

St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology

A.Y. 2023-2024
Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

Experiment – 6: a. To build pipeline of jobs in Jenkins, create a pipeline script to test and deploy an application.
b. To automatically build a job in Jenkins using webhooks
(Topic Beyond Syllabus)

1. **Aim:** To build pipeline of jobs in Jenkins, create a pipeline script to test and deploy an application
2. **Objectives:** Aim of this experiment is that, the students will be able
 - To build pipeline of jobs in Jenkins, create a pipeline script to test and deploy an application
3. **Outcomes:** After study of this experiment, the students will be able
 - To understand the importance of Jenkins to Build and deploy Software Applications on server environment.
4. **Prerequisite:** Knowledge of software engineering concept of integration and deployment
5. **Requirements:** Jenkins, JDK, python, Personal Computer, Windows operating system, browser, Internet Connection, Microsoft Word.
6. **Pre-Experiment Exercise:**
Brief Theory: Refer shared material
7. **Laboratory Exercise**
 - A. **Procedure:**
 - a. **Answer the following:**
 - **What is Jenkins pipeline?**

Jenkins pipelines, typically written in Groovy, enable the automation of complex software delivery processes. They allow for the structured definition of stages, steps, and conditions, facilitating tasks like building, testing, and deployment. Jenkins offers two pipeline types: Declarative for simpler workflows and Scripted for more advanced customization with Groovy scripting. This approach promotes version-controlled, collaborative, and repeatable CI/CD processes, ultimately leading to more efficient and reliable software development and delivery.

- **What are the different ways to write a Jenkins pipeline?**

There are primarily two ways to write Jenkins pipelines:

Declarative Pipeline: This approach uses a simplified and structured DSL (Domain-Specific Language) for defining pipelines. It's suitable for straightforward workflows and follows a more human-readable syntax, making it easier for beginners to understand and use.

Scripted Pipeline: Scripted pipelines utilize Groovy scripting to define pipelines. They offer greater flexibility and control over complex workflows, allowing developers to write custom logic and scripts for intricate CI/CD processes. Scripted

pipelines are ideal for advanced users who require more fine-grained control.

Both approaches enable the definition of continuous integration and continuous delivery (CI/CD) workflows as code, making it possible to version-control and automate software development pipelines.

b. Execute following (Refer the shared material) and attach screenshots:

- Create and build pipeline project with Git
- Create and build pipeline project with pipeline script

8. Post-Experiments Exercise

A. Extended Theory:

Nil

B. Questions:

- Explain the types of agents in a Jenkinsfile?
- What are webhooks?

C. Conclusion:

- Write what was performed in the experiment.
- Write the significance of the topic studied in the experiment.

D. References:

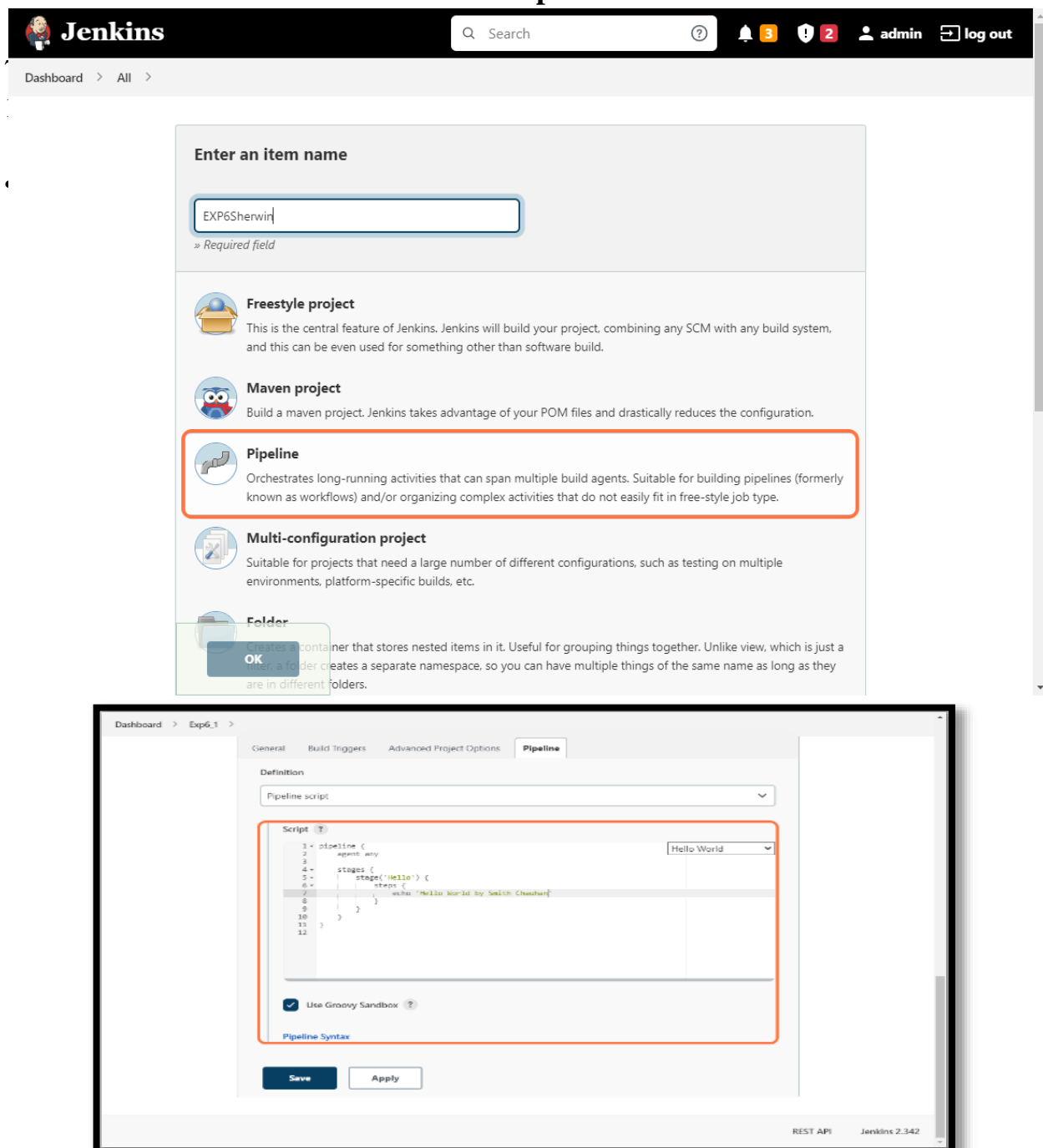
<https://jenkins.io/doc/>
<https://www.jenkins.io/doc/book/pipeline/syntax/>
<https://www.edureka.co/blog/jenkins-pipeline-tutorial-continuous-delivery>
<https://www.slideshare.net/abediaz/introduction-to-jenkins>
<https://www.slideshare.net/jph98/jenkins-ci-presentation>


St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology
A.Y. 2023-2024
Class: TE-ITA/B, Semester: V
Subject: DevOps Lab



Experiment – 6: a. To build pipeline of jobs in Jenkins, create a pipeline script to test and deploy an application.

b. To automatically build a job in Jenkins using webhooks
(Topic Beyond Syllabus)

List of screenshots to be attached for Experiment 6




**Jenkins**


Q Search ?  3  2 admin log out


Dashboard > EXP6Sherwin >


↑ Back to Dashboard


Q Status


 Changes


 Build Now


 Configure

 Delete Pipeline

 Full Stage View



 Rename

 Pipeline Syntax

 **Build History** trend ▾

Q Filter builds...

✓ #1 Sep 6, 2023 10:45 AM


 Atom feed for all  Atom feed for failures

Pipeline EXP6Sherwin

sherwin 63

[Edit description](#)

Disable Project

 [Recent Changes](#)

Stage View

Average stage times:
(Average full run time: ~403ms)

	Hello
	48ms
#1 Sep 06 10:45 No Changes	48ms



Permalinks

REST API Jenkins 2.342

Stage Logs (Hello)

 Print Message -- Hello Sherwin

Hello Sherwin

Q Search ?  2  1 admin log ou

✓ **Console Output**

```
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\exp6_1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Project with your own pipeline script

The screenshot shows the Jenkins project page for 'EXP6Sherwin'. The left sidebar contains navigation links: Build Now, Configure, Delete Pipeline, Full Stage View, Rename, and Pipeline Syntax. Below these is the 'Build History' section with a search bar and a list of builds. The main content area is titled 'Stage View' and shows a pipeline graph with two stages. The first stage, '#1', is highlighted in green and shows a duration of 48ms. The second stage, '#2', is highlighted in blue and shows a duration of 48ms. The pipeline graph also displays 'Average stage times' and 'Average full run time: ~403ms'. A 'Disable Project' button is located in the top right corner. At the bottom right, there are links for 'REST API' and 'Jenkins 2.342'.

Dashboard > EXP6Sherwin >

Build Now
Configure
Delete Pipeline
Full Stage View
Rename
Pipeline Syntax

Build History trend

Filter builds...

#2 Sep 06, 2023 10:49 AM No Changes
#1 Sep 06, 2023 10:45 AM No Changes

Atom feed for all Atom feed for failures

Stage View

Recent Changes

Disable Project

Average stage times:
(Average full run time: ~403ms)

48ms

48ms

Permalinks

- Last build (#2), 13 ms ago
- Last stable build (#1), 4 min 26 sec ago
- Last successful build (#1), 4 min 26 sec ago
- Last completed build (#1), 4 min 26 sec ago

REST API Jenkins 2.342

The screenshot shows the Jenkins 'Enter an item name' page. The top header includes the Jenkins logo, a search bar, and user information (admin, log out). The main content area is titled 'Enter an item name' and features a text input field with the value 'Sherwinexpd'. Below the input field is a list of project types: Freestyle project, Maven project, Pipeline, Multi-configuration project, and Folder. The 'Pipeline' option is highlighted with an orange border. A tooltip is visible over the 'Folder' option, indicating that it creates a container that stores nested items in it.

Jenkins

Search

admin log out

Dashboard > All >

Enter an item name

Sherwinexpd

» 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.

PIPELINE SCRIPT:

```
pipeline {
  agent any
  stages {
    stage('Static Analysis') {
      steps {
        echo 'Run the static analysis on the code'
      }
    }
    stage('Compile') {
      steps {
        echo 'Compile the source code'
      }
    }
    stage('Security Check') {
      steps {
        echo 'Run security checks against the application'
      }
    }
    stage('Run Unit Tests') {
      steps {
        echo 'Run unit tests on the source code'
      }
    }
    stage('Run Integration Tests') {
      steps {
        echo 'Run crucial integration tests on the source code'
      }
    }
    stage('Publish Artifacts') {
      steps {
        echo 'Save the assemblies generated from the compilation'
      }
    }
  }
}
```



OUTPUTS:

The screenshot shows the Jenkins Pipeline Exp6_2 dashboard. On the left is a sidebar with navigation links: Back to Dashboard, Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, Pipeline Syntax, Build History (selected), and trend. The main area displays the Pipeline Exp6_2 details. It includes a 'Recent Changes' section, a 'Stage View' table, and a 'Permalinks' section. The 'Stage View' table shows average stage times and a table of build times for build #1. The 'Permalinks' section shows the last build (#1) completed 1 min 7 sec ago. The bottom right corner shows 'REST API' and 'Jenkins 2.342'.

	Plan phase	code phase	integrate phase	testing phase	unit test
Average stage times: (Average full run time: ~1min 37s)	54ms	40ms	84ms	50ms	51ms
#1 Sep 06 10:54 No Changes	54ms	40ms (paused for 1min 37s)	84ms	50ms	51ms

Stage Logs (Security Check)

Print Message -- Run the security check against the application

Run the security check against the application

Stage Logs (Run Unit Tests)

Print Message -- Run unit tests from the source code (self time 7ms)

Run unit tests from the source code

Stage Logs (Run Integration Tests)

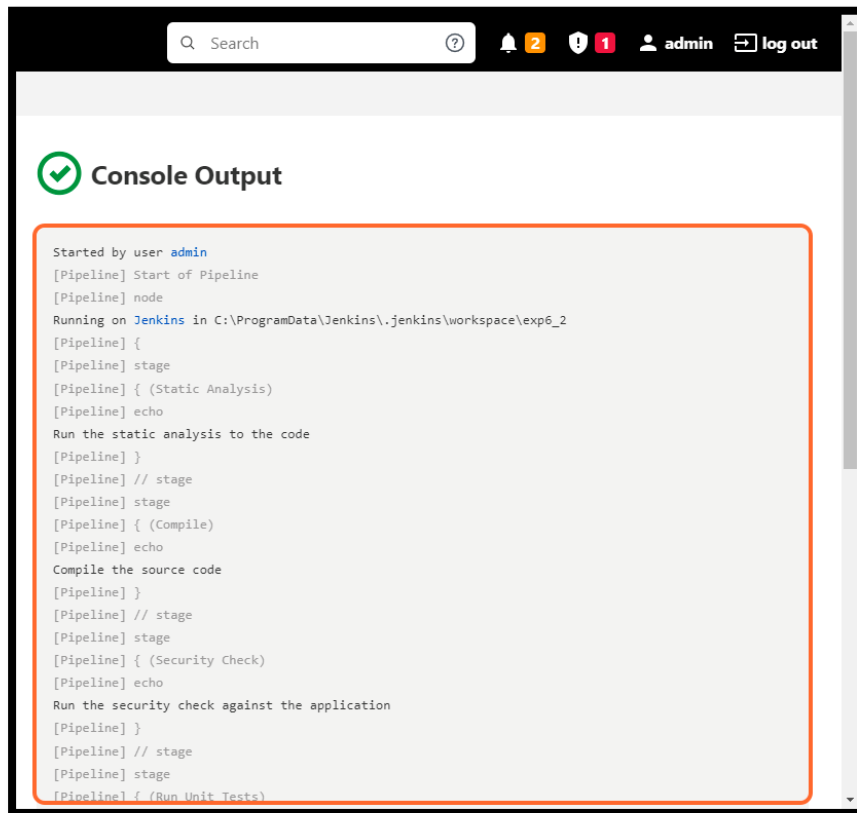
Print Message -- Run only crucial integration tests from the source code (self time 6ms)

Run only crucial integration tests from the source code

Stage Logs (Publish Artifacts)

Print Message -- Save the assemblies generated from the compilation

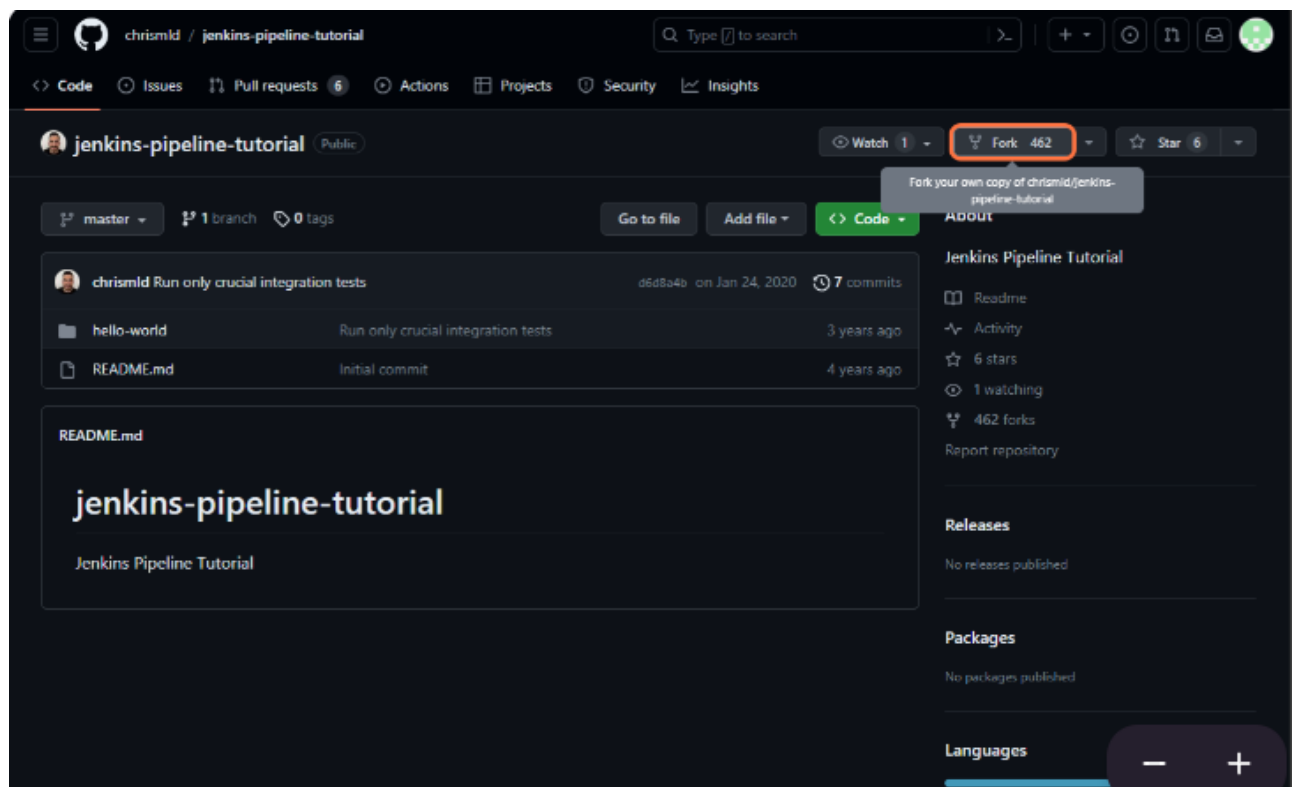
Save the assemblies generated from the compilation



A screenshot of the Jenkins web interface showing the 'Console Output' of a pipeline. The output is displayed in a light gray box with a green checkmark icon and the title 'Console Output'. The text shows the pipeline starting by user 'admin', running on Jenkins in a specific workspace, and executing several stages: 'Static Analysis', 'Compile', 'Security Check', and 'Run Unit Tests'. The output is formatted with color-coded text for different stages and actions.

```
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\exp6_2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Static Analysis)
[Pipeline] echo
Run the static analysis to the code
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Compile)
[Pipeline] echo
Compile the source code
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Security Check)
[Pipeline] echo
Run the security check against the application
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run Unit Tests)
```

2. Create and build pipeline project with Git



Fork repository on GitHub

Fork chrismld/jenkins-pipeline

The screenshot shows the GitHub interface for creating a new fork of the repository 'chrismld / jenkins-pipeline-tutorial'. The page has a dark theme. At the top, there's a navigation bar with links to Code, Issues, Pull requests (6), Actions, Projects, Security, and Insights. Below this, the main heading is 'Create a new fork'. A subtext explains that a fork is a copy of a repository and allows for experimentation without affecting the original project. It also mentions that required fields are marked with an asterisk (*). The form includes two main sections: 'Owner' and 'Repository name'. The 'Owner' dropdown is set to 'Srienzo'. The 'Repository name' dropdown is set to 'jenkins-pipeline-tutorial', with a green checkmark indicating it is available. Below this, there's a note about default naming and a 'Description (optional)' field containing 'Jenkins Pipeline Tutorial'. A checkbox labeled 'Copy the master branch only' is checked, with a link to 'Learn more'. At the bottom, there's a note 'You are creating a fork in your personal account.' and a green 'Create fork' button.

- Create pipeline project with pipeline script from SCM

The screenshot shows the Jenkins web interface. At the top, there's a navigation bar with the Jenkins logo, a search bar, and links for notifications (3), warnings (2), and a user profile (admin) with a 'log out' button. Below this, the breadcrumb trail shows 'Dashboard > Exp6_1 >'. The main content area is titled 'General' and contains a 'Description' field with the text 'SHERWIN RIENZO'. Below the description field, there are several checkboxes for project settings: 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'GitHub project', 'Pipeline speed/durability override', 'Preserve stashes from completed builds', 'This project is parameterized', and 'Throttle builds'. At the bottom, there are 'Save' and 'Apply' buttons.

Build Triggers

☐ Build after other projects are built ?

☐ Build periodically ?

☒ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

☐ Disable this project ?

☐ Quiet period ?

☐ Trigger builds remotely (e.g., from scripts) ?

Save **Apply**

Dashboard > exp6_3 >

General Build Triggers **Advanced Project Options** Pipeline

Pipeline

Definition

Pipeline script from SCM

SCM ?

None

Script Path ?

Jenkinsfile

☒ Lightweight checkout ?

[Pipeline Syntax](#)

Save **Apply**

REST API Jenkins 2.342

General Build Triggers Advanced Project Options **Pipeline**

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

Script Path ?

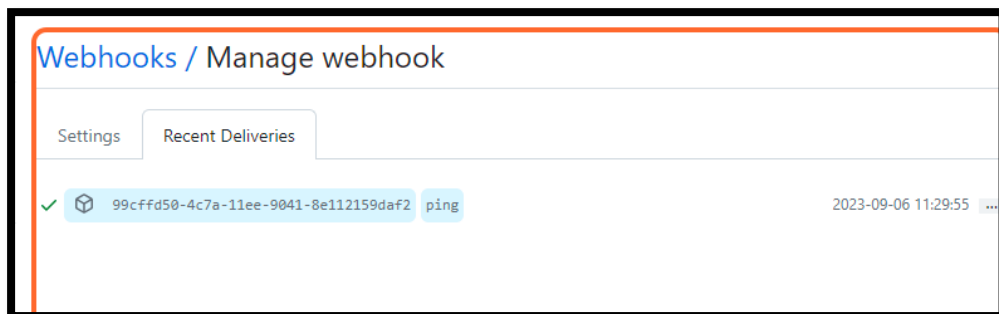
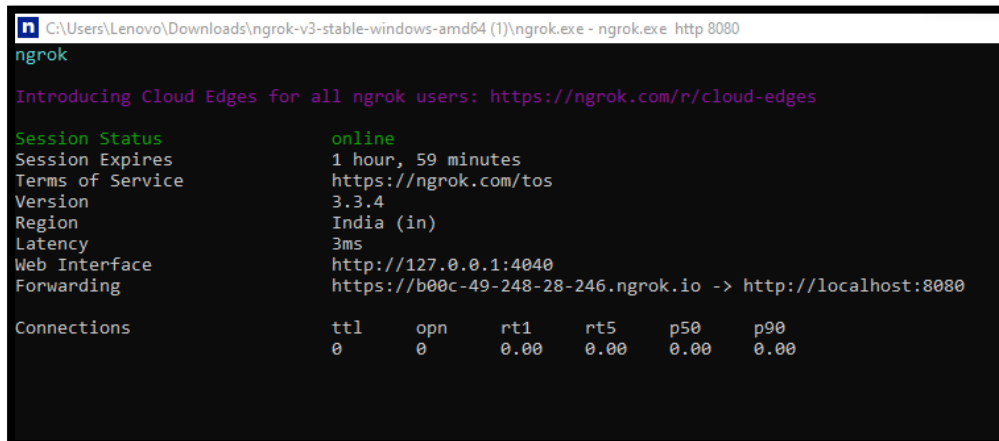
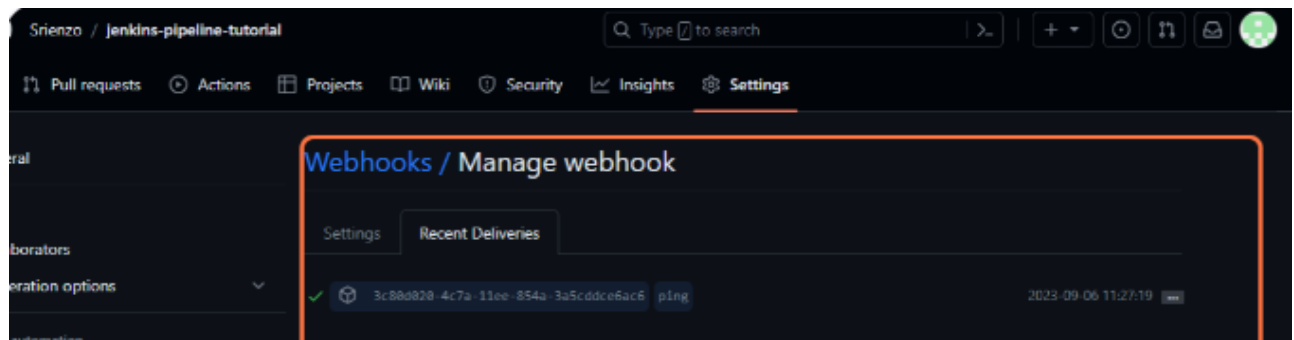
hello-world/Jenkinsfile

☒ Lightweight checkout ?

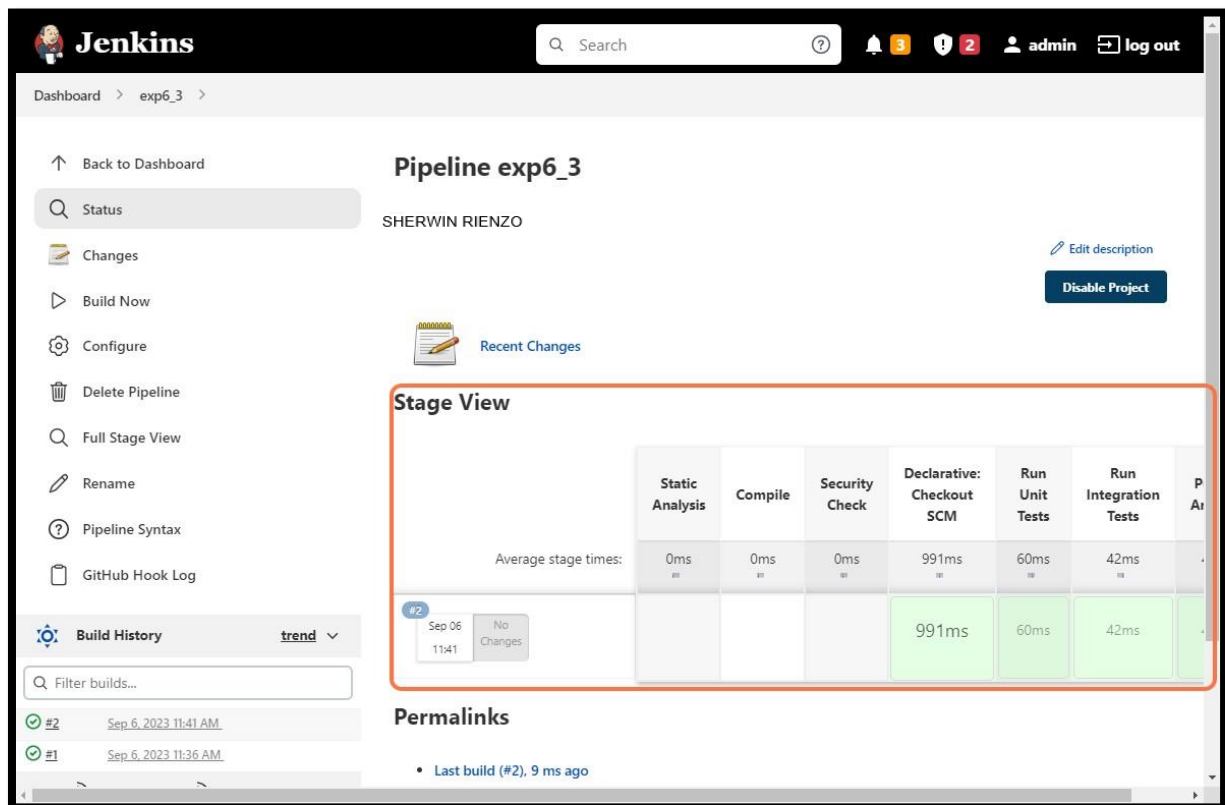
[Pipeline Syntax](#)

Save **Apply**

- Add webhooks to the forked repository



- Build pipeline project



Jenkins Dashboard > exp6_3 >

Search [?] 3 2 admin log out

Dashboard > exp6_3 >

Back to Dashboard

Search Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

GitHub Hook Log

Build History trend

Filter builds...

#2 Sep 6, 2023 11:41 AM

#1 Sep 6, 2023 11:36 AM

Pipeline exp6_3

SHERWIN RIENZO

Edit description

Disable Project

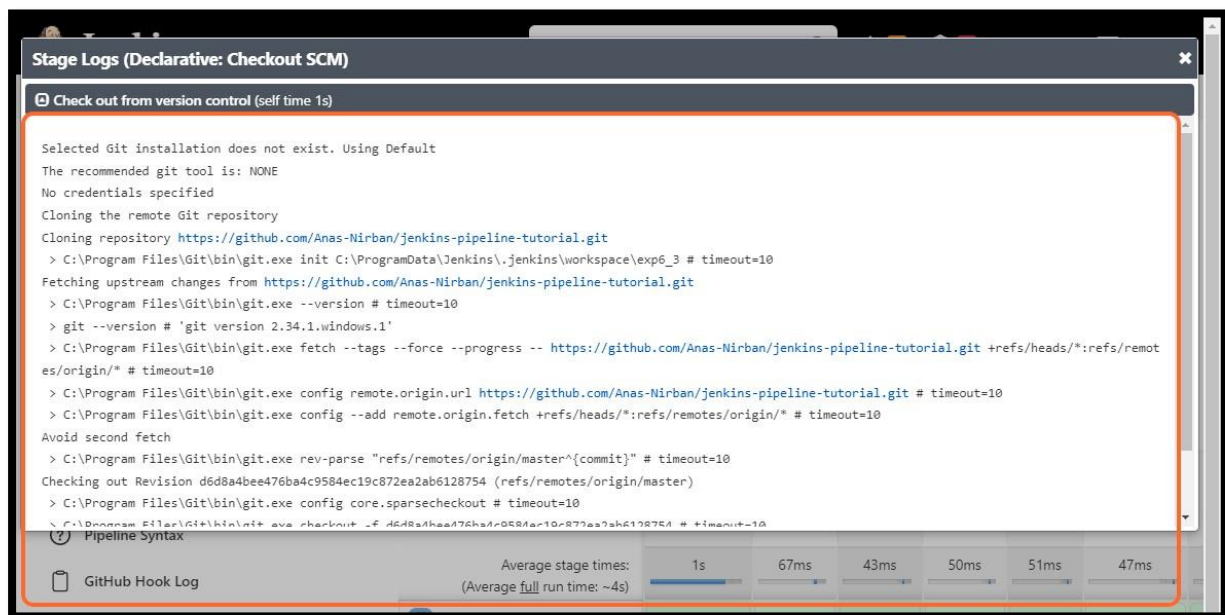
Recent Changes

Stage View

	Static Analysis	Compile	Security Check	Declarative: Checkout SCM	Run Unit Tests	Run Integration Tests	P
Average stage times:	0ms	0ms	0ms	991ms	60ms	42ms	
#2 Sep 06 11:41				991ms	60ms	42ms	

Permalinks

- Last build (#2), 9 ms ago



Stage Logs (Declarative: Checkout SCM)

Check out from version control (self time 1s)

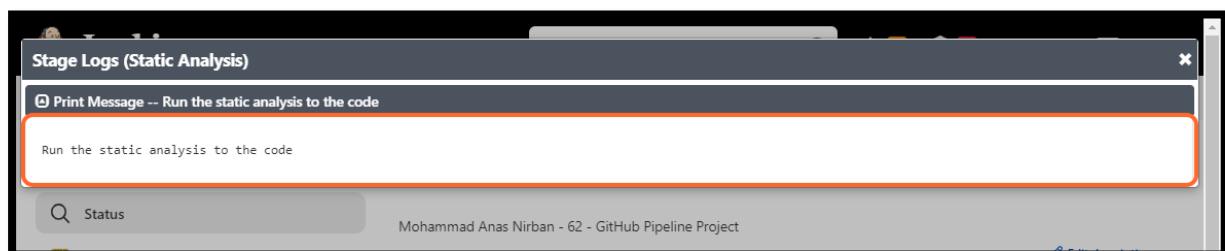
```

Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Anas-Nirban/jenkins-pipeline-tutorial.git
> C:\Program Files\Git\bin\git.exe init C:\ProgramData\Jenkins\workspace\exp6_3 # timeout=10
Fetching upstream changes from https://github.com/Anas-Nirban/jenkins-pipeline-tutorial.git
> C:\Program Files\Git\bin\git.exe --version # timeout=10
> git --version # 'git version 2.34.1.windows.1'
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/Anas-Nirban/jenkins-pipeline-tutorial.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/Anas-Nirban/jenkins-pipeline-tutorial.git # timeout=10
> C:\Program Files\Git\bin\git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision d6d8a4bee476ba4c9584ec19c872ea2ab6128754 (refs/remotes/origin/master)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f d6d8a4bee476ba4c9584ec19c872ea2ab6128754 # timeout=10
  
```

Pipeline Syntax

GitHub Hook Log

Average stage times: 1s 67ms 43ms 50ms 51ms 47ms
(Average full run time: ~4s)



Stage Logs (Static Analysis)

Print Message -- Run the static analysis to the code

Run the static analysis to the code

Search Status

Mohammad Anas Nirban - 62 - GitHub Pipeline Project

Stage Logs (Compile)

Print Message -- Compile the source code

Compile the source code

Stage Logs (Run Unit Tests)

Print Message -- Run unit tests from the source code (self time 7ms)

Run unit tests from the source code

Stage Logs (Publish Artifacts)

Print Message -- Save the assemblies generated from the compilation

Save the assemblies generated from the compilation

Stage Logs (Run Integration Tests)

Print Message -- Run only crucial integration tests from the source code (self time 16ms)

Run only crucial integration tests from the source code

Jenkins

Dashboard > exp6_3 > #4

Build #4 (Sep 6, 2023 11:51:09 AM)

Keep this build forever

Started 10 sec ago
Took 3.3 sec

Changes

- 1. Create sdasd.txt (details / githubweb)
- 2. Create bhv.txt (details / githubweb)

Started by user [admin](#)

Revision: 08e99549f2769d7ac52538158f5a14ca2495474c
Repository: <https://github.com/SmithChauhan2003/jenkins-pipeline-tutorial.git>

- refs/remotes/origin/master

REST API Jenkins 2.342

```

Started by user admin
Obtained hello-world/Jenkinsfile from git https://github.com/Anas-Nirban/jenkins-pipeline-tutorial.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\exp6_3
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Anas-Nirban/jenkins-pipeline-tutorial.git
> C:\Program Files\Git\bin\git.exe init C:\ProgramData\Jenkins\jenkins\workspace\exp6_3 #
timeout=10
Fetching upstream changes from https://github.com/Anas-Nirban/jenkins-pipeline-tutorial.git
> C:\Program Files\Git\bin\git.exe --version # timeout=10
> git --version # 'git version 2.34.1.windows.1'
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/Anas-
Nirban/jenkins-pipeline-tutorial.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/Anas-Nirban/jenkins-
pipeline-tutorial.git # timeout=10
> C:\Program Files\Git\bin\git.exe config --add remote.origin.fetch

```

Pipeline exp6_3

Recent Changes

Stage View

	Static Analysis	Compile	Security Check	Declarative: Checkout SCM	Run Unit Tests	Run Integration Tests	P
Average stage times:	0ms	0ms	0ms	991ms	60ms	42ms	
#2 Sep 06 11:41 No Changes				991ms	60ms	42ms	

Permalinks

- Last build (#2), 9 ms ago

- Add file to forked repository and observe the automated build

The screenshot shows the GitHub interface for a repository named 'jenkins-pipeline-tutorial'. The 'Commit' page displays a commit titled 'Create sherwin.txt' by user 'Srienzo', committed 6 minutes ago. The commit message is 'Create sherwin.txt'. The diff shows a single file 'sherwin.txt' with one addition: 'khud ka likho'. The commit hash is '7922468'. Below the diff, there is a section for comments, currently showing 0 comments. The interface includes navigation links for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.


- Make changes to Jenkins file on forked repository and observe the automated build

The screenshot shows the Jenkins Pipeline Stage View for a pipeline named 'exp6_3'. The 'Stage View' section displays a table of stages and their durations. The 'Recent Changes' section shows a commit by 'Srienzo' on Sep 06, 2023. The 'Build History' section shows a list of builds, with build #3 being the most recent. The 'Permalinks' section provides links to the last build, last stable build, last successful build, and last completed build. The Jenkins version is 2.342.







Stage	Declarative: Checkout SCM	Static Analysis	Compile	Security Check	Run Unit Tests	Run Integration Tests
Average stage times (Average full run time: ~3s)	1s	49ms	49ms	50ms	51ms	52ms
#3 Sep 06 11:49	1s	49ms	49ms	50ms	51ms	52ms

Permalinks

- Last build (#3), 23 sec ago
- Last stable build (#3), 23 sec ago
- Last successful build (#3), 23 sec ago
- Last completed build (#3), 23 sec ago

 **Jenkins**


Q Search ?


     admin  log out


Dashboard > exp6_3 > #4


↑ Back to Project


Q Status


 Changes


 Console Output


 Edit Build Information


 Delete build '#4'

 Git Build Data


 Restart from Stage

 Replay

 Pipeline Steps


 Workspaces


← Previous Build

 **Build #4 (Sep 6, 2023 11:51:09 AM)**


Keep this build forever


Started 10 sec ago
Took 3.3 sec

 Add description

 Changes

1. Create sdasd.txt ([details](#) / [githubweb](#))
2. Create bhv.txt ([details](#) / [githubweb](#))

 Started by user [admin](#)

 git

Revision: 08e99549f2769d7ac52538158f5a14ca2495474c
Repository: <https://github.com/SmithChauhan2003/jenkins-pipeline-tutorial.git>

- refs/remotes/origin/master

REST API Jenkins 2.342