

Exemple avec Pages

Tout d'abord on va créer un nouveau repo sur Gitlab. Puis on va l'ajouter dans un repo local.

```
git init gitlab-pages-test
cd gitlab-pages-test
git remote add origin <lien ssh ou https ici>
```

Ensuite on va créer un nouveau fichier `index.html` dans lequel il est simplement écrit "Hello, World!"

```
echo "Hello, World!" > index.html
```

Maintenant que l'on a créé notre contenu, on va créer le fichier principal `.gitlab-ci.yml` qui va contenir les instructions pour créer notre projet. Ce fichier constitue ce qui s'appelle dans GitLab, une "pipeline"

```
# "pages" est le nom du "job"
pages:
  # Stage indique quel type d'action qui est effectuée (par exemple: test, build, deploy)
  stage: deploy

  # L'image est la base du système dans lequel les commandes d'installation du projet vont être lancées. Dans
  # ce cas ci, debian
  # Nous verrons plus en détail cela dans le chapitre sur Docker
  image: debian

  # Les artifacts sont des fichiers ou dossier qui vont être exporté. Dans cet exemple `public` va être exportée
  # en dehors de notre "pipeline"
  artifacts:
    paths:
      - public

  # Le "before_script" spécifie les commandes d'installation de l'environnement, par exemple ici nous allons
  # utiliser rsync pour déployer notre projet. Donc j'installe rsync
  # Il faut toujours faire en sorte que les commandes ne nécessite pas d'interaction (exemple, en ajoutant -y à
  # la commande APT)
  before_script:
    - apt-get update
```

```
- apt-get install -y rsync
```

```
# Le "script" est la partie principale. Elle indique les commandes à lancer pour déployer notre projet.
```

```
# Ici on ne fait que créer un nouveau dossier public et copier tous les fichiers du projets (à l'exception de "public/") dans le dossier "public/"
```

```
script:
```

```
- mkdir public
```

```
- rsync -rv * public/ --exclude=public/
```

```
# On précise que seul les pushes vers la branche main peuvent appeler le job "pages".
```

```
only:
```

```
- main
```

Une fois nos deux fichiers créés on peut maintenant envoyer le tout sur le serveur

```
git add index.html .gitlab-ci.yml
```

```
git commit -m "Test de CI/CD sur gitlab"
```

```
git push origin main
```

Mainetnant on peut aller dans l'onglet "CI/CD" de Gitlab, puis dans "Jobs", puis sur "Running" pour voir les détails de ce qu'il se passe.

Si tout se passe bien après quelques secondes il devrait être écrit "Passed" et le site devrait être accessible. Le lien est trouvable dans l'onglet paramètre du repo, puis dans "Pages".

Plus d'info

- [Liste de tous les keywords pour .gitlab-ci.yml](#)
- [Gitlab CI/CD Quick Start](#)

Revision #1

Created 26 April 2023 19:09:26 by SnowCode

Updated 26 April 2023 19:09:26 by SnowCode