Jenkins and Tomcat

- Prepare your instance
  - Step 1. Set Up your machine
  - Step 3. Install Jenkins and Maven.
  - Step 4. Check admin Jenkins password.
  - Step 5. Navigate to http://your_ip:8080 and provide password obtained from previous step:
  - Step 6. Install suggested plugins:
  - Step 7. Plugins installations progress
  - Step 8. Finish installation by provisioning Jenkins admin user:
  - Step 9. Jenkins is installed and ready for further configuration.
  - Step 10. Login using your account:
- Configuring plugins and tools
    - Step 1. Install JDK using the next script:
    - Step 2. We do need to do a little configuration.
    - Step 3. Configure plugins installing 'Capitomcat Plugin' and 'Artifact Deployer'
  - Your First Jenkins Build Job
    - Step 4. Once you click on OK, Jenkins will display the project configuration screen.
- Remote deployment to Tomcat 8 container
  - Step 1. Install Jenkins plugin
  - Step 2. Install Tomcat on your machine
  - Step 3. Adapt your /opt/tomcat8/conf/tomcat-users.xml as the following:
  - Step 4. Run tomcat and go to http://your_ip
  - Step 5. Configure Jenkins job. Final step

Prepare your instance

Step 1. Set Up your machine

- Log in to AWS console, go to EC2 service.
- You need a bit of RAM to set up this task, so let's use more powerful machine (t3.small will be enough).
- Choose your instance (it should have stopped state), Actions Instance Settings Change Instance type. Choose t3.small.
- Start your machine and check that inbound connection on ports 22, 80 and 8080 in your machine security group are allowed.

Step 2. As prerequisite for Jenkins download and install Java 8 binaries:

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt update
sudo apt install -y oracle-java8-installer
```

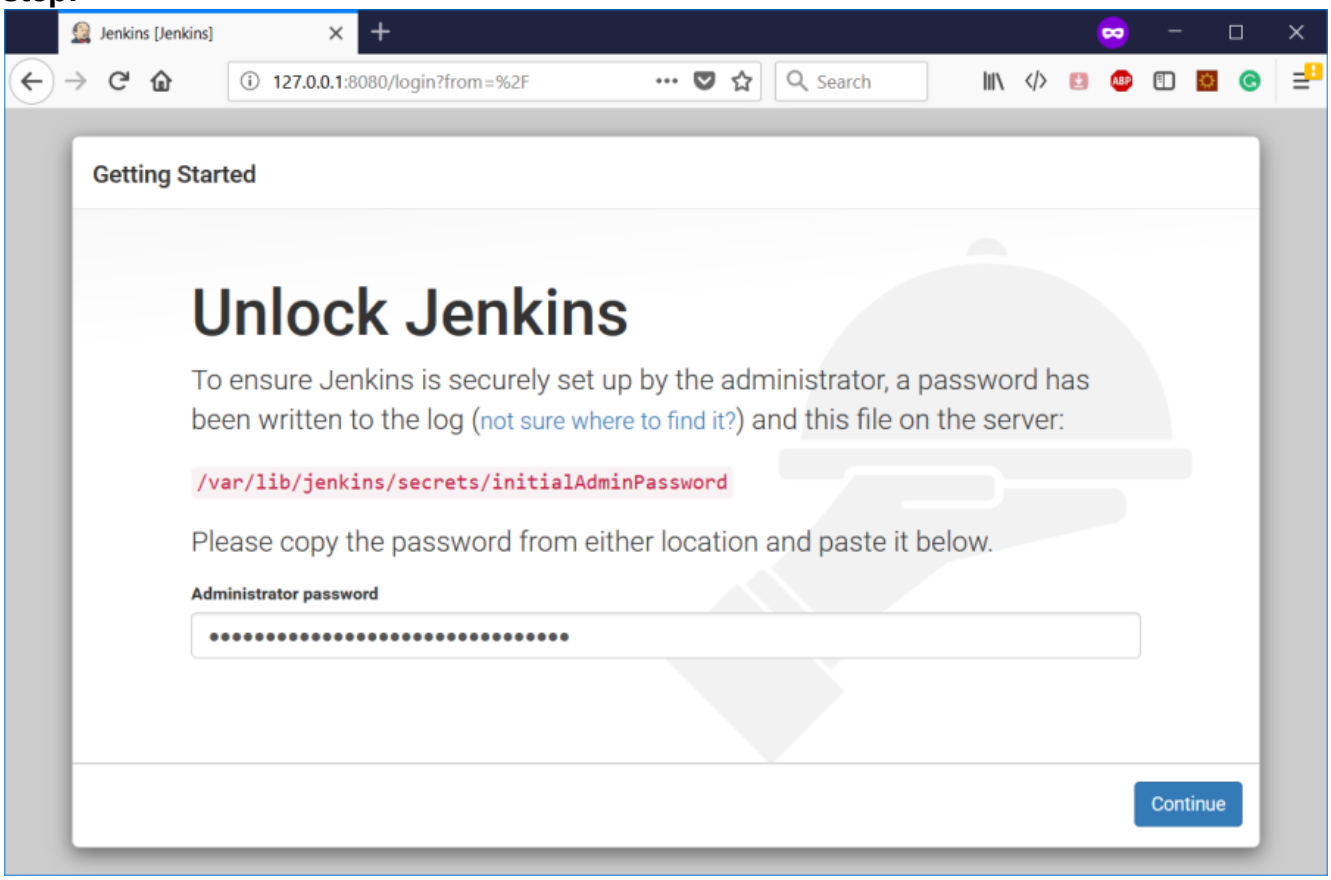
Step 3. Install Jenkins and Maven.

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key
add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt
/sources.list.d/jenkins.list'
sudo apt update
sudo apt install jenkins maven
sudo systemctl start jenkins
sudo systemctl status jenkins
```

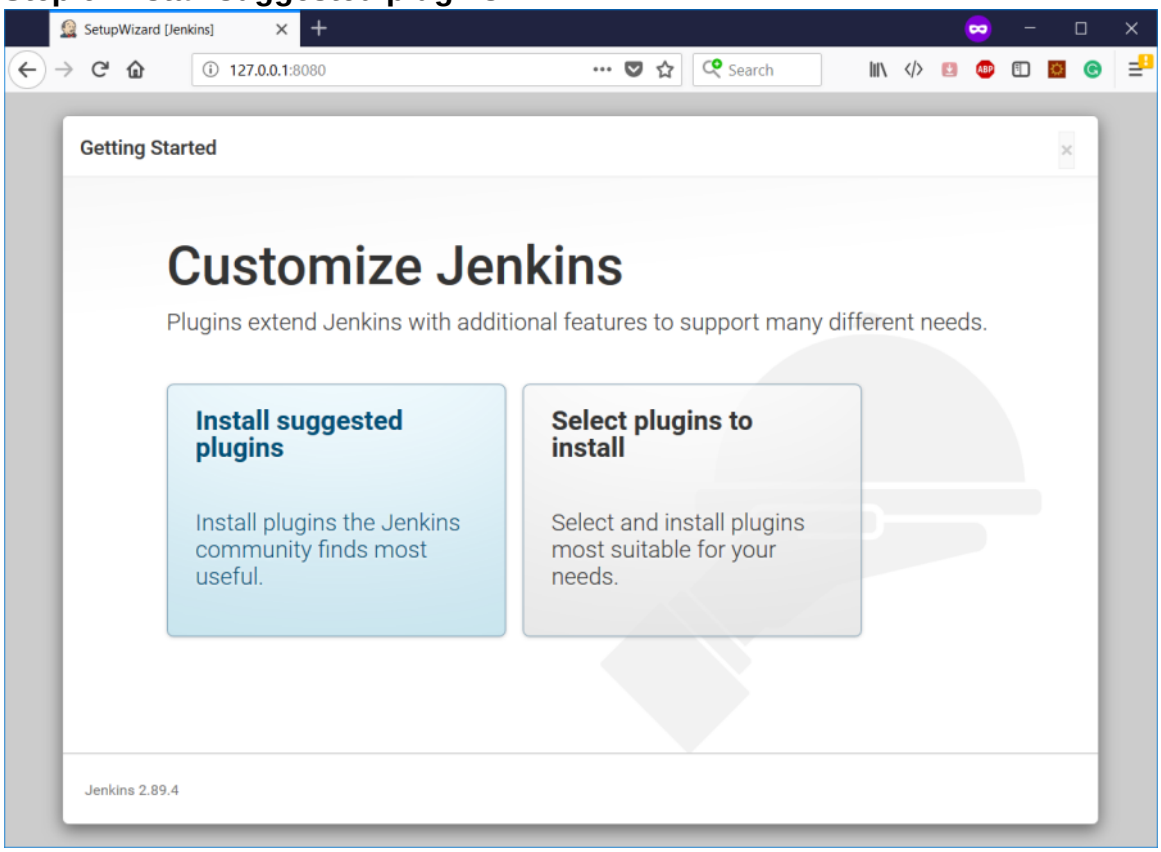
Step 4. Check admin Jenkins password.

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

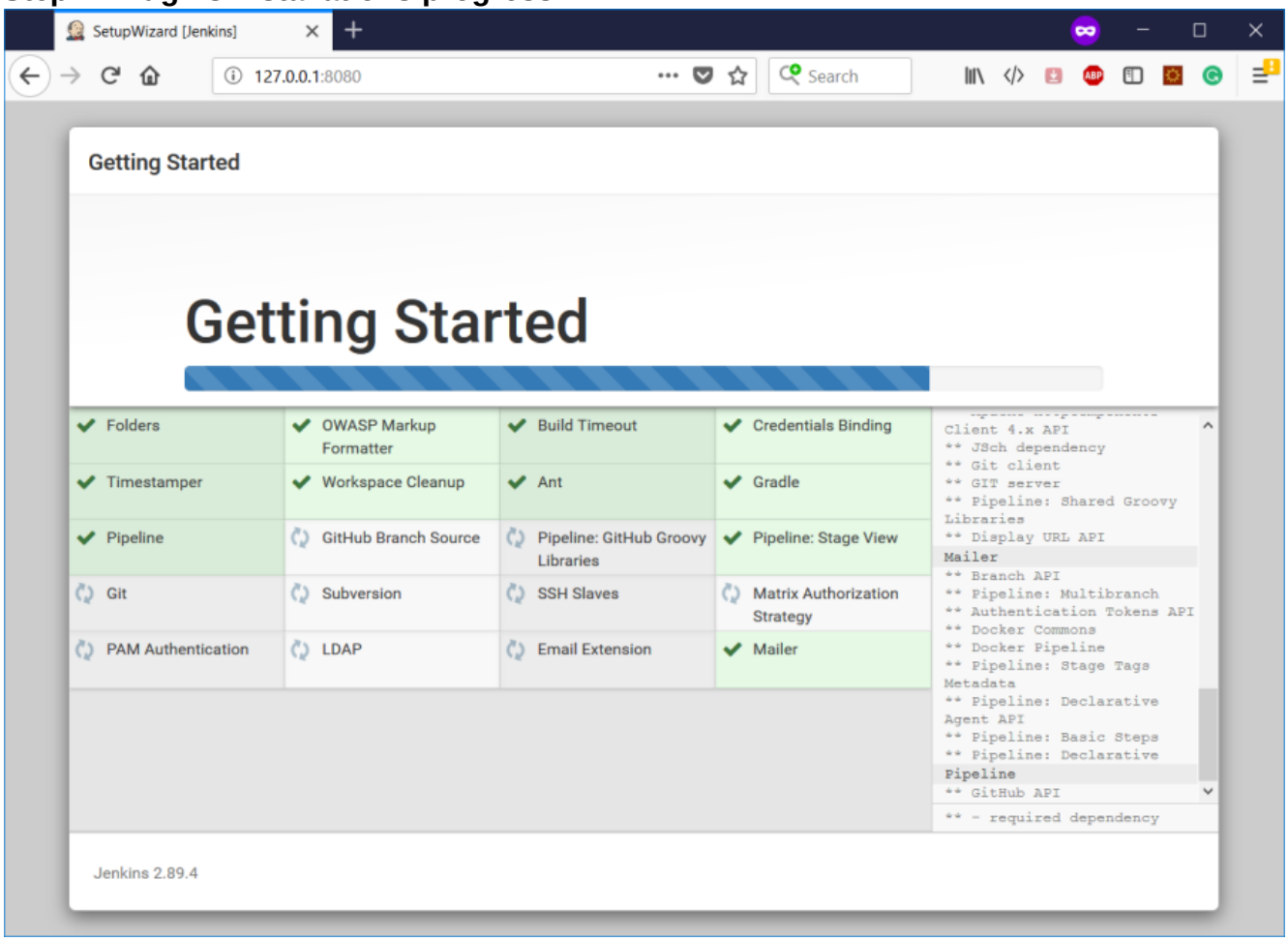
Step 5. Navigate to *http://your_ip:8080* and provide password obtained from previous step:



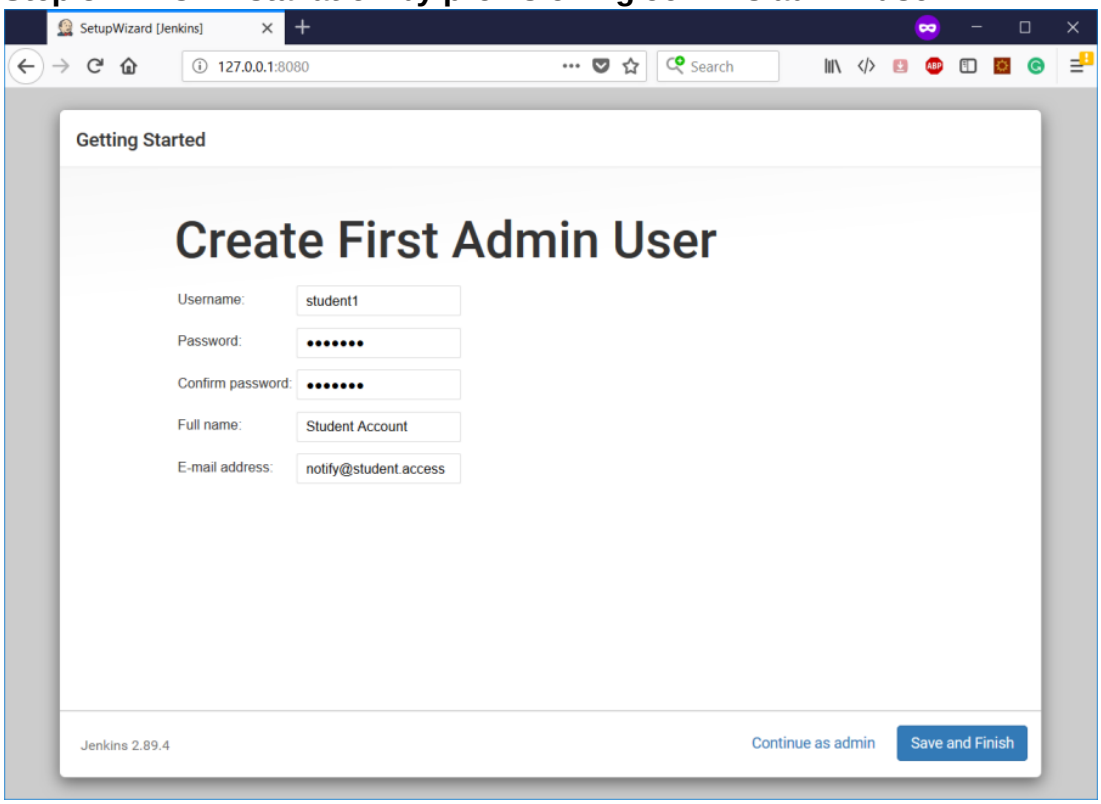
Step 6. Install suggested plugins:



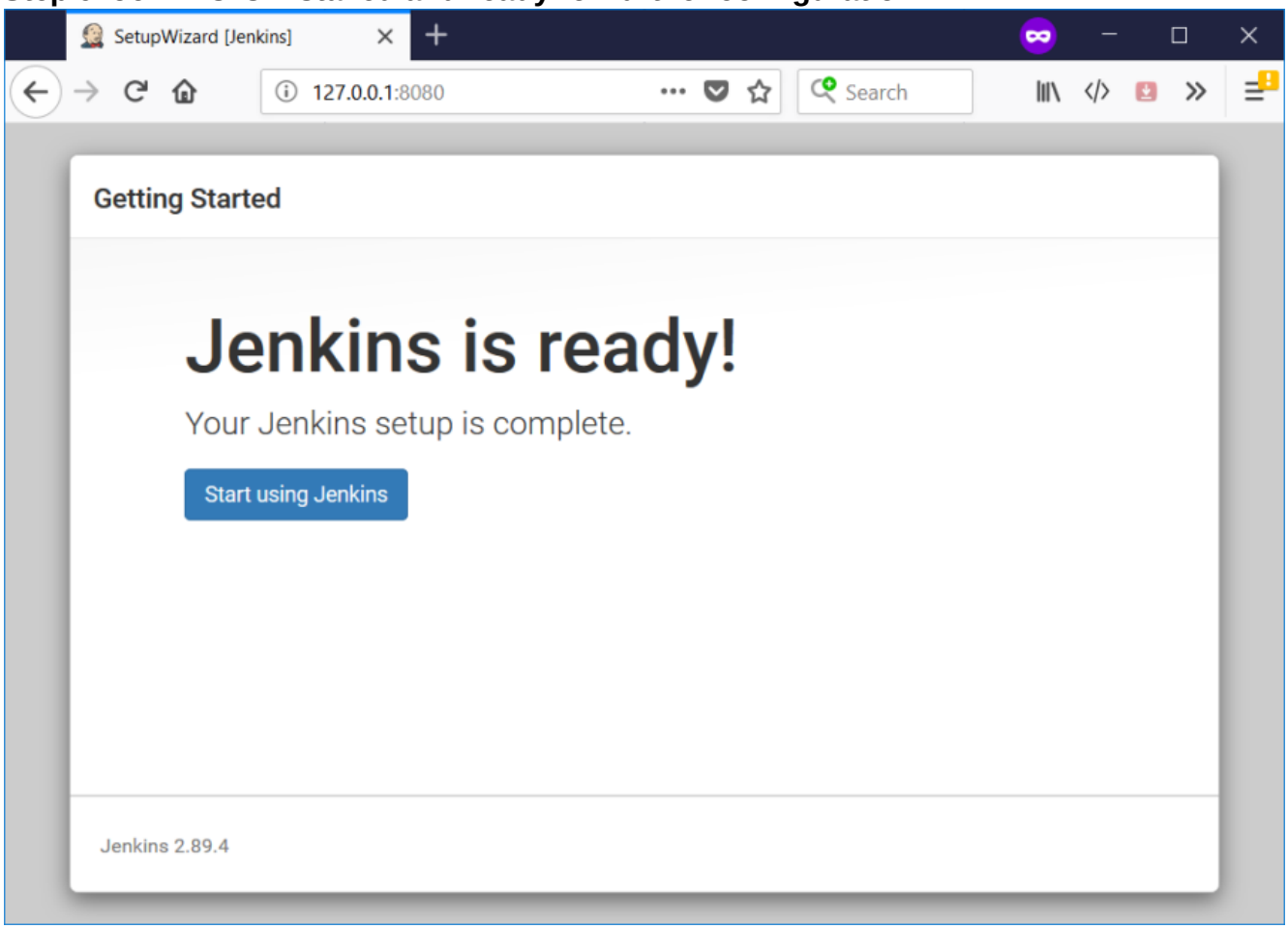
Step 7. Plugins installations progress



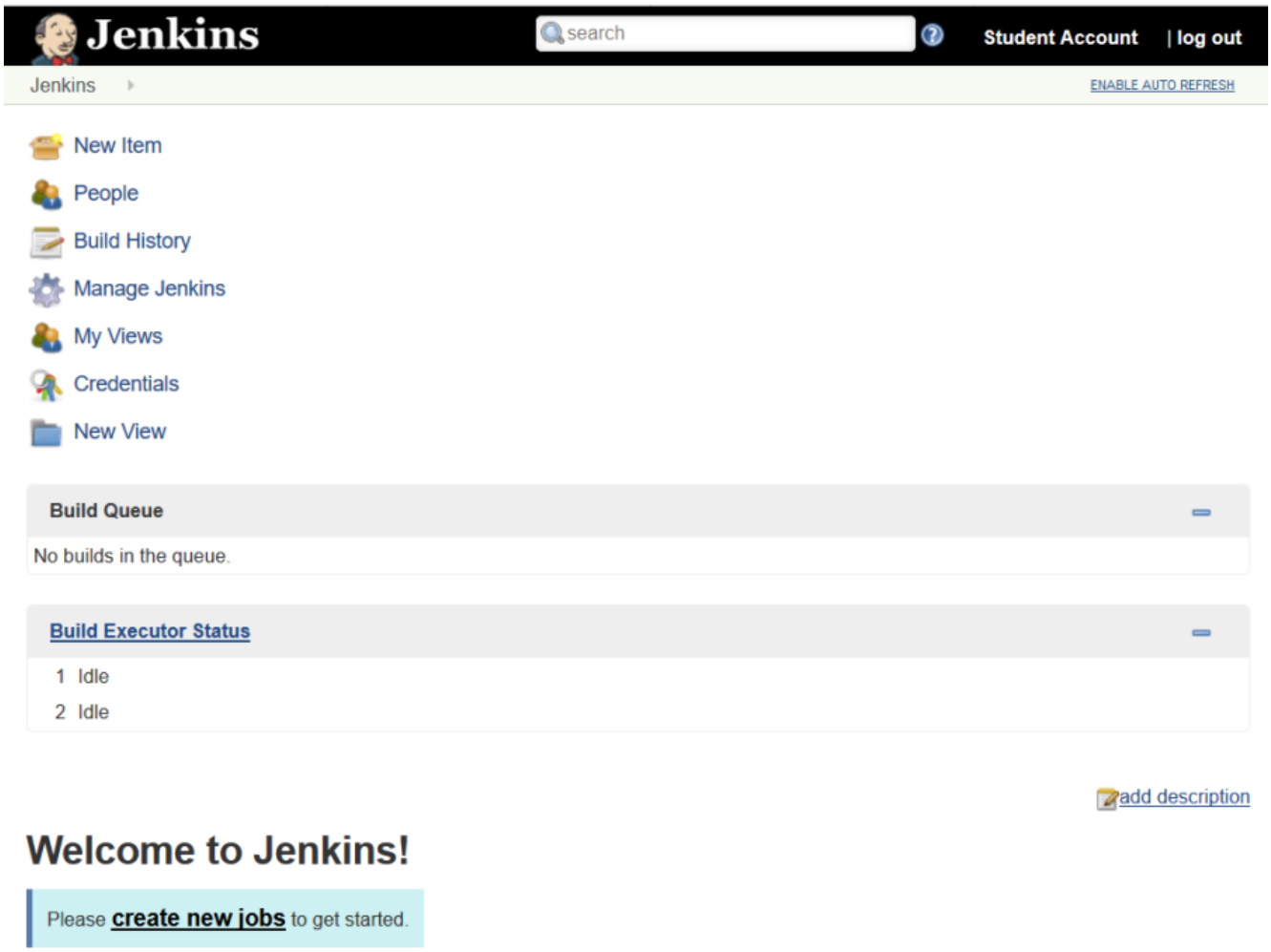
Step 8. Finish installation by provisioning Jenkins admin user:



Step 9. Jenkins is installed and ready for further configuration.



Step 10. Login using your account:



Configuring plugins and tools

Step 1. Install JDK using the next script:

Check new JDK on https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html (for Lector)

```
#!/bin/bash

sudo rm -rf /usr/java

sudo wget --no-cookies \
--no-check-certificate \
--header "Cookie: oraclelicense=accept-securebackup-cookie" \
https://download.oracle.com/otn-pub/java/jdk/8u201-b09
/42970487e3af4f5aa5bca3f542482c60/jdk-8u201-linux-x64.tar.gz \
-O /opt/jdk-8-linux-x64.tar.gz

cd /opt && sudo tar zxf jdk-8-linux-x64.tar.gz && sudo mkdir /usr/java/ &&
```

```
sudo mv /opt/jdk1.8.0_201 /usr/java && sudo rm /opt/jdk-8-linux-x64.tar.gz

#Update alternatives section
sudo update-alternatives --install /usr/bin/java java /usr/java/jdk1.8.0
_201/jre/bin/java 20000
sudo update-alternatives --install /usr/bin/jar jar /usr/java/jdk1.8.0_201
/bin/jar 20000
sudo update-alternatives --install /usr/bin/javac javac /usr/java/jdk1.8.0
_201/bin/javac 20000
sudo update-alternatives --install /usr/bin/javaws javaws /usr/java/jdk1.
8.0_201/jre/bin/javaws 20000
sudo update-alternatives --set java /usr/java/jdk1.8.0_201/jre/bin/java
sudo update-alternatives --set javaws /usr/java/jdk1.8.0_201/jre/bin/javaws
sudo update-alternatives --set javac /usr/java/jdk1.8.0_201/bin/javac
sudo update-alternatives --set jar /usr/java/jdk1.8.0_201/bin/jar

#check version
java -version
```

Result should look like:

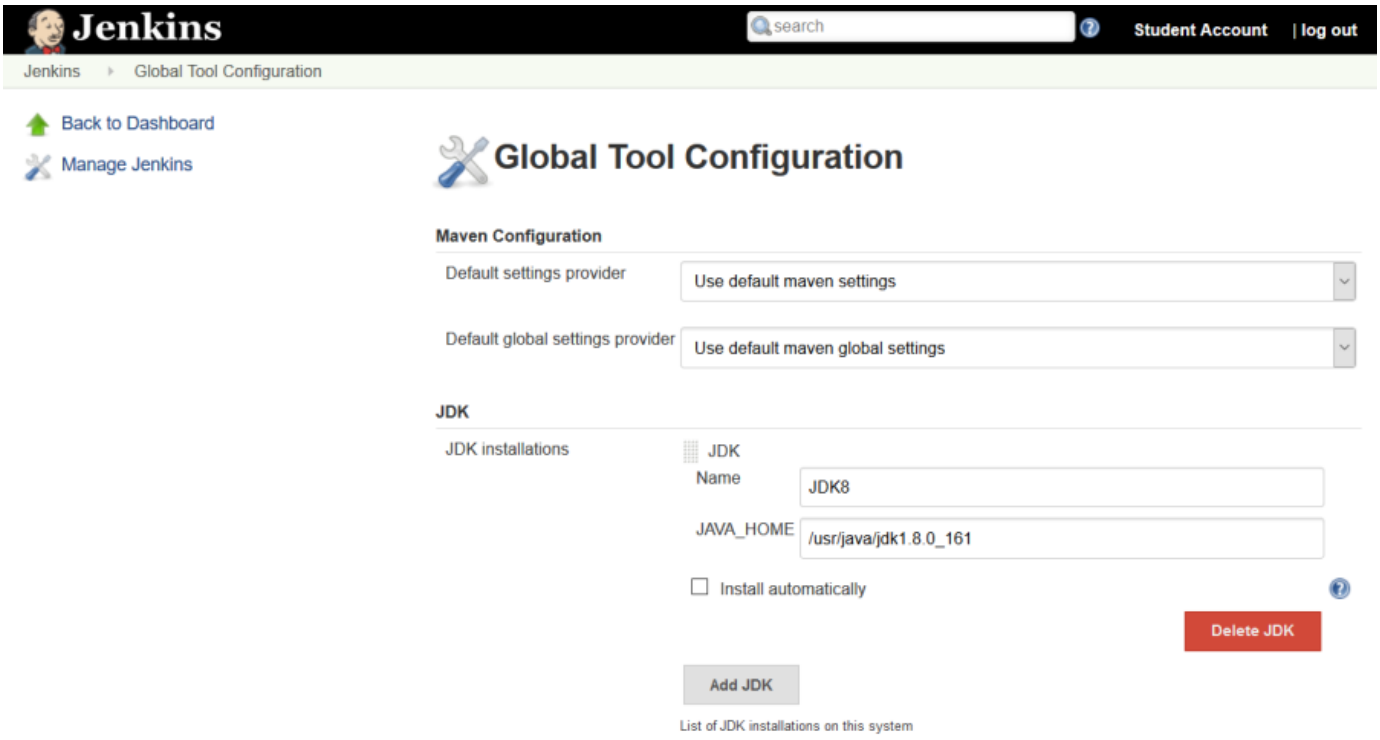
```
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

Step 2. We do need to do a little configuration.

More precisely, we need to tell Jenkins about the build tools and JDK versions we will be using for our builds.

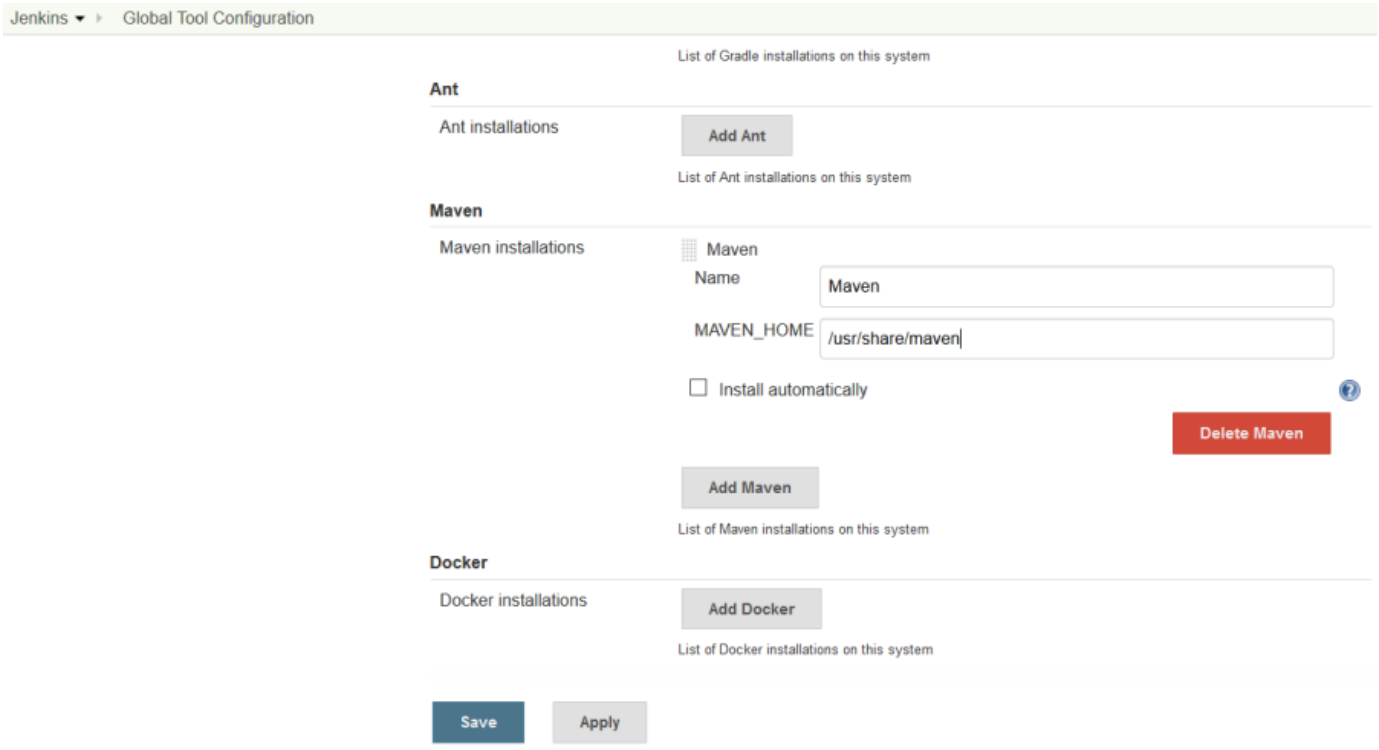
Go to Manage Jenkins Global Tool Configuration (Add JDK, remove checkbox "Install automatically")

```
Name: JDK8
JAVA_HOME: /usr/java/jdk1.8.0_201
```



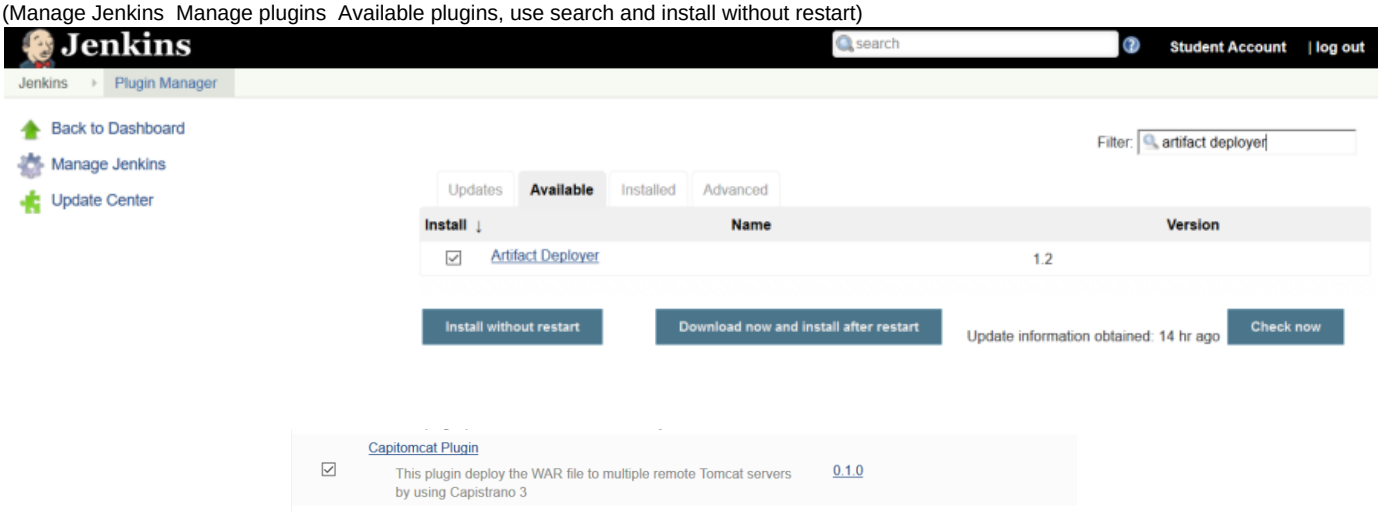
Configure path for Maven applying changes and saving config: (Add Maven, remove checkbox "Install automatically")

```
Name: Maven
MAVEN_HOME: /usr/share/maven
```



Step 3. Configure plugins installing 'Capitomcat Plugin' and 'Artifact Deployer'

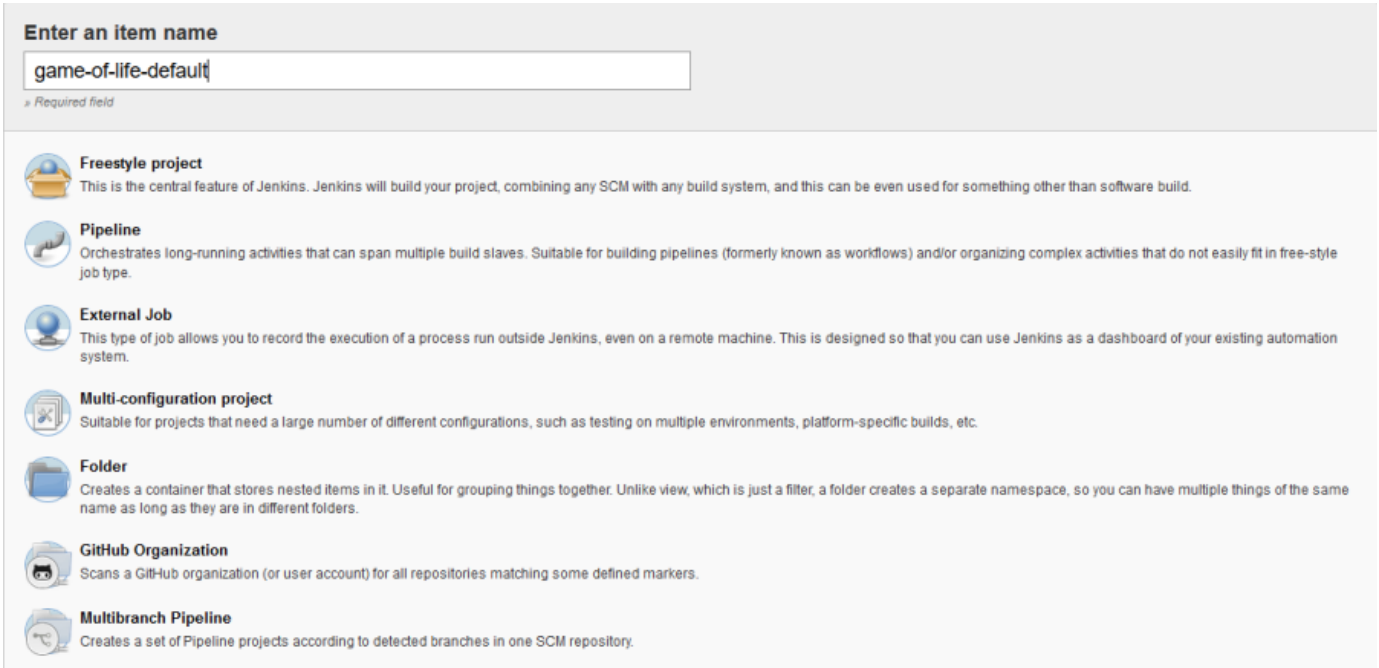
(Manage Jenkins Manage plugins Available plugins, use search and install without restart)



Your First Jenkins Build Job

Build jobs are at the heart of the Jenkins build process. You can think of a Jenkins build job as a particular task or step in your build process. This may involve simply compiling your source code 22 and running your unit tests. Or you might want a build job to do other related tasks, such as running your integration tests, measuring code coverage or code quality metrics, generating technical documentation, or even deploying your application to a web server.
You'll also need to give your build job a sensible name. In this case, call it game-of-life-default, as it will be the default CI build for our Game of Life project.

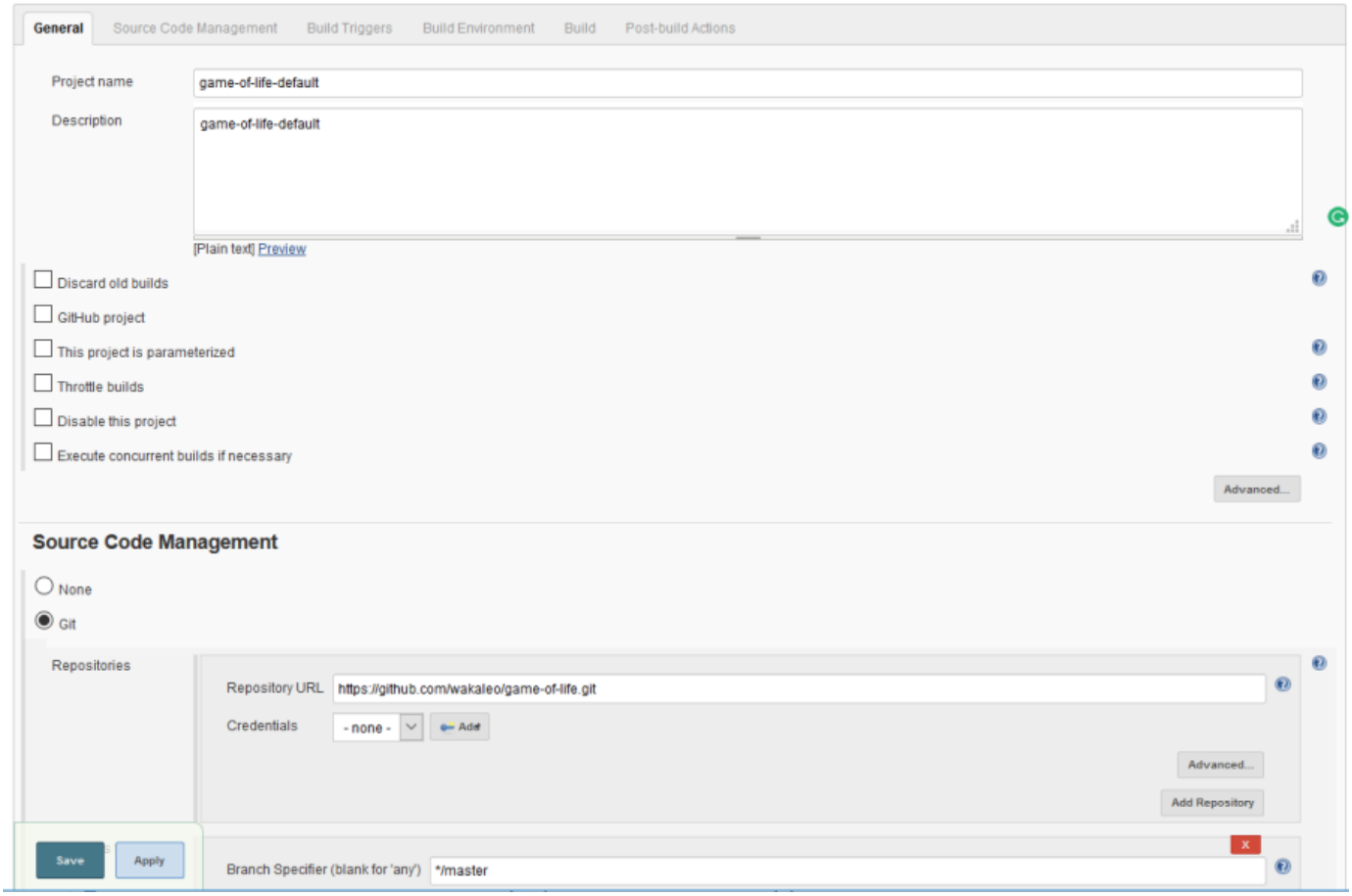
Enter a name of the job and choose Freestyle project



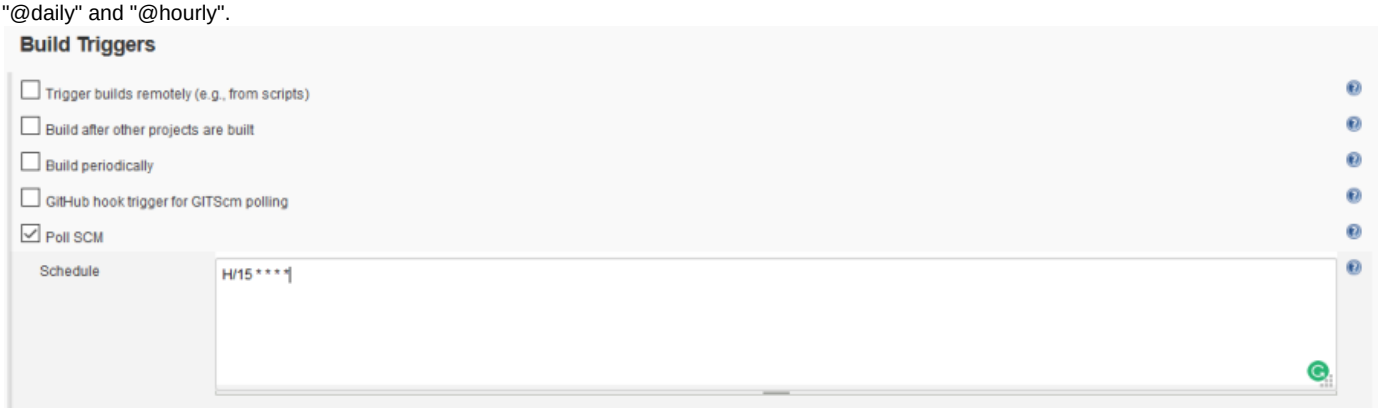
Step 4. Once you click on OK, Jenkins will display the project configuration screen.

Define git repo as:

```
https://github.com/wakaleo/game-of-life.git
```

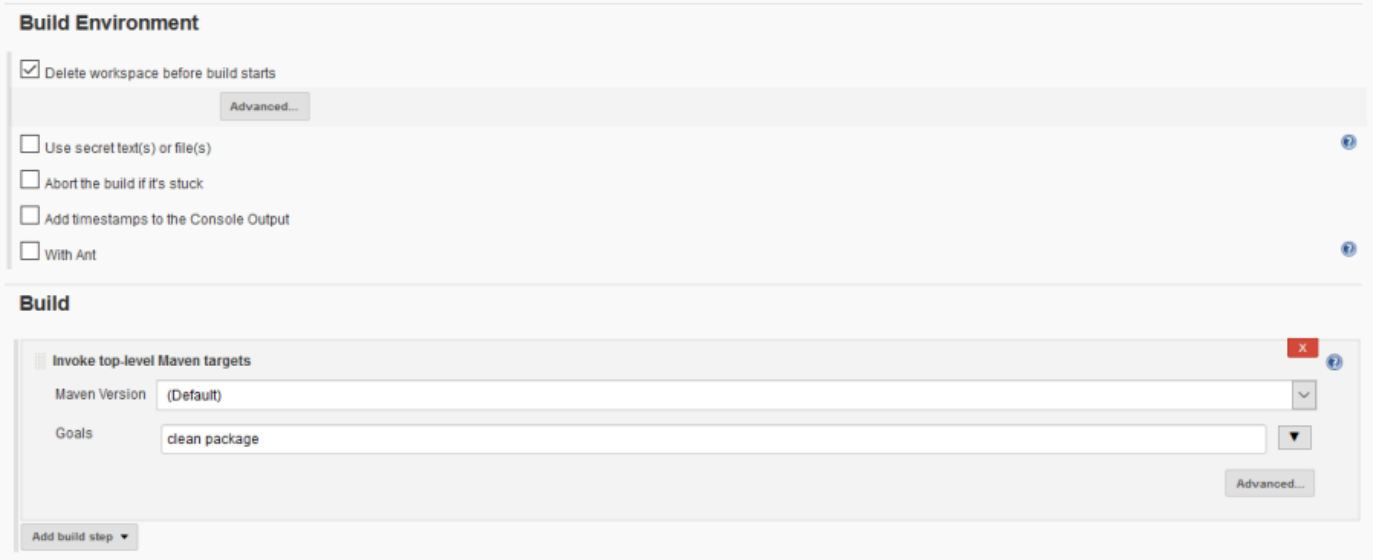


Pick the Poll SCM option and enter "H/15 * * * *" (that's five asterisks separated by spaces) in the Schedule box. Jenkins schedules are configured using the cron syntax, well-known in the Unix world. The cron syntax consists of five fields separated by white space, indicating respectively the minute (0–59), hour (0–23), day of the month (1–31), month (1–12) and the day of the week (0–7, with 0 and 7 being Sunday). The star is a wildcard character which accepts any valid value for that field. So five stars basically means "every minute of every hour of every day." You can also provide ranges of values: "* 9-17 * * *" would mean "every minute of every day, between 9am and 5pm." You can also space out the schedule using intervals: "*/5 * * * *" means "every 5 minutes," for example. Finally, there are some other convenient short-hands, such as "@daily" and "@hourly".



For now, we just want to run a simple Maven build. Scroll down to the Build section and click on the "Add build step" and choose "Invoke top-level Maven targets".

Then enter "clean package" in the Goals field. If you are not familiar with Maven, this will delete any previous build artifacts, compile our code, run our unit tests, and generate a JAR file.



Click save and try to build your Job.

Remote deployment to Tomcat 8 container

Step 1. Install Jenkins plugin

Open your favorite browser and navigate to Jenkins. Log in and select "Manage Jenkins" followed by "Manage Plugins". Select the "Available" tab, locate the "Deploy to container" plugin and install it.

Step 2. Install Tomcat on your machine

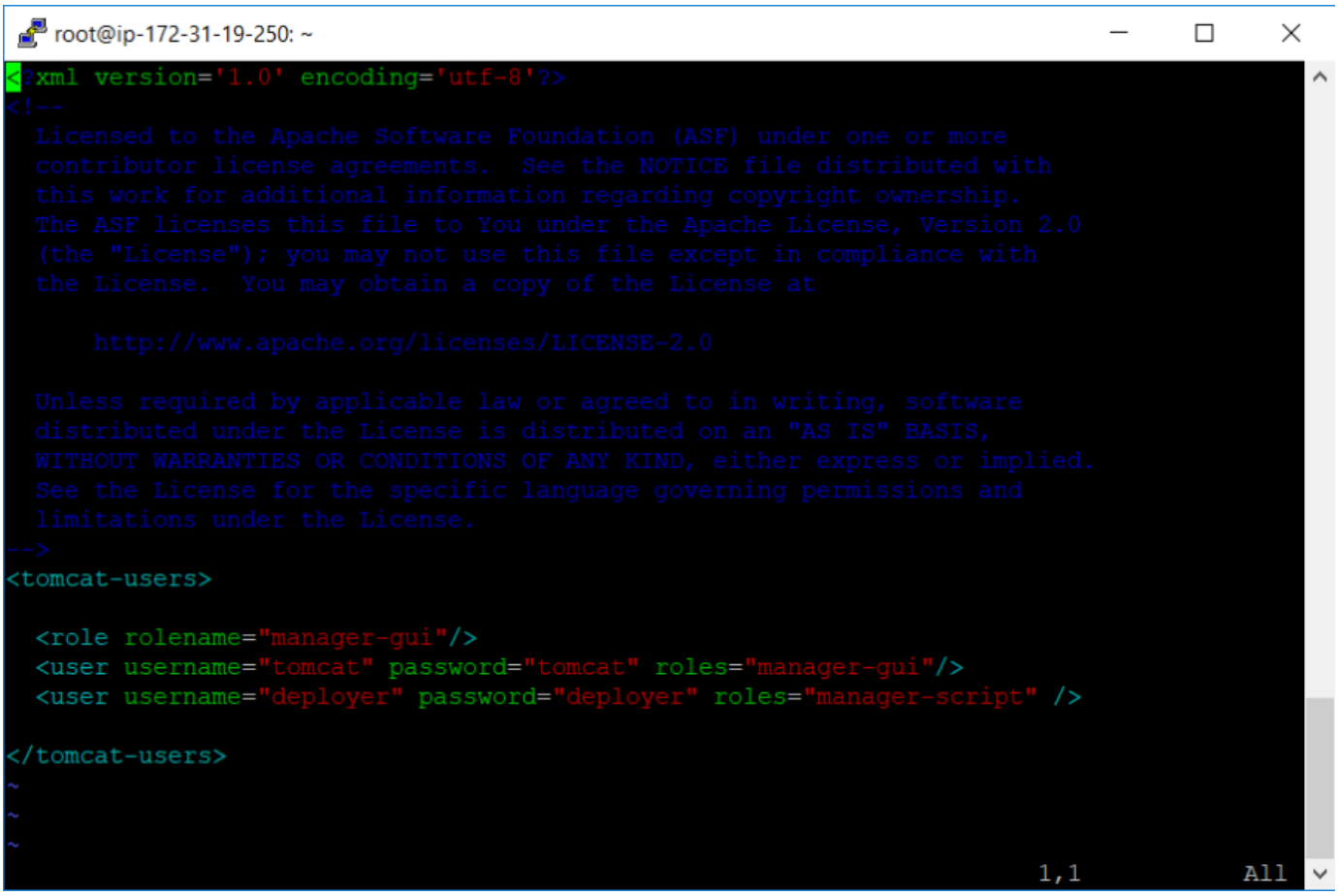
```
#!/bin/bash

sudo wget -c -q "http://apache.volia.net/tomcat/tomcat-8/v8.5.38/bin
/apache-tomcat-8.5.38.tar.gz" -O /opt/apache-tomcat-8.5.38.tar.gz
cd /opt && sudo tar zxf apache-tomcat-8.5.38.tar.gz && sudo mv apache-
tomcat-8.5.38 tomcat8 && sudo rm apache-tomcat-8.5.38.tar.gz
sudo setfacl -m u:jenkins:rwx /opt/tomcat8/webapps
sudo wget "https://bitbucket.org/toorroot/tomcat7/raw
/6284375e551ff0c97986d1f103e45183e52ccfc9/module2_install/tomcat-users.
xml" -O /opt/tomcat8/conf/tomcat-users.xml
sudo sed -i 's/Connector port=\"8080\"/Connector port=\"80\"/g'  /opt
/tomcat8/conf/server.xml
```

Step 3. Adapt your */opt/tomcat8/conf/tomcat-users.xml* as the following:

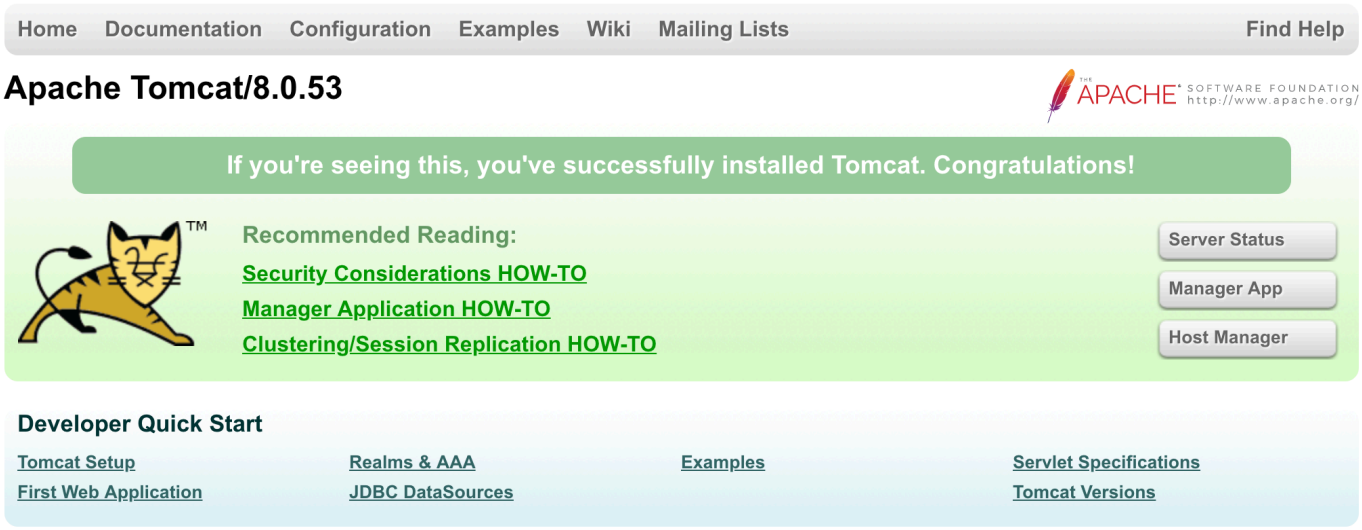
```
sudo nano /opt/tomcat8/conf/tomcat-users.xml
```

```
Add in tomcat users:
<user username="deployer" password="deployer" roles="manager-script"/>
```



Step 4. Run tomcat and go to http://your_ip

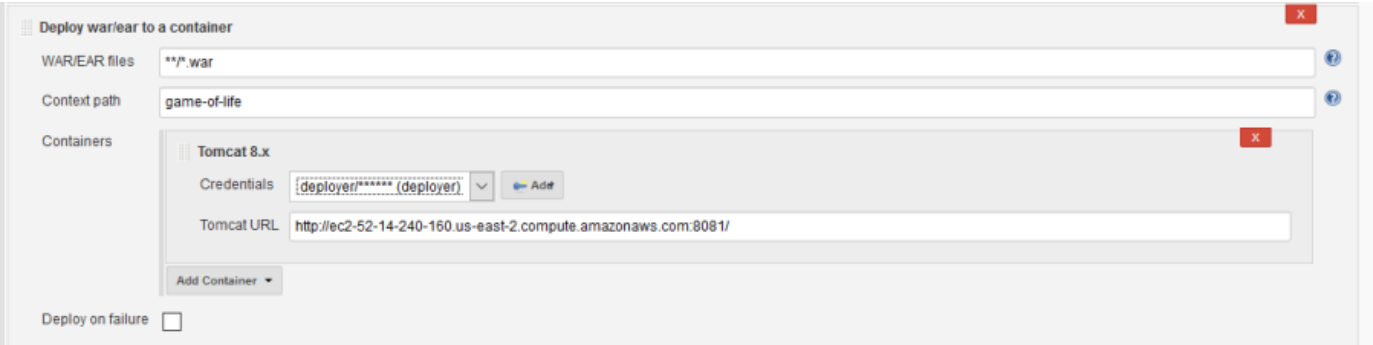
```
sudo /opt/tomcat8/bin/startup.sh
```



Step 5. Configure Jenkins job. Final step

Back in Jenkins, go to your job and select "Configure". Next, scroll down to the bottom of the page to the "Post-build Actions". Select the option "Deploy war/ear to a container" from the "Add post-build action" dropdown button. Fill in the new fields, e.g.:

```
WAR: **/*.war
Context: game-of-life
Tomcat 8.x
Add credentials - user/password/description: deployer
Tomcat_url: http://your_ip
```



Obtain deployment status from console:

```
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 26.803 s
[INFO] Finished at: 2018-03-12T13:40:00+00:00
[INFO] Final Memory: 25M/60M
[INFO] ------------------------------------------------------------------------
Archiving artifacts
Recording test results
Deploying /var/lib/jenkins/workspace/game-of-life-default/gameoflife-web/target/gameoflife.war to container Tomcat 8.x Remote with context game-of-life
  [/var/lib/jenkins/workspace/game-of-life-default/gameoflife-web/target/gameoflife.war] is not deployed. Doing a fresh deployment.
  Deploying [/var/lib/jenkins/workspace/game-of-life-default/gameoflife-web/target/gameoflife.war]
Finished: SUCCESS
```

Go to http://your_ip/game-of-life/ Congratulations!!!

Please, don't forget to switch off your machine!!