

# 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):

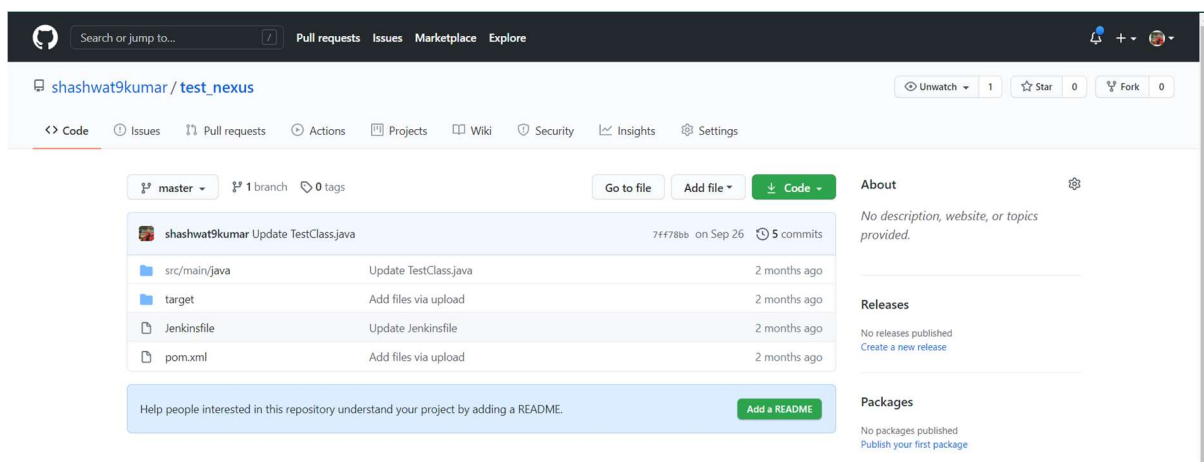
- Java (java 11)
- Maven
- Jenkins
- Sonatype Nexus

## 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 Install 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'
      }
    }
  }
}
```

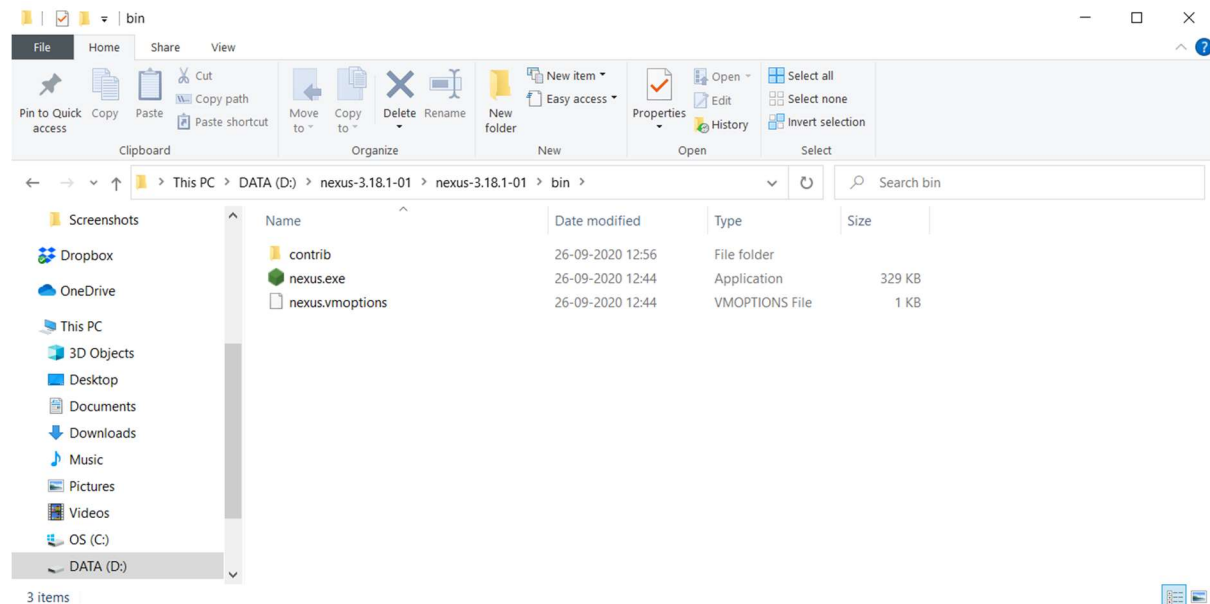
## Step 2:

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**

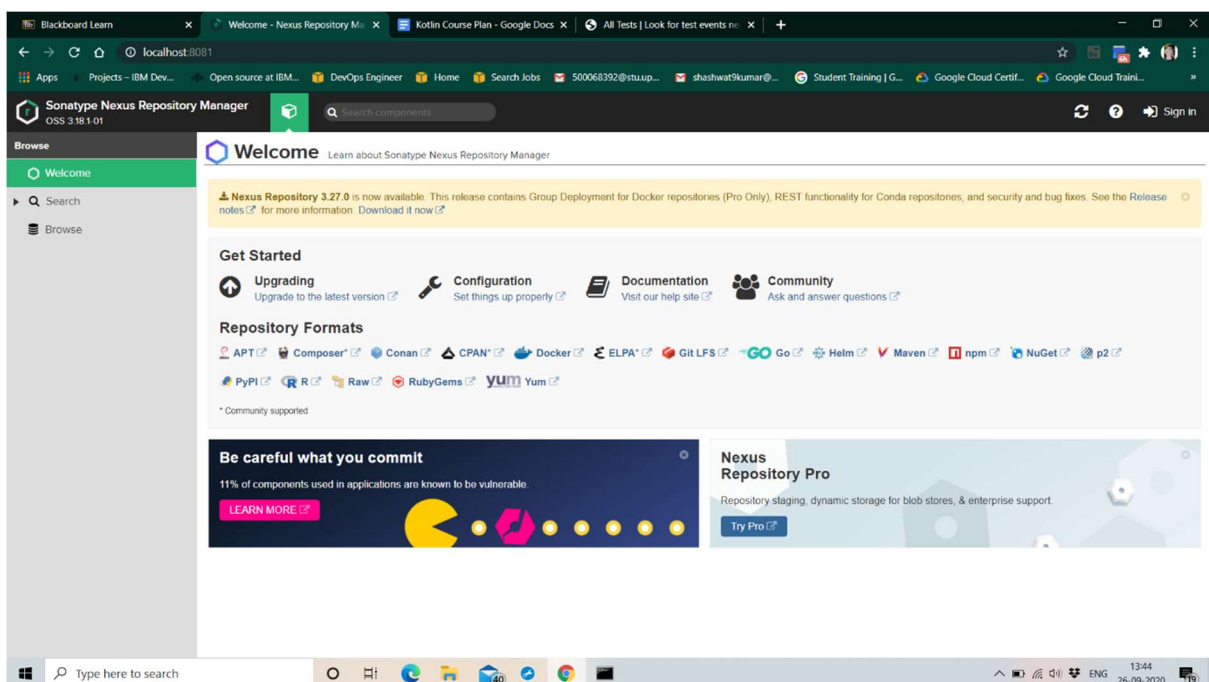


```
D:\nexus\nexus-3.28.0-01-win64\nexus-3.28.0-01\bin>nexus.exe/run
2020-11-20 18:53:04,136+0530 INFO [FelixStartLevel] *SYSTEM org.sonatype.nexus.pax.logging.NexusLogActivator - start
2020-11-20 18:53:04,976+0530 INFO [FelixStartLevel] *SYSTEM org.sonatype.nexus.features.internal.FeaturesWrapper - 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
```

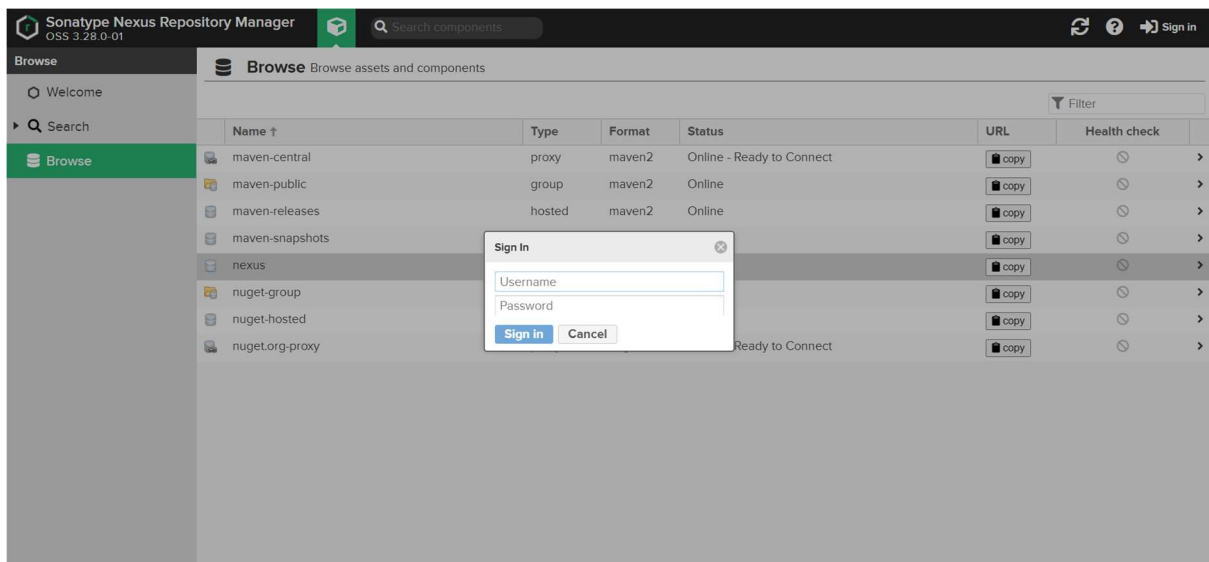
On successful start-up of nexus the terminal shows the following:

```
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
-----
```

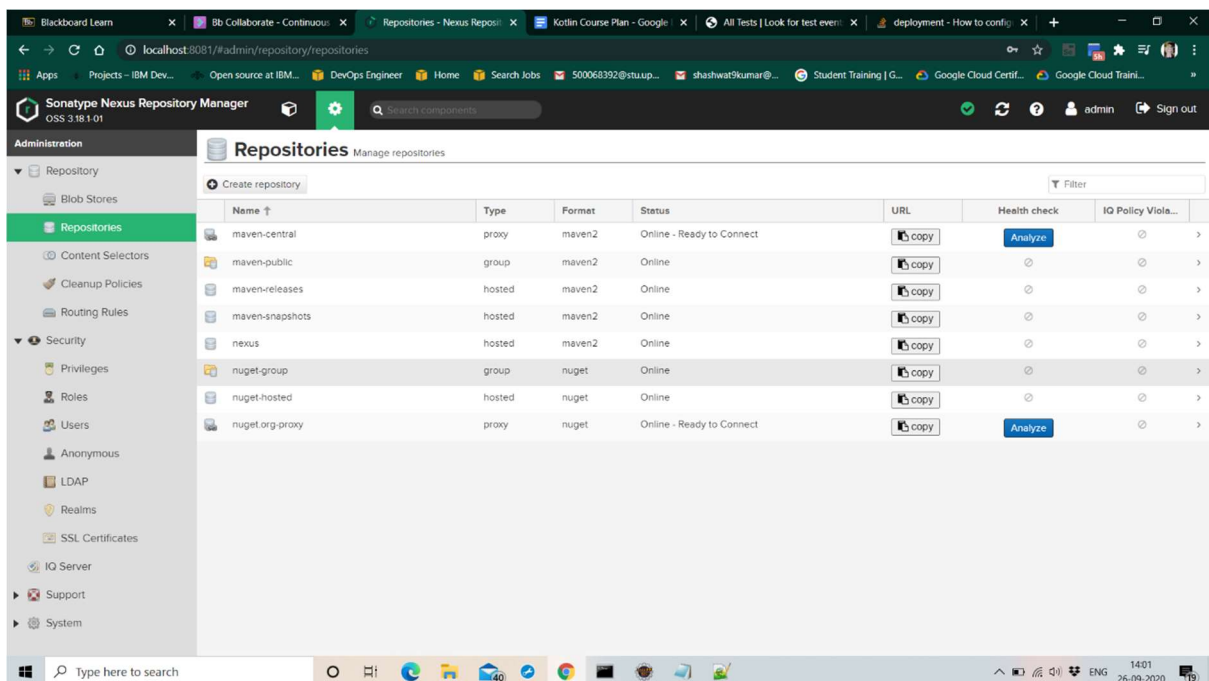
Once the nexus server runs head to **localhost:8081**, the port where Nexus server runs



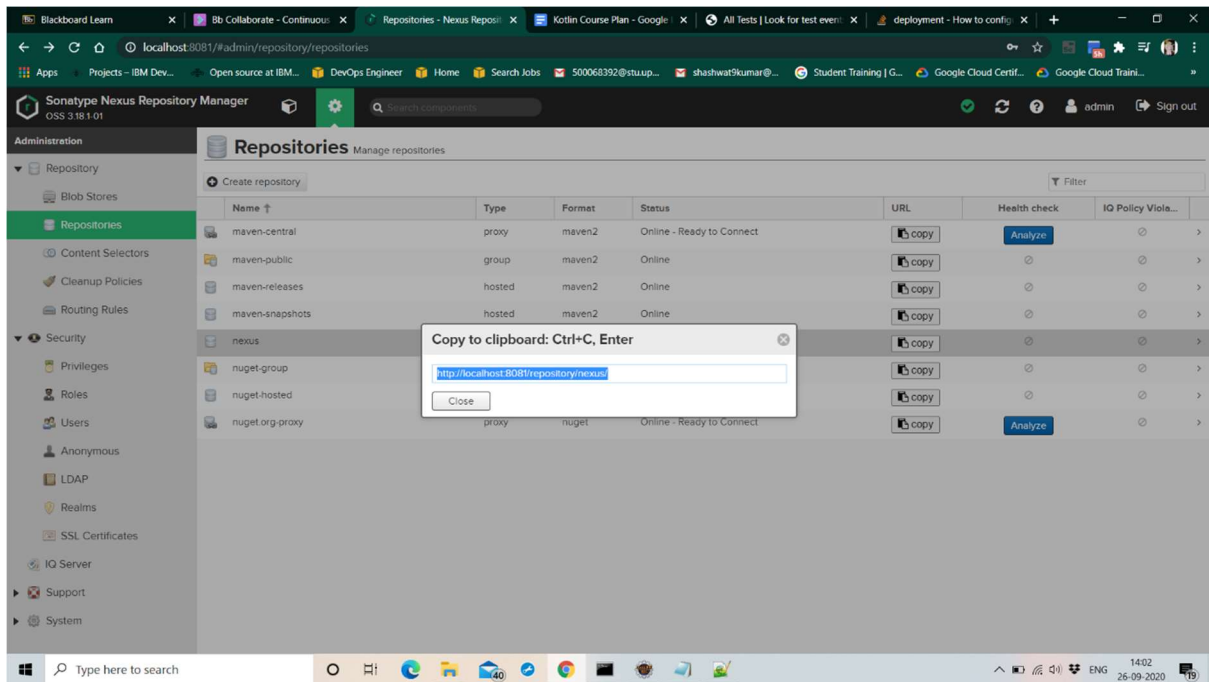
Log in into nexus server using your username and password



On login, Nexus server dashboard should look similar to this



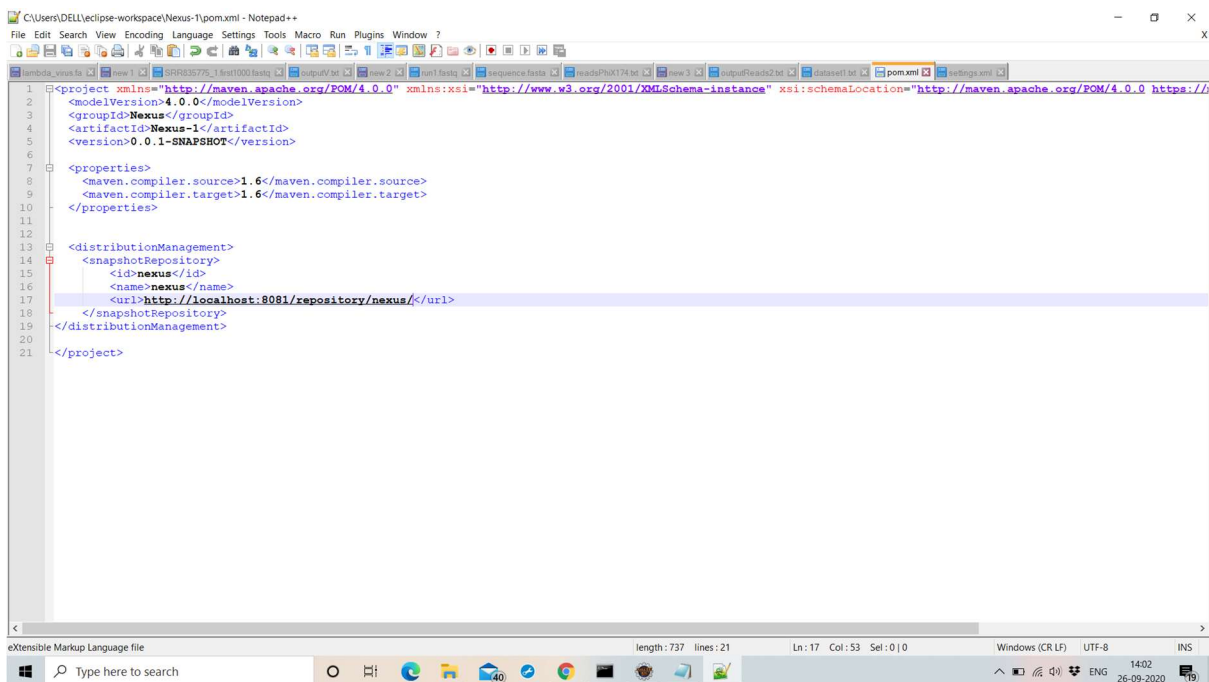
Create a new repository in Nexus (here called “nexus”) and copy its URL



### 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

```
111 <servers>
112 <!-- server
113 | Specifies the authentication information to use when connecting to a particular server, identified by
114 | a unique name within the system (referred to by the 'id' attribute below).
115 |
116 | NOTE: You should either specify username/password OR privateKey/passphrase, since these pairings are
117 | used together.
118 |
119 <server>
120 <id>deploymentRepo</id>
121 <username>repouser</username>
122 <password>repopwd</password>
123 </server>
124 -->
125
126 <!-- Another sample, using keys to authenticate.
127 <server>
128 <id>siteServer</id>
129 <privateKey>/path/to/private/key</privateKey>
130 <passphrase>optional; leave empty if not used.</passphrase>
131 </server>
132 -->
133
134 <server>
135 <id>nexus</id>
136 <username>admin</username>
137 <password>123456</password>
138 </server>
139
140
141 </servers>
142
```

#### Step 4:

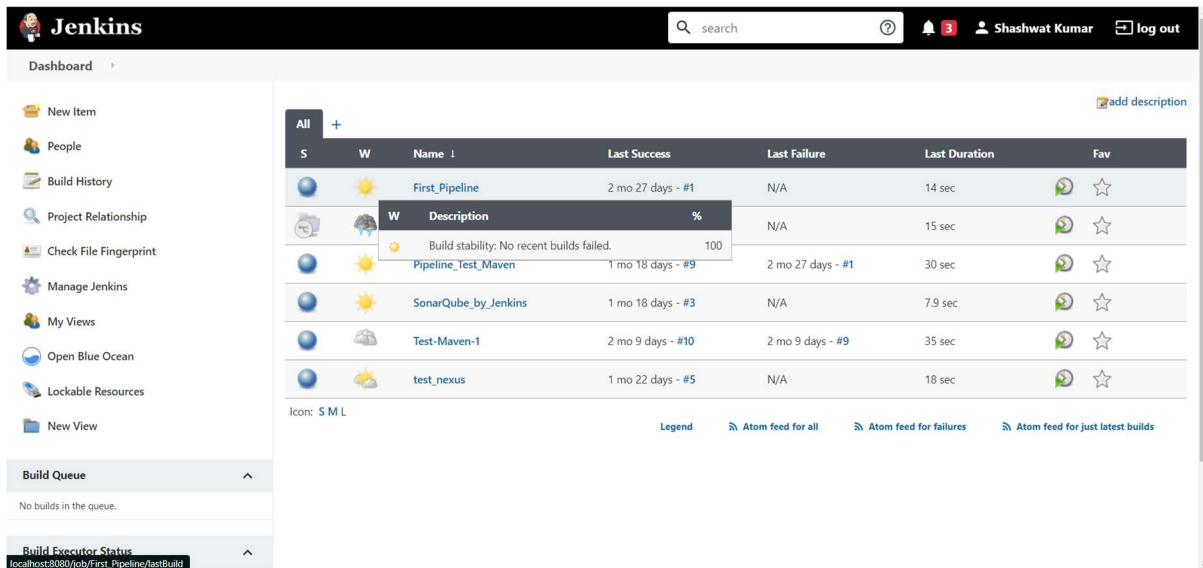
Open the location in terminal where Jenkins is installed and run the following command:

**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-INF/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\DELL\.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}
```



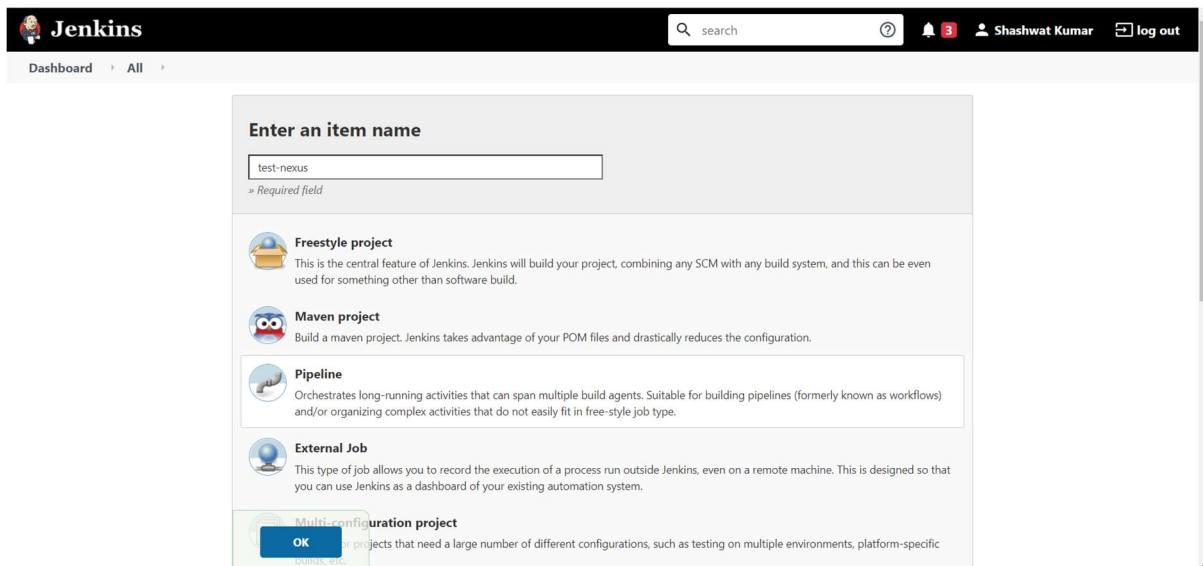
## Log in into Jenkins dashboard



The screenshot shows the Jenkins dashboard. The top navigation bar includes the Jenkins logo, a search bar, a notification bell with 3 alerts, the user name 'Shashwat Kumar', and a 'log out' button. The left sidebar contains a 'Dashboard' link and a list of items: 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'My Views', 'Open Blue Ocean', 'Lockable Resources', and 'New View'. Below this is a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section showing 'localhost:8080/job/First\_Pipeline/lastBuild'. The main content area displays a table of jobs with columns: 'S' (Status), 'W' (Web icon), 'Name', 'Last Success', 'Last Failure', 'Last Duration', and 'Fav'. The table lists several jobs, including 'First\_Pipeline', 'Pipeline\_Test\_Maven', 'SonarQube\_by\_Jenkins', 'Test-Maven-1', and 'test\_nexus'. A tooltip is visible over the 'Description' column for 'Pipeline\_Test\_Maven', showing 'Build stability: No recent builds failed.' and '100%'. At the bottom of the table, there are icons for 'S M L' and a 'Legend' section with links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

S	W	Name	Last Success	Last Failure	Last Duration	Fav
		First_Pipeline	2 mo 27 days - #1	N/A	14 sec	
		W Description %		N/A	15 sec	
		Build stability: No recent builds failed.	100			
		Pipeline_Test_Maven	1 mo 18 days - #9	2 mo 27 days - #1	30 sec	
		SonarQube_by_Jenkins	1 mo 18 days - #3	N/A	7.9 sec	
		Test-Maven-1	2 mo 9 days - #10	2 mo 9 days - #9	35 sec	
		test_nexus	1 mo 22 days - #5	N/A	18 sec	

## Click on new Item to create a new pipeline



The screenshot shows the 'New Item' form in Jenkins. The top navigation bar is the same as the dashboard. The left sidebar shows 'Dashboard' and 'All'. The main content area is titled 'Enter an item name' and contains a text input field with 'test-nexus' entered. Below the input field is a note '» Required field'. The form lists several project types with icons and descriptions: 'Freestyle project' (This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.), 'Maven project' (Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.), 'Pipeline' (Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.), 'External Job' (This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.), and 'Multi-configuration project' (Build projects that need a large number of different configurations, such as testing on multiple environments, platform-specific). At the bottom of the form is a blue 'OK' button.

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'



Dashboard > test\_nexus >

General Build Triggers Advanced Project Options **Pipeline**

### Pipeline

Definition Pipeline script from SCM

SCM Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

Repository browser (Auto)

Additional Behaviours

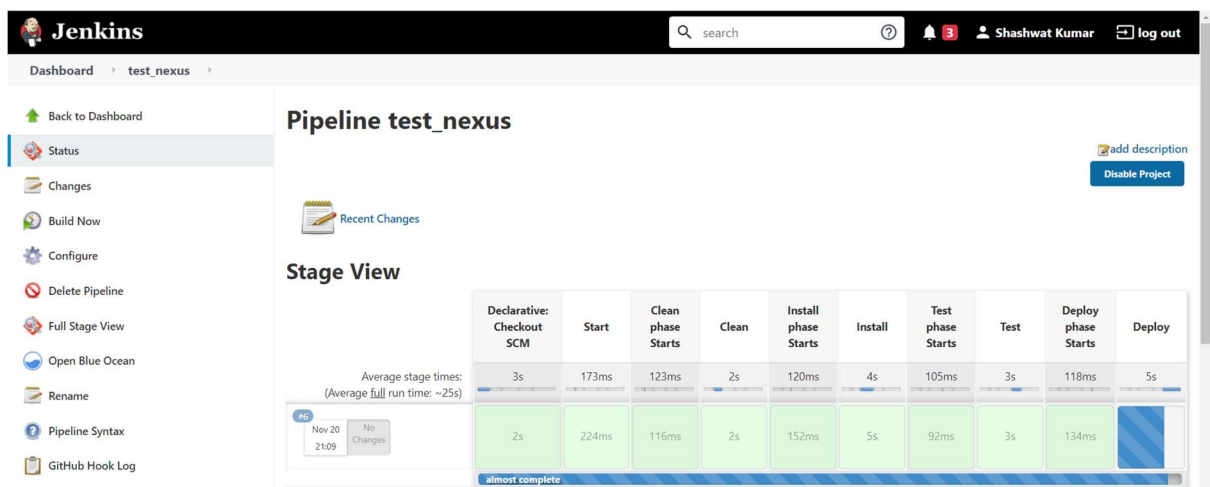
The GitHub repository must contain a Jenkins file that contains “**mvn deploy**” stage. This would help us deploy the artifact on Nexus server.

## Step 5:

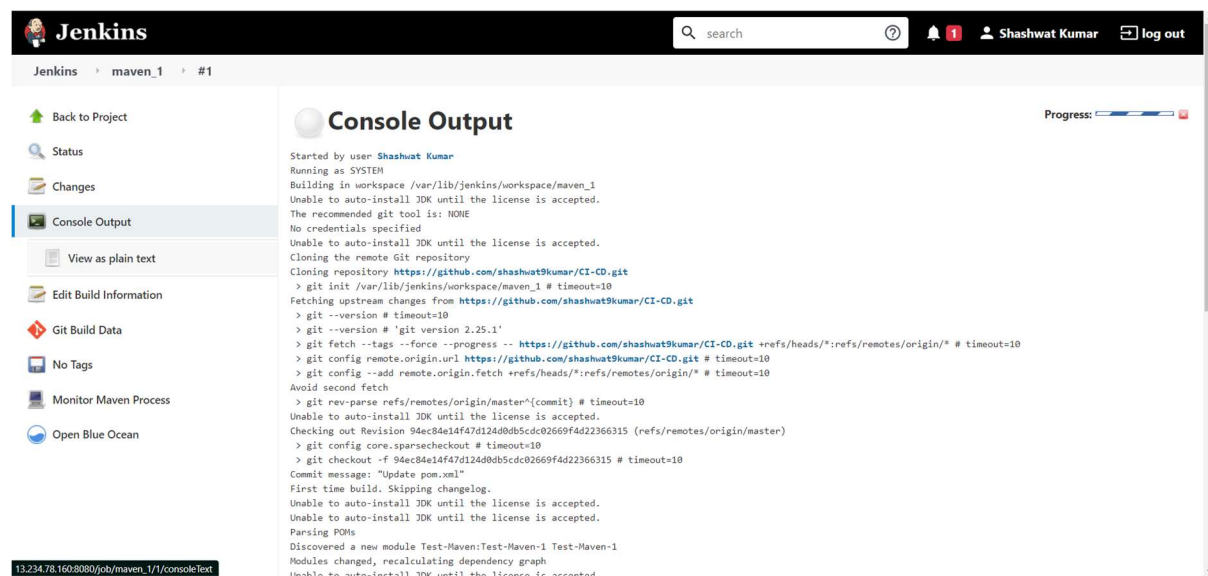
In the screen that pops up, click on Build Now option to build the project.

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



The screenshot shows the Jenkins web interface for job `maven_1` build `#1`. The left sidebar contains navigation links: `Back to Project`, `Status`, `Changes`, `Console Output` (selected), `View as plain text`, `Edit Build Information`, `Git Build Data`, `No Tags`, `Monitor Maven Process`, and `Open Blue Ocean`. The main area is titled `Console Output` and shows the build log. The log starts with user information and workspace details, followed by Git operations: cloning the repository, fetching upstream changes, and checking out a specific revision. It then shows Maven commands being executed, including `mvn test`. The log ends with a progress bar and a status indicator.

```
Started by user Shashwat Kumar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/maven_1
Unable to auto-install JDK until the license is accepted.
The recommended git tool is: NONE
No credentials specified
Unable to auto-install JDK until the license is accepted.
Cloning the remote Git repository
Cloning repository https://github.com/shashwat9kumar/CI-CD.git
> git init /var/lib/jenkins/workspace/maven_1 # timeout=10
Fetching upstream changes from https://github.com/shashwat9kumar/CI-CD.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
> git fetch --tags --force --progress -- https://github.com/shashwat9kumar/CI-CD.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/shashwat9kumar/CI-CD.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Unable to auto-install JDK until the license is accepted.
Checking out Revision 94ec84e14f47d124d0b5cd02669f4d22366315 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 94ec84e14f47d124d0b5cd02669f4d22366315 # timeout=10
Commit message: "Update pom.xml"
First time build. Skipping changelog.
Unable to auto-install JDK until the license is accepted.
Unable to auto-install JDK until the license is accepted.
Parsing POMs
Discovered a new module Test-Maven:Test-Maven-1 Test-Maven-1
Modules changed, recalculating dependency graph
Unable to auto-install JDK until the license is accepted
```

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



The screenshot shows the Jenkins web interface for job `test_nexus` build `#6`. The left sidebar contains navigation links: `Dashboard`, `test_nexus`, and `#6`. The main area is titled `Console Output` and shows the build log. The log starts with a pipeline definition, followed by Maven commands being executed, including `mvn test`. The log ends with a progress bar and a status indicator.

```
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] bat

C:\Users\DELL\.jenkins\workspace\test_nexus>mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] ----- Nexus:Nexus-1 -----
[INFO] Building Nexus-1 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Nexus-1 ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\DELL\.jenkins\workspace\test_nexus\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ Nexus-1 ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Nexus-1 ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\DELL\.jenkins\workspace\test_nexus\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ Nexus-1 ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ Nexus-1 ---
[INFO] No tests to run.
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.252 s
[INFO] Finished at: 2020-11-20T21:09:27+05:30
[INFO]
[INFO] -----
```

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

