**Experiment 5**

**Title:** Sonatype Nexus to Deploy Artifacts

**Aim:** Use Nexus to deploy artifacts after building a Java project hosted on GitHub using Maven via Jenkins

**Pre-requisites:**

1. Maven project hosted on GitHub

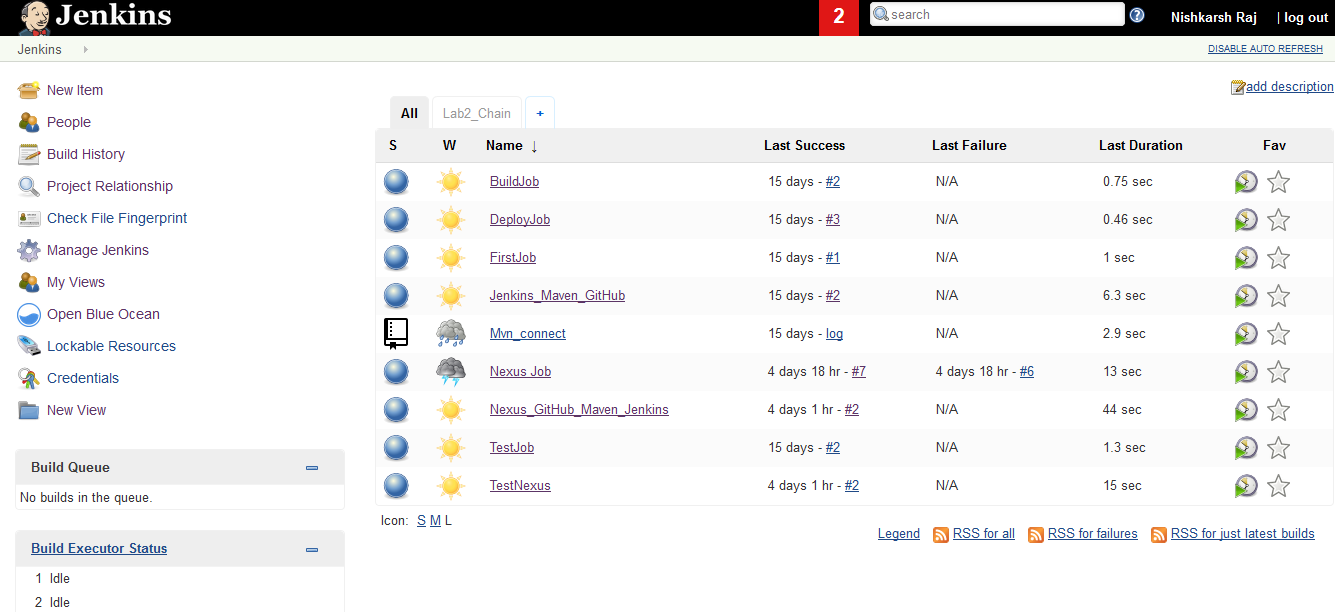
Here, we use <https://www.github.com/NishkarshRaj/Maven-Using-CMD>

1. Jenkins.war installed on our local machine

Go to the command line and type:

**$ java –jar Jenkins.war**

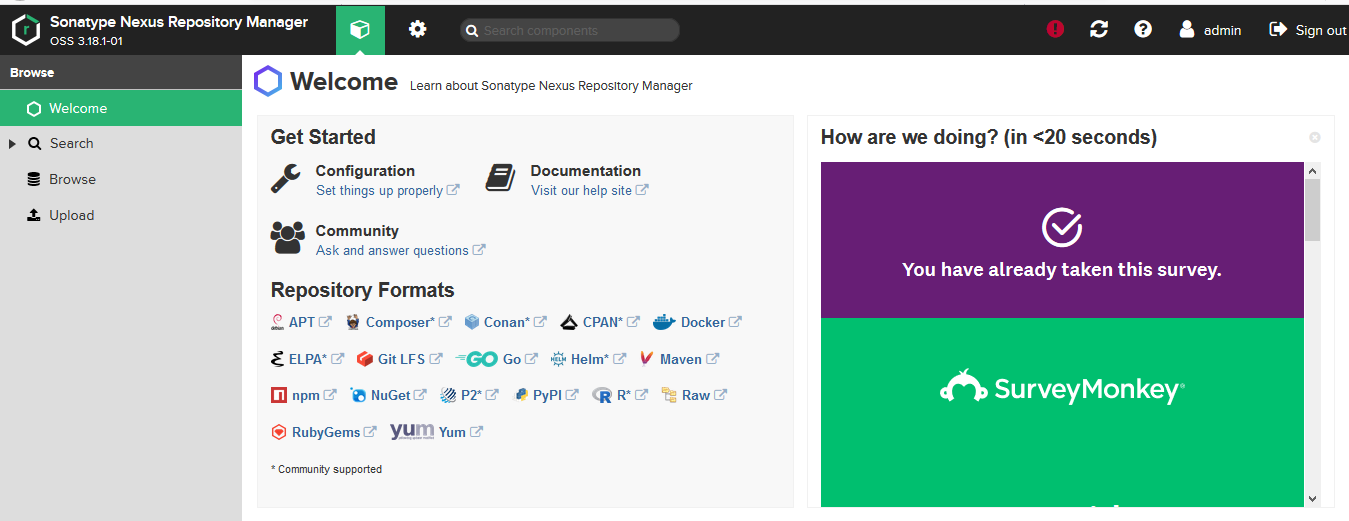
to launch Jenkins and open localhost:8080 on the web browser.



1. Sonatype Nexus installed on our local machine. Go to Nexus repository’s bin folder and use command line to launch Nexus

**$ nexus.exe /run**

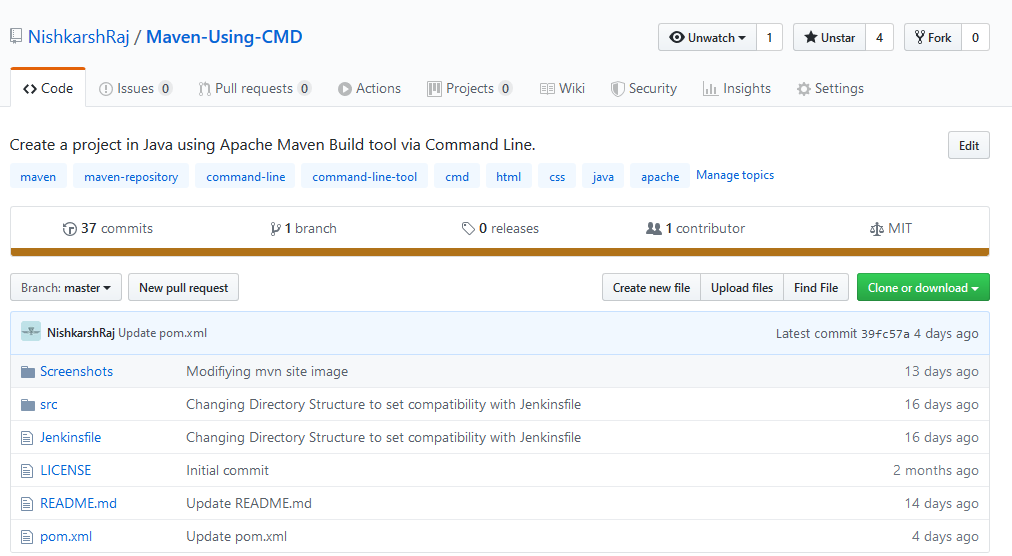
Open localhost:8081 on the web browser to access Nexus repository



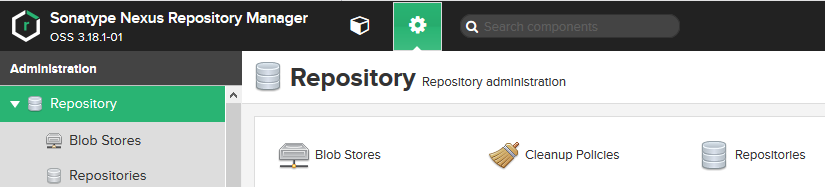
1. Apache Maven installed and configured on the local machine
2. Java JDK 1.8 or above installed and configured.
3. Git Bash installed on local machine.

**Working:**

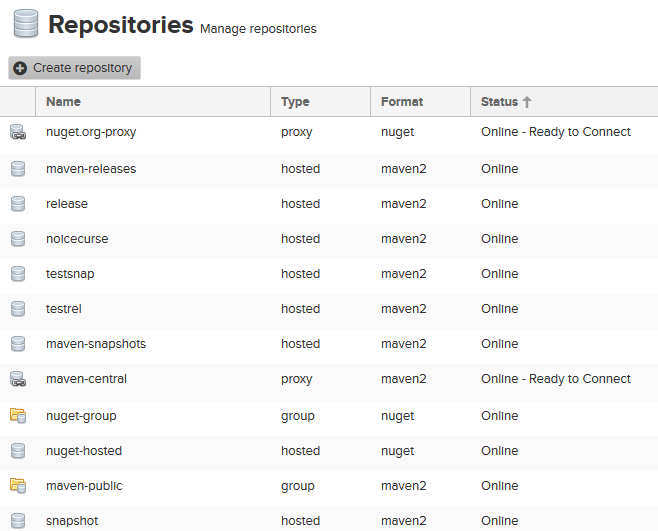
1. Host a Java project on GitHub



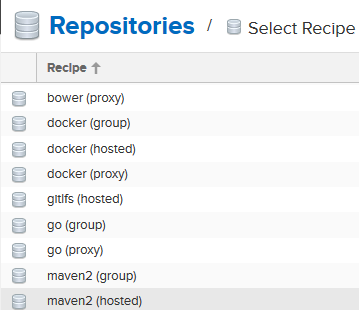
1. Go to Nexus and create repositories where the artifacts will be deployed
2. Click on settings and go to Repository tab.



1. A list of currently present Repositories shows up. Click on Create repository



1. Select **Maven 2 Hosted**  in select Recipe.

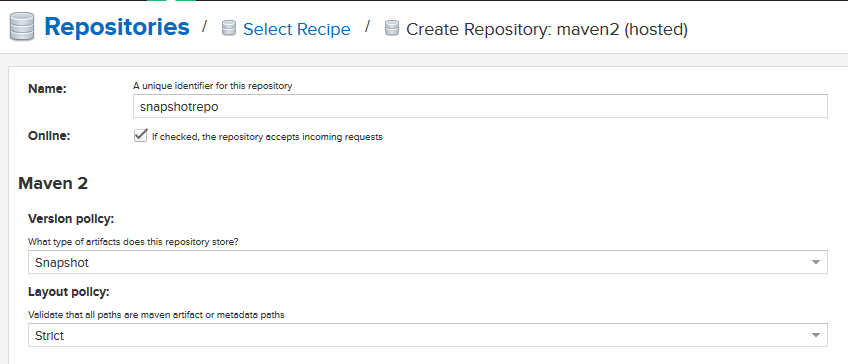


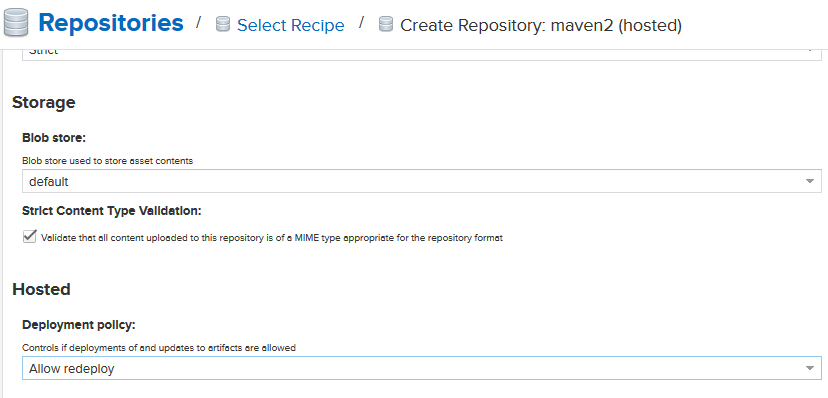
1. Configure the Maven 2 Hosted repository:

**Name:** name of the repository where the artefact will be deployed. Make it unique and meaningful and remember it because it has to be configured in the settings of Maven and POM.xml file of the project.

**Version Policy:** Select snapshot because we want to deploy the binary.

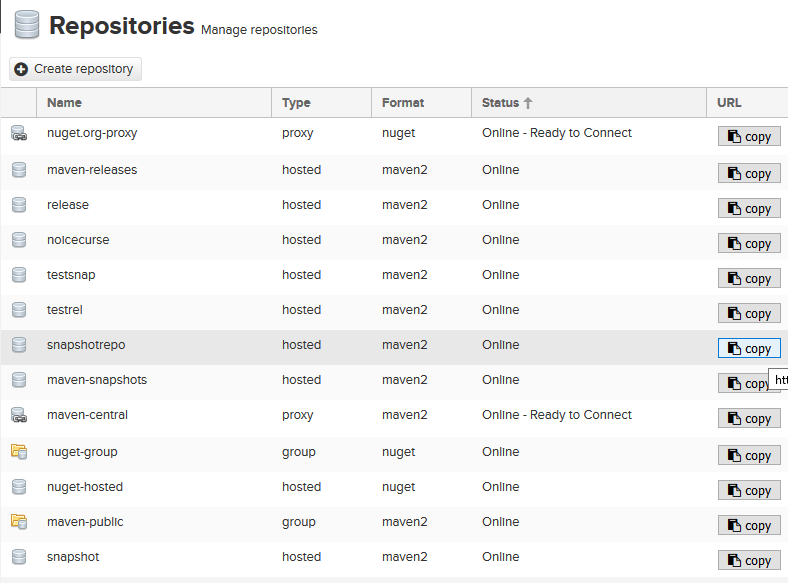
**Depoyment Policy:** Select **Allow Redeploy** to allow redeploy on new build.

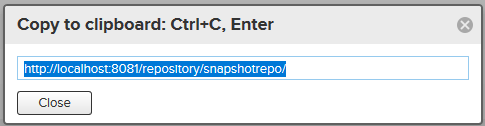
****

****

Click on **Create Repository.**

1. Find the created repository in the list of repositories and click on **Copy** button under **URL** heading.

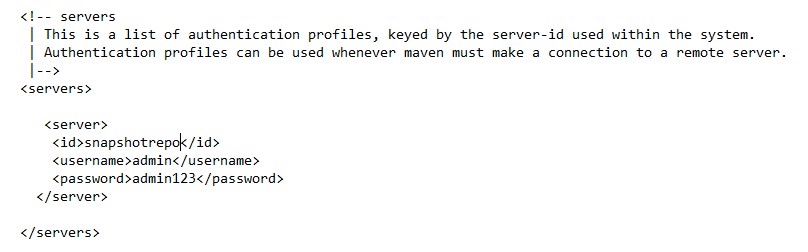




1. Configure the copied snapshot repository link with the POM.xml file of Maven and the settings in Maven configuration.
2. Go to Apache Maven Binaries on local machine. Go in the **conf** folder and open **settings.xml** file

**$ cd <path of maven>/conf/**

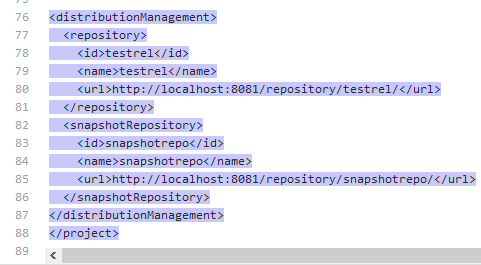
Make the following edits in the settings.xml file under <servers> tag



Here, the server tag has three elements:

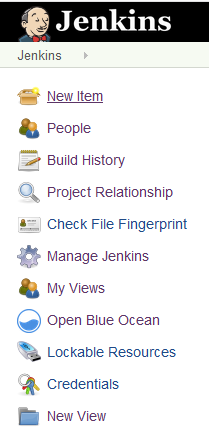
* Id: the **name of the snapshot repository** defined in Nexus
* Username: Nexus user having ownership of the repository
* Password: password of above user for authentication.

1. Configure the POM.xml file on GitHub as:

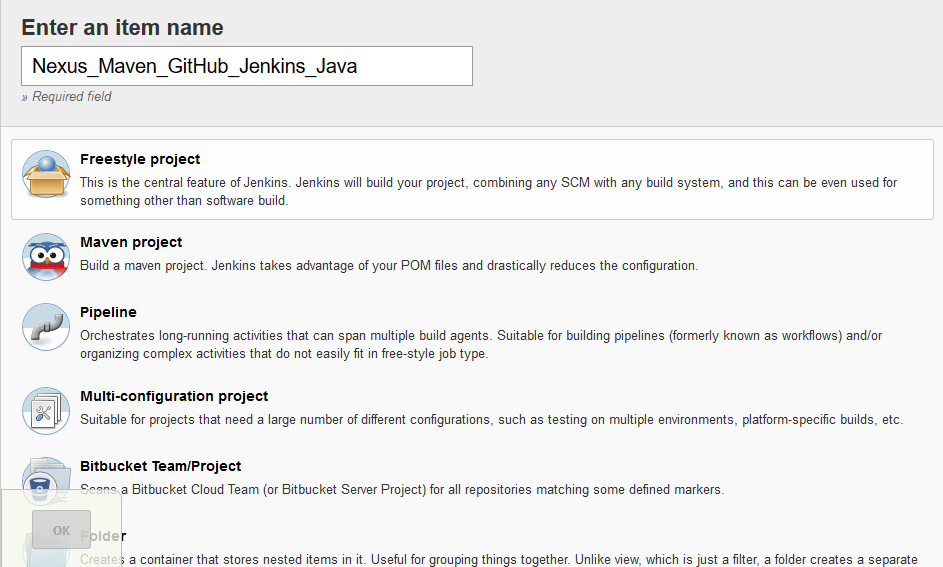


Here, id and name are the name of the repository and the url is of the repository that we copied.

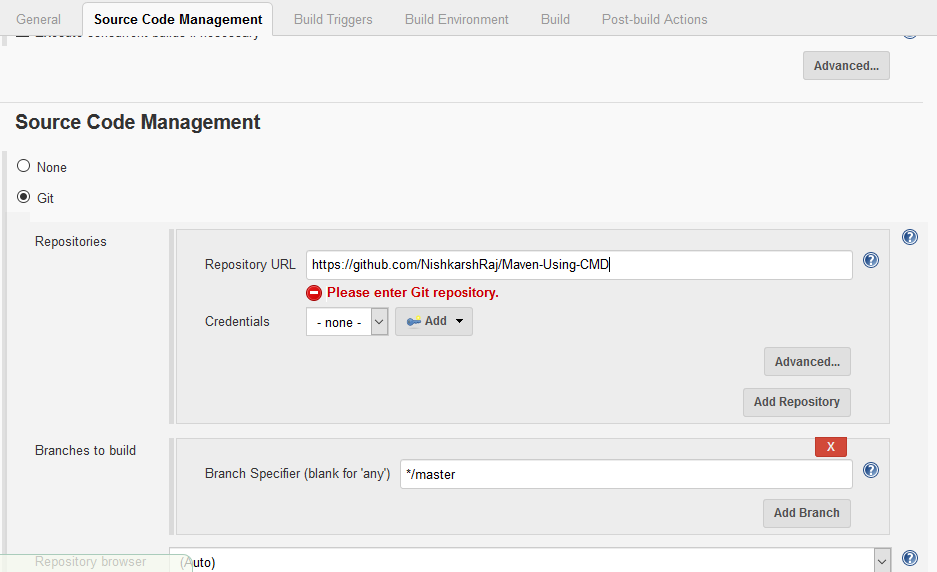
1. Automating the build using Jenkins
2. Create a new job in Jenkins



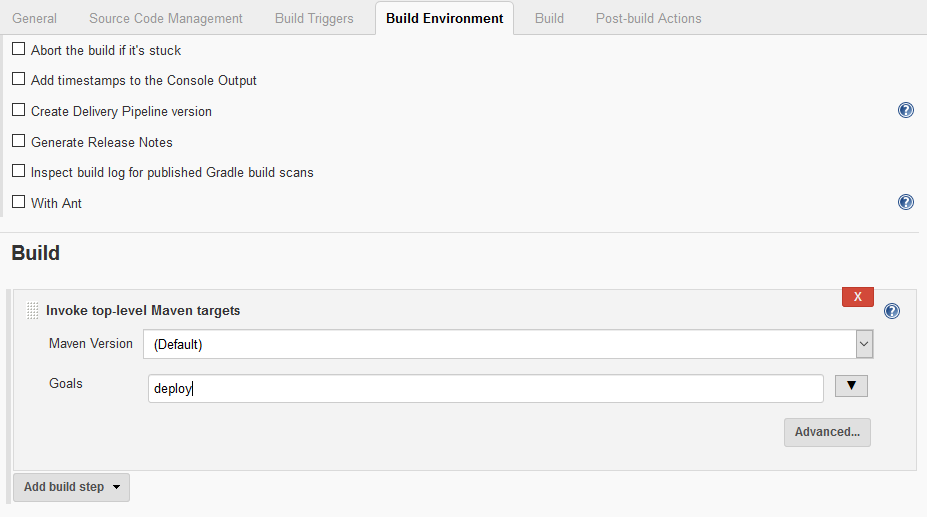
1. Enter name of the Jenkins Job and select **Freestyle Project.**



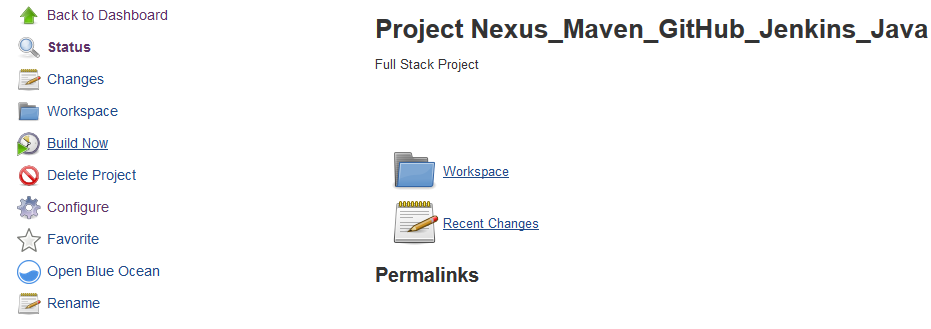
1. Configure the project: In the **Source Code Management** Configuration Tab, add Git and specify the GitHub link of the project



1. In the **Build Environment**, In the **Build** menu, select **Invoke High level Maven targets** and specify **deploy** goal of Maven



1. Click on **Save**. The job is created. Click on **Build Now.**



1. Check the console output:

