Ansible Installation & Configuration on AWS

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

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

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.

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

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

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

7. After installing ansible check the ansible version using

**ansible --version** command

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

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**

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.

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

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

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

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.

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).

But this step did not work for me.

16. Next, we’re going to start the server using **./startup.sh (default)**

17. Now inside the Ansible-Server PuTTY, type **useradd ansadmin**

18. Then type **visudo** and press enter

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

**ansible ALL=(ALL) NOPASSWD: ALL**

**ansadmin ALL=(ALL) NOPASSWD: ALL**

20. Inside the Ansible-Client also add these two lines inside visudo.

21. 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.

22. 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

23. Then, exit and save the file by pressing the esc button and type :wq

24. Then, edit the sshd\_config file using vi command.

25. Inside that file remove the # for PasswordAuthentication yes and add # for PasswordAuthentication no ,

26. Then, restart the service using service sshd restart command.

27. Do the same steps from step 21 to 25 inside the Ansible-Client also.

28. 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.

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

30. Type ssh-keygen command.

31. Now, get out of the ssh and etc folder using cd .. command.

32. Then, go inside the cd /home/ansadmin folder.

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

34. Then, type ssh-copy-id 172.31.4.179 (Ansible-Client’s private IPV4 address)

35. Now, try logging into the machine, with: ssh 172.31.4.179

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

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

37. 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

38. Now, type ansible all -m ping

Here, we’re trying to check whether the connection is stable or not by pinging the Ansible-Client .

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

39. 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 172.31.4.179**

40. Now, ping the Ansible-Client using the command **Ansible all -m ping**

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

41. Create “copyfile.yml” playbook inside /opt/playbooks

42. We need to create a playbook to transfer file from Ansible-Server to Ansible-Client.

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

44. Enable connection between Ansible and Jenkins.

* Manage Jenkins » Configure System » Publish Over SSH » SSH Servers
  + SSH Servers:
    - Hostname: 172.31.12.227
    - username: ansadmin
    - password: \*\*\*\*\*\*\*\*
    - Test the connection "Test Connection"

45. Then, create a Jenkins job named as “Project 2”

* *Source Code Management:*
  + Repository: https://github.com/Harish-77/SimpleDevopsProject.git (GitHub URL)
  + 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

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