


# 1. Automating a Python Application Build using Jenkins.

## Step 1: Install Jenkins and Required Plugins

1. Start the Jenkins service and log in to the Jenkins dashboard.
2. Navigate to Manage Jenkins → Plugins.
3. Install the following plugins:
  - Pipeline Plugin (for defining Jenkins Pipelines)
  - Git Plugin (for pulling code from GitHub)
  - Build Tools Plugin (for executing builds)
4. Restart Jenkins to apply the changes.

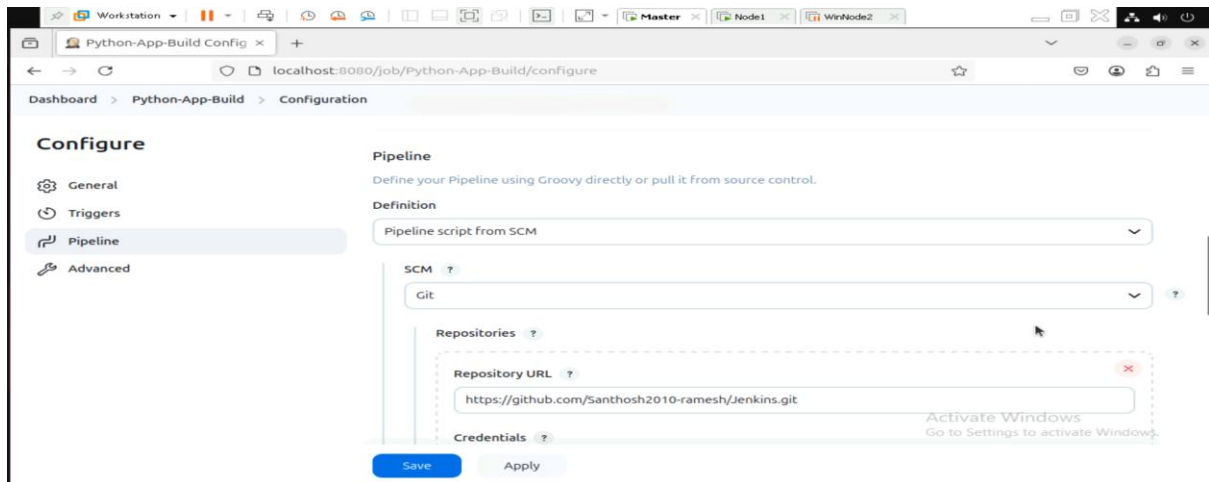
## Step 2: Create a New Jenkins Job

1. Open **Jenkins Dashboard**.
2. Click on **New Item** → Enter a project name.
3. Choose **Freestyle Project** or **Pipeline Project**.
4. Click **OK** to create the project.

S	W	Name ↓	Last Success	Last Failure	Last Duration		F
		My_First_Pipeline	18 hr #10	18 hr #9	26 sec		
		Python-App-Build	6 min 13 sec #1	N/A	8.9 sec		

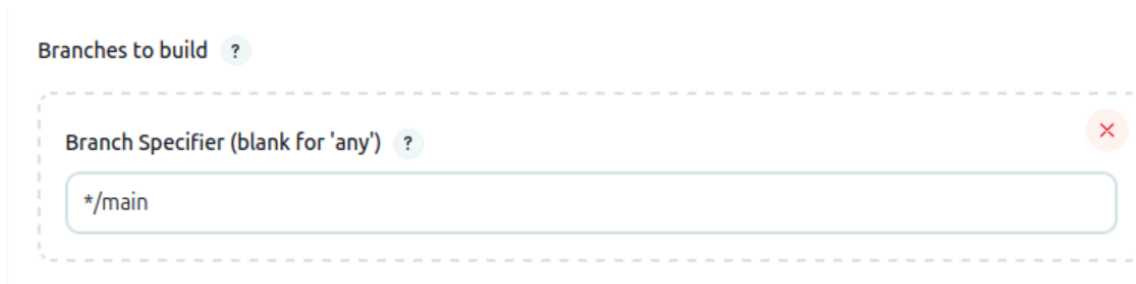
## Step 3: Configure Jenkins to Pull Code from GitHub

1. Open the newly created Jenkins job.
2. Navigate to **Source Code Management**.
3. Select **Git** and enter the repository URL:
  - Example: `https://github.com/your-repo/python-app.git`
4. Configure GitHub credentials if authentication is required.



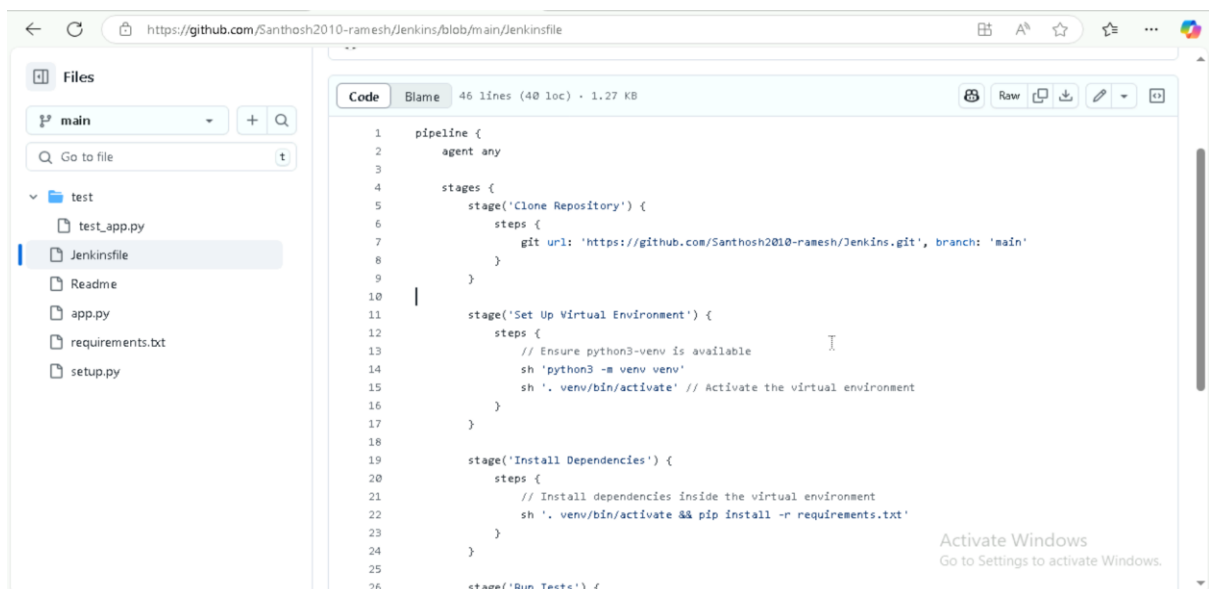
## Step 4: Install Dependencies and Run Tests

Step 1: Adding the desired branch here we use Main branch.



## Step 5: Pipeline with Python Scripts.

Step 1: Building the Declarative: Checkout SCM with the Script.



## Step 2: Sample Python file.

```
Code Blame 5 lines (4 loc) • 92 Bytes

1 def hello_world():
2     print("Hello, World!")
3
4 if __name__ == "__main__":
5     hello_world()
```

## Step 3: Creating Requirements.txt file.

Files

main

Go to file

test

test\_app.py

Jenkinsfile

Readme

app.py

requirements.txt

setup.py

Jenkins / requirements.txt

Santhosh2010-ramesh Update requirements.txt 90d1105 • 6 minutes ago History

Code Blame 3 lines (3 loc) • 27 Bytes

```
1 requests
2 pytest
3 setuptools
```

## Step 4: Creating a Setup.py file.

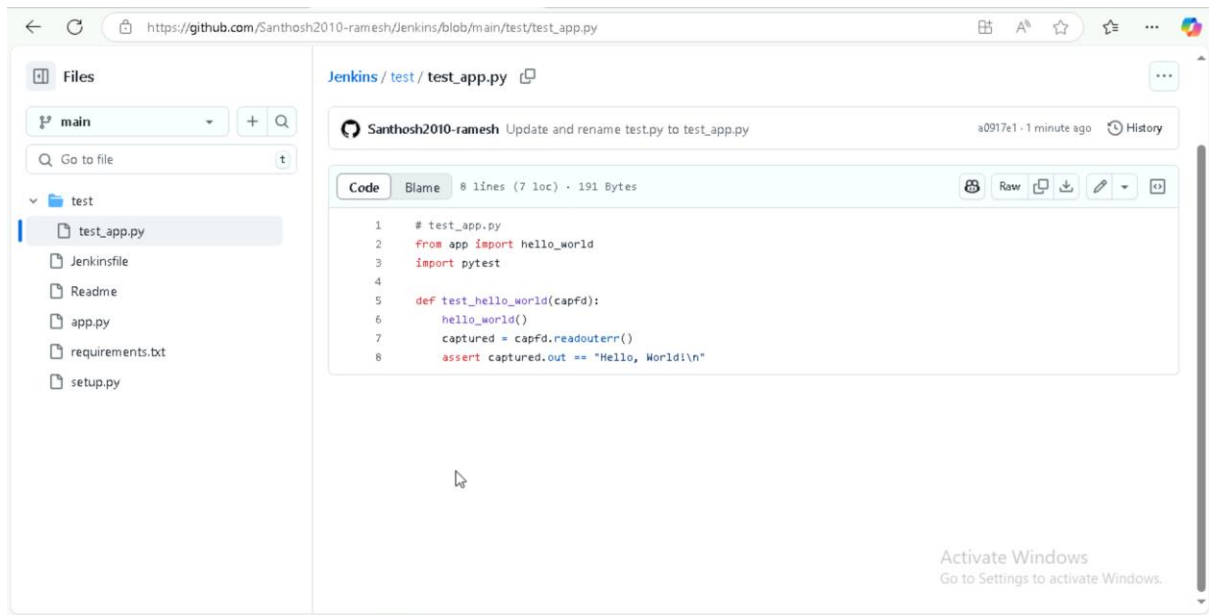
Jenkins / setup.py

Santhosh2010-ramesh Update setup.py 9ed3fed • 6 minutes ago History

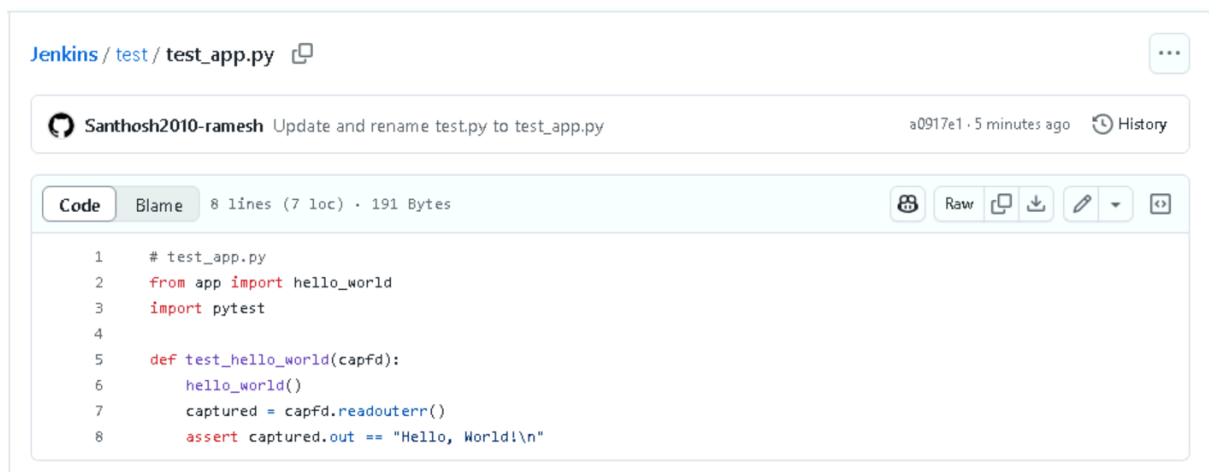
Code Blame 10 lines (9 loc) • 174 Bytes

```
1 from setuptools import setup, find_packages
2
3 setup(
4     name='python-app',
5     version='0.1',
6     packages=find_packages(),
7     install_requires=[
8         'pytest',
9     ],
10 )
```

## Step 5: Creating a sample test.py for checkrun.



## Step 6: Build and Archive Artifacts with Declarative: Checkout SCM, Clone Repository, set up Virtual Environment, Install Dependencies, Run Tests, Build Artifact, Archive Artifact.



## Step 7: Run the Jenkins Job


1. Click **Build Now** to trigger a build.
2. Monitor logs in **Console Output**.

← → ↻

localhost:8080/job/Python-App-Build/15/console

90%

🔍 🔔 👤 admin ⌵ 📄 log out


 Jenkins


🔍 🔔 👤 admin ⌵ 📄 log out


Dashboard > Python-App-Build > #15


Status


</> Changes


 Console Output


 Edit Build Information


 Delete build '#15'


 Timings


 Git Build Data


 See Fingerprints

 Open Blue Ocean

 Pipeline Overview

 Pipeline Console

 Restart from Stage

 Replay

✓ Console Output

Download

Copy

View as plain text

Started by user [admin](#)

Obtained Jenkinsfile from git <https://github.com/Santhosh2010-ramesh/Jenkins.git>

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [jenkins-slave1](#) in /opt/jenkins-slave1/workspace/Python-App-Build

[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

Fetching changes from the remote Git repository

> git rev-parse --resolve-git-dir /opt/jenkins-slave1/workspace/Python-App-Build/.git # timeout=10

> git config remote.origin.url <https://github.com/Santhosh2010-ramesh/Jenkins.git> # timeout=10

Fetching upstream changes from <https://github.com/Santhosh2010-ramesh/Jenkins.git>

> git --version # timeout=10

> git --version # 'git version 2.43.0'

> git fetch --tags --force --progress -- <https://github.com/Santhosh2010-ramesh/Jenkins.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10