



Implementation DevOps using Git, GitHub, Docker and Jenkins on Web CV

Group 2

Names : Andrina Haura Azzahra
Tasya Syafa Kamila

Class : 3SE3

CEP CCIT

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

2024

PROJECT ON

*Implementation DevOps using Git, GitHub, Docker, and Jenkins on
Web CV*

Developed by

- 1. Andrina Haura Azzahra**
- 2. Tasya Syafa Kamila**

The logo for NIIT, consisting of the letters 'NIIT' in a bold, blue, serif font.

Implementation DevOps using Git, GitHub, Docker, and Jenkins on Web CV

Batch Code : 3SE3

Start Date : January 19st 2024

End Date : January 24th 2024

Name of Faculty : Riza Muhammad Nurman

Names of Developer :

1. Andrina Haura Azzahra
2. Tasya Syafa Kamila

Date of Submission: January 24th 2024



CERTIFICATE

This is to certify that this report titled Implementation DevOps using Git, GitHub, and Jenkins on Web CV embodies the original work done by Andrina Haura Azzahra and Tasya Syafa Kamila. Project in partial fulfillment of their course requirement at NIIT.

Coordinator:

Riza Muhammad Nurman

ACKNOWLEDGEMENT

With the name of God, The Most Gracious Most Merciful, we praise over his presence, which has been part of his grace, so that project about introduction to web can be completed which God willing useful for all of us.

Project report has been in rows with its fullest and got help from various parties and the source so that it can facilitate the creation of this report. And especially to Mr. Riza Muhammad Nurman, who has provided guidance in making these Project report.

This report was prepared with a variety of obstacles. Whether it is coming from the internal and external problems. But with great patience and especially the help of Almighty God finally reports Project can be resolved.

Project report this may be able to provide a wider insight and thought-provoking contribution to the readers especially-CCIT FTUI students. This report is probably still a lot of shortcomings and arguably hasn't been perfect. To that end, to give input for the sake of supervising lecturer improvement making this report in the days to come and expect criticism and suggestions from readers

SYSTEM ANALYSIS

System Summary:

The project is titled “Implementation DevOps using Git, GitHub, Jenkins and Docker on CV Website” is it for shows cv on website. And then enables the viewers to see how kind of CV that we made.

This program uses the Html and also integrates with Jenkins, Docker and GitHub.

TECHNOLOGY USED

Git

Git is a free and open-source distributed version control system. It is designed to handle from small to very large projects with speed and efficiency, but it can be used to track changes in any set of files. Git allows a team of people to work together, all using the same files.

GitHub

GitHub is a web-based interface that uses Git, the open source version control software that lets multiple people make separate changes to web pages at the same time. As Carpenter notes, because it allows for real-time collaboration, GitHub encourages teams to work together to build and edit their site content.

Jenkins

Is an open-source continuous integration/ continuous delivery and deployment automation software written in Java. Jenkins helps to automate the non-human part of the software development process, with continuous integration and facilitating technical aspects of continuous delivery. Jenkins can integrate with a number of testing and deployment technologies.

Docker

Docker is an open-source containerization platform used for developing, deploying, and managing applications in lightweight virtualized environments called containers.

It is mainly used as a software development platform for developing distributed applications that work efficiently in different environments. By making the software system agnostic, developers don't have to worry about compatibility issues. Packaging apps into isolated environments (containers) also makes it easier to develop, deploy, maintain, and use applications.

TECHNOLOGY USED

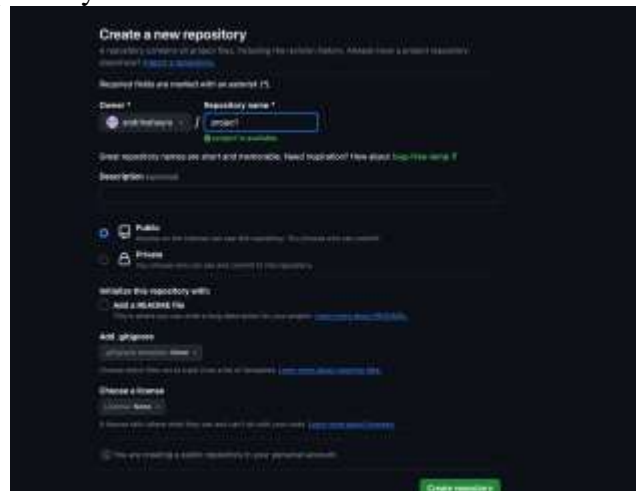
Jenkins Pipeline

Jenkins pipeline is a way to define the steps of a continuous integration and continuous delivery (CI/CD) process in Jenkins. It allows for defining build, test, and deployment steps in a Jenkinsfile, which is stored in source control along with the rest of the code. The pipeline can be visualized in the Jenkins user interface, making it easy to understand and troubleshoot. It also supports features such as parallel execution, conditional branching, and manual approvals.

DEVOPS IMPLEMENTATION

I. USER 1:

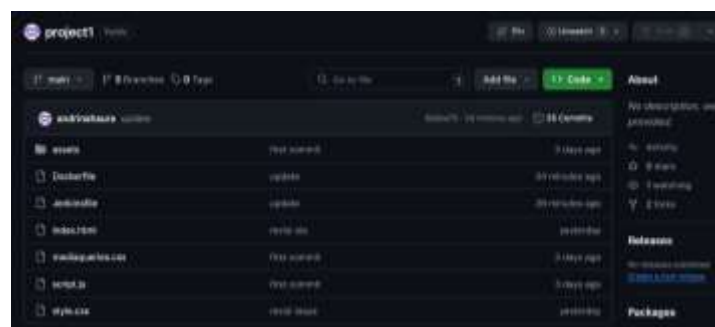
1.1. Create new repository



1.2. Git init, Add, commit and push project to the exist remote repository.

```
heaven-MacBook-Air:documents heaven$ cd project
heaven-MacBook-Air:project heaven$ git init
Initialized empty Git repository in /Users/heaven/Documents/project/.git/
heaven-MacBook-Air:project heaven$ git add .
heaven-MacBook-Air:project heaven$ git commit -m "first commit"
[master (root-commit) 112711a] first commit
10 files changed: 103 insertions(+), 0 deletions(-)
create mode 100644 assets/about-git.png
create mode 100644 assets/arrow.png
create mode 100644 assets/checkmark.png
create mode 100644 assets/education.png
create mode 100644 assets/fossil.png
create mode 100644 assets/expense.png
create mode 100644 assets/github.png
create mode 100644 assets/linkedin.png
create mode 100644 assets/profile-pic-2.png
create mode 100644 assets/profile-pic.png
create mode 100644 assets/project-1.png
create mode 100644 assets/project-2.png
create mode 100644 assets/project-3.png
create mode 100644 assets/README-sample.pdf
create mode 100644 index.html
create mode 100644 main.scss
create mode 100644 script.js
heaven-MacBook-Air:project heaven$ git branch -M main
heaven-MacBook-Air:project heaven$ git remote add origin https://github.com/heavenheaven/project.git
heaven-MacBook-Air:project heaven$ git push -u origin main
Counting objects: 11, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 4.0K | 4.0K/s, done.
Total 11 (delta 0), reused 0 (delta 0)
To https://github.com/heavenheaven/project.git
 * branch 'main' set up to track remote branch 'main' from 'origin'
```

1.3. GitHub Repository



DEVOPS IMPLEMENTATION

1.4. Add backend branch and user 1 merge from user1 which is tasyasyf/main.

```
tasyasyf@macbook-Air:project2 $ git branch backend
* backend
tasyasyf@macbook-Air:project2 $ git checkout backend
Switched to branch 'backend'
tasyasyf@macbook-Air:project2 $ git merge tasyasyf/main
Updating 00000000..00000000
Fast-forward
 1 file changed, 1 insertion(+), 1 deletion(-)
tasyasyf@macbook-Air:project2 $ git push -u origin backend
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote: unable to access 'https://github.com:443/tasyasyf/project2.git/': Could not resolve host: github.com
origin https://github.com:443/tasyasyf/project2.git (fetch)
origin https://github.com:443/tasyasyf/project2.git (push)
tasyasyf@macbook-Air:project2 $ git fetch
tasyasyf@macbook-Air:project2 $ git push -u origin backend
Counting objects: 1, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (1/1), done.
Writing objects: 100% (1/1), 1.25 KiB | 1.25 MiB/s, done.
Total 1 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local objects.
remote:
remote: Create a pull request for 'backend' on GitHub by visiting:
remote:   https://github.com/tasyasyf/project2/pull/new/backend
remote:
To https://github.com:443/tasyasyf/project2.git
 * [new branch]      backend -> origin/backend
Branch 'backend' set up to track remote branch 'backend' from 'origin'.
```

II. USER 2:

2.1.Clone Repository on GitHub

```
Last login: Mon Jan 22 23:12:25 on tty000
tasyasyf@macbook-Air:~ $ git clone https://github.com/tasyasyf/project1.git
Cloning into 'project1'...
remote: Enumerating objects: 73, done.
remote: Counting objects: 100% (73/73), done.
remote: Compressing objects: 100% (58/58), done.
remote: Total 73 (delta 33), reused 34 (delta 16), pack-reused 0
Receiving objects: 100% (73/73), 4.36 MiB | 813.00 KiB/s, done.
Resolving deltas: 100% (33/33), done.
```

2.2.Switching to version1 branch

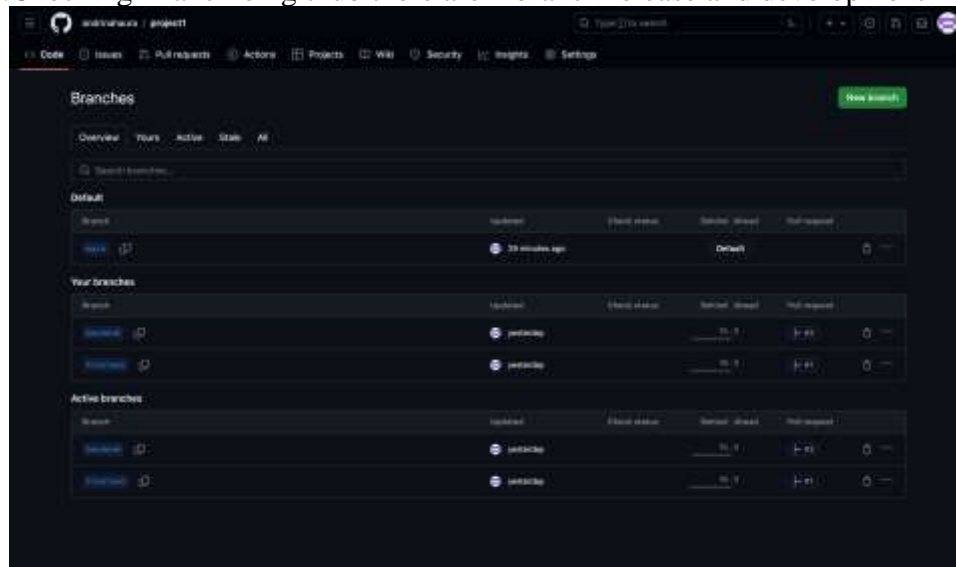
```
tasyasyf@macbook-Air:~ $ cd project1
tasyasyf@macbook-Air:project1 $ git remote -v
origin https://github.com/tasyasyf/project1.git (fetch)
origin https://github.com/tasyasyf/project1.git (push)
tasyasyf@macbook-Air:project1 $ git remote remove origin
tasyasyf@macbook-Air:project1 $ git remote add origin https://github.com/tasyasyf/project1.git
tasyasyf@macbook-Air:project1 $ git branch version1
tasyasyf@macbook-Air:project1 $ git checkout version1
* version1
Switched to branch 'version1'
tasyasyf@macbook-Air:project1 $ git init
Initialized empty Git repository in /Users/tasyasyf/project1/.git/
tasyasyf@macbook-Air:project1 $ git add .
tasyasyf@macbook-Air:project1 $ git commit -m "initial tasya"
[version1 60b6b7] initial tasya
1 file changed, 1 insertion(+), 1 deletion(-)
tasyasyf@macbook-Air:project1 $ git push -u origin version1
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Delta compression using up to 8 threads.
Writing objects: 100% (1/1), 376 bytes | 376.00 KiB/s, done.
Total 1 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local objects.
remote:
remote: Create a pull request for 'version1' on GitHub by visiting:
remote:   https://github.com/tasyasyf/project1/pull/new/version1
remote:
To https://github.com/tasyasyf/project1.git
 * [new branch]      version1 -> origin/version1
tasyasyf@macbook-Air:project1 $
```

DEVOPS IMPLEMENTATION

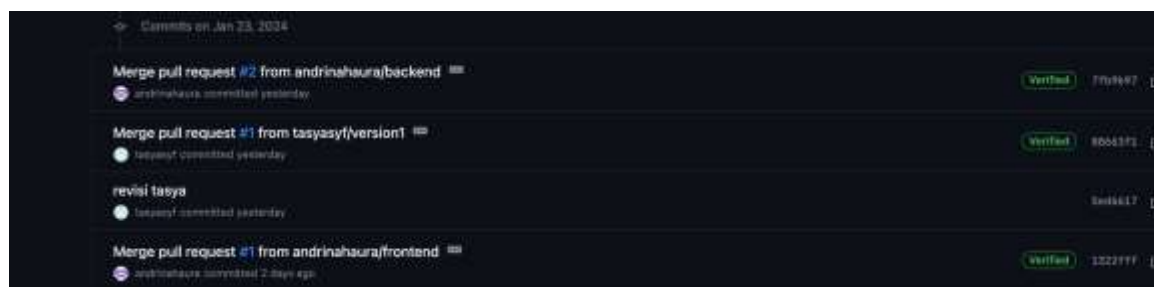
III. GITHUB, JENKINS, and Docker

1. Github

1.1. Checking Branch on github there are 2 branch release and development

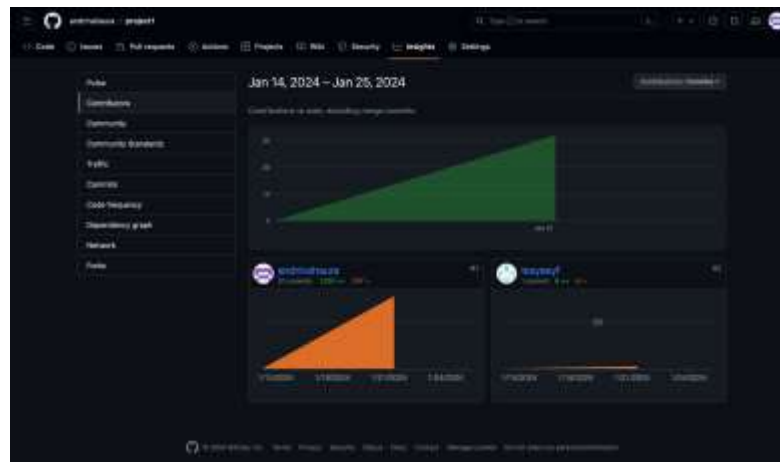


1.2. History Commit



DEVOPS IMPLEMENTATION

1.3.Data contribute for our project



2. Jenkins Pipeline

2.1.Create New item pipeline

The screenshot shows the 'Create new item' dialog in Jenkins. At the top, there is a text input field for 'Enter an item name' with the value 'unclassified'. Below this, there is a list of project types with their descriptions:

- Freestyle project**: Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**: Encapsulate long-running activities that 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.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**: Creates a container that stores related items in its child. It is grouping things together (like a file, a folder, or a workspace). As you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository.
- Organizational Folder**: Creates a set of named workspaces identified by searching for regular expressions.

DEVOPS IMPLEMENTATION

2.2. Pipeline Text

The screenshot shows the Burp Suite interface with the 'HTTP history' tab selected. The selected item is a POST request to 'http://10.10.10.10:8080/submit.php'. The 'Request' tab is active, displaying the raw HTTP request. The request is a POST to 'http://10.10.10.10:8080/submit.php' with a 'Content-Type' of 'application/x-www-form-urlencoded'. The body contains 'username=admin' and 'password=admin'. The 'Response' tab is also visible, showing a '200 OK' status and an 'HTTP/1.1' version.

2.3.Result Console Output

[illegible]

DEVOPS IMPLEMENTATION

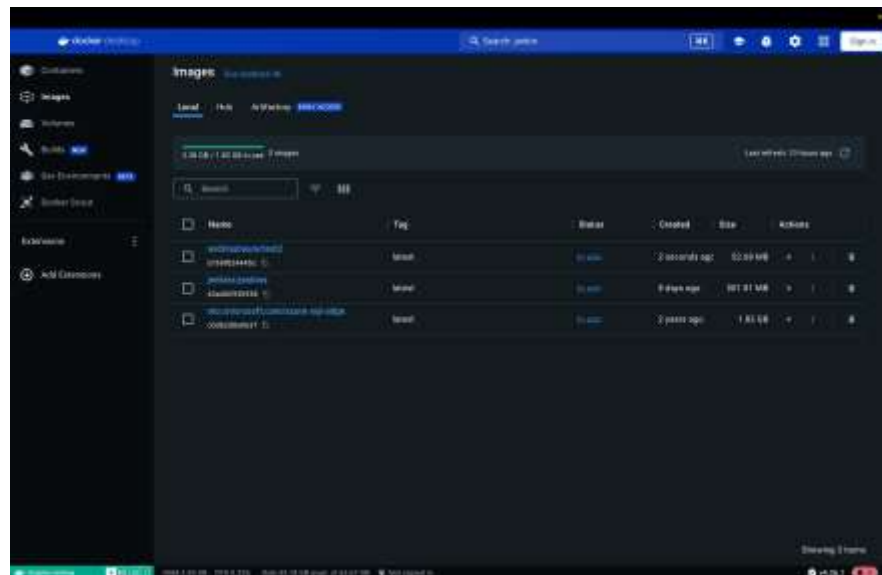
2.4. Result of the item pipeline



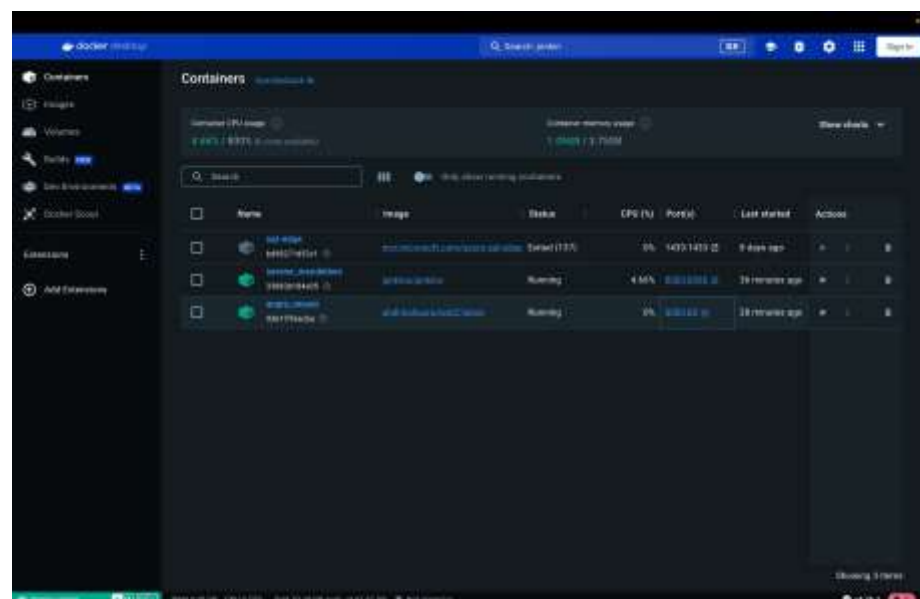
DEVOPS IMPLEMENTATION

3. Docker

3.1.Docker image

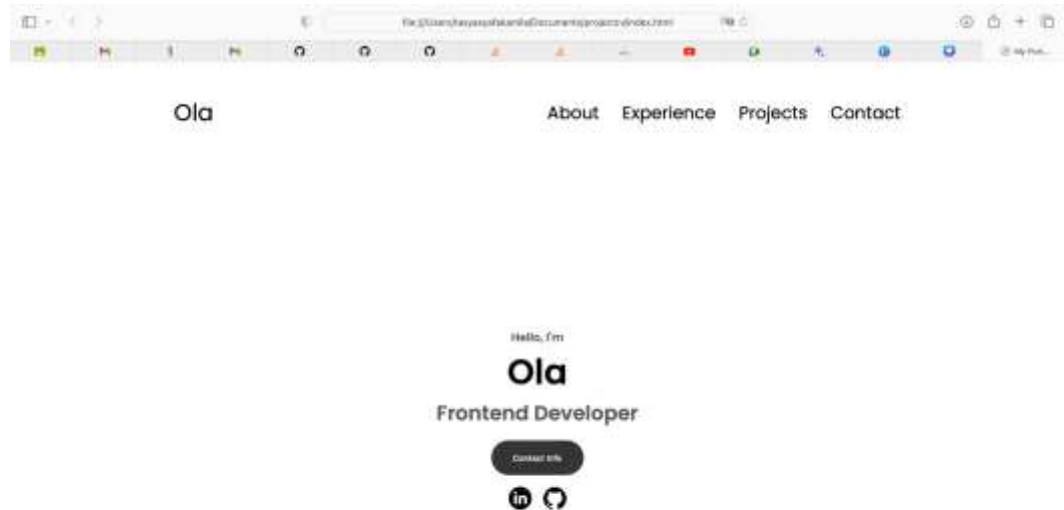


3.2.Docker containers



DEVOPS IMPLEMENTATION

3.3.Run the localhost in website



ACTIVITY LIST

No	Activity	Start Date	End Date	Person(s) Involved
1	Creating Repository	22/1/2024	22/1/2024	Andrina
2	Push Project in Github	22/1/2024	22/1/2024	Andrina
3	Forking Project	22/1/2024	22/1/2024	Tasya
4	Cloning Website using Git	22/1/2024	22/1/2024	Tasya
5	Change some code on website	22/1/2024	22/1/2024	Andrina & Tasya
6	Add and commit to local repository	22/1/2024	22/1/2024	Andrina & Tasya
7	Push Html to GitHub	22/1/2024	23/1/2024	Andrina & Tasya
8	Add pull request	22/1/2024	23/1/2024	Andrina
9	Creating jenkins project	23/1/2024	23/1/2024	Andrina & Tasya
10	Build project in jenkins	23/1/2024	23/1/2024	Andrina & Tasya
11	Build pipeline	23/1/2024	23/1/2024	Andrina & Tasya

CONFIGURATION

Hardware : Macbook M2

Operating System : MacOS

Software : Git, GitHub and Jenkins, Docker, Microsoft Word, Microsoft Power Point

PROJECT FILE DETAILS

No	File Name	Remarks
1.	CV Website.rar	Web folder contains Visual Studio syntax to create CV website
2.	CV Website.docx	The Paper of our project
3.	CV Website.pptx	Our Power Point presentation file
4.	https://github.com/andrinahaura/project1.git	Our Github Repository
5.	http://localhost:8089/ on local Docker andrina haura's laptop	Our DockerHub