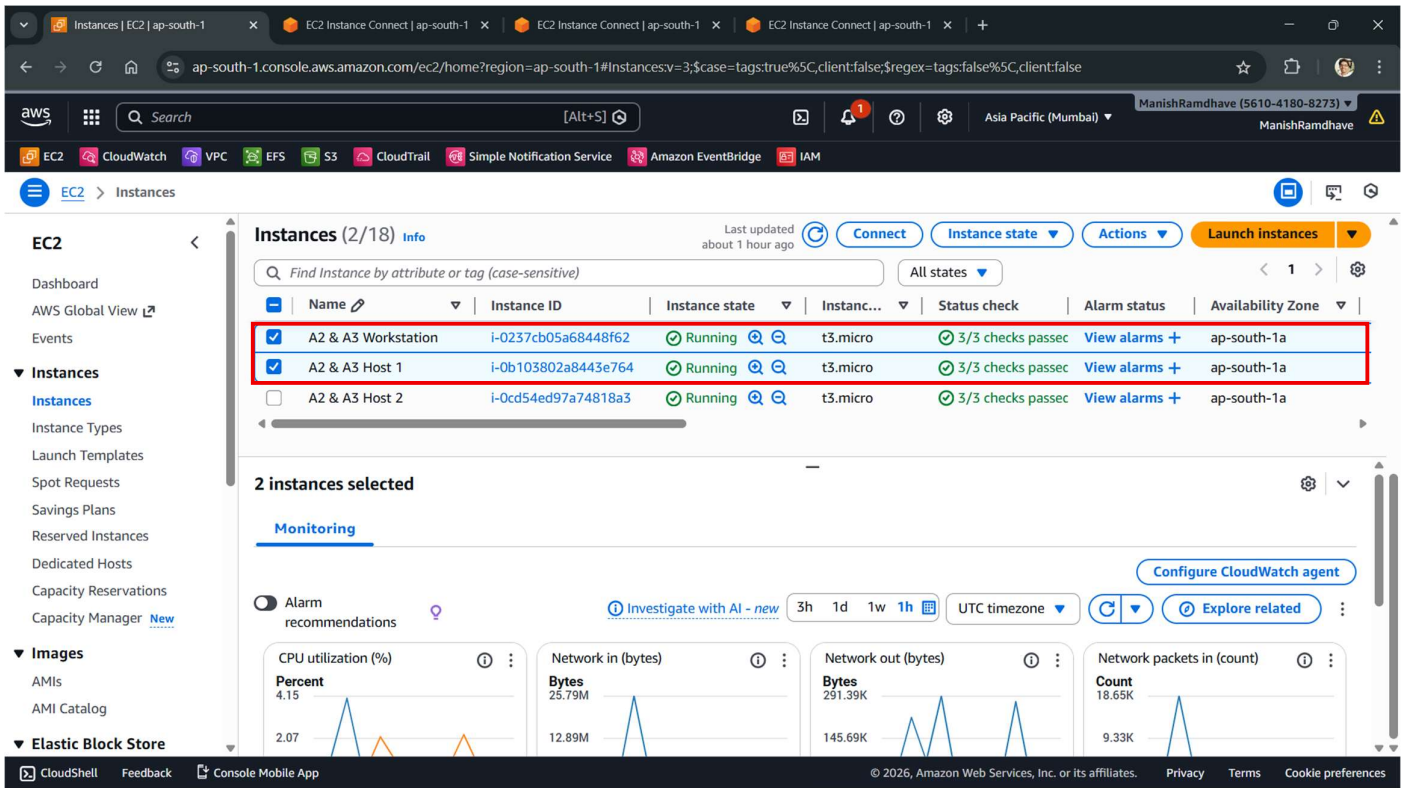
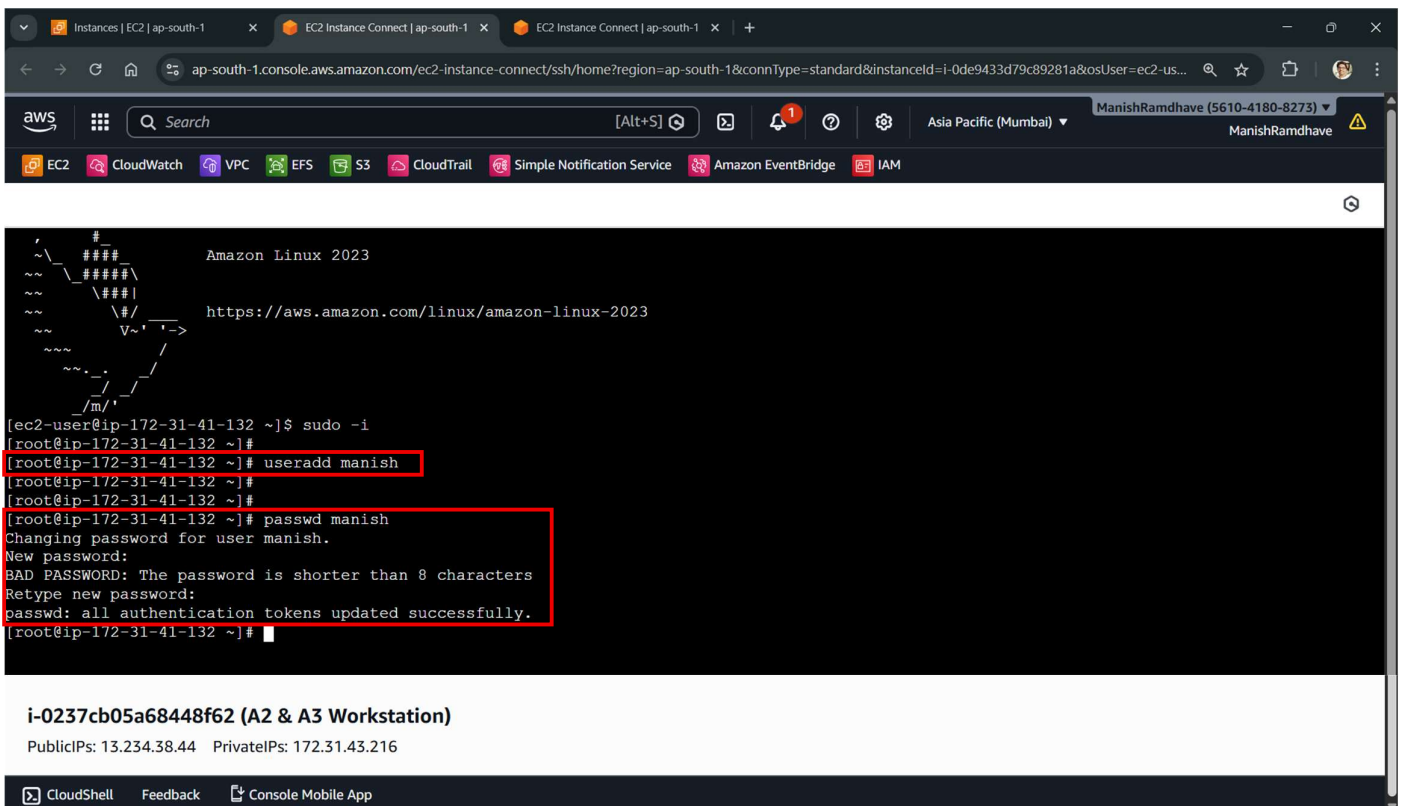


Ansible Assignment 2 (Playbook1)

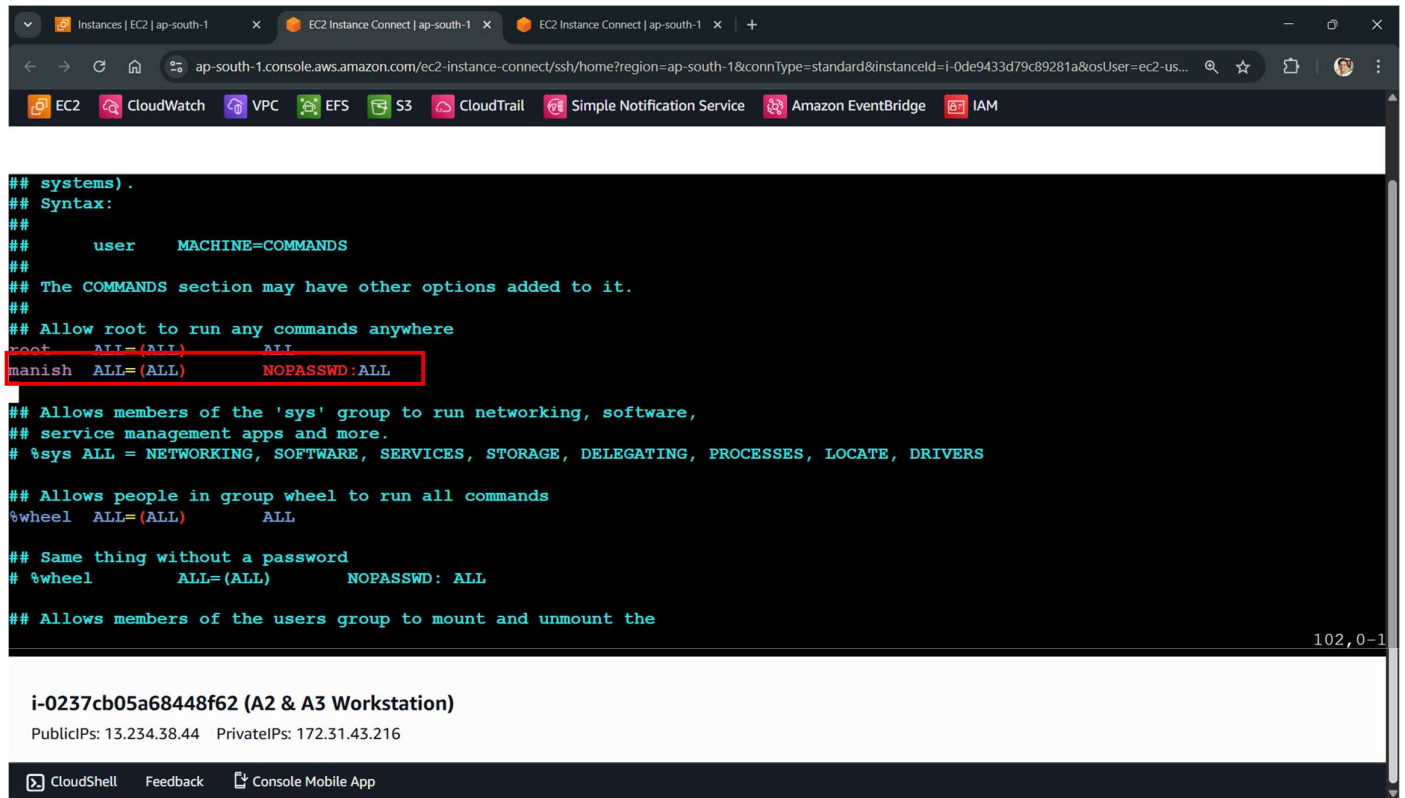
Step 1: Launched two EC2 Instances for our **Ansible Workstation** and **Host1**:



Step 2: Created an user having ‘manish’ and also applied a password to it on both ‘Workstation’ as well as ‘Host1’:



Step 3: Updated the 'sudoers' file in /etc for assigning **superuser privileges** to 'manish' user:



The screenshot shows the AWS Management Console interface with a CloudShell terminal open. The terminal displays the content of the /etc/sudoers file. The line `manish ALL=(ALL) NOPASSWD:ALL` is highlighted with a red box. The terminal output includes comments about the syntax and the COMMANDS section, as well as entries for root, sys, wheel, and users groups.

```
## systems).
## Syntax:
##
##     user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
manish  ALL=(ALL)    NOPASSWD:ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)    ALL

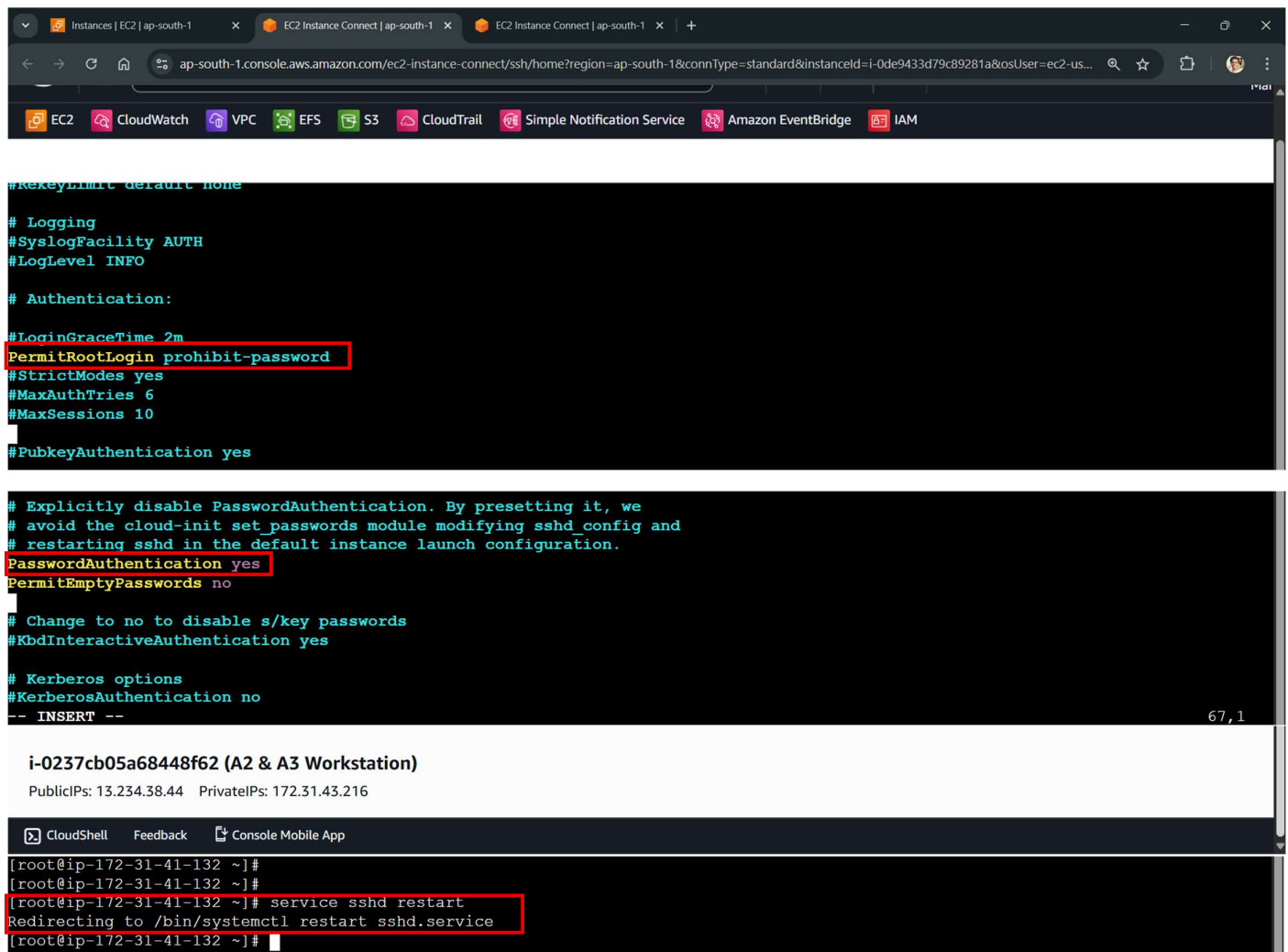
## Same thing without a password
# %wheel    ALL=(ALL)    NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
```

i-0237cb05a68448f62 (A2 & A3 Workstation)
PublicIPs: 13.234.38.44 PrivateIPs: 172.31.43.216

CloudShell Feedback Console Mobile App

Step 4: In /etc/ssh/sshd_config file, made following changes and **restarted the 'sshd service'**:



The screenshot shows the AWS Management Console interface with a CloudShell terminal open. The terminal displays the content of the /etc/ssh/sshd_config file. The lines `PermitRootLogin prohibit-password` and `PasswordAuthentication yes` are highlighted with red boxes. Below the file content, terminal commands to restart the sshd service are shown, with the command `service sshd restart` also highlighted with a red box.

```
#X11UseArbitraryLocales yes
#X11Forwarding no
#XAuthLocation /usr/libexec/openssh/Xauthd

#KeychainAuthentication yes
#KeychainPath /usr/bin/keychain

#PermitTunnel no
#TunnelDevice ipnet10

#Logging
#SyslogFacility AUTH
#LogLevel INFO

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

#PubkeyAuthentication yes

# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords no

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

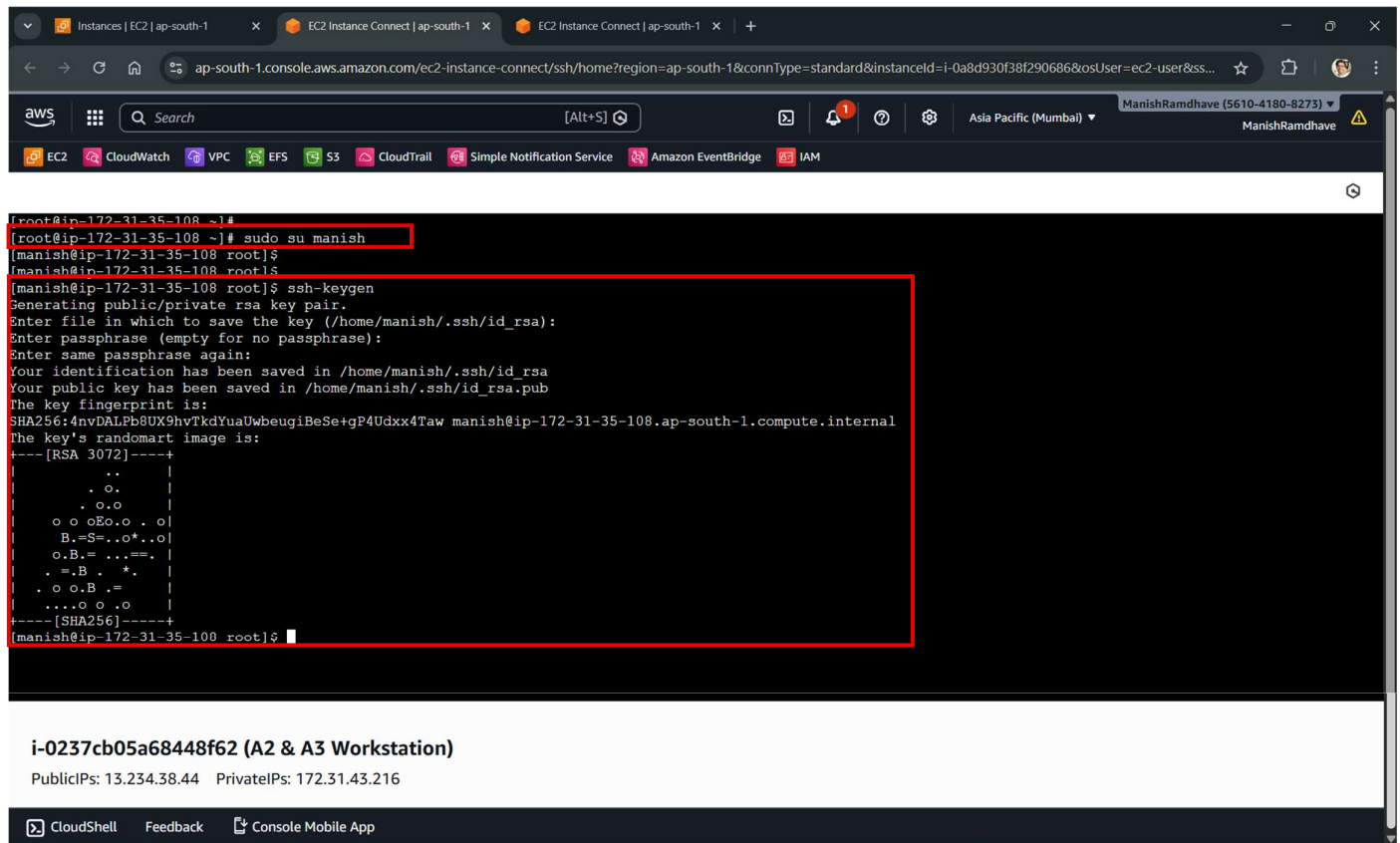
# Kerberos options
#KerberosAuthentication no
-- INSERT --
```

i-0237cb05a68448f62 (A2 & A3 Workstation)
PublicIPs: 13.234.38.44 PrivateIPs: 172.31.43.216

CloudShell Feedback Console Mobile App

```
[root@ip-172-31-41-132 ~]#
[root@ip-172-31-41-132 ~]#
[root@ip-172-31-41-132 ~]# service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[root@ip-172-31-41-132 ~]#
```

Step 5: Logged in to the ‘manish’ user and generated SSH private and public keypairs:



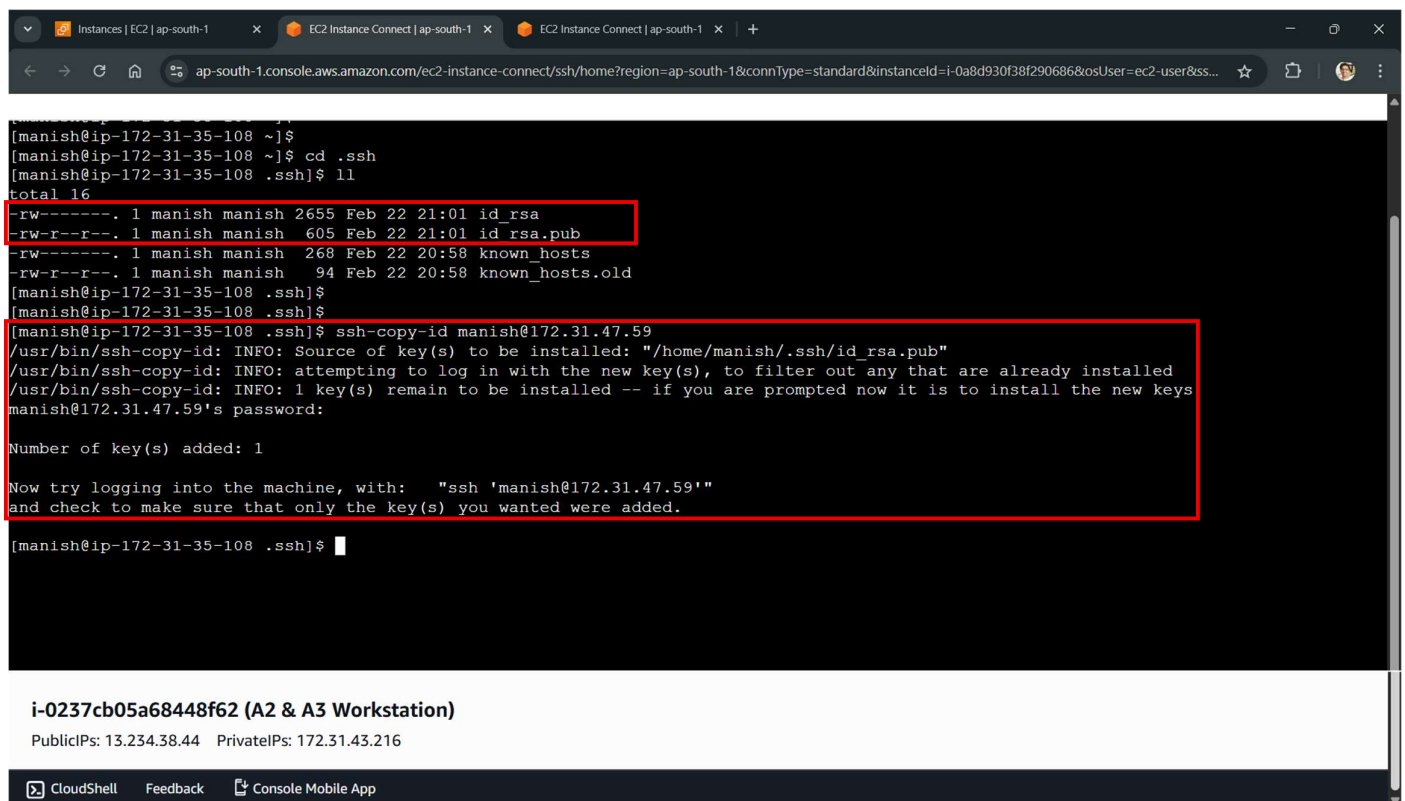
```
[root@ip-172-31-35-108 ~]#
[root@ip-172-31-35-108 ~]# sudo su manish
[manish@ip-172-31-35-108 root]$
[manish@ip-172-31-35-108 root]$
[manish@ip-172-31-35-108 root]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/manish/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/manish/.ssh/id_rsa
Your public key has been saved in /home/manish/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:4nvDALPb8UX9hvTkdYuaUwbeugiBeSe+gP4Udxx4Taw manish@ip-172-31-35-108.ap-south-1.compute.internal
The key's randomart image is:
+---[RSA 3072]-----+
|
|..
|.O.
|.O.O
|o o oEo.o o|
|B.=..o*..o|
|o.B.=...==|
|..B..*|
|.o o.B.=|
|....o o.o|
+---[SHA256]-----+
[manish@ip-172-31-35-108 root]$
```

i-0237cb05a68448f62 (A2 & A3 Workstation)

PublicIPs: 13.234.38.44 PrivateIPs: 172.31.43.216

CloudShell Feedback Console Mobile App

Step 6: Generated both Private (id_rsa) and Public (id_rsa.pub) key pairs and copied them to the host instance using its private IP:



```
[manish@ip-172-31-35-108 ~]$
[manish@ip-172-31-35-108 ~]$ cd .ssh
[manish@ip-172-31-35-108 .ssh]$ ll
total 16
-rw----- 1 manish manish 2655 Feb 22 21:01 id_rsa
-rw-r--r-- 1 manish manish 605 Feb 22 21:01 id_rsa.pub
-rw----- 1 manish manish 268 Feb 22 20:58 known_hosts
-rw-r--r-- 1 manish manish 94 Feb 22 20:58 known_hosts.old
[manish@ip-172-31-35-108 .ssh]$
[manish@ip-172-31-35-108 .ssh]$ ssh-copy-id manish@172.31.47.59
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/manish/.ssh/id_rsa.pub"
/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
manish@172.31.47.59's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'manish@172.31.47.59'"
and check to make sure that only the key(s) you wanted were added.

[manish@ip-172-31-35-108 .ssh]$
```

i-0237cb05a68448f62 (A2 & A3 Workstation)

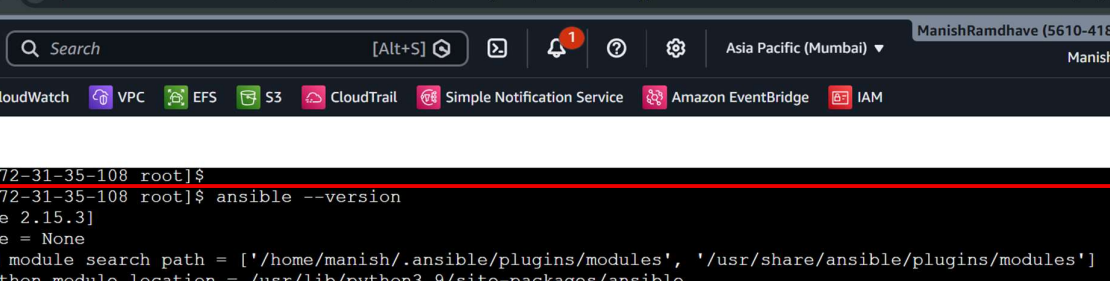
PublicIPs: 13.234.38.44 PrivateIPs: 172.31.43.216

CloudShell Feedback Console Mobile App

Step 7: Successfully logged in to the **Host1** from **Workstation** using SSH and Private IP of Host:

[illegible]

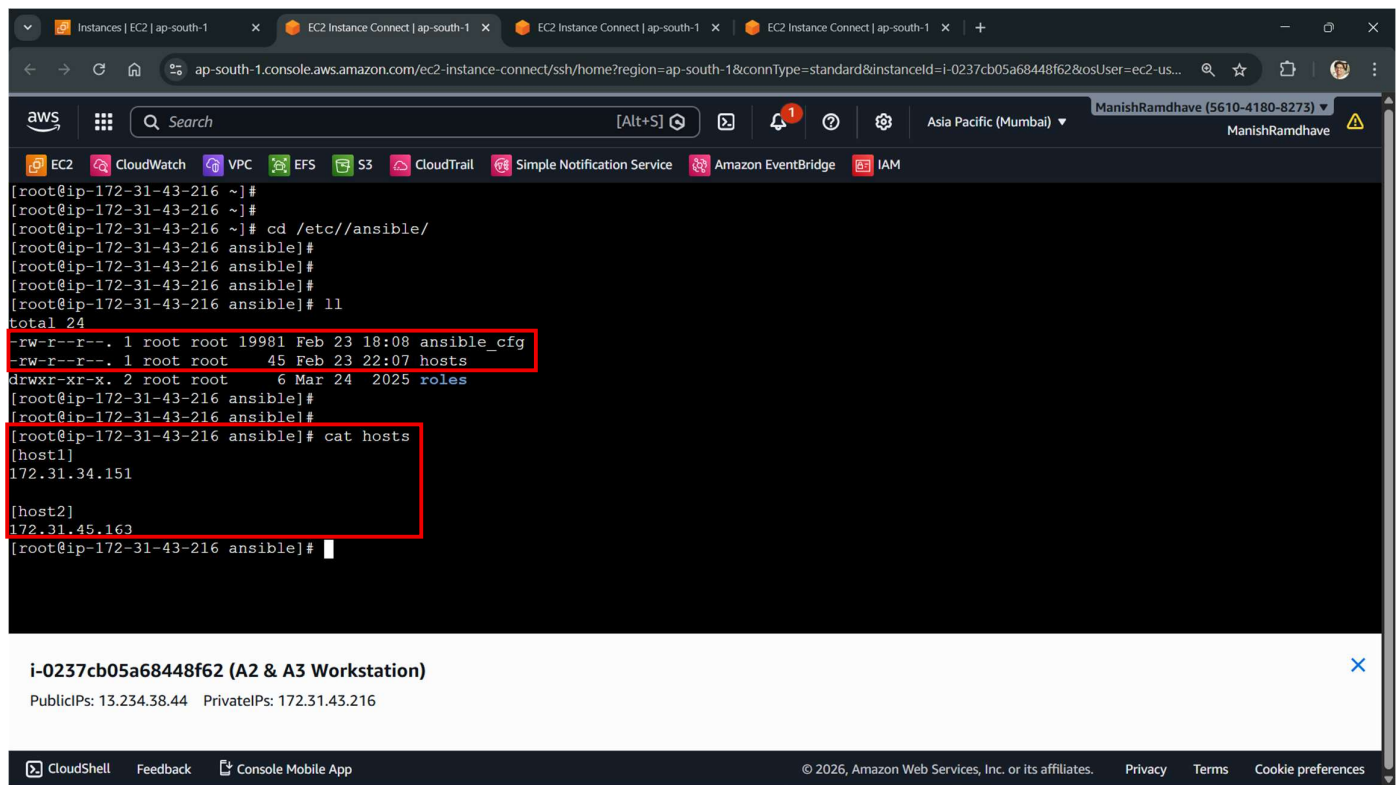
Step 8: Installed **Ansible** on the **Workstation**:



The screenshot shows the AWS Management Console interface. At the top, there are browser tabs for 'Instances | EC2 | ap-south-1', 'EC2 Instance Connect | ap-south-1', and 'EC2 Instance Connect | ap-south-1'. The address bar shows the URL 'ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-0a8d930f38f290686&osUser=ec2-use...'. The console header includes the AWS logo, a search bar, and navigation links for EC2, CloudWatch, VPC, EFS, S3, CloudTrail, Simple Notification Service, Amazon EventBridge, and IAM. The user's name 'ManishRamdhare' and location 'Asia Pacific (Mumbai)' are displayed. The main content area shows a terminal window with the following output:

```
[manish@ip-172-31-35-108 root]$  
[manish@ip-172-31-35-108 root]$ ansible --version  
ansible [core 2.15.3]  
  config file = None  
  configured module search path = ['/home/manish/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']  
  ansible python module location = /usr/lib/python3.9/site-packages/ansible  
  ansible collection location = /home/manish/.ansible/collections:/usr/share/ansible/collections  
  executable location = /usr/bin/ansible  
  python version = 3.9.25 (main, Dec 10 2025, 00:00:00) [GCC 11.5.0 20240719 (Red Hat 11.5.0-5)] (/usr/bin/python3.9)  
  jinja version = 3.1.4  
  libyaml = True  
[manish@ip-172-31-35-108 root]$
```


Step 9: Created two files in `/etc/ansible/` path, `'hosts'` and `'ansible.cfg'`. The `'hosts'` consists of the host instances' private IPs:

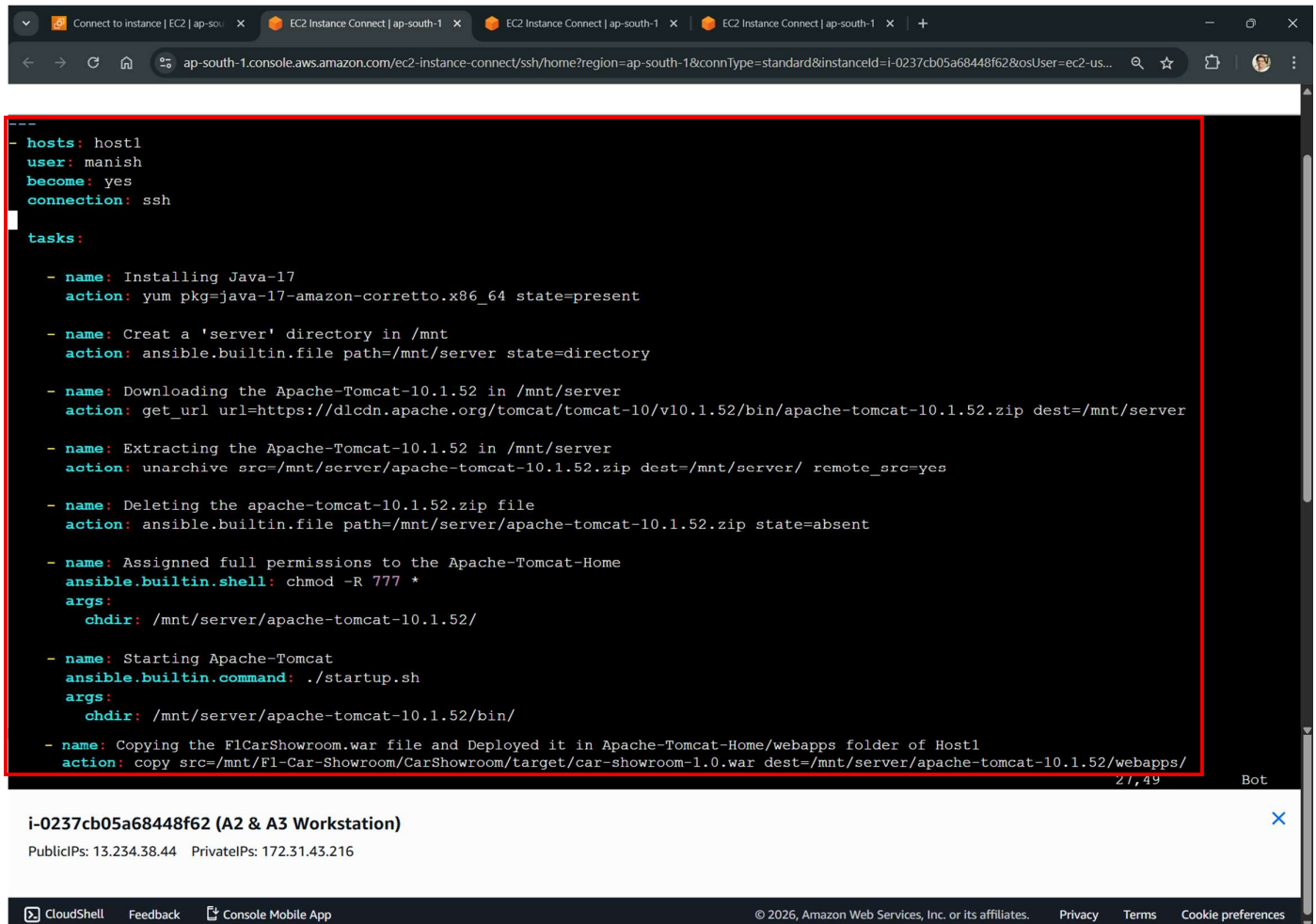


The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output shows the user navigating to `/etc/ansible/` and creating two files: `ansible.cfg` and `hosts`. The `hosts` file contains two host entries: `host1` with IP `172.31.34.151` and `host2` with IP `172.31.45.163`. Below the terminal, a metadata box for instance `i-0237cb05a68448f62` is visible, showing public and private IP addresses.

```
[root@ip-172-31-43-216 ~]#  
[root@ip-172-31-43-216 ~]#  
[root@ip-172-31-43-216 ~]# cd /etc//ansible/  
[root@ip-172-31-43-216 ansible]#  
[root@ip-172-31-43-216 ansible]#  
[root@ip-172-31-43-216 ansible]#  
[root@ip-172-31-43-216 ansible]# ll  
total 24  
-rw-r--r--. 1 root root 19981 Feb 23 18:08 ansible_cfg  
-rw-r--r--. 1 root root 45 Feb 23 22:07 hosts  
drwxr-xr-x. 2 root root 6 Mar 24 2025 roles  
[root@ip-172-31-43-216 ansible]#  
[root@ip-172-31-43-216 ansible]# cat hosts  
[host1]  
172.31.34.151  
[host2]  
172.31.45.163  
[root@ip-172-31-43-216 ansible]#
```

i-0237cb05a68448f62 (A2 & A3 Workstation)
PublicIPs: 13.234.38.44 PrivateIPs: 172.31.43.216

Step 10: Created a **Playbook1** using `'adhoc command with modules'` and named it as a `'host1.yaml'`:

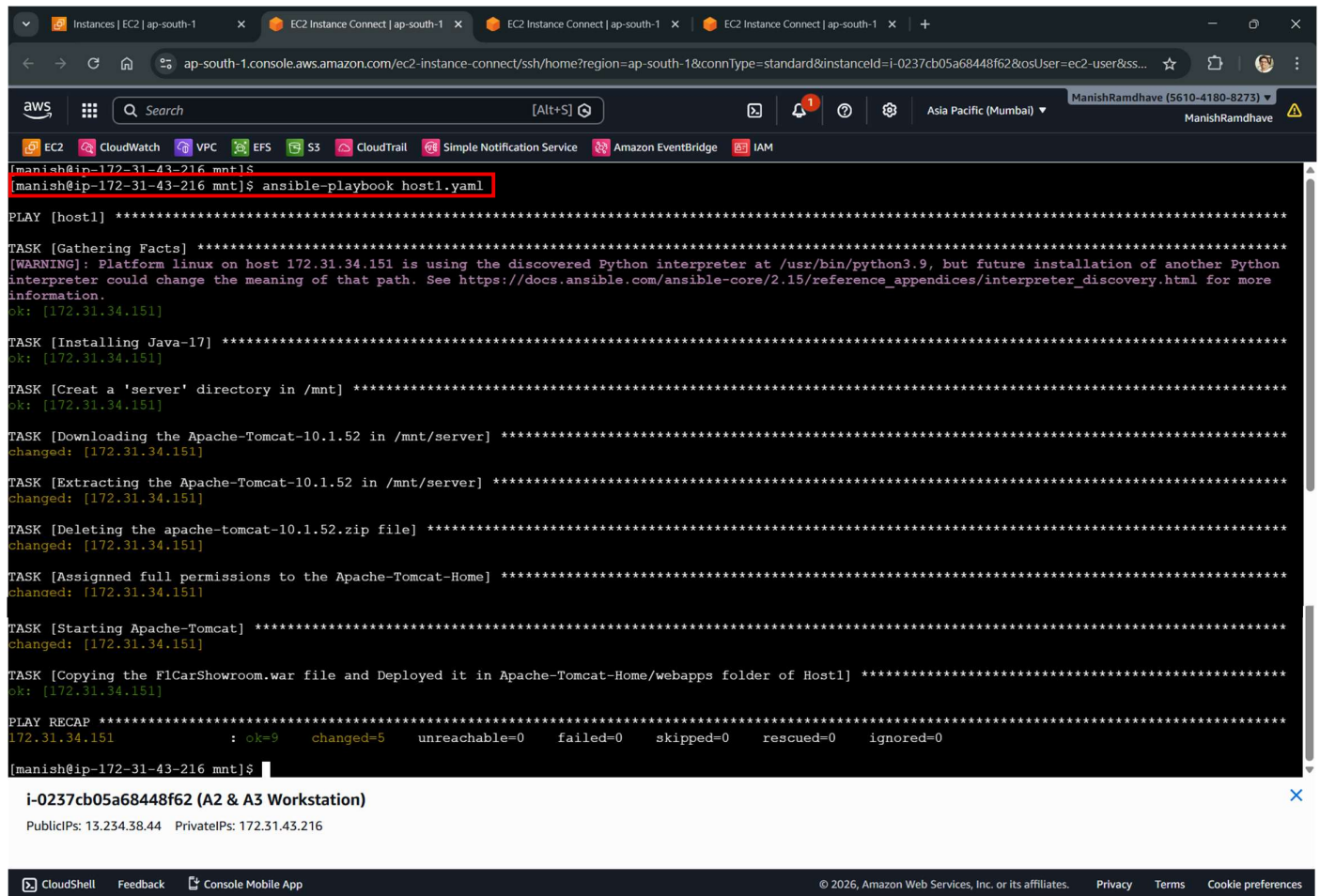


The screenshot shows the AWS CloudShell interface with a terminal window displaying the content of a playbook named `host1.yaml`. The playbook includes a `hosts` section with `host1`, a `user` of `manish`, `become: yes`, and `connection: ssh`. The `tasks` section contains several steps: installing Java-17, creating a `/mnt/server` directory, downloading Apache-Tomcat-10.1.52, extracting it, deleting the zip file, assigning permissions, starting Apache-Tomcat, and copying a `FlCarShowroom.war` file to the `webapps` folder.

```
---  
- hosts: host1  
  user: manish  
  become: yes  
  connection: ssh  
  
tasks:  
  
  - name: Installing Java-17  
    action: yum pkg=java-17-amazon-corretto.x86_64 state=present  
  
  - name: Create a 'server' directory in /mnt  
    action: ansible.builtin.file path=/mnt/server state=directory  
  
  - name: Downloading the Apache-Tomcat-10.1.52 in /mnt/server  
    action: get_url url=https://d1cdn.apache.org/tomcat/tomcat-10/v10.1.52/bin/apache-tomcat-10.1.52.zip dest=/mnt/server  
  
  - name: Extracting the Apache-Tomcat-10.1.52 in /mnt/server  
    action: unarchive src=/mnt/server/apache-tomcat-10.1.52.zip dest=/mnt/server/ remote_src=yes  
  
  - name: Deleting the apache-tomcat-10.1.52.zip file  
    action: ansible.builtin.file path=/mnt/server/apache-tomcat-10.1.52.zip state=absent  
  
  - name: Assigned full permissions to the Apache-Tomcat-Home  
    ansible.builtin.shell: chmod -R 777 *  
    args:  
      chdir: /mnt/server/apache-tomcat-10.1.52/  
  
  - name: Starting Apache-Tomcat  
    ansible.builtin.command: ./startup.sh  
    args:  
      chdir: /mnt/server/apache-tomcat-10.1.52/bin/  
  
  - name: Copying the FlCarShowroom.war file and Deployed it in Apache-Tomcat-Home/webapps folder of Host1  
    action: copy src=/mnt/Fl-Car-Showroom/CarShowroom/target/car-showroom-1.0.war dest=/mnt/server/apache-tomcat-10.1.52/webapps/
```

i-0237cb05a68448f62 (A2 & A3 Workstation)
PublicIPs: 13.234.38.44 PrivateIPs: 172.31.43.216

Step 11: Executed the **Playbook1 Script using ansible command** and all these commands are executed on **'Host1 Instance'**:



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal displays the execution of the command `ansible-playbook host1.yaml` on the host `172.31.34.151`. The output shows the playbook running successfully with various tasks completed, including gathering facts, installing Java, creating a directory, downloading and extracting Apache Tomcat, deleting a zip file, assigning permissions, starting Tomcat, and copying a WAR file. A recap at the bottom shows 9 OK, 5 changed, and 0 failures.

```
[manish@ip-172-31-43-216 mnt]$ ansible-playbook host1.yaml

PLAY [host1] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.34.151 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.34.151]

TASK [Installing Java-17] *****
ok: [172.31.34.151]

TASK [Creat a 'server' directory in /mnt] *****
ok: [172.31.34.151]

TASK [Downloading the Apache-Tomcat-10.1.52 in /mnt/server] *****
changed: [172.31.34.151]

TASK [Extracting the Apache-Tomcat-10.1.52 in /mnt/server] *****
changed: [172.31.34.151]

TASK [Deleting the apache-tomcat-10.1.52.zip file] *****
changed: [172.31.34.151]

TASK [Assigned full permissions to the Apache-Tomcat-Home] *****
changed: [172.31.34.151]

TASK [Starting Apache-Tomcat] *****
changed: [172.31.34.151]

TASK [Copying the F1CarShowroom.war file and Deployed it in Apache-Tomcat-Home/webapps folder of Host1] *****
ok: [172.31.34.151]

PLAY RECAP *****
172.31.34.151      : ok=9    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[manish@ip-172-31-43-216 mnt]$
```

i-0237cb05a68448f62 (A2 & A3 Workstation)
PublicIPs: 13.234.38.44 PrivateIPs: 172.31.43.216

Results:

We have successfully deployed the ‘car-showroom-1.0’ application using ansible ‘adhoc command with modules’ and hosted the same application using Host1 Public IP and Tomcat Server Port No.8080:

