

Ansible Assignment 2 (Playbook1)

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

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, 'A2 & A3 Workstation' and 'A2 & A3 Host 1' are selected. In the main table, these two instances are highlighted with a red border. Both instances are listed as 'Running' with an 't3.micro' instance type. The status check for both is '3/3 checks passed'. The 'Actions' dropdown menu for each instance includes options like 'Stop', 'Start', 'Reboot', and 'Delete'. Below the table, a message says '2 instances selected'. Under the 'Monitoring' tab, there are four line charts: 'CPU utilization (%)', 'Network in (bytes)', 'Network out (bytes)', and 'Network packets in (count)'. The 'Configure CloudWatch agent' button is visible at the top right of the monitoring section.

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

The screenshot shows the AWS EC2 Instance Connect terminal. The user is root on an Amazon Linux 2023 instance. The terminal history shows the following commands:

```
[ec2-user@ip-172-31-41-132 ~]$ sudo -i
[root@ip-172-31-41-132 ~]# useradd manish
[root@ip-172-31-41-132 ~]# passwd manish
Changing password for user manish.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-41-132 ~]#
```

A red box highlights the password creation command and its output. The terminal prompt '[root@ip-172-31-41-132 ~]#' is also highlighted.

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

```
## 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 umount the
102,0-1
```

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’:

```
#RekeyLimit default none

# 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 --
```

67, 1

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:

The screenshot shows a terminal session in the AWS CloudShell. The user has switched to the 'manish' user via 'sudo su manish'. They then run 'ssh-keygen' to generate an RSA key pair. The output shows the key fingerprint and the private key content.

```
[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:
SHA256:4nDALPb8UX9hvTkdyYuaUwbeugiBeSe+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.=S=...o*.o|
| o.B.=...==. |
| . =B . * . |
| . o o.B .= |
| ....o o .o |
+--- [SHA256] ---+
[manish@ip-172-31-35-108 root]$
```

i-0237cb05a68448f62 (A2 & A3 Workstation)
Public IPs: 13.234.38.44 Private IPs: 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:

The screenshot shows the user navigating to the '.ssh' directory and listing files. They then run 'ssh-copy-id' to copy the public key to a host machine at 172.31.47.59. The command prompts for the host's password and adds the key.

```
[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-r----- 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-r----- 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]$
[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.
```

i-0237cb05a68448f62 (A2 & A3 Workstation)
Public IPs: 13.234.38.44 Private IPs: 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:

The screenshot shows a terminal window in the AWS CloudShell interface. The user is connected via SSH to an Amazon Linux 2023 instance (IP: 172.31.43.216). The session starts with root privileges, then switches to the 'manish' user. A red box highlights the output of the 'ssh manish@172.31.34.151' command, which shows the Amazon Linux logo and a link to the distribution's website. The terminal also displays the last login information.

```
[root@ip-172-31-43-216 mnt]#
[root@ip-172-31-43-216 mnt]#
[root@ip-172-31-43-216 mnt]# sudo su manish
[manish@ip-172-31-43-216 mnt]$
[manish@ip-172-31-43-216 mnt]$
[manish@ip-172-31-43-216 mnt]$ ssh manish@172.31.34.151
,      #
~\_\_ #####          Amazon Linux 2023
~~ \_\_\#\#\#\\
~~ \_\_\#\#\#
~~ \_\#\/_\_
~~ V~ ' *-> https://aws.amazon.com/linux/amazon-linux-2023
~~
~~ .-.
~~ /_/
~~ /m,,
Last login: Mon Feb 23 21:09:14 2026 from 172.31.43.216
[manish@ip-172-31-34-151 ~]$ 
```

i-0237cb05a68448f62 (A2 & A3 Workstation)

Public IPs: 13.234.38.44 Private IPs: 172.31.43.216

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 8: Installed Ansible on the Workstation:

The screenshot shows a terminal window in the AWS CloudShell interface. The user is connected via SSH to an Amazon Linux 2023 instance (IP: 172.31.35.108). The session starts with root privileges. A red box highlights the output of the 'ansible --version' command, which displays the installed version of Ansible (2.15.3) along with details about the configured module search path, Python module location, collection location, executable location, Python version, Jinja version, and libyaml status.

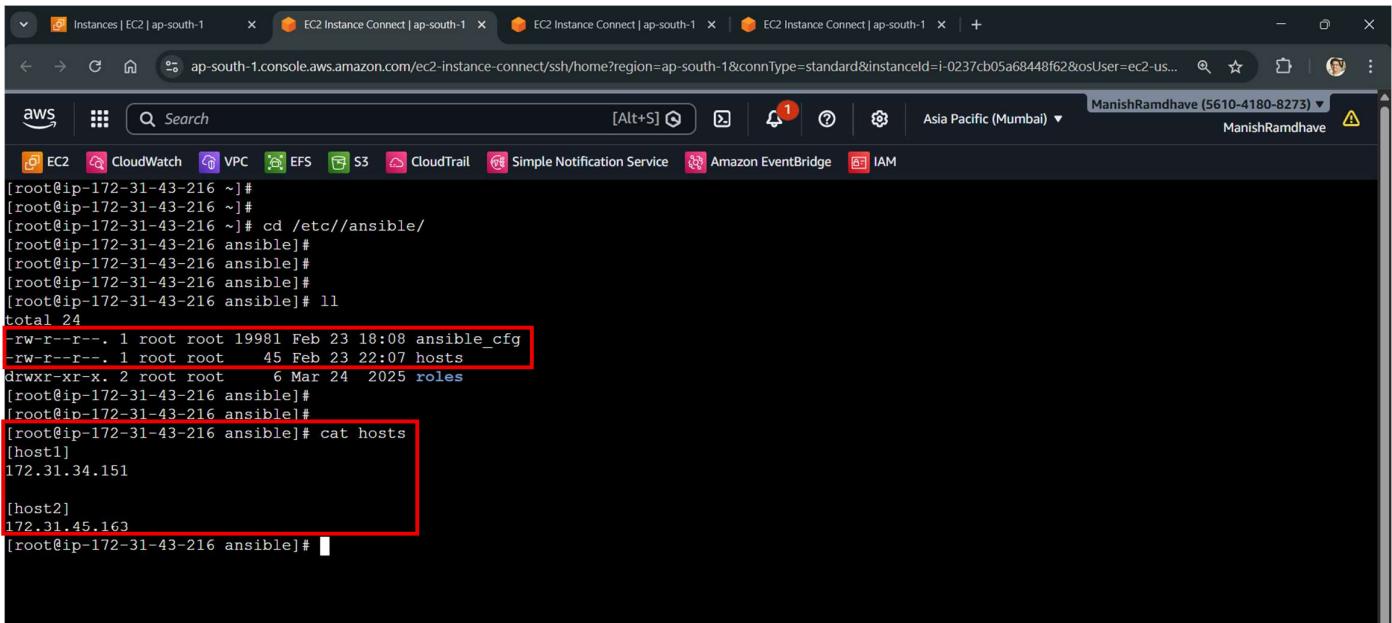
```
[manish@ip-172-31-35-108 root]$
[manish@ip-172-31-35-108 root]$ ansible --version
ansible [core 2.15.3]
  config file = None
  configured module search path = ['/home/manish/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  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]$ 
```

i-0237cb05a68448f62 (A2 & A3 Workstation)

Public IPs: 13.234.38.44 Private IPs: 172.31.43.216

CloudShell Feedback Console Mobile App

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



```
[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]#
[root@ip-172-31-43-216 ansible]# cat hosts
[host1]
172.31.34.151

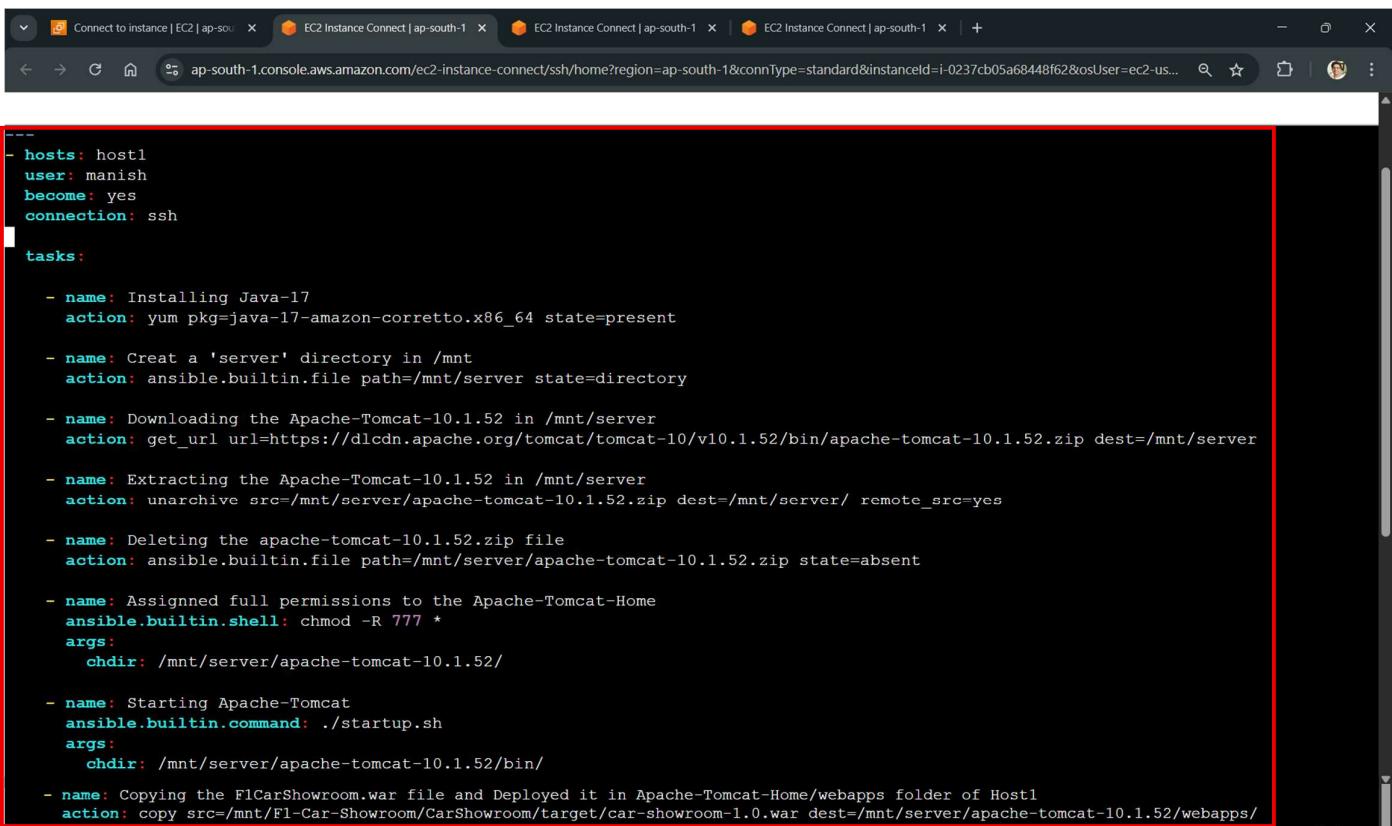
[host2]
172.31.45.163
[root@ip-172-31-43-216 ansible]#
```

i-0237cb05a68448f62 (A2 & A3 Workstation)

PublicIPs: 13.234.38.44 PrivatelPs: 172.31.43.216

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 10: Created a Playbook1 using ‘adhoc command with modules’ and named it as a ‘host1.yaml’:



```
---
- 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://dlcdn.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: Assignned 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/

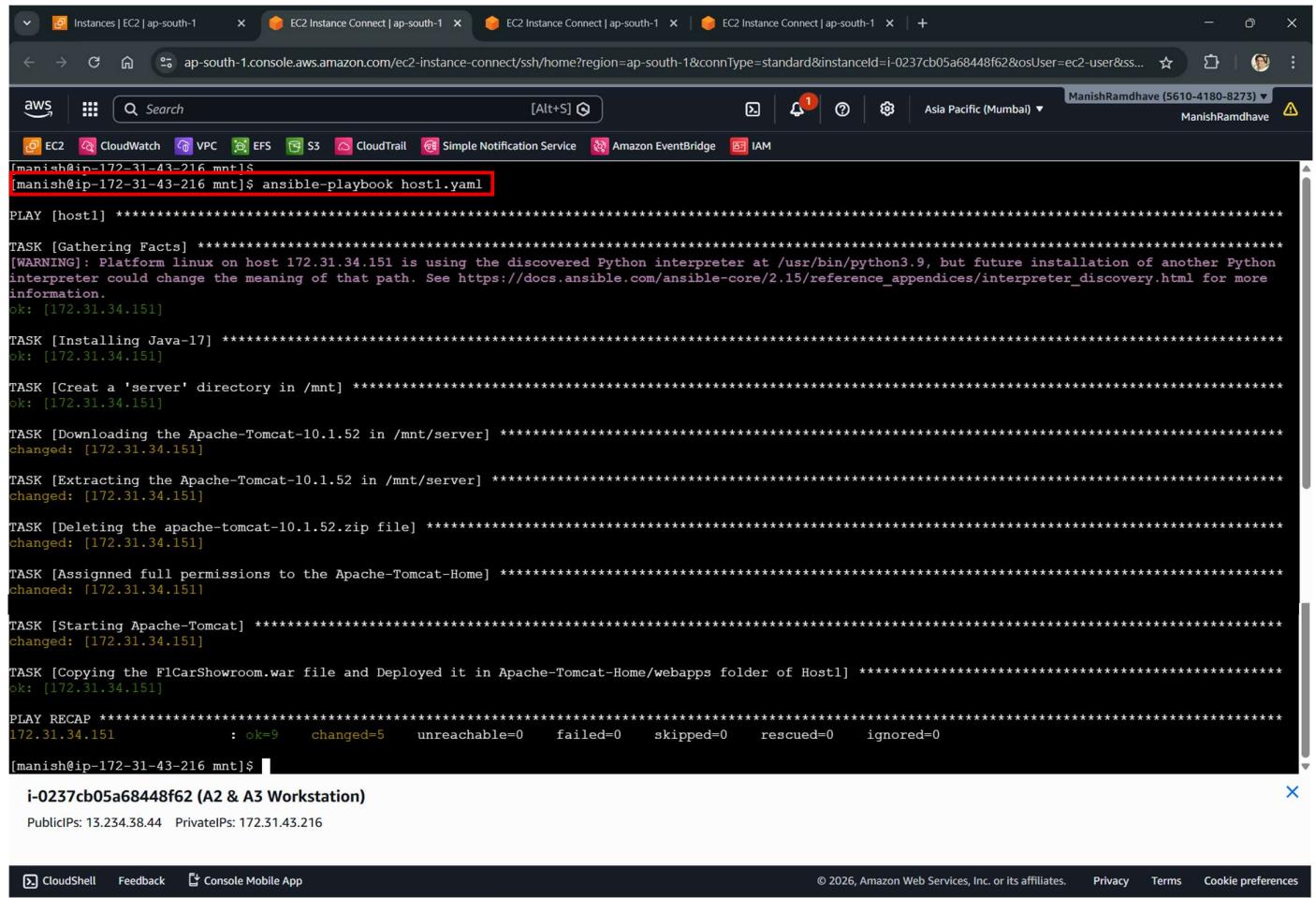
    - name: Copying the F1CarShowroom.war file and Deployed it in Apache-Tomcat-Home/webapps folder of Host1
      action: copy src=/mnt/F1-Car>Showroom/CarShowroom/target/car-showroom-1.0.war dest=/mnt/server/apache-tomcat-10.1.52/webapps/
```

i-0237cb05a68448f62 (A2 & A3 Workstation)

PublicIPs: 13.234.38.44 PrivatelPs: 172.31.43.216

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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



```
[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 [Create 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:

