

**Intégration Continue**  
Tester, c'est douter

L'intégration continue est un ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée.

- *[Wikipédia](#)*

**Qu'est ce que l'intégration continue en pratique ?**

- Lancer des scripts via des Git hooks
- Exécuter des tests
- Analyses statiques de code
- ...
- Alerte si une étape échoue

## Les serveurs de CI

- Gitlab CI
- Circle CI
- Travis CI
- Jenkins
- ...



```
image: node:latest
```

```
# Pick zero or more services to be used on all builds.
```

```
# Only needed when using a docker container to run your tests in.
```

```
# Check out: http://docs.gitlab.com/ce/ci/docker/using\_docker\_images.html#what-is-a-service
```

```
services:
```

- mysql:latest
- redis:latest
- postgres:latest

```
# This folder is cached between builds
```

```
# http://docs.gitlab.com/ce/ci/yaml/README.html#cache
```

```
cache:
```

```
  paths:
```

- node\_modules/

```
test_async:
```

```
  script:
```

- npm install
- node ./specs/start.js ./specs/async.spec.js

```
test_db:
```

```
  script:
```

- npm install
- node ./specs/start.js ./specs/db-postgres.spec.js

**Pourquoi utiliser l'intégration continue?**

- tests: l'app fonctionne
- lint: le code est ok
- ...

La branche est prête a être mergée sans beaucoup de risque !



nat

# Déploiement Continue

Not like this....



1



2



3



4

---

Like this!



1



2



3



4



5

**Qu'est ce que le déploiement continue en pratique ?**

Faire un script qui fait les choses qui vont bien quand les tests vont bien

- *[putaindecode.io](https://putaindecode.io)*

- Lancer des scripts via des Git hooks
- Uniquement si la CI est ok
- Déployer le code

## Les environnements

- Dev
- Acceptance
- Pré-production
- Production

Il ne faut pas confondre la CI et la CD mais les deux sont complémentaires !