

1. Create one Declarative pipeline job

<https://github.com/betawins/hiring-app.git>

stages:

- a. Git Clone
- b. SonarQube Integration
- c. Maven Compilation
- d. Nexus Artifactory
- e. Slack Notification
- f. Deploy On tomcat

a. Git clone:

Click on new item enter name as declarative-pipeline and select type as pipeline.

Jenkins / All / New Item

New Item

Enter an item name

declarative-pipeline

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



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

OK

- Select pipeline for pipeline script and select helloworld.
- In scm select git and give your repository and click on save.



The screenshot shows the Jenkins Pipeline configuration page for a job named 'declarative-pipeline'. The 'Pipeline' tab is selected. In the 'Definition' section, 'Pipeline script' is chosen. The script area contains the following Groovy code:

```
1~ pipeline {  
2     agent any  
3  
4~     stages {  
5~         stage('Hello') {  
6~             steps {  
7                 echo 'Hello World'  
8             }  
9         }  
10    }  
11}  
12|
```

A dropdown menu next to the script area is set to 'Hello World'.

Click on pipeline syntax

- select git Git
- Give repository URL

The screenshot shows the Jenkins Pipeline Syntax configuration page. A 'git' step is selected under 'Sample Step'. The configuration includes:

- Repository URL:** https://github.com/betawins/hiring-app.git
- Branch:** main
- Credentials:** - none -
- Include in polling?** (checkbox checked)
- Include in changelog?** (checkbox checked)

A blue button labeled 'Generate Pipeline Script' is visible. Below it, the generated Jenkinsfile code is shown:

```
git branch: 'main', url: 'https://github.com/betawins/hiring-app.git'
```

Click on generate pipeline script.it will create a link.

The screenshot shows the Jenkins Pipeline Syntax configuration page after generating the pipeline script. The generated Jenkinsfile code is displayed in a text area:

```
git branch: 'main', url: 'https://github.com/betawins/hiring-app.git'
```

git branch: 'main', url: ' https://github.com/betawins/hiring-app.git'

add this link in the script.

Jenkins / declarative-pipeline / Configuration

Configure

Definition

Pipeline

Script

```
1 pipeline {  
2     agent any  
3  
4     stages {  
5         stage('Checkout') {  
6             steps {  
7                 git url: 'https://github.com/betawins/hiring-app.git', branch: 'main'  
8             }  
9         }  
10    }  
11}  
12
```

Use Groovy Sandbox

Click on build now.

← → ⌂ ⚠ Not secure 23.23.26.147:8080/job/declarative-pipeline/

Gmail YouTube Maps

Jenkins / declarative-pipeline

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Credentials

Builds

Filter

declarative-pipeline

Average stage times:
(full run time: ~1s)

Checkout	
#27	Nov 06 21:31
#26	Nov 06 21:29
#25	Nov 06 21:27

478ms

478ms

478ms

Git has cloned successfully.

```
[root@ip-172-31-50-64 workspace]# ls  
declarative-pipeline  declarative-pipeline@tmp  hiring-app  
[root@ip-172-31-50-64 workspace]# cd declarative-pipeline  
[root@ip-172-31-50-64 declarative-pipeline]# ls  
Jenkinsfile  Service-API.dockerfile  XMLSave.txt  build.xml  sonar-project.properties  
README.md   WebContent           build          pom.xml   src  
[root@ip-172-31-50-64 declarative-pipeline]#
```

b. Maven Compilation:

give mvn location script should be like.

```
pipeline {

    agent any

    tools{
        maven "MVN_HOME"

    }

    stages {
        stage('checkout') {
            steps {
                https://github.com/betawins/hiring-app.git
            }
        }
        stage('build'){
            steps{
                sh "mvn clean install -f pom.xml"
            }
        }
    }
}
```

{}

← → ⌂ Not secure 23.23.26.147:8080/job/declarative-pipeline/configure

Gmail YouTube Maps

 Jenkins / declarative-pipeline / Configuration

Configure

Define your Pipeline using Groovy directly or pull it from source control.

Definition

General

Triggers

Pipeline

Advanced

Script ?

```

1 pipeline {
2     agent any
3
4     tools {
5         maven 'MVN_HOME'
6     }
7
8     stages {
9
10        stage('Checkout') {
11            steps {
12                git branch: 'main', url: 'https://github.com/betawins/hiring-app.git'
13            }
14        }
15

```

Click on build now.

← → ⌂ Not secure 23.23.26.147:8080/job/declarative-pipeline/

Gmail YouTube Maps

 Jenkins / declarative-pipeline

Status declarative-pipeline

<> Changes declarative-pipeline

▷ Build Now

Configure

>Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Credentials

Builds

Filter

Today

#29 Nov 06 21:38 No Changes

#28 Nov 06 21:35 No Changes

Average stage times:
(full run time: ~6s)

Declarative: Tool Install	Checkout	Build
100ms	426ms	5s
100ms	426ms	5s

Permalinks

- Last build (#29), 2 min 13 sec ago
- Last stable build (#29), 2 min 13 sec ago

```
[root@ip-172-31-50-64 ~]# cd /var/lib/jenkins/workspace
[root@ip-172-31-50-64 workspace]# ls
declarative-pipeline      hiring-app          parameterized-job      scripted-pipeline
declarative-pipeline@tmp  multi-stage-pipeline  parameterized-pipeline  scripted-pipeline@tmp
helloworld                multi-stage-pipeline@tmp parameterized-pipeline@tmp
[root@ip-172-31-50-64 workspace]# cd hiring-app/
[root@ip-172-31-50-64 hiring-app]# ls
Dockerfile   Jenkinsfile  README.md  'Untitled Diagram.drawio'  jenkinsfile-cicd  pom.xml  src  target
[root@ip-172-31-50-64 hiring-app]# cd target/
[root@ip-172-31-50-64 target]# ls
hiring  hiring.war  maven-archiver
[root@ip-172-31-50-64 target]#
```

c. Sonar integration:

go to pipeline syntax and select with sonarqube env:prepare sonarqube scanner environment click generate pipeline script it will give you script add that in the declarative pipeline script.

The screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. On the left, there's a sidebar with links like Snippet Generator, Declarative Directive Generator, Declarative Online Documentation, Steps Reference, Global Variables Reference, Online Documentation, Examples Reference, and IntelliJ IDEA GDSL. The main area has tabs for Overview and Steps. Under Overview, it says: "This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them blank)." Under Steps, there's a section for "Sample Step" with the code "withSonarQubeEnv: Prepare SonarQube Scanner environment". Below that, there's a "Server authentication token" field with a dropdown menu containing "- none -". At the bottom, there's a blue "Generate Pipeline Script" button and a preview window showing the generated code: "withSonarQubeEnv { // some block }".

```
withSonarQubeEnv {
    // some block
}
```

}

← → ⌂ Not secure 54.208.94.198:8080/job/declarative-pipeline/configure

Gmail YouTube Maps

 Jenkins / declarative-pipeline Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

```
11      steps {  
12          git branch: 'main', url: 'https://github.com/betawins/hiring-app.git'  
13      }  
14  }  
15  
16  stage('Build') {  
17      steps {  
18          withSonarQubeEnv("sonarqube") {  
19              sh 'mvn clean install -f pom.xml'  
20          }  
21      }  
22  }  
23  
24 }  
25 }  
26 }
```

Use Groovy Sandbox ?

click on build now.

Not secure 54.208.94.198:8080/jenkins/jobs/declarative-pipeline

 Jenkins / declarative-pipeline

Status  declarative-pipeline

</> Changes declarative-pipeline

▷ Build Now

Configure

 Delete Pipeline

 Full Stage View

 Stages

 Rename

 Pipeline Syntax

 Credentials

Builds *** 

Filter 

Today

Average stage times:
(full run time: ~7s)

	Declarative: Tool Install	Checkout	Build
#31 Nov 06 21:59 No Changes	160ms	833ms	6s
#30 Nov 06 21:49 No Changes	119ms	426ms	8min 34s
#29 Nov 06 21:38 No Changes	100ms	426ms	5s

d. Nexus Artifactory:

← → ⌛ Not secure 54.208.94.198:8080/job/declarative-pipeline/pipeline-syntax/ ↴

Gmail YouTube Maps

Jenkins / declarative-pipeline / Pipeline Syntax

nexusArtifactUploader

Nexus Details

Nexus Version

NEXUS3

Protocol

HTTP

Nexus URL ?

54.81.238.177:8081

Credentials

admin/***** (nexus)

GroupId

injavahome

← → ⌛ Not secure 54.208.94.198:8080/job/declarative-pipeline/pipeline-syntax/ ↴

Gmail YouTube Maps

Jenkins / declarative-pipeline / Pipeline Syntax

Credentials

admin/***** (nexus)

GroupId

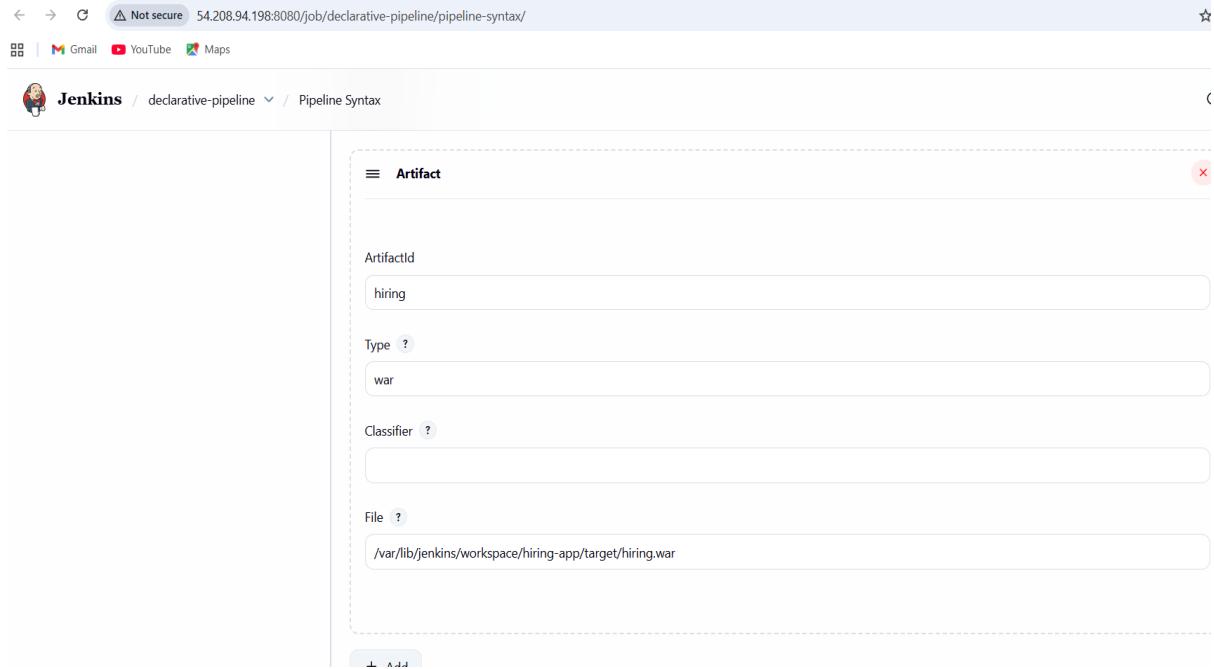
injavahome

Version

0.1

Repository ?

hiring-app



It will give a script add that in the script.

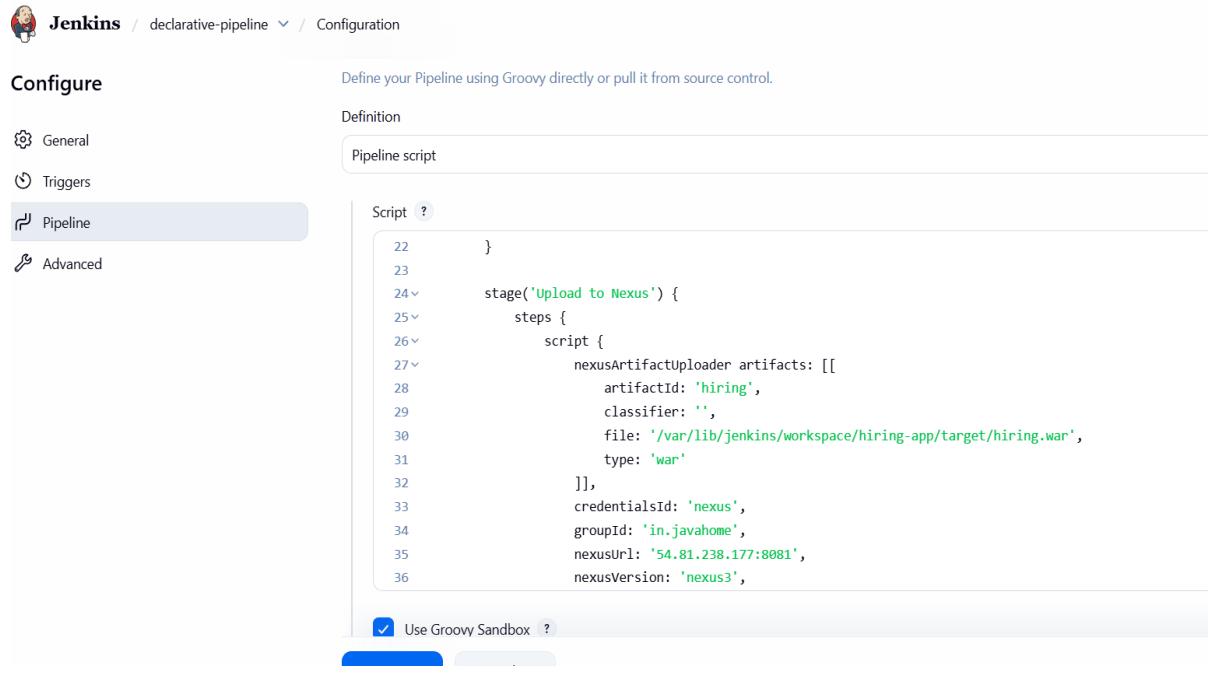
A screenshot of the Jenkins declarative pipeline syntax editor showing the generated Pipeline Script. The URL is 54.208.94.198:8080/job/declarative-pipeline/pipeline-syntax/. The pipeline script is displayed in a code editor. It includes a 'nexusArtifactUploader' step with the following configuration:

```
nexusArtifactUploader artifacts: [[artifactId: 'hiring', classifier: '', file: '/var/lib/jenkins/workspace/hiring-app/target/hiring.war', type: 'war']], credentialsId: 'nexus', groupId: 'in.javahome', nexusUrl: '54.81.238.177:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'hiring-app', version: '0.1'
```

The 'Generate Pipeline Script' button is visible at the bottom of the script area. Below the script, there is a 'Global Variables' section with a note about unsupported Pipeline features.

```
nexusArtifactUploader artifacts: [[artifactId: 'hiring', classifier: '', file: '/var/lib/jenkins/workspace/hiring-app/target/hiring.war', type: 'war']], credentialsId: 'nexus', groupId: 'in.javahome', nexusUrl: '54.81.238.177:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'hiring-app', version: '0.1'
```

`nexusVersion: 'nexus3', protocol: 'http', repository: 'hiring-app', version: '0.1'`



Jenkins / declarative-pipeline / Configuration

Configure

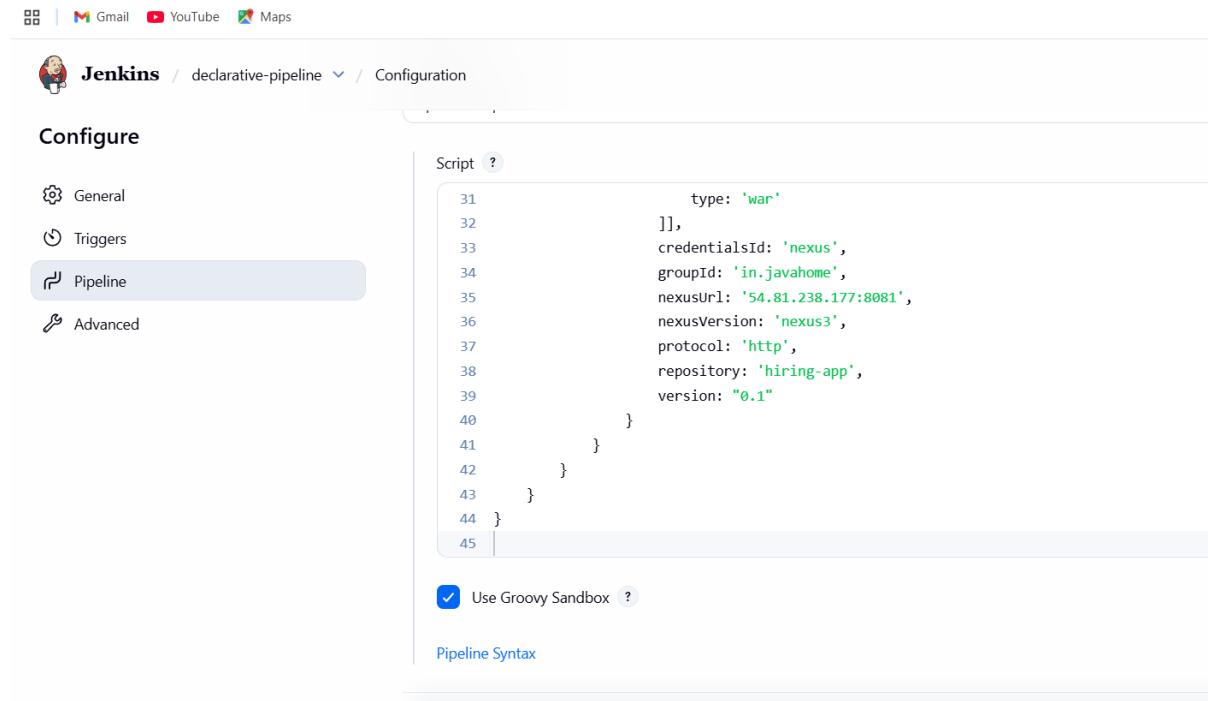
Definition

Pipeline script

Script ?

```
22     }
23
24     stage('Upload to Nexus') {
25         steps {
26             script {
27                 nexusArtifactUploader artifacts: [
28                     artifactId: 'hiring',
29                     classifier: '',
30                     file: '/var/lib/jenkins/workspace/hiring-app/target/hiring.war',
31                     type: 'war'
32                 ],
33                 credentialsId: 'nexus',
34                 groupId: 'in.javahome',
35                 nexusUrl: '54.81.238.177:8081',
36                 nexusVersion: 'nexus3',
37             }
38         }
39     }
40 }
```

Use Groovy Sandbox ?



Gmail YouTube Maps

Jenkins / declarative-pipeline / Configuration

Configure

General

Triggers

Pipeline

Advanced

Script ?

```
31             type: 'war'
32         ],
33         credentialsId: 'nexus',
34         groupId: 'in.javahome',
35         nexusUrl: '54.81.238.177:8081',
36         nexusVersion: 'nexus3',
37         protocol: 'http',
38         repository: 'hiring-app',
39         version: "0.1"
40     }
41 }
42 }
43 }
44 }
45 |
```

Use Groovy Sandbox ?

Pipeline Syntax

Click on build now.

← → ⌂ Not secure 54.196.31.127:8080/job/declarative-pipeline/

Gmail YouTube Maps

 Jenkins / declarative-pipeline

Status declarative-pipeline

</> Changes declarative-pipeline

▷ Build Now

⚙ Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Credentials

Builds

Average stage times:
(full run time: ~8s)

Declarative: Tool Install	Checkout	Build	Upload to Nexus
131ms	610ms	1min 47s	4min 15s
#33 Nov 06 22:44 No Changes	178ms	821ms	6s
#32 Nov 06 22:25 No Changes	101ms	546ms	5s
			8min 30s
			aborted

Sonatype Nexus Repository Manager OSS 3.19.1-01

Search components

Browse

Welcome

Search

Browse

Upload

Upload component HTML View

in

- javahome
- hiring
- 0.1
 - hiring-0.1.war
 - hiring-0.1.war.md5
 - hiring-0.1.war.sha1

f . Deploy On tomcat:

The screenshot shows the Jenkins Pipeline Syntax configuration page. The URL is 54.196.31.127:8080/job/declarative-pipeline/pipeline-syntax/. The configuration is for a 'Sample Step' named 'deploy'. The 'deploy' step is defined with the following parameters:

- WAR/EAR files**: `**/*.war`
- Context path**: (empty)
- Containers**:
 - Tomcat 9.x Remote**:
 - Credentials**: `deployer/******** (tomcat-server)`
 - Tomcat URL**: `http://98.87.16.156:8080/`

The screenshot shows the generated Pipeline Script. The URL is 54.196.31.127:8080/job/declarative-pipeline/pipeline-syntax/. The script is as follows:

```
deploy adapters: [tomcat9(alternativeDeploymentContext: "", credentialsId: 'tomcat-server', path: "", url: 'http://98.87.16.156:8080/']), contextPath: null, war: '**/*.war'
```

deploy adapters: [tomcat9(alternativeDeploymentContext: "", credentialsId: 'tomcat-server', path: "", url: 'http://98.87.16.156:8080/']), contextPath: null, war: '/*.war'**

Jenkins / declarative-pipeline / Configuration

Configure

- General
- Triggers
- Pipeline**
- Advanced

```

46      script {
47          deploy adapters: [tomcat9(
48              alternativeDeploymentContext: '',
49              credentialsId: 'tomcat-server',
50              path: '',
51              url: 'http://98.87.16.156:8080/'
52          )],
53          contextPath: null,
54          war: '**/*.war'
55      }
56  }
57 }
58 }
59 }
60

```

Click on build now.

Not secure 54.196.31.127:8080/job/declarative-pipeline/

declarative-pipeline

- Status
- </> Changes
- ▷ Build Now
- Configure
- Delete Pipeline
- Full Stage View
- Stages
- Rename
- Pipeline Syntax
- Credentials

Builds ...

Average stage times: (full run time: ~8s)	Declarative: Tool Install	Checkout	Build	Upload to Nexus	Deploy to Tomcat
#34 Nov 06 22:53 No Changes	129ms	603ms	1min 30s	2min 50s	1s
#33 Nov 06 22:44 No Changes	178ms	821ms	6s	925ms	
#32 Nov 06 22:25 No Changes	101ms	546ms	5s	8min 30s	aborted

```

[root@ip-172-31-111-184 bin]# cd webapps
-bash: cd: webapps: No such file or directory
[root@ip-172-31-111-184 bin]# cd /opt
[root@ip-172-31-111-184 opt]# ls
apache-tomcat-9.0.111 apache-tomecat-9.0.111.tar.gz aws
[root@ip-172-31-111-184 opt]# cd apache-tomcat-9.0.111/
[root@ip-172-31-111-184 apache-tomcat-9.0.111]# ls
BUILDING.txt CONTRIBUTING.md LICENSE NOTICE README.md RELEASE-NOTES RUNNING.txt bin conf lib logs temp webapps work
[root@ip-172-31-111-184 apache-tomcat-9.0.111]# cd webapps/
[root@ip-172-31-111-184 webapps]# ls
ROOT SimpleCustomerApp-12-SNAPSHOT SimpleCustomerApp-13-SNAPSHOT.war SimpleCustomerApp-21-SNAPSHOT examples host-manager
SimpleCustomerApp-10-SNAPSHOT SimpleCustomerApp-12-SNAPSHOT.war SimpleCustomerApp-20-SNAPSHOT SimpleCustomerApp-21-SNAPSHOT.war hiring manager
SimpleCustomerApp-10-SNAPSHOT.war SimpleCustomerApp-13-SNAPSHOT SimpleCustomerApp-20-SNAPSHOT.war docs hiring.war
[root@ip-172-31-111-184 webapps]#

```

f. Slack Notification:

The screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. The top navigation bar includes the Jenkins logo, the current path "declarative-pipeline > Pipeline Syntax", and a search bar. A sidebar on the left lists various links: Snippet Generator (selected), Declarative Directive Generator, Declarative Online Documentation, Steps Reference, Global Variables Reference, Online Documentation, Examples Reference, and IntelliJ IDEA GDSL. The main content area has a title "Overview" followed by a descriptive text about the Snippet Generator. Below this is a section titled "Steps" with a "Sample Step" example: "slackSend: Send Slack Message". This example is expanded to show configuration options: "Channel" set to "jenkins-integration", "Message" set to "Declarative pipeline executed", and "Advanced" settings with "Edited" status. At the bottom, there is a "Color" section.

This **Snippet Generator** will help you learn the Pipeline Syntax. Click the **Generate Pipeline Script**, and you will see a Pipeline Script generated for you or pick up just the options you care about. (Most parameters are optional)

Steps

Sample Step

slackSend: Send Slack Message

slackSend ?

Channel ?

jenkins-integration

Message ?

Declarative pipeline executed

Advanced ^ Edited

Color ?

→ G Not secure 35.173.132.241:8080/job/declarative-pipeline/pipeline-syntax/

Gmail YouTube Maps

Jenkins / declarative-pipeline / Pipeline Syntax

Credential slack_integration

Custom slack app bot user ?

Notify Committers? ?

Icon Emoji ?

Username ? Shaik Mohammad Mujaheed-15

Override url ?

Fail On Error ?

Generate Pipeline Script

slackSend channel: 'jenkins-integration', message: 'Declarative pipeline executed', tokenCredentialId: 'slack_integration', username: 'Shaik Mohammad Mujaheed-15'

This screenshot shows the Jenkins Pipeline Syntax configuration page. It displays fields for a Slack credential named 'slack_integration'. The 'Fail On Error' checkbox is checked. A large blue button labeled 'Generate Pipeline Script' is visible. Below it, a message box shows a sample slackSend command.

slackSend channel: 'jenkins-integration', message: 'Declarative pipeline executed', tokenCredentialId: 'slack_integration', username: 'Shaik Mohammad Mujaheed-15'

Jenkins / declarative-pipeline / Configuration

Configure General Triggers Pipeline Advanced

Definition Pipeline script

Script ? try sample Pipeline... ▾

```
55      }
56    }
57  }
58
59  stage('Slack Notification') {
60    steps {
61      slackSend(
62        channel: 'jenkins-integration',
63        message: "Declarative pipeline executed successfully - Build #${BUILD_NUMBER}",
64        tokenCredentialId: 'slack_integration',
65        username: 'Shaik Mohammad Mujaheed-15'
66      )
67    }
68  }
69 }
```

Use Groovy Sandbox ?

This screenshot shows the Jenkins Pipeline Configuration page under the 'Pipeline' tab. It displays a Groovy script block containing a stage named 'Slack Notification' that sends a message to a Slack channel using the 'slackSend' step. The 'Use Groovy Sandbox' checkbox is checked.

click on build now.

The screenshot shows the Jenkins declarative pipeline job page. On the left, there's a sidebar with options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, Pipeline Syntax, and Credentials. Below that is a Builds section with a search bar and a list of recent builds: #21 (Nov 04 17:57), #20 (Nov 04 17:40), and #19 (Nov 04 16:51). A Today link is also present. On the right, the main area is titled "declarative-pipeline" with a green checkmark icon. It shows a "Stage View" table with columns: Declarative: Tool Install, Checkout, Build, Upload to Nexus, Deploy to Tomcat, and Slack Notification. The table has three rows corresponding to the builds. The "Build" column shows times: 6s, 6s, and 6s. The "Upload to Nexus" column shows times: 359ms, 422ms, and 1s. The "Deploy to Tomcat" column shows times: 357ms, 724ms, and 1s. The "Slack Notification" column shows times: 810ms, 810ms, and 1s. A note above the table says "Average stage times: (full run time: ~10s)". Below the table is a "Permalinks" section. At the bottom of the page, there's a Jenkins app notification: "jenkins APP 5:57 PM ✓ Declarative pipeline executed successfully – Build #21".

pipeline {

agent any

tools {

maven 'MVN_HOME'

}

stages {

stage('Checkout') {

steps {

```
        git branch: 'main', url:  
        'https://github.com/betawins/hiring-app.git'  
    }  
  
    }  
  
stage('Build') {  
    steps {  
        withSonarQubeEnv("sonarqube") {  
            sh 'mvn clean install -f pom.xml'  
        }  
    }  
}  
  
stage('Upload to Nexus') {  
    steps {  
        script {  
            nexusArtifactUploader artifacts: [[  
                artifactId: 'hiring',  
                classifier: '',  
                file: '/var/lib/jenkins/workspace/hiring-  
app/target/hiring.war',  
            ]]  
        }  
    }  
}
```

```
    type: 'war'  
  ]],  
  credentialsId: 'nexus',  
  groupId: 'in.javahome',  
  nexusUrl: '54.81.238.177:8081',  
  nexusVersion: 'nexus3',  
  protocol: 'http',  
  repository: 'hiring-app',  
  version: "0.1"  
  
}  
}  
}
```

```
stage('Deploy to Tomcat') {  
  steps {  
    script {  
      deploy adapters: [tomcat9(  
        alternativeDeploymentContext: '',  
        credentialsId: 'tomcat-server',  
        path: '',  
        url: 'http://98.87.16.156:8080/'
```

```
        )],  
        contextPath: null,  
        war: '**/*.war'  
    }  
}  
}  
}  
}
```

2. Setup a Jenkins CICD pipeline using Scripted pipeline using <https://github.com/betawins/hiring-app.git>

stages:

- a. Git Clone**
- b. SonarQube Integration**
- c. Maven Compilation**
- d. Nexus Artifactory**
- e. Slack Notification**
- f. Deploy On tomcat**

a. git clone:

go to Jenkins add new item name as scripted pipeline select type as pipeline.

← → ⌂ Not secure 35.173.132.241:8080/view/all/newJob

Gmail YouTube Maps

Jenkins / All / New Item

New Item

Enter an item name

Select an item type

-  Freestyle project
Classic, general-purpose job type that checks out from steps like archiving artifacts and sending email notifications.
-  Maven project
Build a maven project. Jenkins takes advantage of your
-  Pipeline
Orchestrates long-running activities that can span multiple workflows (e.g., parallel builds) and/or organizing complex activities that do
-  Multi-configuration project
Suitable for projects that need a large number of different configurations (e.g., platform-specific builds, etc.)
-  Folder

OK

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

```
1~ node{  
2~   stage("Git clone"){  
3~     }  
4~   }
```

try sample Pipeline... ▾

Save

Apply



Online Documentation

Sample Step

Examples Reference

git: Git

IntelliJ IDEA GDSL

git ?

Repository URL ?

https://github.com/betawins/hiring-app.git

Branch ?

main

Credentials ?

- none -

Include in polling? ?

Include in changelog? ?

Generate Pipeline Script

git branch: 'main', url: 'https://github.com/betawins/hiring-app.git'

git branch: 'main', url: ' https://github.com/betawins/hiring-app.git '

add this in script.

← → ⌂ Not secure 98.91.216.236:8080/job/scripted-pipeline/configure

Gmail YouTube Maps

Jenkins / scripted-pipeline Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

Script ?

```
1 node {  
2   stage("Git Clone") {  
3     git(' https://github.com/betawins/hiring-app.git ')  
4   }  
5 }
```

← → ⌂ Not secure 98.91.216.236:8080/job/scripted-pipeline/

Gmail YouTube Maps

Jenkins / scripted-pipeline

Status ✓ **scripted-pipeline**

</> Changes

▷ Build Now

Configure

>Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Credentials

Builds

Filter

Stage View

Git Clone

Average stage times:
(full run time: ~20s)

#15	Nov 07 12:26	No Changes	489ms
#14	Nov 07 12:25	No Changes	537ms
#13	Nov 07 ...	No Changes	

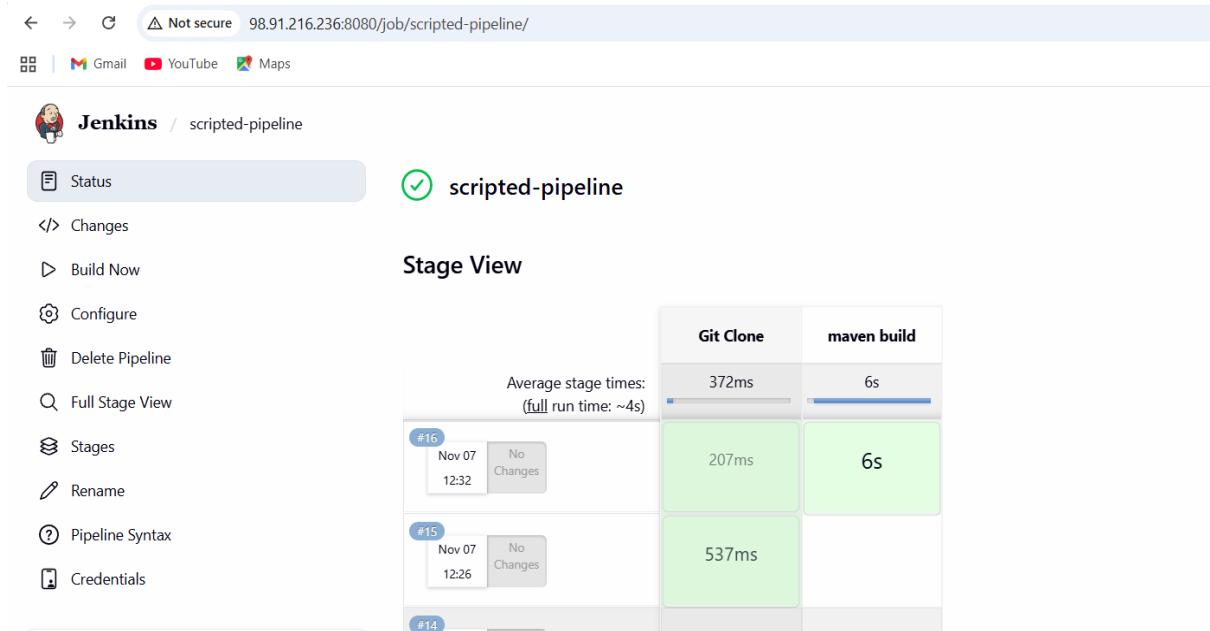
b. maven compilation:

```

node {
    def mvnHome = tool "MVN_HOME"
    stage("Git Clone") {
        git(' https://github.com/betawins/hiring-app.git ')
    }
    stage("maven build"){
        sh "${mvnHome}/bin/mvn clean install -f pom.xml"
    }
}

```

click on build now.



c. sonarqube integration:

click on pipeline syntax and add with sonarqube env click on generate pipeline syntax.add the syntax in the script.

The screenshot shows the Jenkins Snippet Generator interface. On the left, there's a sidebar with links like 'Declarative Directive Generator', 'Declarative Online Documentation', 'Steps Reference', 'Global Variables Reference', 'Online Documentation', 'Examples Reference', and 'IntelliJ IDEA GDSL'. The main area has a title 'Overview' and a description of the Snippet Generator. Below that, under 'Steps', is a 'Sample Step' section for 'withSonarQubeEnv: Prepare SonarQube Scanner environment'. It includes a dropdown for 'Server authentication token' with an option '- none -'. A 'Generate Pipeline Script' button is present, and the generated Groovy code is shown:

```
withSonarQubeEnv {
    // some block
}
```

```
withSonarQubeEnv {  
    // some block  
}
```

The screenshot shows the Jenkins Pipeline configuration page. The left sidebar has tabs for 'General', 'Triggers', 'Pipeline' (which is selected), and 'Advanced'. The right side has a 'Pipeline script' tab with a 'Script' editor. The Groovy script content is:

```
1< node {  
2     def mvnHome = tool "MVN_HOME"  
3     stage("Git Clone") {  
4         git branch: 'main', url: 'https://github.com/betawins/hiring-app.git'  
5     }  
6     stage("Maven Build") {  
7         sh "${mvnHome}/bin/mvn clean install -f pom.xml"  
8     }  
9     stage("SonarQube Analysis") {  
10        withSonarQubeEnv('sonarqube') {  
11            sh "${mvnHome}/bin/mvn sonar:sonar -f pom.xml -Dsonar.qualitygate.wait=false"  
12        }  
13    }  
14}  
15
```

There is also a checked checkbox for 'Use Groovy Sandbox'.

Click on build now.

Jenkins / scripted-pipeline

- Status
- </> Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- SonarQube
- Stages
- Rename
- Pipeline Syntax
- Credentials

Stage View

	Git Clone	Maven Build	SonarQube Analysis
Average stage times: (full run time: ~21s)	453ms	5s	7min 19s
#22 Nov 07 13:25 No Changes	631ms	5s	14s
#21 Nov 07 13:11 No Changes	274ms	5s	13min 38s
#20 Nov 07 No Changes	616ms	5s	3min 1s

d. Nexus Artifactory:

create a repository with scripted pipeline name.

Sonatype Nexus Repository Manager OSS 3.19.1-01

Administration

- Repository
 - Repositories
 - Blob Stores
 - Content Selectors
 - Cleanup Policies
 - Routing Rules
- Security
 - Privileges
 - Roles
 - Users
 - Anonymous
 - LDAP
 - Realms
 - SSL Certificates
- IQ Server
- Support

Repositories / Select Recipe / Create Repository: maven2 (hosted)

Name: A unique identifier for this repository
scripted-pipeline

Online: If checked, the repository accepts incoming requests

Maven 2

Version policy: What type of artifacts does this repository store?
Snapshot

Layout policy: Validate that all paths are Maven artifact or metadata paths
Permissive

Storage

Blob store: Blob store used to store repository contents
default

Strict Content Type Validation: Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format

Hosted

 Jenkins / scripted-pipeline / Pipeline Syntax

② Global Variables Reference
② Online Documentation
② Examples Reference
② IntelliJ IDEA GDSL

Steps

Sample Step

nexusArtifactUploader: Nexus Artifact Uploader

nexusArtifactUploader

Nexus Details

Nexus Version

NEXUS3

Protocol

HTTP

Nexus URL ?

23.22.130.233:8081

← → ⌂ △ Not secure 54.172.130.57:8080/job/scripted-pipeline/pipeline-syntax/

Gmail YouTube Maps

 Jenkins / scripted-pipeline / Pipeline Syntax

Nexus URL ?
23.22.130.233:8081

Credentials
admin/******** (nexus)

GroupId
in.javahome

Version
0.1

Repository ?
scripted-pipeline

Artifacts

Artifact

The screenshot shows two parts of the Jenkins Pipeline Syntax configuration interface.

Top Section (Configuration):

- Artifact:**
 - ArtifactId:** hiring
 - Type:** war
 - Classifier:** (empty)
 - File:** /var/lib/jenkins/workspace/scripted-pipeline/target

Bottom Section (Generated Script):

- File:** /var/lib/jenkins/workspace/hiring-app/hiring.war
- Add** button
- Generate Pipeline Script** button
- Script Output:**

```
nexusArtifactUploader artifacts: [[artifactId: 'hiring', classifier: '', file: '/var/lib/jenkins/workspace/hiring-app/hiring.war', type: 'war']], credentialsId: 'nexus', groupId: 'in.javahome', nexusUrl: '23.22.130.233:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'scripted-pipeline', version: '0.1'
```

```
nexusArtifactUploader artifacts: [[artifactId: 'hiring', classifier: '', file: '/var/lib/jenkins/workspace/ scripted-pipeline/hiring.war', type: 'war']], credentialsId: 'nexus', groupId: 'in.javahome', nexusUrl: '23.22.130.233:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'scripted-pipeline', version: '0.1'
```

← → ⌂ Not secure 98.86.226.194:8080/job/scripted-pipeline/configure

Gmail YouTube Maps

 Jenkins / scripted-pipeline / Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

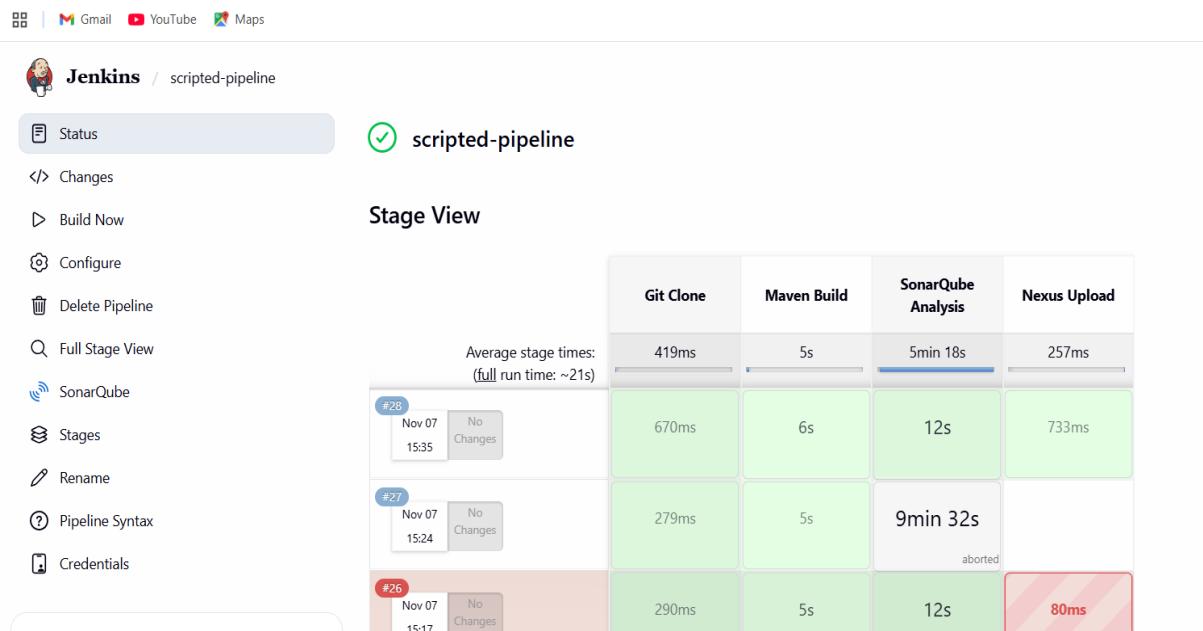
```

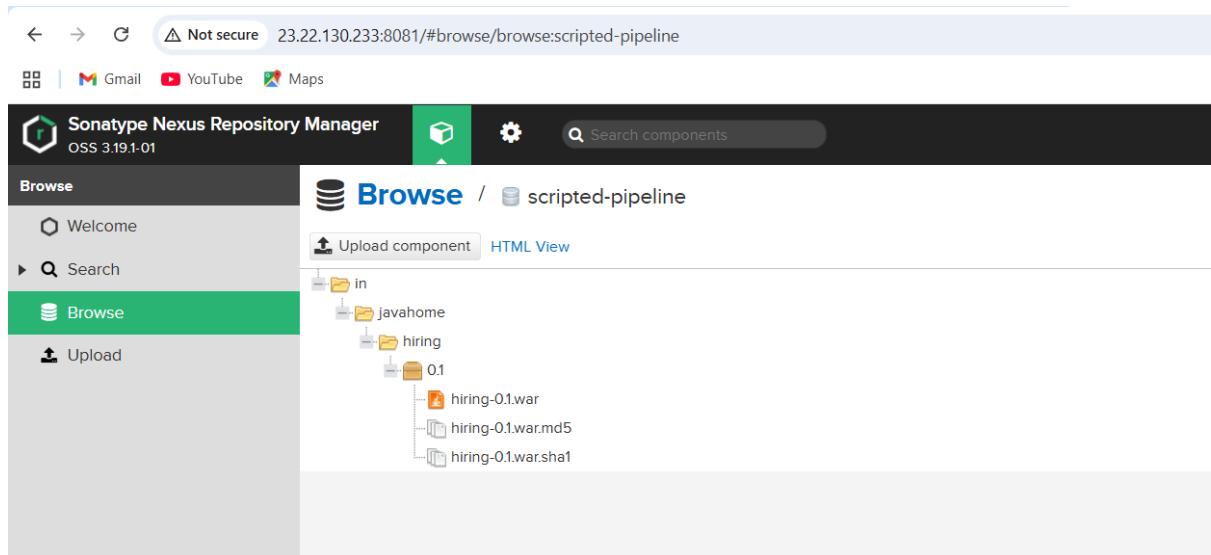
5      }
6  stage("Maven Build") {
7      sh "${mvnHome}/bin/mvn clean install -f pom.xml"
8  }
9  stage("SonarQube Analysis") {
10     withSonarQubeEnv('sonarqube') {
11         sh "${mvnHome}/bin/mvn sonar:sonar -f pom.xml -Dsonar.qualitygate.wait=false"
12     }
13 }
14 stage("Nexus Upload") {
15     nexusArtifactUploader artifacts: [[artifactId: 'hiring', classifier: '', file: '/var/lib/jenkins/']]
16 }
17 }
18

```

Use Groovy Sandbox ?

Click on build now.





e. deploy to tomcat:

A screenshot of the Jenkins Pipeline Syntax configuration page. The top navigation bar shows a warning icon for 'Not secure' and the URL '23.22.130.233:8081/#scripted-pipeline'. Below the bar are links for Gmail, YouTube, and Maps. The main header 'Jenkins / scripted-pipeline / Pipeline Syntax' has a 'Declarative Online Documentation' link. The left sidebar lists 'Steps Reference', 'Global Variables Reference', 'Online Documentation', 'Examples Reference', and 'IntelliJ IDEA GDSL'. The right panel shows a 'Steps' section with a 'Sample Step' labeled 'deploy: Deploy war/ear to a container'. Below it is a 'deploy' step configuration with fields for 'WAR/EAR files' containing '**/*.war', 'Context path' (empty), and 'Containers'. Under 'Containers', there is a 'Tomcat 9.x Remote' section with a 'Credentials' field containing 'deployer/******** (tomcat-server)'.

The screenshot shows the Jenkins Pipeline Syntax configuration page for a job named "scripted-pipeline". The configuration is for a "Tomcat 9.x Remote" adapter. It includes:

- Credentials:** "deployer/******** (tomcat-server)"
- Tomcat URL:** "http://23.20.83.85:8080/"
- Advanced Options:** "Deploy on failure" is checked.
- Generate Pipeline Script:** A button to generate the Groovy script.
- Script Preview:** The generated Groovy code:

```
deploy adapters: [tomcat9(alternativeDeploymentContext: "", credentialsId: 'tomcat-server', path: "", url: 'http://23.20.83.85:8080/')), contextPath: null, war: '**/*.war'
```

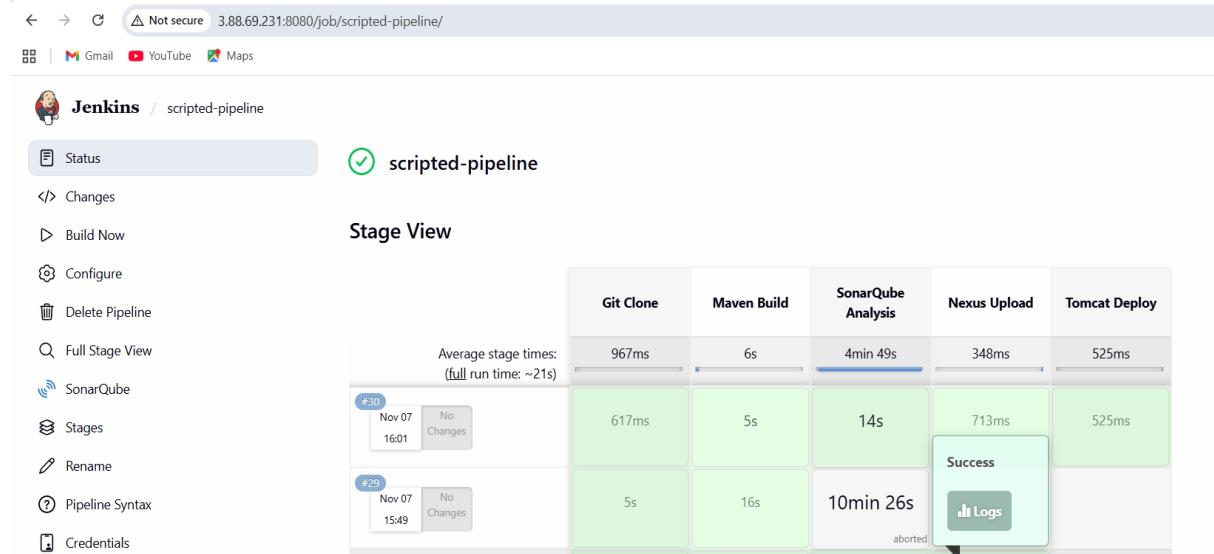
```
deploy adapters: [tomcat9(alternativeDeploymentContext: "",  
credentialsId: 'tomcat-server', path: "", url:  
'http://23.20.83.85:8080/']), contextPath: null, war:  
'**/*.war'
```

The screenshot shows the Jenkins Configuration screen for a job named "scripted-pipeline". The "Pipeline" section is selected in the left sidebar. The main area displays the Groovy pipeline script:

```
8 }  
9 v stage("SonarQube Analysis") {  
10 v   withSonarQubeEnv('sonarqube') {  
11     sh "${mvnHome}/bin/mvn sonar:sonar -f pom.xml -Dsonar.qualitygate.wait=false"  
12   }  
13 }  
14 v stage("Nexus Upload") {  
15   nexusArtifactUploader artifacts: [[artifactId: 'hiring', classifier: '', file: '/var/lib/jenkins/workspace/se  
16 }  
17 v stage("Tomcat Deploy") {  
18   deploy adapters: [tomcat9(alternativeDeploymentContext: "", credentialsId: 'tomcat-server', path: "", url: '  
19 }  
20 }
```

A checkbox for "Use Groovy Sandbox" is visible at the bottom of the script editor.

Click on build now.



f. slack integration:

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration or pick up just the options you care about. (Most parameters are optional and can be omitted in your script)

Steps

Sample Step

```
slackSend: Send Slack Message
```

slackSend ?

Channel ?
jenkins-integration

Message ?
scripted pipeline executed

Advanced ^ Edited

Not secure 13.217.115.239:8080/job/scripted-pipeline/pipeline-syntax/

Gmail YouTube Maps

Jenkins / scripted-pipeline / Pipeline Syntax

Credential : slack_integration

Custom slack app bot user ?

Notify Committers? ?

Icon Emoji ?

Username ? shaik mohammad mujaheed-15

Override url ?

Fail On Error ?

Generate Pipeline Script

slackSend channel: 'jenkins-integration', message: 'scripted pipeline executed', tokenCredentialId: 'slack_integration', username: 'shaik mohammad mujaheed-15'

slackSend channel: 'jenkins-integration', message: 'scripted pipeline executed', tokenCredentialId: 'slack_integration', username: 'shaik mohammad mujaheed-15'

Not secure 34.229.41.116:8080/job/scripted-pipeline/configure

Gmail YouTube Maps

Jenkins / scripted-pipeline / Configuration

Configure

General Triggers Pipeline Advanced

Script ?

```
57      stage("Slack Notification") {  
58        echo "# Sending Slack notification..."  
59        slackSend(  
60          channel: 'jenkins-integration',  
61          message: "Scripted pipeline executed successfully for *simpleCustomerApp* (Build #${BUILD_NUMBER})",  
62          tokenCredentialId: 'slack_integration',  
63          username: 'Shaik Mohammad Mujaheed-15'  
64        )  
65      }  
66    }  
67  
68    } .catch (err) {  
69      // Send failure message to Slack if any stage fails  
70      slackSend(  
71        channel: 'jenkins-integration'  
72      )  
73    }  
74  }  
75  
76  }
```

try sample Pipeline... ▾

Use Groovy Sandbox ?

Pipeline Syntax

Gmail YouTube Maps

Jenkins / scripted-pipeline / Configuration

Configure

- General
- Triggers
- Pipeline**
- Advanced

```

66    }
67
68    } catch (err) {
69      // Send failure message to Slack if any stage fails
70      slackSend(
71        channel: 'jenkins-integration',
72        message: "🔴 Scripted pipeline *FAILED* for *SimpleCustomerApp* (Build #${BUILD_NUMBER})\nError: ${err}"
73        tokenCredentialId: 'slack_integration',
74        username: 'Shaik Mohammad Mujahed-15'
75      )
76      throw err
77    }
78  }
79

```

Use Groovy Sandbox ?

Pipeline Syntax

Not secure 34.229.41.116:8080/job/scripted-pipeline/

Jenkins / scripted-pipeline

Status **scripted-pipeline** Add descri...

Changes Build Now Configure Delete Pipeline Full Stage View SonarQube Stages Rename Pipeline Syntax Credentials

Stage View

Git Clone	Maven Build	SonarQube Analysis	Upload Artifact to Nexus	Deploy to Tomcat	Slack Notification
500ms	5s	15min 30s	15s	622ms	646ms
#12 Nov 04 22:47 No Changes	581ms	5s	11s	858ms	558ms
#11 Nov 04 22:32 No Changes	368ms	5s	9min 47s	Success	646ms

```

node {

  def mvnHome = tool "MVN_HOME"

  stage("Git Clone") {
    git branch: 'main', url:
    'https://github.com/betawins/hiring-app.git'
  }

  stage("Maven Build") {
    sh "${mvnHome}/bin/mvn clean install -f pom.xml"
  }
}

```

```
}

stage("SonarQube Analysis") {

    withSonarQubeEnv('sonarqube') {

        sh "${mvnHome}/bin/mvn sonar:sonar -f pom.xml -Dsonar.qualitygate.wait=false"
    }

    stage("Nexus Upload") {

        nexusArtifactUploader artifacts: [[artifactId: 'hirings', classifier: '', file: '/var/lib/jenkins/workspace/scripted-pipeline/target/hirings.war', type: 'war']], credentialsId: 'nexus', groupId: 'in.javahome', nexusUrl: '23.22.130.233:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'scripted-pipeline', version: '0.1'
    }

    stage("Tomcat Deploy") {

        deploy adapters:
            [tomcat9(alternativeDeploymentContext: '', credentialsId: 'tomcat-server', path: '', url: 'http://23.20.83.85:8080/'), contextPath: null, war: '**/*.war']
    }
}
```

3. create one multi stage pipeline job.

Create a new item name as multi stage pipeline and select pipeline type.

The screenshot shows the Jenkins 'New Item' creation interface. At the top, there is a browser header with tabs for 'Not secure' and the URL '54.90.216.124:8080/view/all/newJob'. Below the header, the Jenkins logo and navigation links ('All', 'New Item') are visible. The main section is titled 'New Item' and contains a text input field labeled 'Enter an item name' with the value 'multi-stage-pipeline'. Below this, a section titled 'Select an item type' lists several options:

- Freestyle project**: Described as a classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**: Described as building a Maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline**: Described as orchestrating 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. This option is highlighted with a light gray background.
- Multi-configuration project**: Described as suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**: A placeholder for creating a folder to organize other Jenkins items.

At the bottom of the form is a blue 'OK' button.

Go the job and write script to clone
<https://github.com/betawins/hiring-app.git> this repository.

← → ⌂ Not secure 3.90.39.73:8080/job/multi-stage-pipeline/configure

Gmail YouTube Maps

Jenkins / multi-stage-pipeline Configuration

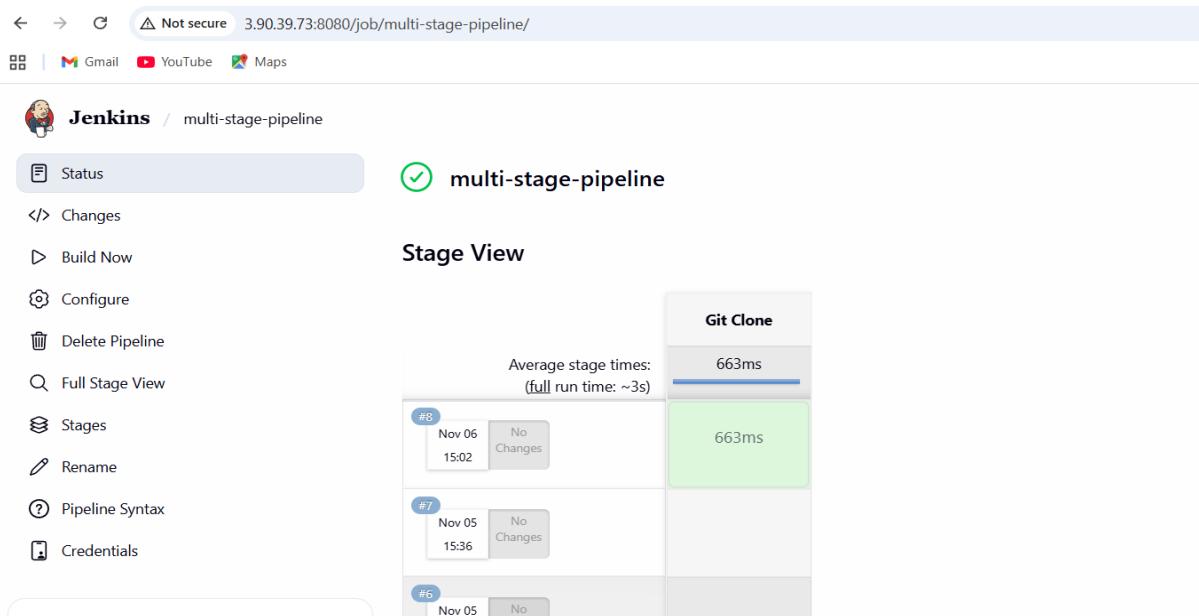
Configure

Definition

Pipeline script

```
1 pipeline {  
2     agent any  
3     stages {  
4         stage('Git Clone') {  
5             steps {  
6                 git branch: 'main', url: 'https://github.com/betawins/hiring-app.git'  
7             }  
8         }  
9     }  
10 }  
11 }
```

Use Groovy Sandbox ?



Configure

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

```

1< pipeline {
2   agent any
3
4<   tools {
5     maven 'MVN_HOME'
6   }
7
8<   stages {
9
10<     stage('Checkout') {
11       steps {
12         git branch: 'main', url: 'https://github.com/betawins/hiring-app.git'
13       }
14     }
15

```

 **Jenkins** / multi-stage-pipeline

Status  **multi-stage-pipeline**

</> Changes

▷ Build Now

Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Stage View

Average stage times:
(full run time: ~9s)

Declarative: Tool Install	Checkout	Build
124ms	776ms	5s
124ms	776ms	5s

#13 Nov 06 16:09 No Changes

Configure

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

```
Script ?  
22 }  
23  
24 stage('Upload to Nexus') {  
25 steps {  
26 script {  
27 nexusArtifactUploader artifacts: [[  
28 artifactId: 'hiring',  
29 classifier: '',  
30 file: '/var/lib/jenkins/workspace/hiring-app/target/hiring.war',  
31 type: 'war'  
32 ]],  
33 credentialsId: 'nexus',  
34 groupId: 'in.javahome',  
35 nexusUrl: '54.81.238.177:8081',  
36 nexusVersion: 'nexus3',  
37  
38  
39  
40  
41  
42  
43  
44  
45
```

Use Groovy Sandbox ?

Configure

General

Triggers

Pipeline

Advanced

Script ?

```
31 type: 'war'  
32 ]],  
33 credentialsId: 'nexus',  
34 groupId: 'in.javahome',  
35 nexusUrl: '54.81.238.177:8081',  
36 nexusVersion: 'nexus3',  
37 protocol: 'http',  
38 repository: 'hiring-app',  
39 version: "0.1"  
40 }  
41 }  
42 }  
43 }  
44 }  
45
```

Use Groovy Sandbox ?

[Pipeline Syntax](#)

← → ⌛ Not secure 54.81.238.177:8081/#browse/browse:hiring-app

Gmail YouTube Maps

Sonatype Nexus Repository Manager OSS 3.19.1-01

Browse / hiring-app

Upload component HTML View

in

- javahome
 - hiring
 - 0.1
 - hiring-0.1.war
 - hiring-0.1.war.md5
 - hiring-0.1.war.sha1

← → ⌛ Not secure 3.82.7.203:8080/job/multi-stage-pipeline/

Gmail YouTube Maps

Jenkins / multi-stage-pipeline

Status multi-stage-pipeline

</> Changes

▷ Build Now

Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Credentials

Stage View

	Declarative: Tool Install	Checkout	Build	Upload to Nexus
Average stage times: (full run time: ~9s)	124ms	776ms	5s	1s
#13 Nov 06 16:09 No Changes	124ms	776ms	5s	1s

Permalinks

- Last build (#13), 7 min 57 sec ago
- Last stable build (#13), 7 min 57 sec ago
- Last successful build (#13), 7 min 57 sec ago
- Last failed build (#11), 44 min ago
- Last unsuccessful build (#12), 42 min ago
- Last completed build (#13), 7 min 57 sec ago

Builds

Filter

Today

#13 10:39 AM

The screenshot shows the Jenkins Pipeline configuration page. On the left, there are tabs: General, Triggers, Pipeline (which is selected and highlighted in blue), and Advanced. The main area is titled "Script" and contains the Groovy code for the pipeline. The code defines a script block with various deployment configurations for a Tomcat adapter.

```
script {
    deploy adapters: [tomcat9(
        alternativeDeploymentContext: '',
        credentialsId: 'tomcat-server',
        path: '',
        url: 'http://98.87.16.156:8080/'
    )],
    contextPath: null,
    war: '**/*.war'
}
```

The screenshot shows the Jenkins interface for a 'multi-stage-pipeline'. On the left, there's a sidebar with links like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, Pipeline Syntax, and Credentials. Below that is a 'Builds' section with a filter and a table for builds from today, showing entries for #13 and #12.

The main area has a title 'multi-stage-pipeline' with a green checkmark icon. It features a 'Stage View' section with a timeline showing stages: Declarative: Tool Install (124ms), Checkout (776ms), Build (5s), Upload to Nexus (1s), and Deploy to Tomcat (716ms). A tooltip for the first stage says 'Average stage times: (full run time: ~9s)'. Below this is a table for build #13, showing the same stages with their respective times.

Underneath the Stage View is a 'Permalinks' section with a bulleted list of links to various build artifacts:

- Last build (#13), 7 min 57 sec ago
- Last stable build (#13), 7 min 57 sec ago
- Last successful build (#13), 7 min 57 sec ago
- Last failed build (#11), 44 min ago
- Last unsuccessful build (#12), 42 min ago
- Last completed build (#13), 7 min 57 sec ago

4. Create one parallel stage pipeline job

Create a new job with the name parallel pipeline and type as pipeline.

New Item

Enter an item name

parallel-pipeline

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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.

Write this script

```
pipeline{
```

```
    agent any
```

```
    stages {
```

```
        stage('stage-1') {
```

```
            steps {
```

```
                echo 'stage-1 started'
```

```
                sleep(time: 5, unit: 'SECONDS')
```

```
                echo 'stage-1 completed'
```

```

    }

}

stage('stage-2') {

    steps {
        echo 'stage-2 started'
        sleep(time: 5, unit: 'SECONDS')
        echo 'stage-2 completed'
    }
}

}

```

← → ⓘ Not secure 13.222.196.20:8080/job/parallel-pipeline/configure

Gmail YouTube Maps

 Jenkins / parallel-pipeline / Configuration

Configure Define your Pipeline using Groovy directly or pull it from source control.

General Triggers Pipeline Advanced

Definition Pipeline script

Script ?

```

4v      stages {
5v        stage('stage-1') {
6v          steps {
7            echo 'stage-1 started'
8            sleep(time: 5, unit: 'SECONDS')
9            echo 'stage-1 completed'
10       }
11     }
12
13v      stage('stage-2') {
14v        steps {
15          echo 'stage-2 started'
16          sleep(time: 5, unit: 'SECONDS')
17          echo 'stage-2 completed'
18        }
}

```

Use Groovy Sandbox ?

Click on build now.

Here 1st stage will be executed after 5 sec the 2nd stage will be started.

The screenshot shows the Jenkins interface for a pipeline named 'parallel-pipeline'. On the left, there's a sidebar with options like Changes, Build Now (which is selected), Configure, Delete Pipeline, Full Stage View, Stages, Rename, Pipeline Syntax, and Credentials. Below that is a 'Builds' section with a 'Filter' input and a 'Today' link. The main area is titled 'Stage View' and displays two parallel stages. A legend indicates that light blue represents 'stage-1' and light green represents 'stage-2'. Stage 1 has a duration of 5s, and Stage 2 also has a duration of 5s. Both stages are currently in progress, as indicated by the progress bars. Below the stages, a message says 'Average stage times: (full run time: ~10s)'. At the bottom, there's a 'Permalinks' section with links to the last build, failed build, and unsuccessful build.

stage-1	stage-2
5s	5s
5s	5s

Average stage times:
(full run time: ~10s)

#2 Nov 05 21:20 No Changes

#1 Nov 05 21:19 No Changes

Permalinks

- Last build (#1), 1 min 32 sec ago
- Last failed build (#1), 1 min 32 sec ago
- Last unsuccessful build (#1), 1 min 32 sec ago

Now we will create parallel stages inside that parallel stage we will keep stage1 and stage 2. Stage 1 will take 5 min to execute after that stage 2 will be executed.

```
pipeline{
```

```
    agent any
```

```
    stages {
```

```
        stage('ParallelStages') {
```

```
            parallel {
```

```
                stage('stage-1') {
```

```
steps {  
    echo 'stage-1 started'  
    sleep(time: 5, unit: 'MINUTES')  
    echo 'stage-1 completed'  
}  
}  
  
stage('stage-2') {  
    steps {  
        echo 'stage-2 started'  
        sleep(time: 5, unit: 'MINUTES')  
        echo 'stage-2 completed'  
    }  
}  
}  
}
```

The screenshot shows the Jenkins Pipeline configuration page for a job named "parallel-pipeline". The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area contains a "Script" editor with the following Groovy code:

```
stage('ParallelStages') {
    parallel {
        stage('stage-1') {
            steps {
                echo 'stage-1 started'
                sleep(time: 5, unit: 'MINUTES')
                echo 'stage-1 completed'
            }
        }
        stage('stage-2') {
            steps {
                echo 'stage-2 started'
                sleep(time: 5, unit: 'MINUTES')
                echo 'stage-2 completed'
            }
        }
    }
}
```

Below the script, there is a checkbox labeled "Use Groovy Sandbox" which is checked.

Click on build now.

← → ⌂ Not secure 13.222.196.20:8080/job/parallel-pipeline/

Gmail YouTube Maps

 Jenkins / parallel-pipeline

Status Changes Build Now Configure Delete Pipeline Full Stage View Stages Rename Pipeline Syntax Credentials

 parallel-pipeline

Stage View

Average stage times:
(full run time: ~5min 0s)

ParallelStages	stage-1	stage-2
83ms	5min 0s	5min 0s

#4 Nov 05 21:42 No Changes

83ms 5min 0s 5min 0s

Permalinks

- Last build (#4), 5 min 29 sec ago
- Last stable build (#4), 5 min 29 sec ago
- Last successful build (#4), 5 min 29 sec ago
- Last failed build (#1), 29 min ago

As we can see in final output the average time took is 5 minutes 0

seconds for executing both stages in parallel way.

- By this parallel pipelines, we can reduce our build time by

50% which are
independent to each other to improve efficiency