



Jenkins vs GitHub CI/CD

TASK 3-REPORT

20F-0408 Muhammad Muaaz Ulhaq

BSE-7A DevOps



Table of Contents

1. Introduction	2
2. Jenkins	2
2.1. Overview	2
2.2. Key Features	2
3. GitHub CI/CD	2
3.1. Overview	2
3.2. Key Features	3
4. Difficulties Encountered with Jenkins on MacBook.....	3
5. Results.....	3
5.1. Jenkins	3
5.2. GitHub CI/CD	4

Comparison of Jenkins and GitHub CI/CD

1. Introduction

Continuous Integration/Continuous Deployment (CI/CD) is an essential practice in modern software development. Two popular tools for implementing CI/CD pipelines are Jenkins and GitHub CI/CD. This report aims to differentiate these two tools and discuss the challenges faced when attempting to use Jenkins on a MacBook.

2. Jenkins

2.1. Overview

- Jenkins is an open-source automation server used for building, testing, and deploying code.
- Jenkins supports a wide range of plugins and integrations, making it highly customizable.
- It can be installed on various platforms, including macOS, Linux, and Windows.

2.2. Key Features

- **Pipeline Support:** Jenkins provides a powerful DSL for defining complex build and deployment pipelines.
- **Wide Plugin Ecosystem:** A vast library of plugins enables extensive integration with third-party tools.
- **Customization:** Jenkins is highly configurable to meet specific project requirements.

3. GitHub CI/CD

3.1. Overview

- GitHub CI/CD is a built-in CI/CD solution provided by GitHub, tightly integrated with the version control platform.
- It uses GitHub Actions as the underlying workflow automation engine.
- GitHub CI/CD is cloud-based and requires no additional server setup.

3.2. Key Features

- **Native Integration:** Seamless integration with GitHub repositories, eliminating the need for external services.
- **YAML Configuration:** Workflow definitions are stored in a repository as YAML files, making them version-controlled and transparent.
- **Scalability:** GitHub CI/CD is designed to scale effortlessly, supporting a wide range of use cases.

4. Difficulties Encountered with Jenkins on MacBook

While trying to get Jenkins up and running on a MacBook, we faced some problems mainly related to how the terminal behaves on macOS. Here's why it didn't work out as expected:

- **Terminal Differences:** Jenkins often needs to work closely with the terminal for tasks like building and deploying code. However, the way the terminal behaves on macOS can be quite different from what Jenkins expects. This caused unexpected problems and made the setup challenging.

5. Results

5.1. Jenkins

S	Build	Time Since ↑	Status	
⊗	DevOps #6	13 min	broken for a long time	🔗
⊗	DevOps #5	1 day 0 hr	broken for a long time	🔗
⊗	DevOps #4	1 day 0 hr	broken for a long time	🔗
⊗	DevOps #3	1 day 0 hr	broken for a long time	🔗
⊗	DevOps #1	1 day 0 hr	broken since this build	🔗

Icon: S M L

Icon legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

5.2. GitHub CI/CD

Re-run triggered 12 minutes ago

Muaaz-Ulhaq a4c7340 main

Status

Success

Total duration

18s

Artifacts

—

file.yml

on: push

✓ build

6s

Re-run all jobs

Latest #3

...

✓ Latest attempt #3

✓ succeeded on Oct 03 by Muaaz-Ulhaq

Attempt #2

✓ succeeded on Oct 02 by Muaaz-Ulhaq

Attempt #1

✓ succeeded on Oct 02 by Muaaz-Ulhaq