Name: Mrunal Vaidya
Roll : 68

XIE ID: 202003060
Batch: C

Experiment 5

AIM: To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and Deploy an application over the tomcat server.

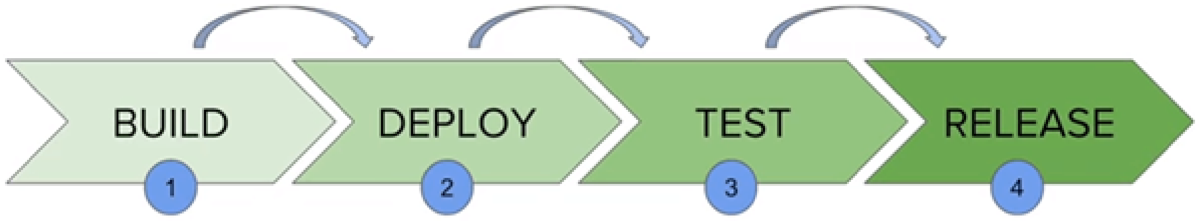
THEORY:

Jenkins Pipeline:-

- In Jenkins, a pipeline is a collection of events or jobs which are interlinked with one another in a sequence.
- It is a combination of plugins that support the integration and implementation of **continuous delivery pipelines** using Jenkins.
- In other words, a Jenkins Pipeline is a collection of jobs or events that brings the software from version control into the hands of the end users by using automation tools. It is used to incorporate continuous delivery in our software development workflow.
- A pipeline has an extensible automation server for creating simple or even complex delivery pipelines "as code", via DSL (Domain-specific language).

What is a Continuous Delivery Pipeline?

- In a Jenkins Pipeline, every job has some sort of dependency on at least one or more jobs or events.



- The above diagram represents a continuous delivery pipeline in Jenkins. It contains a collection of states such as build, deploy, test and release. These jobs or events are interlinked with each other. Every state has its jobs, which work in a sequence called a continuous delivery pipeline.
- A continuous delivery pipeline is an automated expression to show your process for getting software for version control. Thus, every change made in your software goes through a number of complex processes on its manner to being released. It also involves developing the software in a repeatable and reliable manner, and progression of the built software through multiple stages of testing and deployment.

Name: Mrunal Vaidya XIE ID: 202003060
Roll : 68 Batch: C

JenkinsFile:-

Jenkins Pipeline can be defined by a text file called JenkinsFile. You can implement pipeline as code using JenkinsFile, and this can be defined by using a DSL (Domain Specific Language). With the help of JenkinsFile, you can write the steps required for running a Jenkins Pipeline.

The benefits of using JenkinsFile are:

- You can make pipelines automatically for all branches and can execute pull requests with just one JenkinsFile.
- You can review your code on the pipeline.
- You can review your Jenkins pipeline.
- This is the singular source for your pipeline and can be customized by multiple users.

JenkinsFile can be defined by using either Web UI or with a JenkinsFile.

Pipeline syntax:-

Two types of syntax are used for defining your JenkinsFile.

- Declarative
- Scripted

Declarative:

Declarative pipeline syntax offers a simple way to create pipelines. It consists of a predefined hierarchy to create Jenkins pipelines. It provides you the ability to control all aspects of a pipeline execution in a simple, straightforward manner.

Scripted:

Scripted Jenkins pipeline syntax runs on the Jenkins master with the help of a lightweight executor. It uses very few resources to convert the pipeline into atomic commands.

Both scripted and declarative syntax are different from each other and are defined totally differently.

Why Use Jenkins Pipeline?

Jenkins is a continuous integration server which has the ability to support the automation of software development processes. You can create several automation jobs with the help of use cases, and run them as a Jenkins pipeline.

Here are the reasons why you should use Jenkins pipeline:

- Jenkins pipeline is implemented as a code which allows several users to edit and execute the pipeline process.
- Pipelines are robust. So if your server undergoes an unpredicted restart, the pipeline will be automatically resumed.
- You can pause the pipeline process and make it wait to continue until there is an input from the user.
- Jenkins Pipelines support big projects. You can run many jobs, and even use pipelines in a loop.

Jenkins Pipeline Concepts:-

Pipeline: This is the user-defined block, which contains all the processes such as build, test, deploy, etc. it is a group of all the stages in a JenkinsFile. All the stages and steps are defined in this block. It is used in declarative pipeline syntax.

```
pipeline{
}
```

Node: The node is a machine on which Jenkins runs is called a node. A node block is used in scripted pipeline syntax.

```
node{
}
```

Stage: This block contains a series of steps in a pipeline. i.e., build, test, and deploy processes all come together in a stage. Generally, a stage block visualizes the Jenkins pipeline process.

Let's see an example for multiple stages, where each stage performs a specific task:

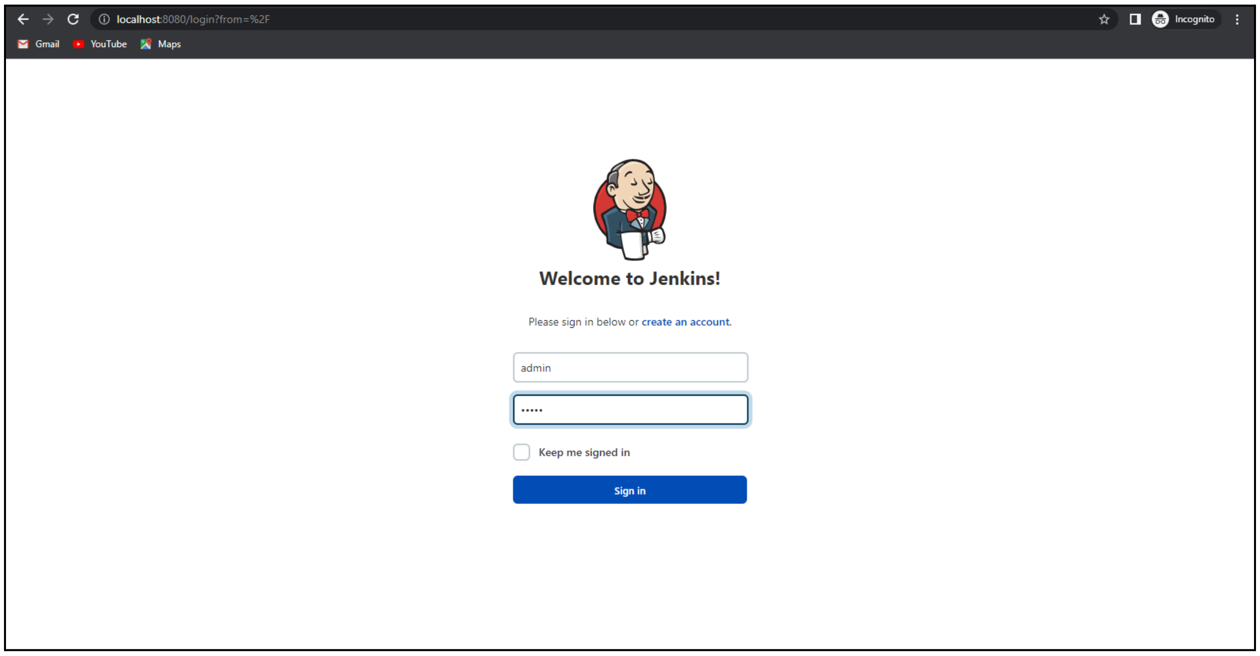
```
pipeline {
   agent any
   stages {
         stage ('Build') {
            ...
         }
         stage ('Test') {
            ...
         }
         stage ('QA') {
            ...
         }
         stage ('Deploy') {
            ...
            }
            stage ('Monitor') {
                ...
            }
    }
}
```

Step: A step is a single task that executes a specific process at a defined time. A pipeline involves a series of steps defined within a stage block.

```
pipeline {
    agent any
    stages {
            stage ('Build') {
                steps {
                        echo 'Running build phase...'
                }
            }
    }
}
```

STEPS TO PERFORM THIS EXPERIMENT:

1. Login to Jenkins through "localhost:8080"



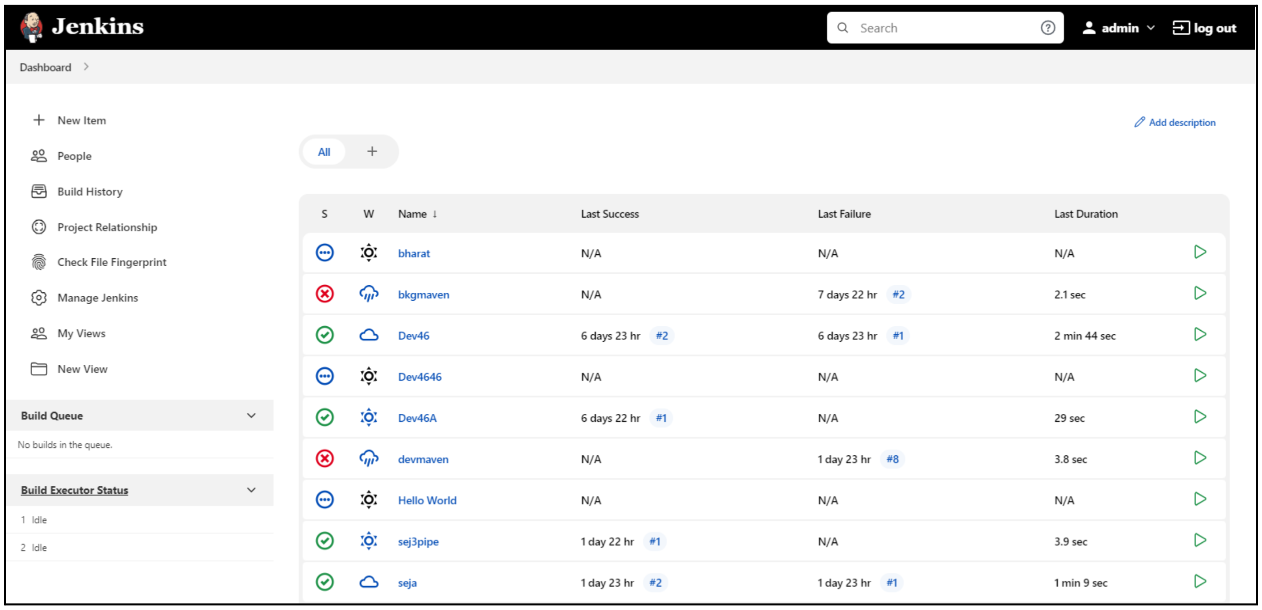
localhost:8080/login?from=%2F

Welcome to Jenkins!

Please sign in below or create an account.

admin

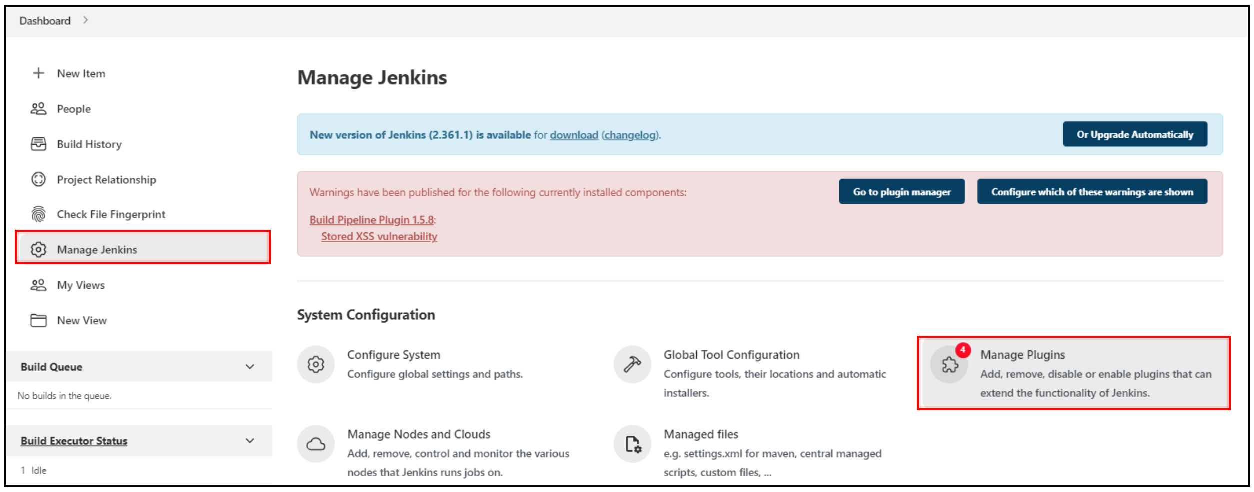
Name: Mrunal Vaidya | XIE ID: 202003060
Roll : 68 | Batch: C


Jenkins | Search | admin | log out
Dashboard
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
New View
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle

Add description
All +

S	W	Name ↓	Last Success	Last Failure	Last Duration
		bharat	N/A	N/A	N/A
		bkgmaven	N/A	7 days 22 hr #2	2.1 sec
		Dev46	6 days 23 hr #2	6 days 23 hr #1	2 min 44 sec
		Dev4646	N/A	N/A	N/A
		Dev46A	6 days 22 hr #1	N/A	29 sec
		devmaven	N/A	1 day 23 hr #8	3.8 sec
		Hello World	N/A	N/A	N/A
		sej3pipe	1 day 22 hr #1	N/A	3.9 sec
		seja	1 day 23 hr #2	1 day 23 hr #1	1 min 9 sec

2. Download the “Build Pipeline Plugin” by following the path “Manage Jenkins > Manage Plugins > Search Build Pipeline”


Dashboard
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
New View
Build Queue
No builds in the queue.
Build Executor Status
1 Idle

Manage Jenkins

New version of Jenkins (2.361.1) is available for download (changelog). | Or Upgrade Automatically

Warnings have been published for the following currently installed components: | Go to plugin manager | Configure which of these warnings are shown
Build Pipeline Plugin 1.5.8:
Stored XSS vulnerability

System Configuration

Configure System
Configure global settings and paths.

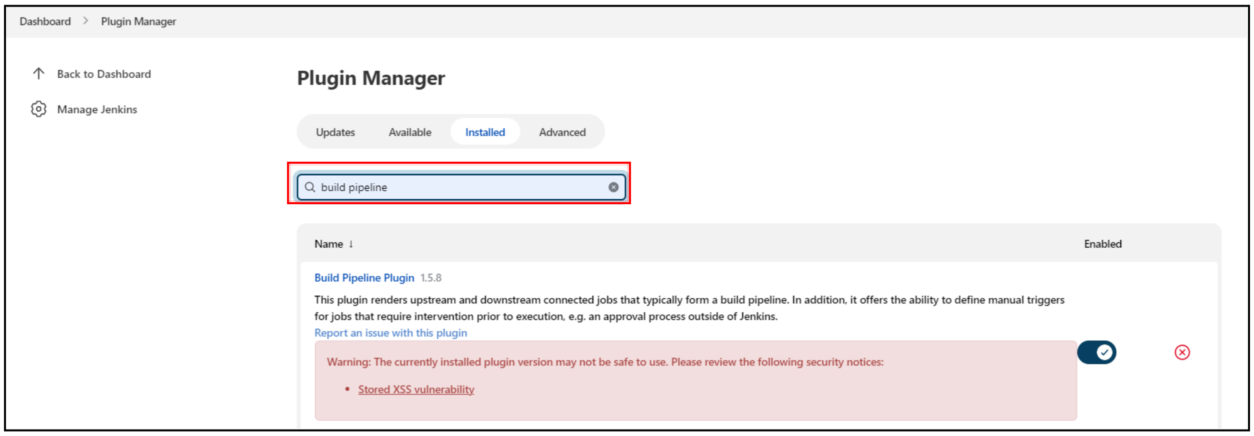
Global Tool Configuration
Configure tools, their locations and automatic installers.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Managed files
e.g. settings.xml for maven, central managed scripts, custom files, ...

Name: Mrunal Vaidya XIE ID: 202003060
Roll : 68 Batch: C


Dashboard > Plugin Manager

Back to Dashboard
Manage Jenkins

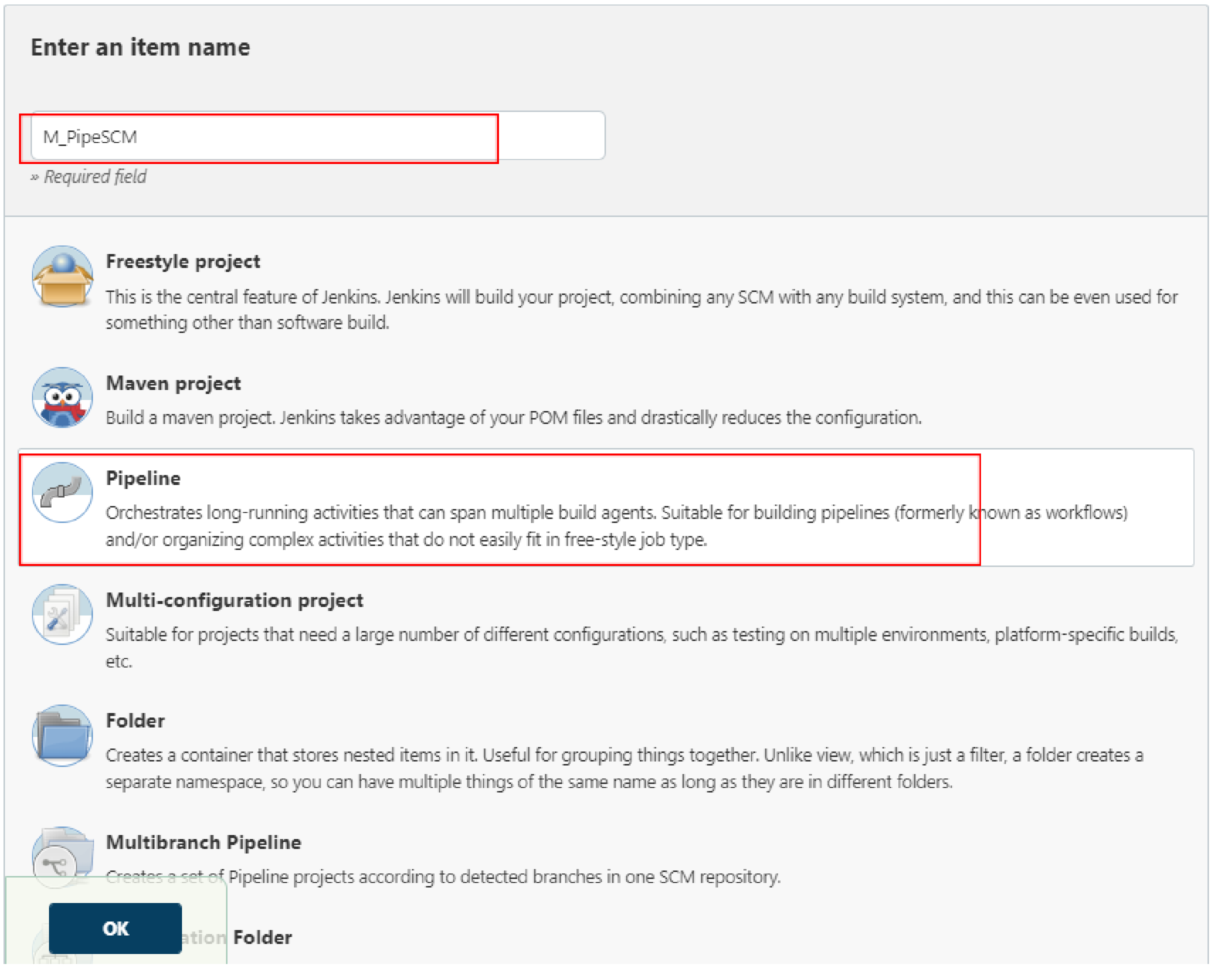
Plugin Manager

Updates | Available | Installed | Advanced

build pipeline

Name ↓	Enabled
Build Pipeline Plugin 1.5.8 This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins. Report an issue with this plugin Warning: The currently installed plugin version may not be safe to use. Please review the following security notices: • Stored XSS vulnerability	✓

3. Now create a New Item and Build a Pipeline Project with a Basic Pipeline Script


Enter an item name

M_PipeSCM

» Required field

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.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

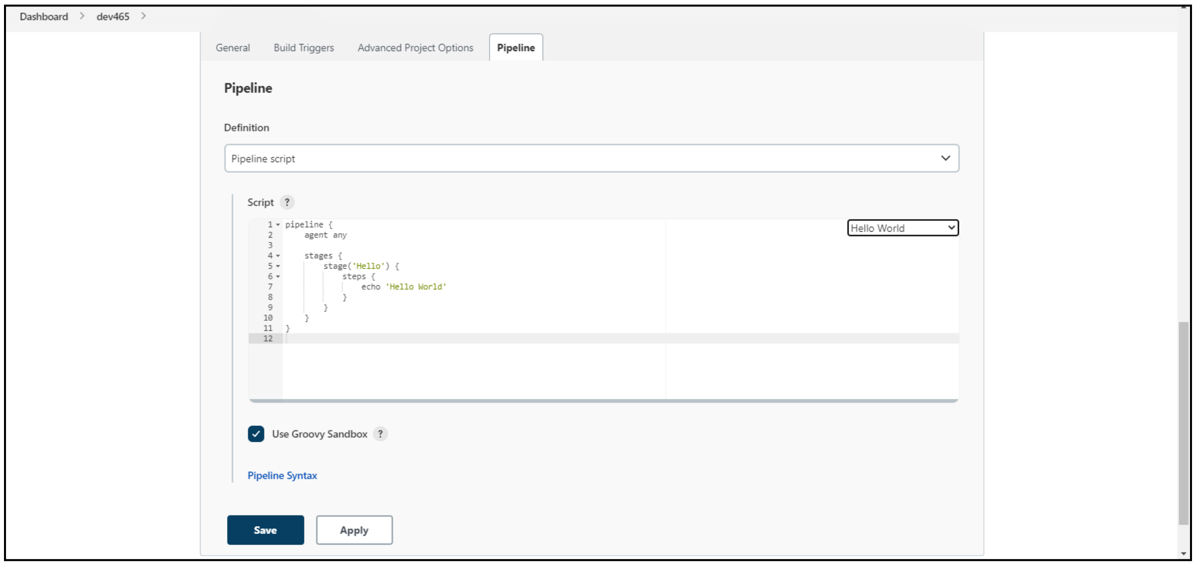
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK ation Folder

Name: Mrunal Vaidya XIE ID: 202003060
Roll : 68 Batch: C

4. In Pipeline Definition, select Pipeline Script and add simple Hello World script



Pipeline

Definition: Pipeline script

Script

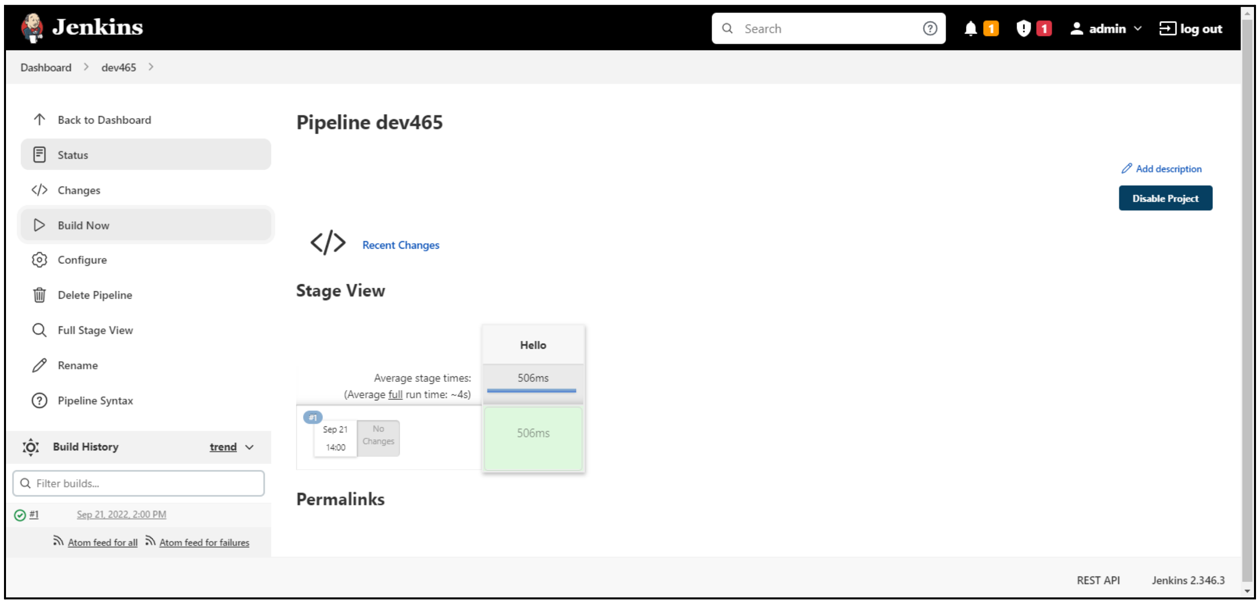
```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
    }
}
```

Hello World

☑ Use Groovy Sandbox

Pipeline Syntax

4. Then in dashboard click on build now



Pipeline dev465

Stage View

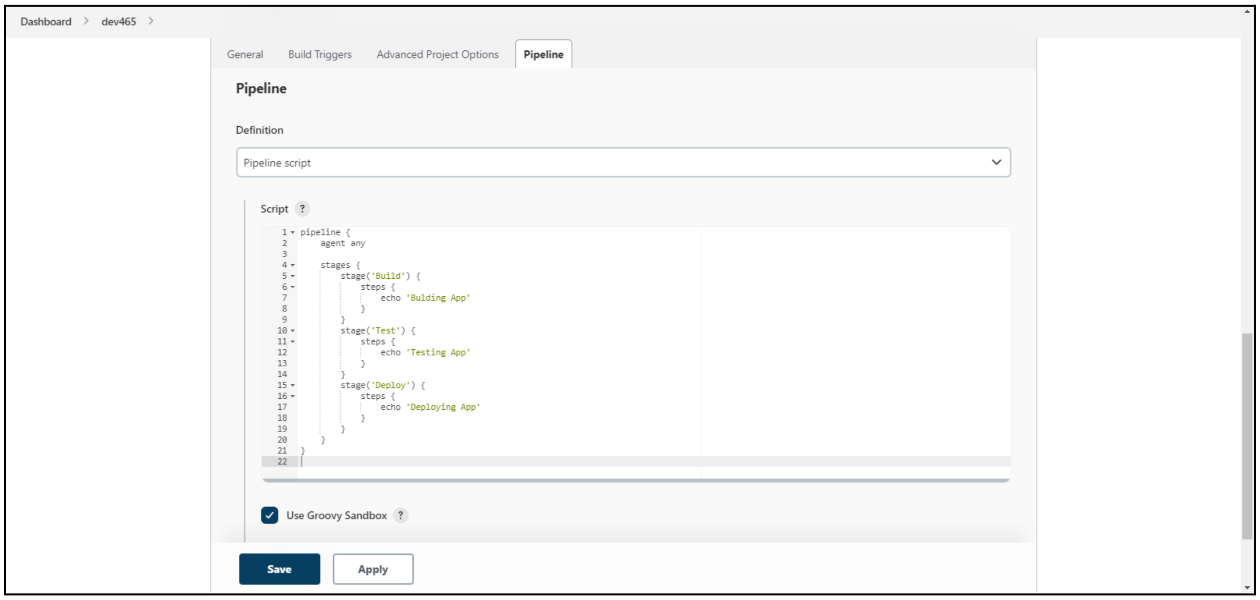
	Hello
Average stage times: (Average full run time: ~4s)	506ms
#1 Sep 21 14:00 No Changes	506ms

Build History: #1 Sep 21, 2022, 2:00 PM

Permalinks

5. Change the Pipeline Script and Build the project again

Name: Mrunal Vaidya XIE ID: 202003060
Roll : 68 Batch: C



Pipeline

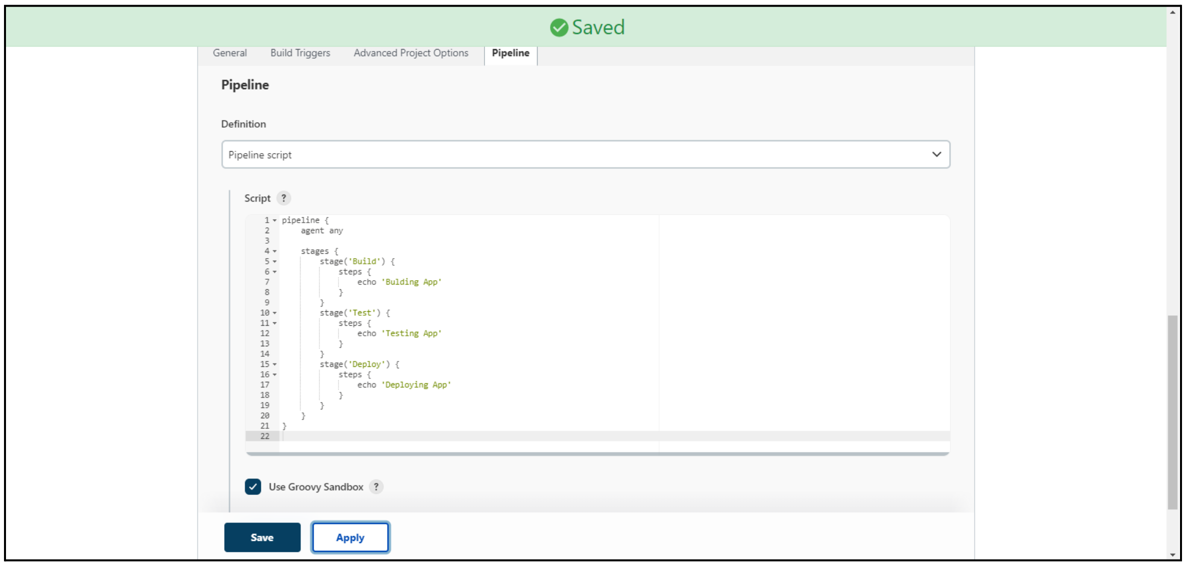
Definition

Pipeline script

Script ?

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo 'Bulding App'
            }
        }
        stage('Test') {
            steps {
                echo 'Testing App'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying App'
            }
        }
    }
}
```

☑ Use Groovy Sandbox ?



Pipeline

Definition

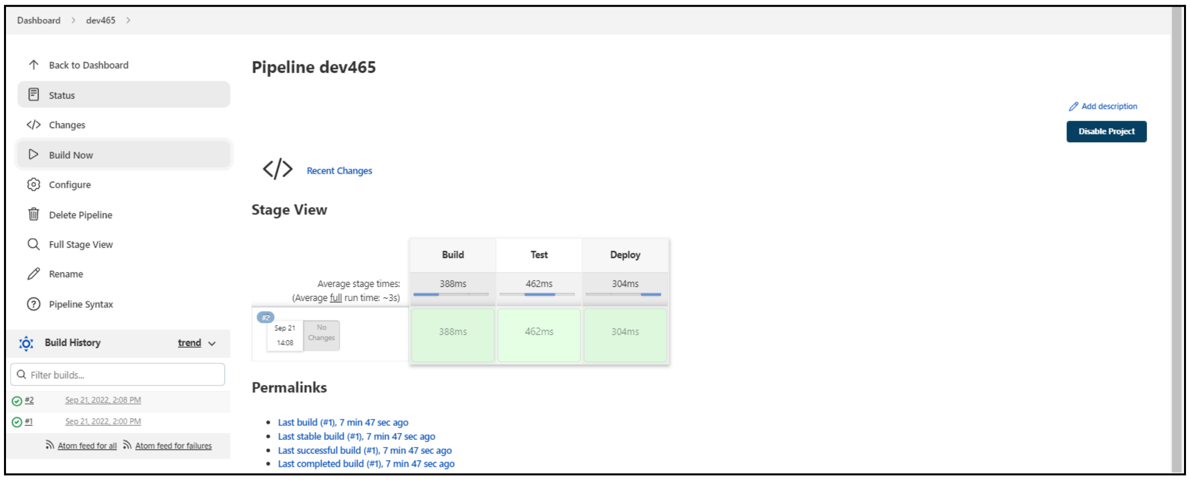
Pipeline script

Script ?

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo 'Bulding App'
            }
        }
        stage('Test') {
            steps {
                echo 'Testing App'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying App'
            }
        }
    }
}
```

☑ Use Groovy Sandbox ?

Name: Mrunal Vaidya XIE ID: 202003060
Roll : 68 Batch: C


Dashboard > dev465 >

Back to Dashboard
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
Rename
Pipeline Syntax

Build History trend

Filter builds...

#2 Sep 21, 2022, 2:08 PM
#1 Sep 21, 2022, 2:00 PM

Atom feed for all Atom feed for failures

Pipeline dev465

Add description
Disable Project

Recent Changes

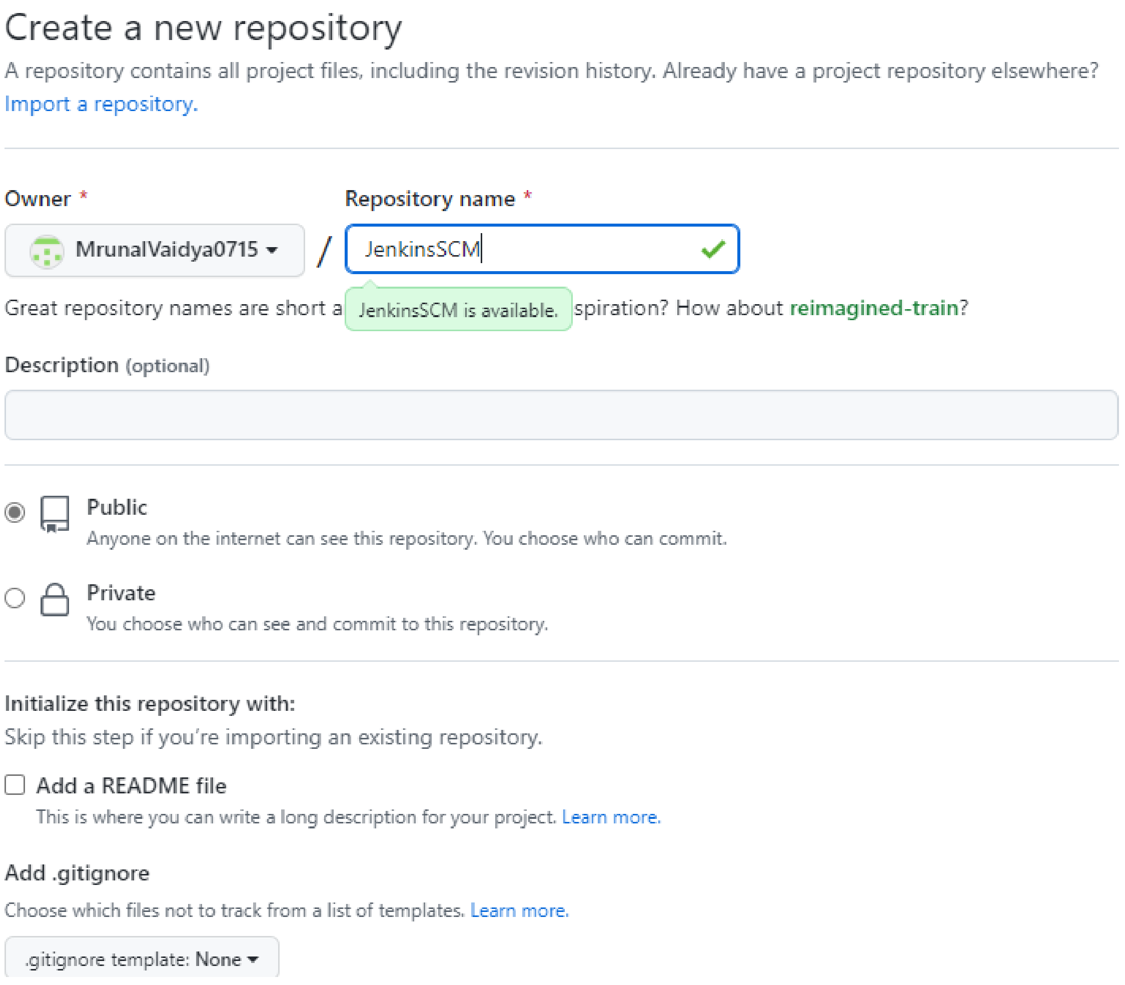
Stage View

	Build	Test	Deploy
Average stage times: (Average full run time: ~3s)	388ms	462ms	304ms
#2 Sep 21 14:08 No Changes	388ms	462ms	304ms

Permalinks

- Last build (#1), 7 min 47 sec ago
- Last stable build (#1), 7 min 47 sec ago
- Last successful build (#1), 7 min 47 sec ago
- Last completed build (#1), 7 min 47 sec ago

6. Login to GitHub & Create a new repository named "JenkinsSCM" and in that create a Jenkinsfile.


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Repository name *

MrunalVaidya0715 / JenkinsSCM

Great repository names are short a JenkinsSCM is available. spiration? How about reimagined-train?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

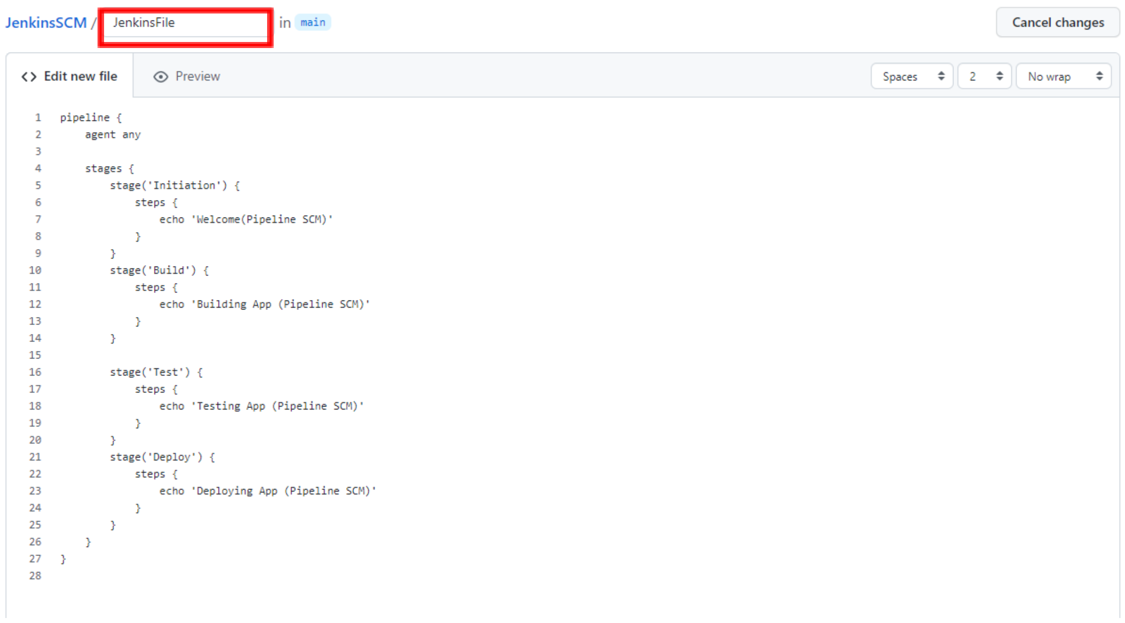
Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. Learn more.

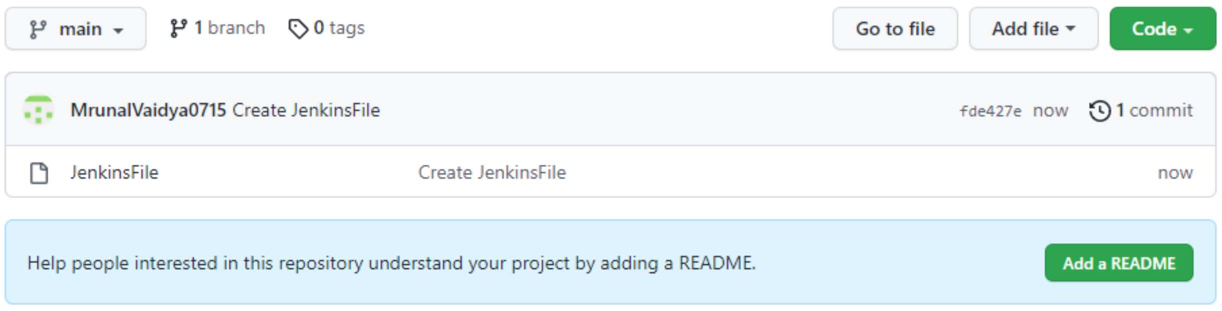
Add .gitignore
Choose which files not to track from a list of templates. Learn more.

.gitignore template: None

Name: Mrunal Vaidya XIE ID: 202003060
Roll : 68 Batch: C

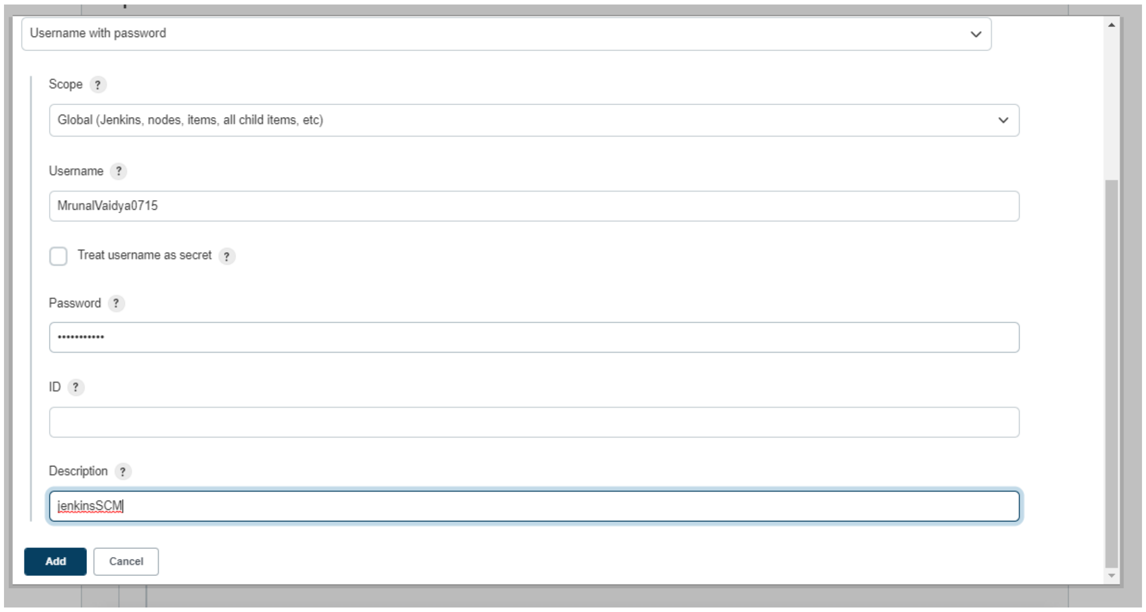


```
pipeline {
    agent any

    stages {
        stage('Initiation') {
            steps {
                echo 'Welcome(Pipeline SCM)'
            }
        }
        stage('Build') {
            steps {
                echo 'Building App (Pipeline SCM)'
            }
        }

        stage('Test') {
            steps {
                echo 'Testing App (Pipeline SCM)'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying App (Pipeline SCM)'
            }
        }
    }
}
```


main
1 branch
0 tags
Go to file
Add file
Code
MrunalVaidya0715 Create JenkinsFile
fde427e now
1 commit
JenkinsFile
Create JenkinsFile
now
Help people interested in this repository understand your project by adding a README.
Add a README

7. Create another Pipeline Project with a pipeline script from SCM

Name: Mrunal Vaidya XIE ID: 202003060
Roll : 68 Batch: C


Username with password

Scope
Global (Jenkins, nodes, items, all child items, etc)

Username
MrunalVaidya0715

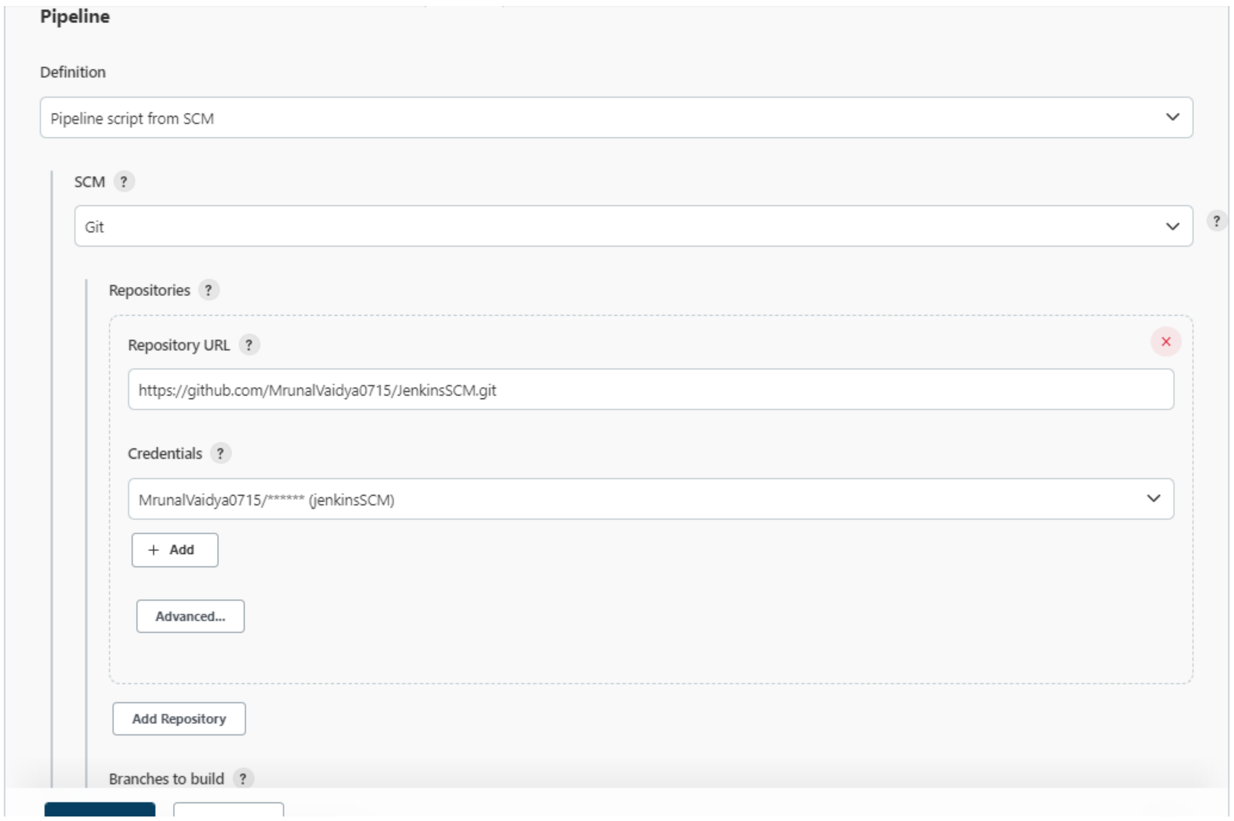
Treat username as secret

Password
•••••••••••

ID

Description
jenkinsSCM

8. Select Git and Paste the link to your repository under Repository URL


Pipeline

Definition
Pipeline script from SCM

SCM
Git

Repositories

Repository URL
https://github.com/MrunalVaidya0715/JenkinsSCM.git

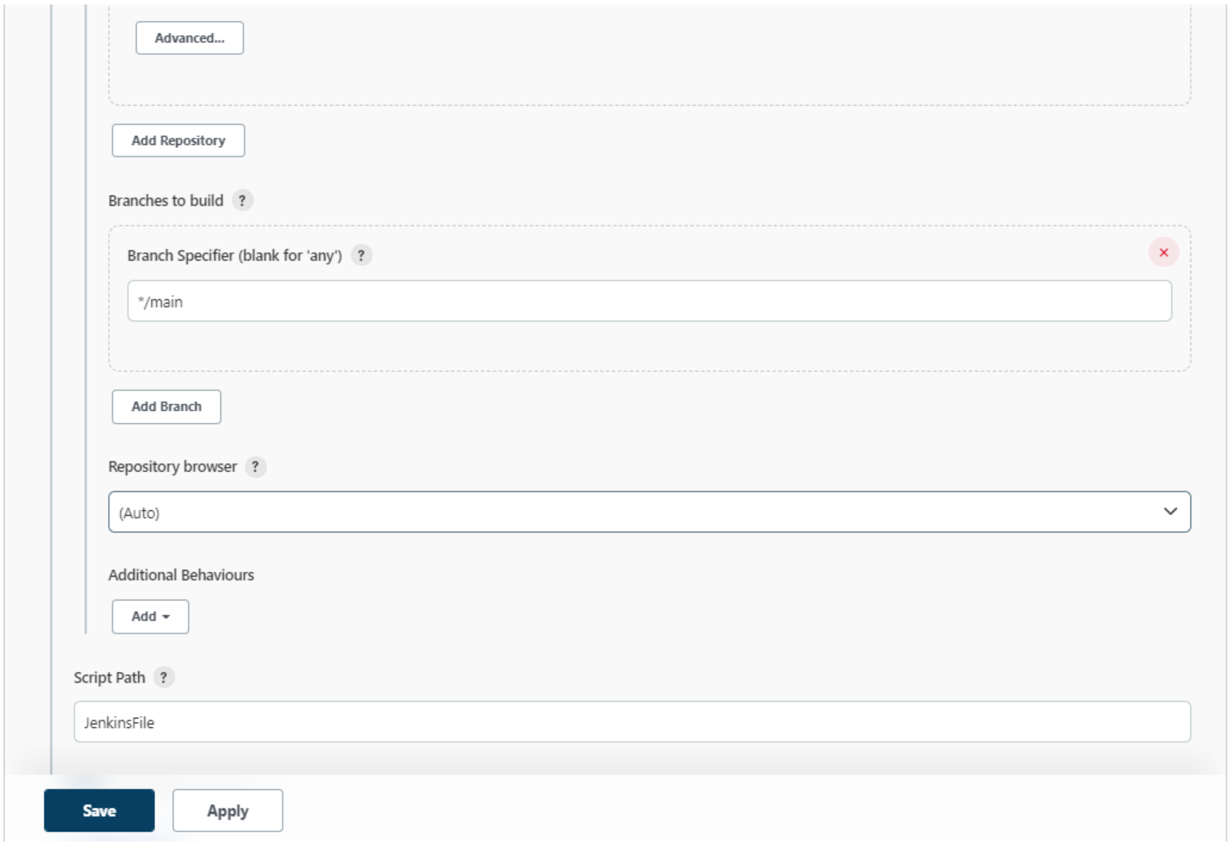
Credentials
MrunalVaidya0715/****** (jenkinsSCM)

Branches to build

Name: Mrunal Vaidya
Roll : 68

XIE ID: 202003060
Batch: C

9. Change the Branch to Build to main and give the file name i.e. Jenkinsfile under Script path



Branches to build

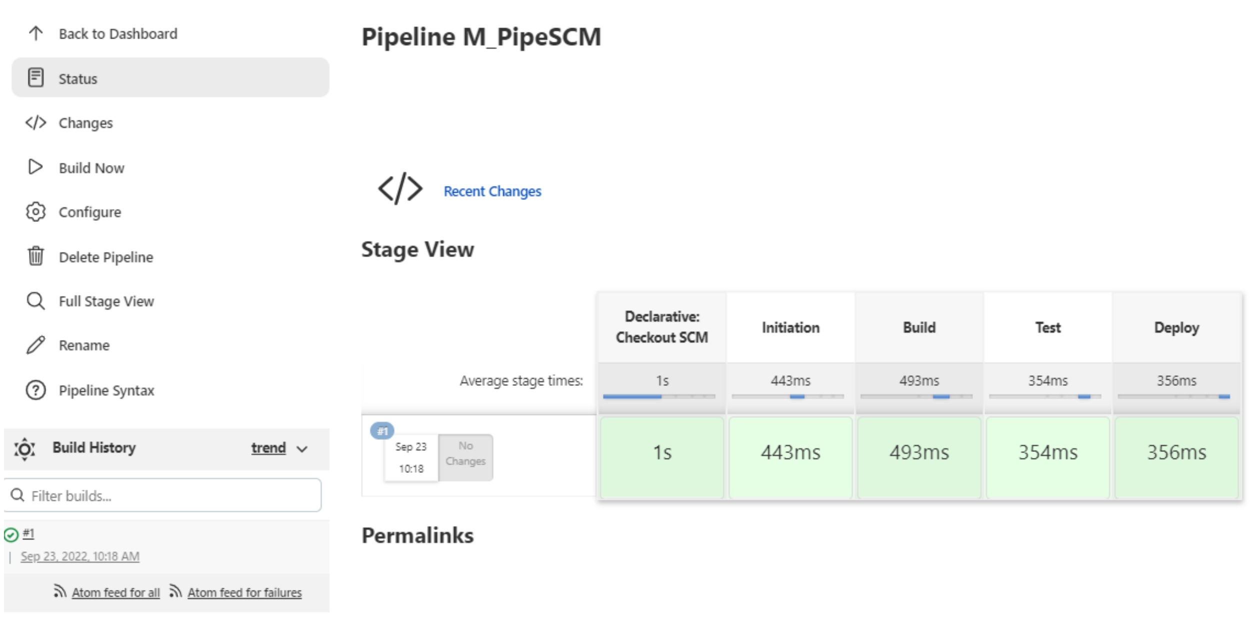
Branch Specifier (blank for 'any'): */main

Repository browser: (Auto)

Script Path: JenkinsFile

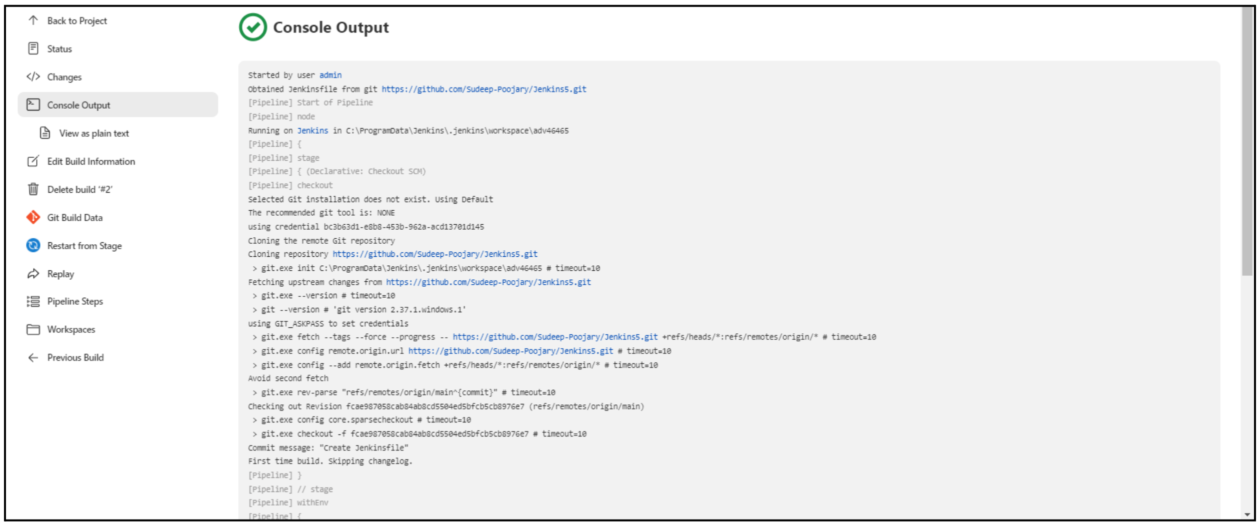
10. Click on Build Now

Name: Mrunal Vaidya XIE ID: 202003060
Roll : 68 Batch: C


Pipeline M_PipeSCM

Stage View

	Declarative: Checkout SCM	Initiation	Build	Test	Deploy
Average stage times:	1s	443ms	493ms	354ms	356ms
#1 Sep 23 10:18 No Changes	1s	443ms	493ms	354ms	356ms



```
Started by user admin
Obtained Jenkinsfile from git https://github.com/Sudeep-Poojary/Jenkins5.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\adv46465
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential bc3b63d1-e8b8-453b-962a-acd13701d145
Cloning the remote Git repository
Cloning repository https://github.com/Sudeep-Poojary/Jenkins5.git
 > git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\adv46465 # timeout=10
Fetching upstream changes from https://github.com/Sudeep-Poojary/Jenkins5.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.37.1.windows.1'
using GIT_ASKPASS to set credentials
 > git.exe fetch --tags --force --progress -- https://github.com/Sudeep-Poojary/Jenkins5.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/Sudeep-Poojary/Jenkins5.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision fcae987058cab84ab8cd5504ed5bfcb5cb8976e7 (refs/remotes/origin/main)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f fcae987058cab84ab8cd5504ed5bfcb5cb8976e7 # timeout=10
Commit message: "Create Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
```

Conclusion:

From the above experiment, it is concluded that we have successfully built a pipeline of jobs using Maven / Gradle / Ant in Jenkins. We also created a pipeline script to Test and Deploy an application over the tomcat server. And hence, with this experiment we have achieved the Lab Outcome Three (LO3).

POs Achieved: PO1, PO5, PO12