M.Mbengue

Introduction

DevOps _ Intégration Continue

INTEGRATION – CONTINUE

Partie I

M.Mbengue

PLAN PARTIE I

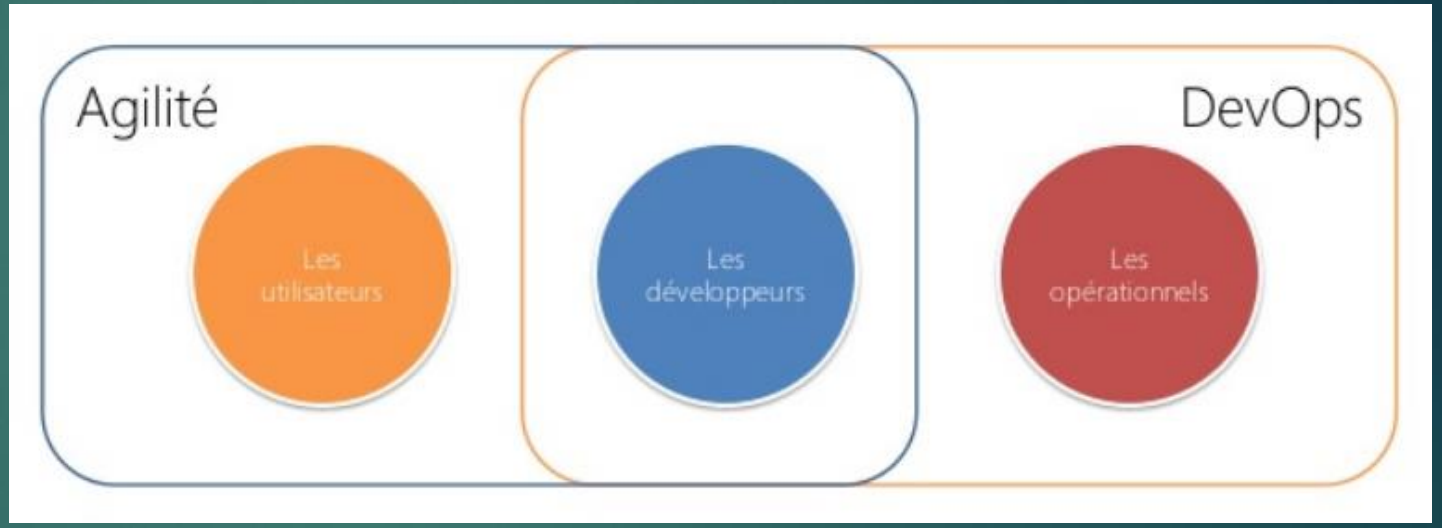
INTRODUCTION

I. Gestion des Versions
II. Gestion des Builds et Dépôts
III. Gestion des Tests

INTRODUCTION

Introduction

- AGILITE ET DEVOPS



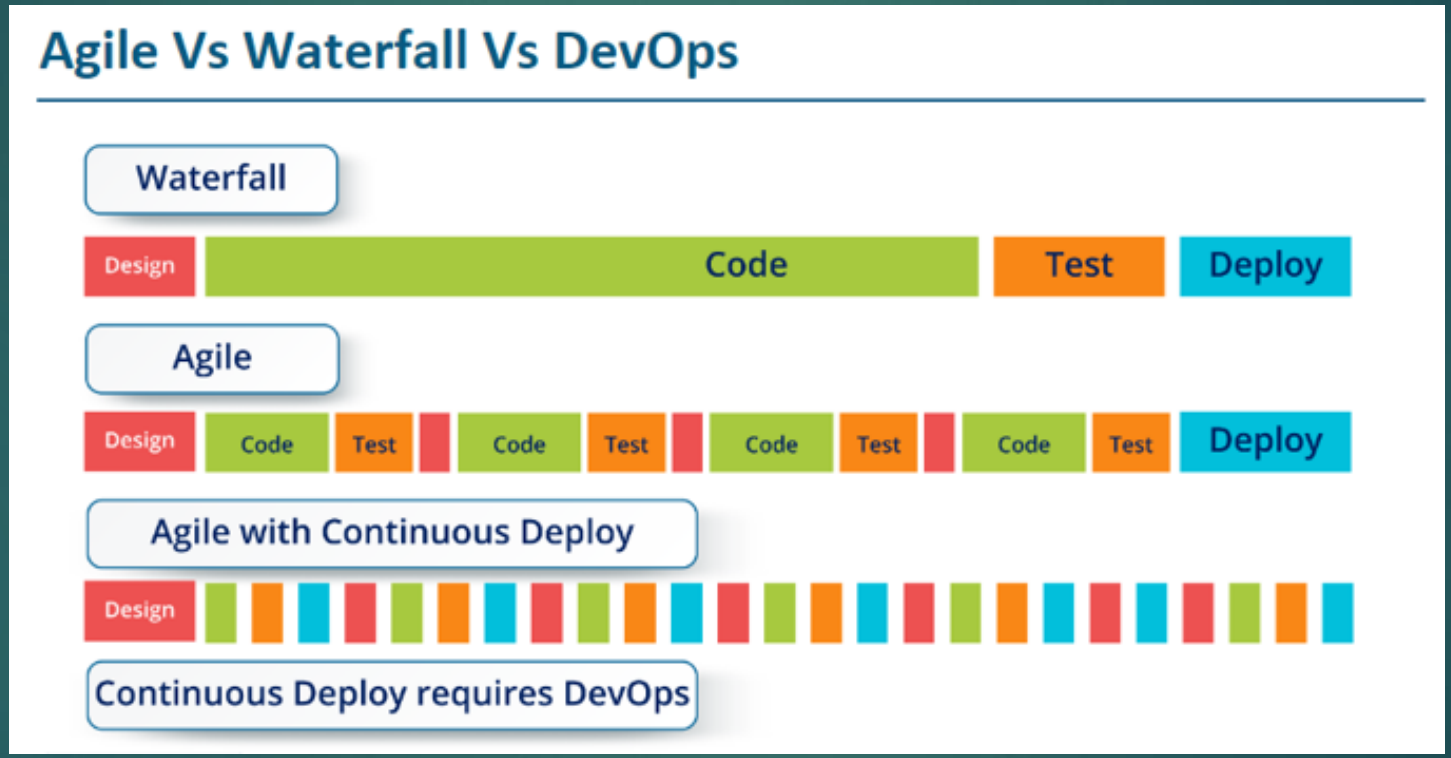
Introduction

- Automatisation
- Agilité
- Culture
- Outils

Introduction

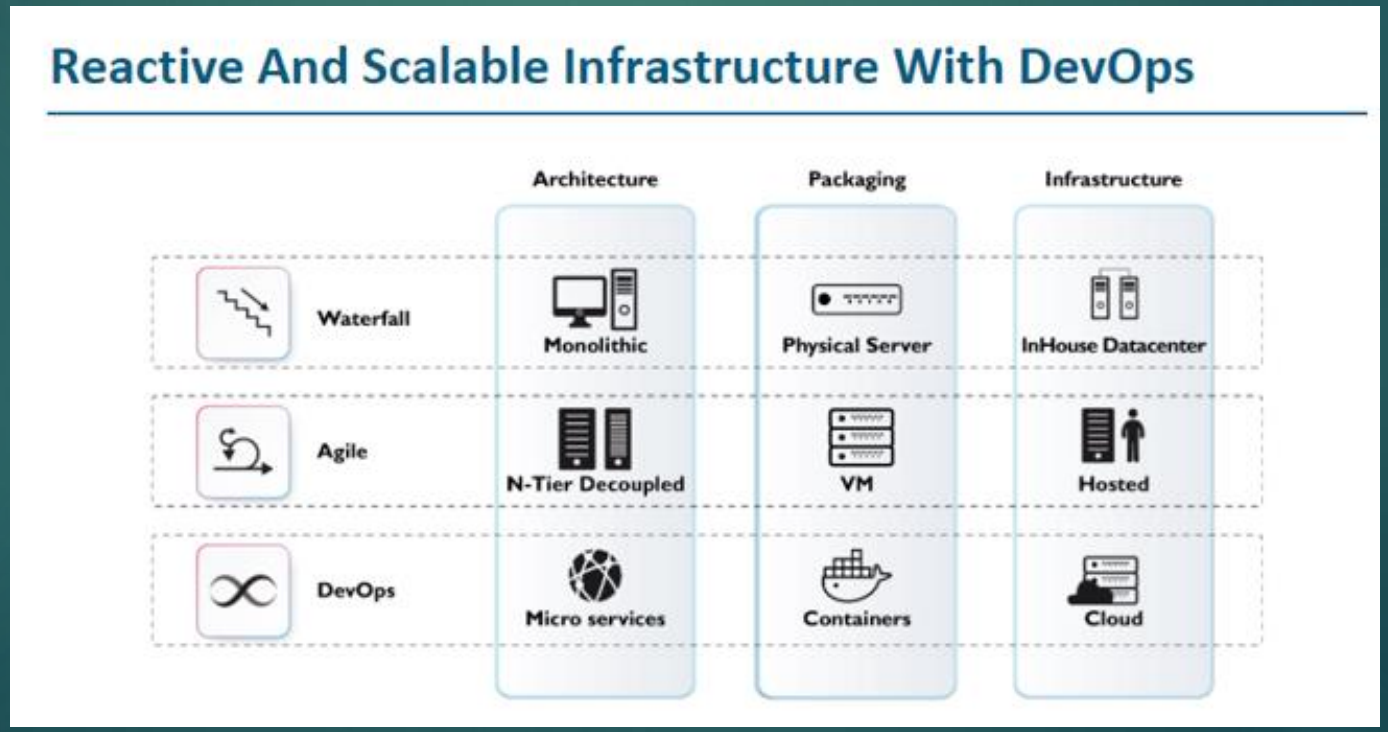
- Développeurs et Intégrateurs collabore pour une meilleure productivité dans :
  - Automatisation des infrastructures
  - Automatisation des workflows
  - Mesure de la performance continue

Introduction



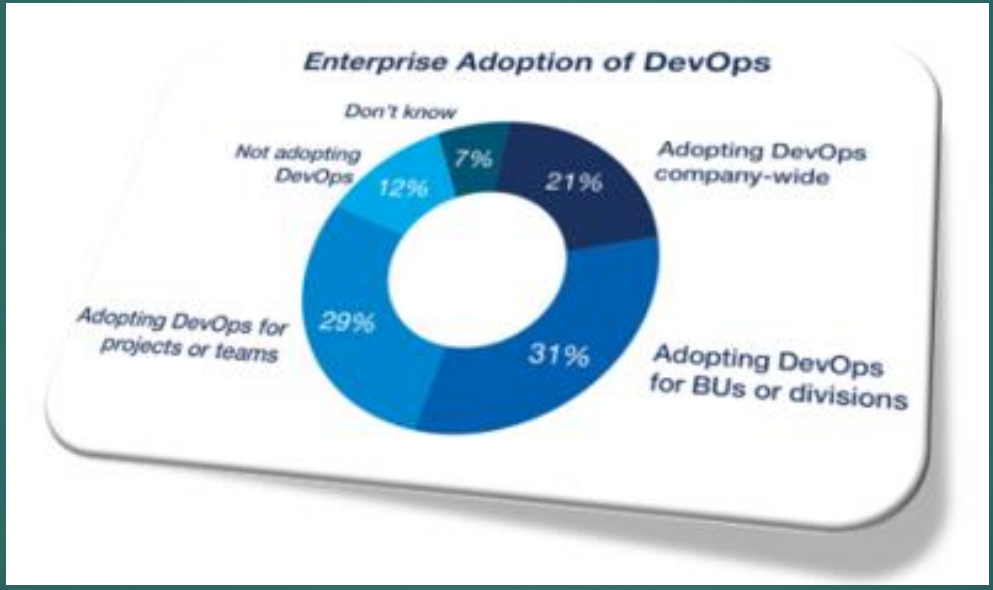
Introduction

- L'architecture, les livrables et les infrastructures d'hier et d'aujourd'hui



Introduction

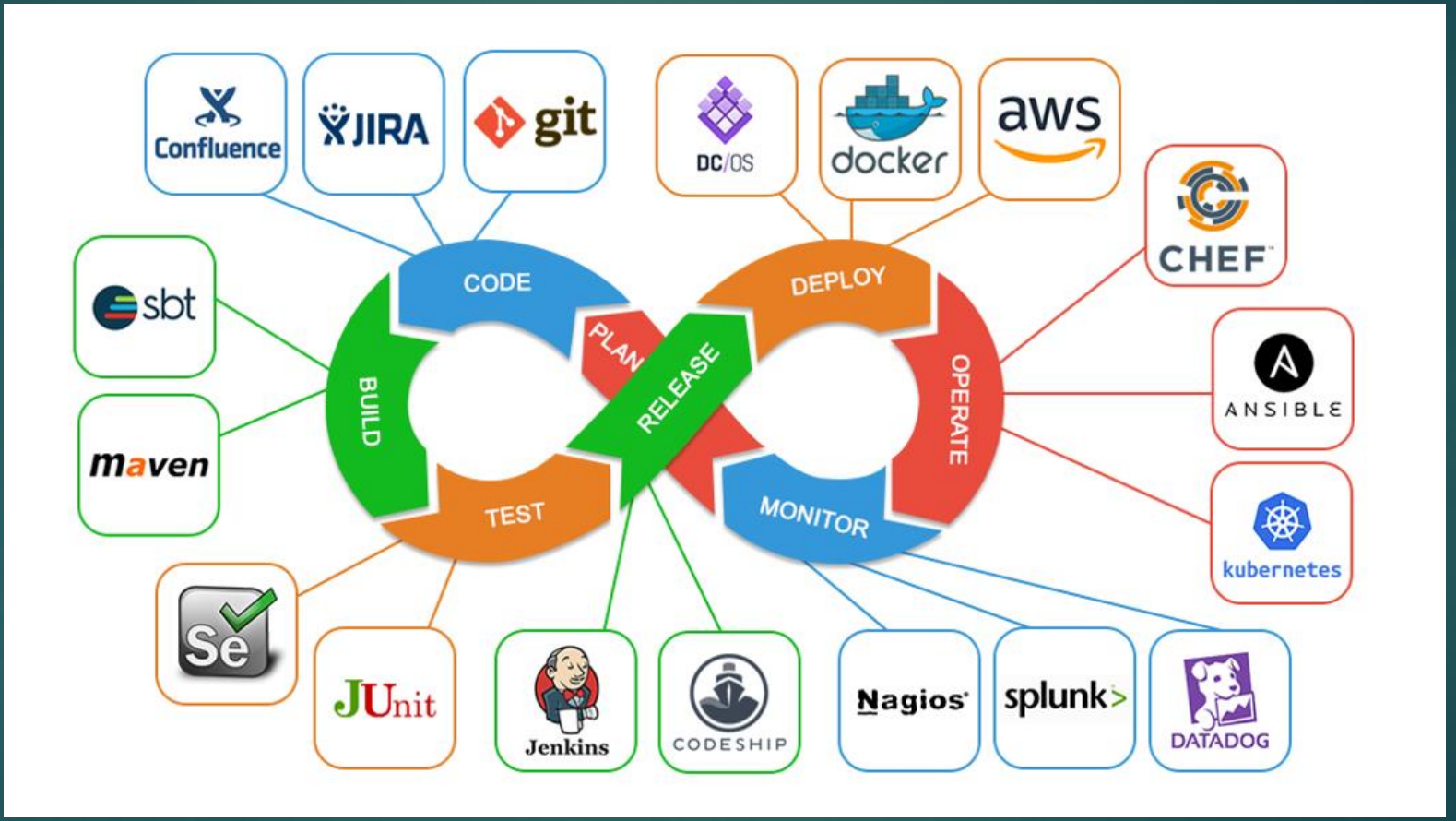
- Adoption



Introduction

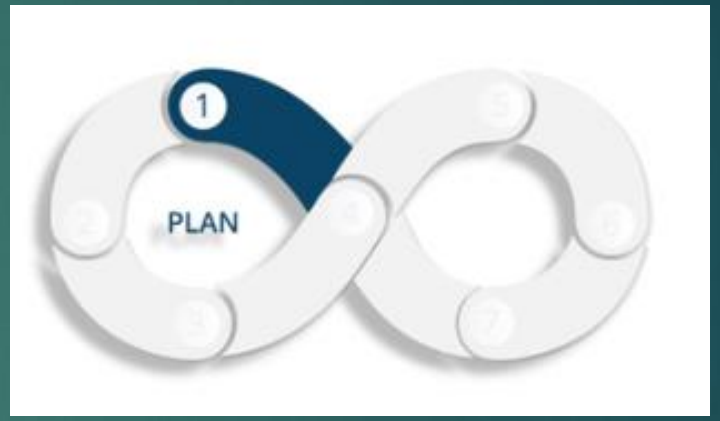
Introduction

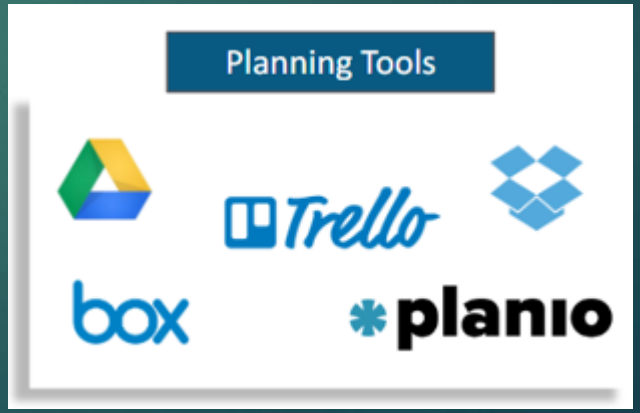
CYCLE DE VIE



Introduction

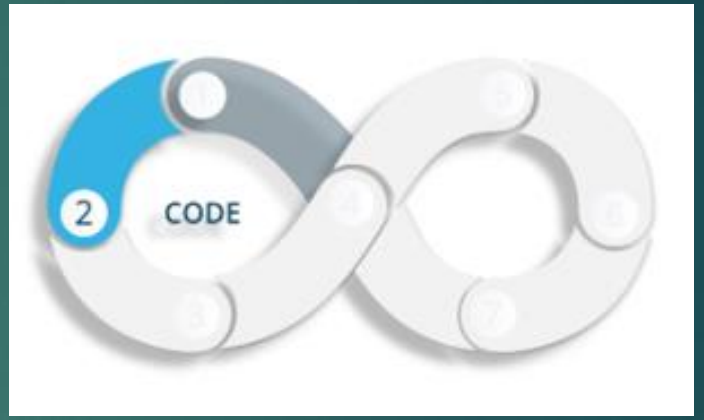
- La première phase est :
  - La planification
  - Visualisation
  - Le suivi





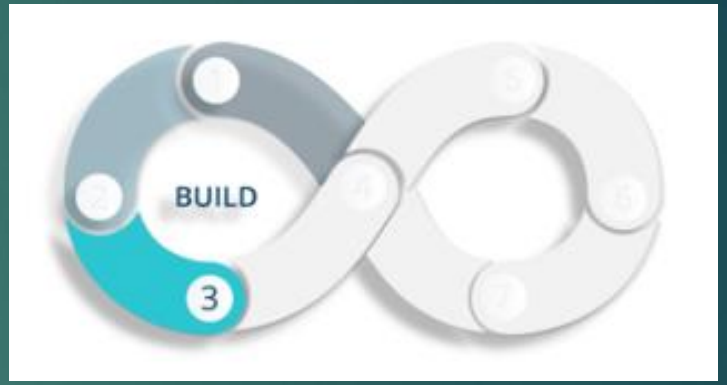
Introduction

- La seconde phase est :
  - L'implémentation



Introduction

- La troisième phase est :
  - Le build qui est une étape de pré-release (différend de l'étape de release)



Introduction

- La quatrième phase est :
  - Exécution automatique des tests dans le but d'avoir un feedback concernant la qualité du produit en cours de déploiement

Introduction

- La cinquième phase est :
  - Intégration du code au sein d'un repository partagé dans lequel l'identification des anomalies peut se faire aisément.

Introduction

- La sixième phase est :
  - La gestion et le provisionning des environnement de déploiement des applications

Introduction

- La septième phase est :
  - S'assurer d'un système à jour et et contenant les dernières évolutions de l'applicatif.

Introduction

- La huitième phase est :
  - Le monitoring

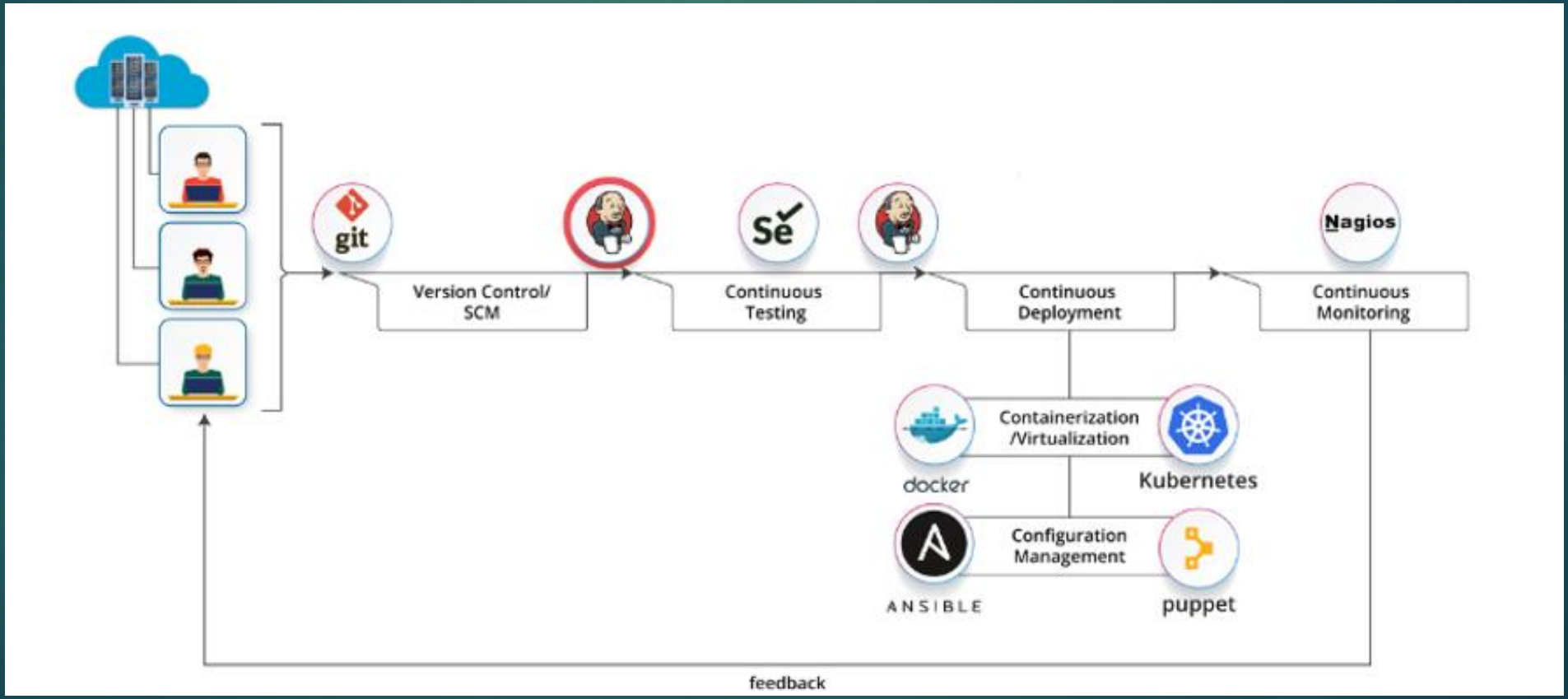
Introduction

M.Mbengue

Continuous Integration

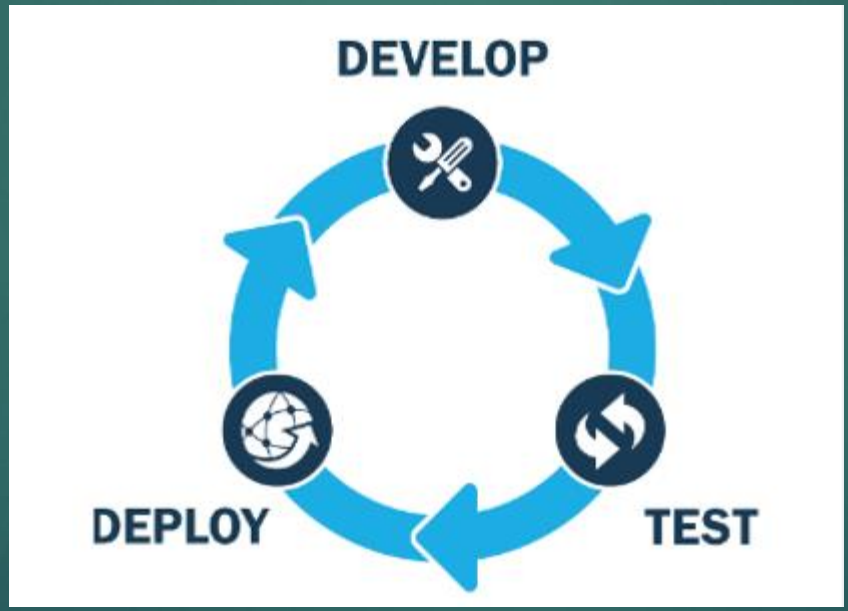
Introduction

- Workflow de l'intégration continue



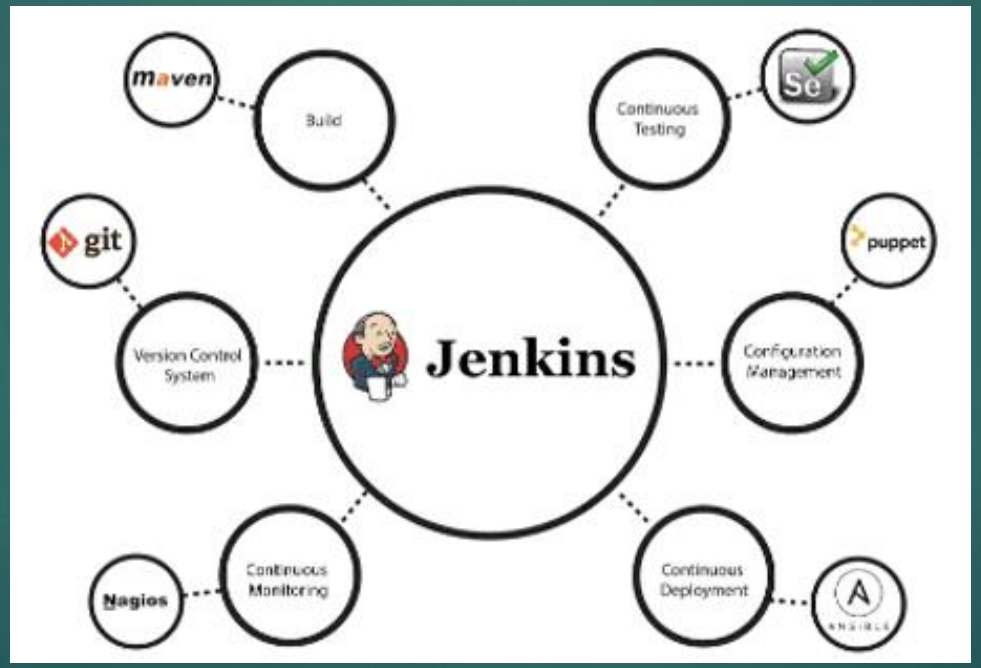
Introduction

- Workflow de l'intégration continue



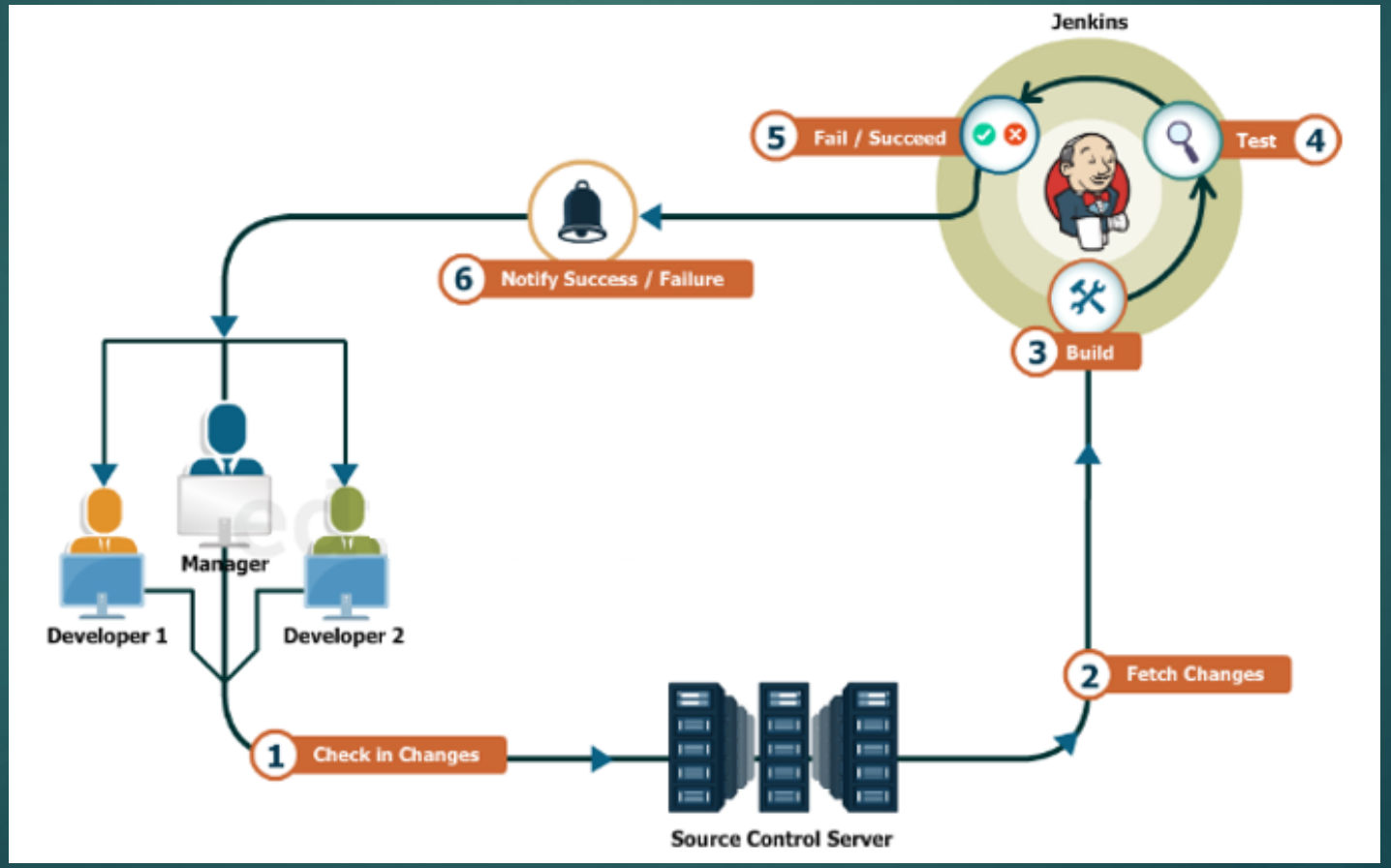
Introduction

- Intégration continue avec Jenkins



Introduction

- Comment ?

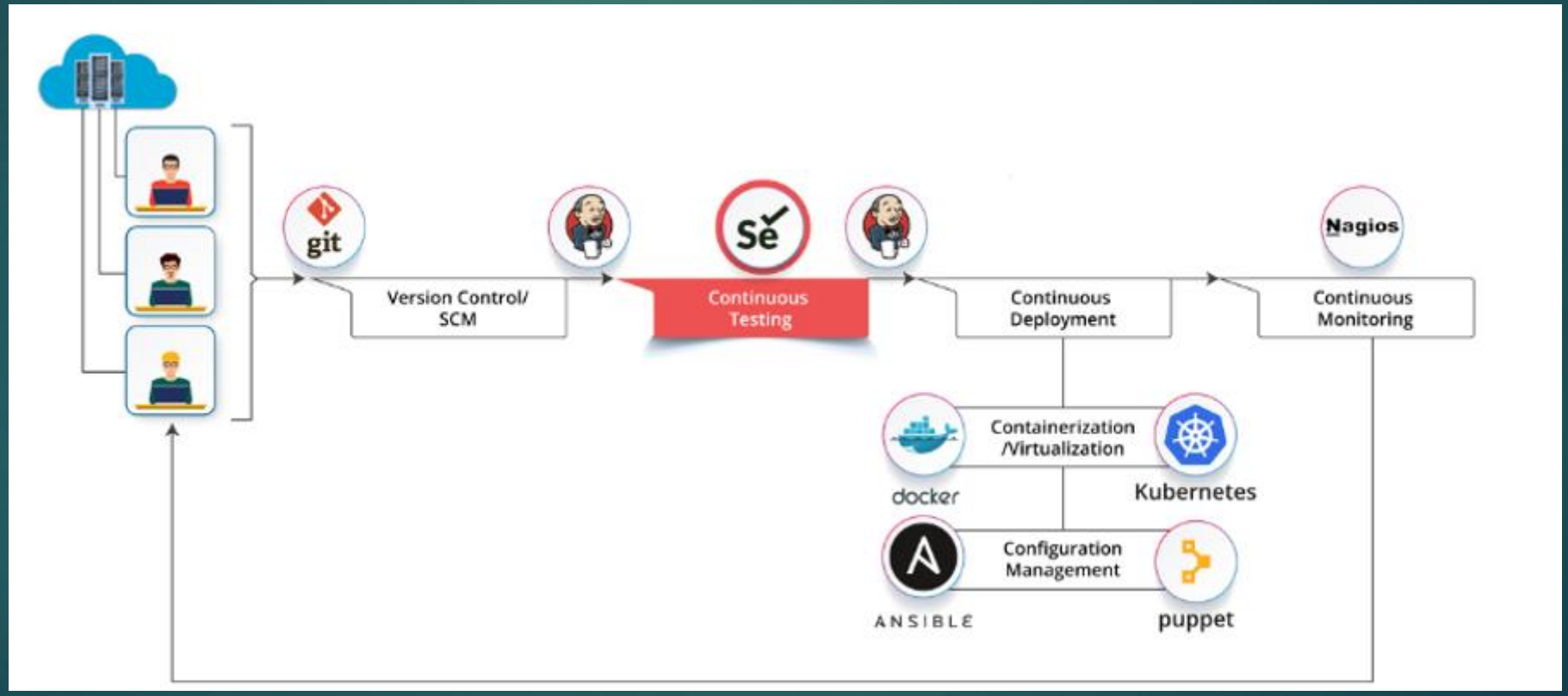


Introduction

Continuous Testing

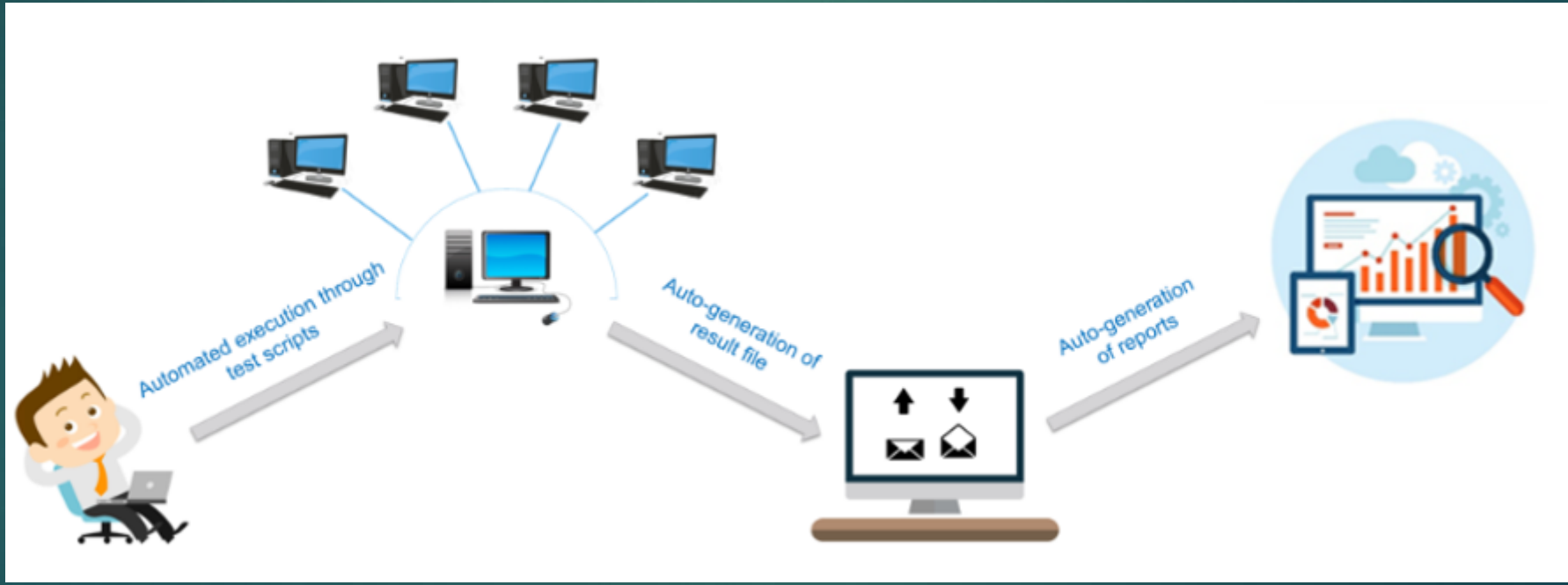
Introduction

- Testing continue avec Selenium



Introduction

- Comment ?

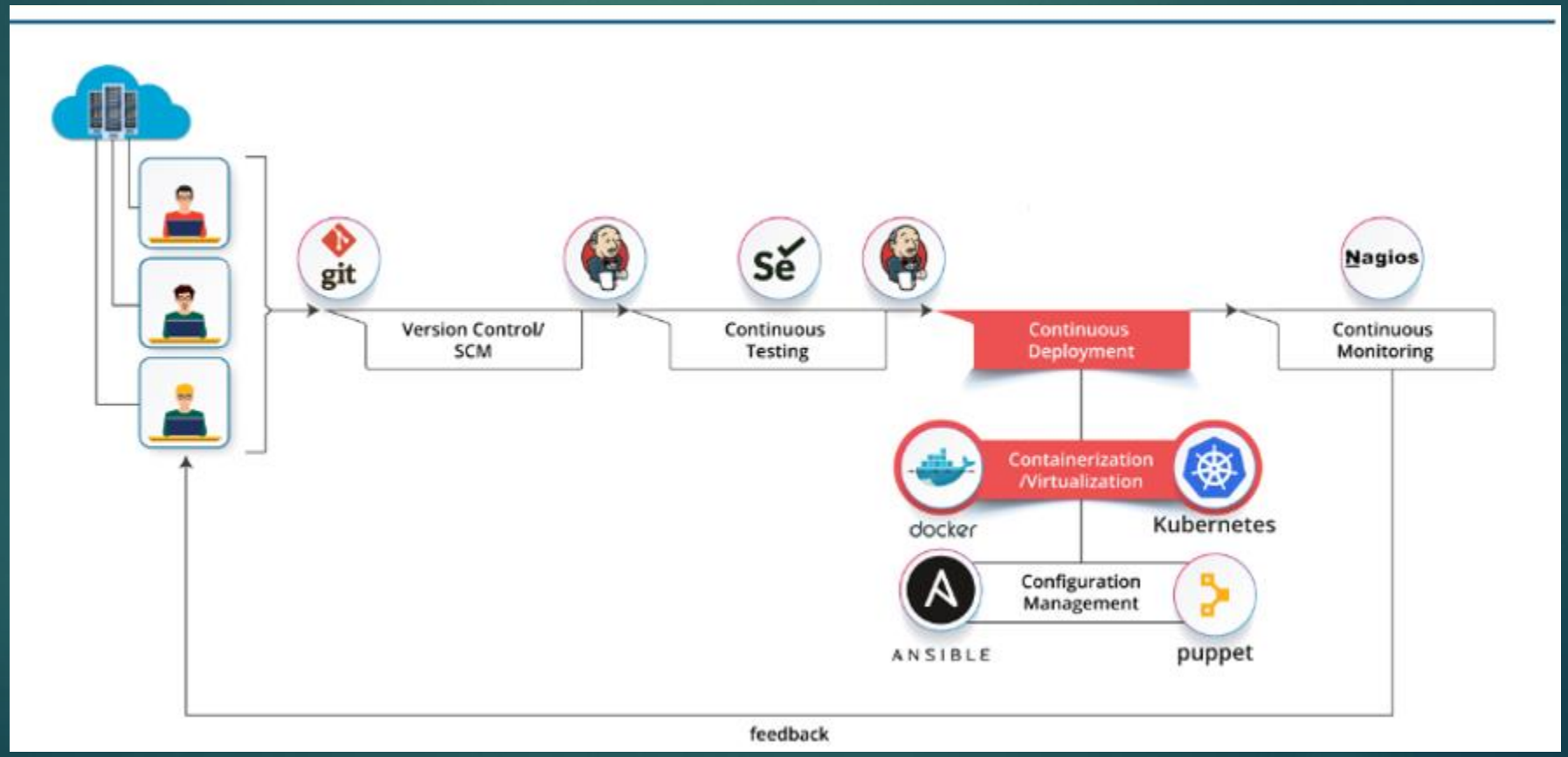


Introduction

Continuous Deployment

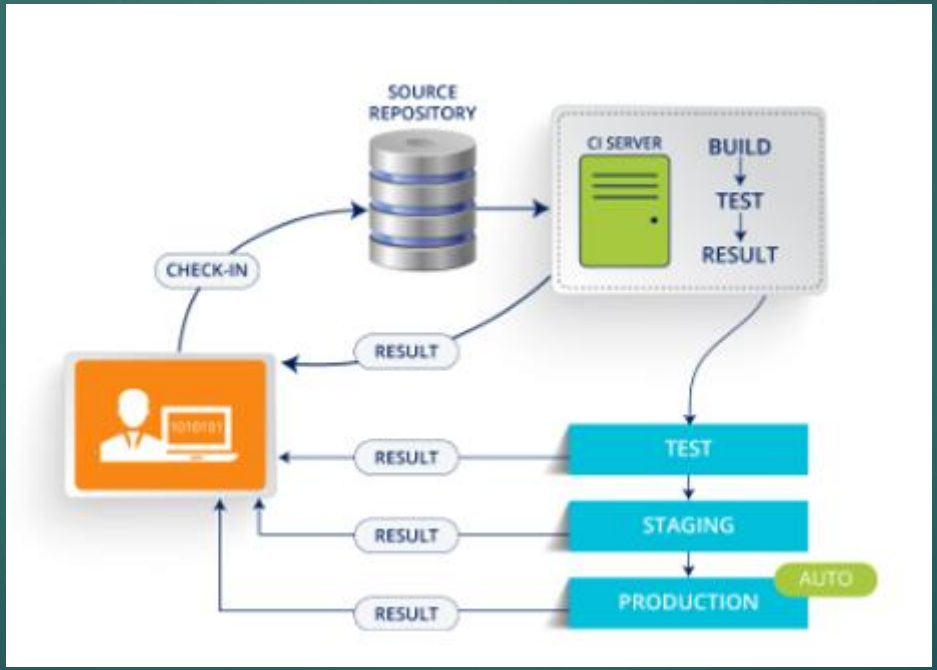
Introduction

- Déploiement continue



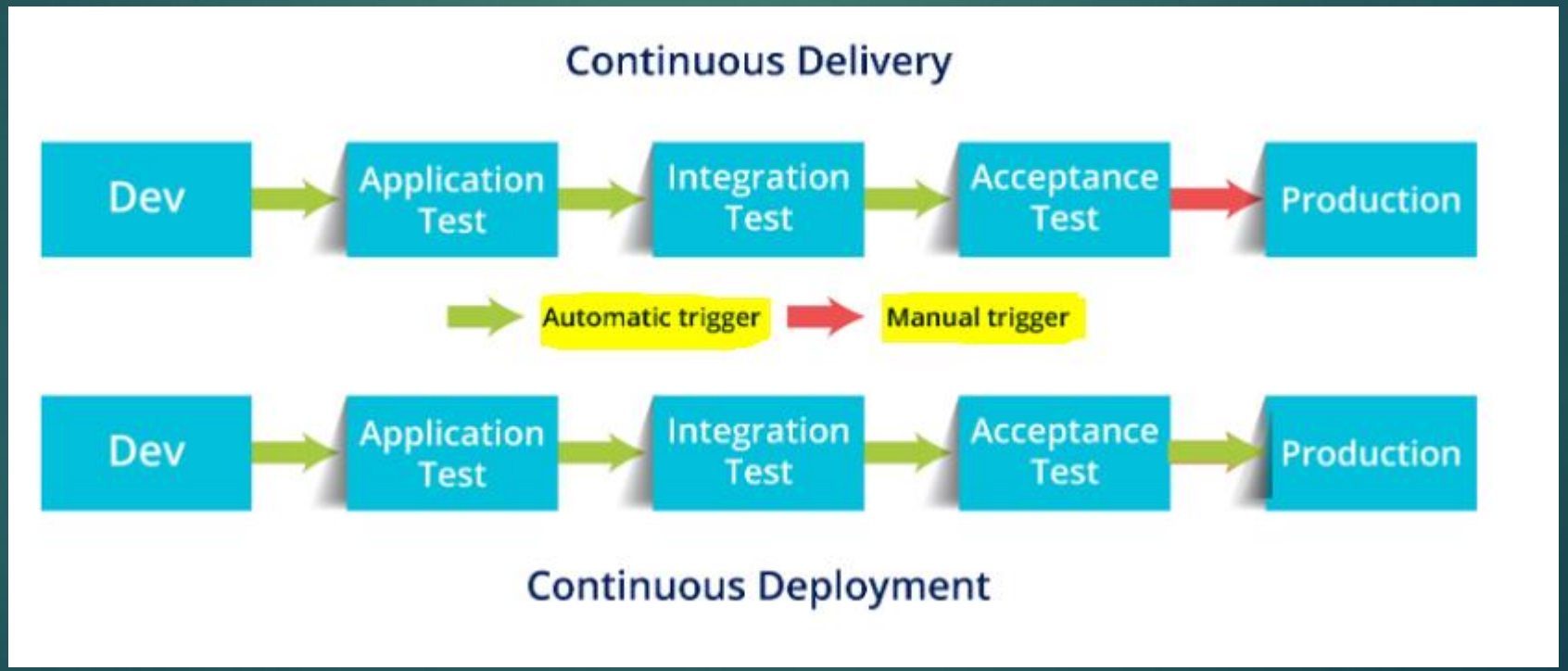
Introduction

- Comment ?



Introduction

- Déploiement continue **VS** Livraison continue



Introduction

Gestion de la configuration avec :

Puppet & Ansible

Introduction

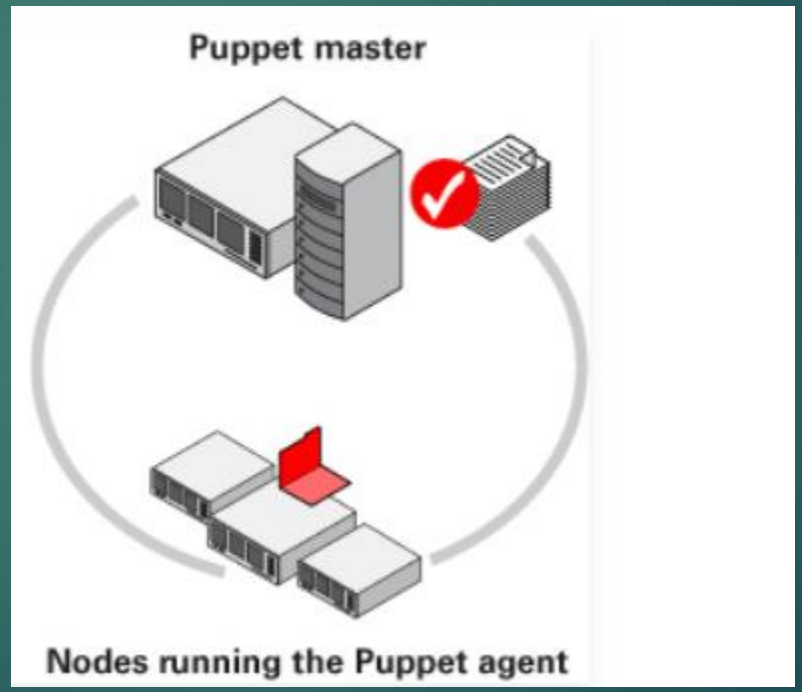
- C’est quoi la gestion de la configuration ?

Elle automatise l'approvisionnement logiciel, la gestion de la configuration et le déploiement applicatif.

Introduction

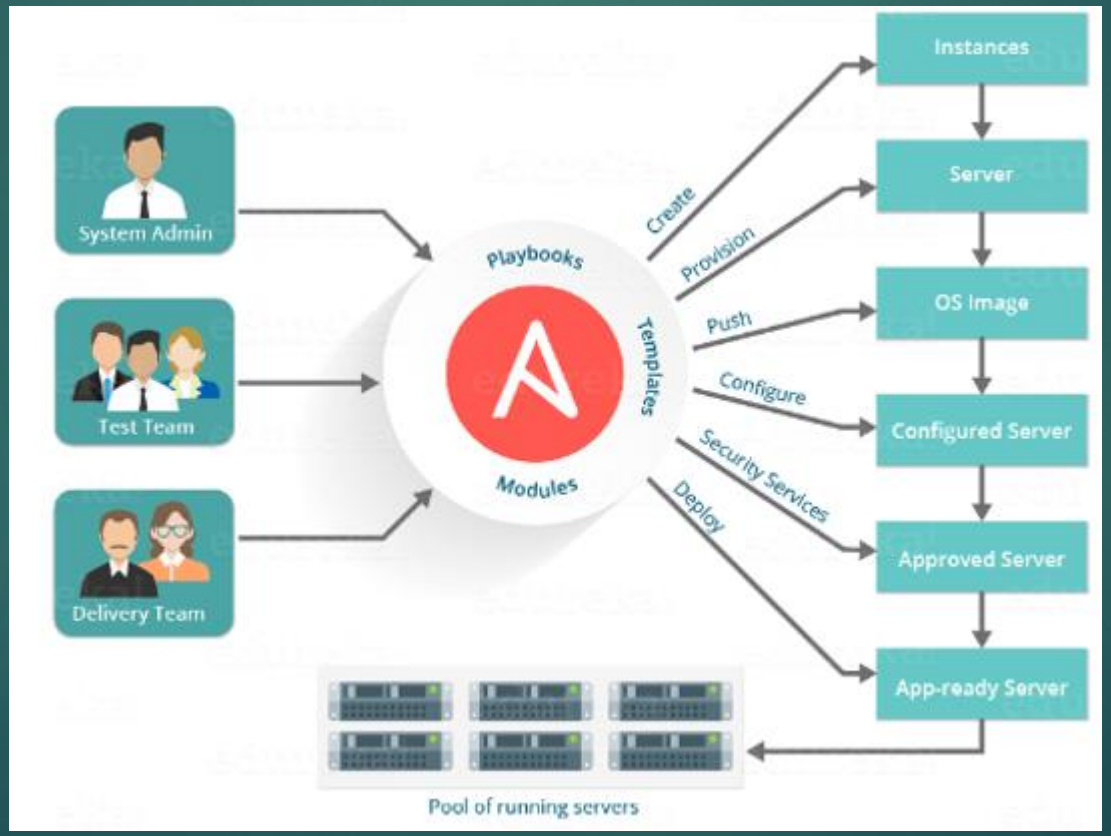
La gestion de la configuration avec PUPPET

1. Node that is running the Puppet agent collects data about itself using facts
2. Agent sends facts to Puppet master
3. Master compiles a catalog based on data for how the node should be configured
4. Master sends catalog back to agent
5. Agent configures itself and reports back to master



Introduction

La gestion de la configuration avec ANSIBLE



Introduction

Conteneurisation avec :

Docker

Introduction

C’est quoi la conteneurisation ?

Elle permet aux instances virtuelles de partager un système d'exploitation hôte unique, avec ses fichiers binaires, bibliothèques ou pilotes.

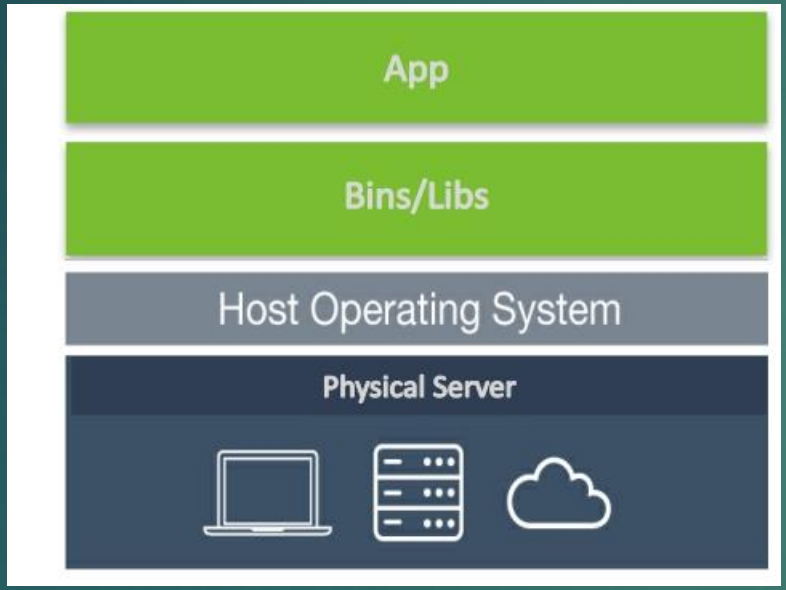
Introduction

C'est quoi un container ?

Un conteneur, c'est un environnement d'exécution complet comprenant :

- Un microservice (application)
- Ses dépendances
- Ses bibliothèques et autres fichiers binaires
- Fichiers de configuration
- Le tout est packagé dans ce que l'on nomme chez docker par exemple **: une image**

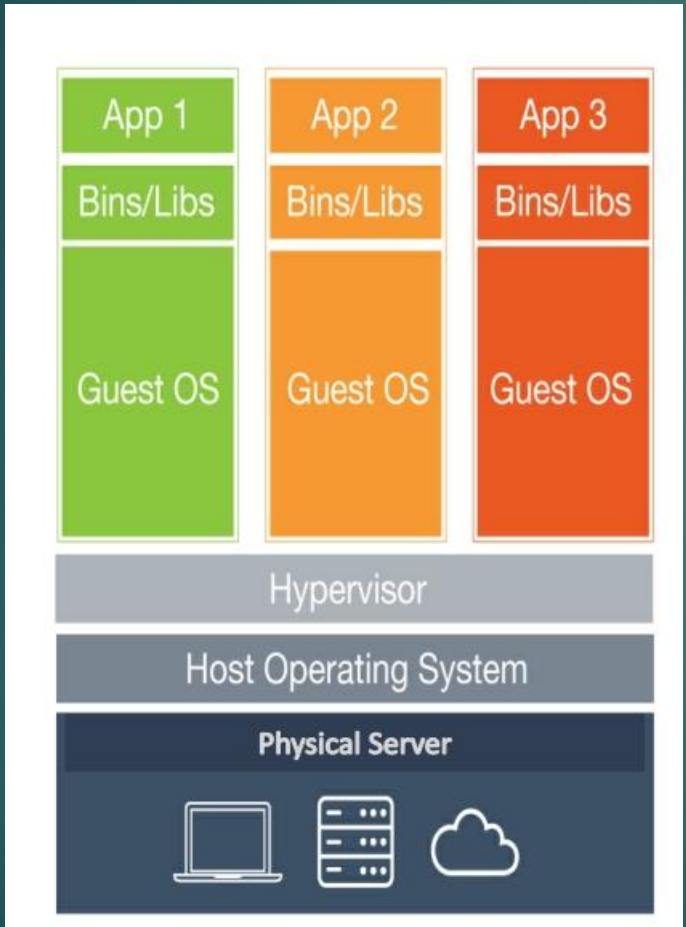
Avant - Virtualisation



Problèmes:

- Coût élevé
- Déploiement lent
- Difficulté de migration

Virtualisation avec hyperviseur



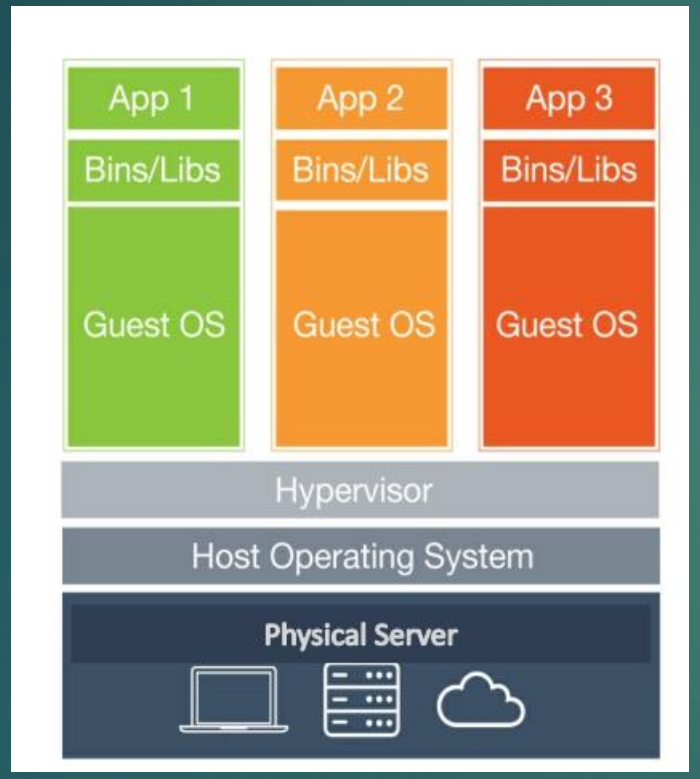
Bénéfices:

- Coût moins élevé
- Easy to Scale

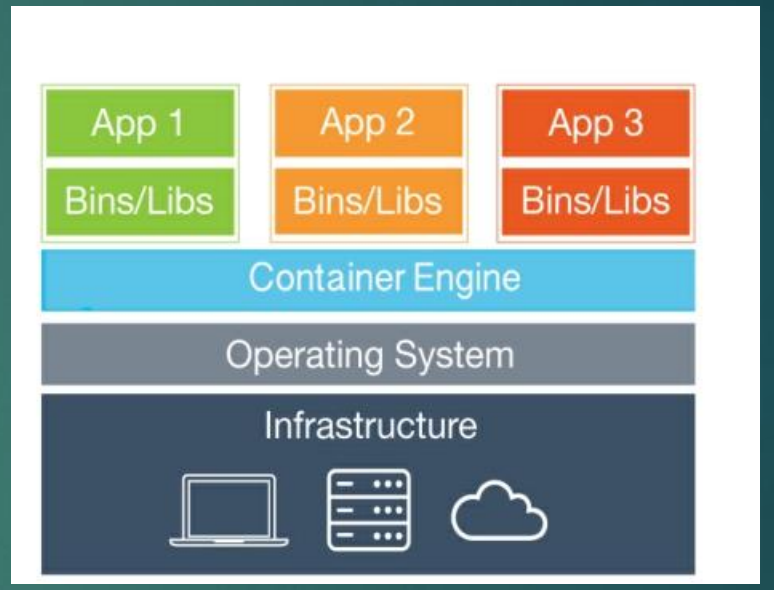
Limitations:

- Duplication de noyau Kernel
- Portatibilité

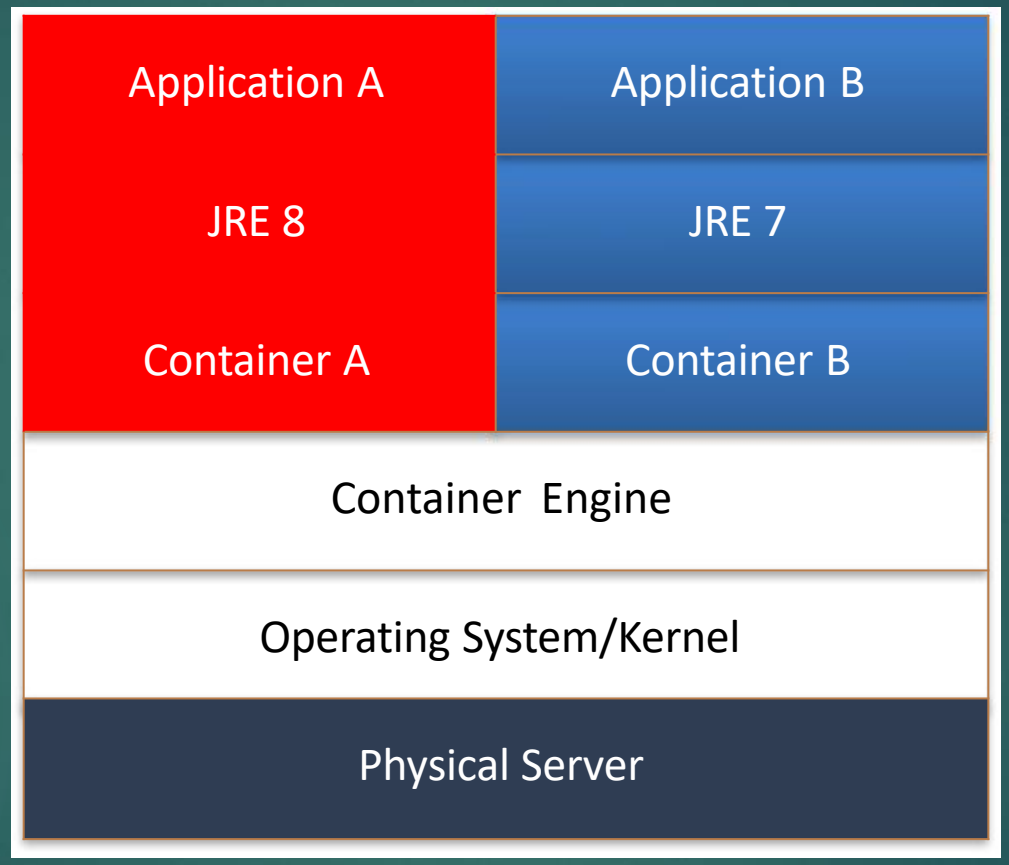
Hyperviseur *VS* Conteneurisation



Containers

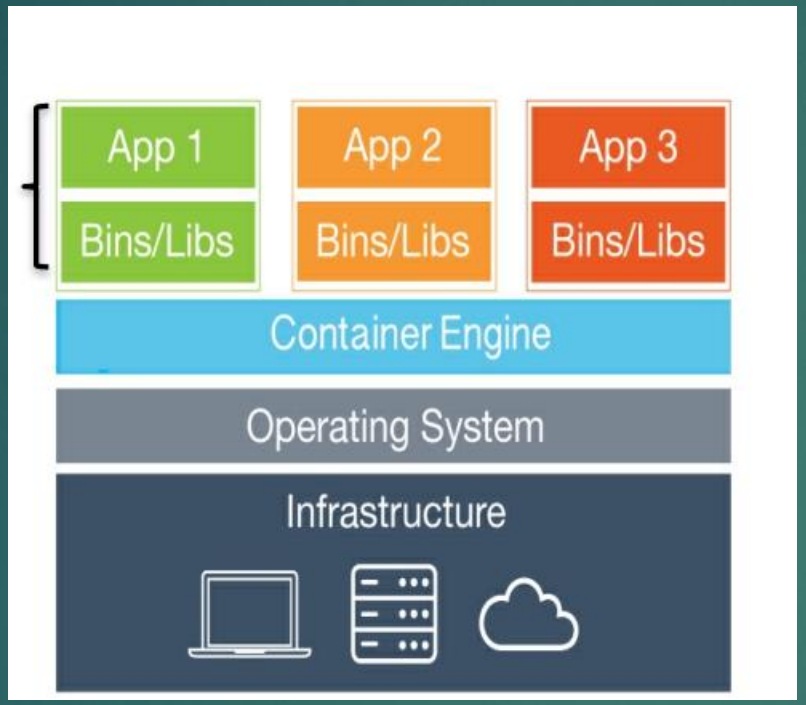


Isolation


Application A
Application B
JRE 8
JRE 7
Container A
Container B
Container Engine
Operating System/Kernel
Physical Server

Conteneurisation

Containers

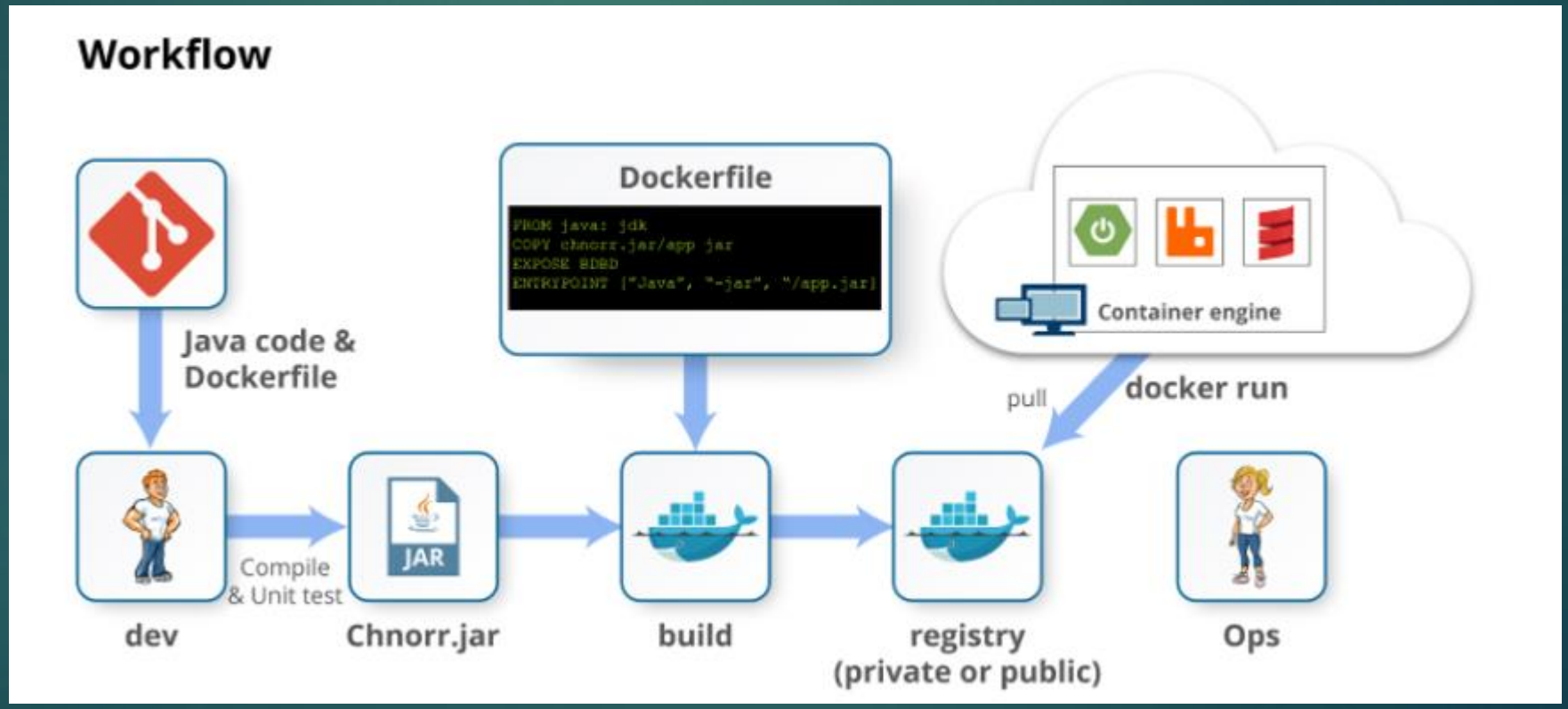


Apports :

- Coût optimisé
- Déploiement rapide
- Portatibilité

Introduction

Conteneuriser avec Docker



Introduction

Avantages de conteneuriser :

- Une taille relativement petite (parfois quelques dizaines de mégaoctets).
- Un conteneur peut démarré presque instantanément
- Ils peuvent être instanciés de manière quasi-immédiat lorsqu'ils sont nécessaires et disparaissent lorsqu'ils ne sont plus nécessaires, libérant ainsi des ressources
- Isolation
- Déploiement multiplateforme
- Maintenabilité et évolutivité du socle applicatif

Introduction

Docker Engine

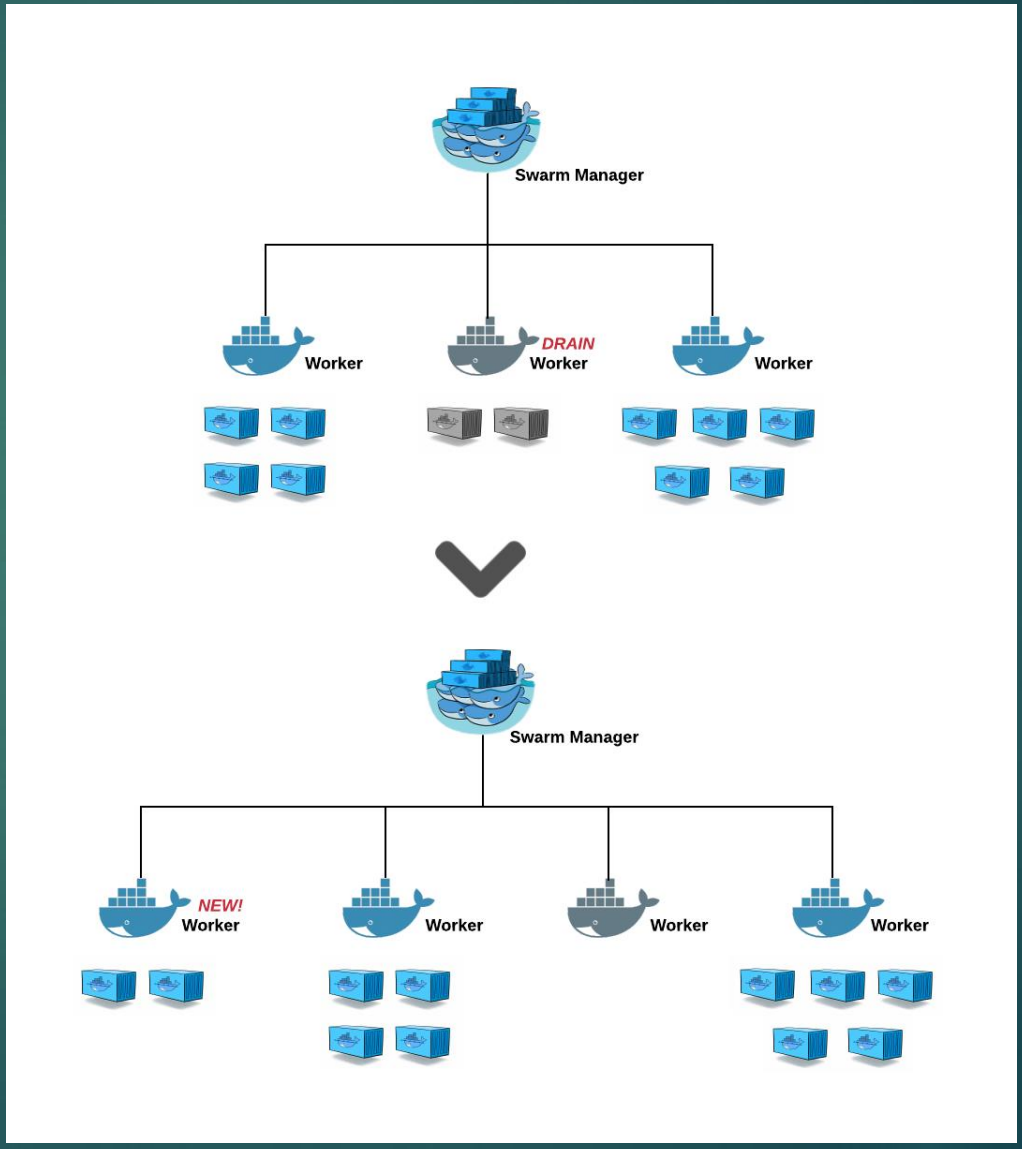
- Docker runtime
- La machine qui fabrique et lance les images

Docker Hub

- Un catalogue en ligne pour trouver ou stocker ses images dans un repository public ou privé
- Permet la fabrication d'image en ligne

Introduction

Docker Swarm



Introduction

Conteneurisation avec :

Kubernetes

Introduction

- Annoncé par Google en 2014. Sa première version sort en 2015
- Il arrête et redémarre pas moins de 2 **milliards** de containers chaque semaine
- Des services comme Gmail, Search, Apps ou Map tournent dans des conteneurs gérés par Kubernetes

Introduction

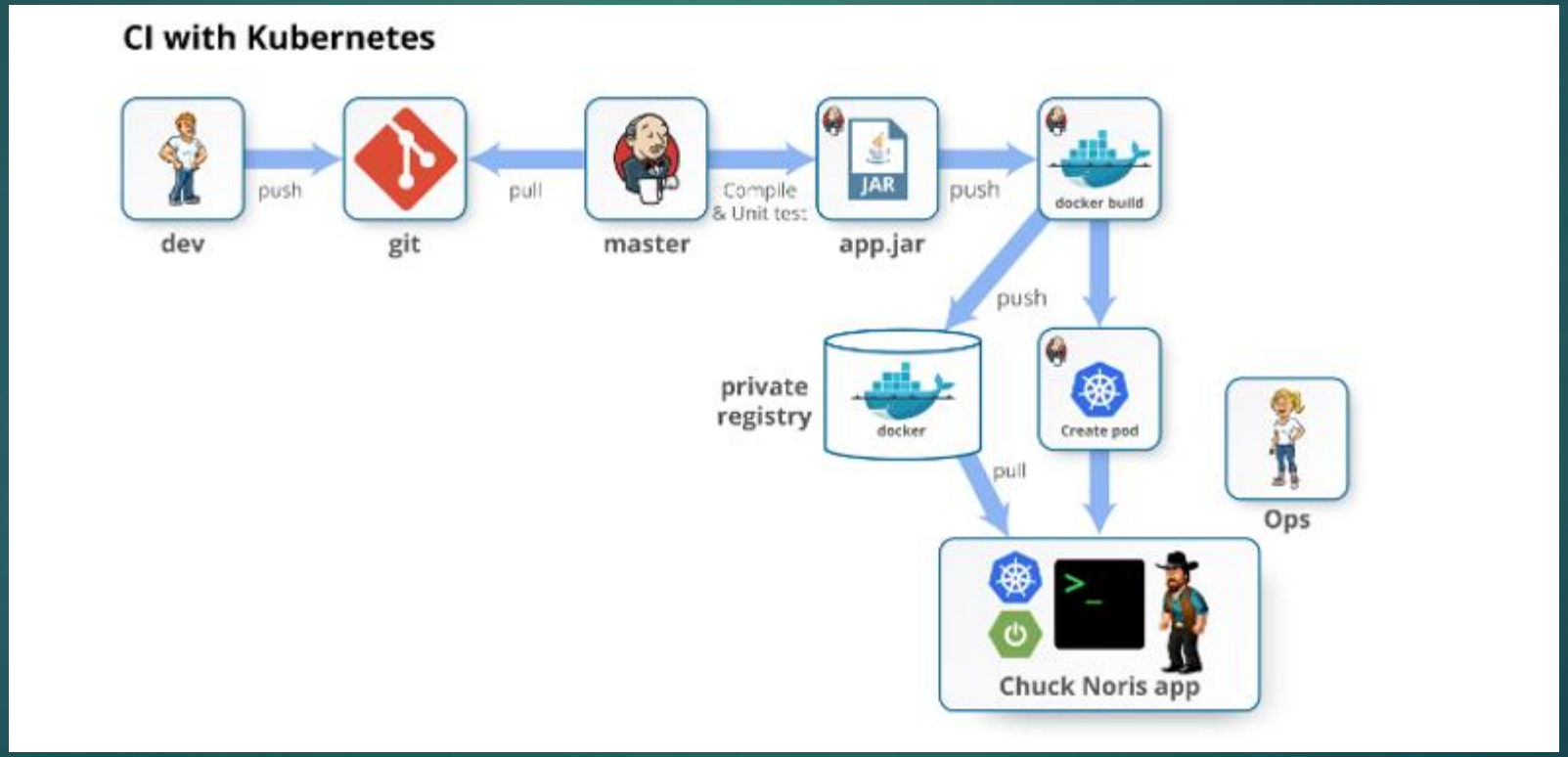
- Kubernetes est un orchestrateur de containers open source
  - Permet de gérer l'état des containers
    - Lancer un container dans un node spécifique
    - Redémarrer un container éteint pour plusieurs raisons pour garantir la disponibilité
    - Déplacer un container d'un node à un node voisin par exemple lors d'une maintenance

Introduction

- Automatise le lancement qui aurait été manuel de containers Docker
- Permet de gérer un réseaux de clusters pouvant atteindre des milliers de nœuds
- Il existe d'autres orchestrateurs de containeurs telles que :
  - Swarm : l' orchestrateur natif de Docker
  - Mesos : Peut faire tourner des container Docker et autres
- Grande modularité : des changements sont intégrables et modifiables
- Open source

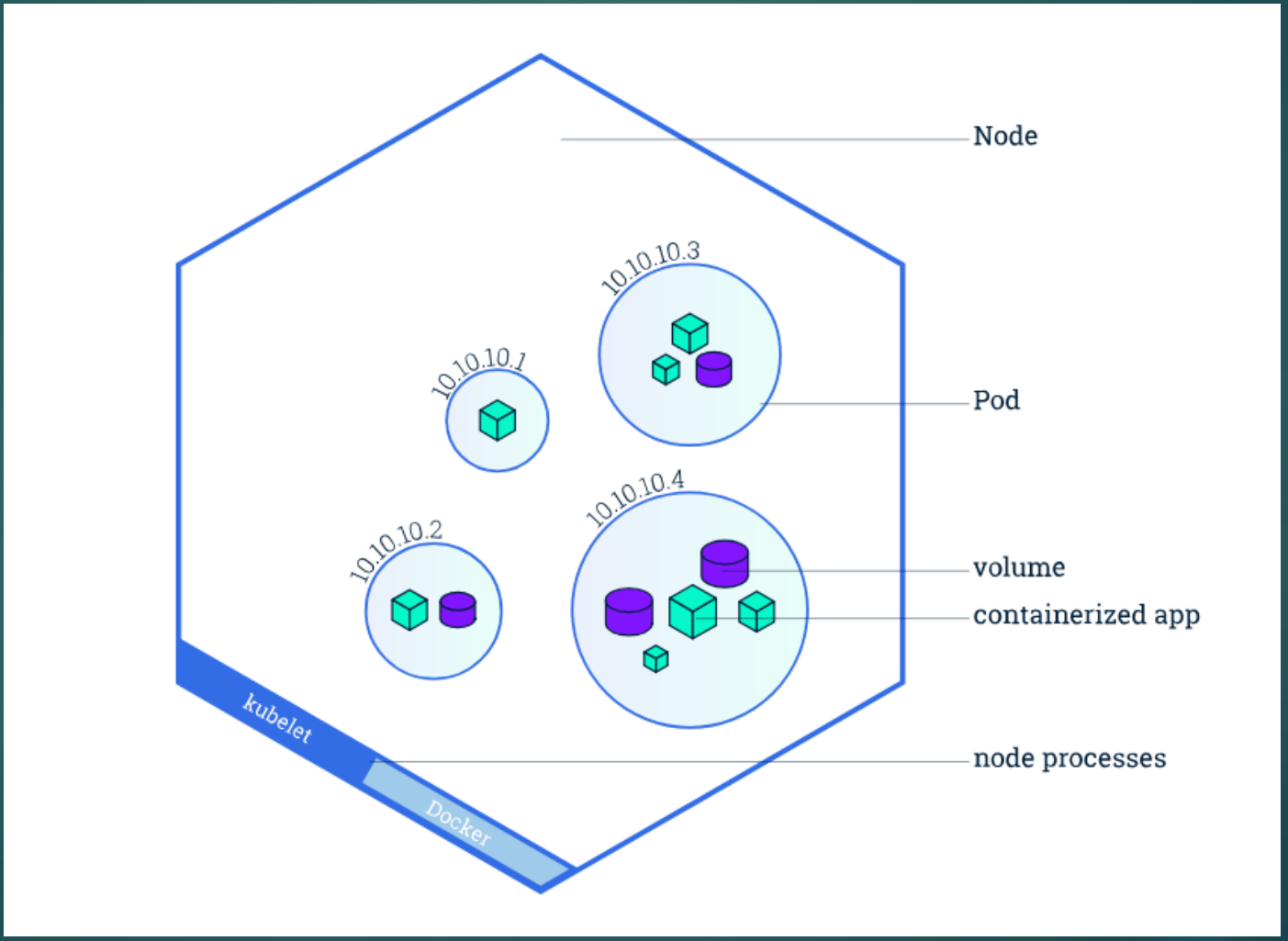
Introduction

Conteneuriser avec Kubernetes



Introduction

Pod & Node

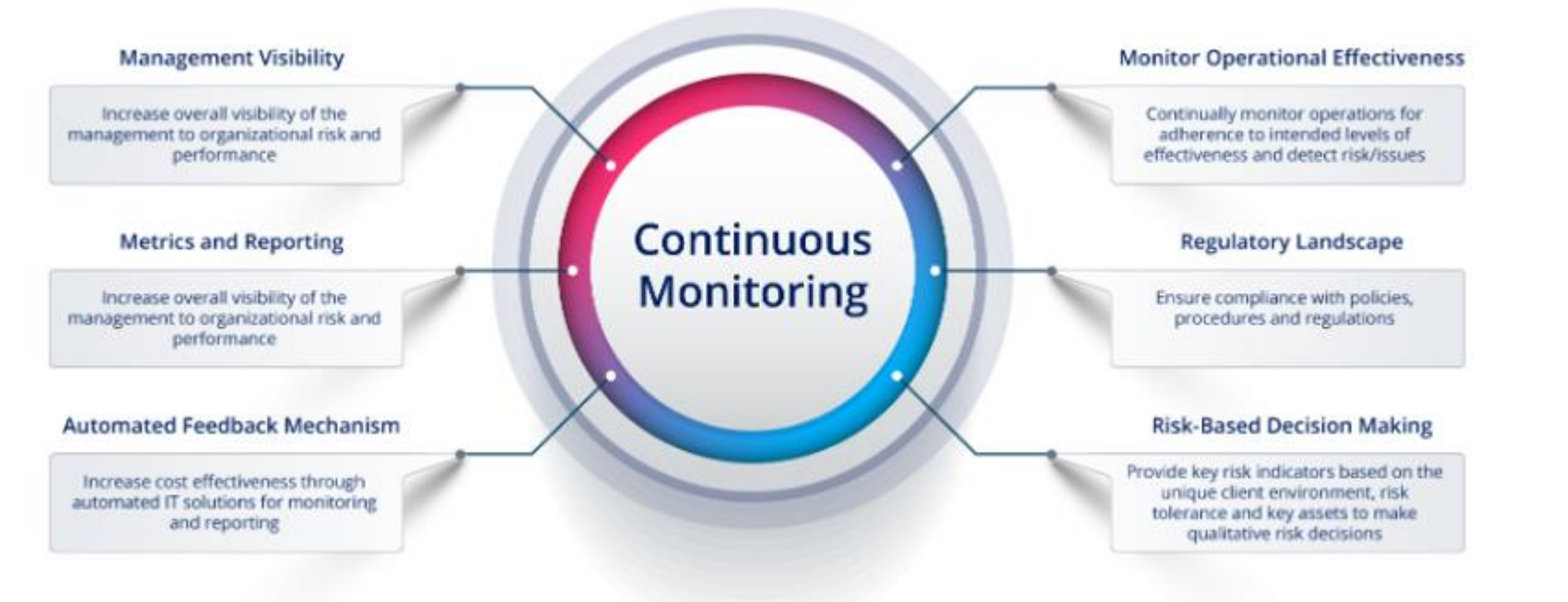


Introduction

Continuous Monitoring

Introduction

- C'est quoi la gestion du monitoring ?

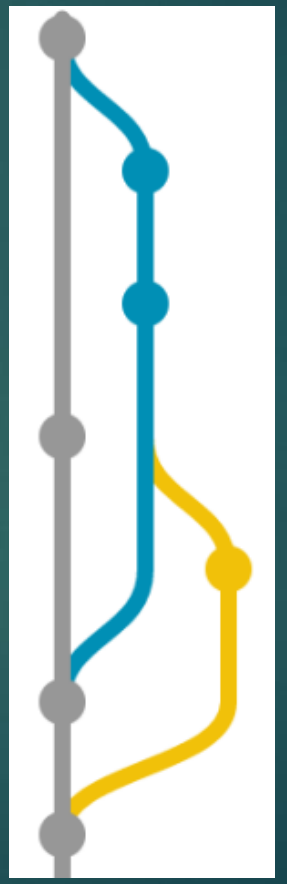


I. Gestion des Versions

Introduction

La gestion de versions (SCM) :

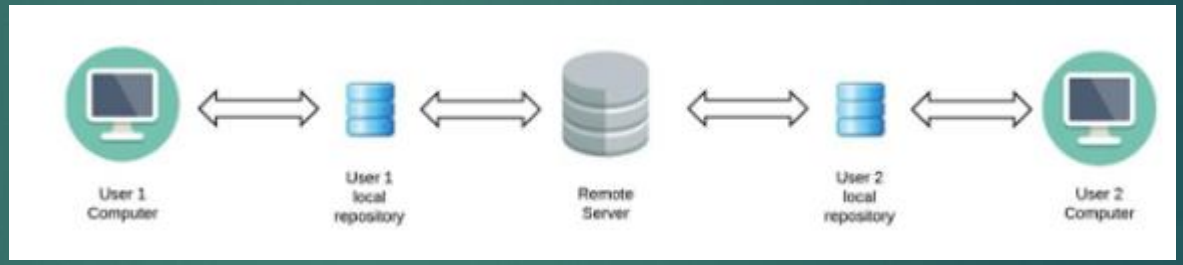
- Elle consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers.



Introduction

La gestion de versions (SCM) :

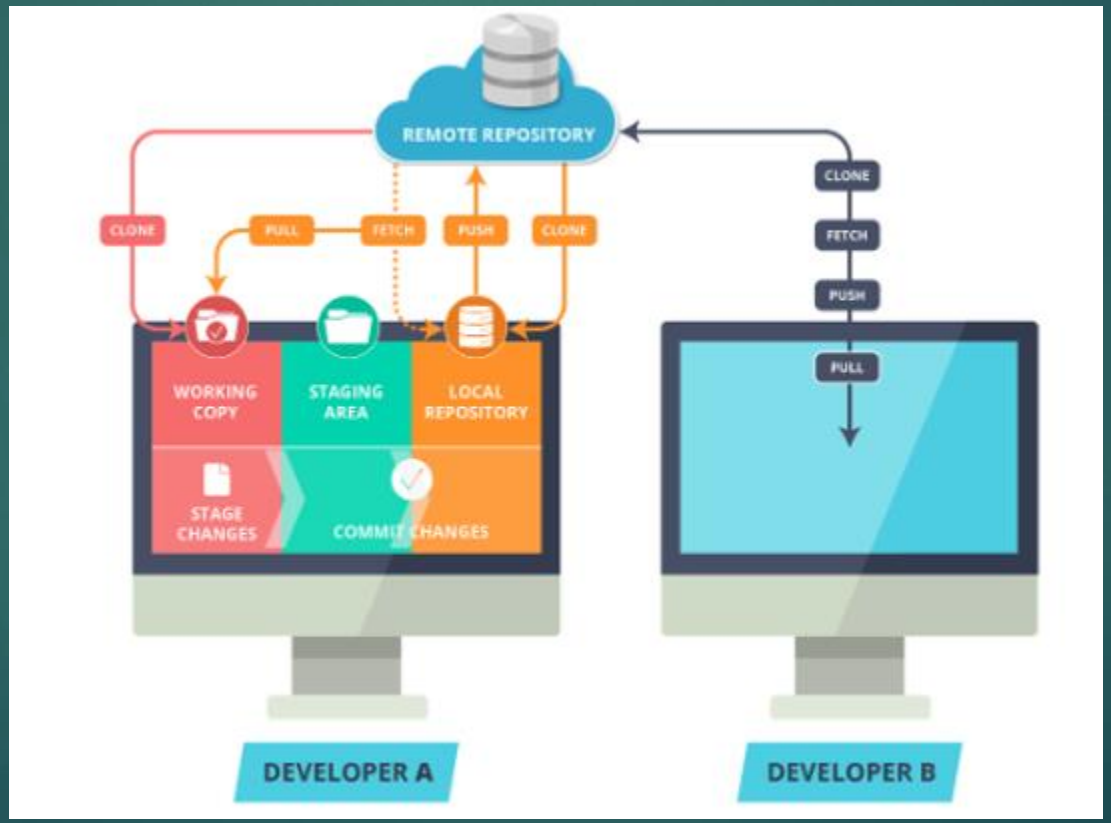
- Décentralisé



- Centralisé

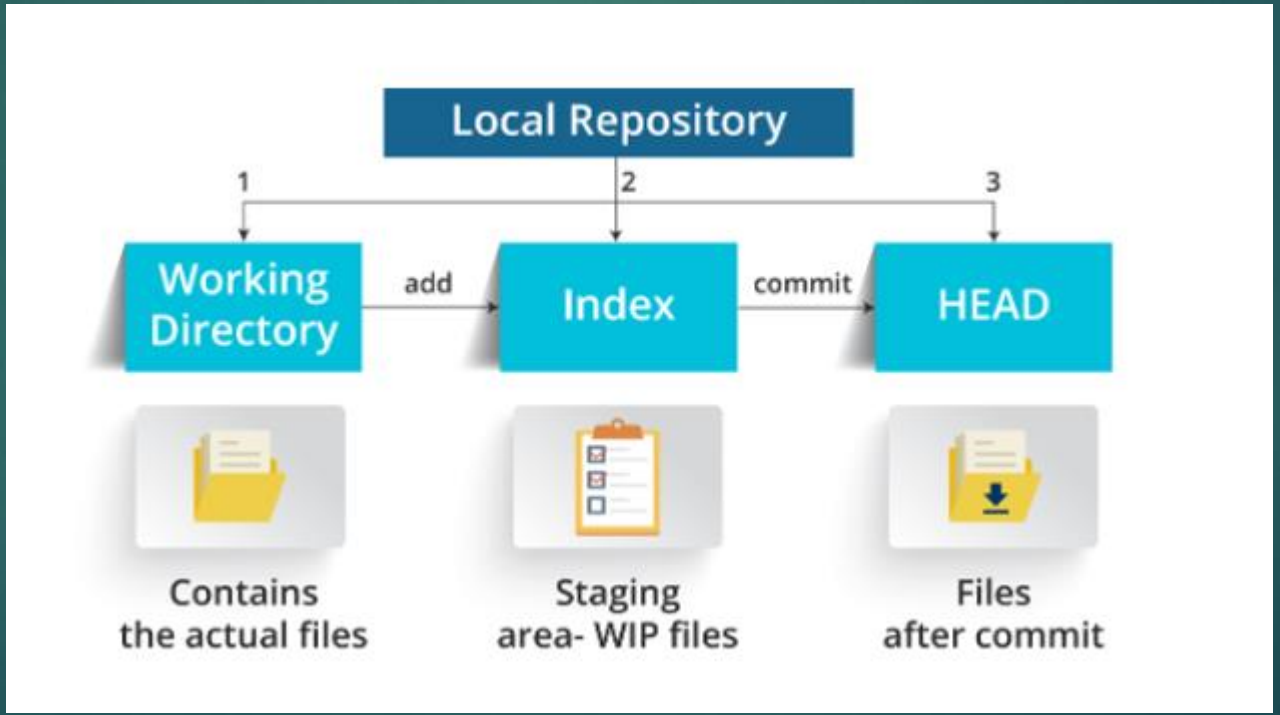
Introduction

La gestion de versions avec GIT



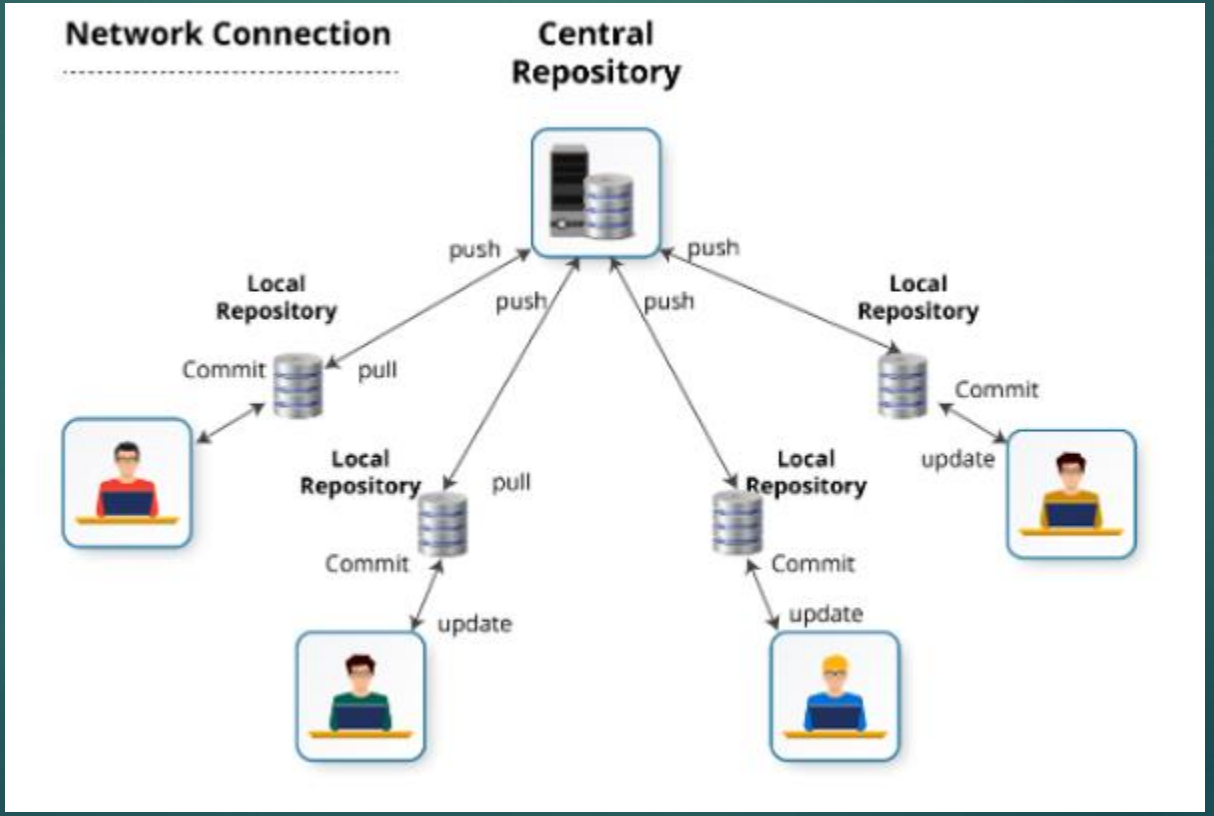
Introduction

Comment ?



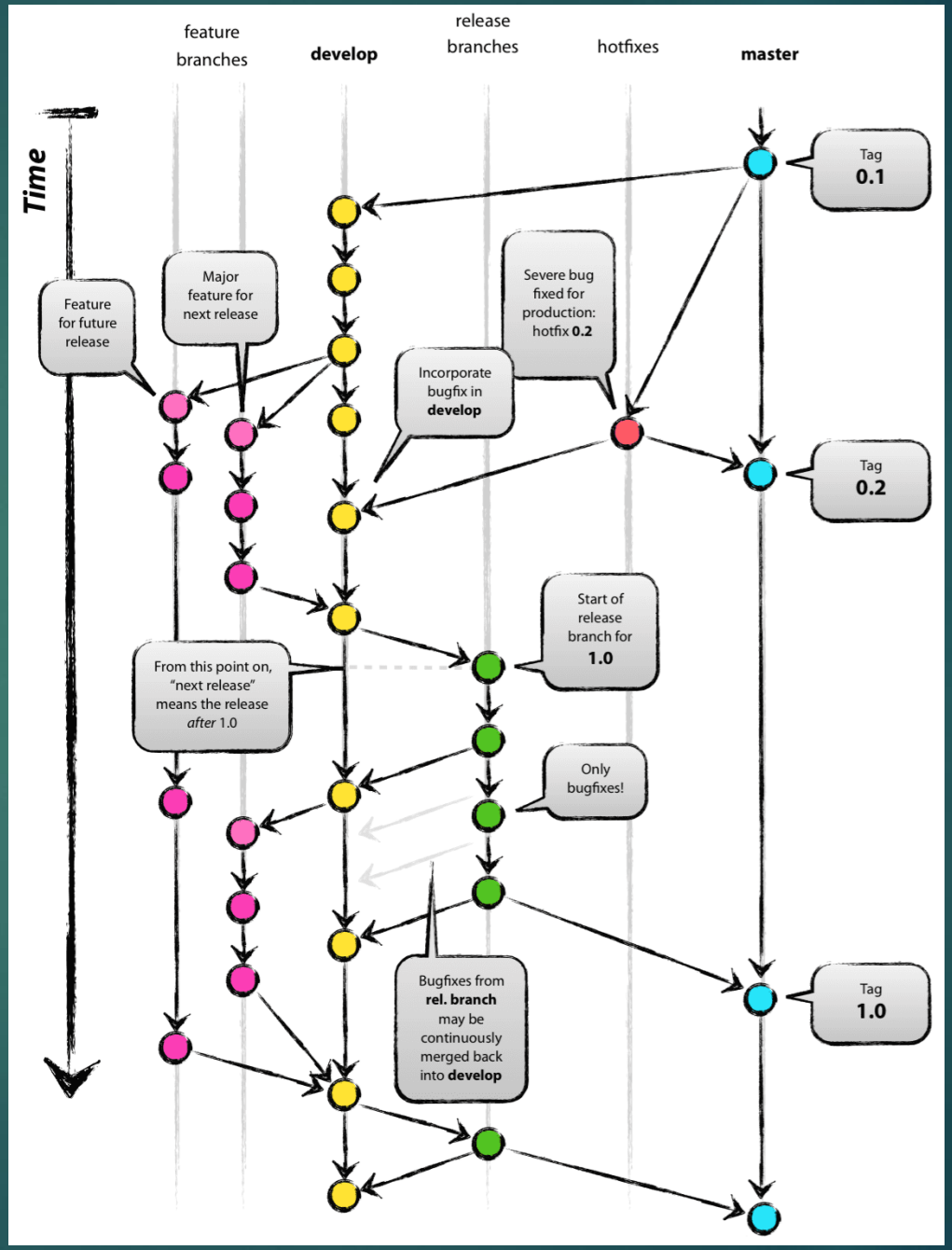
Introduction

Comment ?



Introduction

GitFlow



QUESTIONS ?

MERCI