

# Single Pipeline Git Integration

---

This walkthrough will help integrate a local git server into a Jenkins pipeline to carry out the automation portion of building, testing and deploying a product onto different machines. This will use the **pipeline** project type within Jenkins.

## Table of Contents

---

- [Single Pipeline Git Integration](#)
- [Table of Contents](#)
- [Gitea](#)
- [Jenkins](#)
  - [Pipeline project creation](#)
    - [Project option selection](#)
    - [Pipeline Options](#)
  - [Running a build](#)

## Gitea

---

Gitea will be the internal git server used in this walkthrough. The walkthrough is provided in the "containers" folder at the time of this document. You can search through our repo to find it.

## Jenkins

---


Log into an account that has access to create a new project.

### Pipeline project creation

On the Jenkins UI, after clicking **new item**, you will be able to select the **Pipeline** option. Create a meaningful name.

**Enter an item name**

*» Required field*

**Freestyle project**

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

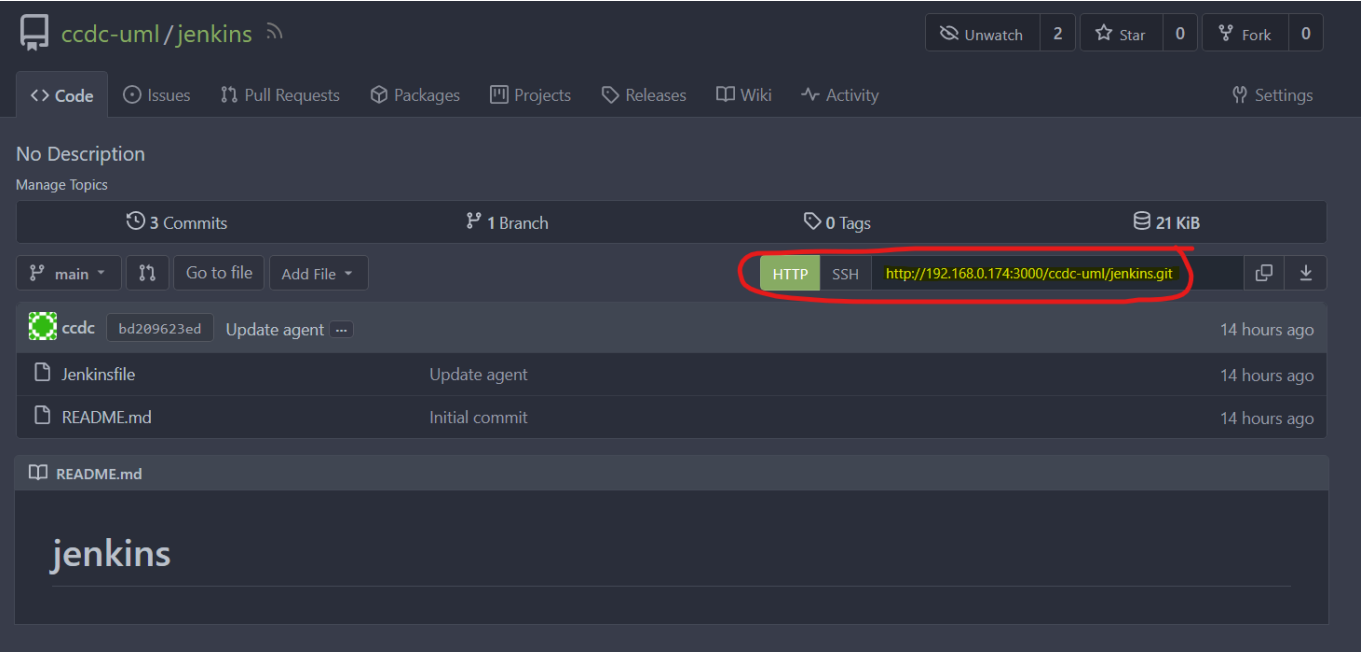
**Pipeline**

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.

Project option selection

- 1. GitHub project creation - Check the option and then put the URL to your internal git server. **Note: This is not a fully secure connection as there is no HTTPS.**

You can find the link to the repo within the GUI of the Gitea server.



Plug this value into the **Project URL** textbox.

### Configure

General

Advanced Project Options

Pipeline

### General

Description

[Plain text] [Preview](#)

☐ Discard old builds ?

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☒ GitHub project

Project url ?

http://192.168.0.174:3000/ccdc-uml/jenkins.git

Advanced...

☐ Pipeline speed/durability override ?

☐ Preserve stashes from completed builds ?

☐ This project is parameterized ?

## Pipeline Options

In the Pipeline section, you'll be able to see the option for **Pipeline script from SCM** under the **Definition** Header. By default, this will be **Pipeline Script**.

The screenshot shows the Jenkins 'Configure' page for a pipeline. On the left, there is a sidebar with three tabs: 'General', 'Advanced Project Options', and 'Pipeline'. The 'Pipeline' tab is selected. The main content area is titled 'Pipeline' and contains a 'Definition' section with a dropdown menu set to 'Pipeline script from SCM'. Below this, there is an 'SCM' section with a dropdown menu set to 'Git'. Under the 'SCM' section, there is a 'Repositories' section with a 'Repository URL' field containing 'http://192.168.0.174:3000/ccdc-uml/jenkins.git' and a 'Credentials' dropdown menu set to '- none -'. There is also an '+ Add' button at the bottom of the 'Repositories' section.

Once again, you will need to fill out the information for the Git repo and then define any credentials if necessary. **Note: Since this repo was made public for internal use, then there is no need for any type of credential.**

If you scroll down, you will reach a section that mentions **Branches to build**. If you create a completely new repo with nothing in it and no branches, then this will give you issues when you try to launch a build. To fix this, simply **blank out** the **Branch Specifier** (blank for 'any') field. Also, make sure the **uncheck** the **Lightweight Checkout** option.

## Configure

- General
- Advanced Project Options
- Pipeline**

Branches to build ?

Branch Specifier (blank for 'any') ?

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Script Path ?

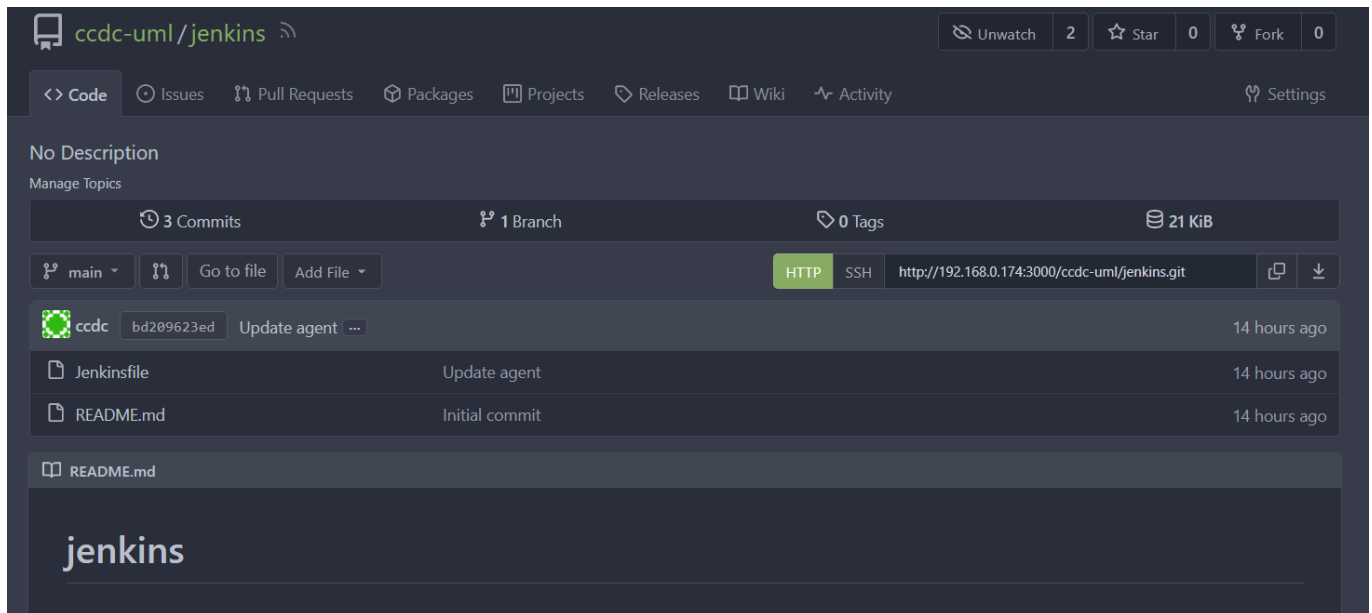
Jenkinsfile

☐ Lightweight checkout ?

[Pipeline Syntax](#)

Save Apply

Finally, you'll notice in the screenshot above that the Script Path is just named Jenkinsfile. This is not always the case as the jenkinsfile could be somewhere else in the repository.




In this demonstration, you'll notice in the repo that the file named "Jenkinsfile" is in the root (/) directory. (Marked in Green)

Save all your changes with **Save** and then you're ready to go if all the other configurations are done.

## Running a build


Once you press save, you'll be brought back to the newly created project's home page that you can press the **Build Now** button to kick off a build.


**Jenkins**


Dashboard > Injects > Gitea-Demonstration >


Status


</> Changes


 Build Now


 Configure


 Delete Pipeline


 Move


 Full Stage View

 GitHub

 Rename

 Pipeline Syntax

 **Build History** trend ▼

 Filter builds...

/

No builds

## Pipeline Gitea-Demonstration

Full project name: Injects/Gitea-Demonstration

## Stage View

No data available. This Pipeline has not yet run.

## Permalinks

If your pipeline is all set to handle all the steps defined in the Dockerfile, then you will see a successful run appear on the screen.

Dashboard > Injects > Gitea-Demonstration >

Status

</> Changes

▶ Build Now

⚙️ Configure

🗑️ Delete Pipeline

↕️ Move

🔍 Full Stage View

🌐 GitHub

✎ Rename

❓ Pipeline Syntax

🌞 Build History trend ▼

🔍 Filter builds... /

✔️ #1

Feb 9, 2023, 10:56 AM EDT

Pipeline Gitea-Demonstration

Full project name: Injects/Gitea-Demonstration

Stage View

Average stage times:  
(Average full run time: ~2s)

#1

Feb 09 10:56

No Changes

Declarative: Checkout SCM	Test - List contents - Agent 0
323ms	410ms
323ms	410ms

Permalinks

7 / 7