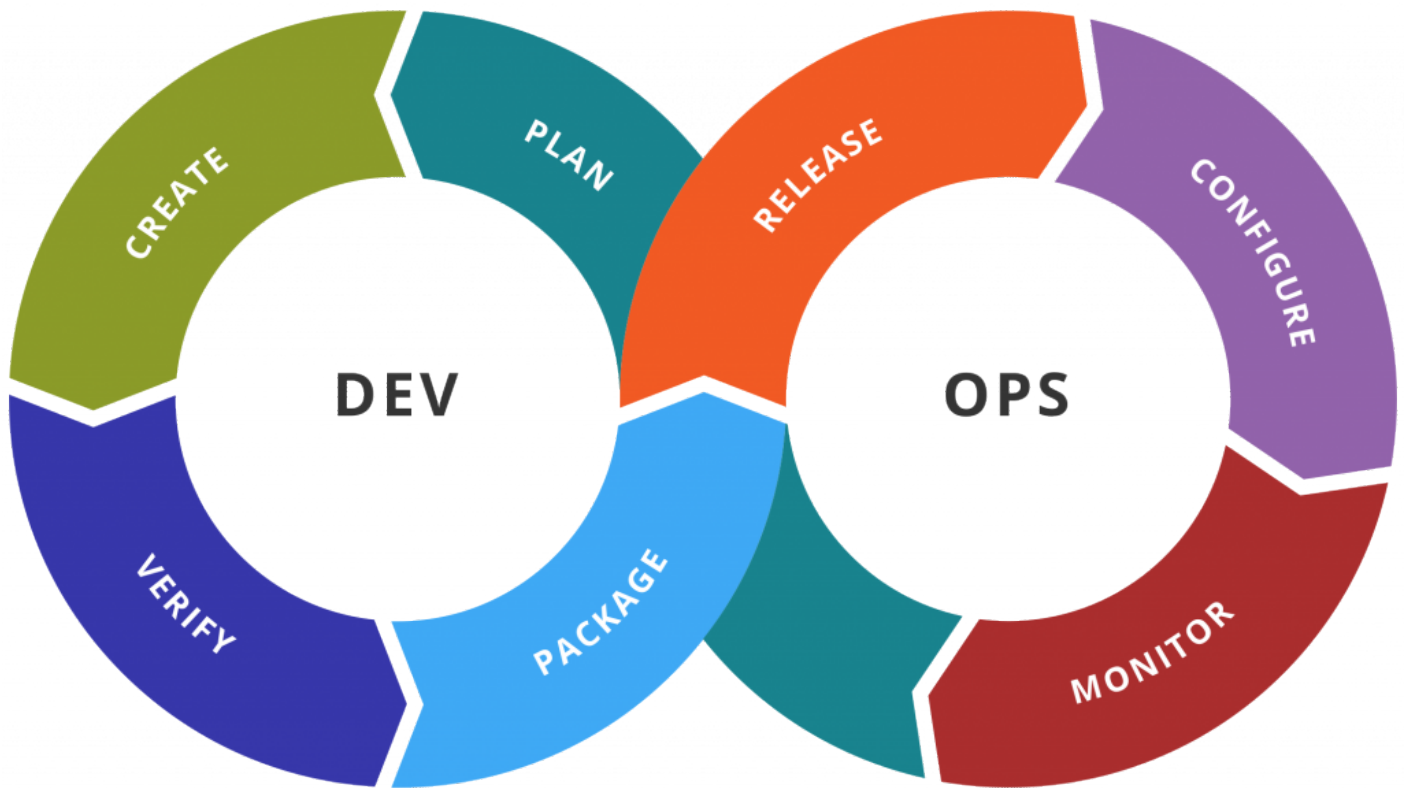


# Glossaire DevOps



## Agile

Un [framework agile](#) est une structure méthodologique qui fournit des lignes directrices, des pratiques, et des processus pour aider les équipes à organiser, planifier, exécuter et suivre leur travail de manière itérative et collaborative. Ce développement itératif découpe un projet en sous-projets pour lesquels, on se fixe des objectifs à court terme atteignables. Ces sous-projets sont appelés des `sprints` et sont de courtes durées.

Le but d'un framework agile est de livrer rapidement au client une première version du produit qui correspond à ses besoins, qui évoluera ensuite au fil des versions.

**Framework agiles** : Scrum, eXtrem Programming, ...

## Air Gap

“Air Gap” (littéralement « coupure d’air ») est une mesure de sécurité dans laquelle un ordinateur ou un réseau est isolé physiquement de tout autre réseau, y compris Internet. Cette isolation est souvent réalisée en s’assurant qu’aucune connexion physique ou sans fil ne relie l’ordinateur ou le réseau en question à des systèmes externes. L’idée est de protéger les données et les systèmes contre les accès non autorisés, les cyberattaques et les logiciels malveillants en éliminant toute voie d’accès directe.

# Artefacts

Les artefacts sont des éléments générés lors du processus d’intégration continue à partir d’un code source. Il peut être composé de bibliothèques, de fichiers de configuration, d’exécutables, de fichiers de documentation, etc.

# Adresse IP

Les adresses IP sont des identifiants numériques attribués aux dispositifs connectés à un réseau, leur permettant de communiquer entre eux sur Internet ou au sein d’un réseau local.

# API

Une API, ou Interface de Programmation d’Applications (Application Programming Interface), est un ensemble de règles et de protocoles pour la construction et l’interaction de logiciels. Elle permet à différents programmes ou composants logiciels de communiquer entre eux. Une API définit la manière dont les fonctions d’un programme peuvent être utilisées par d’autres programmes, facilitant ainsi leur intégration et leur interaction sans avoir besoin de connaître les détails de leur mise en œuvre interne. Cela rend les processus de développement plus rapides et plus flexibles, permettant aux développeurs de créer des applications complexes et interconnectées.

# Automatisation

Fonctionnement d’un système sans intervention humaine, automatisation d’une suite d’opération. Elle peut être faite via l’utilisation de scripts et/ou d’outils qui vont permettre de réduire la charge de travail manuelle et accélérer les tâches répétitives. Autre avantage, elle limite la part des incidents dus à des erreurs humaines.

# Build

Le build est la phase appelée aussi intégration. Elle consiste à assembler le code des développeurs avec toutes ses dépendances afin de fournir un livrable de l'application. Dans cette phase, on retrouve, l'analyse du code (lint), les tests unitaires et la compilation.

# CaaS

CaaS, ou Container as a Service, est un modèle de service cloud qui permet aux utilisateurs de télécharger, organiser, exécuter, échelonner, gérer et arrêter des conteneurs en utilisant une plateforme d'hébergement basée sur le cloud.

# Campagne de patches

Une campagne de patch est un processus visant à appliquer des correctifs (patches) améliorant la sécurité et la stabilité à un ensemble de systèmes, de logiciels ou d'infrastructures informatiques de manière coordonnée. Ces correctifs sont conçus pour résoudre des failles de sécurité, des bogues ou pour améliorer les performances d'un logiciel ou d'un système.

# CI/CD

CI/CD (Continuous Integration / Continuous Deployment) : Pratiques visant à automatiser le processus d'intégration, de test et de déploiement pour livrer rapidement et régulièrement des mises à jour logicielles.

# CLI

Une CLI, ou Command Line Interface (Interface en Ligne de Commande), est une interface utilisateur qui permet aux utilisateurs d'interagir avec un logiciel ou un système d'exploitation en tapant des commandes textuelles dans un terminal ou une console.

# Cloud

Le cloud fait référence à la fourniture de services informatiques tels que le stockage, le calcul, les bases de données, la mise en réseau, l'analyse, etc., via Internet. Plutôt que de posséder et de gérer physiquement des serveurs et des infrastructures informatiques, les entreprises et les individus peuvent accéder à ces ressources à partir de serveurs distants exploités par des fournisseurs de **cloud publics**.

## Cloud Hybride

Un cloud hybride est un environnement informatique qui combine à la fois des ressources de **cloud public** et de **cloud privé**, permettant aux données et aux applications de s'exécuter de manière transparente entre les deux. Concrètement, cela signifie que les entreprises peuvent déployer des applications et des données à la fois sur des serveurs et des infrastructures internes (cloud privé) et sur des ressources de cloud public, le tout dans un environnement cohérent.

## Cluster

Un **cluster** est un groupe de machines (nœuds) interconnectées qui travaillent ensemble pour exécuter des applications conteneurisées. Dans le cadre de Kubernetes, un cluster est constitué d'un plan de contrôle (control plane) et de plusieurs nœuds de travail. Les clusters offrent une **haute disponibilité** et une **scalabilité** en répartissant les charges de travail sur plusieurs machines.

## CMP

Une CMP (Cloud Management Platform) désigne une plateforme logicielle conçue pour gérer et orchestrer les ressources, services et opérations dans un environnement cloud.

## Conteneurs / Conteneurisation

Un conteneur est une unité d'exécution autonome qui regroupe une application, ses dépendances logicielles et toutes les ressources nécessaires à son exécution, y compris des bibliothèques, des fichiers de configuration et des variables d'environnement. Les conteneurs sont conçus pour isoler une application et son environnement de manière à ce qu'elle puisse être déployée et exécutée de manière cohérente. Cette technologie permet de porter une application sur n'importe quel serveur, local ou dans le cloud, à la seule condition de posséder la même architecture et la même famille de système d'exploitation.

# CRD (Custom Resource Definition)

Une **CRD** permet de définir des ressources personnalisées dans Kubernetes. Cela permet aux utilisateurs d'étendre les fonctionnalités de Kubernetes en ajoutant de nouveaux types d'objets spécifiques à leurs besoins.

## Déclaratif vs Impératif

Dans une approche **déclarative**, le code de configuration décrit l'état dans lequel l'environnement de production doit se trouver, sans prendre en compte les étapes pour y arriver. Le système fait en sorte que l'infrastructure atteigne l'état désiré.

Dans une approche **impérative**, le fichier de configuration définit les commandes et les étapes nécessaires pour que l'environnement parvienne à l'état désiré. Le système réalise les commandes indiquées. Exemple : Ansible

## Déploiement Blue-Green

Une méthode de déploiement qui permet de basculer en douceur entre deux environnements (bleu et vert) pour minimiser les interruptions de service.

## Déploiement Continu

Déploiement ou livraison continue est une méthode de développement de logiciels dans le cadre de laquelle les modifications de code sont automatiquement préparées en vue de leur publication dans un environnement de production. La livraison continue étend le principe de l'intégration continue en déployant tous les changements de code dans un environnement de test et/ou de production après l'étape de création. Lorsque la livraison continue est correctement implémentée, les développeurs disposent en permanence d'un artefact de génération prêt pour le déploiement qui a été soumis avec succès à un processus de test standardisé.

## Dettes Techniques

La dette technique est une métaphore utilisée pour décrire le coût caché associé au retard ou à la négligence des bonnes pratiques de développement, de maintenance et de gestion des systèmes

informatiques. Cela se produit lorsque des raccourcis, des compromis ou des décisions techniques inadéquates sont pris pour accélérer le développement d'un logiciel ou pour répondre à des échéances serrées. La dette technique est le reflet des conséquences négatives de ces choix à long terme.

# DevSecOps

Intégration de la sécurité dans les pipelines CI/CD dès le début du développement.

# Edge Computing

Le **Edge Computing** est une architecture informatique décentralisée qui rapproche les ressources de calcul et de traitement des données des périphériques ou des utilisateurs finaux. Cette approche est couramment utilisée pour réduire la latence et la consommation de bande passante dans des environnements distribués comme l'IoT ou les systèmes connectés.

# Élasticité

L'élasticité est la capacité d'un système informatique d'ajuster dynamiquement les ressources (comme la puissance de calcul, la mémoire, le stockage, etc.) en fonction de la demande des utilisateurs.

# Élévation de privilèges

L'élévation de privilèges (ou privilege escalation en anglais) fait référence au processus par lequel un utilisateur ou un processus tente d'acquérir des privilèges ou des autorisations supplémentaires au sein d'un système informatique, lui permettant ainsi d'exécuter des actions pour lesquelles il n'avait pas initialement les droits.

L'élévation de privilèges est également un sujet important dans le domaine de la sécurité informatique, car les vulnérabilités qui permettent une élévation de privilèges peuvent être exploitées pour compromettre la sécurité d'un système. Les correctifs de sécurité sont souvent publiés pour corriger de telles vulnérabilités et empêcher les attaquants d'acquérir des privilèges non autorisés.

# Gestion des configurations

La gestion de configuration est un concept fondamental dans l'administration des systèmes et des infrastructures informatiques. Elle englobe l'ensemble des processus et des outils utilisés pour définir, mettre en place, surveiller et maintenir les configurations, qu'il s'agisse de serveurs, de postes de travail, de logiciels ou d'équipements réseaux.

# Gestionnaire de Contrôle de version

Un gestionnaire de contrôle de version, également appelé logiciel de gestion de version ou système de contrôle de version (VCS), est un outil informatique qui permet de suivre et de gérer les modifications apportées à un ensemble de fichiers au fil du temps. Son principal objectif est de faciliter la collaboration entre plusieurs personnes travaillant sur un même projet informatique, que ce soit un logiciel, un site web, une documentation. Le plus utilisé actuellement s'appelle Git.

# Gestion des secrets

La gestion des secrets désigne la pratique permettant de gérer de manière sécurisée les informations sensibles, telles que les mots de passe, les clés d'authentification, les jetons d'accès, les clés de chiffrement et d'autres données confidentielles, utilisées dans les applications, les systèmes et les infrastructures informatiques.

Cette pratique est essentielle pour garantir la sécurité des systèmes d'informations, car toute faille dans la protection de ces données pourrait entraîner des violations de sécurité, des accès non autorisés, des pertes de données...

# Haute Disponibilité

La haute disponibilité (HA) est un concept en informatique qui se réfère à la capacité d'un système, d'une application ou d'un service à rester opérationnel et accessible pendant une période de temps prolongée, généralement 24 heures sur 24 et 7 jours sur 7. L'objectif de la haute disponibilité est de minimiser les temps d'arrêt non planifiés, d'assurer la continuité des opérations et de garantir que les utilisateurs finaux puissent accéder aux ressources informatiques sans interruption notable.

Pour atteindre un haut niveau de disponibilité, les organisations mettent en place des architectures redondantes, des clusters, des systèmes de basculement et des tests rigoureux de continuité des opérations. Cela garantit que les systèmes restent opérationnels même en cas de défaillance matérielle, de coupures d'électricité, de catastrophes naturelles ou de cyberattaques.

# Helm

**Helm** est un gestionnaire de packages pour Kubernetes. Il permet de simplifier l'installation, la gestion et la mise à jour des applications Kubernetes en utilisant des **charts** (modèles de déploiement). Helm est particulièrement utile pour déployer des applications complexes avec des dépendances multiples.

# IAAS

IaaS est l'acronyme de "Infrastructure as a Service", qui se traduit en français par "Infrastructure en tant que Service". Il s'agit d'un modèle de cloud qui aux utilisateurs de provisionner des ressources d'infrastructures informatiques sous forme de services.

# Idempotence

En algèbre, l'**idempotence** est la propriété d'une opération, d'avoir le même effet qu'on l'applique une ou plusieurs fois. Par exemple, la valeur absolue est idempotente :  $\text{abs}(\text{abs}(-5)) = \text{abs}(-5)$ , les deux membres étant égaux à 5.

En programmation une fonction est idempotente si l'état du système reste le même après un ou plusieurs appels.

# Infrastructure as Code

L'infrastructure en tant que code (Infrastructure as Code) est une infrastructure informatique virtuelle qui est créée, configurée et approvisionnée automatiquement via du code informatique.

# Intégration Continue

CI ou **intégration continue** est une méthode de développement avec laquelle les développeurs intègrent régulièrement leurs modifications de code à un référentiel centralisé, suite à quoi des opérations de création et de tests sont automatiquement menés. L'intégration continue désigne souvent l'étape de création ou d'intégration du processus de publication de logiciel et implique un aspect automatisé. Les principaux objectifs de l'intégration continue sont de trouver et de corriger plus rapidement les bogues, d'améliorer la qualité des logiciels et de réduire le temps nécessaire pour valider et publier de nouvelles mises à jour de logiciels.

## IPAM

IPAM, acronyme de "IP Address Management" (Gestion des adresses IP en français), est un ensemble de pratiques, de techniques et d'outils utilisés pour planifier, gérer et suivre les adresses IP (Internet Protocol) au sein d'un réseau informatique.

## Load balancing ou équilibrage de charge

Un équilibrage de charge, en anglais "load balancer", est un composant matériel ou logiciel utilisé dans les environnements informatiques pour distribuer le trafic réseau entrant de manière équilibrée entre plusieurs serveurs ou ressources, afin d'optimiser les performances, la fiabilité et la disponibilité des systèmes.

## Microservices vs Monolithe

Une architecture microservices est une architecture logicielle basée sur de petits services autonomes qui peuvent être développés, déployés et évolués indépendamment. Cette architecture s'appuie sur des orchestrateurs de conteneurs.

Un monolithe est une architecture logicielle où l'ensemble de l'application est conçu et construit comme une seule entité monolithique. Cela signifie que toutes les fonctionnalités, les composants et les parties de l'application sont regroupés en un seul code source, une seule base de données et un seul exécutable.

## Mutabilité vs Immutabilité

Immutabilité : *Caractère de ce qui ne peut changer.*

Une **infrastructure mutable** est un système qui évolue de **manière incrémentale** et donc mise à jour de manière continue. L'état de l'infrastructure est le résultat d'une **accumulation d'opérations** pouvant être pour certaines manuelles.

Par contre, une **infrastructure immuable** est un système qui n'est jamais modifié après son déploiement. Ce système devra en conséquence avoir le moins de mutation possible, voire aucune.

Par exemple, pour construire des images de VM immuables, vous pouvez utiliser **Packer** d'**HashiCorp** pour ensuite les enregistrer en les versionnant dans votre gestionnaire d'Artéfacts.

## NOC

Un NOC (Network Operations Center) est un centre opérationnel chargé de surveiller, gérer et maintenir un réseau informatique ou de télécommunications. Les équipes du NOC supervisent les performances du réseau, détectent et résolvent les problèmes et effectuent la maintenance préventive pour assurer le bon fonctionnement continu du réseau.

## Observabilité

L'**observabilité** est la capacité de surveiller, d'analyser et de comprendre l'état interne d'un système informatique en temps réel. Elle repose sur trois piliers principaux : les logs, les métriques et les traces. Des outils comme **Prometheus** et **Grafana** sont souvent utilisés pour l'observabilité.

## Orchestration

L'orchestration est un processus automatique de gestion d'un système informatique.

## RPO

Recovery Point Objective (RPO) fait généralement référence à la quantité de données qui peuvent être perdues au cours de la période la plus opportune pour une entreprise, avant qu'un préjudice important ne se produise, à partir d'un événement critique jusqu'à la sauvegarde la plus précédente.

# RTO

Recovery Time Objective (RTO) fait référence au temps pendant lequel une application, un système et/ou un processus peut être en panne sans causer de dommages importants à l'entreprise, ainsi qu'au temps passé à restaurer l'application et ses données.

# PAAS

PaaS, ou "Platform as a Service" en anglais, se traduit en français par "Plateforme en tant que Service". C'est un modèle de cloud qui fournit une plateforme de développement et d'exécution d'applications sur le cloud. Ces services vont permettre aux développeurs de créer, déployer et gérer des applications sans se soucier de la gestion de l'infrastructure sous-jacente.

# Provisioning ou Provisionnement

Le provisioning est un processus d'allocation automatique de ressources informatiques.

# SAAS

SaaS, ou "Software as a Service" en anglais, se traduit en français "logiciel en tant que service". C'est un modèle de distribution de logiciels dans lequel les applications sont hébergées et accessibles en ligne par le biais d'un fournisseur de services cloud. Au lieu d'installer et de gérer des logiciels localement sur un ordinateur ou un serveur, les utilisateurs peuvent accéder à ces applications via Internet, généralement à partir d'un navigateur web.

# Scrum

Un framework de gestion de projet Agile.

# Serverless

Le **Serverless** est un modèle de déploiement dans lequel les développeurs n'ont pas à gérer l'infrastructure sous-jacente. Les fournisseurs cloud allouent dynamiquement les ressources nécessaires à l'exécution d'une application. Les exemples incluent **AWS Lambda**, **Google Cloud Functions**, et **Azure Functions**.

# Service Mesh

Un **Service Mesh** est une infrastructure dédiée à la gestion des communications entre microservices. Il fournit des fonctionnalités comme le **load balancing**, la gestion des échecs, le monitoring, et la sécurisation des connexions via des proxies. Les outils populaires incluent **Istio** et **Linkerd**.

# SLA/SLO/SLI

- **SLA (Service Level Agreement)** : Contrat définissant les attentes en termes de niveaux de service.
- **SLO (Service Level Objective)** : Objectifs précis liés aux niveaux de performance d'un service (ex. : disponibilité à 99,9 %).
- **SLI (Service Level Indicator)** : Indicateur qui mesure un aspect de la performance, comme le temps de réponse ou le taux d'erreur.

# SOC

Un SOC (Security Operations Center) est similaire, mais se concentre spécifiquement sur la sécurité informatique. Les équipes d'un SOC surveillent les menaces potentielles, détectent les activités suspectes, répondent aux incidents de sécurité et mettent en œuvre des mesures de défense pour protéger les systèmes informatiques et les données contre les cyberattaques.

# Supervision

La supervision est un processus de surveillance et de gestion des systèmes informatiques, des réseaux, des serveurs, des applications et d'autres composants de l'infrastructure informatique. Elle vise à garantir que ces systèmes fonctionnent de manière fiable, sécurisée et efficace. La supervision informatique peut être effectuée à l'aide de logiciels spécialisés qui collectent des données en temps réel sur les performances et l'état des équipements informatiques.

# Virtualisation

La virtualisation est une technologie qui permet de créer des versions virtuelles de tous éléments d'une infrastructure informatique comme des serveurs, des équipements réseaux ou d'autres entités informatiques. Elle vise à isoler et à abstraire ces éléments pour les rendre indépendants des ressources physiques sous-jacentes, ce qui permet une utilisation plus efficace et flexible de ces ressources.

Elle est couramment utilisée dans des domaines tels que la virtualisation de serveur, la virtualisation de stockage, la virtualisation de réseau, la virtualisation de bureau et plus encore.

# Vulnérabilité

Une vulnérabilité est une faiblesse du code qui introduit ce qu'on appelle une faille de sécurité. Ces failles peuvent être présentes dans un système d'exploitation, un logiciel, un équipement de réseau ou un processus. Elles peuvent être exploitées par des acteurs malveillants pour causer un dommage, un accès non autorisé, des pertes de données, ou d'autres problèmes de sécurité.

Pour détecter la présence des vulnérabilités, on fait appel à des outils d'analyse de code et de détection.

---

Created 2025-01-31 23:23:32 UTC by Nicolas

Updated 2025-02-01 00:10:39 UTC by Nicolas