Ansible Assignment

1. **Create the file on the target machines or servers as mentioned in the inventory file and the master's group, save the below code with. yml extension and run the playbook.**

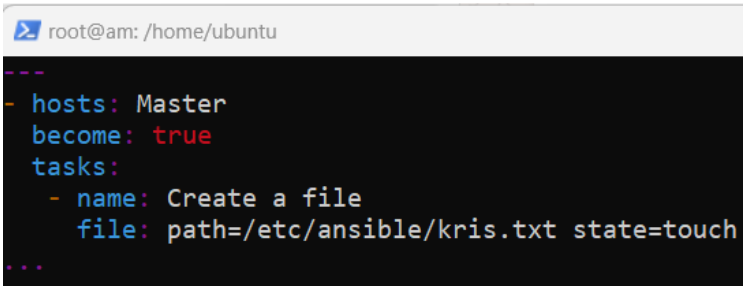
   We will create a playbook for the above task

   vi files.yml

```
root@am:/home/ubuntu# vi files.yml
```

   Contents of the Playbook

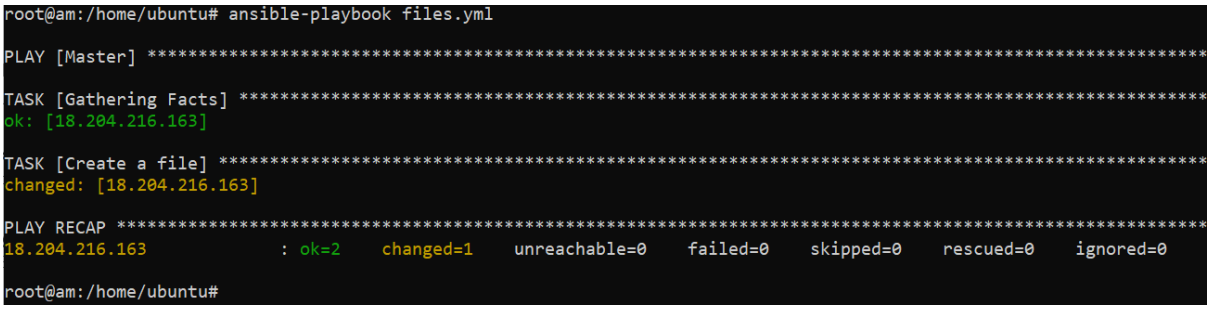
```
---
- hosts: Master
  become: true
  tasks:
   - name: Create a file
     file: path=/etc/ansible/kris.txt state=touch
...
```



   In our case, we have taken hosts as the master.

Now we will run the playbook

ansible-playbook files.yml

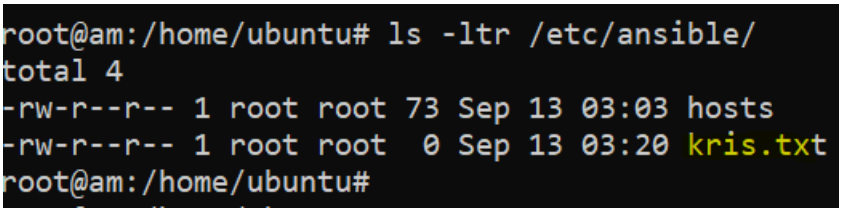


We see the playbook ran successfully.

Output:

Master

ls -ltr /etc/ansible/



2. **Create a directory with the mode as 775 and owner/group as Ansible.**

We will create a playbook for the above task

vi dir.yml

```
root@am:/home/ubuntu#
root@am:/home/ubuntu# vi dir.yml
```

Contents of the Playbook

- hosts: Master

  become: true

  tasks:

  - name: Create directory

    file: path=/etc/ansible/kris state=directory mode=775 owner=root group=root

...

```
---
- hosts: Master
  become: true
  tasks:
  - name: Create directory
    file: path=/etc/ansible/kris state=directory mode=775 owner=root group=root
...
```

In our case, we have taken hosts as the master.

Now we will run the playbook

ansible-playbook dir.yml

```
root@am:/home/ubuntu# ansible-playbook dir.yml

PLAY [Master] ******************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [18.204.216.163]

TASK [Create directory] ********************************************************************************************
changed: [18.204.216.163]

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

root@am:/home/ubuntu#
```

We see the playbook ran successfully.

Output:

Master

ls -ltr /etc/ansible/

```
root@am:/home/ubuntu# ls -ltr /etc/ansible/
total 8
-rw-r--r-- 1 root root   73 Sep 13 03:03 hosts
-rw-r--r-- 1 root root    0 Sep 13 03:20 kris.txt
drwxrwxr-x 2 root root 4096 Sep 13 03:29 kris
root@am:/home/ubuntu#
```
f

3. **Create a user. Let's look at the user module to create and delete users in the playbook.**

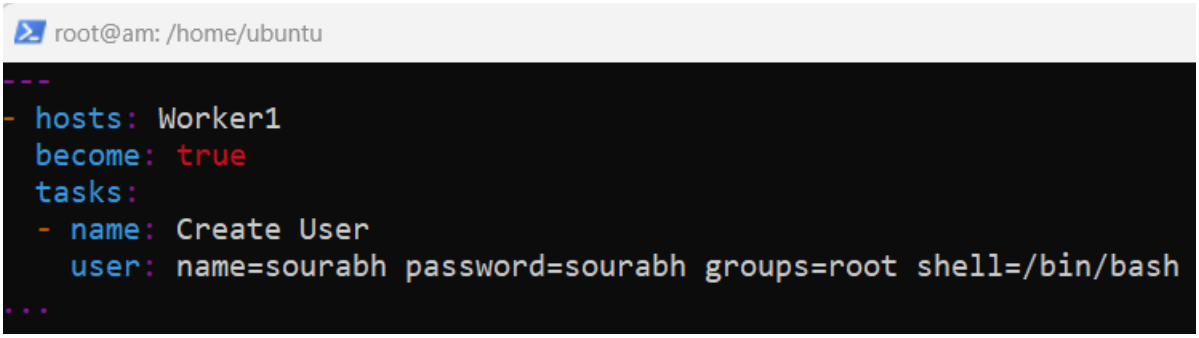
We will create a playbook for the above task

vi useradd.yml

```
root@am:/home/ubuntu# vi useradd.yml
root@am:/home/ubuntu#
```

Contents of the Playbook

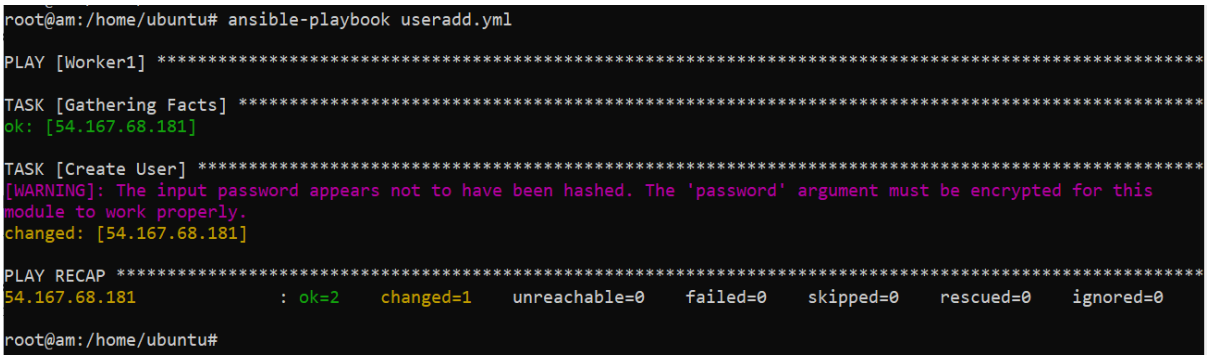
```
---
- hosts: Worker1
  become: true
  tasks:
  - name: Create User
    user: name=sourabh password=sourabh groups=root shell=/bin/bash
...
```



In our case, we have taken hosts as the Worker1.

Now we will run the playbook

ansible-playbook useradd.yml



We see the playbook ran successfully. It also gave the warning to keep password encrypted.

Output:

Worker1

cat /etc/passwd

```
root@aw1:/home/ubuntu# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:996:996:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
syslog:x:102:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:991:991:systemd Resolver:/:/usr/sbin/nologin
uuidd:x:103:103::/run/uuidd:/usr/sbin/nologin
tss:x:104:104:TPM software stack,,,:/var/lib/tpm:/bin/false
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
pollinate:x:106:1::/var/cache/pollinate:/bin/false
tcpdump:x:107:108::/nonexistent:/usr/sbin/nologin
landscape:x:108:109::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:990:990:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
polkitd:x:989:989:User for polkitd:/:/usr/sbin/nologin
ec2-instance-connect:x:109:65534::/nonexistent:/usr/sbin/nologin
root@aw1:/home/ubuntu#
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
sourabh:x:1001:1001::/home/sourabh:/bin/bash
root@aw1:/home/ubuntu#
```

4. **Remove user. Removing a user is very easy and it will need the state to be set to absent. This is equivalent to the userdel command in Linux.**

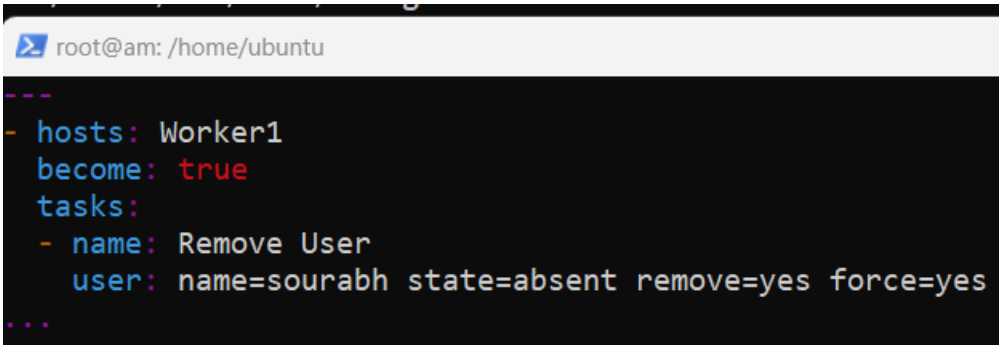
We will create a playbook for the above task

vi userdel.yml

```
root@am:/home/ubuntu#
root@am:/home/ubuntu# vi userdel.yml
```

Contents of the Playbook

```
---
- hosts: Worker1
  become: true
  tasks:
  - name: Remove User
    user: name=sourabh state=absent remove=yes force=yes
...
```



In our case, we have taken hosts as the Worker1.

Now we will run the playbook

ansible-playbook userdel.yml

```
root@am:/home/ubuntu# ansible-playbook userdel.yml

PLAY [Worker1] *****************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [54.167.68.181]

TASK [Remove User] *************************************************************************************************
changed: [54.167.68.181]

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

root@am:/home/ubuntu#
```

We see the playbook ran successfully.

Output:

Worker1

cat /etc/passwd

```
root@aw1:/home/ubuntu# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:996:996:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
syslog:x:102:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:991:991:systemd Resolver:/:/usr/sbin/nologin
uuidd:x:103:103::/run/uuidd:/usr/sbin/nologin
tss:x:104:104:TPM software stack,,,:/var/lib/tpm:/bin/false
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
pollinate:x:106:1::/var/cache/pollinate:/bin/false
tcpdump:x:107:108::/nonexistent:/usr/sbin/nologin
landscape:x:108:109::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:990:990:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
polkitd:x:989:989:User for polkitd:/:/usr/sbin/nologin
ec2-instance-connect:x:109:65534::/nonexistent:/usr/sbin/nologin
_chrony:x:110:112:Chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
root@aw1:/home/ubuntu#
```

5. **Copy content to a file using the copy module. If you need to copy a file to the target machines or servers use the src and dest in the copy module.**

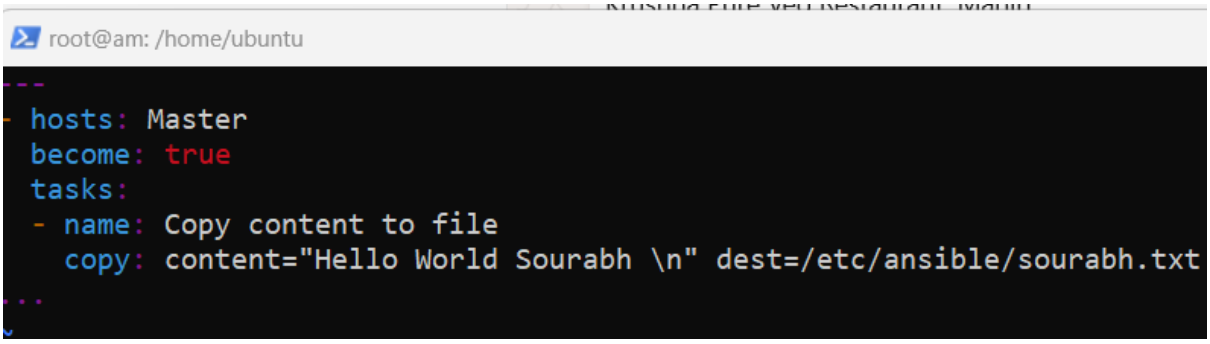
We will create a playbook for the above task

vi copy.yml

```
root@am:/home/ubuntu# vi copy.yml
```

Contents of the Playbook

```
---
- hosts: Master
  become: true
  tasks:
  - name: Copy content to file
    copy: content="Hello World Sourabh \n" dest=/etc/ansible/sourabh.txt
...
```



In our case, we have taken hosts as the master.

Now we will run the playbook

ansible-playbook copy.yml

```
root@am:/home/ubuntu# ansible-playbook copy.yml

PLAY [Master] ******************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [18.204.216.163]

TASK [Copy content to file] ****************************************************************************************
changed: [18.204.216.163]

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

root@am:/home/ubuntu#
```

We see the playbook ran successfully.

Output:

Master

ls -ltr /etc/ansible/

```
root@am:/home/ubuntu# ls -ltr /etc/ansible/
total 12
-rw-r--r-- 1 root root   73 Sep 13 03:03 hosts
-rw-r--r-- 1 root root    0 Sep 13 03:20 kris.txt
drwxrwxr-x 2 root root 4096 Sep 13 03:29 kris
-rw-r--r-- 1 root root   21 Sep 13 03:53 sourabh.txt
root@am:/home/ubuntu#
```

cat /etc/ansible/sourabh.txt

```
root@am:/home/ubuntu# cat /etc/ansible/sourabh.txt
Hello World Sourabh
root@am:/home/ubuntu#
```

6. **Replace all instances of a string. Using replace module we can replace a word with another word. The replace module will need 3 parameters i.e. 'path', 'regexp' (to find the particular word) and 'replace' (providing another word for replacement).**

We will create a playbook for the above task

vi replace.yml

```
root@am:/home/ubuntu# vi replace.yml
root@am:/home/ubuntu#
```

Contents of the Playbook

```
---
- hosts: Master
  tasks:
  - name: Replace example
    replace:
      path: /etc/ansible/sourabh.txt
      regexp: 'Hello'
      replace: ‘World’
...
```

```
---
- hosts: Master
  tasks:
  - name: Replace example
    replace:
      path: /etc/ansible/sourabh.txt
      regexp: 'Hello'
      replace: 'World'
```

In our case, we have taken hosts as the master.

Now we will run the playbook

ansible-playbook replace.yml

```
root@am:/home/ubuntu# ansible-playbook replace.yml

PLAY [Master] ******************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [18.204.216.163]

TASK [Replace example] *********************************************************************************************
changed: [18.204.216.163]

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

root@am:/home/ubuntu#
```

We see the playbook ran successfully.

Output:

Master

cat /etc/ansible/sourabh.txt

```
root@am:/home/ubuntu# cat /etc/ansible/sourabh.txt
World World Sourabh
```

7. **Archive or ZIP files and Folders. Using the Ansible archive module you can compress files or folders to 'zip', '.gz', or 'bz2' format.**

   We will create a playbook for the above task

   vi archive.yml

```
root@am:/home/ubuntu# vi archive.yml
root@am:/home/ubuntu#
```

   Contents of the Playbook

```
---
- hosts: Master
```

```
 become: true
 tasks:
 - name: Ansible zip file example
   archive:
     path: /etc/ansible/sourabh.txt
     dest: /etc/ansible/sourabh.zip
     format: zip
...
```

```yaml
---
- hosts: Master
  become: true
  tasks:
  - name: Ansible zip file example
    archive:
      path: /etc/ansible/sourabh.txt
      dest: /etc/ansible/sourabh.zip
      format: zip
...
```

In our case, we have taken hosts as the master.

Now we will run the playbook

```
ansible-playbook archive.yml
```

```
root@am:/home/ubuntu# ansible-playbook archive.yml

PLAY [Master] ******************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [18.204.216.163]

TASK [Ansible zip file example] ************************************************************************************
changed: [18.204.216.163]

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

root@am:/home/ubuntu#
```

We see the playbook ran successfully.

Output:

Master

ls -ltr /etc/ansible/

```
root@am:/home/ubuntu# ls -ltr /etc/ansible/
total 16
-rw-r--r-- 1 root root   73 Sep 13 03:03 hosts
-rw-r--r-- 1 root root    0 Sep 13 03:20 kris.txt
drwxrwxr-x 2 root root 4096 Sep 13 03:29 kris
-rw-r--r-- 1 root root   21 Sep 13 04:09 sourabh.txt
-rw-r--r-- 1 root root  139 Sep 13 04:16 sourabh.zip
root@am:/home/ubuntu#
```

8. **Playbook to zip multiple files to sourabh.zip file.**

In this we will create two more files at the location /etc/ansible/

touch /etc/ansible/sourabh1.txt

touch /etc/ansible/sourabh2.txt

ls -ltr /etc/ansible/

```
root@am:/home/ubuntu# ls -ltr /etc/ansible/
total 16
-rw-r--r-- 1 root root   73 Sep 13 03:03 hosts
-rw-r--r-- 1 root root    0 Sep 13 03:20 kris.txt
drwxrwxr-x 2 root root 4096 Sep 13 03:29 kris
-rw-r--r-- 1 root root   21 Sep 13 04:09 sourabh.txt
-rw-r--r-- 1 root root  139 Sep 13 04:16 sourabh.zip
-rw-r--r-- 1 root root    0 Sep 13 04:41 sourabh1.txt
-rw-r--r-- 1 root root    0 Sep 13 04:41 sourabh2.txt
root@am:/home/ubuntu#
```

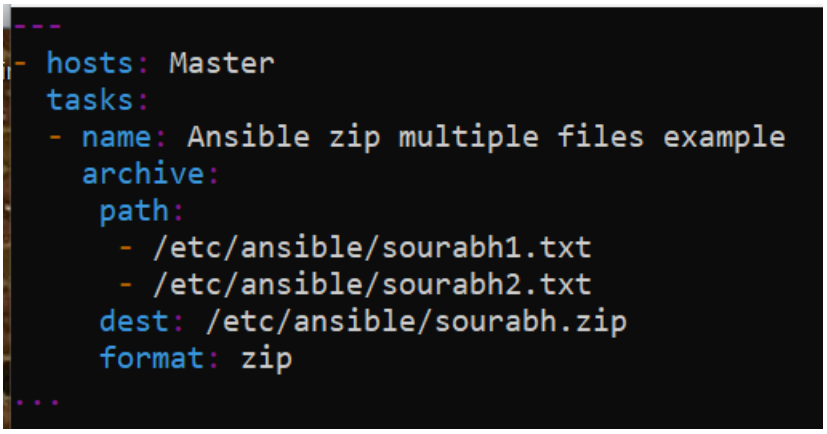
We will create a playbook for the above task

vi multiple.yml

```
root@am:/home/ubuntu# vi multiple.yml
root@am:/home/ubuntu#
```

Contents of the Playbook

```
---
- hosts: Master
  tasks:
  - name: Ansible zip multiple files example
    archive:
     path:
      - /etc/ansible/sourabh1.txt
      - /etc/ansible/sourabh2.txt
     dest: /etc/ansible/sourabh.zip
     format: zip
...
```



In our case, we have taken hosts as the master.

Now we will run the playbook

```
ansible-playbook multiple.yml
```

```
root@am:/home/ubuntu# ansible-playbook multiple.yml

PLAY [Master] ******************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [52.91.254.159]

TASK [Ansible zip multiple files example] **************************************************************************
changed: [52.91.254.159]

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

We see the playbook ran successfully.

Output:

Master

ls -ltr /etc/ansible/

```
root@am:/home/ubuntu# ls -ltr /etc/ansible/
total 16
-rw-r--r-- 1 root root    0 Sep 13 03:20 kris.txt
drwxrwxr-x 2 root root 4096 Sep 13 03:29 kris
-rw-r--r-- 1 root root   21 Sep 13 04:09 sourabh.txt
-rw-r--r-- 1 root root    0 Sep 13 04:41 sourabh1.txt
-rw-r--r-- 1 root root    0 Sep 13 04:41 sourabh2.txt
-rw-r--r-- 1 root root   74 Sep 13 05:42 hosts
-rw-r--r-- 1 root root  226 Sep 13 06:20 sourabh.zip
```

cat /etc/ansible/sourabh.zip

```
root@am:/home/ubuntu# cat /etc/ansible/sourabh.zip
PK  %-Y
        sourabh1.txt PK  "%-Y
                                sourabh2.txt PK    %-Y
                                                        ??sourabh1.txtPK    "%-Y
                                                                                ??,sourabh2.txtPK    tX
buntu#
```

9. **Playbook to zip all files in the /etc/ansible directory**

We will create a playbook for the above task

vi all.yml

```
root@am:/home/ubuntu# vi all.yml
root@am:/home/ubuntu#
```

Contents of the Playbook

```
---
- hosts: Master
  tasks:
  - name: Ansible zip directory example
    archive:
      path:
      - /etc/ansible
      dest: /etc/ansible/sourabh.zip
      format: zip
...
```

In our case, we have taken hosts as the master.

Now we will run the playbook

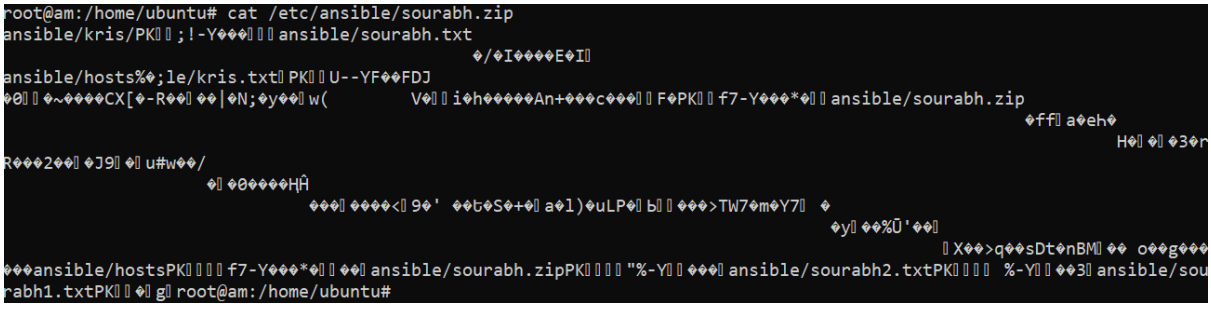
ansible-playbook all.yml

We see the playbook ran successfully.

Output:

Master

cat /etc/ansible/sourabh.zip



10. **Working with date. Using the system date and timestamp helps in certain status or logging purposes. The Ansible facts provide access to remote or target servers date and time. So, we can use the debug**

module to print the output along with the var attribute as shown below.

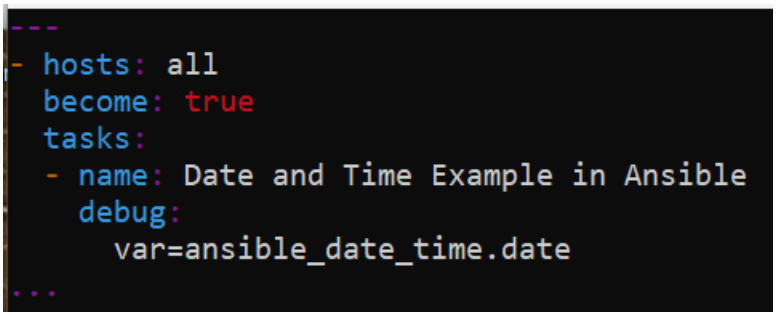
We will create a playbook for the above task

vi date.yml

```
root@am:/home/ubuntu# vi date.yml
root@am:/home/ubuntu#
```

Contents of the Playbook

```
---
- hosts: all
  become: true
  tasks:
  - name: Date and Time Example in Ansible
    debug:
      var=ansible_date_time.date
...
```



In our case, we have taken hosts as all.

Now we will run the playbook

ansible-playbook date.yml

```
root@am:/home/ubuntu# ansible-playbook date.yml

PLAY [all] ***********************************************************************************************************

TASK [Gathering Facts] ***********************************************************************************************
ok: [34.230.58.117]
ok: [44.211.193.203]
ok: [52.91.254.159]

TASK [Date and Time Example in Ansible] ******************************************************************************
ok: [44.211.193.203] => {
    "ansible_date_time.date": "2024-09-13"
}
ok: [34.230.58.117] => {
    "ansible_date_time.date": "2024-09-13"
}
ok: [52.91.254.159] => {
    "ansible_date_time.date": "2024-09-13"
}

PLAY RECAP ***********************************************************************************************************
34.230.58.117              : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
44.211.193.203             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
52.91.254.159              : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

We see the playbook ran successfully.

Output:

Date is displayed in the output of the playbook itself

11. **Working with time. Using the system date and timestamp helps in certain status or logging purposes. The Ansible facts provide access to remote or target servers date and time. So, we can use the debug module to print the output along with the var attribute as shown below.**

We will create a playbook for the above task

vi time.yml

```
root@am:/home/ubuntu# vi time.yml
root@am:/home/ubuntu#
```

Contents of the Playbook

```
---
- hosts: all
  become: true
```

tasks:
- name: Date and Time Example in Ansible
  debug:
    var=ansible_date_time.time
...

```
---
- hosts: all
  become: true
  tasks:
  - name: Date and Time Example in Ansible
    debug:
      var=ansible_date_time.time
...
```

In our case, we have taken hosts as all.

Now we will run the playbook

ansible-playbook time.yml

```
root@am:/home/ubuntu# ansible-playbook time.yml

PLAY [all] ********************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [34.230.58.117]
ok: [44.211.193.203]
ok: [52.91.254.159]

TASK [Date and Time Example in Ansible] ***************************************************************************
ok: [44.211.193.203] => {
    "ansible_date_time.time": "07:23:02"
}
ok: [34.230.58.117] => {
    "ansible_date_time.time": "07:23:03"
}
ok: [52.91.254.159] => {
    "ansible_date_time.time": "07:23:03"
}

PLAY RECAP ********************************************************************************************************
34.230.58.117              : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
44.211.193.203             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
52.91.254.159              : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

We see the playbook ran successfully.

Output:

Time is displayed in the output of the playbook itself

12. Variables Example. Variables are used to store values. In the below Example I am declaring the variable name with value Sourabh. The output will be Sourabh.

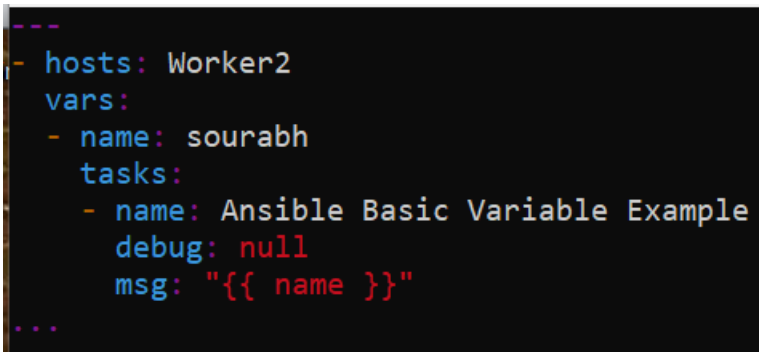
We will create a playbook for the above task

vi variable.yml

```
root@am:/home/ubuntu# vi varaible.yml
root@am:/home/ubuntu#
```

Contents of the Playbook

```
---
- hosts: Worker2
  vars:
  - name: sourabh
    tasks:
    - name: Ansible Basic Variable Example
      debug: null
      msg: "{{ name }}"
...
```



In our case, we have taken hosts as Worker2.

Now we will run the playbook

ansible-playbook variable.yml

```
root@am:/home/ubuntu# ansible-playbook varaible.yml
[DEPRECATION WARNING]: Specifying a list of dictionaries for vars is deprecated in favor of specifying a dictionary.
This feature will be removed in version 2.18. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
[WARNING]: Found variable using reserved name: tasks
[WARNING]: Found variable using reserved name: name

PLAY [Worker2] ***************************************************************************************************

TASK [Gathering Facts] *******************************************************************************************
ok: [34.230.58.117]

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

We see the playbook ran successfully.

Output:

It is displayed in the output of the playbook itself

13. An array or a list of variables

We will create a playbook for the above task

vi array.yml

```
root@am:/home/ubuntu# vi array.yml
```

Contents of the Playbook

```
---
- hosts: all
  vars:
  - username: ubuntu
  - home: /home/ubuntu
```

```
  tasks:
   - name: print variables
     debug:
     msg: "Username: {{ username }}, Home dir: {{ home }}"
...
```

```yaml
---
- hosts: all
  vars:
  - username: ubuntu
  - home: /home/ubuntu
    tasks:
      - name: print variables
        debug:
        msg: "Username: {{ username }}, Home dir: {{ home }}"
...
```

In our case, we have taken hosts as all.

Now we will run the playbook

```
ansible-playbook array.yml
```

```
root@am:/home/ubuntu# ansible-playbook array.yml
[DEPRECATION WARNING]: Specifying a list of dictionaries for vars is deprecated in favor of specifying a dictionary.
This feature will be removed in version 2.18. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
[WARNING]: Found variable using reserved name: tasks

PLAY [all] ********************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [34.230.58.117]
ok: [44.211.193.203]
ok: [52.91.254.159]

PLAY RECAP ********************************************************************************************************
34.230.58.117              : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
44.211.193.203             : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
52.91.254.159              : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

We see the playbook ran successfully.

Output:

It is displayed in the output of the playbook itself

14. **Playbook to install vim editor and GIT on the target servers or machines. In this playbook, we have made use of the apt module to install the latest version of the software packages.**

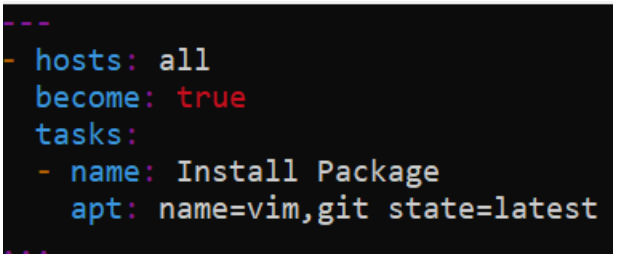
We will create a playbook for the above task

vi gitvim.yml

```
root@am:/home/ubuntu# vi gitvim.yml
root@am:/home/ubuntu#
```

Contents of the Playbook

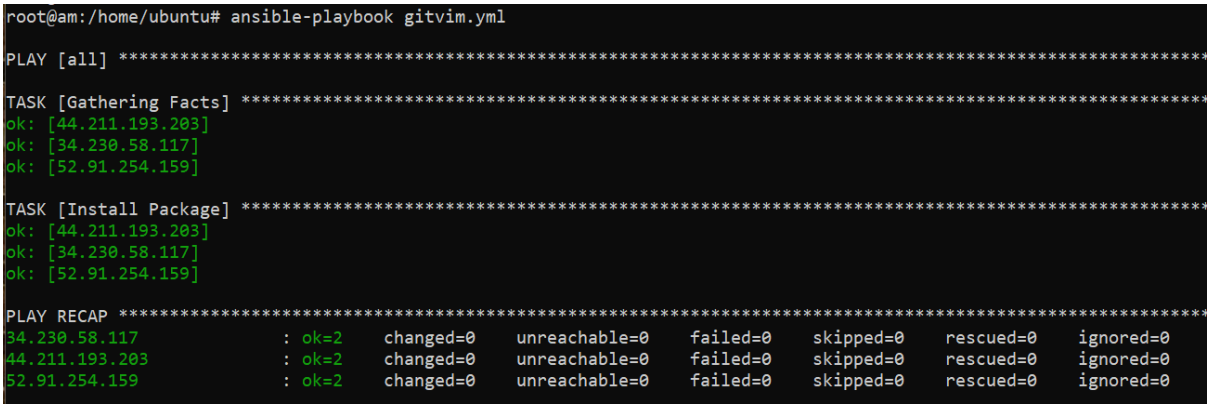
```
---
- hosts: all
  become: true
  tasks:
  - name: Install Package
    apt: name=vim,git state=latest
...
```



In our case, we have taken hosts as all.

Now we will run the playbook

ansible-playbook gitvim.yml



We see the playbook ran successfully without any change as package are already installed.

Output:

It is displayed in the output of the playbook itself as it ran without any change as package are already installed.

15. **Playbook to install mini-httpd on the target servers or machines. In this playbook, we have made use of the service module to install the latest version of the software packages.**

We will create a playbook for the above task

vi mini-httpd.yml

Contents of the Playbook

```
---
- hosts: Worker1
  become: true
  tasks:
  - name: Install Package
```

```
    package: name=mini-httpd state=present
  - name: Start mini-httpd service
    service: name=mini-httpd state=started
...
```

```
---
- hosts: Worker1
  become: true
  tasks:
  - name: Install Package
    package: name=mini-httpd state=present
  - name: Start mini-httpd service
    service: name=mini-httpd state=started
...
```

In our case, we have taken hosts as Worker1.

Now we will run the playbook

```
ansible-playbook mini-httpd.yml
```

```
root@am:/home/ubuntu# ansible-playbook mini-httpd.yml

PLAY [Worker1] ***************************************************************************************************************

TASK [Gathering Facts] *******************************************************************************************************
ok: [44.211.193.203]

TASK [Install Package] *******************************************************************************************************
changed: [44.211.193.203]

TASK [Start mini-httpd service] **********************************************************************************************
ok: [44.211.193.203]

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

We see the playbook ran successfully.

Output:

Worker1

```
root@aw1:/home/ubuntu# systemctl status mini-httpd
● mini-httpd.service - mini_httpd server
     Loaded: loaded (/usr/lib/systemd/system/mini-httpd.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-09-13 08:45:08 UTC; 1min 36s ago
    Process: 3141 ExecStart=/usr/sbin/mini_httpd $DAEMON_OPTS -i /run/mini_httpd.pid (code=exited, status=0/SUCCESS)
   Main PID: 3142 (mini_httpd)
      Tasks: 1 (limit: 1130)
     Memory: 776.0K (peak: 1.1M)
        CPU: 5ms
     CGroup: /system.slice/mini-httpd.service
             └─3142 /usr/sbin/mini_httpd -C /etc/mini-httpd.conf -i /run/mini_httpd.pid

Sep 13 08:45:08 aw1 systemd[1]: Starting mini-httpd.service - mini_httpd server...
Sep 13 08:45:08 aw1 systemd[1]: mini-httpd.service: Can't open PID file /run/mini_httpd.pid (yet?) after start: No su
Sep 13 08:45:08 aw1 systemd[1]: Started mini-httpd.service - mini_httpd server.
Sep 13 08:45:08 aw1 mini_httpd[3142]: /usr/sbin/mini_httpd: started as root without requesting chroot(), warning only
Sep 13 08:45:08 aw1 mini_httpd[3142]: started as root without requesting chroot(), warning only
Sep 13 08:45:08 aw1 mini_httpd[3142]: mini_httpd/1.30 26Oct2018 starting on localhost, port 80
```

16. **Install JDK. The following playbook will automate to install JDK 8 on all target machines or servers. JDK is a pre-requisite for most of the other software packages like Maven or Tomcat.**

    We will create a playbook for the above task

    vi jdk18.yml

```
root@am:/home/ubuntu# vi jdk18.yml
root@am:/home/ubuntu#
```

    Contents of the Playbook

```
- name: Java installation
  hosts: all
  become: true
  tasks:
    - name: Download the JDK binaries
      get_url:
        url: https://download.java.net/java/GA/jdk18.0.2.1/db379da656dc47308e138f21b33976fa/1/GPL/openjdk-18.0.2.1_linux-x64_bin.tar.gz
        dest: /opt/openjdk-18.0.2.1_linux-x64_bin.tar.gz
```

- name: Extract the downloaded file

command: tar xvf /opt/openjdk-18.0.2.1_linux-x64_bin.tar.gz -C /opt

- name: Update profile

copy: content="export JAVA_HOME=/opt/jdk-18.0.2.1 \n" dest=/etc/profile.d/java.sh

- name: Set the JAVA_HOME in /etc/profile.d/java.sh file

lineinfile:

path: /etc/profile.d/java.sh

state: present

line: 'export PATH=${JAVA_HOME}/bin:${PATH}'

- name: Reload /etc/profile file

shell:

cmd: source /etc/profile.d/java.sh

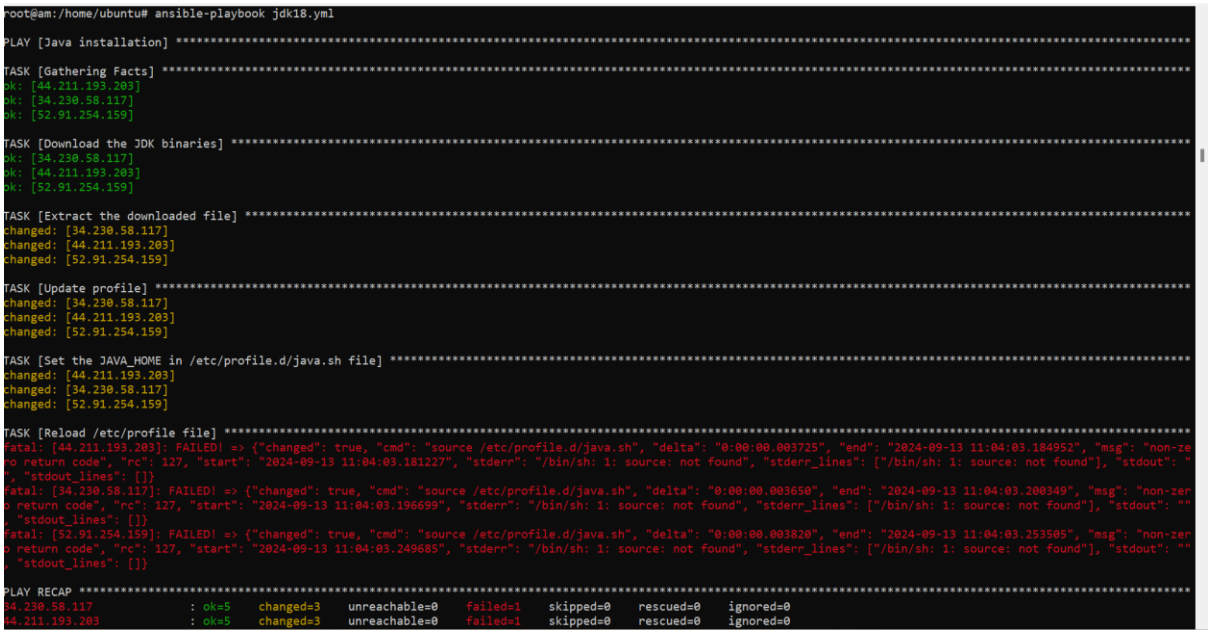
...

```yaml
- name: Java installation
  hosts: all
  become: true
  tasks:
    - name: Download the JDK binaries
      get_url:
        url: https://download.java.net/java/GA/jdk18.0.2.1/db379da656dc47308e138f21b33976fa/1/GPL/openjdk-18.0.2.1_linux-x64_bin.tar.gz
        dest: /opt/openjdk-18.0.2.1_linux-x64_bin.tar.gz

    - name: Extract the downloaded file
      command: tar xvf /opt/openjdk-18.0.2.1_linux-x64_bin.tar.gz -C /opt

    - name: Update profile
      copy: content="export JAVA_HOME=/opt/jdk-18.0.2.1 \n" dest=/etc/profile.d/java.sh

    - name: Set the JAVA_HOME in /etc/profile.d/java.sh file
      lineinfile:
        path: /etc/profile.d/java.sh
        state: present
        line: 'export PATH=${JAVA_HOME}/bin:${PATH}'

    - name: Reload /etc/profile file
      shell:
        cmd: source /etc/profile.d/java.sh
```

In our case, we have taken hosts as all.

Now we will run the playbook

ansible-playbook jdk18.yml

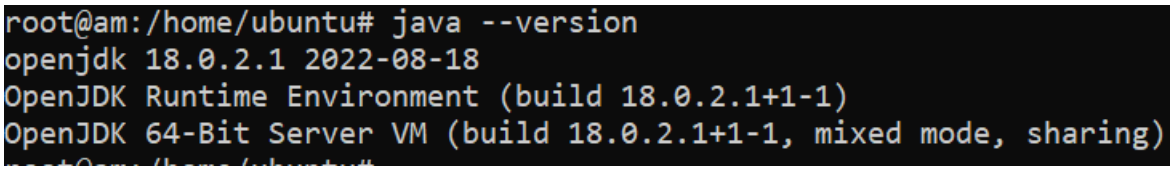


We see the playbook ran successfully.

Output:

Master

java --version



Worker1

java --version

```
root@aw1:/home/ubuntu# java --version
openjdk 18.0.2.1 2022-08-18
OpenJDK Runtime Environment (build 18.0.2.1+1-1)
OpenJDK 64-Bit Server VM (build 18.0.2.1+1-1, mixed mode, sharing)
root@aw1:/home/ubuntu#
```

Worker2

java --version

```
root@aw2:/home/ubuntu# java --version
openjdk 18.0.2.1 2022-08-18
OpenJDK Runtime Environment (build 18.0.2.1+1-1)
OpenJDK 64-Bit Server VM (build 18.0.2.1+1-1, mixed mode, sharing)
root@aw2:/home/ubuntu#
```

17. **Install Maven. The tasks performed are to download the maven file from the URL using the get_url module, extract the file downloaded, move it to a smaller directory, update and run the profile where the maven is added to the path.**

We will create a playbook for the above task

vi mvn.yml

```
root@am:/home/ubuntu#
root@am:/home/ubuntu# vi mvn.yml
```

Contents of the Playbook

```
- name: Maven installation
  hosts: Worker2
  become: true
  tasks:
    - name: Install OpenJDK 17
      apt:
        name: openjdk-17-jdk
        state: present

    - name: Download the Maven binaries
```

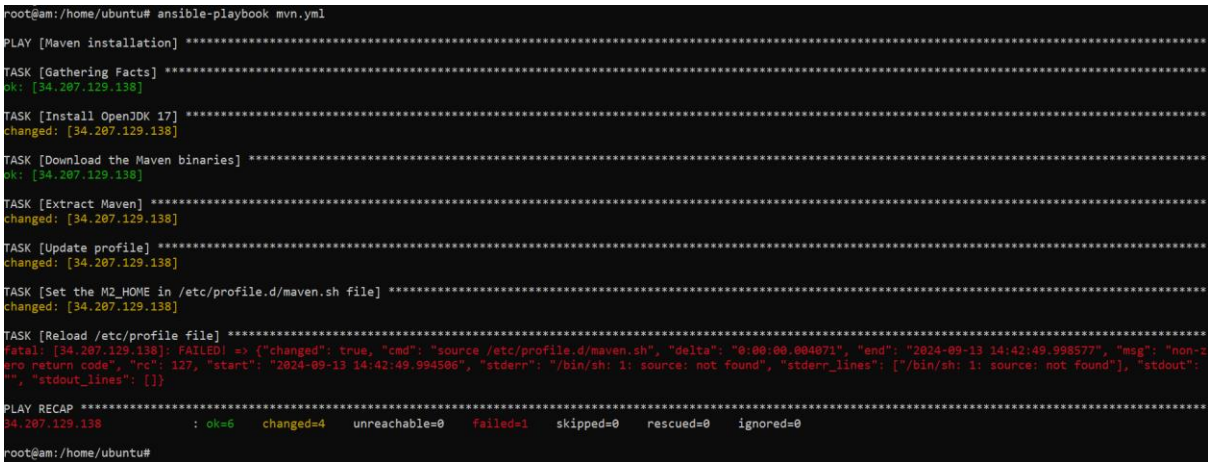
```
    get_url:
      url: https://dlcdn.apache.org/maven/maven-
3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz
      dest: /opt/apache-maven-3.9.9-bin.tar.gz

  - name: Extract Maven
    command: tar xvf /opt/apache-maven-3.9.9-bin.tar.gz -C /opt

  - name: Update profile
    copy: content="export M2_HOME=/opt/apache-maven-3.9.9 \n"
dest=/etc/profile.d/maven.sh

  - name: Set the M2_HOME in /etc/profile.d/maven.sh file
    lineinfile:
      path: /etc/profile.d/maven.sh
      state: present
      line: 'export PATH=${M2_HOME}/bin:${PATH}'

  - name: Reload /etc/profile file
    shell:
      cmd: source /etc/profile.d/maven.sh
...
```

```yaml
- name: Maven installation
  hosts: Worker2
  become: true
  tasks:
    - name: Install OpenJDK 17
      apt:
        name: openjdk-17-jdk
        state: present

    - name: Download the Maven binaries
      get_url:
        url: https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz
        dest: /opt/apache-maven-3.9.9-bin.tar.gz

    - name: Extract Maven
      command: tar xvf /opt/apache-maven-3.9.9-bin.tar.gz -C /opt

    - name: Update profile
      copy: content="export M2_HOME=/opt/apache-maven-3.9.9 \n" dest=/etc/profile.d/maven.sh

    - name: Set the M2_HOME in /etc/profile.d/maven.sh file
      lineinfile:
        path: /etc/profile.d/maven.sh
        state: present
        line: 'export PATH=${M2_HOME}/bin:${PATH}'

    - name: Reload /etc/profile file
      shell:
        cmd: source /etc/profile.d/maven.sh
```

In our case, we have taken hosts as Worker2.

Now we will run the playbook

ansible-playbook mvn.yml



We see the playbook ran successfully.

Output:

Worker2

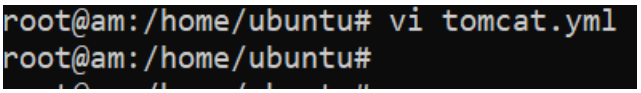
mvn -v

```
root@aw2:/home/ubuntu# mvn -v
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcdc97d260186937)
Maven home: /opt/apache-maven-3.9.9
Java version: 17.0.12, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1012-aws", arch: "amd64", family: "unix"
root@aw2:/home/ubuntu#
```

18. **Install Tomcat. The below playbook helps to install and start Tomcat 8 on to the target machines or servers.**

We will create a playbook for the above task

vi tomcat.yml



Contents of the Playbook

```
---
- hosts: Worker1
  become: true
  tasks:
    - name: Install JDK and Maven
      apt:
        name:
          - openjdk-17-jdk
          - maven
        state: present

    - name: Download Tomcat 9 and unzip the folder
      shell: |
        cd /etc
        wget https://dlcdn.apache.org/tomcat/tomcat-
9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz
        tar xzfv apache-tomcat-9.0.93.tar.gz
        mv apache-tomcat-9.0.93 tomcat
        rm apache-tomcat-9.0.93.tar.gz

    - name: Setup tomcat service
      file: path=/etc/systemd/system/tomcat.service  state=touch

    - name: Add content in service file
      blockinfile:
        dest: /etc/systemd/system/tomcat.service
        block: |
```

```
[Unit]
Description=Tomcat 9 Servlet Container
After=network.target

[Service]
Type=forking

Environment=\"JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64"
Environment=\"JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"
Environment=\"CATALINA_BASE=/etc/tomcat/"
Environment=\"CATALINA_HOME=/etc/tomcat/"
Environment=\"CATALINA_PID=/etc/tomcat/temp/tomcat.pid"
Environment=\"CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"
ExecStart=/etc/tomcat/bin/startup.sh
ExecStop=/etc/tomcat/bin/shutdown.sh

  - name: Restart Services
    shell: |
      systemctl daemon-reload
      systemctl start tomcat
...
```

```yaml
---
- hosts: Worker1
  become: true
  tasks:
    - name: Install JDK and Maven
      apt:
        name:
          - openjdk-17-jdk
          - maven
        state: present

    - name: Download Tomcat 9 and unzip the folder
      shell: |
        cd /etc
        wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz
        tar xzfv apache-tomcat-9.0.93.tar.gz
        mv apache-tomcat-9.0.93 tomcat
        rm apache-tomcat-9.0.93.tar.gz

    - name: Setup tomcat service
      file: path=/etc/systemd/system/tomcat.service  state=touch

    - name: Add content in service file
      blockinfile:
        dest: /etc/systemd/system/tomcat.service
        block: |
          [Unit]
          Description=Tomcat 9 Servlet Container
          After=network.target

          [Service]
          Type=forking

          Environment=\"JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64"
          Environment=\"JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"
          Environment=\"CATALINA_BASE=/etc/tomcat/"
          Environment=\"CATALINA_HOME=/etc/tomcat/"
          Environment=\"CATALINA_PID=/etc/tomcat/temp/tomcat.pid"
          Environment=\"CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"
          ExecStart=/etc/tomcat/bin/startup.sh
          ExecStop=/etc/tomcat/bin/shutdown.sh
```

```
    - name: Restart Services
      shell: |
        systemctl daemon-reload
        systemctl start tomcat
...
```

In our case, we have taken hosts as Worker 1.

Now we will run the playbook

ansible-playbook tomcat.yml

```
root@am:/home/ubuntu# ansible-playbook tomcat.yml

PLAY [Worker1] ***************************************************************************************************************************************************

TASK [Gathering Facts] *******************************************************************************************************************************************
ok: [3.84.15.108]

TASK [Install JDK and Maven] *************************************************************************************************************************************
changed: [3.84.15.108]

TASK [Download Tomcat 9 and unzip the folder] ********************************************************************************************************************
changed: [3.84.15.108]

TASK [Setup tomcat service] **************************************************************************************************************************************
changed: [3.84.15.108]

TASK [Add content in service file] *******************************************************************************************************************************
changed: [3.84.15.108]

TASK [Restart Services] ******************************************************************************************************************************************
changed: [3.84.15.108]

PLAY RECAP *******************************************************************************************************************************************************
3.84.15.108                : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

We see the playbook ran successfully.

Output:

Worker1

systemctl status tomcat

```
root@aw1:/home/ubuntu# systemctl status tomcat
● tomcat.service - Tomcat 9 Servlet Container
     Loaded: loaded (/etc/systemd/system/tomcat.service; static)
     Active: active (running) since Fri 2024-09-13 14:20:12 UTC; 5min ago
    Process: 17700 ExecStart=/etc/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)
   Main PID: 17711 (java)
      Tasks: 30 (limit: 1130)
     Memory: 77.7M (peak: 78.8M)
        CPU: 4.356s
     CGroup: /system.slice/tomcat.service
             └─17711 /usr/bin/java -Djava.util.logging.config.file=/etc/tomcat/conf/logging.properties -Djava.util.lo

Sep 13 14:20:12 aw1 systemd[1]: Starting tomcat.service - Tomcat 9 Servlet Container...
Sep 13 14:20:12 aw1 startup.sh[17700]: Tomcat started.
Sep 13 14:20:12 aw1 systemd[1]: Started tomcat.service - Tomcat 9 Servlet Container.
lines 1-14/14 (END)
```

19. Install Jenkins through ansible.

We will create a playbook for the above task

vi jenkins.yml

```
root@am:/home/ubuntu# vi jenkins.yml
root@am:/home/ubuntu#
```

Contents of the Playbook

```
---
- name: Install Jenkins and Java 17
  hosts: Master
  become: true
  tasks:
    - name: Update and upgrade apt packages
      apt:
        update_cache: yes
        upgrade: dist

    - name: Install OpenJDK 17
      apt:
       name: openjdk-17-jdk
       state: present

    - name: Check Java version
      command: java --version
      register: java_version

    - name: Print Java version
      debug:
        msg: "{{ java_version.stdout }}"

    - name: Add Jenkins repository key
      get_url:
        url: https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
        dest: /usr/share/keyrings/jenkins-keyring.asc
```

```
    - name: Add Jenkins repository
      apt_repository:
        repo: deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]
https://pkg.jenkins.io/debian-stable binary/
        state: present

    - name: Update apt cache again after adding Jenkins repository
      apt:
        update_cache: yes

    - name: Install Jenkins
      apt:
        name: jenkins
        state: present

- name: Start Jenkins service
      systemd:
        name: jenkins
        state: started
        enabled: yes

    - name: Ensure Jenkins on port 8080
      wait_for:
        port: 8080
        delay: 10
        state: started
...
```

```yaml
---
- name: Install Jenkins and Java 17
  hosts: Master
  become: true
  tasks:
    - name: Update and upgrade apt packages
      apt:
        update_cache: yes
        upgrade: dist

    - name: Install OpenJDK 17
      apt:
       name: openjdk-17-jdk
       state: present

    - name: Check Java version
      command: java --version
      register: java_version

    - name: Print Java version
      debug:
        msg: "{{ java_version.stdout }}"

    - name: Add Jenkins repository key
      get_url:
        url: https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
        dest: /usr/share/keyrings/jenkins-keyring.asc

    - name: Add Jenkins repository
      apt_repository:
        repo: deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/
        state: present

    - name: Update apt cache again after adding Jenkins repository
      apt:
        update_cache: yes

    - name: Install Jenkins
      apt:
        name: jenkins
        state: present
```

```yaml
    - name: Start Jenkins service
      systemd:
        name: jenkins
        state: started
        enabled: yes

    - name: Ensure Jenkins on port 8080
      wait_for:
        port: 8080
        delay: 10
        state: started
...
```

In our case, we have taken hosts as Master.

Now we will run the playbook

ansible-playbook jenkins.yml

```
root@am:/home/ubuntu# ansible-playbook jenkins.yml

PLAY [Install Jenkins and Java 17] ********************************************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************************************************************
ok: [54.197.70.185]

TASK [Update and upgrade apt packages] ****************************************************************************************************************************************
ok: [54.197.70.185]

TASK [Install OpenJDK 17] *****************************************************************************************************************************************************
ok: [54.197.70.185]

TASK [Check Java version] ******************************************************************************************************************************************************
changed: [54.197.70.185]

TASK [Print Java version] ******************************************************************************************************************************************************
ok: [54.197.70.185] => {
    "msg": "openjdk 17.0.12 2024-07-16\nOpenJDK Runtime Environment (build 17.0.12+7-Ubuntu-1ubuntu224.04)\nOpenJDK 64-Bit Server VM (build 17.0.12+7-Ubuntu-1ubuntu224.04,
mixed mode, sharing)"
}

TASK [Add Jenkins repository key] **********************************************************************************************************************************************
changed: [54.197.70.185]

TASK [Add Jenkins repository] **************************************************************************************************************************************************
changed: [54.197.70.185]

TASK [Update apt cache again after adding Jenkins repository] ******************************************************************************************************************
changed: [54.197.70.185]

TASK [Install Jenkins] *********************************************************************************************************************************************************
changed: [54.197.70.185]

TASK [Start Jenkins service] ***************************************************************************************************************************************************
ok: [54.197.70.185]

TASK [Ensure Jenkins on port 8080] *********************************************************************************************************************************************
ok: [54.197.70.185]

PLAY RECAP *********************************************************************************************************************************************************************
54.197.70.185              : ok=11   changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

We see the playbook ran successfully.

Output:

Master

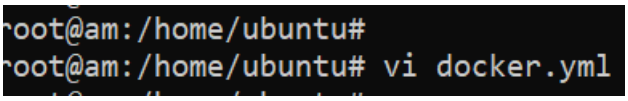
systemctl status jenkins

```
root@am:/home/ubuntu# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-09-13 13:46:36 UTC; 1min 55s ago
   Main PID: 19998 (java)
      Tasks: 40 (limit: 1130)
     Memory: 299.9M (peak: 335.5M)
        CPU: 11.870s
     CGroup: /system.slice/jenkins.service
             └─19998 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Sep 13 13:46:32 am jenkins[19998]: 26e6c7ab08464178a5ff7e21a3f53f75
Sep 13 13:46:32 am jenkins[19998]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Sep 13 13:46:32 am jenkins[19998]: *************************************************************
Sep 13 13:46:32 am jenkins[19998]: *************************************************************
Sep 13 13:46:32 am jenkins[19998]: *************************************************************
Sep 13 13:46:36 am jenkins[19998]: 2024-09-13 13:46:36.723+0000 [id=31]       INFO       jenkins.InitReactorRunner$1#onAttained: Completed initialization
Sep 13 13:46:36 am jenkins[19998]: 2024-09-13 13:46:36.753+0000 [id=24]       INFO       hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
Sep 13 13:46:36 am systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Sep 13 13:46:36 am jenkins[19998]: 2024-09-13 13:46:36.826+0000 [id=47]       INFO       h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson>
Sep 13 13:46:36 am jenkins[19998]: 2024-09-13 13:46:36.826+0000 [id=47]       INFO       hudson.util.Retrier#start: Performed the action check updates server successfull>
lines 1-20/20 (END)
```

20. Install Docker through ansible.

We will create a playbook for the above task

vi docker.yml



Contents of the Playbook

```
---
- hosts: Worker1
  become: true
  tasks:
    - name: Install aptitude
      apt:
        name: aptitude
        state: latest
        update_cache: true

    - name: Install required system packages
      apt:
        pkg:
          - apt-transport-https
          - ca-certificates
          - curl
          - software-properties-common
          - python3-pip
          - virtualenv
          - python3-setuptools
        state: latest
        update_cache: true

    - name: Add Docker GPG apt Key
      apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
        state: present

    - name: Add Docker Repository
      apt_repository:
        repo: deb https://download.docker.com/linux/ubuntu focal stable
```

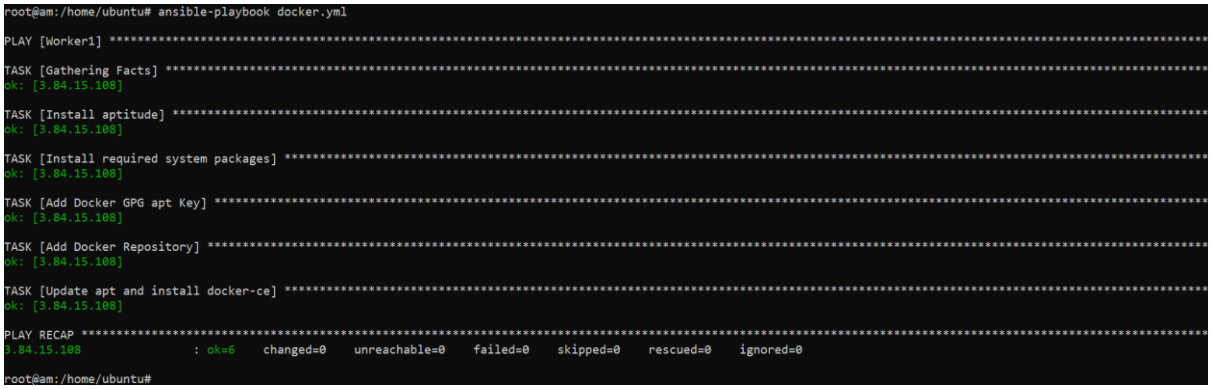
```
    state: present

  - name: Update apt and install docker-ce
    apt:
      name: docker-ce
      state: latest
      update_cache: true
...
```

```yaml
---
- hosts: Worker1
  become: true
  tasks:
    - name: Install aptitude
      apt:
        name: aptitude
        state: latest
        update_cache: true

    - name: Install required system packages
      apt:
        pkg:
          - apt-transport-https
          - ca-certificates
          - curl
          - software-properties-common
          - python3-pip
          - virtualenv
          - python3-setuptools
        state: latest
        update_cache: true

    - name: Add Docker GPG apt Key
      apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
        state: present

    - name: Add Docker Repository
      apt_repository:
        repo: deb https://download.docker.com/linux/ubuntu focal stable
        state: present

    - name: Update apt and install docker-ce
      apt:
        name: docker-ce
        state: latest
        update_cache: true
...
```

In our case, we have taken hosts as Worker1.

Now we will run the playbook

ansible-playbook docker.yml

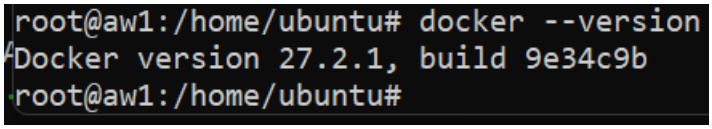


We see the playbook ran successfully.

Output:

Worker1

docker --version



21. Install MySQL through ansible.

For the above task we need to create a directory. In our case, it will “mysql”. This is created to keep our credentials for my sql in it under

subdirectory called group_vars and there will be a file named as “all” which will have our credentials for mysql server.

mkdir mysql

```
root@am:/home/ubuntu# mkdir mysql
root@am:/home/ubuntu#
```

mkdir mysql/group_vars (directory which has the credentials)

ls -ltr

```
root@am:/home/ubuntu/mysql# ls -ltr
total 8
drwxr-xr-x 2 root root 4096 Sep 13 17:08 group_vars
```

vi mysql/group_vars/all

```
root@am:/home/ubuntu/mysql/group_vars# vi all
root@am:/home/ubuntu/mysql/group_vars#
```

Our credentials

```
db_user: my_post
db_pass: my123
db_name: my_db
```

We will create a playbook for the above task in the parent directory

ls -ltr

```
root@am:/home/ubuntu/mysql# ls -ltr
total 8
drwxr-xr-x 2 root root 4096 Sep 13 17:08 group_vars
-rw-r--r-- 1 root root 1145 Sep 13 17:10 mysql.yml
```

vi mysql.yml

```
root@am:/home/ubuntu/mysql# vi mysql.yml
root@am:/home/ubuntu/mysql#
```

Contents of the Playbook

```
- name: setup Mysql with my_db db and remote login
  become: yes
  hosts: Worker3
  tasks:
    - name: Installing Mysql and dependencies
      package:
        name: "{{item}}"
        state: present
        update_cache: yes
      loop:
       - mysql-server
       - mysql-client
       - python3-mysqldb
       - libmysqlclient-dev
      become: yes

    - name: start and enable mysql service
      service:
        name: mysql
        state: started
        enabled: yes

    - name: creating mysql user (my_post)
      mysql_user:
        name: "{{db_user}}"
        password: "{{db_pass}}"
        priv: '*.*:ALL'
        host: '%'
        state: present

    - name: creating medium_db
      mysql_db:
        name: "{{db_name}}"
        state: present
```

```
  - name: Enable remote login to mysql
    lineinfile:
      path: /etc/mysql/mysql.conf.d/mysqld.cnf
      regexp: '^bind-address'
      line: 'bind-address = 0.0.0.0'
      backup: yes
    notify:
      - Restart mysql

 handlers:
  - name: Restart mysql
    service:
      name: mysql
      state: restarted
```

...

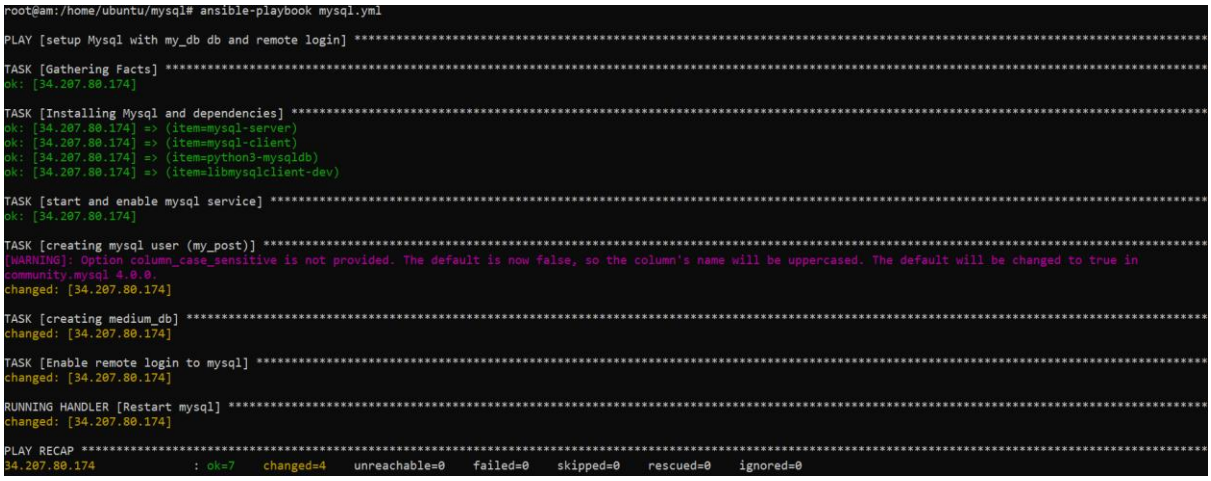
```yaml
- name: setup Mysql with my_db db and remote login
  become: yes
  hosts: Worker3
  tasks:
    - name: Installing Mysql and dependencies
      package:
         name: "{{item}}"
         state: present
         update_cache: yes
      loop:
        - mysql-server
        - mysql-client
        - python3-mysqldb
        - libmysqlclient-dev
      become: yes

    - name: start and enable mysql service
      service:
         name: mysql
         state: started
         enabled: yes

    - name: creating mysql user (my_post)
      mysql_user:
         name: "{{db_user}}"
         password: "{{db_pass}}"
         priv: '*.*:ALL'
         host: '%'
         state: present

    - name: creating medium_db
      mysql_db:
         name: "{{db_name}}"
         state: present

    - name: Enable remote login to mysql
      lineinfile:
         path: /etc/mysql/mysql.conf.d/mysqld.cnf
         regexp: '^bind-address'
         line: 'bind-address = 0.0.0.0'
         backup: yes
```

```
    notify:
      - Restart mysql

handlers:
  - name: Restart mysql
    service:
      name: mysql
      state: restarted
```

In our case, we have taken hosts as Worker3.

Now we will run the playbook

ansible-playbook mysql.yml



We see the playbook ran successfully.

Output:

Worker3

We need run the below command on worker3 as it has mysql client installed on it to connect to our database using the public IP address our EC2 instance.

mysql -u my_post -p -h 34.207.80.174