**DEVOPS PROJECT-2**

**Ansible Installation & Configuration on AWS**

Step 1: Create an instance for Ansible-Server using Amazon Linux 2.

Step 2: Create an instance for Ansible-Client using Amazon Linux 2.

Step 3: Using the public IPV4 addresses of both Ansible-Client and Ansible-Server create a connection using PuTTY. Open both the Ansible-Client & Server in separate PuTTY.

Step 4: In Ansible-Client & Server make them as root using sudo su – command.

Step 5: Then type yum update -y command on both Server & Client PuTTY.

Step 6: Then on Ansible-Server install ansible using amazon-linux-extras install ansible2 command.

Step 7: After installing ansible check the ansible version using

ansible –-version command

Step 8: In Ansible-Client install java using yum install java-1.8\* -y command.

Step 9: Then download the installation package of tomcat 8 by googling the tomcat 8 download. Then go to the 1st website and inside that **Binary Distributions » Core** then, copy the address of the .tar.gz file and inside the Ansible-Client PuTTY paste the address as

wget https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.79/bin/apache-tomcat-8.5.79.tar.gz

Step 10: Then check for the downloaded zip file using ls command. Then, unzip the downloaded file using tar -xvzf /opt/apache-tomcat-8.5.79.tar.gz command.

Step 11: Then using ls command check for the folder as apache-tomcat-8.5.79.

Step 12: Then, get into the folder using cd apache-tomcat-8.5.79 inside that go to bin folder

Step 13: There we can see startup.sh and shutdown.sh.

Step 14: We are making those two script files to executable using chmod +x /opt/apache-tomcat-8.5.79/bin/startup.sh shutdown.sh command.

Step 15: Now we are creating a softlink to star and stop the tomcat server, the command ln -s /opt/apache-tomcat-8.5.79/bin/startup.sh usr/local/bin/tomcatup as well as for shutdown use the command ln -s /opt/apache-tomcat-8.5.79/bin/shutdown.sh usr/local/bin/tomcatdown **(**After executing these commands we can just give tomcatup and tomcatdown to start and stop the server**).**

Step 16: Next, we’re going to start the server using tomcatup.

Step 17: Now inside the Ansible-Server PuTTY, type useradd ansadmin

Step 18: Then type visudo and press enter.

Inside visudo press i button for insert mode then, after root add

ansible ALL=(ALL) NOPASSWD: ALL

ansadmin ALL=(ALL) NOPASSWD: ALL

these two lines like thisText

Description automatically generated

Step 19: Inside the Ansible-Client also add these two lines inside visudo.

Step 20: Then, inside the Ansible-Server after useradd ansadmin type passwd ansadmin give a new password and retype the password after that it will show as passwd: all authentication tokens updated successfully.

Step 21: Then, go to cd /etc/ssh and give ls command you’ll see files and folders in that edit the ssh\_config file using vi ssh\_config

Inside that use the i button for insert mode, then, remove the # for the PasswordAuthentication yes like this

Text

Description automatically generated

Step 22: Then, exit and save the file by pressing the esc buton and type :wq

Step 23: Then, edit the sshd\_config file using vi command.

Step 24: Inside that file remove the # for PasswordAuthentication yes and add an # for PasswordAuthentication no like this

Text

Description automatically generated

Step 25: Then, restart the service using service sshd restart command.

Step 26: Do the same steps from step 21 to 25 inside the Ansible-Client also.

Step 26: Inside the Ansible-Client also type useradd ansadmin and passwd ansadmin. Type the same password what you’ve gave inside the Ansible-Server and retype the same password.

Step 27: Inside the Ansible-Server PuTTY I think we’re inside the ssh folder. Change from root to ansadmin using su ansadmin command.

Step 28: Type ssh-keygen command

Text

Description automatically generated

Step 29: Now, get out of the ssh and etc folder using cd .. command.

Step 30: Then, go inside the cd /home/ansadmin folder.

Step 31: Type ls -lsa command to see the files folders and their permissions. Then, go to .ssh folder using cd .ssh command.

Step 32: Then, type ssh-copy-id 172.31.9.130 (Ansible-Client’s private IPV4 address)

Step 33: Now, try logging into the machine, with: ssh 172.31.9.130

and check to make sure that only the key(s) you wanted were added.

Step 34: Now, you’ve noted that we’re inside the Ansible-Client’s server. Exit from that server using exit command

Step 35: Now, type sudo vi /etc/ansible/hosts then, inside that delete all the contents inside that file then type [web] in 1st line and add the Ansible-Client’s private IPV4 address. Then, save and exit the file using esc button and :wq

Step 36: Now, type ansible all -m ping Here, we’re trying to check whether the connection is stable or not by pinging the Ansible-Client like this

Text

Description automatically generated

We’ve configured the Ansible-Server & Ansible-Client successfully……

Step 37: Check if the connection is stable or not by connecting with the Ansible-Client using ssh followed by Ansible-Clients private IP address like

ssh 172.31.9.130

Step 38: Now, ping the Ansible-Client using the command Ansible all -m ping

If the connection is stable it’ll show as SUCCESS.

Step 39: Create “copyfile.yml” playbook inside /opt/playbooks

Step 40: We need to create a playbook to transfer file from Ansible-Server to Ansible-Client.

Step 41: Go to Jenkins server (webpage) and install “publish over ssh” plugin.

Step 42: Enable connection between Ansible and Jenkins.

* Manage Jenkins » Configure System » Publish Over SSH » SSH Servers
  + SSH Servers:
    - Hostname: 172.31.9.130
    - username: ansadmin
    - password: \*\*\*\*\*\*\*\*
    - Test the connection "Test Connection"

Step 43: Then, create a Jenkins job named as “Maven\_Project-2”

* *Source Code Management:*
  + Repository: https://github.com/AthulyaG/hello-world.git
  + Branches to build: \*/master
* *Build:*
  + Root POM: pom.xml
  + Goals and options : clean install package
* *Add post-build steps*
  + Send files or execute commands over SSH
    - SSH Server : Ansible-Server
    - Source files: webapp/target/\*.war
    - Remote directory: //opt//playbooks
* *Add post-build steps*
  + Send files or execute commands over SSH
    - SSH Server : Ansible-Server
    - Exec command

ansible-playbook /opt/playbooks/copyfile.yml

Step 44: Execute job and you should be able to see that the build has been deployed on Tomcat server.