Assignment 1

Tasks to do:

- **Create a maven project**
- **Push maven project to GitHub**
- **Create Jenkinsfile in the project**
- **Configure Maven, Jenkins server and Nexus Server**

Prerequisites (installed software):

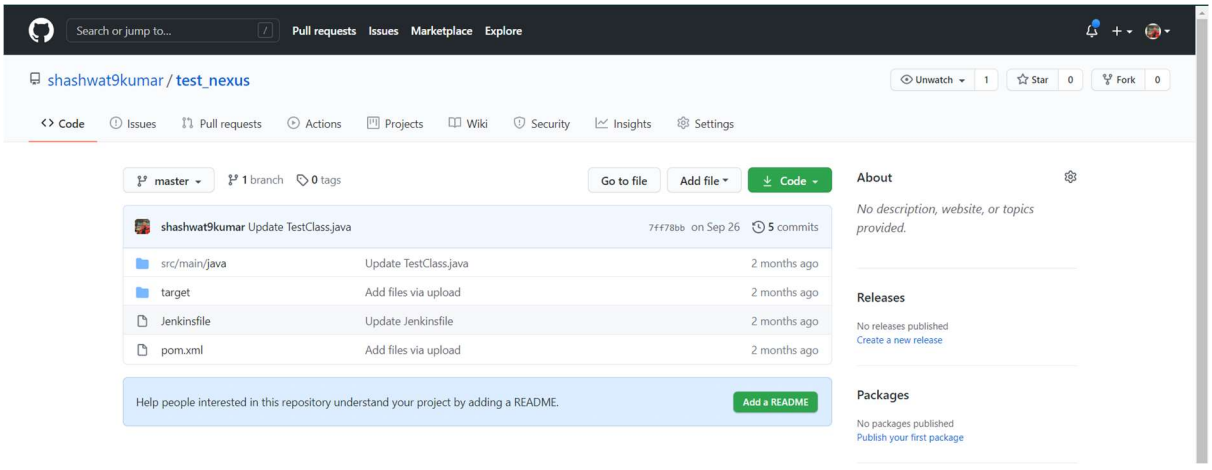
- **Java (java 11)**
- **Maven**
- **Jenkins**
- **Sonatype Nexus**

Solution:

Step 1:

Upload a maven project on GitHub and create a Jenkinsfile in it

The Jenkinsfile will be used by Jenkins to create and run the pipeline



Jenkinsfile would look like this:

```
pipeline{
                agent any
                stages{
                 stage('Start'){
                    steps{
                    echo 'Start the Pipeline'
                    }
                 }


                  stage('Clean phase Starts'){
                    steps{
                    echo 'Start the Clean phase'
                    }
                 }
                 stage('Clean'){
                    steps{
                    bat 'mvn clean'
                    }
                 }


                  stage('Install phase Starts'){
                    steps{
                    echo 'Start the Instll phase'
                    }
                 }
                  stage('Install'){
                    steps{
                    bat 'mvn install'
                    }
                 }


                  stage('Test phase Starts'){
                    steps{
                    echo 'Start the Test phase'
                    }
                 }
                  stage('Test'){
                    steps{
                    bat 'mvn test'
                    }
                 }



                  stage('Deploy phase Starts'){
                    steps{
                    echo 'Start the Deploy phase'
                    }
                 }
                  stage('Deploy'){
                    steps{
                    bat 'mvn deploy'
                    }
                 }
                }
              }
```

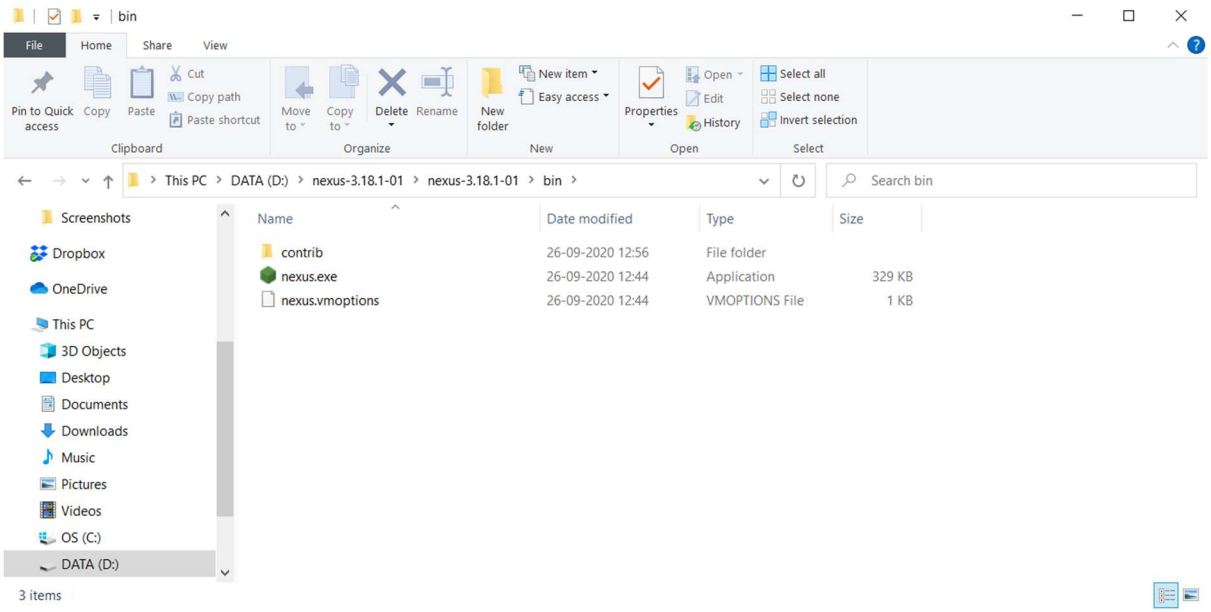
Step 2:

Go to the location on your system where nexus is installed.

Open the bin folder to find nexus.exe file

Open command prompt in the same folder and run the following command:

nexus.exe /run

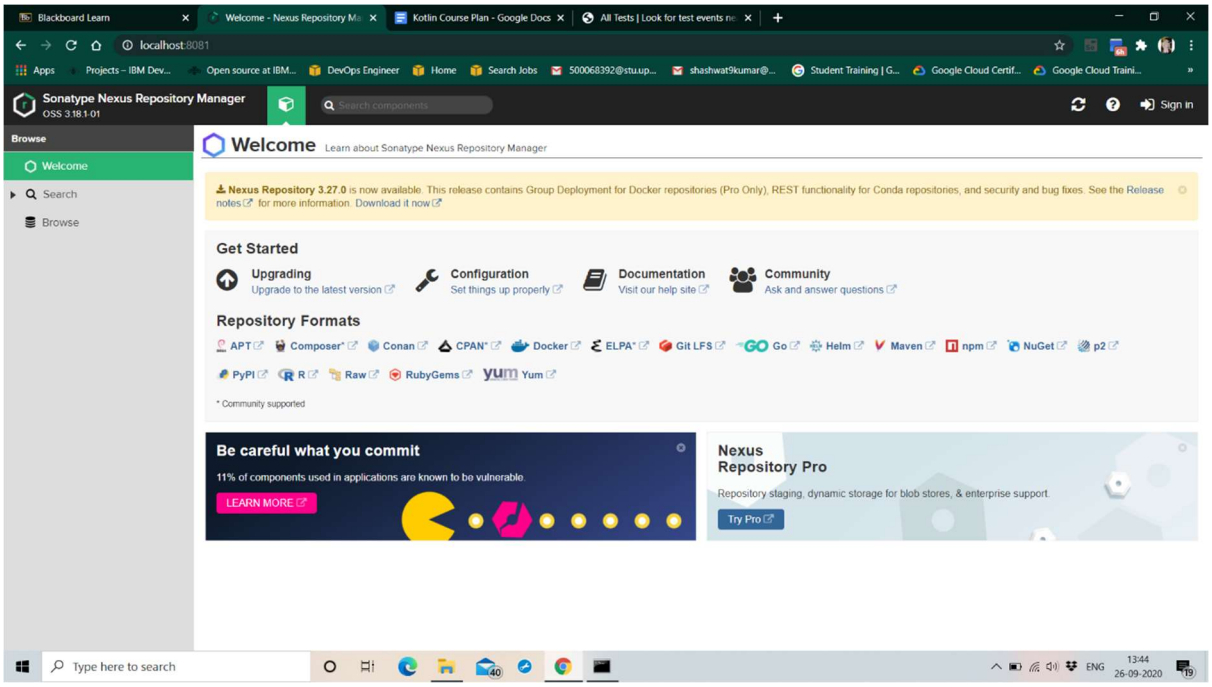



```
D:\nexus\nexus-3.28.0-01-win64\nexus-3.28.0-01\bin>nexus.exe/run
2020-11-20 18:53:04,136+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.pax.logging.NexusLogActivator - start
2020-11-20 18:53:04,976+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.features.internal.FeaturesWrapper - Fast
 FeaturesService starting
2020-11-20 18:53:06,604+0530 WARN  [FelixStartLevel] *SYSTEM uk.org.lidalia.sysoutslf4j.context.SysOutOverSLF4JInitialis
er - Your logging framework class org.ops4j.pax.logging.slf4j.Slf4jLogger is not known - if it needs access to the stand
ard println methods on the console you will need to register it by calling registerLoggingSystemPackage
2020-11-20 18:53:06,604+0530 INFO  [FelixStartLevel] *SYSTEM uk.org.lidalia.sysoutslf4j.context.SysOutOverSLF4J - Packag
e org.ops4j.pax.logging.slf4j registered; all classes within it or subpackages of it will be allowed to print to System.
out and System.err
2020-11-20 18:53:06,604+0530 INFO  [FelixStartLevel] *SYSTEM uk.org.lidalia.sysoutslf4j.context.SysOutOverSLF4J - Replac
ed standard System.out and System.err PrintStreams with SLF4JPrintStreams
2020-11-20 18:53:06,620+0530 INFO  [FelixStartLevel] *SYSTEM uk.org.lidalia.sysoutslf4j.context.SysOutOverSLF4J - Redire
cted System.out and System.err to SLF4J for this context
2020-11-20 18:53:06,620+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder - Propert
ies:
2020-11-20 18:53:06,620+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder -   appli
cation-host='0.0.0.0'
2020-11-20 18:53:06,620+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder -   appli
cation-port='8081'
2020-11-20 18:53:06,620+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder -   fabri
c.etc='D:\nexus\nexus-3.28.0-01-win64\nexus-3.28.0-01/etc/fabric'
2020-11-20 18:53:06,620+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder -   jetty
.etc='D:\nexus\nexus-3.28.0-01-win64\nexus-3.28.0-01/etc/jetty'
2020-11-20 18:53:06,620+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder -   karaf
.base='D:\nexus\nexus-3.28.0-01-win64\nexus-3.28.0-01'
2020-11-20 18:53:06,620+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder -   karaf
.data='D:\nexus\nexus-3.28.0-01-win64\sonatype-work\nexus3'
2020-11-20 18:53:06,620+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder -   karaf
```

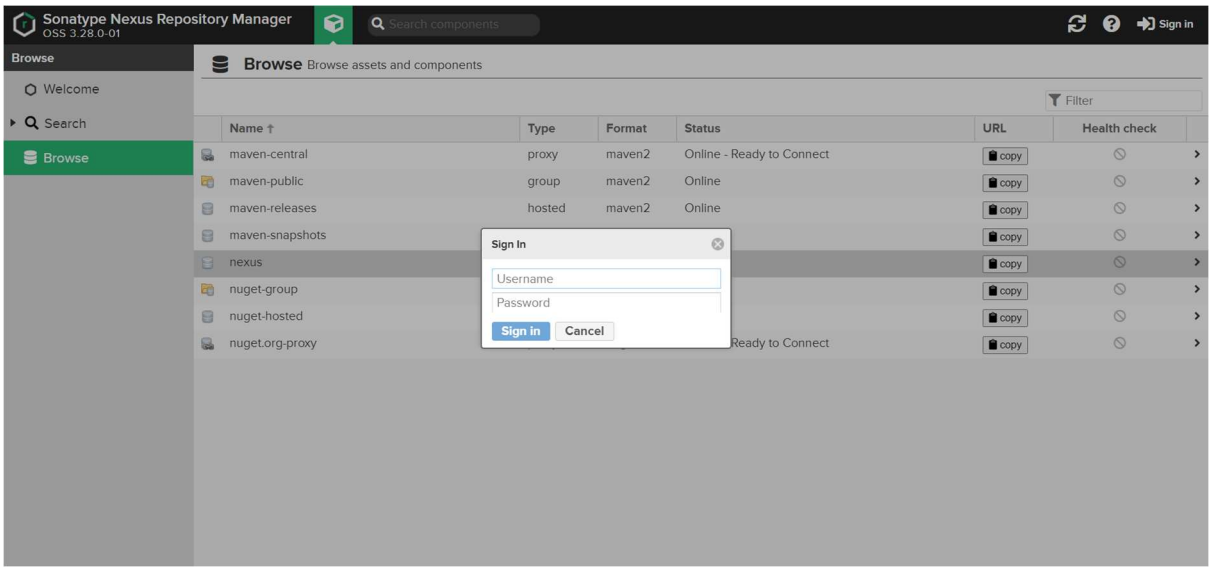
On successful start-up of nexus the terminal shows the following:

```
j.w.WebAppContext@19977182{Sonatype Nexus,/,file:///D:/nexus/nexus-3.28.0-01-win64/nexus-3.28.0-01/public/,AVAILABLE}
2020-11-20 18:54:01,166+0530 INFO  [jetty-main-1] *SYSTEM org.eclipse.jetty.server.AbstractConnector - Started ServerCon
nector@1a9de91f{HTTP/1.1, (http/1.1)}{0.0.0.0:8081}
2020-11-20 18:54:01,166+0530 INFO  [jetty-main-1] *SYSTEM org.eclipse.jetty.server.Server - Started @63782ms
2020-11-20 18:54:01,166+0530 INFO  [jetty-main-1] *SYSTEM org.sonatype.nexus.bootstrap.jetty.JettyServer -
------------------------------------------------

Started Sonatype Nexus OSS 3.28.0-01

------------------------------------------------
```

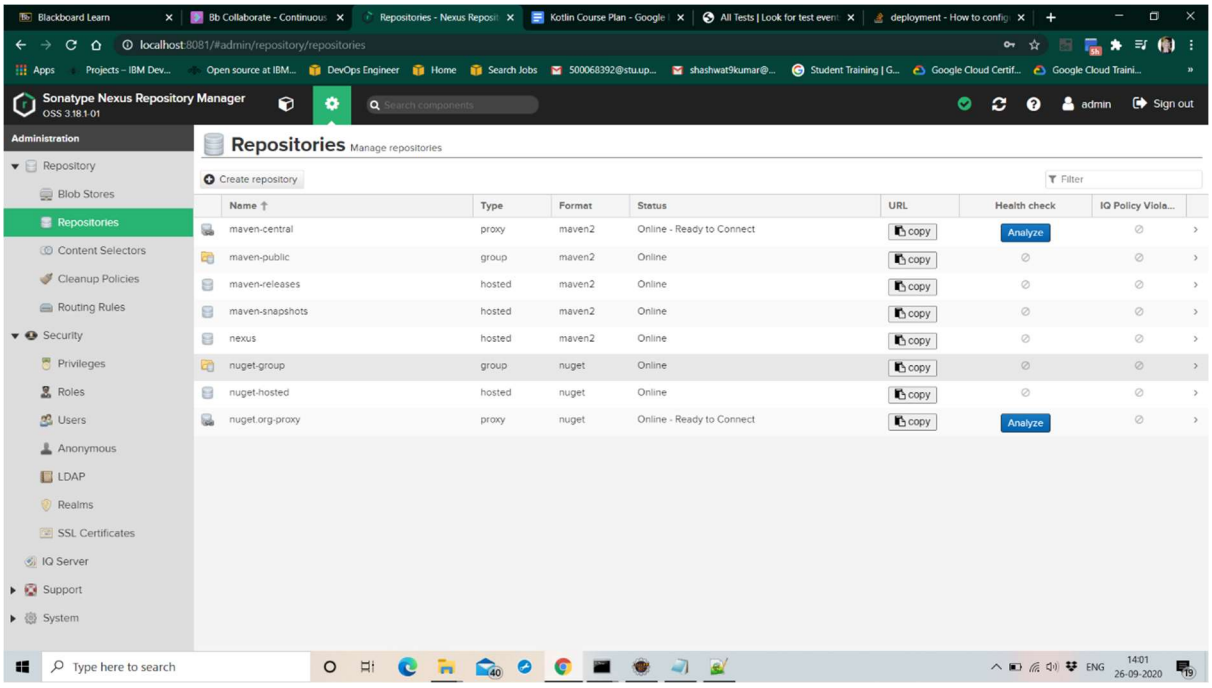
Once the nexus server runs head to **localhost:8081**, the port where Nexus server runs



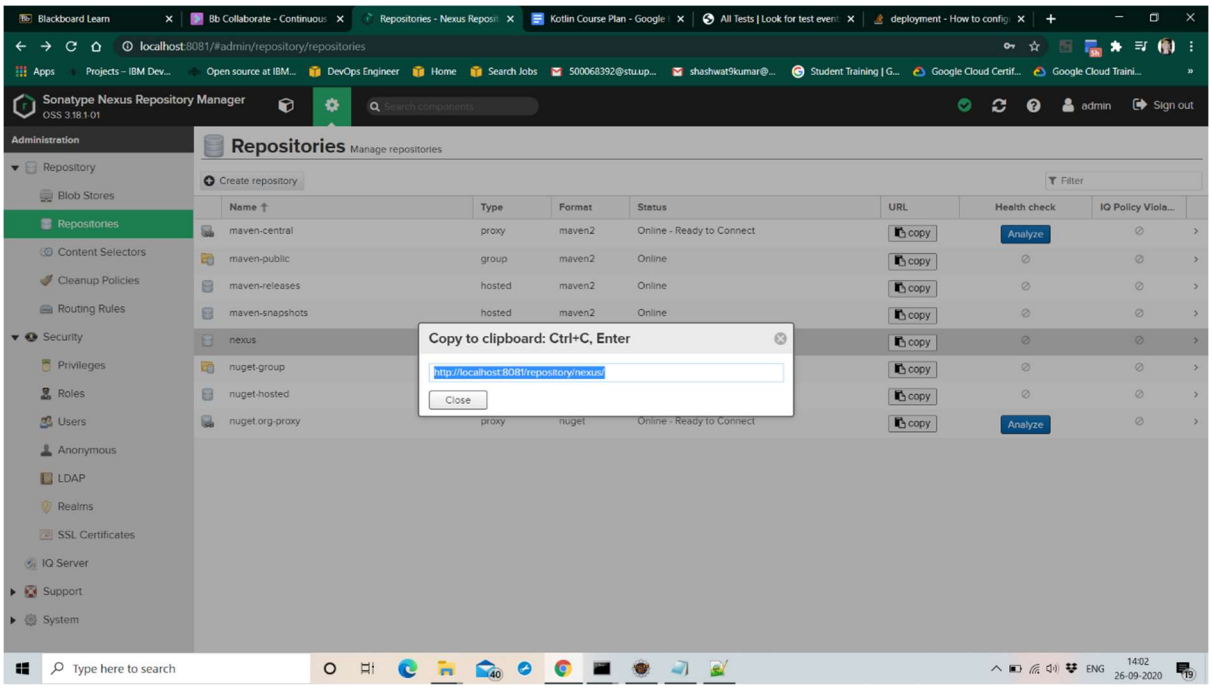
Log in into nexus server using your username and password



On login, Nexus server dashboard should look similar to this



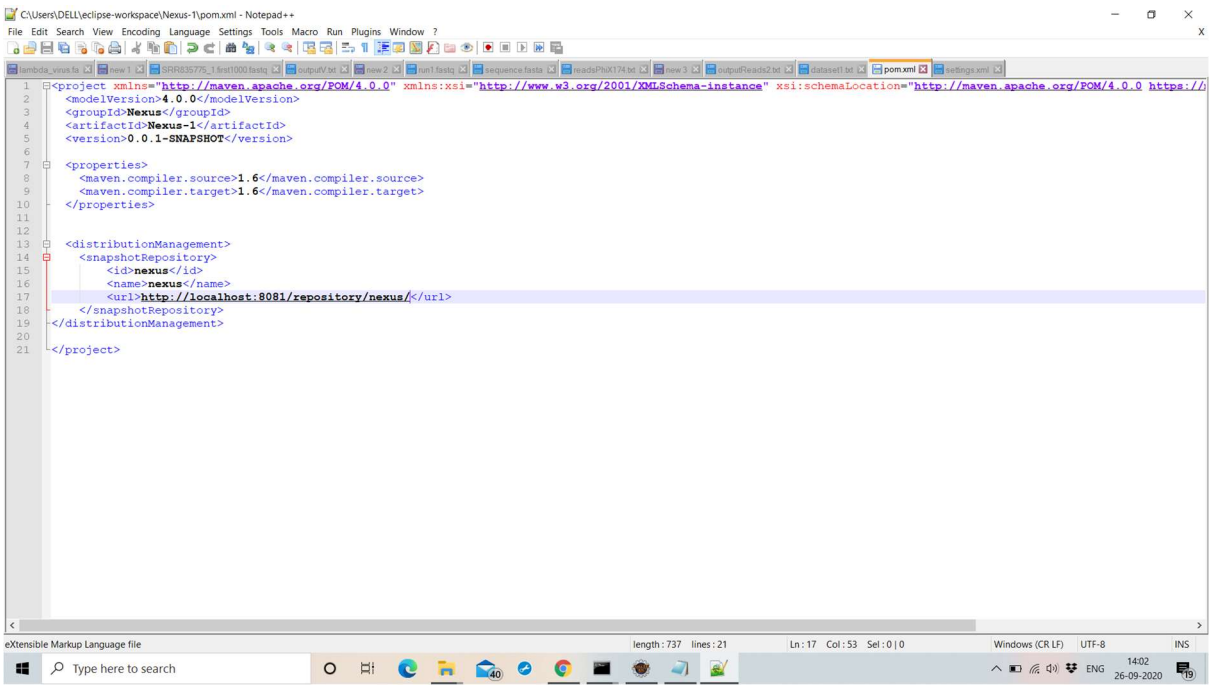
Create a new repository in Nexus (here called “nexus”) and copy its URL



Step 3:

Open the pom.xml of the Maven project whose artifact is to be published to Nexus

Add the snapshot Repository tag into the pom.xml and give the URL of Nexus repository copied



Open the **settings.xml** file from the place where maven is installed and edit the <servers> tag in it to add the username and password for the deployment server

```
<servers>
  <!-- server
   | Specifies the authentication information to use when connecting to a particular server, identified by
   | a unique name within the system (referred to by the 'id' attribute below).
   |
   | NOTE: You should either specify username/password OR privateKey/passphrase, since these pairings are
   |       used together.
   |
  <server>
    <id>deploymentRepo</id>
    <username>repouser</username>
    <password>repopwd</password>
  </server>
  -->

  <!-- Another sample, using keys to authenticate.
  <server>
    <id>siteServer</id>
    <privateKey>/path/to/private/key</privateKey>
    <passphrase>optional; leave empty if not used.</passphrase>
  </server>
  -->

  <server>
     <id>nexus</id>
     <username>admin</username>
     <password>123456</password>
   </server>


</servers>
```

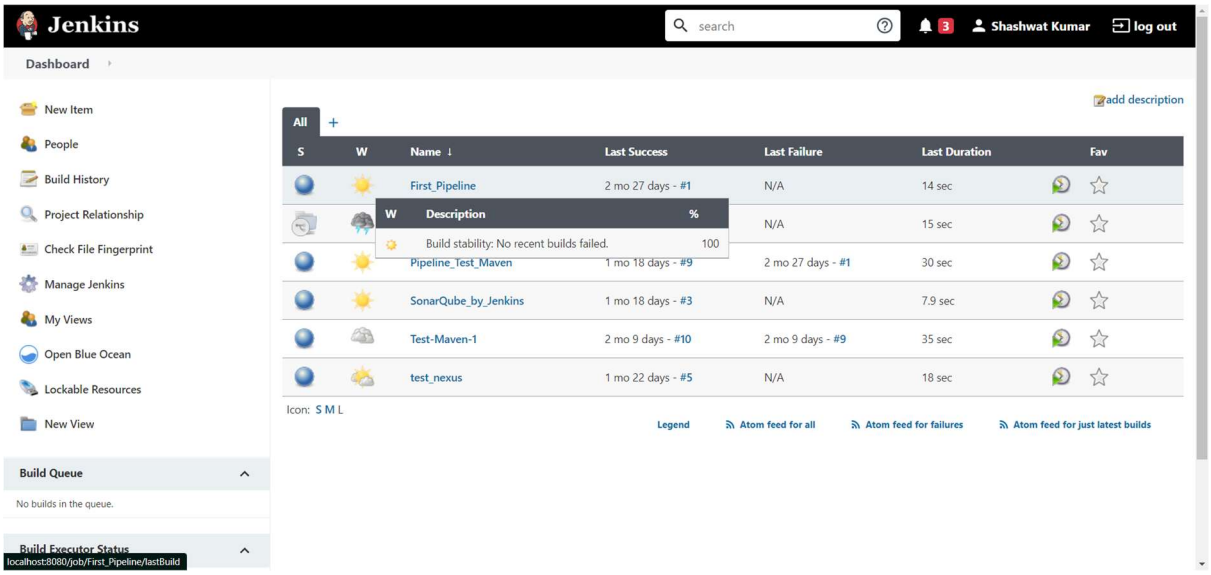
Step 4:

Open the location in terminal where Jenkins is installed and run the following command:

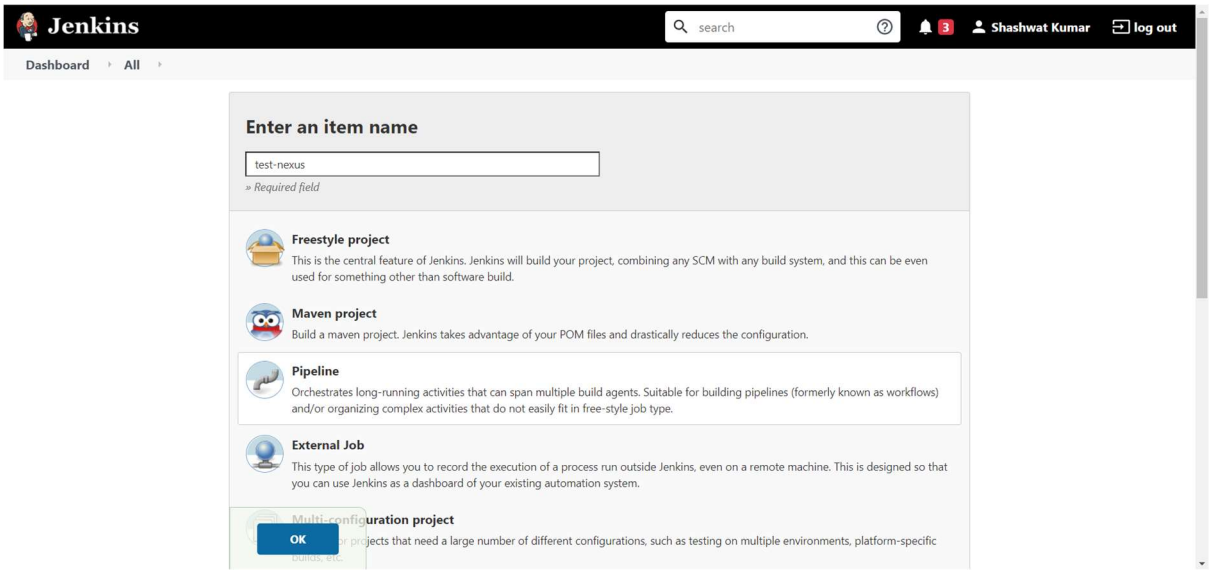
java -jar jenkins.war

```
D:\Jenkins>java -jar jenkins.war
Running from: D:\Jenkins\jenkins.war
webroot: $user.home/.jenkins
2020-11-20 13:20:24.745+0000 [id=1]     INFO    org.eclipse.jetty.util.log.Log#initialized: Logging initialized @1006ms
to org.eclipse.jetty.util.log.JavaUtilLog
2020-11-20 13:20:24.822+0000 [id=1]     INFO    winstone.Logger#logInternal: Beginning extraction from war file
2020-11-20 13:20:24.874+0000 [id=1]     WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2020-11-20 13:20:24.960+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: jetty-9.4.30.v20200611; built:
2020-06-11T12:34:51.929Z; git: 271836e4c1f4612f12b7bb13ef5a92a927634b0d; jvm 11.0.8+10-LTS
2020-11-20 13:20:25.754+0000 [id=1]     INFO    o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /,
did not find org.eclipse.jetty.jsp.JettyJspServlet
2020-11-20 13:20:25.801+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: DefaultSessionIdManager worke
rName=node0
2020-11-20 13:20:25.801+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: No SessionScavenger set, usin
g defaults
2020-11-20 13:20:25.816+0000 [id=1]     INFO    o.e.j.server.session.HouseKeeper#startScavenging: node0 Scavenging every
 600000ms
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.thoughtworks.xstream.core.util.Fields (file:/C:/Users/DELL/.jenkins/war/WEB-IN
F/lib/xstream-1.4.7-jenkins-1.jar) to field java.util.TreeMap.comparator
WARNING: Please consider reporting this to the maintainers of com.thoughtworks.xstream.core.util.Fields
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2020-11-20 13:20:26.428+0000 [id=1]     INFO    hudson.WebAppMain#contextInitialized: Jenkins home directory: C:\Users\D
ELL\.jenkins found at: $user.home/.jenkins
2020-11-20 13:20:26.654+0000 [id=1]     INFO    o.e.j.s.handler.ContextHandler#doStart: Started w.@1e287667{Jenkins v2.2
59,/,file:///C:/Users/DELL/.jenkins/war/,AVAILABLE}{C:\Users\DELL\.jenkins\war}
2020-11-20 13:20:26.716+0000 [id=1]     INFO    o.e.j.server.AbstractConnector#doStart: Started ServerConnector@6913c1fb
{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
```

Log in into Jenkins dashboard



Click on new Item to create a new pipeline

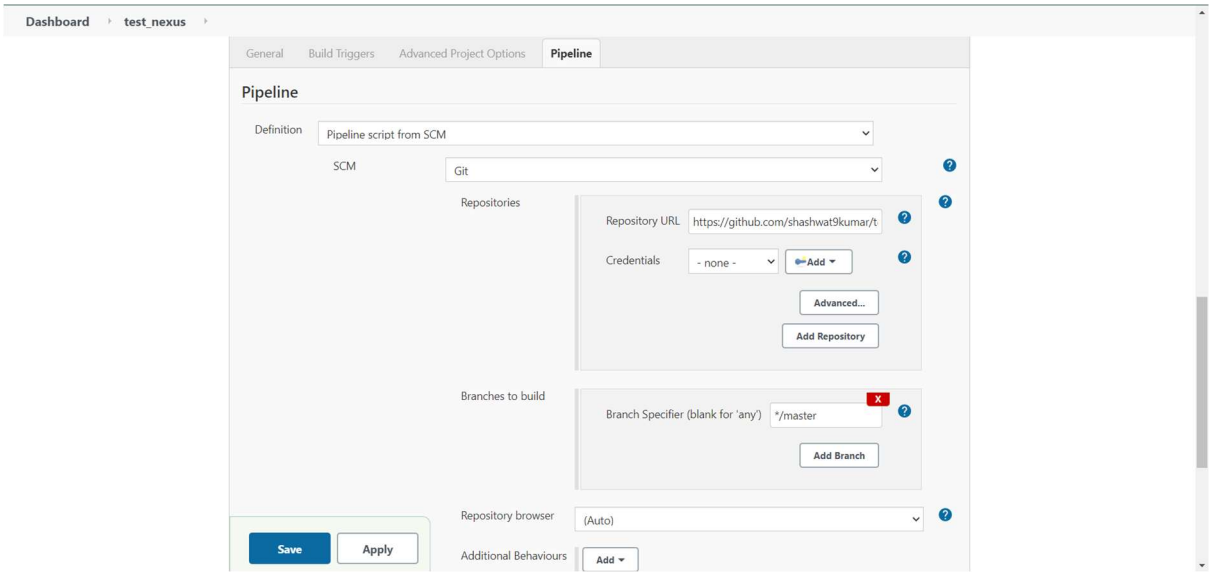


On the page that opens head over to ‘Pipeline’ section and select ‘Pipeline Script from SCM’

In the SCM, select ‘Git’

Provide the URL of the GitHub repository where your project is hosted

Click on ‘Save’ and ‘Apply’



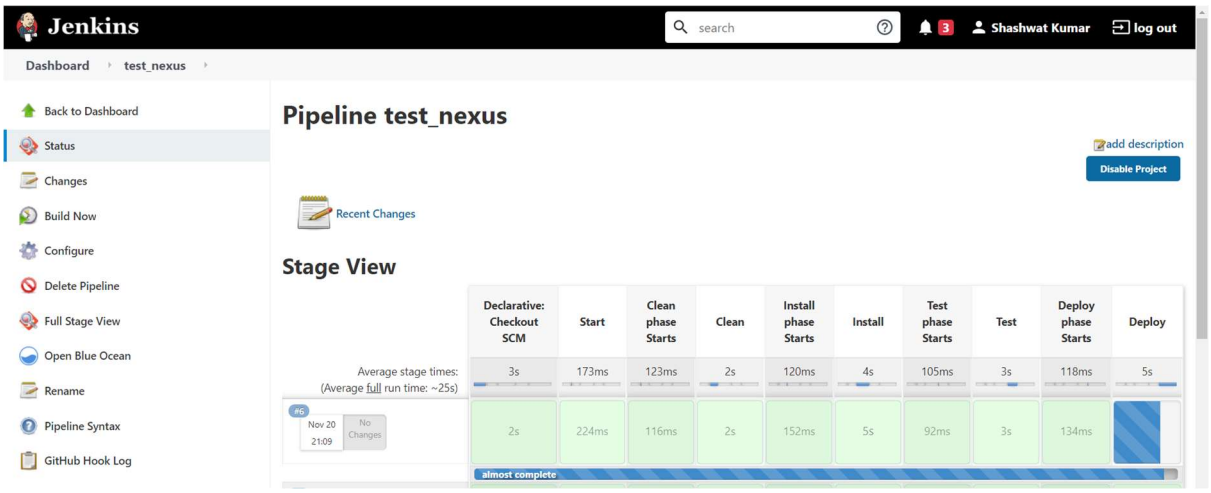
The GitHub repository must contain a Jenkins file that contains "**mvn deploy**" stage. This would help us deploy the artifact on Nexus server.

Step 5:

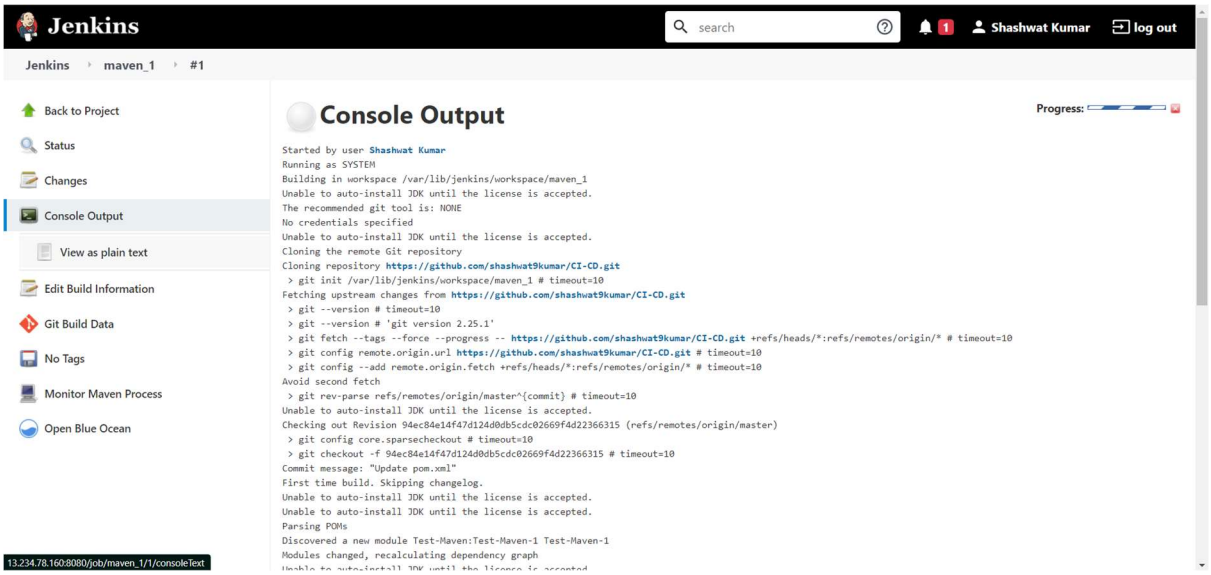
In the screen that pops up, click on Build Now option to build the project.

The project starts to build and run

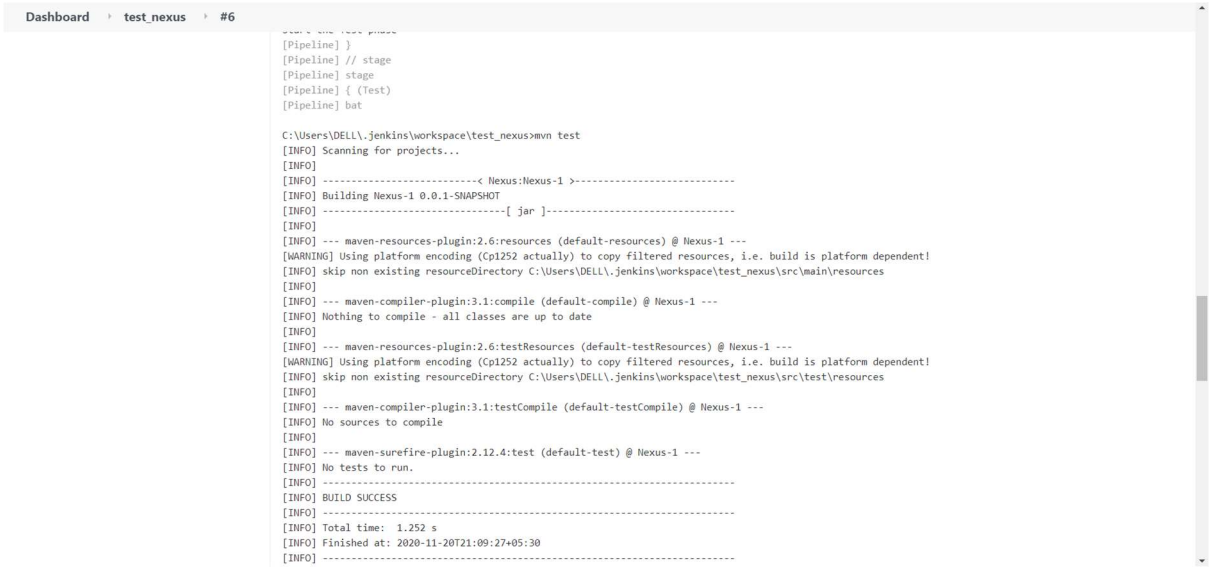
The project pipeline can be seen on the screen depicting the stages defined in the Jenkinsfile



Select the console output option to check how the build is running



If the build runs successfully it would return “BUILD SUCCESS”. It would return failure in case of any errors in the project



Browse the Nexus repository on the running nexus server now to see the deployed jar file