# 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



- 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
  - mysql:latest
  - redis:latest
  - postgres:latest
  - node_modules/
   npm install
   - node ./specs/start.js ./specs/async.spec.js
   npm install
   - node ./specs/start.js ./specs/db-postgres.spec.js
```

Pourquoi utiliser l'intégration continue?

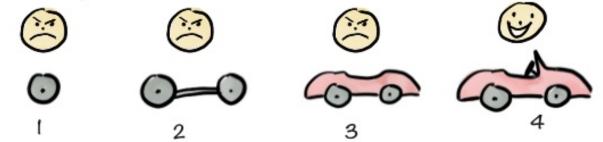
tests: l'app fonctionnelint: le code est ok

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

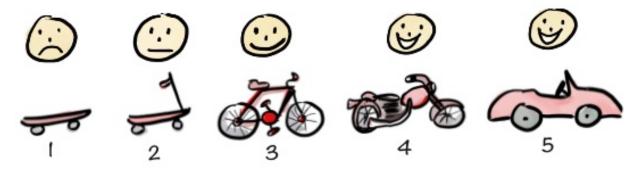
# **Déploiement Continue**

nat

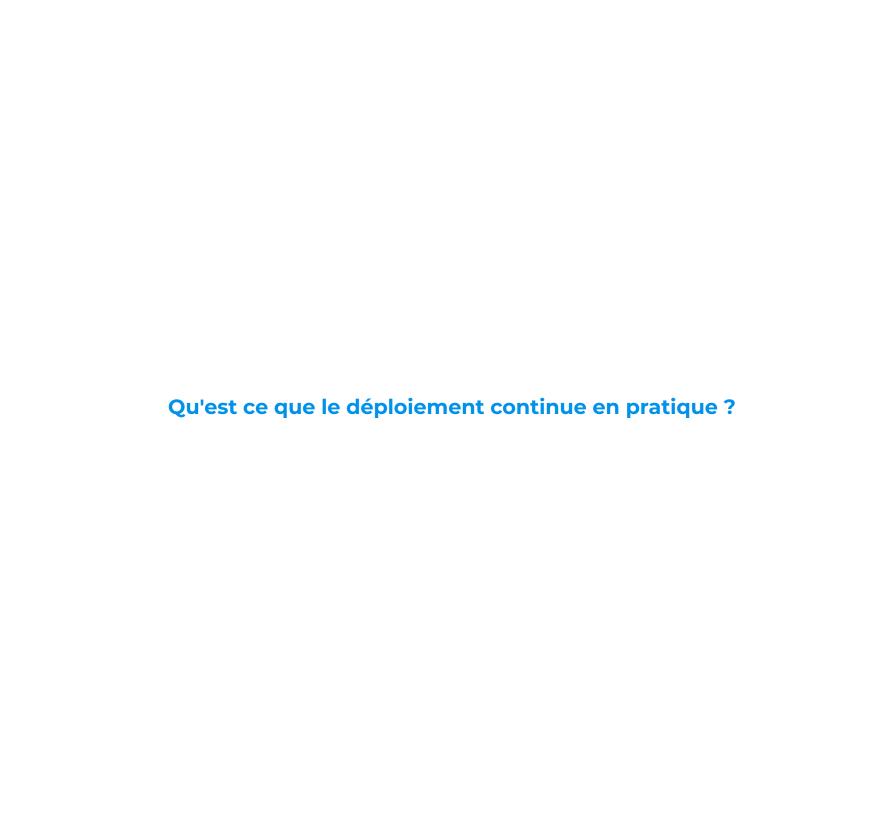
### Not like this....



## Like this!



Henrik Kniberg



Faire un script qui fait les choses qui vont bien quand les tests vont bien - putaindecode.io

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

#### **Les environnements**

- Dev
- Acceptance
- Pré-productionProduction

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