

DevOps

01- Introduction

Jamel ESSOUSSI: Architecte logiciel

jamel.essoussi@gmail.com

<https://github.com/jessoussi>

2025

Présentation

- Organisation du cours
 - 10 heures de cours / 10 heures de TD
- Slides de cours seront disponibles sur:
<https://github.com/jessoussi>

Objectif du cours

Qualité du logiciel

- Des méthodes
- Des outils

Approche DevOps

- Intégration continue
- Livraison continue
- Déploiement continu

Plan du cours

1. Historique
2. La démarche DevOps
3. Concepts DevOps
4. Avantages du DevOps
5. Les pratiques DevOps
6. Outils du DevOps

1. Historique

Il était une fois ...

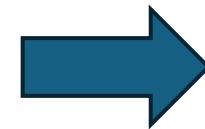
Avant

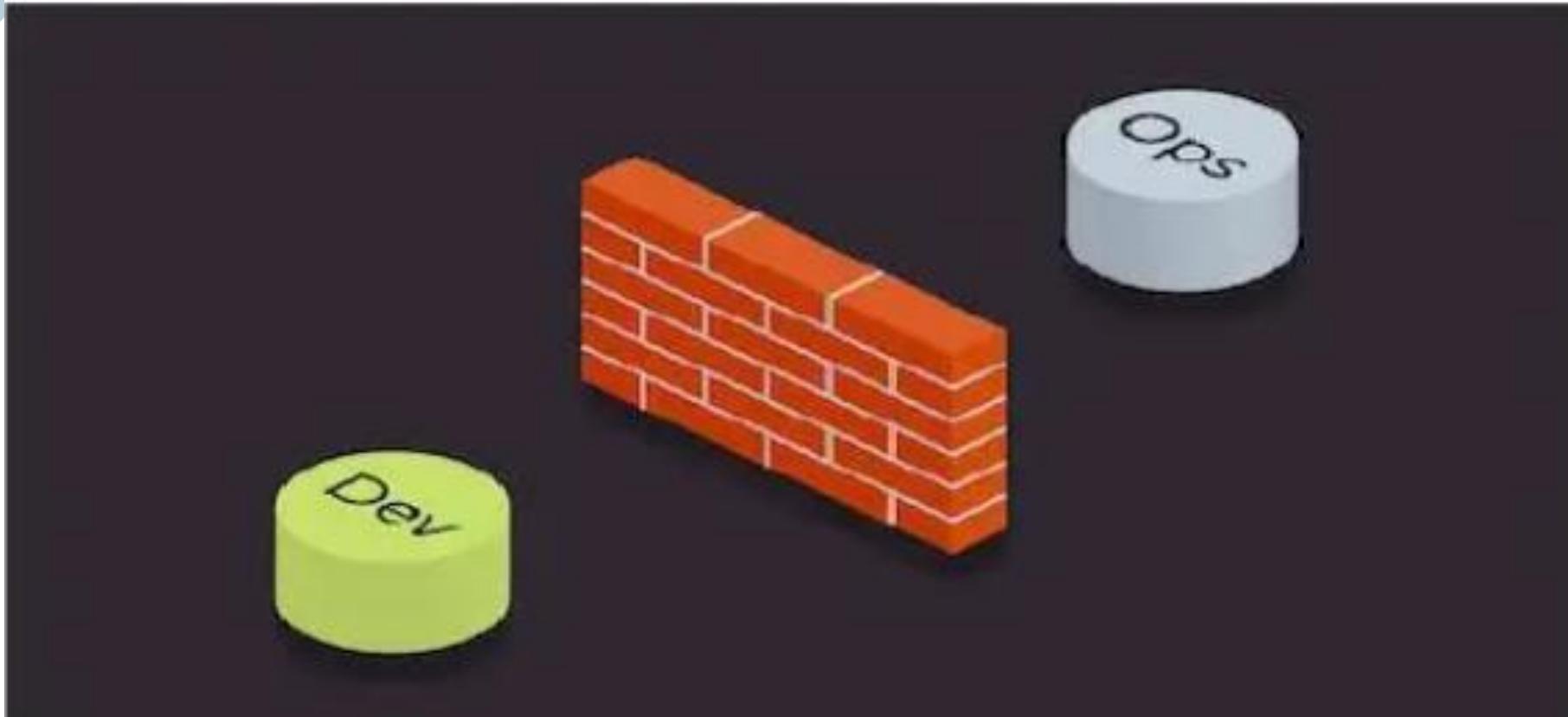


Analystes Programmeurs



Administrateurs système





Le mur de la confusion entre les équipes Dev et Ops

Analystes programmeurs

- ❑ Fonctionnalités
- ❑ Vélocités
- ❑ Nouveautés technologiques

Administrateurs système

- Stabilité (on parle de uptime)
- Scalabilité
- Capacité à réparer rapidement (MTTR)
- Maîtriser le changement
- Mesurer les impacts

Problèmes

- ❑ Mauvaise communication
- ❑ Chez moi ça marche!
- ❑ Moins de responsabilité chez un programmeur

Et puis en 2001

Manifeste Agile

Et puis en 2001

{ Nous découvrons de meilleures approches du développement logiciel en le pratiquant et en aidant les autres à le pratiquer }

Plan du cours

1. Historique
- 2. La démarche DevOps**
3. Concepts DevOps
4. Avantages du DevOps
5. Les pratiques DevOps
6. Outils du DevOps

2. La démarche DevOps: CAMS

- Culture
- Automation
- Measurement
- Sharing

2. La démarche DevOps

Culture

- Casser les murs entre les équipes
- Accepter le changement
- Transparence

2. La démarche DevOps

□ Automation

- Automate all the thinks
- One-click build
- One-click deploy

2. La démarche DevOps

□ Measurement

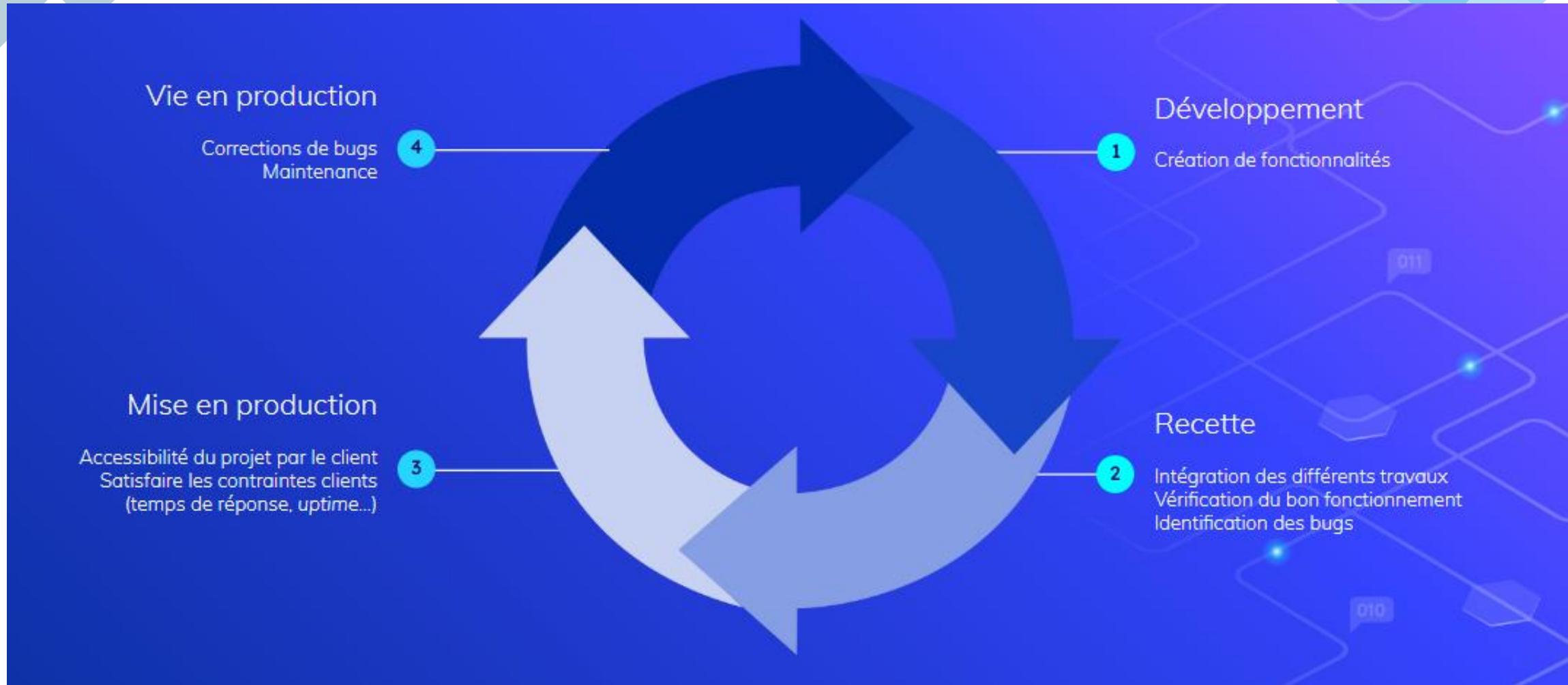
- Monitor all the thinks
- Ce qui ne peut être mesuré n'existe pas
- Distribution

2. La démarche DevOps

□ Sharing

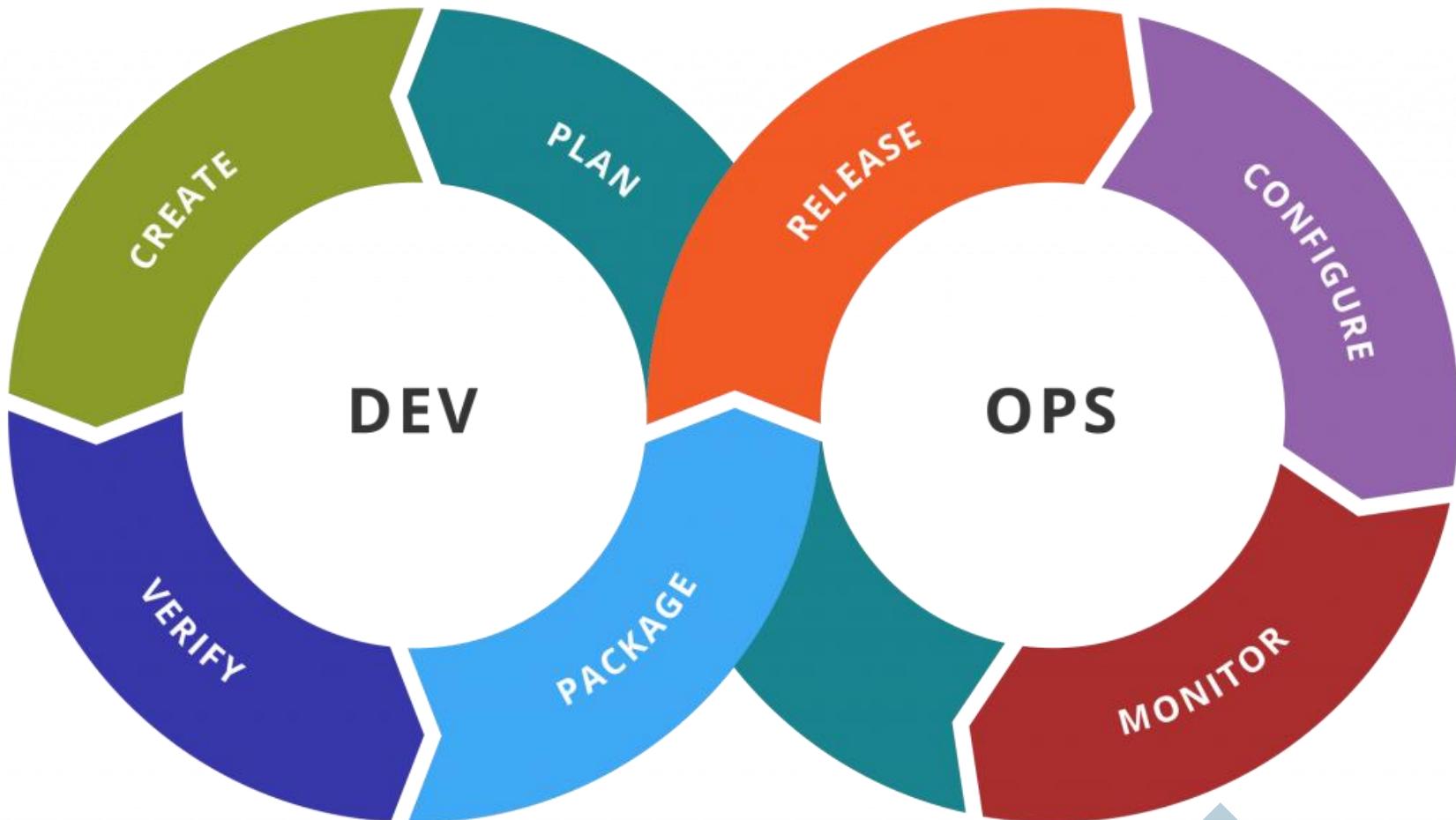
- Code
- Les outils
- Les connaissances

2. La démarche DevOps



DevOps Loop

Deux métiers en un



Plan du cours

1. Historique
2. La démarche DevOps
- 3. Concepts DevOps**
4. Avantages du DevOps
5. Les pratiques DevOps
6. Outils du DevOps

3. Concepts DevOps

□ La collaboration

Devops motive les personnes de différents départements à s'assembler et à réfléchir à la meilleure façon d'améliorer le flux de travail opérationnel d'un produit.

□ La vitesse

DevOps accélère la fréquence et la vitesse à laquelle les entreprises peuvent introduire de nouveaux produits sur le marché. Cette réduction de temps est liée à ce qu'on appelle TTM (Time-To-Market), DevOps accélère ce délai TTM grâce aux tests continues et à l'automatisation.

3. Concepts DevOps

□ TTM

Le TTM Time-To-Market est le temps d'arrivée d'une fonctionnalité sur le marché.

C'est le temps entre le moment de décision de la création de cette fonctionnalité, et son arrivée sur le produit final en production.

Plan du cours

1. Historique
2. La démarche DevOps
3. Concepts DevOps
- 4. Avantages du DevOps**
5. Les pratiques DevOps
6. Outils du DevOps

4. Avantages du DevOps

□ L'agilité

Les pratiques DevOps permettent à une organisation d'être plus flexible lorsqu'il s'agit d'équilibrer sa capacité en conséquence des fluctuations de la demande. L'adoption de DevOps améliore la façon dont la gestion des changements est effectuée et garantit qu'elle ne ralentit pas et n'interrrompt pas le processus en cours.

□ La satisfaction client

DevOps favorise l'amélioration continue de l'expérience client et sa satisfaction car, au bout du compte, l'objectif principale de DevOps est de fournir aux utilisateurs finaux des logiciels plus utiles et de meilleure qualité.

4. Avantages

❑ L'innovation

Le DevOps nourrit l'innovation en permettant aux équipes d'en savoir plus et de mieux comprendre les attentes des clients.

❑ Sécurité

Le DevSecOps suit la philosophie des améliorations itératives constantes, ce qui facilite frandement le processus de gestion de la sécurité. Il accélère également la vitesse de récupération si et quand des incidents se sécurité se prioduisent.

4. Avantages

Pour résumé, le DevOps permet d'améliorer la collaboration entre toutes les parties prenantes, de la planification à la livraison et à l'automatisation du processus de livraison afin de:

- **Améliorer la fréquence de déploiement,**
- **Accélérer la mise sur le marché,**
- **Réduire le taux d'échec de nouvelles livraisons,**
- **Raccourcir le délai entre les correctifs,**
- **Améliorer le temp moyen de récupération,**
- **S'adapter aux changements des besoins clients avec l'agilité,**
- **Posséder un avantage concurrentiel,**
- **Satisfaire les clients,**
- **Accroître l'innovation,**
- **Améliorer la sécurité.**

Plan du cours

1. Historique
2. La démarche DevOps
3. Concepts DevOps
4. Avantages du DevOps
- 5. Les pratiques DevOps**
6. Outils du DevOps

5. Les pratiques DevOps: Automatisation

Contrôle du code source

(SCM Source Code Management ou VCS Version Control Systems)

- Le contrôle du code source est un moyen pour garder trace et historiser les changements apportés à un code source et de collaborer sur ce dernier.
- C'est un élément essentiel pour le développement en général et indispensable pour le DevOps en particulier. Le contrôle du code source permet d'avancer rapidement et intelligemment. Il protège notamment le code source des dommages soudains, des erreurs humaines et des différents conflits.

5. Les pratiques DevOps

Test Continu

- A chaque nouvelle version de code on exécute une série de tests pour avoir des feedbacks le plus rapidement possible. Cela permet de vérifier la validité et la qualité du code avant d'aller en production. Le but est d'éviter toutes sortes de bugs ou problèmes dans l'environnement de production qui peut être très couteux.
- Trois catégories de tests se distinguent:
 - Tests fonctionnels: Tests unitaires, tests d'intégration et tests d'acceptation par l'utilisateur
 - Test non fonctionnels: déterminer les performances et l'endurance du système,
 - Test de maintenance

5. Les pratiques DevOps

Intégration Continue (CI)

- La meilleure façon de comprendre ce concept est de décortiquer son nom. La CI est avant tout continue. Cela signifie qu'il ne peut s'agir d'un processus manuel. L'intégration signifie l'intégration du code et l'intégration de nouvelles fonctionnalités. Il s'agit d'un processus consistant à combiner les mises à jour dans une base de code existant et de tester ces changements afin de ne pas introduire de nouveaux bugs.
- La CI nous permet donc de mettre en commun les différents travaux des développeurs et de s'assurer de la bonne exécution du build et des tests associés d'une façon continue et automatisée.

5. Les pratiques DevOps

Intégration Continue (CI)

- Objectifs:
 - Réduire les impacts des bugs,
 - Améliorer la qualité et la quantité des tests,
 - Augmenter la fréquence des releases,
 - Réduire les risques.

5. Les pratiques DevOps

Livraison Continue (Continus Delivery)

- Se base sur l'intégration continue, où chaque changement devient candidat d'être en production. Un pipeline d'intégration évalue le code, pour qu'ensuite on pourra potentiellement le déployer pour les clients.

5. Les pratiques DevOps

Déploiement Continu (Continus Deployment)

- Ce n'est que la livraison continue avec plus d'automatisation. Les clients finaux en production reçoivent directement les changements apportés au code. Il n'y a aucune intervention humaine. Seuls les tests ratés peuvent empêcher une nouvelle modification d'être mise en production.
- Le déploiement doit être totalement automatisé.
- Eviter l'arrêt de service.
- Automatisation des rollbacks si une erreur survient.

5. Les pratiques DevOps

Déploiement Continu (Continus Deployment) Stratégie de déploiement

Rolling Update	Blue/Green	Canary
Incrément serveur par serveur	Nouvelle version dans un groupe de serveur, redirection du traffic	Nouvelle version dans un groupe de serveur, redirection progressive du traffic

5. Les pratiques DevOps

Monitoring Continue (Continus Monitoring)

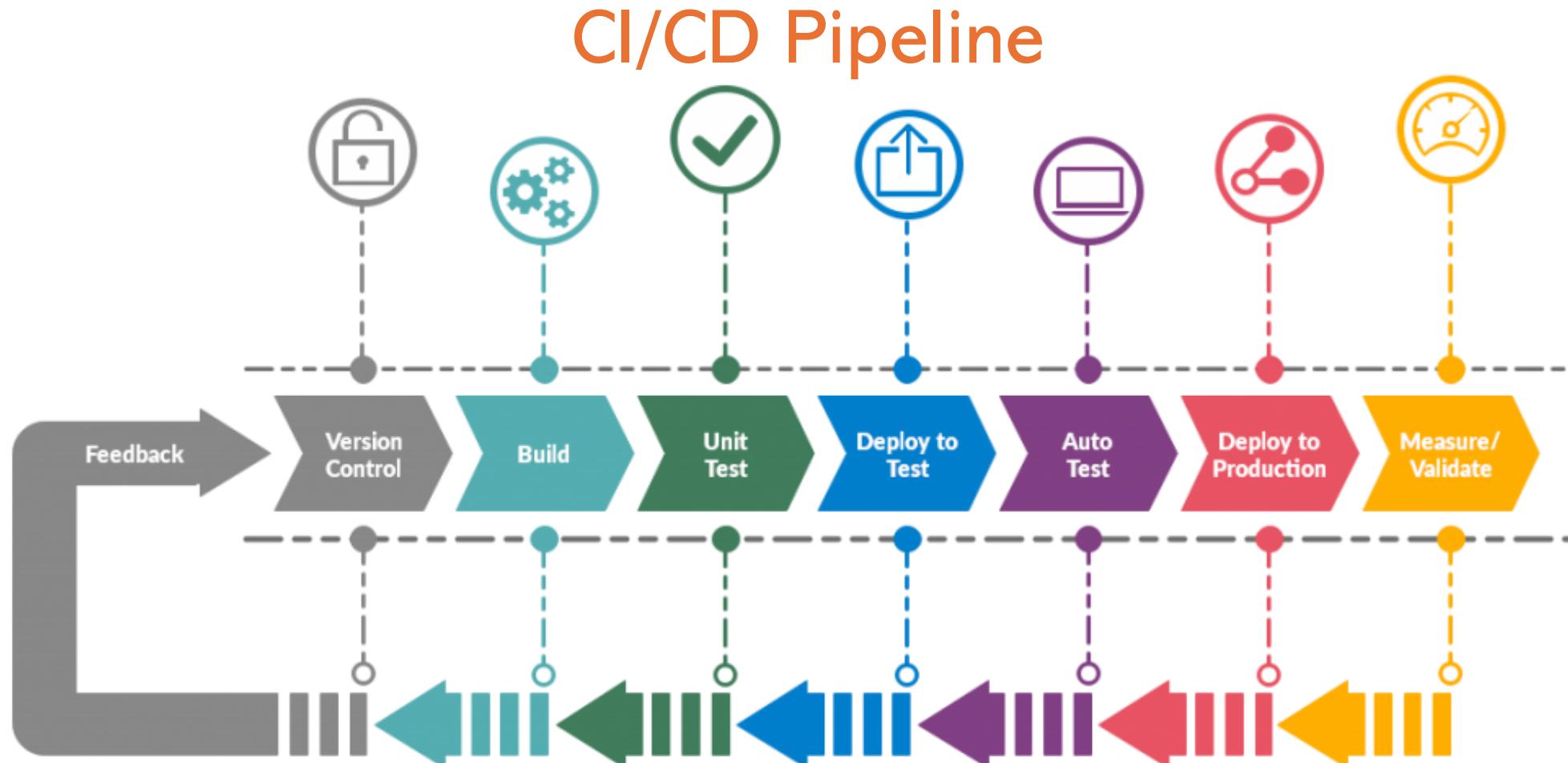
- Un processus automatisé par lequel le personnel DevOps peut observer et détecter les problèmes de conformité et les menaces pour la sécurité au cours de chaque phase du pipeline DevOps. Il aide les équipes à surveiller, détecter et étudier les principales mesures pertinentes, et à trouver des moyens de résoudre les problèmes en temps réel.
- Le continu monitoring se fait à trois niveaux différents:
 - Monitoring de l'infrastructure : surveille et gère l'infrastructure informatique.
 - Monitoring des applications : surveille les performances du logiciel.
 - Monitoring du réseau: Surveille et suit l'activité du réseau.

5. Les pratiques DevOps

CI/CD Pipeline

- Un pipeline DevOps est un ensemble de processus et d'outils automatisés qui permettent aux développeurs et aux experts des opérations de travailler de manière cohérente pour développer et déployer du code dans un environnement de production.

5. Les pratiques DevOps



5. Les pratiques DevOps

CI/CD Pipeline

❑ Exemples de pipeline:

- Build,
- Exécution des tests unitaires,
- Conteneurisation,
- Lancement d'un conteneur dans un environnement,
- Exécution des tests d'intégration,
- Exécution des tests de non régression,
- Release,
- Déploiement sur le serveur de recette,
- Déploiement sur le serveur de pré-production,
- Déploiement sur le serveur de production.

Plan du cours

1. Historique
2. La démarche DevOps
3. Concepts DevOps
4. Avantages du DevOps
5. Les pratiques DevOps
6. Outils du DevOps

6. Outils du DevOps

