**Integrating the GitHub Repository with Jenkins**

**Prerequisites:**

Your Jenkins web console must be accessible over the internet.

A [Github](https://github.com/" \t "https://adamtheautomator.com/jenkins-ci-cd/_blank) account to which you’ll connect with the Jenkins CI CD pipeline.

**Preparing the GitHub Repository**

Create a blank project or fork an existing repository to get started.

[Log in to your GitHub account](https://github.com/login" \t "https://adamtheautomator.com/jenkins-ci-cd/_blank).

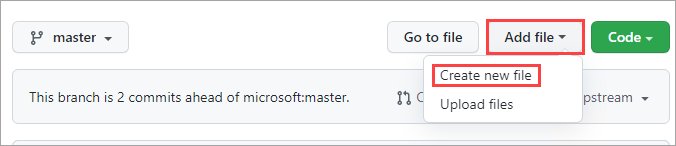
Next, navigate to the  [repository URL](https://github.com/microsoft/MyShuttle2" \t "https://adamtheautomator.com/jenkins-ci-cd/_blank) of your choice. Once you’re on the repo page, click the Fork button near the top-right section of the page.

IMG_256

**Forking a repo**

After the forking process finishes add a new file to the repository named [Jenkinsfile](https://www.jenkins.io/doc/book/pipeline/jenkinsfile/" \t "https://adamtheautomator.com/jenkins-ci-cd/_blank). A Jenkinsfile is a text file that contains the definition of a Jenkins Pipeline. This file should be inside the source code repository.

On your repository page, **click Add file —> Create new file.**

**Adding a new file**

Type in Jenkinsfile as the name of the new file. Next, copy the code below and paste it into your Jenkinsfile.

The term pipeline is a set of instructions written in code for continuous delivery.

A **stage** block contains a series of steps in a pipeline, visualizing the Jenkins pipeline process.

A **step** is a single task that executes a specific process at a defined time. A pipeline contains a series of steps.

This Jenkinsfile is only a basic example of a pipeline script. In production, pipeline scripts will be more complex and include multiple actions on one or all stages.

pipeline {

agent any

stages {

stage('Build') {

steps {

echo 'Building..'

}

}

stage('Test') {

steps {

echo 'Testing..'

}

}

stage('Deploy') {

steps {

echo 'Deploying....'

}

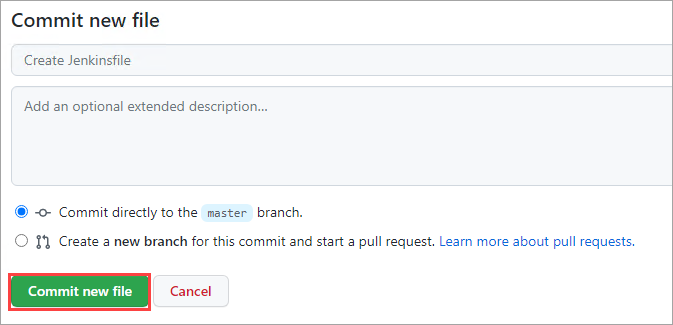
}

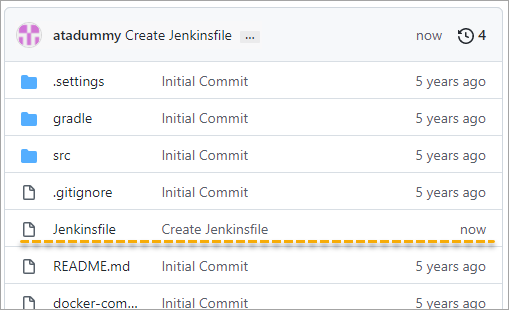
}}

After writing the code, your file will look like the screenshot below.



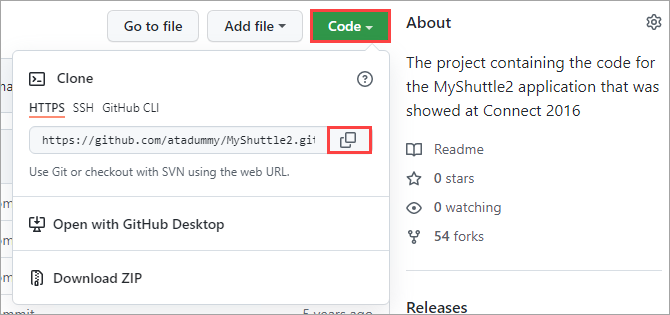
5. Scroll to the bottom of the page and click Commit new file.

Commit the new file



New Jenkinsfile on the GitHub repo

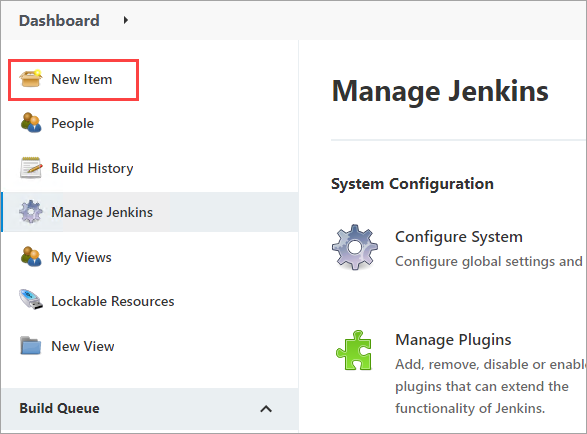
Finally, copy the repository URL. To do so, click Code and click the Copy button. You will need the resulting URL during the Jenkins CI CD pipeline creation.



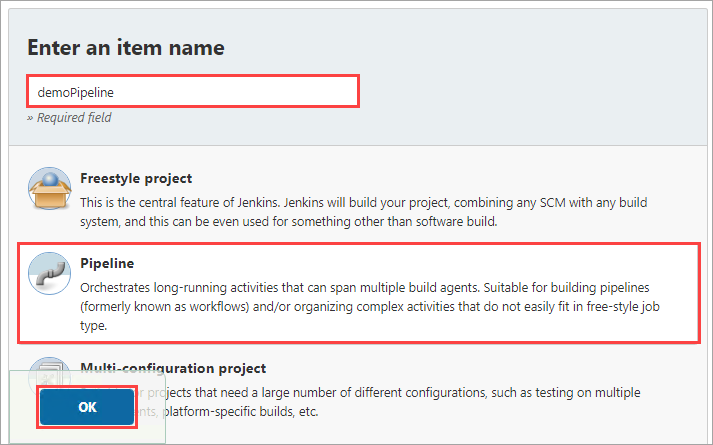
**Creating a Jenkins CI CD Pipeline**

1. Open a browser, navigate to your Jenkins URL, and log in.

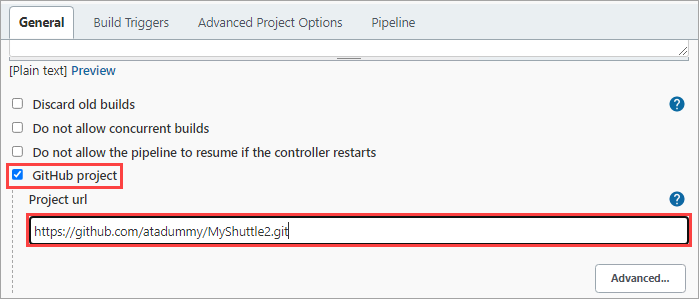
2. On the Jenkins**Dashboard**, click the **New Item** option on the left-hand side menu.



3. Enter the name of the new pipeline. Name it demoPipeline and select the **Pipeline template**. Click OK to finish.

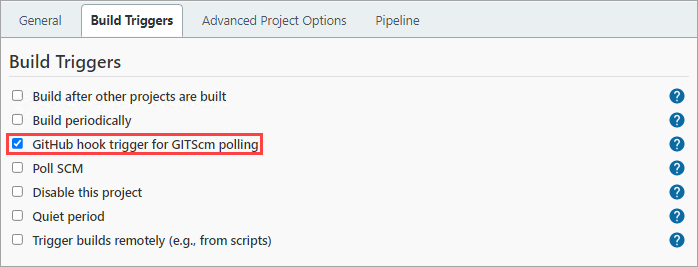


4. Check the GitHub project box under the **General**tab on the configuration page. Next, paste the repository URL you copied earlier into the **Project** **URL** box.



Specifying the GitHub repository URL

1. Scroll down to the ****Build Triggers**** section and check the ****GitHub hook trigger for GITScm polling**** box, like the screenshot below.

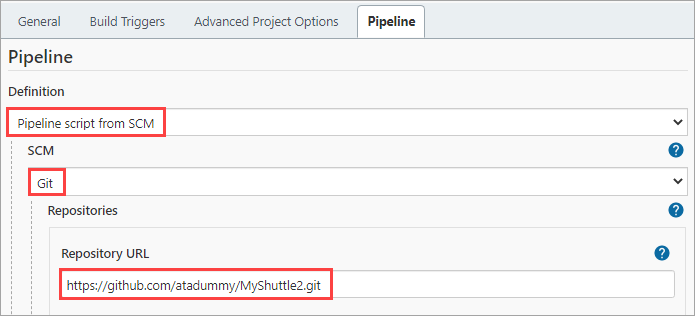
Specifying the build trigger

6. Next, scroll further down to the ****Pipeline**** section and select/specify the following values:

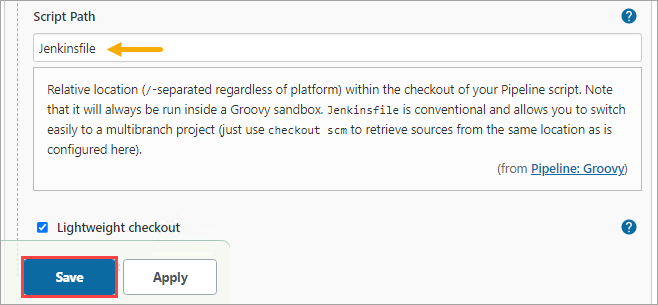
****Definition****: Pipeline script from SCM

****SCM****: Git

****Repository URL****: Your repository URL

Specifying the pipeline values

7. Scroll to the bottom of the page and confirm that the ****Script Path**** value is ****Jenkinsfile****. This value references the Jenkinsfile you created earlier while preparing your Github repo. Finally, click ****Save****.

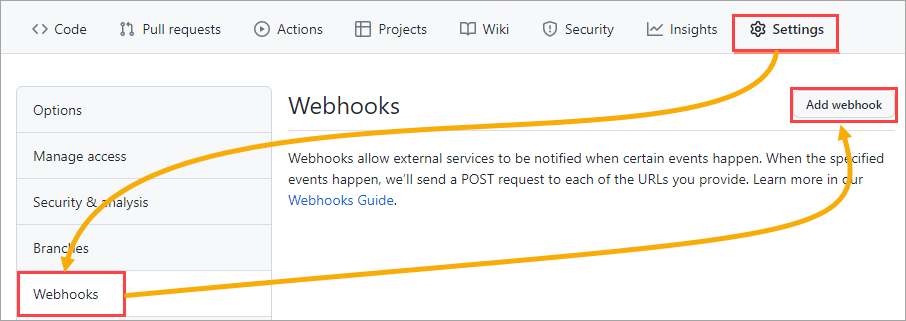
Saving the pipeline configuration

## ****Configuring a Webhook in GitHub****

For Jenkins to run a new job, you must first create a webhook in the GitHub repository. This webhook will notify Jenkins as soon as a new push occurs in the repository.

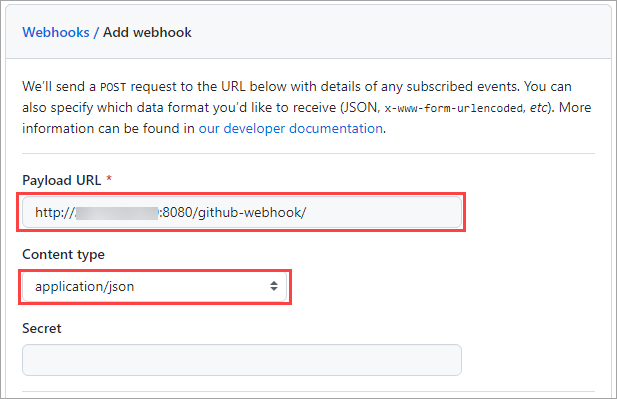
Follow the steps below to create a Webhook.

1. Navigate to the ****Settings**** page and click the ****Webhooks**** tab on your GitHub repository. On the ****Webhooks**** page, click ****Add webhook****.

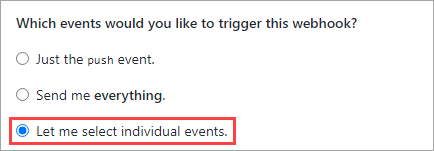
Adding a webhook

2. Specify your Jenkins URL and append /github-webhook/in the ****Payload URL**** field. For example, [HTTP://jenkinsURLhere/github-webhook/](http://jenkinsurlhere/github-webhook/*" \t "https://adamtheautomator.com/jenkins-ci-cd/_blank).

Also, change the ****Content type**** value to application/json.

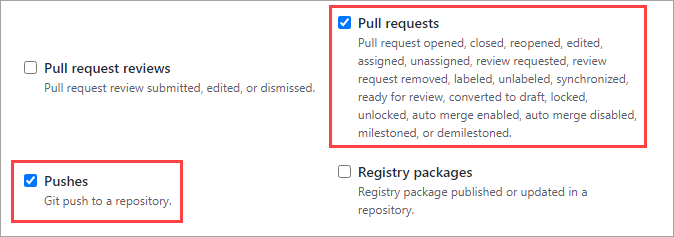
Setting the webhook URL and content type

3. On ****Which events would you like to trigger this webhook**** selection, click the ****Let me select individual events**** option.



Selecting individual events

4. Scroll down and check the ****Pushes**** and ****Pull requests**** checkboxes. These options will ensure that GitHub will send a payload to Jenkins during these specific events.

Enabling Pushes and Pull requests events

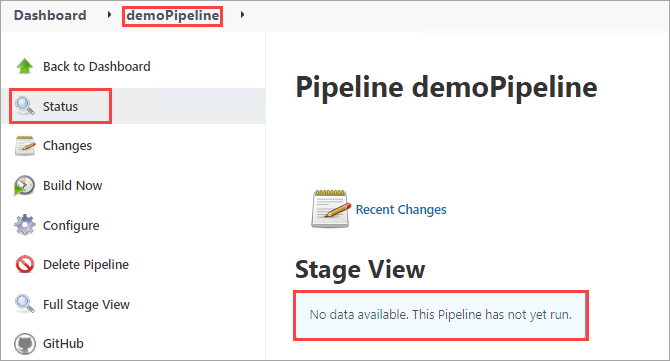
1. Validate the webhook by clicking on the ****Add webhook**** button at the bottom. If the webhook validation went well, you’d see a notification on the top of the page similar to the screenshot below.

IMG_272Webhook creation notification

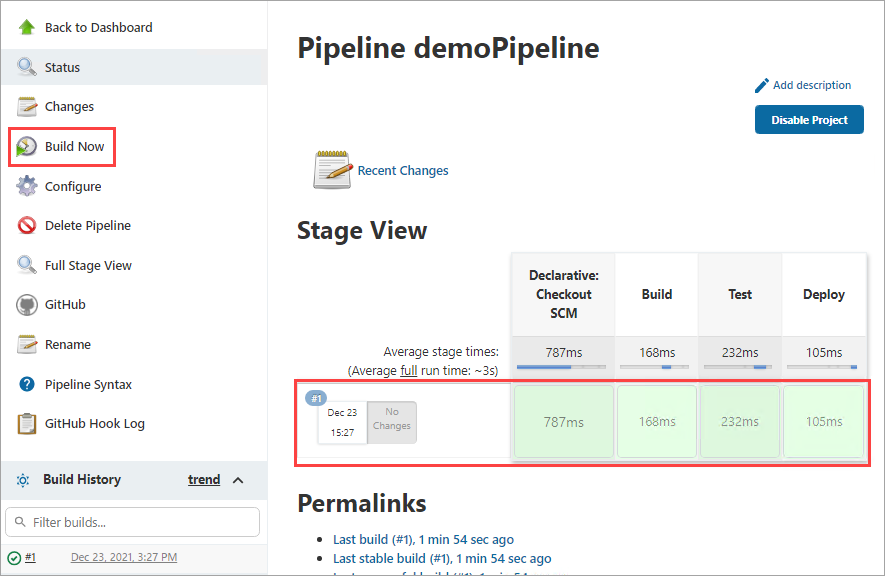
## Executing and Triggering the Jenkins CI CD Pipeline Job

Jenkins provides a ****Pipeline Stage View**** plugin to help you view the pipeline status.

1. To open the pipeline status, open the pipeline in Jenkins and click on ****Status****. As you can see below, the ****Stage View**** status shows ****No data available**** because there hasn’t been an event that would trigger the pipeline.

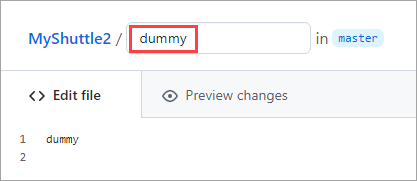
Jenkins pipeline stage view

2. To generate initial build data, click ****Build Now**** and wait for the build process to complete. As you can see below, the Stage View now has one entry, saying that there are no changes in this build.

To generate initial build data, click ****Build Now****

3. Now, test the pipeline by committing a new dummy file to the repository. To do so, go back to your GitHub repository and click click ****Add File —> Create new file****.

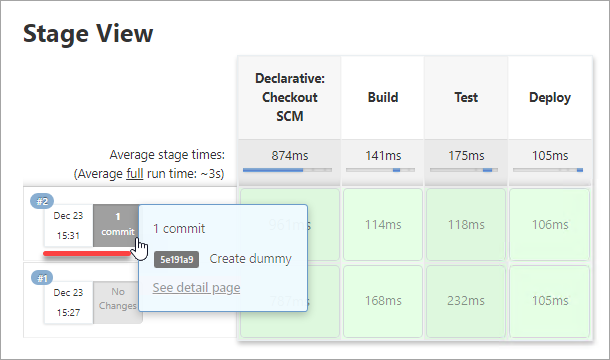
Enter ****dummy**** as the new file’s name and make sure to add content.



Creating a new file

Finally, click ****Commit new file**** at the bottom of the page.

1. Go back to your Jenkins pipeline status page, and you should see a new build entry with one commit, as shown below.

Jenkins pipeline build status