# DevOps Pipeline checklist:

## Setting Up SCM

As a team you should all set up a github page for your project. Make sure you make the page and have everyone on the team connect and download its contents. This will ensure you are all using the same project!

## Start up your RDS

With AWS, go to your RDS section and launch a new RDS. Make sure you check the box for free tier eligibility. You can use Enterprise edition Oracle for the database. Ensure that port 1521 is open for connections.

## Start up your EC2 Continuous Integration Server

Now it is time to start up an EC2 to house your continuous integration tools. The first step would be to make a t2.micro EC2. Through configuration you want to make sure that the following ports are open to all ips:

-22: Used for SSH and file transfers

-8080: Port to access Jenkins gui

Once you configure your server, it will have you download a private key for accessing it. (a pem file)

Save this file somewhere safe as you will need it to access your server.

## Access your ec2 with SSH

**WINDOWS:**

Utilize Puttygen to convert your .pem key to a .ppk.  
Simply load the .pem key and click “save private key”. Be sure the name ends in .ppk.  
Next, with Putty, type in the following url to connect to:

[ec2-user@xxx.xxx.xxx.xxx](mailto:ec2-user@xxx.xxx.xxx.xxx) <-where the xxx represent the **PUBLIC** up address of your EC2

Next, from the side bar, navigate to: connection>SSH>auth and add your .ppk key.  
Save the configurations and connect. (You will be warned the first time you connect. Just click OK.)

## Install all tools necessary for your server.

**JAVA:  
-**Download a jdk to use on your ec2: [The Linux x64 rpm file will suffice](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html).  
-Transfer the .rpm file to your ec2:

**WINDOWS:**   
 You will need to use the cmd to do this:  
 “Directory/To/pscp” -I “directory/to/your/.ppk key” “directory/to/file/to/be/transferred” “ec2-user@xxx.xxx.xxx.xxx:optionalFolder/transferredFileName”

NOTE: If you wish to not transfer the file over to a specific folder, and simply have it transfer to your hme directory in the EC2, then remove ‘optionalFolder/’ from above cmd example.

**MAC:**

Using the shell, use the following command to transfer files:

scp -I “directory/to/.pemKey” “directory/for/file/to/be/transferred” ec2-user@xxx.xxx.xxx.xxx:~/optionalFolder

NOTE: optionalFolder is only for when you don’t want to transfer the file directly to the home directory.

After transferring the file, invoke the following command on the rpm file from the ec2:

sudo rpm -ivh jdk8.rpm 🡨 clarify file name…

Add a path for JAVA\_HOME in your .bash\_profile file:

export JAVA\_HOME=”usr/java/jdk\_8/ 🡨 check folder name

Save the file and reconnect to EC2. Ensure the following command works:

cd ${JAVA\_HOME}

# TOMCAT

Invoke the following commands to get tomcat on your ec2. (Do this within the ec2):

wget apache.mirrors.ionfish.org/tomcat/tomcat-7/v7.0.82/bin/apache-tomcat-7.0.82.tar.gz

tar -zxvf apache-tomcat.tar.gz 🡨 Filename may differ……

Add a path for CATALINA\_HOME in your .bash\_profile file:

export CATALINA=”home/ec2-user/tomcat\_folder/ 🡨 check folder name

Save the file and reconnect to EC2. Ensure the following command works:

cd ${CATALINA\_HOME}

Navigate to: tomcat\_folder/conf/ and edit the tomcat-users.xml file. Add the following code:

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<user username="tomcatgui" password="password" roles="manager-gui"/>

<user username="tomcatscript" password="password" roles="manager-script"/>

**NOTE: You can change username and password how you like, but anything related to role information MUST MATCH.**

## Maven and GIT

Invoke the following commands within EC2 to install maven:

sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo

sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo

sudo yum install -y apache-maven

**Ensure maven installed correctly by typing the following:**

mvn –version

For git, simply invoke:

sudo yum install git

# Jenkins

There are two ways to install Jenkins to your EC2. Step one is to download the [jenkins.war](https://jenkins.io/) (download generic java package) file from their website and transfer it over to the EC2 the same way you did it for java as mentioned above.

Once it is transferred, run the following command on it: java -jar jenkins.war

Then access Jenkins at public.ip:8080 with a browser. You SHOULD get an unlock screen.

If note, there is method 2:

From the ec2 invoke the following commands:

sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo

sudo rpm --import <https://jenkins-ci.org/redhat/jenkins-ci.org.key>

sudo yum install jenkins -y

sudo service jenkins start

Then connect to public.ip:8080