ANSIBLE

INSTALLATION OF ANSIBLE :

- sudo apt-add-repository ppa:ansible/ansible
- sudo apt update
- sudo apt install ansible

MANAGING INFRASTRUCTURE :

- inventory file or Host file at `/etc/ansible/hosts`
- Creating a group with the syntax : “[servers]” and assign the public ip’s of the host with the variable “ansible_host”

Command : sudo vim /etc/ansible/hosts

Edit the vim file and add the the below info and save it

```
server_1 ansible_host=18.222.230.164
server_2 ansible_host=3.19.56.117
server_3 ansible_host=3.17.207.226
```

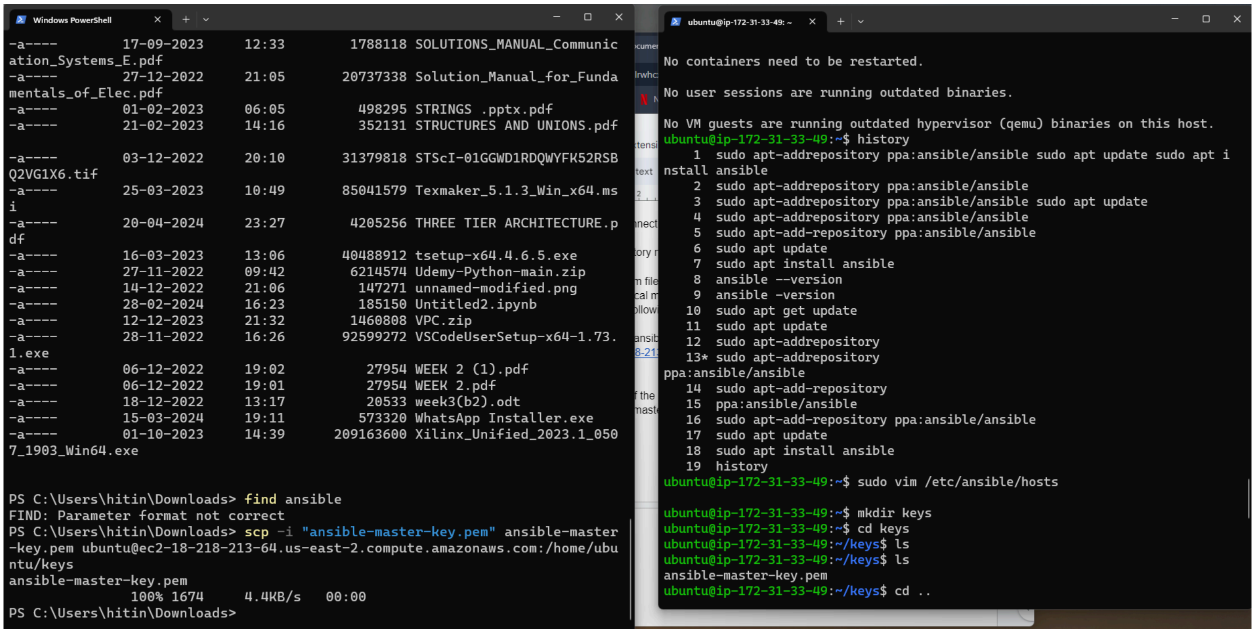
- Now we need a connection with the master and it’s nodes(3 nodes)

  Let’s create a directory name keys in our master instance.

  Navigate to the .pem file on your local machine and type the command to send the file from your local machine to a master machine which is in AWS (EC-2 instance) with the following command

  Command : scp -i "ansible-master-key.pem" ansible-master-key.pem ubuntu@ec2-18-218-213-64.us-east-2.compute.amazonaws.com:/home/ubuntu/keys

  And with the help of the above command we should see the private key(.pem file) on the ansible master instance.



```
-a----         17-09-2023     12:33        1788118 SOLUTIONS_MANUAL_Communic
ation_Systems_E.pdf
-a----         27-12-2022     21:05       20737338 Solution_Manual_for_Funda
mentals_of_Elec.pdf
-a----         01-02-2023     06:05         498295 STRINGS .pptx.pdf
-a----         21-02-2023     14:16         352131 STRUCTURES AND UNIONS.pdf

-a----         03-12-2022     20:10       31379818 STScI-01GGWD1RDQWYFK52RSB
Q2VG1X6.tif
-a----         25-03-2023     10:49       85041579 Texmaker_5.1.3_Win_x64.ms
i
-a----         20-04-2024     23:27        4205256 THREE TIER ARCHITECTURE.p
df
-a----         16-03-2023     13:06       40488912 tsetup-x64.4.6.5.exe
-a----         27-11-2022     09:42        6214574 Udemy-Python-main.zip
-a----         14-12-2022     21:06         147271 unnamed-modified.png
-a----         28-02-2024     16:23         185150 Untitled2.ipynb
-a----         12-12-2023     21:32        1460808 VPC.zip
-a----         28-11-2022     16:26       92599272 VSCodeUserSetup-x64-1.73.
1.exe
-a----         06-12-2022     19:02          27954 WEEK 2 (1).pdf
-a----         06-12-2022     19:01          27954 WEEK 2.pdf
-a----         18-12-2022     13:17          20533 week3(b2).odt
-a----         15-03-2024     19:11         573320 WhatsApp Installer.exe
-a----         01-10-2023     14:39      209163600 Xilinx_Unified_2023.1_050
7_1903_Win64.exe


PS C:\Users\hitin\Downloads> find ansible
FIND: Parameter format not correct
PS C:\Users\hitin\Downloads> scp -i "ansible-master-key.pem" ansible-master
-key.pem ubuntu@ec2-18-218-213-64.us-east-2.compute.amazonaws.com:/home/ubu
ntu/keys
ansible-master-key.pem
               100% 1674     4.4KB/s   00:00
PS C:\Users\hitin\Downloads>
```

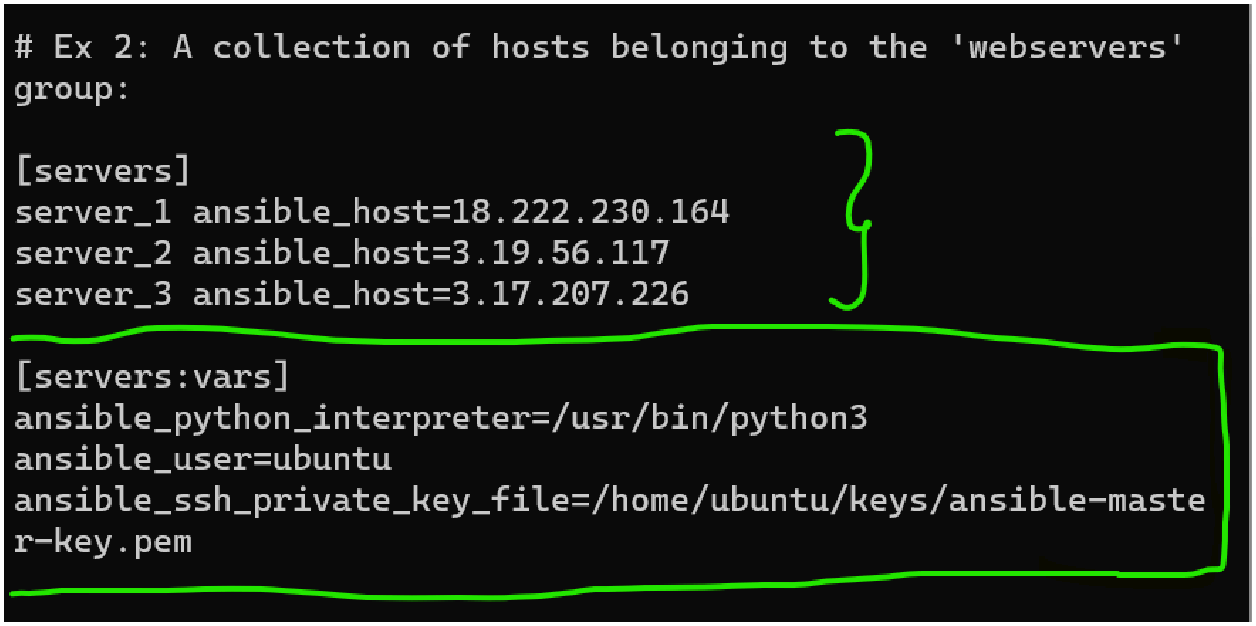
```
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-33-49:~$ history
    1  sudo apt-addrepository ppa:ansible/ansible sudo apt update sudo apt i
nstall ansible
    2  sudo apt-addrepository ppa:ansible/ansible
    3  sudo apt-addrepository ppa:ansible/ansible sudo apt update
    4  sudo apt-addrepository ppa:ansible/ansible
    5  sudo apt-add-repository ppa:ansible/ansible
    6  sudo apt update
    7  sudo apt install ansible
    8  ansible --version
    9  ansible -version
   10  sudo apt get update
   11  sudo apt update
   12  sudo apt-addrepository
   13* sudo apt-addrepository
ppa:ansible/ansible
   14  sudo apt-add-repository
   15  ppa:ansible/ansible
   16  sudo apt-add-repository ppa:ansible/ansible
   17  sudo apt update
   18  sudo apt install ansible
   19  history
ubuntu@ip-172-31-33-49:~$ sudo vim /etc/ansible/hosts

ubuntu@ip-172-31-33-49:~$ mkdir keys
ubuntu@ip-172-31-33-49:~$ cd keys
ubuntu@ip-172-31-33-49:~/keys$ ls
ubuntu@ip-172-31-33-49:~/keys$ ls
ansible-master-key.pem
ubuntu@ip-172-31-33-49:~/keys$ cd ..
```

PERMISSION OF PRIVATE-KEY USAGE TO HOSTS:

- Edit the host file at etc/ansible/hosts with the command

  ansible_ssh_private_key_file=/home/ubuntu/keys/ansible-master-key.pem

  And the python version used is “python3” and the ansible_user is ubuntu

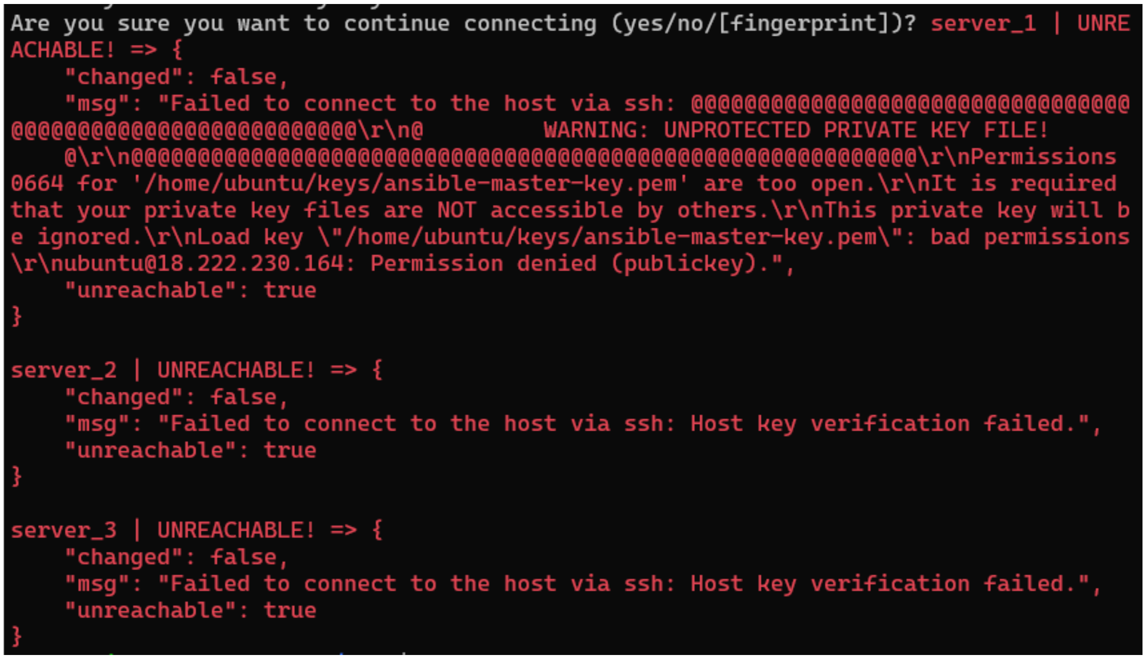


```
# Ex 2: A collection of hosts belonging to the 'webservers'
group:

[servers]
server_1 ansible_host=18.222.230.164
server_2 ansible_host=3.19.56.117
server_3 ansible_host=3.17.207.226

[servers:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/ubuntu/keys/ansible-maste
r-key.pem
```

- Let try pinging the servers/instances from ansible-master

Command : ansible servers -m ping

Output :



```
Are you sure you want to continue connecting (yes/no/[fingerprint])? server_1 | UNRE
ACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@\r\n@         WARNING: UNPROTECTED PRIVATE KEY FILE!
    @\r\n@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\r\nPermissions
0664 for '/home/ubuntu/keys/ansible-master-key.pem' are too open.\r\nIt is required
that your private key files are NOT accessible by others.\r\nThis private key will b
e ignored.\r\nLoad key \"/home/ubuntu/keys/ansible-master-key.pem\": bad permissions
\r\nubuntu@18.222.230.164: Permission denied (publickey).",
    "unreachable": true
}

server_2 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Host key verification failed.",
    "unreachable": true
}

server_3 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Host key verification failed.",
    "unreachable": true
}
```

Server-1 error : Incorrect file permission

SOL : chmod 600 /home/ubuntu/keys/ansible-master-key.pem

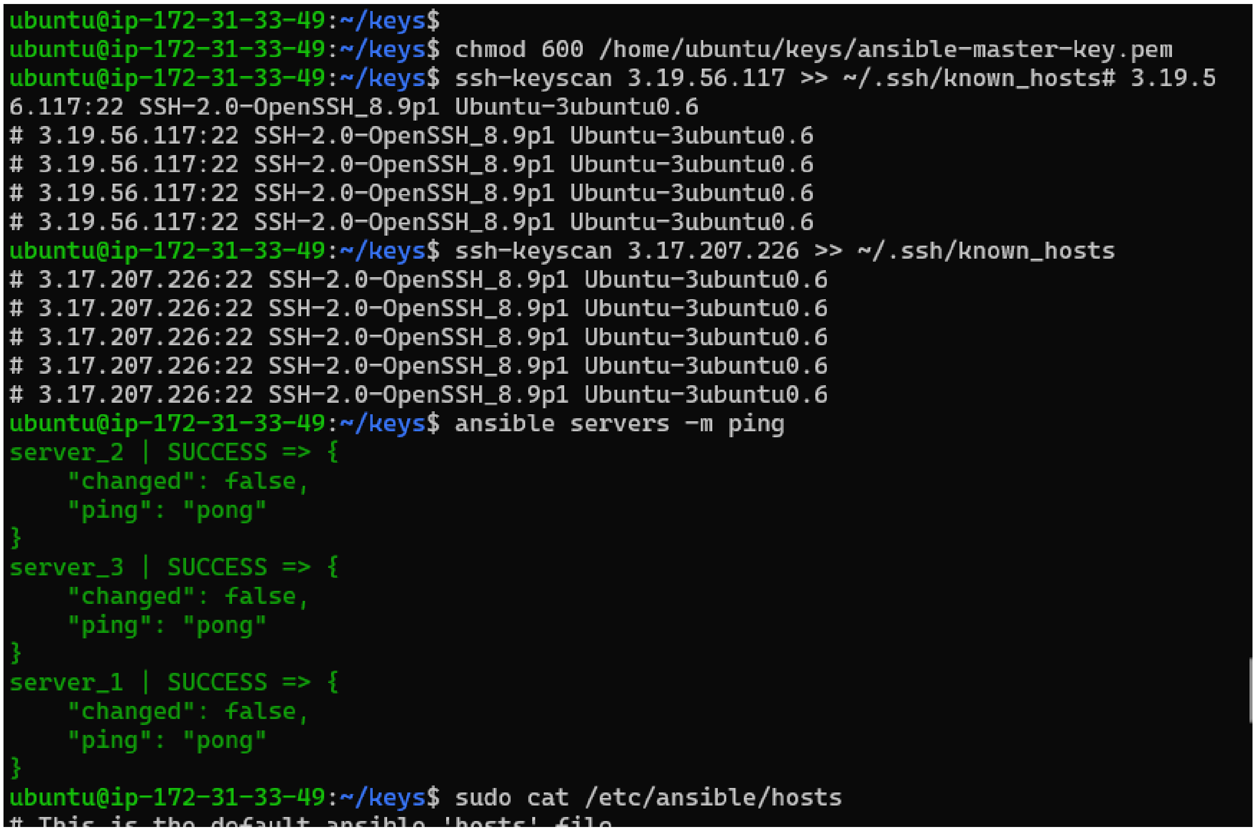
Server2 & 3 error : Host key verification failed

SOL : Host key verification ensures that the server you’re connecting to is the expected one.
Let’s manually add the host key for server2&3 to ~/.ssh/known_hosts

ssh-keyscan 3.19.56.117 >> ~/.ssh/known_hosts – for server2
ssh-keyscan 3.17.207.226 >> ~/.ssh/known_hosts – for server3

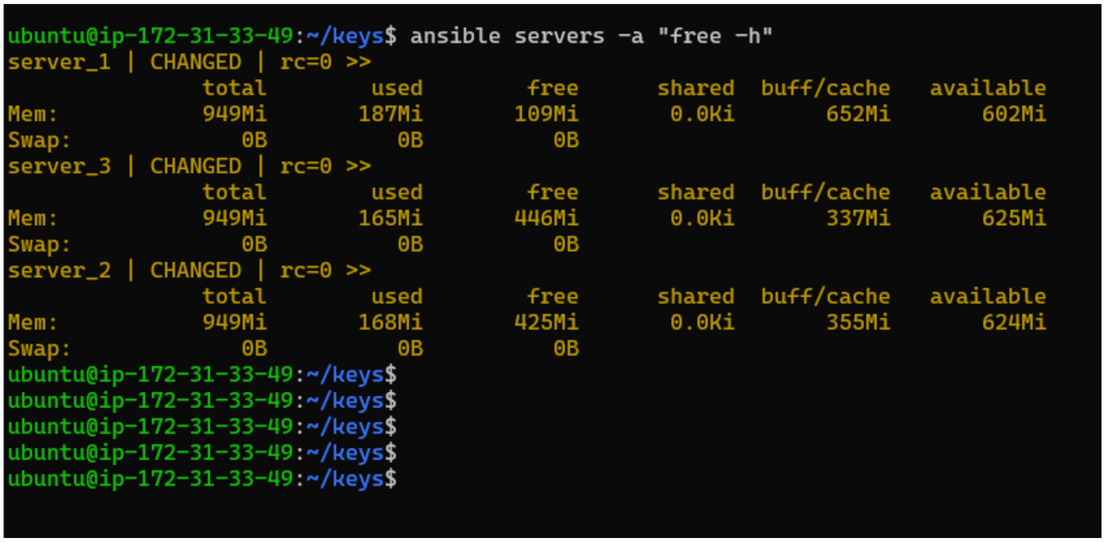
The ssh-keyscan command fetches the host key/s associated with IP addresses

Output : ansible servers -m ping



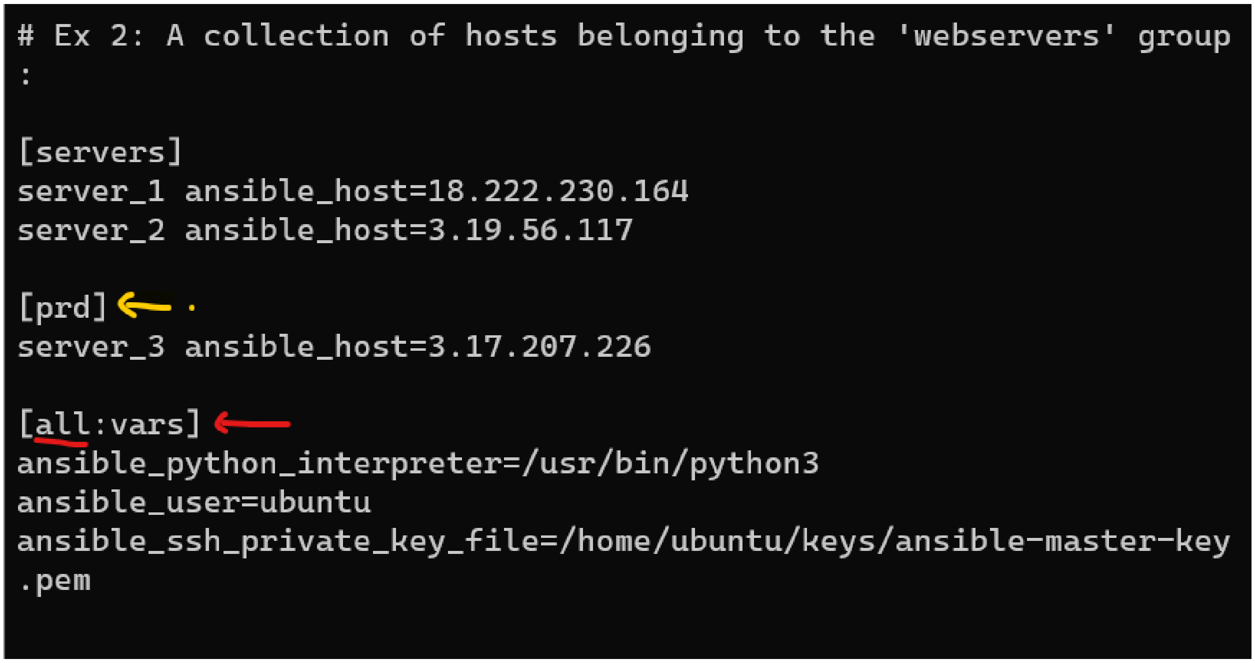
```
ubuntu@ip-172-31-33-49:~/keys$
ubuntu@ip-172-31-33-49:~/keys$ chmod 600 /home/ubuntu/keys/ansible-master-key.pem
ubuntu@ip-172-31-33-49:~/keys$ ssh-keyscan 3.19.56.117 >> ~/.ssh/known_hosts# 3.19.5
6.117:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
# 3.19.56.117:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
# 3.19.56.117:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
# 3.19.56.117:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
# 3.19.56.117:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
ubuntu@ip-172-31-33-49:~/keys$ ssh-keyscan 3.17.207.226 >> ~/.ssh/known_hosts
# 3.17.207.226:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
# 3.17.207.226:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
# 3.17.207.226:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
# 3.17.207.226:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
# 3.17.207.226:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
ubuntu@ip-172-31-33-49:~/keys$ ansible servers -m ping
server_2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
server_3 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
server_1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-33-49:~/keys$ sudo cat /etc/ansible/hosts
# This is the default ansible 'hosts' file
```

From the above image we can see that ping is successful this time.



```
ubuntu@ip-172-31-33-49:~/keys$ ansible servers -a "free -h"
server_1 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:           949Mi       187Mi       109Mi       0.0Ki       652Mi       602Mi
Swap:             0B          0B          0B
server_3 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:           949Mi       165Mi       446Mi       0.0Ki       337Mi       625Mi
Swap:             0B          0B          0B
server_2 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:           949Mi       168Mi       425Mi       0.0Ki       355Mi       624Mi
Swap:             0B          0B          0B
ubuntu@ip-172-31-33-49:~/keys$
ubuntu@ip-172-31-33-49:~/keys$
ubuntu@ip-172-31-33-49:~/keys$
ubuntu@ip-172-31-33-49:~/keys$
ubuntu@ip-172-31-33-49:~/keys$
```

- Let’s move one of the server to a new-group



```
# Ex 2: A collection of hosts belonging to the 'webservers' group
:

[servers]
server_1 ansible_host=18.222.230.164
server_2 ansible_host=3.19.56.117

[prd]
server_3 ansible_host=3.17.207.226

[all:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/ubuntu/keys/ansible-master-key
.pem
```

Command to check which server belong to which group and its functions/properties :

ansible-inventory --list

PLAYBOOK : Playbooks are YAML files that define a set of tasks and configurations to be executed on target hosts.

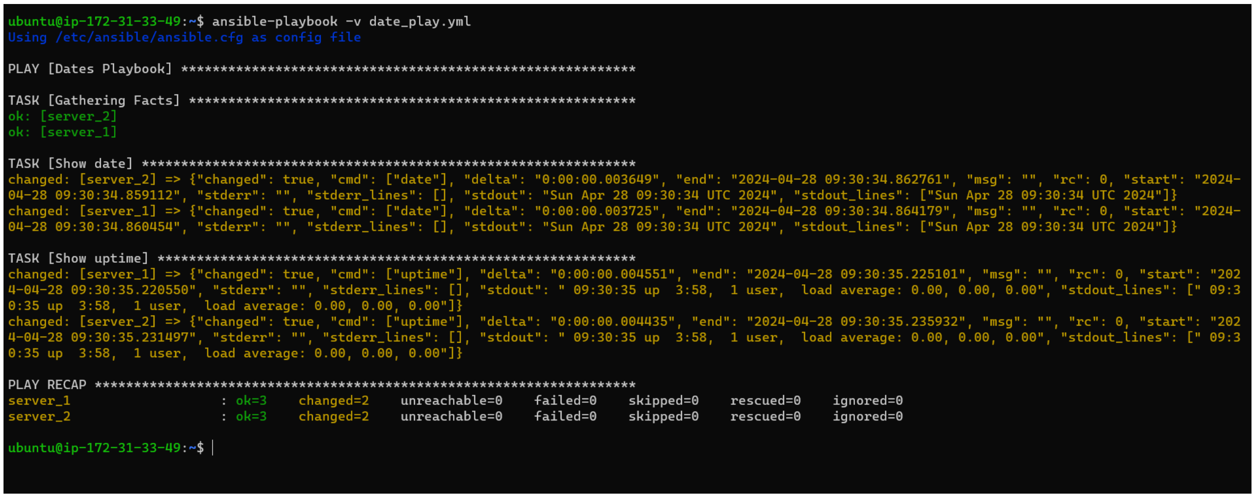
Tasks: Tasks are the individual units of work in Ansible. It represent actions to be performed on target hosts.

- Let’s create a playbook where we display the date and the uptime

Create a .yml file and the code is

```
-
  name: Dates Playbook
  hosts: servers
  tasks:
    - name: Show date
      command: date

    - name: Show uptime
      command: uptime
```

Command for running a ansible playbook is
ansible-playbook date_play.yml — where date_play.yml is playbook name

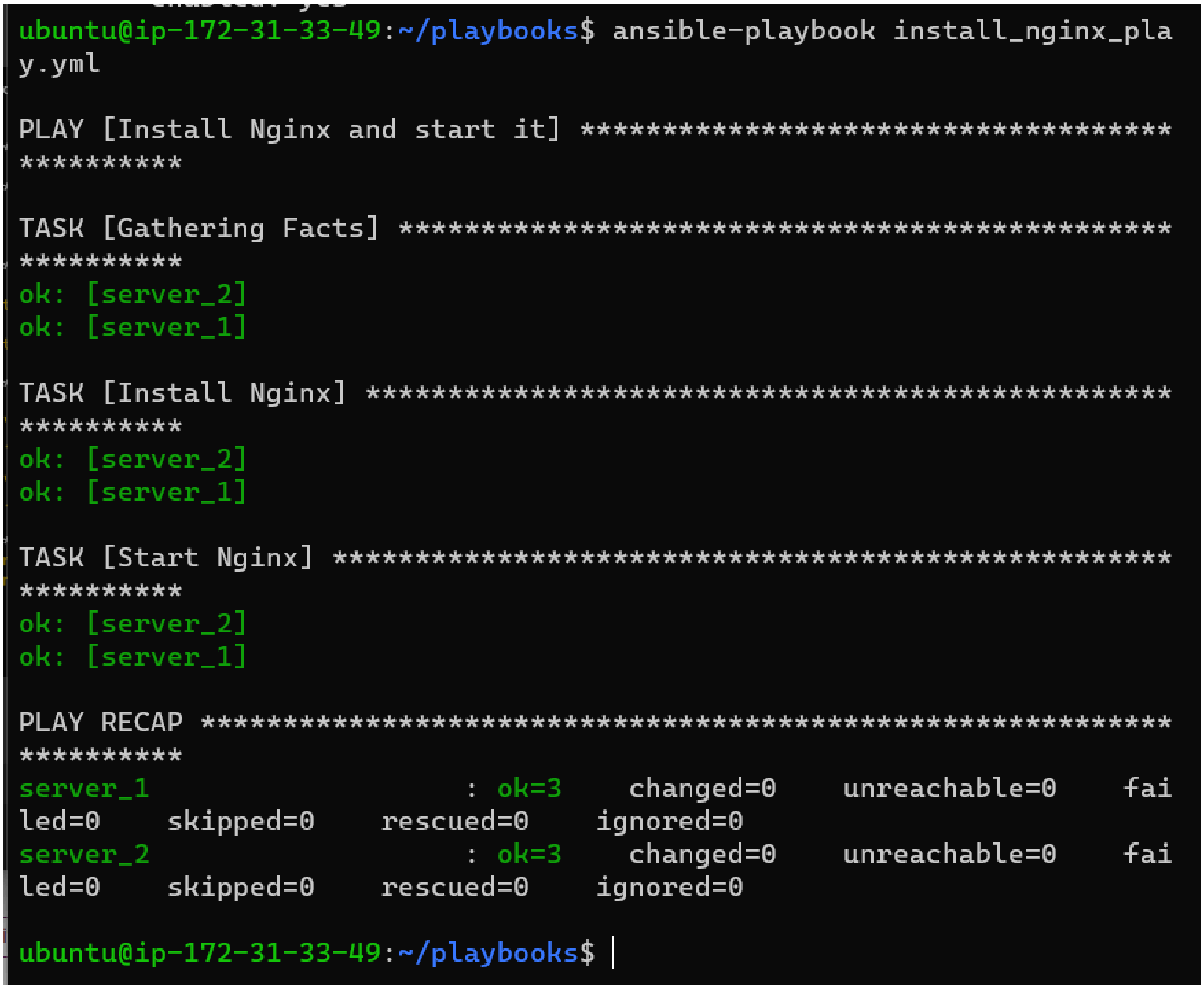


```
ubuntu@ip-172-31-33-49:~$ ansible-playbook -v date_play.yml
Using /etc/ansible/ansible.cfg as config file

PLAY [Dates Playbook] **********************************************************

TASK [Gathering Facts] *********************************************************
ok: [server_2]
ok: [server_1]

TASK [Show date] ***************************************************************
changed: [server_2] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.003649", "end": "2024-04-28 09:30:34.862761", "msg": "", "rc": 0, "start": "2024-
04-28 09:30:34.859112", "stderr": "", "stderr_lines": [], "stdout": "Sun Apr 28 09:30:34 UTC 2024", "stdout_lines": ["Sun Apr 28 09:30:34 UTC 2024"]}
changed: [server_1] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.003725", "end": "2024-04-28 09:30:34.864179", "msg": "", "rc": 0, "start": "2024-
04-28 09:30:34.860454", "stderr": "", "stderr_lines": [], "stdout": "Sun Apr 28 09:30:34 UTC 2024", "stdout_lines": ["Sun Apr 28 09:30:34 UTC 2024"]}

TASK [Show uptime] *************************************************************
changed: [server_1] => {"changed": true, "cmd": ["uptime"], "delta": "0:00:00.004551", "end": "2024-04-28 09:30:35.225101", "msg": "", "rc": 0, "start": "202
4-04-28 09:30:35.220550", "stderr": "", "stderr_lines": [], "stdout": " 09:30:35 up  3:58,  1 user,  load average: 0.00, 0.00, 0.00", "stdout_lines": [" 09:3
0:35 up  3:58,  1 user,  load average: 0.00, 0.00, 0.00"]}
changed: [server_2] => {"changed": true, "cmd": ["uptime"], "delta": "0:00:00.004435", "end": "2024-04-28 09:30:35.235932", "msg": "", "rc": 0, "start": "202
4-04-28 09:30:35.231497", "stderr": "", "stderr_lines": [], "stdout": " 09:30:35 up  3:58,  1 user,  load average: 0.00, 0.00, 0.00", "stdout_lines": [" 09:3
0:35 up  3:58,  1 user,  load average: 0.00, 0.00, 0.00"]}

PLAY RECAP *********************************************************************
server_1                   : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server_2                   : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-33-49:~$ |
```

- Let install nginx and start nginx from ansible

Create a .yml file and the code is

```
-
  name: Install Nginx and start it
  hosts: servers    #group-name
  become: yes    # like a root user
  tasks:
    - name: Install Nginx
      apt:
       name: nginx
       state: latest

    - name: Start Nginx
      service:
        name: nginx
        state: started   #
        enabled: yes   # Nginx service is running and set to start automatically
```



```
        enabled: yes
ubuntu@ip-172-31-33-49:~/playbooks$ ansible-playbook install_nginx_pla
y.yml

PLAY [Install Nginx and start it] ************************************
**********

TASK [Gathering Facts] ***********************************************
**********
ok: [server_2]
ok: [server_1]

TASK [Install Nginx] *************************************************
**********
ok: [server_2]
ok: [server_1]

TASK [Start Nginx] ***************************************************
**********
ok: [server_2]
ok: [server_1]

PLAY RECAP ***********************************************************
**********
server_1                   : ok=3    changed=0    unreachable=0    fai
led=0    skipped=0    rescued=0    ignored=0
server_2                   : ok=3    changed=0    unreachable=0    fai
led=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-33-49:~/playbooks$ 
```