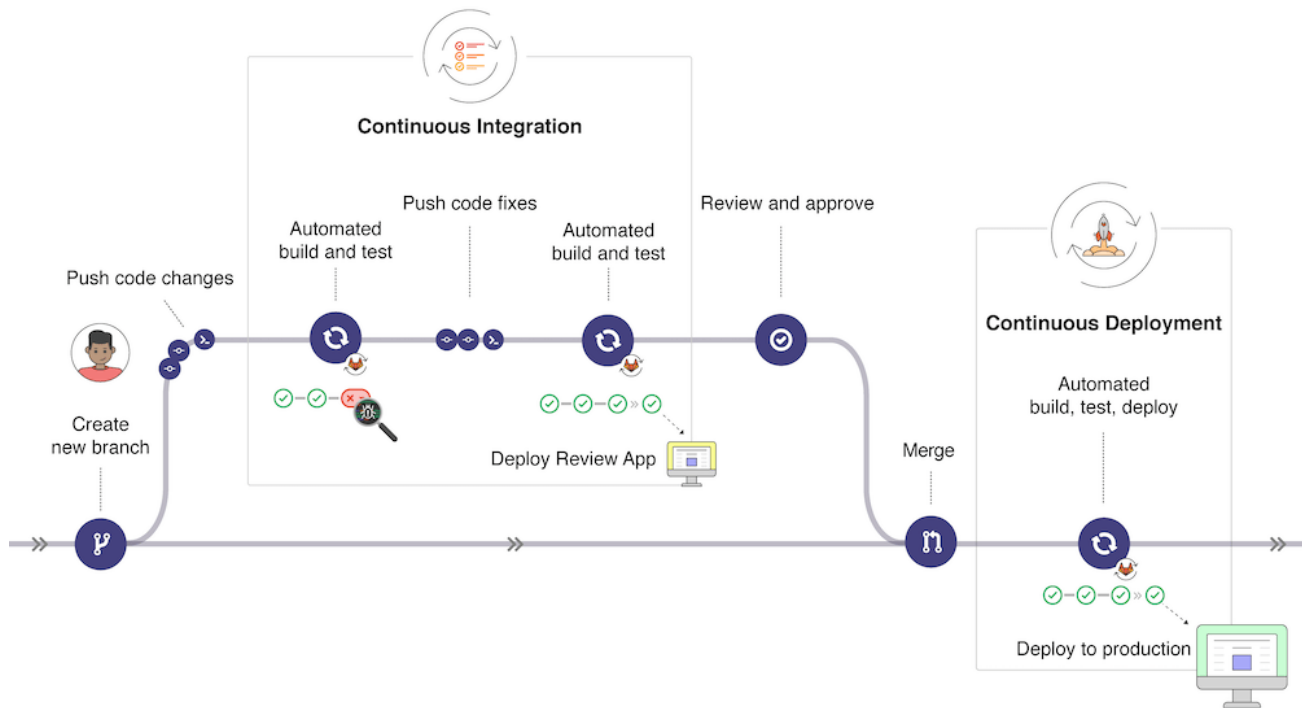# Introduction of GitLAB CI

GitLab CI Key Concept

- GitLab CI Workflow

- Pipelines are the fundamental building blocks for CI/CD in GitLab, includes:
  - Jobs, which define **what** to do.
  - Stages, which define **when** to run the jobs.
- Pipelines is described by `.gitlab-ci.yml`(or other *.yml and can be configed in gitlab) file.
- GitLab Runner is an application that works with GitLab CI/CD to run jobs in a pipeline.

Create GitLab Project

- Run a local GitLab from docker
  - https://gitlab.icelab.lenovo.com/ issue.1: Gitlab runner register error: - x509: certificate...
- Create a project (Test code is from HR650X v2)

New project › Create blank project

Create blank project

Create a blank project to house your files, plan your work, and collaborate on code, among other things.

Project name

Wv2

Project URL

http://localhost/ | root ▾

Project slug

Wv2

Want to house several dependent projects under the same namespace? Create a group.

Project description (optional)

Description format

Visibility Level ⑦

○ 🔒 Private
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.

○ 🛡 Internal
The project can be accessed by any logged in user except external users.

○ 🌐 Public
The project can be accessed without any authentication.

☐ **Initialize repository with a README**
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

[Create project]   [Cancel]

GitLab | Projects ∨ | Groups ∨ | More ∨ | Search or jump to...

Administrator > Wv2

W **Wv2** 🔒
Project ID: 35

🔔∨ | ☆ Star | 0

The repository for this project is empty

You can get started by cloning the repository or start adding files to it with one of the following options.

[Clone ∨] [⊞ New file] [⊞ Add README] [⊞ Add LICENSE] [⊞ Add CHANGELOG] [⊞ Add CONTRIBUTING]

Command line instructions

You can also upload existing files from your computer using the instructions below.

Git global setup

```
git config --global user.name "Administrator"
git config --global user.email "admin@example.com"
```

Create a new repository

```
git clone git@localhost:root/Wv2.git
cd Wv2
touch README.md
git add README.md
git commit -m "add README"
git push -u origin master
```

Push an existing folder

```
cd existing_folder
git init
git remote add origin git@localhost:root/Wv2.git
git add .
git commit -m "Initial commit"
git push -u origin master
```

Push an existing Git repository

```
cd existing_repo
git remote rename origin old-origin
git remote add origin git@localhost:root/Wv2.git
git push -u origin --all
git push -u origin --tags
```

- Init project with project code

Setup GitLab Runner

- **Install on Windows**


```
> ls
```

```
    目录: C:\Users\efika\Desktop\gitlab\ci\runner


Mode                 LastWriteTime        Length Name
----                 -------------        ------ ----
-a----        2021/2/24     14:33      41490944 gitlab-runner.exe
```

- **Register a runner**, URL and token refer to `Settings > CI / CD > Runners`.

```
❯ .\gitlab-runner.exe register
Runtime platform    arch=amd64 os=windows pid=34136 revision=2ebc4dc4
version=13.9.0

Enter the GitLab instance URL (for example, https://gitlab.com/):
http://localhost/

Enter the registration token:
Fxy9zwfX4YsQzDytj_hG

Enter a description for the runner:
[HSuEFI-L3A209-LiuYi28]: build

Enter tags for the runner (comma-separated):
build

Registering runner... succeeded                     runner=Fxy9zwfX

Enter an executor: shell, ssh,... custom:
shell

Runner registered successfully.
❯ ls


    目录: C:\Users\efika\Desktop\gitlab\ci\runner


Mode                 LastWriteTime        Length Name
----                 -------------        ------ ----
-a----         2021/3/9     13:57           332 config.toml
-a----        2021/2/24     14:33      41490944 gitlab-runner.exe
```

- Install GitLab Runner as a service and start it

```
❯ .\gitlab-runner.exe install
❯ .\gitlab-runner.exe start
```

Setup GitLab Pipeline

- Add a `.gitlab-ci.yml` to the root of project code, and submit it.

```yaml
stages:
  - build
  - test

job1:
  stage: build
  script:
    - C:\Users\efika\Desktop\gitlab\ci\tools\AptioV\AptioV.bat
  only:
    - master
  tags:
    - build
  artifacts:
    paths:
      - xge*_*.bin
```

```
::C:\Users\efika\Desktop\gitlab\ci\tools\AptioV\AptioV.bat
@SET WORKSPACE=%CI_PROJECT_DIR%
@SET _APTIO_V_=%~dp0\Aptio_5.x_TOOLS_40
@SET MSVC_HOST=%~dp0\WDDK\7600.16385.1\bin\x86
@SET EWDK_HOME=%~dp0\EWDK
@SET PYTHON_II_HOME=C:\Efika\PYii
::+ Standard env for VEB
@SET CCX86DIR=%MSVC_HOST%\x86
@SET CCX64DIR=%MSVC_HOST%\amd64
@SET EWDK_DIR=%EWDK_HOME%
@SET TOOLS_DIR=%_APTIO_V_%\BuildTools
::----------------------
@SET VEB=WhitleyHR630XV2
@SET PYTHON_COMMAND=%PYTHON_II_HOME%\python.exe
::----------------------
```

```
@SET PATH=%TOOLS_DIR%;%PYTHON_II_HOME%;%PATH%
::- Standard env for VEB
CD /D %WORKSPACE% && make.exe rebuild
@CMD /K
```
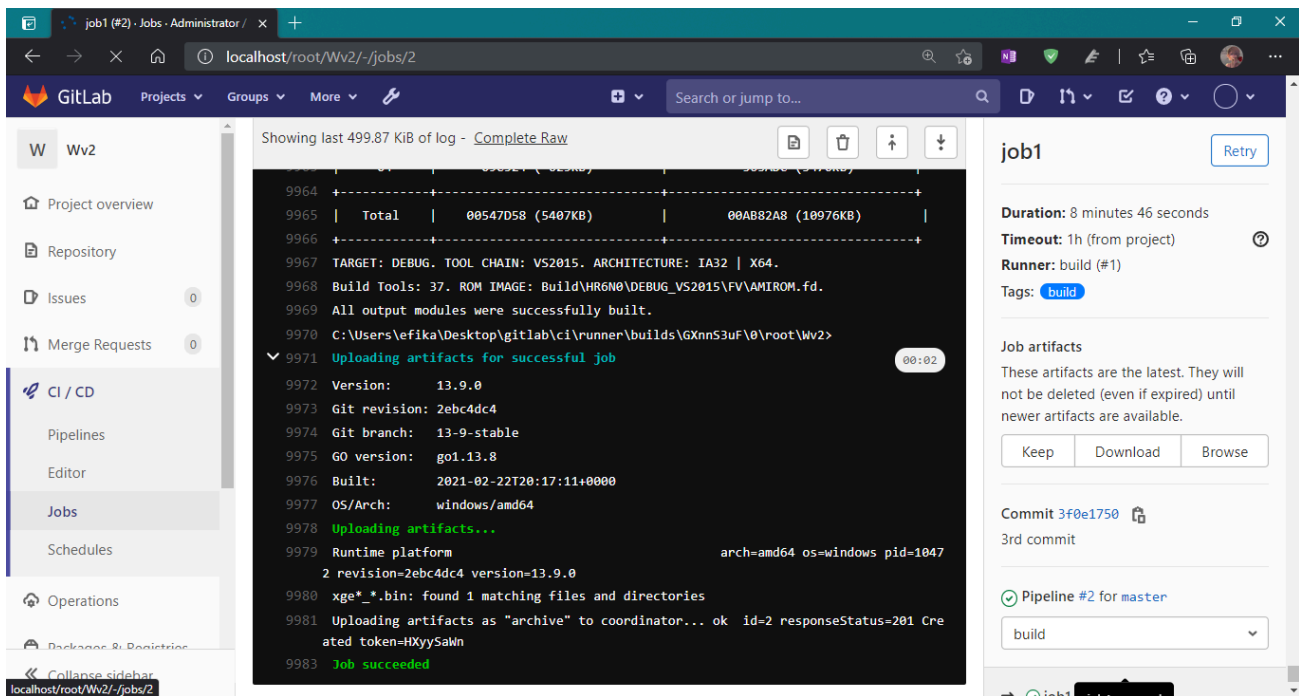
- Then the CI process is running.
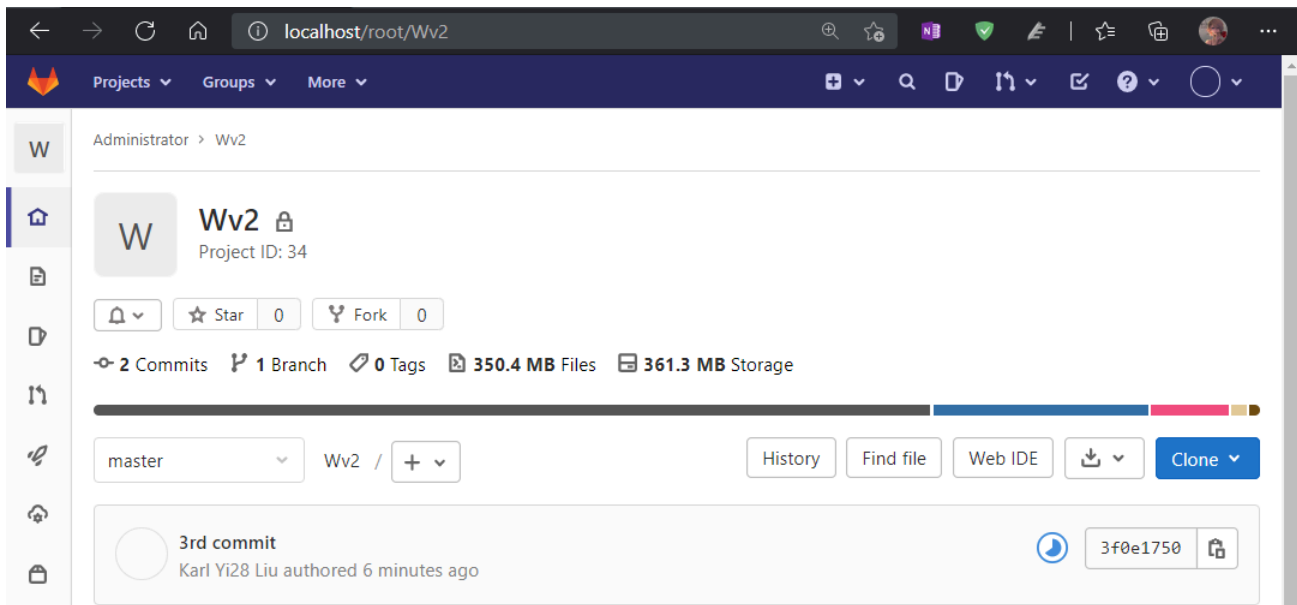
- After pipeline done, the artifacts can be downloaded from `CI/CD > PipeLines` Page.
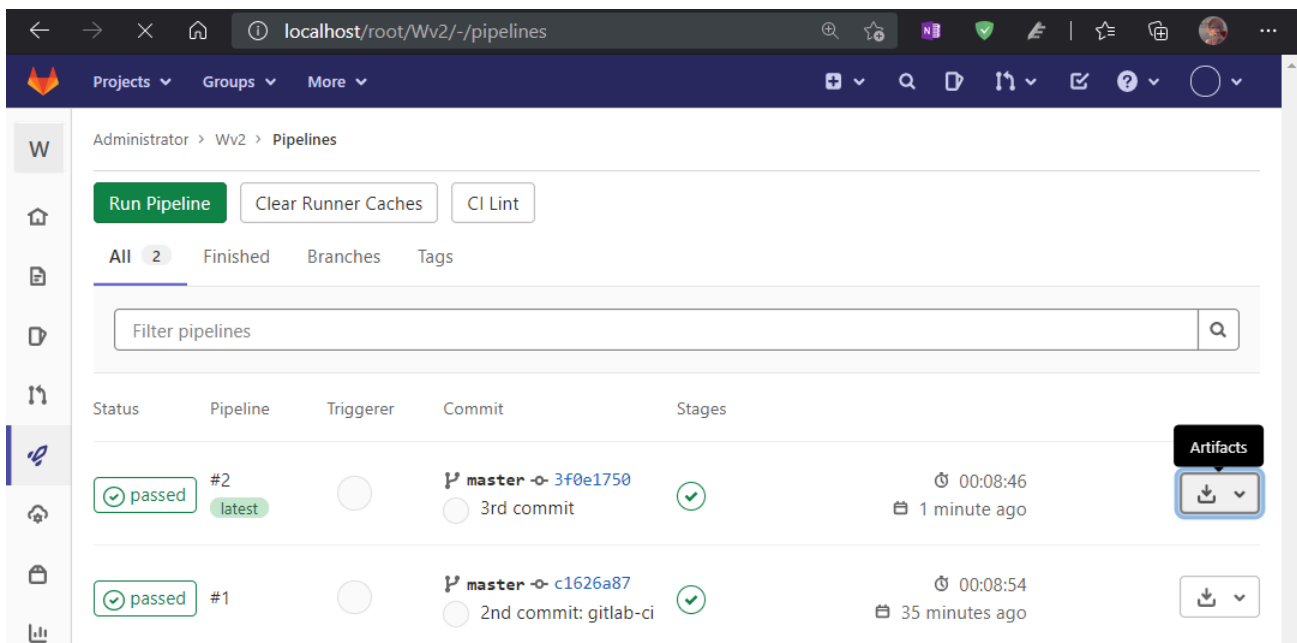
- Checking pipeline process and result log by click into it.

现存的问题

问题：`GitLab`服务器的防火墙限制了`Gitlab Runner (CI)` 的使用。

因为每一次提交的时间具有不确定性，防火墙的会话有效时间是一个小时，假设代码提交的机器并非`Runner`所在的机器，那么此时`Runner`就会运行失败，至少`Runner`上传给`GitLab`的内容会受阻。

解决`A`，以下二者之一：

- 为防火墙添加`IP`白名单功能。
- 将防火墙登录的机制接口化，提供`API`，让`Runner`可以按需登录防火墙，此时`CI`需要专用账户和密码。

解决`B`：

- 代码编译提交的环境与`Runner`环境共用。