

Une pipeline de déploiement aux petits oignons avec Gitlab CI

Par Kevin Gautreau





\$ whois

Kevin Gautreau - @kgaut

Développeur freelance

Drupalistos depuis 2010

<https://kgaut.net>

Twitter / Github / Gitlab : @kgaut

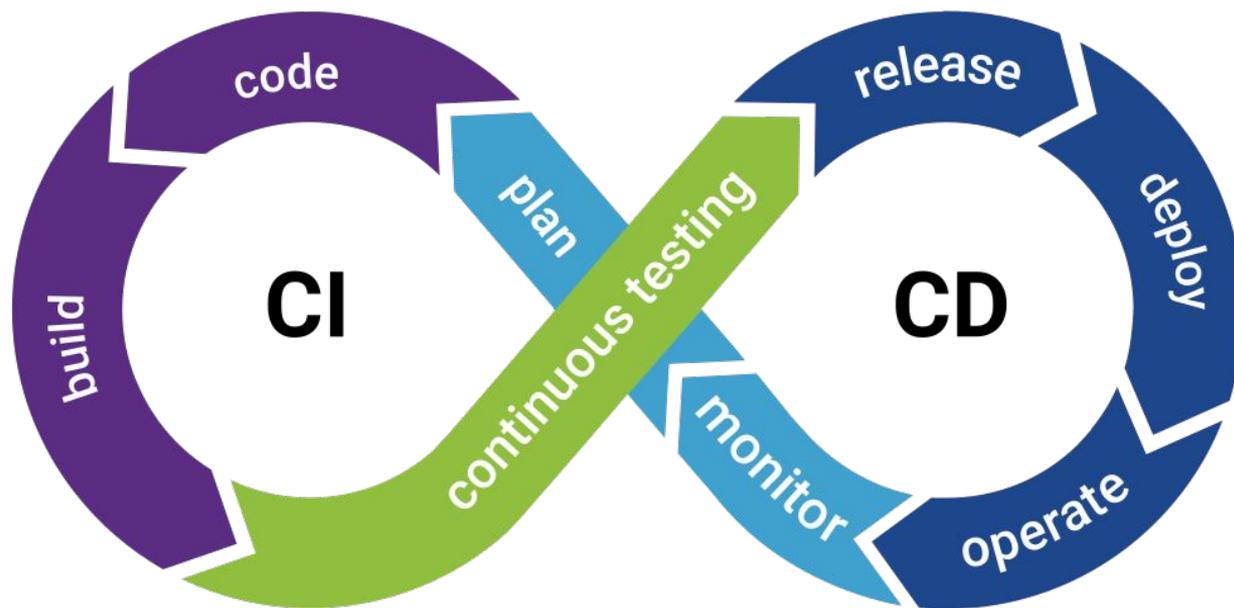
mastodon : kgaut@oisaur.com



Principes de base de la CI/CD

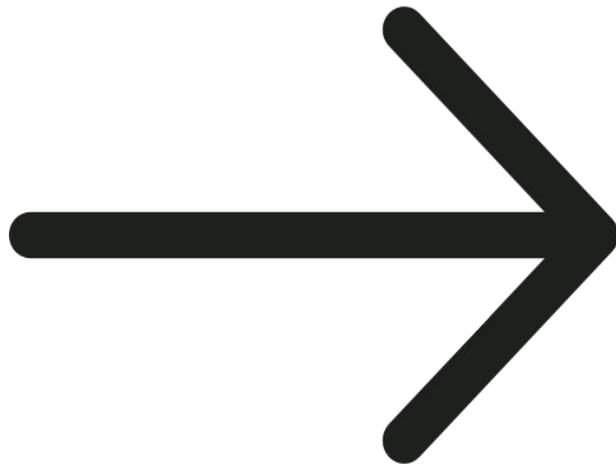


CI / CD



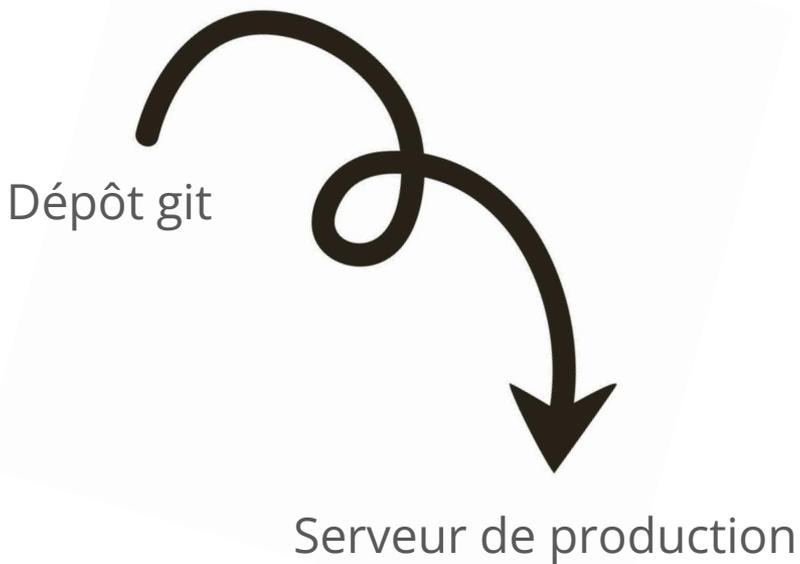


En résumé





Réaction chaînée





Différentes solutions

Github actions

Travis CI

Capistrano

Jenkins

Gitlab CI ...



Pourquoi Gitlab CI

- Directement intégré à Gitlab
- Mode SASS (gitlab.com) ou Autohébergé
- Très puissant
- Adapté aux projets libres comme privés
- Prise en main (relativement) facile



Principe de base

Suite de jobs exécutés après une action sur le dépôt git

Push sur branche preprod ⇒ Déploiement en preprod

Création d'un tag ⇒ Déploiement en production



Découpage des étapes

Création de jobs, séquencés

Si un job plante, réaction (Annulation / Rollback / Alerte)



Qu'est-ce qu'un job

Suite de commandes exécutées par un « Runner »

Généralement dans un conteneur docker

assets:generation:

image: node:19

script:

- npm ci
- npm run sass-compile



.gitlab-ci.yml

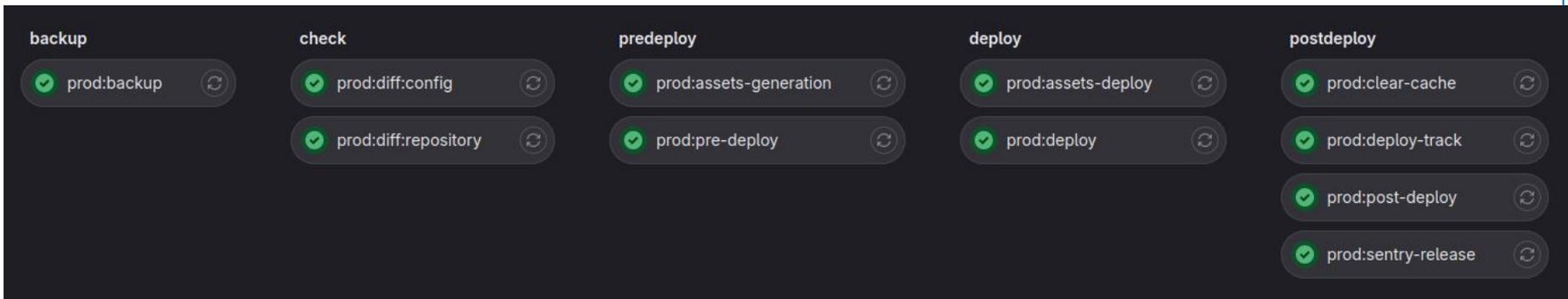
Définition de l'ensemble des jobs, variables...



Étapes d'une mise en production



Étapes d'une mise en production





Étape QA (Assurance qualité)

1. Tests unitaires du code
2. Script de validation du code
 - a. phpcs (php code sniffer)
 - b. php lint
 - c. grumphp ([git blacklist](#))



Étape Backup

```
ssh user@serveur "~/.public_html/vendor/bin/drush -r ~/.public_html"
sql-dump | gzip -9 > "~/.public_html/files/dumps/`date
+%Y-%m-%d_%H-%M-%S`-monsite.com-prod.sql.gz"
```



Étape Check

Vérifier l'intégrité du serveur de déploiement

⇒ Pas de fichier modifiés

⇒ Pas de configuration drupal modifiée

⇒ Sécuriser le déploiement



Étape Check serveur - 1/2

```
ssh user@server 'bash -s' < ./diff-repository.sh
```

```
if [[ $(git diff web) ]]; then
  FILENAME="$(date +"%Y-%m-%d_%H-%M-%S")-diff.patch"
  git diff web > "$DIFFS_PATH/$FILENAME"
  echo "-----";
  echo "modifications en $SCI_ENVIRONMENT_NAME detectées, patch créé : $FILENAME";
  echo "Liste des fichiers modifiés : ";
  echo $(git diff --name-only)
  echo "-----";
  git checkout ./
  FAIL;
```



Étape Check serveur - 2/2

ssh user@server 'bash -s' < ./diff-repository.sh

```
else
  if [[ $(git ls-files --others --exclude-standard) ]]; then
    echo "-----";
    echo "Fichiers non versionnés : "
    echo $(git ls-files --others --exclude-standard)
    echo "-----";
    FAIL;
  else
    echo "Pas de modification en $SCI_ENVIRONMENT_NAME à signaler"
  fi
fi
```



Étape Check config - 1/2

```
ssh user@server 'bash -s' < ./diff-config.sh
```

1. drush config:export
2. git diff config/sync (config modifiée)
3. git ls-files --others --exclude-standard config/sync (config ajoutée)

À utiliser en combinaison avec config-split / config ignore...



Étape pré-déploiement - Scripts pré-déploiements

Point d'entrée pour exécuter des scripts avant le déploiement

- Désactivation module qui seront supprimés via composer
- Tâches non possibles via drush updb / deploy...



Étape pré-déploiement - Générations Assets (CSS / JS)

Utilisation d'une image node via gitlab CI

assets-generation:

image: node:19

script:

- cd web/themes/mon_theme
- npm ci
- npm run sass-compile

artifacts:

paths:

- web/themes/mon_theme/css

expire_in: 20 minutes



Étape déploiement - code

```
ssh user@server 'bash -s' < ./deploy.sh
```

1. git pull / git checkout TAG
2. composer install --no-dev
3. drush deploy



Étape déploiement - assets

Envoi sur le serveur des assets précédemment générés

```
rsync -avhzi --delete web/themes/mon_theme/css -e ssh  
user@server:~/public_html/web/themes/mon_theme
```



Étapes post-déploiement

- Vidage du cache
- scripts de post-déploiement
- Backup db
- webhooks (slack, sentry, dashboard custom...)
- ...



Réutilisation



Problématique 1

10 jobs+...

2, ou 3 environnements...

=> Un fichier .gitlab-ci.yml énorme

450 lignes

```
.gitlab-ci.yml
1 | stages :
2 |   - QA
3 |   - backup
4 |   - predeploy
5 |   - deploy
6 |   - postdeploy
7 |   - scheduled
8 |   - push-prod
9 |
10 | variables:
11 |   PROJECT_NAME: [REDACTED]
12 |   PROJECT_SHORTNAME: [REDACTED]
13 |   DB_PATH: "/data/db"
14 |   DIFFS_PATH: "files/diffs"
15 |
16 | php-pgmd:
17 |   tags:
18 |     - docker
19 |   image: jakzal/phpqa
20 |   stage: QA
21 |   script: phpmd web/modules/custom text ./files/QA/phpmd.xml | tee files/QA/results/phpmd.txt
22 |   except:
23 |     - schedules
24 |     - preprod
25 |     - master
26 |   artifacts:
27 |     when: on_failure
28 |     expire_in: 1 week
29 |     paths:
30 |       - files/QA/results/phpmd.txt
31 |     allow_failure: true
32 |
33 | php-metrics:
34 |   tags:
35 |     - docker
36 |   image: jakzal/phpqa
37 |   stage: QA
38 |   script: phpmetrics --report-html=files/QA/results/php-metrics web/modules/custom | tee files/QA/
39 |   except:
40 |     - schedules
41 |     - preprod
42 |     - master
43 |   artifacts:
44 |     when: always
45 |     expire_in: 1 week
46 |     paths:
47 |       - files/QA/results/php-metrics.txt
48 |       # - files/QA/results/php-metrics
49 |     allow_failure: true
50 |
51 | php-cs:
52 |   tags:
53 |     - docker
54 |   image: jakzal/phpqa
55 |   stage: QA
56 |   script:
57 |     - phpcs web/modules/custom | tee files/QA/results/phpcs.txt
```



Explosion en plusieurs petits fichiers

```
include:  
  - local: '/files/gitlab/extends.yml'  
  - local: '/files/gitlab/jobs-0-scheduled.yml'  
# - local: '/files/gitlab/jobs-1-qa.yml'  
  - local: '/files/gitlab/jobs-2-backup.yml'  
  - local: '/files/gitlab/jobs-3-predeploy.yml'  
  - local: '/files/gitlab/jobs-4-deploy.yml'  
  - local: '/files/gitlab/jobs-5-postdeploy.yml'
```



Factorisation

```
.assets-generation:
  extends:
    - .job-docker
  image: $ASSETS_IMAGE
  stage : predeploy
  script:
    - cd $PATH_TO_THEME
    - npm ci
    - npm run sass-compile
  artifacts:
    paths:
      - $ASSETS_ARTEFACTS
    expire_in: 20 minutes
```

```
prod:assets-generation:
  extends:
    - .prod
    - .assets-generation

preprod:assets-generation:
  extends:
    - .preprod
    - .assets-generation
```



Problématique 2

Un fichier `.gitlab-ci.yml`, qui évolue

Copié / collé de projet en projet

- Comment simplifier la gestion ?
- Comment « backporter » les évolutions ?
- Comment éviter les erreurs ?



Un dépôt pour les contrôler tous

Un dépôt de jobs CI

<https://gitlab.com/kgaut/gitlab-ci-templates>

Inclus par nos différents projets

Choix de job à la carte

Marche sur gitlab.com et instances auto-hébergées

```
include:
  - project: kgaut/gitlab-ci-templates
    file:
      - '/templates/generic/stages-variables-extends.yml'
      - '/templates/drupal/backup.yml'
      - '/templates/drupal/diff-config.yml'
      - '/templates/generic/diff-repository.yml'
      - '/templates/drupal/deploy.yml'
      - '/templates/generic/deploy-tracking.yml'
      - '/templates/drupal/post-deploy-clear-cache.yml'
      - '/templates/generic/scheduled-clean-dump.yml'
      - '/templates/drupal/scheduled-backup.yml'
```



Généralisation à l'aide des variables

script:

```
- ssh user@serveur vendor/bin/drush cr
```



script:

```
- $SSH_CHAIN $DRUSH_BIN cr
```



Prod et Preprod d'un même projet ?

Utilisation des environnement des variables Gitlab CI

PROJECT_ROOT	~/domains/preprod...	preprod
Protected Expanded		
PROJECT_ROOT	~/public_html	prod
Protected Expanded		



Évolutivité sans tout casser ?

Tagging des jobs sources

Référence à la version pour chaque projet

```
include:  
  - project: kgaut/gitlab-ci-templates  
    ref: 0.3.0  
    file:  
      - '/templates/generic/stages-variables-extends.yml'  
      - '/templates/drupal/backup.yml'  
      - '/templates/drupal/diff-config.yml'  
      - '/templates/generic/diff-repository.yml'  
      - '/templates/drupal/deploy.yml'  
      - '/templates/generic/deploy-tracking.yml'
```

Tags give the ability to mark specific p

🔗 0.3.0

571647f1 · feat(changelog): prepare tag 0.3.0

🔗 0.2.2

fe272ca0 · fix(.pre-deploy): ssh connexion w

🔗 0.2.1

d5d27e20 · feat(changelog): prepare tag 0.2.1

🔗 0.2.0

398bbce0 · feat(changelog): preparing tag 0.2.0

🔗 0.1.0

08acd5 · feat(changelog): prepare tag 0.1.0

🔗 0.0.7

b8fefded · feat(templates): revert remove us

🔗 0.0.6

f0b51eb7 · feat(templates): remove useless a



Suivi des déploiements

webhook d'un outil de suivi

Identifiant (ID)	Projet	Env	Version	CI	Créé	Actions
22	drupal.kgaut.xyz	prod	0.0.1	0.3.0	27/03/2024 - 14:05	Pipeline
21	drupal.kgaut.xyz	preprod	preprod-690e0f7c	0.3.0	27/03/2024 - 13:49	Pipeline
20	dashboard.kgaut.net	prod	0.19.5	0.3.0	26/03/2024 - 16:16	Pipeline



Pour finir

Tout est opensourcé : <https://gitlab.com/kgaut/gitlab-ci-templates>

Documenté : <https://gitlab-ci-templates.kgaut.net/>

Forkez, adaptez et déployez.



Merci
pour votre
écoute !



<https://gitlab-ci-templates.kgaut.net/>