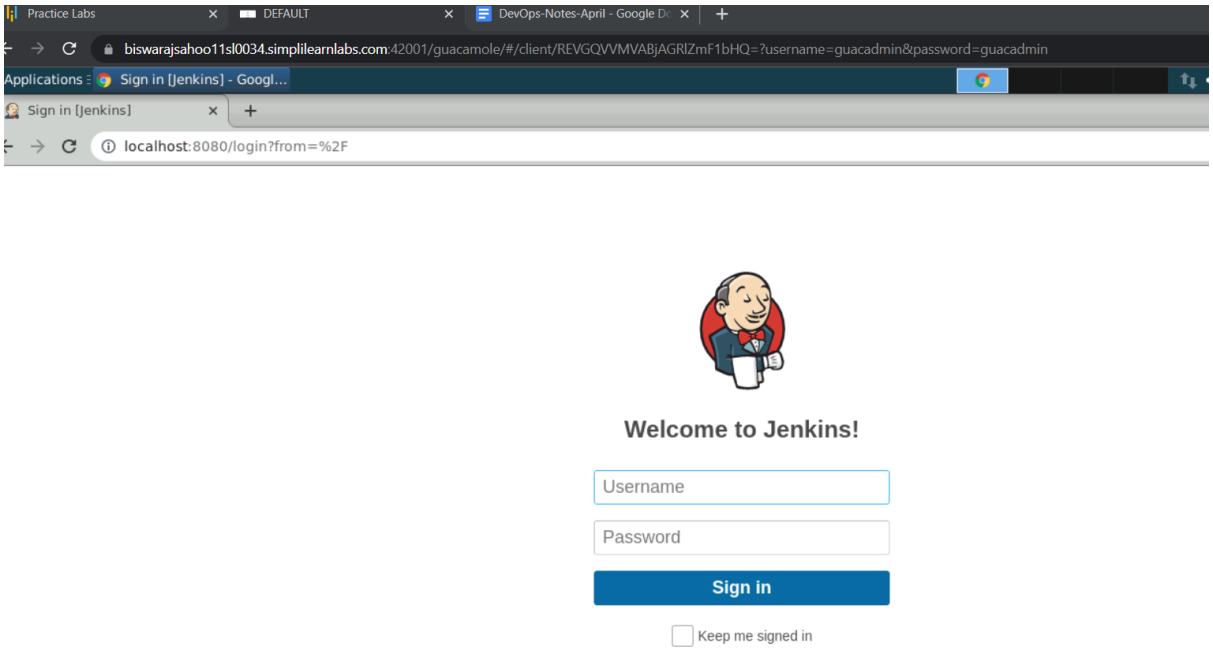
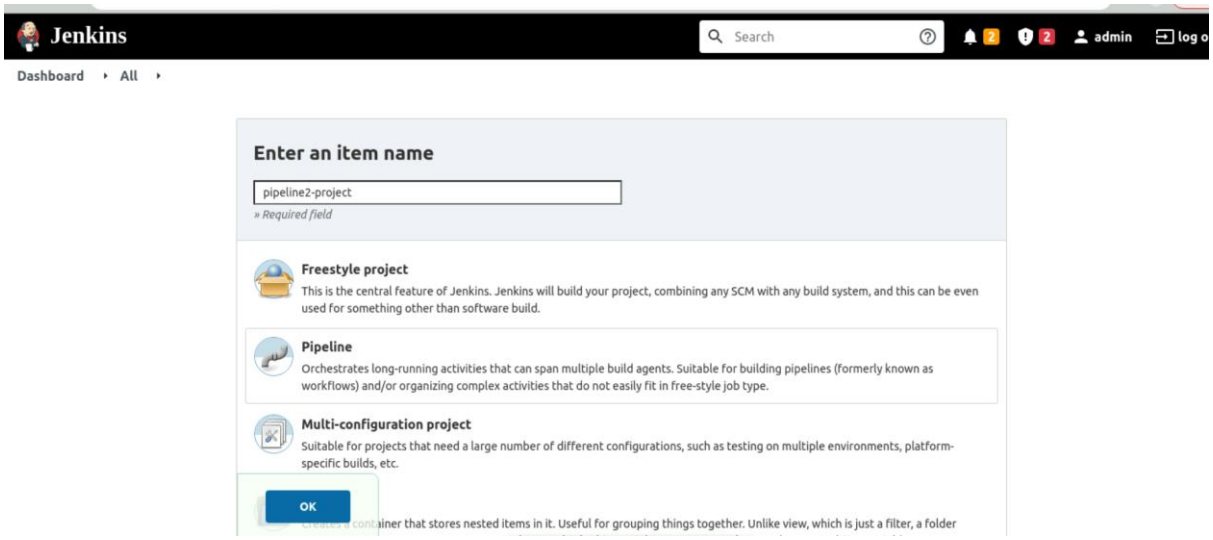
Build a Docker Jenkins Pipeline to Implement CI/CD Workflow

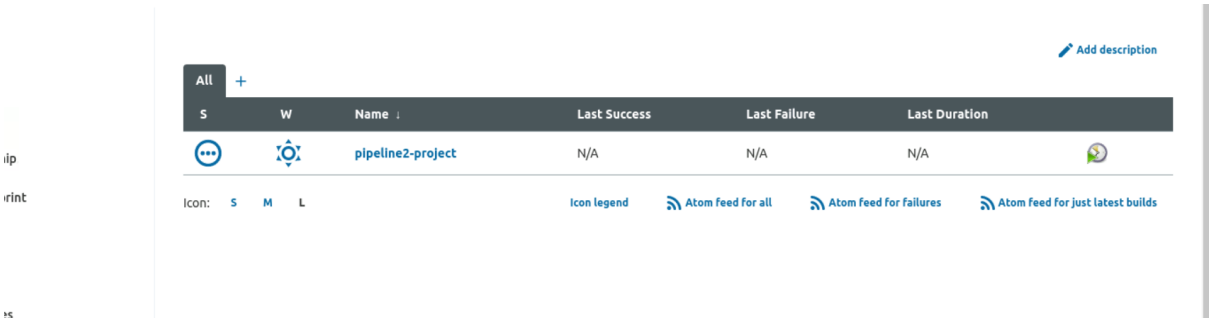
First step we have to go to Jenkins login in browser by giving localhost:8080



Here we have to create a new pipeline named as pipeline2-project

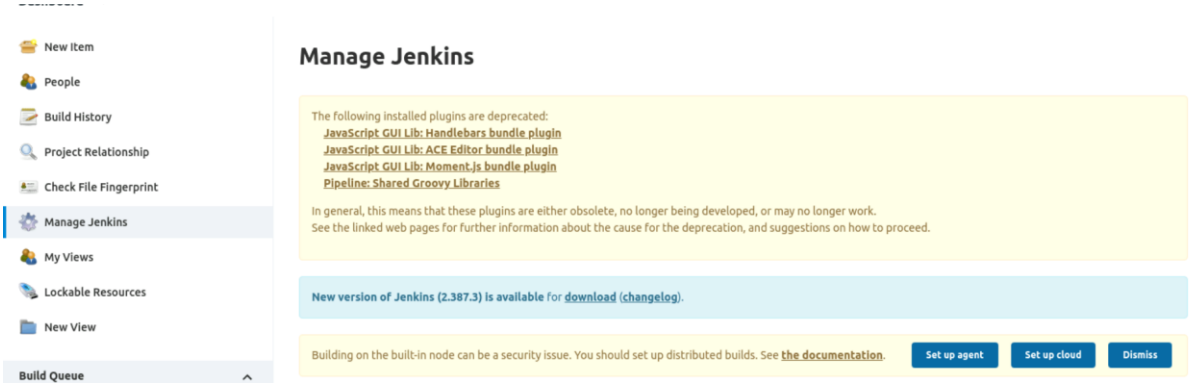


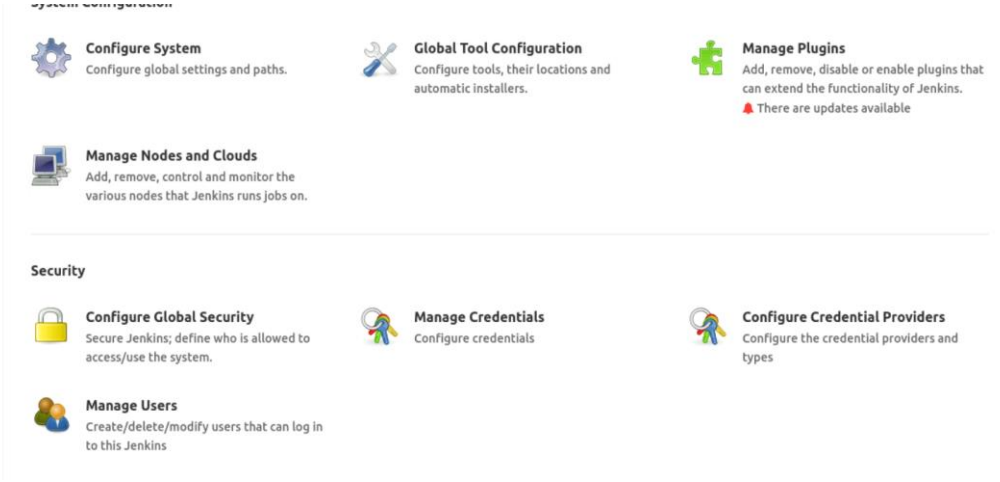
Pipeline2-project created

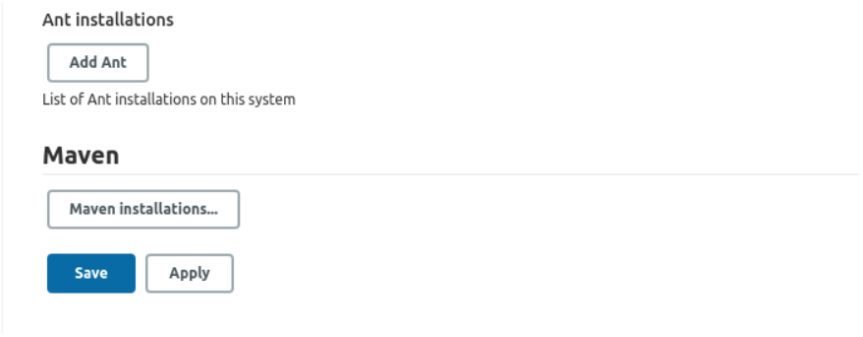


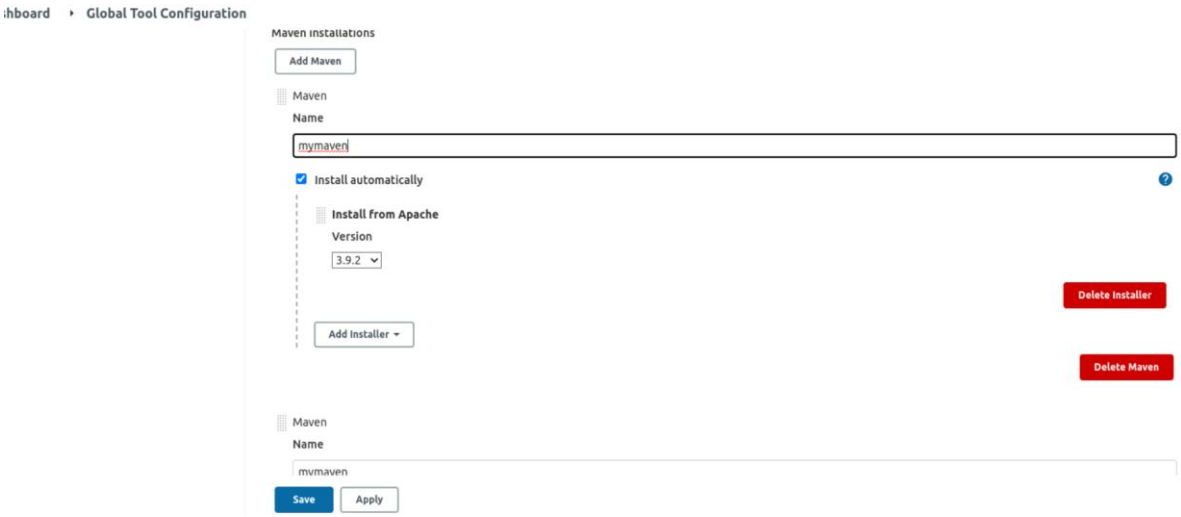
Now we have to set up maven tool fir the project pipeline

> Set up Maven tool in Manage jenkins --> configure global tools--> scroll down --> add maven --> give tool name as mymaven--> save the chnages


New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Lockable Resources
New View
Build Queue
Manage Jenkins
The following installed plugins are deprecated:
JavaScript GUI Lib: Handlebars bundle plugin
JavaScript GUI Lib: ACE Editor bundle plugin
JavaScript GUI Lib: Moment.js bundle plugin
Pipeline: Shared Groovy Libraries
In general, this means that these plugins are either obsolete, no longer being developed, or may no longer work.
See the linked web pages for further information about the cause for the deprecation, and suggestions on how to proceed.
New version of Jenkins (2.387.3) is available for download (changelog).
Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.
Set up agent
Set up cloud
Dismiss

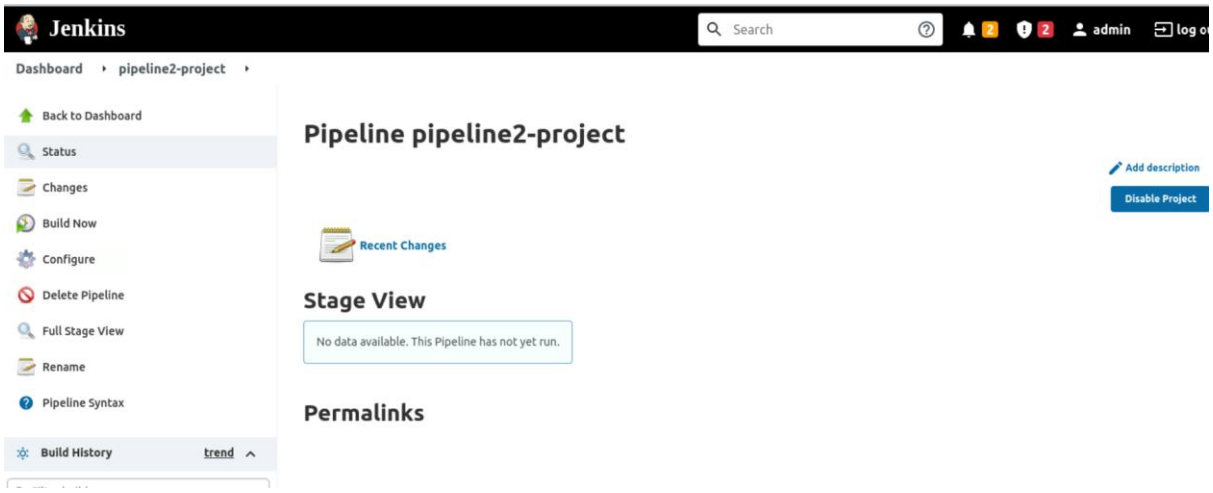

Configure System
Configure global settings and paths.
Global Tool Configuration
Configure tools, their locations and automatic installers.
Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
There are updates available
Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
Security
Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.
Manage Credentials
Configure credentials
Configure Credential Providers
Configure the credential providers and types
Manage Users
Create/delete/modify users that can log in to this Jenkins

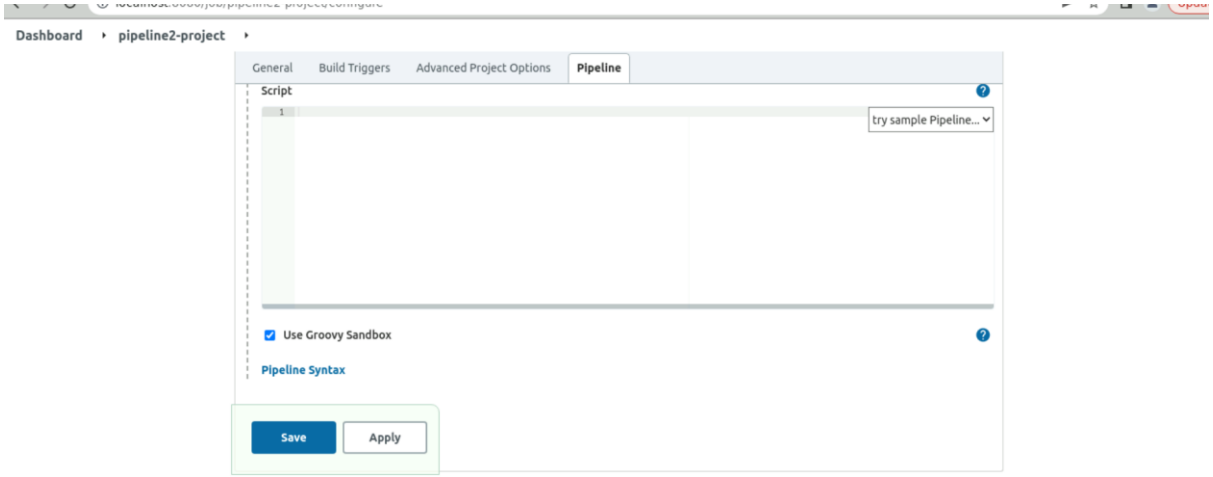

Ant installations
Add Ant
List of Ant installations on this system
Maven
Maven installations...
Save
Apply

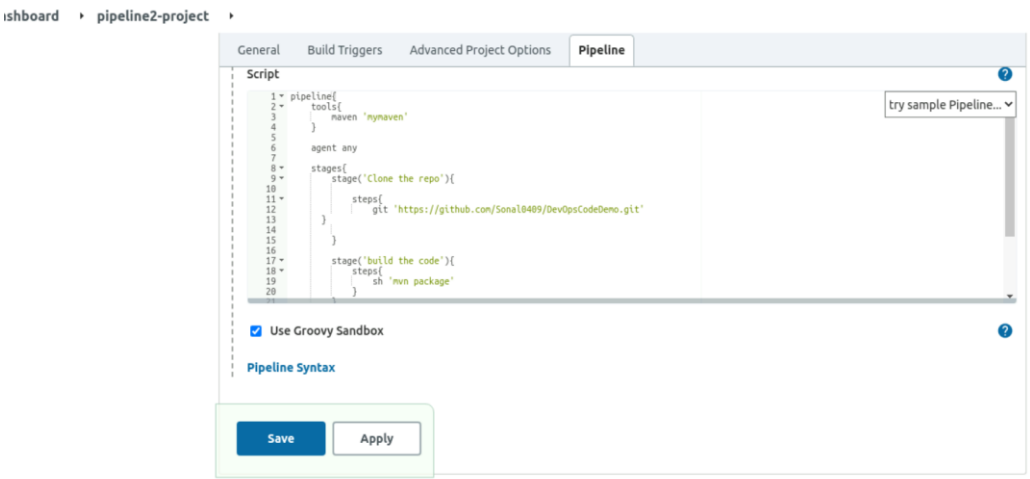


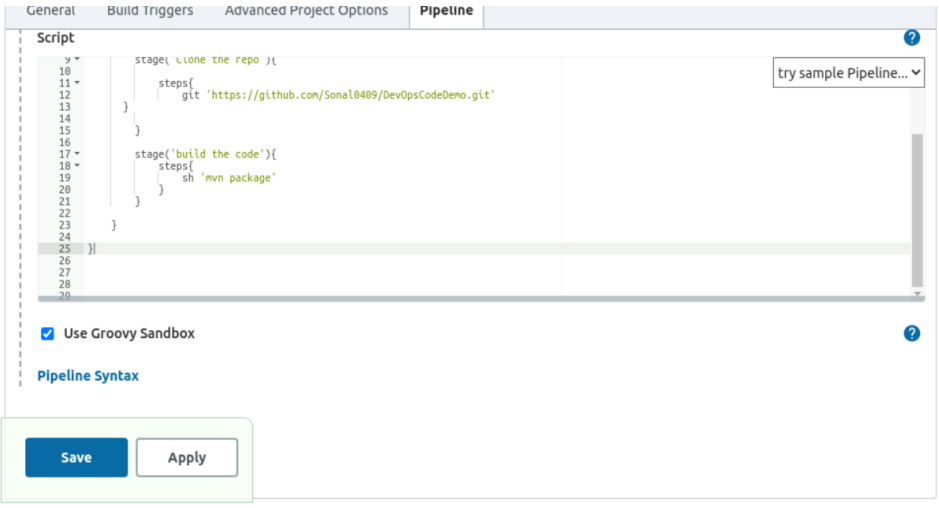
Named the maven tool as “mymeven”

Now w will go to pipeline to build the pipeline



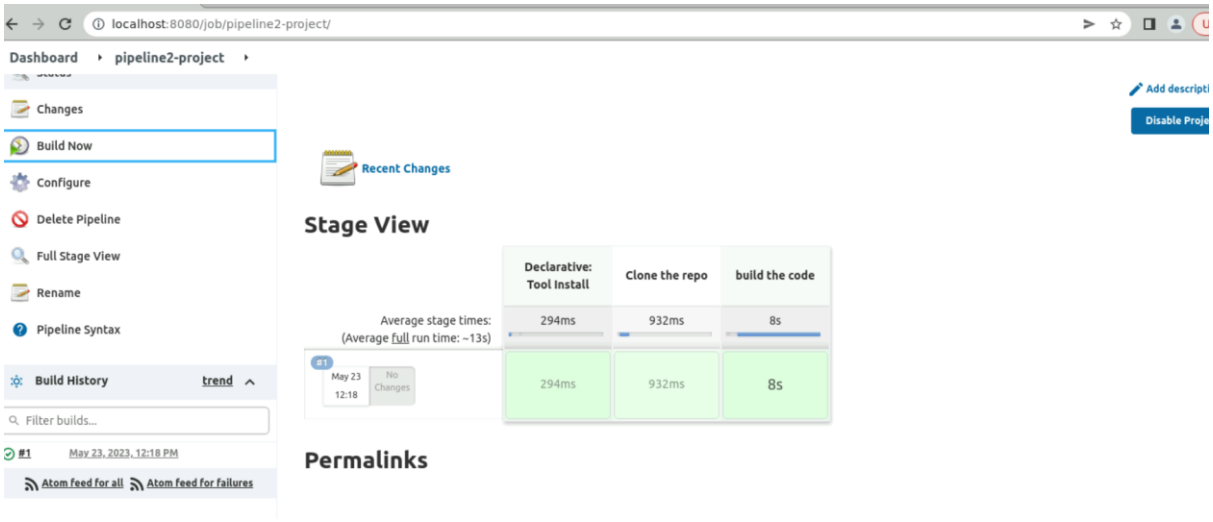


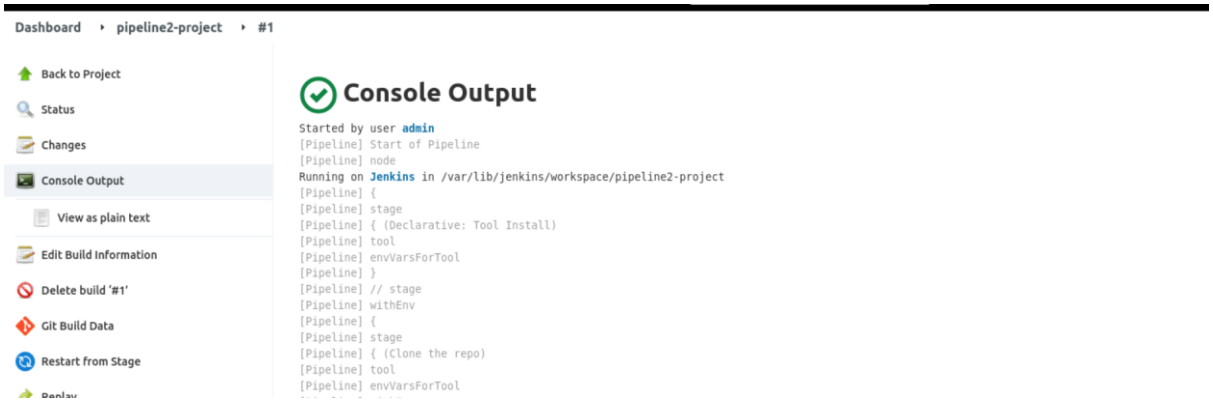




Save it

Now we have to build the pipeline



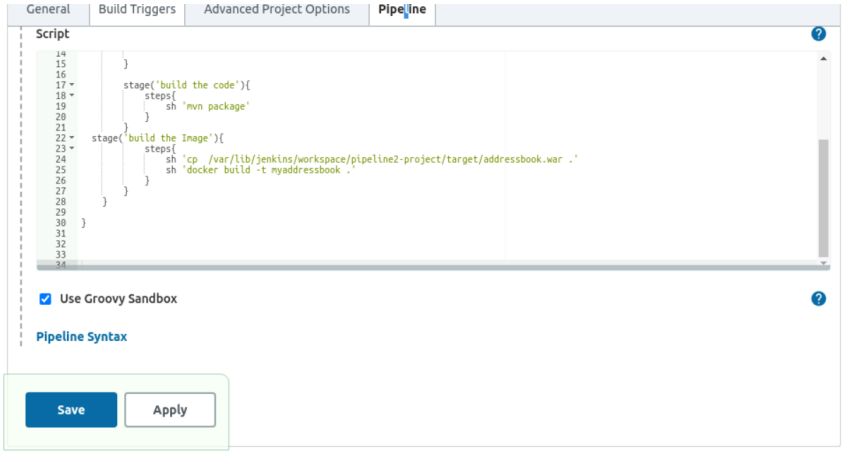


Go to console output of the job, copy the path of war file generated

```
Running com.mydemo.utilities.TestGenericComparator
Tests run: 14, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.042 sec - in com.mydemo.utilities.TestGenericComparator

Results :

Tests run: 23, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- war:3.3.2:war (default-war) @ addressbook ---
[INFO] Packaging webapp
[INFO] Assembling webapp [addressbook] in [/var/lib/jenkins/workspace/pipeline2-project/target/addressbook]
[INFO] Processing war project
[INFO] Building war: /var/lib/jenkins/workspace/pipeline2-project/target/addressbook.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.527 s
[INFO] Finished at: 2023-05-23T12:19:06Z
[INFO] ------------------------------------------------------------------------
[WARNING]
[WARNING] Plugin validation issues were detected in 5 plugin(s)
[WARNING]
```

```
[WARNING]  * org.apache.maven.plugins:maven-resources-plugin:3.3.0
[WARNING]  * org.apache.maven.plugins:maven-enforcer-plugin:1.0
[WARNING]  * org.apache.maven.plugins:maven-war-plugin:3.3.2
[WARNING]
[WARNING] For more or less details, use 'maven.plugin.validation' property with one of the values (case insensitive): [
VERBOSE]
[WARNING]
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Again we have to build the pipeline

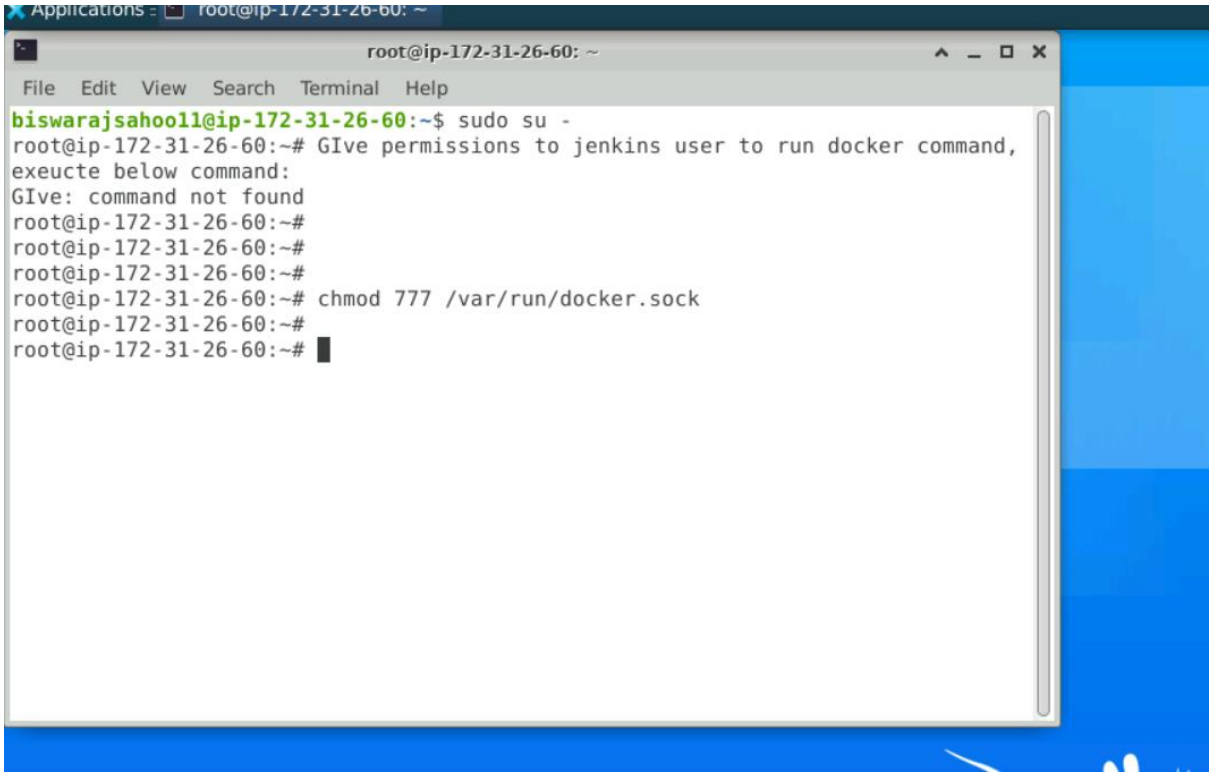



```
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (build the Image)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ cp /var/lib/jenkins/workspace/pipeline2-project/target/addressbook.war .
[Pipeline] sh
+ docker build -t myaddressbook .
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post
"http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?
buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=dockerfile&labels=%7B
%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsize=0&t=myaddressbook&target=&ulimits=null&version=1": dial unix /var/run/docker.sock:
connect: permission denied
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
```

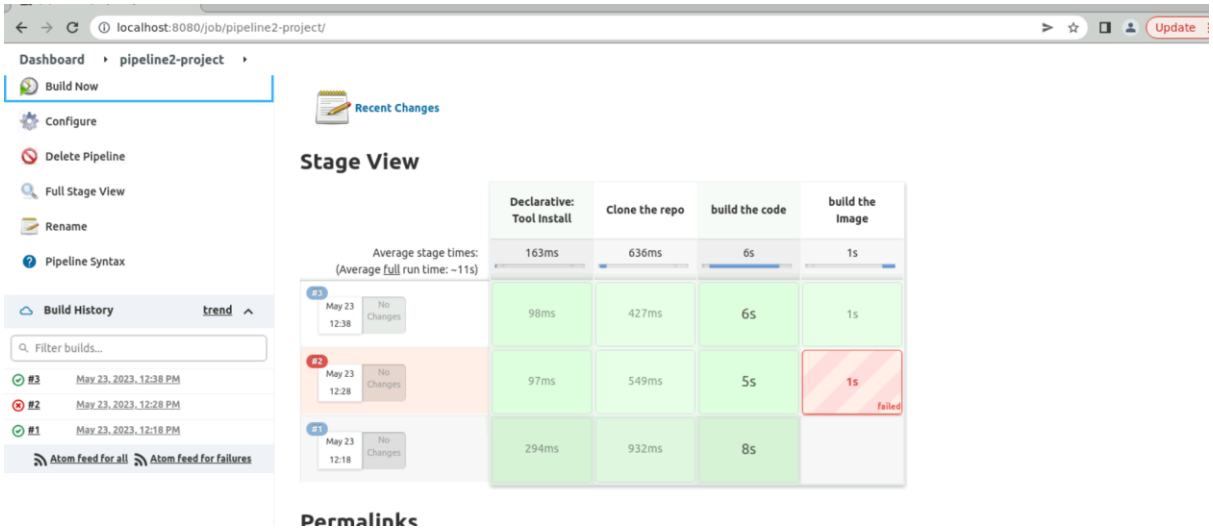
Jenkins cant address the docker image. For this we have to commit a command .

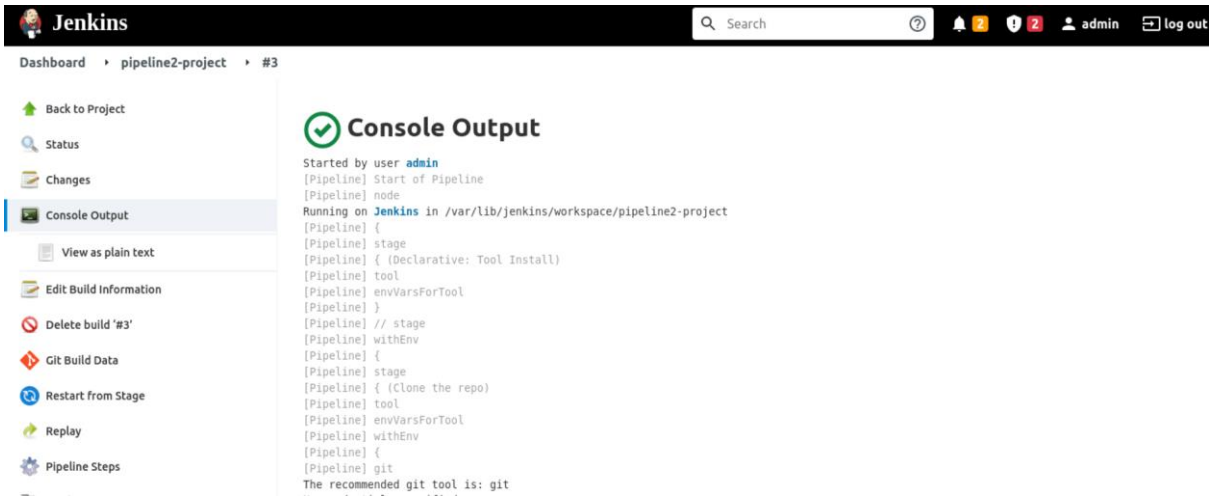
GIve permissions to jenkins user to run docker command, exeucte below command:

chmod 777 /var/run/docker.sock



Now we have to build the pipeline

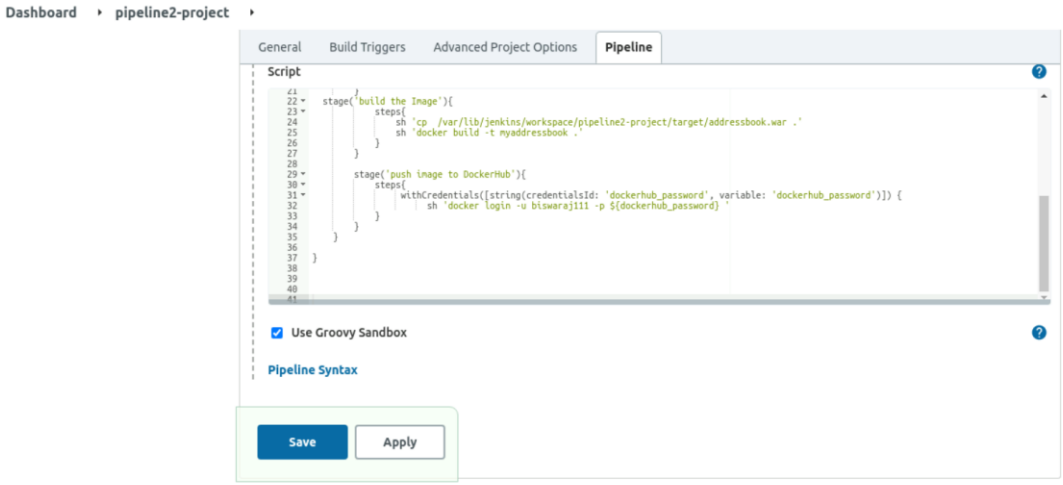


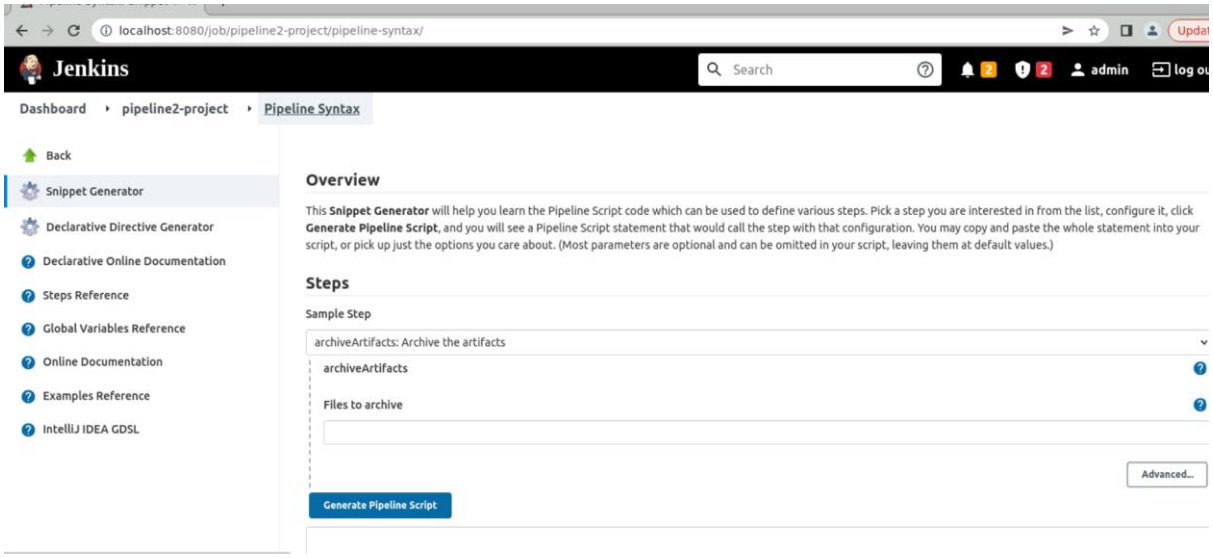


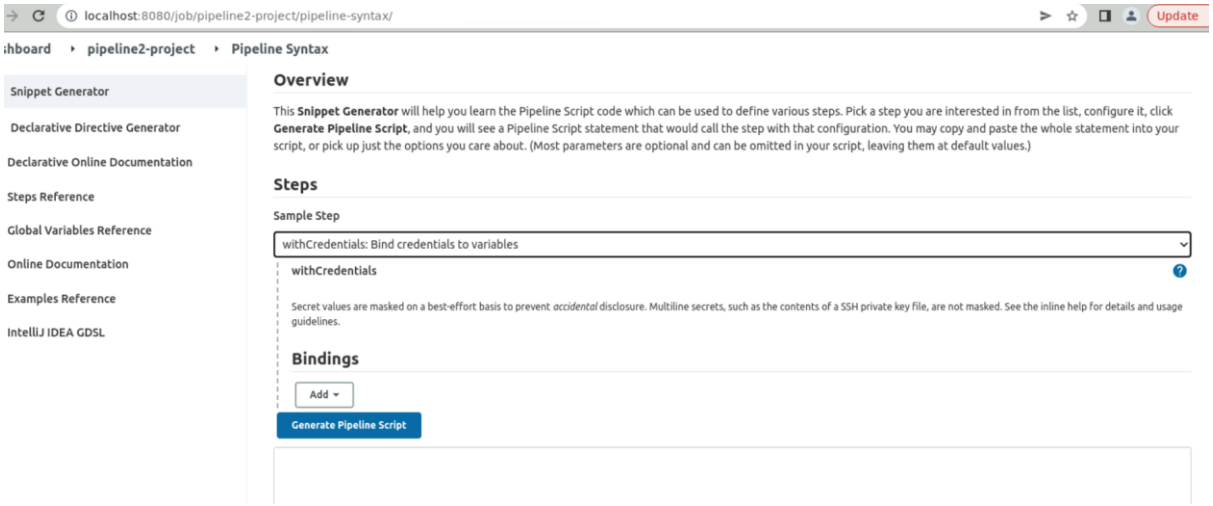
Copy the .war file in current workspace

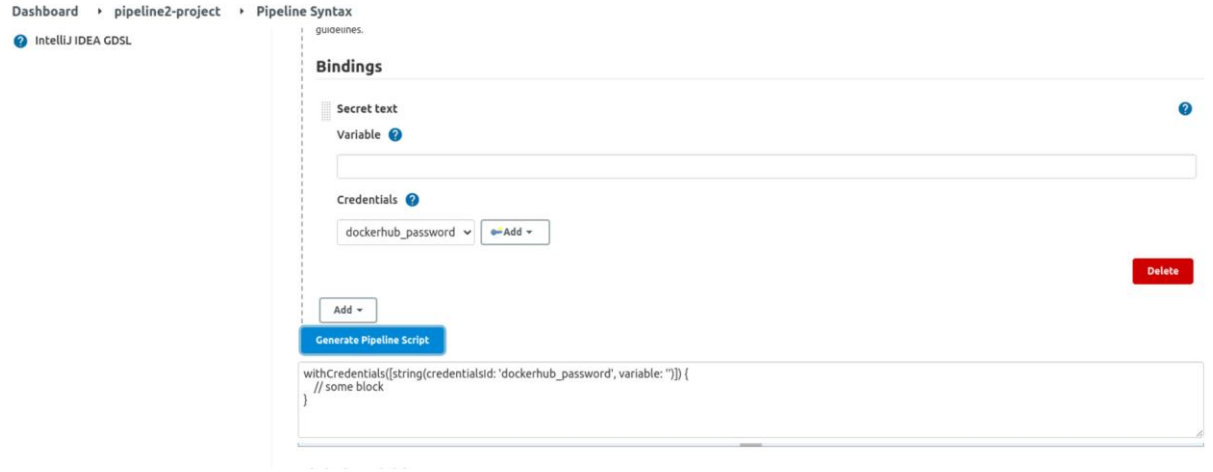
> build the dockerfile which is jobs current workpace

Now again we have to build the pipelinenow we have to push docker image to docker hub account




localhost:8080/job/pipeline2-project/pipeline-syntax/
Jenkins
Search
admin
Dashboard
pipeline2-project
Pipeline Syntax
Back
Snippet Generator
Declarative Directive Generator
Declarative Online Documentation
Steps Reference
Global Variables Reference
Online Documentation
Examples Reference
IntelliJ IDEA GDSL
Overview
This Snippet Generator will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click Generate Pipeline Script, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)
Steps
Sample Step
archiveArtifacts: Archive the artifacts
archiveArtifacts
Files to archive
Advanced...
Generate Pipeline Script


localhost:8080/job/pipeline2-project/pipeline-syntax/
Update
pipeline2-project
Pipeline Syntax
Snippet Generator
Declarative Directive Generator
Declarative Online Documentation
Steps Reference
Global Variables Reference
Online Documentation
Examples Reference
IntelliJ IDEA GDSL
Overview
This Snippet Generator will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click Generate Pipeline Script, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)
Steps
Sample Step
withCredentials: Bind credentials to variables
withCredentials
Secret values are masked on a best-effort basis to prevent accidental disclosure. Multiline secrets, such as the contents of a SSH private key file, are not masked. See the inline help for details and usage guidelines.
Bindings
Add
Generate Pipeline Script

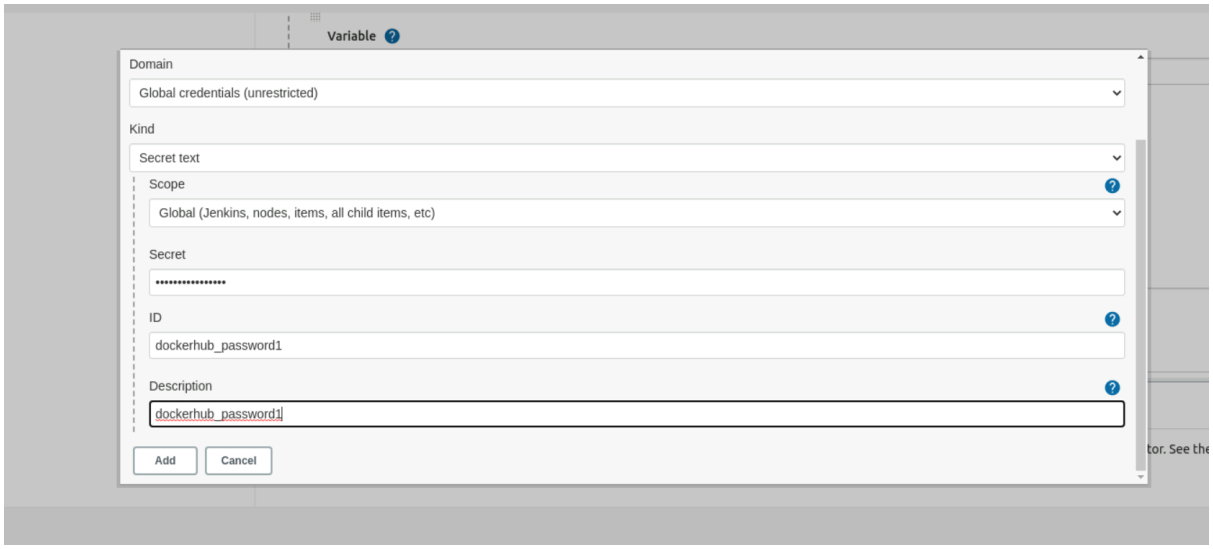

Dashboard
pipeline2-project
Pipeline Syntax
IntelliJ IDEA GDSL
Bindings
Secret text
Variable
Credentials
dockerhub_password
Add
Delete
Add
Generate Pipeline Script
withCredentials([string(credentialsId: 'dockerhub_password', variable: '')]) {
// some block
}

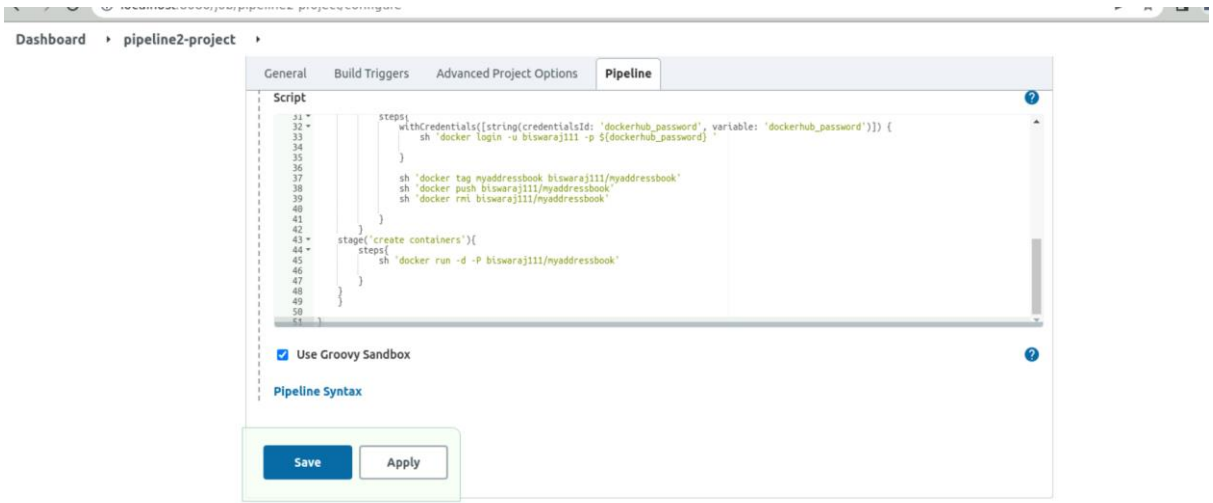
Now generate the pipeline script

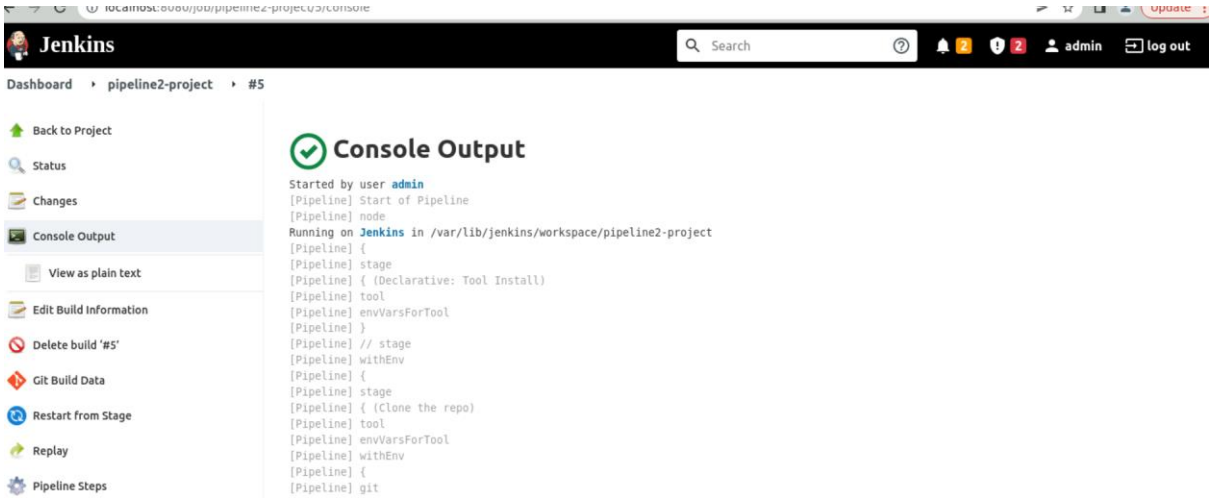
withCredentials([string(credentialsId: 'dockerhub_password', variable: '')]) {

    // some block

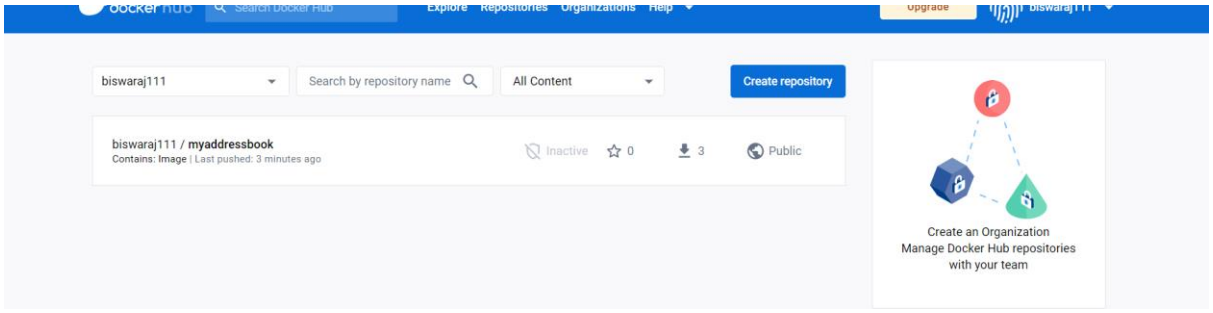
}

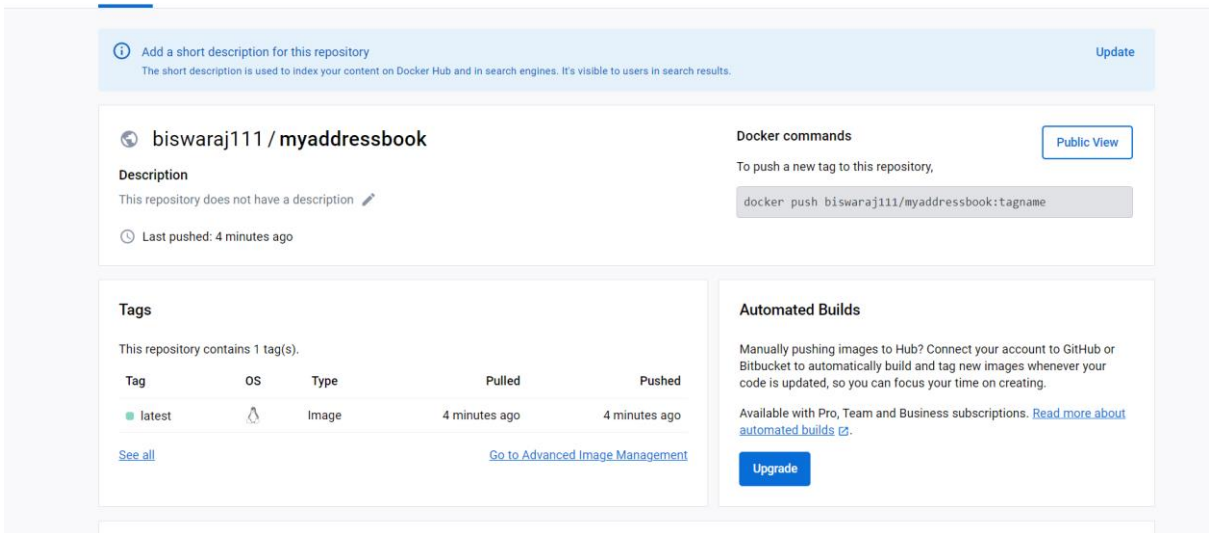




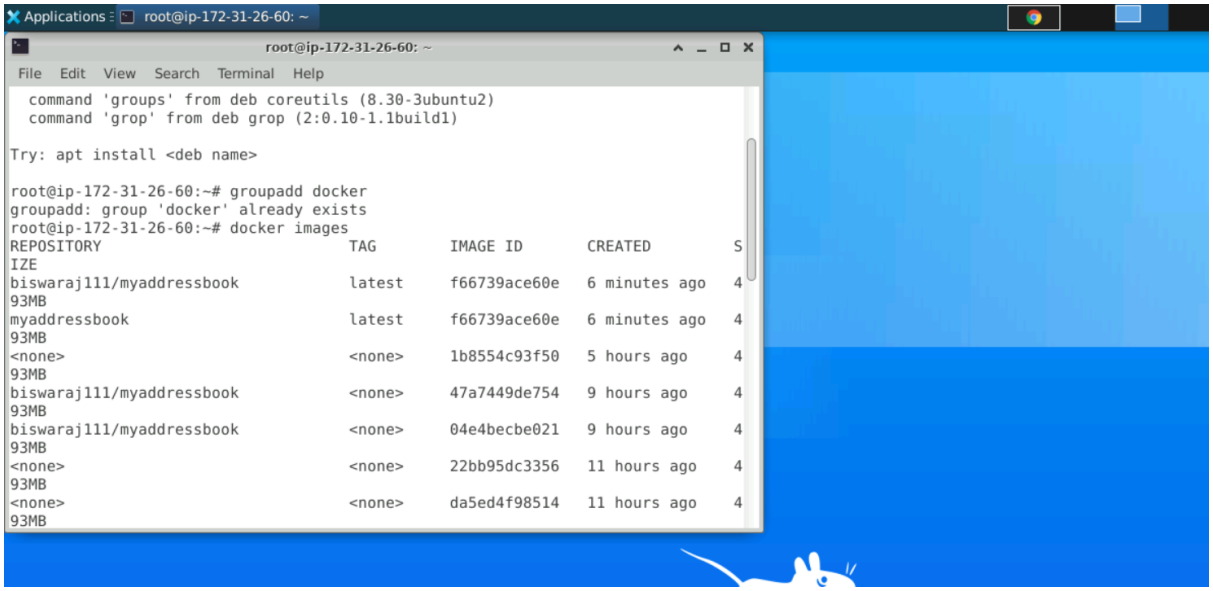


Now we have to sign in our docker hub account



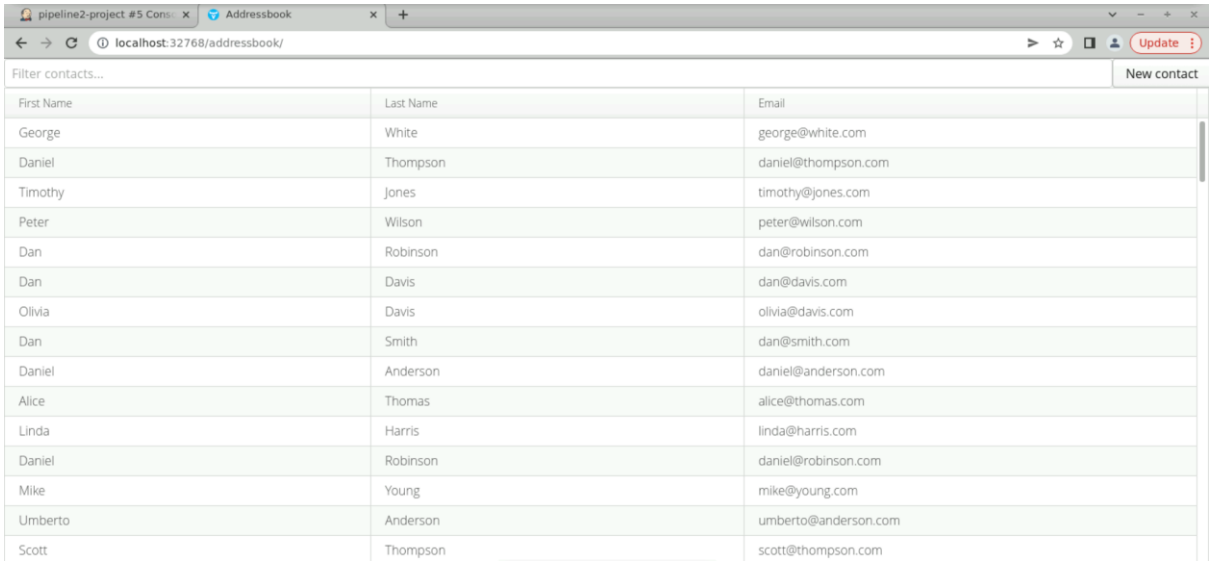


Now we can see the containers and images here



Now we can see the address book

If we want to access the application we have to put localhost:32768/addressbook/



Here comes the end of our project.