Lab1：CI/CD Pipeline Overview– Jenkins & GitLab

1. Installation and First Test

(1) Install docker

sudo apt update

sudo apt install -y docker.io

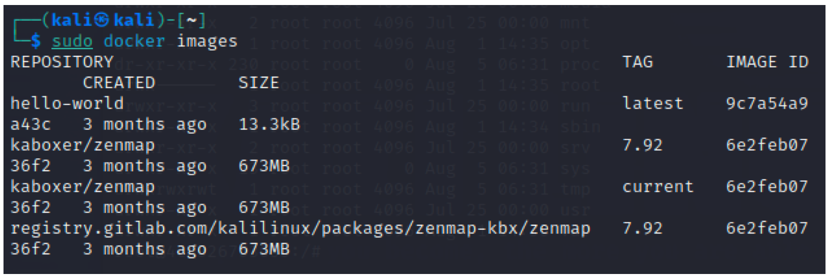
sudo systemctl enable docker --now

(此命令令為了不必每次打 sudo) sudo usermod -aG docker $USER

(然後登出後，重新登入)

(2) 查看 docker images

sudo docker images



```
┌──(kali㉿kali)-[~]
└─$ sudo docker images
REPOSITORY                                           TAG       IMAGE ID
       CREATED        SIZE
hello-world                                          latest    9c7a54a9
a43c    3 months ago    13.3kB
kaboxer/zenmap                                       7.92      6e2feb07
36f2    3 months ago    673MB
kaboxer/zenmap                                       current   6e2feb07
36f2    3 months ago    673MB
registry.gitlab.com/kalilinux/packages/zenmap-kbx/zenmap   7.92      6e2feb07
36f2    3 months ago    673MB
```

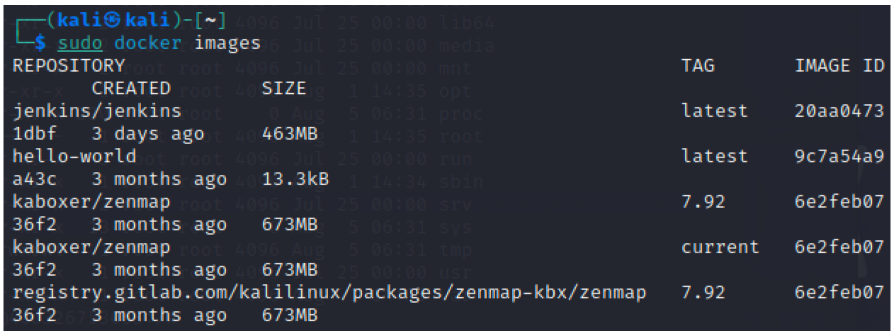
(3) 安裝 Jenkins docker

sudo docker pull jenkins/jenkins

sudo docker run -u 0 -p 8080:8080 -p 50000:50000 -v /pack/jenkins:/var/jenkins_home jenkins/jenkins

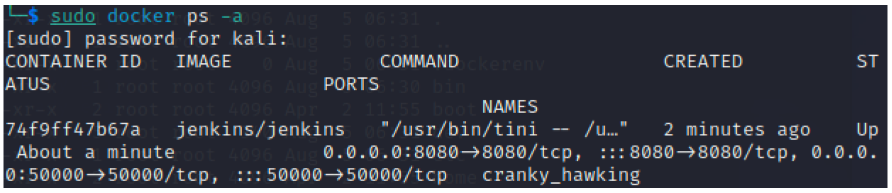
(或使用 docker container run)

查看 docker images



```
┌──(kali㉿kali)-[~]
└─$ sudo docker images
REPOSITORY                                                 TAG       IMAGE ID
       CREATED        SIZE
jenkins/jenkins                                            latest    20aa0473
1dbf    3 days ago      463MB
hello-world                                                latest    9c7a54a9
a43c    3 months ago    13.3kB
kaboxer/zenmap                                             7.92      6e2feb07
36f2    3 months ago    673MB
kaboxer/zenmap                                             current   6e2feb07
36f2    3 months ago    673MB
registry.gitlab.com/kalilinux/packages/zenmap-kbx/zenmap   7.92      6e2feb07
36f2    3 months ago    673MB
```

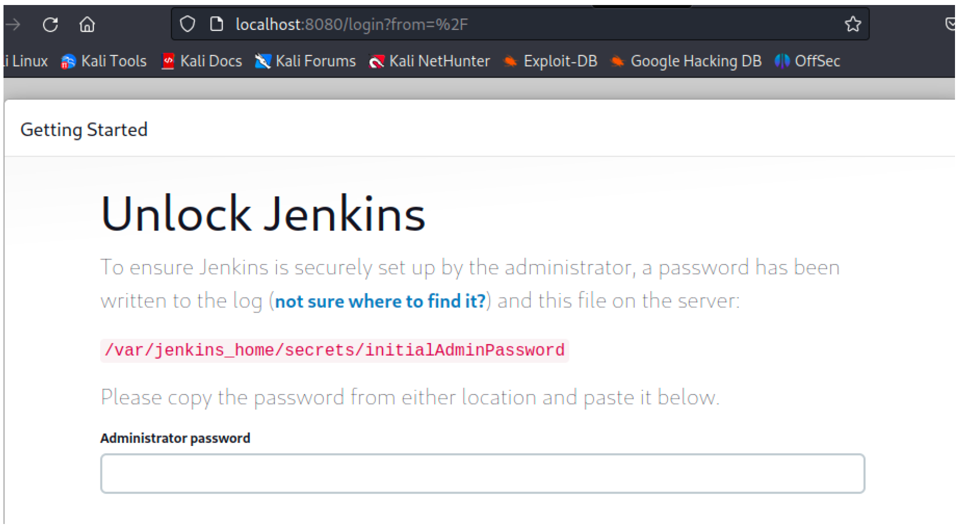
查看執行 (或使用 docker container ls -a)



```
└─$ sudo docker ps -a
[sudo] password for kali:
CONTAINER ID   IMAGE             COMMAND                  CREATED          ST
ATUS            PORTS
                     NAMES
74f9ff47b67a   jenkins/jenkins   "/usr/bin/tini -- /u…"   2 minutes ago    Up
 About a minute   0.0.0.0:8080→8080/tcp, :::8080→8080/tcp, 0.0.0.
0:50000→50000/tcp, :::50000→50000/tcp   cranky_hawking
```

(4) 執行 Jenkins

開啟瀏覽器，執行 localhost:8080



localhost:8080/login?from=%2F

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

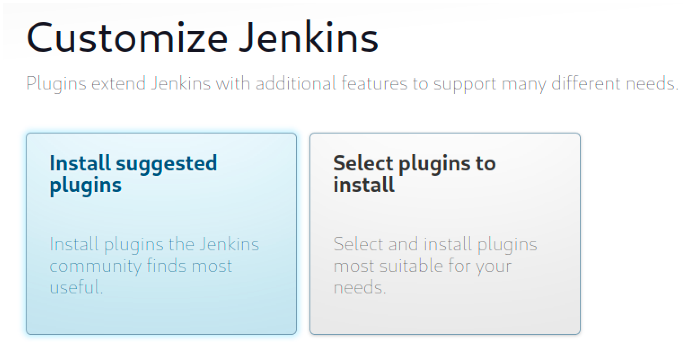
```
/var/jenkins_home/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

解鎖 Jenkins，請查詢 sudo cat /pack/jenkins/secrets/initialAdminPassword

(5) Install (suggested) plugins

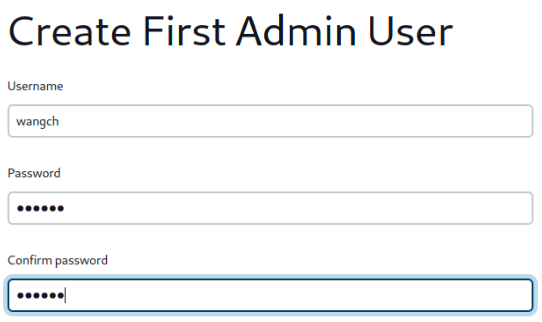


Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.

建立第一個 Admin User



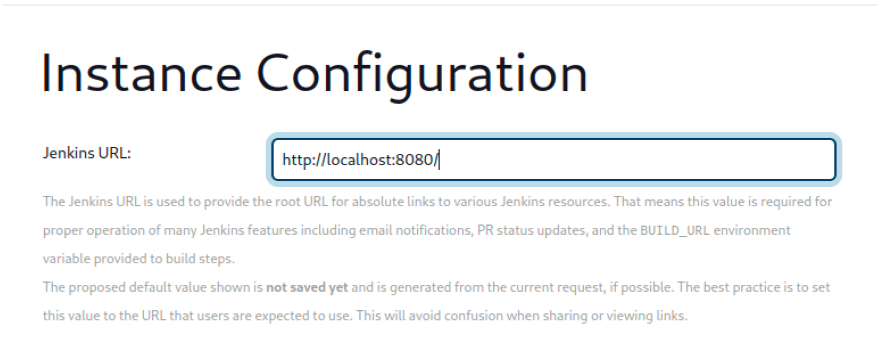
Create First Admin User

Username: wangch

Password: ••••••

Confirm password: ••••••

Instance Configuration



Instance Configuration

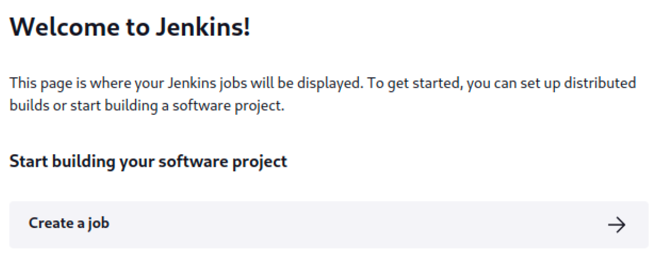
Jenkins URL: http://localhost:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

(6) 簡單使用 freestyle 回應資料

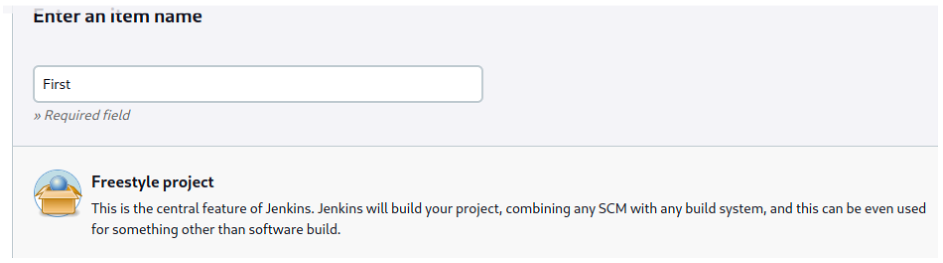
Create a job



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project



Enter an item name

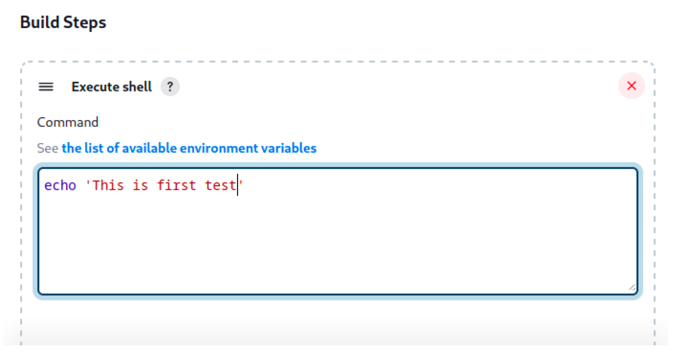
First

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Build Steps 執行簡單的 shell



Build Steps

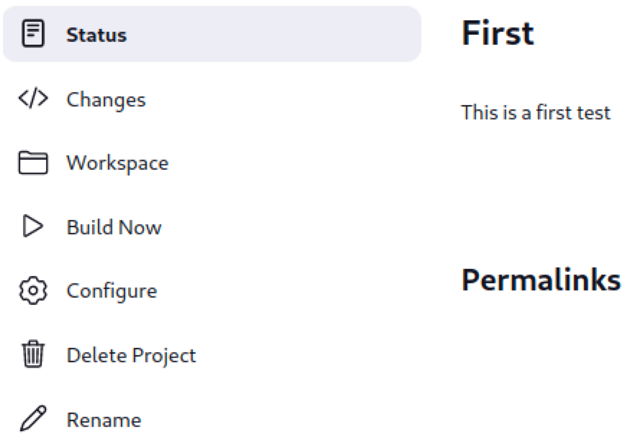
Execute shell

Command

See the list of available environment variables

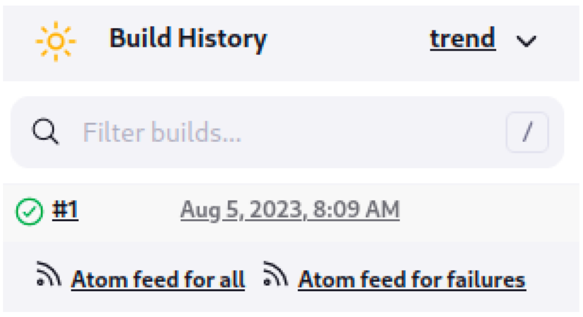
```
echo 'This is first test'
```

(7) 執行 Build Now

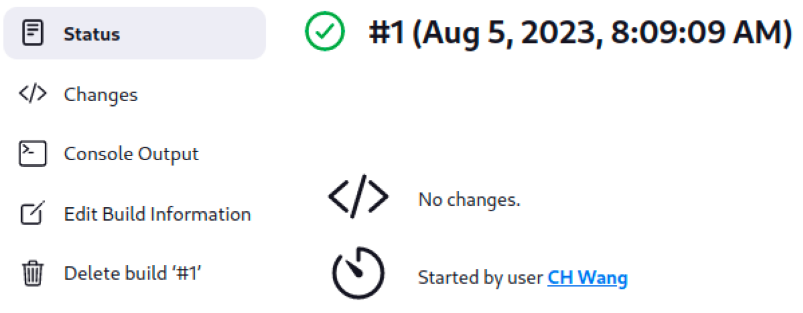


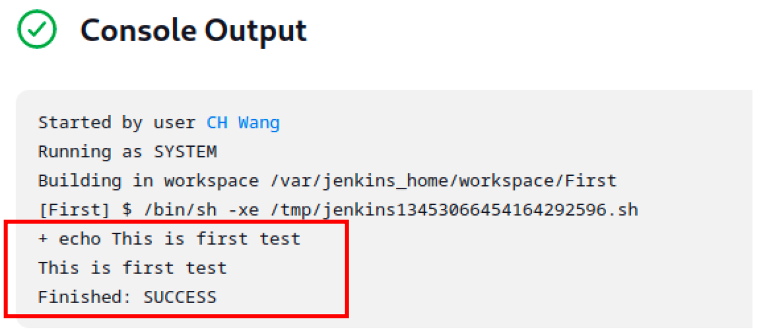
This is a first test

(8) 執行成功



觀看 Console Output





```
Started by user CH Wang
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/First
[First] $ /bin/sh -xe /tmp/jenkins13453066454164292596.sh
+ echo This is first test
This is first test
Finished: SUCCESS
```

2. Simple Demo for Pipeline

(1) 製作一個具有 gcc 以及 flawfinder 的 Jenkins docker

jgcc2 的 檔案內容

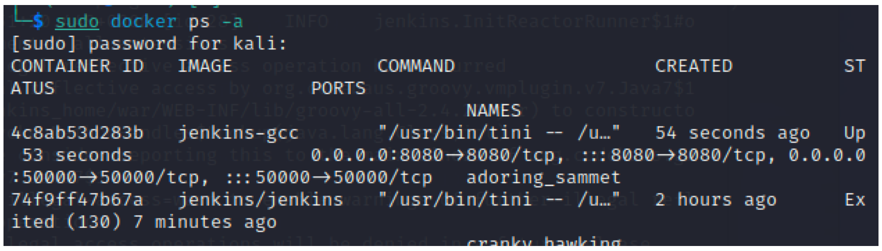
```
FROM jenkins/jenkins
USER root
RUN apt-get update && apt-get install -y build-essential
RUN apt-get install -y flawfinder
USER jenkins
```

執行建置

```
sudo docker build -t jenkins-gcc -f jgcc2 .
```

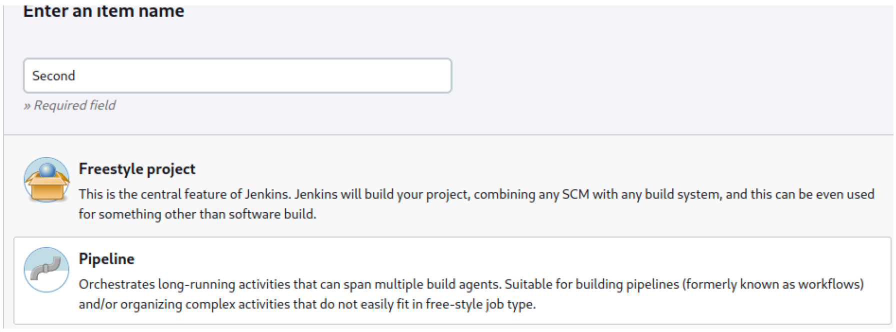
啟動 docker

```
sudo docker run -u 0 -p 8080:8080 -p 50000:50000 -v
 /pack/jenkins:/var/jenkins_home jenkins-gcc
```



```
└─$ sudo docker ps -a
[sudo] password for kali:
CONTAINER ID   IMAGE            COMMAND                  CREATED          ST
ATUS                    PORTS
                                            NAMES
4c8ab53d283b   jenkins-gcc      "/usr/bin/tini -- /u…"   54 seconds ago   Up
 53 seconds            0.0.0.0:8080→8080/tcp, :::8080→8080/tcp, 0.0.0.0
:50000→50000/tcp, :::50000→50000/tcp   adoring_sammet
74f9ff47b67a   jenkins/jenkins  "/usr/bin/tini -- /u…"  2 hours ago      Ex
ited (130) 7 minutes ago
                                            cranky_hawking
```

(2) 開啟 Jenkin，使用 Pipeline



Enter an item name

Second

» Required field

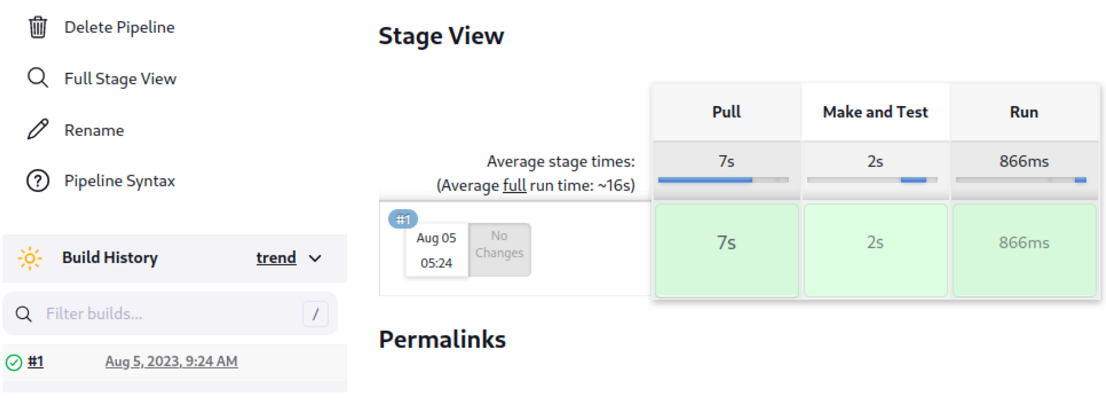
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can 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.

(3) 創建 pipeline script

```
pipeline {
    agent any

    stages {
            stage('Pull') {
                steps {
                    git 'https://github.com/wangch64/test1.git'
                }
            }
            stage('Make and Test') {
                steps {
                    sh '''flawfinder my2.c
                    flawfinder te2.c
                    gcc my2.c -o my2
                    gcc te2.c -o te2'''
                }
            }
            stage('Run') {
                steps {
                    sh './my2'
                    sh './te2'
                }
            }
        }
    }
```

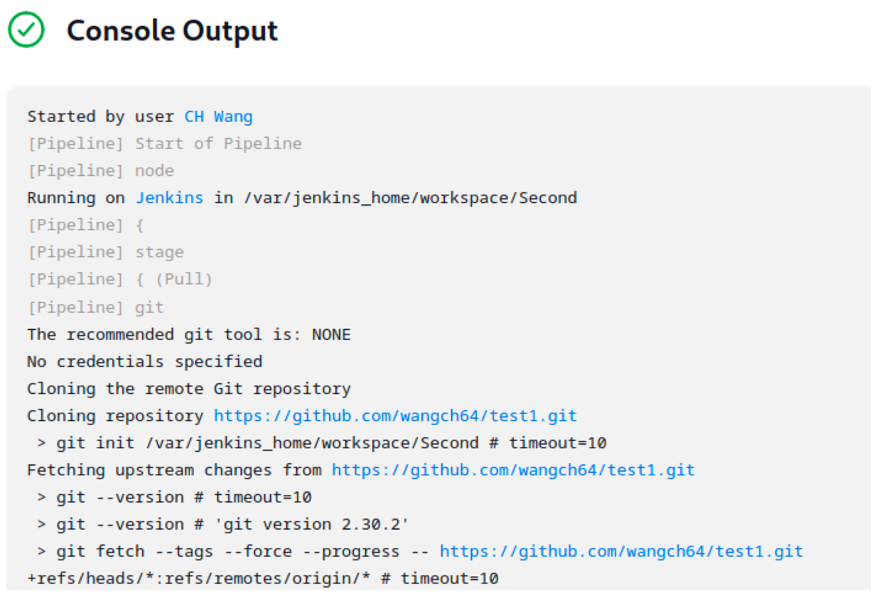
(4) Build Now (see Stage View)



#1 Aug 5, 2023, 9:24 AM

	Pull	Make and Test	Run
Average stage times: (Average full run time: ~16s)	7s	2s	866ms
#1 Aug 05 05:24 No Changes	7s	2s	866ms

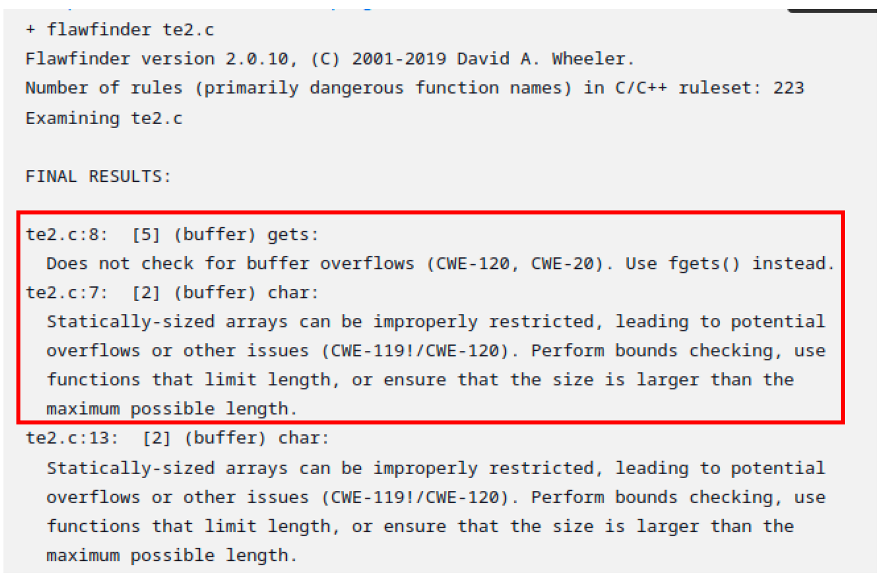
(5) See Console Output



Console Output

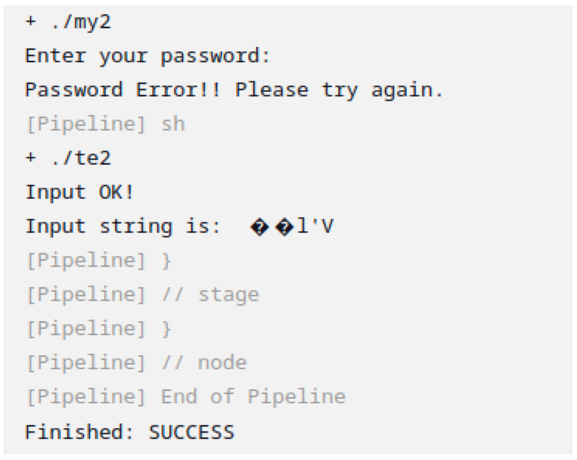
```
Started by user CH Wang
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/Second
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Pull)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/wangch64/test1.git
 > git init /var/jenkins_home/workspace/Second # timeout=10
Fetching upstream changes from https://github.com/wangch64/test1.git
 > git --version # timeout=10
 > git --version # 'git version 2.30.2'
 > git fetch --tags --force --progress -- https://github.com/wangch64/test1.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
```

程式碼測試結果



```
+ flawfinder te2.c
Flawfinder version 2.0.10, (C) 2001-2019 David A. Wheeler.
Number of rules (primarily dangerous function names) in C/C++ ruleset: 223
Examining te2.c

FINAL RESULTS:

te2.c:8:  [5] (buffer) gets:
  Does not check for buffer overflows (CWE-120, CWE-20). Use fgets() instead.
te2.c:7:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
te2.c:13:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
```

執行結果



```
+ ./my2
Enter your password:
Password Error!! Please try again.
[Pipeline] sh
+ ./te2
Input OK!
Input string is:  ��l'V
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

3. **Python flask Demo for Pipeline**

(1) 建構 dockerfile: jgcc3

```
FROM jenkins/jenkins
USER root
RUN apt-get update && apt-get install -y build-essential
RUN apt-get install -y flawfinder
RUN apt-get install -y python3-pip
RUN pip install flask
USER jenkins
```

(2) 建立 docker image

```
sudo docker build -t jenkins-py -f jgcc3 .
```

(3) 執行 docker

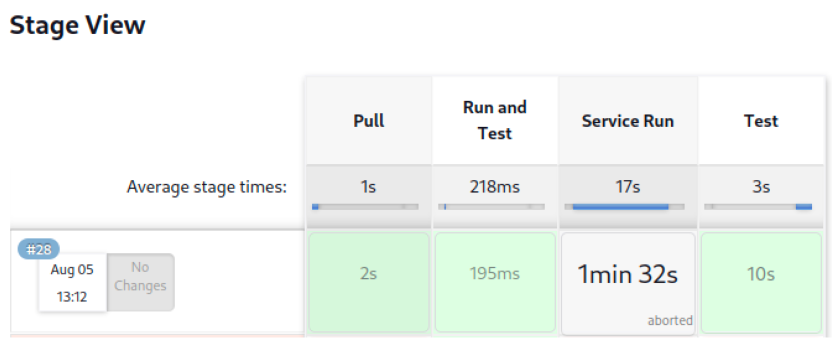
```
sudo docker run -u 0 -p 8080:8080 -p 50000:50000 -p 5000:5000 -v
/pack/jenkins:/var/jenkins_home jenkins-py
```

(4) 建置 pipeline 程式

```
pipeline {
        agent any
        stages {
                stage('Pull') {
                        steps {
                                sh 'if [ -d "flask-test" ]; then rm -R flask-test; fi'
                                sh 'git clone https://github.com/wangch64/flask-test.git'
                        }
                }
                stage('Run and Test') {
                        parallel {
                                stage('Service Run') {
                                        steps {
                                                sh 'nohup python3 flask-test/myform.py'
                                        }
                                }
                                stage('Test') {
                                        steps {
```

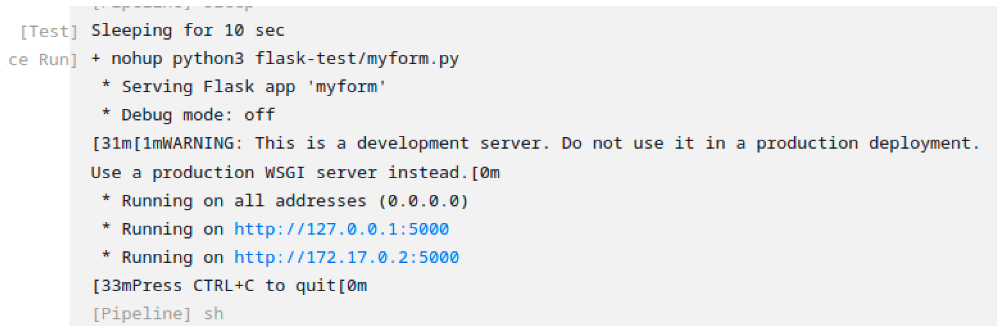
```
                        sleep(10)
                        sh 'curl localhost:5000/myform'
                    }
                }
            }
        }
    }
}
```

(5) Build Now (see Stage View)

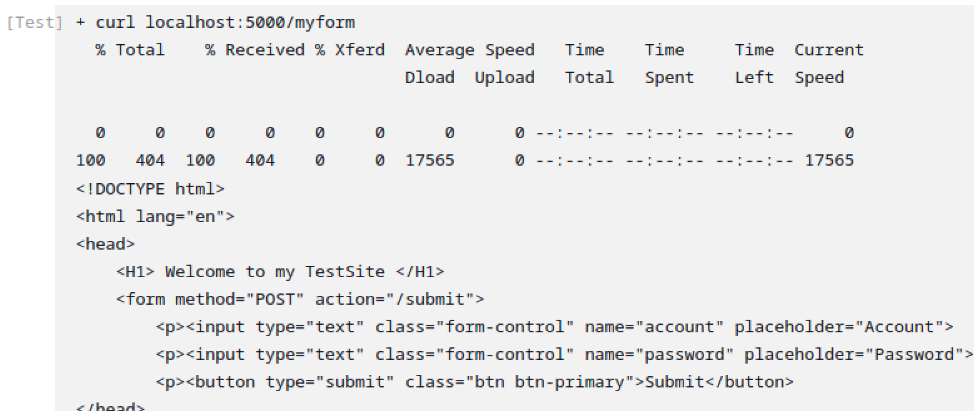

Stage View

	Pull	Run and Test	Service Run	Test
Average stage times:	1s	218ms	17s	3s
#28 Aug 05 13:12 No Changes	2s	195ms	1min 32s aborted	10s

(6) Console Output

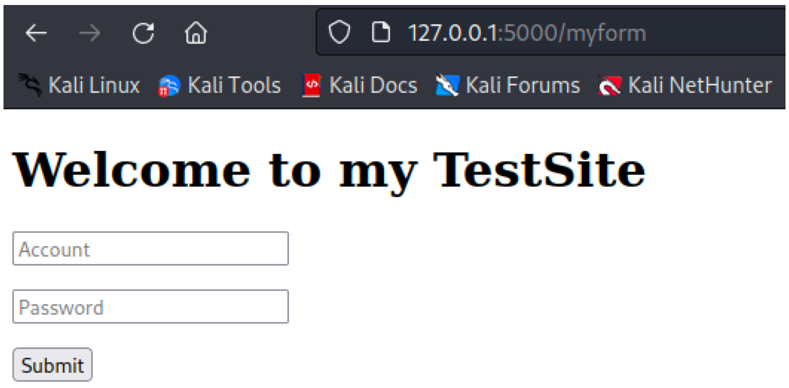


```
  [Test] Sleeping for 10 sec
ce Run] + nohup python3 flask-test/myform.py
         * Serving Flask app 'myform'
         * Debug mode: off
        [31m[1mWARNING: This is a development server. Do not use it in a production deployment.
        Use a production WSGI server instead.[0m
         * Running on all addresses (0.0.0.0)
         * Running on http://127.0.0.1:5000
         * Running on http://172.17.0.2:5000
        [33mPress CTRL+C to quit[0m
        [Pipeline] sh
```



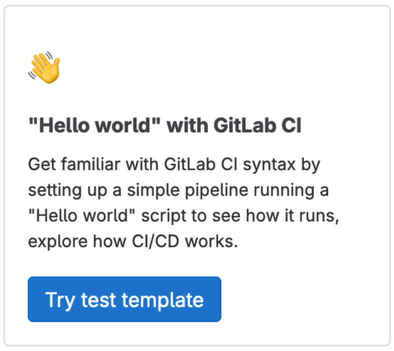
```
[Test] + curl localhost:5000/myform
         % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                        Dload  Upload   Total   Spent    Left  Speed

         0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
       100   404  100   404    0     0  17565      0 --:--:-- --:--:-- --:--:-- 17565
       <!DOCTYPE html>
       <html lang="en">
       <head>
           <H1> Welcome to my TestSite </H1>
           <form method="POST" action="/submit">
               <p><input type="text" class="form-control" name="account" placeholder="Account">
               <p><input type="text" class="form-control" name="password" placeholder="Password">
               <p><button type="submit" class="btn btn-primary">Submit</button>
       </head>
```

(7) 啟動 Flask app

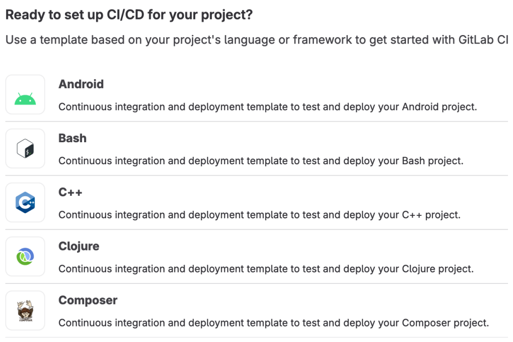

127.0.0.1:5000/myform
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter

Welcome to my TestSite

4. **GitLab CI/CD Pipeline**
   (1) **Register at GitLab**
   (2) **Create a new group**
   (3) **Create a new project**
   (4) **Use sample pipeline yml file (test template)**


"Hello world" with GitLab CI

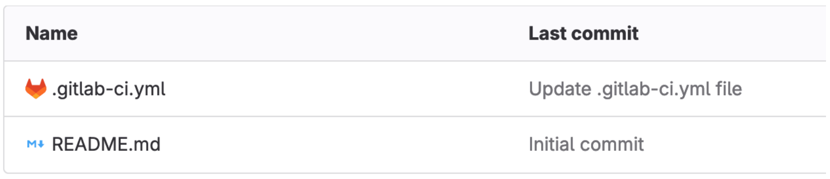
Get familiar with GitLab CI syntax by setting up a simple pipeline running a "Hello world" script to see how it runs, explore how CI/CD works.


Ready to set up CI/CD for your project?

Use a template based on your project's language or framework to get started with GitLab CI

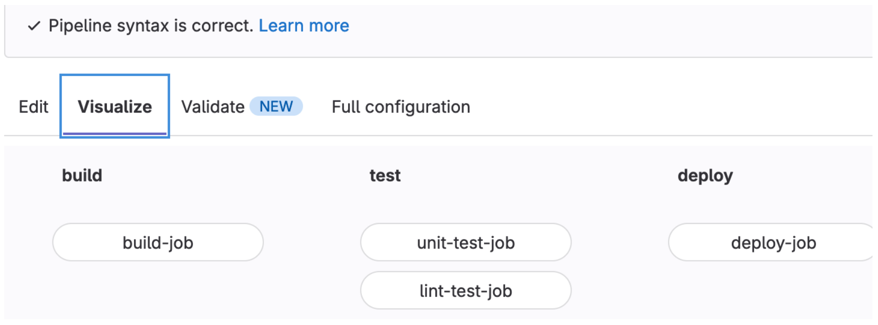
Android	Continuous integration and deployment template to test and deploy your Android project.
Bash	Continuous integration and deployment template to test and deploy your Bash project.
C++	Continuous integration and deployment template to test and deploy your C++ project.
Clojure	Continuous integration and deployment template to test and deploy your Clojure project.
Composer	Continuous integration and deployment template to test and deploy your Composer project.

   (5) **Create a gitlab-ci.yml**

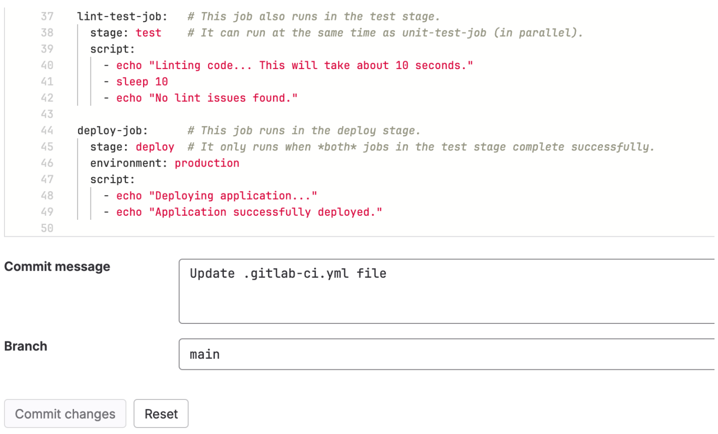


Name	Last commit
.gitlab-ci.yml	Update .gitlab-ci.yml file
README.md	Initial commit

   (6) **Build -> Pipeline editor**


✓ Pipeline syntax is correct. Learn more

(7) Commit and Run

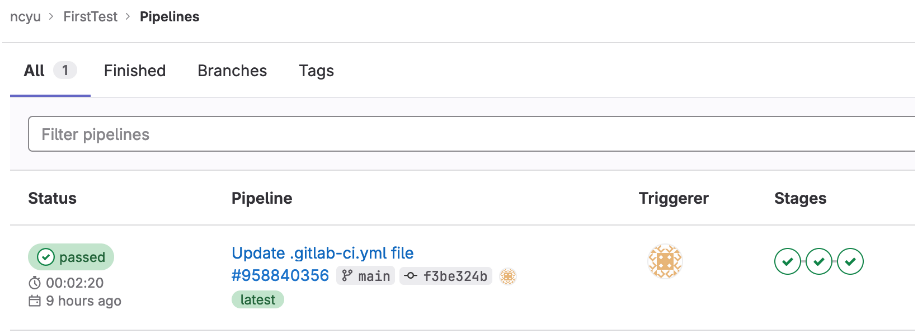


```
lint-test-job:   # This job also runs in the test stage.
  stage: test    # It can run at the same time as unit-test-job (in parallel).
  script:
    - echo "Linting code... This will take about 10 seconds."
    - sleep 10
    - echo "No lint issues found."

deploy-job:      # This job runs in the deploy stage.
  stage: deploy  # It only runs when *both* jobs in the test stage complete successfully.
  environment: production
  script:
    - echo "Deploying application..."
    - echo "Application successfully deployed."

```

Commit message: Update .gitlab-ci.yml file

Branch: main

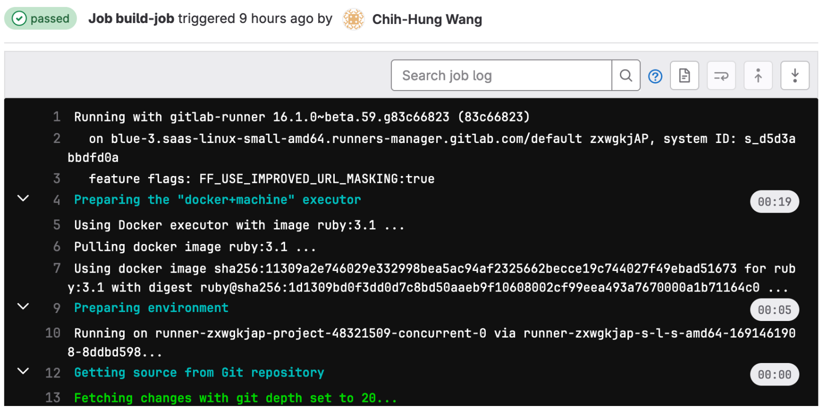
(8) See Pipelines



ncyu > FirstTest > Pipelines

Status	Pipeline	Triggerer	Stages
passed 00:02:20 9 hours ago	Update .gitlab-ci.yml file #958840356 main f3be324b latest		✓ ✓ ✓

(9) See Job Details



passed **Job build-job** triggered 9 hours ago by **Chih-Hung Wang**

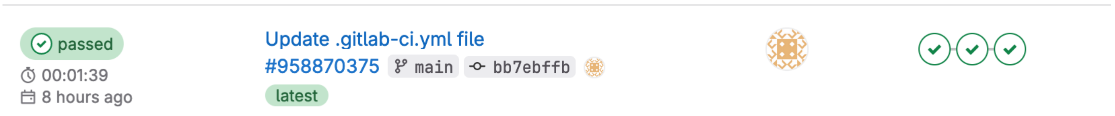
```
Running with gitlab-runner 16.1.0~beta.59.g83c66823 (83c66823)
  on blue-3.saas-linux-small-amd64.runners-manager.gitlab.com/default zxwgkjAP, system ID: s_d5d3a
bbdfd0a
  feature flags: FF_USE_IMPROVED_URL_MASKING:true
Preparing the "docker+machine" executor                                                   00:19
Using Docker executor with image ruby:3.1 ...
Pulling docker image ruby:3.1 ...
Using docker image sha256:11309a2e746029e332998bea5ac94af2325662becce19c744027f49ebad51673 for rub
y:3.1 with digest ruby@sha256:1d1309bd0f3dd0d7c8bd50aaeb9f10608002cf99eea493a7670000a1b71164c0 ...
Preparing environment                                                                     00:05
Running on runner-zxwgkjap-project-48321509-concurrent-0 via runner-zxwgkjap-s-l-s-amd64-169146190
8-8ddbd598...
Getting source from Git repository                                                        00:00
Fetching changes with git depth set to 20...
```

5. GitLab CI/CD Pipeline Using C++ Template

(1) Modify the C++ Template

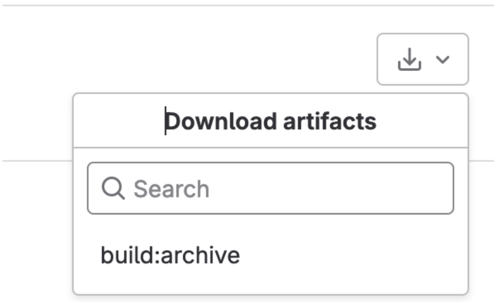
```
# You can copy and paste this template into a new `.gitlab-ci.yml` file.
# You should not add this template to an existing `.gitlab-ci.yml` file
by using the `include:` keyword.
#
# To contribute improvements to CI/CD templates, please follow the
Development guide at:
# https://docs.gitlab.com/ee/development/cicd/templates.html
# This specific template is located at:
# https://gitlab.com/gitlab-org/gitlab/-
/blob/master/lib/gitlab/ci/templates/C++.gitlab-ci.yml

# use the official gcc image, based on debian
# can use versions as well, like gcc:5.2
# see https://hub.docker.com/_/gcc/

image: gcc

build:
  stage: build
  # instead of calling g++ directly you can also use some build toolkit
like make
  # install the necessary build tools when needed
  # before_script:
  #   - apt update && apt -y install make autoconf
  script:
    - gcc te2.c -o mybinary
  artifacts:
    paths:
      - mybinary
      # depending on your build setup it's most likely a good idea to
cache outputs to reduce the build time
      # cache:
      #   paths:
      #     - "*.o"

# run tests using the binary built before
test:
  stage: test
  script:
    - ./mybinary|grep "Input string is"

deploy:
  stage: deploy
  script: echo "Define your deployment script!"
  environment: production
```

(2) Run the pipleline



passed
00:01:39
8 hours ago
Update .gitlab-ci.yml file
#958870375 main bb7ebffb
latest

(3) Download the artifacts



6. GitLab CI/CD Pipeline Sample Project

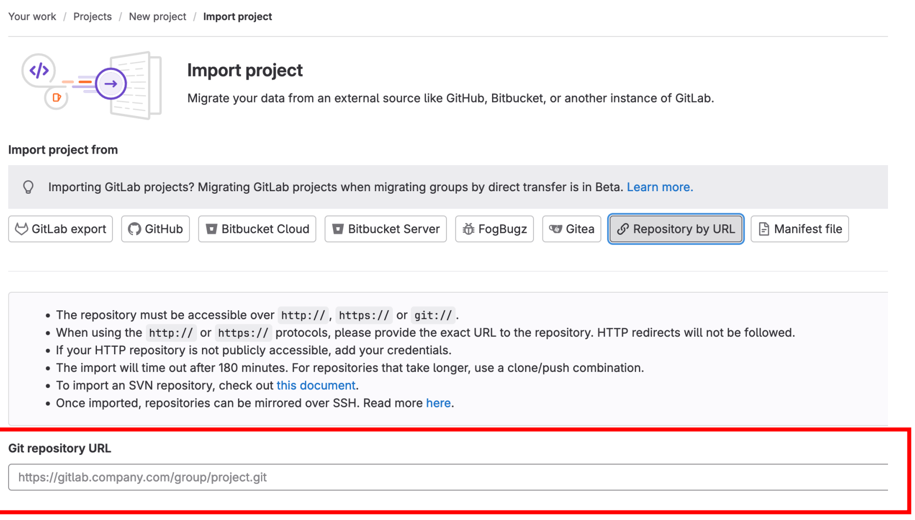
(1) Please connect to: https://gitlab.com/gitlab-learn-labs/sample-projects/quick-devsecops-hands-on-workshop/

(2) Clone to your repository

(a) New Project/Repository

(b) Import Project

(c) Repository by URL



Your work / Projects / New project / Import project

Import project

Migrate your data from an external source like GitHub, Bitbucket, or another instance of GitLab.

Import project from

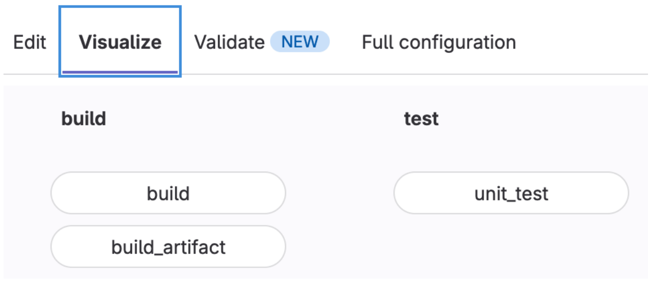
Importing GitLab projects? Migrating GitLab projects when migrating groups by direct transfer is in Beta. Learn more.

- The repository must be accessible over `http://`, `https://` or `git://`.
- When using the `http://` or `https://` protocols, please provide the exact URL to the repository. HTTP redirects will not be followed.
- If your HTTP repository is not publicly accessible, add your credentials.
- The import will time out after 180 minutes. For repositories that take longer, use a clone/push combination.
- To import an SVN repository, check out this document.
- Once imported, repositories can be mirrored over SSH. Read more here.

Git repository URL

https://gitlab.company.com/group/project.git

(3) Try to create a pipeline script…


Edit
Visualize
Validate NEW
Full configuration
build
test
build
unit_test
build_artifact