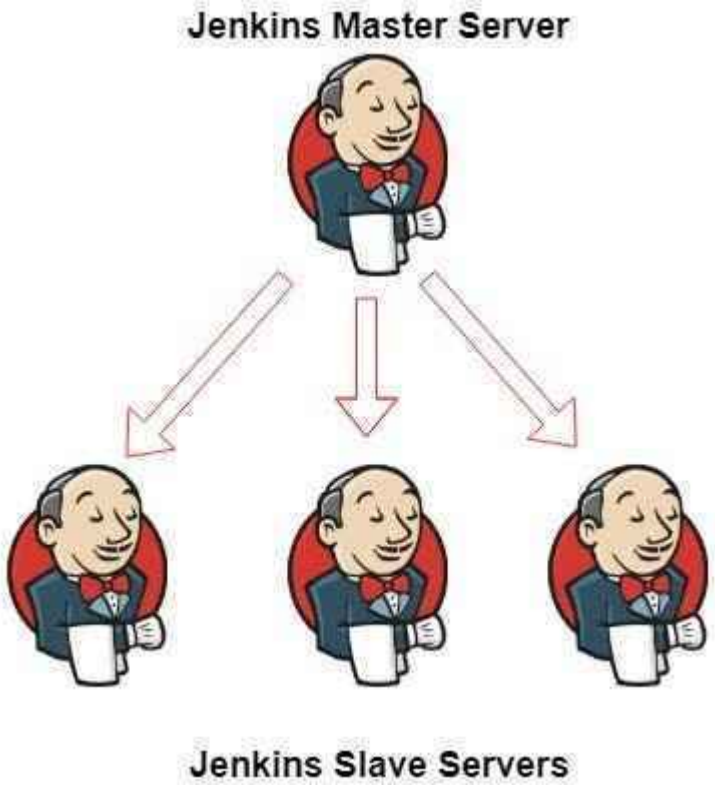
Test Automation: Jenkins Controller Agent setup in GCP: Ubuntu VM (Part 2)

Hello everyone, hope you enjoyed the Part 1 article of the series if you still have not checked it out do take a look here

In this Part 2 article, we will now go ahead and install other essential components such as Java JDK, Maven, Git, and finally Jenkins in our Ubuntu VM instance which is running on Google Cloud Platform (GCP). Let's get started.



Step 1: Install Java 11 JDK in your VM Instance

Pre-requisite: As mentioned in Part 1 of the article, firstly you will need to connect to your VM instance to be able to run commands. You can use the option "Open in Browser window" available under SSH for this. This will open up a new browser window which will be used to SSH and connect to the machine.

Now, we will go ahead and install Java JDK 11 in our instance. You can run the below command to install Java. This might take a few minutes.

sudo apt-get install openjdk-11-jdk -y

Post-installation, you can check if Java JDK version 11 has been successfully installed using the below command.

java -version

You should be seeing something as below:

```
openjdk version "11.0.16" 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Ubuntu-0ubuntu118.04)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Ubuntu-0ubuntu118.04, mixed mode, sharing)
```

Installed JDK details

Now, post the installation of Java JDK an important step which we need to configure is JAVA_HOME, run the below command to get the full path.

readlink -f $(which java)

```
/usr/lib/jvm/java-11-openjdk-amd64/bin/java
```

which java command Output

As seen above, JAVA_HOME will be: /usr/lib/jvm/java-11-openjdk-amd64

You can set JAVA_HOME using the below command.

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/

Note: Do make a note of this path location, as we will need to configure this in our Jenkins Global Tool configuration.

Step 2: Install Maven

To install Maven, just run the below command

sudo apt install maven -y

Post installation, run the command to verify

mvn -version

```
Apache Maven 3.6.0
Maven home: /usr/share/maven
Java version: 11.0.16, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.4.0-1083-gcp", arch: "amd64", family: "unix"
```

Maven Version

As you can see above Maven Home path is /usr/share/maven

Step 3: Install Git

To install Git, run the below command

sudo apt install git

To get the path, run the command,

which git

It will return /usr/bin/git

Step 4: Install Jenkins

Run the below set of four commands one by one in sequence to install Jenkins in the instance.

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/nullecho deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/nullsudo apt-get updatesudo apt-get install jenkins

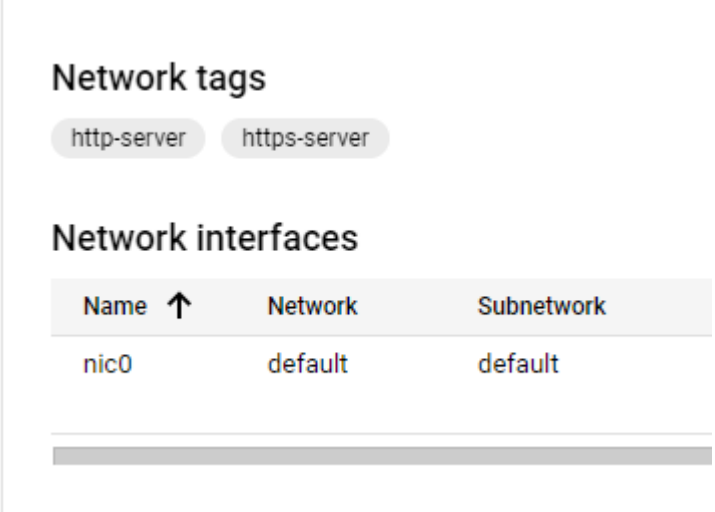
I have referred official Jenkins documentation to install the LTS version for Debian/Ubuntu. You can check the details here.

Post the above step, you can just run the update command again.

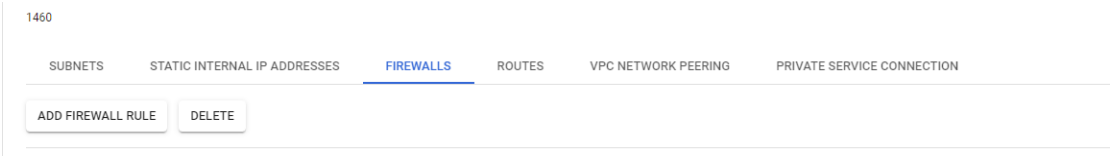
sudo apt update

Step 5: Adding Port 8080 in Firewall Rule for the instance

Now, switch back to your GCP Dashboard and click on the instance name which you had created. Then scroll down to the “Network Interfaces” section and then click on the “default” link available under the Network column.



Navigate to the Firewalls tab and click on the button “Add Firewall Rule”



Create Firewall Rule

In the Create Firewall rule page, specify the following details and click on Create button.

Name: jenkins-port
Direction of traffic: Ingress
Action on match: Allow
Targets: All instances in this network
Source IPv4 ranges: 0.0.0.0/0
Check on the TCP checbox and specify Ports: 8080

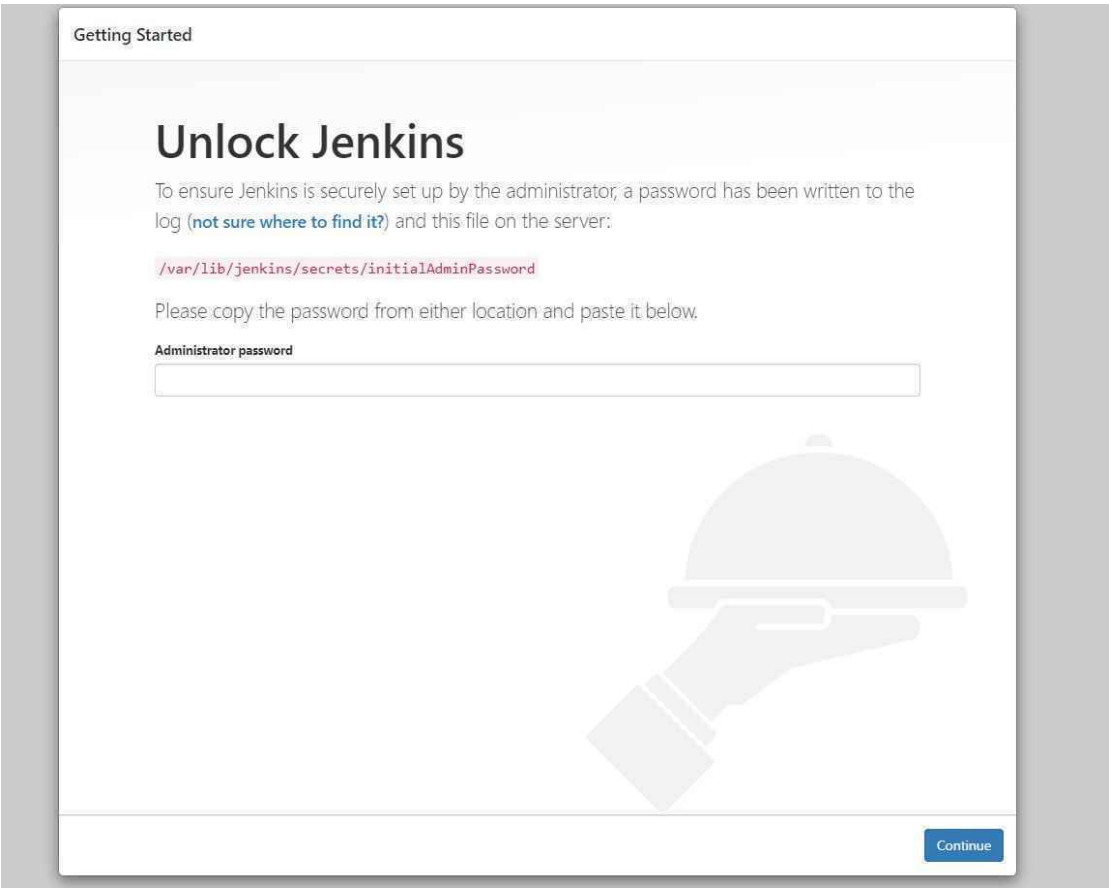
You can give any name for your firewall rule, what we are specifying in the rule is we are allowing incoming traffic on TCP port 8080 from any outside machine (open to the world) to our VM instance. For demo and learning purposes, this should not be an issue.

Note: In production or in real-world use cases, we will not be allowing traffic from any machine (0.0.0.0/0) and will restrict traffic only from a specific group of IP addresses or ranges which is being used in our company.

Step 6: Accessing Jenkins from the browser using port 8080

To access Jenkins, copy the external IP address of your VM instance from your dashboard and hit the below URL in your browser window: public-ip:8080/

For example, if your external IP address is 35.110.150.100 then your URL will be 35.110.150.100:8080 you should be able to view the below screen.



Enter administrator password

If you see this screen, congratulations you are almost done, and Jenkins is successfully installed. We just need to perform a few more steps to complete the setup process. Now, switch back to your SSH browser window and run the following command.

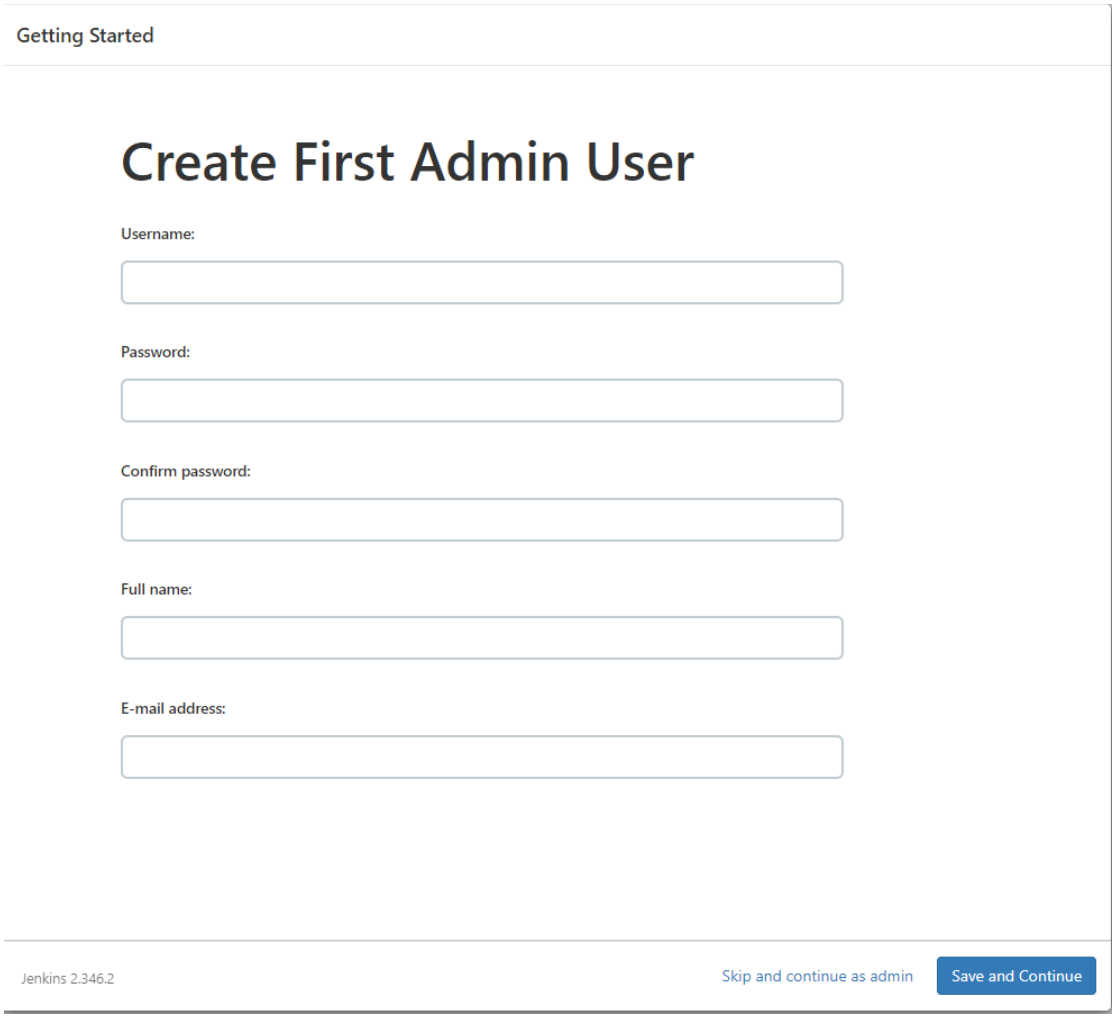
sudo cat /var/lib/jenkins/secrets/initialAdminPassword

The above command return an password string which you will need to paste it into the Administrator password textbox and then click on the Continue button to proceed.

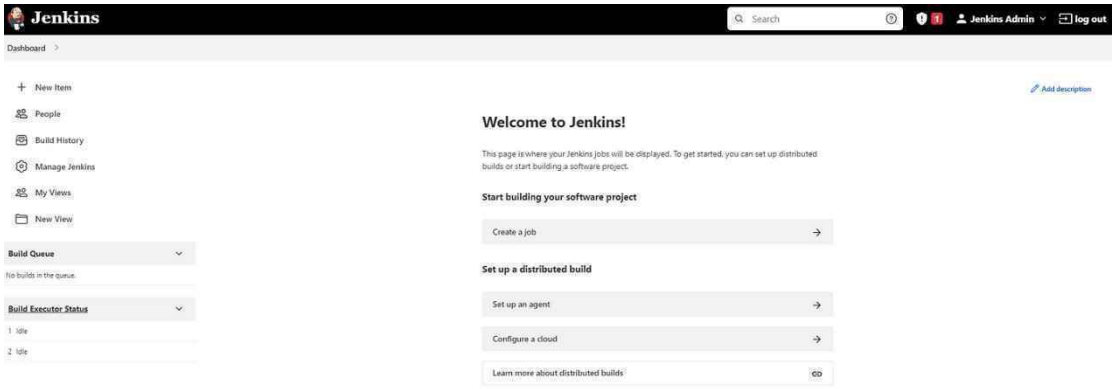
Next, Jenkins will prompt us to install plugins. For now, we will just click on Install suggested plugins and proceed with default plugins, this step might take a few minutes.

Install Jenkins plugins

Post plugin installation, you will have to Create an admin user for Jenkins, enter the following details: username, password, full name, email and click on the “Save and Continue” button.



On the next page, Instance configuration click on Save and Finish button, then click on Start using Jenkins button, you will be navigated to the Jenkins dashboard.



Jenkins Dashboard

Step 7: Configuring Java, Maven & Git details in Global Tool Configuration

In your Jenkins dashboard, click on the option “Manage Jenkins” which is available in the left pane, and then navigate to the “Global Tool Configuration” page available under “Manage Jenkins”

Here, under JDK, click on Add JDK button, un-check the checkbox “Install automatically” and specify the below values, then click on the Save button.

Configure JDK path

Below JDK, you will see the Git section wherein we need to specify “Path to Git executable”, and enter the following details.

Git path details

Finally, scroll down to the Maven section and click on the “Add Maven” button, specify the following details and Save the added details.

Maven_home path

We have completed configuring this instance which can be used as a Jenkins Controller node. In the next part, we will create a new Ubuntu instance configure it to be our Agent instance, and then later we will connect it to our Controller node to complete the Controller Agent setup.