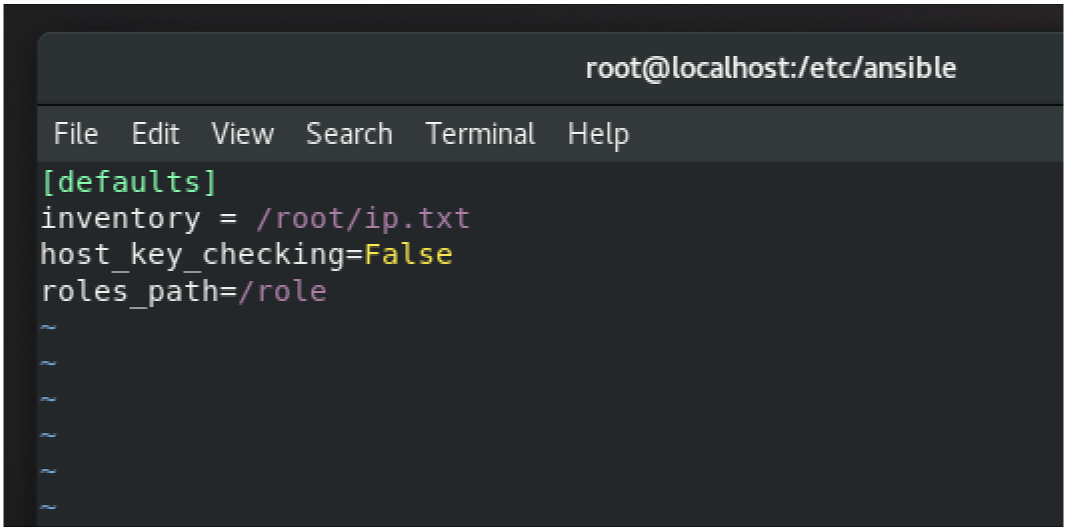
Roles in Ansible

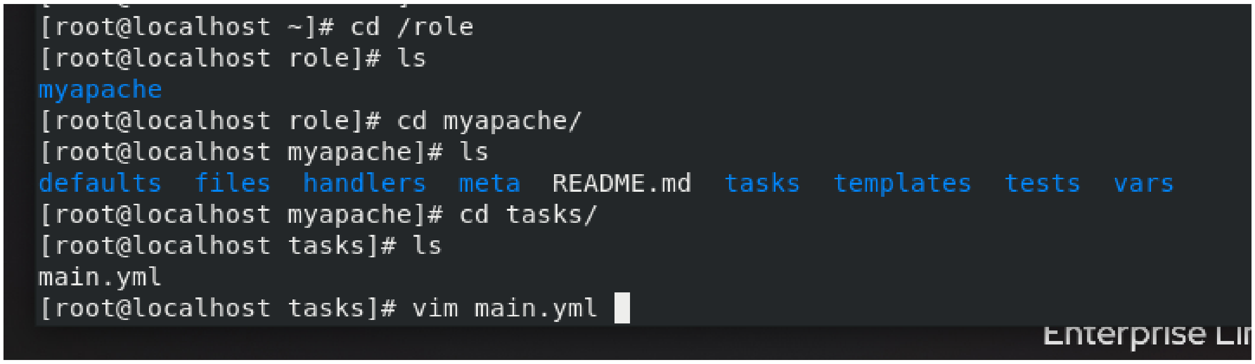
Configuring webserver using Roles

1. Foremost, give the path of the role in the configuration file of ansible. Go to the configuration file using **vim /etc/ansible/ansible.cfg** and write the following.


root@localhost:/etc/ansible
File Edit View Search Terminal Help

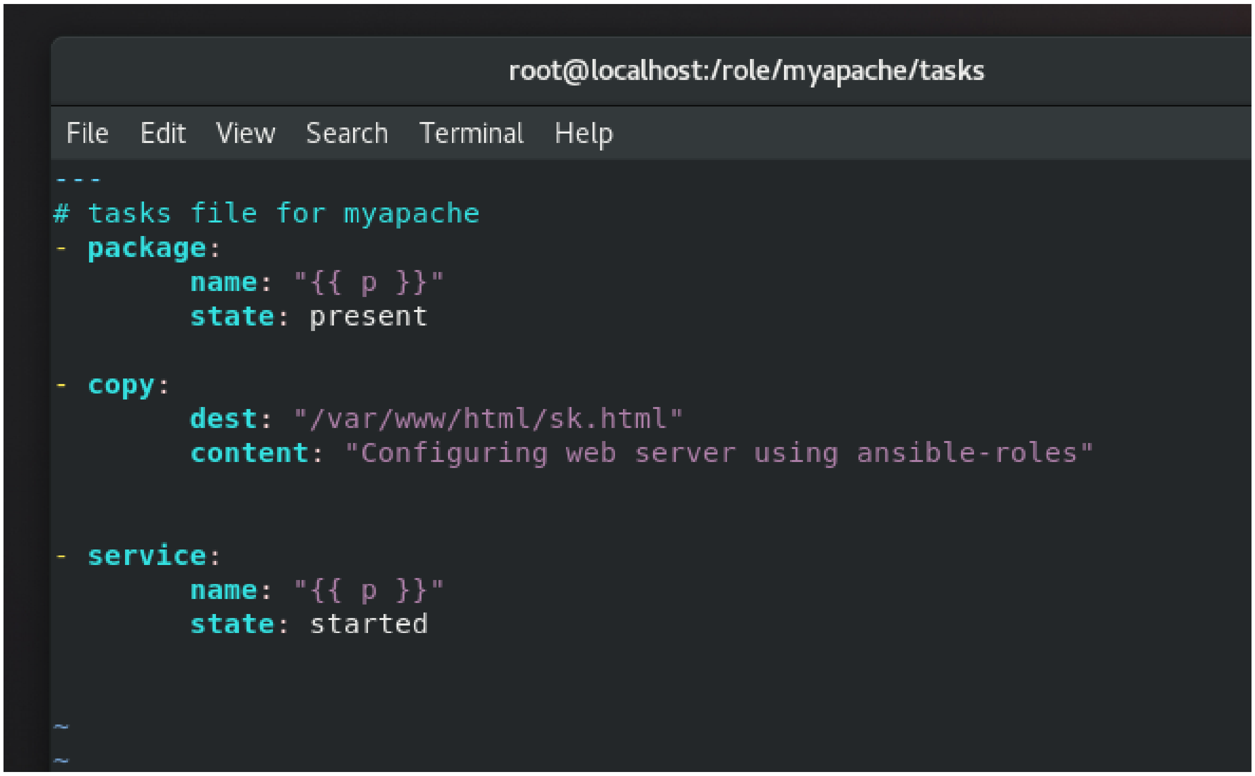
```
[defaults]
inventory = /root/ip.txt
host_key_checking=False
roles_path=/role
~
~
~
~
~
~
```

2. Create a directory of the same name as given in the configuration file in the role_path. Go to that directory and create a role using **ansible-galaxy init myapache**
3. Go inside that directory and use ls.
4. As we created a role named myapache, 8 directories and 1 file is created simultaneously.
5. To write the task of the yml file, go in that directory and open the main.yml file
   cd /myapache
   cd /tasks
   vim main.yml



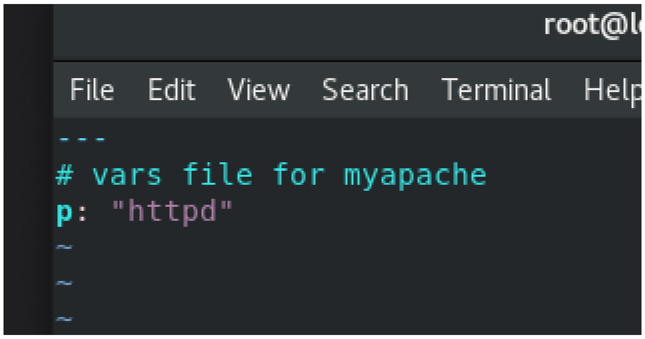
```
[root@localhost ~]# cd /role
[root@localhost role]# ls
myapache
[root@localhost role]# cd myapache/
[root@localhost myapache]# ls
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
[root@localhost myapache]# cd tasks/
[root@localhost tasks]# ls
main.yml
[root@localhost tasks]# vim main.yml
```

Write the following tasks in that file.


root@localhost:/role/myapache/tasks
File Edit View Search Terminal Help

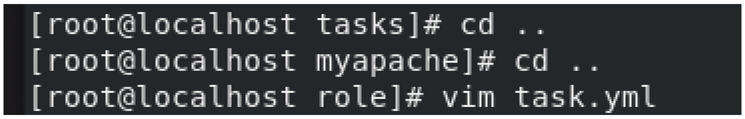
```
---
# tasks file for myapache
- package:
        name: "{{ p }}"
        state: present

- copy:
        dest: "/var/www/html/sk.html"
        content: "Configuring web server using ansible-roles"


- service:
        name: "{{ p }}"
        state: started



~
~
```

6. Now, we go to the vars folder and give the name of the variable that we are using in the task folder’s yml fle.
   cd ..
   cd vars
   vim main.yml
   Write the following in the yml file of the var folder.



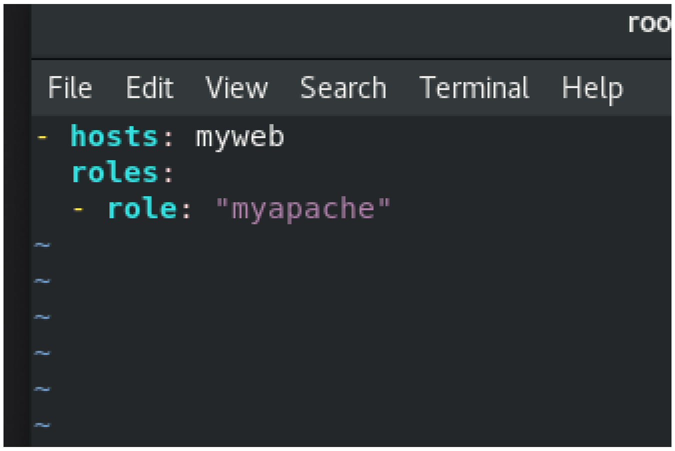
```
---
# vars file for myapache
p: "httpd"
~
~
~
```

7. Now, come back to the main directory and create a yml file.



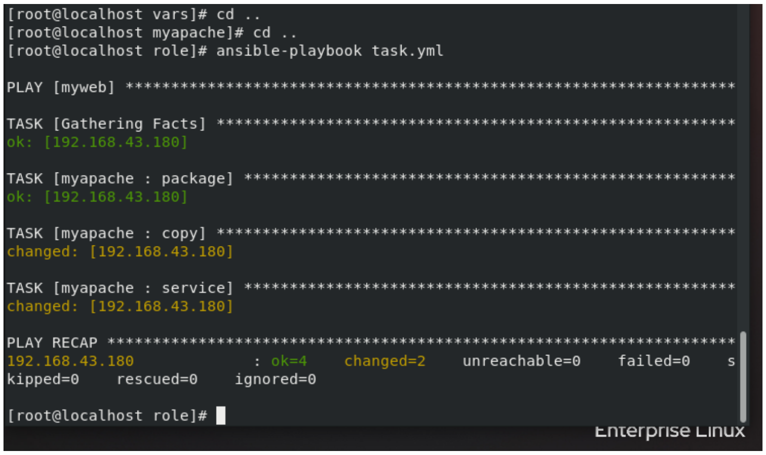
```
[root@localhost tasks]# cd ..
[root@localhost myapache]# cd ..
[root@localhost role]# vim task.yml
```

8. Write the following contents.
   This is the main file that we shall run. It contains the host name and the name of the role that we want to run.

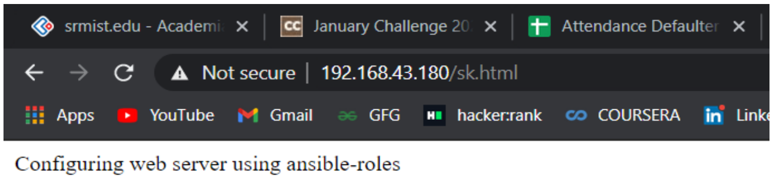


```
- hosts: myweb
  roles:
  - role: "myapache"
~
~
~
~
~
~
```

9. Run this file and use the browser to view the contents.



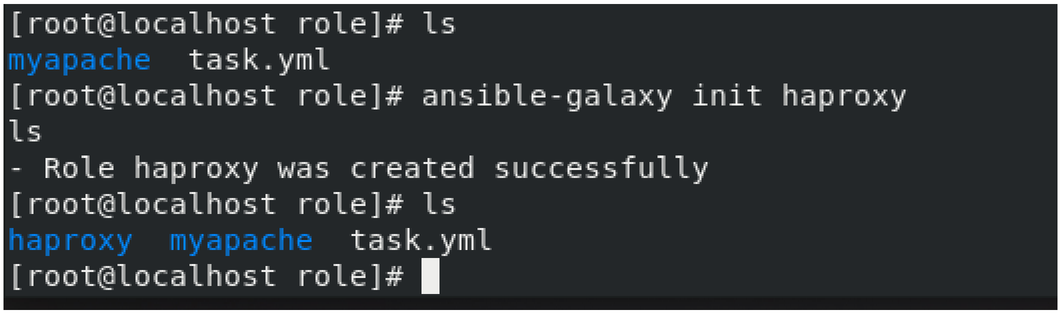
```
[root@localhost vars]# cd ..
[root@localhost myapache]# cd ..
[root@localhost role]# ansible-playbook task.yml

PLAY [myweb] *********************************************************************

TASK [Gathering Facts] ***********************************************************
ok: [192.168.43.180]

TASK [myapache : package] ********************************************************
ok: [192.168.43.180]

TASK [myapache : copy] ***********************************************************
changed: [192.168.43.180]

TASK [myapache : service] ********************************************************
changed: [192.168.43.180]

PLAY RECAP ***********************************************************************
192.168.43.180             : ok=4    changed=2    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

[root@localhost role]# 
```



Configuring web server using ansible-roles

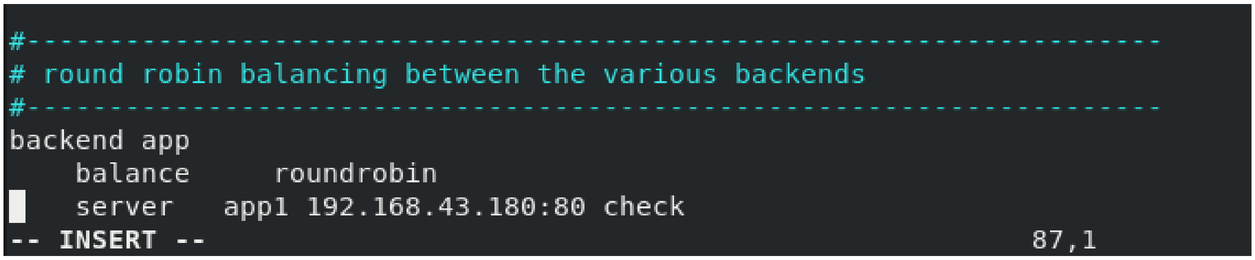
Configuring Haproxy using Roles

1. In the same directory, create one more role using **ansible-galaxy init haproxy**



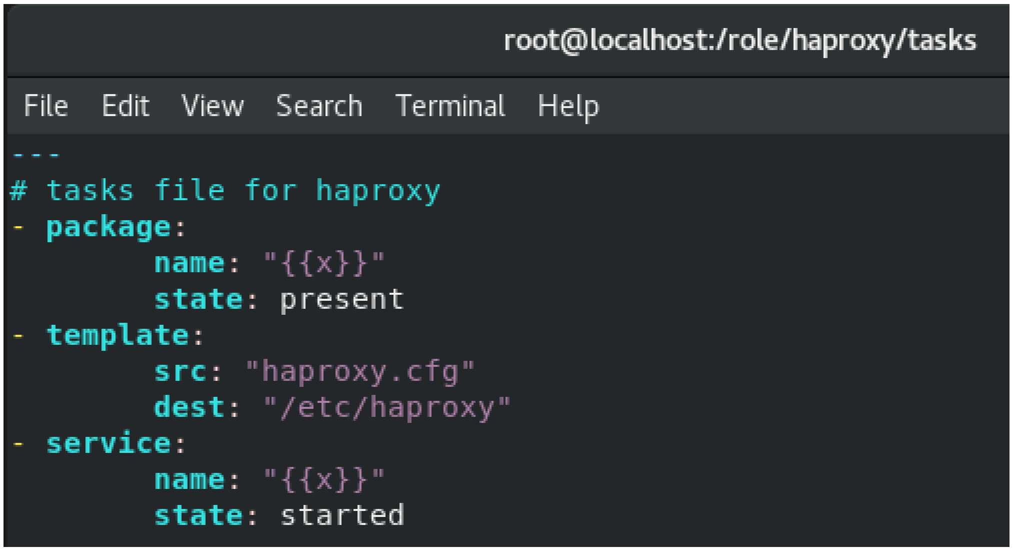
```
[root@localhost role]# ls
myapache  task.yml
[root@localhost role]# ansible-galaxy init haproxy
ls
- Role haproxy was created successfully
[root@localhost role]# ls
haproxy  myapache  task.yml
[root@localhost role]# 
```

2. Copy the configuration file of haproxy and provide the ip of the webserver.



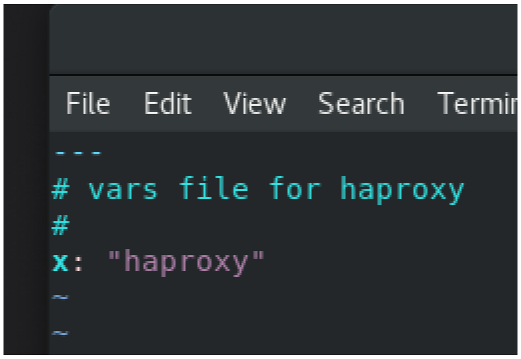
```
#---------------------------------------------------------------------
# round robin balancing between the various backends
#---------------------------------------------------------------------
backend app
    balance     roundrobin
    server   app1 192.168.43.180:80 check
-- INSERT --                                                  87,1
```

3. In the directory named task, open the main.yml file and write the following code.



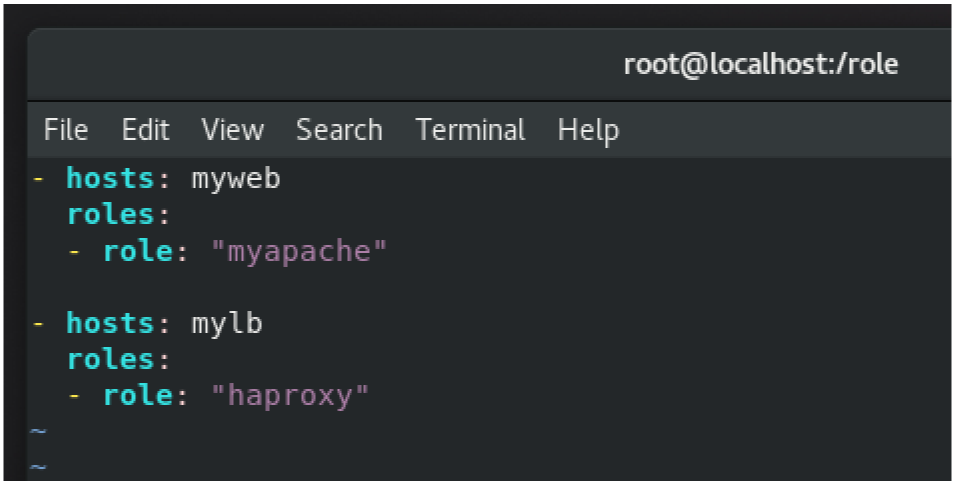
```
---
# tasks file for haproxy
- package:
        name: "{{x}}"
        state: present
- template:
        src: "haproxy.cfg"
        dest: "/etc/haproxy"
- service:
        name: "{{x}}"
        state: started
```

4. In the main.yml file of var directory, write the following code.



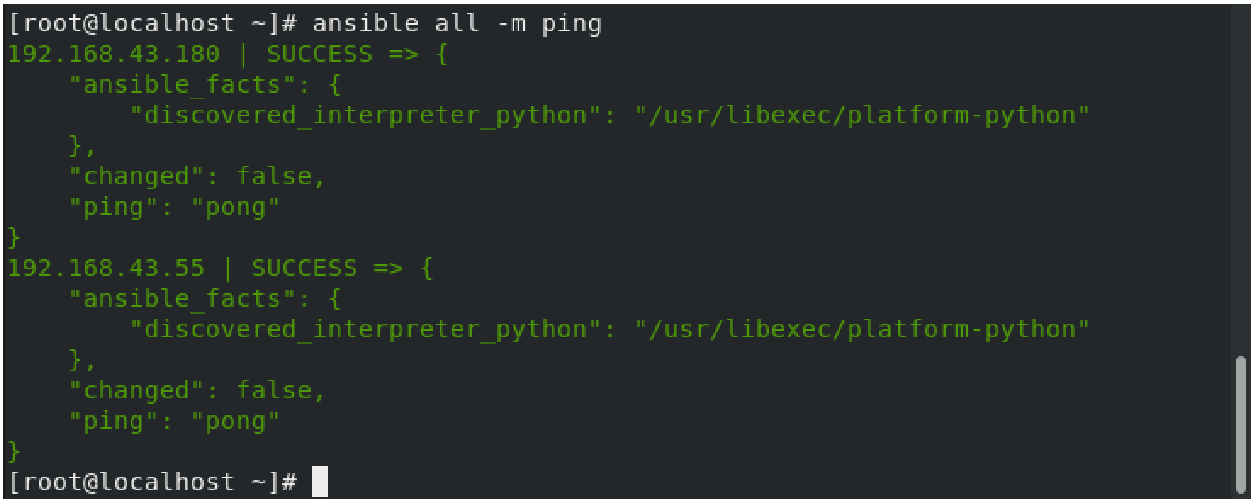
```
---
# vars file for haproxy
#
x: "haproxy"
~
~
```

5. In the yml file of the main directory, we write the following code.



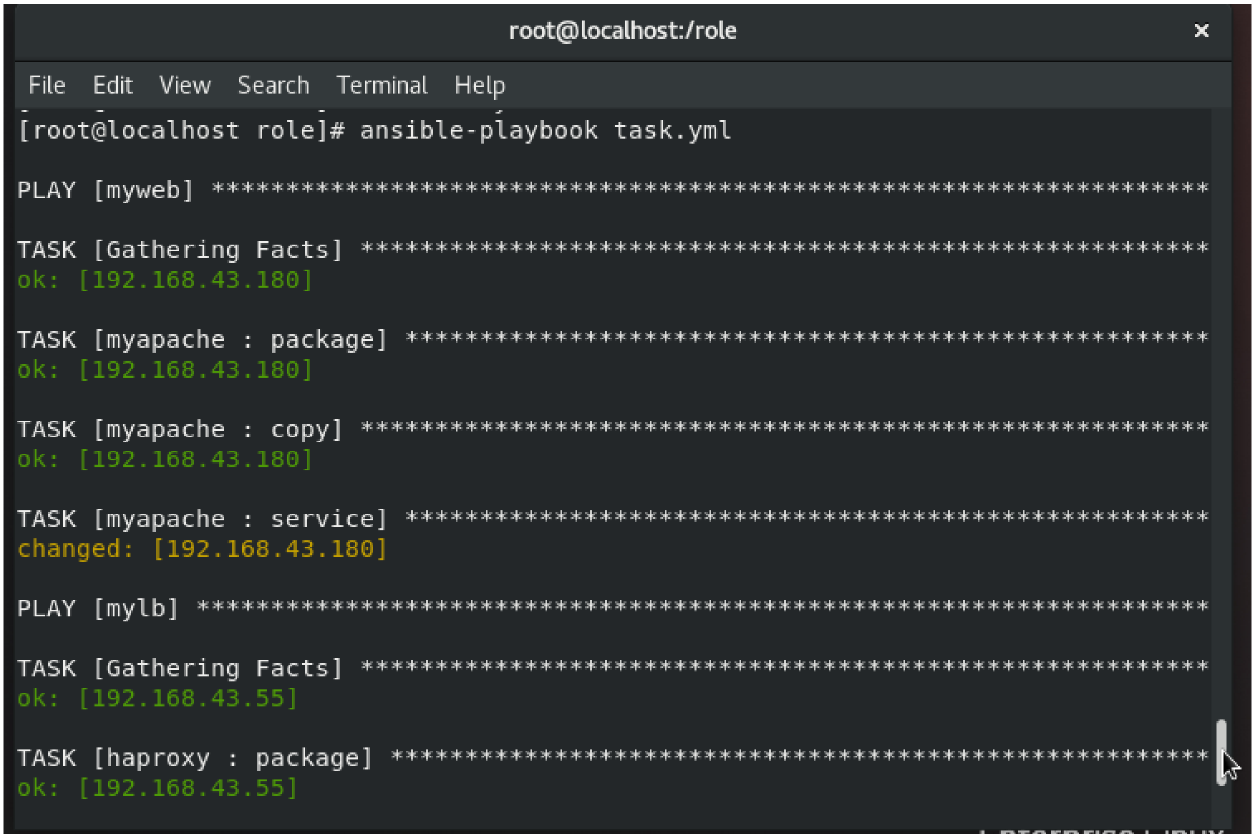
```
- hosts: myweb
  roles:
  - role: "myapache"

- hosts: mylb
  roles:
  - role: "haproxy"
~
~
```

6. Check the connectivity ones using **ansible all –m ping**



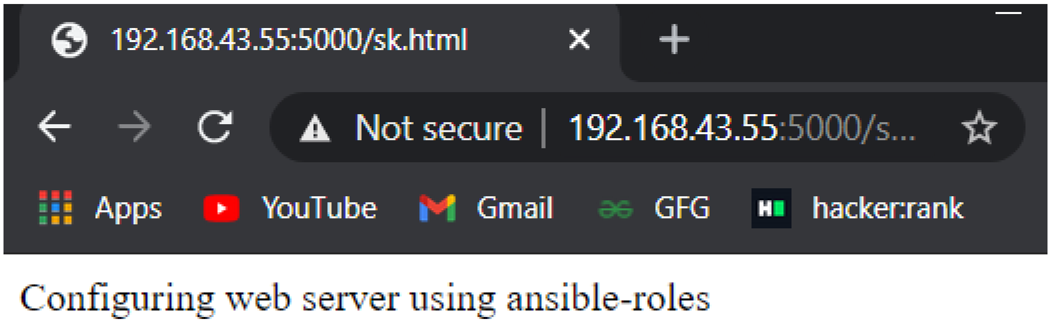
```
[root@localhost ~]# ansible all -m ping
192.168.43.180 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
192.168.43.55 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
[root@localhost ~]# 
```

7. Run the main yml file.



```
[root@localhost role]# ansible-playbook task.yml

PLAY [myweb] *******************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.43.180]

TASK [myapache : package] ******************************************************
ok: [192.168.43.180]

TASK [myapache : copy] *********************************************************
ok: [192.168.43.180]

TASK [myapache : service] ******************************************************
changed: [192.168.43.180]

PLAY [mylb] ********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.43.55]

TASK [haproxy : package] *******************************************************
ok: [192.168.43.55]
```

8. Go to the browser. Instead of the IP of the webserver, provide the ip of the load balancer. Also, provide the port number of haproxy i.e. 5000



Configuring web server using ansible-roles

The task id successful