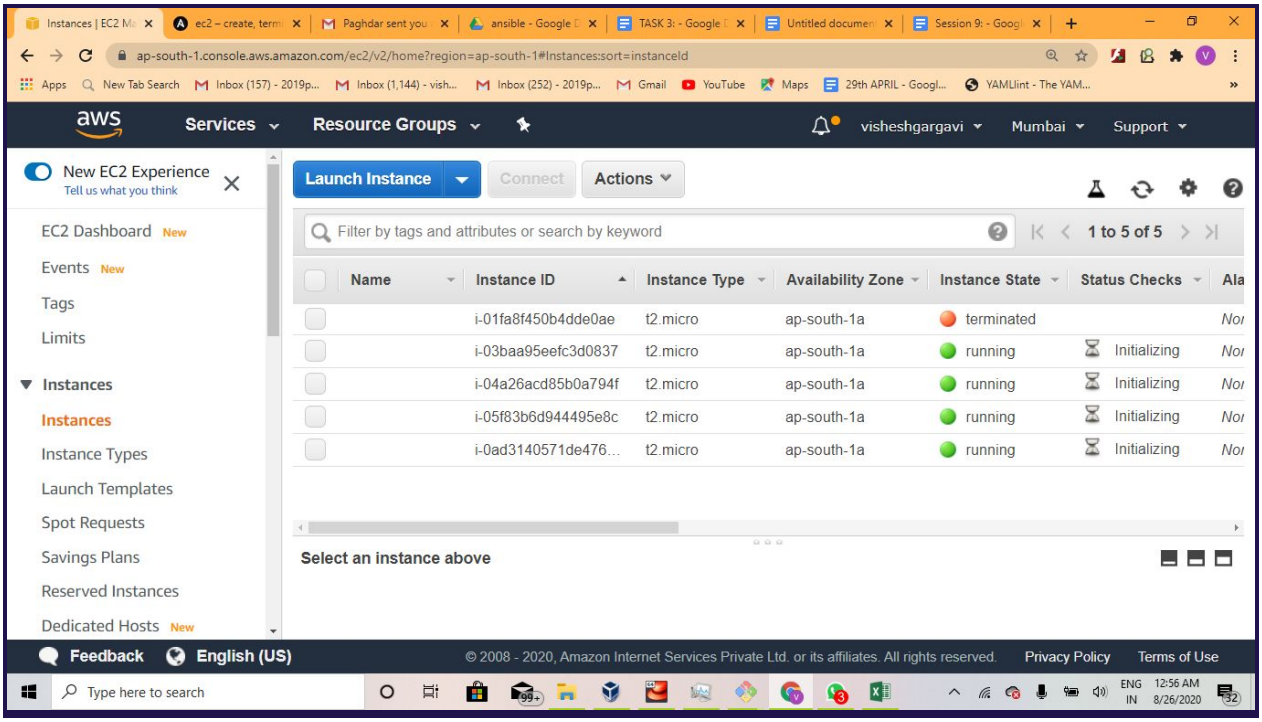
Step 3: Setting up the Dynamic inventory environment

```
root@controllernode:/mypy
ntrib/inventory/ec2.py
--2020-08-20 22:15:29--  https://raw.githubusercontent.com/ansible/ansible/stable-2.9/contrib/in
ventory/ec2.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 199.232.20.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|199.232.20.133|:443... conne
cted.
HTTP request sent, awaiting response... 200 OK
Length: 73130 (71K) [text/plain]
Saving to: 'ec2.py'

ec2.py                  100%[=============================>]  71.42K   422KB/s    in 0.2s

2020-08-20 22:15:30 (422 KB/s) - 'ec2.py' saved [73130/73130]
```

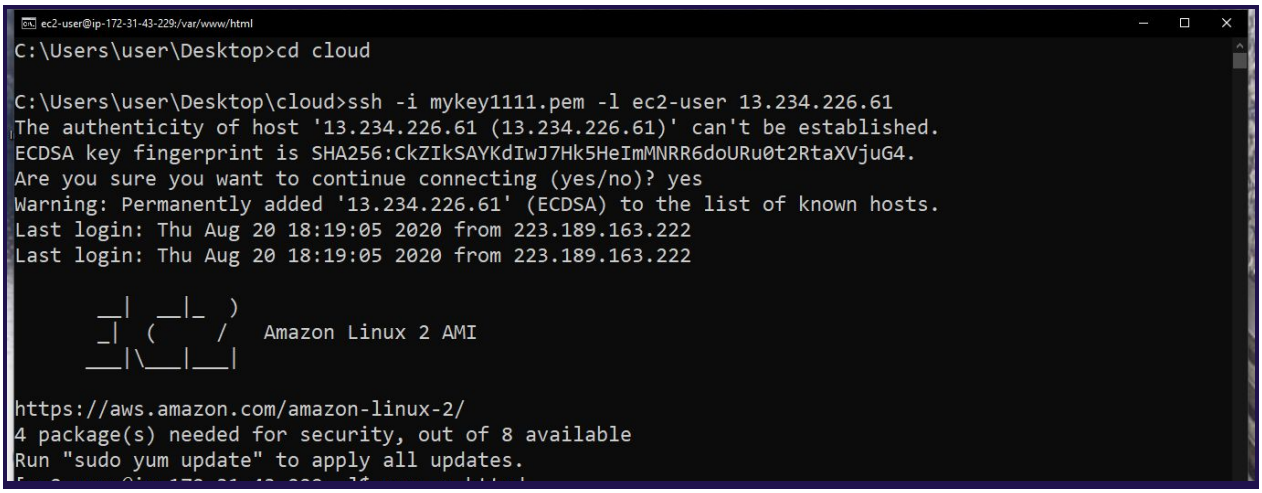
```
[root@controllernode mypy]# wget https://raw.githubusercontent.com/ansible/ansible/stable-2.9/co
ntrib/inventory/ec2.ini
--2020-08-20 22:35:52--  https://raw.githubusercontent.com/ansible/ansible/stable-2.9/contrib/in
ventory/ec2.ini
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 199.232.20.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|199.232.20.133|:443... conne
cted.
HTTP request sent, awaiting response... 200 OK
Length: 9529 (9.3K) [text/plain]
Saving to: 'ec2.ini'

ec2.ini                 100%[=============================>]   9.31K  --.-KB/s    in 0.001s

2020-08-20 22:35:52 (6.31 MB/s) - 'ec2.ini' saved [9529/9529]

[root@controllernode mypy]# ls
ec2.ini  ec2.py
[root@controllernode mypy]# chmod +x ec2.ini
[root@controllernode mypy]# ls
ec2.ini  ec2.py
```

```
[root@controllernode mypy]# export AWS_REGION='ap-south-1'
[root@controllernode mypy]# python3 ec2.py --list
```

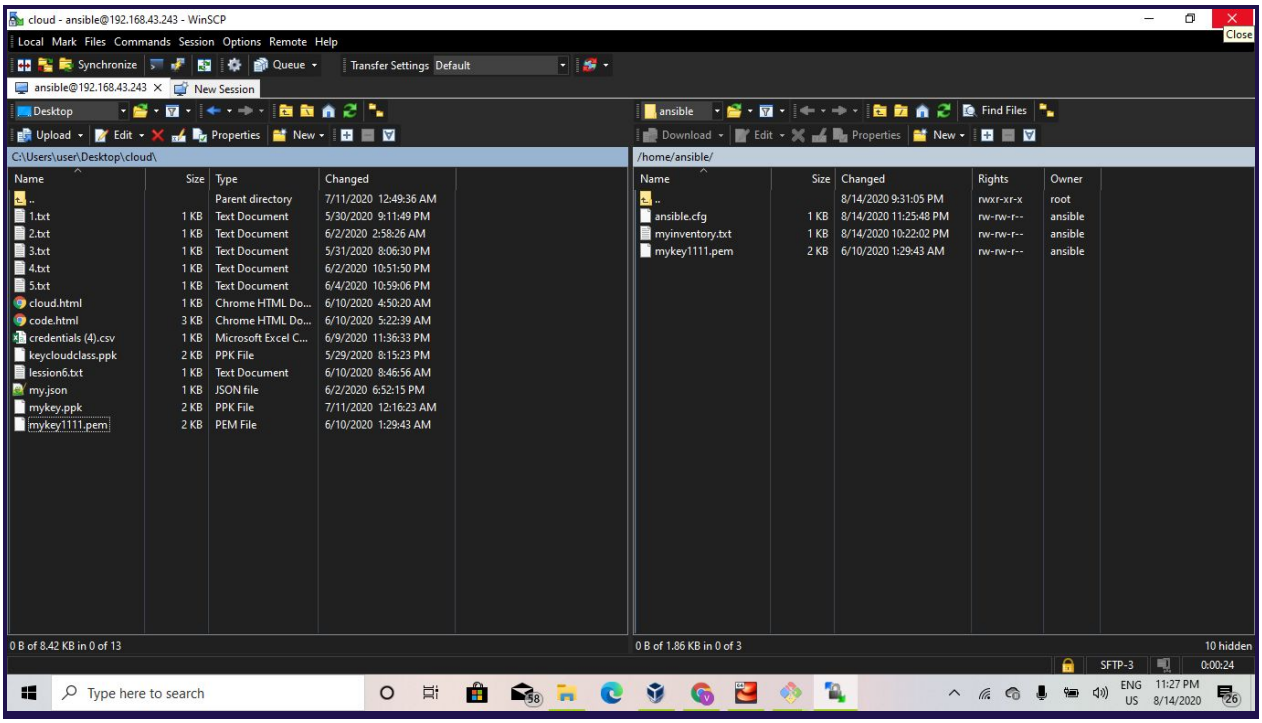
```
[root@controllernode aws_ansible]# ansible all --list-hosts
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see
details
  hosts (4):
    13.234.240.55
    13.127.142.159
    13.234.78.166
    15.206.94.86
```



Step 4: Connect Instance with SSH



Step 5: Copying key-pair using win-scp



Step 6: Configuring host and config file

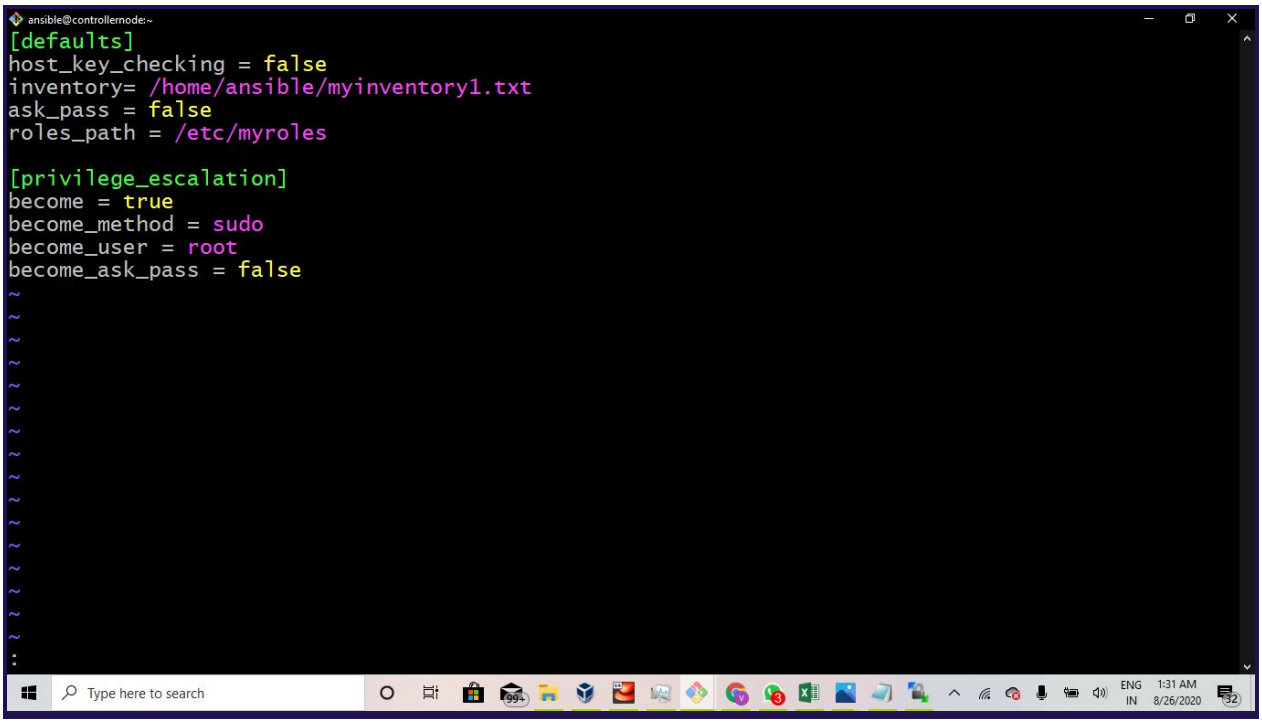
```
[ansible@controllernode ~]$ cat myinventory1.txt
[lb]
13.234.240.55 ansible_user=ec2-user ansible_ssh_private_key_file=/home/ansible/Desktop/mykey1111
.pem

[webserver]
13.127.142.159 ansible_user=ec2-user ansible_ssh_private_key_file=/home/ansible/Desktop/mykey111
1.pem
13.234.78.166 ansible_user=ec2-user ansible_ssh_private_key_file=/home/ansible/Desktop/mykey1111
.pem
15.206.94.86 ansible_user=ec2-user ansible_ssh_private_key_file=/home/ansible/Desktop/mykey1111.
pem
```

```
[ansible@controllernode ~]$
[ansible@controllernode ~]$ cat task4.yml
- hosts: webserver
  roles:
  - role: session14

- hosts: lb
  roles:
  - role: loadbalancersetup
```

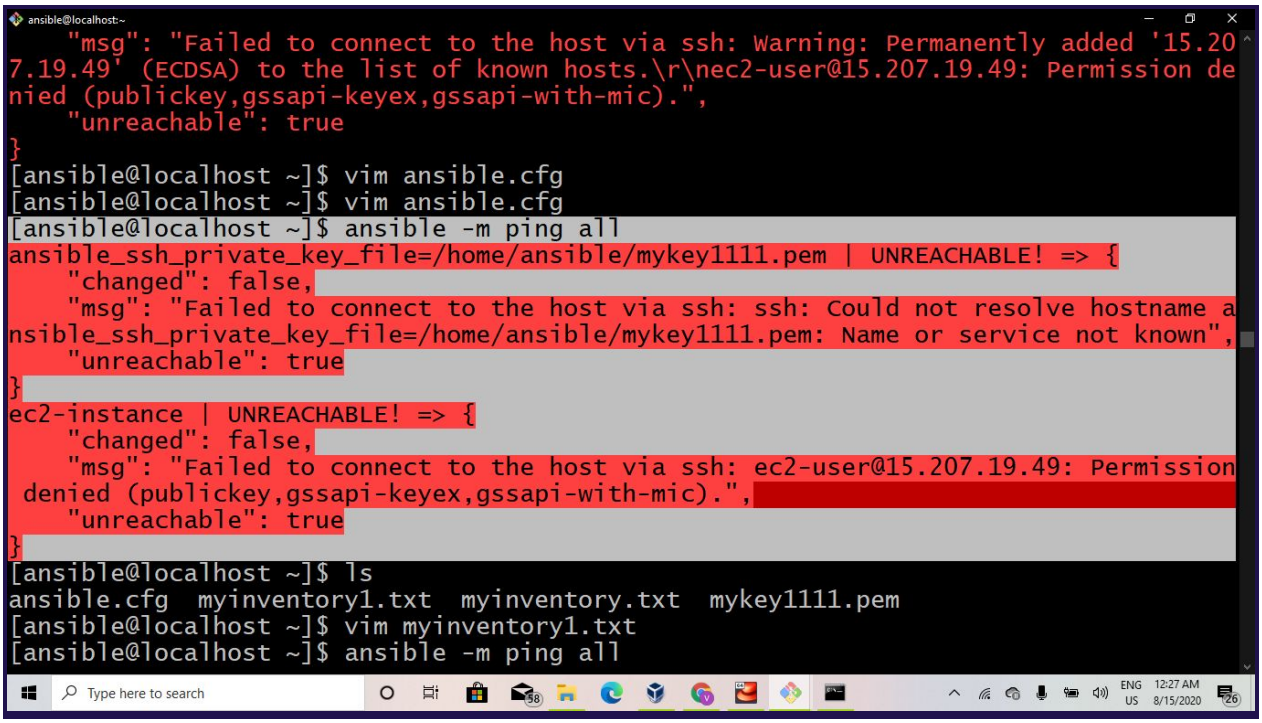
The information you will need is:

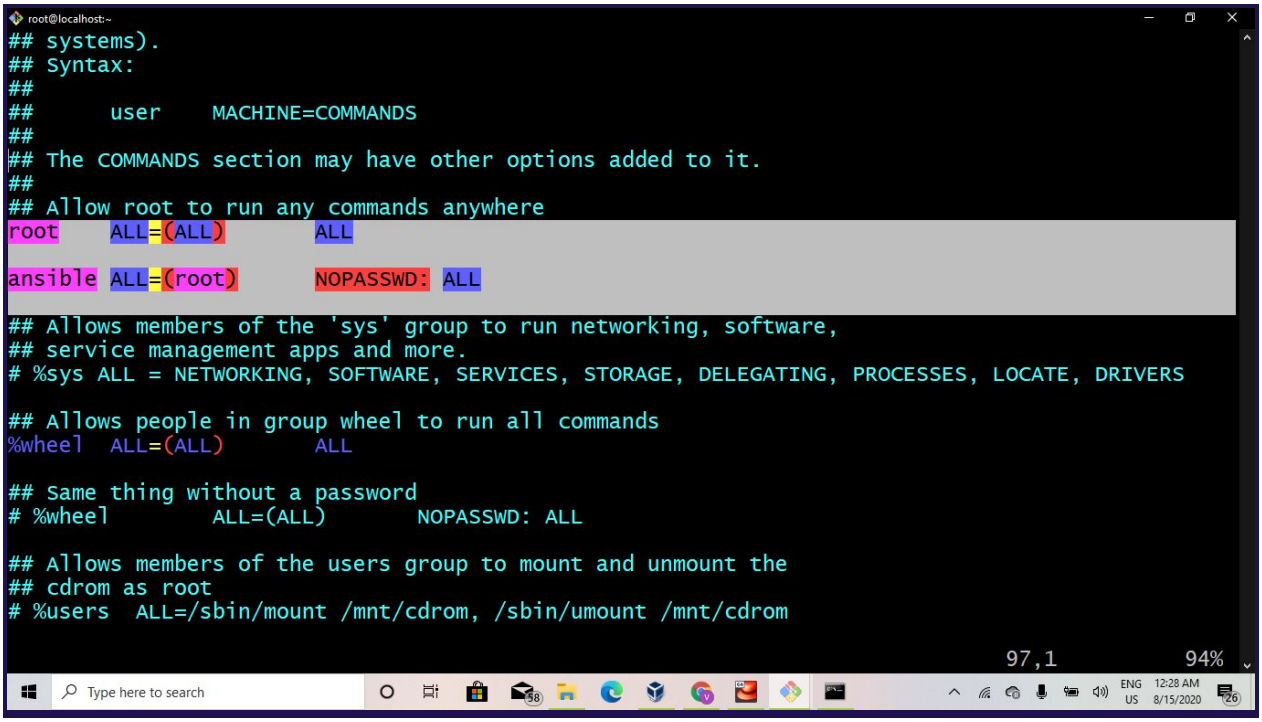
- **Name for the instance**

- IP Address of your AWS instance(public-ip)
- The user present on your AWS instance (ec2-user)
- Location to your private key (.pem) file



Step 7: Configuring the setting and making public-key to private

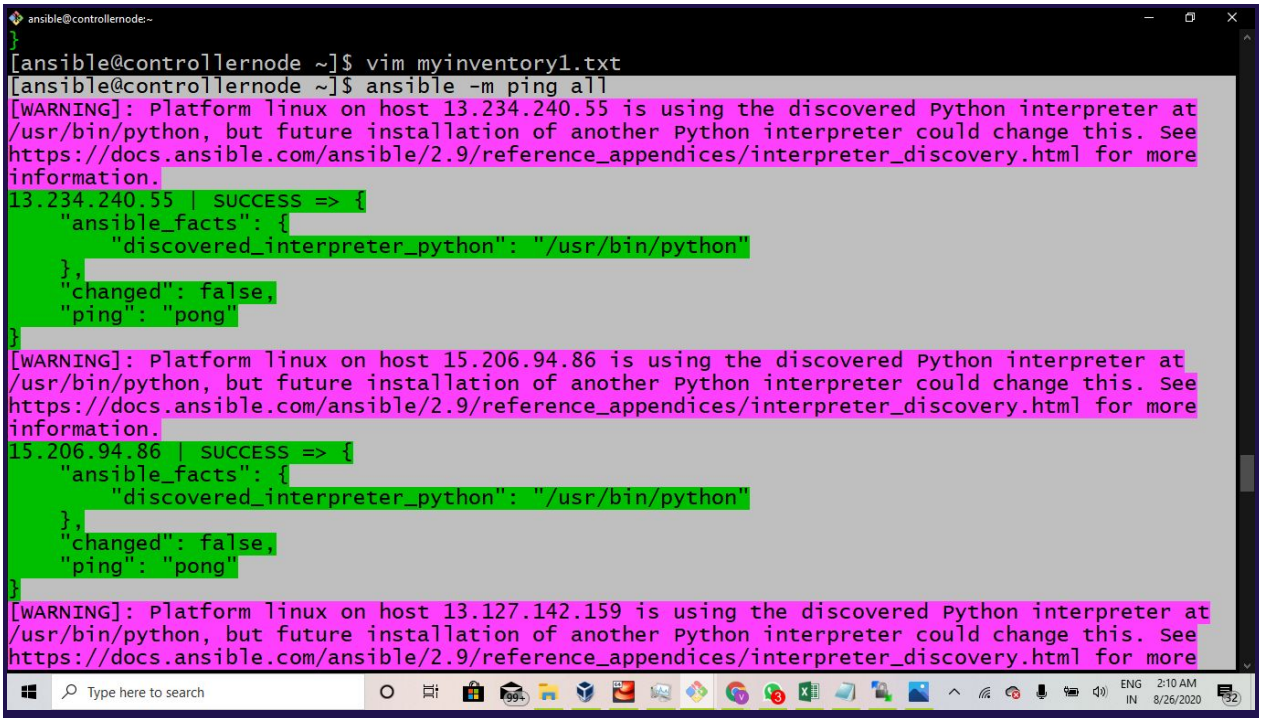





```
sudo chmod 600 /home/ansible/mykey1111.pem
```

```
sudo chmod 755 ~/.ssh
```

Step 8: Run Ansible Ping Module



Step 9: Creating roles load balancer and session14

```
[root@controllernode myroles]# ansible-galaxy list
# /etc/myroles
- webserver, (unknown version)
[root@controllernode myroles]# ansible-galaxy init session14
- Role session14 was created successfully
[root@controllernode myroles]# ansible-galaxy list
# /etc/myroles
- webserver, (unknown version)
- session14, (unknown version)
```


```
[root@controllernode myroles]# ansible-galaxy list
# /etc/myroles
- webserver, (unknown version)
[root@controllernode myroles]# ansible-galaxy init session14
- Role session14 was created successfully
[root@controllernode myroles]# ansible-galaxy list
# /etc/myroles
- webserver, (unknown version)
- session14, (unknown version)
```

Step 10: Configuring Haproxy service inside the load balancer role

root@controllernode:/etc/myroles/loadbalancersetup/tasks

```
---
# tasks file for loadbalancersetup

- name: install haproxy software
  package:
     name: "haproxy"
     state: present

- name: copy my conf file of lb
  template:
     src: "haproxy.cfg"
     dest: /etc/haproxy/haproxy.cfg
  notify: lbrestart

- name: start haproxy service
  service:
          name: "haproxy"
          state: started

~
```

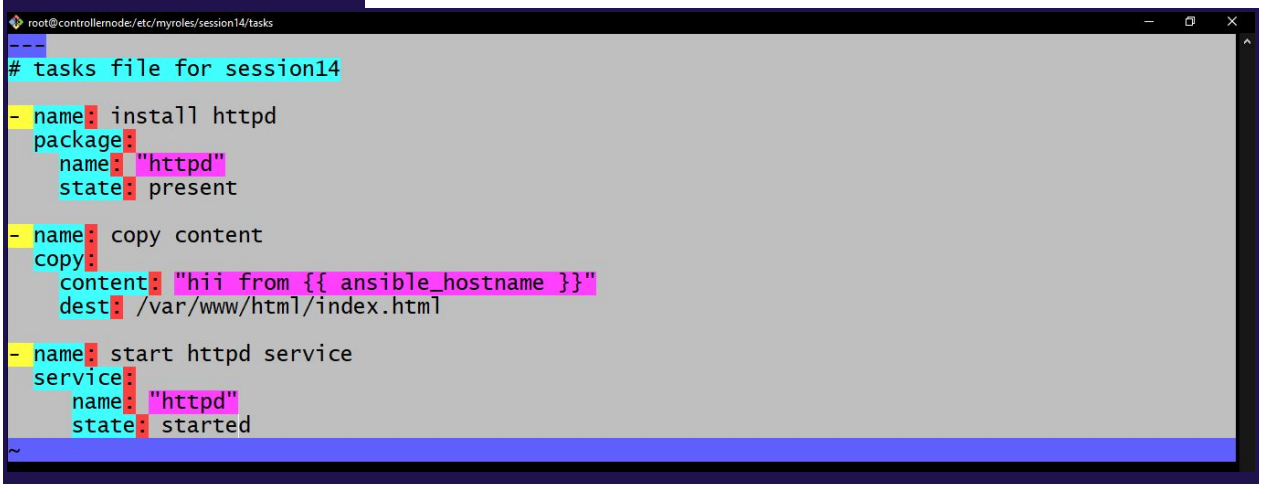
root@controllernode:/etc/myroles/loadbalancersetup/templates

```
frontend main
    bind *:8080
    acl url_static       path_beg       -i /static /images /javascript /stylesheets
    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 i in groups['webserver']   %}

    server  app1 {{ i }}:80 check

{% endfor %}
                                                          89,0-1        Bot
```

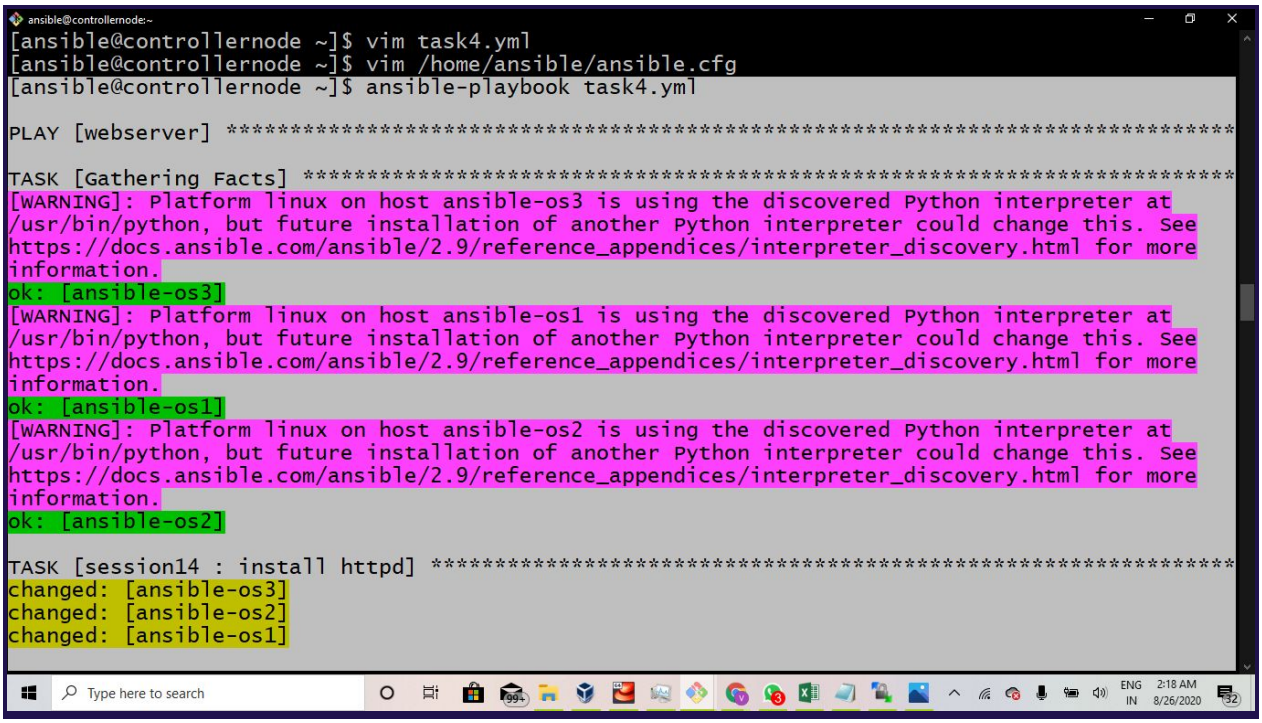

root@controllernode:/etc/myroles/loadbalancersetup/handlers

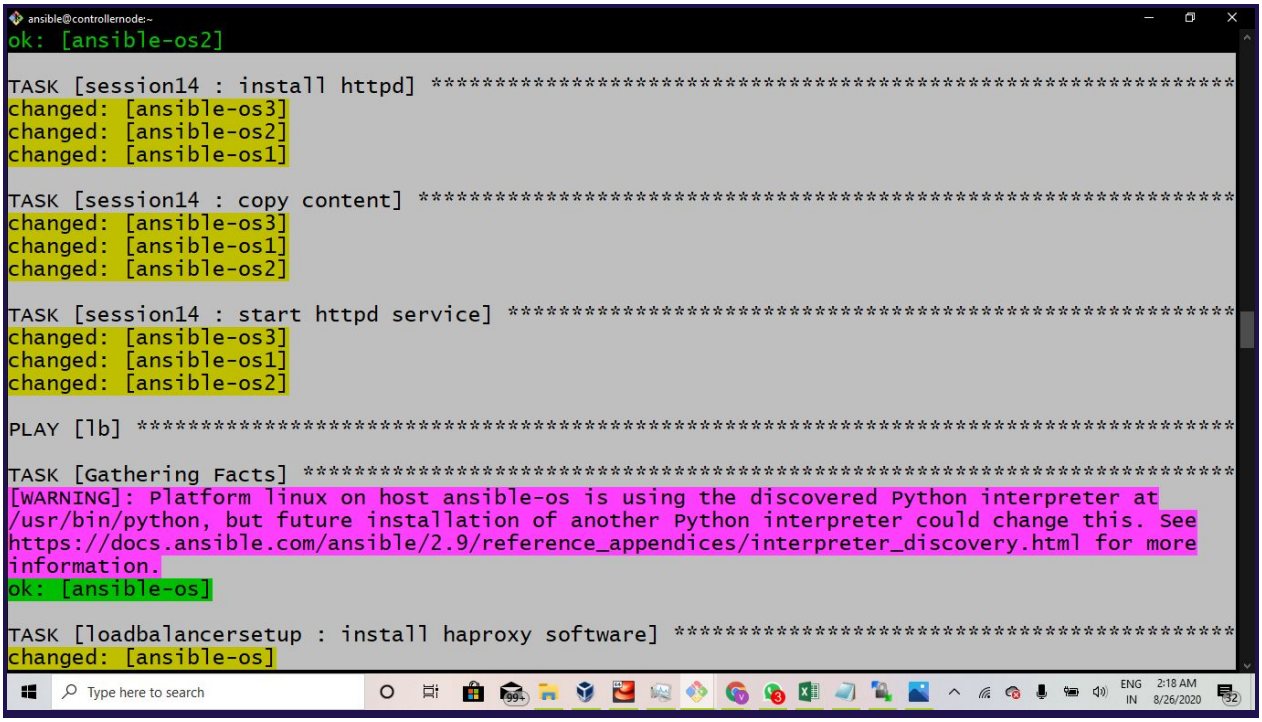
```
---
# handlers file for loadbalancersetup

- name: lbrestart
  service:
          name: "haproxy"
          state: restarted
```

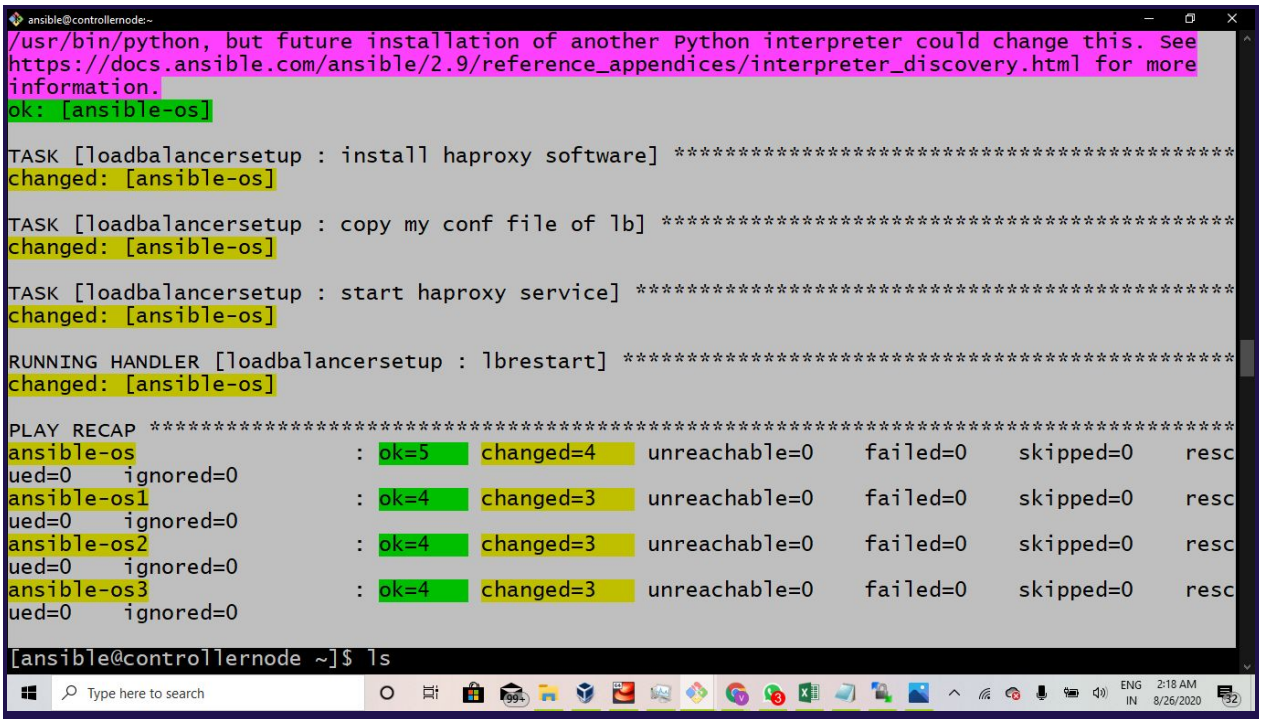
Step 11: Configuring httpd server inside the session14 role



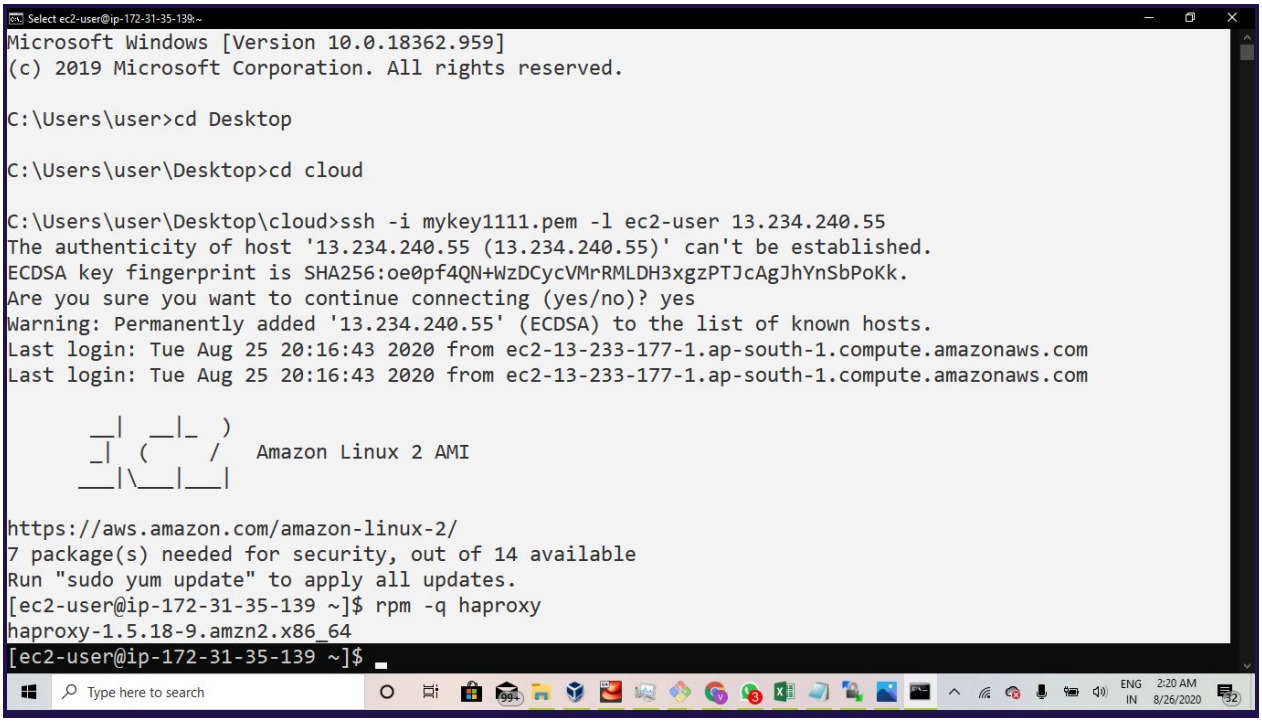
Step 12: Running ansible-playbook command


ansible@controllernode:~
[ansible@controllernode ~]$ vim task4.yml
[ansible@controllernode ~]$ vim /home/ansible/ansible.cfg
[ansible@controllernode ~]$ ansible-playbook task4.yml
PLAY [webserver] ***
TASK [Gathering Facts] ***
[WARNING]: Platform linux on host ansible-os3 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/reference_appendices/interpreter_discovery.html for more
information.
ok: [ansible-os3]
[WARNING]: Platform linux on host ansible-os1 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/reference_appendices/interpreter_discovery.html for more
information.
ok: [ansible-os1]
[WARNING]: Platform linux on host ansible-os2 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/reference_appendices/interpreter_discovery.html for more
information.
ok: [ansible-os2]
TASK [session14 : install httpd] ***
changed: [ansible-os3]
changed: [ansible-os2]
changed: [ansible-os1]
Type here to search
ENG IN 2:18 AM 8/26/2020


ansible@controllernode:~
ok: [ansible-os2]
TASK [session14 : install httpd] ***
changed: [ansible-os3]
changed: [ansible-os2]
changed: [ansible-os1]
TASK [session14 : copy content] **
changed: [ansible-os3]
changed: [ansible-os1]
changed: [ansible-os2]
TASK [session14 : start httpd service] ***
changed: [ansible-os3]
changed: [ansible-os1]
changed: [ansible-os2]
PLAY [lb] **
TASK [Gathering Facts] ***
[WARNING]: Platform linux on host ansible-os 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/reference_appendices/interpreter_discovery.html for more
information.
ok: [ansible-os]
TASK [loadbalancersetup : install haproxy software] ************************************
changed: [ansible-os]
Type here to search
ENG IN 2:18 AM 8/26/2020



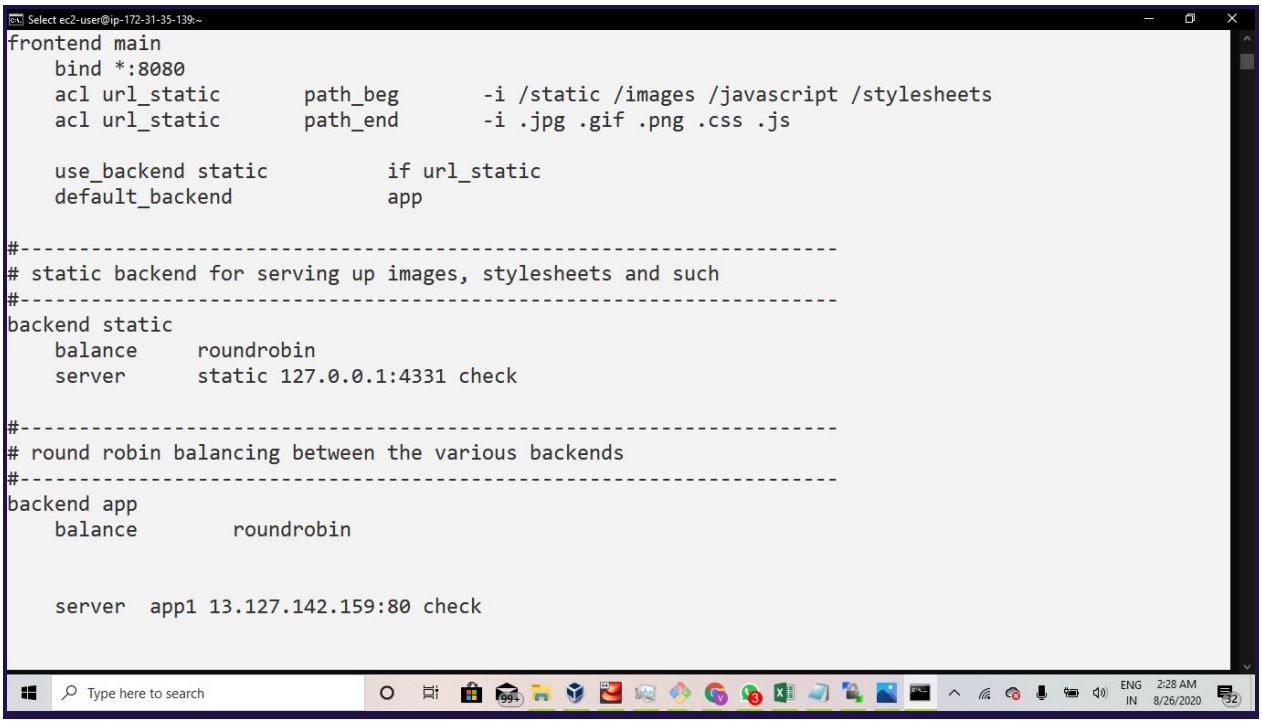
Step 13: final Evaluation



```
[ec2-user@ip-172-31-35-139 ~]$ systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/usr/lib/systemd/system/haproxy.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2020-08-25 20:15:55 UTC; 41min ago
 Main PID: 5737 (haproxy-systemd)
   CGroup: /system.slice/haproxy.service
           ├─5737 /usr/sbin/haproxy-systemd-wrapper -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid
           ├─5738 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -Ds
           └─5739 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -Ds

Aug 25 20:15:55 ip-172-31-35-139.ap-south-1.compute.internal systemd[1]: Started HAProxy Load Balancer.
Aug 25 20:15:55 ip-172-31-35-139.ap-south-1.compute.internal systemd[1]: Starting HAProxy Load Balan...
Aug 25 20:15:55 ip-172-31-35-139.ap-south-1.compute.internal haproxy-systemd-wrapper[5737]: haproxy-...
Aug 25 20:15:55 ip-172-31-35-139.ap-south-1.compute.internal haproxy-systemd-wrapper[5737]: [WARNING...
Aug 25 20:15:55 ip-172-31-35-139.ap-south-1.compute.internal haproxy-systemd-wrapper[5737]: [WARNING...
Hint: Some lines were ellipsized, use -l to show in full.
```

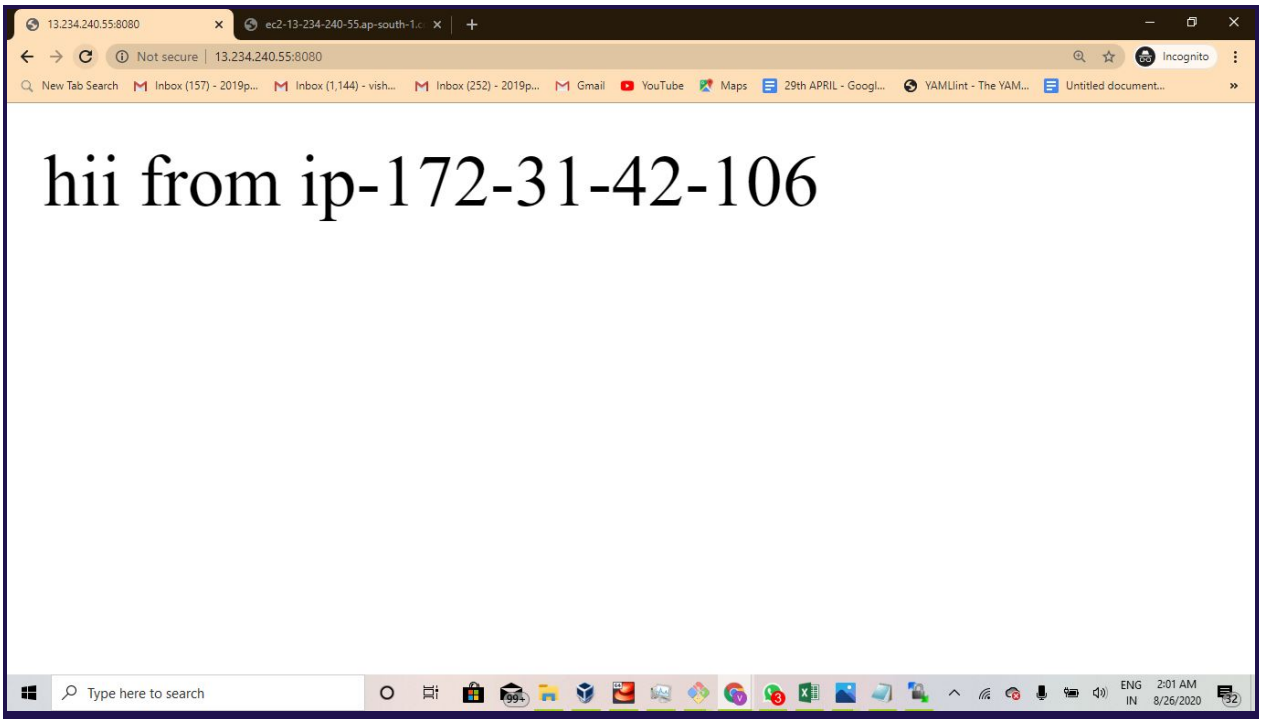
Select ec2-user@ip-172-31-35-139:~

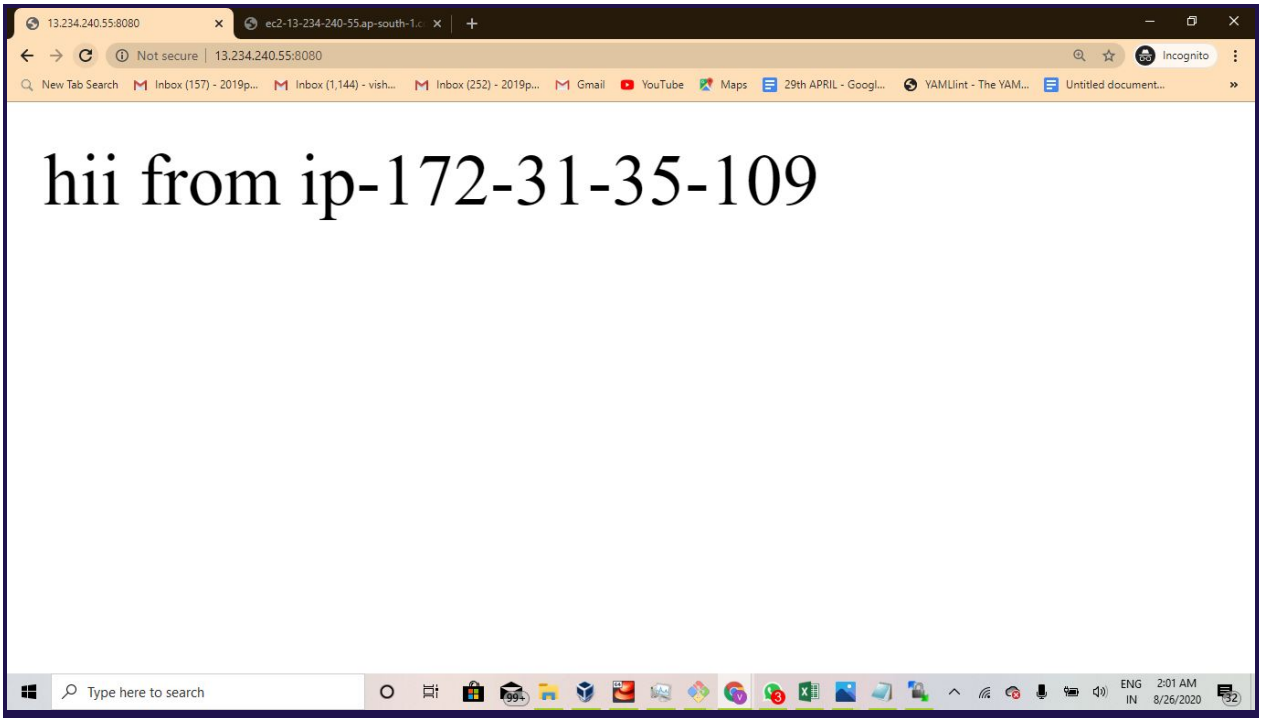
```
Aug 25 20:15:55 ip-172-31-35-139.ap-south-1.compute.internal haproxy-systemd-wrapper[5737]: [WARNING...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-35-139 ~]$ cat /etc/haproxy/haproxy.cfg
#---------------------------------------------------------------------
# Example configuration for a possible web application.  See the
# full configuration options online.
#
#   https://www.haproxy.org/download/1.8/doc/configuration.txt
#
#---------------------------------------------------------------------

#---------------------------------------------------------------------
# Global settings
#---------------------------------------------------------------------
global
    # to have these messages end up in /var/log/haproxy.log you will
    # need to:
    #
    # 1) configure syslog to accept network log events.  This is done
    #    by adding the '-r' option to the SYSLOGD_OPTIONS in
    #    /etc/sysconfig/syslog
    #
    # 2) configure local2 events to go to the /var/log/haproxy.log
    #   file. A line like the following can be added to
    #   /etc/sysconfig/syslog
```

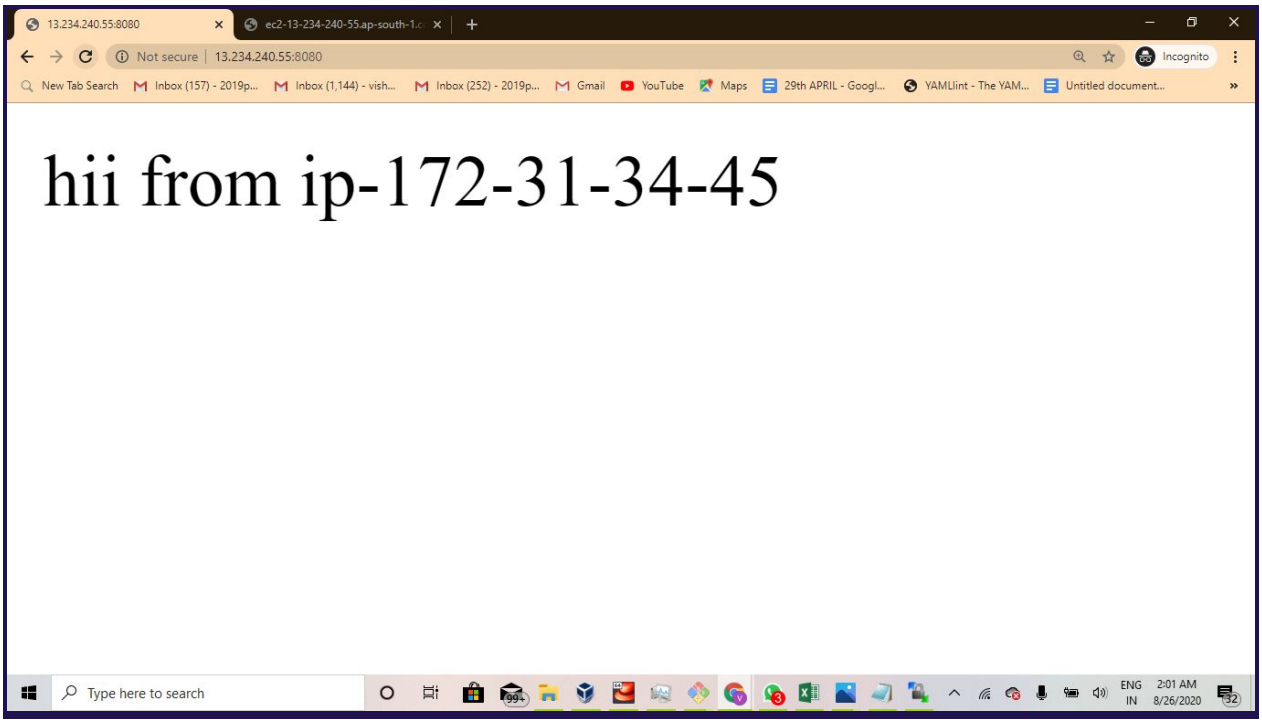


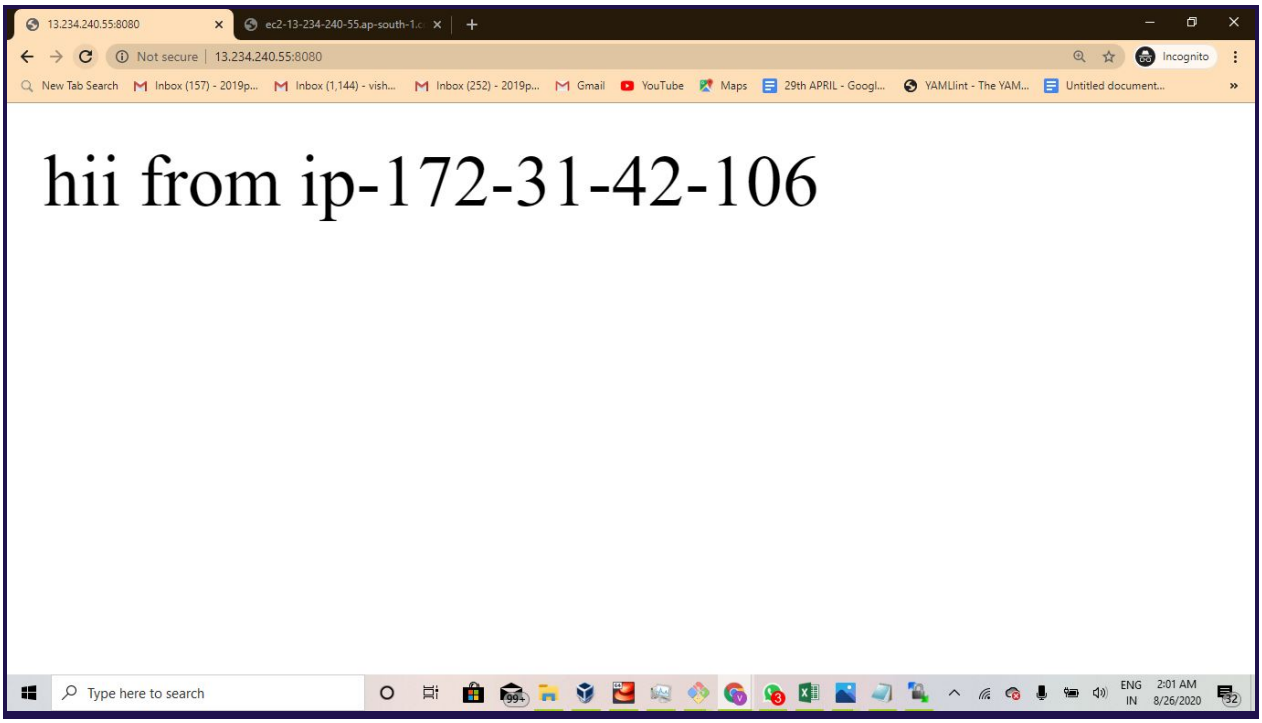

```
    server  app1 13.127.142.159:80 check


    server  app1 13.234.78.166:80 check


    server  app1 15.206.94.86:80 check
```

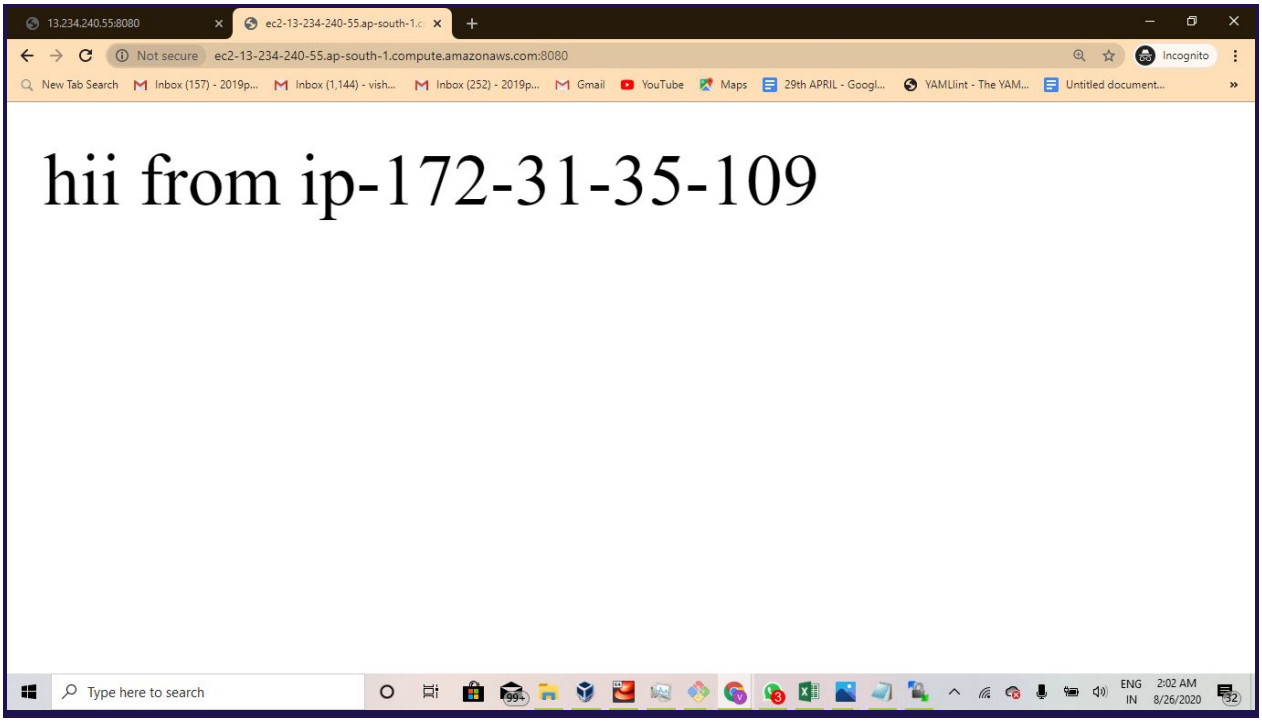
Step 14: Outputs


13.234.240.55:8080
ec2-13-234-240-55.ap-south-1.c
Not secure | 13.234.240.55:8080
Incognito
New Tab Search
Inbox (157) - 2019p...
Inbox (1,144) - vish...
Inbox (252) - 2019p...
Gmail
YouTube
Maps
29th APRIL - Googl...
YAMLint - The YAM...
Untitled document...
hii from ip-172-31-42-106
Type here to search
ENG
IN
2:01 AM
8/26/2020


13.234.240.55:8080
ec2-13-234-240-55.ap-south-1.c
Not secure | 13.234.240.55:8080
Incognito
New Tab Search
Inbox (157) - 2019p...
Inbox (1,144) - vish...
Inbox (252) - 2019p...
Gmail
YouTube
Maps
29th APRIL - Googl...
YAMLint - The YAM...
Untitled document...
hii from ip-172-31-35-109
Type here to search
ENG
IN
2:01 AM
8/26/2020


13.234.240.55:8080
ec2-13-234-240-55.ap-south-1.c
Not secure | 13.234.240.55:8080
Incognito
New Tab Search
Inbox (157) - 2019p...
Inbox (1,144) - vish...
Inbox (252) - 2019p...
Gmail
YouTube
Maps
29th APRIL - Googl...
YAMLint - The YAM...
Untitled document...
hii from ip-172-31-34-45
Type here to search
ENG
IN
2:01 AM
8/26/2020





Note:

Allow all the traffic and launch the instances in the default vpc
Thanks