

La culture et la pratique du DevOps

DevOps : définition

"Le DevOps est **un mouvement** qui s'attaque au conflit existant structurellement entre le développement de logiciels et les opérations. Ce conflit résulte d'objectifs et de motivations divergents. Le DevOps améliore la collaboration entre les départements du développement et des opérations et rationalise l'ensemble de l'organisation. (Citation de Hüttermann2012 - Devops for developers)"

Le problème de la DSI traditionnelle

"In a nutshell, the conflict between development and operations is as follows:

- Need for change: Development produces changes (e.g., new features, bug fixes, and work based on change requests). They want their changes rolled out to production.
- Fear of change: Once the software is delivered, the operations department wants to avoid making changes to the software to ensure stable conditions for the production systems.

(Hüttermann2012)"

Histoire : l'origine du mouvement

- Patrick Debois a animé une session intitulée "Agile Operations and Infrastructure : How infra-gile are you ?" à la conférence Agile 2008 à Toronto et a publié un article portant un nom similaire.
- Il a ensuite inventé le terme DevOps en 2009 pour créer une série de conférences sur la vélocité des infrastructure.

Arbitrage DevOps

- Du côté des directions d'entreprise on **valorise souvent** la **vélocité** pour la compétitivité.

=> **La lenteur des opérations est un problème.**

Arbitrage DevOps

- Du côté des directions d'entreprise on **valorise souvent** la **vélocité** pour la compétitivité.

=> **La lenteur des opérations est un problème.**

- En même temps il faut maintenir la **qualité du service** sinon les clients s'en vont.

Arbitrage DevOps

- Le DevOps consiste à essayer d'**appliquer la méthode agile aux opérations** pour synchroniser les deux mondes Dev et Ops.

Arbitrage DevOps

- Le DevOps consiste à essayer d'**appliquer la méthode agile aux opérations** pour synchroniser les deux mondes Dev et Ops.
- Développer la vitesse et reactivité collaborative tout en conservant voir en améliorant la fiabilité des infrastructures.

Arbitrage DevOps

- Le DevOps consiste à essayer d'**appliquer la méthode agile aux opérations** pour synchroniser les deux mondes Dev et Ops.
- Développer la vitesse et reactivité collaborative tout en conservant voir en améliorant la fiabilité des infrastructures.
- Comment fait-on ?

Le DevOps est

un ensemble de solutions **humaines** et **techniques** liées

Solutions techniques

- Le but est de changer l'administration système pour accélérer les opérations et les rendre plus robustes à la fois.

Comment ?

En **fiabilisant** et **automatisant** les opérations avec de nouveaux outils et principes.

Solutions techniques

Quelques expressions que vous allez beaucoup entendre:

- Infrastructure as Code
- Containerisation
- Technologies de Cloud (infrastructures à la demande)
- CI / CD

CI / CD

(intégration continue et déploiement continu)

- Accélérer la livraison des nouvelles versions du logiciel.
- Des tests systématiques et automatisés pour ne pas se reposer sur la vérification humaine.
- Un déploiement progressif en parallèle (Blue Green) pour pouvoir automatiser le Rollback et être serein.
- A chaque étape le code passe dans un **Pipeline** de validation automatique.

Infrastructure as code

- Permet de régler un problème de l'administration système : Difficulté l'état du système à un instant T ce qui augmente les risques.
- Plutôt que d'appliquer des commandes puis d'oublier si on les a appliqué, On **décrit** le système d'exploitation (l'état du linux) dans un fichier et on utilise un système qui applique cette configuration explicite à tout moment.
- Permet aux Ops/adminsys de travailler comme des développeurs (avec une usine logicielle et ses outils)

Containerisation

Docker (et un peu LXC)

Il s'agit de mettre en quelques sortes les logiciels dans des boîtes :

- Avec tout ce qu'il faut pour qu'il fonctionnent (leurs dépendances).

Containerisation

Docker (et un peu LXC)

Il s'agit de mettre en quelques sortes les logiciels dans des boîtes :

- Avec tout ce qu'il faut pour qu'il fonctionnent (leurs dépendances).
- Ces boîtes sont fermées (on peut ne peut plus les modifier). On parle d'**immutabilité**.

Containerisation

Docker (et un peu LXC)

Il s'agit de mettre en quelques sortes les logiciels dans des boîtes :

- Avec tout ce qu'il faut pour qu'il fonctionnent (leurs dépendances).
- Ces boîtes sont fermées (on peut ne peut plus les modifier). On parle d'**immutabilité**.
- Si on a besoin d'une nouvelle version on fait **un nouveau modèle** de boîte. (on dit une nouvelle image docker)

Containerisation

Docker (et un peu LXC)

Il s'agit de mettre en quelques sortes les logiciels dans des boîtes :

- Avec tout ce qu'il faut pour qu'il fonctionnent (leurs dépendances).
- Ces boîtes sont fermées (on peut ne peut plus les modifier). On parle d'**immutabilité**.
- Si on a besoin d'une nouvelle version on fait **un nouveau modèle** de boîte. (on dit une nouvelle image docker)
- Cette nouvelle image permet de **créer autant d'instances que nécessaire**.

Containerisation - Pourquoi ?

- L'isolation des containers permet d'éviter que les logiciels s'emmêlent entre eux. (Les dépendances ne rentrent pas en conflit)

Containerisation - Pourquoi ?

- L'isolation des containers permet d'éviter que les logiciels s'emmêlent entre eux. (Les dépendances ne rentrent pas en conflit)
 - Les conteneurs non modifiables permettent de savoir exactement l'état de ce qu'on exécute sur l'ordinateur

=> Le risque de bug diminue énormément : **fiabilisation**

- L'agrandissement d'une infrastructure logiciel est beaucoup plus facile lorsqu'on a des boîtes autonomes qu'on peut multiplier.

Le cloud

Plutôt que d'**installer manuellement** de nouveaux serveurs linux pour faire tourner des logiciels on peut utiliser des outils pour **faire apparaître de nouveaux serveurs à la demande**.

Du coup on peut agrandir sans effort l'infrastructure de production pour délivrer une nouvelle version

C'est ce qu'on appelle le IaaS (Infrastructure as a service)

Renforcer la collaboration

Équipes transversales

Dans le cadre d'un produit logiciel, les administrateurs systèmes sont rassemblées avec le développement et le chef produit : tout le monde fait les réunions ensemble pour se parler et se comprendre.

Renforcer la collaboration

Équipes transversales

Dans le cadre d'un produit logiciel, les administrateurs systèmes sont rassemblées avec le développement et le chef produit : tout le monde fait les réunions ensemble pour se parler et se comprendre.

Culture de la polyvalence

- Les développeurs peuvent plus facilement créer un environnement réaliste pour jouer avec et comprendre comment fonctionne l'infrastructure de production (ils progressent dans l'administration système et la compréhension des enjeux opérationnels).

Renforcer la collaboration

Équipes transversales

Dans le cadre d'un produit logiciel, les administrateurs systèmes sont rassemblées avec le développement et le chef produit : tout le monde fait les réunions ensemble pour se parler et se comprendre.

Culture de la polyvalence

- Les développeurs peuvent plus facilement créer un environnement réaliste pour jouer avec et comprendre comment fonctionne l'infrastructure de production (ils progressent dans l'administration système et la compréhension des enjeux opérationnels).
- Les adminsys apprennent à programmer leurs opérations de façon puissante il deviennent donc plus proche de la logique des développeurs. (grâce à l'Infrastructure as Code)

Le profil DevOps

Par abus de langage on dit un ou une DevOps pour parler d'un métier spécifique dans une entreprise. Moronnei je dis que je suis DevOps sur mon CV par exemple.

Vous pouvez retenir :

Un DevOps c'est un **Administrateur Système agile** qui **programme ses outils**.

Le profil DevOps

Il faut être polyvalent : bien connaître l'administration système Linux mais aussi un peu la programmation et le développement.

Il faut connaître les nouvelles bonnes pratiques et les nouveaux outils cités précédemment.

Les prochain module

- Nous allons regarder quelques technologies typiques du devops qui servent à automatiser les déploiements
 - Ansible
 - Docker
 - Puppet
 - Jenkins
 - Elasticsearch

