1️⃣ Planifier (Plan)

- Définir les besoins, les user stories, les fonctionnalités.
- Utiliser des outils comme Jira, Trello, Azure Boards.
- Objectif : aligner Dev et Ops dès la conception.

2️⃣ Développer (Code)

- Écrire le code source.
- Utiliser des systèmes de gestion de versions (Git, GitLab, GitHub).
- Mettre en place des **revues de code** et des **tests unitaires**.

3️⃣ Construire (Build)

- Compilation et assemblage des composants logiciels.
- Utiliser des outils CI/CD (Jenkins, GitLab CI, GitHub Actions).
- Génération automatique d'artifacts (packages, conteneurs Docker).

4️⃣ Tester (Test)

- Tests automatisés : unitaires, intégration, sécurité.
- Vérifier qualité, performance et conformité.
- Objectif : détecter les bugs avant la production.

5️⃣ Publier/Déployer (Release/Deploy)

- Automatisation du déploiement avec des pipelines CI/CD.
- Utilisation de Kubernetes, Ansible, Terraform pour l'infrastructure as code.

- Mise en place de stratégies Blue/Green, Canary pour réduire les risques.

6️⃣ Exploiter (Operate)

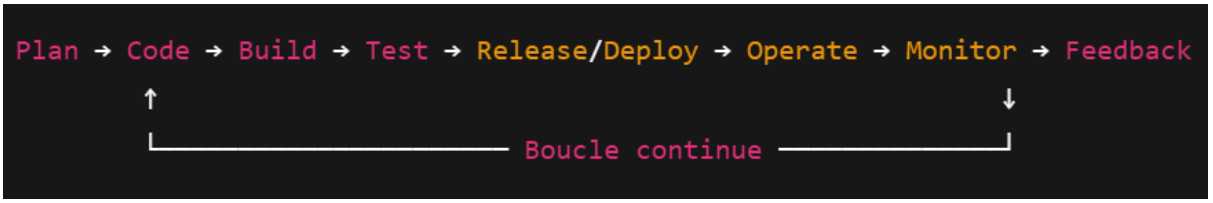
- Mise en production et gestion des environnements.
- Supervision de la disponibilité et du fonctionnement.
- Gestion des incidents et de la capacité.

7️⃣ Surveiller (Monitor)

- Collecte des métriques (logs, traces, APM).
- Utiliser des outils comme Prometheus, Grafana, ELK, Datadog.
- Détection proactive des problèmes.

8️⃣ Boucle de retour (Feedback & Improve)

- Analyse des données issues de la surveillance et des retours utilisateurs.
- Ajuster les prochaines versions.
- Cycle d’amélioration continue (Kaizen).



🔹 Points clés du cycle DevOps

- **Automatisation** : essentielle pour CI/CD et tests.
- **Collaboration** : Dev et Ops travaillent ensemble, plus de silos.
- **Mesure & Feedback** : tout est mesuré pour s'améliorer en continu.
- **Sécurité intégrée** : DevSecOps ajoute la sécurité à chaque étape.