

Olivier RECHER
François TOUREILLE

2A INFRES DL

TP CI/CD N°2

Fait le 05/01/2026

Démarche

On est parti sur 3 stages différents qui sont joués une fois que le précédent est fini. L'ordre choisi a été **tests > minification > deploy**, car les tests ont le double intérêt de vérifier le code avant de procéder à d'autres traitements, ainsi que d'être **joués** sur tous les environnements.

1) Mise en place du projet

On a récupéré le projet fournit et avons créer un nouveau projet gitlab dans lequel nous allons mettre en place notre système de CI/CD.

The screenshot shows the 'Python Project' page in GitLab. At the top, there's a header with the project name and a 'main' branch selector. Below this, a commit message 'ajout des fichiers' by 'FrancoisTourelle' is shown. A table lists the files in the project:

Name	Last commit	Last update
static	ajout des fichiers	12 seconds ago
templates	ajout des fichiers	12 seconds ago
tests	ajout des fichiers	12 seconds ago
.gitignore	ajout des fichiers	12 seconds ago
README.md	Initial commit	4 minutes ago
conf.py	ajout des fichiers	12 seconds ago
hello.py	ajout des fichiers	12 seconds ago
requirements.txt	ajout des fichiers	12 seconds ago

On the right, 'Project information' shows 1 Commit, 1 Branch, 0 Tags, and 4 KIB Project Storage. It also lists integrations like README, Auto DevOps, and various add-ons.

Création de la CI sur un autre dépôt pour ne pas être embêté par les versions de [gitlab-ci.yml](#) selon les branches.

The screenshot shows the 'python-ci / .gitlab-ci.yml' page. A notification at the top says 'The file has been successfully created.' Below, the file name '.gitlab-ci.yml' is shown with a commit hash 'd254027a'. The file content area is empty, with the text 'Empty file' at the bottom.

General pipelines

Customize your pipeline configuration.










- ☒ Project-based pipeline visibility
Allow access to pipelines and job details, including output logs and artifacts, based on your project's visibility settings. ⓘ
- ☒ Auto-cancel redundant pipelines
Pipelines for new changes cause older pending or running pipelines on the same branch to be cancelled. ⓘ
- ☒ Prevent outdated deployment jobs
When a deployment job is successful, prevent older deployment jobs that are still pending. ⓘ
 - ☒ Allow job retries for rollback deployments
Allow job retries even if the deployment job is outdated. ⓘ
- ☒ Use separate caches for protected branches
Unprotected branches will not have access to the cache from protected branches. ⓘ

CI/CD configuration file

The name of the CI/CD configuration file. A path relative to the root directory is optional (for example `my/path/.myfile.yml`). ⓘ

Nous venons de définir le fichier de configuration du CICD qui correspond au fichier de notre nouveau dépôt Python CI.

2) Mise en place des 3 branches

Active branches				
main	 default	protected		
0e4e3e89 · ajout des fichiers · 4 minutes ago				
				
preprod			0 0	
0e4e3e89 · ajout des fichiers · 4 minutes ago				
				
prod			0 0	
0e4e3e89 · ajout des fichiers · 4 minutes ago				
				

3) Début rédaction des pipelines de tests

On définit le noms de venv directory dans une variable gitlab

Edit variable



Type

Variable (default) ▼

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 protected tags only.
- ☐ Expand variable reference
\$ will be treated as the start of a reference to another variable.

Description (optional)

VENV_PATH

The description of the variable's value or usage.

Key

VENV_PATH

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

venv

On avait commencé à faire un stage de setup pour installer tout les dépendances d'un coup, on a supprimé après coup car on veut privilégier une indépendances des jobs et leur cloisonnement (une approche stateless).

Status	Pipeline	Created by	Stages	Actions
Passed 00:00:23 just now	ajout des fichiers #72 1.0.0 8e4e3e89 latest tag			

Edit file

Write

Preview changes

main

.gitlab-ci.yml

Apply a template

1

stages:

2

- tests

3

- minification

4

- deploy

5

6

pylint_test:

7

stage: tests

8

image: python:3.11

9

script:

10

- python -m venv \$VENV_PATH

11

- \$VENV_PATH/bin/pip install -r requirements.txt #on installe Flask

12

- \$VENV_PATH/bin/pip install pylint

13

- \$VENV_PATH/bin/pylint hello.py

14

15

black_test:

16

stage: tests

17

image: python:3.11

18

script:

19

- python -m venv \$VENV_PATH

20

- \$VENV_PATH/bin/pip install -r requirements.txt #on installe Flask

21

- \$VENV_PATH/bin/pip install black

22

- \$VENV_PATH/bin/black --check --diff hello.py

ajout des fichiers

✓ Passed

Created 1 minute ago by cicd, finished just now

For commit

8e4e3e89

In

1.8.2

latest

tag

2 jobs

40 seconds, queued for 68 seconds

Pipeline

Jobs 2

Tests 0

tests

✓ black_test

✓ pylint_test

```

stages:
  - tests
  - minification
  - deploy

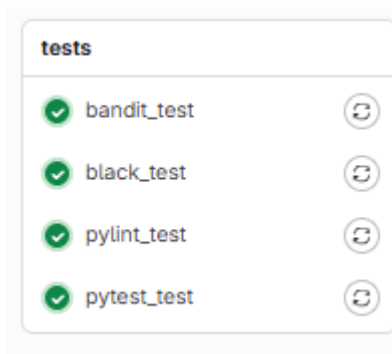
pylint_test:
  stage: tests
  image: python:3.11
  script:
    - python -m venv $VENV_PATH
    - $VENV_PATH/bin/pip install -r requirements.txt pylint
    - $VENV_PATH/bin/pylint hello.py

black_test:
  stage: tests
  image: python:3.11
  script:
    - python -m venv $VENV_PATH
    - $VENV_PATH/bin/pip install black
    - $VENV_PATH/bin/black --check --diff hello.py

bandit_test:
  stage: tests
  image: python:3.11
  script:
    - python -m venv $VENV_PATH
    - $VENV_PATH/bin/pip install bandit
    - $VENV_PATH/bin/bandit hello.py

pytest_test:
  stage: tests
  image: python:3.11
  script:
    - python -m venv $VENV_PATH
    - $VENV_PATH/bin/pip install -r requirements.txt pytest
    - $VENV_PATH/bin/pytest

```



4) Job de Minification des fichiers .js


```

minification_js:
  stage: minification
  image: node:24
  rules:
    - if: $CI_COMMIT_BRANCH != "main"
  script:
    - npm install uglify-js -g
    - uglifyjs static/script.js > static/script.min.js
  artifacts:
    when: on_success
    expire_in: 30 days
    paths:
      - static/script.min.js

```

ajout des fichiers

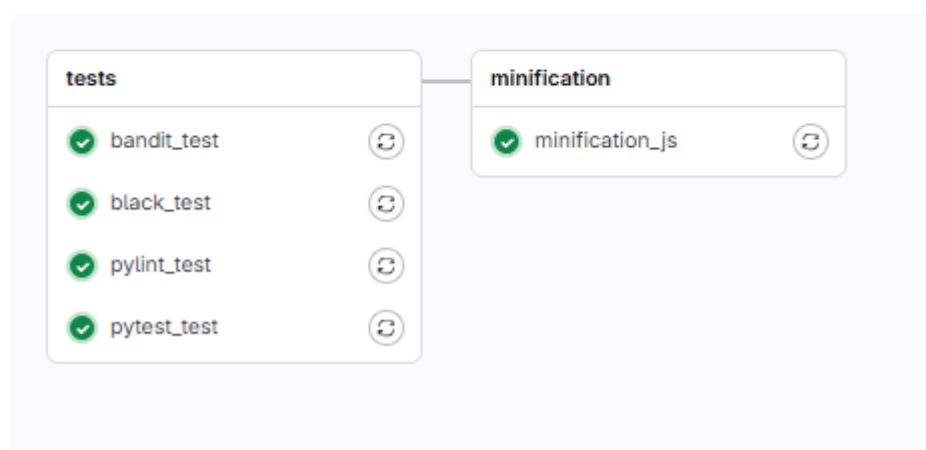
✓ Passed Created 3 minutes ago by **cicd**, finished 1 minute ago

For commit `8e4e3e89` 

In `1.0.5`

latest tag  9 jobs ⌚ 1 minute 11 seconds, queued for 2 seconds

Pipeline Jobs 9 Tests 0



5) Jobs de déploiement preprod et prod

On commence par définir les variables gitlab de de la clé privée SSH et `KNOWN_HOSTS` pour nous permettre de nous connecter à la machine de déploiement et copier les fichier zip depuis notre conteneur.

Add variable



Type

Variable (default)



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 protected 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

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

```
IAAAAAAtzc2gtZWQyNTUxOQAAACB+ynPEVErvfP
rPge9WVYtx9EEU01ftaeNz5Du3kYbaaw
AAAE0kDYu+pQmV/RtT10+8/ws4F2X0FyxUAIUc
1T6gQXk0LH7Kc8RUSu98+s+B71ZVi3H0
QRQ7V+1p43Pk07eRhtprAAAABGNpY2QB
-----END OPENSSH PRIVATE KEY-----
```

Variable value will be evaluated as raw string.

Add variable



Type

Variable (default)



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 protected 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

KNOWN_HOSTS

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

```
QB1jRCWL+/Z0kGhn8glQ4= ecdsa-sha2-  
nistp256  
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbml  
lzdHAyNTYAAABBBBQ4R8D/1MU/hN93pLOCJQPU  
nWDdL5ImFrGQHDpFFx3Hd0tHkQI1yIH+nI0ipv  
mFg6IbFiPc8nKhL2N8EqQCQjo=
```



Variable value will be evaluated as raw string.

Add variable

Cancel

On teste nos pipelines et le déploiement sur preprod et prod

```
deployment_preprod:
  stage: deploy
  image: python:3.11
  rules:
    - if: $CI_COMMIT_BRANCH == "preprod"
  script:
    - apt-get update && apt-get install -y zip openssh-client
    - python -m venv venv_prod
    - venv_prod/bin/pip install -r requirements.txt
    - sed -i "s|COMMIT_DATE=\\\"|COMMIT_DATE=\\\"$CI_COMMIT_TIMESTAMP\\\"|" conf.py
    - sed -i "s|COMMIT_SHORT_HASH=\\\"|COMMIT_SHORT_HASH=\\\"$CI_COMMIT_SHORT_SHA\\\"|" conf.py
    - sed -i "s|COMMIT_AUTHOR=\\\"|COMMIT_AUTHOR=\\\"$CI_COMMIT_AUTHOR\\\"|" conf.py
    - sed -i "s|COMMIT_BRANCH=\\\"|COMMIT_BRANCH=\\\"$CI_COMMIT_BRANCH\\\"|" conf.py
    - zip -r hello_preprod.zip hello.py static conf.py venv_prod
    - mkdir ~/.ssh && chmod 700 ~/.ssh
    - echo "$KNOWN_HOSTS" > ~/.ssh/known_hosts && chmod 400 ~/.ssh/known_hosts
    - echo "$SSH_PRIVATE_KEY" > ~/.ssh/id_rsa && chmod 400 ~/.ssh/id_rsa
    - scp hello_preprod.zip root@192.168.56.38:/root/

deployment_prod:
  stage: deploy
  image: python:3.11
  rules:
    - if: $CI_COMMIT_BRANCH == "prod"
  script:
    - apt-get update && apt-get install -y tar gzip openssh-client
    - python -m venv venv_prod
    - venv_prod/bin/pip install -r requirements.txt
    - sed -i "s|COMMIT_DATE=\\\"|COMMIT_DATE=\\\"$CI_COMMIT_TIMESTAMP\\\"|" conf.py
    - sed -i "s|COMMIT_SHORT_HASH=\\\"|COMMIT_SHORT_HASH=\\\"$CI_COMMIT_SHORT_SHA\\\"|" conf.py
    - sed -i "s|COMMIT_AUTHOR=\\\"|COMMIT_AUTHOR=\\\"$CI_COMMIT_AUTHOR\\\"|" conf.py
    - sed -i "s|COMMIT_BRANCH=\\\"|COMMIT_BRANCH=\\\"$CI_COMMIT_BRANCH\\\"|" conf.py
    - tar -czf hello_prod.tar.gz hello.py static conf.py venv_prod
    - mkdir ~/.ssh && chmod 700 ~/.ssh
    - echo "$KNOWN_HOSTS" > ~/.ssh/known_hosts && chmod 400 ~/.ssh/known_hosts
    - echo "$SSH_PRIVATE_KEY" > ~/.ssh/id_rsa && chmod 400 ~/.ssh/id_rsa
    - scp hello_prod.tar.gz root@192.168.56.38:/root/
```

On a utilisé 'sed' pour mettre à jour le fichier [conf.py](#) en fonction du commit lié.

ajout des fichiers

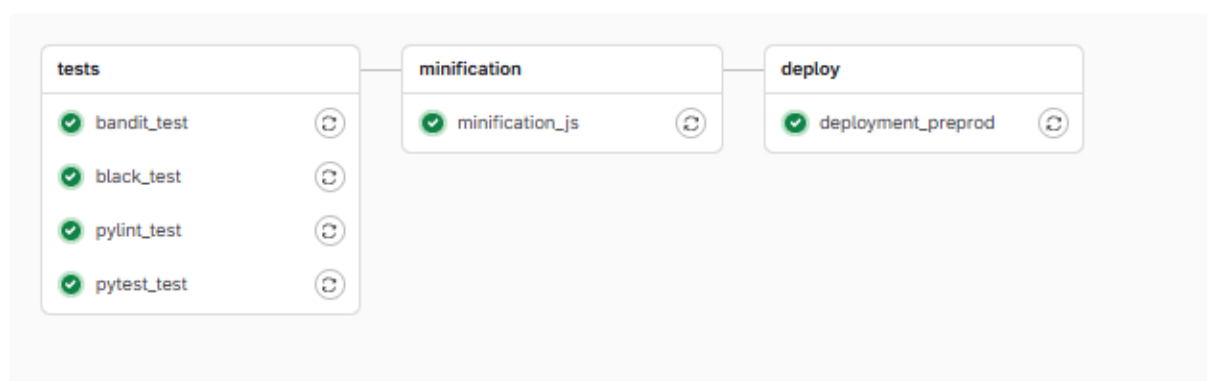
✓ Passed Created 2 minutes ago by **ciccd**, finished 20 seconds ago

For commit [0e4e3e89](#)

In [preprod](#)

[latest](#) [branch](#) 6 jobs ⌚ 2 minutes 2 seconds, queued for 4 seconds


Pipeline Jobs 6 Tests 0

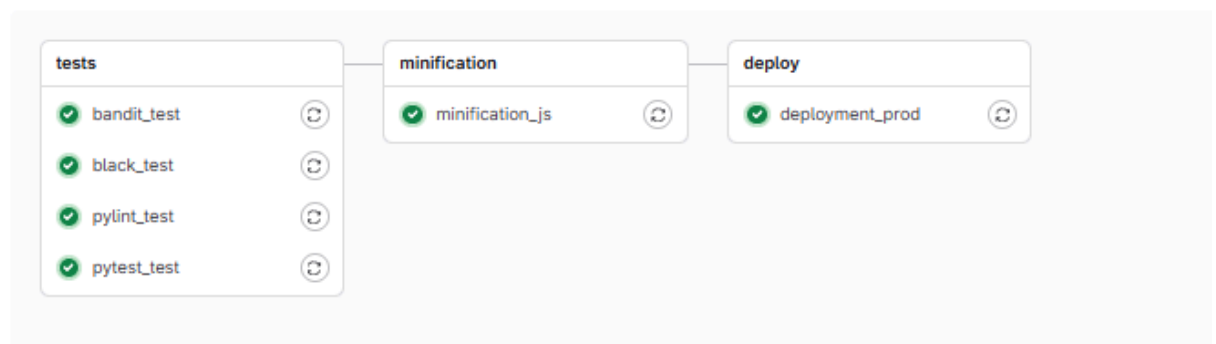


Le fichier de preprod est bien réceptionné.

```
root@cicd-deploy:~# ls /root/  
hello_preprod.zip  
root@cicd-deploy:~#
```

ajout des fichiers

✓ Passed Created 4 minutes ago by **cicd**, finished just now
For commit `0e4e3e89` 
In `prod`
latest branch 6 jobs ⌚ 3 minutes 45 seconds, queued for 3 seconds
Pipeline Jobs 6 Tests 0



Le fichier de preprod est bien réceptionné.

```
root@cicd-deploy:~# ls /root/  
hello_preprod.zip hello_prod.tar.gz  
root@cicd-deploy:~#
```

On a unzip le fichier hello_preprod.zip.

```
root@cicd-deploy:~# cat conf.py  
COMMIT_DATE="2026-01-07T14:16:06+01:00"  
COMMIT_SHORT_HASH="0e4e3e89"  
COMMIT_AUTHOR="FrancoisTourelle <tourellefrancois@gmail.com>"  
COMMIT_BRANCH="preprod"  
root@cicd-deploy:~#
```

On a juste modifié le zip pour que requirements.txt apparaissent lors des prochaines fois.

```

- sed -i "s|COMMIT_BRANCH=\\\"|COMMIT_BRANCH=\\\"$CI_COMMIT_BRANCH\\\"|" conf.py
- zip -r hello_preprod.zip hello.py static conf.py requirements.txt venv_prod
- mkdir ~/.ssh && chmod 700 ~/.ssh
- echo "$KNOWN_HOSTS" > ~/.ssh/known_hosts && chmod 400 ~/.ssh/known_hosts
- echo "$SSH_PRIVATE_KEY" > ~/.ssh/id_rsa && chmod 400 ~/.ssh/id_rsa
- scp hello_preprod.zip root@192.168.56.30:/root/


ant_prod:
ge: deploy
ge: python:3.11
as:
- if: $CI_COMMIT_BRANCH == "prod"
ipt:
- apt-get update && apt-get install -y tar gzip openssh-client
- python -m venv venv_prod
- venv_prod/bin/pip install -r requirements.txt
- sed -i "s|COMMIT_DATE=\\\"|COMMIT_DATE=\\\"$CI_COMMIT_TIMESTAMP\\\"|" conf.py
- sed -i "s|COMMIT_SHORT_HASH=\\\"|COMMIT_SHORT_HASH=\\\"$CI_COMMIT_SHORT_SHA\\\"|" conf.p
- sed -i "s|COMMIT_AUTHOR=\\\"|COMMIT_AUTHOR=\\\"$CI_COMMIT_AUTHOR\\\"|" conf.py
- sed -i "s|COMMIT_BRANCH=\\\"|COMMIT_BRANCH=\\\"$CI_COMMIT_BRANCH\\\"|" conf.py
- tar -czf hello_prod.tar.gz hello.py static conf.py requirements.txt venv_prod
- scp hello_prod.tar.gz root@192.168.56.30:/root/

```

La pipeline s'achève avec succès également pour la branche main.

ajout des fichiers

✓ Passed Created 3 minutes ago by **ci/cd**, finished just now


For commit `0e4e3e89` 


In `main`


`latest` `branch` **40** 4 jobs ⌚ 3 minutes 32 seconds, queued for 4 seconds

Pipeline Jobs **4** Tests **0**

tests

✓ bandit_test 

✓ black_test 

✓ pylint_test 

✓ pytest_test 