**Dockerizing Jenkins Pipeline**

**Description:**

Demonstrate the continuous integration and delivery by Dockerizing Jenkins Pipeline.

**Step 1: Setting up workspace and github repository**

1. Open VS Code
2. Create a directory called dockerizing-jenkins-pipeline in the terminal and change the directory
3. Run “git init” to initialize git repository
4. Create a repository “dockerizing-jenkins-pipeline” in github.com
5. Execute the following commands to push the local repository to github:

$ git add .

$ git commit -m “initial commit”

$ git branch -M master

$ git remote add origin [git@github.com:poojan007/dockerizing-jenkins-pipeline.git](mailto:git@github.com:poojan007/dockerizing-jenkins-pipeline.git)

$ git push -u origin master

**Step 2: Installation and configuration of Jenkins**

1. To use the Debian repository of Jenkins to automate installation and upgrade, first we need to add the key to our system using the following command:

$ wget -q -O – <https://pkg.jenkins.io/debian-stable/jenkins.io.key> | sudo apt-key add –

1. Then, adding the following entry into my **/etc/apt/sources.list**:

$ sudo vi /etc/apt/sources.list

**Now add the following command:**

deb [https://pkg.jenkins.io/debian-stable binary/](https://pkg.jenkins.io/debian-stable%20binary/)

Save and exit using :wq!

1. Update your local package index:

$ sudo apt-get update

1. We need JDK to run Jenkins, to install it we use the following commands:

$ sudo apt-get install openjdk-8-jdk

1. Finally, install Jenkins:

$ sudo apt-get install jenkins

Once Jenkins is installed, open a browser and enter the

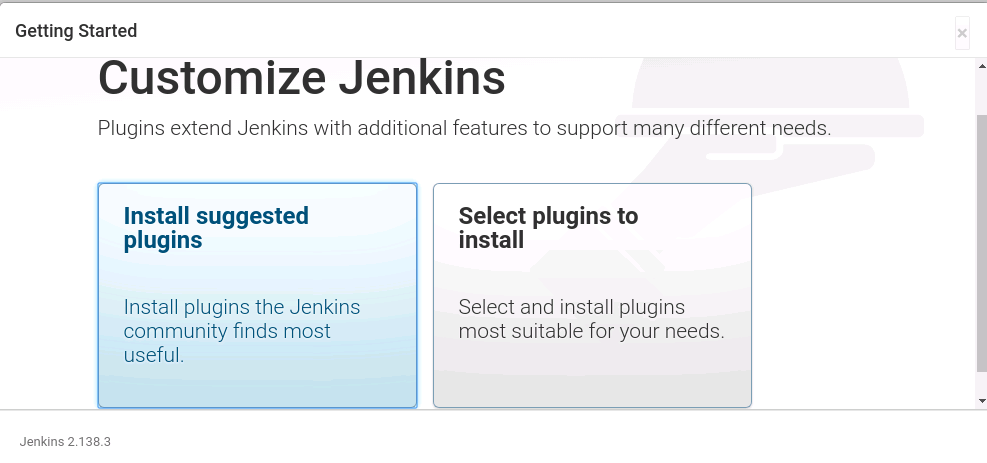
URL: x.x.x.x:8080

Replace x.x.x.x with our machine IP address

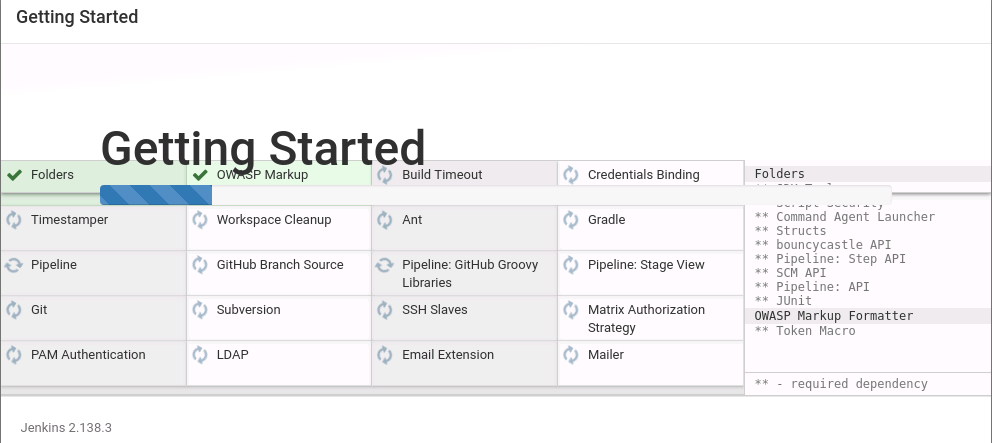
You will need to enter the admin password, to find the password for initial setup, use the following command:

$ sudo cat /var/lib/Jenkins/secrets/initialAdminPassword

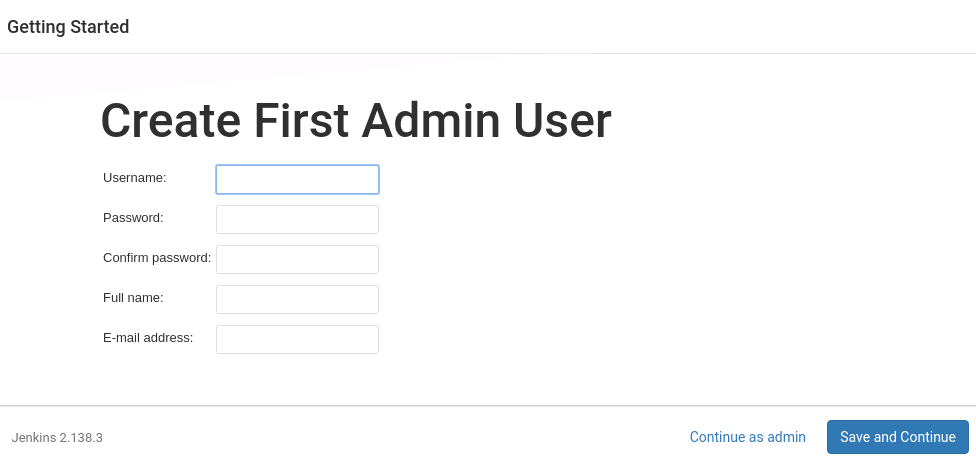
1. Once you are logged in, you will be redirected to the page below:



Select Install suggested plugins. You will be redirected to the page below:



After installing recommended plugins, you can create “First Admin User” or continue as Admin by filling the required details as shown in the screenshot below:



After all the steps are executed we can start using jenkins.

**Step 3: Installation and configuration of Docker**

1. Update software repositories as follows:

$ sudo apt-get update

1. Install docker using the following command:

$ sudo apt-get install docker

1. Start and automate docker as follows:

$ sudo systemctl start docker

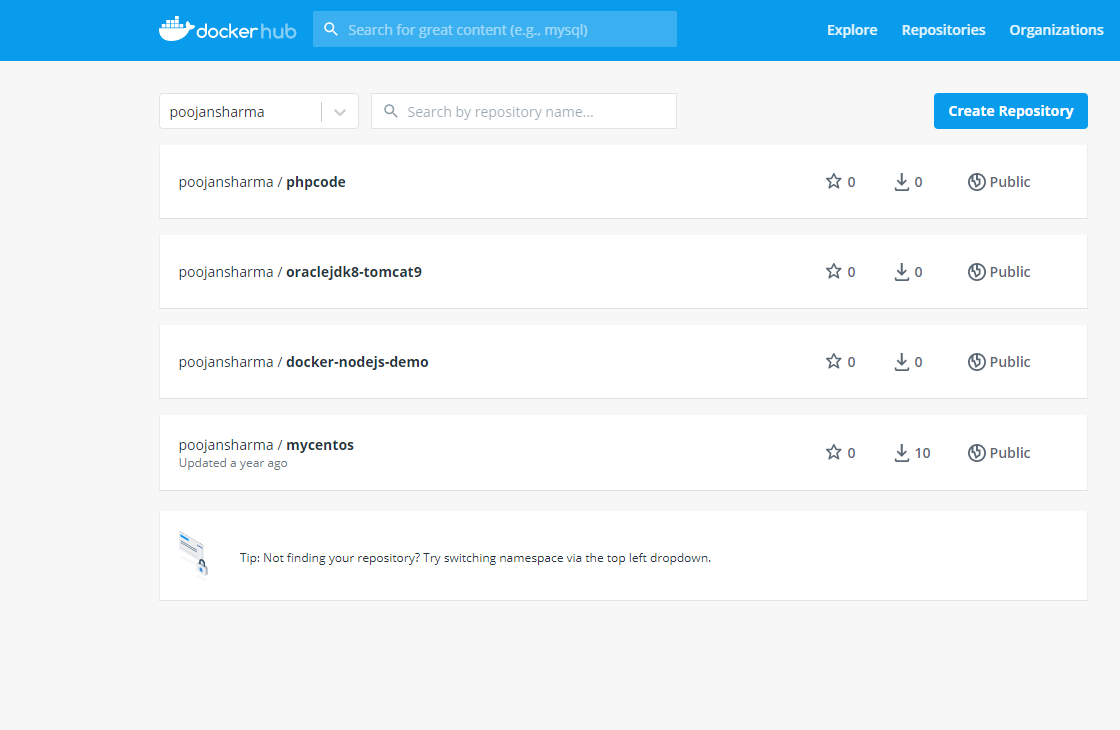
$ sudo systemctl enable docker

1. To verify if installation is successful check the version of docker installed:

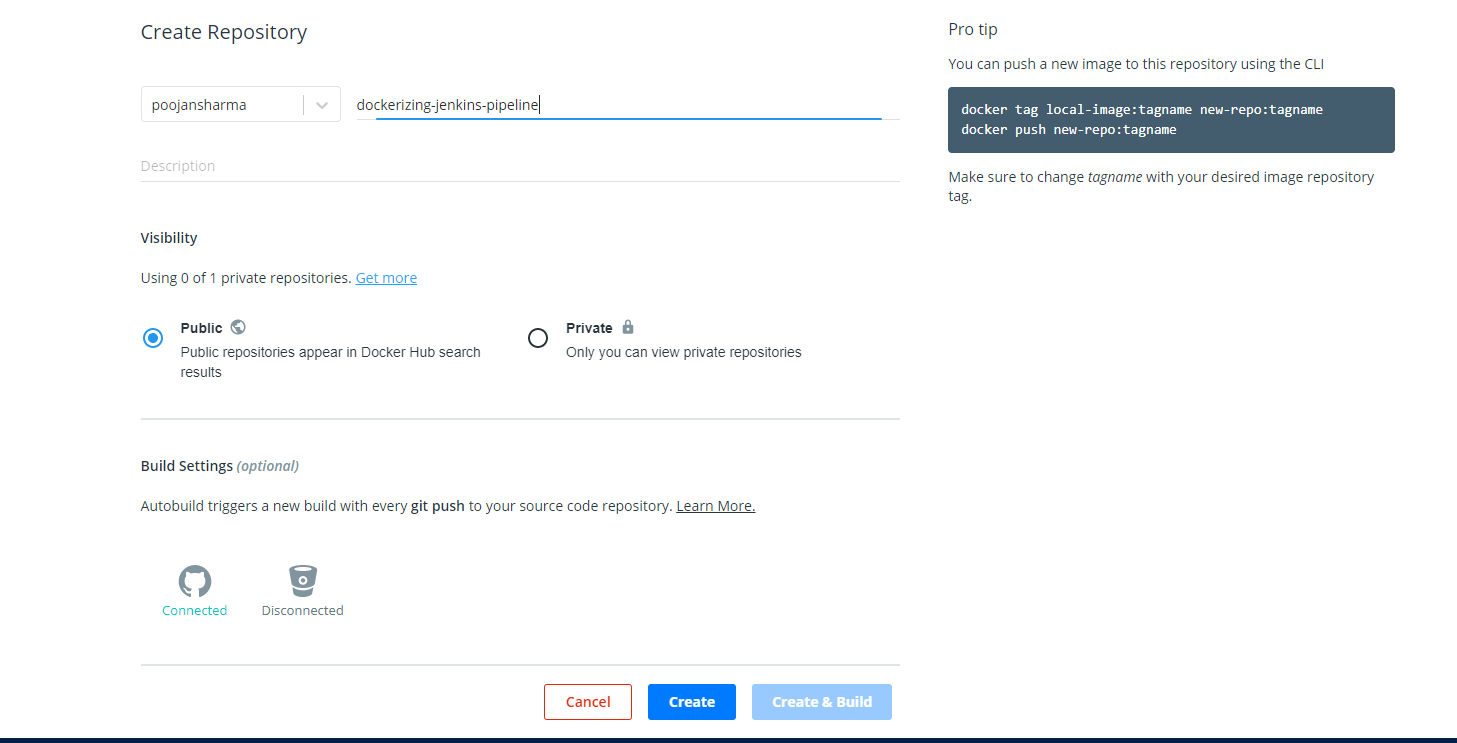
$ docker --version

**Step 4: Creating a new repository in DockerHub**

1. Login to docker hub (<https://hub.docker.com>)



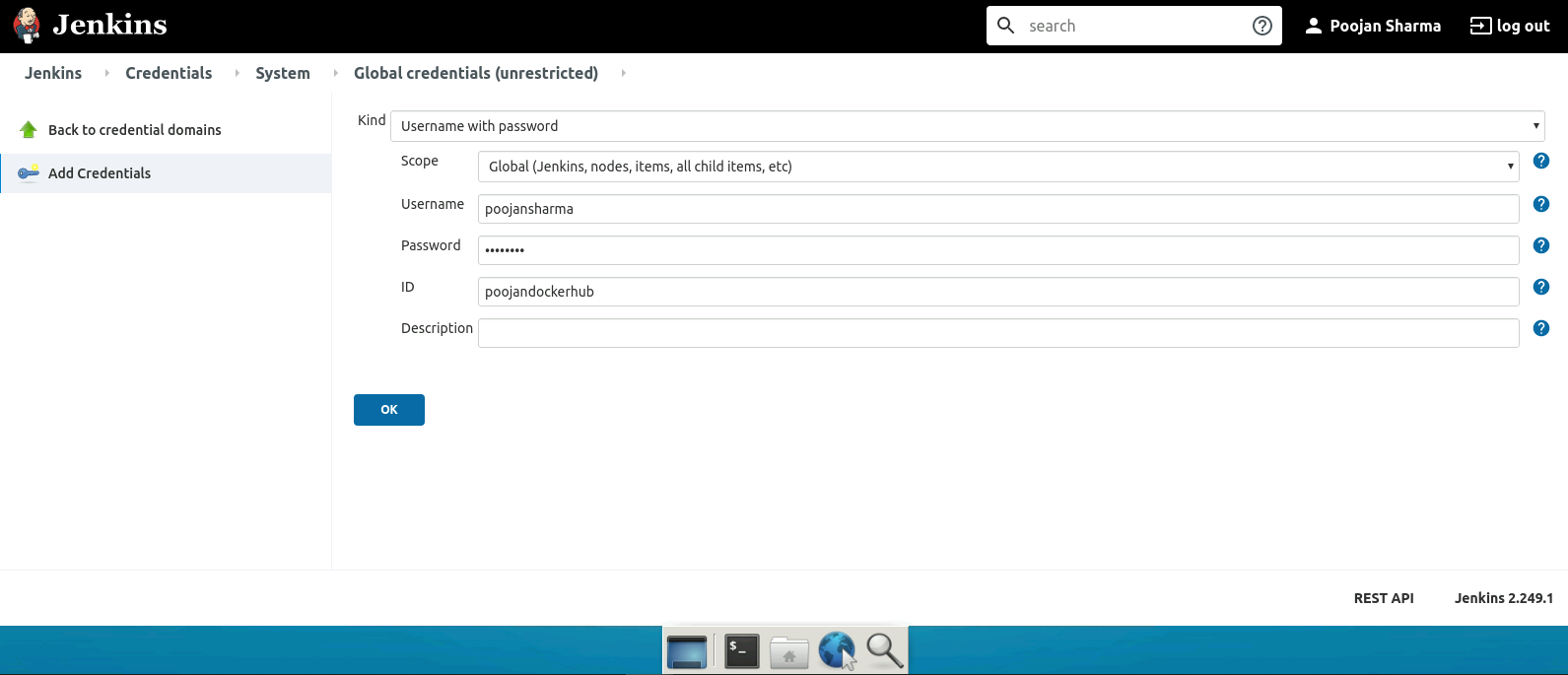
Click on the “Create Repository” button to create a new repository as follows:



**Step 5: Configuring dockerhub credentials in jenkins**

From the jenkins dashboard go to the following location and add your dockerhub credentials as shown in the following screenshot.

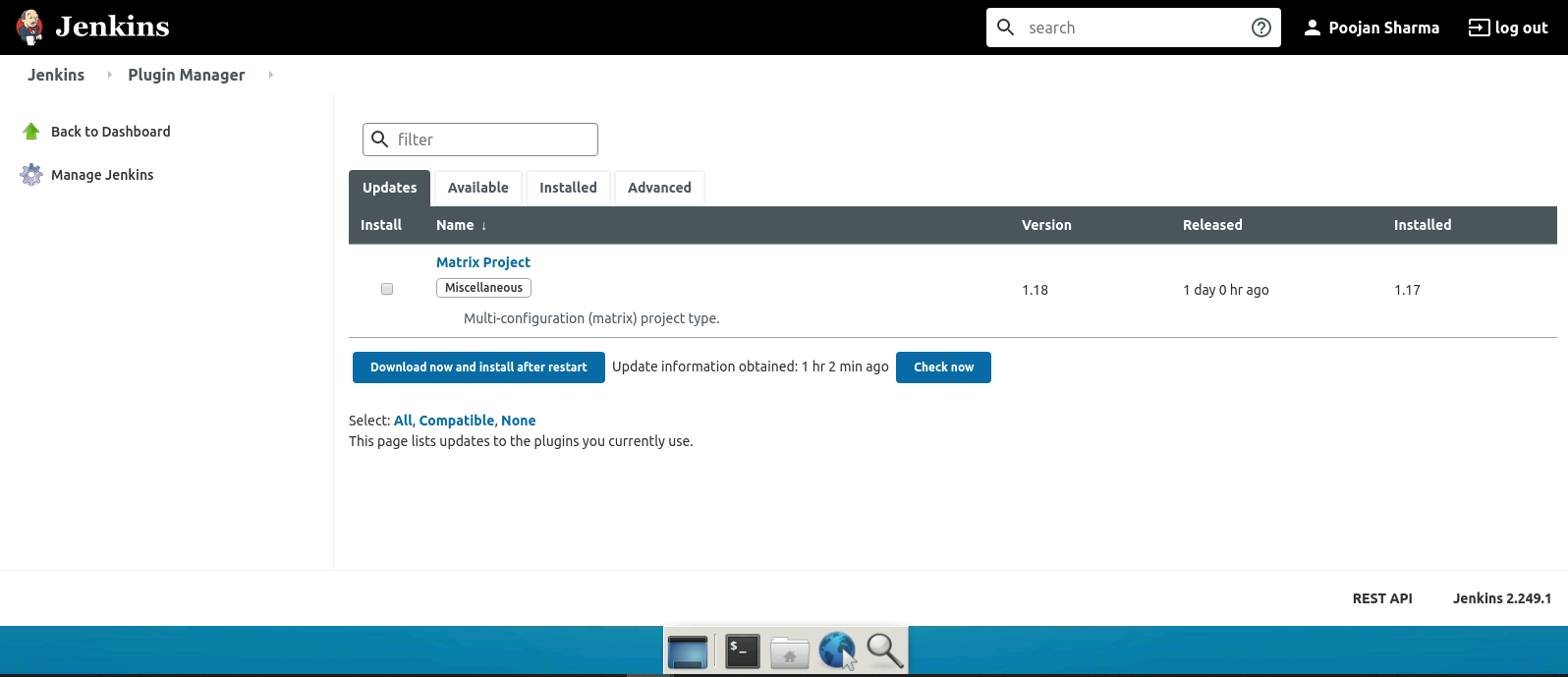
**Location**: Jenkins > Credentials > System > Global credentials



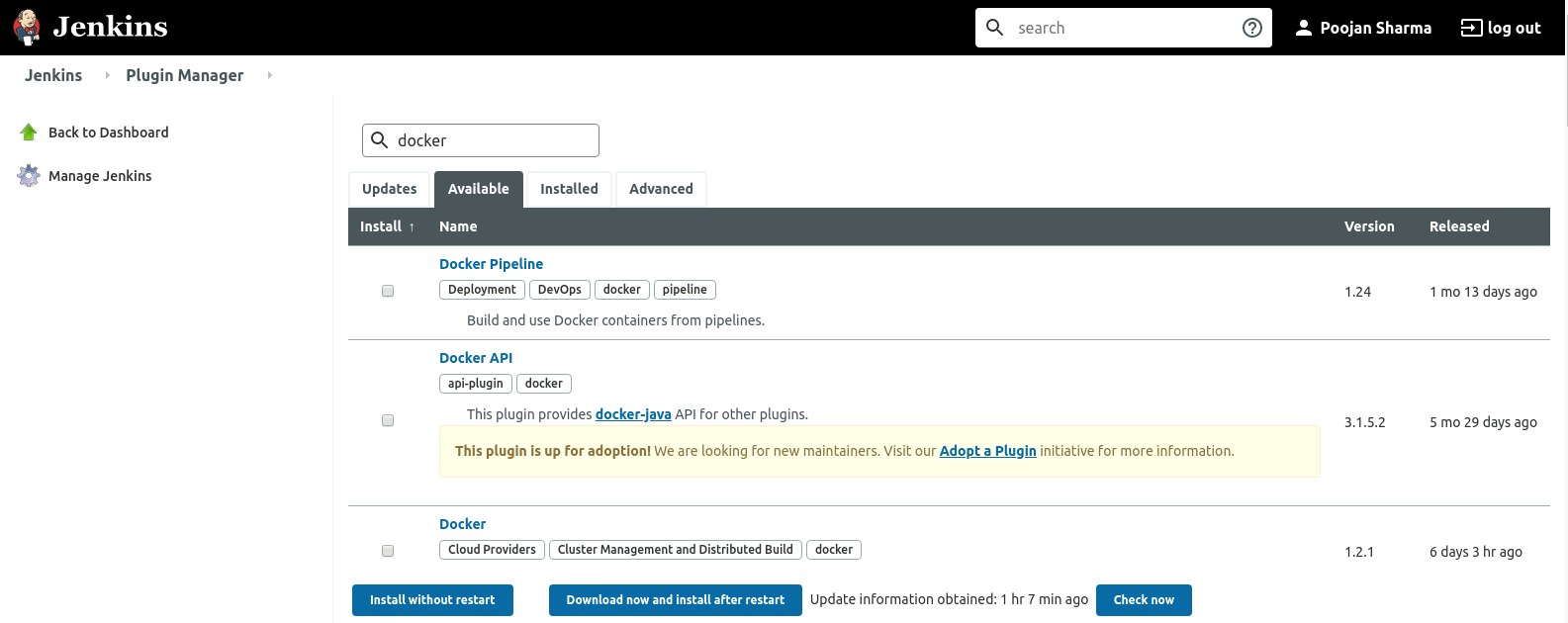
**Step 6: Installing Docker Pipeline plugin in jenkins**

1. From the jenkins dashboard go to

Manage Jenkins > Manage Plugins



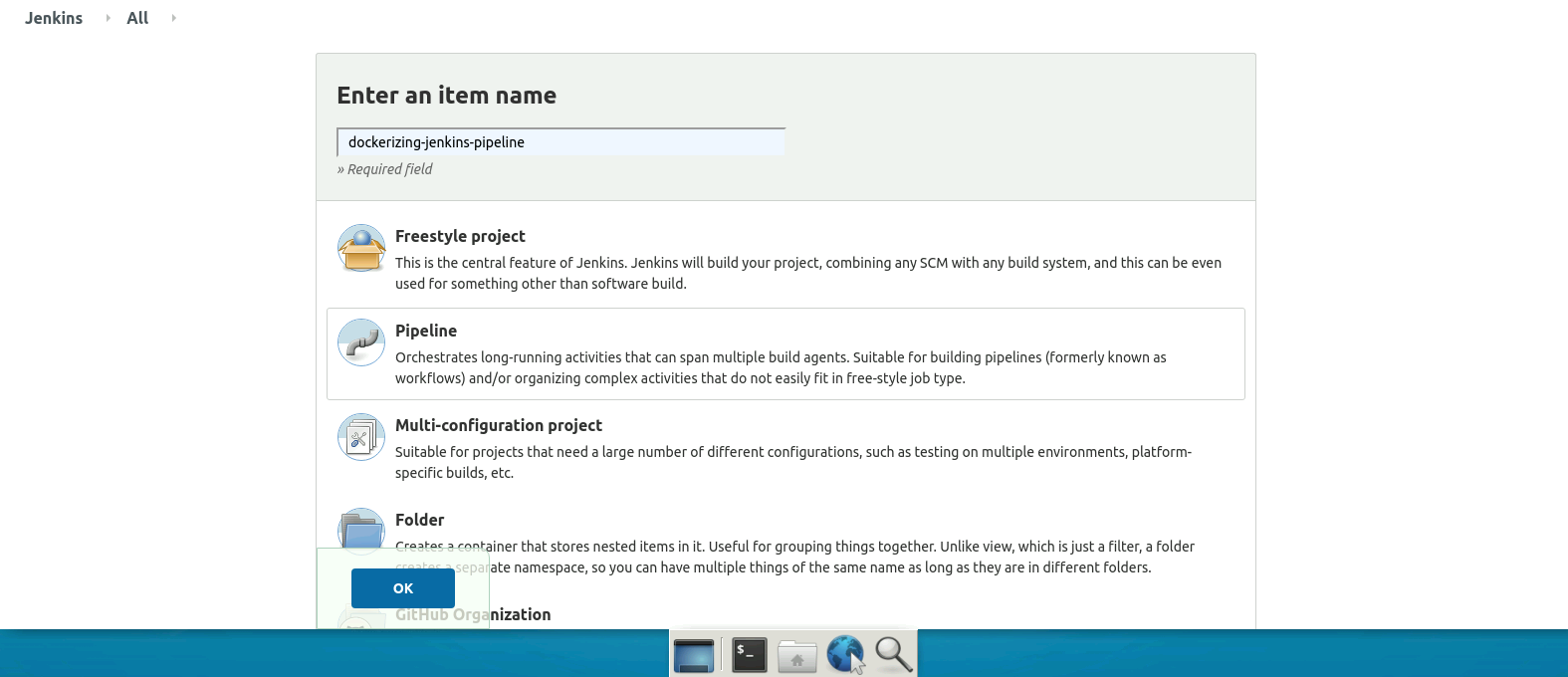
Now select the “Available” tab and search for “Docker”



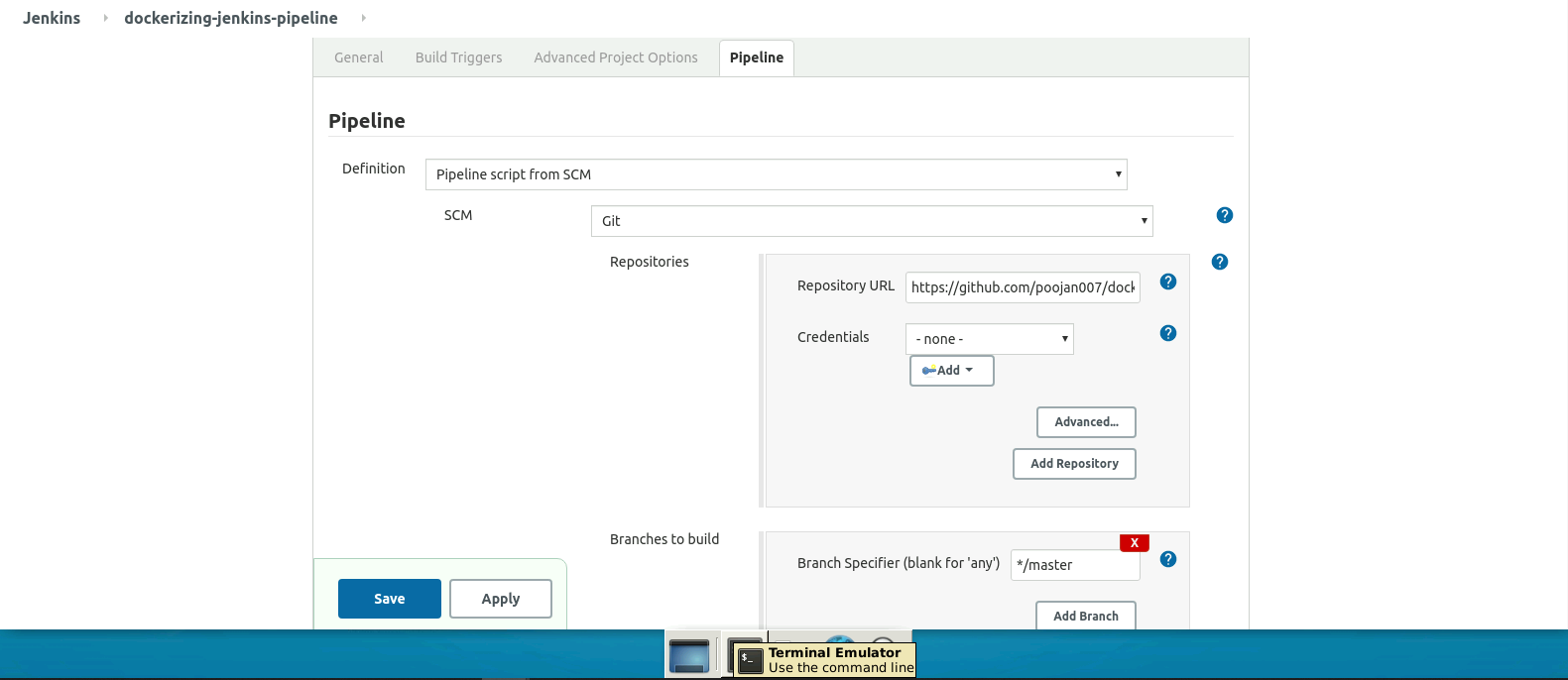
Now check the plugin “Docker Pipeline” and click on “Install without restart” button to install the selected plugin.

**Step 7: Configuring Jenkins pipeline**

1. On the jenkins dashboard click on “New Item” and enter and select the following details to setup a new pipeline.



Will see the following page and then select “Pipeline” from the available tabs



1. Once the pipeline is configured, come back to the project page and click on “Build Now” to start the pipeline and execute the steps mentioned in the Jenkinsfile present in the repository present in GitHub.

