

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 créé un nouveau projet gitlab dans lequel nous allons mettre en place notre système de CI/CD.

Python Project

main python-project

ajout des fichiers

FrançoisTourelle authored 12 seconds ago

0e4e3689 Find file History

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

README.md

Python Project

Project information

- 1 Commit
- 1 Branch
- 0 Tags
- 4 KiB Project Storage

README

- Auto DevOps enabled
- Add LICENSE
- Add CHANGELOG
- Add CONTRIBUTING
- Add Kubernetes cluster
- Add Wiki
- Configure Integrations

Created on

January 07, 2026

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

The file has been successfully created.

main python-ci / .gitlab-ci.yml

.gitlab-ci.yml

Add new file

cicd authored just now

d254027a Find file Blame Edit

.gitlab-ci.yml 0 B

Empty file

Search page

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Prevent outdated deployment jobs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Allow job retries even if the deployment job is outdated.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

.gitlab-ci.yml@cicd/python-ci

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

Branch	Last Commit	Build Status	Actions
main <small>protected</small>	0e4e3e89 · ajout des fichiers · 4 minutes ago	<small>0 0</small>	<small> ⓘ</small> <small>New</small> <small>Download</small> <small>⋮</small>
preprod	0e4e3e89 · ajout des fichiers · 4 minutes ago	<small>0 0</small>	<small> ⓘ</small> <small>New</small> <small>Download</small> <small>⋮</small>
prod	0e4e3e89 · ajout des fichiers · 4 minutes ago	<small>0 0</small>	<small> ⓘ</small> <small>New</small> <small>Download</small> <small>⋮</small>

3) Début rédaction des pipelines de tests

On définit le noms de venv directory dans une variable gitlab

Edit variable

x

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 indépendance des jobs et leur cloisonnement (une approche stateless).

Status	Pipeline	Created by	Stages	Actions
Passed 0 00:00:23 just now	ajout des fichiers #72 ➔ 1.0.0 ➔ 8e4e3e89			

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 $ENV_PATH  
11    - $ENV_PATH/bin/pip install -r requirements.txt #on installe Flask  
12    - $ENV_PATH/bin/pip install pylint  
13    - $ENV_PATH/bin/pylint hello.py  
14  
15 black_test:  
16   stage: tests  
17   image: python:3.11  
18   script:  
19    - python -m venv $ENV_PATH  
20    - $ENV_PATH/bin/pip install -r requirements.txt #on installe Flask  
21    - $ENV_PATH/bin/pip install black  
22    - $ENV_PATH/bin/black --check --diff hello.py
```

cicd / Python Project / Pipelines / #76

ajout des fichiers

 Passed Created 1 minute ago by cicd, finished just now

For commit `0e4e3e89` 

In `1.0.2`

 latest  tag  2 jobs  40 seconds, queued for 68 seconds

[Pipeline](#) [Jobs 2](#) [Tests 0](#)

- | tests |
|---|
|  black_test  |
|  pylint_test  |

```

stages:
  - tests
  - minification
  - deploy

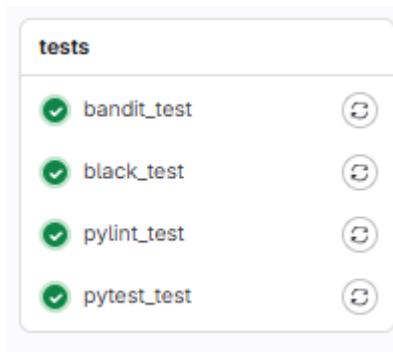
pylint_test:
  stage: tests
  image: python:3.11
  script:
    - python -m venv $ENV_PATH
    - $ENV_PATH/bin/pip install -r requirements.txt pylint
    - $ENV_PATH/bin/pylint hello.py

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

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

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

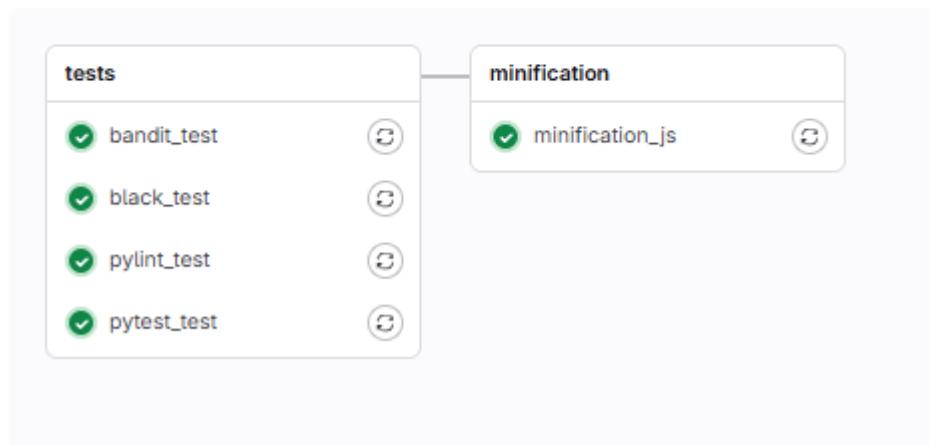
```

ajout des fichiers

Passed Created 3 minutes ago by cicd, finished 1 minute ago
 For commit `0e4e3e89`

In `1.0.5`
latest tag `eo 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 la clé privée SSH et KNOWN_HOSTS pour nous permettre de nous connecter à la machine de déploiement et copier les fichiers zip depuis notre conteneur.

Add variable

X

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

```
IAAAAAtzc2gtZWQyNTUx0QAAACB+ynPEVERvfP  
rPge9WVYtx9EEU01ftaeNz5Du3kYbaaw  
AAAEDkDYu+pQmV/RtT10+8/ws4F2X0FyxUAIUc  
1T6gQXk0LH7Kc8RUSu98+s+B71ZVi3H0  
QRQ7V+1p43Pk07eRhtprAAAABGNpY2QB  
-----END OPENSSH PRIVATE KEY-----
```

Variable value will be evaluated as raw string.

Add variable

X

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+/Z0KGhn8gI0Q4= ecdsa-sha2-nistp256  
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbm  
1zdHAyNTYAAABBBBQ4R8D/1MU/hN93pLOCJQPU  
nWDdL5ImFrGQHDpFFx3Hd0tHkQI1yIH+nI0ipv  
mFg6IbFiPc8nKh12N8EqQCQjo=
```

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@192.168.56.30:/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@192.168.56.30:/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 cied, finished 20 seconds ago

For commit 0e4e3e89 [Diff](#)

In [preprod](#)

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

Pipeline Jobs 6 Tests 0

```
graph LR; A[tests] --> B[minification]; B --> C[deploy]; A --> C;
```

- bandit_test
- black_test
- pylint_test
- pytest_test

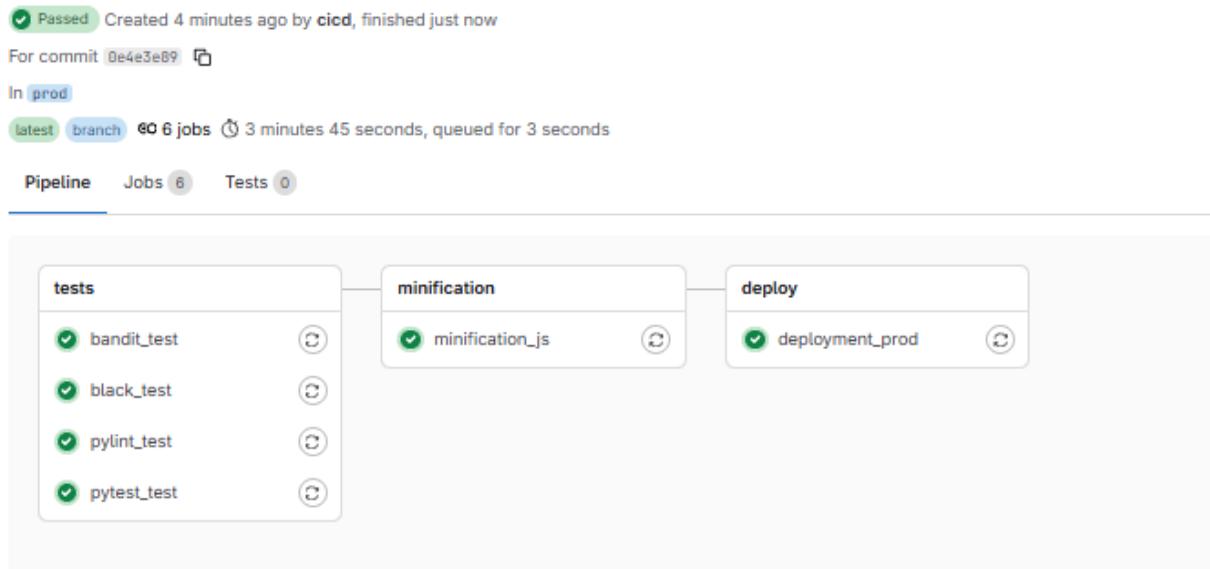
- minification_js

- deployment_preprod

Le fichier de preprod est bien réceptionné.

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

ajout des fichiers



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="FrancoisToureille <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=\"$CI_COMMIT_BRANCH\"|" conf.py
- zip -r hello_preprod.zip hello.py static conf.py requirements.txt venv_prod
- mkdir ~/.ssh && chmod 700 -R ~/.ssh
- echo "$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

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

```

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

ajout des fichiers

