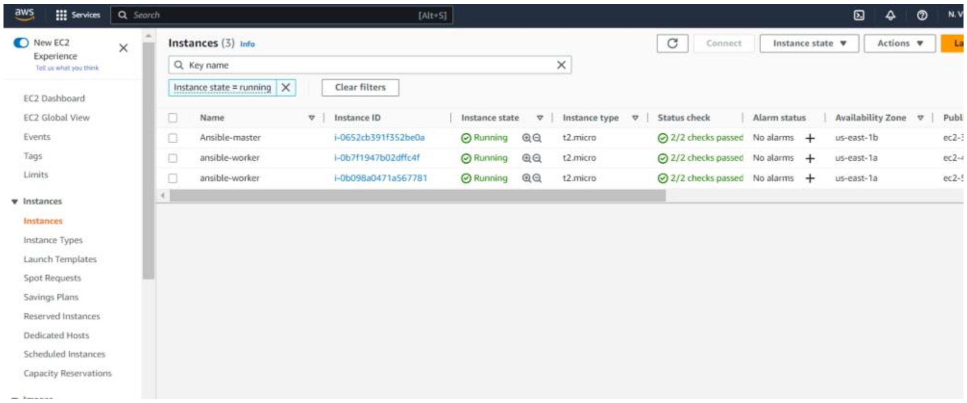
Day 58: Ansible Playbooks

Ansible playbooks run multiple tasks, assign roles, and define configurations, deployment steps, and variables. If you're using multiple servers, Ansible playbooks organize the steps between the assembled machines or servers and get them organized and running in the way the users need them to. Consider playbooks as the equivalent of instruction manuals.

Task-01

- Write an ansible playbook to create a file on a different server

First Launch 3 EC2 instances. 1 for Ansible Master and 2 for Ansible worker.



Instances (3) Info

Key name

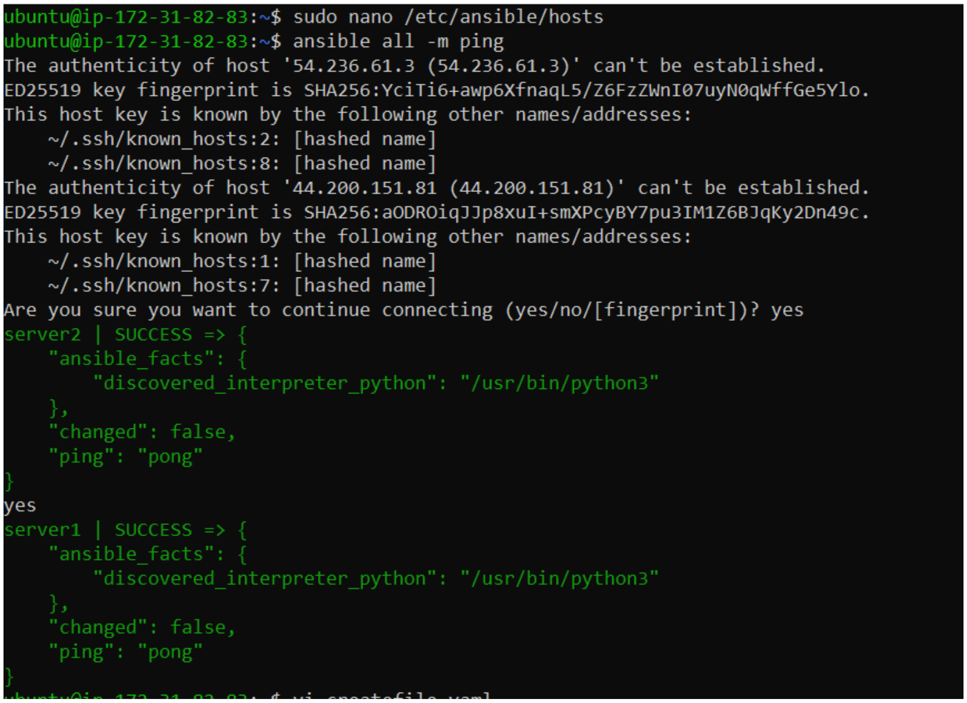
Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
Ansible-master	i-0652cb391f352be0a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-
ansible-worker	i-0b7f1947b02dffc4f	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-
ansible-worker	i-0b098a0471a567781	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-

Go to Ansible Master server and add repository and install ansible.

And try to connect to ansible worker.

As we did last Task, follow previous Tasks.



```
ubuntu@ip-172-31-82-83:~$ sudo nano /etc/ansible/hosts
ubuntu@ip-172-31-82-83:~$ ansible all -m ping
The authenticity of host '54.236.61.3 (54.236.61.3)' can't be established.
ED25519 key fingerprint is SHA256:YciTi6+awp6XfnaqL5/Z6FzZWnI07uyN0qWffGe5Ylo.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:2: [hashed name]
    ~/.ssh/known_hosts:8: [hashed name]
The authenticity of host '44.200.151.81 (44.200.151.81)' can't be established.
ED25519 key fingerprint is SHA256:aODROiqJJp8xuI+smXPcyBY7pu3IM1Z6BJqKy2Dn49c.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:7: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
server2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
yes
server1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-82-83:~$ vi createfile.yaml
```

Create an ansible playbook to for creating a simple text file on all ansible worker servers.

Command Explanation:

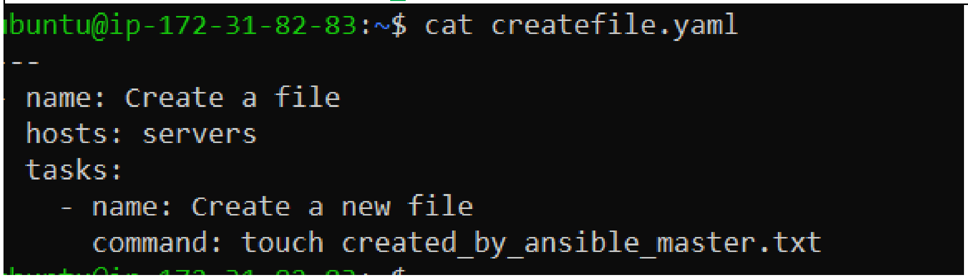
name: Give a simple name
hosts: put ansible worker group name or single worker name
tasks: Create a simple task

- **name:** Task name

Command: command is a module, used to run a command

command: **touch** **created_by_ansible_master.txt**

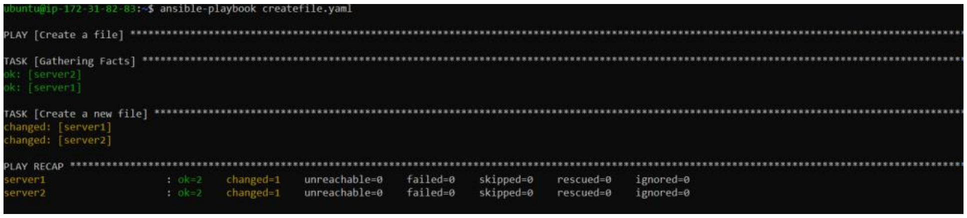
Command_name File name



```
ubuntu@ip-172-31-82-83:~$ cat createfile.yaml
---
- name: Create a file
  hosts: servers
  tasks:
    - name: Create a new file
      command: touch created_by_ansible_master.txt
ubuntu@ip-172-31-82-83:~$
```

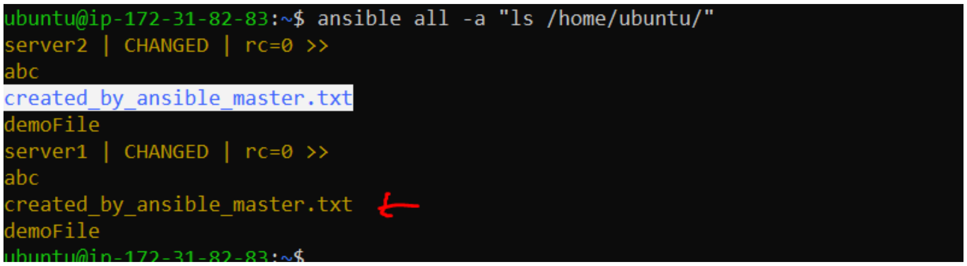
Run Ansible playbook using below command

Ansible-playbook <playbook_name>.yaml



```
ubuntu@ip-172-31-82-83:~$ ansible-playbook createfile.yaml

PLAY [Create a file] *********************************************************************************************************

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

TASK [Create a new file] *****************************************************************************************************
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
```

Verify file created in all ansible worker servers



```
ubuntu@ip-172-31-82-83:~$ ansible all -a "ls /home/ubuntu/"
server2 | CHANGED | rc=0 >>
abc
created_by_ansible_master.txt
demoFile
server1 | CHANGED | rc=0 >>
abc
created_by_ansible_master.txt
demoFile
ubuntu@ip-172-31-82-83:~$
```

- Write an ansible playbook to create a new user.

Create an ansible playbook to for creating a user on all ansible worker servers.

Command Explanation:

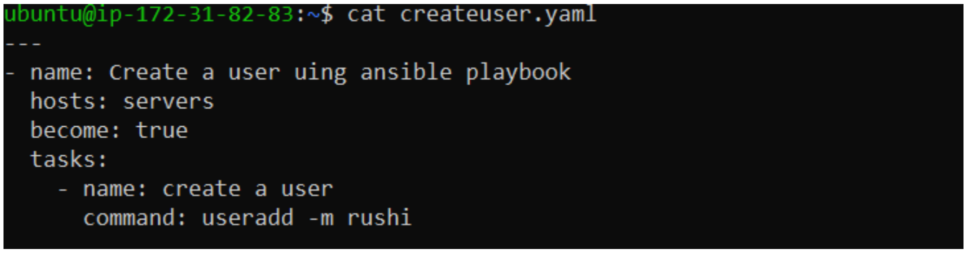
name: Give a simple name
hosts: put ansible worker group name or single worker name
become: true means it allows all permissions i.e., Sudo
tasks: Create a simple task

- **name:** Task name

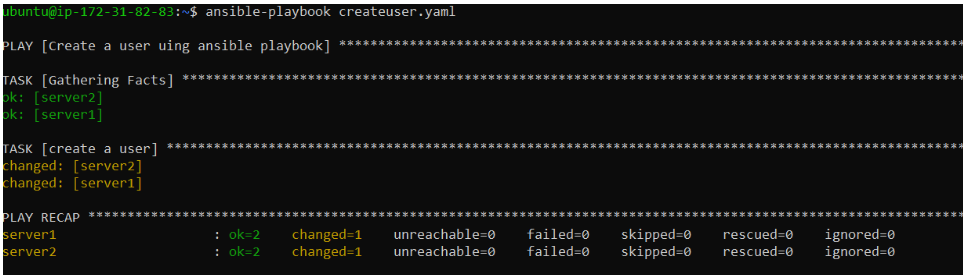
Command: command is a module, used to run a command

command: **useradd -a** **rushi**

Command_name Username

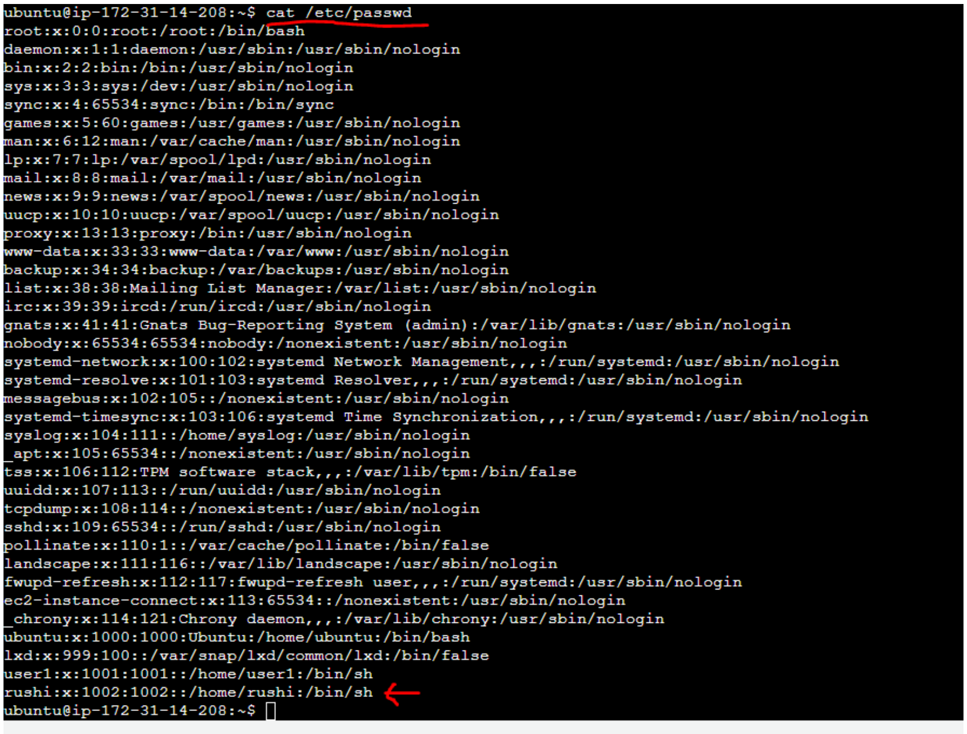


```
ubuntu@ip-172-31-82-83:~$ cat createuser.yaml
---
- name: Create a user uing ansible playbook
  hosts: servers
  become: true
  tasks:
    - name: create a user
      command: useradd -m rushi
```



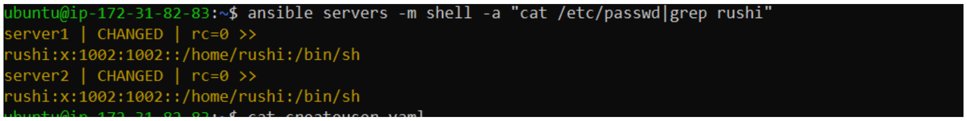
```
ubuntu@ip-172-31-82-83:~$ ansible-playbook createuser.yaml

PLAY [Create a user uing ansible playbook] ********************************************************************************

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

TASK [create a user] ******************************************************************************************************
changed: [server2]
changed: [server1]

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
```

Goto Ansible worker EC2 server and put **cat /etc/passwd** command to see all **users**.



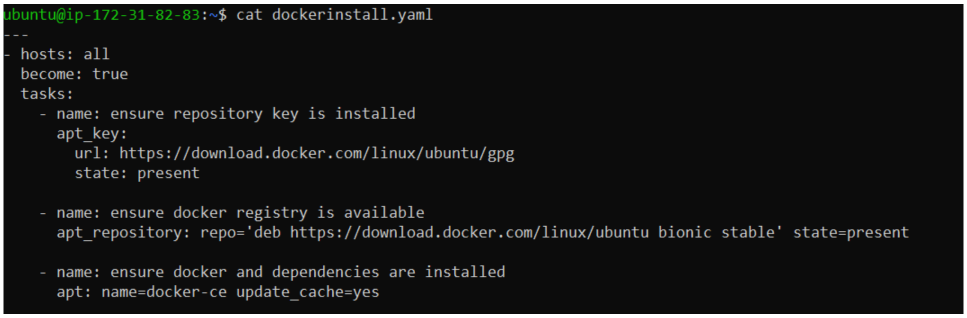
```
ubuntu@ip-172-31-14-208:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
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
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
syslog:x:104:111::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:112:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:113::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:114::/nonexistent:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
landscape:x:111:116::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:112:117:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
ec2-instance-connect:x:113:65534::/nonexistent:/usr/sbin/nologin
_chrony:x:114:121:Chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
user1:x:1001:1001::/home/user1:/bin/sh
rushi:x:1002:1002::/home/rushi:/bin/sh
ubuntu@ip-172-31-14-208:~$
```

Verify the user is created in all ansible worker servers

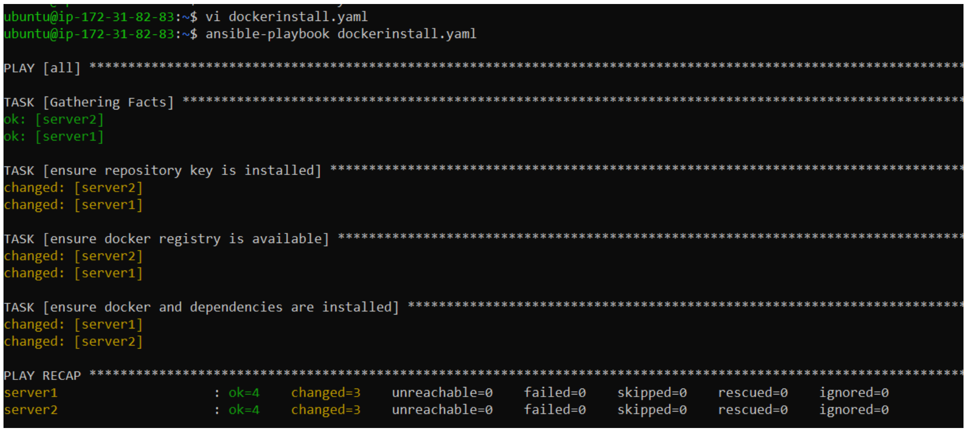


```
ubuntu@ip-172-31-82-83:~$ ansible servers -m shell -a "cat /etc/passwd|grep rushi"
server1 | CHANGED | rc=0 >>
rushi:x:1002:1002::/home/rushi:/bin/sh
server2 | CHANGED | rc=0 >>
rushi:x:1002:1002::/home/rushi:/bin/sh
ubuntu@ip-172-31-82-83:~$ cat createuser.yaml
```

- Write an ansible playbook to install docker on a group of servers

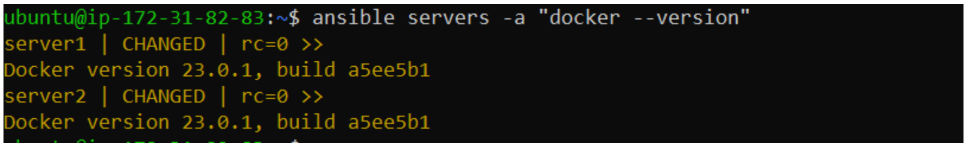


```
ubuntu@ip-172-31-82-83:~$ cat dockerinstall.yaml
---
- hosts: all
  become: true
  tasks:
    - name: ensure repository key is installed
      apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
        state: present

    - name: ensure docker registry is available
      apt_repository: repo='deb https://download.docker.com/linux/ubuntu bionic stable' state=present

    - name: ensure docker and dependencies are installed
      apt: name=docker-ce update_cache=yes
```



```
ubuntu@ip-172-31-82-83:~$ vi dockerinstall.yaml
ubuntu@ip-172-31-82-83:~$ ansible-playbook dockerinstall.yaml

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

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

TASK [ensure repository key is installed] **************************************************************
changed: [server2]
changed: [server1]

TASK [ensure docker registry is available] *************************************************************
changed: [server2]
changed: [server1]

TASK [ensure docker and dependencies are installed] ****************************************************
changed: [server1]
changed: [server2]

PLAY RECAP *********************************************************************************************
server1                    : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server2                    : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Verify the docker is install in all ansible worker servers



```
ubuntu@ip-172-31-82-83:~$ ansible servers -a "docker --version"
server1 | CHANGED | rc=0 >>
Docker version 23.0.1, build a5ee5b1
server2 | CHANGED | rc=0 >>
Docker version 23.0.1, build a5ee5b1
```

Watch this video to learn about ansible Playbooks

Task-02

- Write a blog about writing ansible playbooks with the best practices.

1. “Name” Your Plays and Tasks
Always name your plays and tasks. Adding names with a human meaningful description better communicates the intent to users when running a play.

2. Use Prefixes and Human Meaningful Names with Variables
Ansible has a powerful variable processing system that collects metadata from various sources and manages their merge and context as a play run on your hosts. A lot of effort goes into making that power as easy and transparent as possible to users.

```
apache_max_keepalive: 25
apache_port: 80
tomcat_port: 8080
```

3. Use Modules Before Run Commands
Run commands are what we collectively call the command, shell, raw and script modules that enable users to do command line operations in different ways. They’re a great catch all mechanism for getting things done, but they should be used sparingly and as a last resort.

4. Clean Up Your Debugging Messages
When developing Ansible content, it can be useful to drop in a debug task to display the content of a variable while your play runs. That’s not so nice, even downright disconcerting, when your Playbook goes into production and some unwitting ops manager runs a play and sees your debugging messages on their screen.

Let me or anyone in the community know if you face any challenges

happy Learning :)