

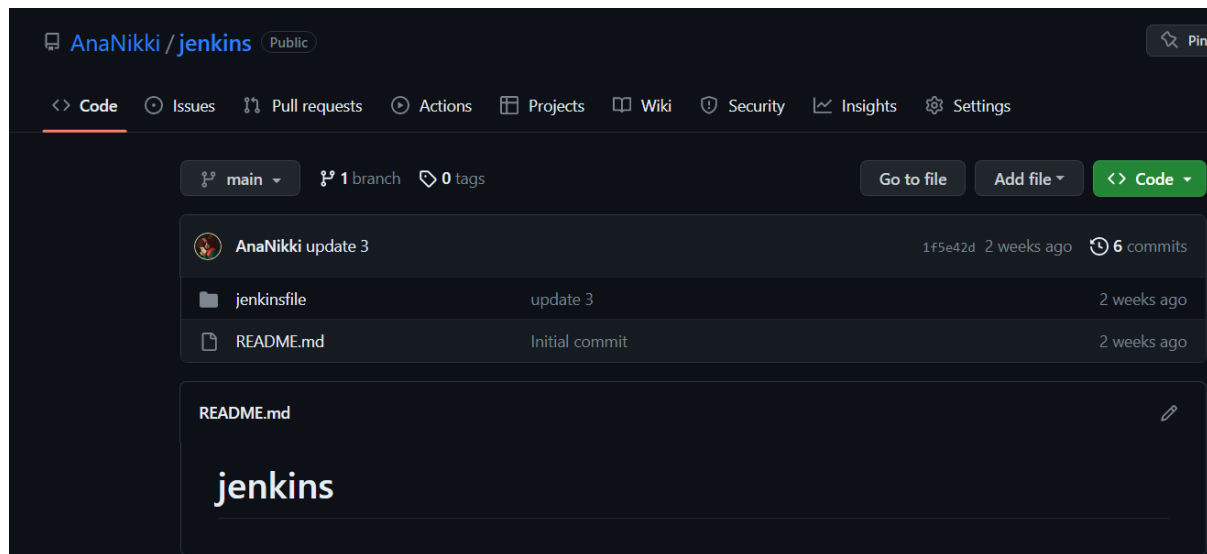
Jenkins

For this exercise, we have Jenkins pipelines. We will see here how to write a basic Jenkins “Hello world” pipeline. But first let’s see what Jenkins pipeline is. Jenkins is basically a sequence of steps that are linked together. There are two ways by which we can define a pipeline:

- Scripted syntax
- Declarative syntax

But most of the time we have our code in GitHub and we need CI/CD tool to pull the code from the version control and perform CI/CD tasks. Here we will define the pipeline in a Jenkins file and upload that file into GitHub. We will see how to write Jenkinsfile and run the Jenkins by specifying the Jenkinsfile .

For this purpose, we will create a GitHub repository for this Jenkins exercise. In the repository we will make a directory named jenkinsfile where we will put our hello_world.jenkinsfile.



This is how our Jenkinsfile looks like:

```
jenkins / jenkinsfile / hello_world.jenkinsfile

Code Blame 20 lines (20 loc) · 359 Bytes

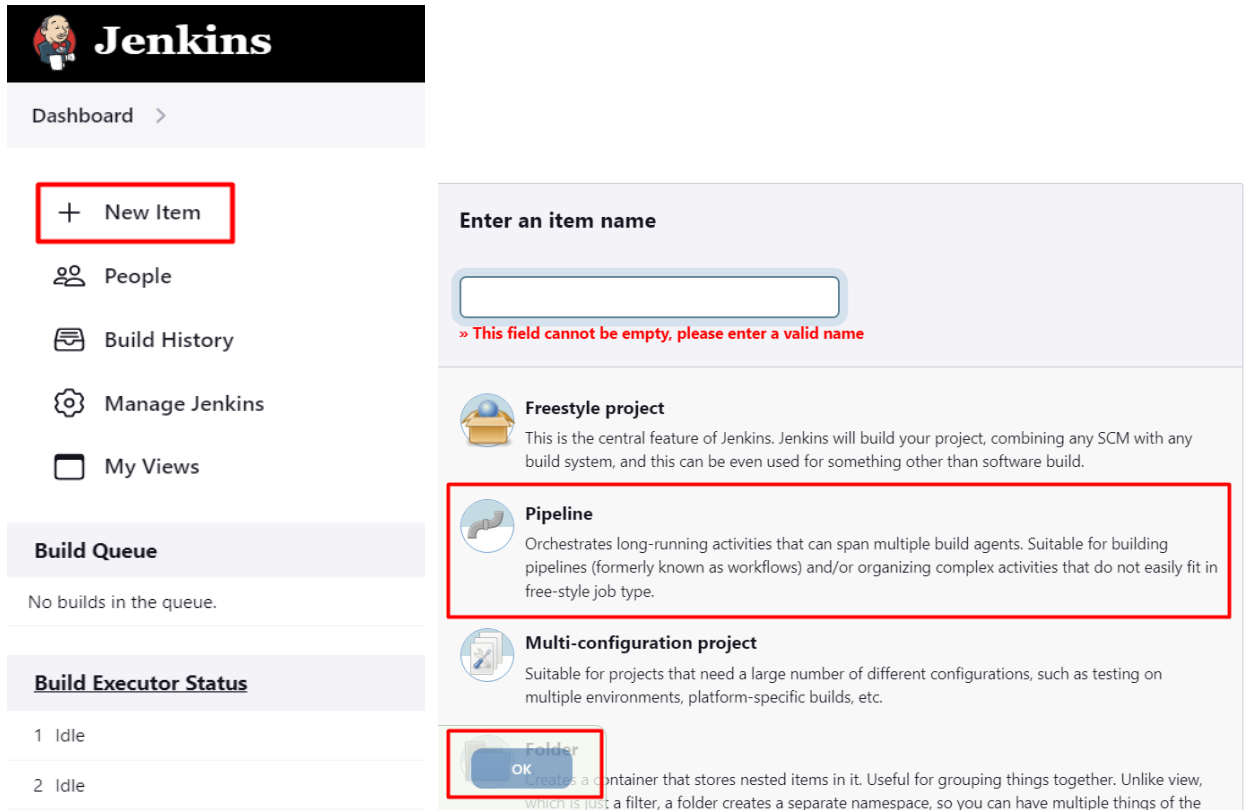
1  pipeline {
2      agent any
3      stages {
4          stage('Hello') {
5              steps {
6                  echo 'Hello World'
7              }
8          }
9          stage('Academy') {
10             steps {
11                 echo 'DevOps Academy'
12             }
13         }
14         stage('Bye') {
15             steps {
16                 echo 'Bye World'
17             }
18         }
19     }
20 }
```

In this code we have:

- pipeline – this is the mandatory block.
- Agent – this is where the Jenkins build job should run. Here we have selected agents as any. Jenkins supports a wide variety of agents.
- Stages/stage – this block consists of different executable stage blocks.
- Steps – this block consists of an actual operation that needs to be performed inside Jenkins.

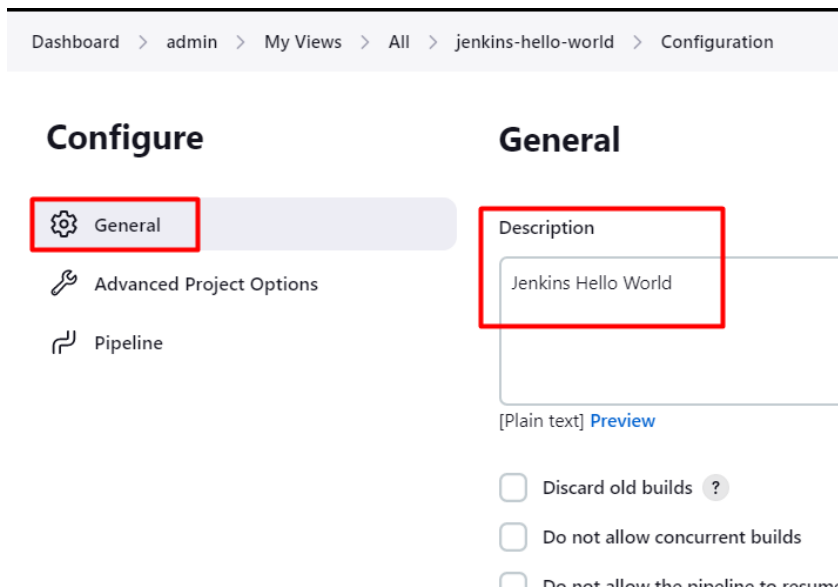
In our example we print “Hello Word”, “DevOps Academy”, “Bye World”

Next, we will create a Jenkins pipeline. On the Jenkins dashboard we click on New Item. When we click on New Item, we need to Enter an Item name, select Pipeline and click ok.



The screenshot shows the Jenkins Dashboard. On the left, there is a sidebar with a 'New Item' button highlighted by a red box. Below it are links for 'People', 'Build History', 'Manage Jenkins', and 'My Views'. The main area is divided into two sections. The top section, 'Build Queue', shows 'No builds in the queue.' The bottom section, 'Build Executor Status', shows two executors in an 'Idle' state. On the right, there is a 'Create New Item' form. The 'Enter an item name' field is empty, with a red error message below it: '» This field cannot be empty, please enter a valid name'. Below the name field, there are three options: 'Freestyle project', 'Pipeline' (highlighted by a red box), and 'Multi-configuration project'. At the bottom of the form, there is a 'Folder' button highlighted by a red box, with a tooltip that says 'OK' and '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'.

Next on the General tab we add a short description.



The screenshot shows the Jenkins Configuration page for a new item. The breadcrumb trail at the top reads: 'Dashboard > admin > My Views > All > jenkins-hello-world > Configuration'. The 'Configure' section on the left has three tabs: 'General' (highlighted by a red box), 'Advanced Project Options', and 'Pipeline'. The 'General' tab is active, showing a 'Description' field with the text 'Jenkins Hello World' (highlighted by a red box). Below the description field, there is a '[Plain text] Preview' link. At the bottom, there are three checkboxes: 'Discard old builds' (checked), 'Do not allow concurrent builds' (unchecked), and 'Do not allow the pipeline to resume' (unchecked).

After that we go to Pipeline tab and then select definition as Pipeline script from SCM. Under SCM we select Git and pass the GitHub repository URL and branch name. Next we add the script path from our GitHub and click on Save to save the pipeline.

Configure

- General
- Advanced Project Options
- Pipeline**

Pipeline

Definition
Pipeline script from SCM

SCM
Git

Repositories

Repository URL
https://github.com/AnaNikki/jenkins

Credentials
- none -

Add +

Advanced

Add Repository

Save
Apply

Configure

- General
- Advanced Project Options
- Pipeline**

Add Repository

Branches to build

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

Add Branch

Repository browser
(Auto)

Additional Behaviours

Add +

Script Path
jenkinsfile/hello_world.jenkinsfile

☒ Lightweight checkout

Pipeline Syntax

Save
Apply

After we save the pipeline, we go to Dashboard and select our pipeline.

Dashboard

+ New Item
People
Build History
Manage Jenkins
My Views

All +

S	W	Name	Last Success	Last Failure	Last Duration
✓	☁	jenkins-hello-world	14 days #10	14 days #8	7 sec
✓	☀	Jenkins_demo	15 days #1	N/A	0.82 sec

Build Queue

No builds in the queue.

Add description

Then we select Build Now.

Pipeline jenkins-

Jenkins Hello World

Stage View

And with that we build our hello-world pipeline. As we can see from the picture below, we had many unsuccessful builds and successful ones. For our first successful build we clicked on it. On the new window we click Console Output and see the output from the build.

Build History

#	Time	Status	Changes
#11	May 11, 2023, 10:51 PM	No Changes	
#10	Apr 27, 2023, 7:05 PM	1 commit	
#9	Apr 27, 2023, 6:39 PM	3 commits	
#8	Apr 27, 2023, 6:34 PM	No Changes	
#7	Apr 27, 2023, 6:21 PM	No Changes	
#5	Apr 27, 2023, 5:46 PM	No Changes	
#5	Apr 27, 2023, 12:23 AM	No Changes	
#4	Apr 27, 2023, 12:23 AM	No Changes	
#3	Apr 27, 2023, 12:21 AM	No Changes	
#2	Apr 27, 2023, 12:16 AM	No Changes	
#1	Apr 27, 2023, 12:13 AM	No Changes	

Console Output

```
Started by user admin
Obtained Jenkinsfile from git https://github.com/andriiki/jenkins
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\workspace\jenkins-hello-world
[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/andriiki/jenkins
> git.exe init C:\ProgramData\Jenkins\workspace\jenkins-hello-world # timeout=10
Fetching upstream changes from https://github.com/andriiki/jenkins
> git.exe --version # timeout=10
> git --version # "git version 2.39.2.windows.1"
> git.exe fetch --tags --force --progress -- https://github.com/andriiki/jenkins -refs/heads/*refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/andriiki/jenkins # 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 4832eb8c475f320b847849eac225c375aa2 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 4832eb8c475f320b847849eac225c375aa2 # timeout=10
Commit message: "Create new hello_world.jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
```

We can go to Status and see the Stage View.

Pipeline jenkins-hello-world

Jenkins Hello World

Stage View

	Declarative: Checkout SCM	Hello	Academy	Bye
Average stage times: (Average full run time: ~7s)	2s	173ms	116ms	104ms
#10 Apr 27 19:05	2s	152ms	112ms	104ms
#9 Apr 27 18:39	2s	160ms	120ms	
#8 Apr 27 18:34	No Changes			
#7 Apr 27	No Changes			

