

Ansible Setup and Target Machine Configuration

Steps to Set Up Ansible in Docker Containers Documentaion 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`****

```

https://ssh.cloud.google.com/v2/ssh/projects/mykubernetiesproject-456622/zones/us-central1-c/instances/ansible-manager?authuser=0&hl=en_...
SSH-in-browser
UPLOAD FILE DOWNLOAD FILE

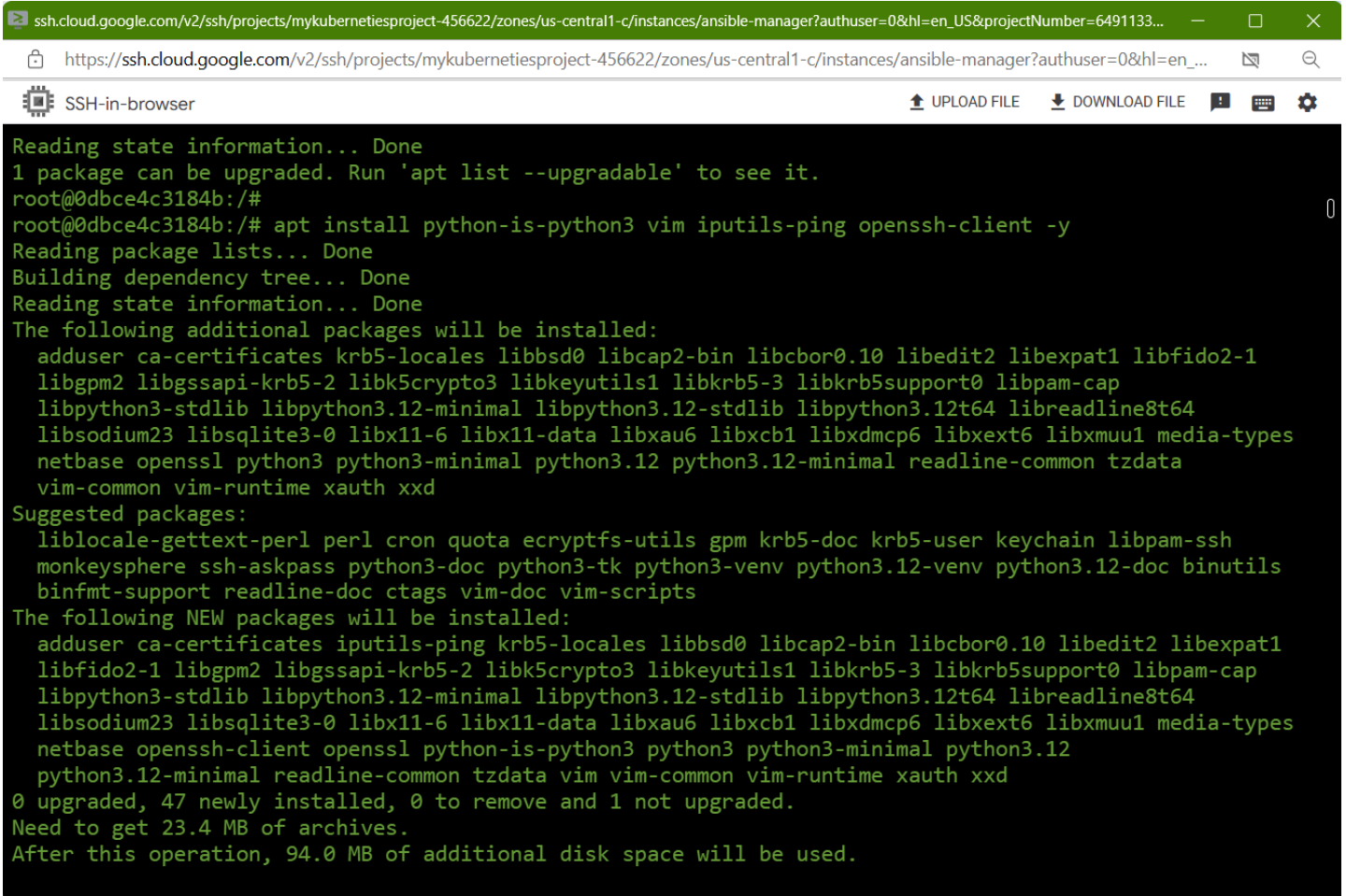
root@ansible-manager:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
3fc45f516187   ubuntu   "/bin/bash"             3 minutes ago Up 3 minutes           target2
cc07d2567798   ubuntu   "/bin/bash"             3 minutes ago Up 3 minutes           target1
0dbce4c3184b   ubuntu   "/bin/bash"             4 minutes ago Up 4 minutes           ansible-master
root@ansible-manager:~# docker exec -it ansible-master bash
root@0dbce4c3184b:/#
root@0dbce4c3184b:/# apt update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [941 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1088 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [21.9 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1073 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1132 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1260 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [26.4 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1369 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [31.8 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [48.0 kB]
Fetched 29.2 MB in 4s (7789 kB/s)

```

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`****
-



```
ssh.cloud.google.com/v2/ssh/projects/mykubernetesproject-456622/zones/us-central1-c/instances/ansible-manager?authuser=0&hl=en_US&projectNumber=6491133...
https://ssh.cloud.google.com/v2/ssh/projects/mykubernetesproject-456622/zones/us-central1-c/instances/ansible-manager?authuser=0&hl=en_...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
root@0dbce4c3184b:/#
root@0dbce4c3184b:/# apt install python-is-python3 vim iputils-ping openssh-client -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adduser ca-certificates krb5-locales libbsd0 libcap2-bin libcbor0.10 libedit2 libexpat1 libfido2-1
  libgpm2 libgssapi-krb5-2 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libpam-cap
  libpython3-stdlib libpython3.12-minimal libpython3.12-stdlib libpython3.12t64 libreadline8t64
  libsodium23 libsqlite3-0 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxext6 libxmuu1 media-types
  netbase openssl python3 python3-minimal python3.12 python3.12-minimal readline-common tzdata
  vim-common vim-runtime xauth xxd
Suggested packages:
  liblocale-gettext-perl perl cron quota ecryptfs-utils gpm krb5-doc krb5-user keychain libpam-ssh
  monkeysphere ssh_askpass python3-doc python3-tk python3-venv python3.12-venv python3.12-doc binutils
  bintfmt-support readline-doc ctags vim-doc vim-scripts
The following NEW packages will be installed:
  adduser ca-certificates iputils-ping krb5-locales libbsd0 libcap2-bin libcbor0.10 libedit2 libexpat1
  libfido2-1 libgpm2 libgssapi-krb5-2 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libpam-cap
  libpython3-stdlib libpython3.12-minimal libpython3.12-stdlib libpython3.12t64 libreadline8t64
  libsodium23 libsqlite3-0 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxext6 libxmuu1 media-types
  netbase openssh-client openssl python-is-python3 python3 python3-minimal python3.12
  python3.12-minimal readline-common tzdata vim vim-common vim-runtime xauth xxd
0 upgraded, 47 newly installed, 0 to remove and 1 not upgraded.
Need to get 23.4 MB of archives.
After this operation, 94.0 MB of additional disk space will be used.
```

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.

https://ssh.cloud.google.com/v2/ssh/projects/mykubernetiesproject-456622/zones/us-central1-c/instances/ansible-manager?authuser=0&hl=en_...



SSH-in-browser

UPLOAD FILE

DOWNLOAD FILE



```
root@0dbce4c3184b:/# apt install software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  appstream dbus dbus-bin dbus-daemon dbus-session-bus-common dbus-system-bus-common dbus-user-session
  dirmngr distro-info-data dmsetup gir1.2-girepository-2.0 gir1.2-glib-2.0 gir1.2-packagekitglib-1.0
  gpg gpg-agent gpgconf iso-codes libapparmor1 libappstream5 libargon2-1 libbrotli1 libcryptsetup12
  libcurl3t64-gnutls libdbus-1-3 libdevmapper1.02.1 libduktape207 libdw1t64 libelf1t64 libfdisk1
  libgirepository-1.0-1 libglib2.0-0t64 libglib2.0-bin libglib2.0-data libgstreamer1.0-0 libicu74
  libjson-c5 libkmod2 libksba8 libldap-common libldap2 libnghttp2-14 libnss-systemd
  libpackagekit-glib2-18 libpam-systemd libpolkit-agent-1-0 libpolkit-gobject-1-0 libpsl5t64 librtmp1
  libsas12-2 libsas12-modules libsas12-modules-db libssh-4 libstemmer0d libsystemd-shared libunwind8
  libxml2 libxmlb2 libyaml-0-2 lsb-release networkd-dispatcher packagekit packagekit-tools
  pinentry-curses polkitd publicsuffix python-apt-common python3-apt python3-blinker
  python3-cffi-backend python3-cryptography python3-dbus python3-distro python3-distro-info python3-gi
  python3-httplib2 python3-jwt python3-launchpadlib python3-lazr.restfulclient python3-lazr.uri
  python3-oauthlib python3-pkg-resources python3-pyparsing python3-six python3-software-properties
  python3-wadllib sgml-base shared-mime-info systemd systemd-dev systemd-resolved systemd-sysv
  systemd-timesyncd ucf unattended-upgrades xdg-user-dirs xml-core xz-utils
Suggested packages:
  apt-config-icons gnupg pinentry-gnome3 tor keyboxd sdaemon isoquery low-memory-monitor
  gstreamer1.0-tools libsas12-modules-gssapi-mit | libsas12-modules-gssapi-heimdal
  libsas12-modules-ldap libsas12-modules-otp libsas12-modules-sql iw | wireless-tools pinentry-doc
  polkitd-pkla python-apt-doc python-blinker-doc python-cryptography-doc python3-cryptography-vectors
  python-dbus-doc python3-crypto python3-keyring python3-testresources python3-setuptools
  python-pyparsing-doc sgml-base-doc systemd-container systemd-homed systemd-userdbd systemd-boot
  libip4tc2 libqrencode4 libtss2-esys-3.0.2-0 libtss2-mu-4.0.1-0 libtss2-rc0 libtss2-tcti-device0
```

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*`****

SSH-in-browser

UPLOAD FILE

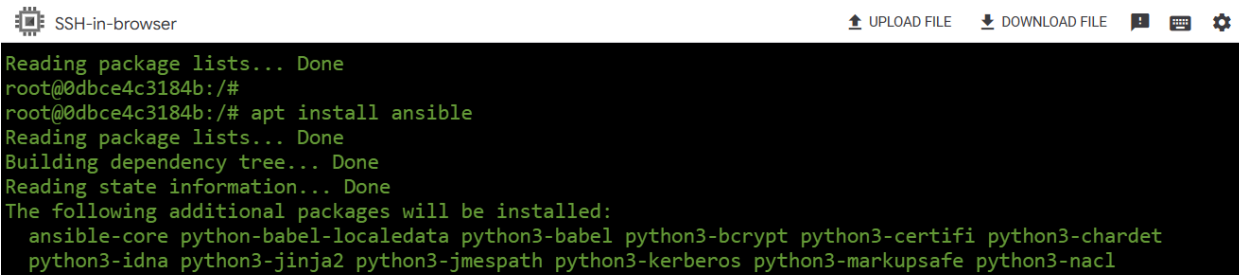
DOWNLOAD FILE

Setting up packagekit-tools (1.2.8-2ubuntu1.2) ...
Setting up software-properties-common (0.99.49.2) ...
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
root@0dbce4c3184b:/#
root@0dbce4c3184b:/# add-apt-repository --yes --update ppa:ansible/ansible
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/
Suites: noble
Components: main
'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to d
eploy. Avoid writing scripts or custom code to deploy and update your applications– automate in a languag
e that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:2 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [10.1 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [65.6 kB]
Hit:4 http://archive.ubuntu.com/ubuntu noble InRelease
Get:5 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [157 B]
Get:6 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [158 B]
Hit:7 http://archive.ubuntu.com/ubuntu noble-updates InRelease

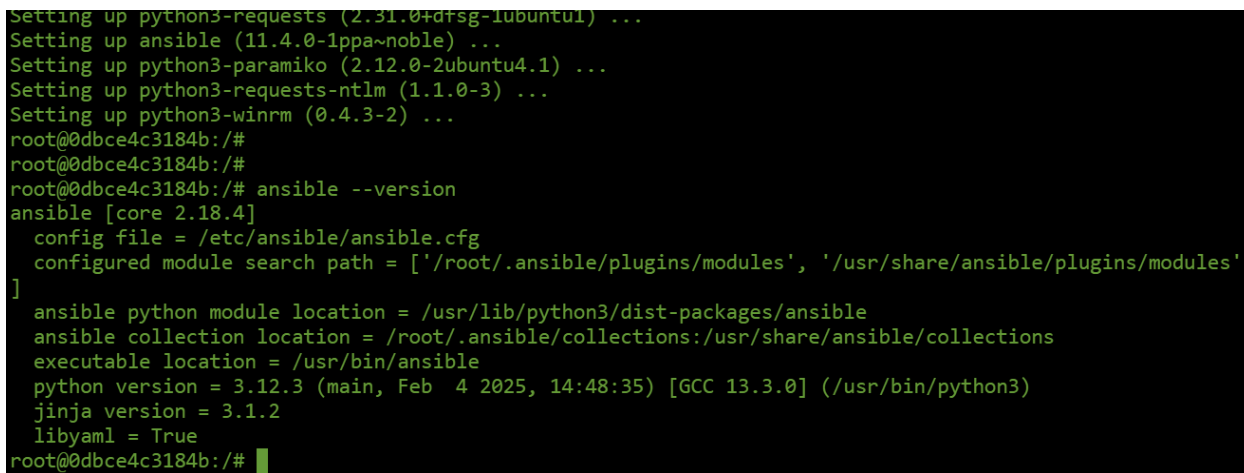
⇒ Install Ansible



SSH-in-browser

```
Reading package lists... Done
root@0dbce4c3184b:/#
root@0dbce4c3184b:/# apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python-babel-localedata python3-babel python3-bcrypt python3-certifi python3-chardet
  python3-idna python3-jinja2 python3-jmespath python3-kerberos python3-markupsafe python3-nacl
```

Check the version ⇒ ansible --version

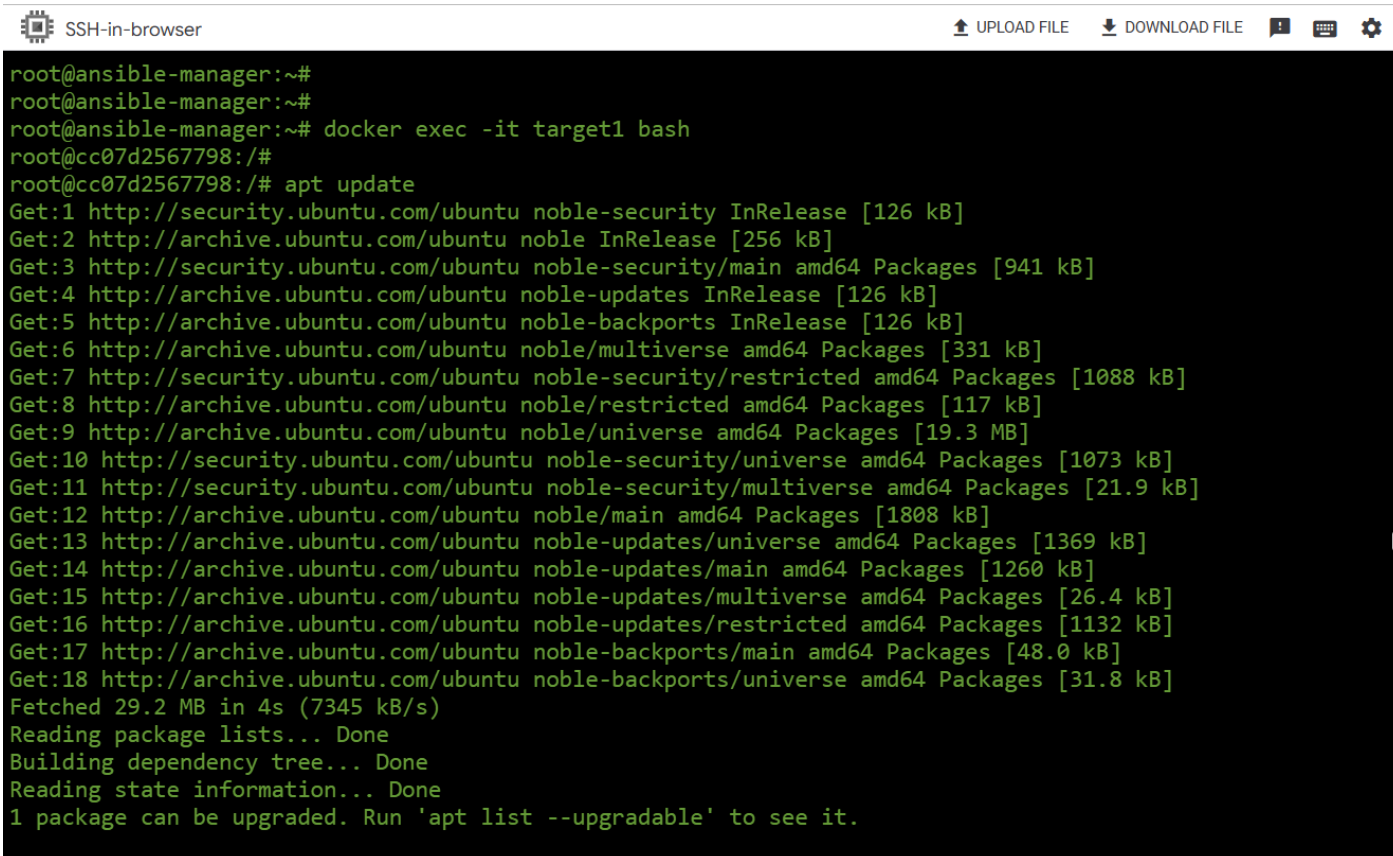


```
Setting up python3-requests (2.31.0+dfsg-1ubuntu1) ...
Setting up ansible (11.4.0-1ppa~noble) ...
Setting up python3-paramiko (2.12.0-2ubuntu4.1) ...
Setting up python3-requests-ntlm (1.1.0-3) ...
Setting up python3-winrm (0.4.3-2) ...
root@0dbce4c3184b:/#
root@0dbce4c3184b:/#
root@0dbce4c3184b:/# ansible --version
ansible [core 2.18.4]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Feb  4 2025, 14:48:35) [GCC 13.3.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
root@0dbce4c3184b:/#
```

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`

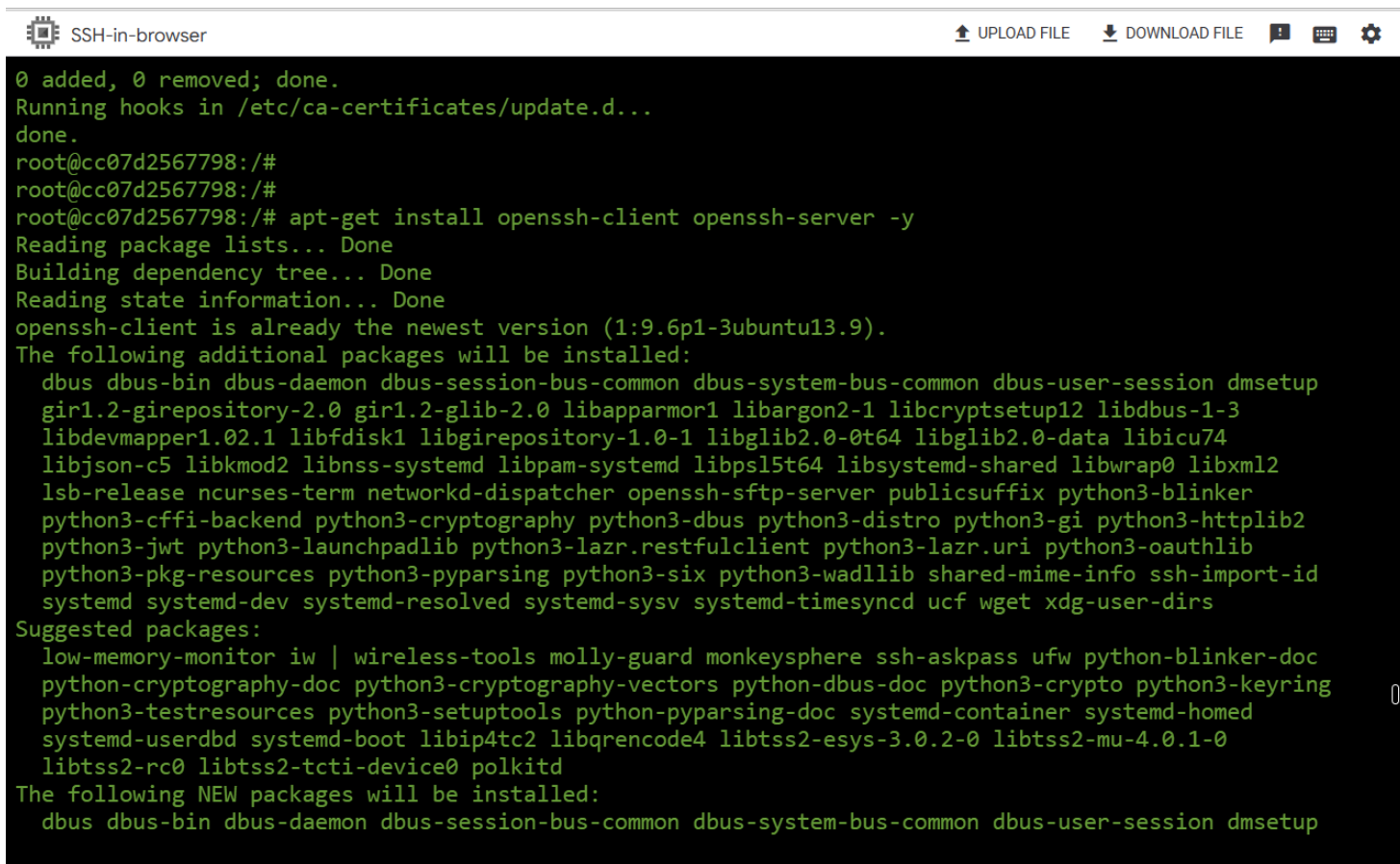


The image shows a terminal window titled "SSH-in-browser" with a toolbar containing "UPLOAD FILE", "DOWNLOAD FILE", and other icons. The terminal output shows a user logging into an "ansible-manager" container and then into a "target1" container. Inside the "target1" container, the user runs "apt update", which fetches package lists from various Ubuntu repositories (security, archive, main, universe, restricted, multiverse) and reports that 1 package can be upgraded.

```
root@ansible-manager:~#
root@ansible-manager:~#
root@ansible-manager:~# docker exec -it target1 bash
root@cc07d2567798:/#
root@cc07d2567798:/# apt update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [941 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1088 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1073 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [21.9 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1369 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1260 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [26.4 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1132 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [48.0 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [31.8 kB]
Fetched 29.2 MB in 4s (7345 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
```


Step 2: Install dependencies as installed for master

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

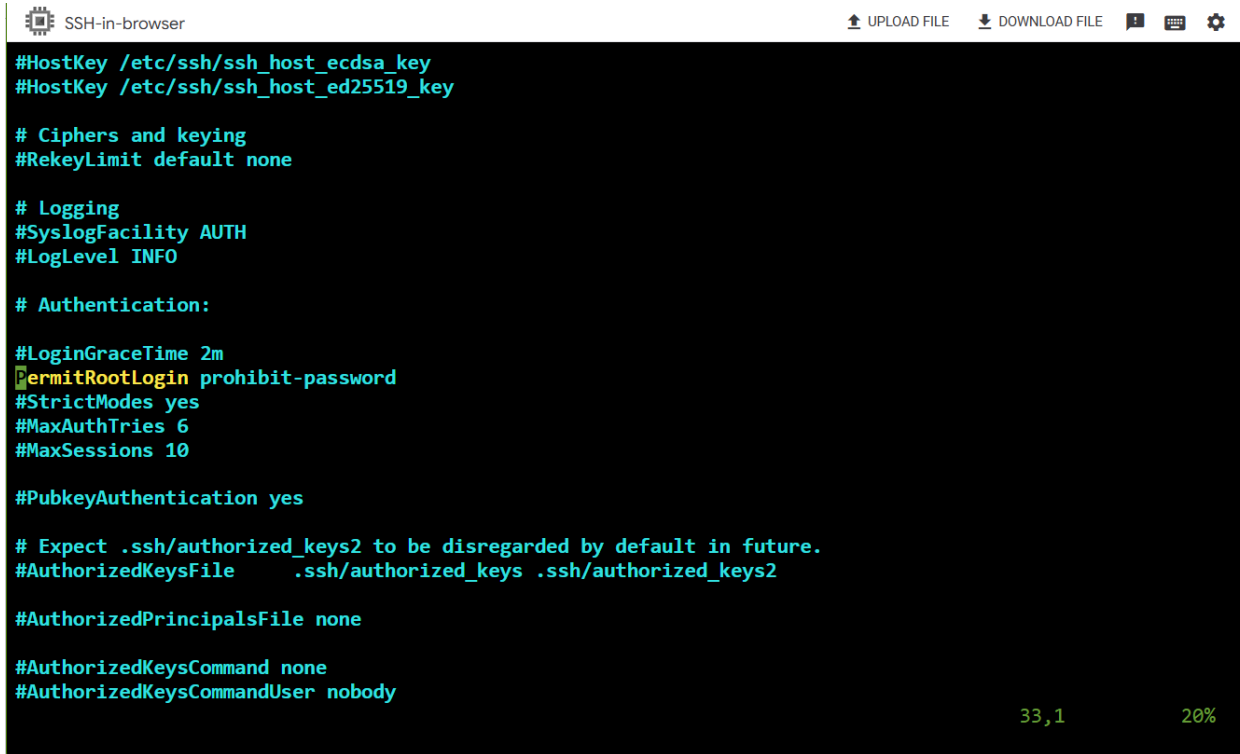


SSH-in-browser

0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@cc07d2567798:/#
root@cc07d2567798:/#
root@cc07d2567798:/# apt-get install openssh-client openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-client is already the newest version (1:9.6p1-3ubuntu13.9).
The following additional packages will be installed:
 dbus dbus-bin dbus-daemon dbus-session-bus-common dbus-system-bus-common dbus-user-session dmsetup
 gir1.2-girepository-2.0 gir1.2-glib-2.0 libapparmor1 libargon2-1 libcryptsetup12 libdbus-1-3
 libdevmapper1.02.1 libfdisk1 libgirepository-1.0-1 libglib2.0-0t64 libglib2.0-data libicu74
 libjson-c5 libkmod2 libnss-systemd libpam-systemd libpsl5t64 libsystemd-shared libwrap0 libxml2
 lsb-release ncurses-term networkd-dispatcher openssh-sftp-server publicsuffix python3-blinker
 python3-cffi-backend python3-cryptography python3-dbus python3-distro python3-gi python3-httplib2
 python3-jwt python3-launchpadlib python3-lazr.restfulclient python3-lazr.uri python3-oauthlib
 python3-pkg-resources python3-pyparsing python3-six python3-wadllib shared-mime-info ssh-import-id
 systemd systemd-dev systemd-resolved systemd-sysv systemd-timesyncd ucf wget xdg-user-dirs
Suggested packages:
 low-memory-monitor iw | wireless-tools molly-guard monkeysphere ssh-askpass ufw python-blinker-doc
 python-cryptography-doc python3-cryptography-vectors python-dbus-doc python3-crypto python3-keyring
 python3-testresources python3-setuptools python-pyparsing-doc systemd-container systemd-homed
 systemd-userdbd systemd-boot libip4tc2 libqrencode4 libtss2-esys-3.0.2-0 libtss2-mu-4.0.1-0
 libtss2-rc0 libtss2-tcti-device0 polkitd
The following NEW packages will be installed:
 dbus dbus-bin dbus-daemon dbus-session-bus-common dbus-system-bus-common dbus-user-session dmsetup

Step 3: Go to the ssh and edit the file

- `Cd /etc/ssh`
- `Vi sshd_config`



```
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
```

33,1 20%

⇒ From Prohibited to change it to Yes

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

⇒ 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

```
    "Networks": {
      "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "MacAddress": "02:42:ac:11:00:03",
        "NetworkID": "578594f0acd0c4254b683d0c8bb44466114dc70f25fcae25e1f7d5cc73038555",
        "EndpointID": "398eae99f9864ca42b5f69dd6047338dbaca1eece6d3294b1baedace8164023e",
        "Gateway": "172.17.0.1",
        "IPAddress": "172.17.0.3",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "DriverOpts": null,
        "DNSNames": null
      }
    }
  }
}
root@ansible-manager:~#
```

⇒ GO to the Ansible Master and edit the hosts file

Add the IP address to the Host file – 172.17.0.3

```
root@ansible-manager:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
3fc45f516187   ubuntu   "/bin/bash"             About an hour ago   Up About an hour               target2
cc07d2567798   ubuntu   "/bin/bash"             About an hour ago   Up About an hour               target1
0dbce4c3184b   ubuntu   "/bin/bash"             About an hour ago   Up About an hour               ansible-master
root@ansible-manager:~# docker exec -it ansible-master bash
root@0dbce4c3184b:/# cd /etc/ansible/
root@0dbce4c3184b:/etc/ansible#
root@0dbce4c3184b:/etc/ansible#
root@0dbce4c3184b:/etc/ansible# ls
ansible.cfg  hosts  roles
root@0dbce4c3184b:/etc/ansible#
root@0dbce4c3184b:/etc/ansible#
root@0dbce4c3184b:/etc/ansible# vi hosts
root@0dbce4c3184b:/etc/ansible#
```

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:yIgyacoEvKnByxHo3iyjxmDojn8urjGI7DLWDjDpWI 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]

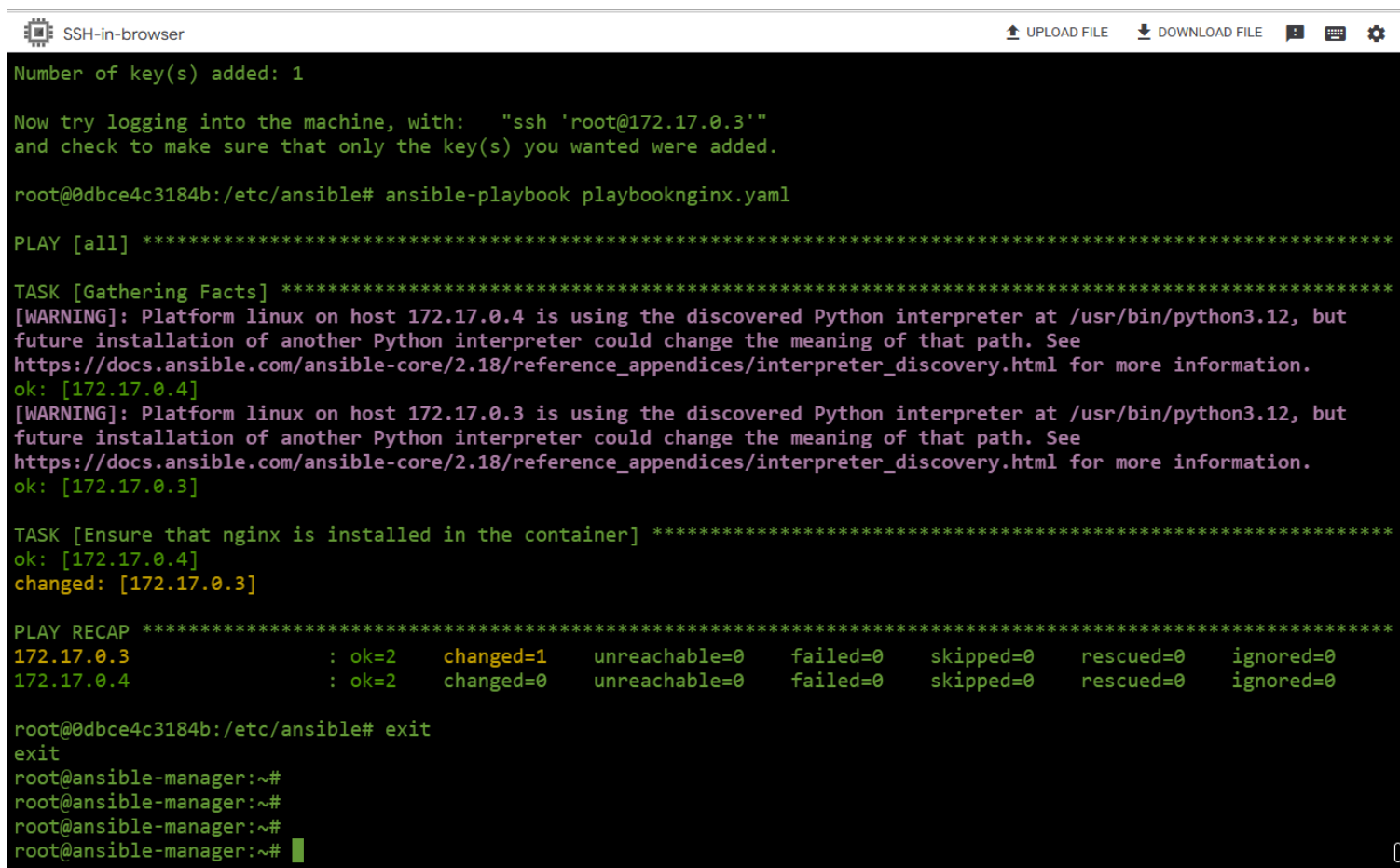
TASK [Ensure that nginx is installed in the container] *****
changed: [172.17.0.3]

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.



The screenshot shows a terminal window titled "SSH-in-browser" with a dark background and light green text. The terminal output shows the execution of an Ansible playbook named "playbooknginx.yaml". The output includes a warning about the Python interpreter path on hosts 172.17.0.3 and 172.17.0.4, and a task to ensure nginx is installed in the container. The task is successful on both hosts. The terminal also shows a "PLAY RECAP" section with a table of results for both hosts.

```
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# ansible-playbook playbooknginx.yaml

PLAY [all] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.17.0.4 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: [172.17.0.4]
[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] *****
ok: [172.17.0.4]
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
172.17.0.4      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@0dbce4c3184b:/etc/ansible# exit
exit
root@ansible-manager:~#
root@ansible-manager:~#
root@ansible-manager:~#
root@ansible-manager:~#
```

Submitted: for Approval.

Engineer - Parvez M B

Verification: