**Configuring Build Agents in Jenkins**

**Introduction**

In this hands-on lab, we will create build agent accounts on our remote agents that will then be used to connect those to the Jenkins server to allow us to build remote jobs in a more secure way. We will use specific accounts for the build agents that are not used by normal users and are accessed by the Jenkins server via SSH keys. This can be done as part of a more secure strategy for deploying build agents, and it also allows for better Audit Trail management.

**Log In to the Jenkins Master, and Set Up Keys for the cloud\_user on the Agent Server**

1. Open a terminal session, and log in to the Jenkins instance via SSH using the credentials provided:

ssh cloud\_user@<JENKINS\_SERVER\_PUBLIC\_IP>

The password is the one provided with the public IP.

1. From the Jenkins server, access the agent instance as the jagent user:

[cloud\_user@jenkins]$ ssh jagent@<AGENT\_SERVER\_PUBLIC\_IP>

The password is Zaq1Xsw2.

1. Make sure jagent has Java installed:

[jagent@jenkins]$ java -version

1. Change to the home directory:

[jagent@jenkins]$ cd ~

1. Ensure jagent's home directory is in /var/lib:

[jagent@jenkins]$ pwd

We should see /var/lib/jagent.

1. See what files are there:

[jagent@jenkins]$ ls -a

We'll see bash files listed.

1. Exit the agent machine:

[jagent@jenkins]$ exit

1. Back on the Jenkins server, see what files are there:

[cloud\_user@jenkins]$ ls -a

This time, we'll see bash files and .ssh.

1. Change to the .ssh directory:

[cloud\_user@jenkins]$ cd ./.ssh

1. List its contents:

[cloud\_user@jenkins .ssh]$ ls

1. View the contents of known\_hosts:

[cloud\_user@jenkins .ssh]$ cat known\_hosts

We'll see there's a fingerprint for the agent server.

1. Generate an SSH key:

[cloud\_user@jenkins .ssh]$ ssh-keygen

1. Press **Enter** when prompted to store it in the default location without a passphrase.
2. List the contents of the directory again:

[cloud\_user@jenkins .ssh]$ ls

This time, we'll also see id\_rsa and id\_rsa.pub.

1. Copy the keys from cloud\_user on the Jenkins server to jagent on the agent server:

[cloud\_user@jenkins .ssh]$ ssh-copy-id jagent@<AGENT\_SERVER\_PUBLIC\_IP>

The password is Zaq1Xsw2.

1. Verify you are able to log in to the agent server as jagent without a password:

[cloud\_user@jenkins .ssh]$ ssh jagent@<AGENT\_SERVER\_PUBLIC\_IP>

This time, it should log us in without prompting for a password.

**Copy the .ssh Information from cloud\_user to the Jenkins User**

1. Exit back to the Jenkins server:

[jagent@jenkins]$ exit

1. Get back to the home directory:

[cloud\_user@jenkins]$ cd ~

1. Become the root user:

[cloud\_user@jenkins]$ sudo su

1. Verify there is not an .ssh directory in the Jenkins user's home directory:

[root@jenkins cloud\_user]# ls -a /var/lib/jenkins/

We shouldn't see one included on the list.

1. Copy the contents of the .ssh directory from /home/cloud\_user/.ssh to /var/lib/jenkins/.ssh:

[root@jenkins cloud\_user]# cp -r ./.ssh /var/lib/jenkins/

1. List the contents of the directory again:

[root@jenkins cloud\_user]# ls -a /var/lib/jenkins/

This time, we should see .ssh included.

1. Check the permissions on .ssh:

[root@jenkins cloud\_user]# ls -la /var/lib/jenkins/

We'll see it's owned by root, rather than jenkins.

1. Correct the permissions on the /var/lib/jenkins/.ssh directory so jenkins is the owner:

[root@jenkins cloud\_user]# chown -R jenkins:jenkins /var/lib/jenkins

1. Check the permissions on .ssh:

[root@jenkins cloud\_user]# ls -la /var/lib/jenkins/

This time, we'll see it's owned by jenkins.

**Configure the Agent in the Jenkins GUI**

1. Change directory to /jenkins/:

[root@jenkins cloud\_user]# cd /var/lib/jenkins/

1. Change directory to .ssh:

[root@jenkins jenkins]# cd ./.ssh

1. List its contents:

[root@jenkins .ssh]# ls

1. View id\_rsa:

[root@jenkins .ssh]# cat id\_rsa

1. Copy the entire contents of id\_rsa, and paste it into a text file — we'll need it for the last objective of the lab.
2. Access the Jenkins GUI by navigating in the browser to the Jenkins server public IP with :8080 appended at the end: <PUBLIC\_IP>:8080
3. Log in using the following credentials:
   * *Username*: **student**
   * *Password*: **OmgPassword!**
4. Click **Manage Jenkins** in the left-hand menu.
5. Click **Manage Nodes and Clouds**.
6. Click **New Node** in the left-hand menu.
7. For *Node name*, enter **Ubuntu\_agent**.
8. Select **Permanent Agent**.
9. Click **OK**.
10. On the next screen, set the following values:
    * *# of executors*: **4**
    * *Remote root directory*: **/var/lib/jagent**
    * *Labels*: **ubuntu\_agent**
    * *Usage*: **Only build jobs with label expressions matching this node**
    * *Launch method*: **Launch agents via SSH**
    * *Host*: Enter the agent server's public IP address
    * *Credentials*: **Add**>**Jenkins**
      + *Kind*: **SSH Username with private key**
      + *Description*: **ubuntuagent**
      + *Username*: **jagent**
      + *Private Key*: **Enter directly**, **Add**, and paste in the id\_rsa private key you copied in the terminal
      + Click **Add**
11. Once it's added, for *Credentials*, select **jagent**.
12. Click **Save**.
13. Select **Ubuntu\_agent**.
14. Click to **See log for more details**. Verify the agent comes online by making sure there's a line saying, "Agent successfully connected and online."

**Conclusion**

Congratulations on successfully completing this hands-on lab!

Graphical user interface, diagram

Description automatically generated