**Install Jenkins**

**Prerequisites:**

1. Create EC2 Instance

* With Internet Access
* Security Group with Port 8080 for internet
* You may create a new security group.
* You should see preconfigured rule for SSH on port 22
* Add a new rule and choose your ‘Type’ to **Custom TCP,** port range should be **8080, and Source** should set to custom and in the space beside it, input **0.0.0.0/0**
* SSH into the machine once it is created
* Become the root by running this command: sudo su -

**Steps:**

1. Install Java

(First, check your Java version by running this command: java -version)

NOTE: By the default, Amazon Linux comes with the 1.7 version. It is recommended to remove 1.7 and install 1.8. To remove 1.7, run this command: yum remove java-1.7.0\*. To confirm its removal, run this command: java -version. If it says No such file or directory, it means the removal was successful.

* Now we can install Java 1.8 by running this command:

**yum install java-1.8\***

**You will be asked to confirm with this prompt [y/d/N]:** choose or type **y** (means yes)

**IMPORTANT:**

After Java is installed, we need to set HOME path for our Java. For that, we need to find where our JRE (Java Runtime Environment) exists. To find JRE, run this command:

**find /usr/lib/jvm/java-1.8\* | head -n 3**

The above command lets us know the path of the JRE. It should like this: **/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-0.43.amznl.x86\_64**. (And we can confirm by running **ls -l**. Press control C abort the confirmation and regain user control).

(You may now clear your screen by typing clear as we are about set our HOME path)

* To set home path, run this command: **cd ~** to return to the home directory of our user
* Then, run: vi .bash\_profile. [This command will return an output under which we should provide our HOME path. Click on i to invoke INSERT command

We should see something like this on the upper part of the terminal:

# Get the aliases and functions

If [ -f ~/.bashrc ]; then

. ~/ .bashrc

f1

#user specific environment and startup progress

**-------------------------------------------**

PATH=$PATH:$HOME/bin

export PATH

The red dashes above will not be shown on the terminal. It is in this documentation to let us know that we have to type something in here:

JAVA\_HOME=**path-we-retrived-using**:**find /usr/lib/jvm/java-1.8\* | head -n 3 (Remember we retrieved our JRE path earlier with the command: find /usr/lib/jvm/java-1.8\* | head -n 3)**

Therefore, assuming the path we retrieved was **/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-0.43.amznl.x86\_64,** what we would have and input into the red dashes space should be:

**JAVA\_HOME**=**/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-0.43.amznl.x86\_64**

Now,the **PATH=$PATH:$HOME/bin** argument needs to be extended to have the below:

**PATH=$PATH:$HOME/bin:$JAVA\_HOME** #(we are simply adding the HOME to our path)

Then, input (**:wq**)to save the file [Do not include the bracket]. #(:**wq** is used to save a file in Vim and quit and the editor)

**Once we save the file, the terminal would look like this:**

**[root@ip-172-31—31-222 ~]#**

**[root@ip-172-31—31-222 ~]# cd ~**

**[root@ip-172-31—31-222 ~]# vi .bash\_profile**

**[root@ip-172-31—31-222 ~]# ….**

We have to type the following command on the red dotted lines above (Remember, no red dotted lines in the terminal. It is just in this documentation to let you where to type):

**echo $JAVA\_HOME**

When we do, the whole lines will look like this:

**[root@ip-172-31—31-222 ~]#**

**[root@ip-172-31—31-222 ~]# cd ~**

**[root@ip-172-31—31-222 ~]# vi .bash\_profile**

**[root@ip-172-31—31-222 ~]# echo $JAVA\_HOME**

When we press **ENTER,** we will have the following:

**/usr/lib/jvm/jre**

Now, this does not show the full path because we need to log out and log in for it to take effect.

So, we look out by running the exit command. At this point, our terminal should look like this:

**[root@ip-172-31—31-222 ~]#**

**[root@ip-172-31—31-222 ~]# cd ~**

**[root@ip-172-31—31-222 ~]# vi .bash\_profile**

**[root@ip-172-31—31-222 ~]# echo $JAVA\_HOME**

**/usr/lib/jvm/jre**

**[root@ip-172-31—31-222 ~]# exit**

**Logout**

**[ec2-user@ ip-172-31—31-222 ~]$**

Then, again, we login as root with **sudo su –**

Now, execute the same command: **echo $JAVA\_HOME** and you will see the latest path

**Now we install Jenkins:**

1. Go to Jenkins.io/download (this takes us to Jenkins’ website)
2. On the page, we will be presented with two types of release: long term support (LTS) or weekly. It is best practice to use LTS.
3. Under LTS, you will see compatible operating systems. Choose the one that Linux supports unless you have launched a Windows virtual machine. I recommend choosing CentOS **/Fedora/Red Hat**

This will take you to a page that gives the instructions on what to do to create Jenkins.

The first two commands we need to run (one before the next) are:

**sudo wget -O /etc/yum.repos.d/jenkins.repo** [**https://pkg.jenkins.io/redhat-stable/jenkins.repo**](https://pkg.jenkins.io/redhat-stable/jenkins.repo)

**sudo rpm --import** [**https://pkg.jenkins.io/redhat-stable/jenkins.io.key**](https://pkg.jenkins.io/redhat-stable/jenkins.io.key)

NOTE: If you've previously imported the key from Jenkins, the rpm --import will fail because you already have a key. Please ignore that and move on.

Then, run the following command:

**yum install jenkins**

You will get this prompt:

**Is this ok [y/d/N]:**

Type **y**

**It will run and we will receive a confirmation that our Jenkins installation is successful.**

Now run the following: **service jenkins status**

This will show stopped. It is set to show this by default.

So, run this command: **service jenkins start**

And you will see it saying **starting Jenkins**

Now, when yourun **service jenkins status** again, you should have:

**jenkins is running….**

Now, to access the Jenkins application from the browser, we go to our EC2 Instance and copy public IP address, paste it on a browser include **:8080**

It should bring a page that asks for **Administrator password.**

Copy the link in red on this page that says:

**/var/lib/Jenkins/secrets/initialAdminPassword**

Cat this file by pasting it with the cat command:

**Cat /var/lib/Jenkins/secrets/initialAdminPassword** and you will see the auto generated password. Copy this password.

Go back to the browser and paste this browser. Click continue. It will take you to a page that asks you install plugins. (You may skip this stage for now) Now your Jenkins is ready.

Click on start using Jenkins and you will be taken to your Jenkins console

Now, we need to change our password. Inside the Jenkins console, go to admin, then, go to Configure, then find Password and change your password, click apply and save. Upon successfully changing your password, you will get logged out and the browser will bring HTTP ERROR 403

Copy the public IP address of your EC2 Instance and include :8080 into a browser. It will bring a page to sign in. Your username by default is admin and you will enter the password you just created and click **Sign in** and it will take you to your Jenkins console**.**

Now you are in your Jenkins console homepage. However, you need set your Java HOME path. For that, you go to **Manage Jenkins** on your Jenkins console homepage and find **Global Tool Configuration.** Click on it, scroll down and find **JDK,** click on **Add JDK** and you will Name and **JAVA\_HOME** pop up with a space for each. In the Name space, enter **JAVA\_HOME. And in the JAVA\_HOME space,** enter your version of **/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-0.43.amznl.x86\_64.** You can also retrieve it from your EC2 Instance Linux terminal by typing **echo $JAVA\_HOME.** Click **Apply and Save.**

**RUN FIRST JENKINS JOB**

(Please, know that you will always login to your Jenkins console with your EC2 public IP together with :8080 pasted into your browser)

* On your Jenkins page, click on **New Item.** Enter a name for your Jenkins project. Enter a description for it.
* Scroll down and leave Source Code Management as **None** because we have not installed any plugins for it.
* Scroll down to **Build and** click on the **Add build step** button and choose **Execute Shell** because our target system is Linux. When we click this, it will ask for a command. In the big space provided, we can input anything of our choice, i.e., echo “welcome to DevOps project”. Click **Apply** and **Save**.
* Now you can find **Build Now on the** right-hand side of the page**. Click on it,** and under **Build History** we should a small rolling ball**.** If it is blue, it means it is successful. clicking on the ball takes you to the Console Output.

**Install and setup Git on Jenkins**

* Go to your Jenkins server on your Linux terminal. You may change the host name to make it easy for you to identify by running this command: **hostname Jenkins.**
* Run **sudo su –**
* Type: **yum install git -y**
* Go back to your Jenkins console, find **Manage Jenkins. And from there, find Manage Plugins,** find and click on **Available,** scroll down and find **GitHub,** select **GitHub** and click on **Install without restart. This will install GitHub and other dependency plugins.**
* Once successful, go back to your Jenkins console home page and set up **Git** in the **Global Tool Configuration**. So, we go to Manage Jenkins and **Global Tool Configuration.** Now, when we scroll down, we will see **Git Installations** because we have just installed the plugins. We would need to put a name in the **Name** space and a path in the **Path to Git executable** space. But we can just name it **github**, and input **git** (if not there by default) in the **Path to Git executable** space, then click on **Apply** and **Save**.