

CI/CD Pipeline Implementation Using GitLab

As part of my DevOps practice, I implemented a CI/CD pipeline using GitLab for a basic web application. The pipeline is designed to automate key stages of development and deployment. The first step in the pipeline focuses on the build process to ensure that the application compiles correctly and is free from build-time errors.

1. Make a server in AWS EC2 instance and ssh from local

```
.ssh — ubuntu@ip-10-0-0-13: ~ — ssh -i ~/.ssh/my-key-pair.pem ubuntu@34.232.64.137 — 80x24
ubuntu@ip-10-0-0-13:~$
```

- Run:

```
sudo apt update
sudo apt install nginx
```

```
.ssh — ubuntu@ip-10-0-0-13: ~ — ssh -i ~/.ssh/my-key-pair.pem ubuntu@34.232.64.137 — 80x24
Get:21 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [160 kB]
Get:22 https://packages.grafana.com/oss/deb stable/main amd64 Packages [407 kB]
Get:23 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [2 1.5 kB]
Get:24 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [851 kB]
Get:25 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [187 kB]
Get:26 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:27 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:28 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Fetched 7651 kB in 2s (3216 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
64 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://packages.grafana.com/oss/deb/dists/stable/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-10-0-0-13:~$
```

```
.ssh — ubuntu@ip-10-0-0-13: ~ — ssh -i ~/.ssh/my-key-pair.pem ubuntu@34.232.64.137 — 80x24
Preparing to unpack .../nginx-common_1.24.0-2ubuntu7.3_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7.3) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.24.0-2ubuntu7.3_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7.3) ...
Setting up nginx (1.24.0-2ubuntu7.3) ...
Setting up nginx-common (1.24.0-2ubuntu7.3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-0-0-13:~$
```

- Make a directory for deploy

```
.ssh — ubuntu@ip-10-0-0-13: ~ — ssh -i ~/.ssh/my-key-pair.pem ubuntu@34.232.64.137 — 80x24
ubuntu@ip-10-0-0-13:~$ sudo mkdir -p /var/www/html
ubuntu@ip-10-0-0-13:~$ sudo chown -R ubuntu:ubuntu /var/www/html
ubuntu@ip-10-0-0-13:~$
```

- On the local computer create an ssh keygen for gitlab and the ec2 server

```
.ssh — ubuntu@ip-10-0-0-13: ~ — zsh — 108x24
(base) dynoaryawana@Dynos-MacBook-Pro .ssh % ssh-keygen -t rsa -b 4096 -C "gitlab-ci" -f gitlab-deploy-key
Generating public/private rsa key pair.
Enter passphrase for "gitlab-deploy-key" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in gitlab-deploy-key
Your public key has been saved in gitlab-deploy-key.pub
The key fingerprint is:
SHA256:p5vMh1m0QgUD2dq7RSBHwP8slW2KjCtmSRNr4noZOS8 gitlab-ci
The key's randomart image is:
+---[RSA 4096]-----+
|          .+..+..+          |
|         oo*.O..o         |
|        *.Oo*o          |
|       ..++B o          |
|      .S+..            |
|     .oB+             |
|    .EB.              |
|   ++.               |
|  =.                 |
+---[SHA256]-----+
(base) dynoaryawana@Dynos-MacBook-Pro .ssh %
```

- Put public key into EC2 server

```
.ssh — ubuntu@ip-10-0-0-13: ~ — zsh — 118x24
(base) dynoaryawana@Dynos-MacBook-Pro .ssh % scp -i my-key-pair.pem gitlab-deploy-key.pub ubuntu@34.232.64.137
(base) dynoaryawana@Dynos-MacBook-Pro .ssh %
```

- Log in to Gitlab, create a new project, create a variable and enter the private key in the variable.

Environments ?

All (default)

Visibility ?

☐ Visible
Can be seen in job logs.
 ☒ Masked
Masked in job logs but value can be revealed in CI/CD settings. Requires values to meet regular expressions requirements.
 ☐ Masked and hidden
Masked in job logs, and can never be revealed in the CI/CD settings after the variable is saved.

Flags

☒ Protect variable
Export variable to pipelines running on protected branches and tags only.
 ☒ Expand variable reference
\$ will be treated as the start of a reference to another variable.

Description (optional)

The description of the variable's value or usage.

Key

SSH_PRIVATE_KEY_BASE64

You can use CI/CD variables with the same name in different places, but the variables might overwrite each other. What is the order of precedence for variables?

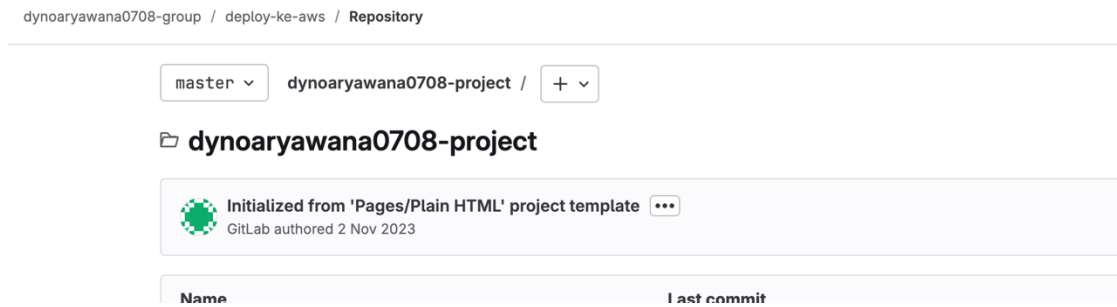
Value

NSZTVUV2xJmNGa0tvckdQaG13L2LUYTF3MzBpSFhNLCxZy9EK1d0dwpTeXp4c3JleW1pZV86bjJWbW1sOjhpYzZUZWx5ZS2xL3BUNG1ybnZldGx8BTlpKMTl4U0UvVkg3Z6NFd3EVM8SMbT8mZS9NCnFdB8RejRWKlpd21ESzUzZjV3Uys4eU91NU91WVXUnhNdHA2V3UvaFgwbmVWZjJzenN5

Add variable

Cancel

- Display that has been created from the project



- Adding origin into git

```
CICD Gitlab — ubuntu@ip-10-0-0-13: ~ — zsh — 86x24
(base) dynoaryawana@Dynos-MacBook-Pro CICD Gitlab % git remote add origin https://gitlab.com/dynoaryawana0708-group/dynoaryawana0708-project
(base) dynoaryawana@Dynos-MacBook-Pro CICD Gitlab %
```

- Create index.html and .gitlab-ci.yml in the same directory on the local computer

The screenshot shows a code editor with two tabs: 'index.html' and '.gitlab-ci.yml'. The 'index.html' tab is active, displaying the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Welcome to My Simple Page</title>
</head>
<body>
  <h1>Hello from EC2!</h1>
  <p>This static page was deployed automatically via GitLab CI/CD 🚀</p>
</body>
</html>
```

The screenshot shows the same code editor with the '.gitlab-ci.yml' tab active. The file is titled '.gitlab-ci - GitLab CI Configuration file (ci.json)'. The content of the file is as follows:

```
stages:
  - build

build:
  stage: build
  image: alpine
  script:
    - echo "HTML build selesai (tidak ada proses build)"
  only:
    - master
```

- Push to gitlab with :

`git add .`

`git status`

`git commit -m`

```
CICD Gitlab --zsh-- 197x55
(base) dynoaryawana@Dynos-MacBook-Pro CICD Gitlab % git add .
(base) dynoaryawana@Dynos-MacBook-Pro CICD Gitlab % git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitlab-ci.yml

(base) dynoaryawana@Dynos-MacBook-Pro CICD Gitlab % git commit -m "Update .gitlab-ci.yml untuk build stage"

[master 7d5ae01] Update .gitlab-ci.yml untuk build stage
Committer: Dyno Aryawana <dynoaryawana@Dynos-MacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+), 18 deletions(-)
(base) dynoaryawana@Dynos-MacBook-Pro CICD Gitlab %
```

- `git push origin master`

```
CICD Gitlab --zsh-- 197x55
(base) dynoaryawana@Dynos-MacBook-Pro CICD Gitlab % git push origin master
warning: redirecting to https://gitlab.com/dynoaryawana0708-group/dynoaryawana0708-project.git/
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 344 bytes | 344.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
To https://gitlab.com/dynoaryawana0708-group/dynoaryawana0708-project
 b965a47..7d5ae01 master -> master
(base) dynoaryawana@Dynos-MacBook-Pro CICD Gitlab %
```

- Check if it has been successfully pushed and built on Gitlab

The screenshot displays the GitLab CI/CD interface. On the left, a sidebar shows the project structure with 'Jobs' selected. The main area shows the job output for 'deploy-ke-aws'. The output is a log of commands and their results, including Docker image pulling, Git repository initialization, and a successful push to the 'master' branch. The job status is 'Succeeded'.

Status	Pipeline	Created by	Stages	Actions
Passed 00:00:11 1 minute ago	Update .gitlab-ci.yml untuk build stage #1876681775 latest branch		✓	

- Successfully pushed and built on gitlab

THANK YOU