Module: Initiation à l'automatisation

Chapitre 1
Vue d'ensemble

Soumaya MBAREK
Soumaya.mbarek@esprit.tn

A.U. 2021/2022

Automatisation ?

- L'automatisation correspond à l'utilisation de technologies pour **effectuer certaines tâches sans intervention humaine**.
- De nombreux secteurs y ont recours (la fabrication, la robotique, le contrôle des véhicules, etc.)
  - **L'automatisation informatique** consiste à utiliser un système d'instructions pour effectuer un ensemble répétitif de processus qui remplace le travail manuel réalisé sur les systèmes informatiques.
  - **L'automatisation métier** consiste à aligner la gestion des processus métier et la gestion des règles métier sur le développement d'applications modernes.

https://www.redhat.com/fr/topics/automation

Pourquoi automatiser ?

Pourquoi automatiser ?

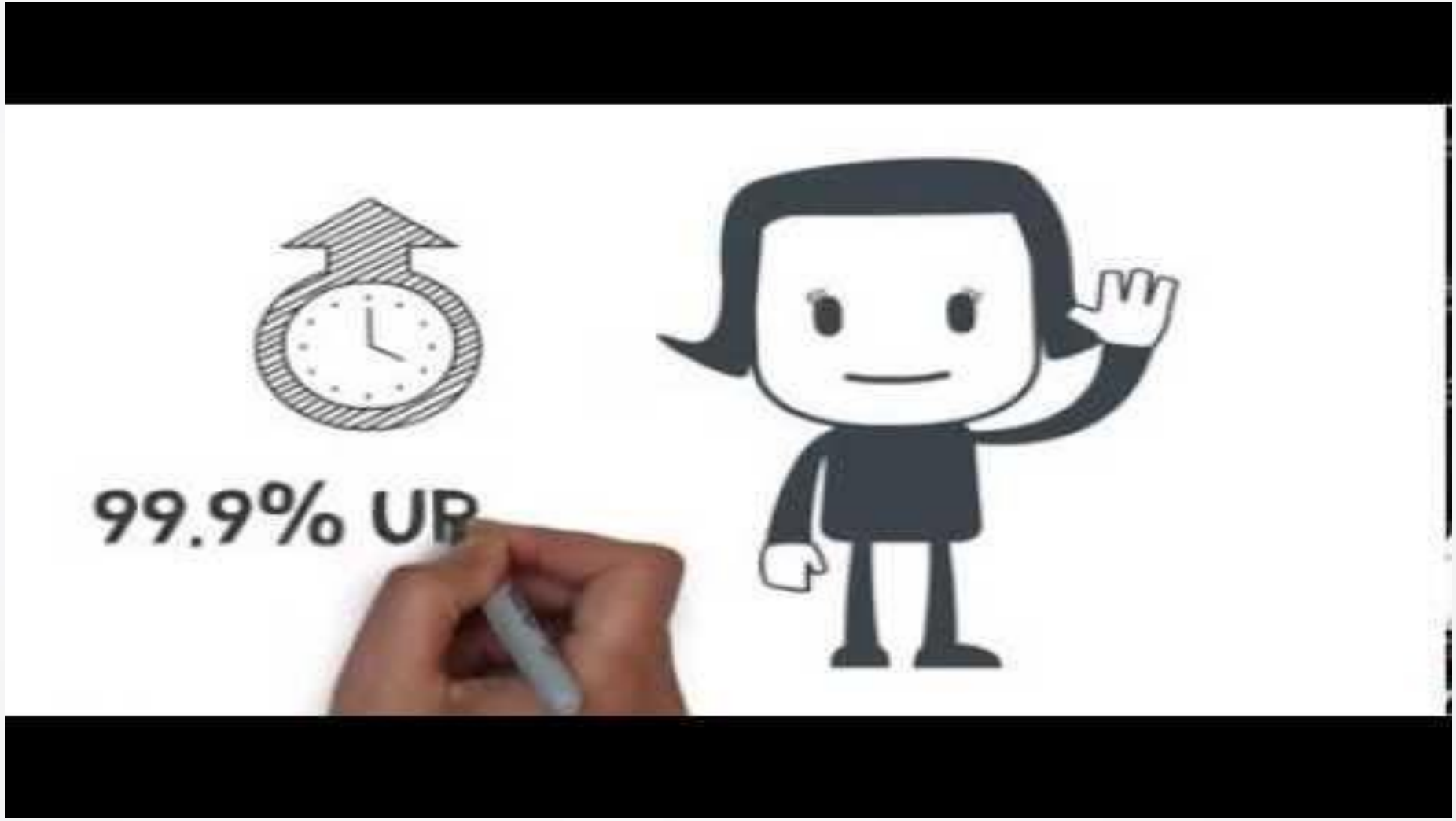
- Parque c’est une technologie clé utlisé dans plusieurs domaines, par exemple:
  - DevOps
  - Native Cloud Applications

DevOps: Définition

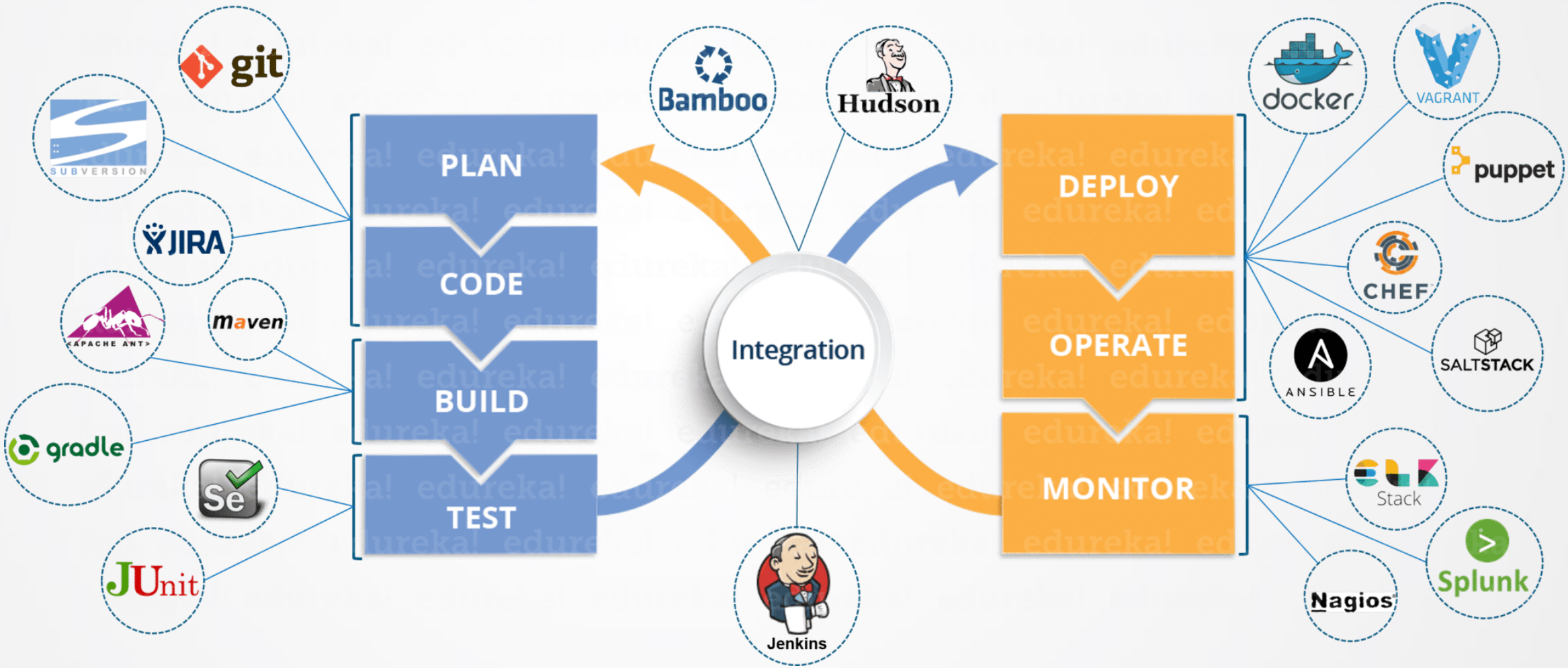
- DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

https://aws.amazon.com/devops/what-is-devops/

DevOps: Définition



Les outils DevOps



https://aws.amazon.com/devops/what-is-devops/

Les pratiques clé de DevOps

- Continuous Integration
- Continuous Delivery
- Microservices
- Infrastructure as Code
- Monitoring and Logging
- Communication and Collaboration

https://aws.amazon.com/devops/what-is-devops/

CI/CD

- L’approche CI/CD se rapporte à un processus, souvent représenté sous forme de pipeline, qui consiste à introduire un haut degré d'automatisation et de surveillance continues dans le processus de développement des applications.

https://www.redhat.com/fr/topics/devops/what-is-ci-cd

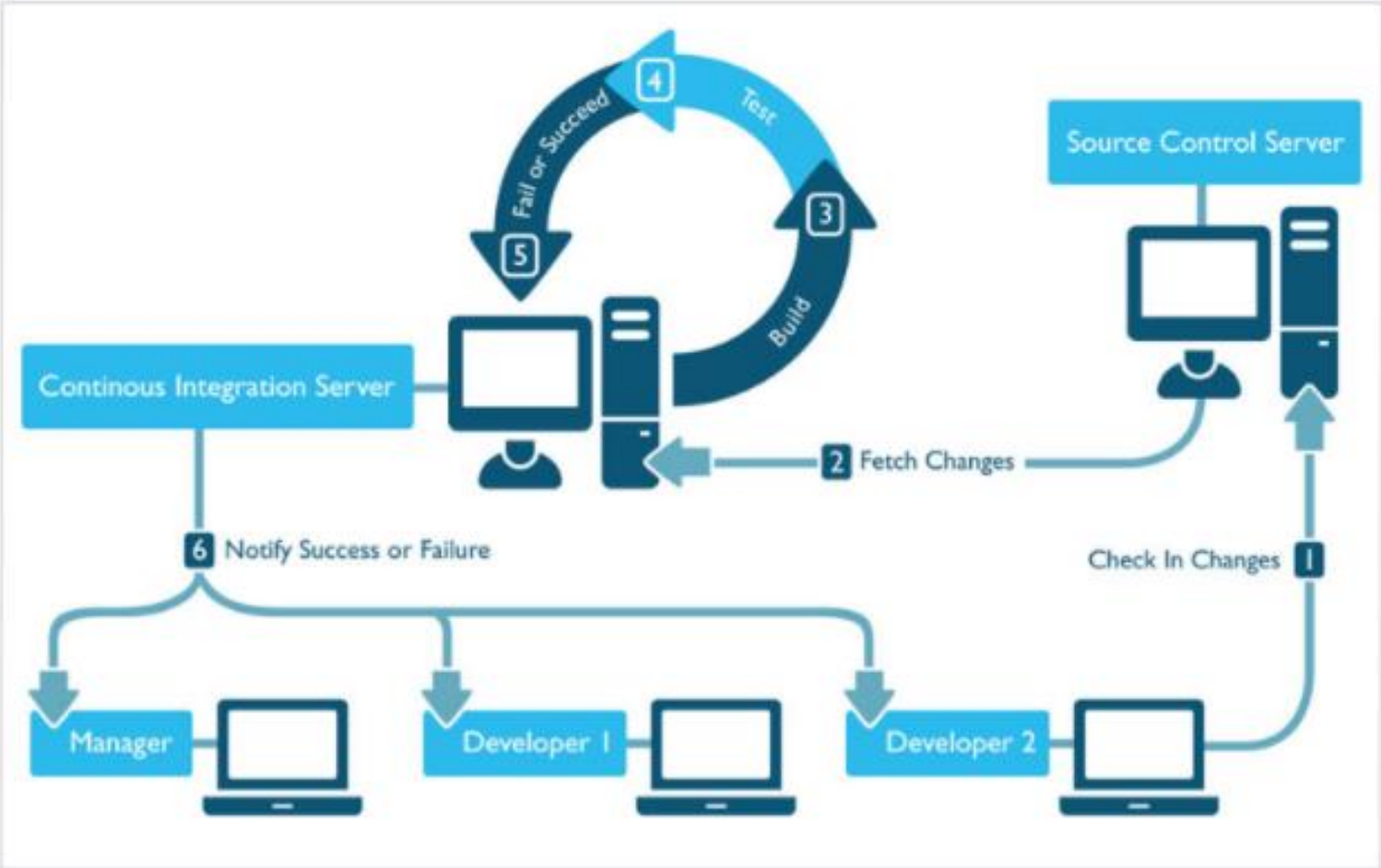
L'intégration continue

- L’Intégration Continue ou Continuous Integration est un processus consistant à compiler, tester et déployer sur un environnement d’intégration **d’une manière automatisée.**

- Le but est de tester aussi souvent et autant que possible les livrables pour détecter plus tôt possible.

- La plupart du travail est réalisé par des outils de test. Le déploiement sur la plateforme d’intégration devient simple et peut être réalisé par des outils/pipelines sans faire intervenir le développeur/testeur.

L'intégration continue : processus



https://developers.redhat.com/blog/2017/09/06/continuous-integration-a-typical-process/

Livraison Continue

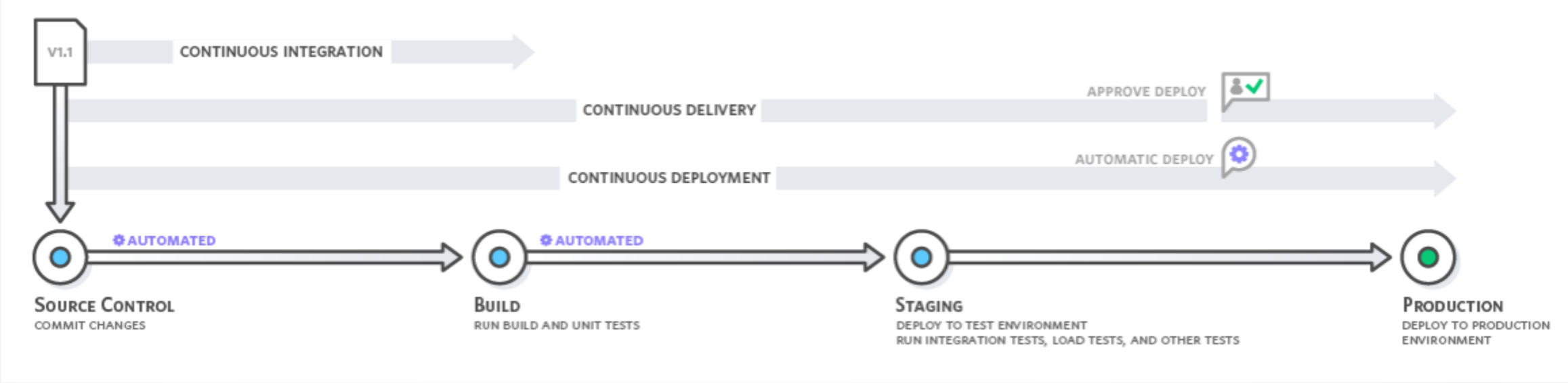
- Après l'automatisation de la création et des tests unitaires et d'intégration dans le cadre de l'intégration continue, la distribution continue automatise la publication du code validé dans un référentiel.
- Chaque étape implique l'automatisation des processus de test et de publication du code.
- Output final: une version de l’application prête pour la production

Déploiement continue

- En complément du processus de distribution continue, qui automatise la publication d'une version prête pour la production dans un référentiel de code, le déploiement continu automatise le lancement d'une application dans un environnement de production.
- Output : l’application en exécution dans son environnement de production.

Livraison Continue vs Déploiement continue

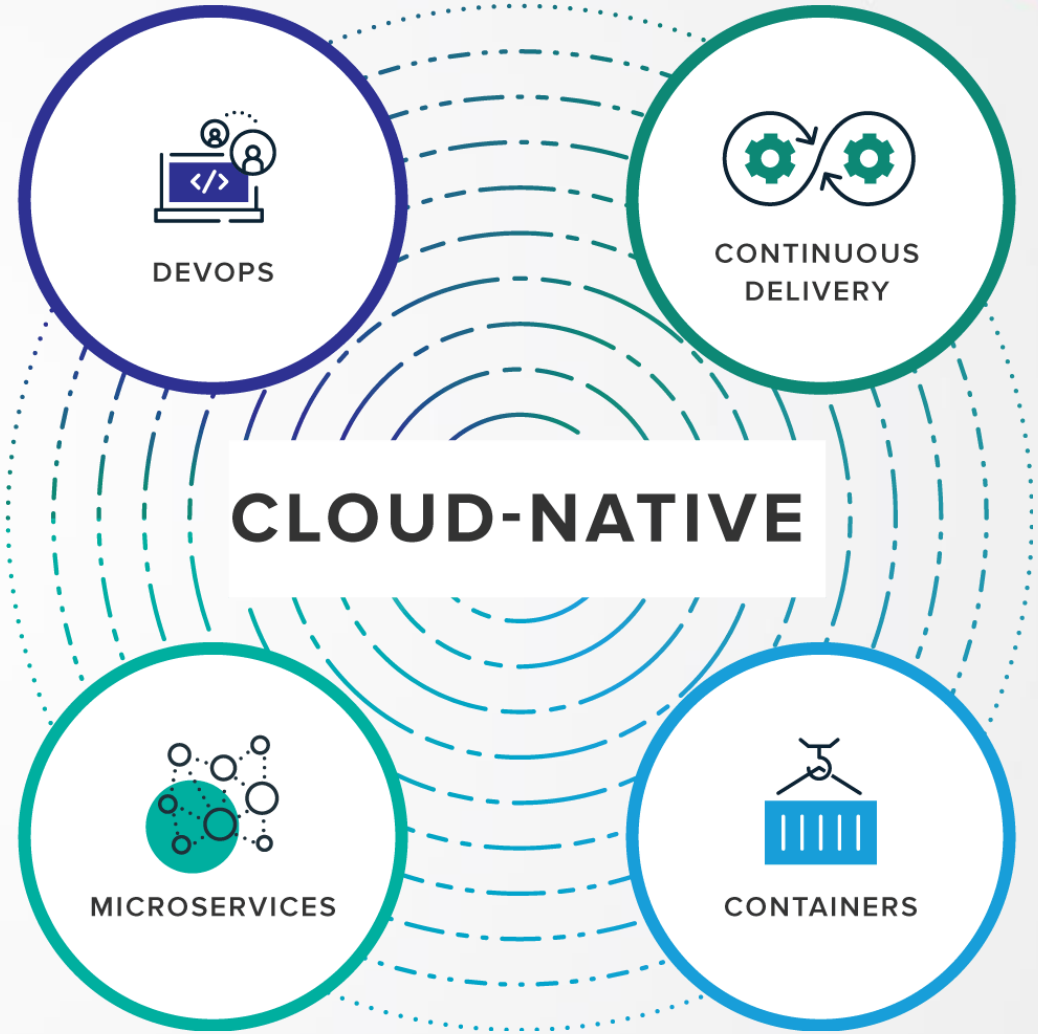
- La différence entre la livraison continue et le déploiement continu réside dans la présence d'une approbation manuelle pour mettre à jour et produire. Avec le déploiement continu, la production se fait automatiquement, sans approbation explicite.



Cloud Native

Cloud-native est une approche pour construire et exécuter des applications qui exploitent pleinement les avantages du modèle de livraison du cloud computing. L'approche Cloud-native est la façon dont les applications sont créées et déployées, non pas où elles sont exécutées.

Pour créer et exploiter des applications natives du cloud , Les organisations ont besoin d'une plate-forme qui **automatisent** et intègrent les concepts de **DevOp**s, de **livraison en continue**, de **micro-services** et de **conteneurs.**



https://www.supinfo.com/articles/single/5015-qu-est-ce-qu-une-application-cloud-native

Applications traditionnelles Vs Applications Cloud native

- **Applications traditionnelles**
  - Imprévisible
  - Dépendance au système d'exploitation
  - Capacité surdimensionnée
  - Isolée
  - Baisse de rendement
  - Dépendante
  - Mise à l'échelle manuelle
  - Restauration lente

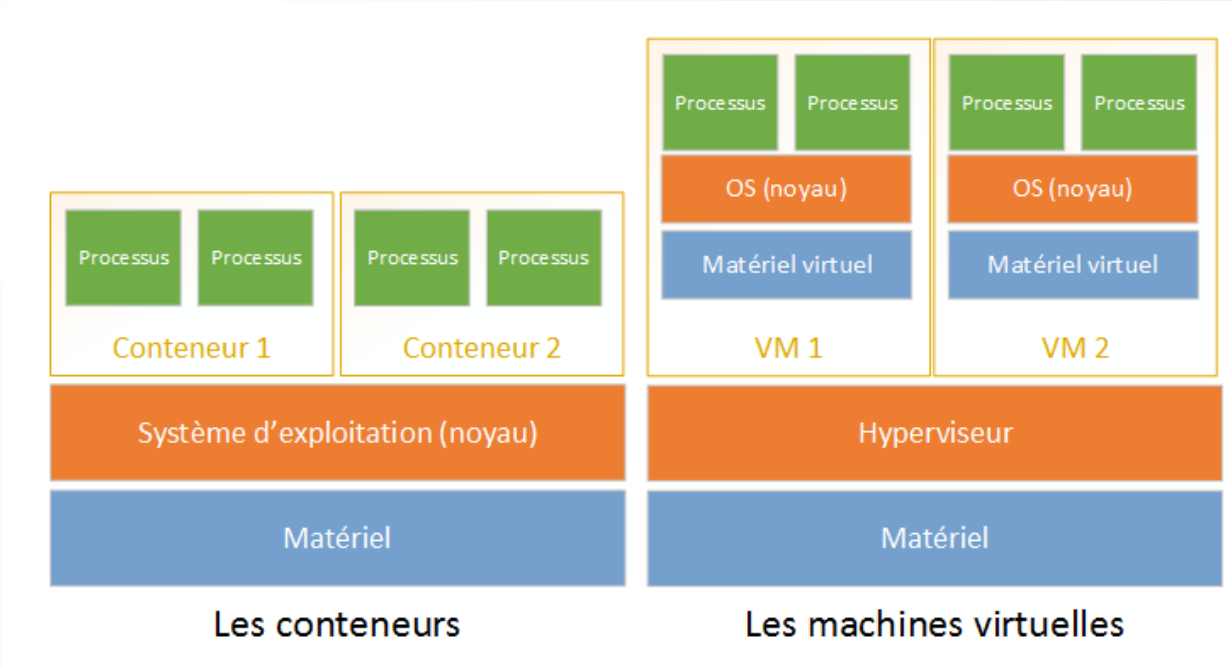
- **Applications Cloud native**
  - Prévisible
  - Abstraction du système d'exploitation
  - Bon dimensionnement
  - Collaboration
  - Déploiement en continue
  - Indépendante
  - Mise à l’échelle automatisé (Auto-Scaling)
  - Restauration rapide

Conteneurs

Un conteneur consiste en un environnement d'exécution complet : une application, plus toutes ses dépendances, ses bibliothèques et autres fichiers binaires, ainsi que les fichiers de configuration nécessaires pour l'exécuter, regroupés dans un seul package.

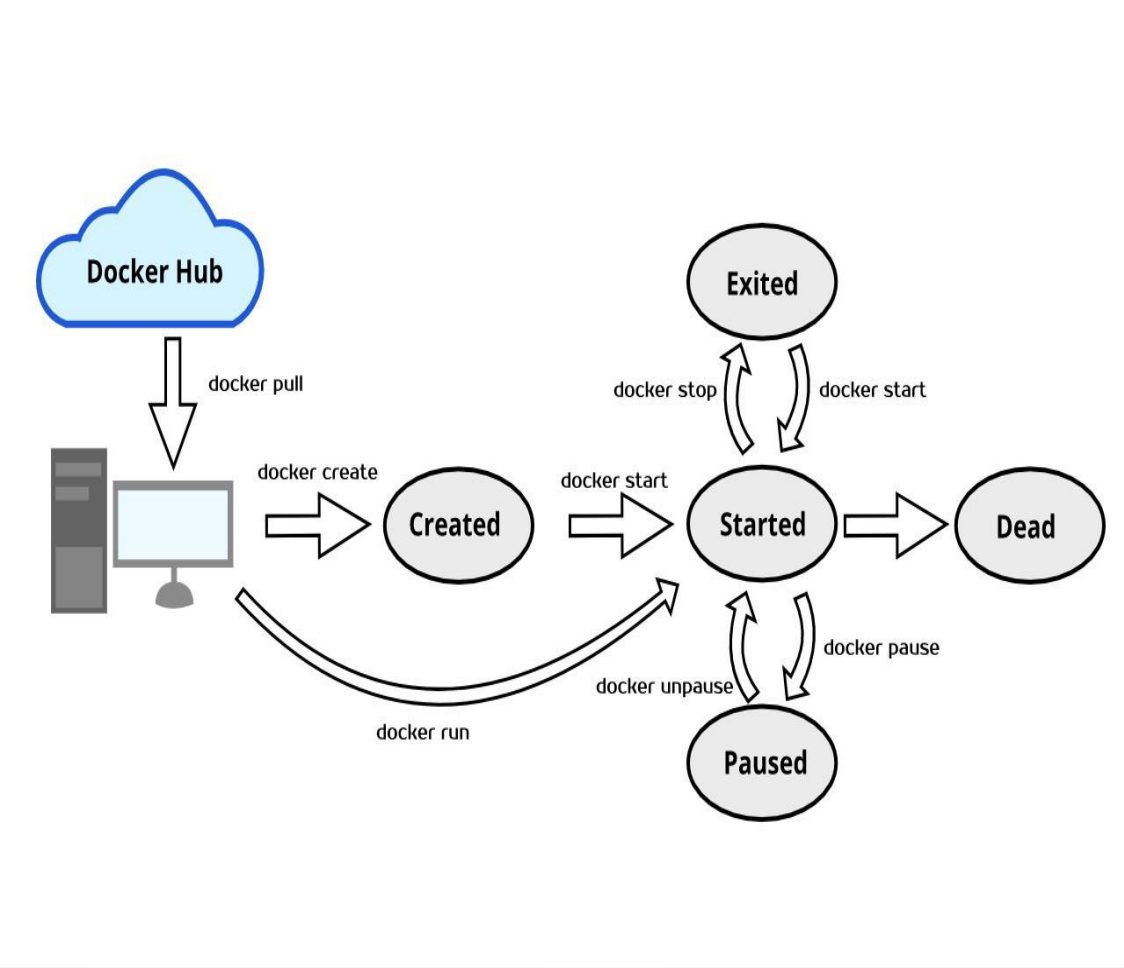
Avec la virtualisation, le package qui peut être transmis est une machine virtuelle, et comprend un système d'exploitation complet ainsi que l'application.

un serveur exécutant trois applications conteneurisées exécute un seul système d'exploitation et chaque conteneur partage le noyau du système d'exploitation avec les autres conteneurs



https://www.supinfo.com/articles/single/5473-conteneurs-pourquoi-t-on-besoin

Conteneur comment ça marche : Exemple Docker

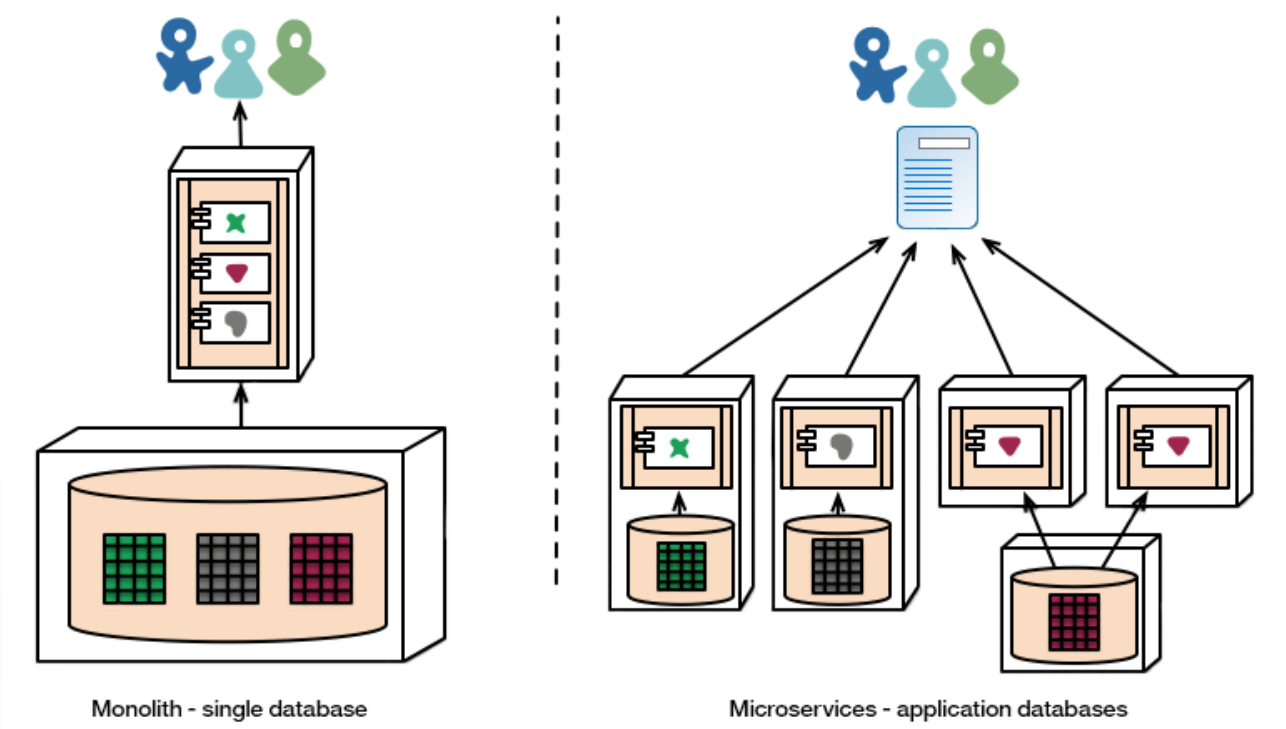


Avantages

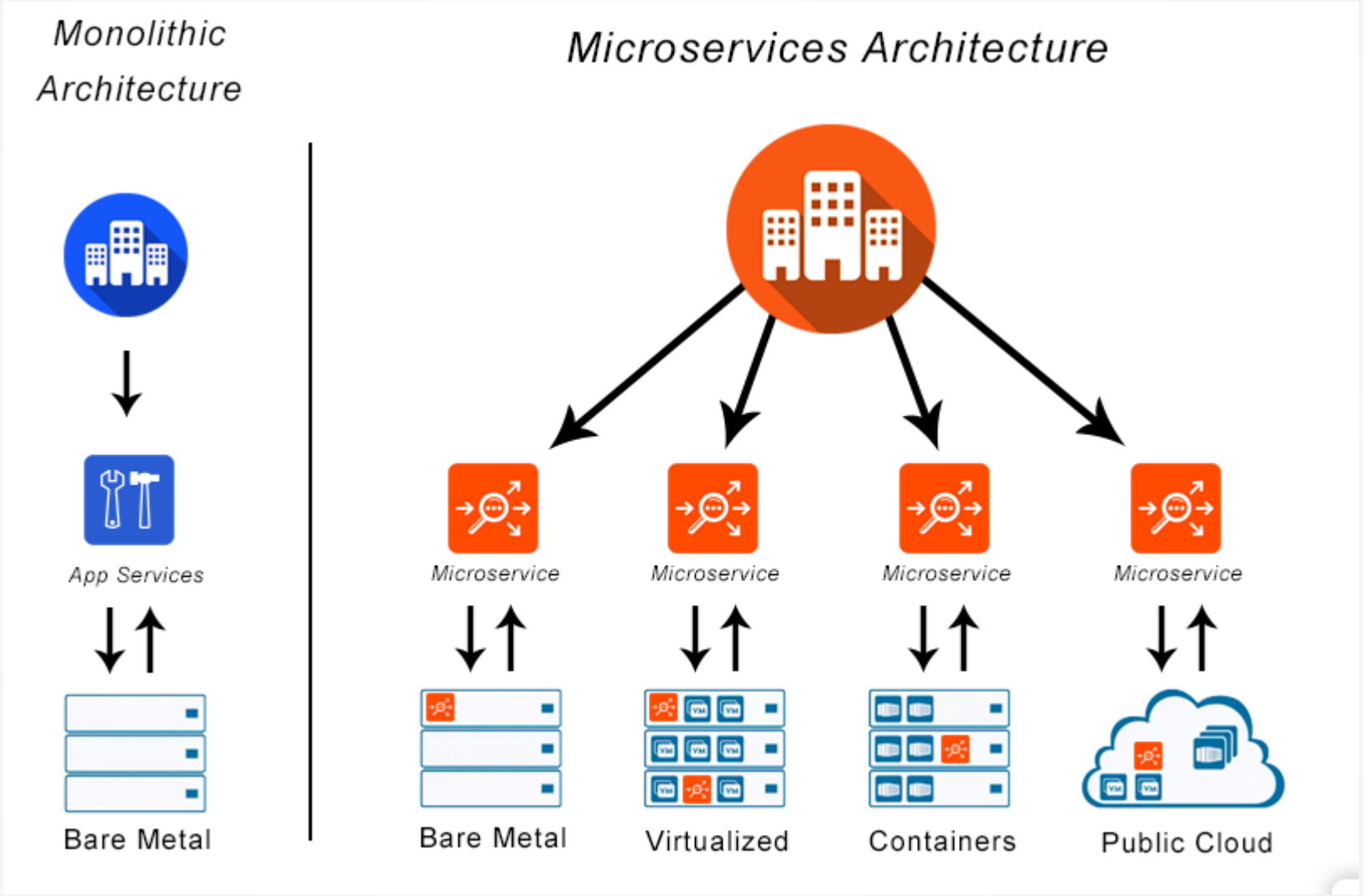
- Optimisation de l’utilisation des ressources
- Rapidité de déploiement des applications et de la libération des ressources.
- Meilleur disponibilité
- Modularité.

Microservices

les microservices sont une technique de développement logiciel — une variante du style architectural de l'architecture orientée services (SOA) — qui structure une application comme un ensemble de services faiblement couplés. Les microservices indépendants communiquent les uns avec les autres en utilisant des API indépendantes du langage de programmation.



https://fr.wikipedia.org/wiki/Microservices

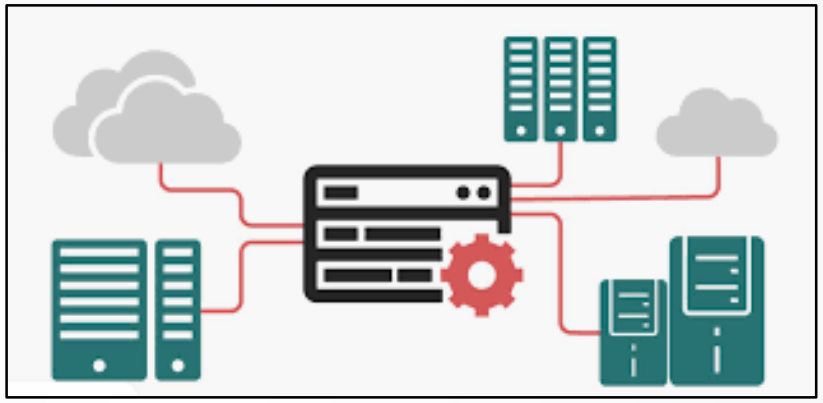


https://avinetworks.com/glossary/microservice/

Automatisation: Cas d’utilisations

Cas d’utilisations

Gestion des configurations



Approvisionnement

Déploiement d'applications

Orchestration

Sécurité et conformité

Gestion de configurations

- La **gestion de configuration** consiste à gérer la description technique d'un système (et de ses divers composants), ainsi qu'à gérer l'ensemble des modifications apportées au cours de l'évolution du système.
- la **gestion de configuration** peut être utilisée à plusieurs fins :
  - Pour stocker et tracer les différentes versions ou révisions de toute information destinée à être utilisée par un système (matériel, logiciel, document, donnée unitaire, etc.).
  - Pour déployer des configurations à travers un parc informatique sous formes de fichiers et données.

https://fr.wikipedia.org/wiki/Gestion_de_configuration

Approvisionnement de ressources

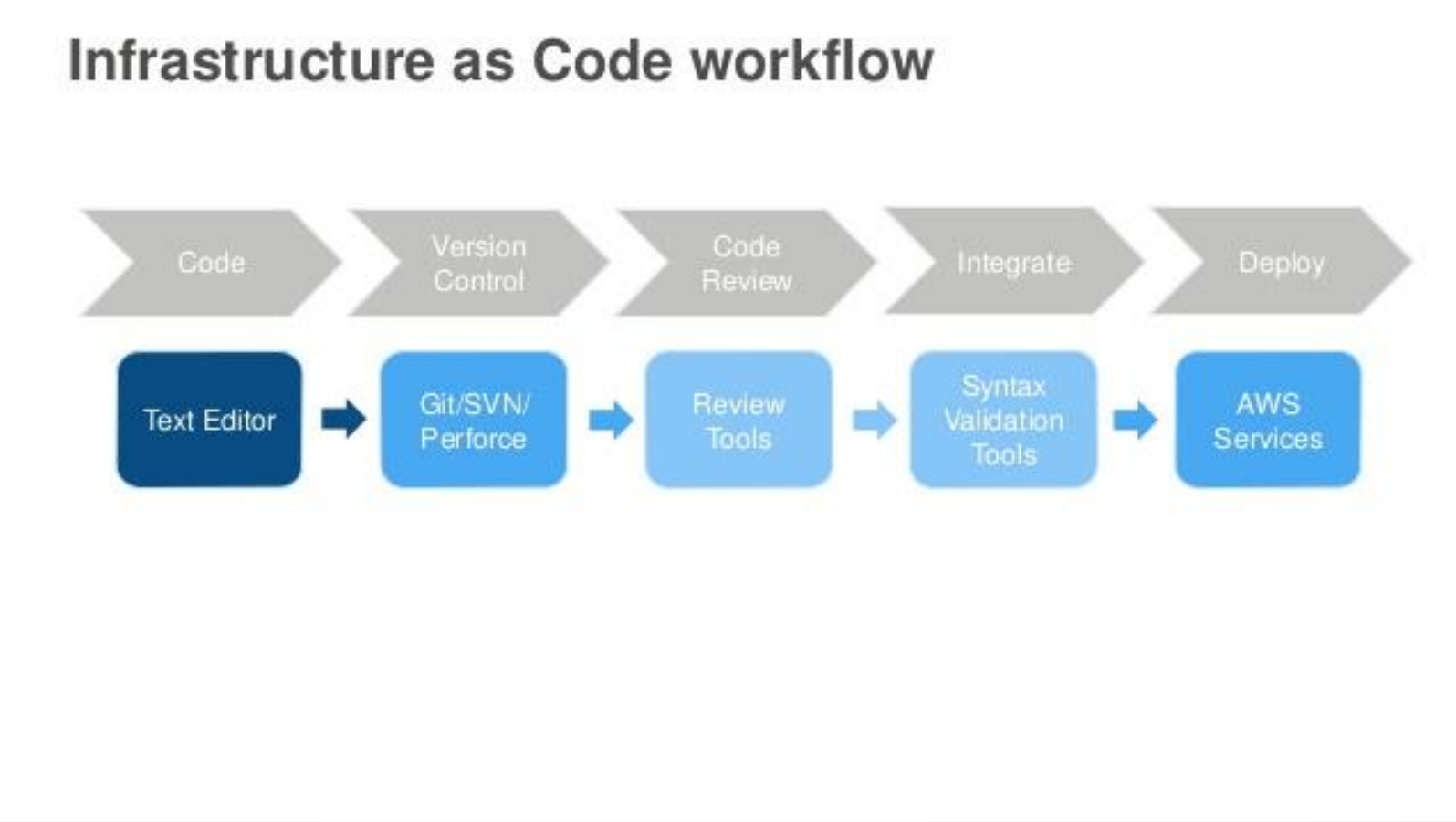
- Il s'agisse d'approvisionner un système physique ou un cloud privé, hybride ou public: création de machines virtuelles, réseaux, stockage, et conteneurs … software defined data centers

IaC: Infrastructure as a Code

- Infrastructure as code is the approach to **defining computing and network infrastructure** through **source code** that can then be treated just like any software system. Such code can be kept in source control to allow auditability and Reproducible Builds, subject to testing practices, and the full discipline of Continuous Delivery. It's an approach that's been used over the last decade to deal with growing Cloud Computing platforms and will become the dominant way to handle computing infrastructure in the next.

Infrastructure as Code: Managing Servers in the Cloud, by Kief Morris

IaC workflow



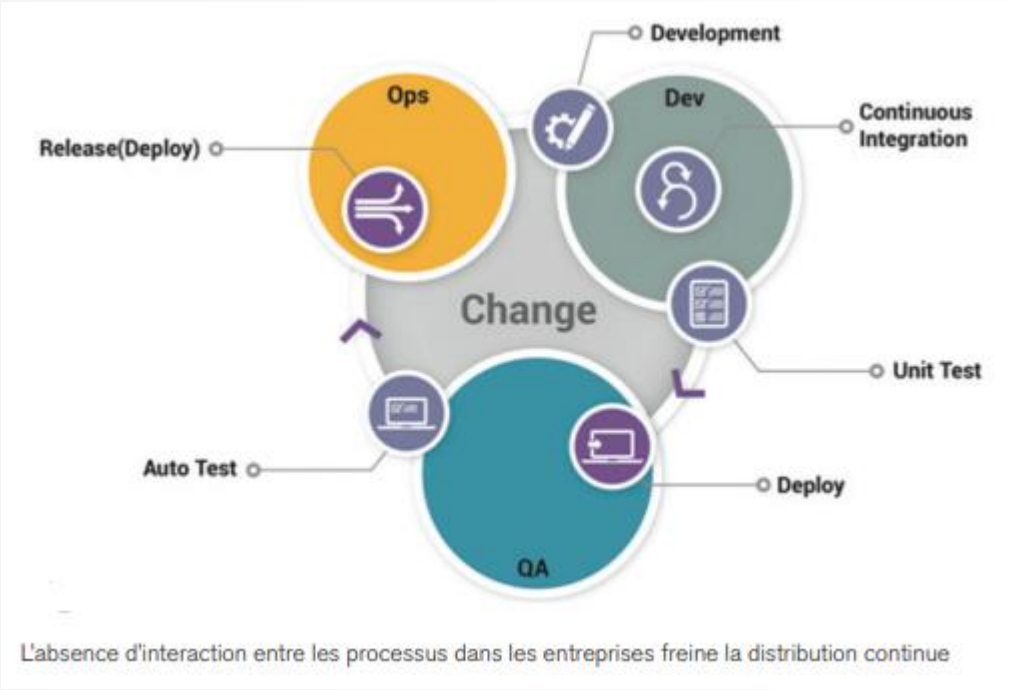
Avantages

- Standardisation
- Gestion des changement et contrôle plus simple des versions
- Réduction du temps et de la productivité
- Amélioration de la fiabilité
- Amélioration des processus CI/CD

Déploiement d’applications: livraison continue

- La livraison continue est une pratique de distribution de logiciels qui nécessite de créer et déployer des logiciels qui peuvent être mis en production à tout moment. Pour satisfaire cette condition , il faut:
  - Une intégration de manière continue les modifications logicielles apportées lors du développement, de la conception, des tests et du déploiement des exécutables, et être prêt à les mettre en production lorsque l’activité le nécessite.
- Le pipeline de déploiement constitue le cœur de la distribution continue. Il s’agit de la mise en œuvre automatique de votre processus de conception, de déploiement, de test et de distribution des applications. Le pipeline de déploiement est instancié dès lors qu’une application est modifiée.

Déploiement d’applications: livraison continue

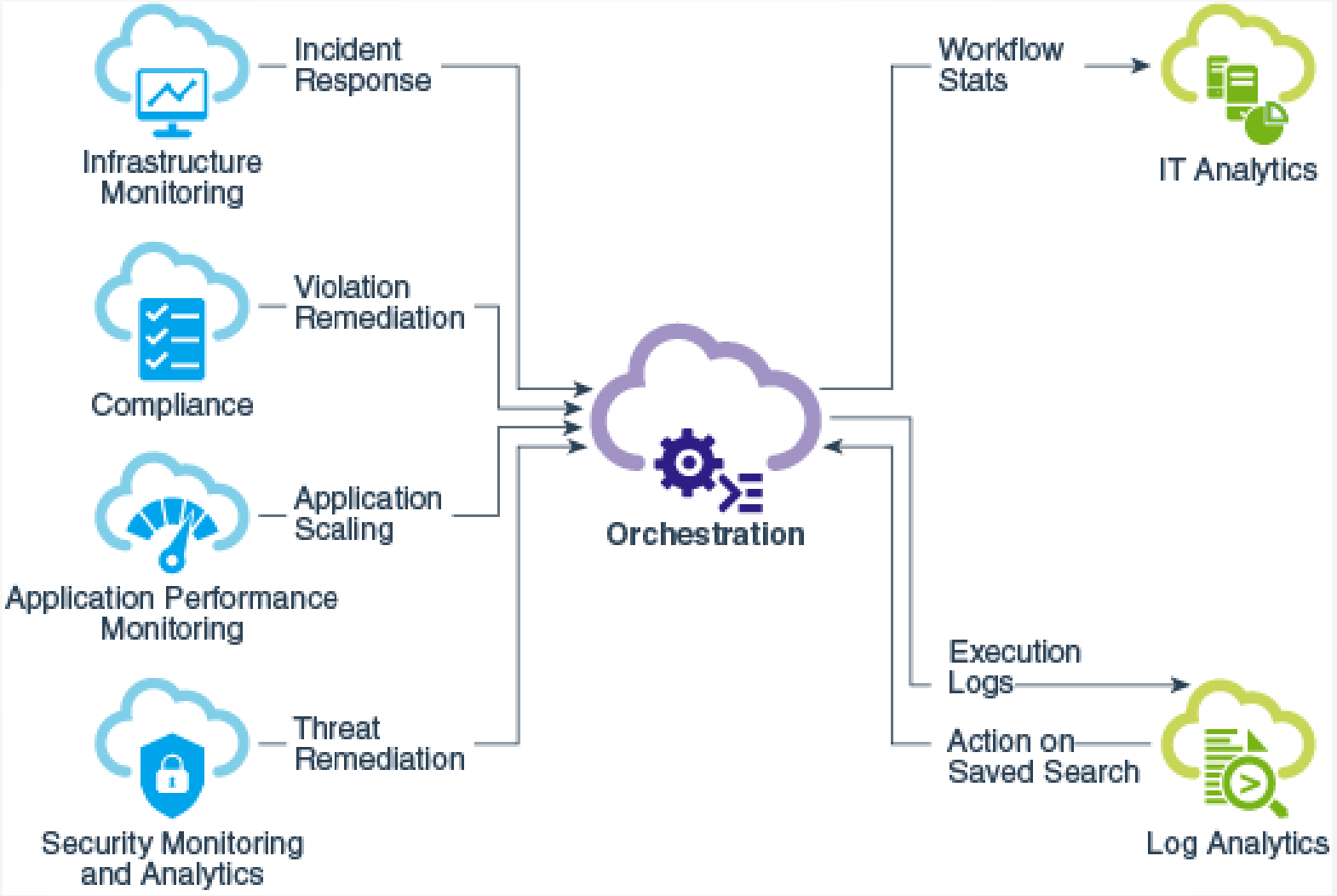


L'absence d'interaction entre les processus dans les entreprises freine la distribution continue

Orchestration

- L'orchestration permet de **coordonner et gérer l'exécution des flux de travail (**Workflows) et des **processus** à travers de multiples **outils, matériels, processus et services** pour fournir un service défini.
- Permet entre autre de :
  - **Coordonner les différents flux de travail et réduire** de manière très significative les **délais de traitement**,
  - **Fiabiliser l'ensemble des processus** et des opérations à forte occurrence et à faible valeur ajoutée,
  - **Superviser les ressources** et **informer l'utilisateur sur sa consommation**, sur le taux de **disponibilité** du service, etc.
  - **Fournir un service à la demande en libre-service** en tenant compte d'un « capacity planning »,
  - Gérer le **cycle de vie des ressources** (création, suspension, fin d'un engagement).

Orchestration



Sécurité et conformité

- Pour faire face aux problèmes de sécurité et de conformité, l'automatisation des processus informatiques et réseau basés sur les données, dans l'ensemble de l'environnement peut constituer une solution. . Les éléments à automatiser doivent inclure:
  - les systèmes d'exploitation :
    - gestion des paquets
    - gestion des correctifs
    - renforcement du système d'exploitation selon des normes de sécurité au moment de l'approvisionnement afin d'assurer la cohérence et l'immuabilité du système d'exploitation.

Sécurité et conformité

- l'infrastructure et la sécurité « en tant que code » : répétabilité, capacité de partage et de vérification, et aide pour réussir les audits de sécurité et de conformité
- la sécurité et la surveillance en continu avec les opérations de suivi de la sécurité :
    - gestion des correctifs
    - identification et gestion des vulnérabilités (bilans de santé, etc.)
    - gouvernance proactive des politiques de sécurité, de contrôle et de conformité
    - correction : génération et automatisation de correctifs

Plusieurs outils

- Ansible
- Puppet
- SaltSatck
- Chef
- Aws OpsWorks
- CFEngine
- Powershell DSC
- Terraform
- Cloud Formation
- ….

Critères de comparaison ?

Activité 1

- Etape 1: Identifier les critères de comparaison entre les différentes solutions d’automatisation
- Etape 2: Comparaison entre ces différentes solution selon les critères précédemment déduit