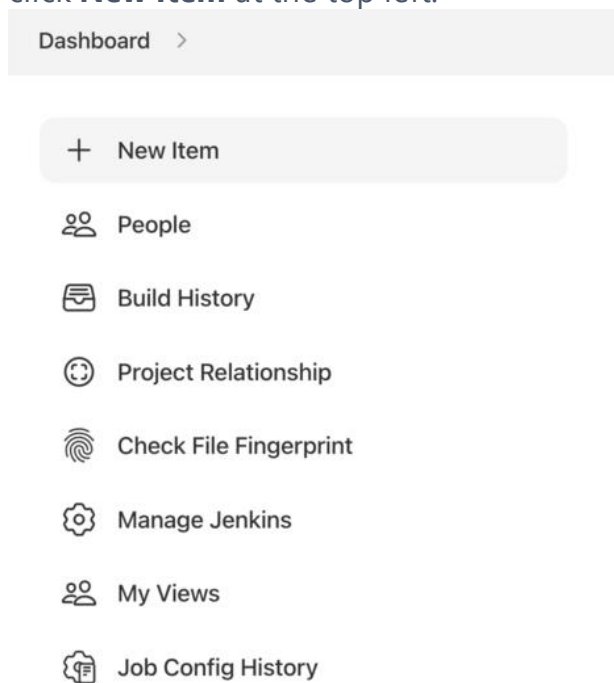


A Jenkinsfile created using the classic UI is stored by Jenkins itself (within the Jenkins home directory).

To create a basic Pipeline through the Jenkins classic UI:

1. If required, ensure you are logged in to Jenkins.
2. From the Jenkins home page (i.e. the Dashboard of the Jenkins classic UI), click **New Item** at the top left.



3. In the **Enter an item name** field, specify the name for your new Pipeline project.  
**Caution:** Jenkins uses this item name to create directories on disk. It is recommended to avoid using spaces in item names, since doing so may uncover bugs in scripts that do not properly handle spaces in directory paths.
4. Scroll down and click **Pipeline**, then click **OK** at the end of the page to open the Pipeline configuration page (whose **General** tab is selected).

## Enter an item name

» *Required field*



### Freestyle project

This is the central feature of Jenkins. Jenkins will build your project system, and this can be even used for something other than software.



### Pipeline

Orchestrates long-running activities that can span multiple builds (formerly known as workflows) and/or organizing complex activities as a job type.



### Multi-configuration project

Suitable for projects that need a large number of different configurations, environments, platform-specific builds, etc.

5. Click the **Pipeline** tab at the top of the page to scroll down to the **Pipeline** section.  
**Note:** If instead you are defining your Jenkinsfile in source control, follow the instructions in [In SCM](#) below.
6. In the **Pipeline** section, ensure that the **Definition** field indicates the **Pipeline script** option.
7. Enter your Pipeline code into the **Script** text area.  
For instance, copy the following Declarative example Pipeline code (below the *Jenkinsfile* ( ... ) heading) or its Scripted version equivalent and paste this into the **Script** text area. (The Declarative example below is used throughout the remainder of this procedure.)

## Jenkinsfile (Declarative Pipeline)

```
pipeline {  
  agent any  
  stages {  
    stage('Stage 1') {  
      steps {  
        echo 'Hello world!'  
      }  
    }  
  }  
}
```

## [Toggle Scripted Pipeline](#) (Advanced)

	agent instructs Jenkins to allocate an executor (on any available agent/node in the Jenkins environment) and workspace for the entire Pipeline.
	echo writes simple string in the console output.
	node effectively does the same as agent (above).

GeneralBuild TriggersAdvanced Project OptionsPipeline

Definition

Pipeline script

Script ?

```
1 pipeline {  
2     agent any  
3     stages {  
4         stage('Stage 1') {  
5             steps {  
6                 echo 'Hello world!'  
7             }  
8         }  
9     }  
10 }
```

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

SaveApply

**Note:** You can also select from canned *Scripted* Pipeline examples from the **try sample Pipeline** option at the top right of the **Script** text area. Be aware that there are no canned Declarative Pipeline examples available from this field.

- Click **Save** to open the Pipeline project/item view page.
- On this page, click **Build Now** on the left to run the Pipeline.



↑ Back to Dashboard

🔍 Status

📝 Changes

▶ Build Now

⚙️ Configure

🗑️ Delete Pipeline

🔍 Full Stage View

🌊 Open Blue Ocean

✎️ Rename

❓ Pipeline Syntax

10. Under **Build History** on the left, click **#1** to access the details for this particular Pipeline run.
11. Click **Console Output** to see the full output from the Pipeline run. The following output shows a successful run of your Pipeline.



## Console Output

```
Started by user Mmesoma Ikechukwu
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /Users/macbookair/.jenkins/workspace/example-pi
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Stage 1)
[Pipeline] echo
Hello world!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```