Task 15.1

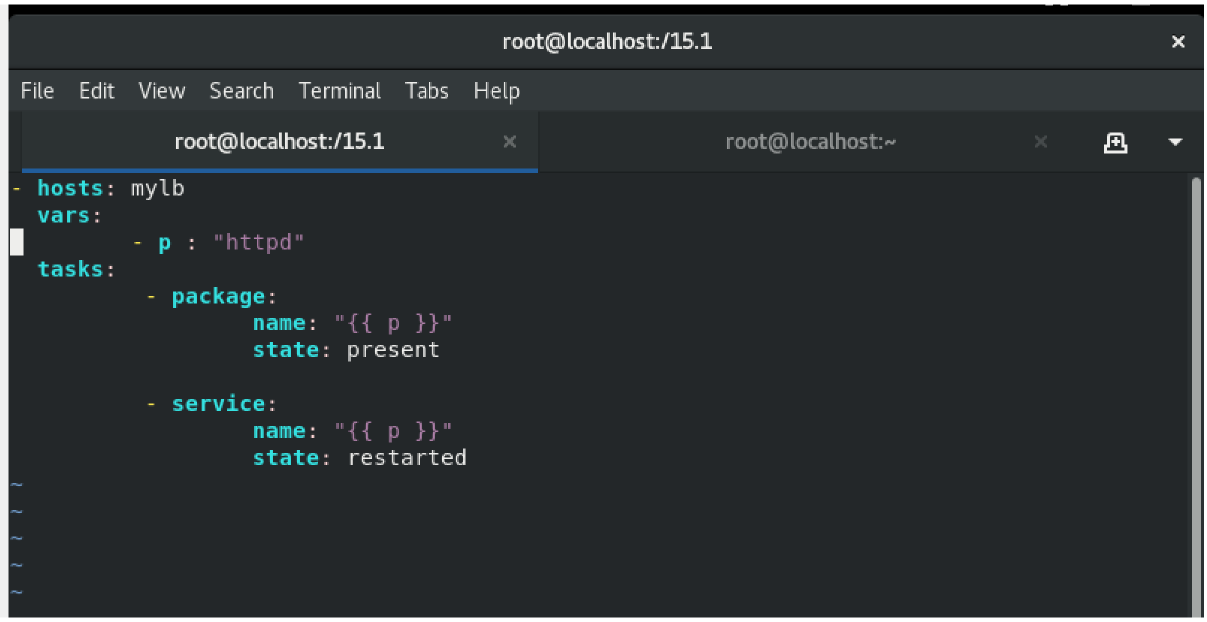
Writing the ansible code to install and restart httpd without using role

Making a workspace:

mkdir /15.1

cd /15.1

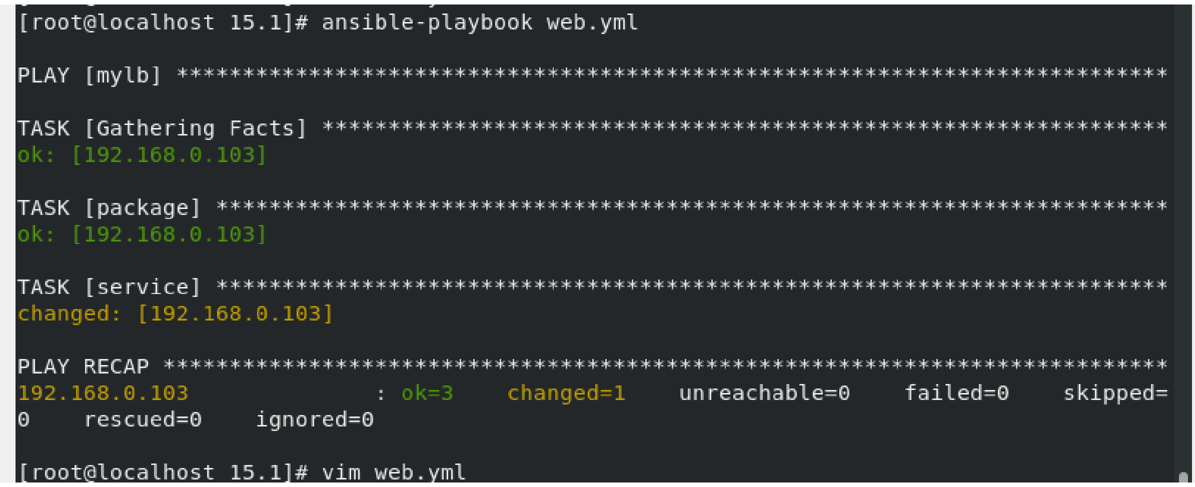
vim web.yml



```
- hosts: mylb
  vars:
        - p : "httpd"
  tasks:
         - package:
                name: "{{ p }}"
                state: present

         - service:
                name: "{{ p }}"
                state: restarted
~
~
~
~
~
```

Running the code.

ansible-playbook web.yml



```
[root@localhost 15.1]# ansible-playbook web.yml

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

TASK [Gathering Facts] *************************************************************
ok: [192.168.0.103]

TASK [package] *********************************************************************
ok: [192.168.0.103]

TASK [service] *********************************************************************
changed: [192.168.0.103]

PLAY RECAP *************************************************************************
192.168.0.103              : ok=3    changed=1    unreachable=0    failed=0    skipped=
0    rescued=0    ignored=0

[root@localhost 15.1]# vim web.yml
```

..

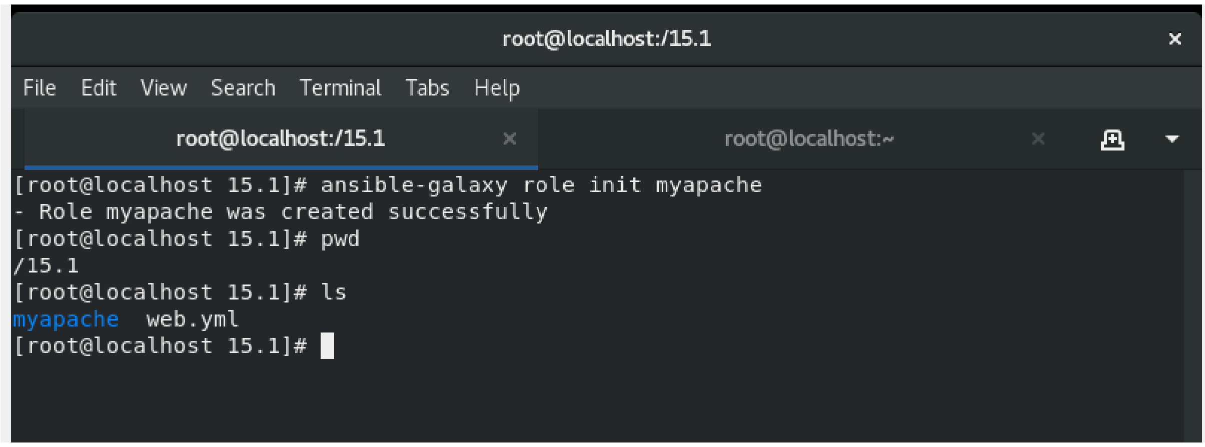
Task 15.1

Let’s Create an ansible role myapache to configure Httpd WebServer

FAQ’s on ROLES

👉What is role in ansible?
✔ It is just the folder name, where the code Is individually organized in the subfolders having a main.yml file in it.
👉How to create role?
✔. command: ansible-galaxy role init <role_name>
👉How to set the path of role?
✔. ansible-galaxy role list –roles-path </dir/Of/Path/Of/Role>
👉What is ansible galaxy?
✔. It is an online community where several pre-created roles are published by the developers.
👉How to see available role in our system?
✔. ansible-galaxy role list

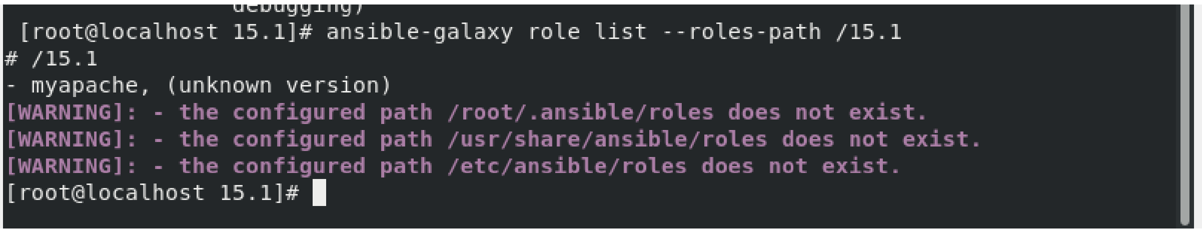
1. Creating a role.



```
[root@localhost 15.1]# ansible-galaxy role init myapache
- Role myapache was created successfully
[root@localhost 15.1]# pwd
/15.1
[root@localhost 15.1]# ls
myapache  web.yml
[root@localhost 15.1]#
```

Let’s list the role, but before that we have give the path of the role too.

Ansible-galaxy role list –roles-path /15.1



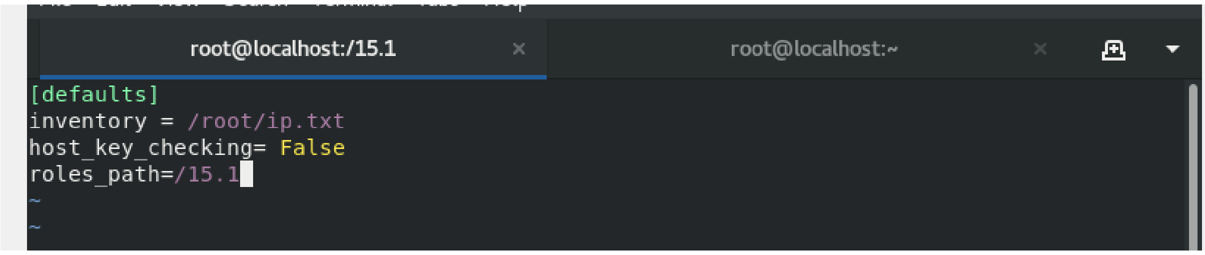
```
 debugging)
 [root@localhost 15.1]# ansible-galaxy role list --roles-path /15.1
# /15.1
- myapache, (unknown version)
[WARNING]: - the configured path /root/.ansible/roles does not exist.
[WARNING]: - the configured path /usr/share/ansible/roles does not exist.
[WARNING]: - the configured path /etc/ansible/roles does not exist.
[root@localhost 15.1]#
```

Task 15.1

2.

Or I can add the path the role in the config file.

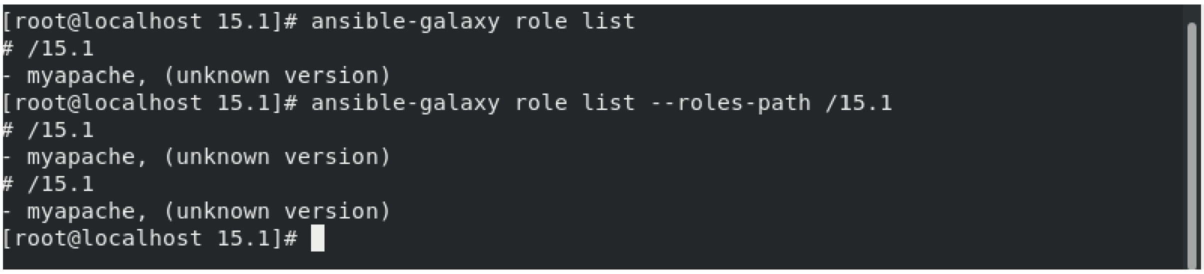
Vim /etc/ansible/ansible.cfg



```
[defaults]
inventory = /root/ip.txt
host_key_checking= False
roles_path=/15.1
~
~
```

Now we can check the role list without specifying the path also

ansible-galaxy role list



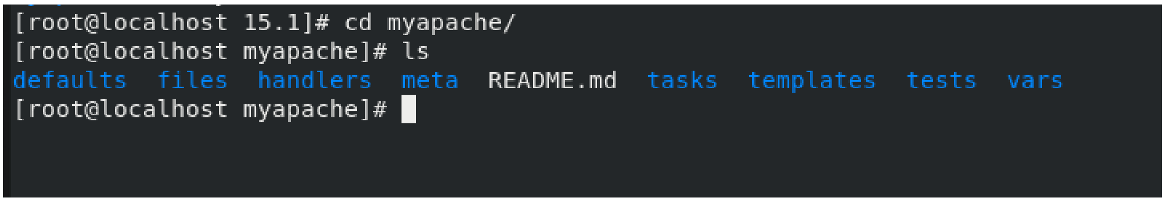
```
[root@localhost 15.1]# ansible-galaxy role list
# /15.1
- myapache, (unknown version)
[root@localhost 15.1]# ansible-galaxy role list --roles-path /15.1
# /15.1
- myapache, (unknown version)
# /15.1
- myapache, (unknown version)
[root@localhost 15.1]#
```

3.

Getting into the role.

cd myapache/

ls - list the files.

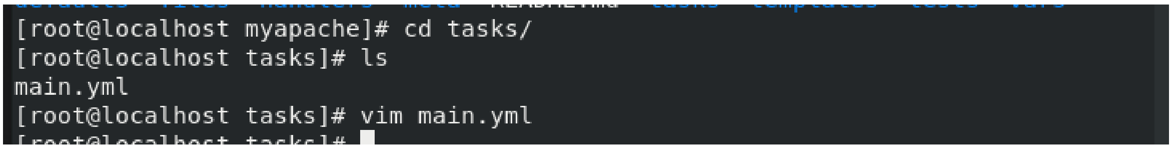


```
[root@localhost 15.1]# cd myapache/
[root@localhost myapache]# ls
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
[root@localhost myapache]#
```

It is the just the folder name, where the code Is individually organized in the subfolders having main.yml file in it.

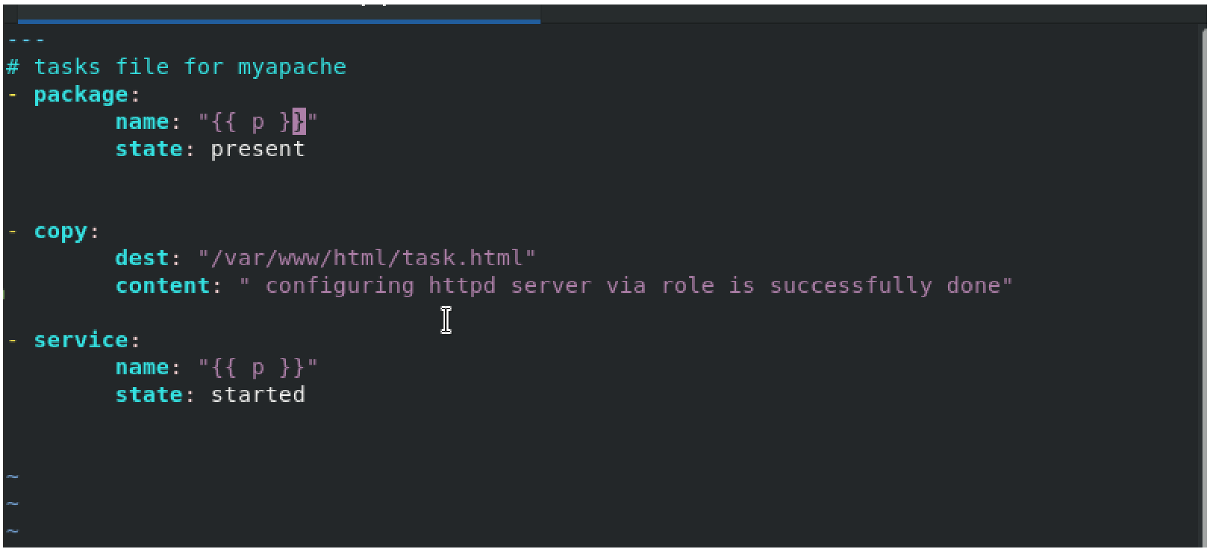
It has 9 subfolders: defaults, files, handler, meta, README.md , tasks, templates, tests, vars

Now, writing the above code in the specific folders,



```
[root@localhost myapache]# cd tasks/
[root@localhost tasks]# ls
main.yml
[root@localhost tasks]# vim main.yml
[root@localhost tasks]#
```

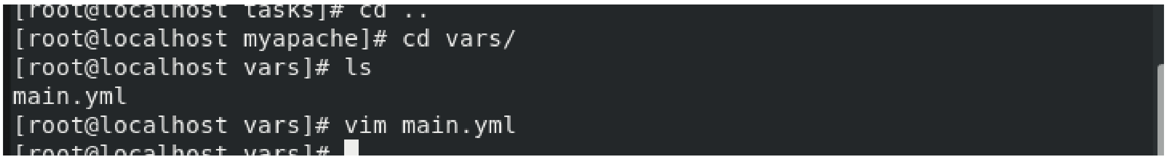
Task 15.1



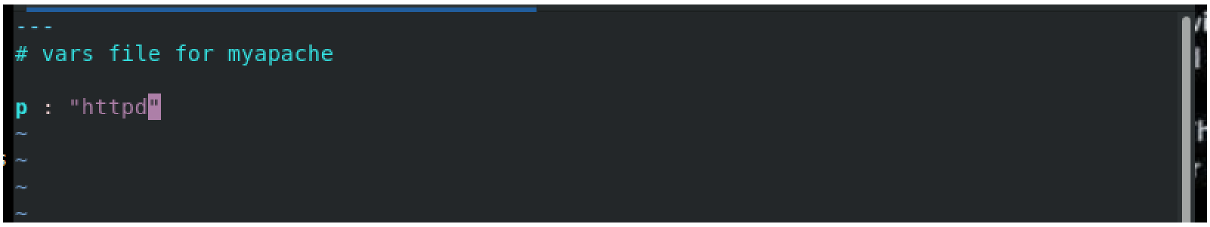
```
---
# tasks file for myapache
- package:
        name: "{{ p }}"
        state: present


- copy:
        dest: "/var/www/html/task.html"
        content: " configuring httpd server via role is successfully done"

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



~
~
~
```

Variables in vars folder

vim vars/main.yml



```
[root@localhost tasks]# cd ..
[root@localhost myapache]# cd vars/
[root@localhost vars]# ls
main.yml
[root@localhost vars]# vim main.yml
[root@localhost vars]#
```

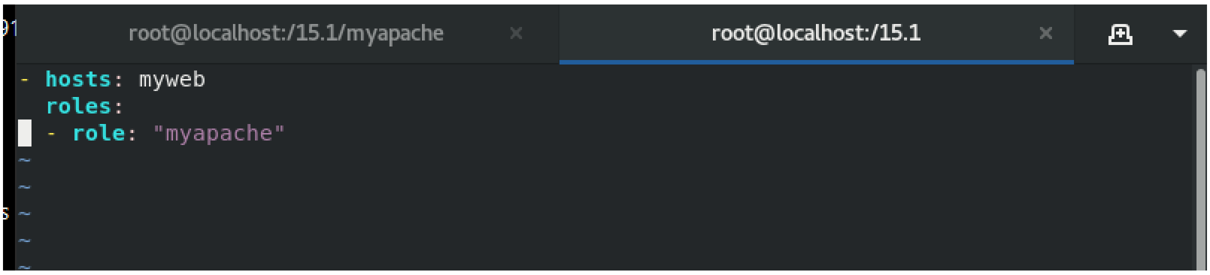


```
---
# vars file for myapache

p : "httpd"
~
~
~
~
```

4.

Creating another yml file to mentions the use of the role.

vim task.yml



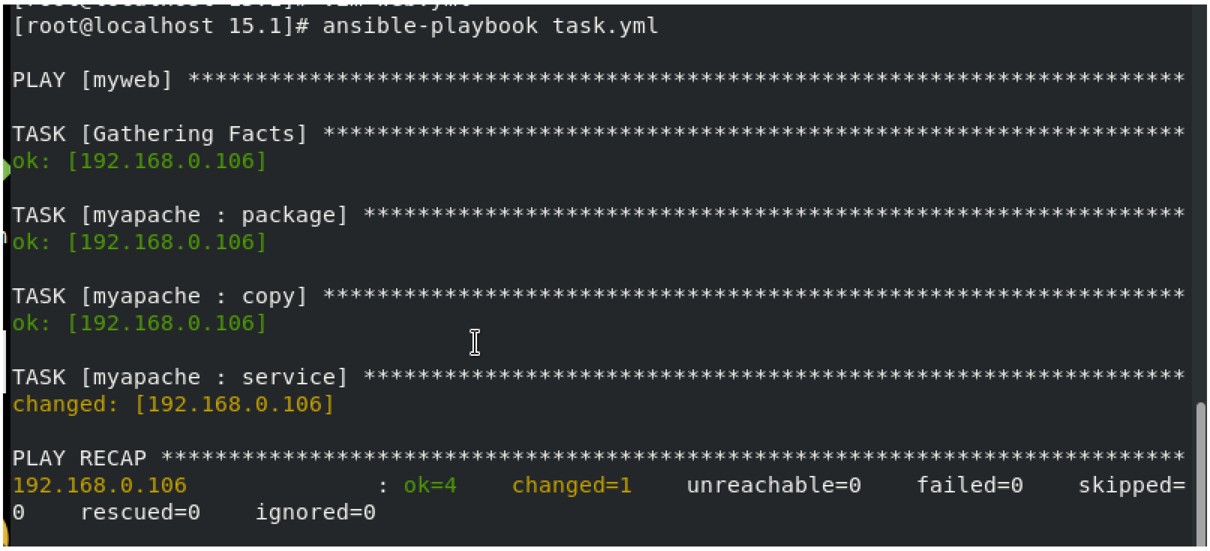
```
- hosts: myweb
  roles:
  - role: "myapache"
~
~
~
~
~
```

5.

Run the playbook….

ansible-playbook task.yml

Task 15.1

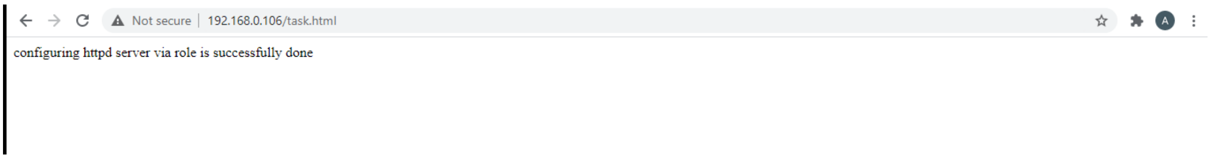


```
[root@localhost 15.1]# ansible-playbook task.yml

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.0.106]

TASK [myapache : package] ******************************************************
ok: [192.168.0.106]

TASK [myapache : copy] *********************************************************
ok: [192.168.0.106]

TASK [myapache : service] ******************************************************
changed: [192.168.0.106]

PLAY RECAP *********************************************************************
192.168.0.106              : ok=4    changed=1    unreachable=0    failed=0    skipped=
0    rescued=0    ignored=0
```

Checking if server is running.

->Go to a browser

->Search url : 192.168.0.106:80/task.html



configuring httpd server via role is successfully done

YES............it’s working....

Task 15.1

☀Create another ansible role myloadbalancer to configure HAProxy LB.

Intuition

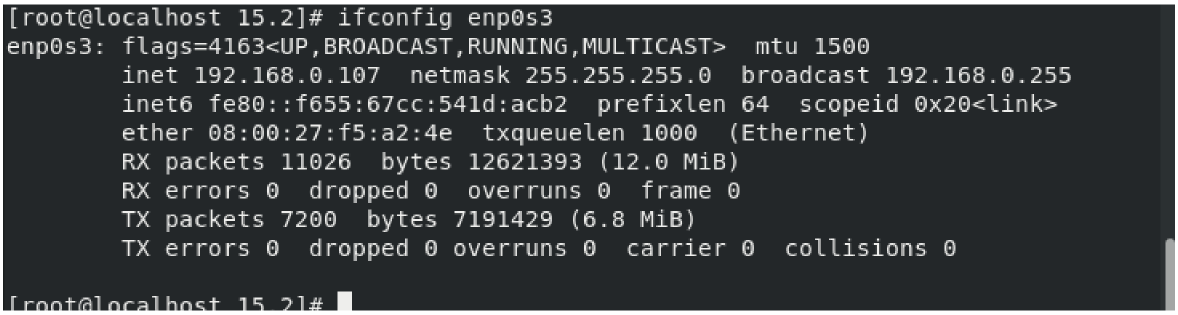
Step 1: adding the two ip in inverntory as mylb and myweb

Step 2: making two role myhaproxy and myapache and writing the code.

Step 3: making a new yml file and mentioning the role name to be performed on which host.

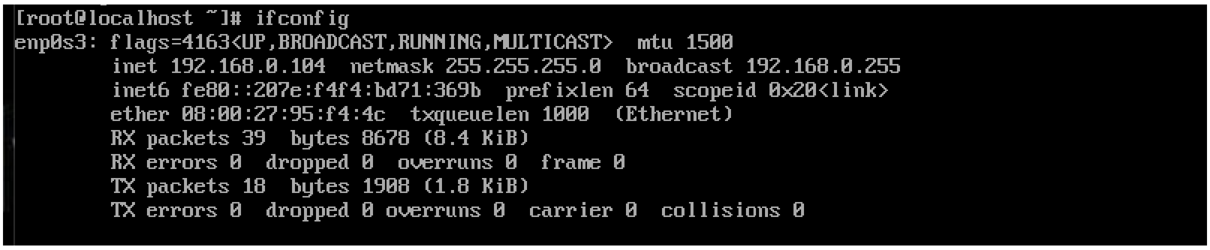
Step 4: Verification: access the server

CONTROL NODE IP



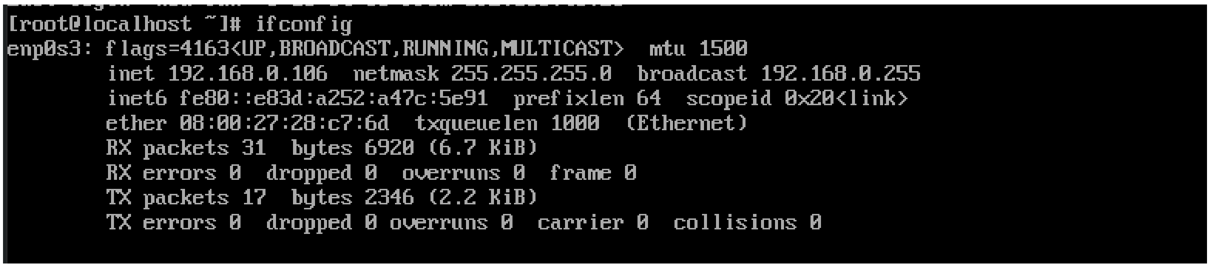
```
[root@localhost 15.2]# ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.107  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::f655:67cc:541d:acb2  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:f5:a2:4e  txqueuelen 1000  (Ethernet)
        RX packets 11026  bytes 12621393 (12.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7200  bytes 7191429 (6.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost 15.2]#
```

TARGET NODE IP: CONFIGURING AS LOAD BALANCER



```
[root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.104  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::207e:f4f4:bd71:369b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:95:f4:4c  txqueuelen 1000  (Ethernet)
        RX packets 39  bytes 8678 (8.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 1908 (1.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

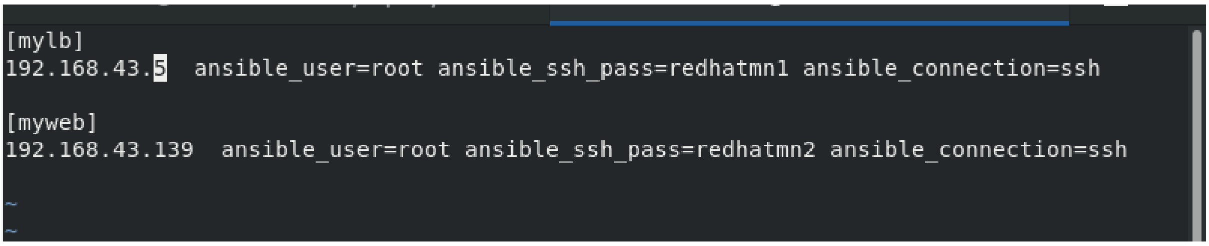
TARGET NODE IP: CONFIGURING AS apache backend server



```
[root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.106  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::e83d:a252:a47c:5e91  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:28:c7:6d  txqueuelen 1000  (Ethernet)
        RX packets 31  bytes 6920 (6.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17  bytes 2346 (2.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Task 15.1

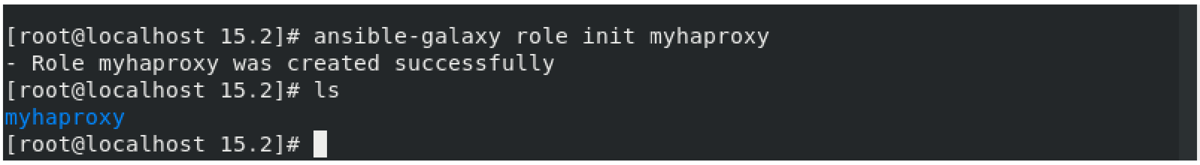
Inventory of Control Node.



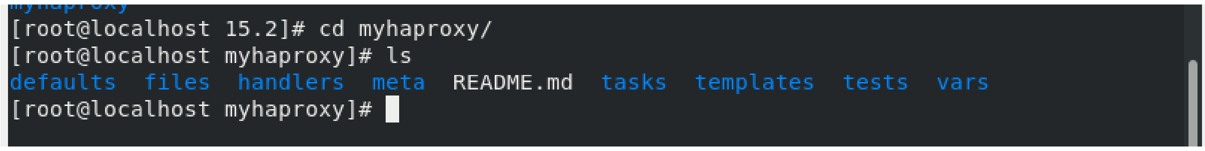
```
[mylb]
192.168.43.5  ansible_user=root ansible_ssh_pass=redhatmn1 ansible_connection=ssh

[myweb]
192.168.43.139  ansible_user=root ansible_ssh_pass=redhatmn2 ansible_connection=ssh

~
~
```

1. Creating a role

ansible-galaxy role init myhaproxy



```
[root@localhost 15.2]# ansible-galaxy role init myhaproxy
- Role myhaproxy was created successfully
[root@localhost 15.2]# ls
myhaproxy
[root@localhost 15.2]#
```

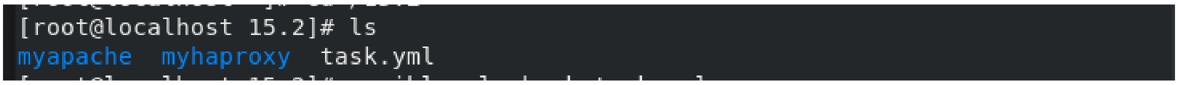


```
[root@localhost 15.2]# cd myhaproxy/
[root@localhost myhaproxy]# ls
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
[root@localhost myhaproxy]#
```

Similarly, for myapache role

ansible-galaxy role init myapache

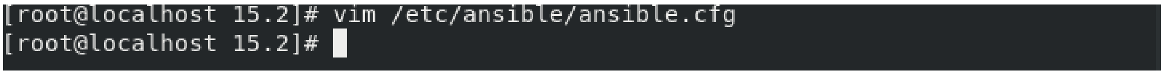
ls : to check the files



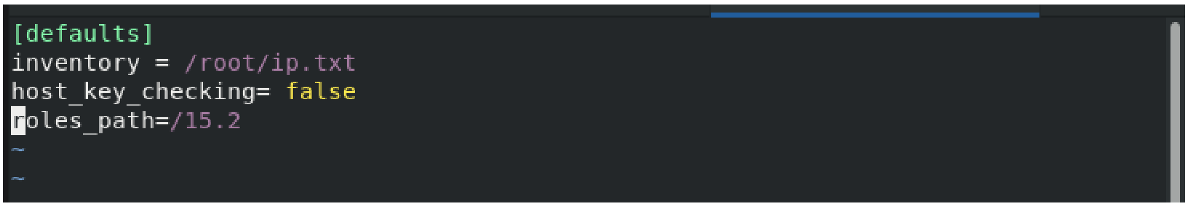
```
[root@localhost 15.2]# ls
myapache  myhaproxy  task.yml
```

Specifying the roles path in ansible.cfg

Vim /etc/ansible/ansible.cfg



```
[root@localhost 15.2]# vim /etc/ansible/ansible.cfg
[root@localhost 15.2]#
```

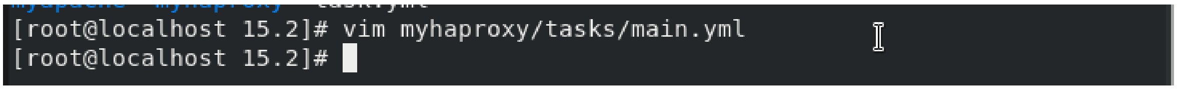


```
[defaults]
inventory = /root/ip.txt
host_key_checking= false
roles_path=/15.2
~
~
```

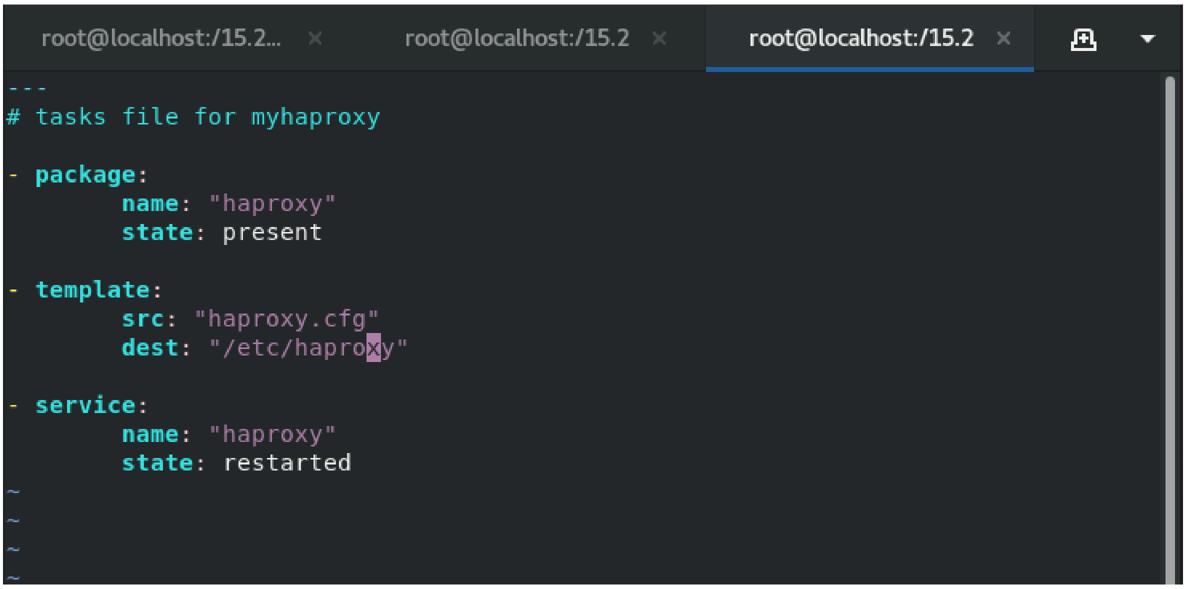
2. Writing the code in role myhaproxy-> tasks-> main.yml

vim myhaproxy/tasks/main.yml

Task 15.1

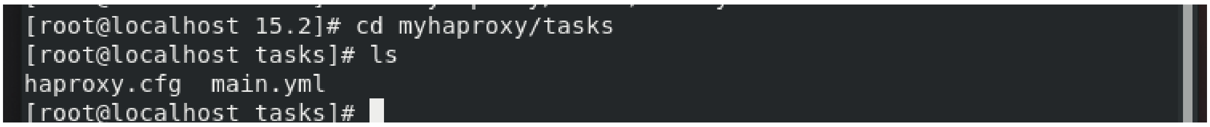


```
myapache  myhaproxy  task.yml
[root@localhost 15.2]# vim myhaproxy/tasks/main.yml
[root@localhost 15.2]# 
```



```
---
# tasks file for myhaproxy

- package:
        name: "haproxy"
        state: present

- template:
        src: "haproxy.cfg"
        dest: "/etc/haproxy"

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

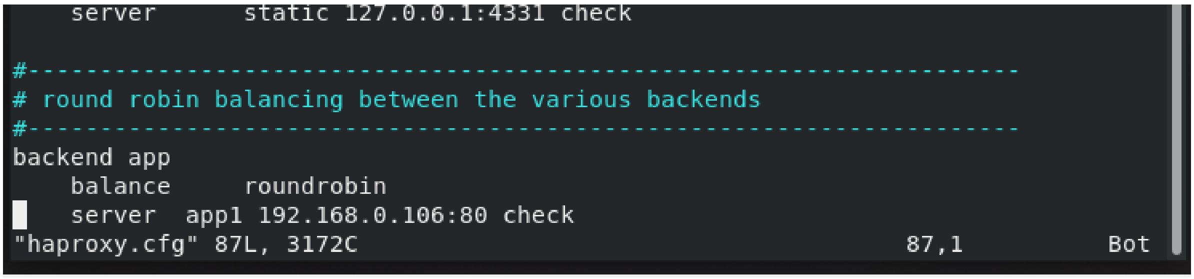
And copying the file of haproxy.cfg in the tasks folder and mention the ip of target node to be configured as webserver



```
[root@localhost 15.2]# cd myhaproxy/tasks
[root@localhost tasks]# ls
haproxy.cfg  main.yml
[root@localhost tasks]# 
```

Vim haproxy.cfg

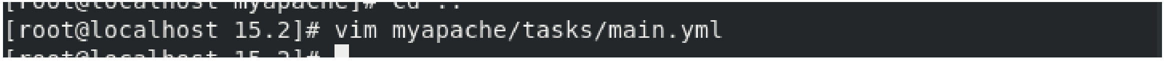
Adding the target node IP to be configured as backend httpd server in the config file.



```
    server      static 127.0.0.1:4331 check

#---------------------------------------------------------------------
# round robin balancing between the various backends
#---------------------------------------------------------------------
backend app
    balance     roundrobin
    server  app1 192.168.0.106:80 check
"haproxy.cfg" 87L, 3172C                                      87,1          Bot
```

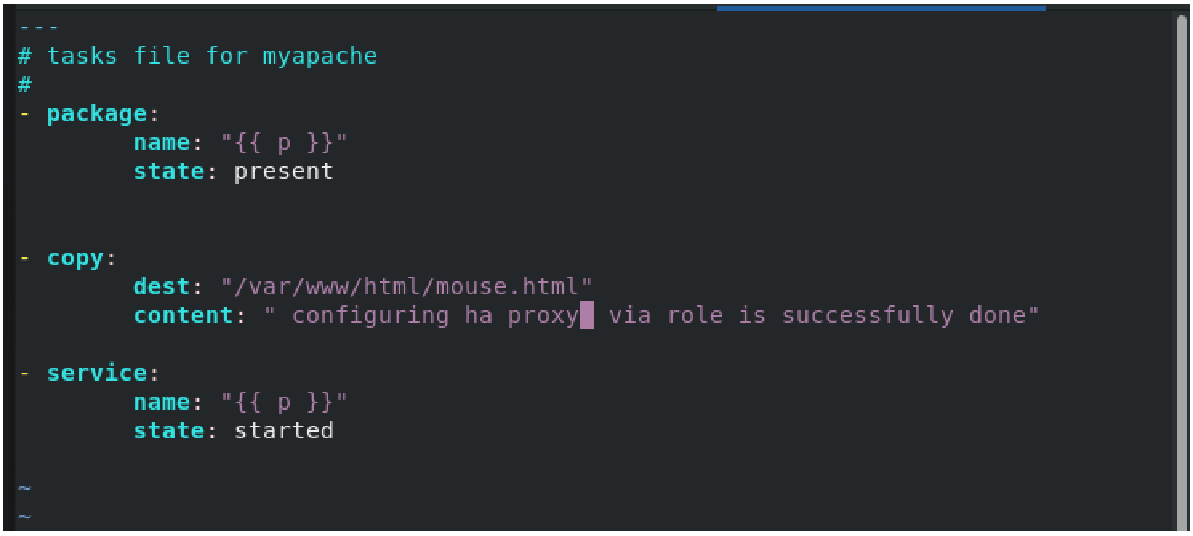
Writing the code in the myapache->tasks->main.yml : actually using the above same code.

Vim myapache/tasks/main.yml



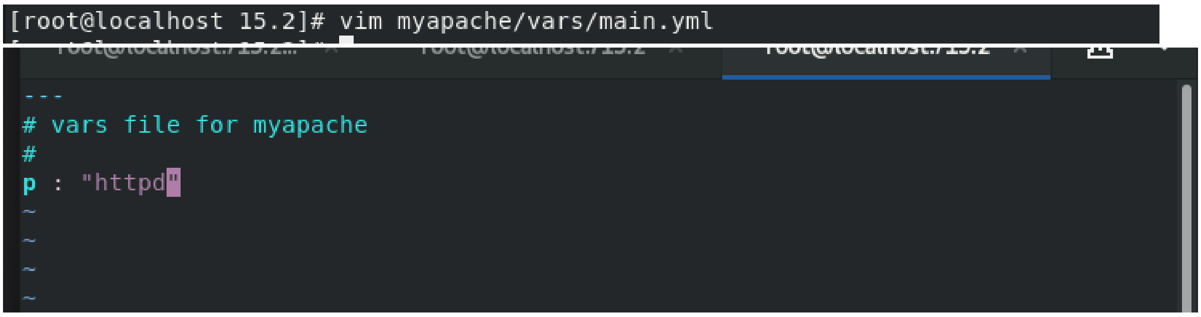
```
[root@localhost myapache]# cd ..
[root@localhost 15.2]# vim myapache/tasks/main.yml
[root@localhost 15.2]# 
```

Task 15.1



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


- copy:
        dest: "/var/www/html/mouse.html"
        content: " configuring ha proxy via role is successfully done"

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

~
~
```

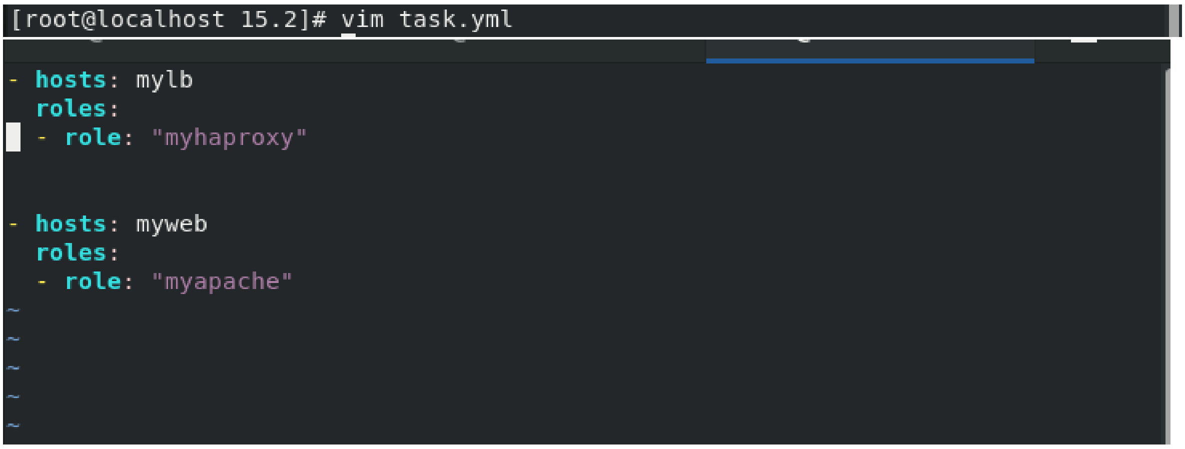
code in vars-> main.yml file



```
[root@localhost 15.2]# vim myapache/vars/main.yml
```

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

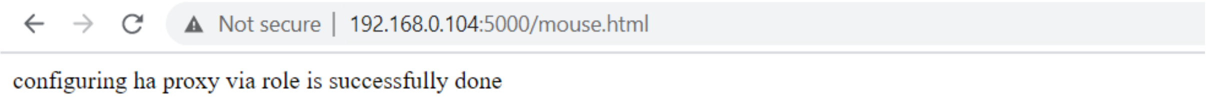
3. Making a new yml file to specify the use of roles.
Vim task.yml



```
[root@localhost 15.2]# vim task.yml
```

```
- hosts: mylb
  roles:
  - role: "myhaproxy"


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

4. Verification
https://IPofLB:portnumber/<htmlfilename>



configuring ha proxy via role is successfully done

Done…………..

Task 15.1