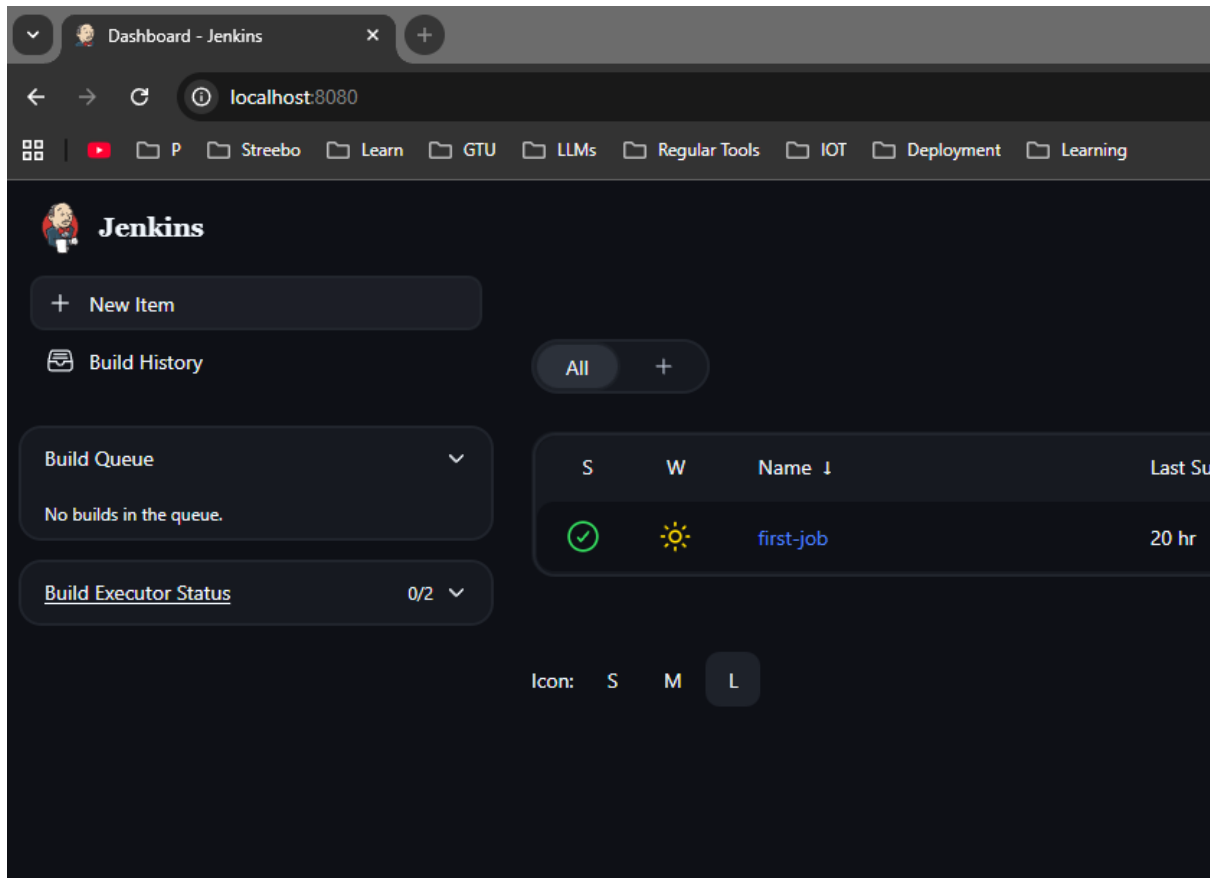
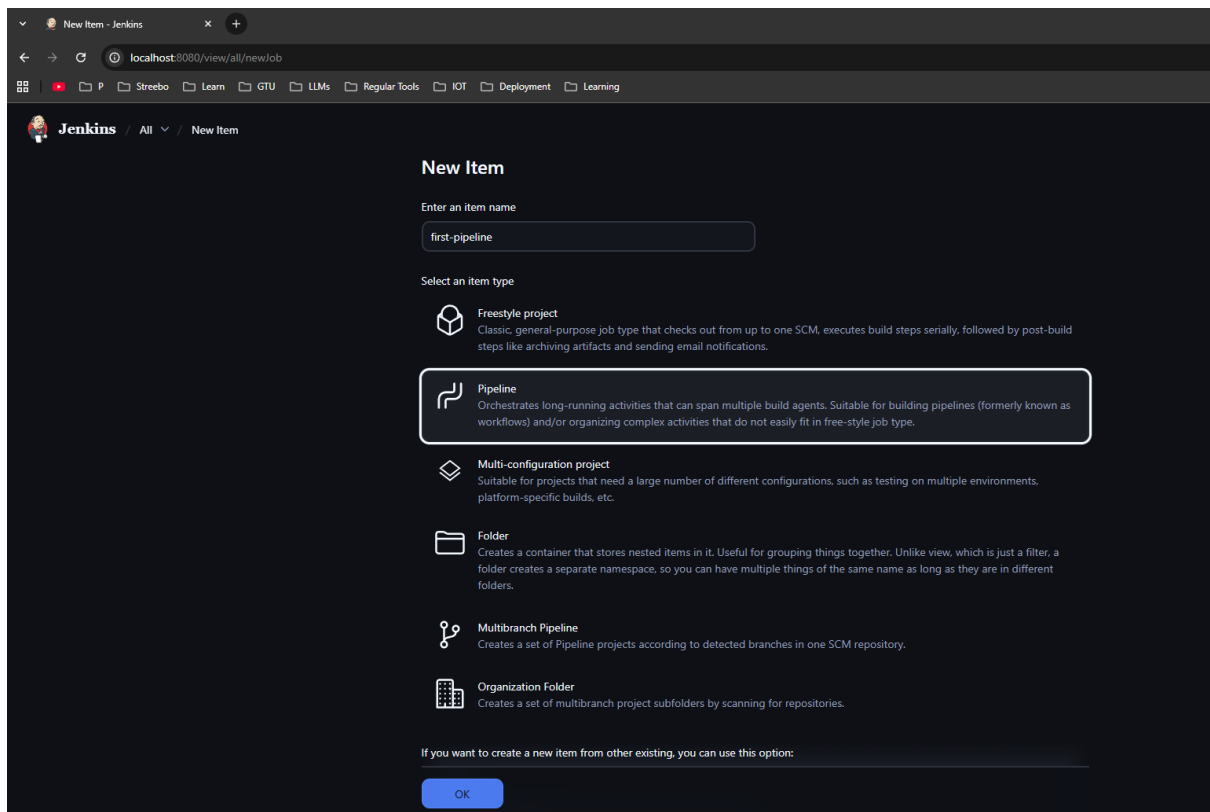


Pipeline Job



A screenshot of the Jenkins Dashboard. The browser tab is 'Dashboard - Jenkins' and the address bar shows 'localhost:8080'. The dashboard includes a 'New Item' button, a 'Build History' link, and a 'Build Queue' section stating 'No builds in the queue.' Below this is a 'Build Executor Status' section showing '0/2' executors. A table displays build history with columns 'S', 'W', 'Name', and 'Last Success'. One build is listed: 'first-job' with a green checkmark icon and a duration of '20 hr'. At the bottom, there are icons for 'S', 'M', and 'L'.

S	W	Name ↓	Last Success
✓	☀	first-job	20 hr



A screenshot of the Jenkins 'New Item' form. The browser tab is 'New Item - Jenkins' and the address bar shows 'localhost:8080/view/all/newJob'. The form has a breadcrumb 'Jenkins / All / New Item'. It includes a text input for 'Enter an item name' with the value 'first-pipeline'. Below is a section 'Select an item type' with several options: 'Freestyle project', 'Pipeline' (highlighted with a red border), 'Multi-configuration project', 'Folder', 'Multibranch Pipeline', and 'Organization Folder'. Each option has a brief description. At the bottom, there is a note: 'If you want to create a new item from other existing, you can use this option:' followed by an 'OK' button.

New Item

Enter an item name

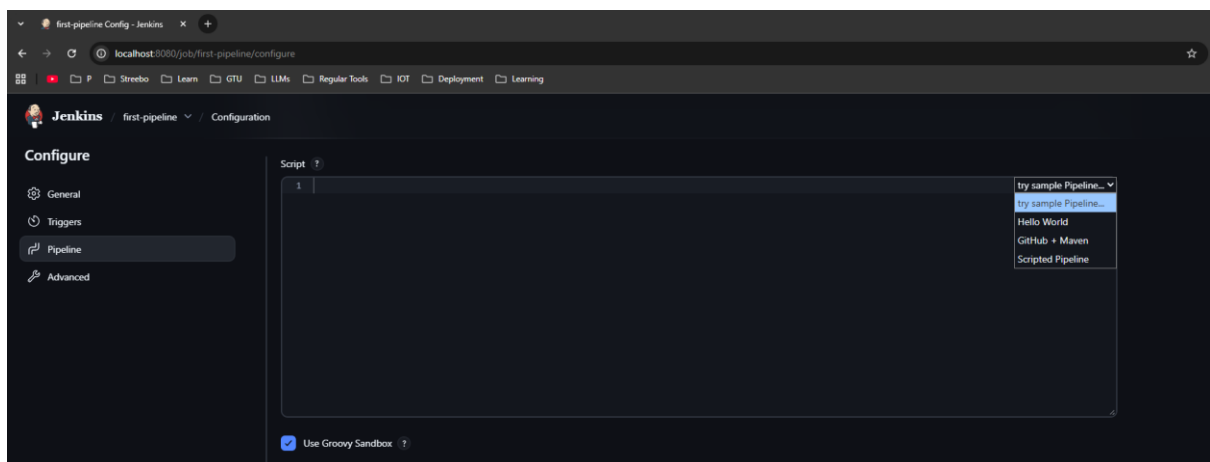
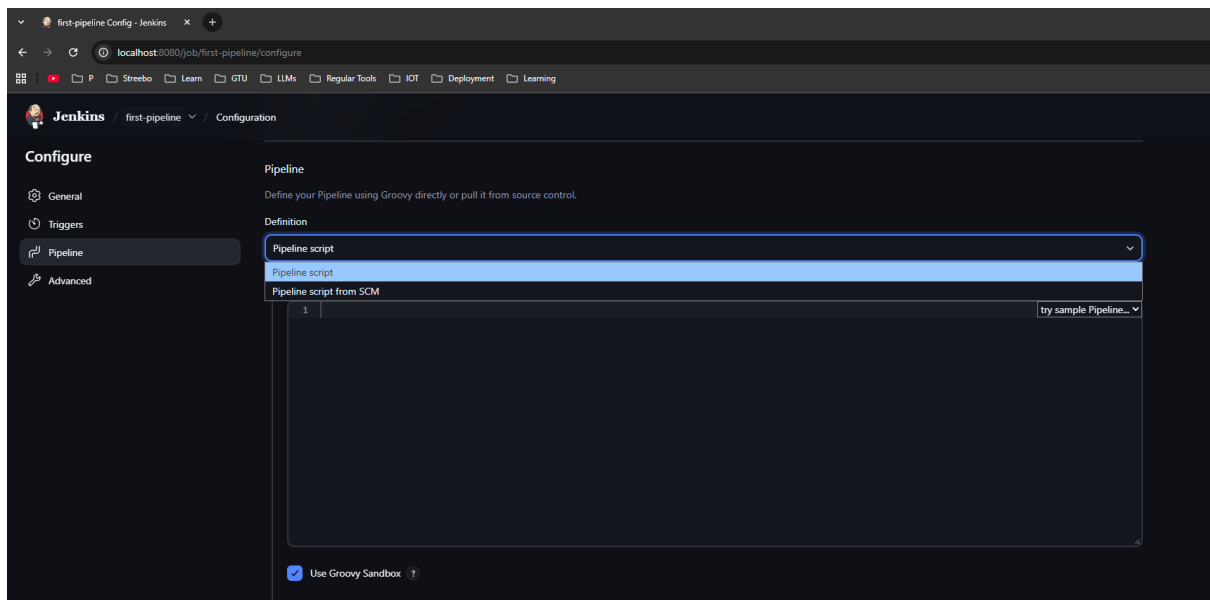
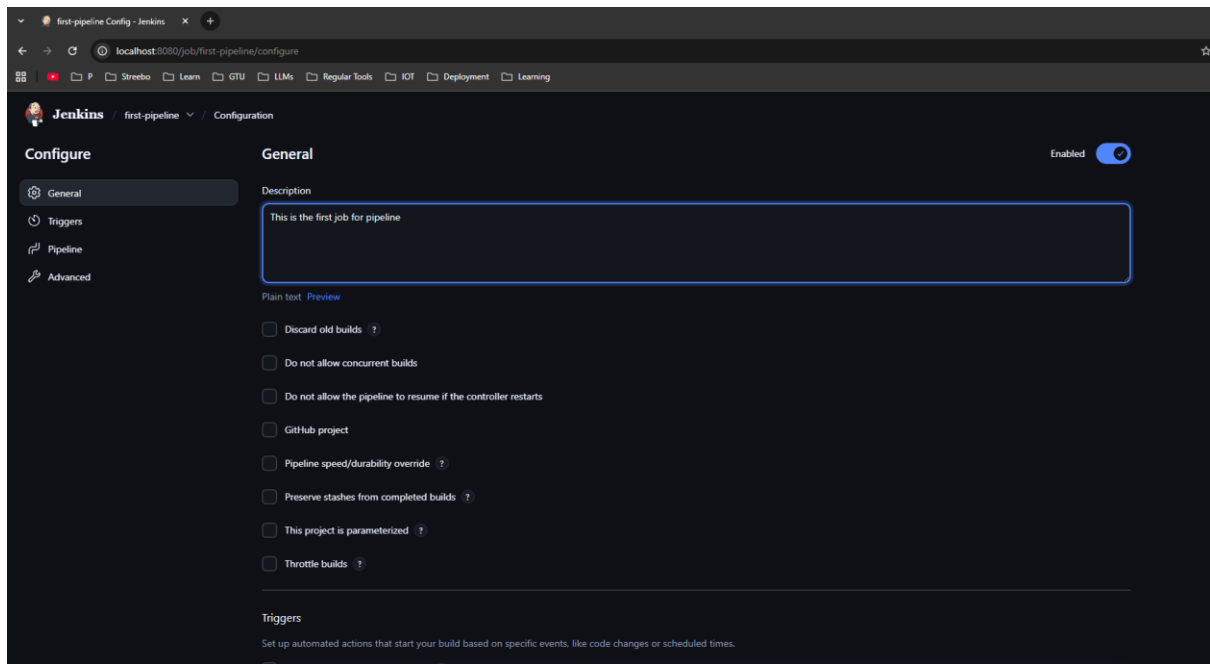
first-pipeline

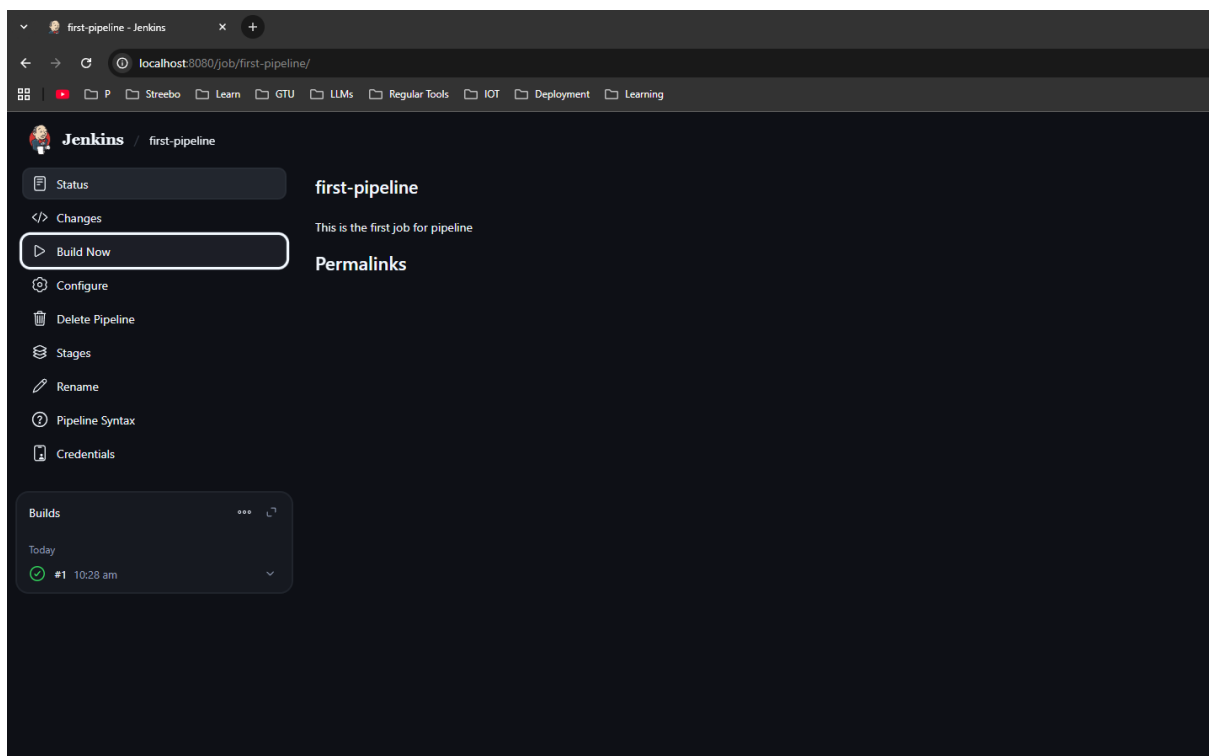
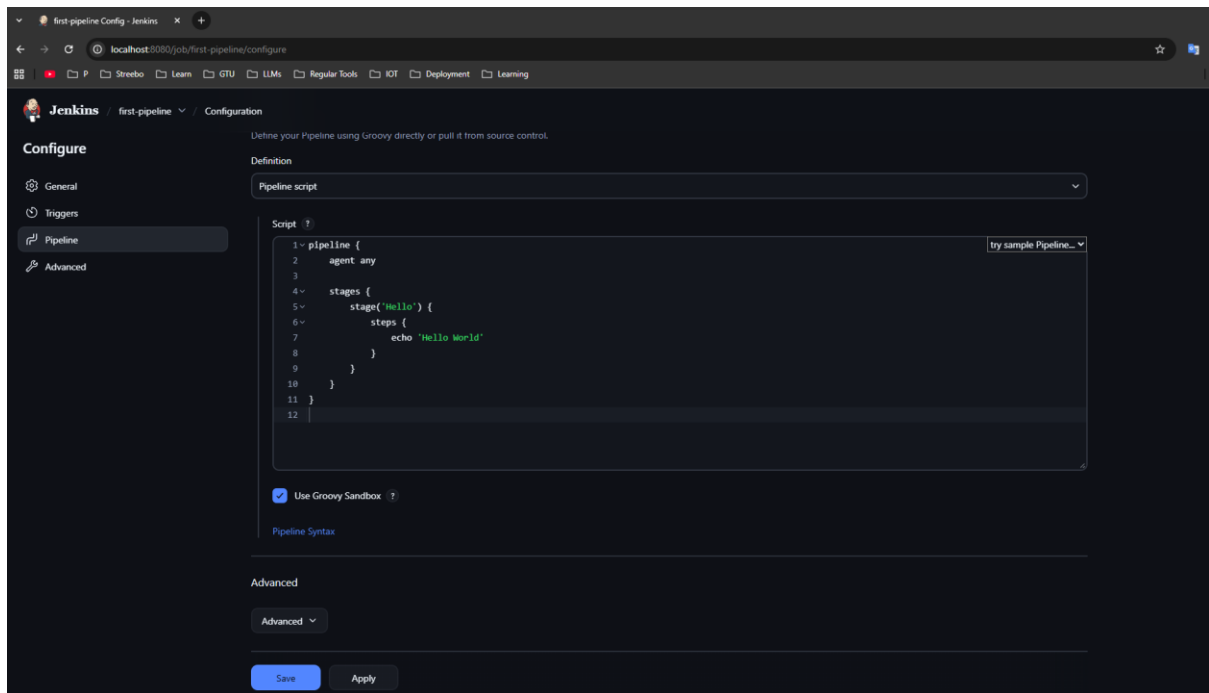
Select an item type

- 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.
- 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.
- Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

OK





first-pipeline #1 Console - Jenkins

localhost:8080/job/first-pipeline/1/console

Streebo Learn GTU LLMs Regular Tools IOT Deployment Learning

Jenkins / first-pipeline / #1 / Console Output

Status

Changes

Console Output

Edit Build Information

Delete build '#1'

Timings

Pipeline Overview

Restart from Stage

Replay

Pipeline Steps

Workspaces

Console Output

Started by user unknown or anonymous

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Jenkins in C:\ProgramData\Jenkins\workspace\first-pipeline

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Hello)

[Pipeline] echo

Hello World

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS

first-pipeline Config - Jenkins

localhost:8080/job/first-pipeline/configure

Streebo Learn GTU LLMs Regular Tools IOT Deployment Learning

Jenkins / first-pipeline / Configuration

Configure

General

Triggers

Pipeline

Advanced

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Build') {
6       steps {
7         echo 'Building Project...'
8       }
9     }
10  }
11  stages {
12    stage('Test') {
13      steps {
14        echo 'Running Tests...'
15      }
16    }
17  }
18 }
```

☒ Use Groovy Sandbox ?

Pipeline Syntax

Advanced

Advanced

Save Apply

first-pipeline Config - Jenkins

localhost:8080/job/first-pipeline/configure

Jenkins / first-pipeline / Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

```
12 stage('Test') {
13     steps {
14         echo 'Running Tests...'
15     }
16 }
17
18 stages {
19     stage('Deploy') {
20         steps {
21             echo 'Deploying Application...'
22         }
23     }
24 }
25 }
26
```

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

Advanced

Advanced

Save Apply

first-pipeline #2 Console - Jenkins

localhost:8080/job/first-pipeline/2/console

Jenkins / first-pipeline / #2 / Console Output

Console Output

Download

```
Started by user unknown or anonymous
org.codehaus.groovy.control.MultipleCompilationErrorsException: startup failed:
WorkflowScript: 11: Multiple occurrences of the stages section @ line 11, column 5.
    stages {
    ^

WorkflowScript: 18: Multiple occurrences of the stages section @ line 18, column 5.
    stages {
    ^

2 errors

at org.codehaus.groovy.control.ErrorCollector.failIfErrors(ErrorCollector.java:389)
at org.codehaus.groovy.control.CompilationUnit.applyToPrimaryClassNodes(CompilationUnit.java:1107)
at org.codehaus.groovy.control.CompilationUnit.doPhaseOperation(CompilationUnit.java:624)
at org.codehaus.groovy.control.CompilationUnit.processPhaseOperations(CompilationUnit.java:602)
at org.codehaus.groovy.control.CompilationUnit.compile(CompilationUnit.java:579)
at groovy.lang.GroovyClassLoader.doParseClass(GroovyClassLoader.java:323)
at groovy.lang.GroovyClassLoader.parseClass(GroovyClassLoader.java:293)
at PluginClassLoader for script-security//org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.GroovySandbox$Scope.parse(GroovySandbox.java:162)
at PluginClassLoader for workflow-cps//org.jenkinsci.plugins.workflow.cps.CpsGroovyShell.doParse(CpsGroovyShell.java:202)
at PluginClassLoader for workflow-cps//org.jenkinsci.plugins.workflow.cps.CpsGroovyShell.reparse(CpsGroovyShell.java:186)
at PluginClassLoader for workflow-cps//org.jenkinsci.plugins.workflow.cps.CpsFlowExecution.parseScript(CpsFlowExecution.java:669)
at PluginClassLoader for workflow-cps//org.jenkinsci.plugins.workflow.cps.CpsFlowExecution.start(CpsFlowExecution.java:615)
at PluginClassLoader for workflow-job//org.jenkinsci.plugins.workflow.job.WorkflowRun.run(WorkflowRun.java:341)
at hudson.model.ResourceController.execute(ResourceController.java:101)
at hudson.model.Executor.run(Executor.java:460)

Finished: FAILURE
```

first-pipeline Config - Jenkins

localhost:8080/job/first-pipeline/configure

Jenkins / first-pipeline / Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Build') {
6       steps {
7         echo 'Building Project...'
8       }
9     }
10    stage('Test') {
11      steps {
12        echo 'Running Tests...'
13      }
14    }
15    stage('Deploy') {
```

try sample Pipeline_

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Advanced

Advanced

Save Apply

first-pipeline Config - Jenkins

localhost:8080/job/first-pipeline/configure

Jenkins / first-pipeline / Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

Script ?

```
8       }
9     }
10    stage('Test') {
11      steps {
12        echo 'Running Tests...'
13      }
14    }
15    stage('Deploy') {
16      steps {
17        echo 'Deploying Application...'
18      }
19    }
20  }
21 }
22
```

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Advanced

Advanced

Save Apply

first-pipeline #3 Console - Jeni

localhost:8080/job/first-pipeline/3/console

StreeboLearnGTU LLMsRegular ToolsIOTDeploymentLearning

Jenkins

first-pipeline / #3 / Console Output

Status

Changes

Console Output

Edit Build Information

Delete build '#3'

Timings

Pipeline Overview

Restart from Stage

Replay

Pipeline Steps

Workspaces

Previous Build

Console Output

Started by user unknown or anonymous

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\first-pipeline

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Build)

[Pipeline] echo

Building Project...

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Test)

[Pipeline] echo

Running Tests...

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Deploy)

[Pipeline] echo

Deploying Application...

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS