

TP CI/CD

Sur le gitlab de MiNET

clonage du repo TP

git clone https://gitlab.minet.net/Balmine/formation-ci-cd.git

clonage des données existantes du TP à son état premier

---

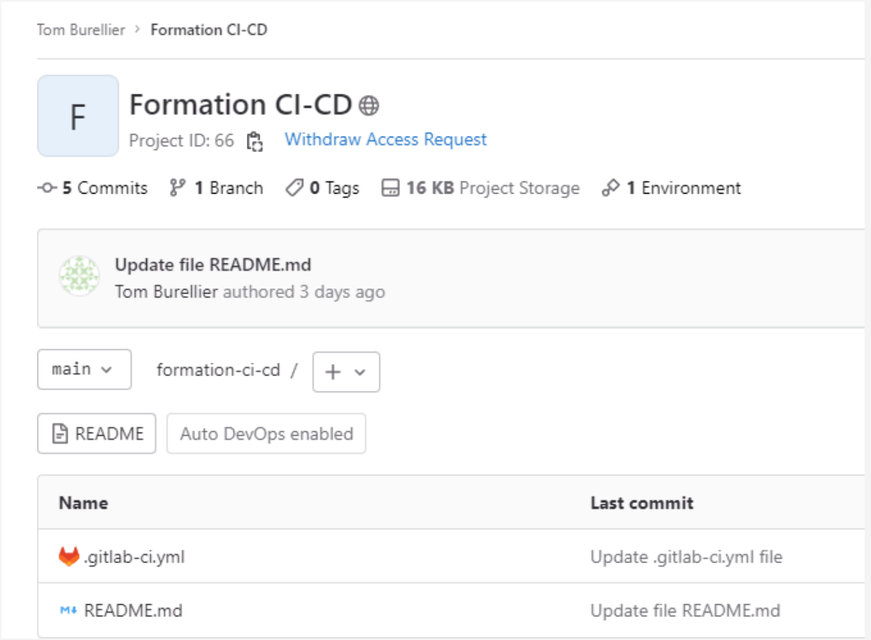
cd formation-ci-cd

déplacement dans le dossier du repository

---

git status

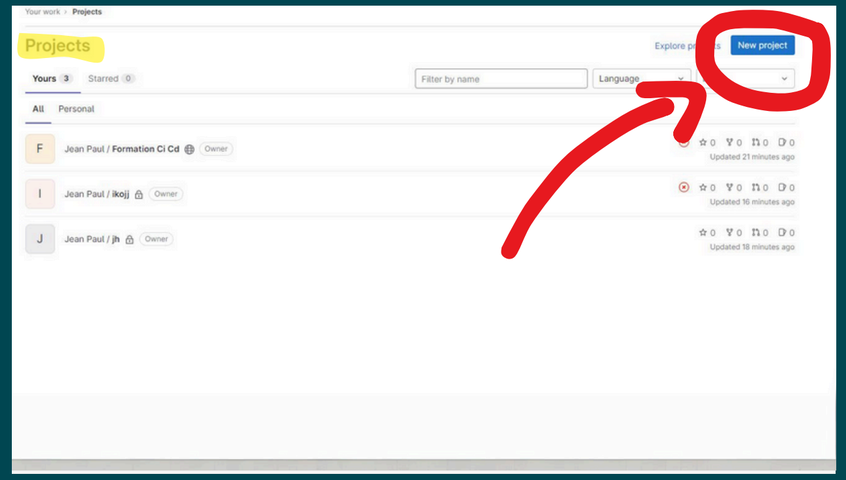
vérifiez que tout va bien dans le projet !



Créez votre propre repo



- Créez vous un compte sur gitlab.minet.net
- Connectez vous ensuite :)



- Créez vous un super projet sur votre gitlab
- Faites en sorte qu’il soit vierge (sans README) et qu’il soit publique pour le debugging :)

Push sur votre repo personnel

git remote rename origin old-origin

changer l’ancien remote origin pour le remplacer

---

git remote add origin https://gitlab.minet.net/toi/ton-projet.git

Ajouter un nouveau remote qui correspond à votre nouveau projet fraîchement crée

---

git push -u origin --all

push mon projet sur votre repo

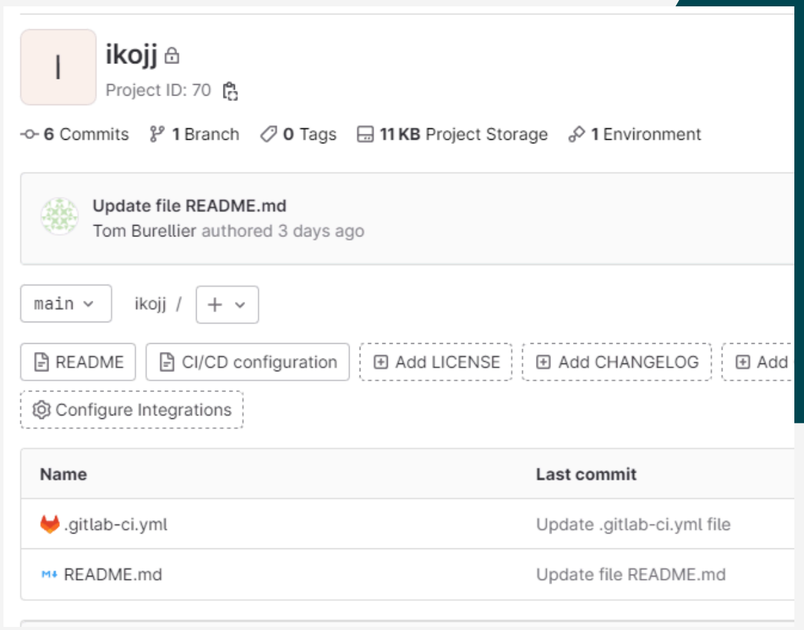
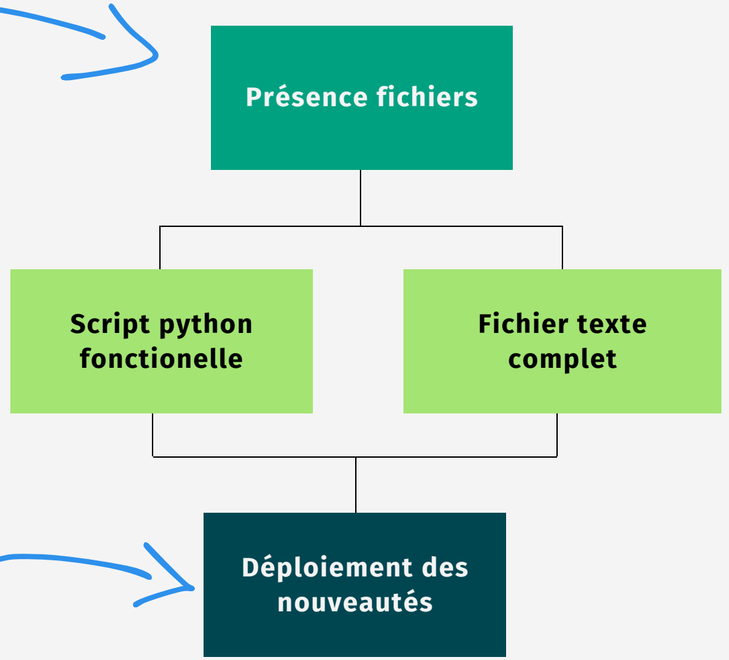


Diagramme simple du CI/CD

.gitlab-ci.yml

```yaml
completeness-job:
  stage: completeness
  script:
    - ls data.txt
    - ls script.py
```




```yaml
txt-test-job:   # Ce job s'exé
  stage: tests    # Il démarre
  tags: [production]
  script:
    - if ! grep -q "printemps"

python-test-job:   # Ce job s'
  stage: tests    # Il peut s'
  script:
    - python3 script.py
```

```yaml
deploy-job:      # Ce job s'exécute dans l'étape
  stage: deploy  # Il s'exécute seulement lorsqu
  script:
    - echo "Vous gagnez un point Tux ! Bravo !"
    - echo "Le déploiement est un succès."
```

Présence des fichiers

Dans le menu déroulant à gauche, trouvez dabs CI/CD : Pipelines

Prenez la dernière pipeline exécutée et ouvrez le premier job

```
Running with gitlab-runner 16.6.0 (3046fee8)
  on gitlab-runner-dev JhVVn3aKq, system ID: s_68ba1c4ad5c1
Preparing the "shell" executor                                    00:00
Using Shell (bash) executor...
Preparing environment                                             00:00
Running on gitlab-runner-dev...
Getting source from Git repository                                00:00
Fetching changes with git depth set to 20...
Reinitialized existing Git repository in /home/gitlab-runner/builds/JhVVn3aKq/0/Balmine/formation-ci-cd/.git/
Checking out f3132fd1 as detached HEAD (ref is main)...
Skipping Git submodules setup
Executing "step_script" stage of the job script                   00:00
$ ls data.txt
data.txt
$ ls script.py
script.py
Cleaning up project directory and file based variables            00:00
Job succeeded
```

Astuce : Au début lire ce genre de choses ça fais un peu peur mais dans l'idée outre la partie récupération du code ceci n'est juste qu'une console qui exécute des commandes et voit les retours d'exécution (comme par exemple les erreurs ou les prints).

Script python fonctionelle

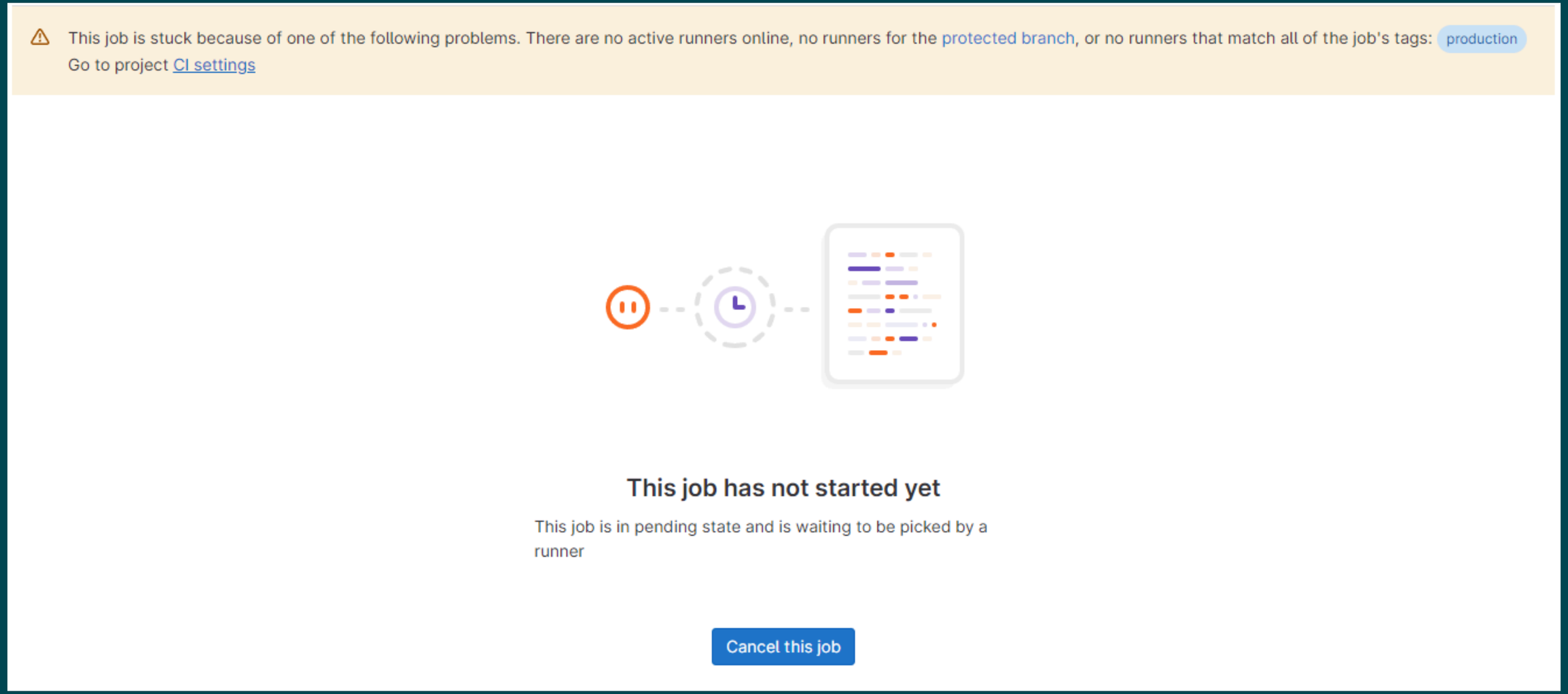
Prenez la dernière pipeline exécutée et ouvrez le deuxième job

```
Running with gitlab-runner 16.6.0 (3046fee8)
  on gitlab-runner-dev JhVVn3aKq, system ID: s_68ba1c4ad5c1
Preparing the "shell" executor                                    00:00
Using Shell (bash) executor...
Preparing environment                                             00:00
Running on gitlab-runner-dev...
Getting source from Git repository                                00:00
Fetching changes with git depth set to 20...
Reinitialized existing Git repository in /home/gitlab-runner/builds/JhVVn3aKq/0/Balmine/formation-ci-cd/.git/
Checking out f3132fd1 as detached HEAD (ref is main)...
Skipping Git submodules setup
Executing "step_script" stage of the job script                   00:00
$ python3 script.py
  File "/home/gitlab-runner/builds/JhVVn3aKq/0/Balmine/formation-ci-cd/script.py", line 13
    for valeur in valeurs:
IndentationError: unexpected indent
Cleaning up project directory and file based variables            00:00
ERROR: Job failed: exit status 1
```

Astuce : En vert, c'est basiquement ce qui s'est bien passé, en bleu c'est des messages de logs, et en rouge c'est les erreurs.

Présence de texte

Prenez la dernière pipeline exécutée et ouvrez le deuxième job



Installation de son propre runner

Le créer du pdv de gitlab

Settings > CI/CD > Runners > Project Runners > New project runner

(ne pas oublier de mettre le tag production)

Installer le runner sur sa machine

Liste de commandes fastidieuses, cliquer sur le texte pour suivre les étapes

. How do I install GitLab Runner?

Register votre runner

Suivre les étapes indiqués sur gitlab pour enregistrer la connexion

(choisir **shell** comme **executor**)

Vérifier si le runner est fonctionnel

> gitlab-runner run

Déploiement en production

Prenez la dernière pipeline exécutée et ouvrez le **troisième** job

```
Running with gitlab-runner 16.6.0 (3046fee8)
  on feur-laptop 2XUz58ZF3, system ID: s_d5f146d69be6
Preparing the "shell" executor                                    00:00
Using Shell (bash) executor...
Preparing environment                                             00:00
Running on feur-laptop...
Getting source from Git repository                                00:00
Fetching changes with git depth set to 20...
Dépôt Git existant réinitialisé dans /home/balmine/builds/2XUz58ZF3/0/Balmine/formation-ci-cd/.git/
Checking out d5daf532 as detached HEAD (ref is main)...
Skipping Git submodules setup
Executing "step_script" stage of the job script                   00:00
$ echo "Deploying application..."
Deploying application...
$ echo "Application successfully deployed."
Application successfully deployed.
Cleaning up project directory and file based variables            00:00
Job succeeded
```

C’est gagné !!

