DVS Technologies Aws & Devops

Compiled and Scrutinized by

Mr. Shaan Shaik

(Senior DevOps Lead)

Words To The Students

Though we have taken utmost efforts to present you this book error free, but still it may contain some errors or mistakes. Students are encouraged to bring, if there are any mistakes or errors in this document to our notice. So that it may be rectified in the next edition of this document.

"Suppressing your doubts is Hindering your growth".

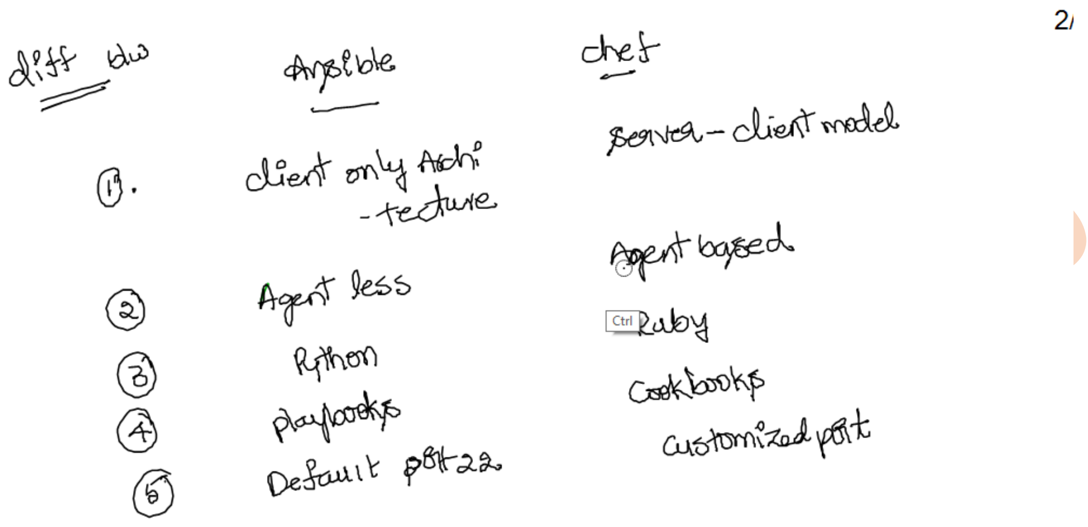
We urge you to work hard and make use of the facilities we are providing to you, because there is no substitute for hard work. We wish you all the best for your future.

"The grass isn't greener on the other side; the grass is greener where you water it."

You and your suggestions are valuable to us; Help us to serve you better. In case of any suggestions, grievance, or complaints, please feel free to write us your suggestions, grievance and feedback on the following

Dvs.training@gmail.com

1. Introduction



diff b/w	Ansible	chef
1.	client only Archi-tecture	Server-client model
2	Agent less	Agent based
3	Python	Ruby
4	playbooks	Cookbooks
5	Default port22	customized port

Complete theory about the background process of ansible execution

What exactly happens in background?

1. Generate a python script
2. Copy the script to respective servers
3. Execute the script on respective servers
4. Wait for the script to complete execution on all hosts

It's Important To note that:

1. Ansible runs each task in parallel across all hosts
2. Ansible waits until all hosts have completed a task befor moving to the next node
3. Ansible runs the tasks in the order that you specify them.

4. To manage a server with ansible, the server needs to have ssh & python 2.5 or later installed or python 2.4 with the python simple json library installed
5. Ansible is an agent less configuration tool which works on ssh port (22)

6. Pushbased: Unlike chef and puppet ansible is push based . It simply executes the playbook from parent/ansible server to its clients. Whereas puppet and ansible are agent based.
7. Ansible scales down: Which means you can define a single host to "n" number of hosts for configuring.

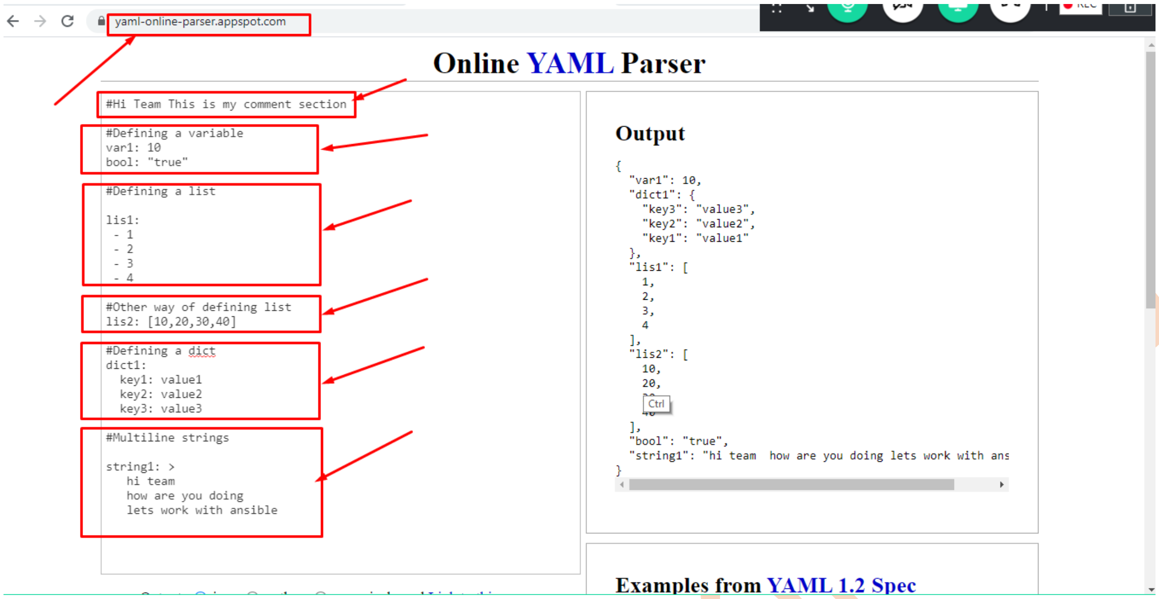
*** Idempotent**

Idempotent in simple: If its installed then it won't do anything, if its not installed it will install it. If any changes are made then it will revert back. Hence these type of behaviour is called Idempotent.

YAML Basics:

Examples For YML :

```
#Hello all this is the comment section


val1: 430

bool1: "Ture"

l1:
  - 1
  - 2
  - 3

l2: [1,2,3,4]


dict1:
   - name: "n1"

dict2:
  - name: "dvs"
  - 1
  - 2
  - 3
  - place: "blr"
  - l1:
     - 1
     - 2
     - 3
```



Online YAML Parser

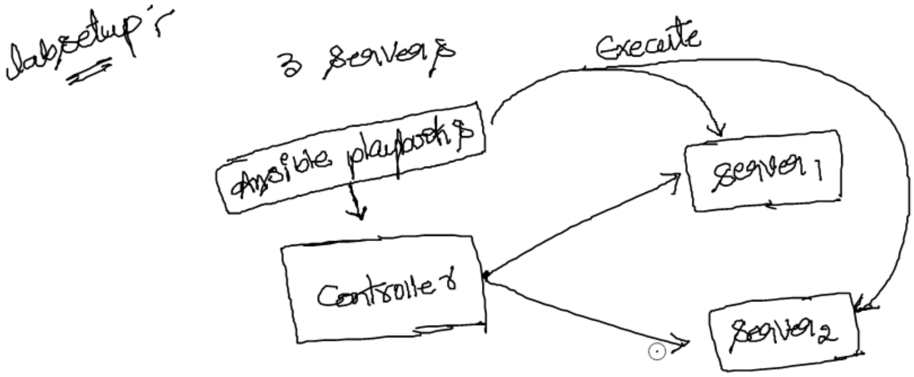
```
#Hi Team This is my comment section

#Defining a variable
var1: 10
bool: "true"

#Defining a list

lis1:
 - 1
 - 2
 - 3
 - 4

#Other way of defining list
lis2: [10,20,30,40]

#Defining a dict
dict1:
   key1: value1
   key2: value2
   key3: value3

#Multiline strings

string1: >
    hi team
    how are you doing
    lets work with ansible
```

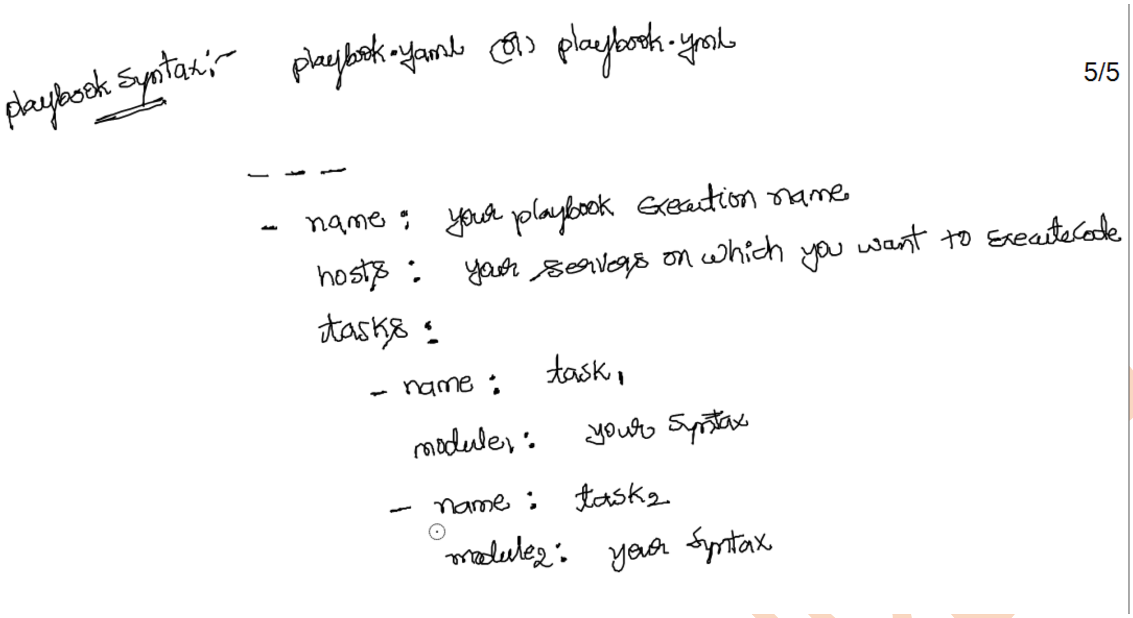
Output

```
{
  "var1": 10,
  "dict1": {
    "key3": "value3",
    "key2": "value2",
    "key1": "value1"
  },
  "lis1": [
    1,
    2,
    3,
    4
  ],
  "lis2": [
    10,
    20,
    30,
    40
  ],
  "bool": "true",
  "string1": "hi team  how are you doing lets work with ans
}
```

Examples from YAML 1.2 Spec

Lab Setup:





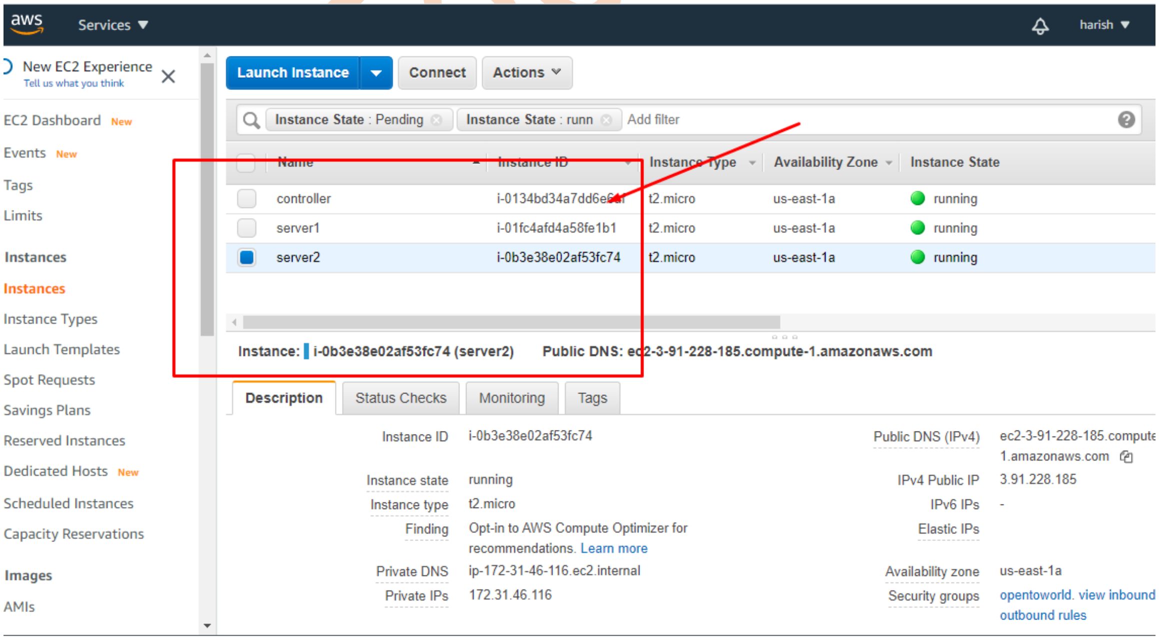
playbook Syntax:- playbook.yaml (or) playbook.yml

5/5

```
---
- name: your playbook Creation name
  hosts: your servers on which you want to Execute code
  tasks:
    - name: task1
      module1: your Syntax
    - name: task2
      module2: your Syntax
```

2. Installation & Configuration of Ansible

Let's spin up three servers 1 controller & 2 servers (slaves)



Instance State : Pending | Instance State : runn

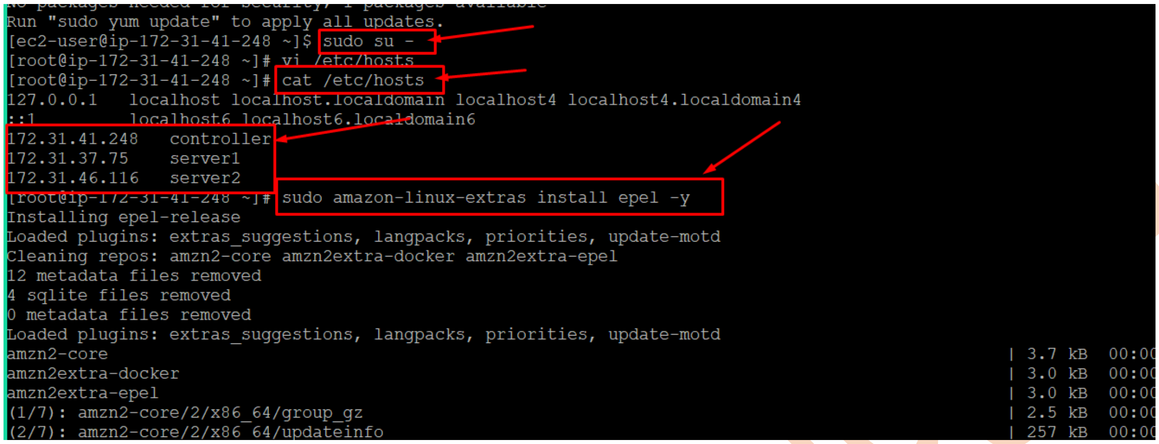
	Name	Instance ID	Instance Type	Availability Zone	Instance State
□	controller	i-0134bd34a7dd6e64	t2.micro	us-east-1a	● running
□	server1	i-01fc4afd4a58fe1b1	t2.micro	us-east-1a	● running
■	server2	i-0b3e38e02af53fc74	t2.micro	us-east-1a	● running

Instance: i-0b3e38e02af53fc74 (server2) Public DNS: ec2-3-91-228-185.compute-1.amazonaws.com

Instance ID	i-0b3e38e02af53fc74	Public DNS (IPv4)	ec2-3-91-228-185.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	3.91.228.185
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-172-31-46-116.ec2.internal	Availability zone	us-east-1a
Private IPs	172.31.46.116	Security groups	opentoworld. view inbound outbound rules

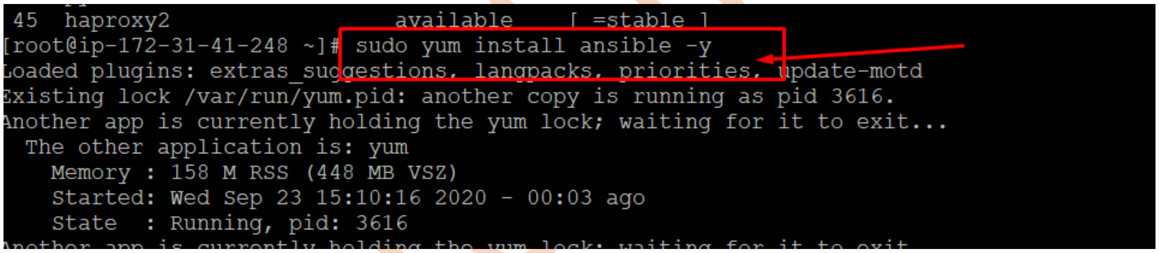
Installing ansible package:

Make sure you are adding the local dns

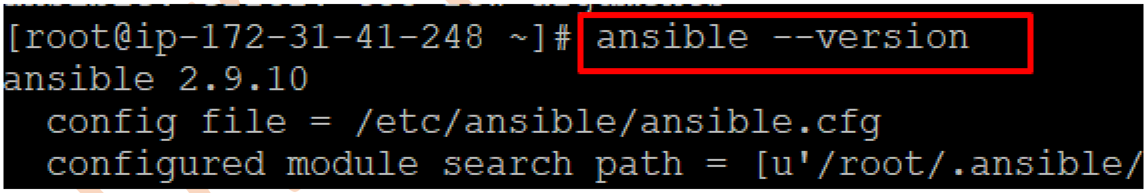


```
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-41-248 ~]$ sudo su -
[root@ip-172-31-41-248 ~]# vi /etc/hosts
[root@ip-172-31-41-248 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6
172.31.41.248   controller
172.31.37.75    server1
172.31.46.116   server2
[root@ip-172-31-41-248 ~]# sudo amazon-linux-extras install epel -y
Installing epel-release
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-epel
12 metadata files removed
4 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                  | 3.7 kB  00:00
amzn2extra-docker                                           | 3.0 kB  00:00
amzn2extra-epel                                             | 3.0 kB  00:00
(1/7): amzn2-core/2/x86_64/group_gz                         | 2.5 kB  00:00
(2/7): amzn2-core/2/x86_64/updateinfo                       | 257 kB  00:00
```

Installing

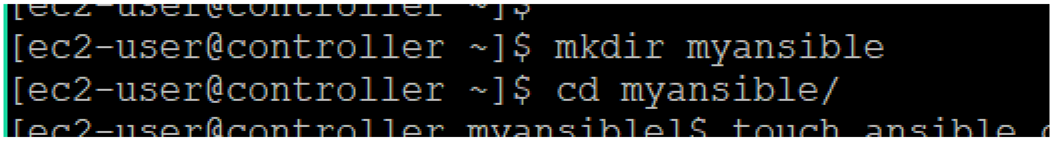


```
 45  haproxy2                 available    [ =stable ]
[root@ip-172-31-41-248 ~]# sudo yum install ansible -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Existing lock /var/run/yum.pid: another copy is running as pid 3616.
Another app is currently holding the yum lock; waiting for it to exit...
  The other application is: yum
    Memory : 158 M RSS (448 MB VSZ)
    Started: Wed Sep 23 15:10:16 2020 - 00:03 ago
    State  : Running, pid: 3616
Another app is currently holding the yum lock; waiting for it to exit...
```

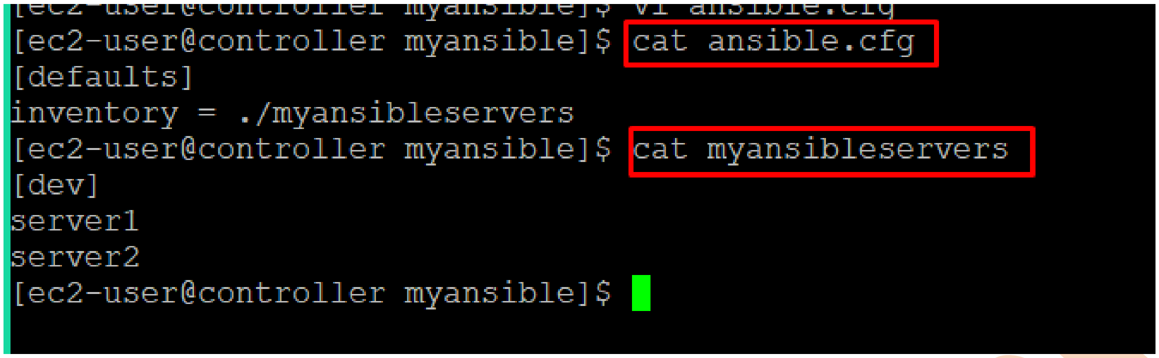


```
[root@ip-172-31-41-248 ~]# ansible --version
ansible 2.9.10
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/
```

Ansible Custom configuration:



```
[ec2-user@controller ~]$
[ec2-user@controller ~]$ mkdir myansible
[ec2-user@controller ~]$ cd myansible/
[ec2-user@controller myansible]$ touch ansible.c
```

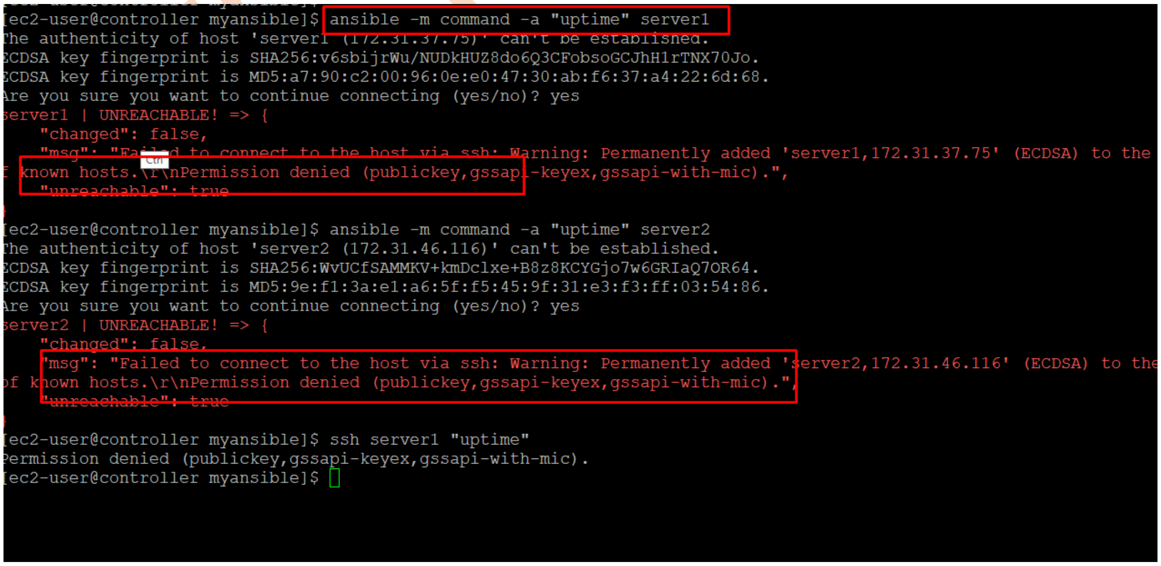


```
[ec2-user@controller myansible]$ vi ansible.cfg
[ec2-user@controller myansible]$ cat ansible.cfg
[defaults]
inventory = ./myansibleservers
[ec2-user@controller myansible]$ cat myansibleservers
[dev]
server1
server2
[ec2-user@controller myansible]$
```

Text Data:

[ec2-user@controller ~]$ mkdir myansible
[ec2-user@controller ~]$ cd myansible/
[ec2-user@controller myansible]$ touch ansible.cfg
[ec2-user@controller myansible]$ vi ansible.cfg
[ec2-user@controller myansible]$ cat ansible.cfg
[defaults]
inventory = ./myansibleservers
[ec2-user@controller myansible]$ touch myansibleservers
[ec2-user@controller myansible]$ vi myansibleservers
[ec2-user@controller myansible]$ cat myansibleservers
[dev]
server1
server2
[ec2-user@controller myansible]$

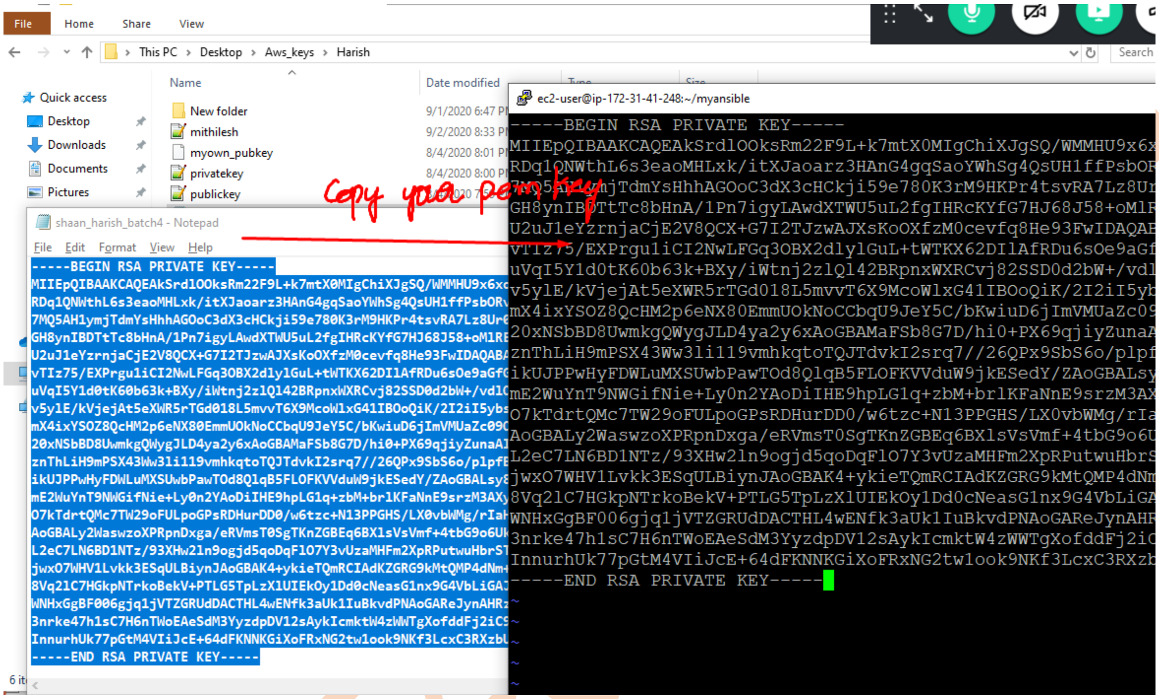
Common Issue with access:



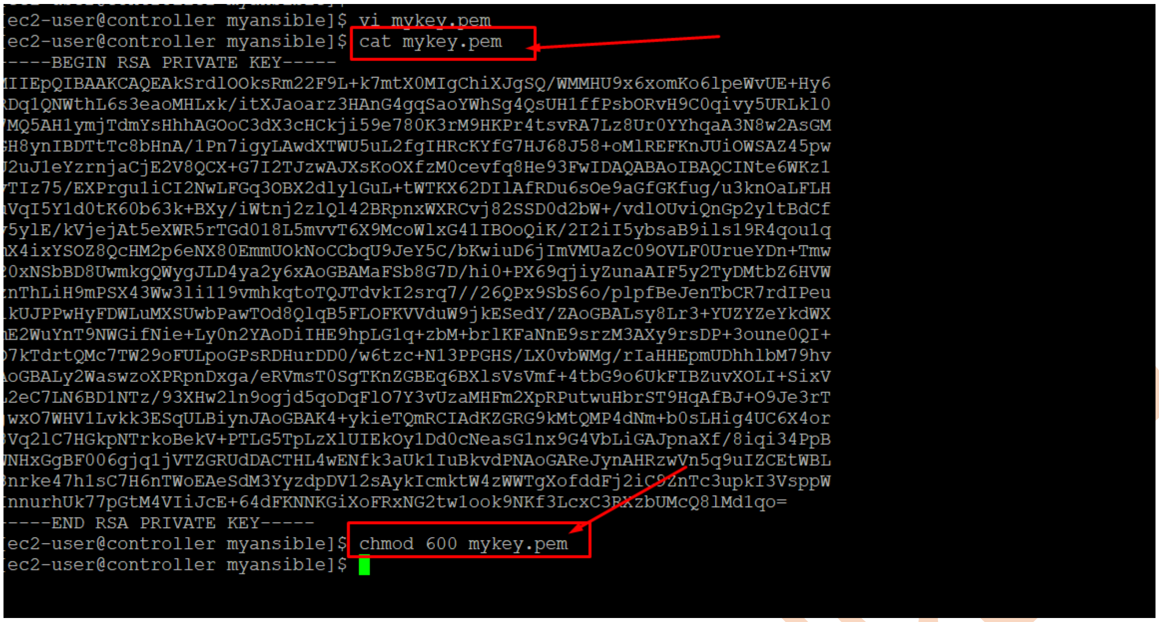
```
[ec2-user@controller myansible]$
[ec2-user@controller myansible]$ ansible -m command -a "uptime" server1
The authenticity of host 'server1 (172.31.37.75)' can't be established.
ECDSA key fingerprint is SHA256:v6sbijrWu/NUDkHUZ8do6Q3CFobsoGCJhH1rTNX70Jo.
ECDSA key fingerprint is MD5:a7:90:c2:00:96:0e:e0:47:30:ab:f6:37:a4:22:6d:68.
Are you sure you want to continue connecting (yes/no)? yes
server1 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Warning: Permanently added 'server1,172.31.37.75' (ECDSA) to the
f known hosts.\r\nPermission denied (publickey,gssapi-keyex,gssapi-with-mic).",
    "unreachable": true
}
[ec2-user@controller myansible]$ ansible -m command -a "uptime" server2
The authenticity of host 'server2 (172.31.46.116)' can't be established.
ECDSA key fingerprint is SHA256:WvUCfSAMMKV+kmDclxe+B8z8KCYGjo7w6GRIaQ7OR64.
ECDSA key fingerprint is MD5:9e:f1:3a:e1:a6:5f:f5:45:9f:31:e3:f3:ff:03:54:86.
Are you sure you want to continue connecting (yes/no)? yes
server2 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Warning: Permanently added 'server2,172.31.46.116' (ECDSA) to the
of known hosts.\r\nPermission denied (publickey,gssapi-keyex,gssapi-with-mic).",
    "unreachable": true
}
[ec2-user@controller myansible]$ ssh server1 "uptime"
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@controller myansible]$
```

Solution:

Add your pem key to the ansible.cfg

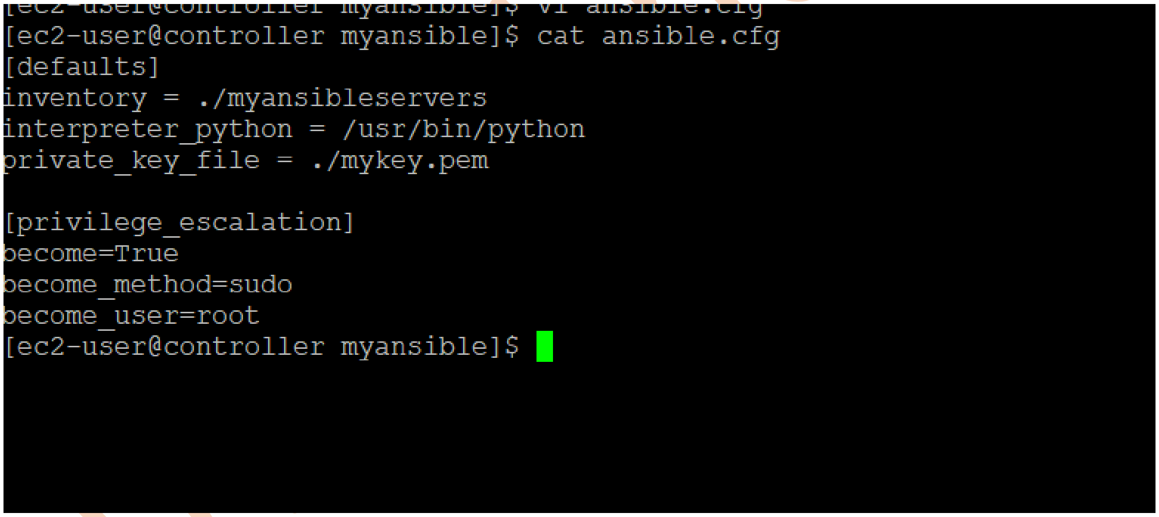


```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAkSrdlOOksRm22F9L+k7mtX0MIgChiXJgSQ/WMMHU9x6x
RDq1QNWthL6s3eaoMHLxk/itXJaoarz3HAnG4gqSaoYWhSg4QsUH1ffPsbOR
7MQ5AH1ymjTdmYsHhhAGOoC3dX3cHCkji59e780K3rM9HKPr4tsvRA7Lz8Ur
GH8ynIBDTtTc8bHnA/1Pn7igyLAwdXTWU5uL2fgIHRcKYfG7HJ68J58+oMlR
U2uJ1eYzrnjaCjE2V8QCX+G7I2TJzwAJXsKoOXfzM0cevfq8He93FwIDAQAB
vTIz75/EXPrguliCI2NwLFGq3OBX2dly1GuL+tWTKX62DIlAfRDu6sOe9aGf
uVqI5Y1d0tK60b63k+BXy/iWtnj2zlQl42BRpnxWXRCvj82SSD0d2bW+/vdl
v5ylE/kVjejAt5eXWR5rTGd018L5mvvT6X9McoWlxG41IBOoQiK/2I2iI5yb
mX4ixYSOZ8QcHM2p6eNX80EmmUOkNoCCbqU9JeY5C/bKwiuD6jImVMUaZc09
20xNSbBD8UwmkgQWygJLD4ya2y6xAoGBAMaFSb8G7D/hi0+PX69qjiyZunaA
znThLiH9mPSX43Ww3li119vmhkqtoTQJTdvkI2srq7//26QPx9SbS6o/plpf
ikUJPPwHyFDWLuMXSUwbPawTOd8QlqB5FLOFKVVduW9jkESedY/ZAoGBALsy
mE2WuYnT9NWGifNie+Ly0n2YAoDiIHE9hpLG1q+zbM+brlKFaNnE9srzM3AX
O7kTdrtQMc7TW29oFULpoGPsRDHurDD0/w6tzc+N13PPGHS/LX0vbWMg/rIa
AoGBALy2WaswzoXPRpnDxga/eRVmsT0SgTKnZGBEq6BXlsVsVmf+4tbG9o6U
L2eC7LN6BD1NTz/93XHw2ln9ogjd5qoDqFlO7Y3vUzaMHFm2XpRPutwuHbrS
jwxO7WHV1Lvkk3ESqULBiynJAoGBAK4+ykieTQmRCIAdKZGRG9kMtQMP4dNm
8Vq2lC7HGkpNTrkoBekV+PTLG5TpLzXlUIEkOy1Dd0cNeasG1nx9G4VbLiGA
WNHxGgBF006gjq1jVTZGRUdDACTHL4wENfk3aUk1IuBkvdPNAoGAReJynAHR
3nrke47h1sC7H6nTWoEAeSdM3YyzdpDV12sAykIcmktW4zWWTgXofddFj2iC
InnurhUk77pGtM4VIiJcE+64dFKNNKGiXoFRxNG2tw1ook9NKf3LcxC3RXzb
-----END RSA PRIVATE KEY-----
~
~
~
~
~
```



```
[ec2-user@controller myansible]$ vi mykey.pem
[ec2-user@controller myansible]$ cat mykey.pem
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAkSrdlOOksRm22F9L+k7mtX0MIgChiXJgSQ/WMMHU9x6xomKo6lpeWvUE+Hy6
RDq1QNWthL6s3eaoMHLxk/itXJaoarz3HAnG4gqSaoYWhSg4QsUH1ffPsbORvH9C0qivy5URLkl0
VMQ5AH1ymjTdmYsHhhAGOoC3dX3cHCkji59e780K3rM9HKPr4tsvRA7Lz8Ur0YYhqaA3N8w2AsGM
GH8ynIBDTtTc8bHnA/1Pn7igyLAwdXTWU5uL2fgIHRcKYfG7HJ68J58+oMlREFKnJUiOWSAZ45pw
J2uJ1eYzrnjaCjE2V8QCX+G7I2TJzwAJXsKoOXfzM0cevfq8He93FwIDAQABAoIBAQCINte6WKz1
vTIz75/EXPrgu1iCI2NwLFGq3OBX2dlylGuL+tWTKX62DIlAfRDu6sOe9aGfGKfug/u3knOaLFLH
aVqI5Y1d0tK60b63k+BXy/iWtnj2zlQl42BRpnxWXRCvj82SSD0d2bW+/vdlOUviQnGp2yltBdCf
y5ylE/kVjejAt5eXWR5rTGd018L5mvvT6X9McoWlxG41IBOoQiK/2I2iI5ybsaB9ils19R4qou1q
nX4ixYSOZ8QcHM2p6eNX80EmmUOkNoCCbqU9JeY5C/bKwiuD6jImVMUaZc09OVLF0UrueYDn+Tmw
20xNSbBD8UwmkgQWygJLD4ya2y6xAoGBAMaFSb8G7D/hi0+PX69qjiyZunaAIF5y2TyDMtbZ6HVW
znThLiH9mPSX43Ww3li119vmhkqtoTQJTdvkI2srq7//26QPx9SbS6o/plpfBeJenTbCR7rdIPeu
lkUJPPwHyFDWLuMXSUwbPawTOd8QlqB5FLOFKVVduW9jkESedY/ZAoGBALsy8Lr3+YUZYZeYkdWX
nE2WuYnT9NWGifNie+Ly0n2YAoDiIHE9hpLG1q+zbM+brlKFaNnE9srzM3AXy9rsDP+3oune0QI+
O7kTdrtQMc7TW29oFULpoGPsRDHurDD0/w6tzc+N13PPGHS/LX0vbWMg/rIaHHEpmUDhhlbM79hv
AoGBALy2WaswzoXPRpnDxga/eRVmsT0SgTKnZGBEq6BXlsVsVmf+4tbG9o6UkFIBZuvXOLI+SixV
L2eC7LN6BD1NTz/93XHw2ln9ogjd5qoDqFlO7Y3vUzaMHFm2XpRPutwuHbrST9HqAfBJ+O9Je3rT
gwxO7WHV1Lvkk3ESqULBiynJAoGBAK4+ykieTQmRCIAdKZGRG9kMtQMP4dNm+b0sLHig4UC6X4or
3Vq2lC7HGkpNTrkoBekV+PTLG5TpLzXlUIEkOy1Dd0cNeasG1nx9G4VbLiGAJpnaXf/8iqi34PpB
WNHxGgBF006gjq1jVTZGRUdDACTHL4wENfk3aUk1IuBkvdPNAoGAReJynAHRzwVn5q9uIZCEtWBL
3nrke47h1sC7H6nTWoEAeSdM3YyzdpDV12sAykIcmktW4zWWTgXofddFj2iC9ZnTc3upkI3VsppW
InnurhUk77pGtM4VIiJcE+64dFKNNKGiXoFRxNG2tw1ook9NKf3LcxC3RXzbUMcQ81Md1qo=
-----END RSA PRIVATE KEY-----
[ec2-user@controller myansible]$ chmod 600 mykey.pem
[ec2-user@controller myansible]$
```

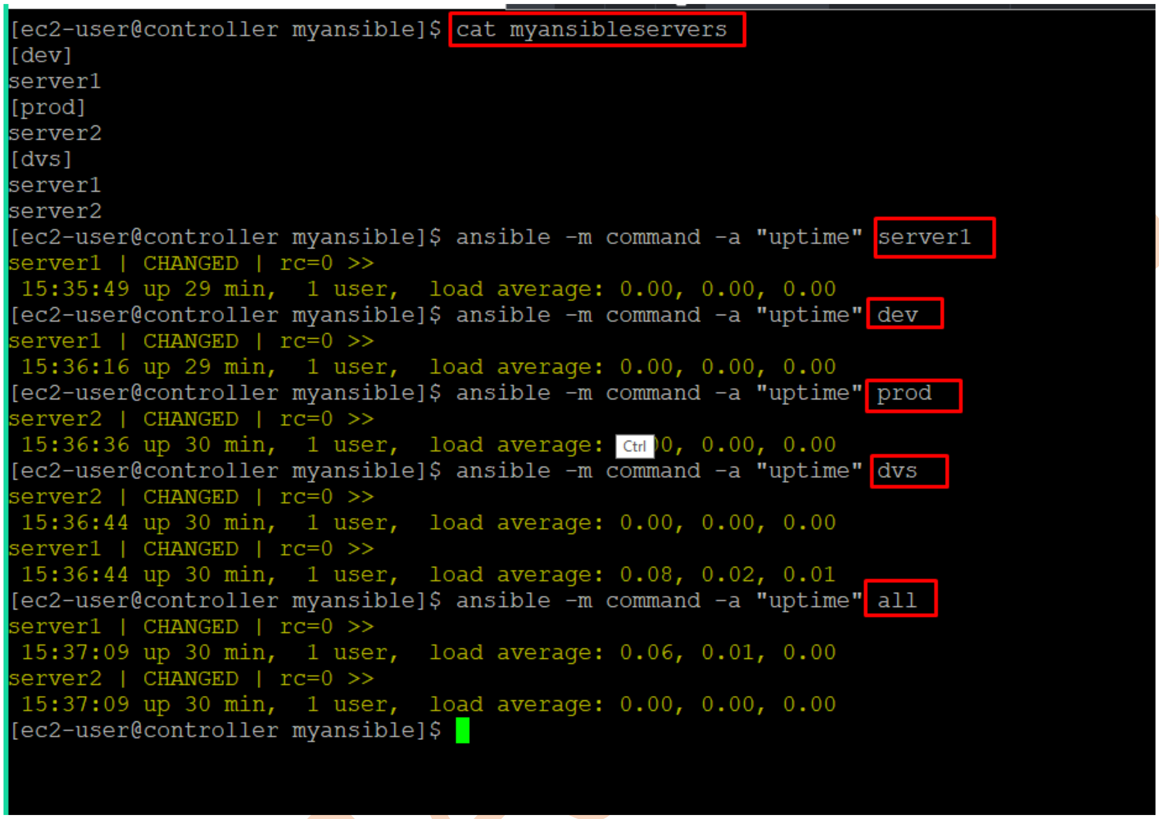
Finally make sure that you are adding the content to the ansible.cfg file as below.



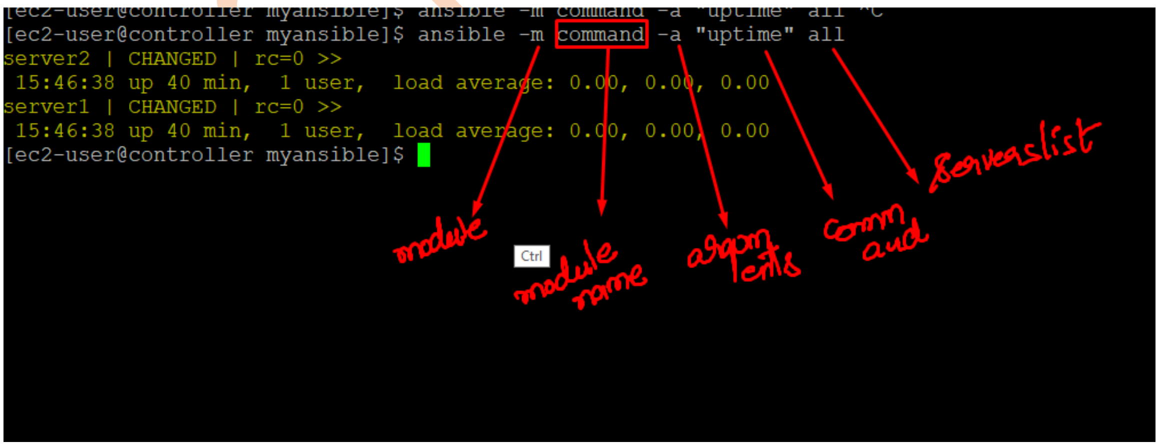
```
[ec2-user@controller myansible]$ vi ansible.cfg
[ec2-user@controller myansible]$ cat ansible.cfg
[defaults]
inventory = ./myansibleservers
interpreter_python = /usr/bin/python
private_key_file = ./mykey.pem

[privilege_escalation]
become=True
become_method=sudo
become_user=root
[ec2-user@controller myansible]$
```

3. Working with Ansible

Executing code in single, group, all servers as part of our ansible hosts file like below.

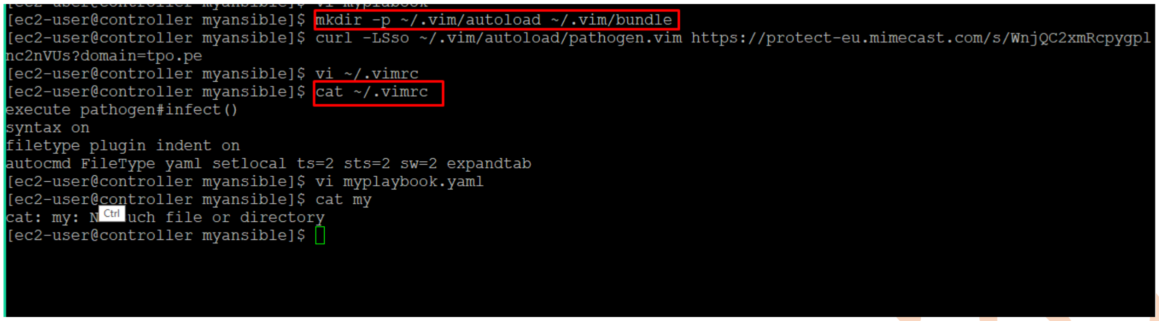


```
[ec2-user@controller myansible]$ cat myansibleservers
[dev]
server1
[prod]
server2
[dvs]
server1
server2
[ec2-user@controller myansible]$ ansible -m command -a "uptime" server1
server1 | CHANGED | rc=0 >>
 15:35:49 up 29 min,  1 user,  load average: 0.00, 0.00, 0.00
[ec2-user@controller myansible]$ ansible -m command -a "uptime" dev
server1 | CHANGED | rc=0 >>
 15:36:16 up 29 min,  1 user,  load average: 0.00, 0.00, 0.00
[ec2-user@controller myansible]$ ansible -m command -a "uptime" prod
server2 | CHANGED | rc=0 >>
 15:36:36 up 30 min,  1 user,  load average: 0.00, 0.00, 0.00
[ec2-user@controller myansible]$ ansible -m command -a "uptime" dvs
server2 | CHANGED | rc=0 >>
 15:36:44 up 30 min,  1 user,  load average: 0.00, 0.00, 0.00
server1 | CHANGED | rc=0 >>
 15:36:44 up 30 min,  1 user,  load average: 0.08, 0.02, 0.01
[ec2-user@controller myansible]$ ansible -m command -a "uptime" all
server1 | CHANGED | rc=0 >>
 15:37:09 up 30 min,  1 user,  load average: 0.06, 0.01, 0.00
server2 | CHANGED | rc=0 >>
 15:37:09 up 30 min,  1 user,  load average: 0.00, 0.00, 0.00
[ec2-user@controller myansible]$
```



```
[ec2-user@controller myansible]$ ansible -m command -a "uptime" all ^C
[ec2-user@controller myansible]$ ansible -m command -a "uptime" all
server2 | CHANGED | rc=0 >>
 15:46:38 up 40 min,  1 user,  load average: 0.00, 0.00, 0.00
server1 | CHANGED | rc=0 >>
 15:46:38 up 40 min,  1 user,  load average: 0.00, 0.00, 0.00
[ec2-user@controller myansible]$
```

Re-modifying vim editor as per out requirement:



```
[ec2-user@controller myansible]$ vi myplaybook
[ec2-user@controller myansible]$ mkdir -p ~/.vim/autoload ~/.vim/bundle
[ec2-user@controller myansible]$ curl -LSso ~/.vim/autoload/pathogen.vim https://protect-eu.mimecast.com/s/WnjQC2xmRcpygpl
nc2nVUs?domain=tpo.pe
[ec2-user@controller myansible]$ vi ~/.vimrc
[ec2-user@controller myansible]$ cat ~/.vimrc
execute pathogen#infect()
syntax on
filetype plugin indent on
autocmd FileType yaml setlocal ts=2 sts=2 sw=2 expandtab
[ec2-user@controller myansible]$ vi myplaybook.yaml
[ec2-user@controller myansible]$ cat my
cat: my: N uch file or directory
[ec2-user@controller myansible]$
```

Configuring VIM as per our requirement ::

highlight-ansible-yaml-and-jinja-syntax-in-vim-editor ::

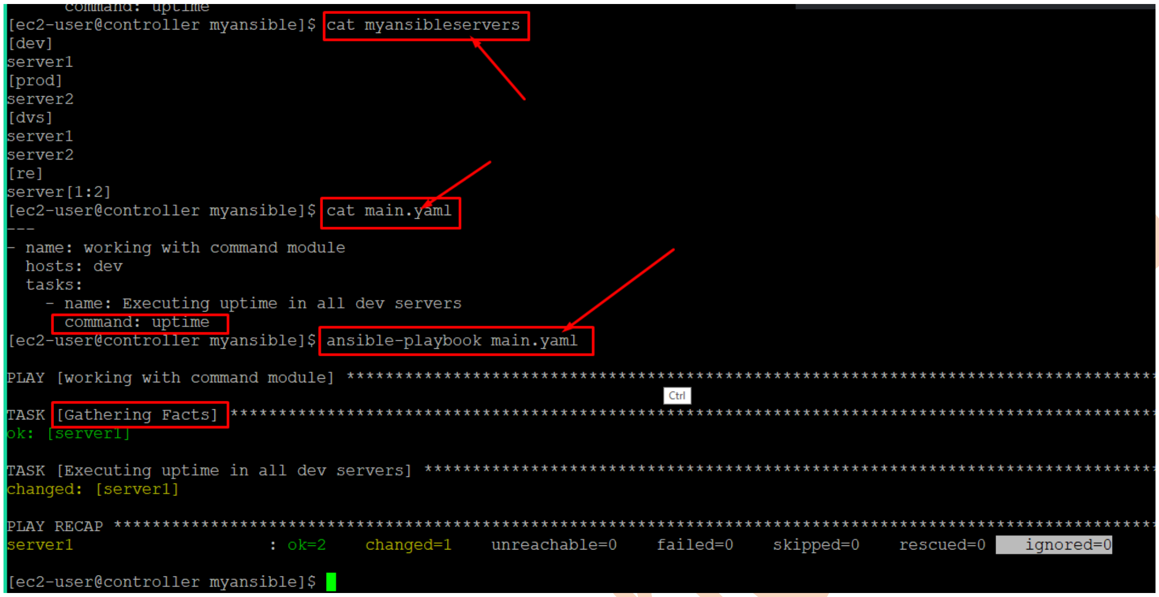
mkdir -p ~/.vim/autoload ~/.vim/bundle
curl -LSso ~/.vim/autoload/pathogen.vim https://protect-eu.mimecast.com/s/WnjQC2xmRcpygplnc2nVUs?domain=tpo.pe

Now add the following lines to vim ~/.vimrc to activate this and start autoloading bundles.

[root@slave1 ansible]# vim ~/.vimrc
execute pathogen#infect()
syntax on
filetype plugin indent on
autocmd FileType yaml setlocal ts=2 sts=2 sw=2 expandtab

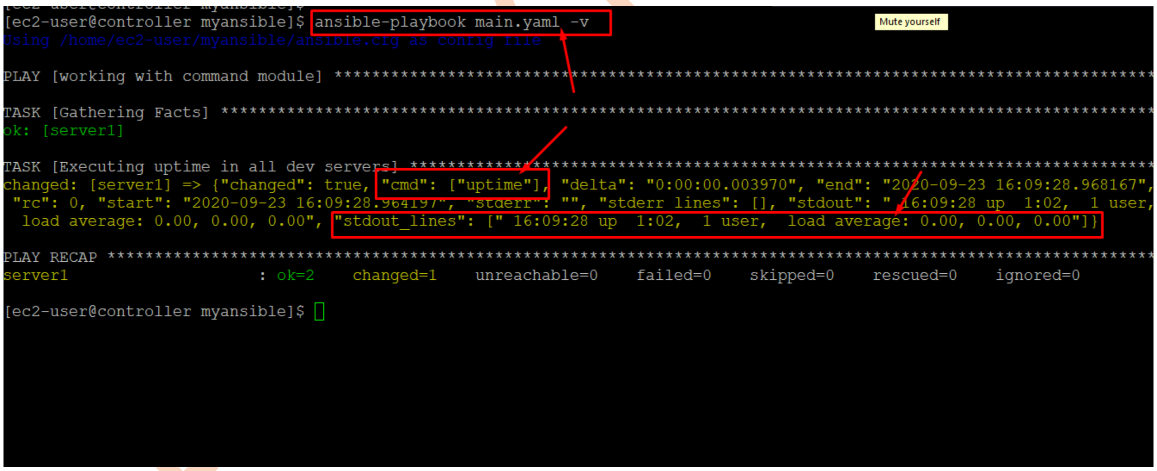
4. Working with Modules

Command Module:



```
      command: uptime
[ec2-user@controller myansible]$ cat myansibleservers
[dev]
server1
[prod]
server2
[dvs]
server1
server2
[re]
server[1:2]
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with command module
  hosts: dev
  tasks:
    - name: Executing uptime in all dev servers
      command: uptime
[ec2-user@controller myansible]$ ansible-playbook main.yaml

PLAY [working with command module] ****************************************************************

TASK [Gathering Facts] ****************************************************************************
ok: [server1]

TASK [Executing uptime in all dev servers] ********************************************************
changed: [server1]

PLAY RECAP ****************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

If you want to get the output of your playbook execution you can use "-v" option

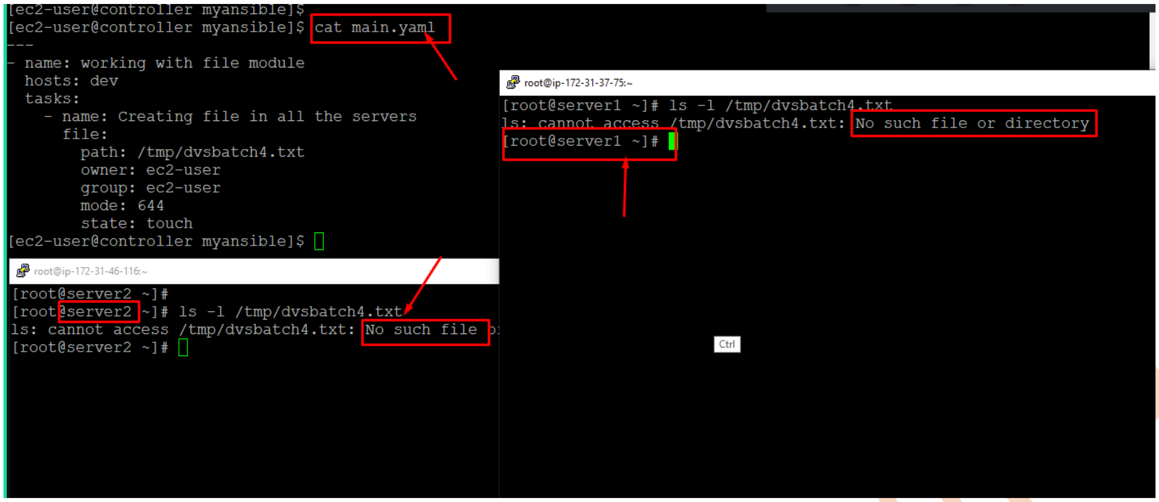


```
[ec2-user@controller myansible]$
[ec2-user@controller myansible]$ ansible-playbook main.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with command module] ****************************************************************

TASK [Gathering Facts] ****************************************************************************
ok: [server1]

TASK [Executing uptime in all dev servers] ********************************************************
changed: [server1] => {"changed": true, "cmd": ["uptime"], "delta": "0:00:00.003970", "end": "2020-09-23 16:09:28.968167",
 "rc": 0, "start": "2020-09-23 16:09:28.964197", "stderr": "", "stderr_lines": [], "stdout": " 16:09:28 up  1:02,  1 user,
  load average: 0.00, 0.00, 0.00", "stdout_lines": [" 16:09:28 up  1:02,  1 user,  load average: 0.00, 0.00, 0.00"]}

PLAY RECAP ****************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

Note: -v stands for verbose

File Module:

Before playbook execution:



```
[ec2-user@controller myansible]$
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with file module
  hosts: dev
  tasks:
    - name: Creating file in all the servers
      file:
        path: /tmp/dvsbatch4.txt
        owner: ec2-user
        group: ec2-user
        mode: 644
        state: touch
[ec2-user@controller myansible]$
```

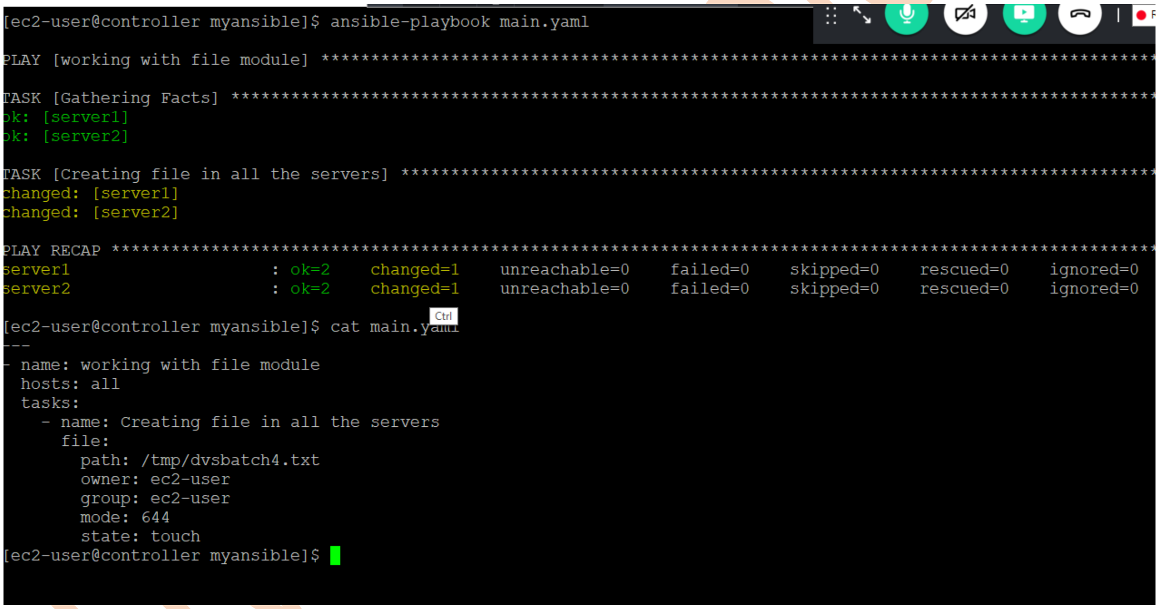
root@ip-172-31-46-116:~

```
[root@server2 ~]#
[root@server2 ~]# ls -l /tmp/dvsbatch4.txt
ls: cannot access /tmp/dvsbatch4.txt: No such file o
[root@server2 ~]#
```

root@ip-172-31-37-75:~

```
[root@server1 ~]# ls -l /tmp/dvsbatch4.txt
ls: cannot access /tmp/dvsbatch4.txt: No such file or directory
[root@server1 ~]#
```

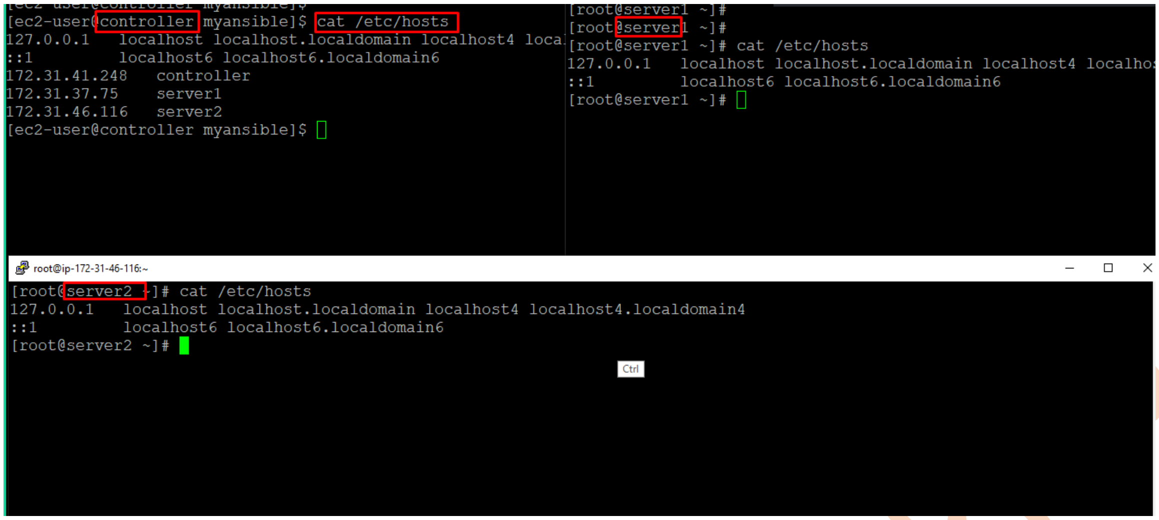
Post playbook execution:



```
[ec2-user@controller myansible]$ ansible-playbook main.yaml

PLAY [working with file module] ***********************************************************************

TASK [Gathering Facts] ********************************************************************************
ok: [server1]
ok: [server2]

TASK [Creating file in all the servers] ***************************************************************
changed: [server1]
changed: [server2]

PLAY RECAP ********************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with file module
  hosts: all
  tasks:
    - name: Creating file in all the servers
      file:
        path: /tmp/dvsbatch4.txt
        owner: ec2-user
        group: ec2-user
        mode: 644
        state: touch
[ec2-user@controller myansible]$
```

Copy Module:

Before Execution:

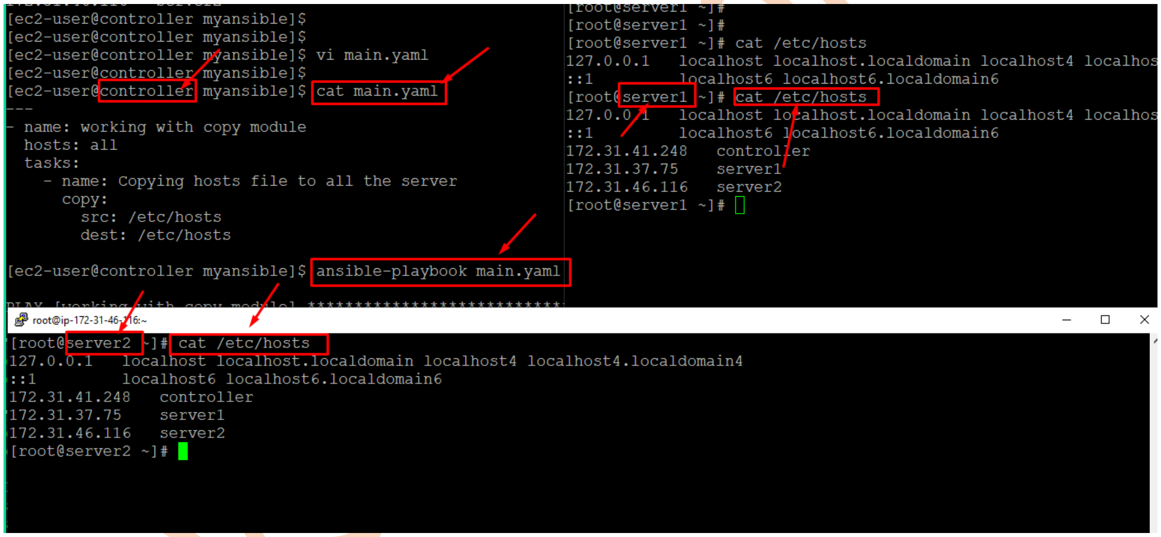


```
[ec2-user@controller myansible]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 loca
::1         localhost6 localhost6.localdomain6
172.31.41.248   controller
172.31.37.75    server1
172.31.46.116   server2
[ec2-user@controller myansible]$
```

```
[root@server1 ~]#
[root@server1 ~]#
[root@server1 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localho
::1         localhost6 localhost6.localdomain6
[root@server1 ~]#
```

```
[root@server2 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6
[root@server2 ~]#
```

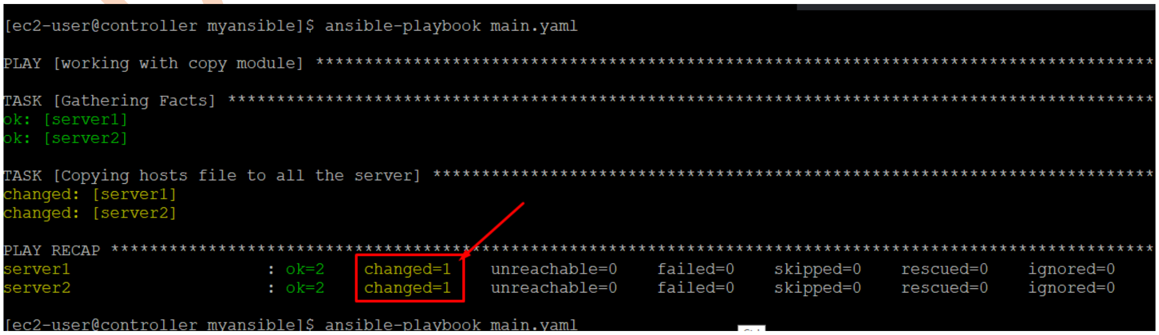
Code:



```
[ec2-user@controller myansible]$
[ec2-user@controller myansible]$
[ec2-user@controller myansible]$ vi main.yaml
[ec2-user@controller myansible]$
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with copy module
  hosts: all
  tasks:
    - name: Copying hosts file to all the server
      copy:
        src: /etc/hosts
        dest: /etc/hosts

[ec2-user@controller myansible]$ ansible-playbook main.yaml
```

```
[root@server1 ~]#
[root@server1 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhos
::1         localhost6 localhost6.localdomain6
[root@server1 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhos
::1         localhost6 localhost6.localdomain6
172.31.41.248   controller
172.31.37.75    server1
172.31.46.116   server2
[root@server1 ~]#
```

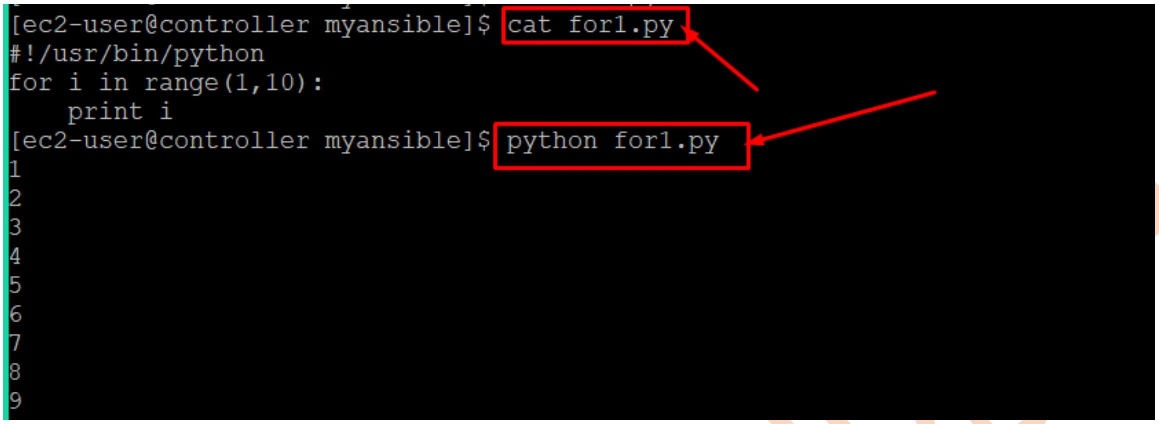
```
[root@server2 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6
172.31.41.248   controller
172.31.37.75    server1
172.31.46.116   server2
[root@server2 ~]#
```



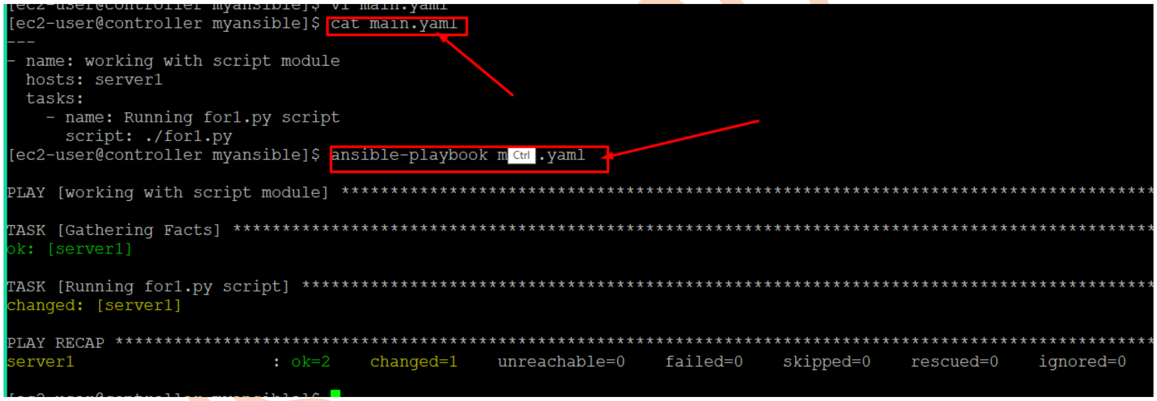
```
[ec2-user@controller myansible]$ ansible-playbook main.yaml

PLAY [working with copy module] ***********************************************************************

TASK [Gathering Facts] ********************************************************************************
ok: [server1]
ok: [server2]

TASK [Copying hosts file to all the server] ***********************************************************
changed: [server1]
changed: [server2]

PLAY RECAP ********************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ ansible-playbook main.yaml
```

Script Module:

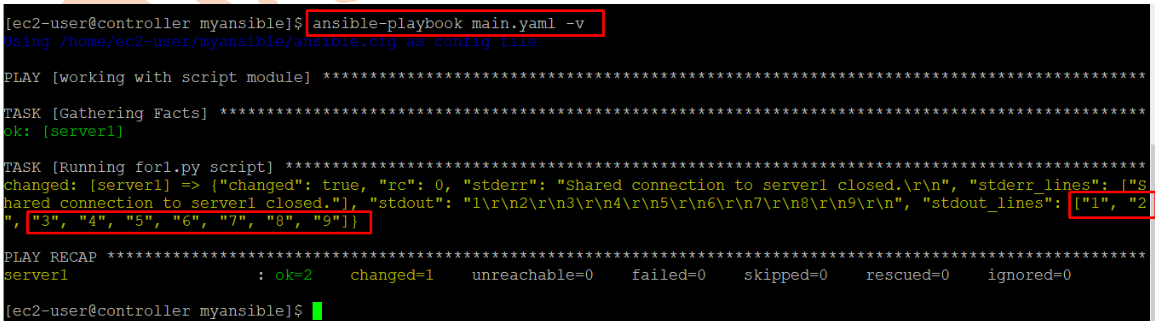
Executing python script via ansible



```
[ec2-user@controller myansible]$ cat for1.py
#!/usr/bin/python
for i in range(1,10):
    print i
[ec2-user@controller myansible]$ python for1.py
1
2
3
4
5
6
7
8
9
```

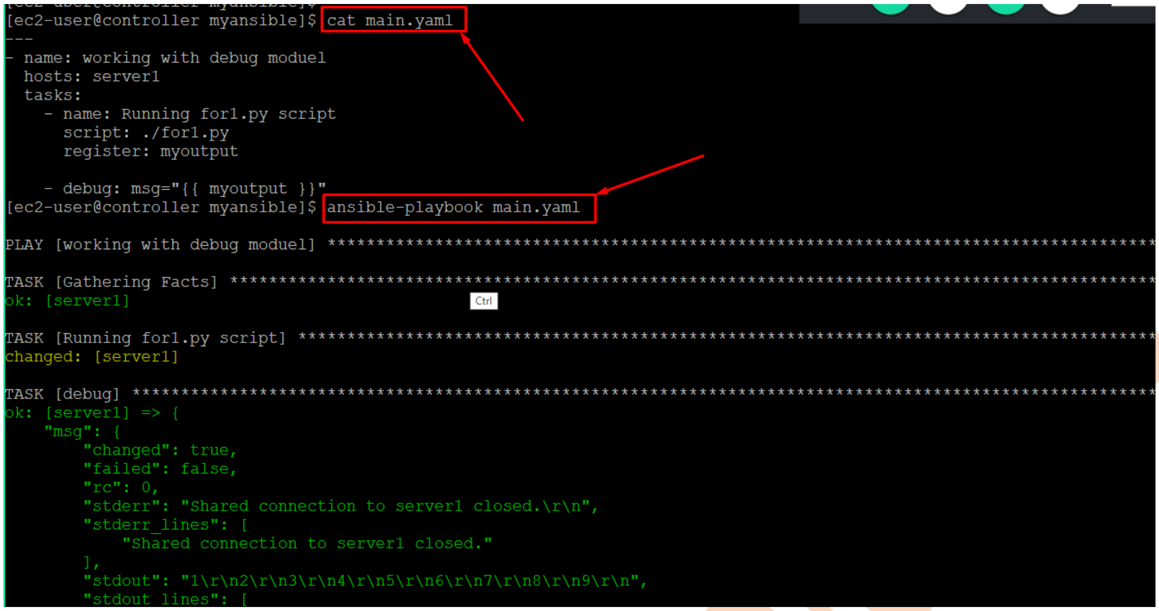


```
[ec2-user@controller myansible]$ vi main.yaml
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with script module
  hosts: server1
  tasks:
    - name: Running for1.py script
      script: ./for1.py
[ec2-user@controller myansible]$ ansible-playbook m    .yaml

PLAY [working with script module] *********************************************************************

TASK [Gathering Facts] *******************************************************************************
ok: [server1]

TASK [Running for1.py script] ************************************************************************
changed: [server1]

PLAY RECAP *******************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

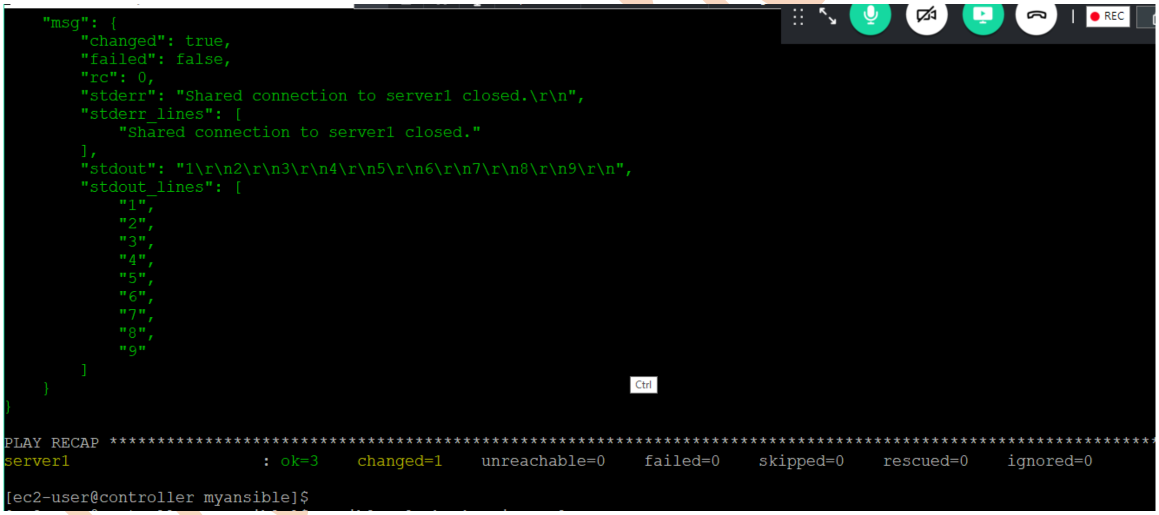


```
[ec2-user@controller myansible]$ ansible-playbook main.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with script module] *********************************************************************

TASK [Gathering Facts] *******************************************************************************
ok: [server1]

TASK [Running for1.py script] ************************************************************************
changed: [server1] => {"changed": true, "rc": 0, "stderr": "Shared connection to server1 closed.\r\n", "stderr_lines": ["S
hared connection to server1 closed."], "stdout": "1\r\n2\r\n3\r\n4\r\n5\r\n6\r\n7\r\n8\r\n9\r\n", "stdout_lines": ["1", "2
", "3", "4", "5", "6", "7", "8", "9"]}

PLAY RECAP *******************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

Debug Module:

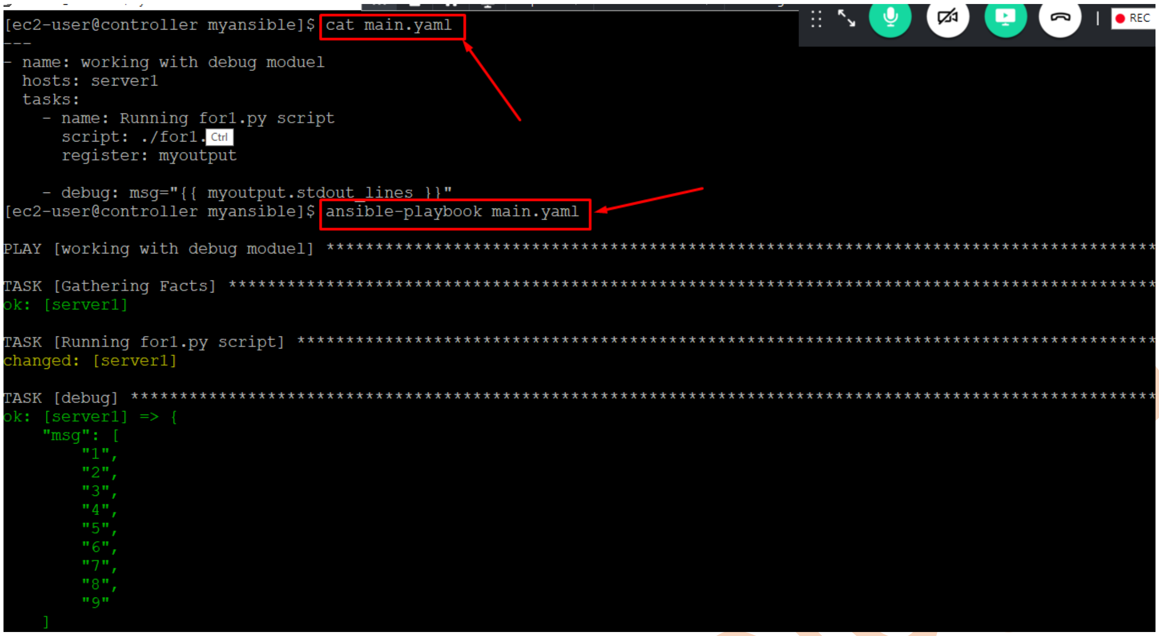


```
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with debug moduel
  hosts: server1
  tasks:
    - name: Running for1.py script
      script: ./for1.py
      register: myoutput

    - debug: msg="{{ myoutput }}"
[ec2-user@controller myansible]$ ansible-playbook main.yaml

PLAY [working with debug moduel] ******************************************************************************

TASK [Gathering Facts] ****************************************************************************************
ok: [server1]

TASK [Running for1.py script] *********************************************************************************
changed: [server1]

TASK [debug] **************************************************************************************************
ok: [server1] => {
    "msg": {
        "changed": true,
        "failed": false,
        "rc": 0,
        "stderr": "Shared connection to server1 closed.\r\n",
        "stderr_lines": [
            "Shared connection to server1 closed."
        ],
        "stdout": "1\r\n2\r\n3\r\n4\r\n5\r\n6\r\n7\r\n8\r\n9\r\n",
        "stdout_lines": [
```

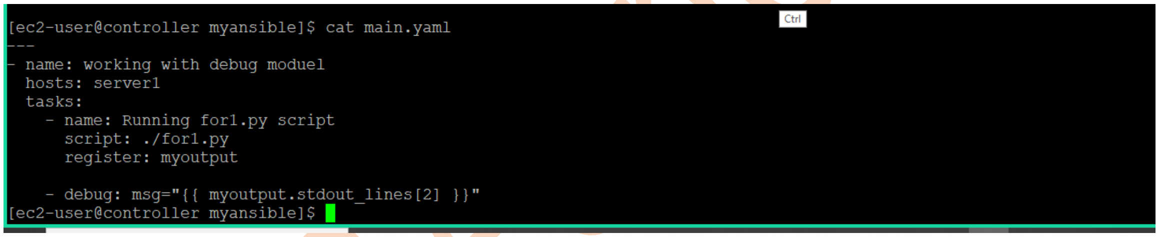


```
    "msg": {
        "changed": true,
        "failed": false,
        "rc": 0,
        "stderr": "Shared connection to server1 closed.\r\n",
        "stderr_lines": [
            "Shared connection to server1 closed."
        ],
        "stdout": "1\r\n2\r\n3\r\n4\r\n5\r\n6\r\n7\r\n8\r\n9\r\n",
        "stdout_lines": [
            "1",
            "2",
            "3",
            "4",
            "5",
            "6",
            "7",
            "8",
            "9"
        ]
    }
}

PLAY RECAP ****************************************************************************************************
server1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

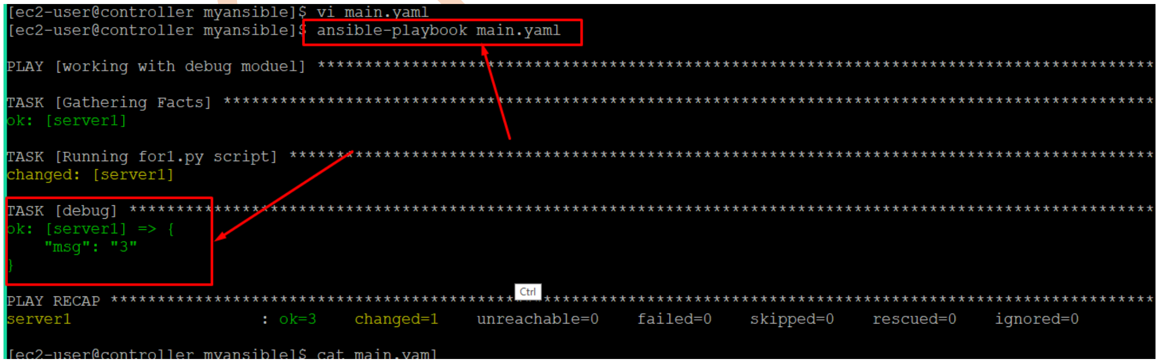
Let's stream line the output:



```
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with debug moduel
  hosts: server1
  tasks:
    - name: Running for1.py script
      script: ./for1.
      register: myoutput

    - debug: msg="{{ myoutput.stdout_lines }}"
[ec2-user@controller myansible]$ ansible-playbook main.yaml

PLAY [working with debug moduel] ***********************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server1]

TASK [Running for1.py script] **************************************************************************
changed: [server1]

TASK [debug] *******************************************************************************************
ok: [server1] => {
    "msg": [
        "1",
        "2",
        "3",
        "4",
        "5",
        "6",
        "7",
        "8",
        "9"
    ]
```



```
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with debug moduel
  hosts: server1
  tasks:
    - name: Running for1.py script
      script: ./for1.py
      register: myoutput

    - debug: msg="{{ myoutput.stdout_lines[2] }}"
[ec2-user@controller myansible]$
```



```
[ec2-user@controller myansible]$ vi main.yaml
[ec2-user@controller myansible]$ ansible-playbook main.yaml

PLAY [working with debug moduel] ***********************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server1]

TASK [Running for1.py script] **************************************************************************
changed: [server1]

TASK [debug] *******************************************************************************************
ok: [server1] => {
    "msg": "3"
}

PLAY RECAP *********************************************************************************************
server1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ cat main.yaml
```

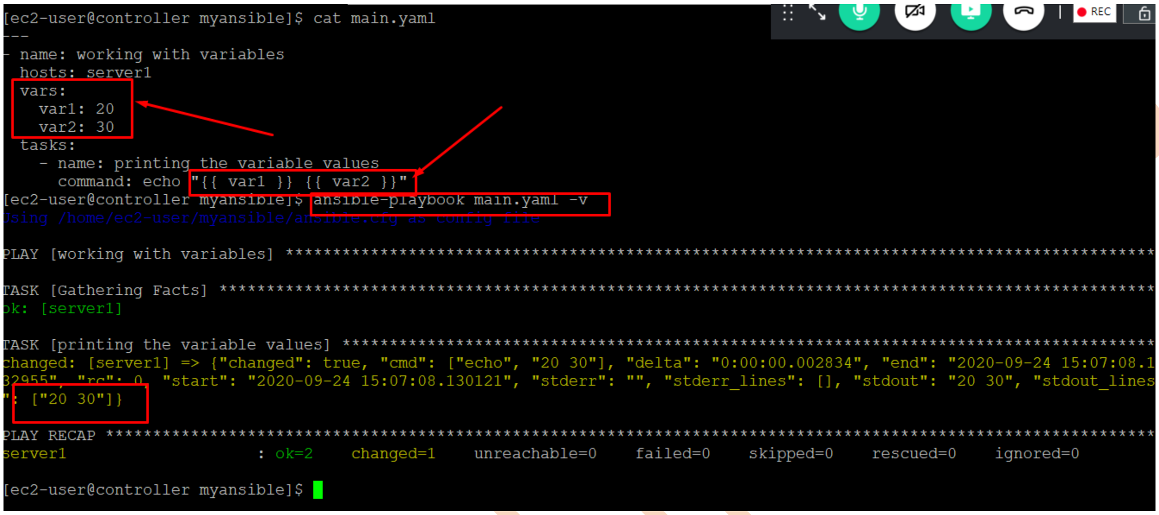
Final Code:

```
[ec2-user@master ansible]$ cat modules.yml
---
- name: working with playbooks
  hosts: all
  tasks:
    - name: working with command module
      shell: "uptime;df -hT"

    - name: working with file module
      file:
        path: /etc/foo.conf
        owner: root
        mode: '0644'
        state: touch

    - name: working with copy module
      copy:
        src: /etc/hosts
        dest: /etc/hosts
        owner: root
        group: root
        mode: '0644'

    - name: working with template
      template:
        src: /etc/passwd
        dest: /tmp/dvsbatch/
        force: no

    - name: working with script
      script: ./for1.py
      register: script_output

    - debug: msg="{{script_output.stdout_lines[0]}}"
```

5. Variables

We have many ways of variable declarations few of them are as below

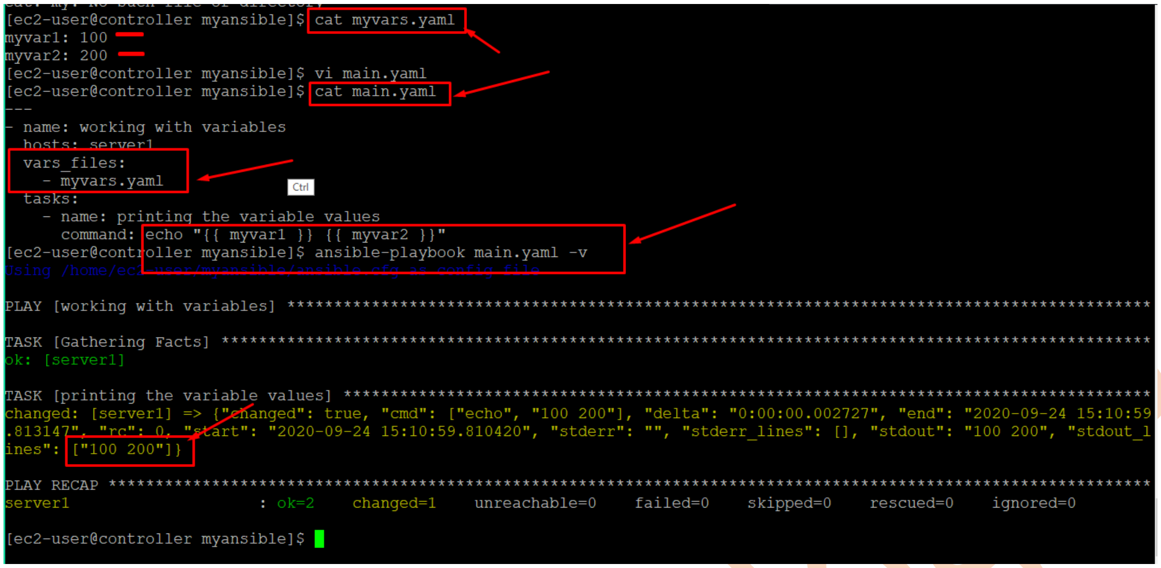
vars section:



```
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with variables
  hosts: server1
  vars:
    var1: 20
    var2: 30
  tasks:
    - name: printing the variable values
      command: echo "{{ var1 }} {{ var2 }}"
[ec2-user@controller myansible]$ ansible-playbook main.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with variables] ***********************************************************************************

TASK [Gathering Facts] ******************************************************************************************
ok: [server1]

TASK [printing the variable values] *****************************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "20 30"], "delta": "0:00:00.002834", "end": "2020-09-24 15:07:08.1
32955", "rc": 0, "start": "2020-09-24 15:07:08.130121", "stderr": "", "stderr_lines": [], "stdout": "20 30", "stdout_lines
": ["20 30"]}

PLAY RECAP ******************************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

```
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with variables
  hosts: server1
  vars:
    var1: 20
    var2: 30
  tasks:
    - name: printing the variable values
      command: echo "{{ var1 }} {{ var2 }}"
```

vars_files:

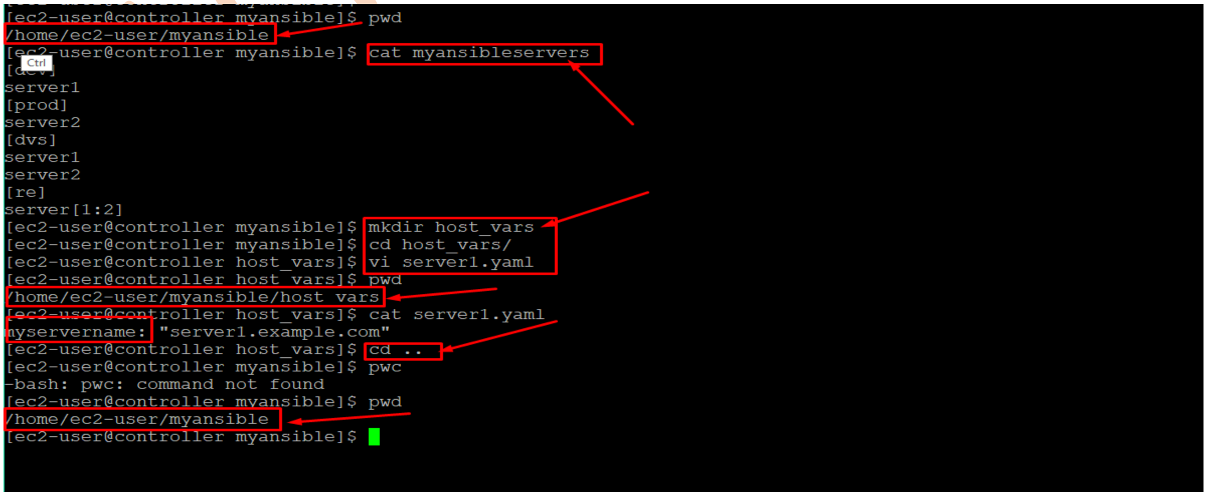


```
cat: my: No such file or directory
[ec2-user@controller myansible]$ cat myvars.yaml
myvar1: 100
myvar2: 200
[ec2-user@controller myansible]$ vi main.yaml
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with variables
  hosts: server1
  vars_files:
    - myvars.yaml
  tasks:
    - name: printing the variable values
      command: echo "{{ myvar1 }} {{ myvar2 }}"
[ec2-user@controller myansible]$ ansible-playbook main.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with variables] ***************************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server1]

TASK [printing the variable values] *********************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "100 200"], "delta": "0:00:00.002727", "end": "2020-09-24 15:10:59
.813147", "rc": 0, "start": "2020-09-24 15:10:59.810420", "stderr": "", "stderr_lines": [], "stdout": "100 200", "stdout_l
ines": ["100 200"]}

PLAY RECAP **********************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

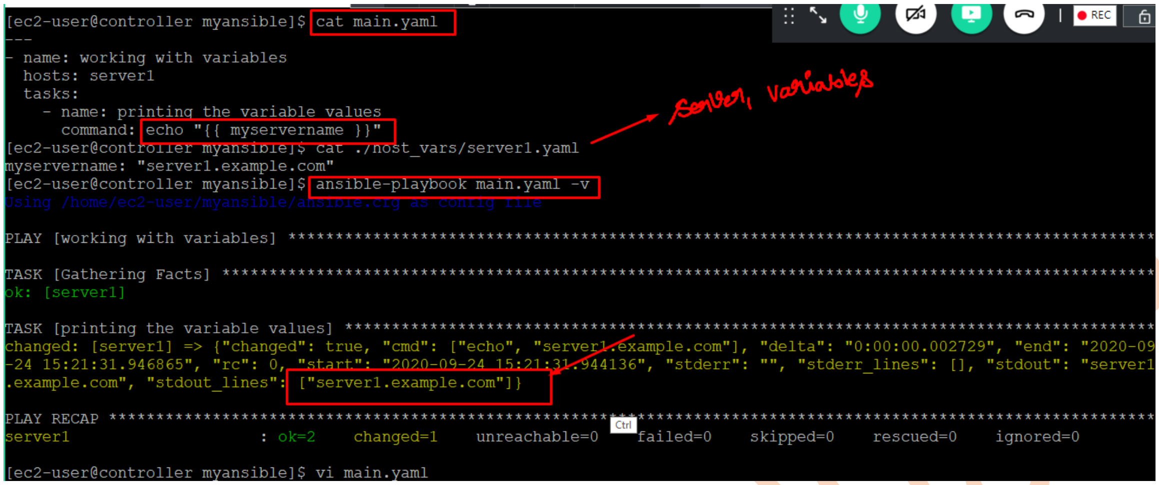
[ec2-user@controller myansible]$ cat main.yaml

```
---
- name: working with variables
  hosts: server1
  vars_files:
    - myvars.yaml
  tasks:
    - name: printing the variable values
      command: echo "{{ myvar1 }} {{ myvar2 }}"
```

host_vars Section:

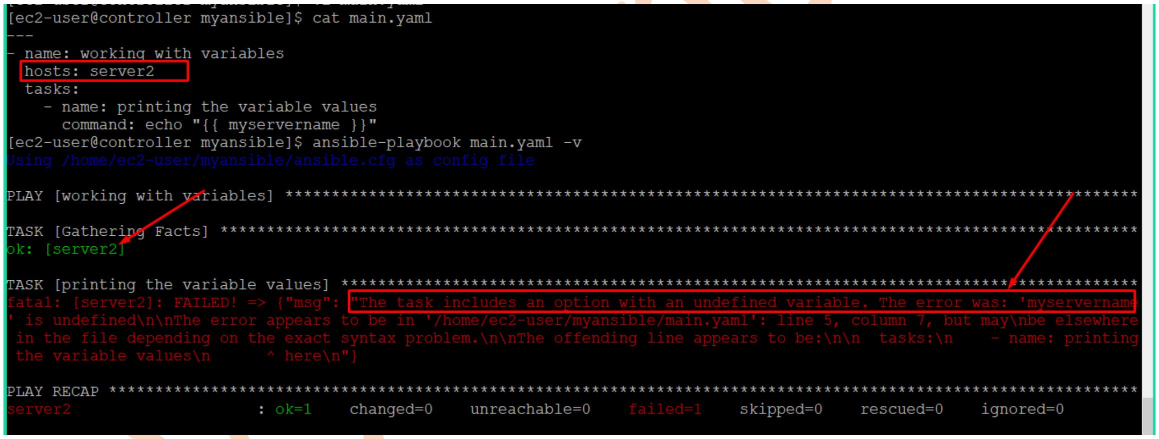


```
[ec2-user@controller myansible]$ pwd
/home/ec2-user/myansible
[ec2-user@controller myansible]$ cat myansibleservers
[dev]
server1
[prod]
server2
[dvs]
server1
server2
[re]
server[1:2]
[ec2-user@controller myansible]$ mkdir host_vars
[ec2-user@controller myansible]$ cd host_vars/
[ec2-user@controller host_vars]$ vi server1.yaml
[ec2-user@controller host_vars]$ pwd
/home/ec2-user/myansible/host_vars
[ec2-user@controller host_vars]$ cat server1.yaml
myservername: "server1.example.com"
[ec2-user@controller host_vars]$ cd ..
[ec2-user@controller myansible]$ pwc
-bash: pwc: command not found
[ec2-user@controller myansible]$ pwd
/home/ec2-user/myansible
[ec2-user@controller myansible]$
```



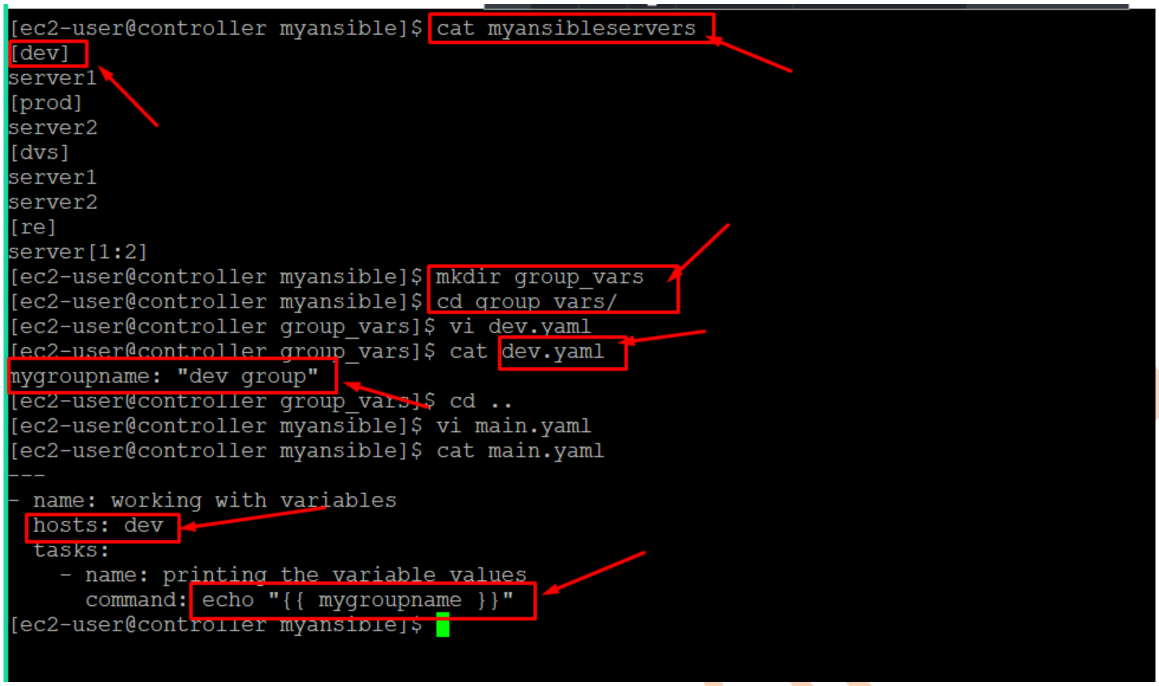
```
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with variables
  hosts: server1
  tasks:
    - name: printing the variable values
      command: echo "{{ myservername }}"
[ec2-user@controller myansible]$ cat ./host_vars/server1.yaml
myservername: "server1.example.com"
[ec2-user@controller myansible]$ ansible-playbook main.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with variables] ****************************************************************************************

TASK [Gathering Facts] ***********************************************************************************************
ok: [server1]

TASK [printing the variable values] **********************************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "server1.example.com"], "delta": "0:00:00.002729", "end": "2020-09
-24 15:21:31.946865", "rc": 0, "start": "2020-09-24 15:21:31.944136", "stderr": "", "stderr_lines": [], "stdout": "server1
.example.com", "stdout_lines": ["server1.example.com"]}

PLAY RECAP ***********************************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ vi main.yaml
```

Now test the same execution on server2

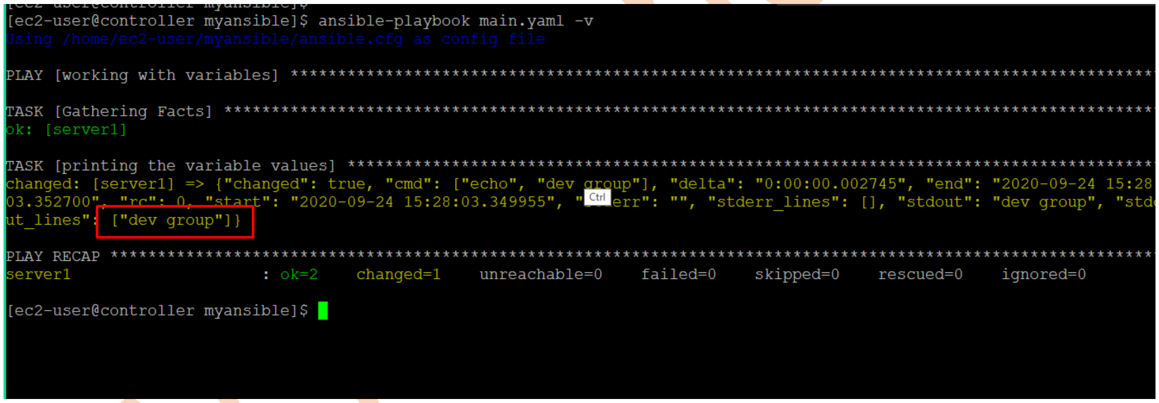


```
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with variables
  hosts: server2
  tasks:
    - name: printing the variable values
      command: echo "{{ myservername }}"
[ec2-user@controller myansible]$ ansible-playbook main.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with variables] ****************************************************************************************

TASK [Gathering Facts] ***********************************************************************************************
ok: [server2]

TASK [printing the variable values] **********************************************************************************
fatal: [server2]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'myservername
' is undefined\n\nThe error appears to be in '/home/ec2-user/myansible/main.yaml': line 5, column 7, but may\nbe elsewhere
 in the file depending on the exact syntax problem.\n\nThe offending line appears to be:\n\n  tasks:\n    - name: printing
 the variable values\n      ^ here\n"}

PLAY RECAP ***********************************************************************************************************
server2                    : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

group_vars:

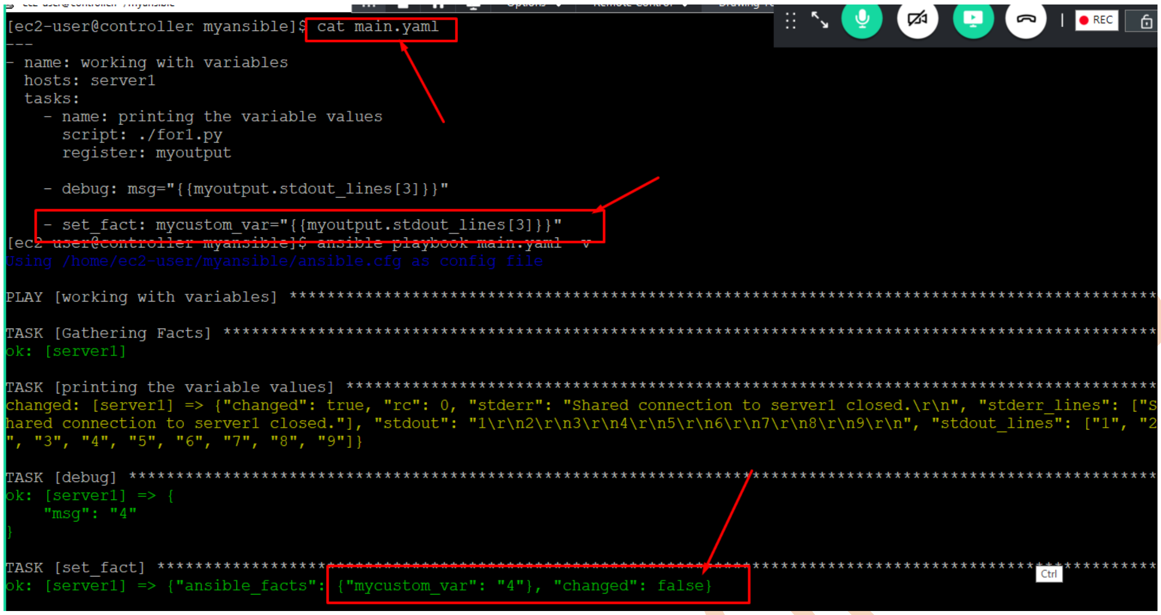


```
[ec2-user@controller myansible]$ cat myansibleservers
[dev]
server1
[prod]
server2
[dvs]
server1
server2
[re]
server[1:2]
[ec2-user@controller myansible]$ mkdir group_vars
[ec2-user@controller myansible]$ cd group_vars/
[ec2-user@controller group_vars]$ vi dev.yaml
[ec2-user@controller group_vars]$ cat dev.yaml
mygroupname: "dev group"
[ec2-user@controller group_vars]$ cd ..
[ec2-user@controller myansible]$ vi main.yaml
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with variables
  hosts: dev
  tasks:
    - name: printing the variable values
      command: echo "{{ mygroupname }}"
[ec2-user@controller myansible]$
```



```
[ec2-user@controller myansible]$
[ec2-user@controller myansible]$ ansible-playbook main.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with variables] **************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [server1]

TASK [printing the variable values] ********************************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "dev group"], "delta": "0:00:00.002745", "end": "2020-09-24 15:28
03.352700", "rc": 0, "start": "2020-09-24 15:28:03.349955", "Ctrl err": "", "stderr_lines": [], "stdout": "dev group", "std
ut_lines": ["dev group"]}

PLAY RECAP *********************************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

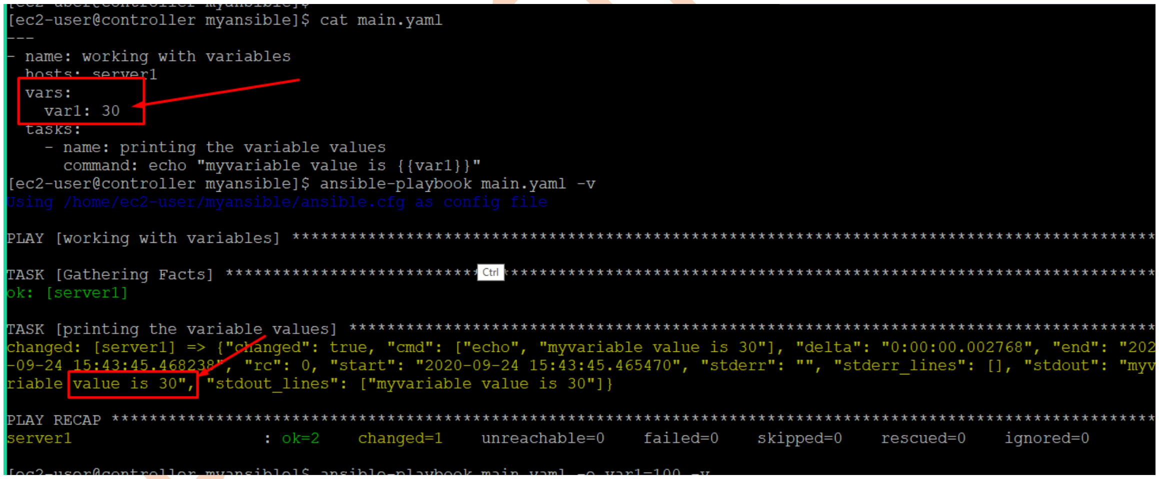
set_fact section:



```
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with variables
  hosts: server1
  tasks:
    - name: printing the variable values
      script: ./for1.py
      register: myoutput

    - debug: msg="{{myoutput.stdout_lines[3]}}"

    - set_fact: mycustom_var="{{myoutput.stdout_lines[3]}}"
[ec2-user@controller myansible]$ ansible-playbook main.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with variables] ******************************************************************************

TASK [Gathering Facts] *************************************************************************************
ok: [server1]

TASK [printing the variable values] ************************************************************************
changed: [server1] => {"changed": true, "rc": 0, "stderr": "Shared connection to server1 closed.\r\n", "stderr_lines": ["S
hared connection to server1 closed."], "stdout": "1\r\n2\r\n3\r\n4\r\n5\r\n6\r\n7\r\n8\r\n9\r\n", "stdout_lines": ["1", "2
", "3", "4", "5", "6", "7", "8", "9"]}

TASK [debug] ***********************************************************************************************
ok: [server1] => {
    "msg": "4"
}

TASK [set_fact] ********************************************************************************************
ok: [server1] => {"ansible_facts": {"mycustom_var": "4"}, "changed": false}
```

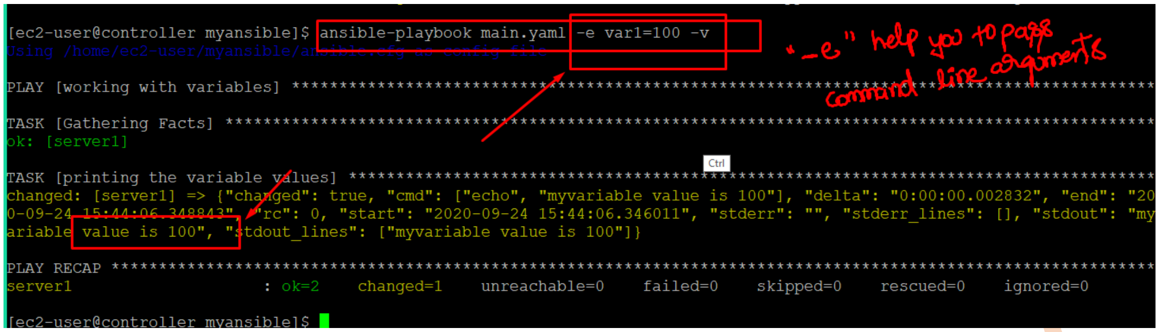
Run time variables:

Normal execution:



```
[ec2-user@controller myansible]$ cat main.yaml
---
- name: working with variables
  hosts: server1
  vars:
    var1: 30
  tasks:
    - name: printing the variable values
      command: echo "myvariable value is {{var1}}"
[ec2-user@controller myansible]$ ansible-playbook main.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with variables] ******************************************************************************

TASK [Gathering Facts] *************************************************************************************
ok: [server1]

TASK [printing the variable values] ************************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "myvariable value is 30"], "delta": "0:00:00.002768", "end": "202
-09-24 15:43:45.468238", "rc": 0, "start": "2020-09-24 15:43:45.465470", "stderr": "", "stderr_lines": [], "stdout": "myv
riable value is 30", "stdout_lines": ["myvariable value is 30"]}

PLAY RECAP *************************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ ansible-playbook main.yaml -e var1=100 -v
```

Passing parameters from command line:



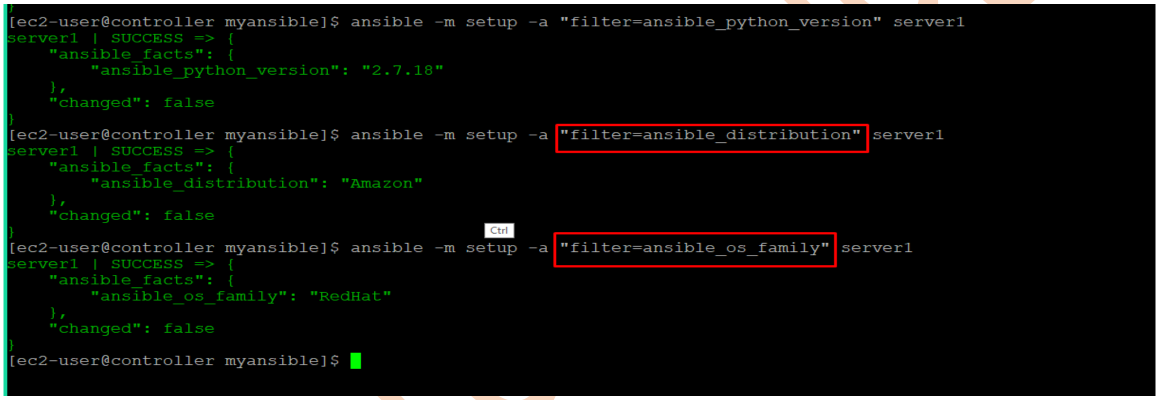
```
[ec2-user@controller myansible]$ ansible-playbook main.yaml -e var1=100 -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with variables] **************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server1]

TASK [printing the variable values] ********************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "myvariable value is 100"], "delta": "0:00:00.002832", "end": "20
0-09-24 15:44:06.348843", "rc": 0, "start": "2020-09-24 15:44:06.346011", "stderr": "", "stderr_lines": [], "stdout": "my
ariable value is 100", "stdout_lines": ["myvariable value is 100"]}

PLAY RECAP *********************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

"-e" help you to pass command line arguments

\

6. Conditions

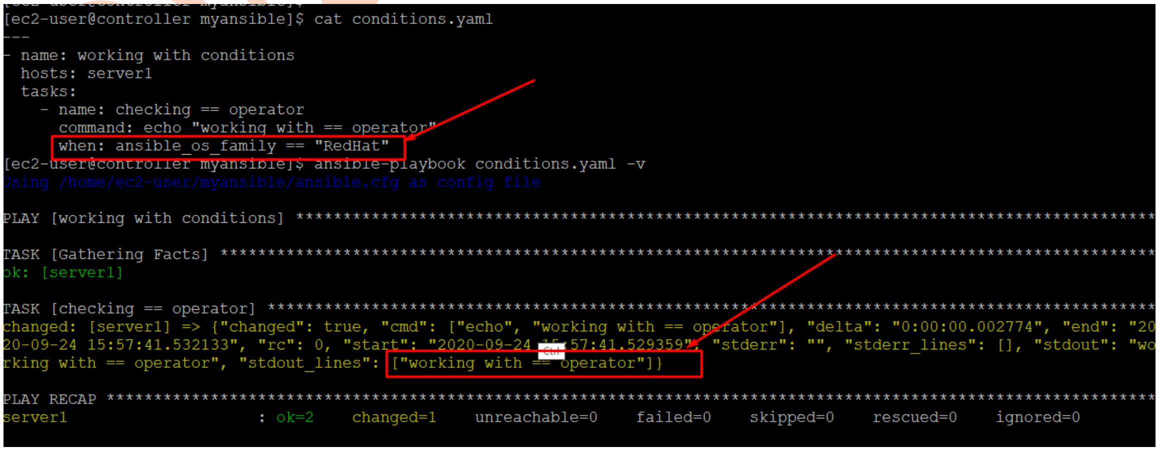
Setup module:



```
[ec2-user@controller myansible]$ ansible -m setup -a "filter=ansible_python_version" server1
server1 | SUCCESS => {
    "ansible_facts": {
        "ansible_python_version": "2.7.18"
    },
    "changed": false
}
[ec2-user@controller myansible]$ ansible -m setup -a "filter=ansible_distribution" server1
server1 | SUCCESS => {
    "ansible_facts": {
        "ansible_distribution": "Amazon"
    },
    "changed": false
}
[ec2-user@controller myansible]$ ansible -m setup -a "filter=ansible_os_family" server1
server1 | SUCCESS => {
    "ansible_facts": {
        "ansible_os_family": "RedHat"
    },
    "changed": false
}
[ec2-user@controller myansible]$
```

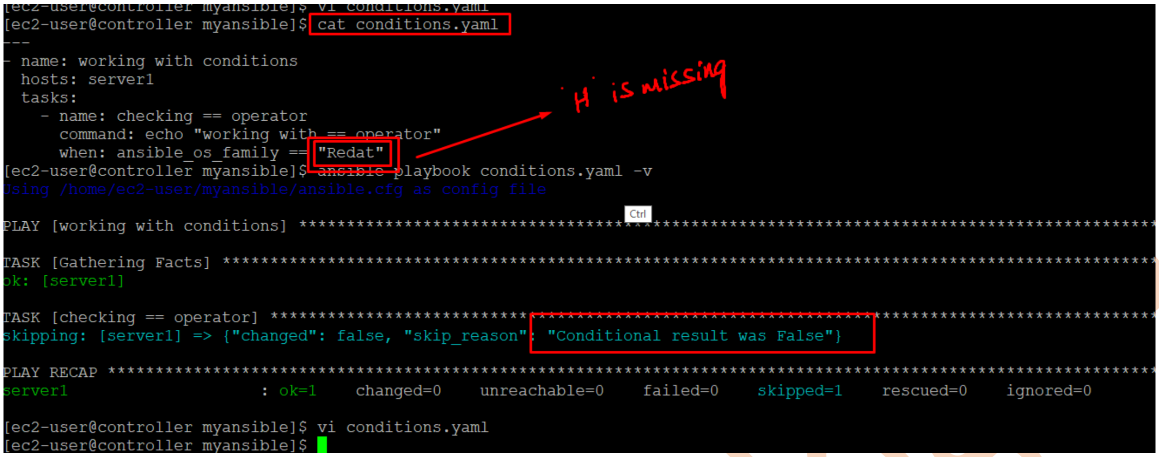
Let's start working on conditions

== operator:



```
[ec2-user@controller myansible]$ cat conditions.yaml
---
- name: working with conditions
  hosts: server1
  tasks:
    - name: checking == operator
      command: echo "working with == operator"
      when: ansible_os_family == "RedHat"
[ec2-user@controller myansible]$ ansible-playbook conditions.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with conditions] *************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server1]

TASK [checking == operator] ****************************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "working with == operator"], "delta": "0:00:00.002774", "end": "20
20-09-24 15:57:41.532133", "rc": 0, "start": "2020-09-24 15:57:41.529359", "stderr": "", "stderr_lines": [], "stdout": "wo
rking with == operator", "stdout_lines": ["working with == operator"]}

PLAY RECAP *********************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

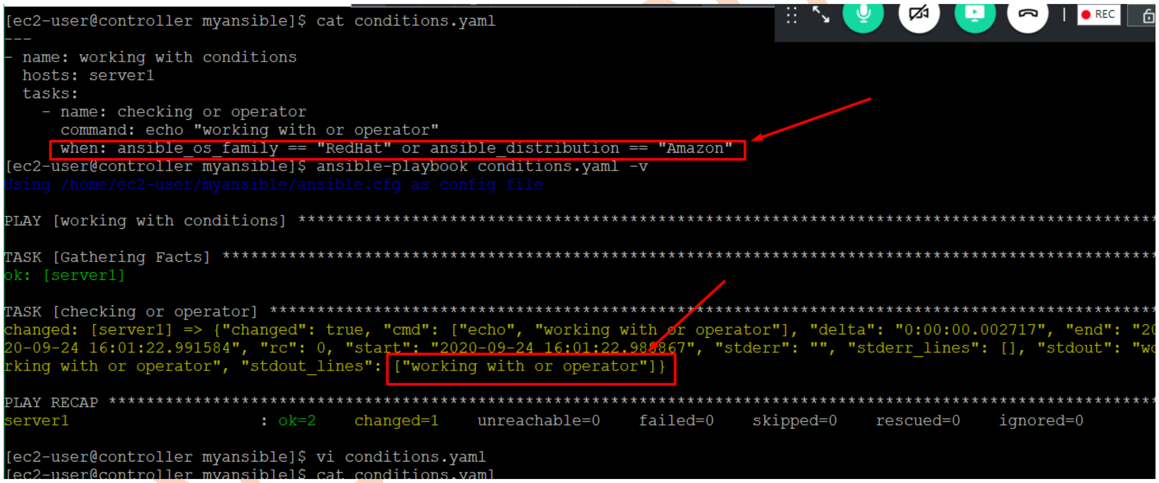
Let's check the false condition



```
[ec2-user@controller myansible]$ vi conditions.yaml
[ec2-user@controller myansible]$ cat conditions.yaml
---
- name: working with conditions
  hosts: server1
  tasks:
    - name: checking == operator
      command: echo "working with == operator"
      when: ansible_os_family == "Redat"
[ec2-user@controller myansible]$ ansible-playbook conditions.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with conditions] ************************************************************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [server1]

TASK [checking == operator] ***************************************************************************************
skipping: [server1] => {"changed": false, "skip_reason": "Conditional result was False"}

PLAY RECAP ********************************************************************************************************
server1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

[ec2-user@controller myansible]$ vi conditions.yaml
[ec2-user@controller myansible]$
```

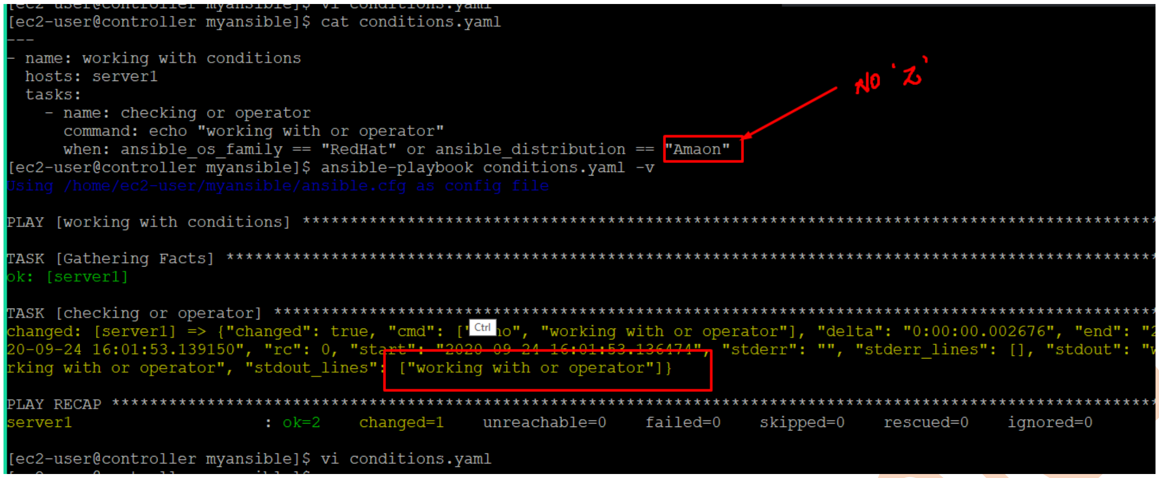
'H' is missing

Or operator:



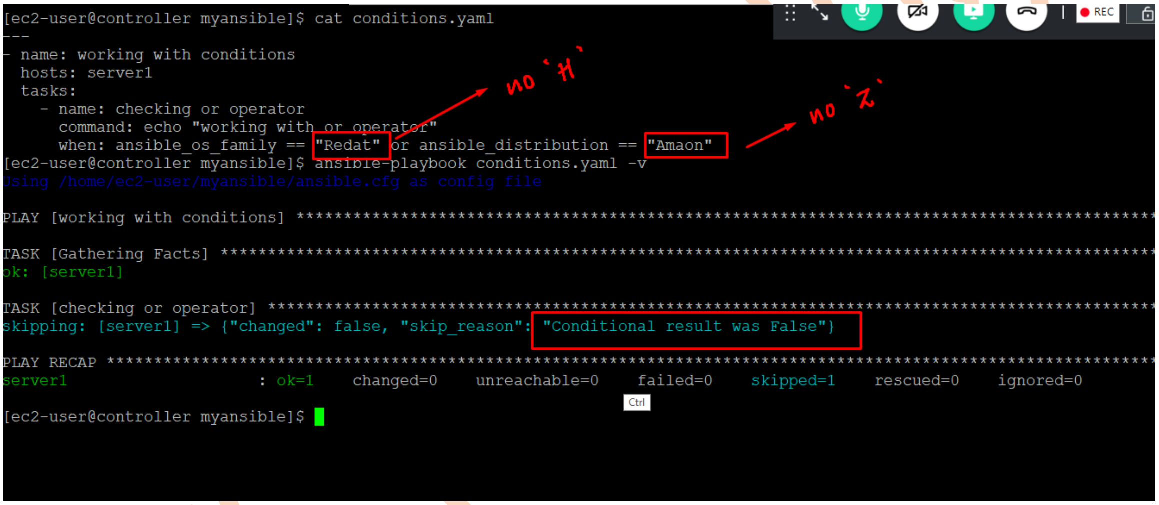
```
[ec2-user@controller myansible]$ cat conditions.yaml
---
- name: working with conditions
  hosts: server1
  tasks:
    - name: checking or operator
      command: echo "working with or operator"
      when: ansible_os_family == "RedHat" or ansible_distribution == "Amazon"
[ec2-user@controller myansible]$ ansible-playbook conditions.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with conditions] ************************************************************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [server1]

TASK [checking or operator] ***************************************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "working with or operator"], "delta": "0:00:00.002717", "end": "20
20-09-24 16:01:22.991584", "rc": 0, "start": "2020-09-24 16:01:22.988867", "stderr": "", "stderr_lines": [], "stdout": "wo
rking with or operator", "stdout_lines": ["working with or operator"]}

PLAY RECAP ********************************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ vi conditions.yaml
[ec2-user@controller myansible]$ cat conditions.yaml
```

Case1:



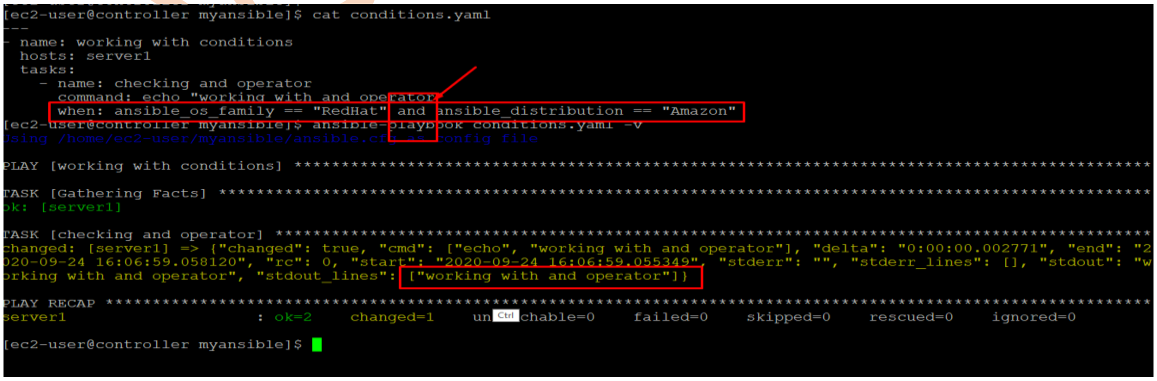
```
[ec2-user@controller myansible]$ vi conditions.yaml
[ec2-user@controller myansible]$ cat conditions.yaml
---
- name: working with conditions
  hosts: server1
  tasks:
    - name: checking or operator
      command: echo "working with or operator"
      when: ansible_os_family == "RedHat" or ansible_distribution == "Amaon"
[ec2-user@controller myansible]$ ansible-playbook conditions.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with conditions] ***********************************************************************

TASK [Gathering Facts] *******************************************************************************
ok: [server1]

TASK [checking or operator] **************************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "working with or operator"], "delta": "0:00:00.002676", "end": "2
20-09-24 16:01:53.139150", "rc": 0, "start": "2020-09-24 16:01:53.136474", "stderr": "", "stderr_lines": [], "stdout": "w
rking with or operator", "stdout_lines": ["working with or operator"]}

PLAY RECAP *******************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ vi conditions.yaml
```

Case2:

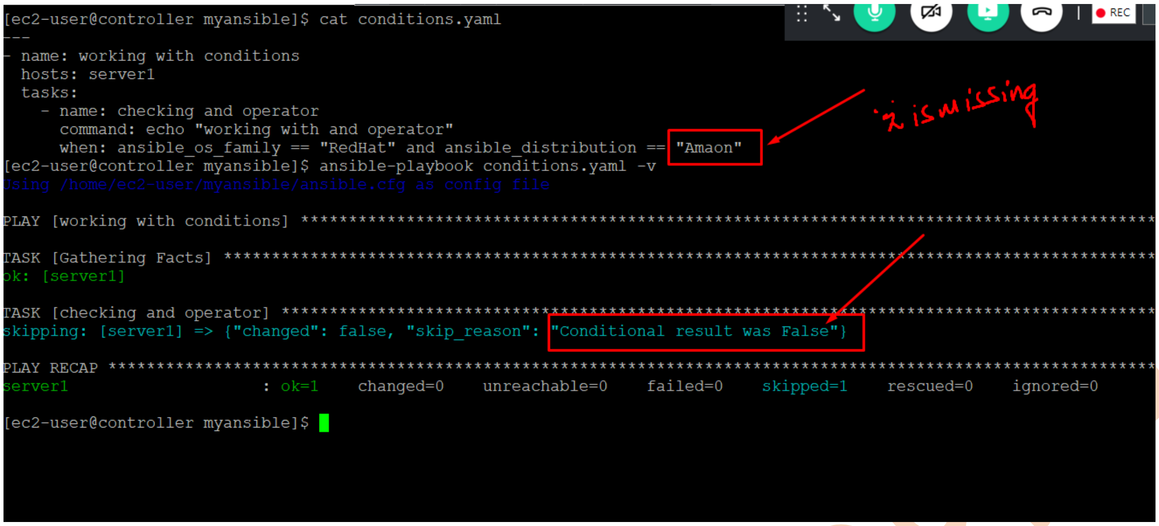


```
[ec2-user@controller myansible]$ cat conditions.yaml
---
- name: working with conditions
  hosts: server1
  tasks:
    - name: checking or operator
      command: echo "working with or operator"
      when: ansible_os_family == "Redat" or ansible_distribution == "Amaon"
[ec2-user@controller myansible]$ ansible-playbook conditions.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with conditions] ***********************************************************************

TASK [Gathering Facts] *******************************************************************************
ok: [server1]

TASK [checking or operator] **************************************************************************
skipping: [server1] => {"changed": false, "skip_reason": "Conditional result was False"}

PLAY RECAP *******************************************************************************************
server1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

and operator:



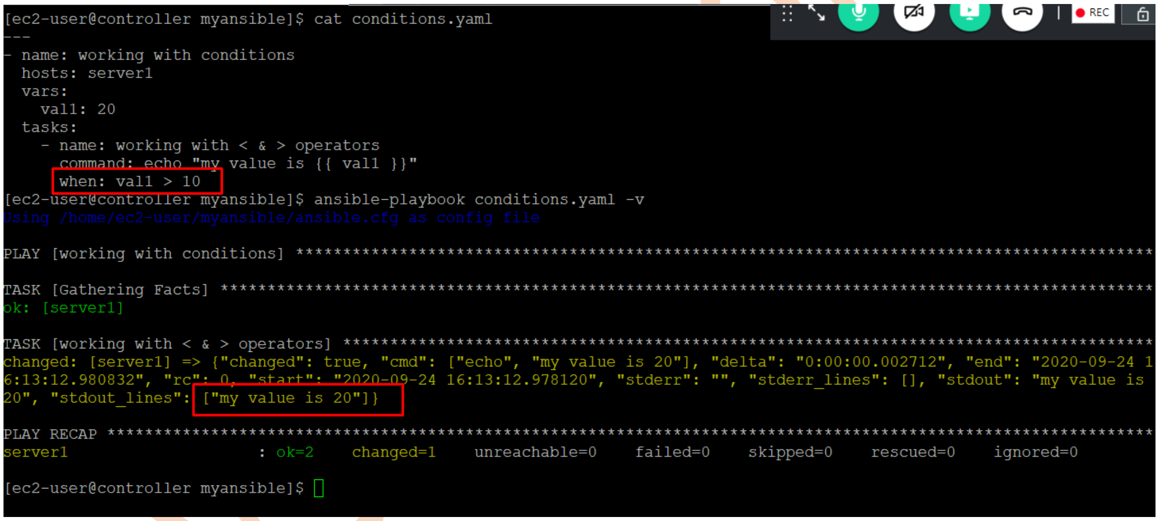
```
[ec2-user@controller myansible]$ cat conditions.yaml
---
- name: working with conditions
  hosts: server1
  tasks:
    - name: checking and operator
      command: echo "working with and operator"
      when: ansible_os_family == "RedHat" and ansible_distribution == "Amazon"
[ec2-user@controller myansible]$ ansible-playbook conditions.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with conditions] ***********************************************************************

TASK [Gathering Facts] *******************************************************************************
ok: [server1]

TASK [checking and operator] *************************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "working with and operator"], "delta": "0:00:00.002771", "end": "2
020-09-24 16:06:59.058120", "rc": 0, "start": "2020-09-24 16:06:59.055349", "stderr": "", "stderr_lines": [], "stdout": "w
orking with and operator", "stdout_lines": ["working with and operator"]}

PLAY RECAP *******************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```



```
[ec2-user@controller myansible]$ cat conditions.yaml
---
- name: working with conditions
  hosts: server1
  tasks:
    - name: checking and operator
      command: echo "working with and operator"
      when: ansible_os_family == "RedHat" and ansible_distribution == "Amaon"
[ec2-user@controller myansible]$ ansible-playbook conditions.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with conditions] **************************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server1]

TASK [checking and operator] ****************************************************************************
skipping: [server1] => {"changed": false, "skip_reason": "Conditional result was False"}

PLAY RECAP **********************************************************************************************
server1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

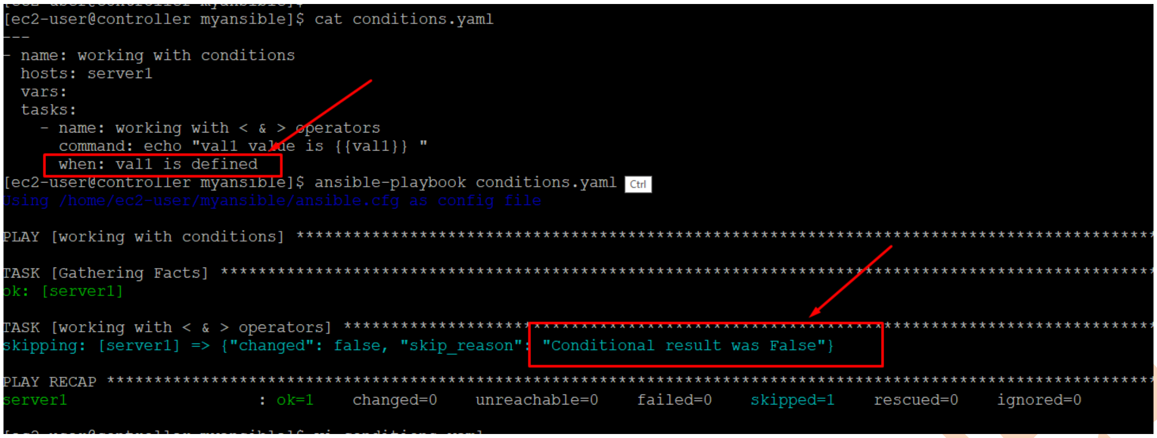
'z is missing

< & > operators:

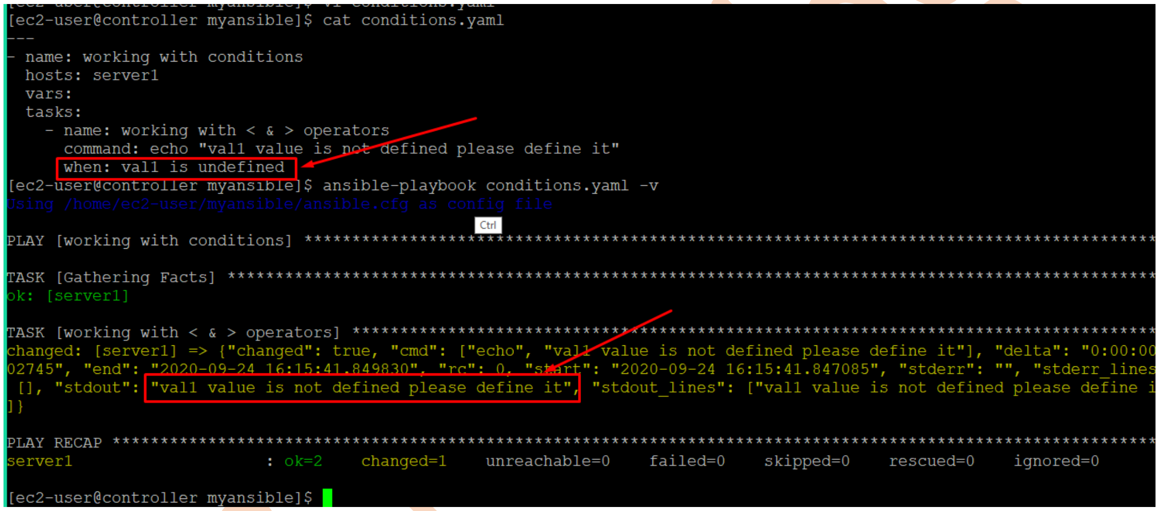


```
[ec2-user@controller myansible]$ cat conditions.yaml
---
- name: working with conditions
  hosts: server1
  vars:
    val1: 20
  tasks:
    - name: working with < & > operators
      command: echo "my value is {{ val1 }}"
      when: val1 > 10
[ec2-user@controller myansible]$ ansible-playbook conditions.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with conditions] **************************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server1]

TASK [working with < & > operators] *********************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "my value is 20"], "delta": "0:00:00.002712", "end": "2020-09-24 1
6:13:12.980832", "rc": 0, "start": "2020-09-24 16:13:12.978120", "stderr": "", "stderr_lines": [], "stdout": "my value is
20", "stdout_lines": ["my value is 20"]}

PLAY RECAP **********************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

defined & undefined:



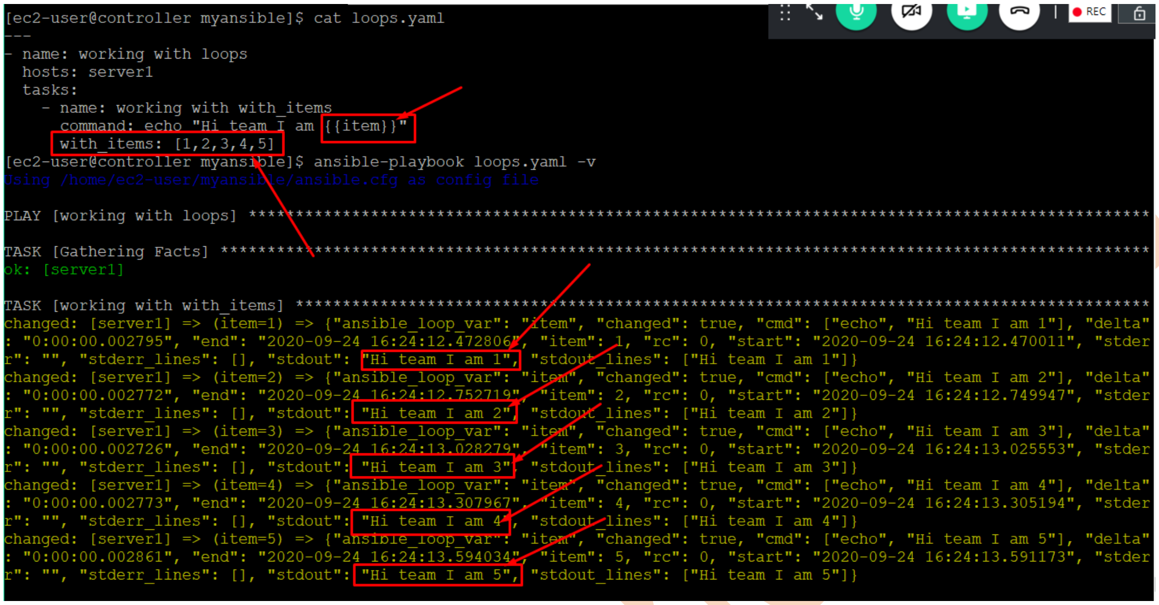
```
[ec2-user@controller myansible]$ cat conditions.yaml
---
- name: working with conditions
  hosts: server1
  vars:
  tasks:
    - name: working with < & > operators
      command: echo "val1 value is {{val1}} "
      when: val1 is defined
[ec2-user@controller myansible]$ ansible-playbook conditions.yaml
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with conditions] *************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server1]

TASK [working with < & > operators] ********************************************************************
skipping: [server1] => {"changed": false, "skip_reason": "Conditional result was False"}

PLAY RECAP *********************************************************************************************
server1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```



```
[ec2-user@controller myansible]$ cat conditions.yaml
---
- name: working with conditions
  hosts: server1
  vars:
  tasks:
    - name: working with < & > operators
      command: echo "val1 value is not defined please define it"
      when: val1 is undefined
[ec2-user@controller myansible]$ ansible-playbook conditions.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with conditions] *************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server1]

TASK [working with < & > operators] ********************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "val1 value is not defined please define it"], "delta": "0:00:00
02745", "end": "2020-09-24 16:15:41.849830", "rc": 0, "start": "2020-09-24 16:15:41.847085", "stderr": "", "stderr_lines
 [], "stdout": "val1 value is not defined please define it", "stdout_lines": ["val1 value is not defined please define i
]}

PLAY RECAP *********************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

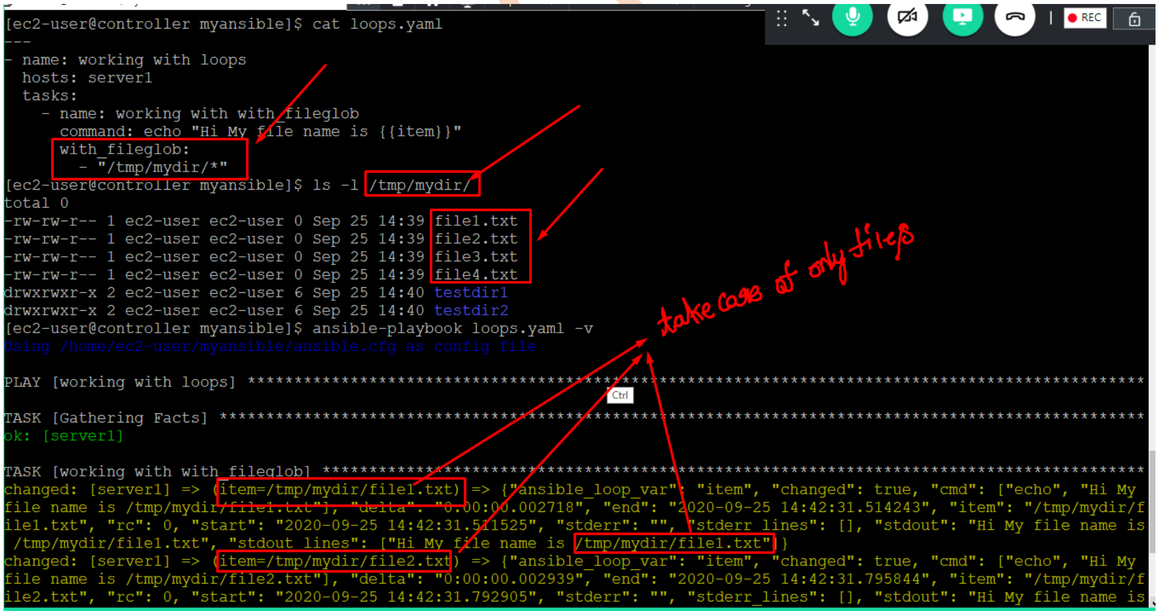
7. Loops

With_items:



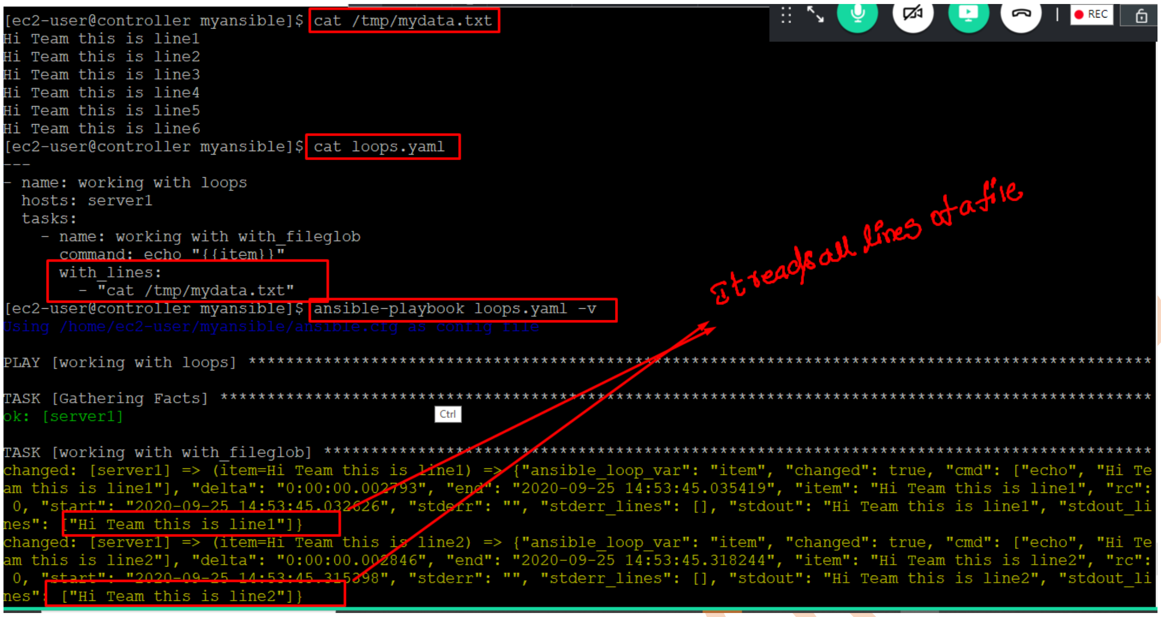
```
[ec2-user@controller myansible]$ cat loops.yaml
---
- name: working with loops
  hosts: server1
  tasks:
    - name: working with with_items
      command: echo "Hi team I am {{item}}"
      with_items: [1,2,3,4,5]
[ec2-user@controller myansible]$ ansible-playbook loops.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with loops] ******************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server1]

TASK [working with with_items] *************************************************************************
changed: [server1] => (item=1) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "Hi team I am 1"], "delta"
: "0:00:00.002795", "end": "2020-09-24 16:24:12.472806", "item": 1, "rc": 0, "start": "2020-09-24 16:24:12.470011", "stder
r": "", "stderr_lines": [], "stdout": "Hi team I am 1", "stdout_lines": ["Hi team I am 1"]}
changed: [server1] => (item=2) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "Hi team I am 2"], "delta"
: "0:00:00.002772", "end": "2020-09-24 16:24:12.752719", "item": 2, "rc": 0, "start": "2020-09-24 16:24:12.749947", "stder
r": "", "stderr_lines": [], "stdout": "Hi team I am 2", "stdout_lines": ["Hi team I am 2"]}
changed: [server1] => (item=3) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "Hi team I am 3"], "delta"
: "0:00:00.002726", "end": "2020-09-24 16:24:13.028279", "item": 3, "rc": 0, "start": "2020-09-24 16:24:13.025553", "stder
r": "", "stderr_lines": [], "stdout": "Hi team I am 3", "stdout_lines": ["Hi team I am 3"]}
changed: [server1] => (item=4) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "Hi team I am 4"], "delta"
: "0:00:00.002773", "end": "2020-09-24 16:24:13.307967", "item": 4, "rc": 0, "start": "2020-09-24 16:24:13.305194", "stder
r": "", "stderr_lines": [], "stdout": "Hi team I am 4", "stdout_lines": ["Hi team I am 4"]}
changed: [server1] => (item=5) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "Hi team I am 5"], "delta"
: "0:00:00.002861", "end": "2020-09-24 16:24:13.594034", "item": 5, "rc": 0, "start": "2020-09-24 16:24:13.591173", "stder
r": "", "stderr_lines": [], "stdout": "Hi team I am 5", "stdout_lines": ["Hi team I am 5"]}
```

With_fileglob:



```
[ec2-user@controller myansible]$ cat loops.yaml
---
- name: working with loops
  hosts: server1
  tasks:
    - name: working with with_fileglob
      command: echo "Hi My file name is {{item}}"
      with_fileglob:
        - "/tmp/mydir/*"
[ec2-user@controller myansible]$ ls -l /tmp/mydir/
total 0
-rw-rw-r-- 1 ec2-user ec2-user 0 Sep 25 14:39 file1.txt
-rw-rw-r-- 1 ec2-user ec2-user 0 Sep 25 14:39 file2.txt
-rw-rw-r-- 1 ec2-user ec2-user 0 Sep 25 14:39 file3.txt
-rw-rw-r-- 1 ec2-user ec2-user 0 Sep 25 14:39 file4.txt
drwxrwxr-x 2 ec2-user ec2-user 6 Sep 25 14:40 testdir1
drwxrwxr-x 2 ec2-user ec2-user 6 Sep 25 14:40 testdir2
[ec2-user@controller myansible]$ ansible-playbook loops.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with loops] ******************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server1]

TASK [working with with_fileglob] **********************************************************************
changed: [server1] => (item=/tmp/mydir/file1.txt) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "Hi My
file name is /tmp/mydir/file1.txt"], "delta": "0:00:00.002718", "end": "2020-09-25 14:42:31.514243", "item": "/tmp/mydir/f
ile1.txt", "rc": 0, "start": "2020-09-25 14:42:31.511525", "stderr": "", "stderr_lines": [], "stdout": "Hi My file name is
 /tmp/mydir/file1.txt", "stdout_lines": ["Hi My file name is /tmp/mydir/file1.txt"]}
changed: [server1] => (item=/tmp/mydir/file2.txt) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "Hi My
file name is /tmp/mydir/file2.txt"], "delta": "0:00:00.002939", "end": "2020-09-25 14:42:31.795844", "item": "/tmp/mydir/f
ile2.txt", "rc": 0, "start": "2020-09-25 14:42:31.792905", "stderr": "", "stderr_lines": [], "stdout": "Hi My file name is
```

With_lines:

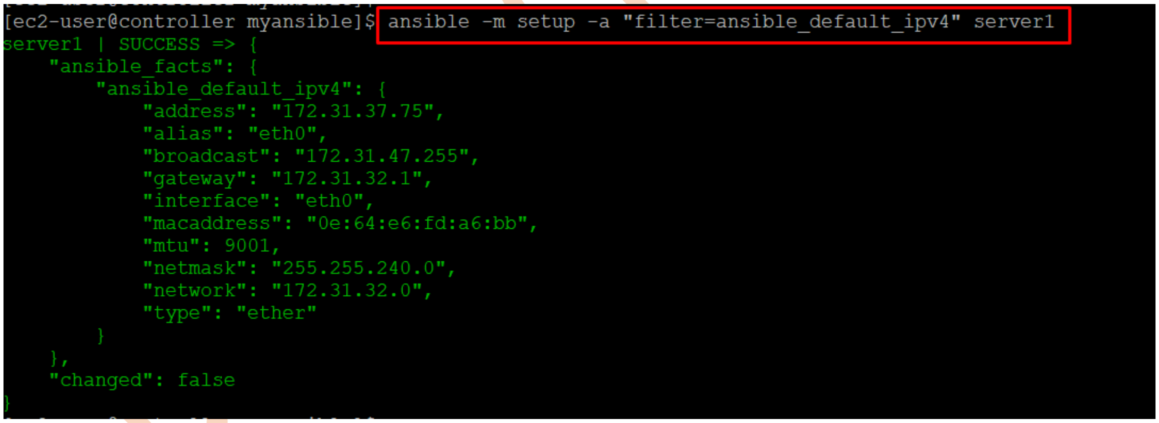


```
[ec2-user@controller myansible]$ cat /tmp/mydata.txt
Hi Team this is line1
Hi Team this is line2
Hi Team this is line3
Hi Team this is line4
Hi Team this is line5
Hi Team this is line6
[ec2-user@controller myansible]$ cat loops.yaml
---
- name: working with loops
  hosts: server1
  tasks:
    - name: working with with_fileglob
      command: echo "{{item}}"
      with_lines:
        - "cat /tmp/mydata.txt"
[ec2-user@controller myansible]$ ansible-playbook loops.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with loops] ******************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server1]

TASK [working with with_fileglob] **********************************************************************
changed: [server1] => (item=Hi Team this is line1) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "Hi Te
am this is line1"], "delta": "0:00:00.002793", "end": "2020-09-25 14:53:45.035419", "item": "Hi Team this is line1", "rc":
 0, "start": "2020-09-25 14:53:45.032626", "stderr": "", "stderr_lines": [], "stdout": "Hi Team this is line1", "stdout_li
nes": ["Hi Team this is line1"]}
changed: [server1] => (item=Hi Team this is line2) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "Hi Te
am this is line2"], "delta": "0:00:00.002846", "end": "2020-09-25 14:53:45.318244", "item": "Hi Team this is line2", "rc":
 0, "start": "2020-09-25 14:53:45.315398", "stderr": "", "stderr_lines": [], "stdout": "Hi Team this is line2", "stdout_li
nes": ["Hi Team this is line2"]}
```

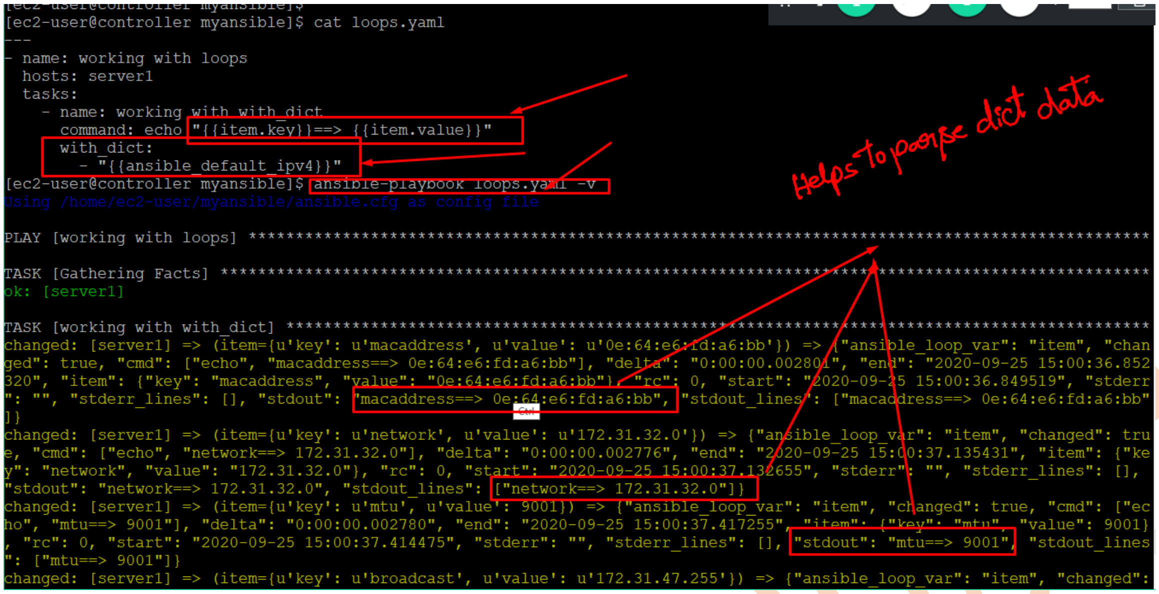
It reads all lines of a file

with_dict:

Input data i.e dict:

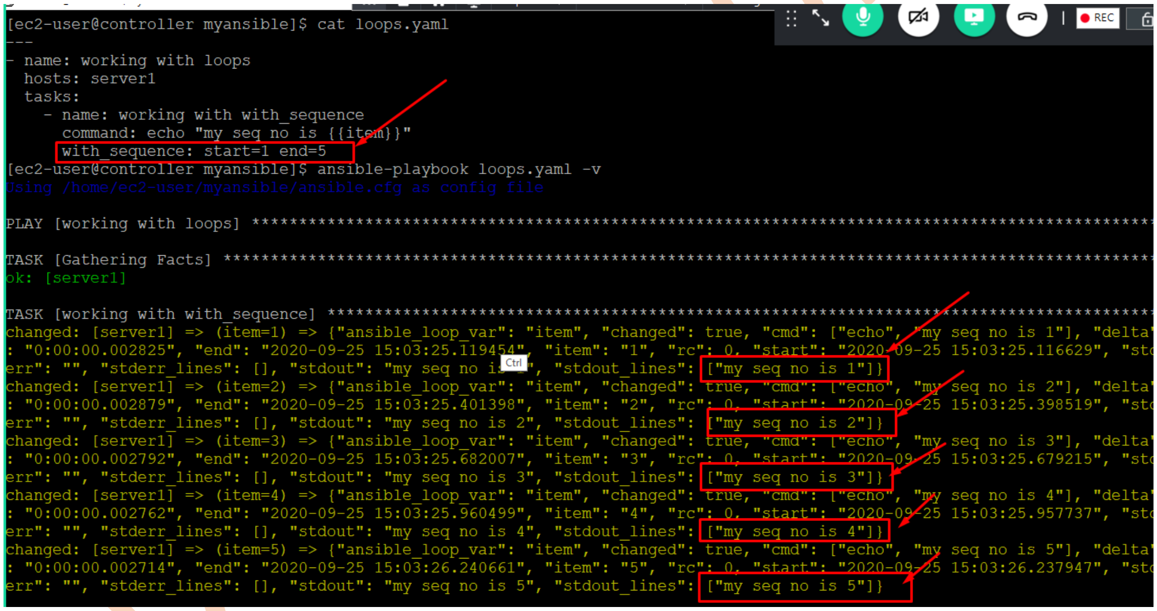


```
[ec2-user@controller myansible]$ ansible -m setup -a "filter=ansible_default_ipv4" server1
server1 | SUCCESS => {
    "ansible_facts": {
        "ansible_default_ipv4": {
            "address": "172.31.37.75",
            "alias": "eth0",
            "broadcast": "172.31.47.255",
            "gateway": "172.31.32.1",
            "interface": "eth0",
            "macaddress": "0e:64:e6:fd:a6:bb",
            "mtu": 9001,
            "netmask": "255.255.240.0",
            "network": "172.31.32.0",
            "type": "ether"
        }
    },
    "changed": false
}
```



```
[ec2-user@controller myansible]$ cat loops.yaml
---
- name: working with loops
  hosts: server1
  tasks:
    - name: working with with_dict
      command: echo "{{item.key}}==> {{item.value}}"
      with_dict:
        - "{{ansible_default_ipv4}}"
[ec2-user@controller myansible]$ ansible-playbook loops.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with loops] ****************************************************************************************

TASK [Gathering Facts] *******************************************************************************************
ok: [server1]

TASK [working with with_dict] ************************************************************************************
changed: [server1] => (item={u'key': u'macaddress', u'value': u'0e:64:e6:fd:a6:bb'}) => {"ansible_loop_var": "item", "chan
ged": true, "cmd": ["echo", "macaddress==> 0e:64:e6:fd:a6:bb"], "delta": "0:00:00.002801", "end": "2020-09-25 15:00:36.852
320", "item": {"key": "macaddress", "value": "0e:64:e6:fd:a6:bb"}, "rc": 0, "start": "2020-09-25 15:00:36.849519", "stderr
": "", "stderr_lines": [], "stdout": "macaddress==> 0e:64:e6:fd:a6:bb", "stdout_lines": ["macaddress==> 0e:64:e6:fd:a6:bb"
]}
changed: [server1] => (item={u'key': u'network', u'value': u'172.31.32.0'}) => {"ansible_loop_var": "item", "changed": tru
e, "cmd": ["echo", "network==> 172.31.32.0"], "delta": "0:00:00.002776", "end": "2020-09-25 15:00:37.135431", "item": {"ke
y": "network", "value": "172.31.32.0"}, "rc": 0, "start": "2020-09-25 15:00:37.132655", "stderr": "", "stderr_lines": [],
"stdout": "network==> 172.31.32.0", "stdout_lines": ["network==> 172.31.32.0"]}
changed: [server1] => (item={u'key': u'mtu', u'value': 9001}) => {"ansible_loop_var": "item", "changed": true, "cmd": ["ec
ho", "mtu==> 9001"], "delta": "0:00:00.002780", "end": "2020-09-25 15:00:37.417255", "item": {"key": "mtu", "value": 9001}
, "rc": 0, "start": "2020-09-25 15:00:37.414475", "stderr": "", "stderr_lines": [], "stdout": "mtu==> 9001", "stdout_lines
": ["mtu==> 9001"]}
changed: [server1] => (item={u'key': u'broadcast', u'value': u'172.31.47.255'}) => {"ansible_loop_var": "item", "changed":
```

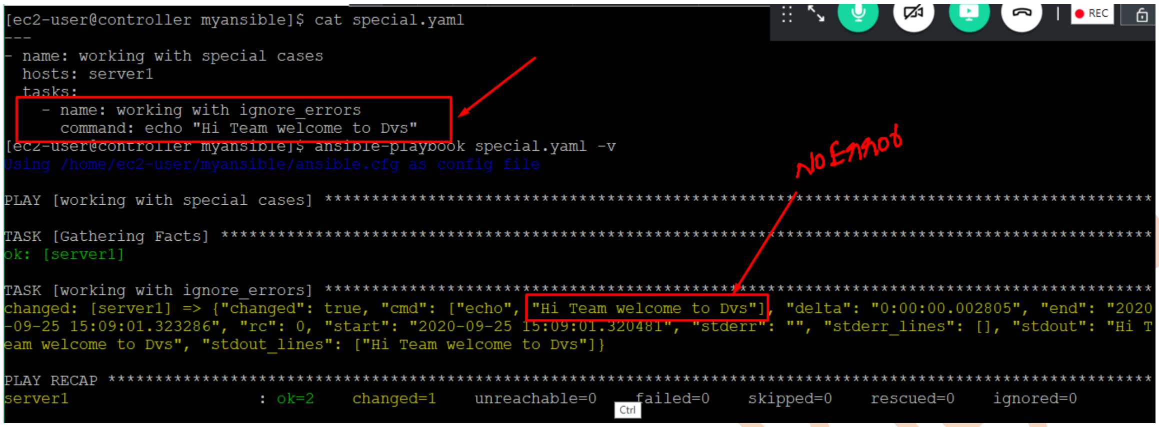
with_sequence:



```
[ec2-user@controller myansible]$ cat loops.yaml
---
- name: working with loops
  hosts: server1
  tasks:
    - name: working with with_sequence
      command: echo "my seq no is {{item}}"
      with_sequence: start=1 end=5
[ec2-user@controller myansible]$ ansible-playbook loops.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with loops] ****************************************************************************************

TASK [Gathering Facts] *******************************************************************************************
ok: [server1]

TASK [working with with_sequence] ********************************************************************************
changed: [server1] => (item=1) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "my seq no is 1"], "delta"
: "0:00:00.002825", "end": "2020-09-25 15:03:25.119454", "item": "1", "rc": 0, "start": "2020-09-25 15:03:25.116629", "std
err": "", "stderr_lines": [], "stdout": "my seq no is 1", "stdout_lines": ["my seq no is 1"]}
changed: [server1] => (item=2) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "my seq no is 2"], "delta"
: "0:00:00.002879", "end": "2020-09-25 15:03:25.401398", "item": "2", "rc": 0, "start": "2020-09-25 15:03:25.398519", "std
err": "", "stderr_lines": [], "stdout": "my seq no is 2", "stdout_lines": ["my seq no is 2"]}
changed: [server1] => (item=3) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "my seq no is 3"], "delta"
: "0:00:00.002792", "end": "2020-09-25 15:03:25.682007", "item": "3", "rc": 0, "start": "2020-09-25 15:03:25.679215", "std
err": "", "stderr_lines": [], "stdout": "my seq no is 3", "stdout_lines": ["my seq no is 3"]}
changed: [server1] => (item=4) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "my seq no is 4"], "delta"
: "0:00:00.002762", "end": "2020-09-25 15:03:25.960499", "item": "4", "rc": 0, "start": "2020-09-25 15:03:25.957737", "std
err": "", "stderr_lines": [], "stdout": "my seq no is 4", "stdout_lines": ["my seq no is 4"]}
changed: [server1] => (item=5) => {"ansible_loop_var": "item", "changed": true, "cmd": ["echo", "my seq no is 5"], "delta"
: "0:00:00.002714", "end": "2020-09-25 15:03:26.240661", "item": "5", "rc": 0, "start": "2020-09-25 15:03:26.237947", "std
err": "", "stderr_lines": [], "stdout": "my seq no is 5", "stdout_lines": ["my seq no is 5"]}
```

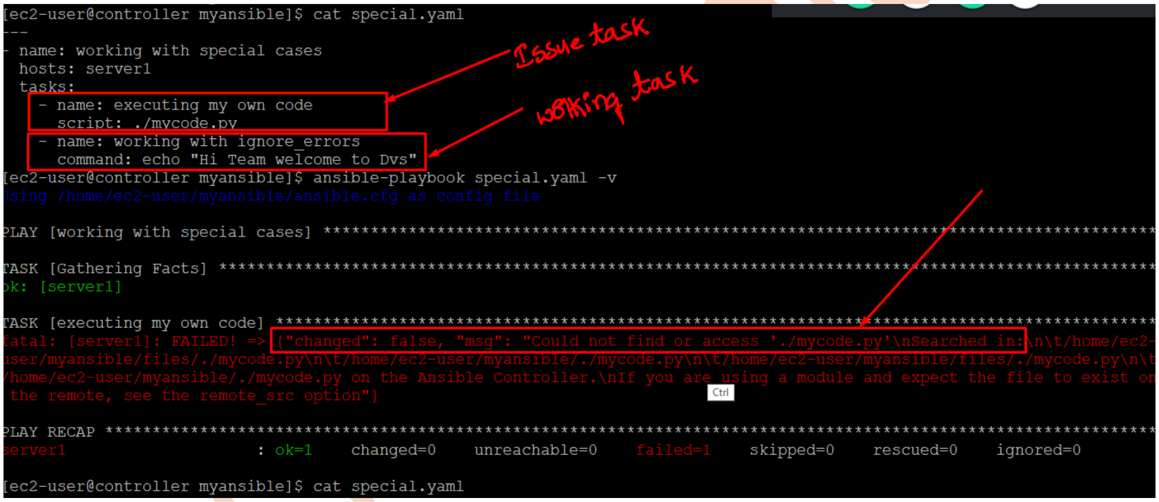
8. Special Cases with ansible

ignore_errors:



```
[ec2-user@controller myansible]$ cat special.yaml
---
- name: working with special cases
  hosts: server1
  tasks:
    - name: working with ignore_errors
      command: echo "Hi Team welcome to Dvs"
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] ***********************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server1]

TASK [working with ignore_errors] ***********************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "Hi Team welcome to Dvs"], "delta": "0:00:00.002805", "end": "2020
-09-25 15:09:01.323286", "rc": 0, "start": "2020-09-25 15:09:01.320481", "stderr": "", "stderr_lines": [], "stdout": "Hi T
eam welcome to Dvs", "stdout_lines": ["Hi Team welcome to Dvs"]}

PLAY RECAP **********************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

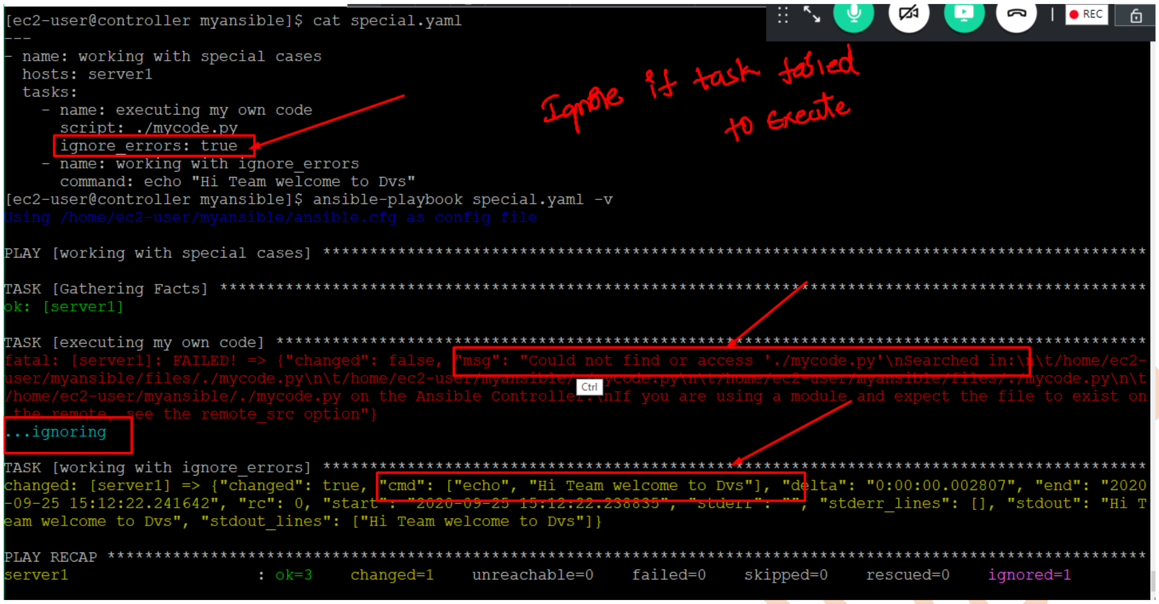
No Error



```
[ec2-user@controller myansible]$ cat special.yaml
---
- name: working with special cases
  hosts: server1
  tasks:
    - name: executing my own code
      script: ./mycode.py
    - name: working with ignore_errors
      command: echo "Hi Team welcome to Dvs"
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] ***********************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server1]

TASK [executing my own code] ****************************************************************************
fatal: [server1]: FAILED! => {"changed": false, "msg": "Could not find or access './mycode.py'\nSearched in:\n\t/home/ec2-
user/myansible/files/./mycode.py\n\t/home/ec2-user/myansible/./mycode.py\n\t/home/ec2-user/myansible/files/./mycode.py\n\t
/home/ec2-user/myansible/./mycode.py on the Ansible Controller.\nIf you are using a module and expect the file to exist on
 the remote, see the remote_src option"}

PLAY RECAP **********************************************************************************************
server1                    : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ cat special.yaml
```

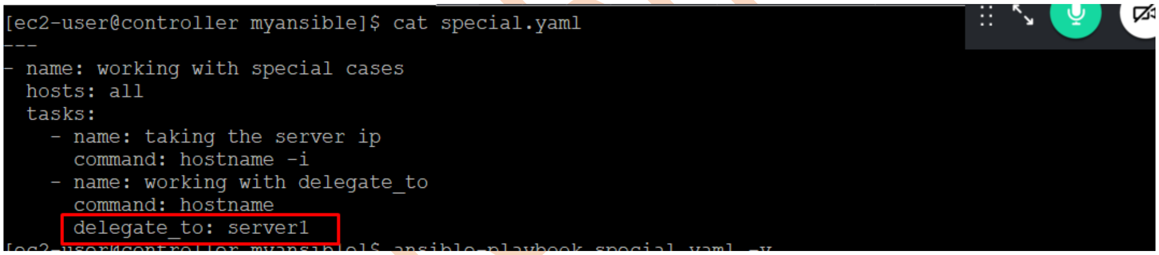
Issue task

woking task

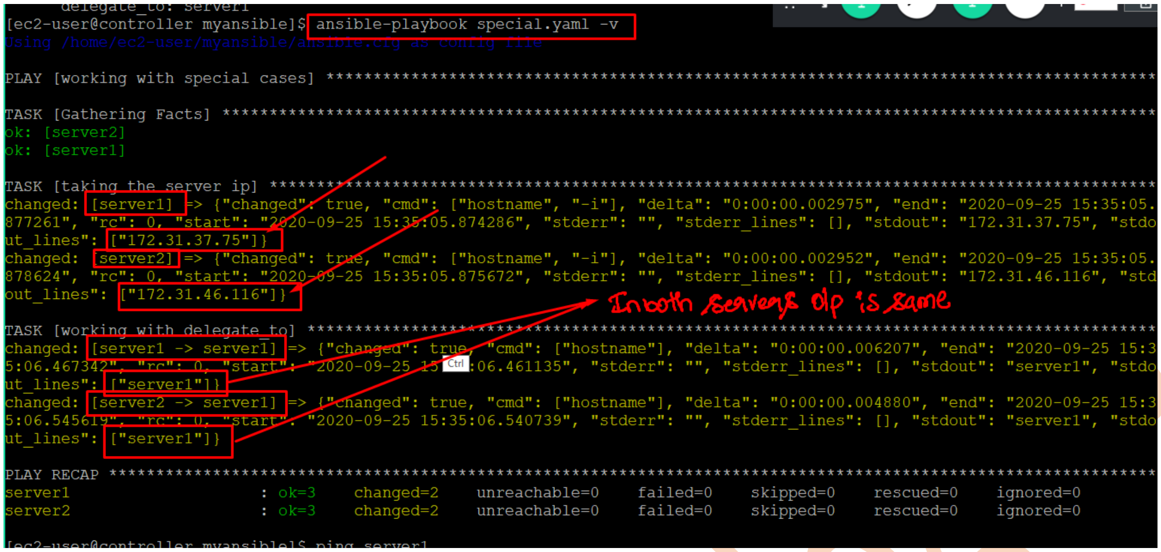


```
[ec2-user@controller myansible]$ cat special.yaml
---
- name: working with special cases
  hosts: server1
  tasks:
    - name: executing my own code
      script: ./mycode.py
      ignore_errors: true
    - name: working with ignore_errors
      command: echo "Hi Team welcome to Dvs"
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] ***********************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server1]

TASK [executing my own code] ****************************************************************************
fatal: [server1]: FAILED! => {"changed": false, "msg": "Could not find or access './mycode.py'\nSearched in:\n\t/home/ec2-
user/myansible/files/./mycode.py\n\t/home/ec2-user/myansible/./mycode.py\n\t/home/ec2-user/myansible/files/./mycode.py\n\t
/home/ec2-user/myansible/./mycode.py on the Ansible Controller.\nIf you are using a module and expect the file to exist on
 the remote, see the remote_src option"}
...ignoring

TASK [working with ignore_errors] ***********************************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "Hi Team welcome to Dvs"], "delta": "0:00:00.002807", "end": "2020
-09-25 15:12:22.241642", "rc": 0, "start": "2020-09-25 15:12:22.238835", "stderr": "", "stderr_lines": [], "stdout": "Hi T
eam welcome to Dvs", "stdout_lines": ["Hi Team welcome to Dvs"]}

PLAY RECAP **********************************************************************************************
server1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=1
```

Delegate_to:



```
[ec2-user@controller myansible]$ cat special.yaml
---
- name: working with special cases
  hosts: all
  tasks:
    - name: taking the server ip
      command: hostname -i
    - name: working with delegate_to
      command: hostname
      delegate_to: server1
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v
```

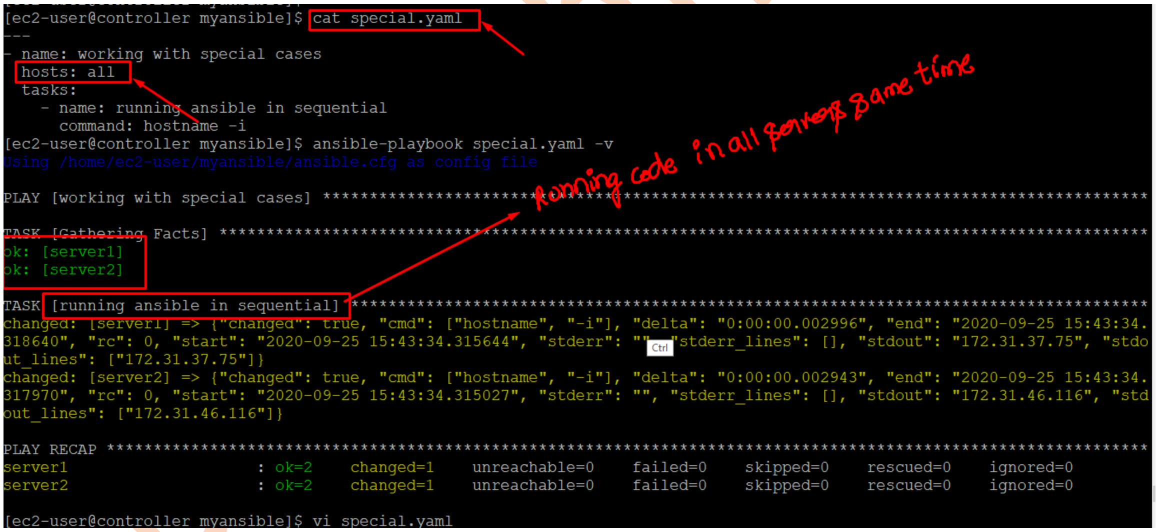


```
      delegate_to: server1
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] **********************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server2]
ok: [server1]

TASK [taking the server ip] ****************************************************************************
changed: [server1] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.002975", "end": "2020-09-25 15:35:05.
877261", "rc": 0, "start": "2020-09-25 15:35:05.874286", "stderr": "", "stderr_lines": [], "stdout": "172.31.37.75", "stdo
ut_lines": ["172.31.37.75"]}
changed: [server2] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.002952", "end": "2020-09-25 15:35:05.
878624", "rc": 0, "start": "2020-09-25 15:35:05.875672", "stderr": "", "stderr_lines": [], "stdout": "172.31.46.116", "std
out_lines": ["172.31.46.116"]}

TASK [working with delegate_to] ************************************************************************
changed: [server1 -> server1] => {"changed": true, "cmd": ["hostname"], "delta": "0:00:00.006207", "end": "2020-09-25 15:3
5:06.467342", "rc": 0, "start": "2020-09-25 15:35:06.461135", "stderr": "", "stderr_lines": [], "stdout": "server1", "stdo
ut_lines": ["server1"]}
changed: [server2 -> server1] => {"changed": true, "cmd": ["hostname"], "delta": "0:00:00.004880", "end": "2020-09-25 15:3
5:06.545619", "rc": 0, "start": "2020-09-25 15:35:06.540739", "stderr": "", "stderr_lines": [], "stdout": "server1", "stdo
ut_lines": ["server1"]}

PLAY RECAP *********************************************************************************************
server1                    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server2                    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ ping server1
```

In both servers o/p is same

Serial:

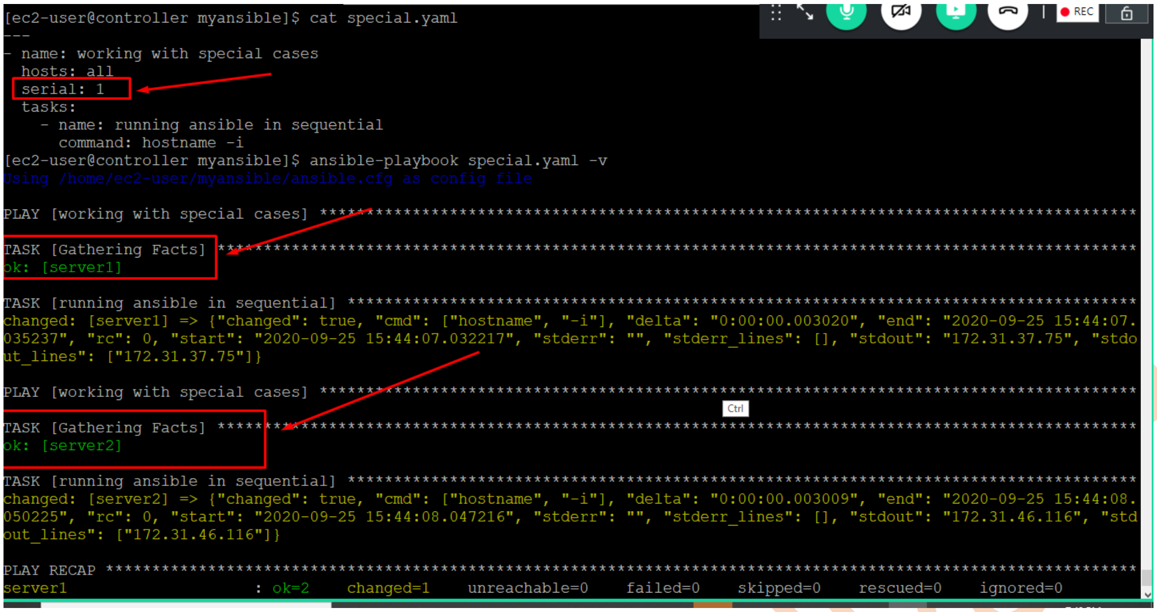
Parallel execution example:



```
[ec2-user@controller myansible]$ cat special.yaml
---
- name: working with special cases
  hosts: all
  tasks:
    - name: running ansible in sequential
      command: hostname -i
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] **********************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [server1]
ok: [server2]

TASK [running ansible in sequential] *******************************************************************
changed: [server1] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.002996", "end": "2020-09-25 15:43:34.
318640", "rc": 0, "start": "2020-09-25 15:43:34.315644", "stderr": "", "stderr_lines": [], "stdout": "172.31.37.75", "stdo
ut_lines": ["172.31.37.75"]}
changed: [server2] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.002943", "end": "2020-09-25 15:43:34.
317970", "rc": 0, "start": "2020-09-25 15:43:34.315027", "stderr": "", "stderr_lines": [], "stdout": "172.31.46.116", "std
out_lines": ["172.31.46.116"]}

PLAY RECAP *********************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ vi special.yaml
```

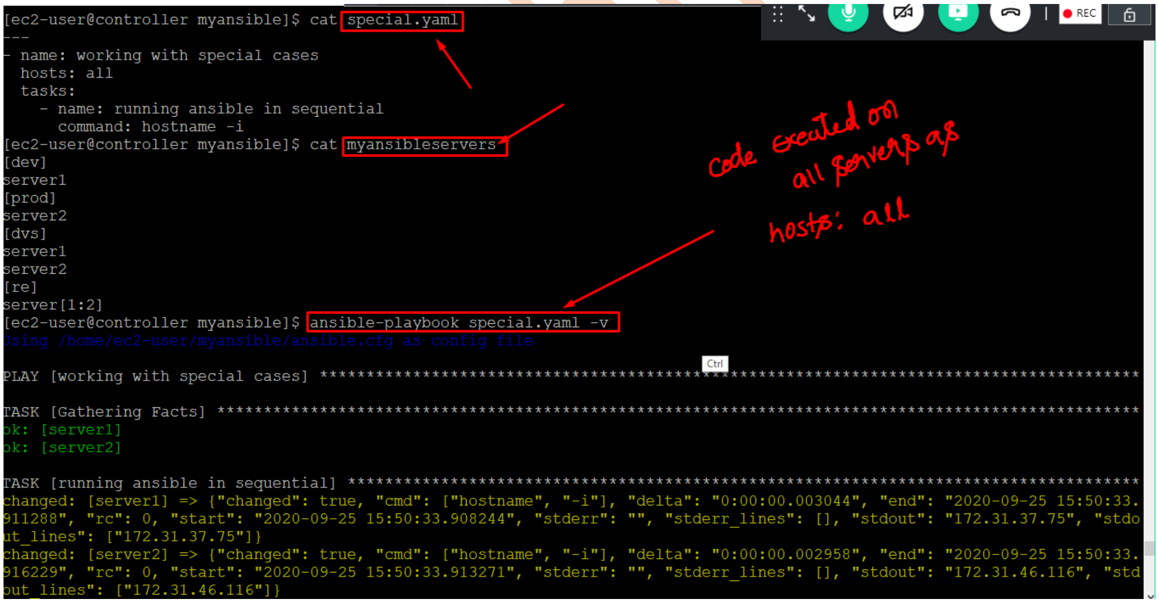
Running code in all servers same time

Converting parallel to sequential execution:



```
[ec2-user@controller myansible]$ cat special.yaml
---
- name: working with special cases
  hosts: all
  serial: 1
  tasks:
    - name: running ansible in sequential
      command: hostname -i
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] **************************************************************************************

TASK [Gathering Facts] *************************************************************************************************
ok: [server1]

TASK [running ansible in sequential] ***********************************************************************************
changed: [server1] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.003020", "end": "2020-09-25 15:44:07.
035237", "rc": 0, "start": "2020-09-25 15:44:07.032217", "stderr": "", "stderr_lines": [], "stdout": "172.31.37.75", "stdo
ut_lines": ["172.31.37.75"]}

PLAY [working with special cases] **************************************************************************************

TASK [Gathering Facts] *************************************************************************************************
ok: [server2]

TASK [running ansible in sequential] ***********************************************************************************
changed: [server2] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.003009", "end": "2020-09-25 15:44:08.
050225", "rc": 0, "start": "2020-09-25 15:44:08.047216", "stderr": "", "stderr_lines": [], "stdout": "172.31.46.116", "std
out_lines": ["172.31.46.116"]}

PLAY RECAP *************************************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

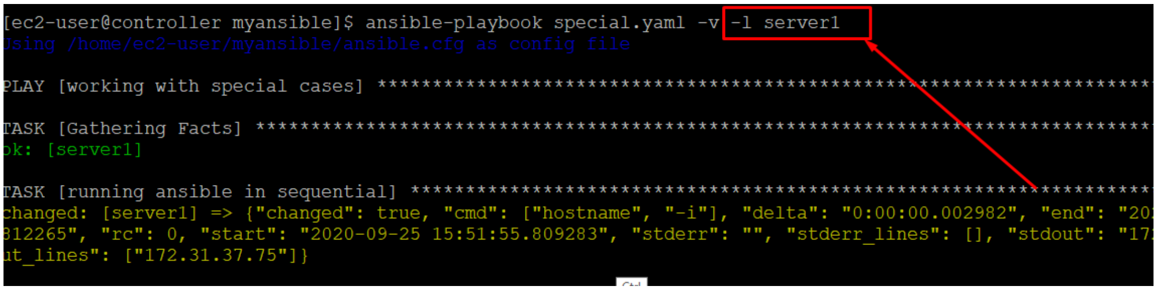
Executing code on a single server/group/all from commandline:



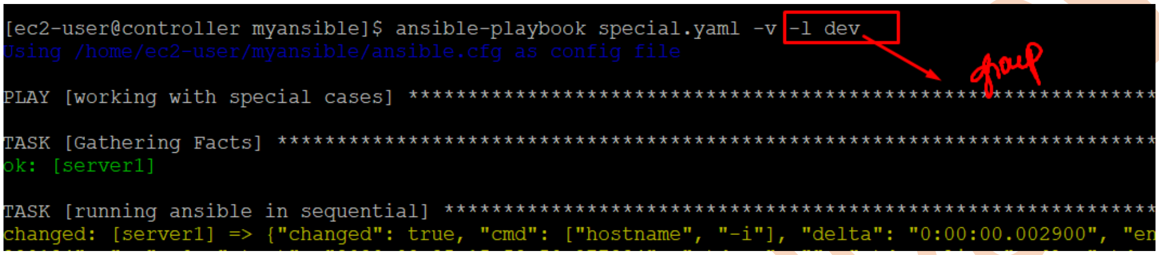
```
[ec2-user@controller myansible]$ cat special.yaml
---
- name: working with special cases
  hosts: all
  tasks:
    - name: running ansible in sequential
      command: hostname -i
[ec2-user@controller myansible]$ cat myansibleservers
[dev]
server1
[prod]
server2
[dvs]
server1
server2
[re]
server[1:2]
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] **************************************************************************************

TASK [Gathering Facts] *************************************************************************************************
ok: [server1]
ok: [server2]

TASK [running ansible in sequential] ***********************************************************************************
changed: [server1] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.003044", "end": "2020-09-25 15:50:33.
911288", "rc": 0, "start": "2020-09-25 15:50:33.908244", "stderr": "", "stderr_lines": [], "stdout": "172.31.37.75", "stdo
ut_lines": ["172.31.37.75"]}
changed: [server2] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.002958", "end": "2020-09-25 15:50:33.
916229", "rc": 0, "start": "2020-09-25 15:50:33.913271", "stderr": "", "stderr_lines": [], "stdout": "172.31.46.116", "std
out_lines": ["172.31.46.116"]}
```

Code Executed on all Servers as hosts: all

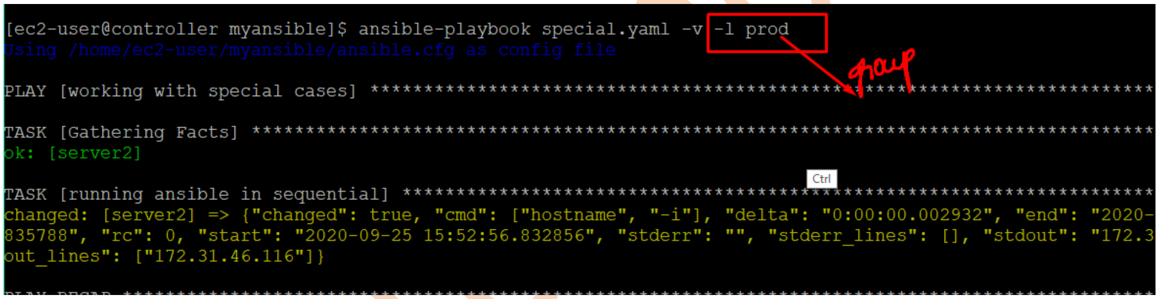
Now I want to execute the code in a sinle/group/all servers without modify the files/code



```
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v -l server1
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] ***********************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server1]

TASK [running ansible in sequential] ********************************************************************
changed: [server1] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.002982", "end": "20
812265", "rc": 0, "start": "2020-09-25 15:51:55.809283", "stderr": "", "stderr_lines": [], "stdout": "17
ut_lines": ["172.31.37.75"]}
```

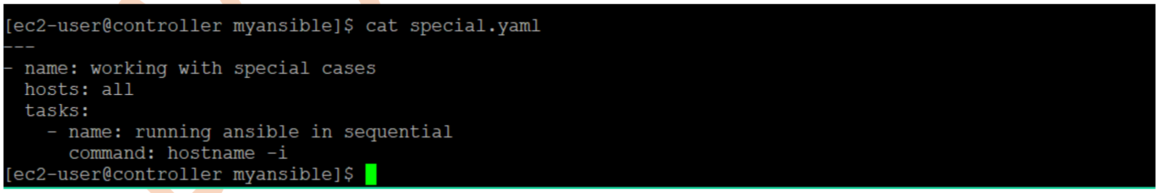


```
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v -l dev
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] ************************************************************************

TASK [Gathering Facts] ***********************************************************************************
ok: [server1]

TASK [running ansible in sequential] *********************************************************************
changed: [server1] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.002900", "en
```

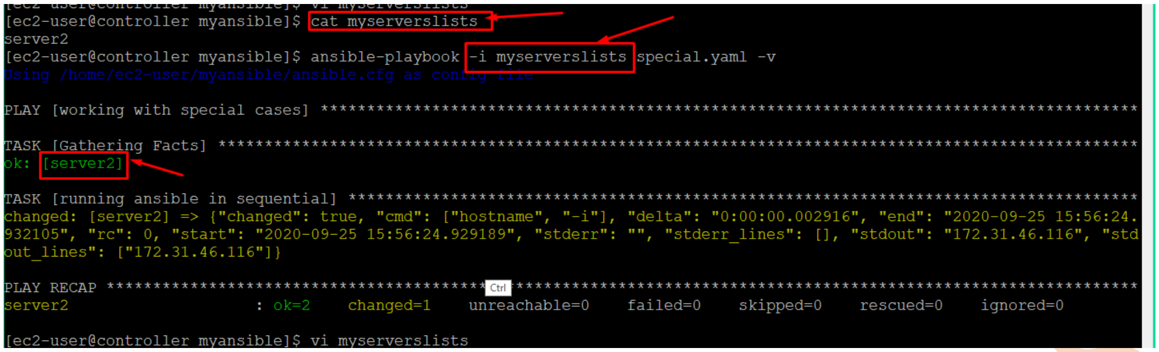


```
[ec2-user@controller myansible]$ ansible-playbook special.yaml -v -l prod
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] ************************************************************************

TASK [Gathering Facts] ***********************************************************************************
ok: [server2]

TASK [running ansible in sequential] *********************************************************************
changed: [server2] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.002932", "end": "2020-
835788", "rc": 0, "start": "2020-09-25 15:52:56.832856", "stderr": "", "stderr_lines": [], "stdout": "172.3
out_lines": ["172.31.46.116"]}
```

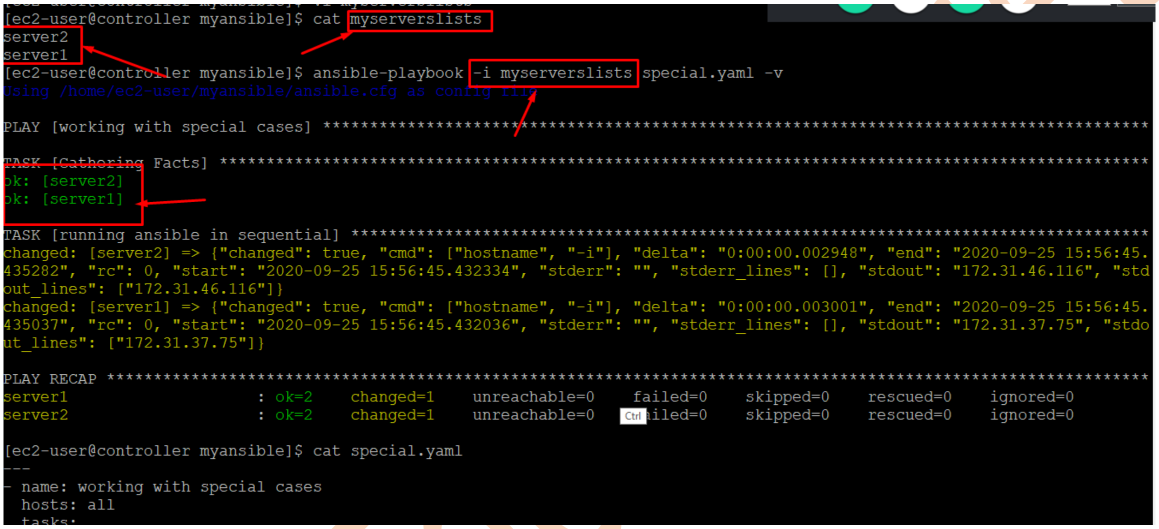
Executing code from custom servers list :



```
[ec2-user@controller myansible]$ cat special.yaml
---
- name: working with special cases
  hosts: all
  tasks:
    - name: running ansible in sequential
      command: hostname -i
[ec2-user@controller myansible]$
```



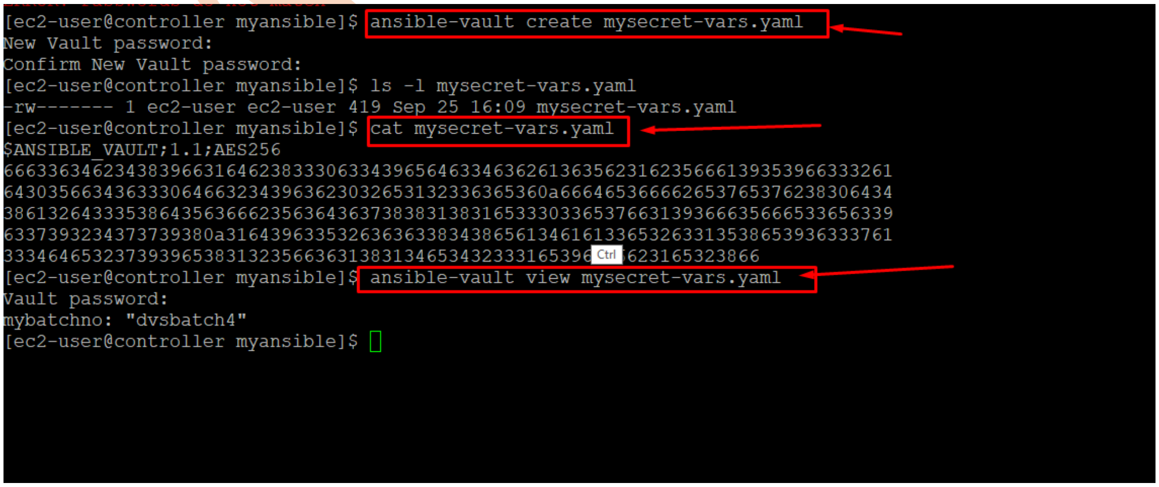
```
[ec2-user@controller myansible]$ cat myserverslists
server2
[ec2-user@controller myansible]$ ansible-playbook -i myserverslists special.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] *************************************************************************

TASK [Gathering Facts] ************************************************************************************
ok: [server2]

TASK [running ansible in sequential] **********************************************************************
changed: [server2] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.002916", "end": "2020-09-25 15:56:24.
932105", "rc": 0, "start": "2020-09-25 15:56:24.929189", "stderr": "", "stderr_lines": [], "stdout": "172.31.46.116", "std
out_lines": ["172.31.46.116"]}

PLAY RECAP ************************************************************************************************
server2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ vi myserverslists
```



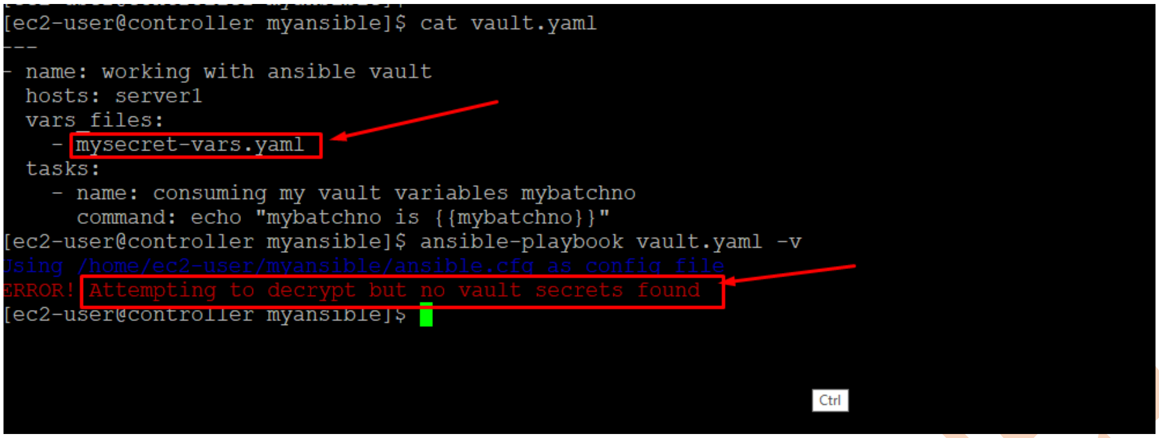
```
[ec2-user@controller myansible]$ cat myserverslists
server2
server1
[ec2-user@controller myansible]$ ansible-playbook -i myserverslists special.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with special cases] *************************************************************************

TASK [Gathering Facts] ************************************************************************************
ok: [server2]
ok: [server1]

TASK [running ansible in sequential] **********************************************************************
changed: [server2] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.002948", "end": "2020-09-25 15:56:45.
435282", "rc": 0, "start": "2020-09-25 15:56:45.432334", "stderr": "", "stderr_lines": [], "stdout": "172.31.46.116", "std
out_lines": ["172.31.46.116"]}
changed: [server1] => {"changed": true, "cmd": ["hostname", "-i"], "delta": "0:00:00.003001", "end": "2020-09-25 15:56:45.
435037", "rc": 0, "start": "2020-09-25 15:56:45.432036", "stderr": "", "stderr_lines": [], "stdout": "172.31.37.75", "stdo
ut_lines": ["172.31.37.75"]}

PLAY RECAP ************************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ cat special.yaml
---
- name: working with special cases
  hosts: all
  tasks:
```

Ansible Vault:

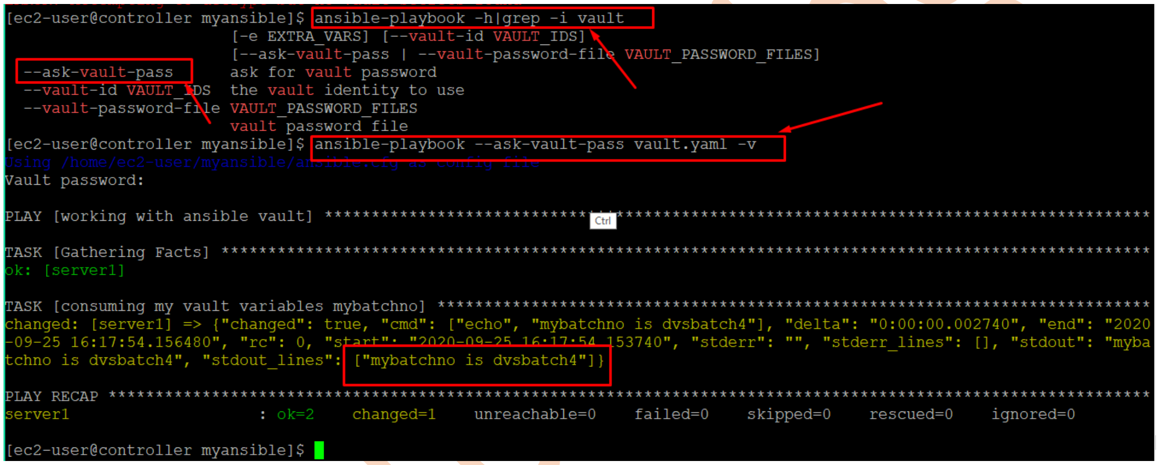
Creating a secret file with all the variables



```
[ec2-user@controller myansible]$ ansible-vault create mysecret-vars.yaml
New Vault password:
Confirm New Vault password:
[ec2-user@controller myansible]$ ls -l mysecret-vars.yaml
-rw------- 1 ec2-user ec2-user 419 Sep 25 16:09 mysecret-vars.yaml
[ec2-user@controller myansible]$ cat mysecret-vars.yaml
$ANSIBLE_VAULT;1.1;AES256
66633634623438396631646238333063343965646334636261363562316235666139353966333261
6430356634363330646632343963623032653132336365360a666465366662653765376238306434
38613264333538643563666235636436373838313831653330336537663139366635666533656339
6337393234373739380a31643963353236363633383438656134616133653263313538653936333761
33346465323739396538313235663631383134653432333165396 623165323866
[ec2-user@controller myansible]$ ansible-vault view mysecret-vars.yaml
Vault password:
mybatchno: "dvsbatch4"
[ec2-user@controller myansible]$ 
```

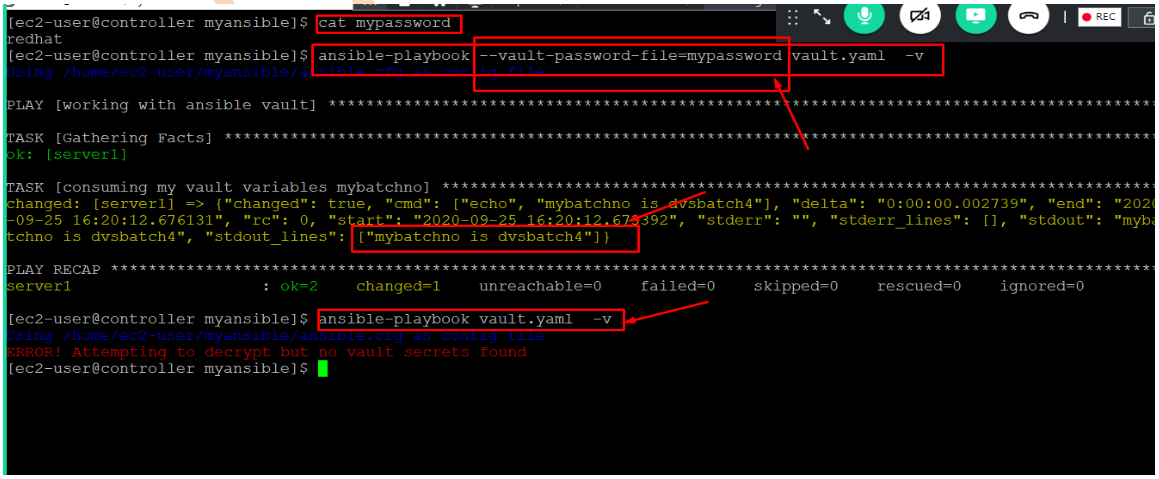


```
[ec2-user@controller myansible]$ cat vault.yaml
---
- name: working with ansible vault
  hosts: server1
  vars_files:
    - mysecret-vars.yaml
  tasks:
    - name: consuming my vault variables mybatchno
      command: echo "mybatchno is {{mybatchno}}"
[ec2-user@controller myansible]$ ansible-playbook vault.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file
ERROR! Attempting to decrypt but no vault secrets found
[ec2-user@controller myansible]$
```



```
[ec2-user@controller myansible]$ ansible-playbook -h|grep -i vault
                                 [-e EXTRA_VARS] [--vault-id VAULT_IDS]
                                 [--ask-vault-pass | --vault-password-file VAULT_PASSWORD_FILES]
  --ask-vault-pass      ask for vault password
  --vault-id VAULT_IDS  the vault identity to use
  --vault-password-file VAULT_PASSWORD_FILES
                        vault password file
[ec2-user@controller myansible]$ ansible-playbook --ask-vault-pass vault.yaml -v
Using /home/ec2-user/myansible/ansible.cfg as config file
Vault password:

PLAY [working with ansible vault] ********************************************************************

TASK [Gathering Facts] *******************************************************************************
ok: [server1]

TASK [consuming my vault variables mybatchno] ********************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "mybatchno is dvsbatch4"], "delta": "0:00:00.002740", "end": "2020
-09-25 16:17:54.156480", "rc": 0, "start": "2020-09-25 16:17:54.153740", "stderr": "", "stderr_lines": [], "stdout": "myba
tchno is dvsbatch4", "stdout_lines": ["mybatchno is dvsbatch4"]}

PLAY RECAP *******************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

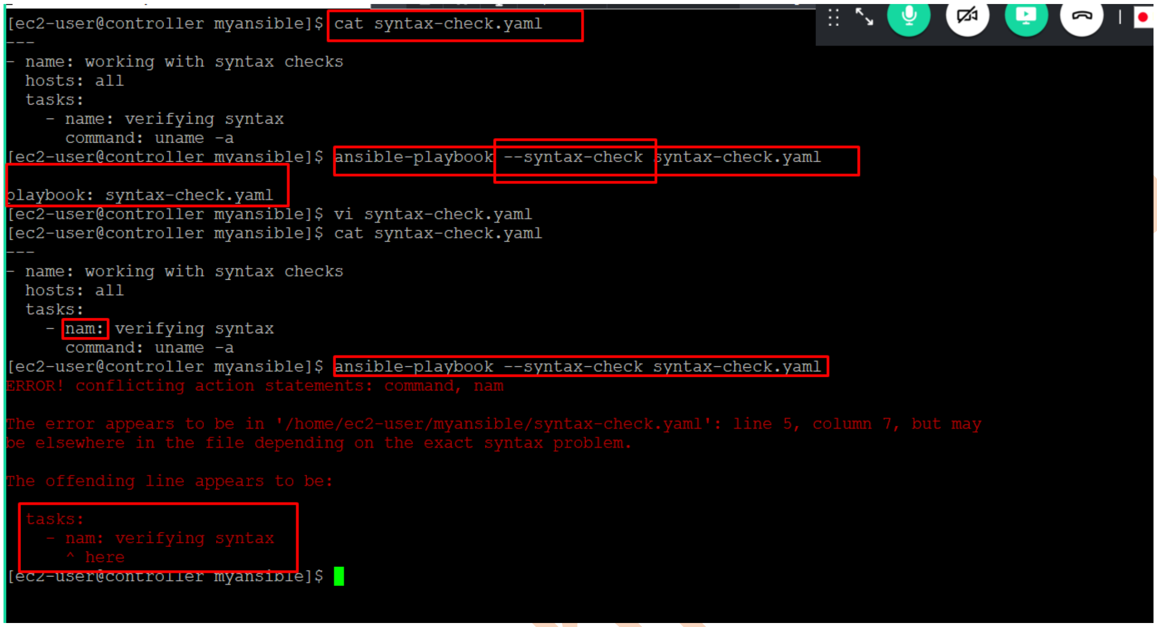
Passing password from a file:



```
[ec2-user@controller myansible]$ cat mypassword
redhat
[ec2-user@controller myansible]$ ansible-playbook --vault-password-file=mypassword vault.yaml  -v
Using /home/ec2-user/myansible/ansible.cfg as config file

PLAY [working with ansible vault] ********************************************************************

TASK [Gathering Facts] *******************************************************************************
ok: [server1]

TASK [consuming my vault variables mybatchno] ********************************************************
changed: [server1] => {"changed": true, "cmd": ["echo", "mybatchno is dvsbatch4"], "delta": "0:00:00.002739", "end": "2020
-09-25 16:20:12.676131", "rc": 0, "start": "2020-09-25 16:20:12.673392", "stderr": "", "stderr_lines": [], "stdout": "myba
tchno is dvsbatch4", "stdout_lines": ["mybatchno is dvsbatch4"]}

PLAY RECAP *******************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ ansible-playbook vault.yaml  -v
Using /home/ec2-user/myansible/ansible.cfg as config file
ERROR! Attempting to decrypt but no vault secrets found
[ec2-user@controller myansible]$
```

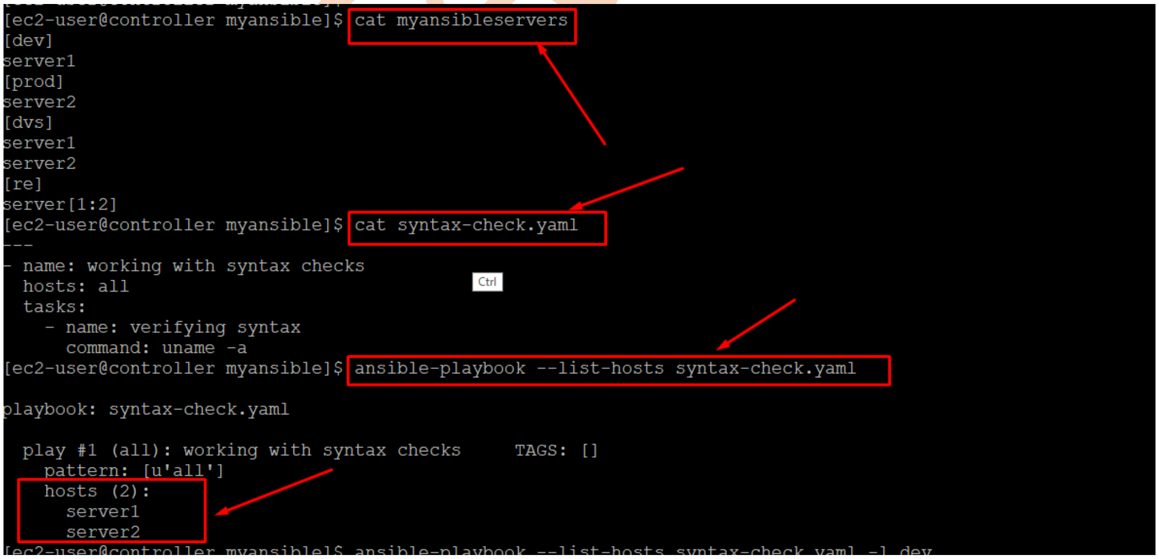
9. Checking Your Playbook before Execution

1. Syntax Check: (--syntax-check)



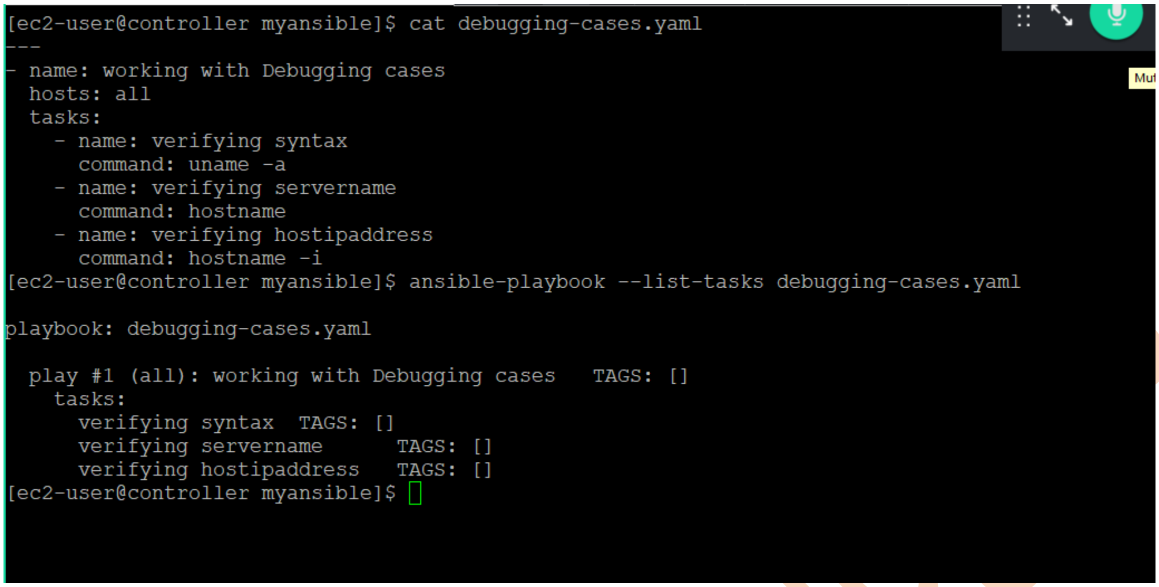
```
[ec2-user@controller myansible]$ cat syntax-check.yaml
---
- name: working with syntax checks
  hosts: all
  tasks:
    - name: verifying syntax
      command: uname -a
[ec2-user@controller myansible]$ ansible-playbook --syntax-check syntax-check.yaml

playbook: syntax-check.yaml
[ec2-user@controller myansible]$ vi syntax-check.yaml
[ec2-user@controller myansible]$ cat syntax-check.yaml
---
- name: working with syntax checks
  hosts: all
  tasks:
    - nam: verifying syntax
      command: uname -a
[ec2-user@controller myansible]$ ansible-playbook --syntax-check syntax-check.yaml
ERROR! conflicting action statements: command, nam

The error appears to be in '/home/ec2-user/myansible/syntax-check.yaml': line 5, column 7, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

  tasks:
    - nam: verifying syntax
      ^ here
[ec2-user@controller myansible]$
```

2. List Hosts: (--list-hosts)



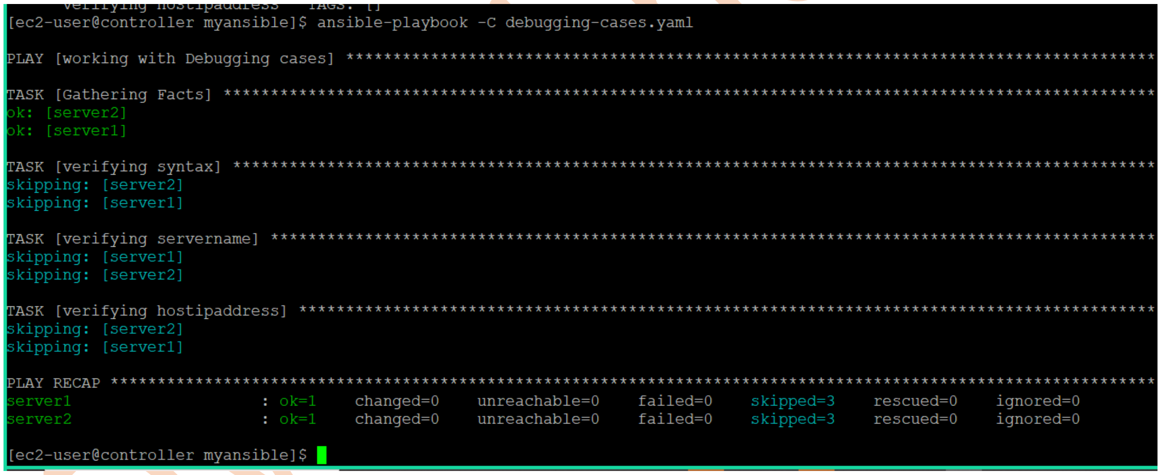
```
[ec2-user@controller myansible]$ cat myansibleservers
[dev]
server1
[prod]
server2
[dvs]
server1
server2
[re]
server[1:2]
[ec2-user@controller myansible]$ cat syntax-check.yaml
---
- name: working with syntax checks
  hosts: all
  tasks:
    - name: verifying syntax
      command: uname -a
[ec2-user@controller myansible]$ ansible-playbook --list-hosts syntax-check.yaml

playbook: syntax-check.yaml

  play #1 (all): working with syntax checks     TAGS: []
    pattern: [u'all']
    hosts (2):
      server1
      server2
[ec2-user@controller myansible]$ ansible-playbook --list-hosts syntax-check.yaml -l dev
```

3. Listing All the Tasks: (--list-tasks)



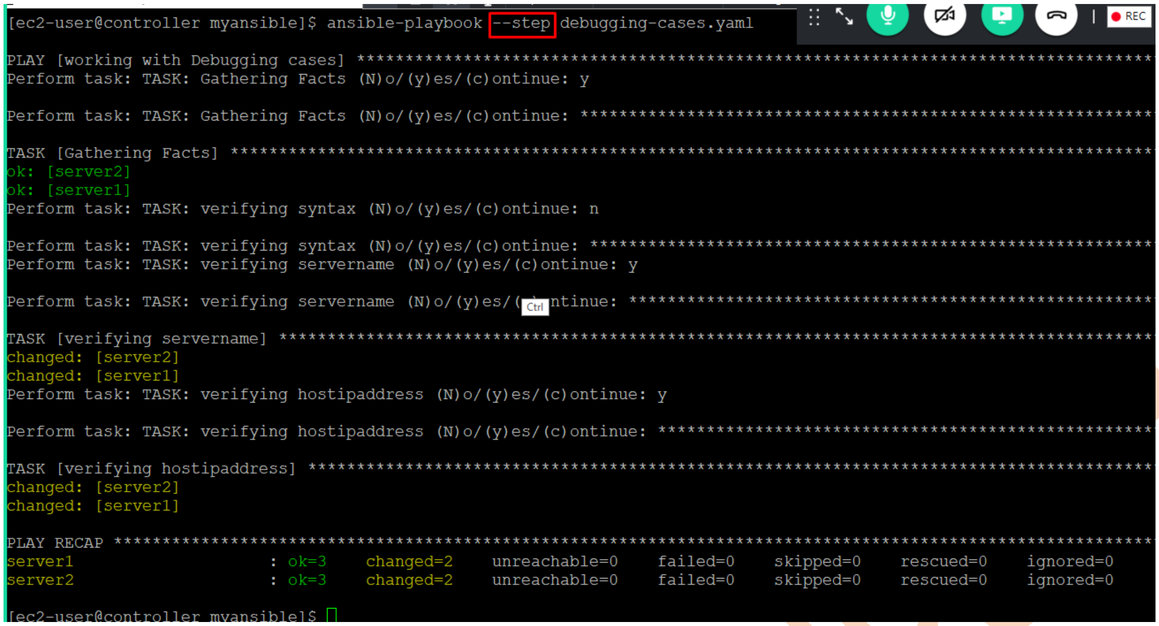
```
[ec2-user@controller myansible]$ cat debugging-cases.yaml
---
- name: working with Debugging cases
  hosts: all
  tasks:
    - name: verifying syntax
      command: uname -a
    - name: verifying servername
      command: hostname
    - name: verifying hostipaddress
      command: hostname -i
[ec2-user@controller myansible]$ ansible-playbook --list-tasks debugging-cases.yaml

playbook: debugging-cases.yaml

  play #1 (all): working with Debugging cases   TAGS: []
    tasks:
      verifying syntax  TAGS: []
      verifying servername      TAGS: []
      verifying hostipaddress   TAGS: []
[ec2-user@controller myansible]$
```

4. Check Mode: (-C)



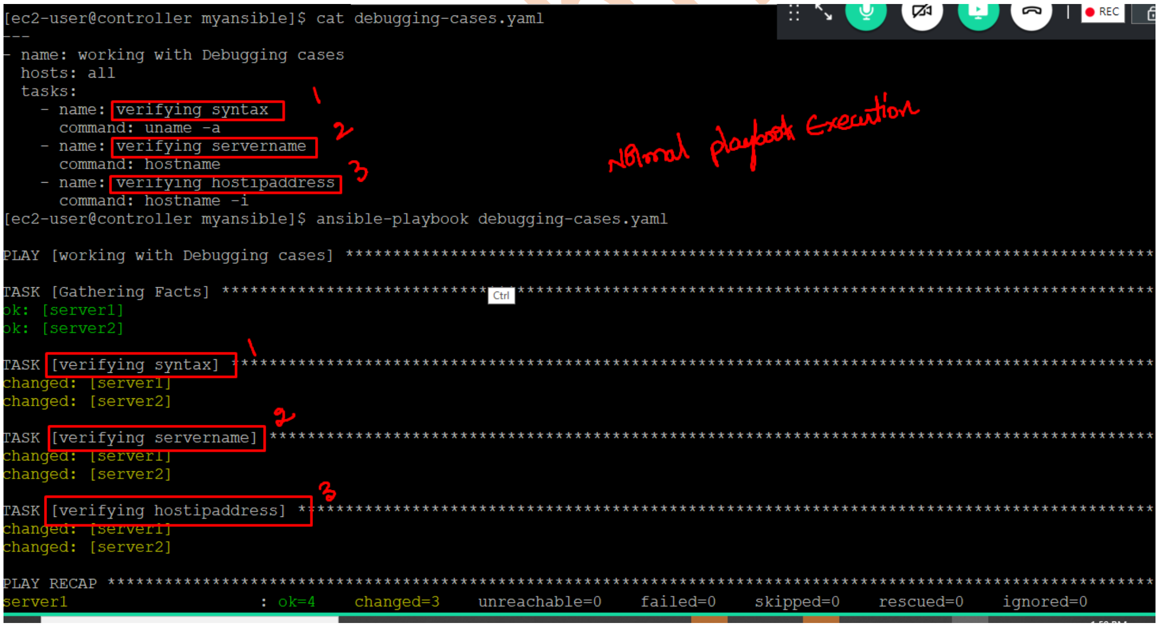
```
      verifying hostipaddress   TAGS: []
[ec2-user@controller myansible]$ ansible-playbook -C debugging-cases.yaml

PLAY [working with Debugging cases] ***************************************************************************

TASK [Gathering Facts] ****************************************************************************************
ok: [server2]
ok: [server1]

TASK [verifying syntax] ***************************************************************************************
skipping: [server2]
skipping: [server1]

TASK [verifying servername] ***********************************************************************************
skipping: [server1]
skipping: [server2]

TASK [verifying hostipaddress] ********************************************************************************
skipping: [server2]
skipping: [server1]

PLAY RECAP ****************************************************************************************************
server1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
server2                    : ok=1    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

5. Step: (--step)

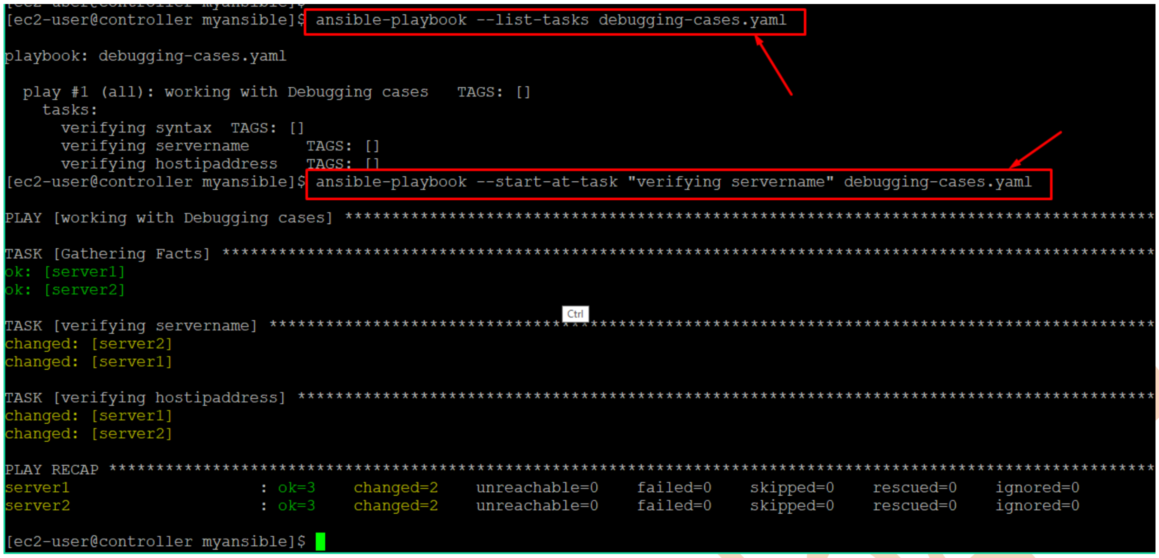


```
[ec2-user@controller myansible]$ ansible-playbook --step debugging-cases.yaml

PLAY [working with Debugging cases] ****************************************************************
Perform task: TASK: Gathering Facts (N)o/(y)es/(c)ontinue: y

Perform task: TASK: Gathering Facts (N)o/(y)es/(c)ontinue: ****************************************

TASK [Gathering Facts] *****************************************************************************
ok: [server2]
ok: [server1]
Perform task: TASK: verifying syntax (N)o/(y)es/(c)ontinue: n

Perform task: TASK: verifying syntax (N)o/(y)es/(c)ontinue: ***************************************
Perform task: TASK: verifying servername (N)o/(y)es/(c)ontinue: y

Perform task: TASK: verifying servername (N)o/(y)es/(c)ontinue: ***********************************

TASK [verifying servername] ************************************************************************
changed: [server2]
changed: [server1]
Perform task: TASK: verifying hostipaddress (N)o/(y)es/(c)ontinue: y

Perform task: TASK: verifying hostipaddress (N)o/(y)es/(c)ontinue: ********************************

TASK [verifying hostipaddress] *********************************************************************
changed: [server2]
changed: [server1]

PLAY RECAP *****************************************************************************************
server1                    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server2                    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

6. Start-at-Task: (--start-at-task)

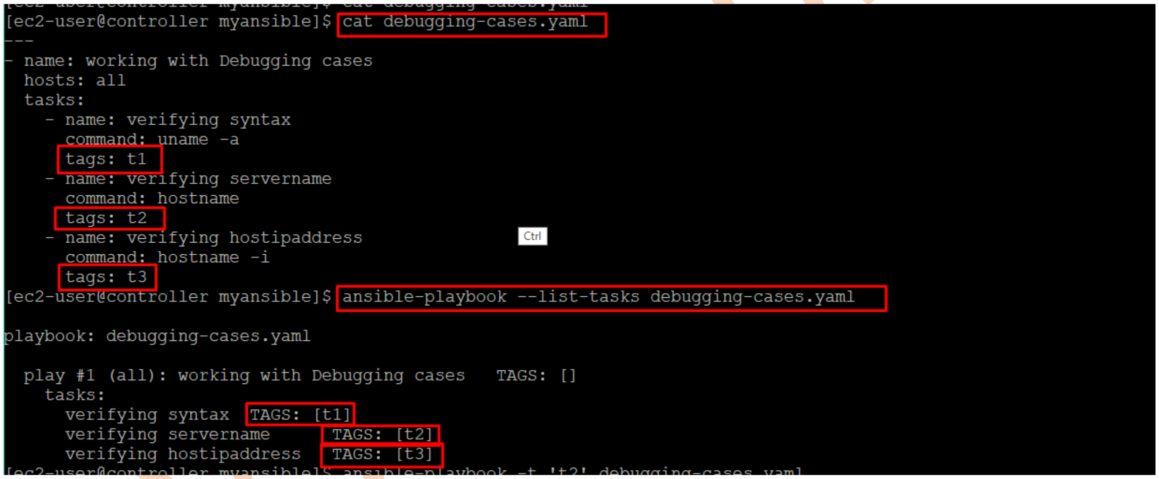


```
[ec2-user@controller myansible]$ cat debugging-cases.yaml
---
- name: working with Debugging cases
  hosts: all
  tasks:
    - name: verifying syntax
      command: uname -a
    - name: verifying servername
      command: hostname
    - name: verifying hostipaddress
      command: hostname -i
[ec2-user@controller myansible]$ ansible-playbook debugging-cases.yaml

PLAY [working with Debugging cases] ****************************************************************

TASK [Gathering Facts] *****************************************************************************
ok: [server1]
ok: [server2]

TASK [verifying syntax] ****************************************************************************
changed: [server1]
changed: [server2]

TASK [verifying servername] ************************************************************************
changed: [server1]
changed: [server2]

TASK [verifying hostipaddress] *********************************************************************
changed: [server1]
changed: [server2]

PLAY RECAP *****************************************************************************************
server1                   : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

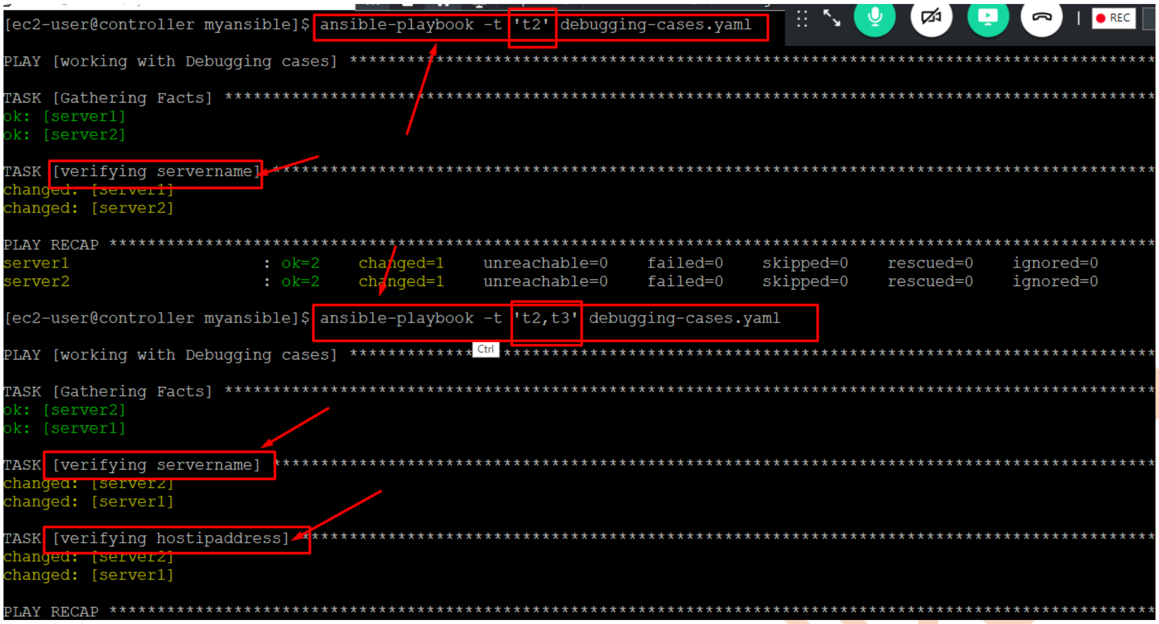


```
[ec2-user@controller myansible]$ ansible-playbook --list-tasks debugging-cases.yaml

playbook: debugging-cases.yaml

  play #1 (all): working with Debugging cases   TAGS: []
    tasks:
      verifying syntax  TAGS: []
      verifying servername      TAGS: []
      verifying hostipaddress   TAGS: []
[ec2-user@controller myansible]$ ansible-playbook --start-at-task "verifying servername" debugging-cases.yaml

PLAY [working with Debugging cases] ********************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [server1]
ok: [server2]

TASK [verifying servername] ****************************************************************************************
changed: [server2]
changed: [server1]

TASK [verifying hostipaddress] *************************************************************************************
changed: [server1]
changed: [server2]

PLAY RECAP *********************************************************************************************************
server1                    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server2                    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$
```

7. Tagging: (tags/-t)

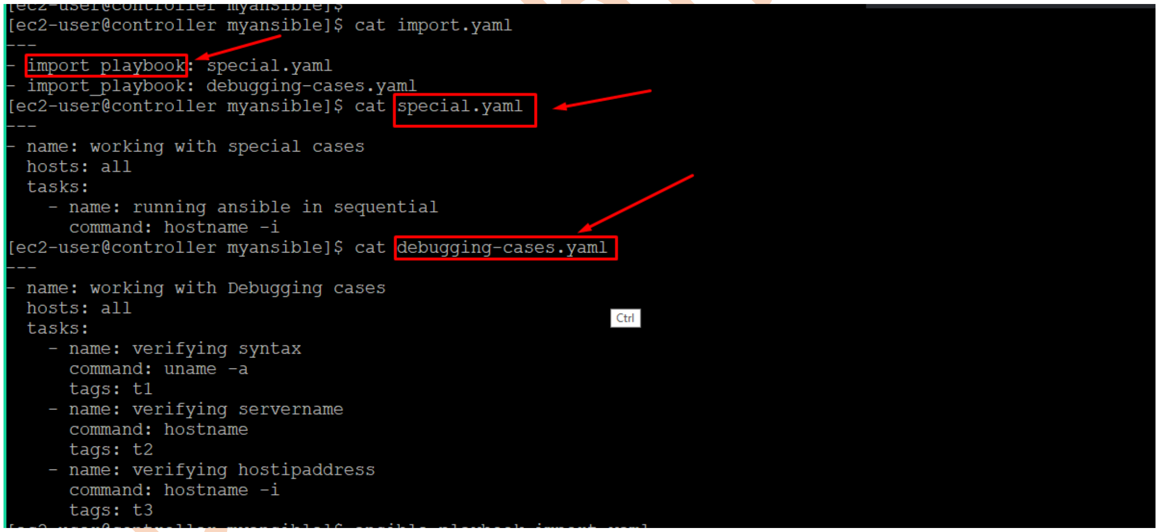


```
[ec2-user@controller myansible]$ cat debugging-cases.yaml
---
- name: working with Debugging cases
  hosts: all
  tasks:
    - name: verifying syntax
      command: uname -a
      tags: t1
    - name: verifying servername
      command: hostname
      tags: t2
    - name: verifying hostipaddress
      command: hostname -i
      tags: t3
[ec2-user@controller myansible]$ ansible-playbook --list-tasks debugging-cases.yaml

playbook: debugging-cases.yaml

  play #1 (all): working with Debugging cases   TAGS: []
    tasks:
      verifying syntax  TAGS: [t1]
      verifying servername      TAGS: [t2]
      verifying hostipaddress   TAGS: [t3]
[ec2-user@controller myansible]$ ansible-playbook -t 't2' debugging-cases.yaml
```

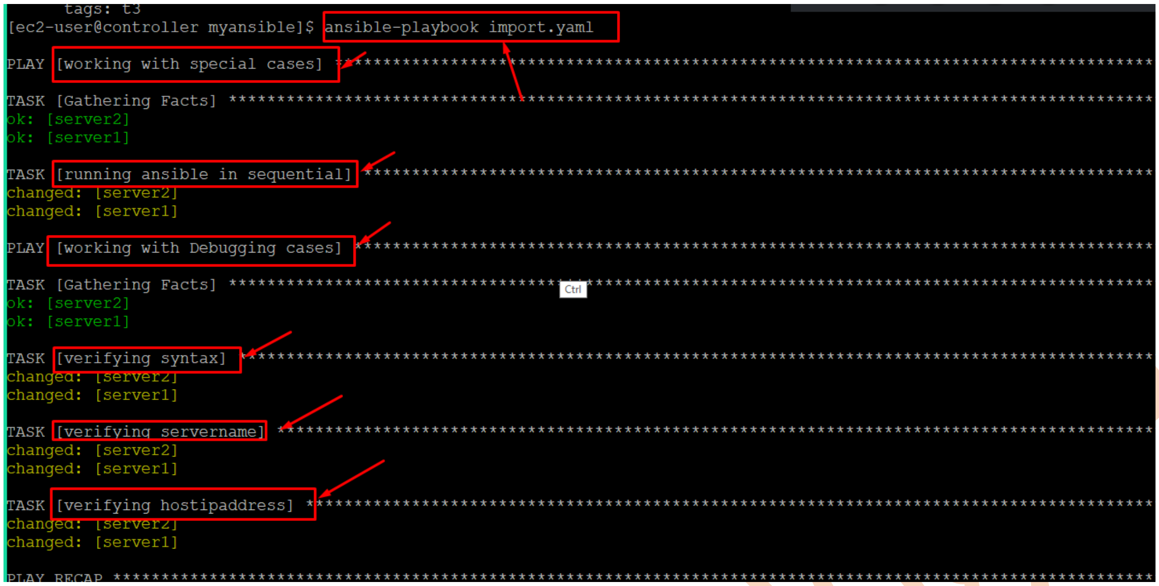


```
[ec2-user@controller myansible]$ ansible-playbook -t 't2' debugging-cases.yaml

PLAY [working with Debugging cases] *********************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server1]
ok: [server2]

TASK [verifying servername] *****************************************************************************
changed: [server1]
changed: [server2]

PLAY RECAP **********************************************************************************************
server1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@controller myansible]$ ansible-playbook -t 't2,t3' debugging-cases.yaml

PLAY [working with Debugging cases] *********************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server2]
ok: [server1]

TASK [verifying servername] *****************************************************************************
changed: [server2]
changed: [server1]

TASK [verifying hostipaddress] **************************************************************************
changed: [server2]
changed: [server1]

PLAY RECAP **********************************************************************************************
```

8. import_playbook



```
[ec2-user@controller myansible]$
[ec2-user@controller myansible]$ cat import.yaml
---
- import_playbook: special.yaml
- import_playbook: debugging-cases.yaml
[ec2-user@controller myansible]$ cat special.yaml
---
- name: working with special cases
  hosts: all
  tasks:
    - name: running ansible in sequential
      command: hostname -i
[ec2-user@controller myansible]$ cat debugging-cases.yaml
---
- name: working with Debugging cases
  hosts: all
  tasks:
    - name: verifying syntax
      command: uname -a
      tags: t1
    - name: verifying servername
      command: hostname
      tags: t2
    - name: verifying hostipaddress
      command: hostname -i
      tags: t3
[ec2-user@controller myansible]$ ansible-playbook import.yaml
```



```
     tags: t3
[ec2-user@controller myansible]$ ansible-playbook import.yaml

PLAY [working with special cases] ***********************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server2]
ok: [server1]

TASK [running ansible in sequential] ********************************************************************
changed: [server2]
changed: [server1]

PLAY [working with Debugging cases] *********************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [server2]
ok: [server1]

TASK [verifying syntax] *********************************************************************************
changed: [server2]
changed: [server1]

TASK [verifying servername] *****************************************************************************
changed: [server2]
changed: [server1]

TASK [verifying hostipaddress] **************************************************************************
changed: [server2]
changed: [server1]

PLAY RECAP **********************************************************************************************
```

10. Working with ansible playbooks

Example1: Installing and configuring httpd using ansible playbook

```
[ec2-user@controller myansible]$ cat httpd.yaml
---
- name: working with apache installation & configuration
  hosts: all
  tasks:
    - name: Gathering public ipaddress of the server
      command: curl http://169.254.169.254/latest/meta-data/public-ipv4
      register: myip

    - set_fact: mypubip="{{myip.stdout}}"

    - name: Installing apache package
      yum:
        name: httpd
        state: present
    - name: enabling apache package
      systemd:
        name: httpd
        enabled: yes
    - name: copying index.html file
      template: src=index.html dest=/var/www/html/
      notify: httpd restart

    - name: modifying httpd.conf file
      replace:
        path: /etc/httpd/conf/httpd.conf
        regexp: '^#ServerName www.example.com'
        replace: 'ServerName {{mypubip}}'
      notify: httpd restart


  handlers:
    - name: httpd restart
      systemd:
        name: httpd
        state: restarted
```

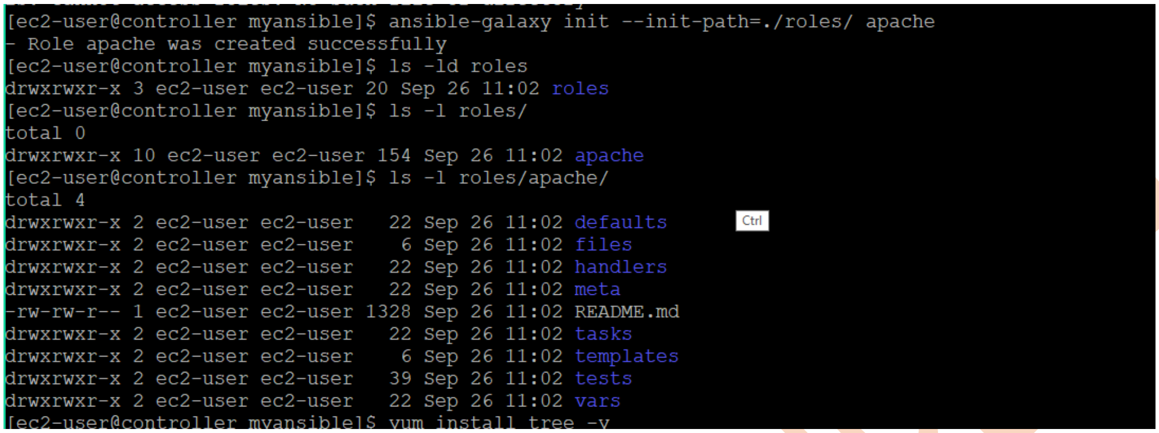
Execution:

```
[ec2-user@controller myansible]$ ansible-playbook httpd.yaml

PLAY [working with apache installation & configuration]
****************************************************************************

TASK [Gathering Facts]
******************************************************************************************
************************
ok: [server2]
ok: [server1]

TASK [Gathering public ipaddress of the server]
**************************************************************************************
[WARNING]: Consider using the get_url or uri module rather than running 'curl'.  If you
need to use command because
get_url or uri is insufficient you can add 'warn: false' to this command task or set
'command_warnings=False' in
ansible.cfg to get rid of this message.
changed: [server2]
changed: [server1]

TASK [set_fact]
******************************************************************************************
********************************
ok: [server1]
ok: [server2]
```

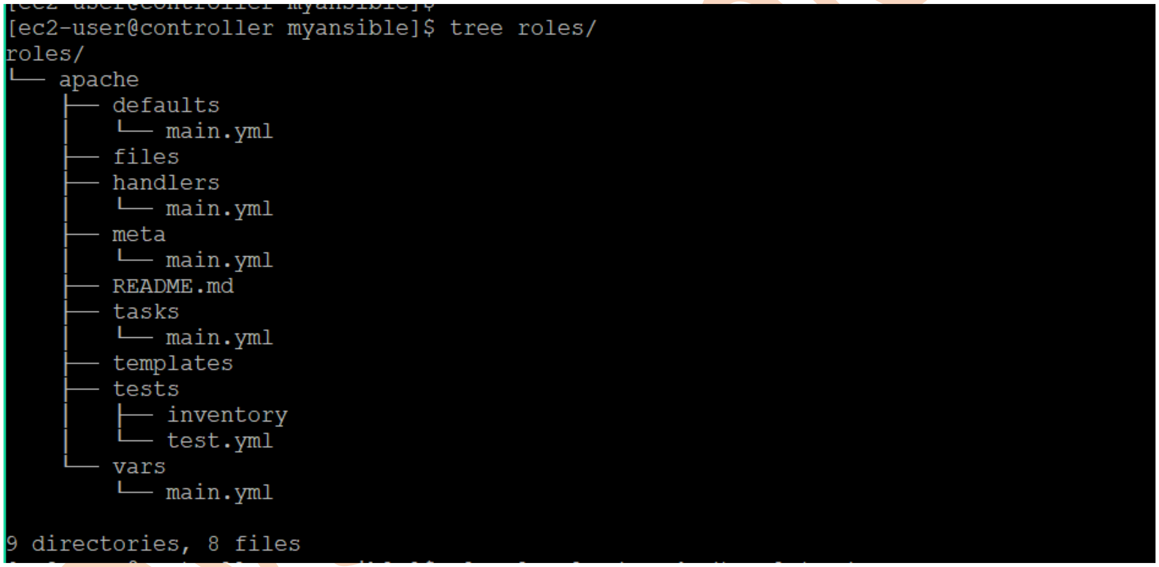
```
TASK [Installing apache package]
*******************************************************************************
***********
ok: [server2]
ok: [server1]

TASK [enabling apache package]
*******************************************************************************
*************
ok: [server2]
ok: [server1]

TASK [copying index.html file]
*******************************************************************************
*************
ok: [server1]
changed: [server2]

TASK [modifying httpd.conf file]
*******************************************************************************
***********
ok: [server2]
ok: [server1]

RUNNING HANDLER [httpd restart]
*******************************************************************************
************
changed: [server2]

PLAY RECAP
*******************************************************************************
*********************************
server1                : ok=7    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
server2                : ok=8    changed=3    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

11. Working with roles

Aligning ansible playbooks to roles



```
[ec2-user@controller myansible]$ ansible-galaxy init --init-path=./roles/ apache
- Role apache was created successfully
[ec2-user@controller myansible]$ ls -ld roles
drwxrwxr-x 3 ec2-user ec2-user 20 Sep 26 11:02 roles
[ec2-user@controller myansible]$ ls -l roles/
total 0
drwxrwxr-x 10 ec2-user ec2-user 154 Sep 26 11:02 apache
[ec2-user@controller myansible]$ ls -l roles/apache/
total 4
drwxrwxr-x 2 ec2-user ec2-user   22 Sep 26 11:02 defaults
drwxrwxr-x 2 ec2-user ec2-user    6 Sep 26 11:02 files
drwxrwxr-x 2 ec2-user ec2-user   22 Sep 26 11:02 handlers
drwxrwxr-x 2 ec2-user ec2-user   22 Sep 26 11:02 meta
-rw-rw-r-- 1 ec2-user ec2-user 1328 Sep 26 11:02 README.md
drwxrwxr-x 2 ec2-user ec2-user   22 Sep 26 11:02 tasks
drwxrwxr-x 2 ec2-user ec2-user    6 Sep 26 11:02 templates
drwxrwxr-x 2 ec2-user ec2-user   39 Sep 26 11:02 tests
drwxrwxr-x 2 ec2-user ec2-user   22 Sep 26 11:02 vars
[ec2-user@controller myansible]$ yum install tree -y
```



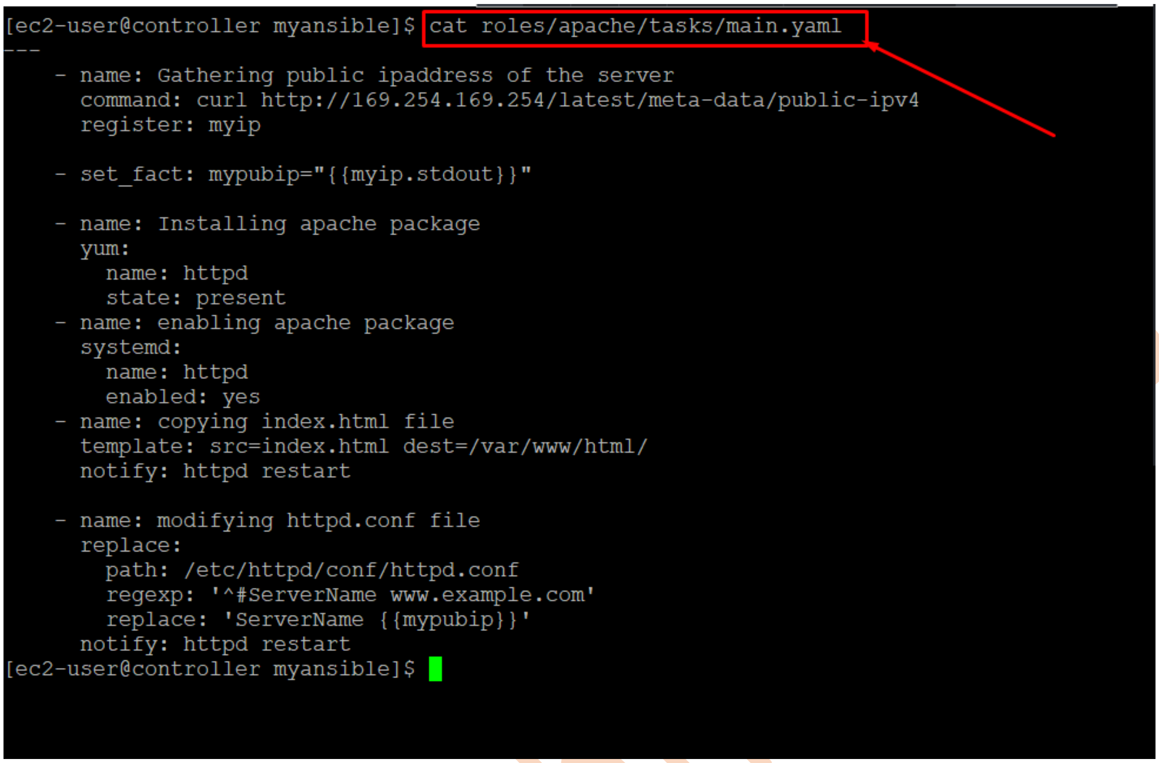
```
[ec2-user@controller myansible]$
[ec2-user@controller myansible]$ tree roles/
roles/
└── apache
    ├── defaults
    │   └── main.yml
    ├── files
    ├── handlers
    │   └── main.yml
    ├── meta
    │   └── main.yml
    ├── README.md
    ├── tasks
    │   └── main.yml
    ├── templates
    ├── tests
    │   ├── inventory
    │   └── test.yml
    └── vars
        └── main.yml

9 directories, 8 files
```

Roles:

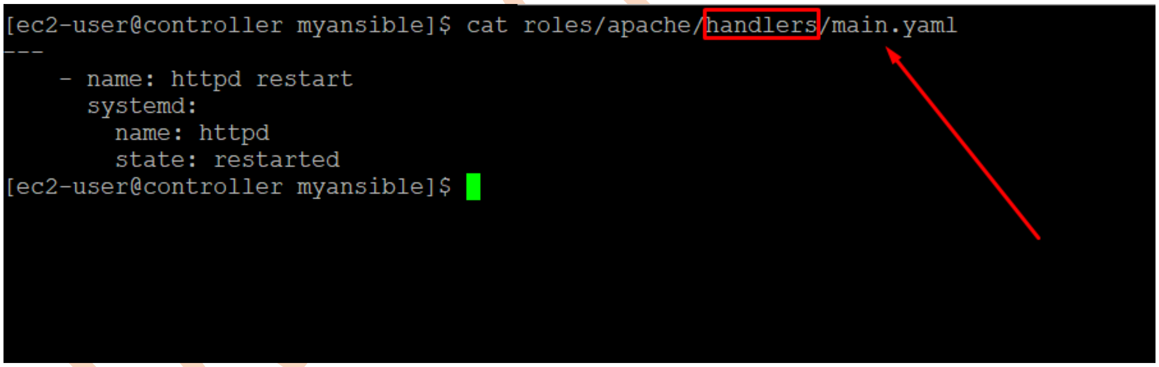
Follow the below steps:

mkdir ./roles/apache -p
mkdir ./roles/apache/templates -p
mkdir ./roles/apache/tasks -p
mkdir ./roles/apache/handlers -p

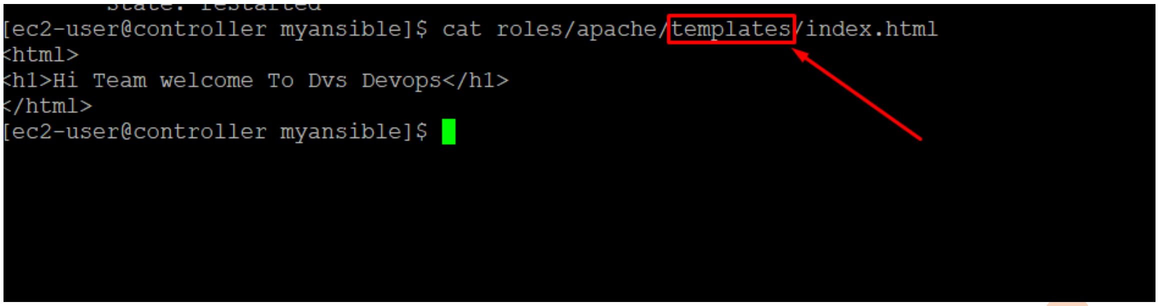
Know segregate your ansible playbook the respective folders like below



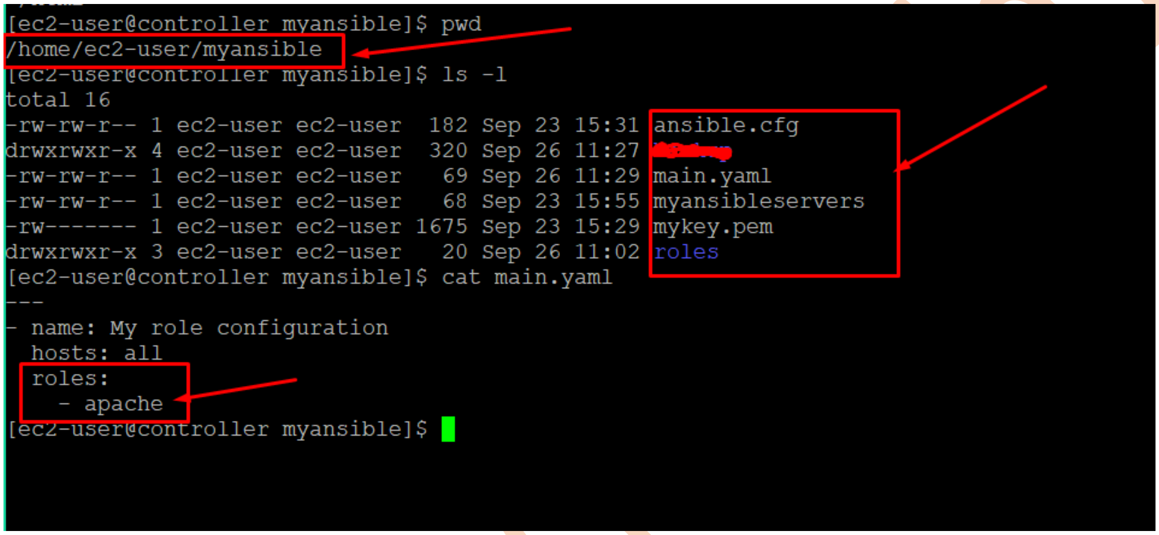
```
[ec2-user@controller myansible]$ cat roles/apache/tasks/main.yaml
---
    - name: Gathering public ipaddress of the server
      command: curl http://169.254.169.254/latest/meta-data/public-ipv4
      register: myip

    - set_fact: mypubip="{{myip.stdout}}"

    - name: Installing apache package
      yum:
        name: httpd
        state: present
    - name: enabling apache package
      systemd:
        name: httpd
        enabled: yes
    - name: copying index.html file
      template: src=index.html dest=/var/www/html/
      notify: httpd restart

    - name: modifying httpd.conf file
      replace:
        path: /etc/httpd/conf/httpd.conf
        regexp: '^#ServerName www.example.com'
        replace: 'ServerName {{mypubip}}'
      notify: httpd restart
[ec2-user@controller myansible]$
```



```
[ec2-user@controller myansible]$ cat roles/apache/handlers/main.yaml
---
    - name: httpd restart
      systemd:
        name: httpd
        state: restarted
[ec2-user@controller myansible]$
```



```
      state: restarted
[ec2-user@controller myansible]$ cat roles/apache/templates/index.html
<html>
<h1>Hi Team welcome To Dvs Devops</h1>
</html>
[ec2-user@controller myansible]$
```



```
[ec2-user@controller myansible]$ pwd
/home/ec2-user/myansible
[ec2-user@controller myansible]$ ls -l
total 16
-rw-rw-r-- 1 ec2-user ec2-user  182 Sep 23 15:31 ansible.cfg
drwxrwxr-x 4 ec2-user ec2-user  320 Sep 26 11:27 [redacted]
-rw-rw-r-- 1 ec2-user ec2-user   69 Sep 26 11:29 main.yaml
-rw-rw-r-- 1 ec2-user ec2-user   68 Sep 23 15:55 myansibleservers
-rw------- 1 ec2-user ec2-user 1675 Sep 23 15:29 mykey.pem
drwxrwxr-x 3 ec2-user ec2-user   20 Sep 26 11:02 roles
[ec2-user@controller myansible]$ cat main.yaml
---
- name: My role configuration
  hosts: all
  roles:
    - apache
[ec2-user@controller myansible]$
```

12. Configuring Word press application

Let's configure our word press application manually.

1. Manual Installation of Database ::

```
yum install mariadb-server
systemctl enable mariadb.service
systemctl restart mariadb.service
```

mysql:

```
CREATE USER 'admin'@'%' IDENTIFIED BY 'admin123';
GRANT ALL PRIVILEGES ON * . * TO 'admin'@'%';
FLUSH PRIVILEGES;
create database devopsdb;
```

Verification::

```
select host, user, password from mysql.user;
DROP USER 'admin'@'localhost';
```

Automating the configuration using ansible roles ::

2. Manual Installation of application

```
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
sudo cp -r wordpress/* /var/www/html/
cd /var/www/html/
sudo cp wp-config-sample.php wp-config.php
```

Post db installation do the below.

```
sudo vi wp-config.php --> add the database details to this file
sudo chown -R apache /var/www
sudo chgrp -R apache /var/www
sudo chmod 775 /var/www
sudo find /var/www -type d -exec sudo chmod 2775 {} \;
sudo find /var/www -type f -exec sudo chmod 0664 {} \;
sudo systemctl restart httpd
```

Automating the configuration using ansible roles ::

Final Code ::
https://github.com/shan5a6/wordpress-mariadb.git

13. Integrating Ansible with AWS

Basic Requirement:

```
# aws configure
[root@master playbooks]# aws configure
AWS Access Key ID [****************EBFA]:
AWS Secret Access Key [****************DOqF]:
Default region name [us-west-2]: eu-west-2
Default output format [json]: json
```

Installtin boto3

```
pip install boto3 boto
```

Note: If anisble is not there make sure you are installing it with "pip install ansible"

If you are getting below error then make sure you are adding "become: False" in the playbook.

```
TASK [creating a normal basic instance]
****************************************************************
fatal: [localhost]: FAILED! => {"changed": false, "msg": "No handler was ready to authenticate. 1 handlers were checked. ['HmacAuthV4Handler'] Check your credentials"}
```

Examples:

1. Creating Ec2 instance using "ec2" module

```
[root@master playbooks]# cat ec2-create.yml
---
- name: working with ec2 instance creation
  vars:
    - serv_count: 0
  hosts: localhost
  become: False
  tasks:
    - name: creating a normal basic instance
      ec2:
           key_name: newkeys_swamy_useast1
           instance_type: t2.micro
           image: ami-97785bed
           wait: yes
           group: opentoworld
           count: "{{serv_count}}"
           vpc_subnet_id: subnet-d88cbcf4
           assign_public_ip: yes
           region: us-east-1
      register: output

    - debug: msg="{{output}}"
```

2. Creating a VPC with a public subnet and security groups.

```
---
- name: creating a vpc
  hosts: localhost
  become: False
  tasks:
    - name: working with VPC creation
      ec2_vpc_net:
          name: "mytestvpc"
          cidr_block: "192.168.0.0/16"
          region: "us-east-1"
          state: "present"
      register: myvpc

# Assigning the vpc id to a variable

    - set_fact: vpc_id="{{myvpc.vpc.id}}"
    - command: echo "{{vpc_id}}"

# Creating a public Subnet

    - name: creating a public Subnet
      ec2_vpc_subnet:
        state: "present"
        vpc_id: "{{vpc_id}}"
        cidr: "192.168.10.0/24"
        resource_tags:
```

```
        Name: "Public Subnet"

    register: my_public_subnet

# Assigning public subnet id to a variable

  - set_fact: pub_sub_id="{{my_public_subnet.subnet.id}}"

# Creating IGW for the vpc

  - name: creating the IGW
    ec2_vpc_igw:
      vpc_id: "{{vpc_id}}"
      region: "us-east-1"
      state:  "present"
    register: my_vpc_igw

# Assigning igw id to a variable

  - set_fact: igw_id="{{my_vpc_igw.gateway_id}}"

# Creating the route table

  - name: creating our own route table
    ec2_vpc_route_table:
      vpc_id: "{{vpc_id}}"
      region: "us-east-1"
      tags:
        Name: "public"
      subnets:
        - "{{pub_sub_id}}"
      routes:
        - dest: "0.0.0.0/0"
          gateway_id: "{{igw_id}}"


# Creating our own security group

  - name: creating our own security group
    ec2_group:
      name: "Ansible securitygroup"
      description: "Ansible SG"
      vpc_id: "{{vpc_id}}"
      region: "us-east-1"
      rules:
```

```
- proto: "tcp"
  from_port: 0
  to_port:  65000
  cidr_ip: "0.0.0.0/0"
```