**Ansible Exercise**

**Q)** Problem: a user want to install tomcat using ansible.

Write down all the steps given in manual process through command line:

we have used a lot of Ansible modules to facilitate various tasks.  This playbook consists of the following steps or tasks

* **Update and Upgrade packages –** Update the system package cache by running.
* **Download Open JDK**  – We are using the apt module to install java as a system package.
* **Validate if Java is available** –   Shell module is used to run a java -version command, the playbook would fail if the command fails (or) not found
* **Create a Group** – We are using a group module is used to create linux user group named tomcat
* **Create a User** – We are creating a new user named  tomcat using user module, this user and the group would be used by the tomcat application server as its not recommended to run servers on Root
* **Directory creation**– The file module is used to create a directory where the tomcat server be installed and operated. CATALINA\_BASE /opt/tomcat in our case
* **Download Tomcat installable tar** – The unarchive module is used to download tomcat from the URL and to untar once the file is downloaded. we have combined the efforts of get\_url and unarchive together as unarchive can have URL as a source.
* **Change Ownership of tomcat directory-** Changing the ownership of tomcat directory to /opt/tomcat
* **Creating a Service file for tomcat** – we are using the copy module to create a file in remote and the content of the service file is hard coded inside the playbook itself. another way to do it is to use template but this is convenient. we are using |- to maintain the line break while assigning the multiline data to the content variable.
* **Reloading SystemD** – Once the service file is created and placed on the appropriate /etc/systemd directory, we need to reload the system daemon to sync the changes.  Ansible has a dedicated module named systemd for to interact with SystemDaemon.
* **Enabling the tomcat to Auto-Start** – We are enabling the tomcat service we just created, for the boot time auto-start. this is done using systemd module too
* **Connect to the tomcat web interface and validate** – We are using the uri module to connect to the http://localhost:8080 to ensure that the tomcat is started and running fine.  we are also using [until and retry](https://www.middlewareinventory.com/blog/ansible-retry-examples/) to wait for the server to come up

Ansible.cfg file:

[defaults]

inventory = dev.inv

forks = 5

dev.inv file:

worker

Play.yml file:

---

- name: Install Tomcat to remote server

  hosts: all

  become: yes

  vars:

    download\_url: https://dlcdn.apache.org/tomcat/tomcat-10/v10.0.27/bin/apache-tomcat-10.0.27.tar.gz

  tasks:

   - name: Update and upgrade apt packages

     apt:

      upgrade: yes

      update\_cache: yes

   - name: Download Open JDK

     apt:

      name: default-jdk

      update\_cache: yes

      state: present

   - name: Check if Java is Installed

     shell:

      java -version

   - name: Create Group

     group:

      name: tomcat

      state: present

   - name: Create User

     user:

        name: tomcat

        state: present

   - name: Create a Directory /opt/tomcat

     file:

      path: /opt/tomcat

      state: directory

      mode: 0755

      owner: tomcat

      group: tomcat

   - name: Download Tomcat using unarchive

     unarchive:

      src: "{{download\_url}}"

      dest: /opt/tomcat

      remote\_src: yes

      extra\_opts: [--strip-components=1]

   - name: Change ownership of tomcat directory

     file:

      path: /opt/tomcat

      owner: tomcat

      group: tomcat

      mode: "u+rwx,g+rx,o=rx"

      recurse: yes

      state: directory

   - name: Creating a service file

     copy:

      content: |-

        [Unit]

        Description=Tomcat Service

        Requires=network.target

        After=network.target

        [Service]

        Type=forking

        User=tomcat

        Group=tomcat

        Environment="JAVA\_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64"

        Environment="JAVA\_OPTS=-Djava.security.egd=file:///dev/urandom"

        Environment="CATALINA\_PID=/opt/tomcat/logs/tomcat.pid"

        Environment="CATALINA\_BASE=/opt/tomcat"

        Environment="CATALINA\_HOME=/opt/tomcat"

        Environment="CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

        ExecStart=/opt/tomcat/bin/startup.sh

        ExecStop=/opt/tomcat/bin/shutdown.sh

        Restart=always

        RestartSec=10

        [Install]

        WantedBy=multi-user.target

      dest: /etc/systemd/system/tomcat.service

   - name: Reload the SystemD to re-read configurations

     systemd:

        daemon-reload: yes

   - name: Enable the tomcat service and start

     systemd:

        name: tomcat

        enabled: yes

        state: started

   - name: Connect to Tomcat server on port 8080 and check status 200 - Try 5 times

     tags: test

     uri:

       url: http://localhost:8080

     register: result

     until: "result.status == 200"

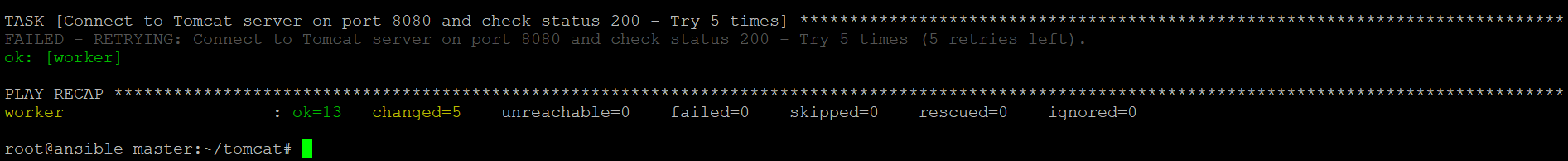
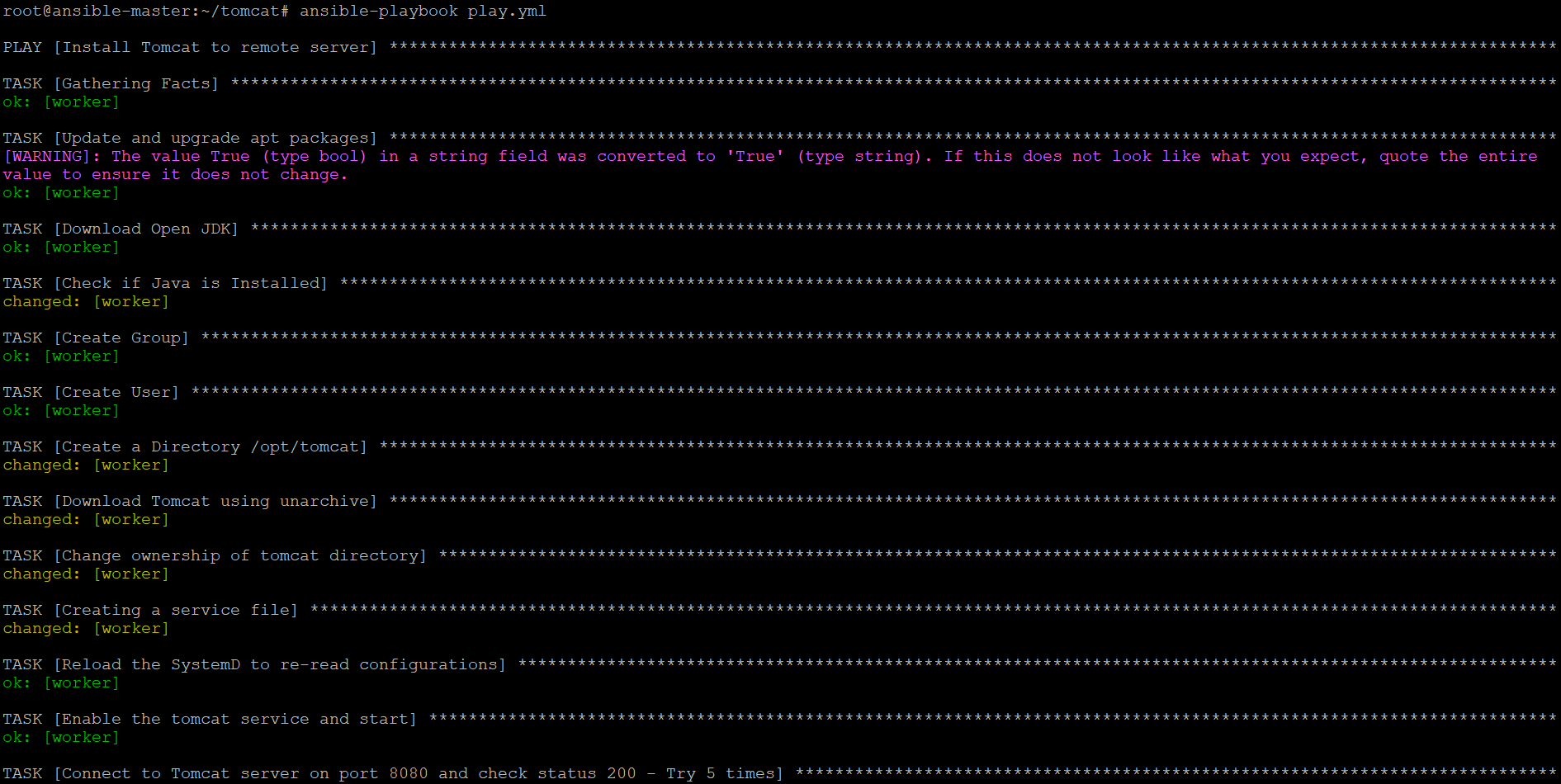
     retries: 5

     delay: 10

Master node console output:

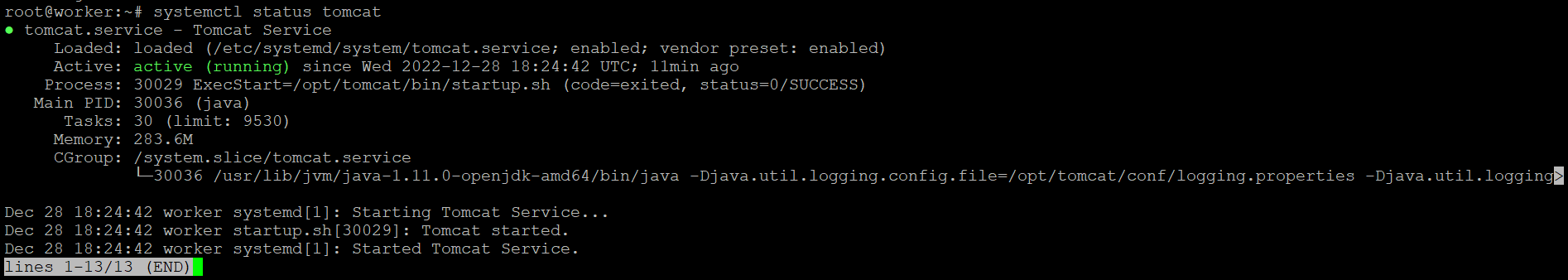
After running a playbook i.e., play.yml

**Ansible-playbook play.yml**



Console output of worker node:

Check status of tomcat in worker node using **systemctl status tomcat** command



Web server output:

This is running on public ip(52.188.200.206) of worker node followed by :8080

