Ansible playbook for installing Apache, Git, JAVA, Tomcat, Tree....

Playbook:

```
---
- hosts: dev
  user: aniket
  become: yes
  connection: ssh
  gather_facts: yes
  tasks:
        #############install Apache2##############
      - name: Install Apache
        apt: name=apache2 update_cache=yes state=present
      - name: start Apache
        action: service name=apache2 state=started


        ############install git##################
      - name: Install git
        action: apt pkg=git state=present
```

```
    ############install java##############

  - name: Install java11

    action: apt pkg=openjdk-11-jdk state=present

  ##############Install Unzip##############

  - name: install unzip

    action: apt name=unzip state=present

  ############Install tomcat############

  - name: create directory

    action: file path=/home/aniket/tomcat state=directory

  - name: Install tomcat

    action: get_url url=https://downloads.apache.org/tomcat/tomcat-
9/v9.0.73/bin/apache- tomcat-9.0.73.zip dest=/home/aniket/tomcat

  - name: unzip tomcat

    action: unarchive src=/home/aniket/tomcat/apache-tomcat-9.0.73.zip
dest=/home/aniket/tomcat remote_src=yes


  - name: Set ownership and permissions for Apache Tomcat

    action: file path=/home/aniket/tomcat/apache-tomcat-9.0.73
state=directory  owner=aniket group=aniket mode=777 recurse=yes

  - name: Start tomcat

    command: "nohup /home/aniket/tomcat/apache-tomcat-
9.0.73/bin/startup.sh"
```

```
###############Install tree###############

 - name: Install tree

   action: apt name=tree state=present
```

Explanation:

1. '- **hosts: dev**' is the beginning of an Ansible playbook YAML file. It specifies the target hosts on which the tasks in the playbook will be executed. In this case, the playbook is targeting the hosts with the name "dev". It indicates that the tasks following this line will be applied to the "dev" host(s) specified in the Ansible inventory file.

2. `**user: aniket**` in the Ansible playbook specifies the username to be used when connecting to the target hosts. In this case, the playbook will connect to the target hosts using the username "aniket". This user should have the necessary permissions and privileges to perform the tasks defined in the playbook.

3. The line '**become: yes'** in the Ansible playbook enables privilege escalation. When become is set to yes, Ansible will attempt to execute the tasks with elevated privileges, typically by using the sudo command

4.The line '**connection: ssh'** in the Ansible playbook specifies the connection method to be used when communicating with the target hosts. In this case, the playbook will establish an SSH connection to the target hosts.

5.The line '**gather_facts: yes'** in the Ansible playbook enables the gathering of facts from the target hosts before executing tasks. Facts are variables containing information about the target hosts, such as network interfaces, operating system details, disk usage, and more.

6.The '**tasks**:' section in an Ansible playbook contains a list of individual tasks that will be executed on the target hosts. Each task describes a specific action to be performed.

7.**Install Apache** -These tasks ensure that Apache is installed on the target hosts and that the Apache service is started. The *apt* module handles the installation of the package, and the *service* module is responsible for starting and managing services on the target hosts.

8.**Install git**-This task will use the apt module to install the "git" package. The name parameter specifies the package name as "git", and the state parameter is set to "present" to ensure that Git is installed on the target hosts.

9.**Install JAVA** -This task utilizes the apt module to install the "openjdk-11-jdk" package, which corresponds to Java 11 development kit. The name parameter specifies the package name, and the state parameter is set to "present" to ensure that Java 11 is installed on the target hosts.

10.**Install unzip -**This task uses the apt module to install the unzip package. The name parameter specifies the package name as unzip, and the state parameter is set to present to ensure that unzip is installed on the target hosts.

11.**Create directory:**This task uses the file module to create a directory. The path parameter specifies the directory path as /home/aniket/tomcat, and the state parameter is set to directory to ensure that the directory exists on the target hosts.

12.**Install Tomcat** -This task uses the get_url module to download the Apache Tomcat zip file from the specified URL and save it to the /home/aniket/tomcat directory with the filename apache-tomcat-9.0.73.zip.

13.**Unzip tomcat -**This task uses the unarchive module to extract the contents of the apache-tomcat-9.0.73.zip file located at /home/aniket/tomcat and extract them to the /home/aniket/tomcat directory on the target hosts. The remote_src parameter is set to yes to indicate that the source file is on the remote hosts.

14.**Set ownership and permissions for Apache Tomcat -** This task uses the file module to set the ownership, group, and permissions for the /home/aniket/tomcat/apache-tomcat-9.0.73 directory on the target hosts. The path parameter specifies the directory path, the state parameter is set to

directory, and the owner, group, and mode parameters are used to set the desired ownership, group, and permissions.

15. **Start Tomcat-** This task uses the command module to execute the command "nohup /home/aniket/tomcat/apache-tomcat-9.0.73/bin/startup.sh". The nohup command is used to run the Tomcat startup script in the background and prevent it from being terminated when the SSH session ends.

16. **Install Tree -** This task uses the apt module to install the tree package. The name parameter specifies the package name as tree, and the state parameter is set to present to ensure that tree is installed on the target hosts.