DevSecOps

Définition
(9 slides)

Plan du cours

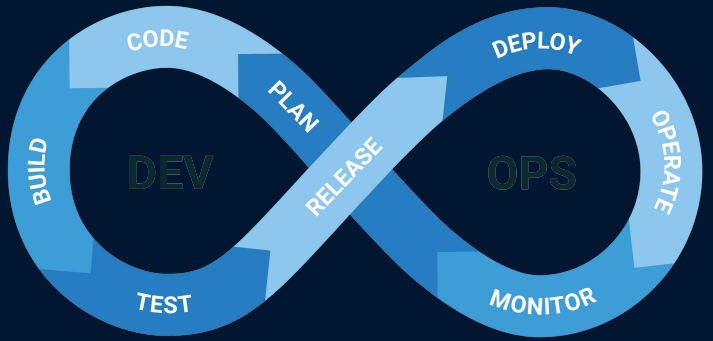
- **Le plan des 3 semaines :**
- DEV → semaine 1
- OPS → semaine 2
- SEC → semaine 3

Avant : Dev et Ops travaillaient séparément

- Le Dev travaillait sur son application
- L’Ops la déployait et la maintenait
- **Problème : chacun travaille dans son coin → communication faible, perte de temps, erreurs, etc**

Le DevOps : collaboration et automatisation

- DevOps rapproche Dev et Ops.
- Objectif : travailler ensemble, automatiser, livrer plus vite.
- **Idée clé : intégration (de code) continue (CI) et déploiement continu (CD).**

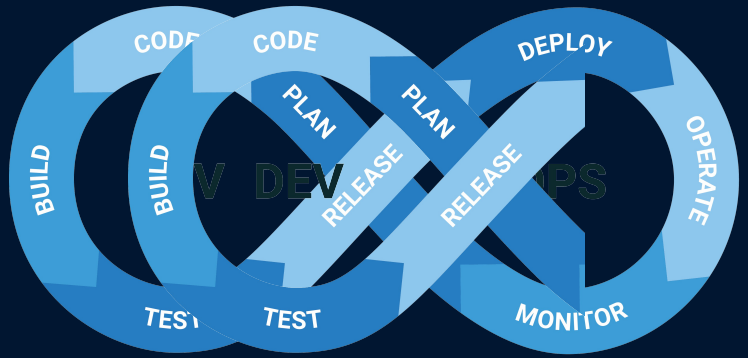


Continuous Integration (CI)

- **1. Plan (*ex : Jira*)**
  On crée ou améliore l'application
- **2. Code (*ex : VSCode*)**
  On code les fonctionnalités prévues
- **3. Build (*ex : python –m build / npm*)**
  On construit l'application

4. Test (*ex : PyTest*)
On teste que l'application fonctionne comme prévu

5. Release (*ex : GitHub Actions / GitLab CI*)
On prépare une version stable de l'application à mettre en ligne



Continuous Deployment (CD)

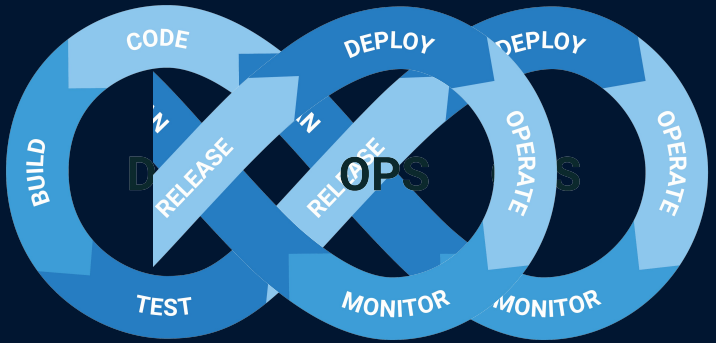
- **5. Release (*ex : GitHub Actions / GitLab CI*)**
  On prépare une version stable de l'application à mettre en ligne

- **6. Deploy (*ex : Docker / Kubernetes*)**
  On déploie cette version de l'application sur les serveurs

- **7. Operate (*ex : Kubernetes / AWS*)**
  On fait tourner l'application et on surveille qu'elle fonctionne bien

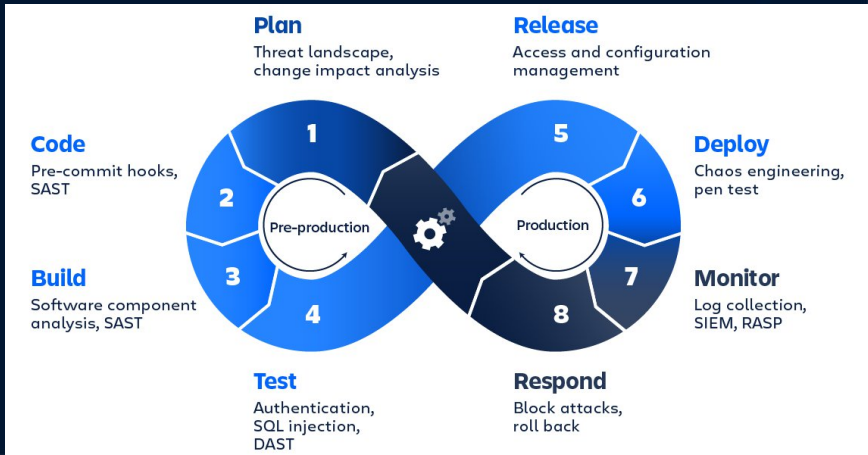
8. Monitor (*ex : Prometheus + Grafana*)
On collecte des informations sur l'application pour détecter les problèmes

9. Respond (*ex : Sentry*)
On réagit rapidement pour corriger les incidents ou les failles détectés



Alors, qu’est ce que le DevSecOps

- Avec DevOps, on va vite… parfois trop vite pour la sécurité
- Avant, la sécurité intervenait seulement à la fin → trop tard.
- DevSecOps insère la sécurité dès le début, et à chaque étape.
- Security by Design



Alors, qu’est ce que le DevSecOps

- **1. Plan : Threat Modeling Tool**
  Réfléchir aux risques de sécurité dès la conception

- **2. Code SAST : Semgrep + Secure Coding**
  Analyse le code pour détecter des failles

- **3. Build : Safety ou Pip-audit**
  Vérifie que les bibliothèques utilisées
  ne contiennent pas de vulnérabilités connues

4. Test : OWASP ZAP (DAST)
Simule des attaques sur l'application pour trouver des failles

5. Release : Image/Artifact Scan : Trivy
Analyse les images Docker ou artefacts avant leur mise en ligne.

Ce qu’il faut retenir

- DevOps = Communication Dev et Ops pour + d’efficacité

- DevSecOps = efficacité en sécurité

- **Le but n’est pas la perfection,**
  mais des petites améliorations continues

« Le but n’est pas la perfection, mais des petites améliorations continues. »