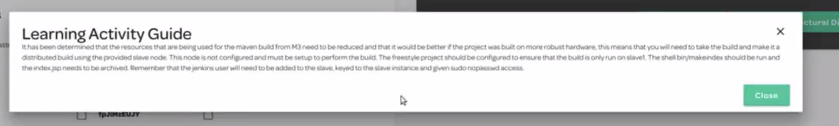
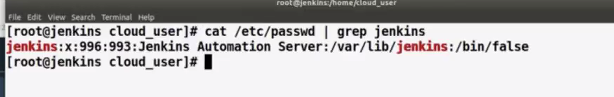
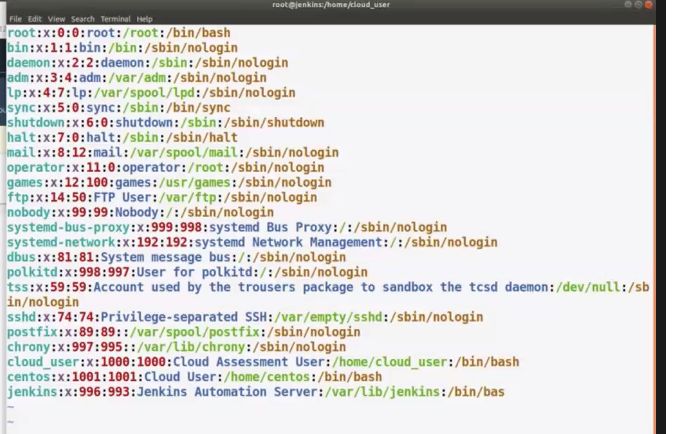


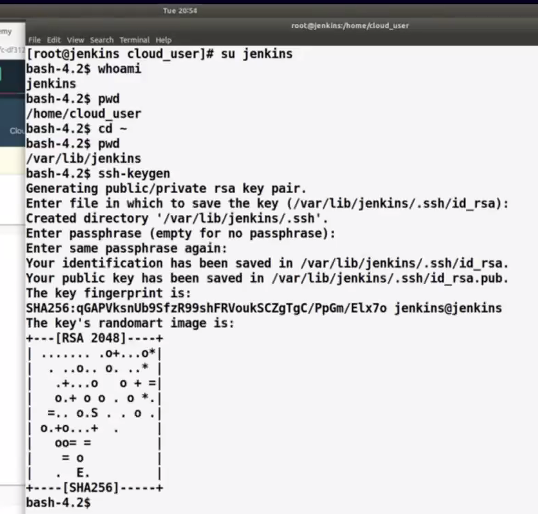
The URI of the git repository for this lesson is <https://github.com/linuxacademy/content-cje-prebuild.git>

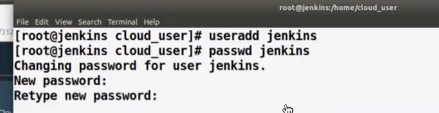
**Distributing the Build**

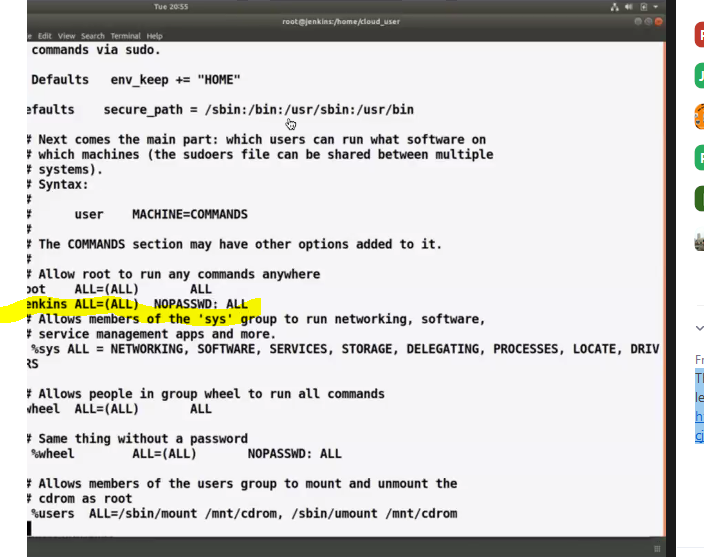














**Distributing a Build**

**Introduction**

**In this hands-on lab, we will configure Maven to build a project pulled from SCM — but we will configure a slave node to build the project instead of building the project on the master node.**

**Solution**

**Log in to the Jenkins master server using the credentials provided:**

**ssh cloud\_user@<MASTER\_PUBLIC\_IP\_ADDRESS>**

**Become root:**

**sudo su**

**Configure the Slave Machine for Use with the Jenkins Master**

**Open the /etc/passwd file:**

**[root@master]$ vim /etc/passwd**

**In the last line in the file (beginning with jenkins), change /bin/false to /bin/bash to allow the jenkins user a shell login.**

**Save and exit the file by pressing Escape followed by :x.**

**Change the password for the jenkins user:**

**[root@master]$ passwd jenkins**

**Enter a password of your choice that you'll easily remember.**

**Switch to jenkins:**

**[root@master]$ su jenkins**

**Change directory:**

**[jenkins@master]$ cd ~**

**Generate a public/private RSA key pair:**

**[jenkins@master]$ ssh-keygen**

**Log in to the slave server:**

**[jenkins@master]$ ssh cloud\_user@<SLAVE\_PUBLIC\_IP\_ADDRESS>**

**Become root:**

**[cloud\_user@slave]$ sudo su**

**Create a jenkins user:**

**[root@slave]$ useradd jenkins**

**Create a password:**

**[root@slave]$ passwd jenkins**

**Open the sudoers file:**

**[root@slave]$ visudo**

**In the Defaults section, beneath root, add:**

**jenkins ALL=(ALL) NOPASSWD: ALL**

**Save and exit the file by pressing Escape followed by :x.**

**Exit root:**

**[root@slave]$ exit**

**See who you're logged in as:**

**[cloud\_user@slave]$ whoami**

**You should see you're cloud\_user.**

**Switch to jenkins:**

**[cloud\_user@slave]$ su jenkins**

**Enter the password you created.**

**Change directory:**

**[jenkins@slave]$ cd ~**

**Enter exit twice to exit back to the master server.**

**See who you're signed in as:**

**whoami**

**You should see you're jenkins.**

**As the jenkins user on the master server, copy the jenkins user's ssh keys to the slave server:**

**[jenkins@master]$ ssh-copy-id jenkins@<SLAVE\_PUBLIC\_IP\_ADDRESS>**

**Run the following:**

**cat ./.ssh/id\_rsa**

**Keep the output listed, as we'll need it for a later step.**

**Run the Maven Build on the Remote Agent**

**In a new browser tab, navigate to http://<JENKINS\_MASTER\_SERVER\_PUBLIC\_IP>:8080.**

**Log in to Jenkins using the following credentials:**

**User: student**

**Password: OmgPassword!**

**Click Manage Jenkins in the left-hand menu.**

**Click Manage Nodes and Clouds.**

**Click New Node.**

**Give it a name of slave1.**

**Select Permanent Agent.**

**Click OK.**

**For Remote root directory, enter /home/jenkins.**

**For Labels, enter slave1.**

**For Host, enter the slave server's public IP address.**

**Next to Credentials, click Add > Jenkins.**

**Set the following values:**

**Kind: SSH Username with private key**

**Username: jenkins**

**Private Key: Enter directly**

**Copy the entire RSA key in the terminal (from dashes to dashes) and paste it into the Key window**

**ID: jkey**

**Description: jenkinsuser**

**Click Add.**

**Set Credentials to jenkins (jenkinsuser).**

**Click Save.**

**In the upper-left corner, click Jenkins > New Item.**

**Enter an item name of mavenproject.**

**Select Freestyle project.**

**Click OK.**

**Set the following values:**

**General**

**Restrict where this project can be run: Check**

**Label Expression: slave1**

**Source Code Management**

**Git: Check**

**Repository URL: https://github.com/linuxacademy/content-cje-prebuild.git**

**Click outside the box to make sure the red text goes away.**

**Build**

**Click Add build step > Invoke top-level Maven targets.**

**Goals: clean package**

**Click Add build step > Execute shell.**

**Command: bin/makeindex**

**Post-build Actions**

**Click Add post-build action > Archive the artifacts.**

**Files to archive: index.jsp**

**Click Advanced....**

**Fingerprint all archived artifacts: Check**

**Leave other default boxes checked.**

**Click Save.**

**In the upper-left corner, click Jenkins > Manage Jenkins > Global Tool Configuration.**

**In the Maven section, click Add Maven.**

**Give it the name M3.**

**Click Save.**

**In the upper-left corner, click Jenkins.**

**Click mavenproject.**

**Click Configure in the left-hand menu.**

**In the Build section, set Maven Version to M3.**

**Click Save.**

**Click Build Now in the left-hand menu.**

**Once the build starts, click the dropdown icon next to #1 and select Console Output and observe its progress.**

**Solution**

**Configure Maven Installer**

**Use a browser to navigate to the provided public IP address for the server. Remember to add ":8080" to the end of the IP address to specify the default Jenkins port.**

**Log in with the credentials provided in the lab guide instructions.**

**Click Manage Jenkins.**

**Click Global Tool Configuration.**

**Under Maven installations, click Add Maven.**

**In the Name box, enter "M3".**

**Make sure Install automatically is checked.**

**Click Save.**

**Configure the Build to Use Maven and Make the Index File**

**Click New Item.**

**From Jignesh Makwana to Me: (Privately) 07:22 PM**

**The URI of the git repository for this lesson is https://github.com/linuxacademy/content-cje-prebuild.git**

**From surender Singh to Me: (Privately) 07:40 PM**

**https://docs.microsoft.com/en-us/learn/certifications/exams/az-400 , content changing on 24 Nov**

**Enter an item name of "mavenproject" in the box provided.**

**Select Freestyle project.**

**Click OK.**

**Click the Source Code Management tab at the top of the screen.**

**Select the option for a Git repository.**

**Copy the git repository link from the lab instructions and enter it into the Repository URL box.**

**Click the Build tab at the top of the screen.**

**Click Add build step and select the Invoke top-level Maven targets option.**

**Under Maven Version, select M3.**

**In the Goals box, enter "clean package".**

**Click Add build step and select the Execute shell option.**

**In the Command window, enter "bin/makeindex".**

**Click Add post-build action and select the Archive the artifacts option.**

**Inside the Archive the artifacts box, click Advanced...**

**Check the option for Fingerprint all archived artifacts.**

**In the Files to archive box, enter "index.jsp".**

**Click Save.**

**Click Build Now.**

**Refresh the window and click the View link next to index.jsp. Verify the contents of the index.jsp file.**