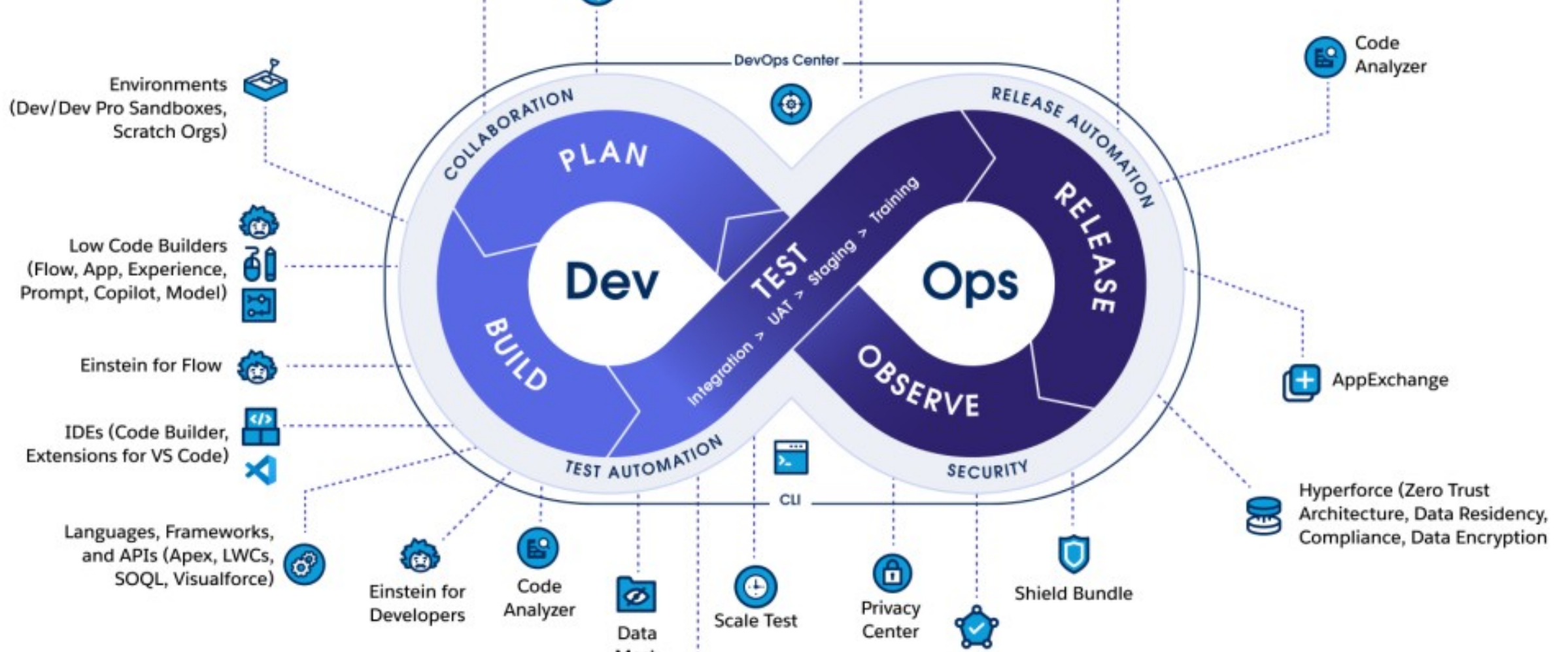




Qu'est-ce que DevOps ?

Une vue d'ensemble de la méthodologie DevOps et de ses principaux avantages pour le développement et le déploiement de logiciels.

Qu'est-ce que DevOps ?



DevOps, en termes simples

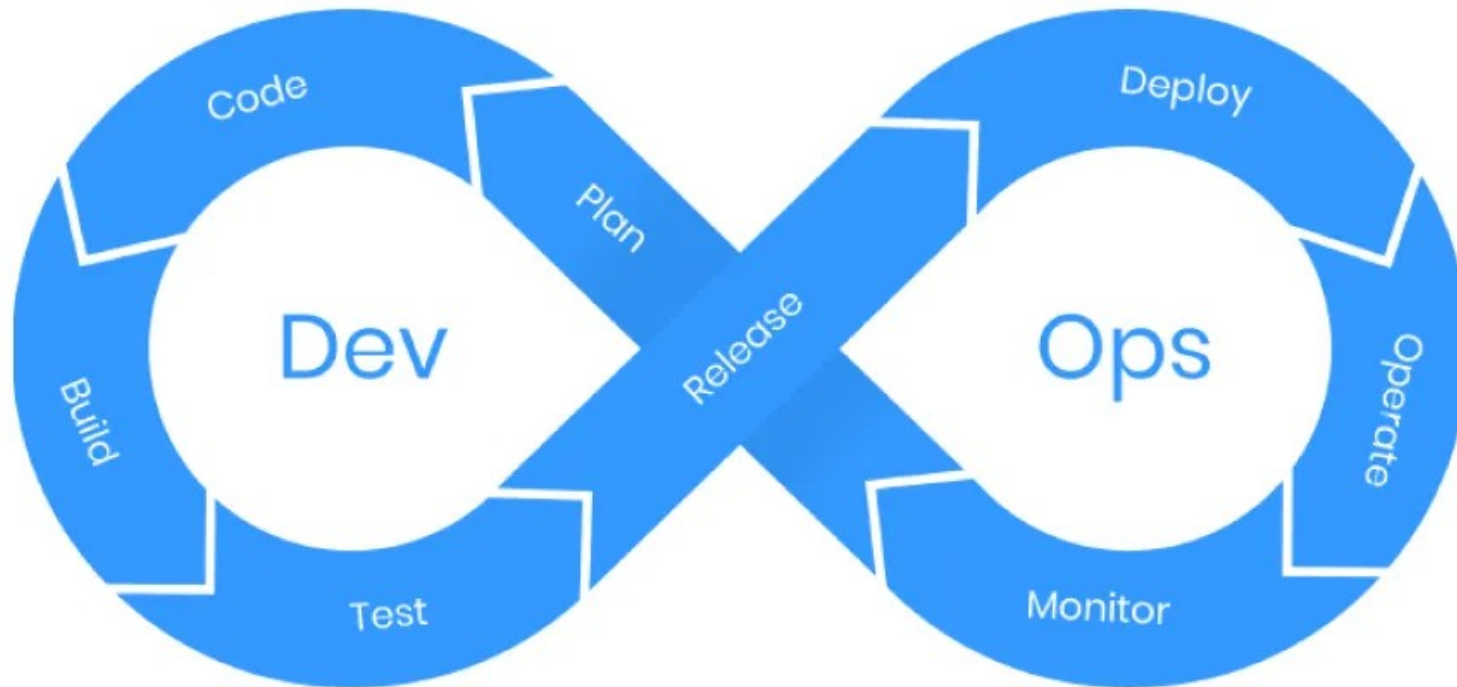
c'est "la façon dont vous concevez, construisez et déployez votre logiciel".



L'objectif

est de réunir les équipes de développement et d'exploitation autour d'un but commun et d'une vision unifiée.

DevOps n'est plus un facteur de différenciation, c'est un élément essentiel pour rester compétitif et survivre dans l'industrie logicielle.



Introduction to DevOps automation tools: A comprehensive overview

Définition (Méthodologie)

DevOps est une méthodologie de développement logiciel qui améliore la collaboration entre les équipes de développement et d'exploitation grâce à l'automatisation. Ces outils d'automatisation s'intègrent à chaque étape du cycle de vie DevOps.

Pourquoi DevOps ?



Pourquoi DevOps ?



Silos entre les équipes de développement et d'exploitation (Dev et Ops)

Les équipes Dev et Ops travaillaient en silo.



Manual processes

Les processus manuels ont créé des bottlenecks et des dépendances, ce qui a freiné la productivité.



Recurring issues

Des erreurs, des lacunes et des problèmes récurrents ont conduit à l'adoption de DevOps

DevOps a été adopté pour relever les défis posés par l'approche traditionnelle en cascade (ou cycle en V), dans laquelle les équipes de développement et d'exploitation travaillaient en silos, tandis que les processus manuels freinaient la productivité et entraînaient des problèmes récurrents.

Architecture applicative actuelle

L'équipe de développement planifie et conçoit les **fonctionnalités** ainsi que l'**architecture** de l'application.

Plan

L'équipe de développement rédige le **code source** de l'application.

Code

L'équipe de développement **compile le code** et crée les **artéfacts déployables** nécessaires à la mise en production de l'application.

Build

L'équipe de développement **teste l'application** afin de s'assurer qu'elle **répond aux exigences** et **fonctionne comme prévu**.

Test

Monitor

L'équipe d'exploitation **surveille l'application en cours d'exécution** afin de s'assurer qu'elle **fonctionne correctement**.

Release

L'équipe d'exploitation **prépare l'application pour sa mise en production**, en veillant à ce qu'elle soit **prête à être déployée**.

Deploy

L'équipe d'exploitation **déploie l'application** dans l'**environnement cible**.

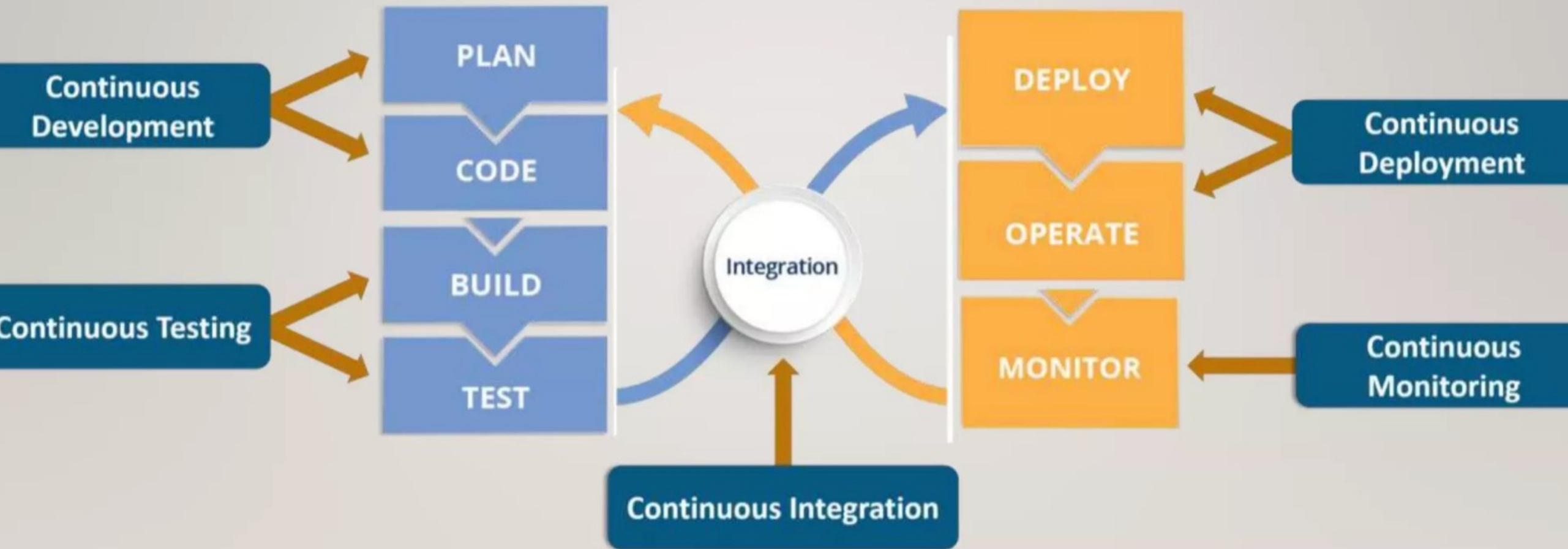
Operate

L'équipe d'exploitation **gère et maintient l'application en fonctionnement**, en veillant à ce qu'elle **continue de fonctionner correctement**.

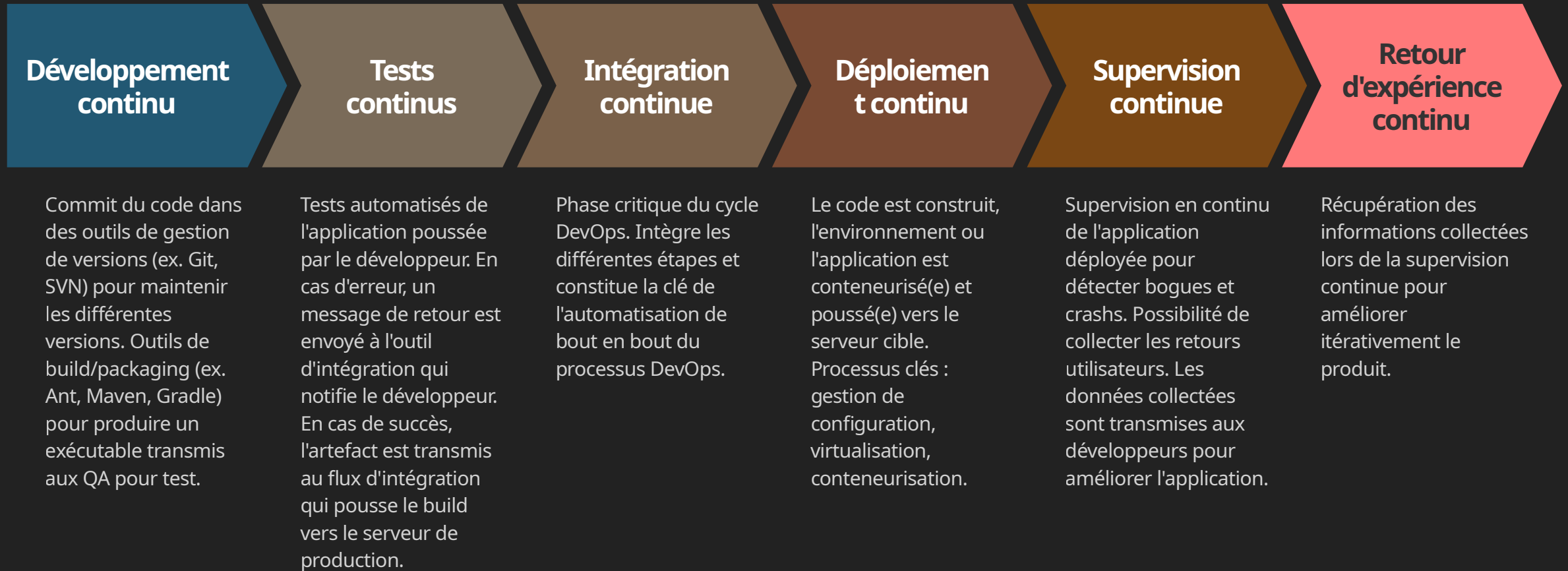
Present Application Architecture



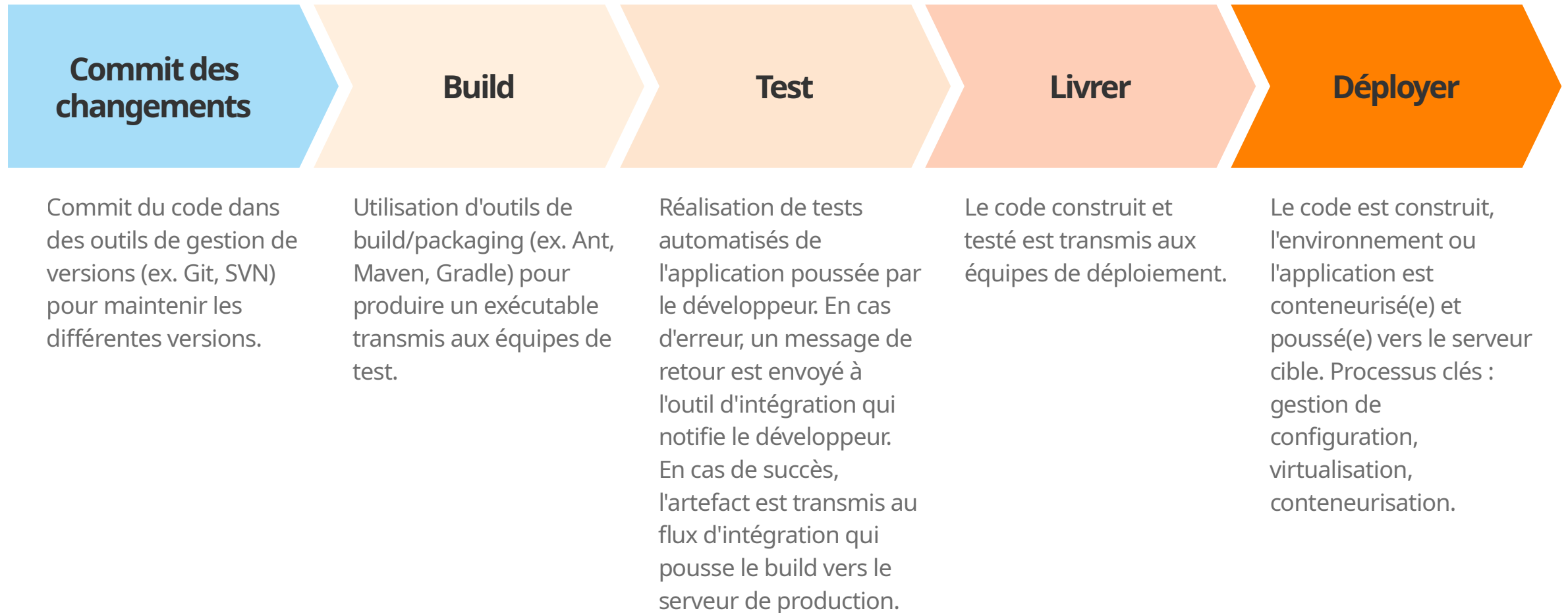
Les phases du DEVOPS



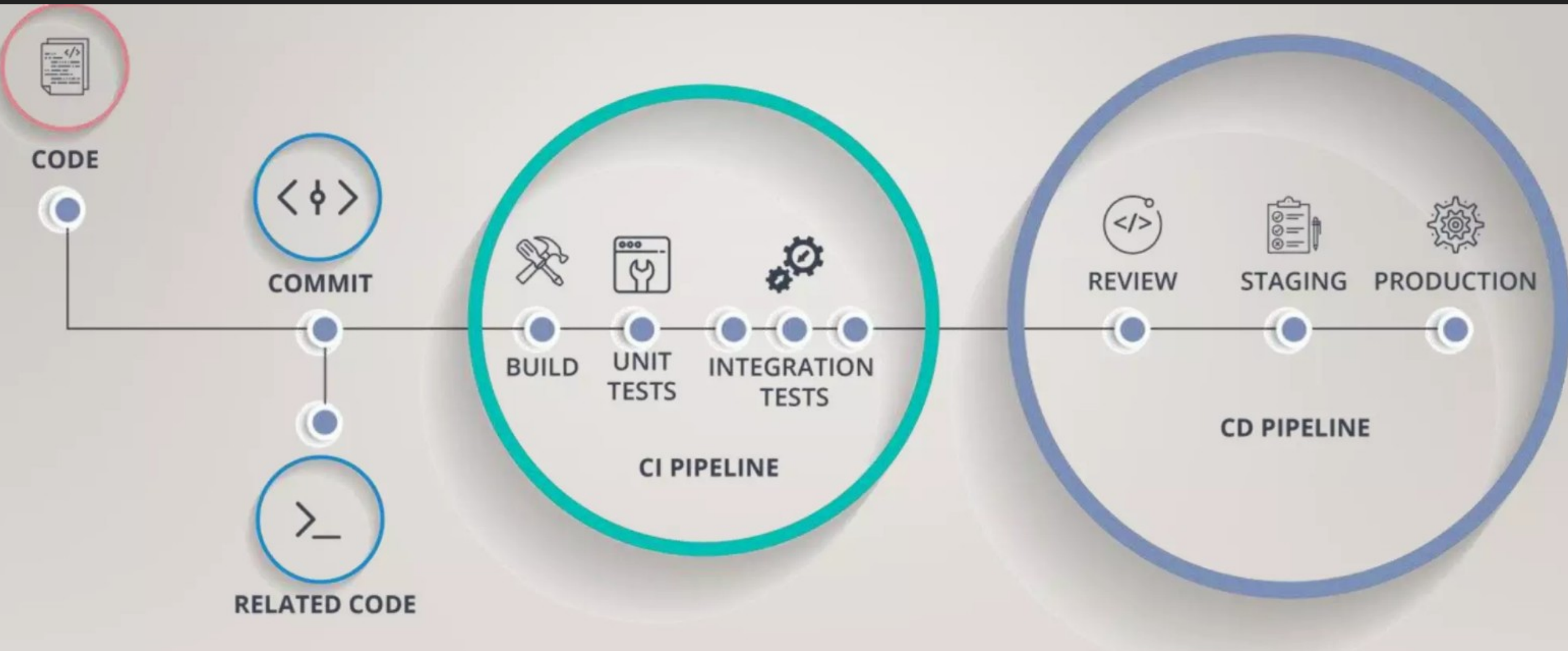
DevOps cycle de vie (Vue d'ensemble)



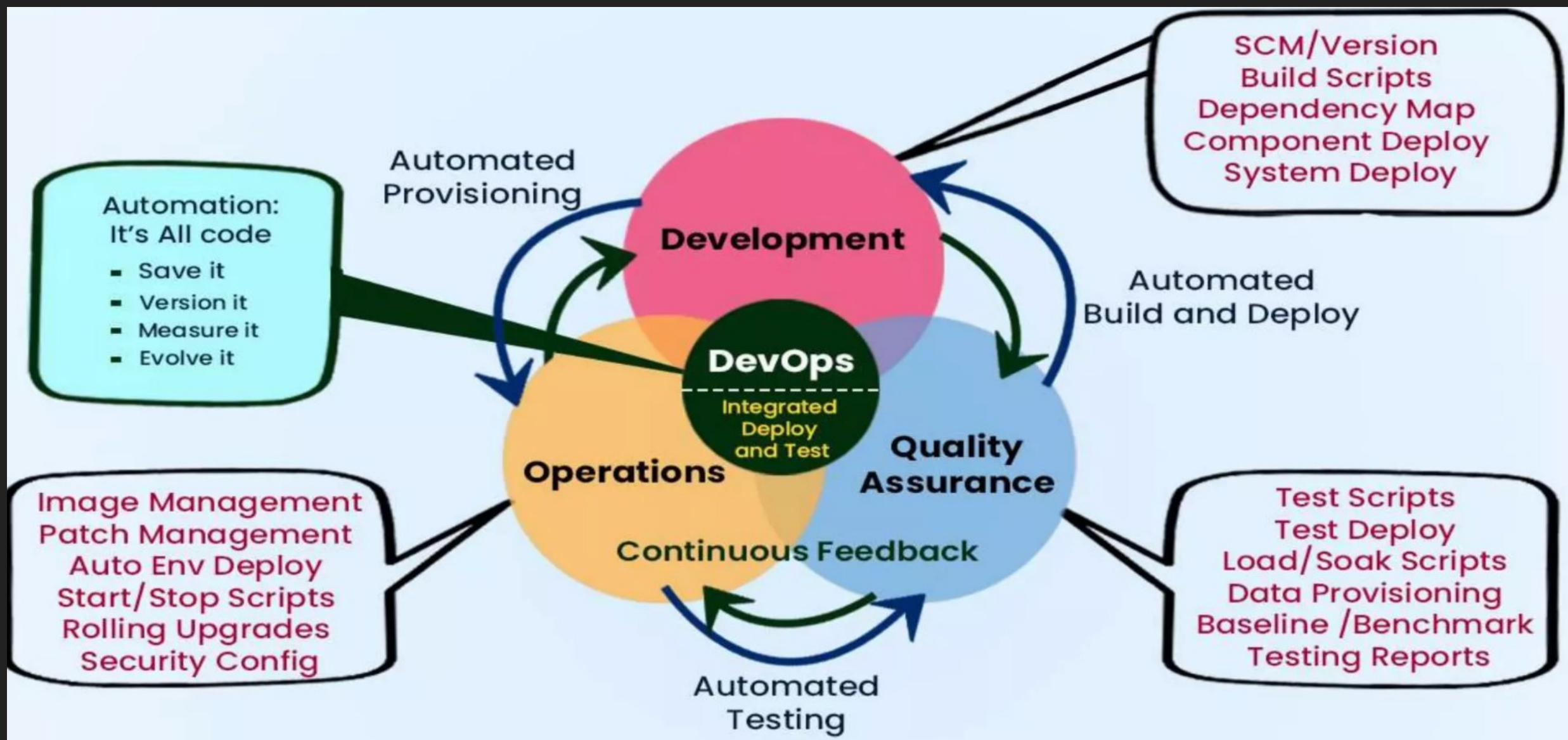
Processus CI/CD



Pipeline CI/CD automatisé



Bonnes pratiques d'automatisation



Bonnes pratiques d'automatisation

- **Gestion d'images**

Automatiser la gestion et la maintenance des images Docker pour les déploiements

- **Patch management**

Automatiser les processus de mise à jour et de correction des systèmes

- **Déploiement d'environnements**

Créer et déployer des environnements de test et de production de manière automatisée

- **Scripts start/stop**

Utiliser des scripts pour démarrer et arrêter les applications de manière standardisée

- **Mises à jour progressives**

Effectuer des mises à jour progressives des applications pour minimiser les risques

- **Sécurité/configuration**

Automatiser la gestion de la sécurité et de la configuration des systèmes

- **SCM/versions**

Utiliser un système de gestion de versions pour le code source

- **Scripts de build**

Automatiser la construction des applications avec des scripts de build

- **Cartographie des dépendances**

Identifier et gérer les dépendances entre les différents composants du système

- **Déploiement composant/système**

Automatiser le déploiement des différents composants du système

- **Scripts de test**

Créer des scripts de test automatisés pour valider le fonctionnement de l'application

- **Déploiement des tests**

Automatiser le déploiement des environnements de test

- **Tests de charge/soak**

Effectuer des tests de charge et de durabilité de manière automatisée

- **Provisionnement des données**

Automatiser la création et le provisionnement des données de test

- **Benchmarks**

Réaliser des benchmarks pour évaluer les performances de l'application

- **Rapports de test**

Générer des rapports de test automatiquement pour suivre les résultats

Bénéfices clés

- **Livraisons rapides**

Déployer des mises à jour plus fréquemment et de manière automatisée pour répondre plus rapidement aux besoins des utilisateurs.

- **Cycles plus courts**

Réduire la durée du cycle de développement grâce à l'automatisation des processus et à une meilleure collaboration entre les équipes.

- **Risques réduits**

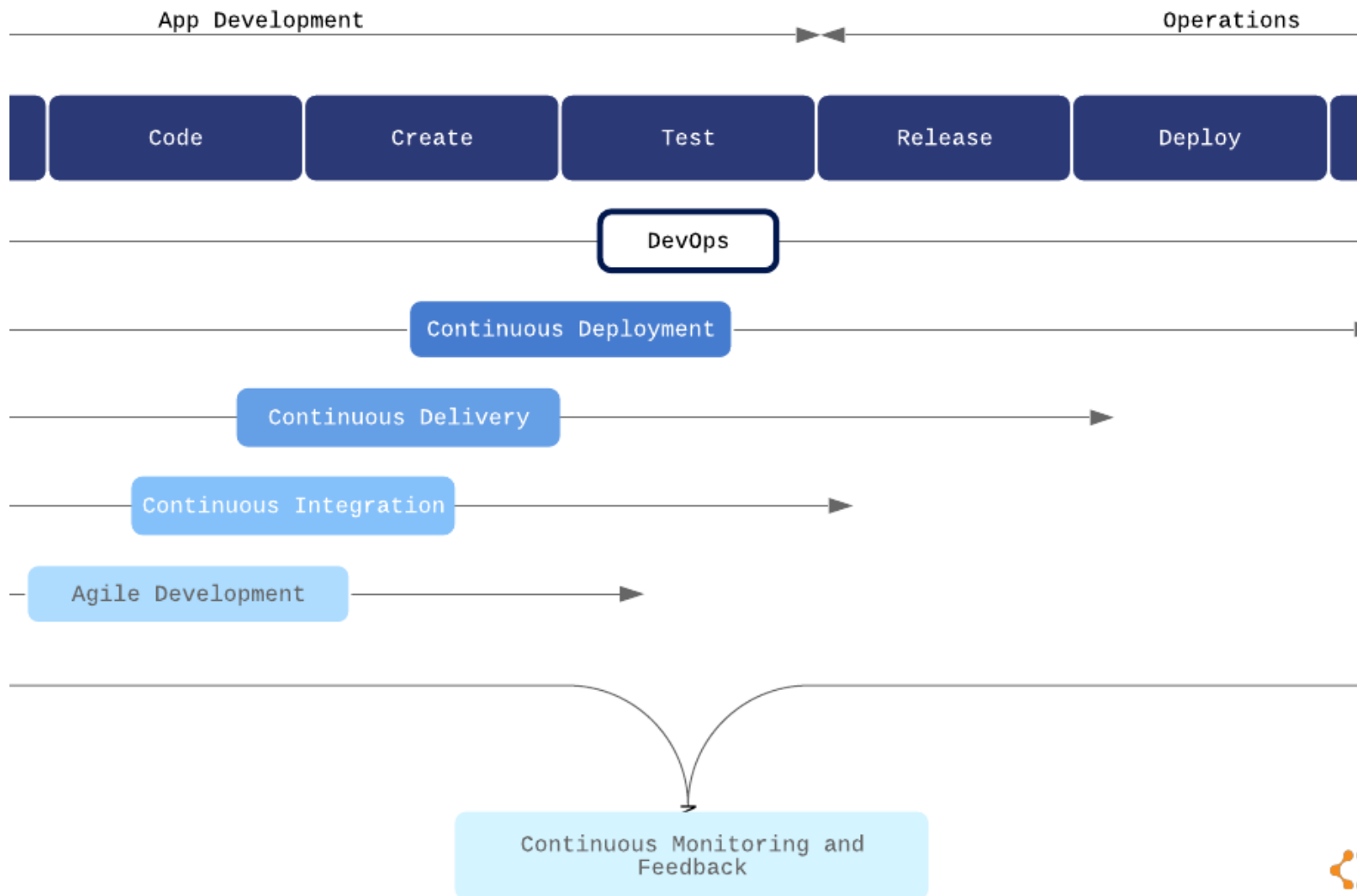
Diminuer les risques liés aux déploiements grâce à des tests automatisés et à une surveillance continue des performances de l'application.

- **Résolution plus rapide des incidents**

Identifier et résoudre les problèmes plus rapidement grâce à une meilleure visibilité sur l'état de l'application et à des processus de débogage automatisés.

- **Meilleure productivité**

Optimiser les processus et éliminer les tâches manuelles répétitives pour permettre aux équipes de se concentrer sur des tâches à plus forte valeur ajoutée.



**DevOps est devenu
une exigence
essentielle pour les
entreprises
souhaitant rester
compétitives et prospérer
dans l'industrie du logiciel.
Ce n'est plus un facteur de
différenciation, mais
un élément
indispensable pour
assurer une livraison
rapide et fiable des
applications logicielles.**



“aha moments”!

Merci & à la prochaine séance