

# Assignment 4 - CI/CD Pipeline with Jenkins on Kubernetes

---

**Course:** DevOps - Container Orchestration

**Assignment:** Jenkins CI/CD Pipeline Integration

**Date:** November 29, 2025

## Team Members

- **Nikhil Shankar Chirakkal Sivasankaran** - 9026254
  - **Richard Andrey Biscazzi** - 8903530
- 

## Overview

This assignment demonstrates a complete CI/CD pipeline using Jenkins deployed on Kubernetes, integrated with both GitHub and self-hosted Gitea for automated build triggers.

---

## Architecture

### Infrastructure Setup

- **Jenkins Server:** Deployed on Kubernetes cluster
- **Version Control:** GitHub + Self-hosted Gitea (from Assignment 3)
- **CI/CD Runner:** Jenkins on K8s
- **Trigger Method:** Webhooks (GitHub & Gitea)

### Repositories

Repository	Purpose	Branch	URL
Jenkins Infrastructure	Jenkins deployment on K8s	assignment4	<a href="#">container-asssignment3</a>
Python Test App	Sample Flask application with CI/CD	master	<a href="#">PythonTest-CICD-Assignment4</a>

### Deployment URLs

- **Jenkins Server:** <http://jenkins.devsecmindset.dev/>
  - **Gitea Instance:** Running locally from Assignment 3
- 

## Implementation Steps

### 1. Jenkins Setup on Kubernetes

Richard deployed Jenkins on the Kubernetes cluster using configurations from the [container-asssignment3](#) repository (branch: [assignment4](#)).

The screenshot shows a browser window with multiple tabs open. The active tab is for the Jenkins pipeline 'assignment4-pipeline-niks' at the URL [jenkins.devsecmindset.dev/job/assignment4-pipeline-niks/](http://jenkins.devsecmindset.dev/job/assignment4-pipeline-niks/). The Jenkins logo is visible on the left. The main content area shows the pipeline name 'assignment4-pipeline-niks' with a red 'X' icon. Below it is a 'Permalinks' section with a list of recent builds:

- Last build (#2), 6.1 sec ago
- Last failed build (#1), 10 min ago
- Last unsuccessful build (#1), 10 min ago
- Last completed build (#1), 10 min ago

On the left sidebar, there are several options: Status (selected), Changes, Build Now, Configure, Delete Pipeline, Rename, Pipeline Syntax, and GitHub Hook Log. A 'Builds' section displays two entries under 'Today':

- #2 4:05 AM (with a progress bar)
- #1 3:55 AM

A red box highlights the first entry, #2 4:05 AM.

## 2. Python Application Repository

Created a Flask-based Python application with the following structure:

- [app.py](#) - Simple Flask web application
- [requirements.txt](#) - Python dependencies
- [Jenkinsfile](#) - Pipeline configuration

## 3. GitHub Integration

### Webhook Configuration:

- URL: <https://jenkins.devsecmindset.dev/github-webhook/>
- Trigger: Push events
- Content-Type: application/json

The screenshot shows a GitHub project settings page for 'PythonTest-CI\_CD-Assignment4'. A red box highlights the project name in the header. Another red box highlights the 'Webhooks' section in the main content area. The 'Webhooks' section displays a single entry for the Jenkins webhook, which has never been triggered.

Okay, that hook was successfully created. We sent a ping payload to test it out! Read more about it at <https://docs.github.com/webhooks/#ping-event>.

**General**

**Access**

**Collaborators**

**Moderation options**

**Code and automation**

**Branches**

**Tags**

**Rules**

**Actions**

**Models**

**Webhooks** Preview

**Copilot**

**Environments**

**Codespaces**

**Pages**

**Security**

**Advanced Security**

**Deploy keys**

**Secrets and variables**

**Integrations**

**Webhooks**

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

● <https://jenkins.devsecmindset.dev/github-webhook/> (push)

This hook has never been triggered.

**Edit** **Delete**

### Pipeline Trigger Test:

The screenshot shows the Jenkins interface for a job named 'assignment4-pipeline-niks'. The build number is #6, which was started by GitHub push by NikhilShankar on Nov 29, 2025, at 4:22:31 AM. The build took 31 seconds and completed 5 minutes 22 seconds ago. The build status is green with a checkmark. A red box highlights the build number and timestamp. Another red box highlights the 'Started by GitHub push by NikhilShankar' message. The sidebar on the left lists various Jenkins management options like Status, Changes, Console Output, and Git Build Data. The bottom right corner shows the Jenkins version as 2.528.2.

## 4. Gitea Integration

### **Repository Clone:**

Cloned the GitHub repository to self-hosted Gitea instance.

niks1267 / PythonTest-CI\_CD-Assignt4

**Code** Issues Pull Requests Actions Packages Projects Releases Wiki Activity

8 Commits 1 Branch 0 Tags

master Go to file Add File

**niks1267** 565d165f5c Update Jenkinsfile 1 minute ago  
app.py Update app.py 8 minutes ago  
Jenkinsfile Update Jenkinsfile 1 minute ago  
requirements.txt App and Jenkins file created 1 hour ago

Description  
No description provided  
Manage Topics  
31 KiB

Languages  
Python 100%

Powered by Gitea Version: 1.24.6 Page: 61ms Template: 5ms English Licenses API

## Generic Webhook Configuration:

Installed Generic Webhook Trigger Plugin in Jenkins and configured Gitea webhook.

Webhooks - PythonTest-CI\_CD-Assignt4

Triggering Jenkins pipeline from

Issues Pull Requests Milestones Explore

**niks1267 / PythonTest-CI\_CD-Assignt4**

Code Issues Pull Requests Actions Packages Projects Releases Wiki Activity Settings

Webhooks	
Webhooks automatically make HTTP POST requests to a server when certain Gitea events trigger. Read more in the <a href="#">webhooks guide</a> .	
●	<a href="https://jenkins.devsecmindset.dev/generic-webhook-trigger/invoke?token=my-secret-gitea-token-123">https://jenkins.devsecmindset.dev/generic-webhook-trigger/invoke?token=my-secret-gitea-token-123</a>

Powered by Gitea Version: 1.24.6 Page: 12ms Template: 4ms English Licenses API

## Code Change Test:

Modified `app.py` in Gitea to test pipeline trigger.

```

1  from flask import Flask
2  from datetime import datetime
3
4  app = Flask(__name__)
5
6  # This will be set during deployment
7  DEPLOYMENT_TIME = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
8
9  @app.route('/')
10 def hello():
11     return f"""Hello from Richard and Nikhil. This is the version deployed on {DEPLOYMENT_TIME}
12
13 Triggering pipeline for CICD Assignment 4.
14
15 Triggering pipeline from gitea|
16 """
17
18 if __name__ == '__main__':
19     app.run(host='0.0.0.0', port=9876)
20

```

## Successful Pipeline Execution:

#8 (Nov 29, 2025, 4:40:12 AM)

Triggered by Gitea

This run spent:

- 19 sec waiting;
- 38 sec build duration;
- 46 sec total from scheduled to completion.

git Revision: ab2852462ab9dfc8e10c8c6a42103d91baf8b5b  
Repository: https://github.com/NikhilShankar/PythonTest-CICD-Assignment4  
• refs/remotes/origin/master

</> No changes.

## Jenkinsfile Pipeline

The pipeline consists of three stages:

```

pipeline {

    triggers {
        GenericTrigger(
            genericVariables: [

```

```
[key: 'ref', value: '$.ref']
],
causeString: 'Triggered by Gitea',
token: 'my-secret-gitea-token-123', // <-- This is YOUR_TOKEN
printContributedVariables: true,
printPostContent: true
)
}

agent any

stages {
    stage('Checkout') {
        steps {
            echo 'Code checked out from GitHub'
            sh 'ls -la'
        }
    }

    stage('Validate') {
        steps {
            echo 'Validating project structure...'
            sh '''
                echo "Checking for required files..."
                test -f app.py && echo "✓ app.py found" || echo "X app.py
missing"
                test -f requirements.txt && echo "✓ requirements.txt found"
                || echo "X requirements.txt missing"
                test -f Jenkinsfile && echo "✓ Jenkinsfile found" || echo "X
Jenkinsfile missing"
                ...
            '''
        }
    }

    stage('Build Info') {
        steps {
            echo 'Displaying build information...'
            sh '''
                echo "Repository: PythonTest-CICD-Assignment4"
                echo "Branch: ${GIT_BRANCH}"
                echo "Commit: ${GIT_COMMIT}"
                cat app.py
                ...
            '''
        }
    }

    stage('Deploy') {
        steps {
            echo '✓ Pipeline completed successfully!'
            echo "Build finished at: ${new Date()}"
        }
    }
}
```

```
post {
    success {
        echo '✓ Build SUCCESS - All stages passed!'
    }
    failure {
        echo '✗ Build FAILED'
    }
}
```

## Key Features Implemented

- Jenkins deployed on Kubernetes cluster
- Automated GitHub webhook integration
- Self-hosted Gitea integration using Generic Webhook Plugin
- Multi-source repository support (GitHub + Gitea)
- Automated pipeline triggers on code push
- Successful validation and build processes

## Testing Results

Test Case	Source	Result	Screenshot
Manual Trigger	Jenkins UI	<input checked="" type="checkbox"/> Success	Screenshot 1
GitHub Push	GitHub Webhook	<input checked="" type="checkbox"/> Success	Screenshot 4
Gitea Push	Gitea Webhook	<input checked="" type="checkbox"/> Success	Screenshot 7

## Collaboration

- **Richard:** Set up Jenkins infrastructure on Kubernetes, configured cluster access
- **Nikhil:** Created Python application, configured webhooks (GitHub & Gitea), tested pipeline triggers

## Conclusion

Successfully implemented a complete CI/CD pipeline using Jenkins on Kubernetes with dual repository integration (GitHub and self-hosted Gitea). The pipeline automatically triggers on code changes from both sources, demonstrating enterprise-grade DevOps practices.