

Initiation en DevOps

Mohamed Taher BEN BAHRI



Plan

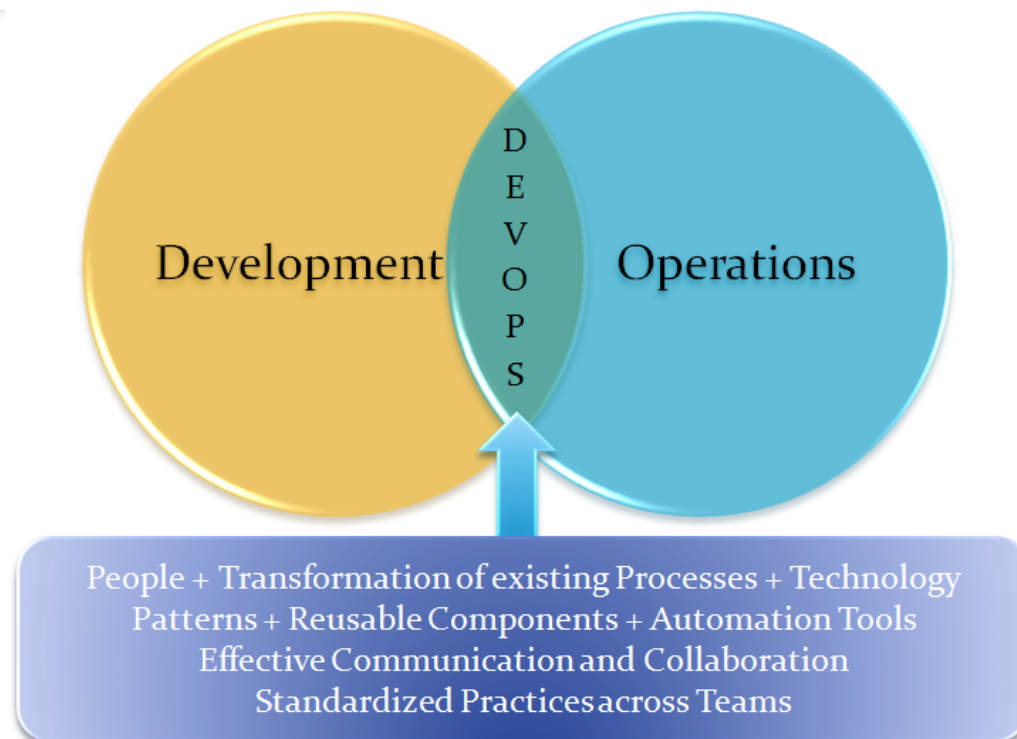
- DevOps
- Automatisation des constructions: Maven
- Versionnement du code: Git
- Integration Continue: Gitlab CI
- Environnement de déploiement: Docker



C'est Quoi DevOps aujourd'hui ?

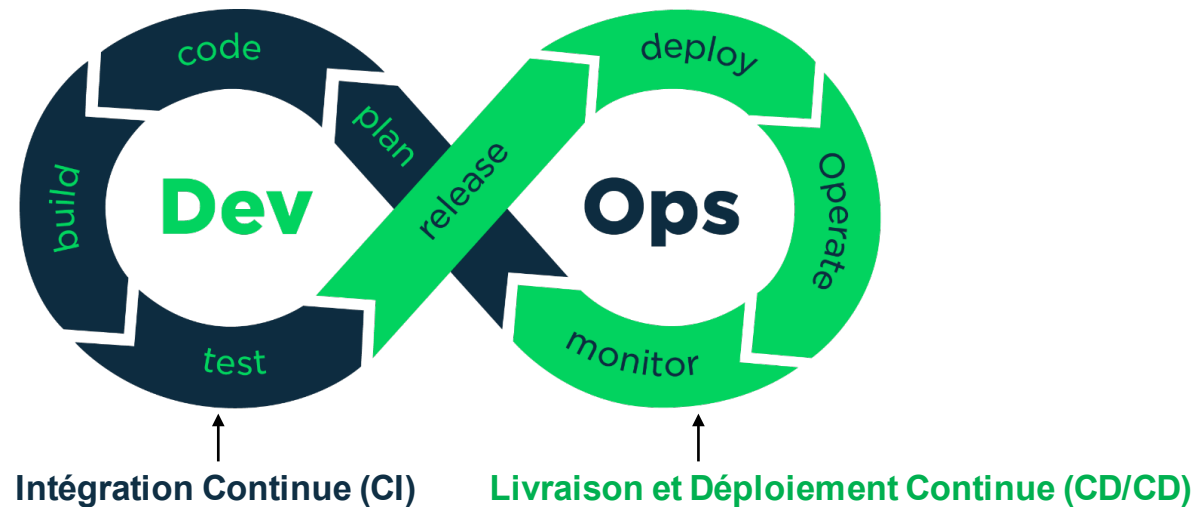
- DevOps est tout au sujet de la culture d'une organisation, des processus et de la technologie
- Développer la communication et la collaboration entre les équipes de développement et d'opérations informatiques afin de gérer le cycle de vie des applications plus efficacement.
- Nous avons souvent tendance à travailler sur la base de modèles pour trouver des solutions réutilisables à des types de problèmes ou de défis similaires.

C'est Quoi DevOps aujourd'hui ?

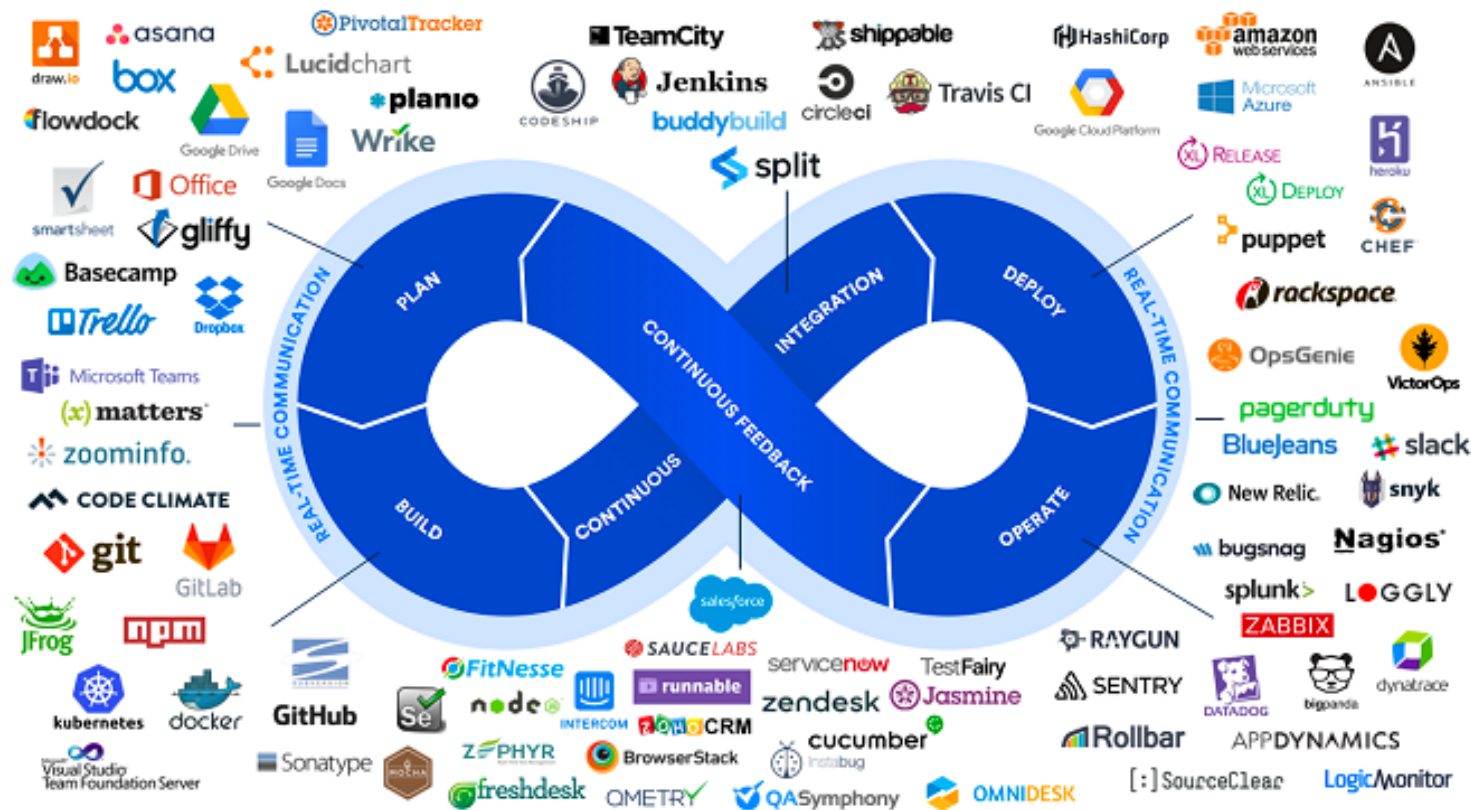


Culture DevOps: Mindset

- Étend et complète le processus d'intégration et de diffusion continue en s'assurant que le code est prêt à la production (CI)
- DevOps permet un flux de travail beaucoup plus continu dans les opérations IT (CD)



Ecosystème DevOps





Industrialisation des déploiements : Mise en place d'une usine logicielle DevOps

- Outils nécessaires
 - Gestionnaire de code sources
 - Construction logicielle (Build)
 - Serveur d'Intégration Continue (compilation, test, génération d'artefacts)

Standardisation des Livrables

- Des outils qui permettent de standardiser les livrables en fonctions des technologies

- Maven
- Gradle
- Npm
- Composer

Maven™



Standardisation des Livrables

- Construction logicielle
 - La compilation du code source.
 - La résolution des dépendances.
 - Le packaging des fichiers compilés dans un format compressé tel que jar, zip, war, etc.
 - La production des installeurs.
 - Etc...

Gestionnaires du Code Source & Version Control



- C'est très nécessaire lorsque plusieurs développeurs travaillent/modifient en permanence le code source.

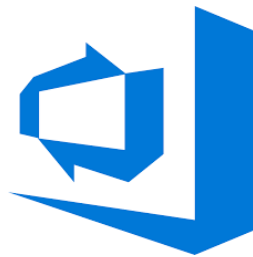
Gestionnaires du Code Source & Version Control

- **Git**



- **SVN**

- **Mercurial**



- **VSTS**

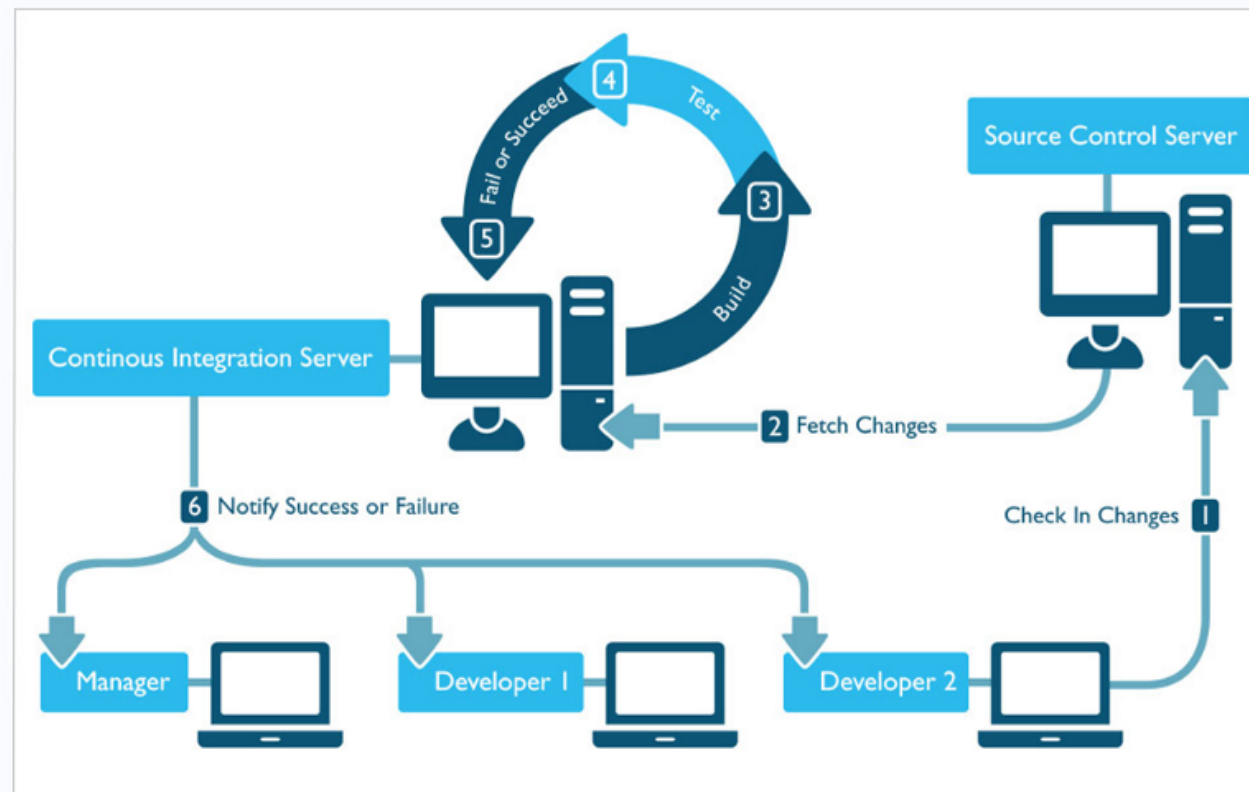


Gestionnaires du Code Source & Version Control

- Gestionnaire de code sources git
 - Git s'impose comme un quasi standard dans la gestion de code sources
 - Un système distribué
 - Un système rapide
 - Travail collaboratif avec github, bitbucket ou gitlab



Continuous Integration



Qu'est-ce que l'intégration continue?

- Fusionner (Merge) toutes les copies de travail des développeurs dans un environnement partagés.
- Améliorer la qualité du code.
- Isoler les changements qui sont effectués et rapporte immédiatement chaque fois qu'ils sont ajoutés à une base de code.
- Rétroaction rapide lorsqu'un défaut est détecté dans la base de code
- Testing et Reporting automatique.
- un pipeline: Le plan, le code, la construction et le test

GitLab CI/CD



- [GitLab](#) CI/CD est un outil intégré que chaque utilisateur de GitLab peut utiliser.
- Il vous permet d'héberger plusieurs GitLab runners sur des serveurs et de leur attribuer des étiquettes.
- Cela vous donne une batterie de serveurs de build où les builds peuvent être alloués à n'importe quel serveur selon les besoins.
- Cela permet des opportunités d'évolutivité massives, que des outils comme Jenkins ne fournissent pas.

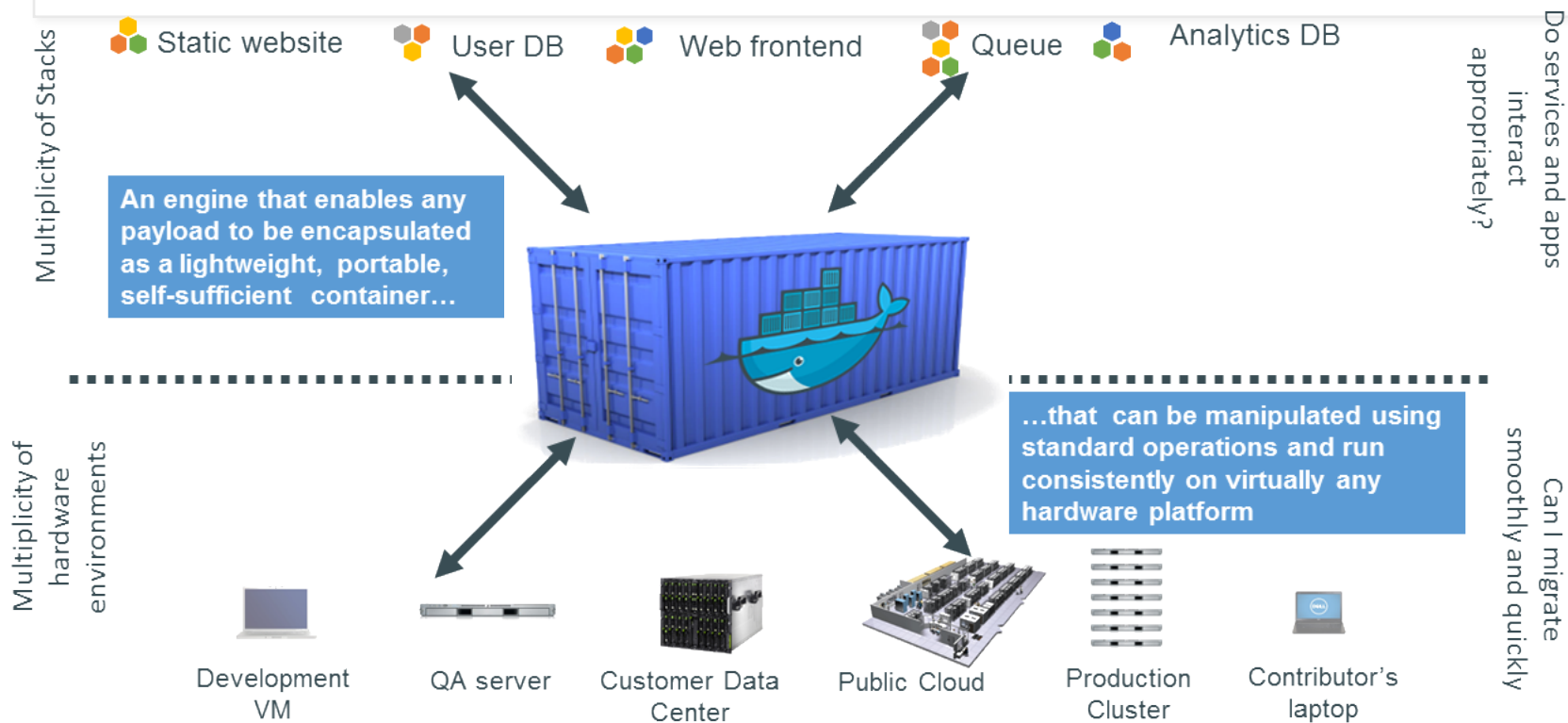
GitLab CI/CD



GitLab CI / CD est votre meilleur pari si:

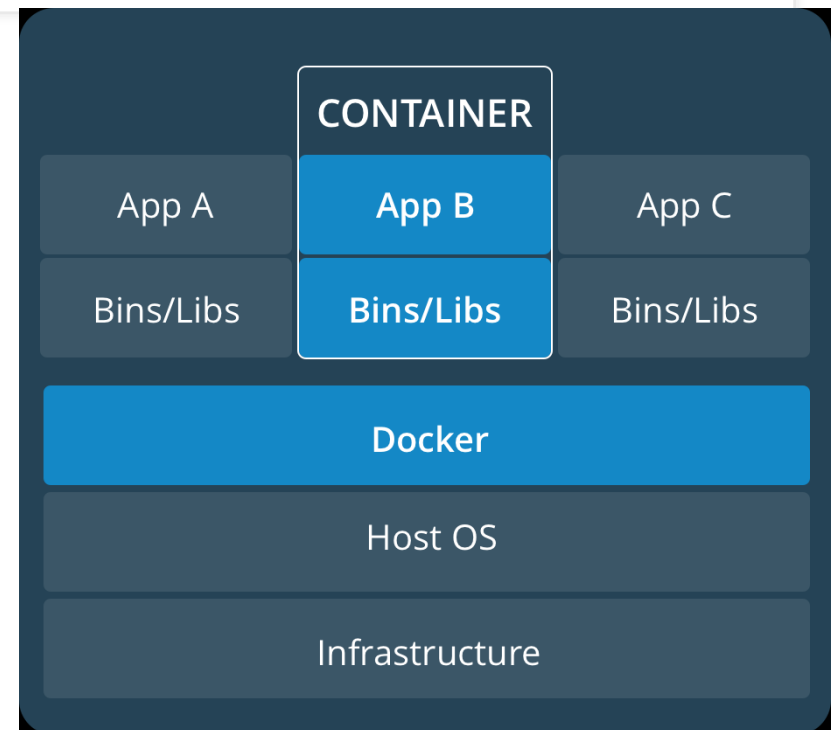
- votre code est hébergé dans GitLab;
- vous souhaitez externaliser votre cycle de vie DevOps;
- vous ne voulez pas avoir à installer et configurer un outil entièrement nouveau;
- vous bénéficierez d'un taux constant de publication de fonctionnalités;
- vous avez besoin d'un registre Docker intégré;
- vous n'avez pas besoin de plugins.

La virtualisation des environnements: Les conteneurs

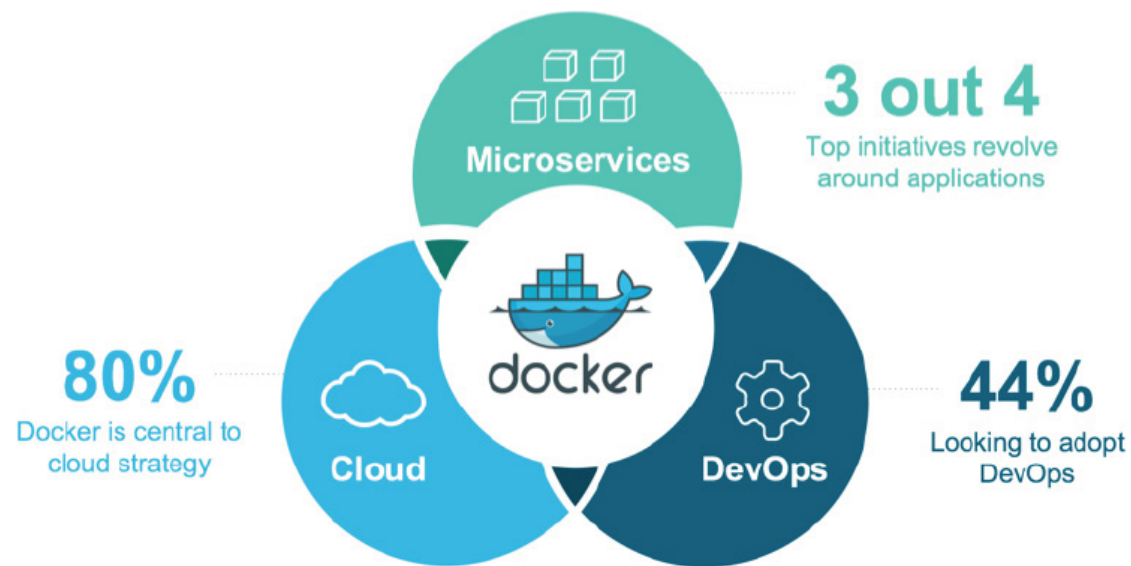


La virtualisation des environnements : Les conteneurs

- Containerisation
 - Code + dépendances
 - Isolation des applications
 - Autonomie pour les DEVs
 - Déploiement simple et rapide
 - Modularité et portabilité
 - Plus légers qu'une VM
 - Même kernel que l'O/S hôte



La virtualisation des environnements : Les conteneurs



State of App Development Survey: Q1 2016

https://www.innovate-systems.de/downloads/docker/The_definitive_Guide_to_Docker.pdf

La virtualisation des environnements:

Les conteneurs – Transition Monolithique vers Microservices

1

- Conteneuriser les applications monolithiques
- Garantir la portabilité, l'optimisation des ressources et la sécurité



2

- Conteneuriser les applications monolithiques puis les transformer en microservices (Refactoring)
- Transformation selon l'importance de l'application

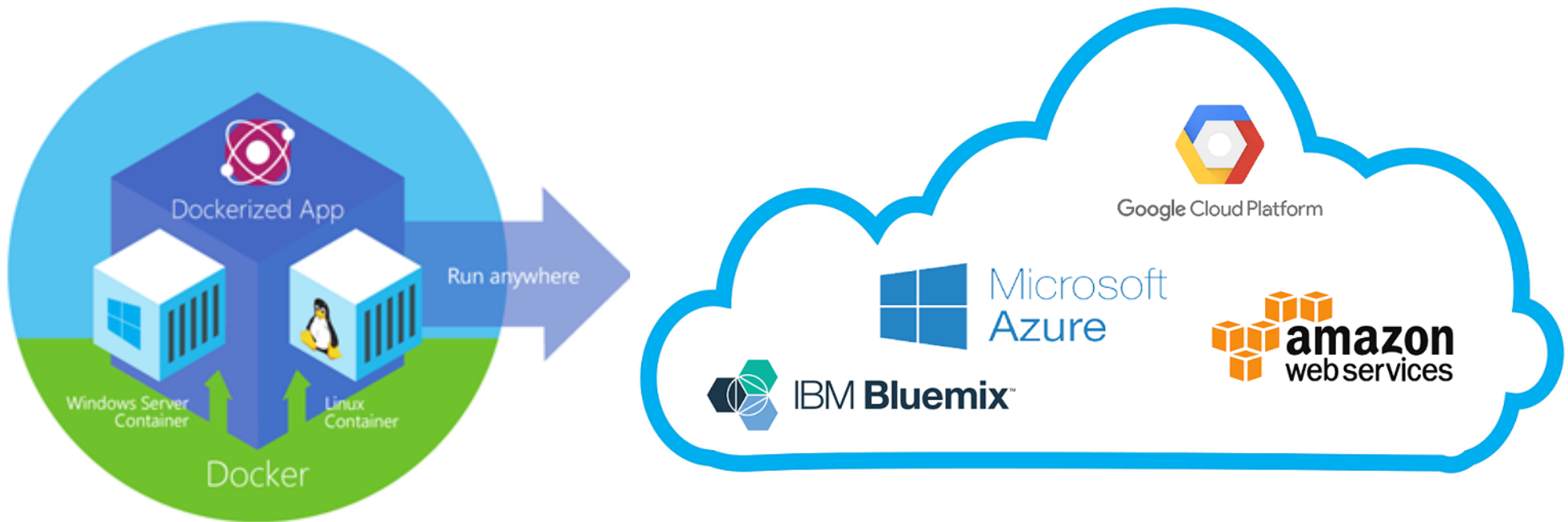


3

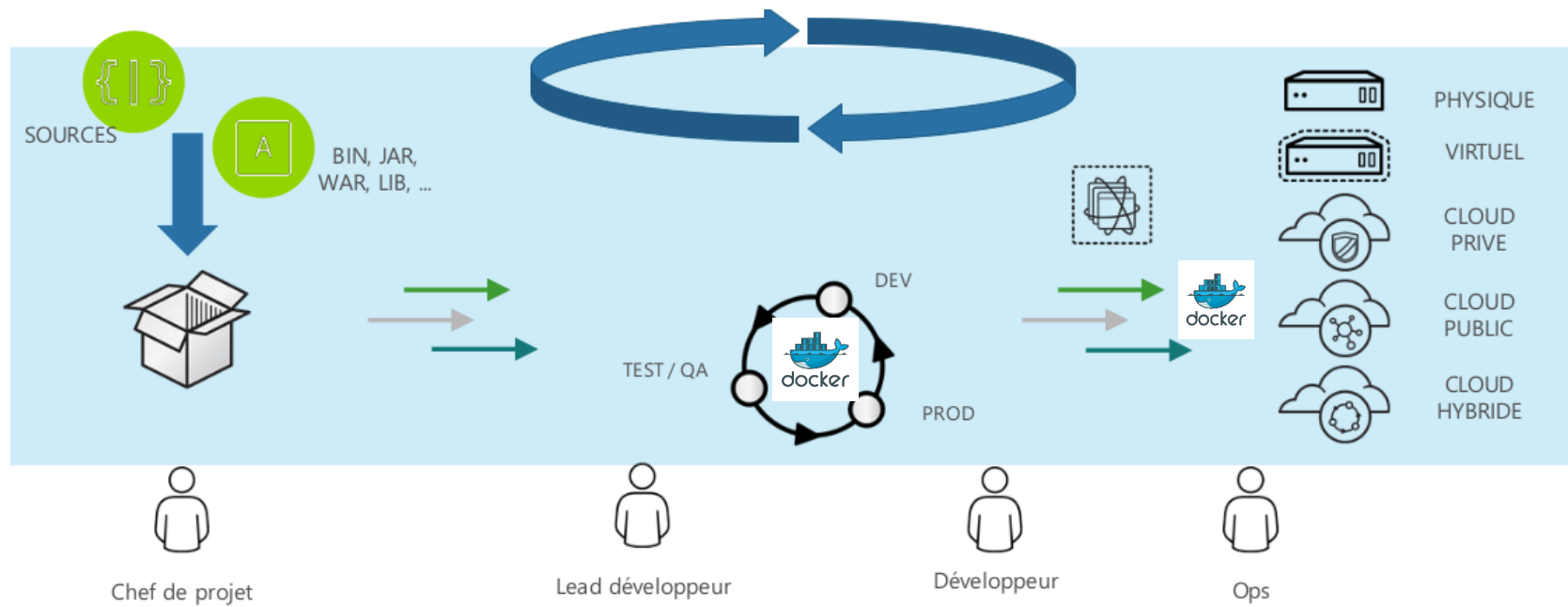
- Créer des nouvelles applications microservices (from scratch)
- Conteneuriser tous les microservices distribués et les déployer en une seule application en production.



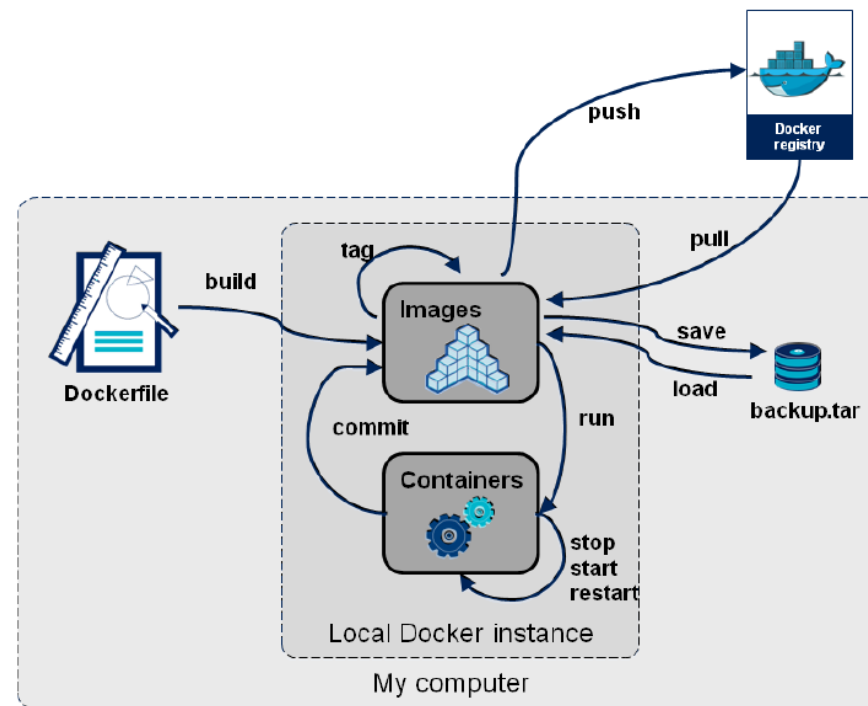
La virtualisation des environnements: Les conteneurs révolutionnent le Cloud



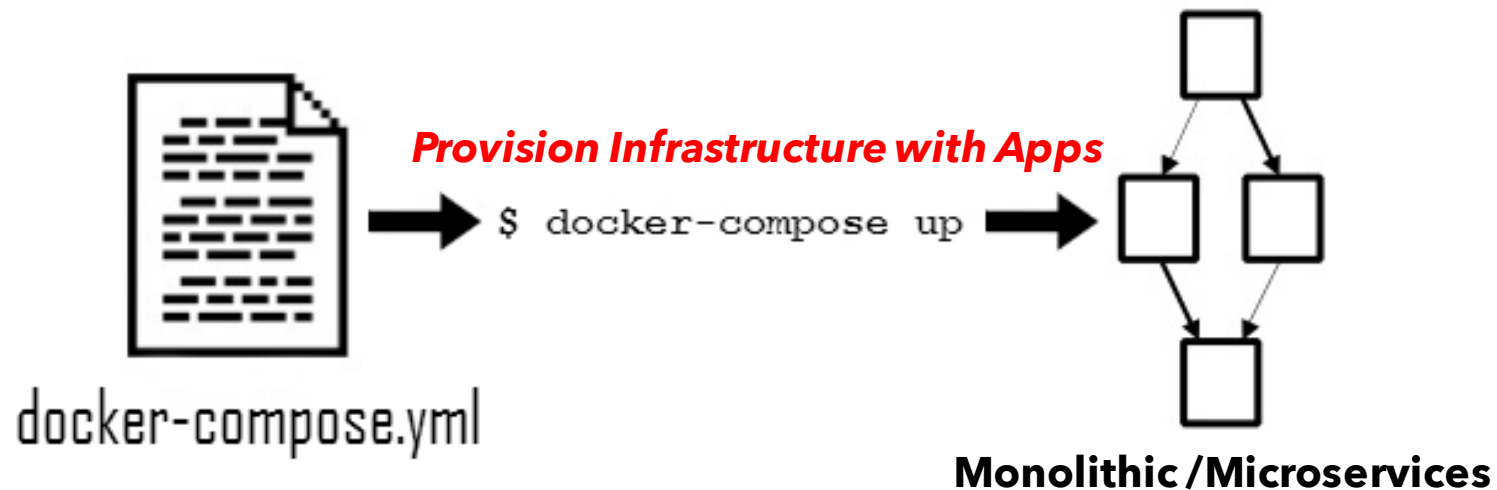
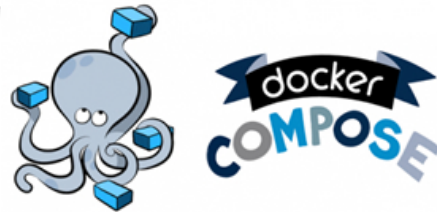
La virtualisation des environnements : Les conteneurs – La voix vers le DevOps



La virtualisation des environnements : Les conteneurs - Cycle de vie



La virtualisation des environnements: Les conteneurs – Docker Compose



Questions

