**Capstone - Exercise 5: Setting Up Jenkins with GitHub Project**

Open the Cloud Platform Console at [https://console.cloud.google.com](https://console.cloud.google.com/).

Click on the three horizontal bars at the left most side of the blue bar near the top of the browser window. *Select Compute Engine*.

Select *VM Instances*. You should see the virtual machine you created earlier.

Click on the checkbox to the left of the VM name and then select *START*. It will take a few moments to start.

Click on *SSH* to start a terminal window.

**Change the host name to student:** Find the icon that looks like a gear in the upper right-hand corner of this terminal browser window and select *Change Linux User Name*. Enter *student* and *click Change*. Now, notice the prompt that says "student@lab:~$"



**Setting Up Jenkins with GitHub Project**

**Step 2**

Run Jenkins container if it’s stopped. Follow instructions from a previous lab to do this.

**Step 3**

Create a GitHub account and log in. Click on *Import Repository*.

**Step 4**

Enter the URL:

*https://github.com/simplilearn-devops/devops-capstone/*

Give a name to your repository like *Devops,* and click on import.

**Step 5**

Once import is complete, click on the repository name and you will have the *DevOps* project in that.

**Step 6**

Copy the URL of your DevOps project. You will need this to set up Jenkins.

**Step 7**

Start a browser on your local machine and enter the URL x.x.x.x:8080, replacing x.x.x.x with the external IP address of your virtual machine.

You will see *Jenkins Login* screen. Enter user ID and password and hit *Continue*.

**Step 8**

Select *create new jobs*.  
Enter the name *Build*.  
Select *Freestyle project*.  
Hit *OK*.

**Step 9**

Under Source Code Management, select Git and paste your GitHub URL that you copied in step 6.

**Step 10**

Select *build steps*.  
Select *Top Level Maven Targets.*

Select Default Maven and in goals, enter *-P metrics pmd:pmd*

In Post Build Actions, select Publish PMD analysis results and select location as *\*\*/\*.xml*

Apply and save.

Click on the Jenkins project Build, and click on Build Now.

Click on last build history, and click on Console Output.

**Step 12**

Select *create new jobs*.  
Enter the name *Test*.  
Select *Freestyle project*.  
Hit *OK*.

**Step 13**

Under Source Code Management, select Git and paste your GitHub URL that you copied in step 6.

**Step 14**

Select *build steps*.  
Select *Top Level Maven Targets*

Select Default Maven and in goals, enter *test.*

Apply and save.

Click on the Jenkins project Test, and click on Build Now.

Click on last build history, and click on Console Output.

**Step 15**

Select *create new jobs*.  
Enter the name *Code\_Metrics*.  
Select *Freestyle project*.  
Hit *OK*.

**Step 13**

Under Source Code Management, select Git and paste your GitHub URL that you copied in step 6.

**Step 14**

Select *build steps*.  
Select *Top Level Maven Targets*

Select Default Maven and in goals, enter *cobertura:cobertura*

In Post Build Actions, select Publish Cobertura Coverage Report and select location as *\*\*/targets/cobertura/coverage.xml*

Apply and save.

Click on the Jenkins project Code\_Metrics, and click on Build Now.

Click on last build history, and click on Console Output.

You will need to stop the lab computer at the end of each day to prevent it from accumulating costs during the evening and night.

From the Web UI, you can navigate to the Compute Engine section and select your lab computer. When it is selected, click on the icon representing the "Stop" operation as shown below:

