



Centre de Calcul de l'Institut National de Physique Nucléaire et de Physique des Particules

GitLab CI & Docker

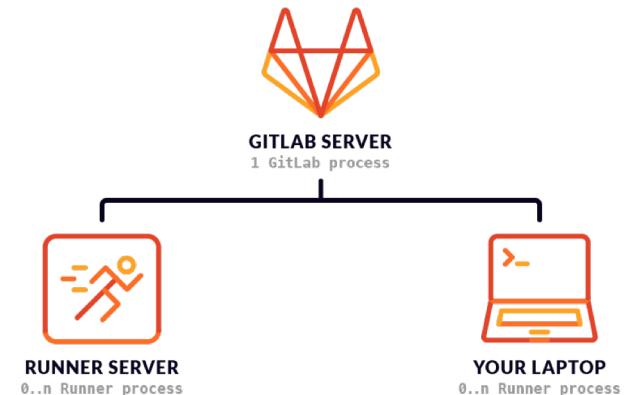
École conteneurs en production

4-8 juin 2018

Jean-René ROUET

- ▶ Tuyauterie d'intégration et livraison continue pour construire, tester, déployer et montrer votre code
 - Intégrée à GitLab
 - Apprentissage facile
 - Scalable et Rapide

- ▶ Multi plateforme: Unix, Windows, OSX, a juste besoin de Go.
- ▶ Multi language
- ▶ Stable, les runners sont différents de la plate-forme GitLab
- ▶ Parallélisme
- ▶ Gitlab Runner supporte docker
- ▶ Processus DevOps
- ▶ Production d'artefacts



► Ajouter un fichier .gitlab-ci.yml à la racine de son projet

```
# This file is a template, and might need editing before it works on your project.
# use the official gcc image, based on debian
# can use verions as well, like gcc:5.2
# see https://hub.docker.com/_/gcc/
image: gcc

stages:
  - build
  - test

build:
  stage: build
  # instead of calling g++ directly you can also use some build toolkit like make
  # install the necessary build tools when needed
  # before_script:
  #   - apt update && apt -y install make autoconf
  script:
    - g++ helloworld.cpp -o mybinary
  artifacts:
    paths:
      - mybinary
  # depending on your build setup it's most likely a good idea to cache outputs to reduce the build time
  # cache:
  #   paths:
  #     - "*.*"

# run tests using the binary built before
test:
  stage: test
  script:
    - ./runmytests.sh
```

Exemple plus complexe

```
# This file is a template, and might need editing before it works on your project.
# Read more about this script on this blog post https://about.gitlab.com/2016/11/30/setting-up-gitlab-ci-for-android-projects/, by Greysen Parrelli
image: openjdk:8-jdk

variables:
  ANDROID_COMPILE_SDK: "25"
  ANDROID_BUILD_TOOLS: "24.0.0"
  ANDROID_SDK_TOOLS: "24.4.1"

before_script:
  - apt-get --quiet update --yes
  - apt-get --quiet install --yes wget tar unzip lib32stdc++6 lib32z1
  - wget --quiet --output-document=android-sdk.tgz https://dl.google.com/android/android-sdk_r${ANDROID_SDK_TOOLS}-linux.tgz
  - tar --extract --gzip --file=android-sdk.tgz
  - echo y | android-sdk-linux/tools/android --silent update sdk --no-ui --all --filter android-${ANDROID_COMPILE_SDK}
  - echo y | android-sdk-linux/tools/android --silent update sdk --no-ui --all --filter platform-tools
  - echo y | android-sdk-linux/tools/android --silent update sdk --no-ui --all --filter build-tools-${ANDROID_BUILD_TOOLS}
  - echo y | android-sdk-linux/tools/android --silent update sdk --no-ui --all --filter extra-android-m2repository
  - echo y | android-sdk-linux/tools/android --silent update sdk --no-ui --all --filter extra-google-google_play_services
  - echo y | android-sdk-linux/tools/android --silent update sdk --no-ui --all --filter extra-google-m2repository
  - export ANDROID_HOME=$PWD/android-sdk-linux
  - export PATH=$PATH:$PWD/android-sdk-linux/platform-tools/
  - chmod +x ./gradlew

stages:
  - build
  - test
```

Schéma d'exemple de fonctionnement

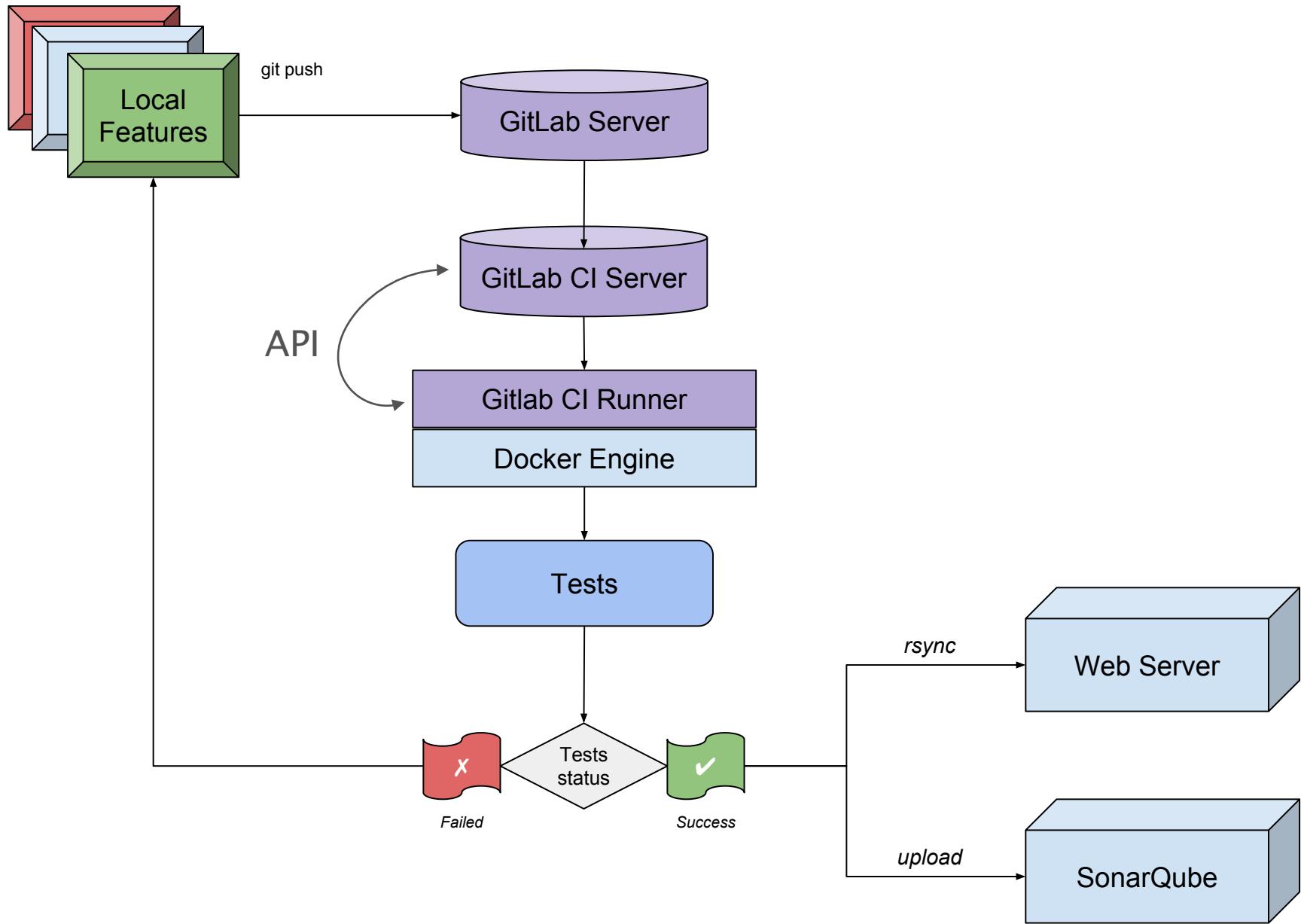


Schéma d'exemple de fonctionnement

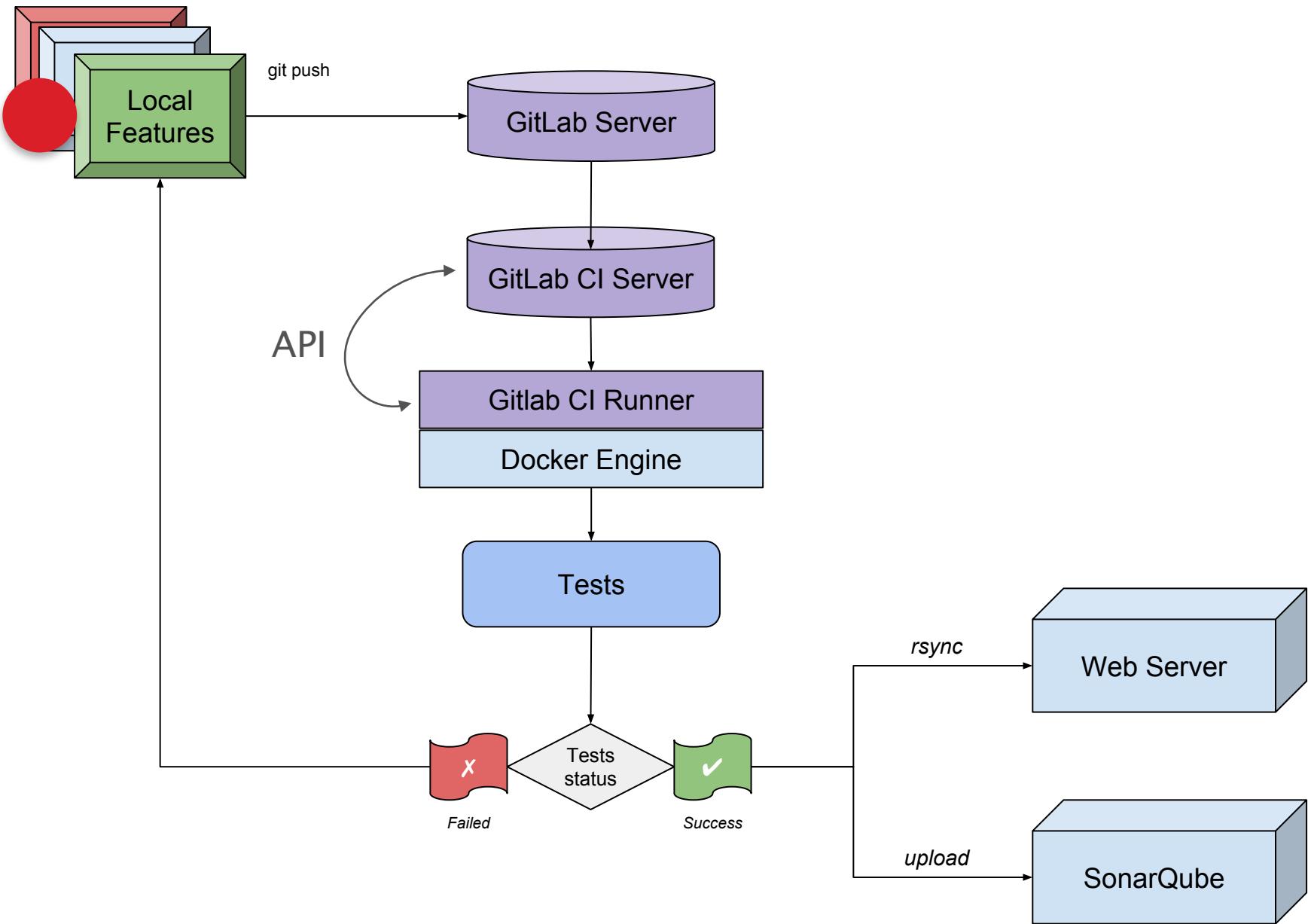


Schéma d'exemple de fonctionnement

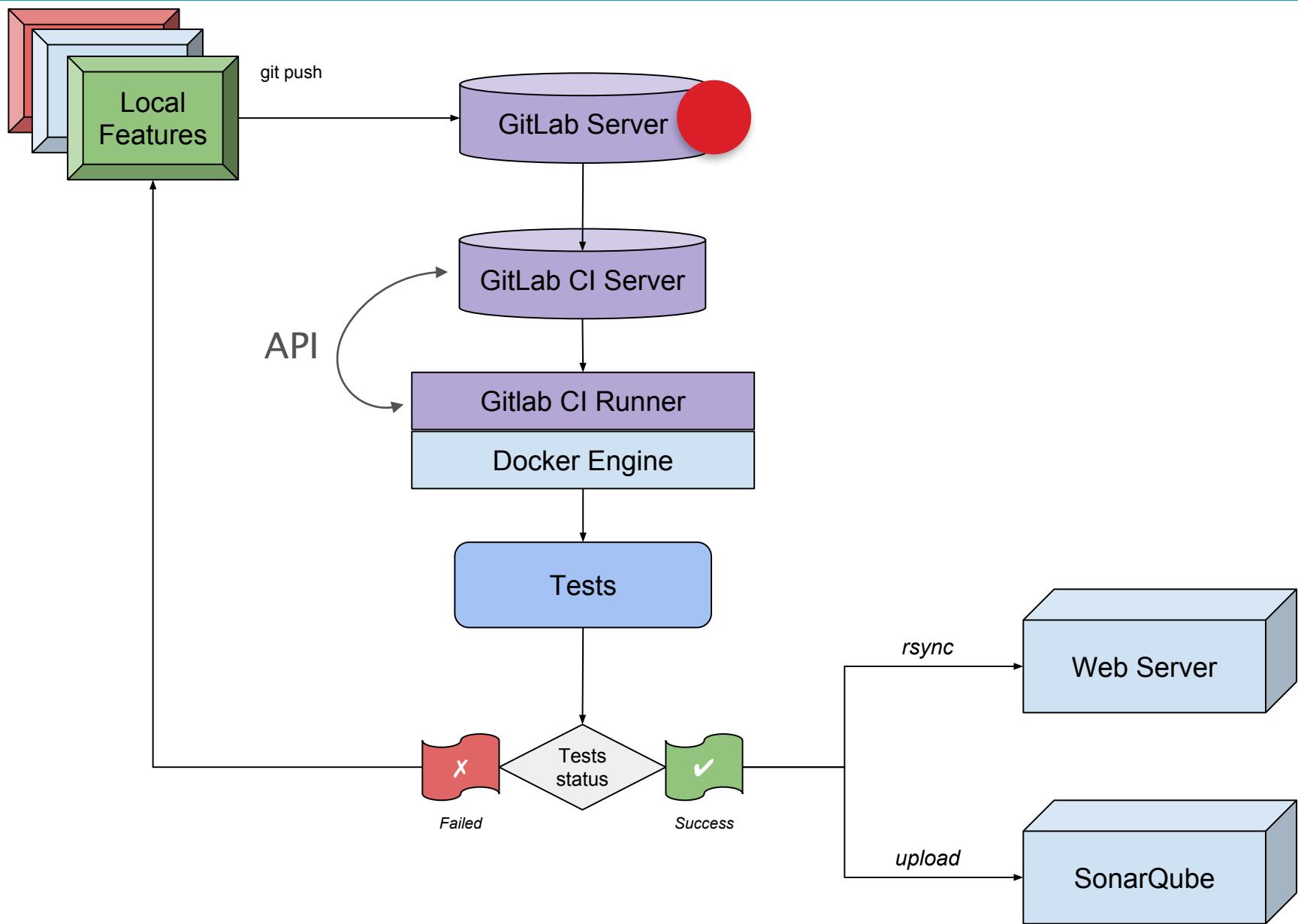


Schéma d'exemple de fonctionnement

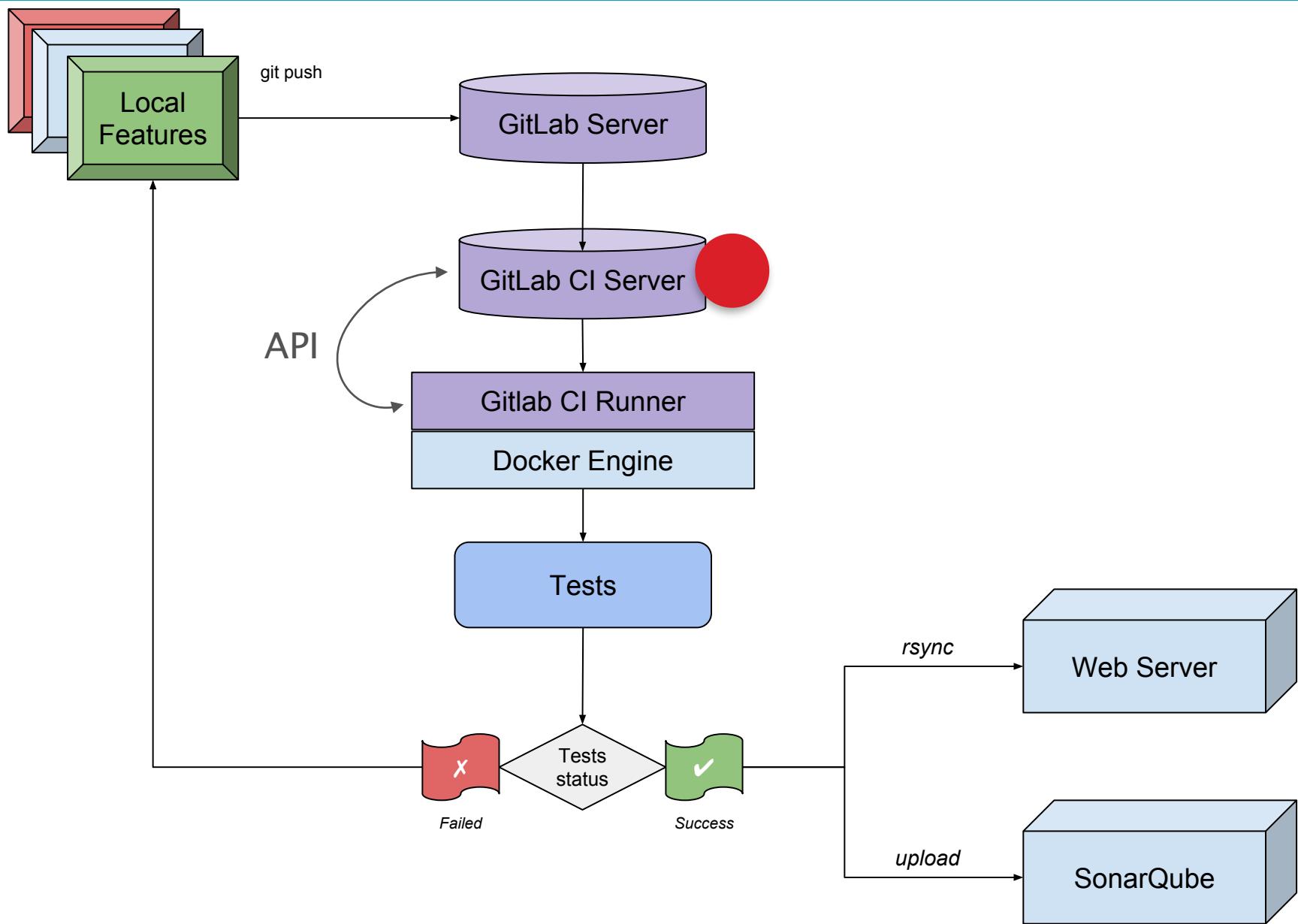


Schéma d'exemple de fonctionnement

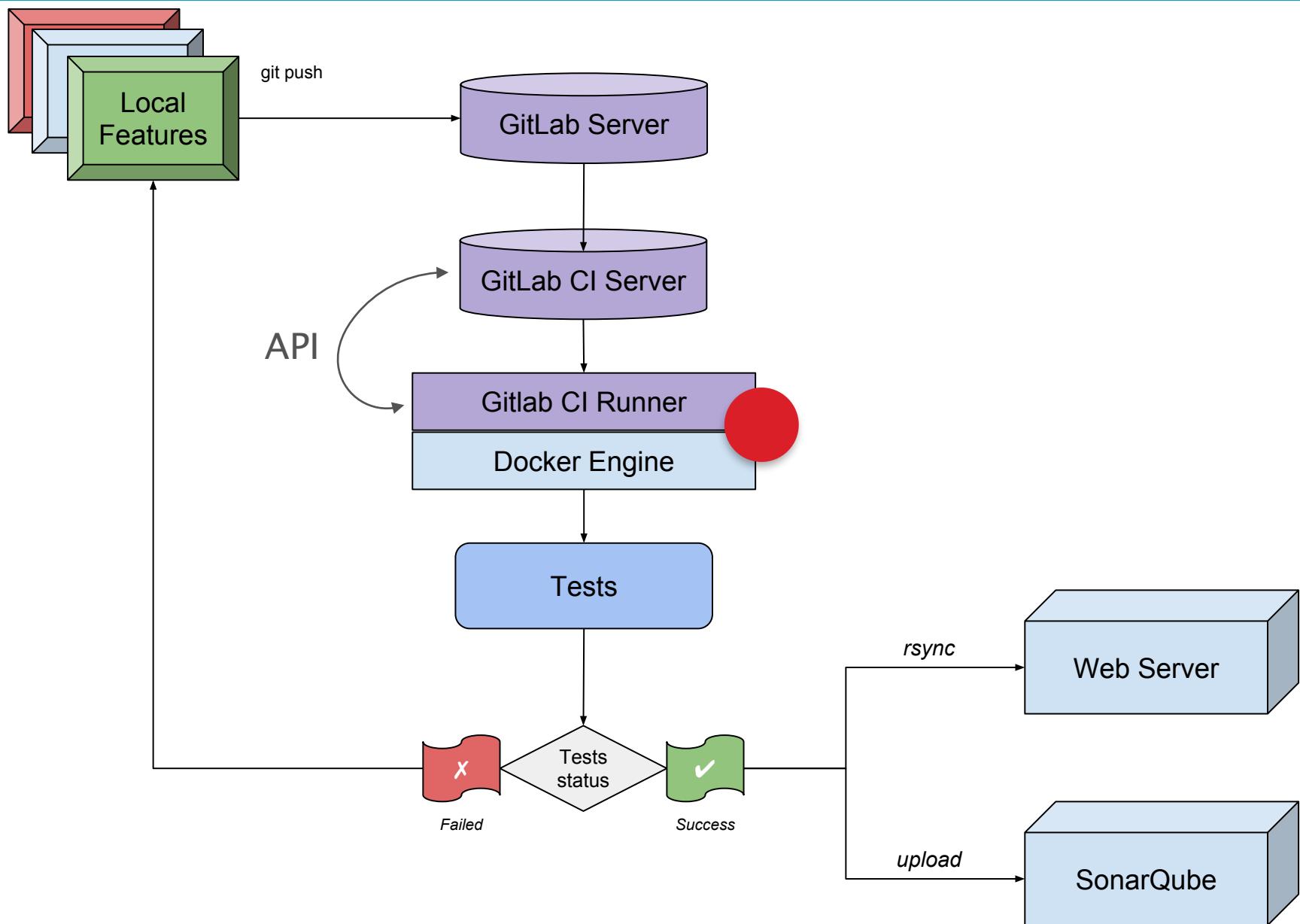


Schéma d'exemple de fonctionnement

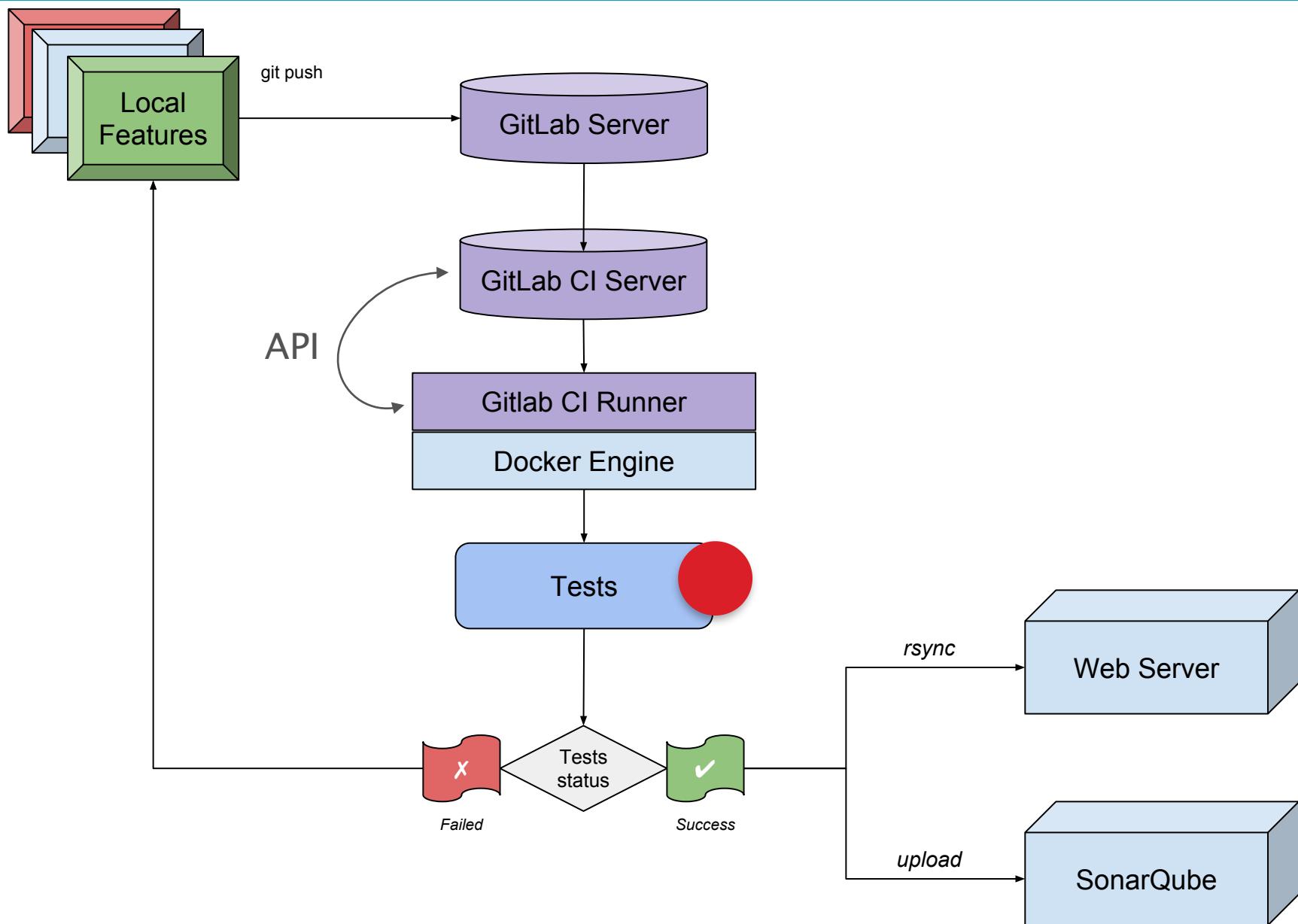


Schéma d'exemple de fonctionnement

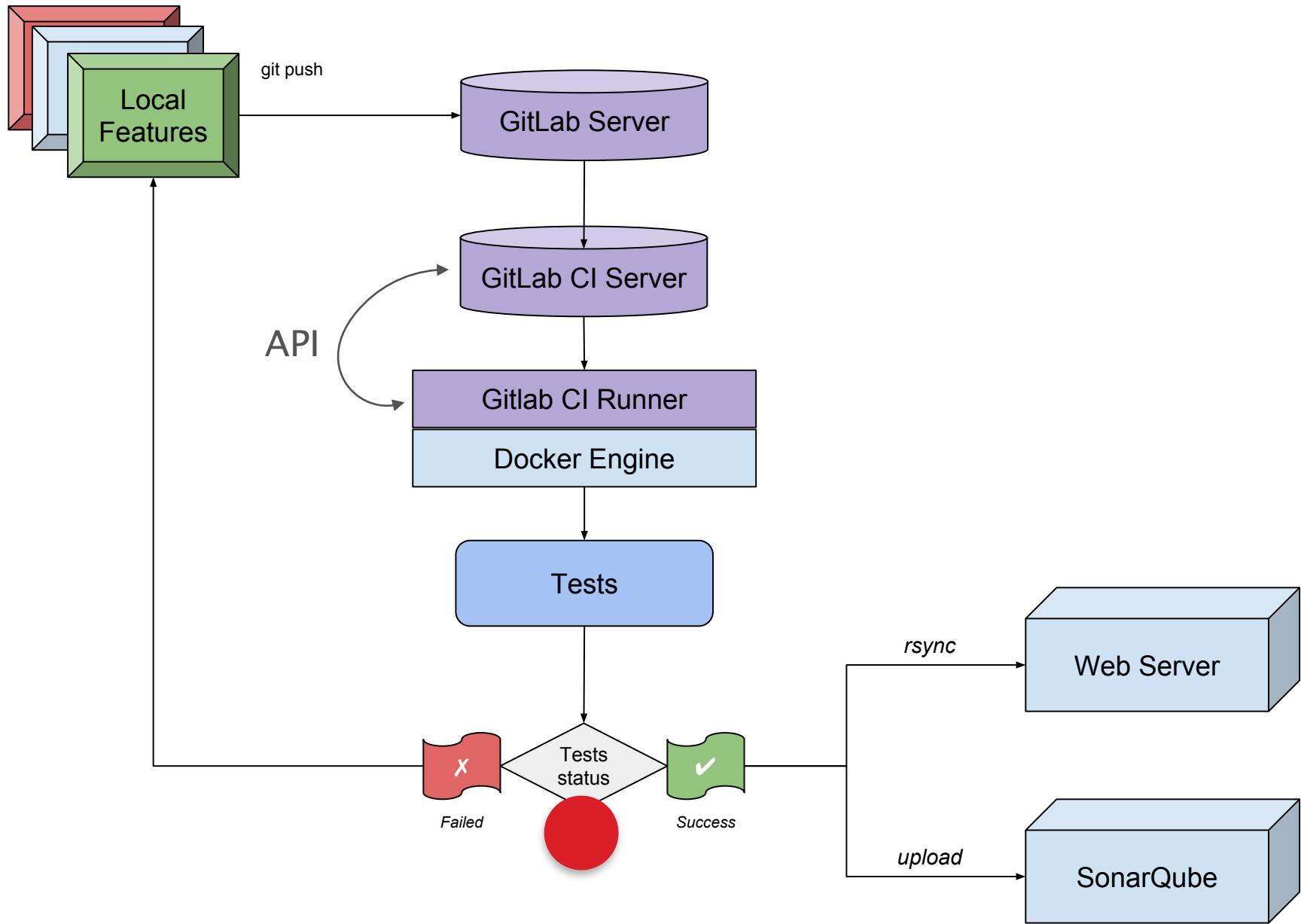


Schéma d'exemple de fonctionnement

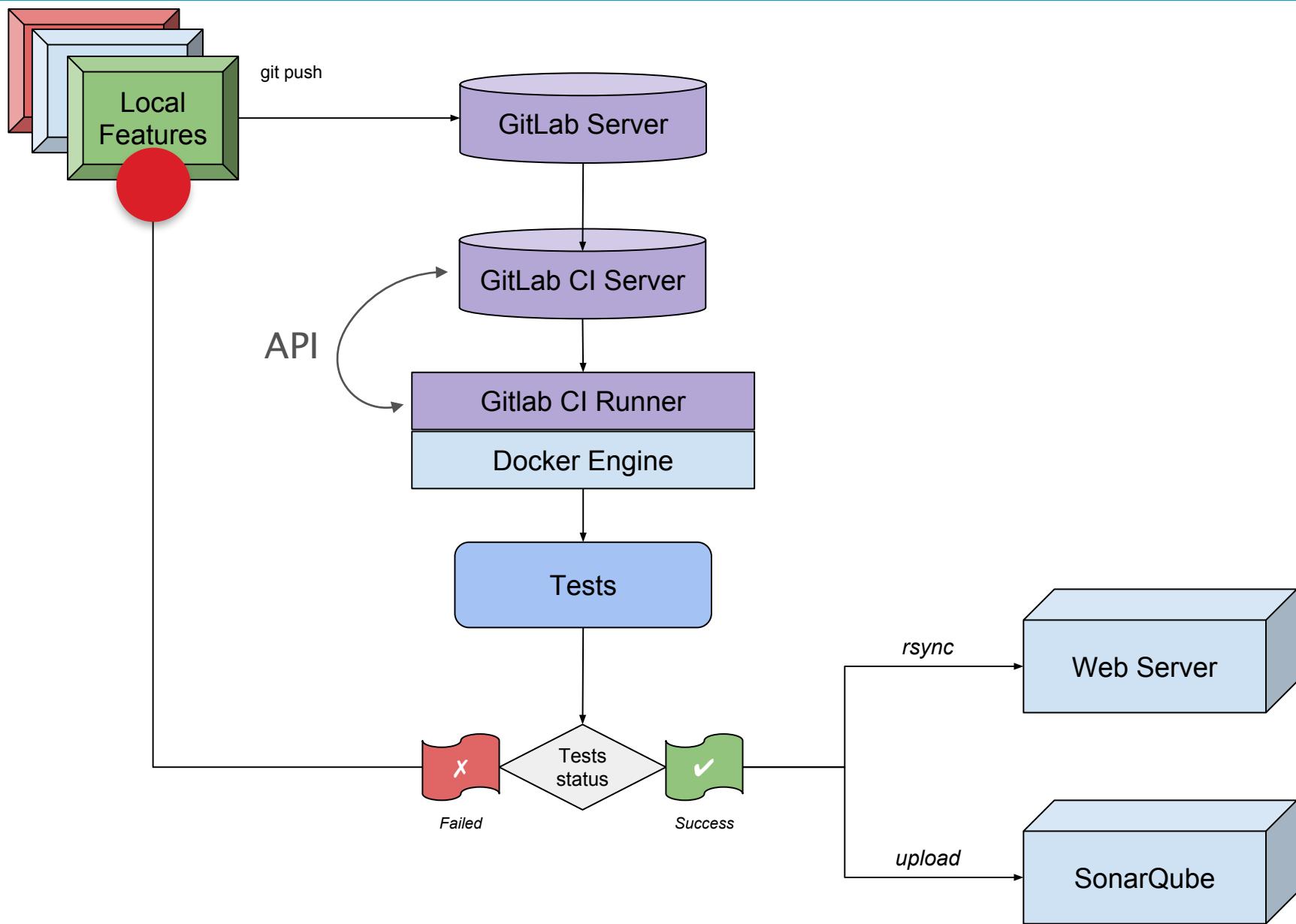


Schéma d'exemple de fonctionnement

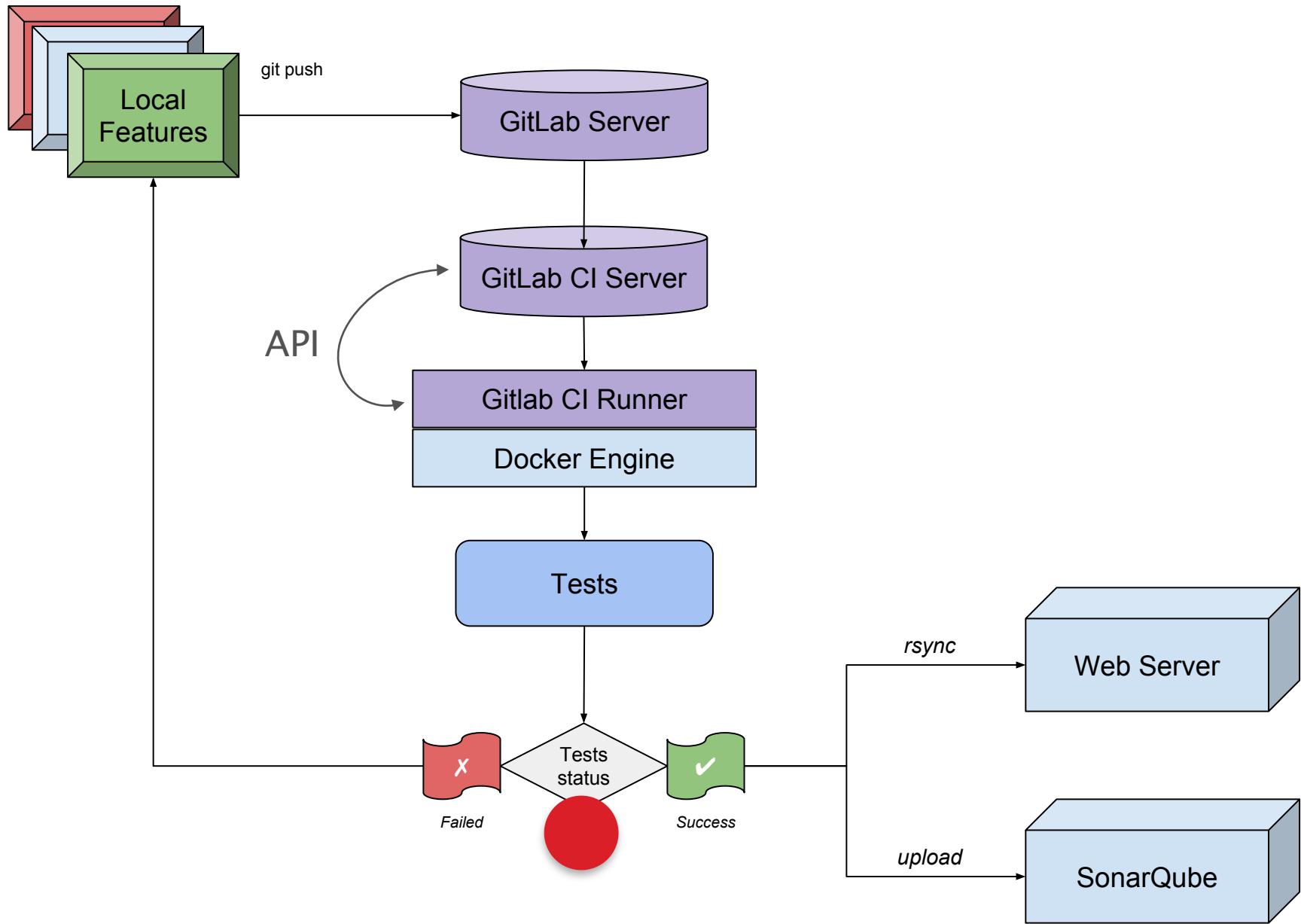


Schéma d'exemple de fonctionnement

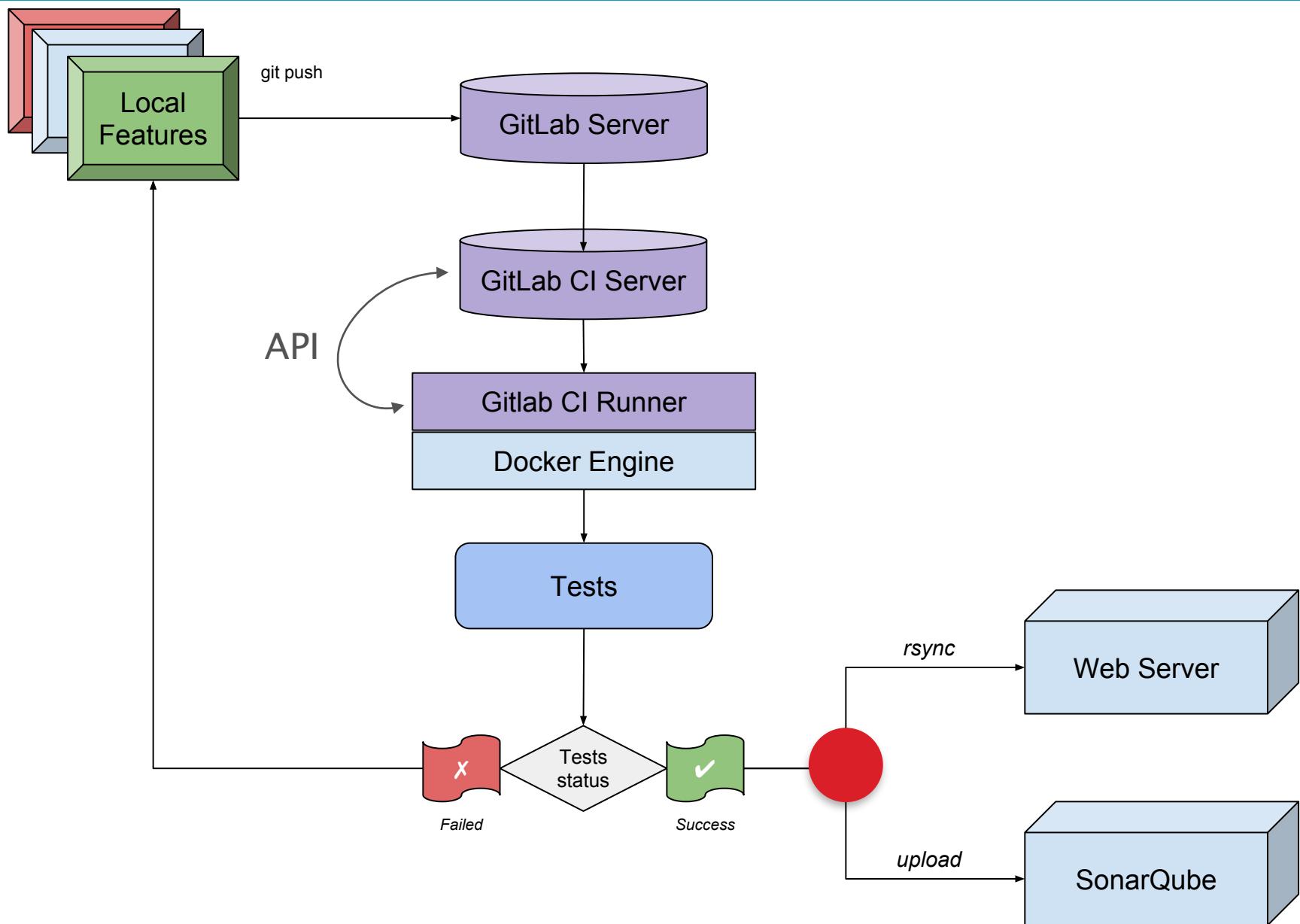
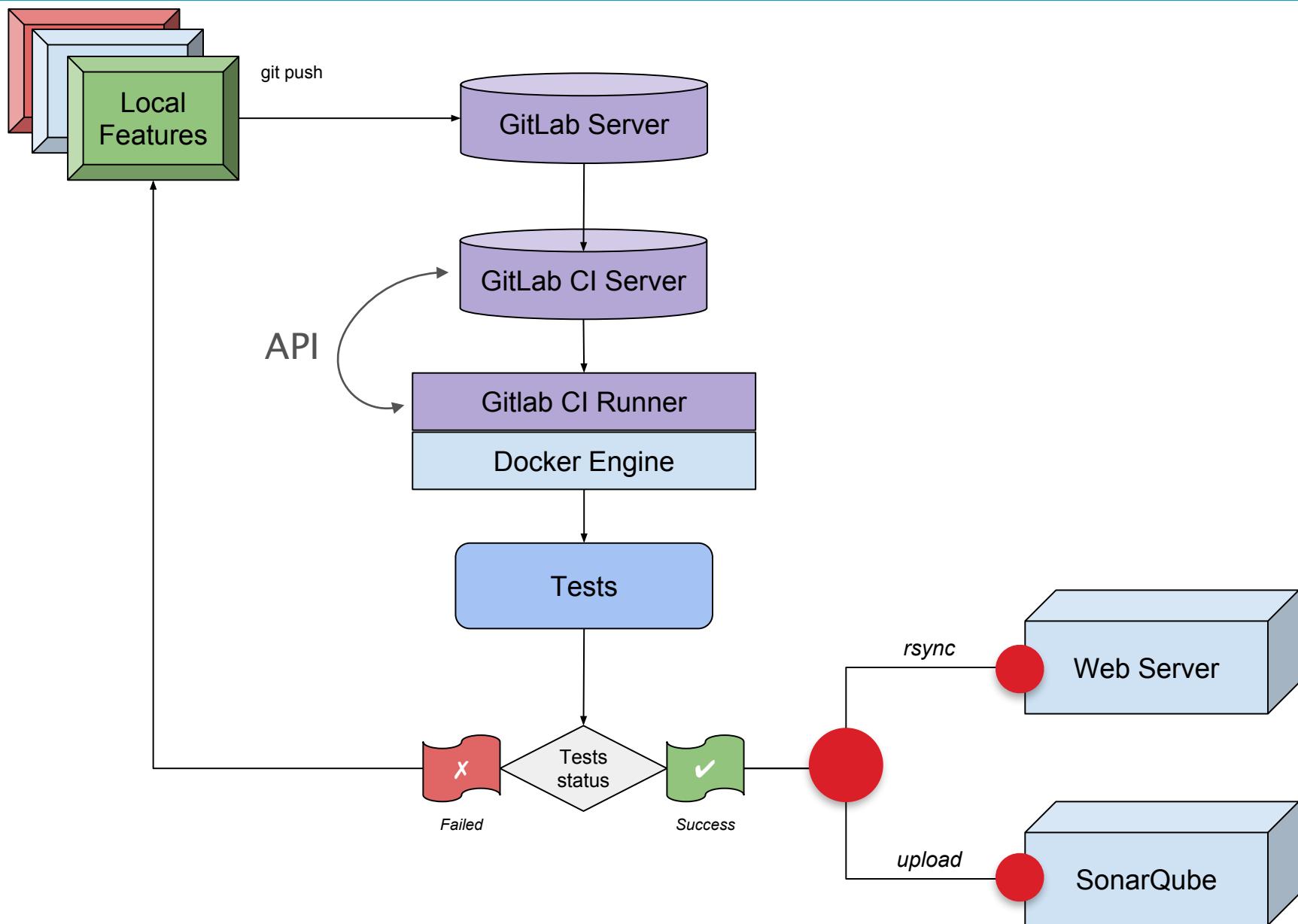


Schéma d'exemple de fonctionnement



.gitlab-ci.yml

```
image: ccin2p3/php-xdebug-sonar:latest

cache:
  paths:
    - vendor/

before_script:
  # run ssh-agent
  - eval $(ssh-agent -s)
  # add ssh key stored in SSH_PRIVATE_KEY variable to the agent store
  - ssh-add <(echo "$SSH_PRIVATE_KEY")
  [...]
  - php composer.phar install

stages:
  - test
  - deploy
```

```
image: ccin2p3/php-xdebug-sonar:latest
cache:
  paths:
    - vendor/
before_script:
  # run ssh-agent
  - eval $(ssh-agent -s)
  # add ssh key stored in SSH_PRIVATE_KEY variable to the agent store
  - ssh-add <(echo "$SSH_PRIVATE_KEY")
  [...]
  - php composer.phar install
stages:
  - test
  - deploy
```

Nom de l'image docker

```
image: ccin2p3/php-xdebug-sonar:latest

cache:
  paths:
    - vendor/

before_script:
  # run ssh-agent
  - eval $(ssh-agent -s)
  # add ssh key stored in SSH_PRIVATE_KEY variable to the agent store
  - ssh-add <(echo "$SSH_PRIVATE_KEY")
  [...]
  - php composer.phar install

stages:
  - test
  - deploy
```

Mise en cache de fichiers

```
image: ccin2p3/php-xdebug-sonar:latest

cache:
  paths:
    - vendor/

before_script: ← Script shell exécuté par le runner
  # run ssh-agent
  - eval $(ssh-agent -s)

  # add ssh key stored in SSH_PRIVATE_KEY variable to the agent store
  - ssh-add <(echo "$SSH_PRIVATE_KEY")  
[...]
  - php composer.phar install

stages:
  - test
  - deploy
```

```
image: ccin2p3/php-xdebug-sonar:latest

cache:
  paths:
    - vendor/

before_script:

# run ssh-agent
- eval $(ssh-agent -s)

# add ssh key stored in SSH_PRIVATE_KEY variable to the agent store
- ssh-add <(echo "$SSH_PRIVATE_KEY")

[...]

- php composer.phar install

stages:
  - test
  - deploy
```

Ordonnancement de l'exécution et
regroupement des jobs

```
image: ccin2p3/php-xdebug-sonar:latest

cache:
  paths:
    - vendor/

before_script:
  # run ssh-agent
  - eval $(ssh-agent -s)
  # add ssh key stored in SSH_PRIVATE_KEY variable to the agent store
  - ssh-add <(echo "$SSH_PRIVATE_KEY")
  [...]
  - php composer.phar install

stages:
  - test
  - deploy
```

.gitlab-ci.yml

```
job_phpunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phpunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}

job_phpunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    - php app/console doctrine:database:drop --env=test --force
    [...]
    - php phpunit.phar --configuration app/phpunit.xml.dist

job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
webcast2@webcast2.in2p3.fr:/www/htdocs
```

Nom des jobs

```
job_phpunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phpunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}

job_phpunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    - php app/console doctrine:database:drop --env=test --force
    [...]
    - php phpunit.phar --configuration app/phpunit.xml.dist

job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
      webcast2@webcast2.in2p3.fr:/www/htdocs
```

.gitlab-ci.yml

```
job_phpunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phpunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}

job_phpunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    - php app/console doctrine:database:drop --env=test -force
    [...]
    - php phpunit.phar --configuration app/phpunit.xml.dist

job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
webcast2@webcast2.in2p3.fr:/www/htdocs
```

.gitlab-ci.yml

```
job_phpunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phpunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}

job_phpunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    - php app/console doctrine:database:drop --env=test -force
    [...]
    - php phpunit.phar --configuration app/phpunit.xml.dist

job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
webcast2@webcast2.in2p3.fr:/www/htdocs
```

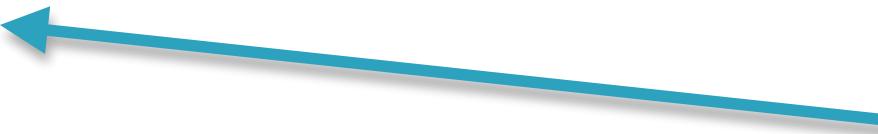
.gitlab-ci.yml

```
job_phunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}
```

```
job_phunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    [...]
    - php app/console doctrine:database:drop --env=test -force
    - php phunit.phar --configuration app/phpunit.xml.dist
```

```
job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
  webcast2@webcast2.in2p3.fr:/www/htdocs
```

Étape dans laquelle va être exécuté le job

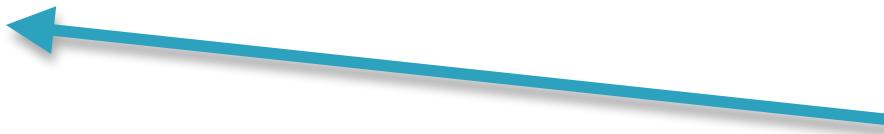


.gitlab-ci.yml

```
job_phunit_sonar:  
  stage: test  
  only:  
    - develop  
  tags:  
    - docker  
  script:  
    [...]  
    - php -dzend_extension=xdebug.so phunit.phar --configuration app/phpunit.xml.dist  
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}
```

```
job_phunit:  
  stage: test  
  only:  
    - tags  
  tags:  
    - docker  
  script:  
    [...]  
    - php app/console doctrine:database:drop --env=test -force  
    - php phunit.phar --configuration app/phpunit.xml.dist
```

```
job_deploy_prod:  
  stage: deploy  
  only:  
    - tags  
  script:  
    - echo "${PARAMETERS}" > app/config/parameters.yml  
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .  
      webcast2@webcast2.in2p3.fr:/www/htdocs
```



Étape dans laquelle va être exécuté le job

```
stages:  
  - test  
  - deploy
```

.gitlab-ci.yml

```
job_phunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}
```

```
job_phunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    [...]
    - php app/console doctrine:database:drop --env=test -force
    - php phunit.phar --configuration app/phpunit.xml.dist
```

```
job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
      webcast2@webcast2.in2p3.fr:/www/htdocs
```

Restriction d'exécution à
une ou plusieurs



.gitlab-ci.yml

```
job_phunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}
```

```
job_phunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    [...]
    - php app/console doctrine:database:drop --env=test -force
    - php phunit.phar --configuration app/phpunit.xml.dist
```

```
job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
  webcast2@webcast2.in2p3.fr:/www/htdocs
```

Choix du runner utilisé

.gitlab-ci.yml

```
job_phunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${{SONAR_JDBC_URL}} -Dsonar.jdbc.username=${{SONAR_JDBC_USERNAME}} -Dsonar.jdbc.password=${{SONAR_JDBC_PASSWORD}} -Dsonar.projectVersion=${{CI_BUILD_REF}}
```

Script shell exécuté par
le runner

```
job_phunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    - php app/console doctrine:database:drop --env=test --force
    [...]
    - php phunit.phar --configuration app/phpunit.xml.dist
```

```
job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
      webcast2@webcast2.in2p3.fr:/www/htdocs
```

.gitlab-ci.yml

```
job_phpunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phpunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}

job_phpunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    - php app/console doctrine:database:drop --env=test -force
    [...]
    - php phpunit.phar --configuration app/phpunit.xml.dist

job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
webcast2@webcast2.in2p3.fr:/www/htdocs
```

.gitlab-ci.yml

```
job_phpunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phpunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}

job_phpunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    - php app/console doctrine:database:drop --env=test -force
    [...]
    - php phpunit.phar --configuration app/phpunit.xml.dist

job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
webcast2@webcast2.in2p3.fr:/www/htdocs
```

.gitlab-ci.yml

```
job_phpunit_sonar:  
  stage: test  
  only:  
    - develop  
  tags:  
    - docker  
  script:  
    [...]  
    - php -dzend_extension=xdebug.so phpunit.phar --configuration app/phpunit.xml.dist  
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}
```

```
job_phpunit:  
  stage: test  
  only:  
    - tags  
  tags:  
    - docker  
  script:  
    [...]  
    - php app/console doctrine:database:drop --env=test  
    - php phpunit.phar --configuration app/phpunit.xml.dist
```

```
job_deploy_prod:  
  stage: deploy  
  only:  
    - tags  
  script:  
    - echo "${PARAMETERS}" > app/config/parameters.yml  
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .  
      webcast2@webcast2.in2p3.fr:/www/htdocs
```

Variables
d'environnement

.gitlab-ci.yml

```
job_phunit_sc
stage: test
only:
  - develop
tags:
  - docker
script:
  [...]
  - php -dzlib=1
  - /sonar-s
{SONAR_JDBC_UF
{SONAR_JDBC_PA
```

```
job_phunit:
stage: test
only:
  - tags
tags:
  - docker
script:
  - php app/
  [...]
  - php phpu
```

```
job_deploy_pro
stage: deploy
only:
  - tags
script:
  - echo "$@"
  - rsync -a
webcast2@webca
```

The screenshot shows the 'Secret Variables' section of a GitLab project settings page. It displays three environment variables:

Key	Value
PARAMETERS	# This file is auto-generated during the composer install parameters ..
SONAR_HOST_URL	http://ccsonar.in2p3.fr:9000
SONAR_JDBC_URL	jdbc:postgresql://ccpgsql.in2p3.fr/ccsonar

Each variable has a 'Remove this variable' button to its right.

it.xml.dist
ar.jdbc.url=\$
d=\$

es
ment

.gitlab-ci.yml

```
job_phpunit_sonar:
  stage: test
  only:
    - develop
  tags:
    - docker
  script:
    [...]
    - php -dzend_extension=xdebug.so phpunit.phar --configuration app/phpunit.xml.dist
    - /sonar-scanner-2.5/bin/sonar-runner -Dsonar.host.url=${SONAR_HOST_URL} -Dsonar.jdbc.url=${SONAR_JDBC_URL} -Dsonar.jdbc.username=${SONAR_JDBC_USERNAME} -Dsonar.jdbc.password=${SONAR_JDBC_PASSWORD} -Dsonar.projectVersion=${CI_BUILD_REF}

job_phpunit:
  stage: test
  only:
    - tags
  tags:
    - docker
  script:
    - php app/console doctrine:database:drop --env=test -force
    [...]
    - php phpunit.phar --configuration app/phpunit.xml.dist

job_deploy_prod:
  stage: deploy
  only:
    - tags
  script:
    - echo "${PARAMETERS}" > app/config/parameters.yml
    - rsync -az --delete --exclude=web/media --exclude=/web/cache -e "ssh" .
webcast2@webcast2.in2p3.fr:/www/htdocs
```

Exemple plus complexe

```
build:
  stage: build
  script:
    - ./gradlew assembleDebug
artifacts:
  paths:
    - app/build/outputs/
  unitTests:
    stage: test
    script:
      - ./gradlew test
  functionalTests:
    stage: test
    script:
      - wget --quiet --output-document=android-wait-for-emulator https://raw.githubusercontent.com/travis-ci/travis-cookbooks/0f497eb71291b52a703143c5cd63a217c8766dc9/community-cookbooks/android-sdk/files/default/android-wait-for-emulator
      - chmod +x android-wait-for-emulator
      - echo y | android-sdk-linux/tools/android --silent update sdk --no-ui --all --filter sys-img-x86-google_apis-$ANDROID_COMPILE_SDK
      - echo no | android-sdk-linux/tools/android create avd -n test -t android-$ANDROID_COMPILE_SDK --abi google_apis/x86
      - android-sdk-linux/tools/emulator64-x86 -avd test -no-window -no-audio &
      - ./android-wait-for-emulator
      - adb shell input keyevent 82
      - ./gradlew cAT
```

Troisième exemple

```
# This file is a template, and might need editing before it
works on your project.
# Full project: https://gitlab.com/pages/plain-html
pages:
  stage: deploy
  script:
    - mkdir .public
    - cp -r * .public
    - mv .public public
  artifacts:
    paths:
      - public
  only:
    - master
```

- ▶ Compte GitLab
- ▶ Fork du projet
- ▶ Ajout du .gitlab-ci.yml
 - Exécuter les tests
 - Documentation gitlab-ci
- ▶ Push
- ▶ Examiner le log du runner
- ▶ Bonus
 - Ajouter une variable pour la chaîne de connexion à mongodb

GitLab-CI toutes les options (enfin presque)

- ▶ Jobs
- ▶ image; services
- ▶ before_script; after_script
- ▶ stages
- ▶ script
- ▶ only; except
- ▶ tags
- ▶ when
- ▶ environment
- ▶ cache
- ▶ artifacts
- ▶ dependencies
- ▶ job templates
- ▶ Le reste sur : <https://docs.gitlab.com/ce/ci/yaml/README.html>

GitLab-CI exemples de la vie réelle

```
image: docker:latest

services:
- docker:dind

build:
  stage: build
  before_script:
    - docker login -u "gitlab-ci-token" -p "$CI_BUILD_TOKEN" $CI_REGISTRY
    - echo -n "${APIKEY}" > APIKEY.txt
    - sed -i "s/DB_CHARSET/${DB_CHARSET}/" settings.json
    - sed -i "s/DB_HOST/${DB_HOST}/" settings.json
    - sed -i "s/DB_NAME/${DB_NAME}/" settings.json
    - sed -i "s/DB_PASSWORD/${DB_PASSWORD}/" settings.json
    - sed -i "s/DB_USER/${DB_USER}/" settings.json
    - sed -i "s/REDIS_DB/${REDIS_DB}/" settings.json
    - sed -i "s/REDIS_HOST/${REDIS_HOST}/" settings.json
    - sed -i "s/REDIS_PORT/${REDIS_PORT}/" settings.json
    - sed -i "s/USER_ADMIN_PASSWORD/${USER_ADMIN_PASSWORD}/" settings.json
  script:
    - docker build --pull -t "$CI_REGISTRY_IMAGE:$CI_BUILD_TAG" .
    - docker push "$CI_REGISTRY_IMAGE:$CI_BUILD_TAG"
  only:
    - tags
```

```

---
stages:
  - unit
variables:
  RIEMANN_VERSION: 0.3.0
before_script:
  - apt update
  - apt install openjdk-8-jre-headless -y
  - wget -nv -O riemann.deb
https://github.com/riemann/riemann/releases/download/${RIEMANN_VERSION}/riemann_${RIEMANN_VERSION}_all.deb
  - dpkg -i riemann.deb
  - gem --version
  - gem install --no-ri --no-rdoc bundler
  - bundle -v
  - bundle install --without system_tests

.test-ruby: &test_ruby
stage: unit
variables:
  GITLAB_CI_PUPPET VERSIONS: "4.10.4 4.10.10 5.0.1 5.3.5 5.4.0 5.5.0"
script:
  - bundle exec puppet module build
  - bundle exec puppet module install pkg/*.tar.gz
  - ./run_test

test-ruby_2.4.1-puppet_4.10.4:
<<: *test_ruby
image: ruby:2.4.1-stretch
variables:
  GITLAB_CI_PUPPET VERSIONS: 4.10.4

test-ruby_2.4.1-puppet_4.10.10:
<<: *test_ruby
image: ruby:2.4.1-stretch
variables:
  GITLAB_CI_PUPPET VERSIONS: 4.10.10

test-ruby_2.4.1-puppet_5.0.1:
<<: *test_ruby
image: ruby:2.4.1-stretch
variables:
  GITLAB_CI_PUPPET VERSIONS: 5.0.1

test-ruby_2.4.1-puppet_5.4.0:
<<: *test_ruby
image: ruby:2.4.1-stretch
variables:
  GITLAB_CI_PUPPET VERSIONS: 5.4.0

test-ruby_2.4.1-puppet_5.5.0:
<<: *test_ruby
image: ruby:2.4.1-stretch
variables:
  GITLAB_CI_PUPPET VERSIONS: 5.5.0

test-ruby_2.5:
<<: *test_ruby
image: ruby:2.5-stretch

#.rvm:
# before_script:
#   - apt-get update
#   - apt-get install ruby wget ruby-dev libffi-dev gcc make -y
#   - curl -SSL https://rvm.io/mpapis.asc | gpg --import -
#   - curl -SSL https://get.rvm.io | bash -s stable
#   - . /etc/profile.d/rvm.sh
#   - rvm install $RUBY_VERSION
#   - rvm use $RUBY_VERSION
#
#   - gem --version
#   - gem update --system 2.7.4
#   - gem --version
#   - gem install --no-ri --no-rdoc bundler
#   - bundle -v

```

```
# - bundle install --without system_tests
# - bundle exec puppet module build
# - bundle exec puppet module install pkg/*.tar.gz
# - gem install --no-ri --no-rdoc pdk
# - wget -nv -O riemann.deb
https://github.com/riemann/riemann/releases/download/0.3.0/riemann\_0.3.0\_all.deb
# - dpkg -i riemann.deb
```

```

image: ccin2p3/php7-xdebug-sonar:latest

variables:
# Configure mysql environment variables (https://hub.docker.com/r/\_/mysql/)
MYSQL_DATABASE: hito
MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}

before_script:

# run ssh-agent
- eval $(ssh-agent -s)

# add ssh key stored in SSH_PRIVATE_KEY variable to the agent store
- ssh-add <(echo "$SSH_PRIVATE_KEY")

# disable host key checking (NOTE: makes you susceptible to man-in-the-middle attacks)
# WARNING: use only in docker container, if you use it with shell you will overwrite your
user's ssh config
- mkdir -p ~/.ssh
- echo -e "Host *\n\tStrictHostKeyChecking no\n\n" > ~/.ssh/config

# Set up php.ini
- echo "date.timezone=Europe/Paris" > /usr/local/etc/php/conf.d/timezone.ini
- echo "memory_limit=8G" > /usr/local/etc/php/conf.d/memory.ini
- echo "opcache.enable_cli=1" > /usr/local/etc/php/conf.d/opcache.ini
- echo "opcache.revalidate_freq=500" >> /usr/local/etc/php/conf.d/opcache.ini
- echo "opcache.memory_consumption=128" >> /usr/local/etc/php/conf.d/opcache.ini
- echo "opcache.interned_strings_buffer=8" >> /usr/local/etc/php/conf.d/opcache.ini
- echo "opcache.max_accelerated_files=4000" >> /usr/local/etc/php/conf.d/opcache.ini
- echo "opcache.fast_shutdown=1" >> /usr/local/etc/php/conf.d/opcache.ini
- rm -f /usr/local/etc/php/conf.d/xdebug.ini

stages:
- install
- test
- sonar
- deploy

job_install:
stage: install
only:
- tags
- develop
script:
# Install composer
- curl -sS https://getcomposer.org/installer | php
- php composer.phar global require hirak/prestissimo
- php composer.phar install
# Install node and bower
- ln -s /usr/bin/nodejs /usr/bin/node
- npm install -g bower
- /usr/local/lib/node_modules/bower/bin/bower install --allow-root
# Install CKEditor
- app/console ckeditor:install
- app/console assets:install web
artifacts:
paths:
- vendor/
- web/assets/
- web/bundles/
- app/bootstrap.php.cache
- app/SymfonyRequirements.php
- app/cache/
- bin/
- composer.phar

# TEST UNITAIRES
job_phunit_unitaire_develop:
stage: test
services:
- mysql:5.6
only:
- tags

```

```

- develop
script:
- echo ${ROLE_TEST}
- echo "${PARAMETERS_TEST}" > app/config/parameters.yml
- app/console doctrine:schema:create --env=test
- app/console doctrine:fixtures:load --env=test
- app/console hito:create-view-agent --env=test
- app/console hito:create-user correia philippe.correia@cc.in2p3.fr
ROLE_SUPER_ADMIN --env=test
- bin/phpunit --configuration app/phpunit_unitaire.xml
dependencies:
- job_install

# TEST FONCTIONNELS - ROLE SUPER ADMIN
job_phunit_fonctionnel_develop_ROLE_SUPER_ADMIN:
stage: test
services:
- mysql:5.6
only:
- tags
- develop
variables:
ROLE_TEST: "ROLE_SUPER_ADMIN"
script:
- echo ${ROLE_TEST}
- echo "${PARAMETERS_TEST}" > app/config/parameters.yml
- app/console doctrine:schema:create --env=test
- app/console doctrine:fixtures:load --env=test
- app/console hito:create-view-agent --env=test
- app/console hito:create-user correia philippe.correia@cc.in2p3.fr
ROLE_SUPER_ADMIN --env=test
- bin/phpunit --configuration app/phpunit_fonctionnel.xml
dependencies:
- job_install

# TEST FONCTIONNELS - ROLE SRH
job_phunit_fonctionnel_develop_ROLE_SRH:
stage: test
services:
- mysql:5.6
only:
- tags
- develop
variables:
ROLE_TEST: "ROLE_SRH"
script:
- echo ${ROLE_TEST}
- echo "${PARAMETERS_TEST}" > app/config/parameters.yml
- app/console doctrine:schema:create --env=test
- app/console doctrine:fixtures:load --env=test
- app/console hito:create-view-agent --env=test
- app/console hito:create-user correia philippe.correia@cc.in2p3.fr
ROLE_SUPER_ADMIN --env=test
- bin/phpunit --configuration app/phpunit_fonctionnel.xml
dependencies:
- job_install

# TEST FONCTIONNELS - ROLE_ADMINISTRATIF
job_phunit_fonctionnel_develop_ROLE_ADMINISTRATIF:
stage: test
services:
- mysql:5.6
only:
- tags
- develop
variables:
ROLE_TEST: "ROLE_ADMINISTRATIF"
script:
- echo ${ROLE_TEST}
- echo "${PARAMETERS_TEST}" > app/config/parameters.yml
- app/console doctrine:schema:create --env=test
- app/console doctrine:fixtures:load --env=test
- app/console hito:create-view-agent --env=test
- app/console hito:create-user correia philippe.correia@cc.in2p3.fr
ROLE_SUPER_ADMIN --env=test
- bin/phpunit --configuration app/phpunit_fonctionnel.xml

```

```
- bin/phpunit --configuration app/phpunit_fonctionnel.xml
dependencies:
- job_install

# TEST FONCTIONNELS - ROLE_RESP
job_phpunit_fonctionnel_develop_ROLE_RESP:
stage: test
services:
- mysql:5.6
only:
- tags
- develop
variables:
ROLE_TEST: "ROLE_RESP"
script:
- echo ${ROLE_TEST}
- echo "${PARAMETERS_TEST}" > app/config/parameters.yml
- app/console doctrine:schema:create --env=test
- app/console doctrine:fixtures:load --env=test
- app/console hito:create-view-agent --env=test
- app/console hito:create-user correia philippe.correia@cc.in2p3.fr
ROLE_SUPER_ADMIN --env=test
- bin/phpunit --configuration app/phpunit_fonctionnel.xml
dependencies:
- job_install

# TEST FONCTIONNELS - ROLE_COFO
job_phpunit_fonctionnel_develop_ROLE_COFO:
stage: test
services:
- mysql:5.6
only:
- tags
- develop
variables:
ROLE_TEST: "ROLE_COFO"
script:
- echo ${ROLE_TEST}
- echo "${PARAMETERS_TEST}" > app/config/parameters.yml
- app/console doctrine:schema:create --env=test
- app/console doctrine:fixtures:load --env=test
- app/console hito:create-view-agent --env=test
- app/console hito:create-user correia philippe.correia@cc.in2p3.fr
ROLE_SUPER_ADMIN --env=test
- bin/phpunit --configuration app/phpunit_fonctionnel.xml
dependencies:
- job_install

# TEST FONCTIONNELS - ROLE_AP
job_phpunit_fonctionnel_develop_ROLE_AP:
stage: test
services:
- mysql:5.6
only:
- tags
- develop
variables:
ROLE_TEST: "ROLE_AP"
script:
- echo ${ROLE_TEST}
- echo "${PARAMETERS_TEST}" > app/config/parameters.yml
- app/console doctrine:schema:create --env=test
- app/console doctrine:fixtures:load --env=test
- app/console hito:create-view-agent --env=test
- app/console hito:create-user correia philippe.correia@cc.in2p3.fr
ROLE_SUPER_ADMIN --env=test
- bin/phpunit --configuration app/phpunit_fonctionnel.xml
dependencies:
- job_install

# TEST FONCTIONNELS - ROLE_AGENT
job_phpunit_fonctionnel_develop_ROLE_AGENT:
stage: test
services:
- mysql:5.6
only:
```

```

tags
- develop
variables:
  ROLE_TEST: "ROLE_AGENT"
script:
  - echo ${ROLE_TEST}
  - echo "${PARAMETERS_TEST}" > app/config/parameters.yml
  - app/console doctrine:schema:create --env=test
  - app/console doctrine:fixtures:load --env=test
  - app/console hito:create-view-agent --env=test
  - app/console hito:create-user correia philippe.correia@cc.in2p3.fr
ROLE_SUPER_ADMIN --env=test
  - bin/phpunit --configuration app/phpunit_fonctionnel.xml
dependencies:
- job_install

job_phunit_sonar:
stage: test
services:
- mysql:5.6
only:
- tags
- develop
script:
  - echo "${PARAMETERS_TEST}" > app/config/parameters.yml
  - app/console doctrine:schema:create --env=test
  - app/console doctrine:fixtures:load --env=test
  - app/console hito:create-view-agent --env=test
  - app/console hito:create-user correia philippe.correia@cc.in2p3.fr
ROLE_SUPER_ADMIN --env=test
  - echo "zend_extension=xdebug.so" > /usr/local/etc/php/conf.d/xdebug.ini
  - bin/phpunit --configuration app/phpunit_coverage.xml
  - /sonar-scanner-2.8/bin/sonar-scanner -Dsonar.host.url=${SONAR_HOST_URL} -
Dsonar.login=${SONAR_LOGIN_TOKEN} -Dsonar.projectVersion=${CI_BUILD_REF}
dependencies:
- job_install
when: manual

#Déploiement en préprod
job_deploy_preprod:
stage: deploy
only:
- tags
script:
  - echo "${PARAMETERS_PREPROD}" > app/config/parameters.yml
  - app/console cache:clear --env=prod
  - app/console hito:create-view-agent --env=prod
  - app/console --no-interaction doctrine:migrations:migrate
  - rsync -az --delete -e "ssh" . hitoprep@hitopreprod.in2p3.fr:/www/htdocs
  - rm -rf app/cache/*
  - rsync -az --delete -e "ssh" . hitoprep@hitopreprod.in2p3.fr:/www/htdocs
dependencies:
- job_install
environment: preproduction

#Déploiement pour le CC
job_deploy_prod_cc:
stage: deploy
only:
- tags
script:
  - echo "${PARAMETERS}" > app/config/parameters.yml
  - app/console cache:clear --env=prod
  - app/console hito:create-view-agent --env=prod
  - app/console --no-interaction doctrine:migrations:migrate
  - rsync -az --delete -e "ssh" . hito@hito.in2p3.fr:/www/htdocs
  - rm -rf app/cache/*
  - rsync -az --delete -e "ssh" . hito@hito.in2p3.fr:/www/htdocs
dependencies:
- job_install
when: manual
environment: production

#Déploiement pour l'ipnl
job_deploy_prod_ipnl:

```

```
stage: deploy
only:
  - tags
script:
  - echo "${PARAMETERS_IPNL}" > app/config/parameters.yml
  - app/console cache:clear --env=prod
  - app/console hito:create-view-agent --env=prod
  - app/console --no-interaction doctrine:migrations:migrate
  - rsync -az --delete -e "ssh" . hitoipnl@hito-ipnl.in2p3.fr:/www/htdocs
  - rm -rf app/cache/*
  - rsync -az --delete -e "ssh" . hitoipnl@hito-ipnl.in2p3.fr:/www/htdocs
dependencies:
- job_install
when: manual
environment: production_ipnl
```

- ▶ Nous allons construire une image avec notre application pour la distribuer
- ▶ Docker In Docker
- ▶ Les jobs tournent dans un docker donc on va utiliser un autre service docker pour exécuter les commandes docker.
 - build
 - push

Construction d'image

```
# This file is a template, and might need editing before it works on your project.
# Official docker image.
image: docker:latest

services:
- docker:dind

before_script:
- docker login -u "$CI_REGISTRY_USER" -p "$CI_REGISTRY_PASSWORD" $CI_REGISTRY

build-master:
stage: build
script:
- docker build --pull -t "$CI_REGISTRY_IMAGE" .
- docker push "$CI_REGISTRY_IMAGE"
only:
- master

build:
stage: build
script:
- docker build --pull -t "$CI_REGISTRY_IMAGE:$CI_COMMIT_REF_SLUG" .
- docker push "$CI_REGISTRY_IMAGE:$CI_COMMIT_REF_SLUG"
except:
- master
```

- ▶ Tout est dit
 - Construire l'image de l'applications avec GitLab-CI
 - La mettre à disposition dans le registry lié à votre dépôt
- ▶ Bonus
 - Construire des images tagguées
 - Ajouter un docker-compose
 - Node app + Mongodb
 - Construire les 2 images dans le même registry
- ▶ Déploiement ready to go !