Ansible Setup and Target Machine Configuration

Steps to Set Up Ansible in Docker Containers Documantaion od project

1. **Set up the VM:**
   - **Create a GCP or AWS VM with Ubuntu Server as the OS.**
   - **Update the system using:** *apt update && apt upgrade -y*
   - **Install Docker:** *apt install docker.io -y* **and verify the Ubuntu image.**
2. **Create Docker containers:**
   - **Create three containers:**
     - **Ansible-master:**
       docker run -dit --name ansible-master ubuntu /bin/bash
     - **Target1:**
       docker run -it -d --name target1 ubuntu /bin/bash
     - **Target2:**
       docker run -it -d --name target2 ubuntu /bin/bash

   – *Note: Flag dit and -it -d are the same.*

   - **Verify running containers with:** *docker ps*
3. **Deploy packages in Ansible-master:**
   - **Login to the container:**
     docker exec -it ansible-master bash.
   - **Install required packages:** python3 or minimal would work pointing py3
     apt install python-is-python3 vim iputils-ping openssh-client -y.
4. **Install Software Properties Common:**
   - **Run:** *apt install software-properties-common*
     to enable repository management tools.
5. **Add Ansible Repository and Install Ansible:**
   - **Add the PPA:**
     add-apt-repository --yes --update ppa:ansible/ansible
   - **Install Ansible:** *apt install ansible*

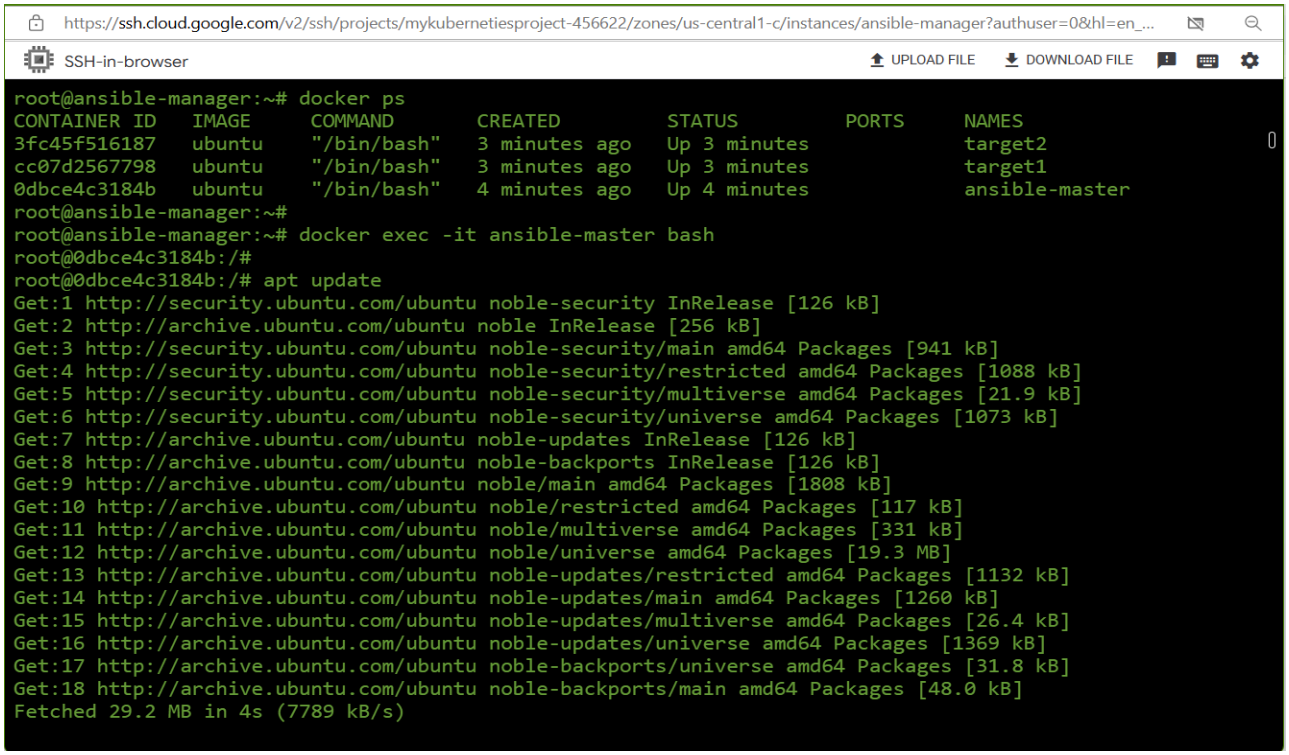
Next stage: Project implementation as follows in page 2

Step 1: Setting up the Virtual Machine (VM)

- ├── **Create a VM**
- │ - Choose GCP VM or AWS VM.
- │ - Select Ubuntu Server as the operating system.
- │
- ├── **Update the VM**
- │ - **Run**: **`*apt update && apt upgrade -y*`**
- └── **Install Docker**
- - Run: **`*apt install docker.io -y*`**
- - Pull the Ubuntu image and verify the container.
-

Step 2: Create Docker Containers

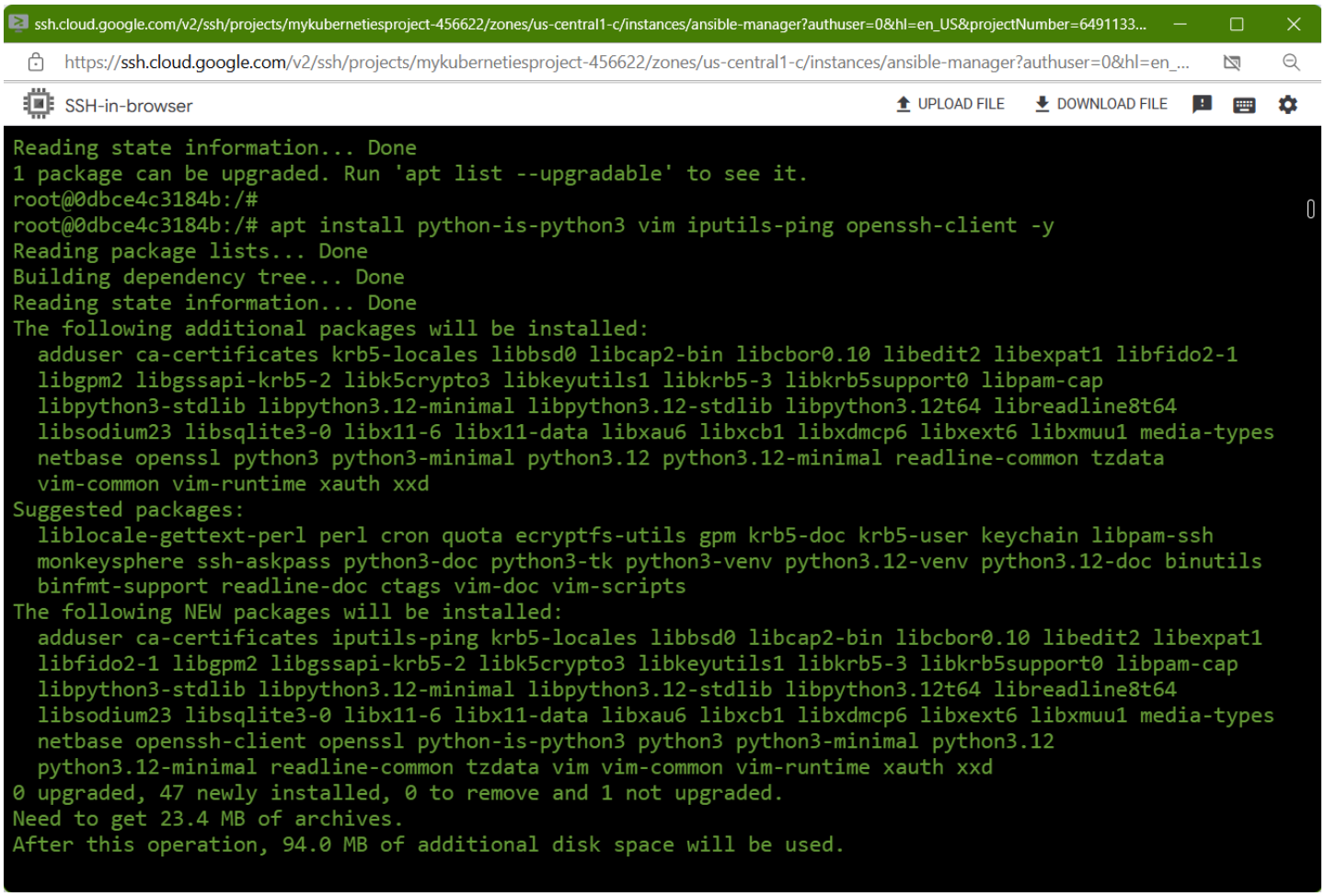
- ├── **Create containers**
- │ - **Run**: **`*docker run -it -d --name ansible-master ubuntu bash*`**
- │ - **Run**: **`*docker run -it -d --name target1 ubuntu /bin/bash*`**
- │ - **Run**: **`*docker run -it -d --name target2 ubuntu /bin/bash*`**
- │
- └── **Check running containers**
- - **Run**: **`*docker ps*`**



Step 3: Deploying Packages in the Ansible-Master Container

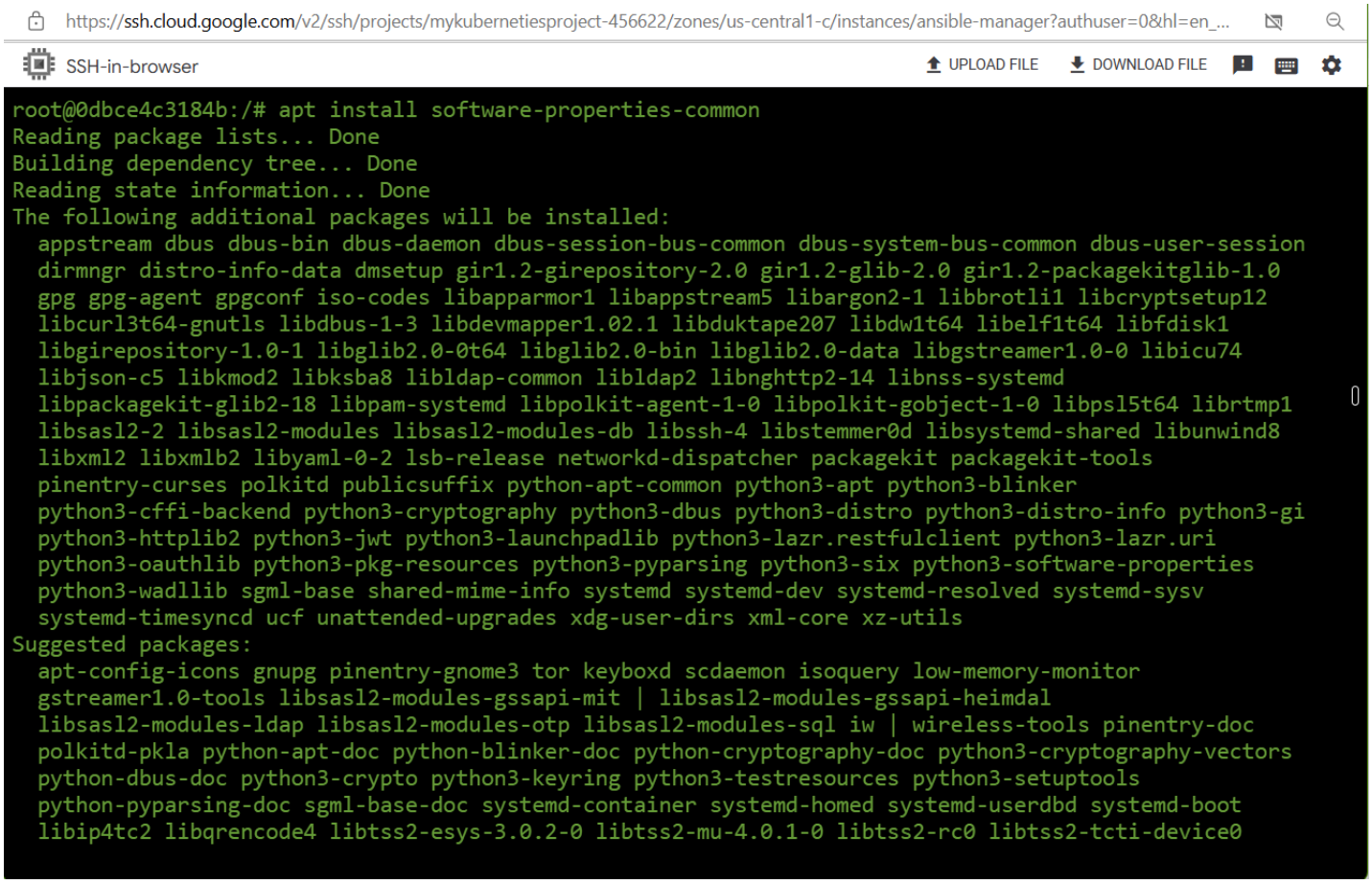
- ├── **Login to the container**
- │ - **Run**: **`*docker exec -it ansible-master bash*`**

- ├── **Install the required packages**
- │ - Run: **`*apt install python-is-python3 vim iputils-ping openssh-client -y*`**
-



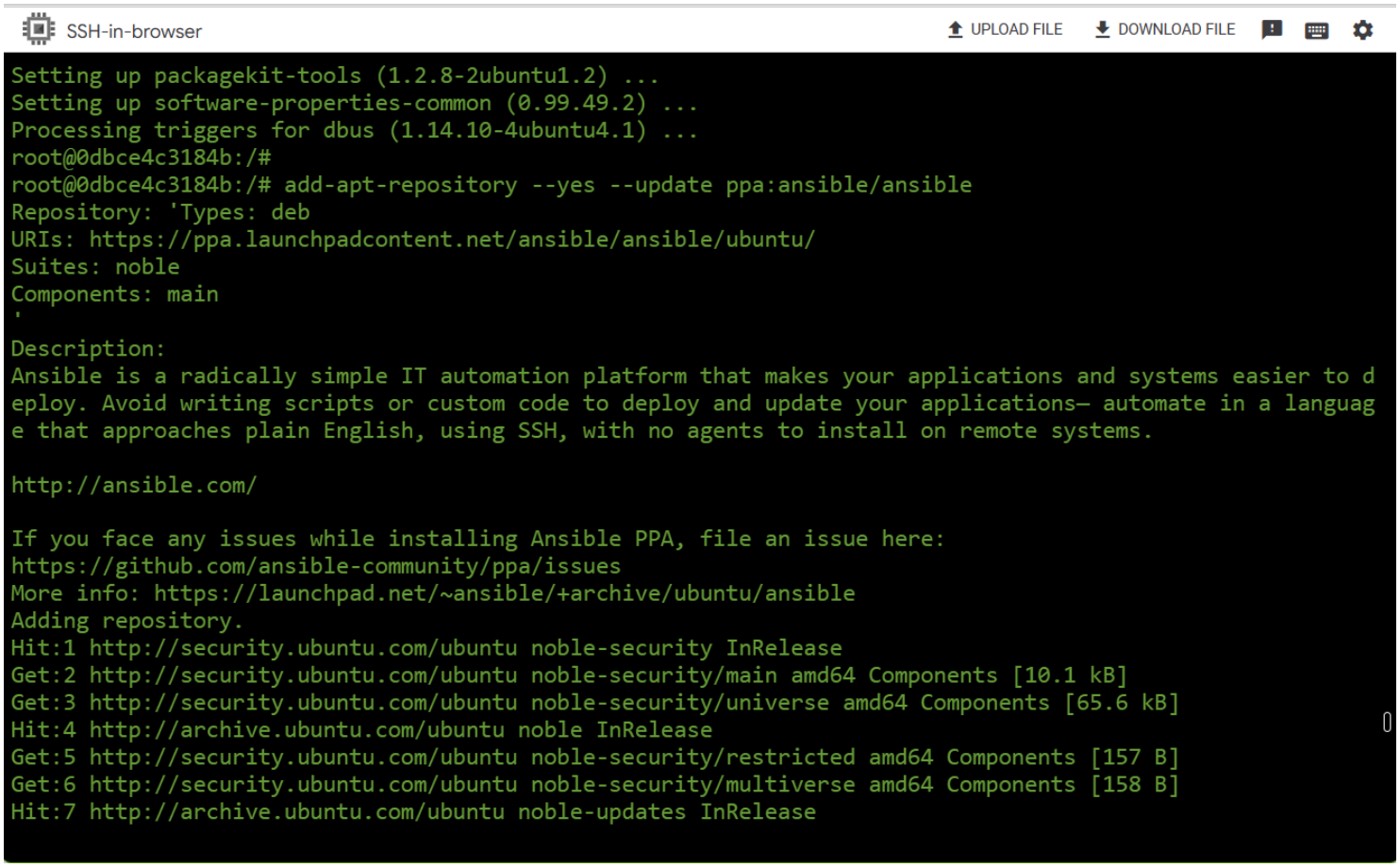
Step 4: Installing Software Properties

- ├── **Command Explanation**
- │ - The command **`*apt install software-properties-common*`** is used on Debian-based systems (e.g., Ubuntu).
- │ - It installs the package **`software-properties-common`**, which includes tools for managing software repositories.

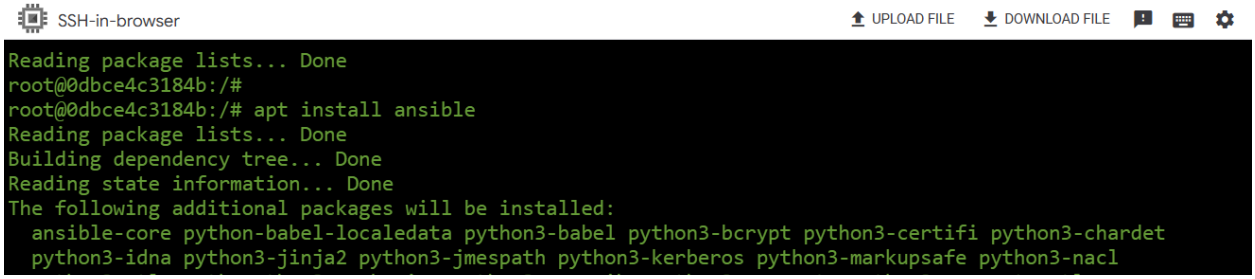


Step 5: Adding Ansible Repository and Installing Ansible

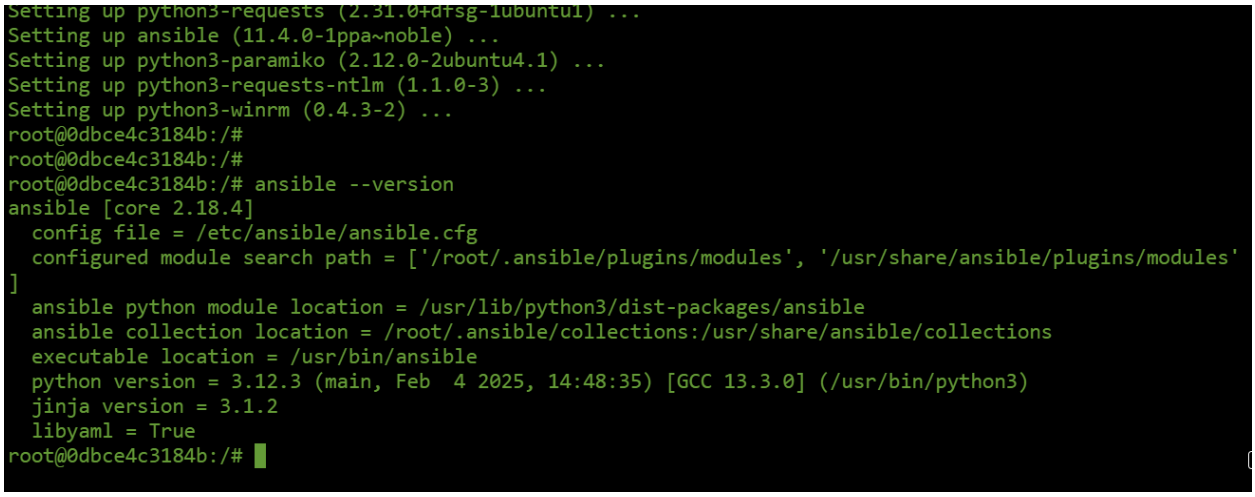
- ├── **Add Ansible's PPA (Personal Package Archive)**
- │ - Run: **`*add-apt-repository --yes --update ppa:ansible/ansible*`**
- │
- ├── **Install Ansible**
- │ - Run: **`*apt install ansible*`**



⇒ **Install Ansible**



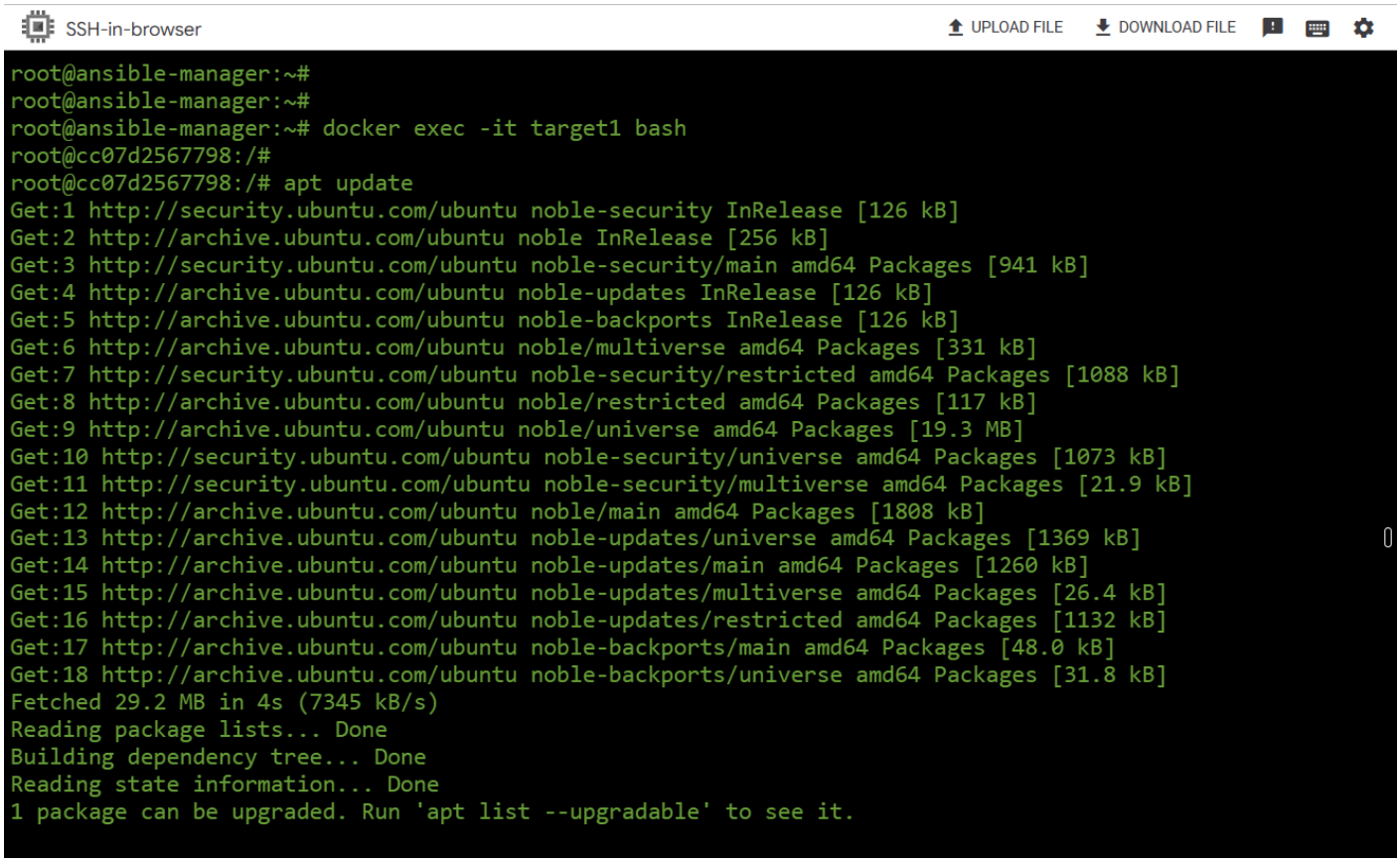
Check the version ⇒ ansible –version



Part 1 Completed Above ⇒ Ansible is installed on the Master

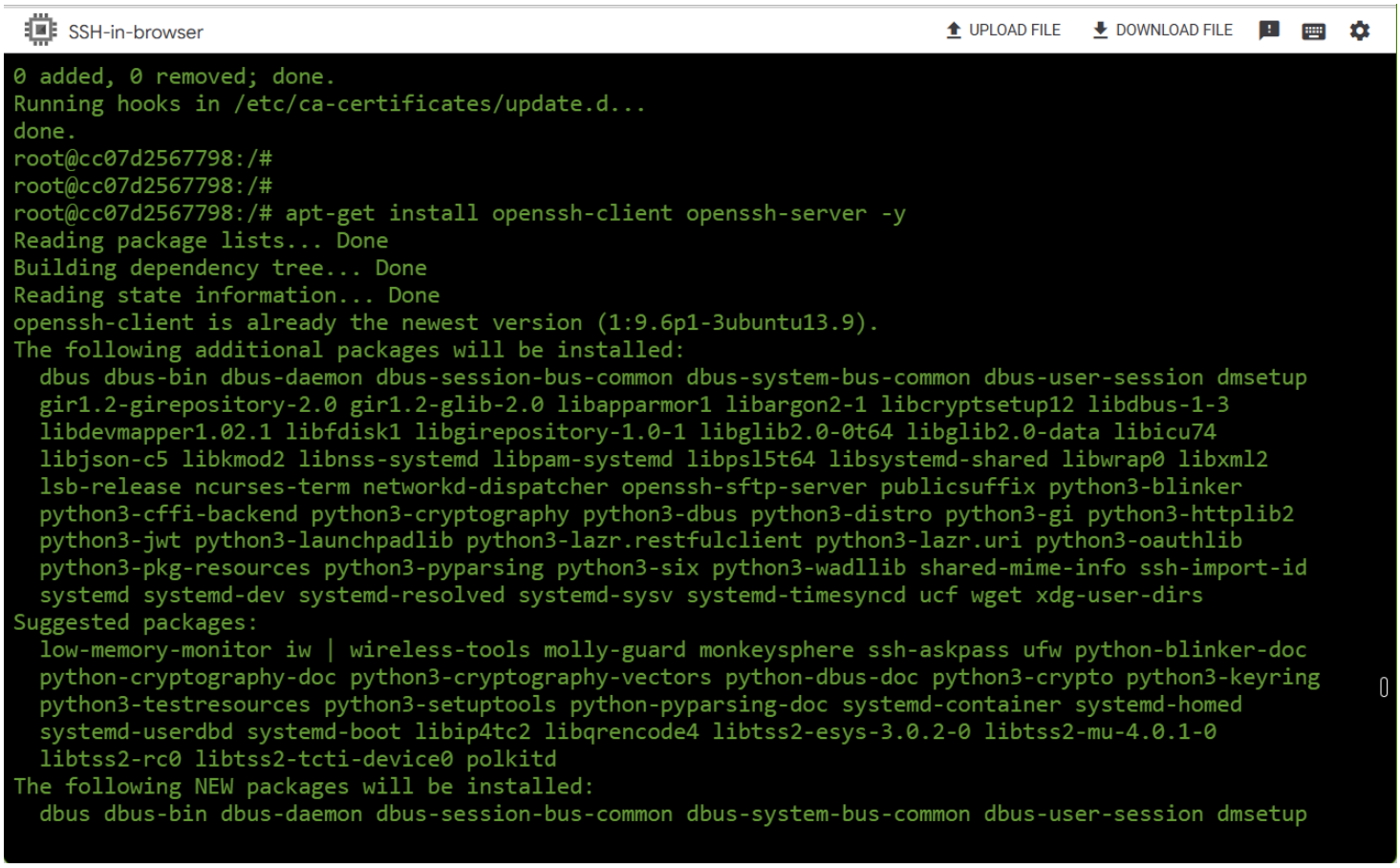
⇒ **Part 2: Install the Target machines. The steps are as follows.**

- Log in to the Target1 Container using the command
- **Run *docker exec -it target1 bash***



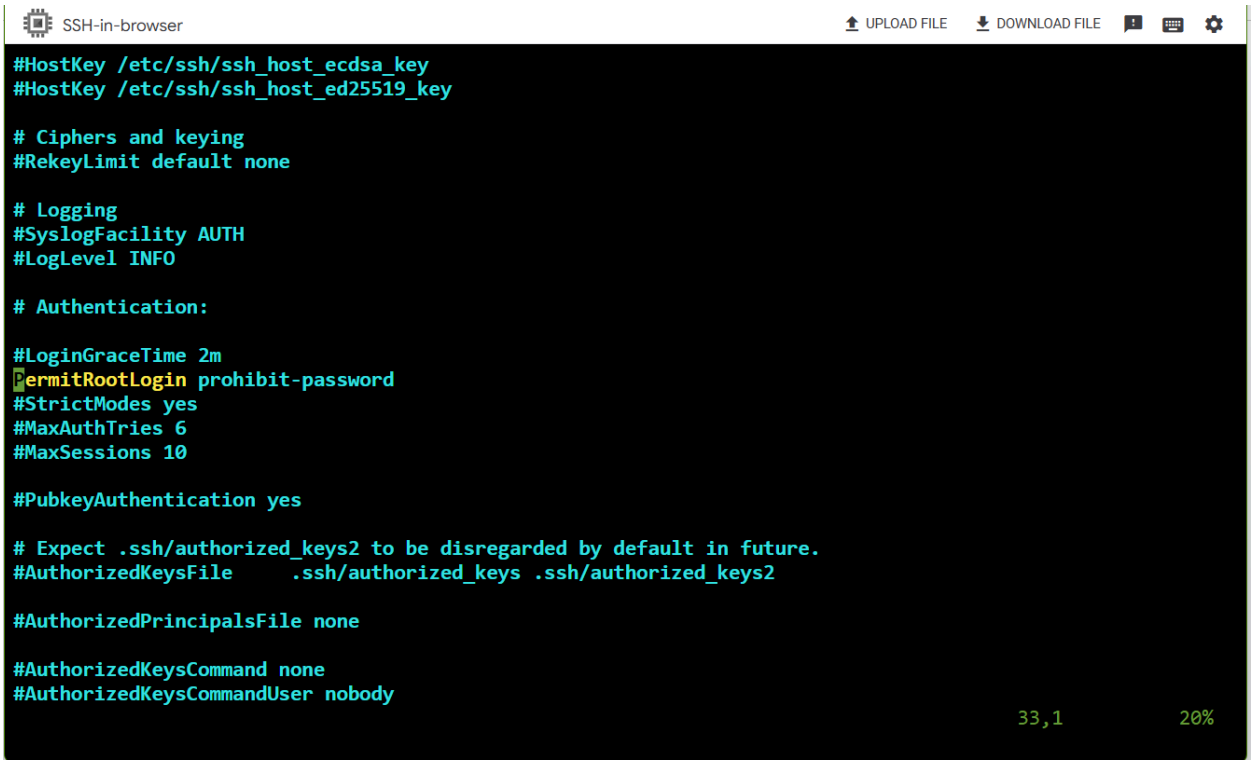
Step 2: Install dependencies as installed for master

- *apt install vim python-is-python3 iputils-ping*
- *apt-get install openssh-client openssh-server -y*



Step 3: Go to the ssh and edit the file

- *Cd /etc/ssh*
- *Vi sshd_config*



⇒ From Prohibited to change it to **Yes**

⇒ Change the password authentication, **which is commented**

```
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no
-- INSERT --
```

```
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no
-- INSERT --
```

Exit from the ssh config and verify the service.

```
root@cc07d2567798:~#
root@cc07d2567798:~#
root@cc07d2567798:~# cd /etc/ssh
root@cc07d2567798:/etc/ssh#
root@cc07d2567798:/etc/ssh#
root@cc07d2567798:/etc/ssh# ls
moduli      ssh_config.d       ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub  sshd_config
ssh_config  ssh_host_ecdsa_key  ssh_host_ed25519_key    ssh_host_rsa_key          ssh_import_id         sshd_config.d
root@cc07d2567798:/etc/ssh#
root@cc07d2567798:/etc/ssh# vi sshd_config
root@cc07d2567798:/etc/ssh#
root@cc07d2567798:/etc/ssh#
root@cc07d2567798:/etc/ssh#
```

⇒ Check the service if not running start

- Run *service ssh status*
- Run *service ssh start*

```
root@cc07d2567798:/etc/ssh# service ssh status
 * sshd is not running
root@cc07d2567798:/etc/ssh# service ssh start
 * Starting OpenBSD Secure Shell server sshd
root@cc07d2567798:/etc/ssh# service ssh status
 * sshd is running
root@cc07d2567798:/etc/ssh#
root@cc07d2567798:/etc/ssh#
root@cc07d2567798:/etc/ssh#
```

Change the root password

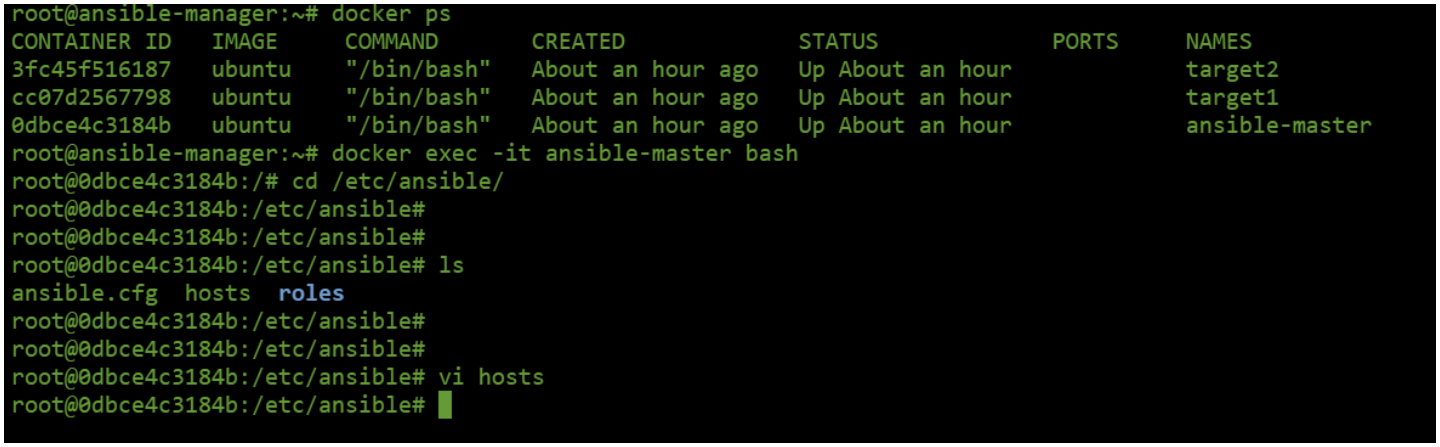
SSH-in-browser

```
root@cc07d2567798:/etc/ssh# passwd root
New password:
Retype new password:
passwd: password updated successfully
root@cc07d2567798:/etc/ssh#
root@cc07d2567798:/etc/ssh#
root@cc07d2567798:/etc/ssh# exit
exit
root@ansible-manager:~#
```

⇒ Using ***sudo docker inspect target1*** find the ipaddress

⇒ **GO to the Ansible Master and edit the hosts file**

Add the IP address to the Host file – **172.17.0.3**



Add Ip address to host file. 172.17.0.3

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

172.17.0.3

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

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

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:

## www[001:006].example.com

-- INSERT --
```

⇒ **Using a keygen to generate a Public and private key**

- **RUN** *ssh-keygen*

```
root@0dbce4c3184b:/etc/ansible# vi hosts
root@0dbce4c3184b:/etc/ansible#
root@0dbce4c3184b:/etc/ansible#
root@0dbce4c3184b:/etc/ansible# ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:yIgyacoEvKnByxHo3iyjxmDojn8urjGIt7DLWDjDpWI root@0dbce4c3184b
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|o                |
|oo               |
|+.+. o .         |
|*O... o S        |
|#*B              |
|^Eoo             |
|B&+..            |
|XB++.            |
+----[SHA256]-----+
root@0dbce4c3184b:/etc/ansible#
root@0dbce4c3184b:/etc/ansible#
root@0dbce4c3184b:/etc/ansible#
```

⇒ **Add the key to the remote machine 172.17.0.3**

- *ssh-copy-id root@172.17.0.3*

```
root@0dbce4c3184b:/etc/ansible# ssh-copy-id root@172.17.0.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_ed25519.pub"
The authenticity of host '172.17.0.3 (172.17.0.3)' can't be established.
ED25519 key fingerprint is SHA256:UTDRt/rsxL+VsRGfddfboZlg+b8P2VDNn+32kNfiHko.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.17.0.3's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@172.17.0.3'"
and check to make sure that only the key(s) you wanted were added.

root@0dbce4c3184b:/etc/ansible#
```

⇒ Connect client without password now using ***ssh root@172.17.0.3***

```
root@0dbce4c3184b:/etc/ansible# ssh-copy-id root@172.17.0.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_ed25519.pub"
The authenticity of host '172.17.0.3 (172.17.0.3)' can't be established.
ED25519 key fingerprint is SHA256:UTDRt/rsxL+VsRGfddfboZlg+b8P2VDNn+32kNfiHko.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.17.0.3's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@172.17.0.3'"
and check to make sure that only the key(s) you wanted were added.

root@0dbce4c3184b:/etc/ansible# ssh root@172.17.0.3
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 5.15.0-1078-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
root@cc07d2567798:~# exit
logout
Connection to 172.17.0.3 closed.
root@0dbce4c3184b:/etc/ansible#
```

⇒ **Add code to yaml file for the playbook**

To verify the yaml file use yaml lint check indentations.

- *Cd /etc/ansible*
- *VI playbook-name*

SSH-in-browser

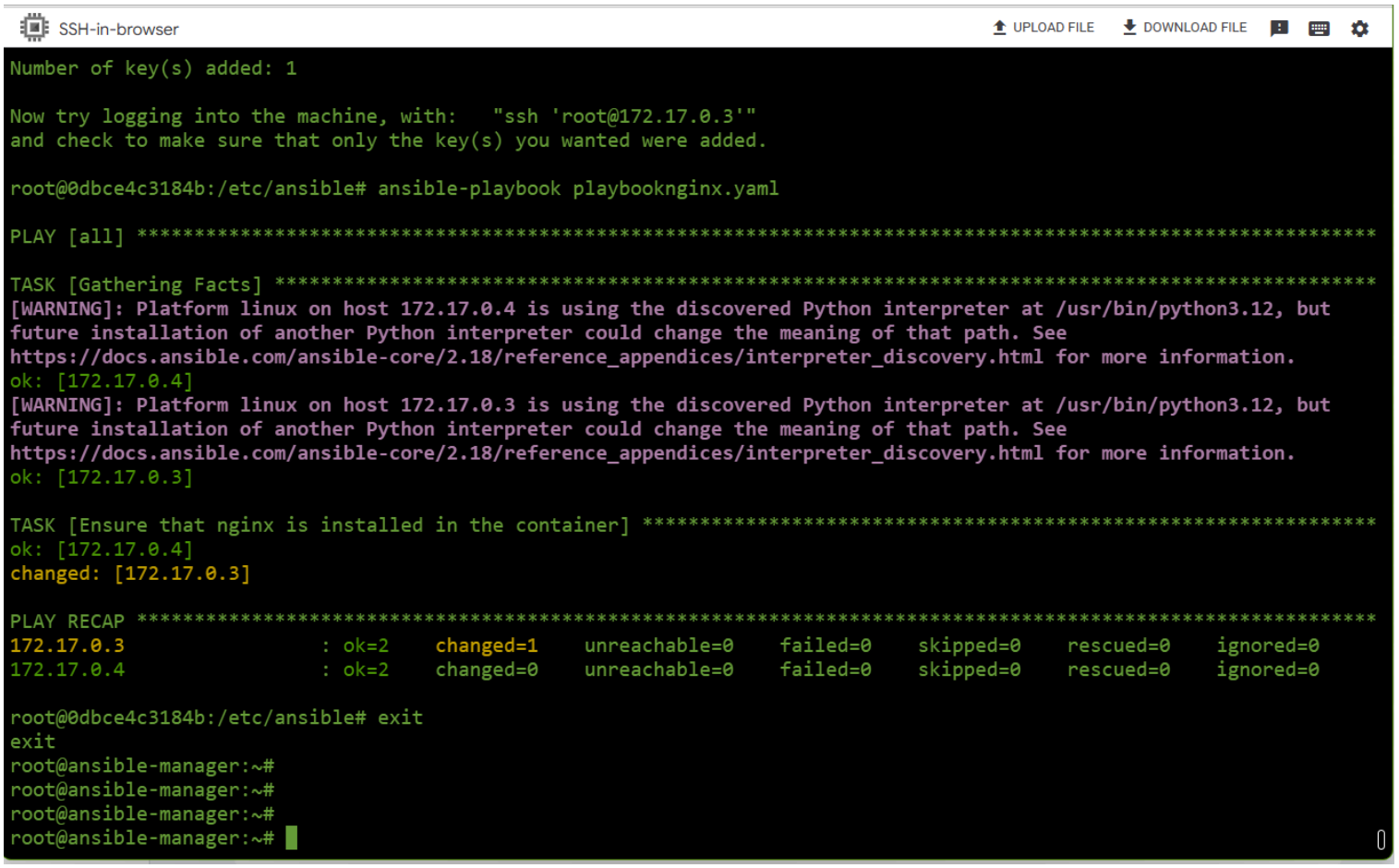
```
---
- hosts: all
  tasks:
    - name: Ensure that nginx is installed in the container
      apt: name=nginx state=latest
~
~
~
~
~
~
~
~
~
~
```

Run the playbook using cmd

⇒ *ansible-playbook playbooknginx.yaml*

```
root@0dbce4c3184b:/#
root@0dbce4c3184b:/# cd /etc/ansible/
root@0dbce4c3184b:/etc/ansible# vi playbooknginx.yaml
root@0dbce4c3184b:/etc/ansible# ls
ansible.cfg  hosts  playbooknginx.yaml  roles
root@0dbce4c3184b:/etc/ansible#
root@0dbce4c3184b:/etc/ansible# vi playbooknginx.yaml
root@0dbce4c3184b:/etc/ansible#
root@0dbce4c3184b:/etc/ansible# ansible-playbook playbooknginx.yaml

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

TASK [Gathering Facts] *********************************************************************************************
[WARNING]: Platform linux on host 172.17.0.3 is using the discovered Python interpreter at /usr/bin/python3.12, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [172.17.0.3]

TASK [Ensure that nginx is installed in the container] *************************************************************
changed: [172.17.0.3]

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

root@0dbce4c3184b:/etc/ansible#
```

Repeat the same to the target2 and the following result
⇒ Run all as similar to the target1 and the output as desired, shown in the image.



Submitted: for Approval. Engineer - Parvez M B

Verification: