Lesson-End Project

Installing Apache Tomcat Using Ansible Playbook

Project agenda: To install Apache Tomcat using an Ansible playbook for automated deployment, configuration, and management of the application server

Description: As a DevOps engineer at FutureTech Solutions, you are tasked with automating the deployment of Apache Tomcat on multiple Ubuntu servers. Your objectives include creating an Ansible playbook for consistent installation and configuration, implementing Ansible Vault to manage sensitive data securely, and developing a dynamic inventory system to adapt to server changes, ensuring scalability and reducing manual intervention.

Tools required: Ansible

Prerequisites: You must have Ansible installed in the lab to proceed.

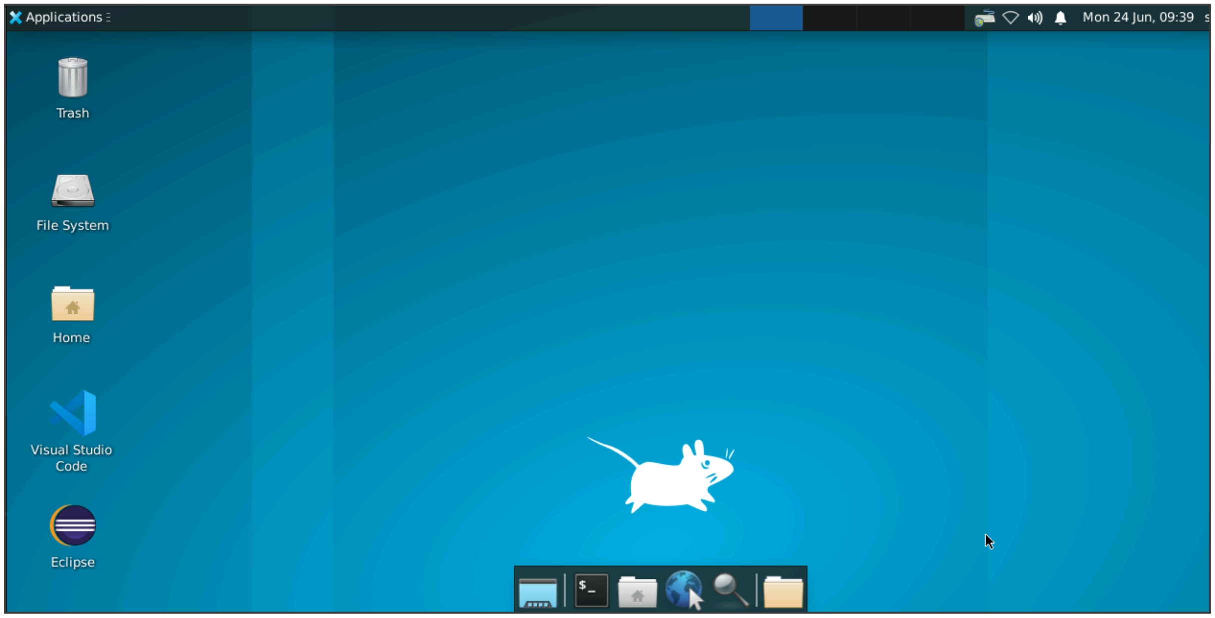
Expected deliverables: A step-by-step guide to install Tomcat; begin by writing a playbook that outlines all the necessary tasks for downloading, configuring, and starting the Tomcat service.

Steps to be followed:

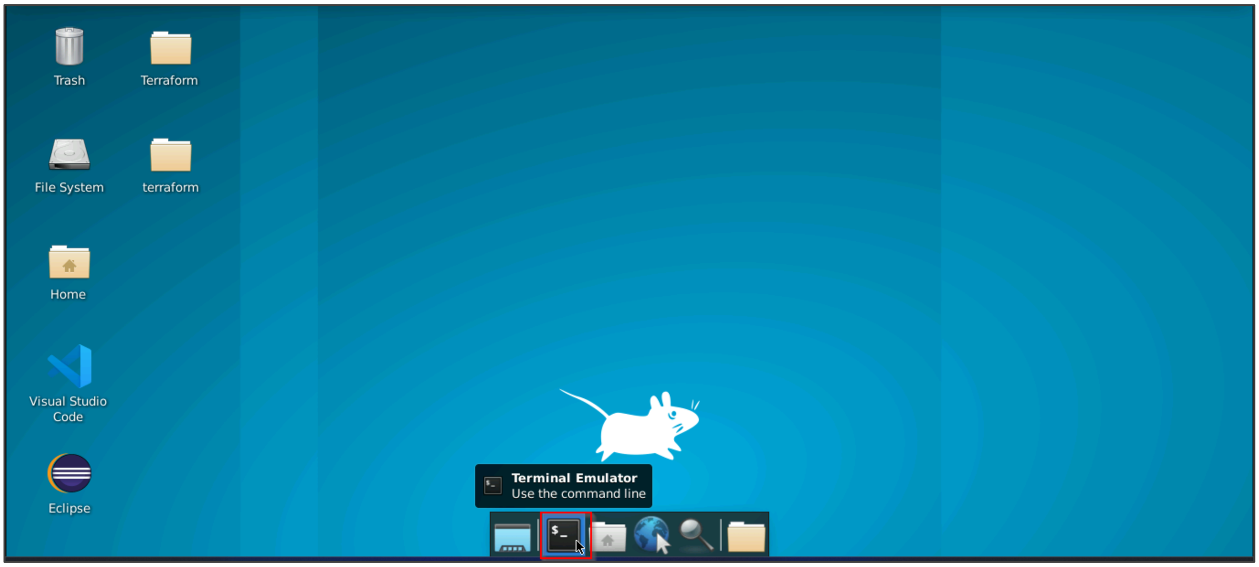
1. Run Ansible and update packages
2. Configure Ansible
3. Establish connectivity between the Ansible controller and the node machine
4. Create an Apache playbook
5. Execute the playbook
6. Confirm the installation

Step 1: Run Ansible and update packages

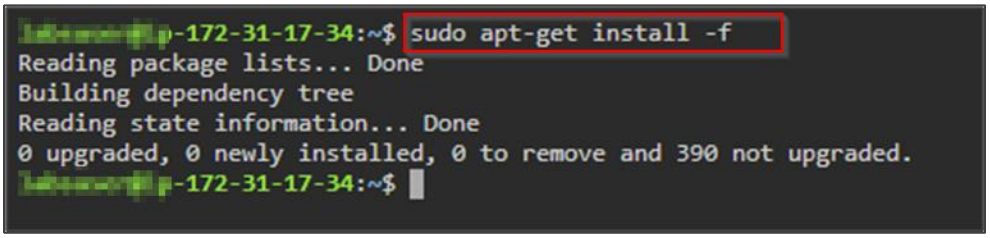
1.1 Open the DevOps lab:



1.2 Open the terminal:

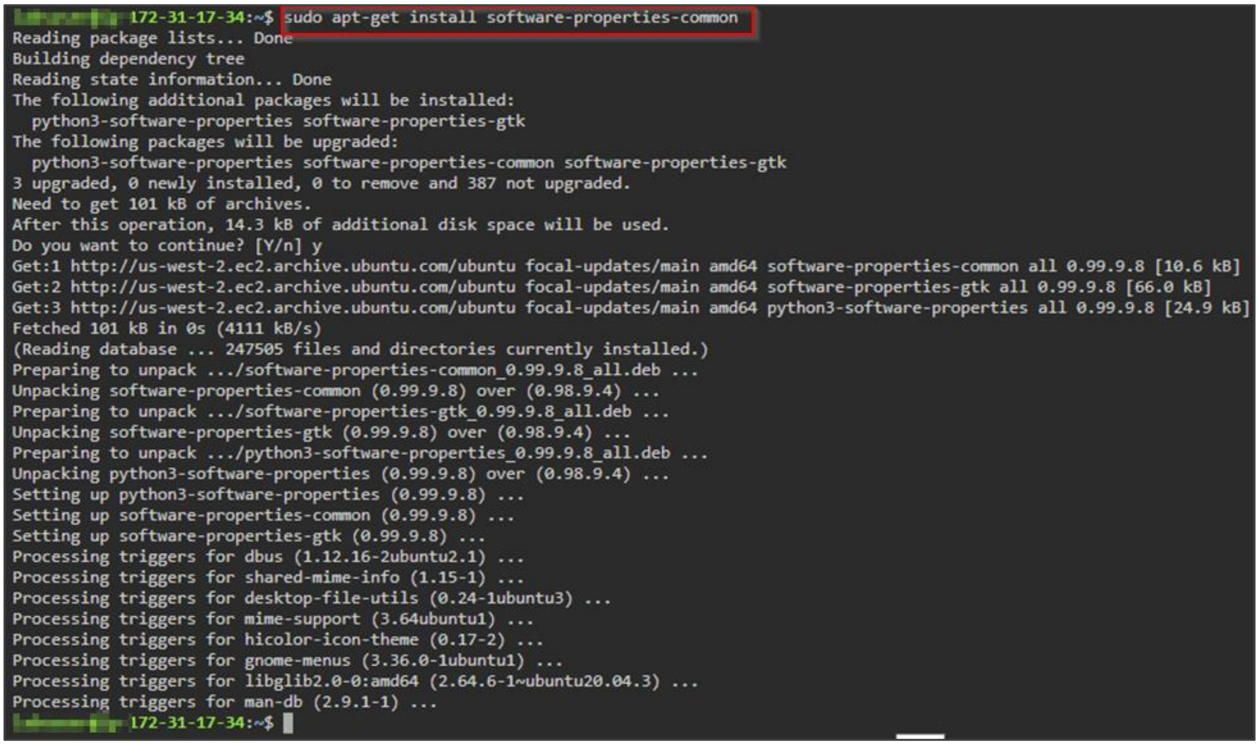


1.3 Run the below command to fix and install any broken dependencies:
sudo apt-get install -f



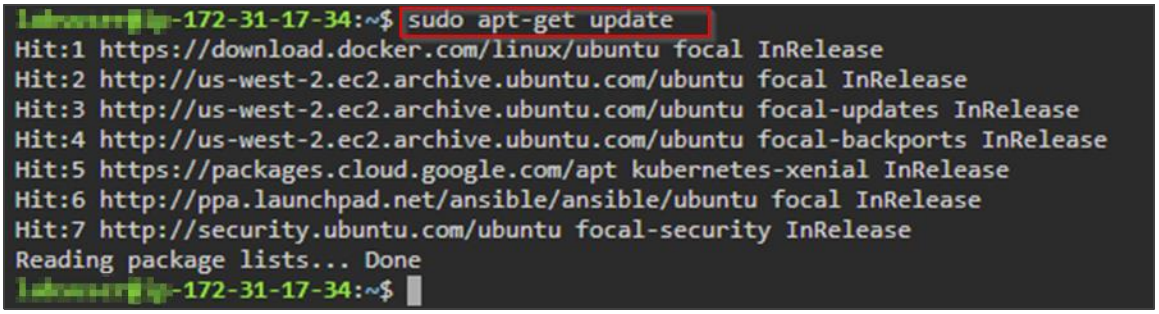
```
[redacted]-172-31-17-34:~$ sudo apt-get install -f
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 390 not upgraded.
[redacted]-172-31-17-34:~$
```

1.4 Enter the command given below to update package repositories and get the latest package information:
sudo apt-get install software-properties-common



```
[redacted] 172-31-17-34:~$ sudo apt-get install software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties software-properties-gtk
The following packages will be upgraded:
  python3-software-properties software-properties-common software-properties-gtk
3 upgraded, 0 newly installed, 0 to remove and 387 not upgraded.
Need to get 101 kB of archives.
After this operation, 14.3 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 software-properties-common all 0.99.9.8 [10.6 kB]
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 software-properties-gtk all 0.99.9.8 [66.0 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-software-properties all 0.99.9.8 [24.9 kB]
Fetched 101 kB in 0s (4111 kB/s)
(Reading database ... 247505 files and directories currently installed.)
Preparing to unpack .../software-properties-common_0.99.9.8_all.deb ...
Unpacking software-properties-common (0.99.9.8) over (0.98.9.4) ...
Preparing to unpack .../software-properties-gtk_0.99.9.8_all.deb ...
Unpacking software-properties-gtk (0.99.9.8) over (0.98.9.4) ...
Preparing to unpack .../python3-software-properties_0.99.9.8_all.deb ...
Unpacking python3-software-properties (0.99.9.8) over (0.98.9.4) ...
Setting up python3-software-properties (0.99.9.8) ...
Setting up software-properties-common (0.99.9.8) ...
Setting up software-properties-gtk (0.99.9.8) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for shared-mime-info (1.15-1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libglib2.0-0:amd64 (2.64.6-1~ubuntu20.04.3) ...
Processing triggers for man-db (2.9.1-1) ...
[redacted] 172-31-17-34:~$
```

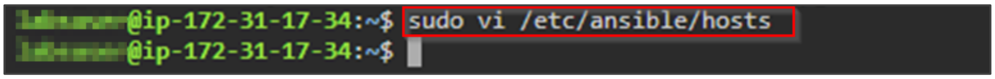
1.5 Run the following command to update the package lists from the repositories:
sudo apt-get update



```
[redacted]-172-31-17-34:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Hit:6 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease
Hit:7 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
[redacted]-172-31-17-34:~$
```

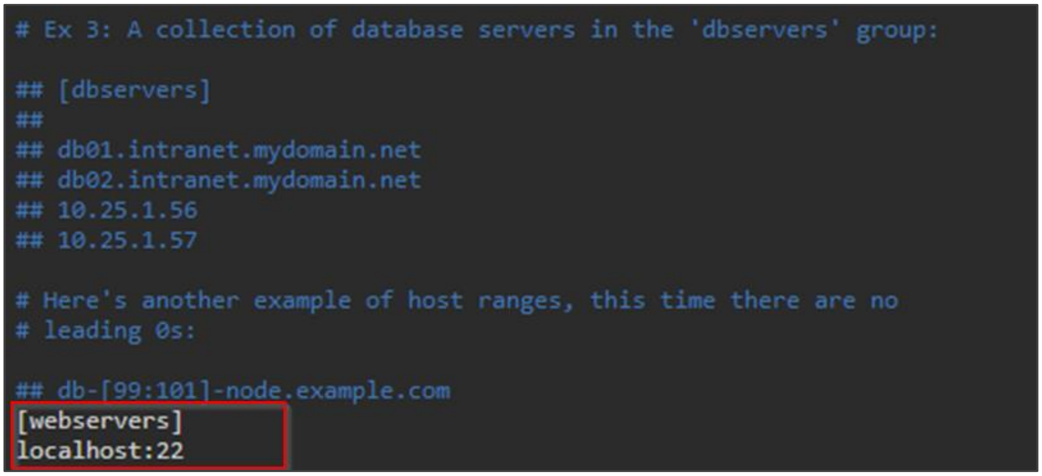
Step 2: Configure Ansible

2.1 Enter the given command to add the host localhost in the Ansible host file:
sudo vi /etc/ansible/hosts



```
[redacted]@ip-172-31-17-34:~$ sudo vi /etc/ansible/hosts
[redacted]@ip-172-31-17-34:~$
```

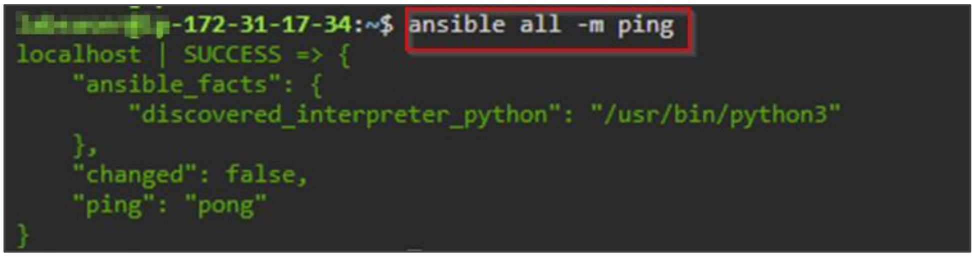
2.2 Define a host group named **[webservers]** that includes a single host accessible via localhost on port **22**
[webservers]
localhost:22



```
# Ex 3: A collection of database servers in the 'dbservers' group:

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com
[webservers]
localhost:22
```

Step 3: Establish connectivity between the Ansible controller and the node machine

3.1 Execute the command given below to validate the host inventory file by attempting to ping all hosts listed in the inventory:
ansible all -m ping



```
[redacted]p-172-31-17-34:~$ ansible all -m ping
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

The command executes successfully, confirming that Ansible can reach and communicate with the hosts.

Step 4: Create an Apache playbook

4.1 Run the following command to create and open a YAML file:
sudo nano tomcat-playbook.yml

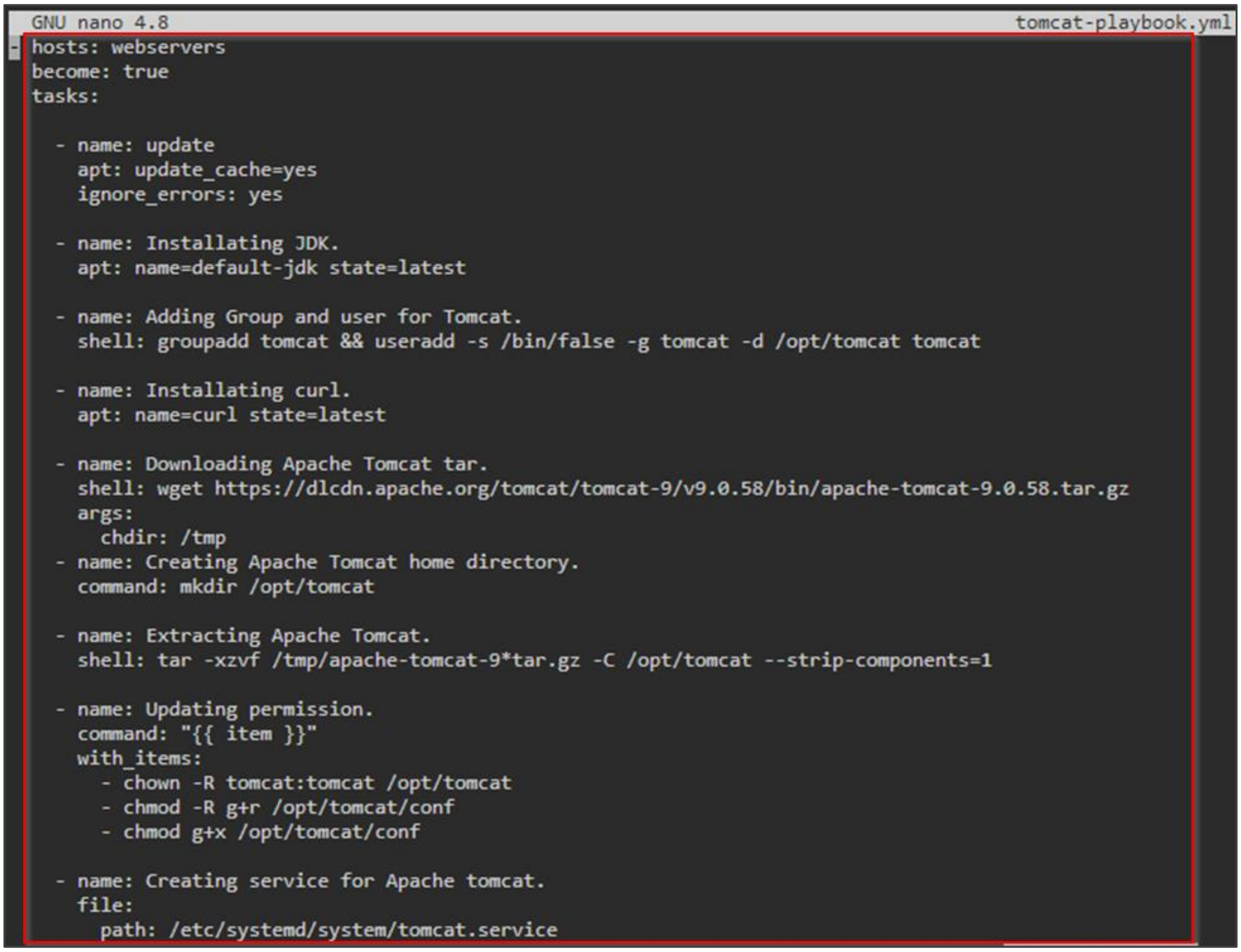
```
[redacted]@ip-172-31-17-34:~$ sudo nano tomcat-playbook.yml
```

4.2 Add the following YAML script in the **tomcat-playbook.yml** file:

```
- hosts: webservers
  become: true
  tasks:

    - name: update
      apt: update_cache=yes
      ignore_errors: yes

    - name: Installating JDK.
```

```
      apt: name=default-jdk state=latest

    - name: Adding Group and user for Tomcat.
      shell: groupadd tomcat && useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat

    - name: Installating curl.
      apt: name=curl state=latest

    - name: Downloading Apache Tomcat tar.
      shell: wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.58/bin/apache-tomcat-9.0.58.tar.gz
      args:
        chdir: /tmp
    - name: Creating Apache Tomcat home directory.
      command: mkdir /opt/tomcat

    - name: Extracting Apache Tomcat.
      shell: tar -xzvf /tmp/apache-tomcat-9*tar.gz -C /opt/tomcat --strip-components=1

    - name: Updating permission.
      command: "{{ item }}"
      with_items:
        - chown -R tomcat:tomcat /opt/tomcat
        - chmod -R g+r /opt/tomcat/conf
        - chmod g+x /opt/tomcat/conf
    - name: Creating service for Apache tomcat.
      file:
        path: /etc/systemd/system/tomcat.service
```

```
       state: touch
       mode: u+rwx,g-rwx,o-x
  - name: download foo.conf
     get_url:
       url: https://raw.githubusercontent.com/aftab70/Apache_Tomcat/master/tomcat_services
       dest: /etc/systemd/system/tomcat.service
  - name: Deamon reload.
     command: systemctl daemon-reload
  - name: Starting Apache Tomcat service.
     service: name=tomcat state=started
```



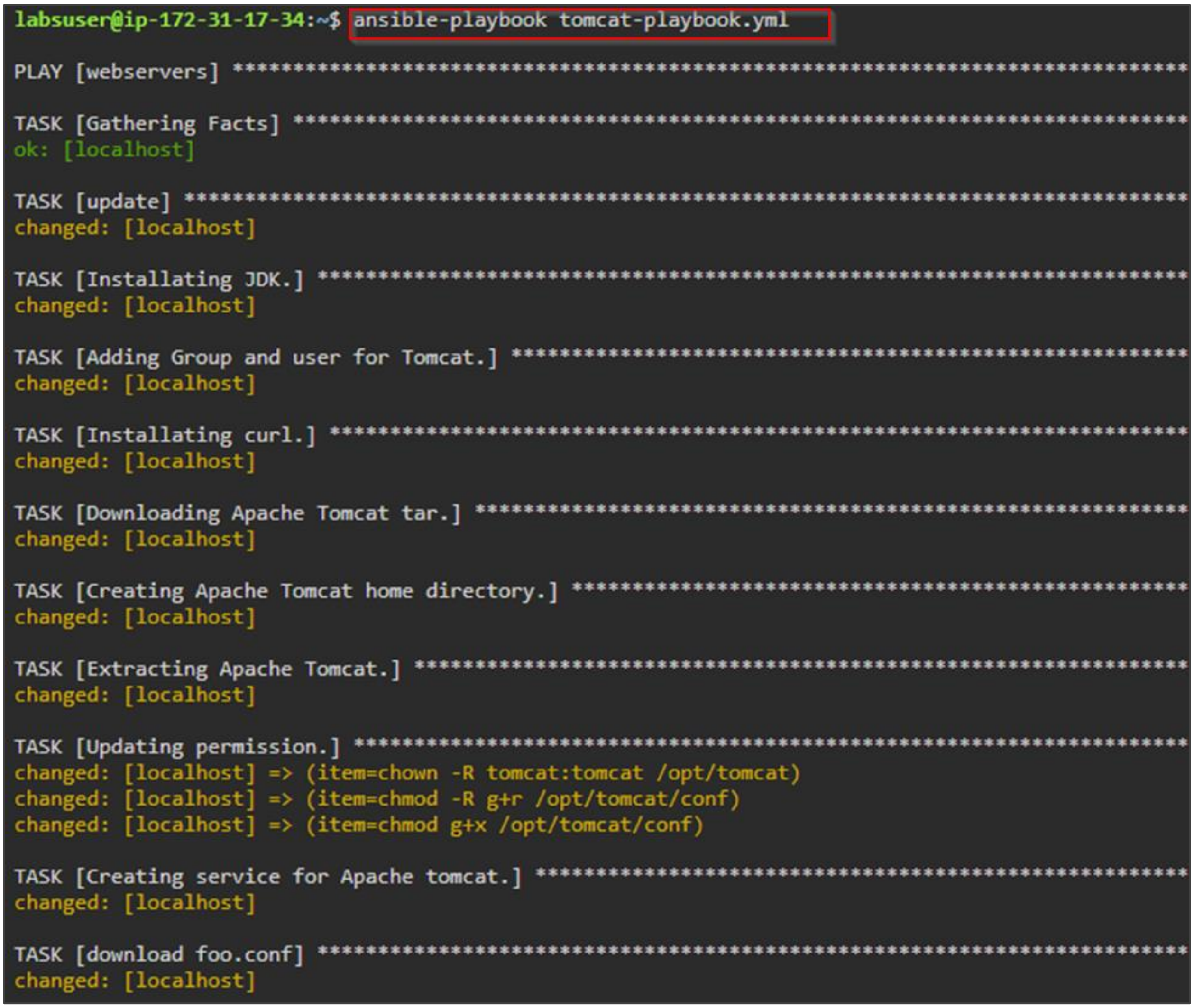
```
- hosts: webservers
  become: true
  tasks:

    - name: update
      apt: update_cache=yes
      ignore_errors: yes

    - name: Installating JDK.
      apt: name=default-jdk state=latest

    - name: Adding Group and user for Tomcat.
      shell: groupadd tomcat && useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat

    - name: Installating curl.
      apt: name=curl state=latest

    - name: Downloading Apache Tomcat tar.
      shell: wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.58/bin/apache-tomcat-9.0.58.tar.gz
      args:
        chdir: /tmp
    - name: Creating Apache Tomcat home directory.
      command: mkdir /opt/tomcat

    - name: Extracting Apache Tomcat.
      shell: tar -xzvf /tmp/apache-tomcat-9*tar.gz -C /opt/tomcat --strip-components=1

    - name: Updating permission.
      command: "{{ item }}"
      with_items:
        - chown -R tomcat:tomcat /opt/tomcat
        - chmod -R g+r /opt/tomcat/conf
        - chmod g+x /opt/tomcat/conf

    - name: Creating service for Apache tomcat.
      file:
        path: /etc/systemd/system/tomcat.service
```

This Ansible playbook performs several tasks on hosts in the [webservers] group, including updating the package cache, installing JDK and curl, creating a user and group for Tomcat, downloading and extracting Apache Tomcat, setting permissions, creating a Tomcat service, and starting the service.

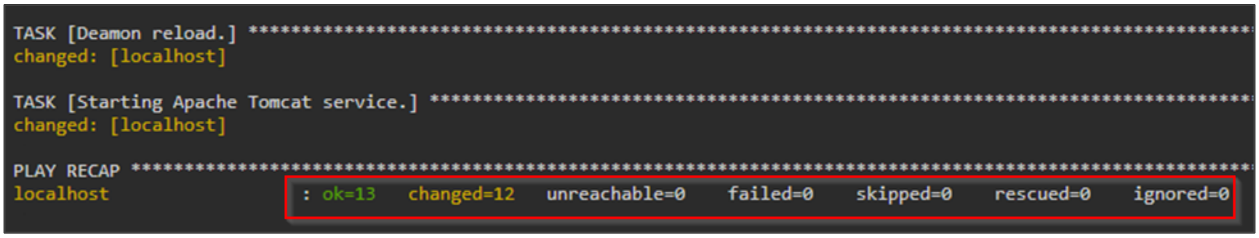
Note: Save the file and exit using **Ctrl + X**, type **Y** for yes, and then press enter.

Step 5: Execute the playbook

5.1 Execute the below command to run the playbook:
ansible-playbook tomcat-playbook.yml



```
labsuser@ip-172-31-17-34:~$ ansible-playbook tomcat-playbook.yml

PLAY [webservers] ****************************************************************************

TASK [Gathering Facts] ***********************************************************************
ok: [localhost]

TASK [update] ********************************************************************************
changed: [localhost]

TASK [Installating JDK.] *********************************************************************
changed: [localhost]

TASK [Adding Group and user for Tomcat.] *****************************************************
changed: [localhost]

TASK [Installating curl.] ********************************************************************
changed: [localhost]

TASK [Downloading Apache Tomcat tar.] ********************************************************
changed: [localhost]

TASK [Creating Apache Tomcat home directory.] ************************************************
changed: [localhost]

TASK [Extracting Apache Tomcat.] *************************************************************
changed: [localhost]

TASK [Updating permission.] ******************************************************************
changed: [localhost] => (item=chown -R tomcat:tomcat /opt/tomcat)
changed: [localhost] => (item=chmod -R g+r /opt/tomcat/conf)
changed: [localhost] => (item=chmod g+x /opt/tomcat/conf)

TASK [Creating service for Apache tomcat.] ***************************************************
changed: [localhost]

TASK [download foo.conf] *********************************************************************
changed: [localhost]
```



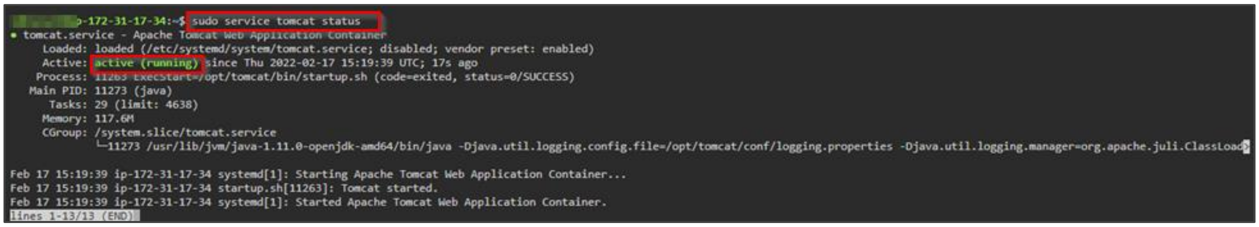
```
TASK [Deamon reload.] ***********************************************************************************
changed: [localhost]

TASK [Starting Apache Tomcat service.] ******************************************************************
changed: [localhost]

PLAY RECAP **********************************************************************************************
localhost                   : ok=13   changed=12   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

The playbook is successfully executed.

Step 6: Confirm the installation

6.1 Run the below command to check the status of the Tomcat:

sudo service tomcat status



```
[redacted]p-172-31-17-34:~$ sudo service tomcat status
● tomcat.service - Apache Tomcat Web Application Container
     Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: enabled)
     Active: active (running) since Thu 2022-02-17 15:19:39 UTC; 17s ago
    Process: 11263 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)
   Main PID: 11273 (java)
      Tasks: 29 (limit: 4638)
     Memory: 117.6M
     CGroup: /system.slice/tomcat.service
             └─11273 /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoad>

Feb 17 15:19:39 ip-172-31-17-34 systemd[1]: Starting Apache Tomcat Web Application Container...
Feb 17 15:19:39 ip-172-31-17-34 startup.sh[11263]: Tomcat started.
Feb 17 15:19:39 ip-172-31-17-34 systemd[1]: Started Apache Tomcat Web Application Container.
lines 1-13/13 (END)
```

By following the above steps, you have successfully installed Apache Tomcat using an Ansible playbook for automated deployment, configuration, and management of the application server.