SMB111 - Systèmes et applications répartis pour le cloud

- **François Lacomme <francois.lacomme@2isa.net>**

- Document provisoire.
  Copie et diffusion non autorisées sans accord écrit.
  Documents liés aux cours et TP : smb111.seancetenante.com

Systèmes et applications répartis pour le cloud

SMB111
Plan du cours

1. La virtualisation système
   - Définitions ; Hyperviseur ; Machine Virtuelle
   - Avantages / Inconvénients
   - Types de virtualisation
   - Logiciels de virtualisation
2. Conteneurs
   - Définitions ; Comparaison avec les machines virtuelles
   - Orchestration des conteneurs
   - Fournisseurs de conteneurs : Docker, RKT, LXC
   - Précisions sur Docker : Installation, utilisation
   - Kubernetes : Les bases, architecture d'un cluster Kubernetes, TP.

Systèmes et applications répartis pour le cloud

SMB111
Plan du cours

4. La virtualisation réseau
   - Caractéristiques principales
   - Types de virtualisation réseau
   - Applications
5. La virtualisation de stockage
   - Fonctionnement
   - Types de virtualisation de stockage
   - Exemples de fournisseurs

Systèmes et applications répartis pour le cloud

SMB111
Plan du cours

6. Cloud Computing
   - Rappels (déjà introduit avec RSX102)
   - Cloud privé, public et hybride
   - Modèles de service : IaaS, PaaS, SaaS, ServerLess, etc.
   - Exemples de fournisseurs
7. Big Data ; Hadoop (déjà introduit avec RSX102)
   - Entrepôts et lacs de données
   - Fonctionnement de Hadoop

La virtualisation système

Chapitre 1

1 - Définitions

- La virtualisation ; définitions
  - La virtualisation de l'infrastructure informatique consiste à créer des versions virtuelles des composants matériels traditionnels, tels que les serveurs, les serveurs de stockage et les réseaux.
  - Au lieu de s'appuyer sur des ressources physiques dédiées, la virtualisation permet de tirer parti de l'abstraction des ressources, ce qui signifie que ces ressources sont définies et gérées par des logiciels au lieu d'être liées à un matériel physique spécifique.
  - Cela consiste à exécuter sur une machine hôte, dans un environnement isolé,
    - des systèmes d'exploitation en cas de virtualisation système
    - des applications — on parle alors de virtualisation applicative.
  - Ces ordinateurs virtuels sont appelés serveur privé virtuel (*Virtual Private Server* ou VPS) ou encore environnement virtuel (*Virtual Environment* ou VE).

La virtualisation système

Chapitre 1

1 - Définitions

- Qu’est-ce qu’un hyperviseur ?
  - Un hyperviseur est le composant central de la virtualisation.
  - Il divise un serveur physique en plusieurs machines virtuelles, chacune fonctionnant de manière isolée.

- Types d’hyperviseurs :
  - Hyperviseur de type 1 (natif ou bare-metal) : il s’exécute directement sur le matériel sans nécessiter de système d’exploitation hôte.
    - Meilleures performances
    - Isolation plus forte.
    - Exemples d’hyperviseurs de type 1 : VMware vSphere/ESXi, Proxmox et KVM (*Kernel-based Virtual Machine*).
  - Hyperviseur de type 2 (hosted) : Il s’exécute sur un système d’exploitation hôte.
    - Plus simples à configurer,
    - moindre performance.
    - Exemples : VirtualBox et VMware Workstation.

La virtualisation système

Chapitre 1

1 - Définitions

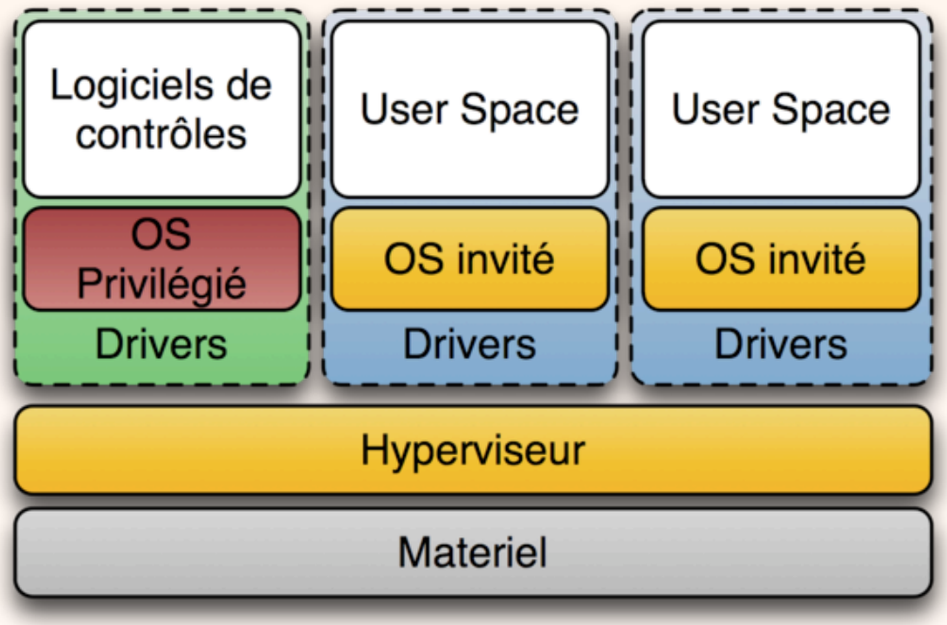


Fig 1.1 - Hyperviseur de type 1

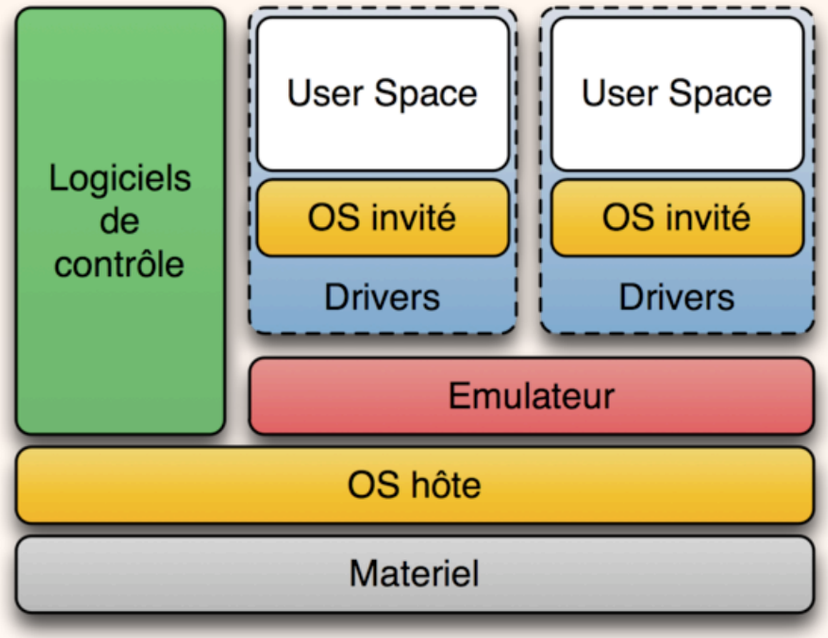


Fig 1.2 - Hyperviseur de type 2

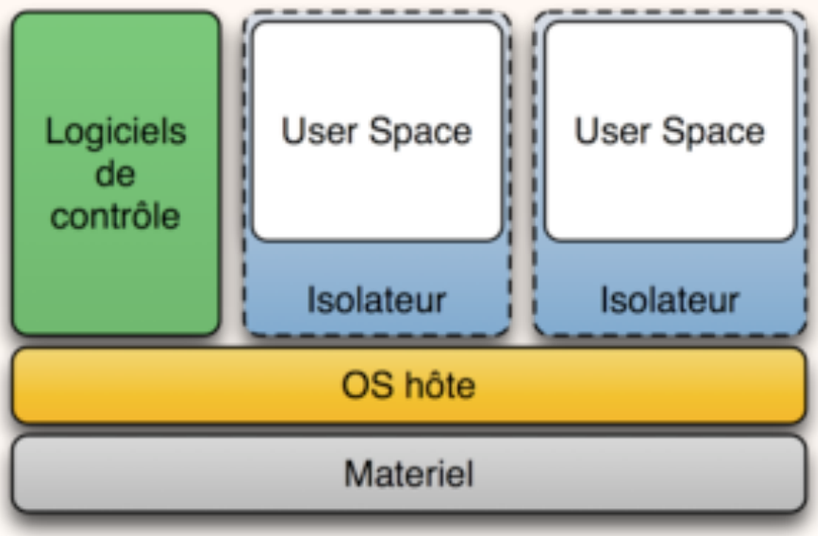


Fig 2.1 - Conteneurs

La virtualisation système

Chapitre 1

1 - Définitions

- Machine Virtuelle (VM)
  - Les machines virtuelles sont des instances logicielles complètes d'ordinateurs physiques.
  - Chaque VM fonctionne :
    - sur un hyperviseur (un logiciel de virtualisation)
    - dispose de son propre système d'exploitation et de ses applications.
    - Cela permet d'exécuter plusieurs systèmes d'exploitation et charges de travail sur un seul serveur physique.
- Conteneur
  - La virtualisation au niveau du système d'exploitation est un paradigme de système d'exploitation (SE) dans lequel le noyau permet l'existence de plusieurs instances d'espace utilisateur isolées, appelées **conteneurs**.
  - Exemples : LXC, Solaris containers, Docker, Podman, OpenVZ

2 - Avantages / Inconvénients

- **Avantages** de la virtualisation système
  - Optimisation des ressources :
    - La virtualisation permet une meilleure utilisation des ressources matérielles en consolidant **plusieurs systèmes virtuels sur un seul serveur physique.**
  - Réduction des coûts :
    - Elle diminue les dépenses liées à l'infrastructure, notamment en termes d'achat de matériel, d'espace de stockage et de consommation énergétique.
  - Flexibilité et évolutivité :
    - Il est facile de créer, modifier ou supprimer rapidement des environnements virtuels selon les besoins.

2 - Avantages / Inconvénients

- **Avantages** de la virtualisation système, suite…
  - Compatibilité :
    - Les machines virtuelles peuvent exécuter des systèmes d'exploitation et des applications anciens ou obsolètes, même si le système hôte est incompatible.
  - Meilleure gestion :
    - La virtualisation permet une allocation dynamique et personnalisée des ressources pour chaque machine virtuelle.
  - Consolidation des serveurs :
    - Elle améliore l'utilisation des processeurs et la distribution du stockage.

2 - Avantages / Inconvénients

- **Inconvénients** de la virtualisation système
  - Complexité :
    - La mise en œuvre et la gestion de la virtualisation nécessitent des **compétences techniques spécifiques**.
  - Coûts initiaux :
    - L'investissement initial peut être élevé (achat de **serveurs puissants et licences logicielles**).
  - Risque accru en cas de panne :
    - Une défaillance du serveur hôte peut affecter tous les systèmes virtuels hébergés.

- **Inconvénients** de la virtualisation système, suite…
  - Problèmes de sécurité :
    - La compromission d'une machine virtuelle peut potentiellement affecter l'ensemble de l'infrastructure.
  - Performance :
    - Les performances peuvent être affectées par la concurrence entre machines virtuelles sur le même système hôte.
  - Overhead :
    - La réplication de l'environnement matériel et du système d'exploitation pour chaque machine virtuelle génère un surcoût.

3 - Types de virtualisation

- Les différents types de virtualisation :
  - Virtualisation des serveurs :
    - Elle permet d'exécuter plusieurs systèmes d'exploitation virtuels sur un seul serveur physique.
  - Virtualisation des systèmes d'exploitation :
    - Similaire à la virtualisation des serveurs, elle permet d'exécuter plusieurs systèmes d'exploitation isolés (comme Linux et Windows) sur une seule machine.
  - Virtualisation des postes de travail :
    - Elle sépare l'environnement du poste de travail des périphériques qui l'utilisent, permettant le déploiement d'un même environnement sur plusieurs machines.
    - On distingue :
      - Infrastructure de bureau virtuel (VDI, *Virtual desktop infrastructure*)
      - Services de bureau à distance (RDS, *Remote Desktop Services*)
      - Desktop-as-a-Service (DaaS)

3 - Types de virtualisation

- Les différents types de virtualisation, suite…
  - Virtualisation du stockage :
    - Elle regroupe les ressources de stockage de plusieurs périphériques en un seul grand périphérique de stockage virtuel.
    - Elle se compose de deux catégories :
      - La virtualisation des fichiers : Appliqué aux systèmes de stockage en réseau (NAS). À l'aide des protocoles SMB (Server Message Block) ou NFS (Network File System).
      - La virtualisation des blocs : Les systèmes basés sur des blocs font abstraction du stockage logique et sépare celui-ci du stockage physique afin que l'utilisateur / administrateur puisse y accéder sans avoir à accéder au stockage physique, ce qui donne à l'administrateur plus de flexibilité dans la gestion des différents stockage.

3 - Types de virtualisation

- Les différents types de virtualisation, suite…
  - Virtualisation du réseau :
    - Elle reproduit logiciellement un réseau physique complet, combinant les ressources réseau en une seule entité virtuelle.
  - Virtualisation des applications :
    - Elle permet d'exécuter des applications de manière encapsulée, indépendamment du système d'exploitation sous-jacent.

4 - Logiciels de virtualisation

- Hyperviseur de type 1 :
  - Un **hyperviseur de type 1** fonctionne directement sur le matériel physique (*bare-metal*) ou intégré au système d'exploitation hôte (natif).
  - Logiciels :
    - Microsoft Hyper-V Manager, sur Windows 64 bits 10 Entreprise/Pro/Education, gratuit.
      - Base du **Windows Subsystem for Linux** (WSL/WSL2).
    - KVM, *Kernel-based Virtual Machine*, sur Linux 32 et 64 bits, gratuit.
      - Intégrée au noyau Linux et en est la technologie de virtualisation de base.
      - KVM permet d'exploiter plusieurs machines virtuelles sur un hôte Linux.
      - Chaque VM se voit attribuer son propre matériel virtualisé. Cela comprend les cœurs de processeur, la mémoire vive, mais aussi des adaptateurs réseau et graphiques ainsi que la mémoire de masse.

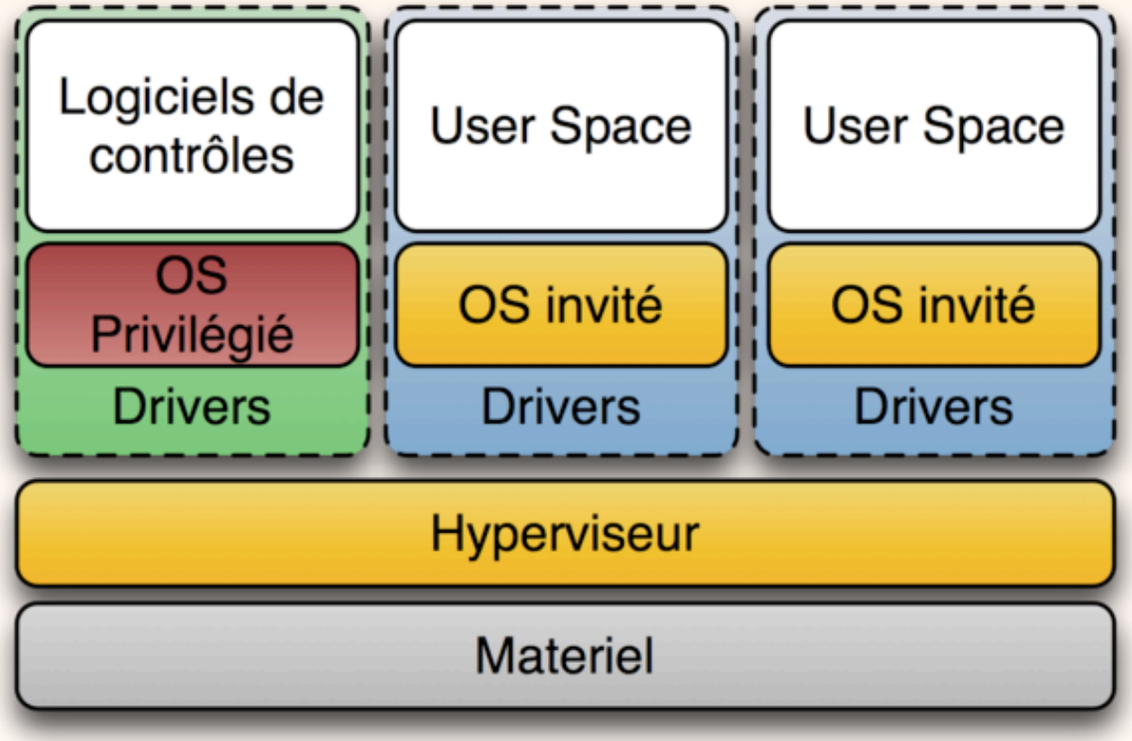


Fig 1.1 - Hyperviseur de type 1

4 - Logiciels de virtualisation

- Hyperviseur de type 2 :
  - Un **hyperviseur de type 2** (hébergé ou *hosted*) fonctionne sur un système d'exploitation hôte.
  - Logiciels :
    - Oracle VirtualBox, sur Windows ou Linux 64 bit, gratuit.
      - Logiciel libre et open source.
      - Simple d'utilisation et convivial.
    - VMware Fusion Pro, sur macOs, gratuit pour usage personnel.
      - Permet d'exécuter sur Mac des VM avec macOS, Windows et Linux comme OS.
      - il comprend des fonctionnalités pour la création, la gestion et l'exécution de conteneurs OCI, *Open Container Initiative,* et de clusters Kubernetes.
    - VMware Workstation Pro, sur Windows ou Linux 64 bit, gratuit pour usage personnel.
      - Permet la virtualisation de la plupart des systèmes d'exploitation x86 sur un poste standard.

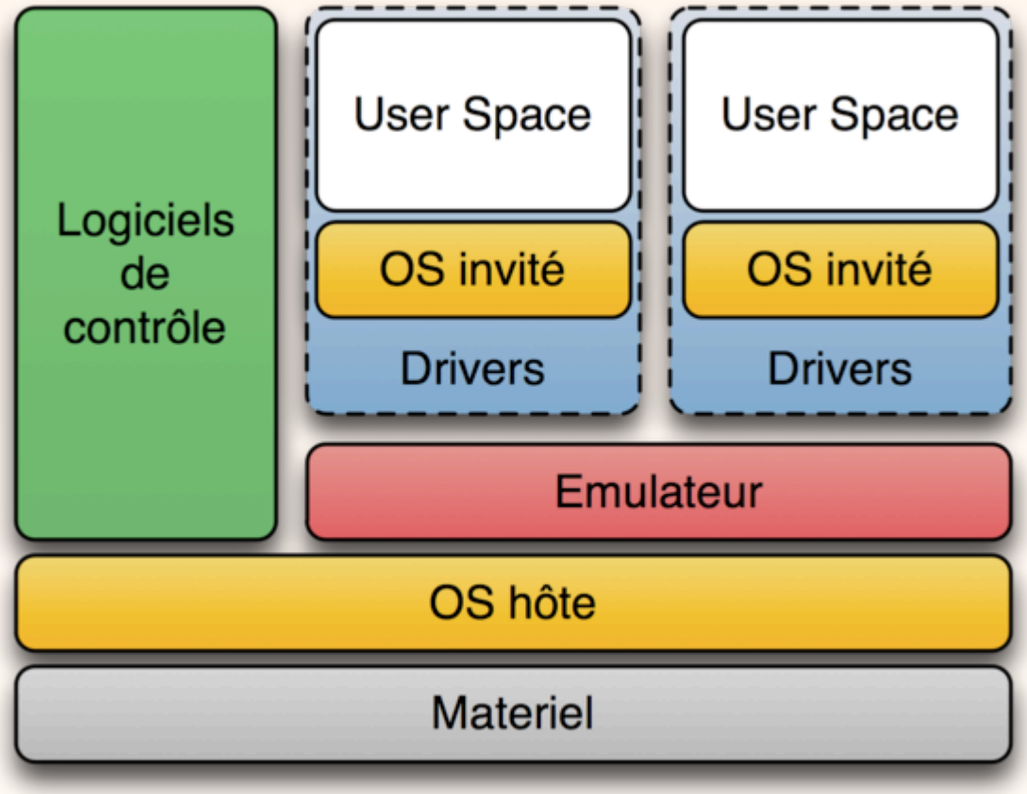


Fig 1.2 - Hyperviseur de type 2

4 - Logiciels de virtualisation

- Hyperviseur de type 2, suite :
  - Logiciels, suite :
    - Parallels Desktop, sur macOs, gratuit pour usage personnel.
      - Environnement de bureau Windows fonctionnant en parallèle avec macOS.
      - Ce logiciel de virtualisation permet de déplacer et de partager des contenus entre Mac et Windows de manière transparente.
      - Grâce au mode Coherence, on peut utiliser des applications Windows comme des applications Mac.
    - QEMU, *Quick Emulator*, sur Linux, gratuit.
      - Logiciel de virtualisation complexe.
      - En plus de la virtualisation complète de matériel x86, QEMU maîtrise l'émulation d'autres architectures de processeur (d'où son nom).
      - Il est possible d'exécuter des fichiers binaires écrits pour des processeurs qui n'existent pas physiquement dans le système. Et même également de traduire en direct des programmes individuels afin de les exécuter.

Conteneurs

Chapitre 2

1 - Définitions ; comparaison avec les machines virtuelles

- Définitions et comparaison avec les machines virtuelles
  - Les **conteneurs** sont une technologie de virtualisation au niveau du système d'exploitation qui permet d'isoler et d'exécuter des applications avec leurs dépendances dans des environnements légers et portables.
  - Contrairement aux machines entièrement virtualisées, les conteneurs ne simulent pas leur matériel.
  - Au lieu de cela, ils partagent le noyau du système d'exploitation de la machine hôte, mais isolent l'application spécifique, ses bibliothèques et les dépendances nécessaires à l'exécution.

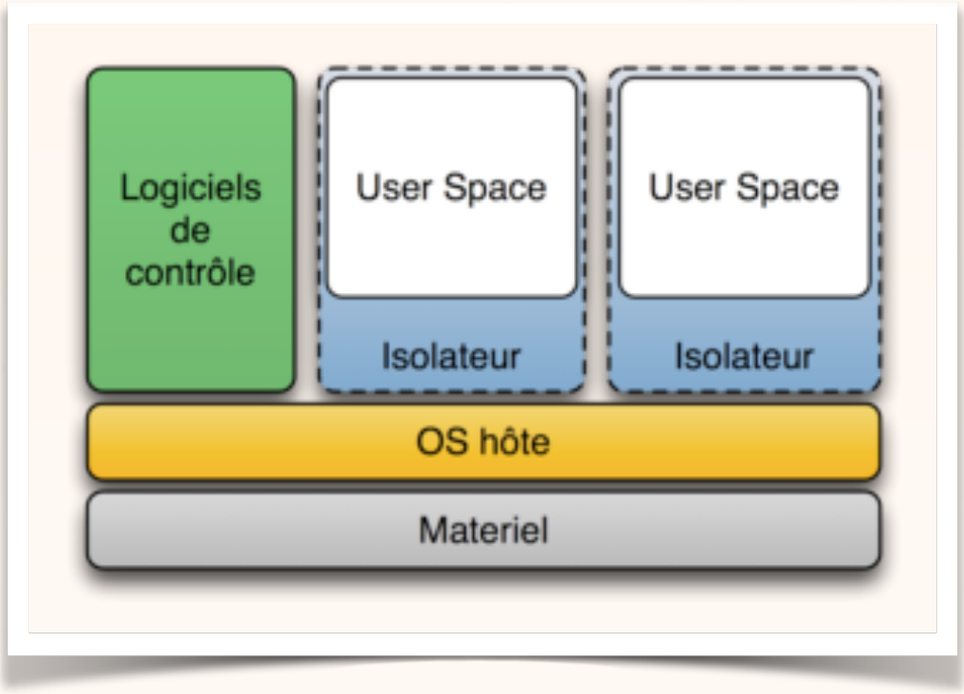


Fig 2.1 - Conteneurs

  - Les conteneurs sont particulièrement adaptés aux applications cloud-natives, aux microservices et aux pratiques DevOps, offrant une approche agile et efficace pour le développement et le déploiement d'applications modernes.

1 - Définitions ; comparaison avec les machines virtuelles

- Conteneurs vs machines virtuelles :
  - **Isolation**
    Les conteneurs partagent le noyau du système d'exploitation hôte, tandis que les VM ont chacune leur propre système d'exploitation complet.
  - **Ressources**
    Les conteneurs sont plus légers et utilisent moins de ressources que les VM, ce qui permet d'exécuter davantage de conteneurs sur un même hôte.

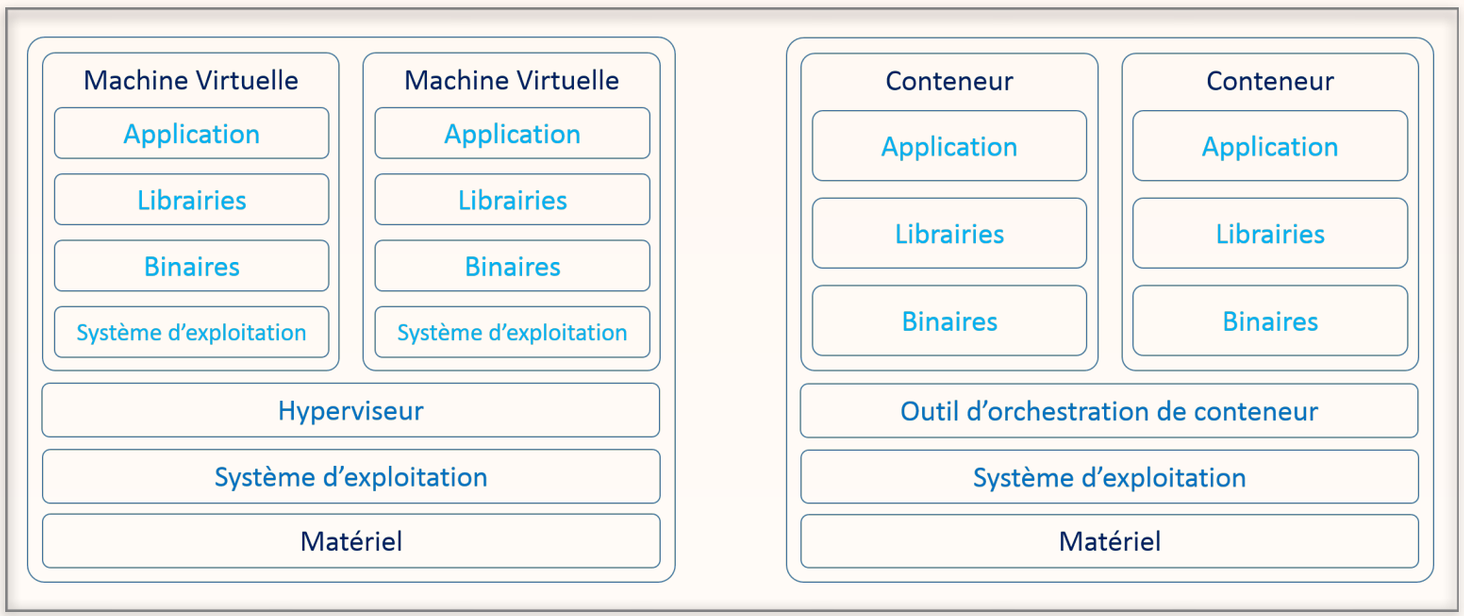


Fig 2.2 - VM vs Conteneurs

1 - Définitions ; comparaison avec les machines virtuelles

Conteneurs vs machines virtuelles, suite…

- **Portabilité**
  Les conteneurs sont hautement portables et peuvent fonctionner de manière cohérente sur différents systèmes.
- **Démarrage**
  Les conteneurs démarrent beaucoup plus rapidement que les VM, souvent en quelques secondes.
- **Taille**
  Les conteneurs sont généralement mesurés en mégaoctets, contrairement aux VM qui sont plus volumineuses.

2 - Orchestration des conteneurs

Orchestration des conteneurs

- L'orchestration des conteneurs est essentielle pour gérer efficacement de nombreux conteneurs dans des environnements de production.
- Les outils d'orchestration de conteneurs automatisent et gèrent le déploiement, la mise à l'échelle et le fonctionnement des applications conteneurisées.
- Les plateformes d'orchestration comme **Kubernetes** et **Red Hat OpenShift** permettent de provisionner, gérer et mettre à l'échelle les conteneurs automatiquement.
- En automatisant ces tâches complexes, les outils d'orchestration comme ***Kubernetes***, ***Docker Swarm*** ou ***Apache Mesos*** permettent aux équipes *DevOps* de gérer efficacement des applications conteneurisées à grande échelle.

2 - Orchestration des conteneurs

- Les plateformes d'orchestration
  - **Kubernetes (**La plus populaire et largement adoptée) :
    - Développé initialement par Google, maintenant géré par la Cloud Native Computing Foundation ;
    - Offre une gestion automatisée du déploiement, de la mise à l'échelle et des mises à jour ;
    - Dispose de fonctionnalités avancées comme l'autoscaling et le monitoring ;
    - Complexe à prendre en main, nécessitant une montée en compétences.
  - **Docker Swarm** (Solution native d'orchestration de Docker) :
    - Plus simple à configurer et utiliser que Kubernetes ;
    - Intégration native avec l'écosystème Docker ;
    - Adapté pour des besoins d'orchestration moins complexes.

2 - Orchestration des conteneurs

- Les plateformes d'orchestration, suite…
  - **Amazon ECS, Elastic Container Service** (Service d'orchestration proposé par AWS)
    - Intégré avec d'autres services AWS comme CloudWatch, Elastic Load Balancer, et CloudTrail ;
    - Permet de provisionner les instances EC2 et de gérer les clusters ;
    - Offre une approche "*Do it yourself*" avec contrôle sur l'infrastructure.
  - **GKE, Google Kubernetes Engine** (Solution d'orchestration de conteneurs de Google Cloud)
    - Basée sur Kubernetes, il facilite le déploiement et l'exécution des applications conteneurisées ;
    - Permet une gestion automatisée du cycle de vie des conteneurs.

2 - Orchestration des conteneurs

- Les plateformes d'orchestration, suite…
  - **Red Hat OpenShift** (plateforme pour développer, moderniser et déployer des applications à grande échelle)
    - Applications de cloud hybride ;
    - Différentes solutions : OpenShift Kubernetes Engine , OpenShift Container Platform.
  - **Marathon**, l'orchestrateur pour conteneurs d'Apache Mesos
    - Il est équipé d'une API REST pour démarrer et stopper les applications.
  - Et aussi :
    - Hashicorp Nomad, Rancher Labs Cattle, CoreOS Fleet, Cloud Foundry Diego, etc.

2 - Orchestration des conteneurs

- Fonctionnement des outils d’orchestration
  - **Planification et déploiement :**
    - L'orchestrateur décide sur quel nœud chaque conteneur doit s'exécuter en fonction des ressources disponibles et des exigences de l'application.
    - Il déploie ensuite automatiquement les conteneurs sur les nœuds appropriés du cluster.
  - **Gestion du cycle de vie :**
    - L'outil surveille en permanence l'état des conteneurs.
    - Si un conteneur échoue, l'orchestrateur le redémarre automatiquement ou le déploie sur un autre nœud pour maintenir la disponibilité de l’application.
  - **Mise à l'échelle automatique :**
    - En fonction de la charge de travail, l'orchestrateur peut augmenter ou diminuer automatiquement le nombre d'instances de conteneurs pour répondre à la demande.

2 - Orchestration des conteneurs

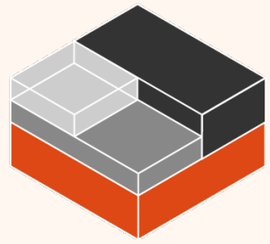
- Fonctionnement des outils d’orchestration, suite…
  - **Gestion des configurations et des secrets :**
    - Les outils d'orchestration gèrent de manière sécurisée les données sensibles comme les identifiants de base de données, en les rendant accessibles uniquement aux conteneurs concernés.
  - **Équilibrage de charge et routage :**
    - L'orchestrateur distribue le trafic entre les différentes instances de conteneurs pour optimiser les performances et la disponibilité.
  - **Surveillance et récupération :**
    - Les outils surveillent en permanence l'intégrité des conteneurs et de l'infrastructure, et peuvent prendre des mesures correctives automatiques en cas de problème.

3 - Fournisseurs de conteneurs

- **Docker :**

  - C'est la plateforme de conteneurisation la plus populaire, offrant des outils pour la création, le déploiement et la gestion des applications conteneurisées.
  - Docker est un outil qui peut empaqueter une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur.
  - Techniquement, Docker étend le format de conteneur Linux standard, LXC, avec une API de haut niveau fournissant une solution pratique de virtualisation qui exécute les processus de façon isolée.
  - Plus de détails sur Docker ci-après.

3 - Fournisseurs de conteneurs



- **LXC (Linux Containers) :**
  - Une technologie de conteneurisation au niveau du système d'exploitation, offrant une virtualisation légère pour Linux.
  - Linux Container (LXC) désigne d'un part les applications virtualisées basées sur Linux et la plateforme et d'autre part la technologie de conteneur sous-jacente. Certaines plateformes de conteneurs alternatives utilisent également les Linux Container comme technologie.
    - Par ex. : CRI-O (voir ci-après) et OpenVZ, qui nécessite un noyau Linux spécifiquement modifié, mais qui est plus facile à utiliser
- **CRI-O :**
  - C'est une implémentation de l'interface CRI, *Container Runtime Interface,* de Kubernetes pour permettre l'utilisation de systèmes d'exécution (*runtimes*) compatibles avec l'OCI, *Open Container Initiative.* Il s'agit d'une alternative légère à l'utilisation de Docker comme runtime pour Kubernetes.

3 - Fournisseurs de conteneurs

- **Podman :**

  - Il s’agit d’une alternative à Docker, qui permet de lancer les commandes sans les permissions root.
  - Podman (version contractée de « POD manager ») est un outil Open Source qui sert à développer, gérer et exécuter des conteneurs.
  - À l’inverse de Docker, Podman n’intègre pas de daemon nécessaire à son fonctionnement.
  - Podman s'exécute sur diverses distributions Linux, notamment Red Hat Enterprise Linux, Fedora, CentOS et Ubuntu.
  - Podman Desktop est une interface utilisateur graphique pour Podman.
- **RKT (Rocket) :**
  - C’était une alternative à Docker, développée par CoreOS et arrêté en 2018, mettant l'accent sur la sécurité et la simplicité.

4 - Précisions sur Docker

- **Docker :**
  - Docker est un outil qui peut empaqueter une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur.
  - La plateforme Docker a été créée en 2013 par Solomon Hykes. Docker est distribué en tant que projet open source.
  - Docker utilise entre autres LXC, cgroups et le noyau Linux lui-même.
    - cgroups, *control groups,* est une fonctionnalité du noyau Linux pour limiter, compter et isoler l'utilisation des ressources (processeur, mémoire, utilisation disque, etc.).
  - Un conteneur Docker s'appuie sur les fonctionnalités du système d'exploitation fournies par la machine hôte.

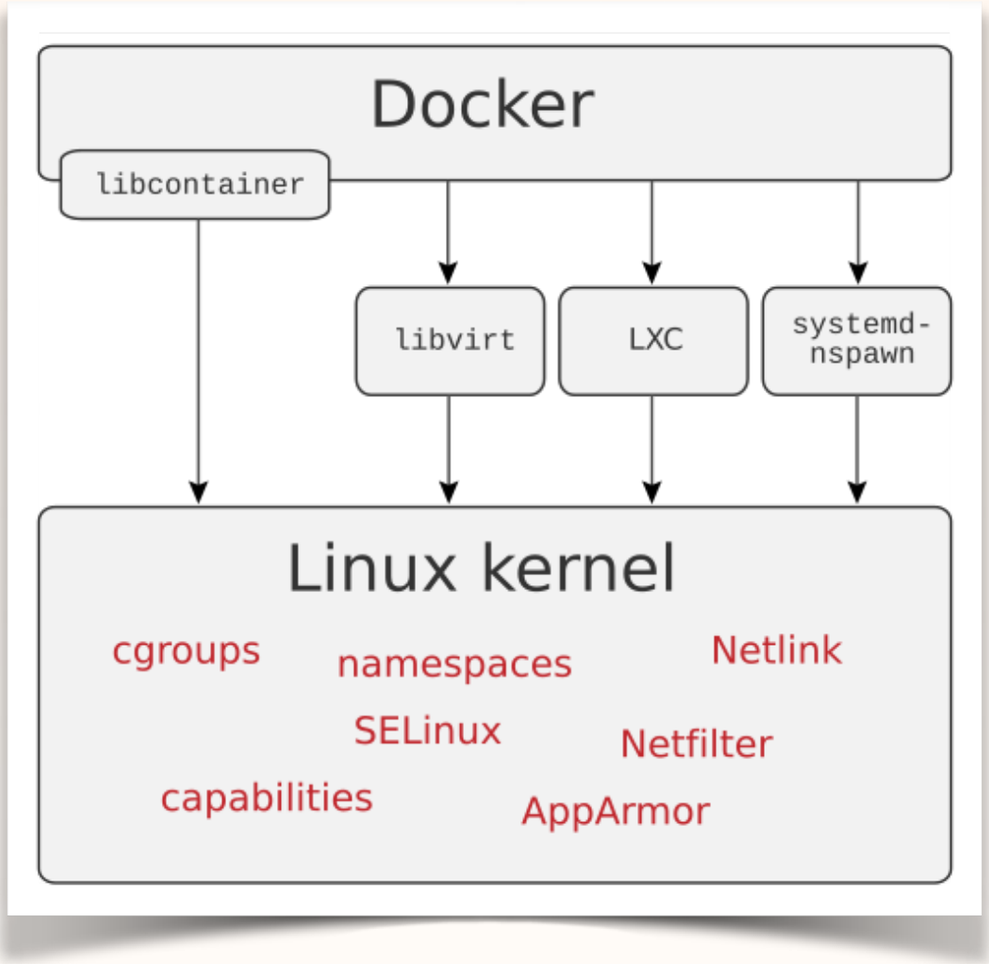


Fig 2.3 - Schéma des interfaces de Docker

4 - Précisions sur Docker

- Installation de Docker :
  - Docker Desktop – Le moyen le plus simple de conteneuriser des applications
  - Docker Desktop vous fournit tous les outils dont vous avez besoin pour exécuter des applications Docker sur votre bureau :
    - le moteur Docker,
    - les outils Docker d'interface de ligne de commande (CLI, *Command line interface*),
    - la fonction Docker Compose.
- Docker desktop se compose d'outils de développement, de Docker App, de Kubernetes et de la synchronisation des versions.
- Il nous permet de créer des images certifiées et des modèles de langues et d'outils de notre choix.

4 - Précisions sur Docker

- Installation de Docker, suite…
  - Docker Desktop est gratuit si vous ne l'utilisez pas à des fins commerciales.
  - Docker se décline principalement en deux éditions, l'édition ***Community*** et l'édition ***Enterprise***.
  - L'édition communautaire est livrée avec un ensemble gratuit de produits Docker.
  - L'édition *Enterprise* est une plateforme de conteneurs certifiée qui offre aux utilisateurs commerciaux des fonctionnalités supplémentaires telles que :
    - la sécurité des images,
    - la gestion des images,
    - l'orchestration et la gestion de l'exécution des conteneurs.

4 - Précisions sur Docker

Installation de Docker, suite…

- Voir Installer et utiliser Docker sur Windows 11.
- Dans Windows, pour utiliser Docker, vous devez activer la virtualisation, car la technologie de conteneur requiert un noyau Linux :
  - Avec Windows Famille, il est nécessaire de passer par **WSL 2** pour exécuter Docker Desktop.
  - Windows Professionnel et Entreprise prennent à la fois en charge **Hyper-V** et le **WSL 2.**
- Pré-requis pour Windows :
  - Processeur (CPU) : 64 bits avec traduction d'adresse de deuxième niveau (« Second level address translation » ou SLAT)
  - Mémoire vive (RAM) : 4 Go
  - Espace en disque dur : 20 Go minimum

4 - Précisions sur Docker

- Installation de Docker, suite…
  - Sur mac, voir Install Docker Desktop on Mac
  - ouvrir le programme d'installation *Docker.dmg*, puis glisser l'icône *Docker* dans le dossier Applications.
  - Ouvrir *Docker.app* pour démarrer *Docker Desktop*.
  - Comme *Docker Desktop* pour Windows, *Docker Desktop* pour MAC propose *Docker Engine*, *Notary*, *Docker compose*, *Kubernetes* et *Credential helper*.

4 - Précisions sur Docker

- Utilisation de Docker
  - Une fois installé, vérifiez la version du moteur Docker installée.
    - `docker --version`
  - Vous pouvez alors
    - Créer votre propre application conteneurisée,
    - Ou télécharger des images conteneurisées sur Docker Hub
- Docker maintient des images conteneurisées sur Docker Hub, et elles peuvent être facilement téléchargées à l'aide de la simple commande docker run.
- Par ex., pour extraire l'image *Redis* Puis démarrer le conteneur :
  - `docker pull redis`
  - `docker run -p 6379 Redis`

4 - Précisions sur Docker

Fonctionnalités de Docker Desktop

- Prise en charge d'une grande variété d'outils et de langages de développement.
- Fournit un moyen rapide et optimisé de créer et de partager une image conteneurisée sur n'importe quelle plateforme cloud.
- Facile à installer et à configurer un environnement Docker complet
- Meilleures performances avec la virtualisation native Hyper-V sur Windows et HyperKit sur MAC.
- Possibilité de travailler nativement sur Linux grâce à WSL 2 sur les machines Windows.
- Accès facile aux conteneurs en cours d'exécution sur le réseau local.
- Possibilité de partager n'importe quelle application sur la plateforme cloud, dans différents langages et frameworks.
- Les dernières versions de Kubernetes sont incluses.
- **Remarque** : Docker Desktop n'est pas destiné à un environnement de production, mais plutôt à un environnement de bureau et de développement.

4 - Précisions sur Docker

- Des liens pour Docker :
  - https://www.docker.com/
  - Docker - Wikilivres
  - Docker : qu'est-ce que c'est et comment l'utiliser ? - DataScientest
  - Optimisez votre déploiement en créant des conteneurs avec Docker - OpenClassrooms
  - AWS :
    - Qu'est-ce que Docker ?
    - Amazon ECS - Amazon Elastic Container Service
    - AWS Fargate - moteur de calcul sans serveur
  - Tutoriel Docker - IONOS
  - Docker Swarm pour l'orchestration des conteneurs - Geekflare
  - Construire et Orchestrer des conteneurs - Stéphane ROBERT
  - Docker Cheat Sheet - Stéphane ROBERT

5 - Kubernetes

Les bases de Kubernetes

- **Kubernetes**, alias K8s, est une plateforme open source pour **automatiser** le déploiement, la montée en charge et la mise en œuvre de conteneurs d'application sur des grappes de serveurs
- Il fonctionne avec différentes séries de technologies de conteneurisation.
- Mais il est souvent utilisé avec **Docker**.
- Il a été conçu à l'origine par **Google**, en 2014, puis offert à la **Cloud Native Computing Foundation**.
- Kubernetes est utilisé par Red Hat pour son produit **OpenShift**.
- Voir sur kubernetes.io :
  - Vue d'ensemble
  - Apprendre les bases de Kubernetes

5 - Kubernetes

- Les bases de Kubernetes, suite…
  - L'unité de base de l'ordonnancement dans Kubernetes est appelée « *pod* ».
    - C'est une vue abstraite de composants conteneurisés.
  - Les éléments de Kubernetes :
    - Un **pod** désigne un groupe de conteneur géré par Kubernetes :
      - Il possède une adresse IP unique ;
      - Il consiste en un ou plusieurs conteneurs qui ont la garantie d'être co-localisés sur une machine hôte et peuvent en partager les ressources.

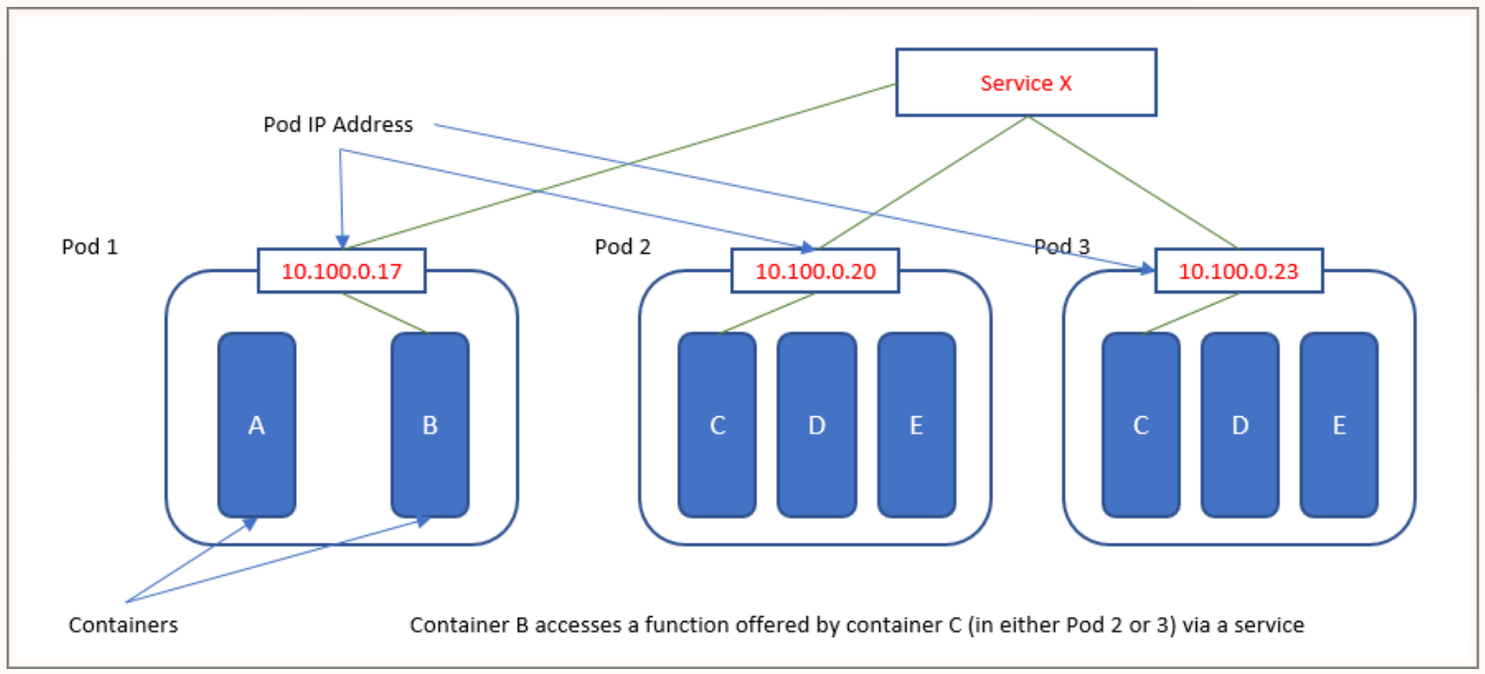


Fig 1.6 - Mise en réseau des pods et les services pour résoudre les dépendances réseau.
Par Marvin The Paranoid, CC BY-SA 4.0

5 - Kubernetes

- Les bases de Kubernetes, suite…
  - Les éléments de Kubernetes, suite…
    - Un **service** Kubernetes est un groupe de pods travaillant ensemble.
    - Labels et selectors :
      - Kubernetes permet à des clients (utilisateurs et composants internes) d'attacher des paires clés-valeurs appelées « labels » à n'importe quel objet d'API dans le système, par exemple les pods et les nodes.
      - Ils constituent le premier mécanisme de groupement dans Kubernetes, et sont utilisés pour déterminer les composants sur lesquels appliquer une opération
      - Un *label selectors* est une interrogation faite avec des labels
    - Contrôleur :
      - C’est une boucle d'arbitrage qui pilote l'état courant d'un cluster vers son état désiré.
      - Il effectue cette action en gérant un ensemble de pods.

5 - Kubernetes

- Architecture d'un cluster Kubernetes

Plan de contrôle

Nœuds

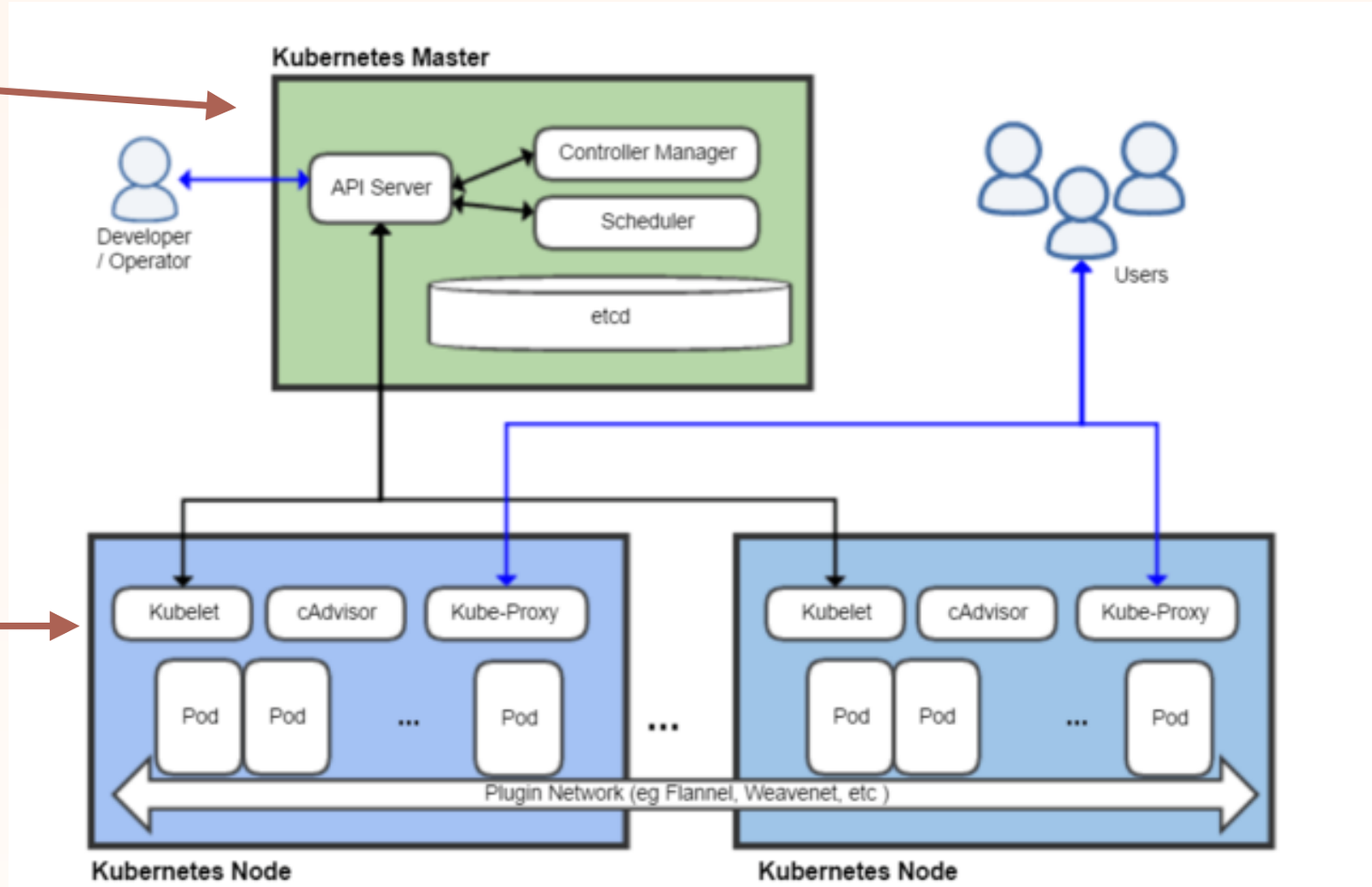


Fig 2.4 - Diagramme d'architecture de Kubernetes. *Par Khtan66 , CC BY-SA 4.0*

5 - Kubernetes

Architecture d'un cluster Kubernetes, suite…

- Un **cluster Kubernetes** est un ensemble de machines (les nœuds ou **nodes**) qui permettent d'exécuter des applications conteneurisées.
  - Si vous exécutez Kubernetes, vous exécutez un cluster.
- Un cluster comprend au minimum un **plan de contrôle** avec une ou plusieurs machines de calcul (ou **nœuds**).
- Le **plan de contrôle** est responsable du maintien du cluster dans un état souhaité ;
  - il vérifie, par exemple, les applications exécutées et les images de conteneurs utilisées.
  - Le **maître Kubernetes** est l'unité de contrôle principale qui gère la charge de travail et dirige les communications dans le système.
  - Les composants d'un plan de contrôle sont : **etcd**, le **serveur d'API**, L'**ordonnanceur** (*scheduler*), le **gestionnaire de contrôle** (controller manager).
    - Ces composants sont détaillés ci-après.
- Ce sont les **nœuds,** ou ***nodes,*** qui exécutent concrètement les applications et les charges de travail, grâce aux **pods** qu'ils contiennent.

5 - Kubernetes

- Architecture d'un cluster Kubernetes, suite…
  - Les composants d'un plan de contrôle sont :
    - **etcd**, une unité de stockage distribuée persistante et légère de données clé-valeur, pour stocker de manière fiable les données de configuration du cluster.
    - Le **serveur d'API** : il gère et valide des requêtes REST et met à jour l'état des objets de l'API dans etcd, permettant ainsi aux clients de configurer la charge de travail et les containers sur les nœuds de travail.
    - L'**ordonnanceur** (*scheduler*) : il gère l'utilisation des ressources sur chaque nœud afin de s'assurer que la charge de travail n'est pas en excès par rapport aux ressources disponibles.
    - le **gestionnaire de contrôle** (*controller manager*) :
      - c'est le processus dans lequel s'exécutent les contrôleurs principaux de Kubernetes tels que *DaemonSet Controller* et le *Replication Controller*.
      - Les contrôleurs communiquent avec le serveur d'API pour créer, mettre à jour et effacer les ressources qu'ils gèrent (*pods*, *service endpoints*, etc.)

5 - Kubernetes

Architecture d'un cluster Kubernetes, suite…

- Un **nœud**, appelé **node**, *worker* ou *minion* est une machine unique (ou une machine virtuelle) où des conteneurs (charges de travail) sont déployés.
- Chaque node du cluster doit exécuter le programme de conteneurisation (par exemple Docker), ainsi que les composants mentionnés ci-dessous, pour communiquer avec le maître afin de configurer la partie réseau de ces conteneurs :
  - **Kubelet** est responsable de l'état d'exécution de chaque nœud. Il surveille l'état d'un pod et s'il n'est pas dans l'état voulu, le pod sera redéployé sur le même node.
  - **Kube-proxy** est l'implémentation d'un proxy réseau et d'un répartiteur de charge.
  - **cAdvisor** est un agent qui surveille et récupère les données de consommation des ressources et des performances comme le processeur, la mémoire, ainsi que l'utilisation disque et réseau des conteneurs de chaque node.

5 - Kubernetes

- TP sur Kubernetes
  - Using Minikube to Create a Cluster.
  - Using kubectl to Create a Deployment.

La virtualisation réseau

Chapitre 3

1 - Introduction

- La virtualisation réseau :
  - La virtualisation réseau est une technologie qui permet d'abstraire les ressources physiques du réseau pour créer des réseaux logiques et virtuels.
  - Elle transforme un réseau dépendant du matériel en un réseau basé sur des logiciels
- Le plan de ce chapitre est :
  - 1 - Introduction
  - 2 - Caractéristiques principales
  - 3 - SDN, *Software Defined Networking*
  - 4 - Autres technologies de virtualisation réseau
  - 5 - Outils et applications

2 - Caractéristiques principales

- Définition :
  - La virtualisation réseau est une technologie qui permet d'abstraire les ressources physiques du réseau pour créer des réseaux logiques et virtuels.
  - Elle transforme un réseau dépendant du matériel en un réseau basé sur des logiciels.

- Principales caractéristiques de la virtualisation réseau :
  - **Abstraction** des ressources physiques :
    La virtualisation réseau extrait les ressources physiques du réseau pour créer des réseaux logiques et virtuels.
  - **Segmentation** du réseau :
    Elle permet de créer plusieurs réseaux virtuels indépendants sur une seule infrastructure physique.

2 - Caractéristiques principales

- Principales caractéristiques de la virtualisation réseau
  - Découplage des **plans de contrôle et de données** :
    Le plan de contrôle (décisions de routage) est séparé du plan de données (transfert de paquets).
  - **Gestion centralisée** :
    Les administrateurs peuvent configurer et gérer les ressources réseau par programmation via des panneaux de contrôle centralisés.
  - **Flexibilité et évolutivité** :
    Les réseaux virtuels peuvent être rapidement configurés et redimensionnés selon les besoins.
  - **Optimisation des ressources** :
    Elle permet une utilisation plus efficace des ressources matérielles, réduisant les coûts.

2 - Caractéristiques principales

- Principales caractéristiques de la virtualisation réseau
  - **Indépendance du matériel** :
    Les services réseau (routage, commutation, pare-feu) sont implémentés en logiciel, indépendamment du matériel propriétaire.
  - **Types de virtualisation** :
    Elle peut être externe (combinant des systèmes physiques en VLAN) ou interne (au sein d'un seul serveur).
  - **Automatisation** :
    De nombreuses tâches de gestion réseau peuvent être automatisées, réduisant la complexité et les coûts.
  - **Support pour le cloud** :
    C’est un élément clé des solutions cloud, offrant une infrastructure IT plus agile et flexible.

2 - Caractéristiques principales

- Les principaux types de virtualisation de réseau :
  - **VLAN,** Virtual LAN
    - Créer des segments de réseau logique au sein d'un même réseau physique pour isoler et regrouper les ressources selon des critères spécifiques.
  - **SDN**, *Software Defined Networking,* permet la gestion centralisée des réseaux en séparant le plan de contrôle du plan de données pour une configuration dynamique et flexible.
  - **VPC** , *Virtual Private Cloud,* fournit un réseau isolé et privé dans le cloud, permettant aux entreprises de déployer des ressources cloud en toute sécurité.
  - **NFV**, *Network Function Virtualization*, convertit les fonctions réseau traditionnelles, telles que les pare-feu et équilibreurs de charge en logiciel pour permettre une gestion évolutive.
    - On remplace le matériel des appareils de réseau par des machines virtuelles efficaces.
    - Ces machines virtuelles ont besoin d'un hyperviseur pour exécuter les processus de mise en réseau, tels que l'équilibrage des charges et le routage.

3 - SDN, *Software Defined Networking*

- Nouveau concept d'architecture de réseaux
  - 2008 : recherches d'équipe des universités de Berkeley et Stanford
  - 2011 : Open Networking foundation
    - pour la promotion du SDN,
    - créé par Deutsche Telekom, Facebook, Google, Microsoft, Verizon, et Yahoo.
    - Rejoint par tous les grands constructeurs/éditeurs IT tels que Cisco, Juniper, HP, Dell, Broadcom, IBM, etc.
- Principe
  - Séparer la partie opérationnelle liée au fonctionnement des routeurs et commutateurs de la partie décisionnelle réalisée par un contrôleur ;
  - S'affranchir des spécificités des équipements ;

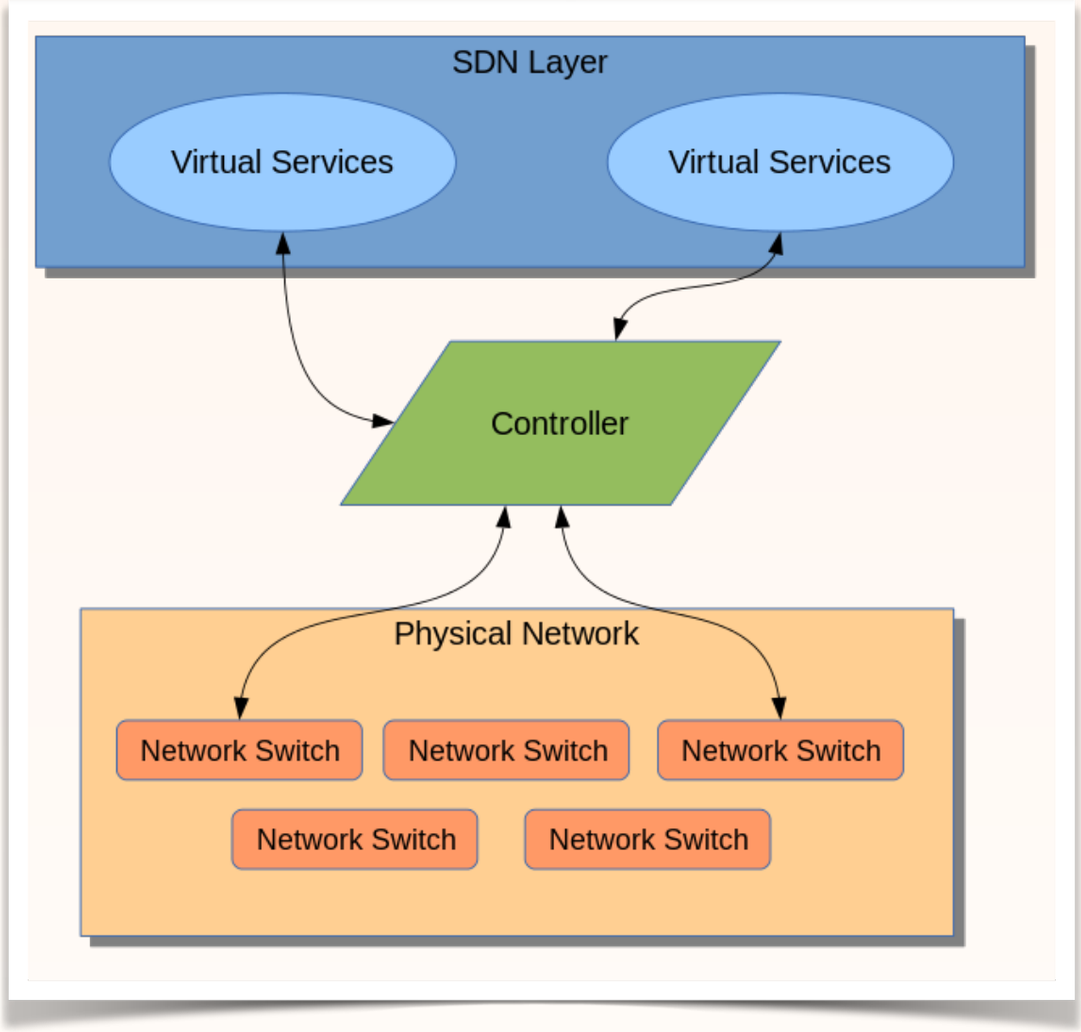
3 - SDN, *Software Defined Networking*

- Principe, suite
  - Faciliter grâce à une API réseau standard le développement de services réseaux à forte valeur ajoutée
    - Équilibrage de charge
    - Configuration
    - Planification
    - Routage intelligent
- Définitions
  - Le réseau à définition logicielle, ou SDN, *Software-Defined Networking*, est une approche de la gestion des réseaux dans laquelle le contrôle est dissocié du matériel et transféré à une application logicielle appelée contrôleur.

3 - SDN, *Software Defined Networking*

Les plans d'équipements réseaux

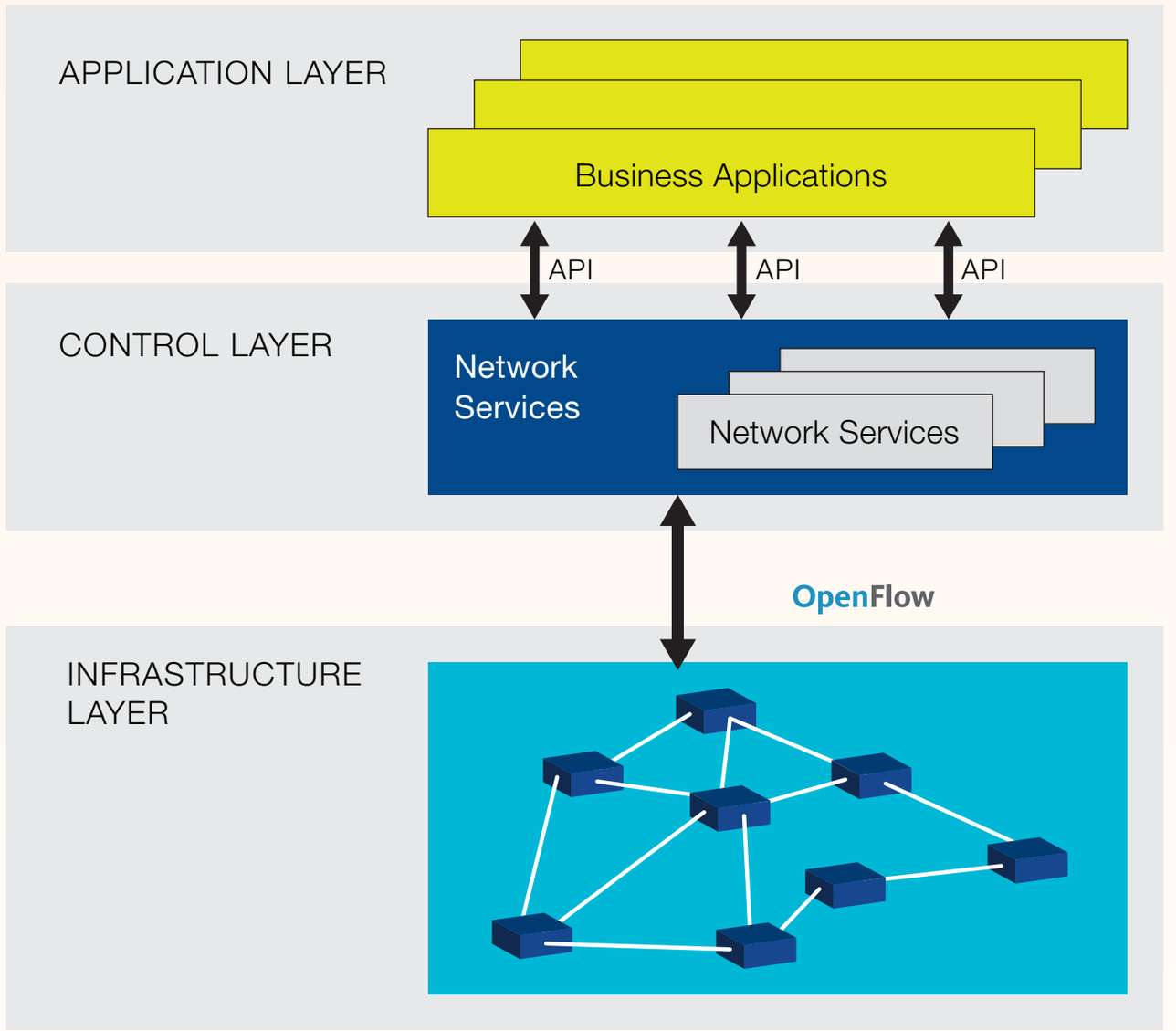
- Le **plan de gestion** (*management plane*) concerne l'administration et la configuration des équipements, à l'aide de flux SSH, *Secure Shell* ou SNMP, *Simple Network Management Protocol*. Il est parfois considéré comme un sous-ensemble du plan de contrôle.
- Le **plan de contrôle** (*control plane*) va contrôler le plan de données suivant des règles établies. Les protocoles comme OSPF, STP, ARP, BGP et leurs tables participent à ce plan de contrôle.
- Le **plan de données** (*data plane*) est liée à l'infrastructure, soit l'ensemble des équipements, switches, routeurs… permettant l'acheminement des données.



3 - SDN, *Software Defined Networking*

- **Plan de gestion**
  - Applications logicielles
- **Plan de contrôle**
  - Processus réseau qui dirigent le trafic réseau
- **Plan de données**
  - Données traversant le réseau

FIGURE 1
ONF/SDN architecture



3 - SDN, *Software Defined Networking*

- Architecture SDN
  - Trois composantes importantes définissent une architecture SDN :
    - La décorrélation du plan de contrôle et du plan de données ;
    - L'abstraction du réseau physique ;
    - La programmabilité du réseau.
  - Le SDN propose de créer un point central, le contrôleur, pour gérer le plan de contrôle des équipements.
    - Ce contrôleur transmet des instructions à l'aide d'un protocole standardisé par ONF : OpenFlow
    - D'autres protocoles sont possibles :
      - XMPP, *Extensible Messaging and Presence Protocol* ;
      - *Networking Configuration Protocol* (protocole Netconf) ;
      - Cisco utilise leurs propres protocoles propriétaires pour leurs solutions SDN, comme OpFlex pour le système Cisco ONE.

3 - SDN, *Software Defined Networking*

- Le contrôleur
  - Le contrôleur (SDN Controler) transmet des instructions vers le plan de données (les équipements réseau) à l'aide d'un protocole standardisé par ONF : OpenFlow
  - OpenFlow est un protocole réseau standard qui permet de réaliser une architecture SDN ;
    il permet l'administration à distance de commutateurs de niveau 3
  - Exemple de contrôleur open source proposé par ONF : ONOS, *Open Network Operating System.*
    - ONOS Overview
    - What is ONOS (Open Network Operating System) - TechTarget
  - Le contrôleur central reçoit également, via OpenFlow, des informations des équipements (switches, routeurs, etc.)
  - Il possède une vue logique du réseau (abstraction de la topologie réseau), utilisée pour toutes les décisions prisent par le plan de contrôle.

3 - SDN, *Software Defined Networking*

- La programmabilité du réseau
  - Le contrôleur présente l'abstraction du réseau et une API pour les applications SDN
  - Ces applications SDN dialoguent avec le contrôleur et implémentent des services tels que routage, sécurité, QoS, monitoring, etc.

- Voir :
  - Software-Defined Networking (SDN) Definition
  - Les risques d'OpenFlow et du SDN - ANSSI
  - https://www.opennetworking.org

4 - Autres technologies de virtualisation réseau

- NFV, *Network functions virtualization*
  - Cela consiste à virtualiser les services et fonctions réseaux actuellement mis à disposition par un matériel dédié et propriétaire.
  - Réalisée dans les règles de l'art, la NFV diminue la quantité de matériel propriétaire nécessaire au lancement et à l'exploitation de services réseau.
- RAN, *Radio Access Network*, Réseau d'accès radio
  - Technologie de connexion de stations mobiles à un réseau 3G, 4G ou 5G.
- Open RAN
  - initiative axée sur la virtualisation et la désagrégation des réseaux télécoms.
  - L'alliance O-RAN, fondée par 5 opérateurs (dont Orange), cherche à créer des standards et des interfaces ouvertes.
- SD-RAN
  - Plateforme de l'ONF pour le software-defined RAN conforme à la norme 3GPP ;
  - Il est compatible avec l'architecture O-RAN.

5 - Outils et applications

- Les outils
  - **Terraform :**
    - Outil unifié qui simplifie la gestion de l'infrastructure, y compris l'infrastructure réseau.
    - IaC, *Infrastructure as Code,* consiste à gérer et à approvisionner une **infrastructure** à l'aide de lignes de code plutôt que par des processus manuels.
    - Le langage utilisé pour définir l'infrastructure est HCL, *HashiCorp Configuration Langage.*
    - Voir : Terraform Infrastructure as Code
  - **Ansible** :
    - Outils d'automatisation les plus populaires en administration réseau.
    - C'est un moteur Open Source qui automatise le provisionnement, la gestion des configurations, le déploiement des applications, l'orchestration, etc.
    - Voir : Apprendre les bases d'Ansible et Automatiser les réseaux avec Red Hat Ansible Automation Platform.

6 - Conclusion

- Voir aussi :
  - La virtualisation des réseaux, qu'est-ce que c'est ? - RedHat
  - Les réseaux définis par logiciel (SDN) expliqués en 5 minutes ou moins - Geekflare
  - La virtualisation des fonctions réseau expliquée - Geekflare

La virtualisation de stockage

Chapitre 4

1 - Introduction

- La virtualisation de stockage :
  - La virtualisation de stockage consiste à assembler la capacité de stockage de multiples appareils de stockage en réseau sous forme d'un seul appareil de stockage (virtuel) pouvant être géré depuis une console centrale.
  - Le plan de ce chapitre est :
    - 1 - Introduction
    - 2 - Systèmes de stockage : Historique ; Types de stockage ; Protocoles de stockage
    - 3 - Fonctionnement
    - 4 - Types de virtualisation du stockage
    - 5 - Network File System
    - 6 - Autres systèmes de fichiers
    - 7 - Fournisseurs et solutions

2 - Systèmes de stockage

- Histoire des systèmes de stockage de données et de fichiers
  - Systèmes de stockage :
    - Cartes perforées (1890) puis bandes magnétiques (années 1950) ;
    - Disque dur (IBM, 1956) ;
    - Disquettes (IBM, 1971, 8 pouces) ;
    - CD (1980, 74 mn de musique) ;
    - CD-ROM et DVD (années 1980-1990) ;
    - SSD, *Solid State Drive* et mémoire flash (années 2000) ;
    - SAN, *Storage Area Network*, réseau de stockage (années 2000) ;
    - Stockage cloud (années 2010) et développement du stockage unifié en mode fichier et objet (UFFO, *Unified Fast File and Object*).

La virtualisation de stockage

Chapitre 4

2 - Systèmes de stockage

- Histoire des systèmes de stockage de données et de fichiers :
  - Systèmes de fichiers :
    - DECtape (DEC, 1964)
    - CP/M (1974, pour disquettes et disques durs)
    - FAT12 (1980) et FAT16 (1984) (*File Allocation Table*)
    - NTFS, *New Technology File System* (1993) et FAT32 (1996)
  - Systèmes de fichiers en réseau :
    - SMB/CIFS, *Server Message Block* / *Common Internet File System* pour Windows et Samba pour Linux ;
    - NFS, *Network File System*
  - Systèmes de fichiers distribués :
    - AFS, *Andrew File System* ; HDFS, *Hadoop Distributed File System* ; Lustre…
  - Virtualisation du stockage, alias *Software-Defined Storage* ou SAN virtuel (depuis les années 1990).

2 - Systèmes de stockage

- Types de stockage
  - Stockage de fichiers :
    - Les données sont stockées comme une seule pièce d'informations au sein d'un dossier, afin de faciliter l'organisation. On parle aussi de stockage hiérarchique.
  - Stockage de blocs :
    - Cela consiste à diviser un fichier en plusieurs blocs de données, pour ensuite les stocker séparément.
    - Ceci permet de répartir les blocs au sein du système de stockage de façon plus efficiente.

- Types de stockage, suite
  - Stockage objet (ou stockage orienté objet) :
    - Le stockage objet ne repose pas sur une hiérarchie de répertoires.
    - Les données sont stockées sous forme d'objets dans un espace d'adressage linéaire, appelé pool de stockage.
    - Chaque objet est constitué de la donnée, d'un identifiant unique et de métadonnées.
      - L'identifiant unique correspond au chemin d'accès de la donnée.
      - Les métadonnées sont les informations liées au contexte de l'objet (type de donnée, structure…). Elles permettent de lier les données qui comportent les mêmes métadonnées.
    - Le stockage objet est idéal pour l'analyse de données.
    - Amazon S3, *Simple Storage Service,* est un service de stockage objet proposé par AWS, *Amazon Web Services*.

- Protocoles de stockage
  - **iSCSI**, *Internet Small Computer System Interface*.
    - Protocole de stockage en réseau basé sur le protocole IP destiné à relier les installations de stockage de données.
    - iSCSI est utilisé pour transmettre des données sur des LAN ou à travers Internet.
  - **Fibre Channel** est un protocole standard des SAN, où il transporte les données entre un serveur et la mémoire d'une baie de stockage.
    - Débits jusqu'à 16 Gbit/s et bientôt 32 Gbit/s.
    - Topologies : point à point, commutée (*fabric*) ou avec boucle arbitrée (FC-AL pour *Arbitrated Loop*)
  - **FCoE**, *Fibre channel Over Ethernet* : les trames du protocole Fibre Channel sont transmises sur un réseau Ethernet.

La virtualisation de stockage

Chapitre 4

3 - Fonctionnement

- La virtualisation de stockage :
  - La virtualisation du stockage sépare le logiciel de gestion du stockage de l'infrastructure matérielle sous-jacente afin d'offrir :
    - plus de flexibilité ;
    - des pools de ressources de stockage évolutifs.
  - De plus, elle permet de virtualiser le matériel de stockage (baies et disques) sous forme de pools de stockage virtuel.
  - Cette virtualisation s'est développée depuis les années 1990.
  - Elle permet de regrouper et de gérer tous les stockages existants sous une seule console.

  - Voir : Virtualisation du stockage - Étude approfondie - DataCore

3 - Fonctionnement

- Principe de base :
  - Les disques physiques sont séparés du volume virtuel par une couche de virtualisation.
  - Le logiciel de stockage virtuel gère les demandes d'entrée/sortie (I/O) et les redirige vers les dispositifs appropriés dans le pool global.
  - Regroupement des ressources : Plusieurs disques physiques sont combinés en un groupe sur un seul serveur.
  - Création de blocs virtuels : Des blocs de stockage virtuels ou logiques sont affectés au serveur.
  - Redirection du trafic : Le trafic d'I/O est redirigé vers les ressources physiques appropriées.

4 - Types de virtualisation du stockage

- Virtualisation basée sur le réseau :
  - La forme la plus courante de virtualisation du stockage.
  - Les dispositifs de stockage sont reliés à un réseau FC (*Fibre Channel*) ou iSCSI.
  - Présente un pool virtuel unique au sein du réseau de stockage.
- Virtualisation basée sur l'hôte :
  - Cela utilise un logiciel pour diriger le trafic.
  - Couramment utilisée dans les systèmes HCI, *Hyper-Converged Infrastructure,* et le stockage cloud.
    - L'hyperconvergence est un type d'architecture matérielle qui agrège les composants de traitement, de stockage, de réseau et de virtualisation de plusieurs serveurs physiques.
  - Permet d'attribuer le stockage physique à presque n'importe quel dispositif ou baie.

4 - Types de virtualisation du stockage

- Virtualisation basée sur les baies :
  - Une baie de stockage sert de contrôleur principal.
  - Regroupe les ressources de stockage d'autres baies.
  - Peut présenter différents types de stockage physique comme des niveaux distincts.
- Virtualisation en mode bloc et fichier
  - Mode bloc : Ajoute un niveau d'abstraction entre le serveur et le système de stockage.
  - Mode fichier : Masque les dépendances vis-à-vis de l'emplacement physique des données dans un NAS.

5 - Network File System

- NFS ; *Network File System* :
  - NFS est souvent utilisé dans les environnements de virtualisation pour fournir un stockage partagé aux machines virtuelles.
  - NFS peut donc être considéré comme un composant d'une solution de virtualisation du stockage plus large.
  - NFS est un **système de fichiers en réseau** développé par Sun Microsystems (racheté par Oracle en 2009). NFS est supporté par un grand nombre de systèmes d'exploitation.
  - Avec NFS, utilisateurs et programmes accèdent aux dossiers et fichiers distants comme s'ils étaient locaux.
  - C'est un système client-serveur.
  - Le serveur rend des répertoires accessibles : il exporte ou partage des répertoires
  - Le client monte (*mount*) un répertoire lorsqu'il attache un répertoire distant à un système de fichier local.

5 - Network File System

- NFS fonctionne grâce des différents démons :

Démon	Serveur	Client	Détail
nfsd	√		réponses aux requêtes client
mountd	√		demande de montage
nfslogd	√		journal
rquotad	√	√	quota
lockd	√	√	verrouillage
statd	√	√	surveillance de l'état du réseau

5 - Network File System

- Configuration du serveur
  - Le but du partage de système de fichier est de :
    - partager des données entre utilisateur d'un même groupe
    - éviter la duplication de répertoires
    - offrir des programmes et des données centralisés
    - fournir de l'espace disque à des clients sans disque
  - Commande ***share*** (système Solaris) :
    - elle lit les répertoire à exporter dans /etc/dfs/dfstab
  - Commande ***exportfs*** (système Linux)
    - elle lit en /etc/exports : les répertoires à exporter, les machines qui peuvent y accéder et les droits d'accès associés.

5 - Network File System

- Configuration de clients
  - Commande ***showmount*** : obtenir des informations sur un serveur ;
  - Commande ***mount*** : montage de répertoire :
    - `mount -t nfs nom du serveur:dossier distant dossier local`
  - Commande ***umount*** : démontage de répertoire.
- NFSv4
  - L'IETF, *Internet Engineering Task Force*, est maintenant chargé du développement des protocoles NFS.
  - La version 4 de NFS est révisée en 2003 (RFC 3530) .
  - NFSv4.1 (RFC 5661) a été publié en 2010 et **NFSv4.2** (RFC 7862) a été publié en 2016.

5 - Network File System

- **NFSv4,** par rapport à ses prédécesseurs, embarque de nombreuses avancées telles que :
  - Une technologie de cache agressive (délégation) ;
  - Le regroupement des requêtes réseau (*Compound request*) ;
  - La sécurisation négociée et le chiffrement des données : Kerberos 5, Certificats (SPKM), Clefs publiques/privées (LIPKEY) ;
  - La capacité pour les clients de maintenir des sessions ou de les récupérer malgré un crash serveur ou une panne du réseau ;
  - La possibilité (à terme) de rediriger la charge de serveurs saturés vers un autre serveur, de manière transparente pour les clients ;
  - Le support d'attributs fichier nommés par l'utilisateur (ex. un attribut ‘photos’).

6 - Autres systèmes de fichiers

- Autres systèmes de fichiers réseaux :
  - **SMB, Server Message Block** ou **SMB/CIFS** permet le partage de ressources (fichiers et imprimantes) sur des réseaux locaux avec des PC sous Windows.
  - **Samba** est une implémentation libre des protocoles SMB/CIFS pour Linux et Unix.
    - Samba v3 fournit sur un réseau local des fichiers et services d'impression pour divers clients Windows et peut s'intégrer à un domaine Windows Server ou faire partie d'un domaine Active Directory.
  - Voir aussi les logiciels de **gestion de versions** : Apache Subversion et Git.

6 - Autres systèmes de fichiers

- Systèmes de fichiers distribués
  - **AFS**, *Andrew File System*, est un système d'archivage distribué ; il est maintenant supplanté par NFSv4 et Lustre.
  - **Lustre** est un système de fichiers distribué libre, généralement utilisé pour **de très grandes grappes de serveurs**.

6 - Autres systèmes de fichiers

- Systèmes de fichiers distribués
  - **HDFS**, *Hadoop Distributed File System :*
    - Système de fichiers distribué au cœur du framework **Apache Hadoop.**
    - Apache Hadoop est un framework open source permettant de stocker et de traiter efficacement de grands ensembles de données (du gigaoctet au pétaoctet).
  - Au lieu d'utiliser un vaste système informatique pour stocker et traiter les données, Hadoop permet de regrouper des machines en clusters pour analyser plus rapidement des ensembles de données volumineux en parallèle.
  - HDFS est un système de fichiers distribué qui fonctionne sur du matériel standard ou bas de gamme.
  - HDFS offre un meilleur débit de données que les systèmes de fichiers traditionnels, en plus d'une tolérance élevée aux pannes et d'une prise en charge native de grands jeux de données.

7 - Fournisseurs et solutions

- Exemples de fournisseurs et solutions
  - **DataCore SANsymphony** :
    - Pionnier dans le domaine, permet de placer une couche de virtualisation évolutive sur les infrastructures de stockage existantes.
  - **HPE GreenLake Cloud** :
    - Solution de cloud hybride qui offre des services de stockage avancés avec sa plateforme *Edge to Cloud*, proposant du stockage à la demande à l'échelle de l'entreprise.
  - **IBM SVC** (*SAN Volume Controller*) :
    - Dispositif de virtualisation du stockage en bloc qui appartient à la famille de produits IBM System Storage. SVC met en œuvre une couche de virtualisation dans un SAN.
    - Propose des fonctionnalités d'auto-tiering, de cache en lecture et écriture, et de thin provisioning.

7 - Fournisseurs et solutions

- Exemples de fournisseurs et solutions
  - Dell **EMC VPLEX** :
    - Implémente une virtualisation du stockage distribuée sur plusieurs sites.
  - **NetApp ONTAP** :
    - Une plateforme offrant une haute disponibilité, des performances élevées et une scalabilité linéaire grâce à sa technologie de clusterisation.
  - Microsoft **Azure Local** **(ex Azure Stack HCI)** :
    - Une solution hyperconvergée permettant de moderniser l'infrastructure des centres de données tout en s'intégrant avec les services cloud Azure.
  - **VMware vSAN** :
    - Solution de virtualisation de stockage entièrement intégrée dans l'infrastructure VMware.

7 - Fournisseurs et solutions

- Exemples de fournisseurs et solutions
  - **Nutanix Cloud Infrastructure** :
    - Solution d'hyperconvergence simplifiant la gestion des infrastructures IT en intégrant calcul, stockage et mise en réseau.
  - **Cisco HyperFlex :** abandonné par Cisco au profit de *Cisco Compute Hyperconverged with Nutanix.*
  - **HPE Alletra dHCI** :
    - Solution d'hyperconvergence conçue pour transformer et simplifier la gestion des infrastructures IT.
  - IBM Spectrum Virtualize for Public Cloud :
    - Permet de répliquer ou de migrer des données sur des systèmes de stockage hétérogènes entre les installations sur site et IBM Cloud, Amazon Web Services ou Microsoft Azure.

7 - Fournisseurs et solutions

- Exemples de fournisseurs et solutions
  - **Starwind HCA** :
    - Une solution de stockage hyperconvergée qui consolide les besoins de stockage et de virtualisation
  - Google **Cloud Storage** :
    - Service web de stockage de données non structurées permettant de stocker et d'accéder à des données sur l'infrastructure de *Google Cloud Platform*.
  - Google **Persistent Disk** :
    - Service de stockage de blocs fiable et performant pour les instances de machines virtuelles.

- Voir : Solutions de Virtualisation de Stockage - Foxeet

Le Cloud Computing

Chapitre 5

1 - Introduction

- Cloud computing : l'informatique dans les nuages :
  - Le cloud computing consiste à héberger et à accéder à des services informatiques (stockage, calcul, logiciels) sur des serveurs distants, plutôt que sur des infrastructures locales.
    Les utilisateurs peuvent accéder à ces ressources depuis n'importe quel appareil connecté à Internet.
  - Le plan de ce chapitre est :
    - 1 - Introduction
    - 2 - Caractéristiques
    - 3 - Cloud privé, public et hybride
    - 4 - Modèles de services
    - 5 - Fournisseurs et solutions

Le Cloud Computing

Chapitre 5

1 - Introduction

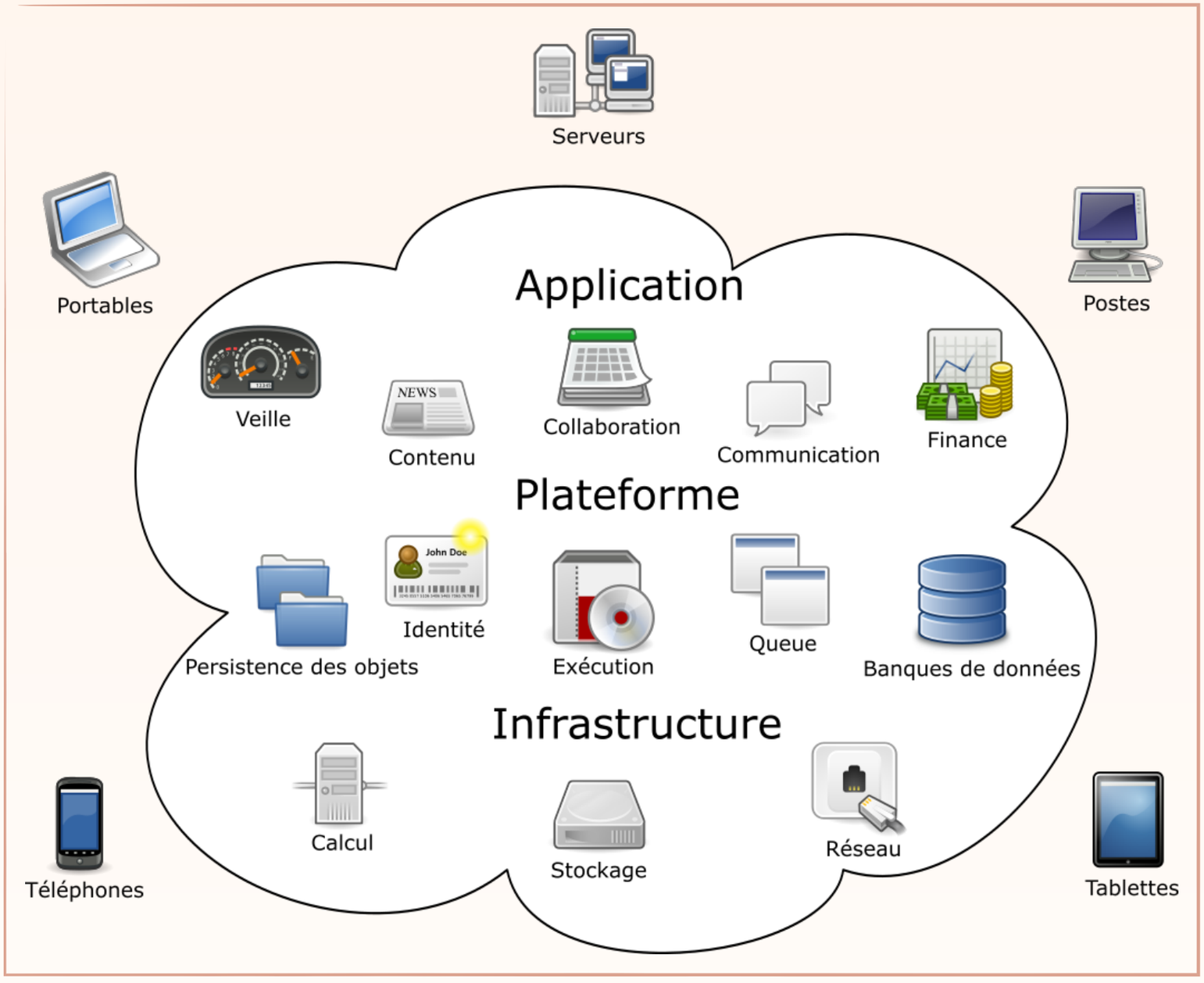


Fig. the Cloud

https://commons.wikimedia.org/wiki/File:Infonuagique.svg

Icons:Tango!-Project
Layout:Sam Johnston
Translation: Urhixidur,

CC BY-SA 3.0,
via Wikimedia Commons

Le Cloud Computing

Chapitre 5

1 - Introduction

- Histoire
  - Années 1980 :
    - La virtualisation (hyperviseur)
    - l'infogérance
    - l'externalisation
  - Dernière décennie :
    - Généralisation d’internet, des FAI, des hébergeurs
    - développement des réseaux à haut débit
    - la location d’application
    - le paiement à l’usage
    - la quête de mobilité...

2 - Caractéristiques

- Cloud computing ≈ Virtualization + pay as you go + self service
  - Virtualisation (***Virtualization***) :
    - Partage des ressources
      - Permet à plusieurs VM ou conteneurs de coexister sur un même serveur physique ;
      - Exemple : Un serveur physique de 64 cœurs peut héberger 16 VM de 4 cœurs chacune.
    - Abstraction sur la localisation
      - Dissocie les ressources logiques (CPU, RAM) de leur support physique réel ;
      - Permet la migration transparente de charges de travail entre datacenters.
    - Élasticité.
      - Capacité à ajuster automatiquement les ressources allouées
      - Exemple : Montée en puissance pendant les pics de trafic, réduction en période creuse.

2 - Caractéristiques

- Cloud computing ≈ Virtualization + pay as you go + self service
  - Cette combinaison crée un écosystème où :
    - La **virtualisation** fournit l'agilité technique,
    - Le **paiement à l'usage** optimise les coûts,
    - Le **self-service** permet l'innovation rapide.
  - Exemple concret : Une startup peut déployer une architecture mondiale en quelques heures, payer exactement ce qu'elle utilise, et ajuster ses ressources en temps réel sans intervention humaine.

2 - Caractéristiques

- Cloud computing ≈ Virtualization + pay as you go + self service, suite
  - ***Pay as you go*** :
    - Le **paiement à l'usage,** soit le paiement des ressources effectivement utilisées (processeur, stockage, réseau) est une révolution économique.
  - Modèle de facturation granulaire
    - Facturation à la seconde pour le calcul (ex : AWS EC2) ;
    - Stockage payé au Go/mois (ex : Azure Blob Storage) ;
    - Réseau facturé au Go transféré.
  - Avantages :
    - Élimination des investissements initiaux (CapEx) ;
    - Optimisation des coûts par alignement parfait usage/dépense ;
    - Exemple : 100 Go de stockage utilisés 15 jours = 50 Go/mois facturés.

2 - Caractéristiques

- Cloud computing ≈ Virtualization + pay as you go + self service, suite
  - ***Self service*** :
    Allocation de ressources d'exécution via une interface web, avec effet en quelques minutes.
  - Autonomie opérationnelle
  - Interface de gestion intuitive
    - Portails web (AWS Console, Azure Portal) ;
    - API REST pour l'automatisation
    - CLI (Command Line Interface) pour les power users

2 - Caractéristiques

- Cloud computing ≈ Virtualization + pay as you go + self service, suite
  - ***Self service*, suite :**
    - Délais de provisionnement réduits
      - Création d'un serveur virtuel en moins de 60 secondes
      - Déploiement d'une base de données managée en 5 clics
      - Mise à l'échelle horizontale automatique via des règles prédéfinies
  - Impact organisationnel :
    - Réduction de la dépendance aux équipes infrastructure
    - Accélération des cycles de développement (DevOps)
    - Expérimentation à moindre risque (environnements éphémères)
  - ***Auto scalling*** **:** Possibilité de définir des fonctions de mise à l'échelle automatique.

3 - Cloud privé, public et hybride

- Les formes de cloud computing
  - les **clouds privés internes**, gérés en interne par une entreprise pour ses besoins. Avantages :
    - L'entreprise a un **contrôle total** sur l'infrastructure et les données.
    - Sécurité renforcée : Les données sensibles restent dans l'environnement de l'entreprise.
    - Personnalisation : L'infrastructure peut être adaptée précisément aux besoins spécifiques de l'entreprise.
  - Cependant, cette option nécessite des investissements importants en matériel, logiciels et expertise interne.
  - Voir :
    - Cloud privé : les 12 plateformes techniques de 2025 - LeMagIT

- Les formes de cloud computing, suite
  - Les **clouds privés externes**, dédiés aux besoins propres d'une seule entreprise, mais dont la gestion est externalisée chez un prestataire. Avantages :
    - Réduction des coûts d'infrastructure : L'entreprise n'a pas besoin d'investir dans son propre matériel.
    - Expertise spécialisée : Le prestataire apporte son savoir-faire en gestion de cloud.
    - Flexibilité : L'entreprise peut ajuster les ressources selon ses besoins.

3 - Cloud privé, public et hybride

- Les formes de cloud computing, suite
  - Les **clouds publics**, gérés par des fournisseurs spécialisés qui louent leur services à de nombreuses entreprises. Avantages :
    - Économies d'échelle : Les coûts sont répartis entre de nombreux clients.
    - Scalabilité : Les ressources peuvent être ajustées rapidement en fonction des besoins.
    - Large gamme de services : Les fournisseurs proposent souvent une variété d'outils et de services intégrés.
  - Cependant, les entreprises ont moins de contrôle sur l'infrastructure et doivent être attentives aux questions de sécurité et de conformité.

3 - Cloud privé, public et hybride

- Les formes de cloud computing, suite
  - un **cloud hybride** consiste à associer un ou plusieurs clouds publics à un cloud privé. Un système d'orchestration connecte et ordonnance les ressources de cloud privé avec des ressources de cloud public. Avantages :
    - Optimisation des coûts : Les charges de travail peuvent être réparties entre cloud privé et public selon les besoins.
    - Conformité : Les données sensibles peuvent rester dans le cloud privé tout en profitant des ressources du cloud public pour d'autres tâches.
    - Résilience : La redondance entre cloud privé et public améliore la continuité des activités.
    - Optimisation des performances : L'orchestrateur peut répartir les charges de travail de manière optimale entre les différents clouds.

4 - Les modèles de service

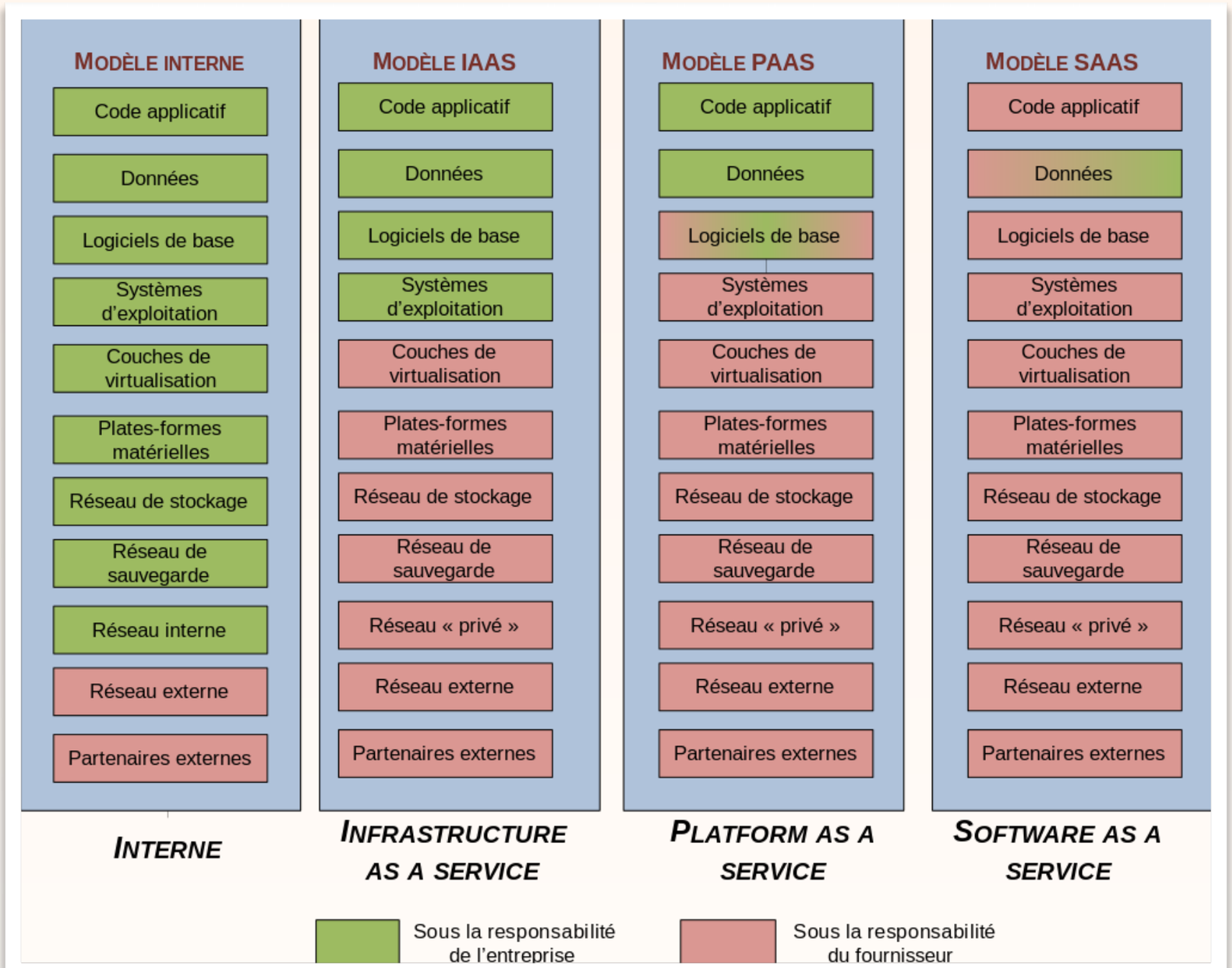


Fig-Les modèles de cloud
https://fr.wikipedia.org/wiki/Cloud_computing#/media/File:Cloud_Computing_-_les_différents_modèles_de_service.svg

4 - Les modèles de service

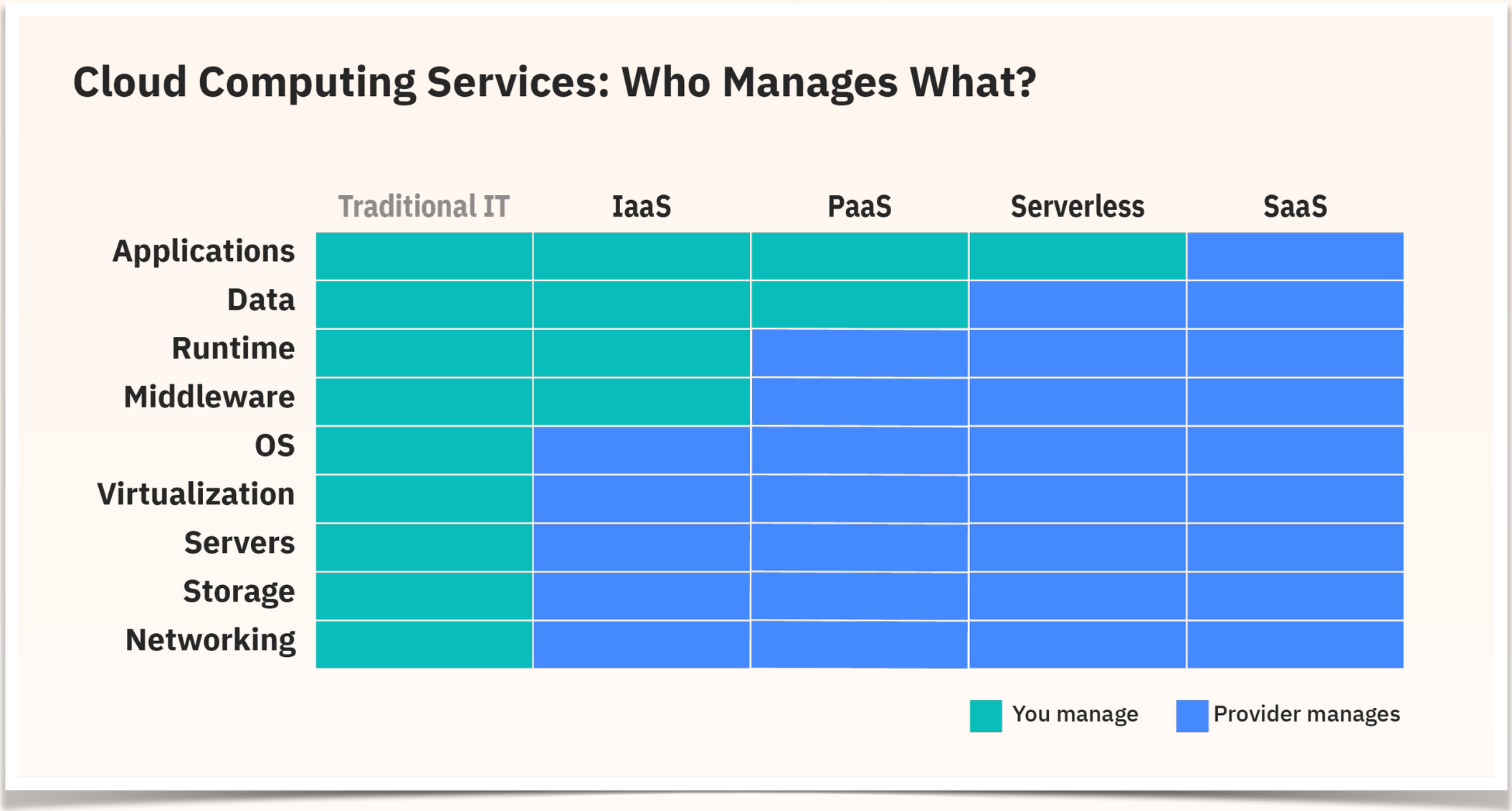
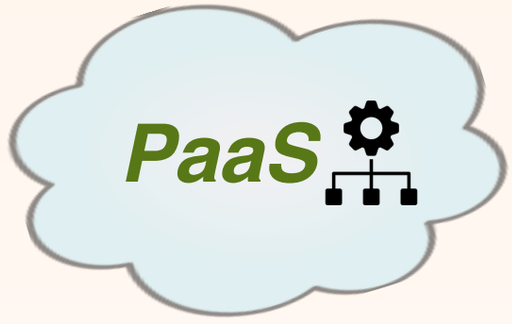


Fig-Les modèles de cloud - D'après : What is Cloud Computing? | IBM

4 - Les modèles de service

- **Infrastructure as a Service** (IaaS)
  - L'infrastructure est virtualisée ;
  - Le fournisseur Cloud fournit et maintient la virtualisation, le matériel serveur, le stockage et les réseaux.
    - Ces services sont facturés selon votre utilisation.
  - Vous êtes responsable du système d'exploitation ainsi que des données, applications, solutions de middleware et environnements d'exécution.
  - Inconvénients :
    - potentiels problèmes de sécurité chez le fournisseur,
    - fiabilité du service car le fournisseur partage des ressources de l'infrastructure entre plusieurs clients.
  - Exemples de fournisseur : AWS, Microsoft Azure et Google Compute Engine.

4 - Les modèles de service



- **Platform as a Service** (PaaS)
  - La plateforme est louée par le client ;
  - Le fournisseur héberge le matériel et les logiciels sur sa propre infrastructure et met à disposition de l'utilisateur une plateforme via Internet, sous la forme d'une solution intégrée, d'une pile de solutions ou d'un service.
  - Vous écrivez le code, créez et gérez vos applications, le tout, sans avoir à vous préoccuper des mises à jour logicielles ou de la maintenance du matériel. L'environnement de développement et de déploiement vous est fourni.
  - Exemples : AWS Elastic Beanstalk, Google App Engine, Heroku et Red Hat OpenShift.

4 - Les modèles de service

- **Software as a Service** (SaaS)

  - Logiciel à la demande : utilisation de services proposés en abonnement ou payés à la demande.
  - Le fournisseur gére, met à jour et propose une application complète par l'intermédiaire d'un navigateur web.
    - Aucun logiciel n'est installé sur les machines du client et l'accès au programme est plus fluide et plus fiable.
    - Cela permet aux utilisateurs de travailler de n’importe où.

- **Software as a Service** (SaaS), suite

  - Inconvénients : Bien qu'il limite les opérations de maintenance, le modèle SaaS réduit le niveau de contrôle et peut nuire à la sécurité et aux performances.
  - Exemples : Dropbox, Salesforce, Google Apps et Red Hat Insights. Netflix, Disney+, Apple TV, Prime Video, YouTubeTV, World of Warcraft, etc.
  - Voir :
    - Qu'est-ce que le SaaS (Software en tant que service) ? - Oracle

4 - Les modèles de service

- **Serverless**
  - Construction et exécution d'applications sans gérer les serveurs sous-jacents.
  - Le fournisseur Cloud gère le provisionnement et la surveillance des serveurs.
  - Le développeur se concentre sur le code de l'application et le déploiement de ses API.
  - Le terme "**serverless**" est trompeur car les serveurs sont toujours impliqués.
    - Mais du point de vue du développeur, il n'y a pas de serveurs visibles à gérer.
  - Le modèle **serverless** est proche du modèle PaaS, avec, pour le serverless :
    - Scalabilité automatique, sans configuration ni intervention du développeur ;
    - Facturation basée sur l'utilisation réelle ;
    - Offre moins de contrôle sur l'environnement d'exécution, mais simplifie considérablement la gestion.

4 - Les modèles de service

- **Serverless**
  - Le modèle serverless comprend des plate-formes du type **FaaS** et **BaaS.**
  - **Faas,** *Function-as-a-Service :*
    - Ce modèle permet aux développeurs d'exécuter de petites parties de code en périphérie du réseau.
    - Grâce à l'approche FaaS, ces derniers peuvent développer une architecture modulaire, afin de créer une base de code plus évolutive, sans se soucier de mobiliser des ressources pour l'entretien de l'infrastructure back-end sous-jacente.
  - **BaaS**, *Backend-as-a-Service :*
    - Avec ce modèle, un fournisseur cloud propose des services back-end afin de permettre aux développeurs de se concentrer sur l'écriture de leur code front-end. Par exemple :
      - le stockage de données, la gestion des bases de données, l'authentification des utilisateurs, la mise à jour à distance, etc.

4 - Les modèles de service

- **Serverless**
  - Exemples d'offres serverless :
    - Firebase (Alphabet),
    - AWS Lambda et autres ,
    - Cloudflare Workers,
    - Microsoft Azure serverless.
- Voir :
  - Architectures serverless : une révolution pour le développement web - Johnstyle
  -

4 - Les modèles de service

- Autres modèles :
  - XaaS, *Anything as a Service* ; Tout et n'importe quoi en tant que Service.
  - AaaS, *Analytics as a Service* ;
  - DaaS, *Desktop as a Service* ; Exemple : Citrix
  - FaaS, Functions as a Service ; Exemple : Google Cloud Functions
  - STaaS, Storage as a Service ;
  - CaaS, Containers as a Service ; Exemple : Portainer.
  - DBaaS, Database as a Service ; Exemple : Oracle Database.
  - AaaS, Authentification as a service ; Exemple : Duo Security

---

- Voir : XaaS : Le modèle Anything as a Service (avec 10 exemples de catégories) - Kinsta

5 - Fournisseurs et solutions

- IaaS, Infrastructure as a Service
  - Amazon EC2, *Elastic Compute Cloud*
    - Service d'hébergement cloud évolutif proposé par Amazon Web Services (AWS)
    - EC2 utilise la virtualisation Xen. Chaque VM, appelée « instance », fonctionne comme un serveur virtuel privé.
  - Caractéristiques principales :
    - Flexibilité : EC2 offre une capacité de mise à l'échelle à la demande, permettant d'augmenter ou de réduire rapidement les ressources en fonction des besoins.
    - Variété d'instances : Plus de 750 types d'instances sont disponibles, avec différentes configurations de CPU, mémoire, stockage et capacités réseau.
    - Contrôle total : Les utilisateurs peuvent gérer leurs instances, configurer la sécurité, les réseaux et le stockage selon leurs besoins.
    - Coût optimisé : Le modèle de tarification est basé sur l'utilisation réelle, ce qui permet de réduire les coûts matériels.

5 - Fournisseurs et solutions

- IaaS, Infrastructure as a Service, suite
  - Google Compute Engine
    - Compute Engine propose des VM avec un hyperviseur KVM, des systèmes d'exploitation pour Linux et Windows, ainsi que des options de stockage local et durable.
    - Chaque instance Google Compute Engine démarre avec une ressource disque appelée *persistent disk*. Ce disque persistant fournit l'espace disque pour les instances et contient le système de fichiers racine à partir duquel l'instance démarre.
    - Caractéristiques principales :
      - Évolutivité rapide et flexible ; Large choix de configurations de VM ;
      - Performance et fiabilité élevées avec une disponibilité garantie de 99,9% ;
      - Tarification à la seconde et options de réduction pour usage prolongé ;
      - Fonctionnalités réseau avancées, incluant VPC et équilibrage de charge global ;
      - Mesures de sécurité robustes et intégration facile avec d'autres services Google Cloud

5 - Fournisseurs et solutions

- IaaS, Infrastructure as a Service, suite
  - Microsoft Azure
    - Les principaux éléments de l'offre IaaS d'Azure sont :
      - Déploiement de VMs Windows et Linux hautement personnalisables et adaptables ;
      - Stockage cloud : Solutions de stockage sécurisées et extensibles pour les fichiers, bases de données et disques ;
      - Mise en réseau : VPN, CDN et connexions ExpressRoute ;
      - Sécurité : Protections intégrées pour les données et applications, y compris la gestion sécurisée des identités et des accès (IAM) ;
      - Évolutivité ; Tarification flexible ;
      - Intégration : Compatibilité aisée avec d'autres outils Microsoft et services Azure, comme Power Platform et les services d'analyse de données.

5 - Fournisseurs et solutions

- IaaS, Infrastructure as a Service, suite
  - OpenStack
    - IAAS Open source, qui a rejoint la Fondation Linux en 2025.
    - Ensemble de logiciels open source permettant de déployer des infrastructures.
    - Architecture modulaire composée de plusieurs projets corrélés (Nova, Swift, Glance…)
    - www.openstack.org

5 - Fournisseurs et solutions

- PaaS, Platform as a Service
  - AWS, *Amazon Web Services*, offre des services PaaS comme *AWS Elastic Beanstalk* et *AWS Lambda*.
  - Microsoft Azure propose une gamme complète de services PaaS, se positionnant comme l'un des leaders du marché.
  - Google Cloud Platform fournit des services PaaS tels que Google App Engine.
  - Heroku est une plateforme PaaS populaire, particulièrement auprès des développeurs.
  - Oracle Cloud : Offre des solutions PaaS avec sa suite de services cloud.
  - IBM Cloud Foundry.
  - Red Hat OpenShift : Une plateforme PaaS basée sur Kubernetes.

5 - Fournisseurs et solutions

- SaaS, Software as a Service
  - Google Workspace (ex G Suite)
    - Inclus : Gmail, Agenda, Docs, Drive, Forms, Sites, etc.
  - Microsoft 365
    - Il est constitué de la suite Office (Word, Excel, PowerPoint, Outlook, OneNote, Publisher et Access), ainsi que des services en ligne tels que OneDrive, Exchange Online, SharePoint Online, Teams et Yammer.
  - Salesforce , dominant le secteur de la gestion de la relation client (CRM *)
  - Adobe : Troisième fournisseur important de solutions SaaS

* CRM : *Customer Relationship Management*

5 - Fournisseurs et solutions

- SaaS, Software as a Service
  - Atlassian : Il propose des outils essentiels pour le développement de logiciels et la gestion de projet, notamment *Jira* et *Confluence*. Cf. atlassian.com/fr
  - Dassault Systèmes : Leader français du marché des éditeurs de logiciels, reconnue pour ses logiciels de conception par ordinateur et ses solutions de jumeaux numériques. Cf. 3DEXPERIENCE
  - Criteo : entreprise française de reciblage publicitaire sur internet

* CRM : *Customer Relationship Management*

6 - Conclusion

- Voir :
  - 11 Avantages du Cloud Computing en 2025 - Kinsta
  - Qu'est-ce que le cloud ? | Définition du cloud - Cloudflare
  - Administration : 5 outils Open source pour surveiller le cloud - LeMagIT
  - Cloud computing - OVHcloud

Big Data & Hadoop

Chapitre 6

1 - Big Data - définition et caractéristiques

- Le Big Data
  - Le Big Data désigne des **ensembles de données massifs et complexes,** caractérisés par **les 5V :**
    - **Volume :** Traitement de données atteignant des pétaoctets (Près de 200 Zo de données numériques créées dans le monde en 2025)
    - **Vitesse :** Collecte et analyse en temps réel (flux sociaux, IoT)
    - **Variété :** Données structurées, semi-structurées et non structurées (texte, images, logs)
    - **Véracité :** Enjeux de fiabilité et qualité des données
    - **Valeur :** Extraction d'informations exploitables (*actionnables insights*)

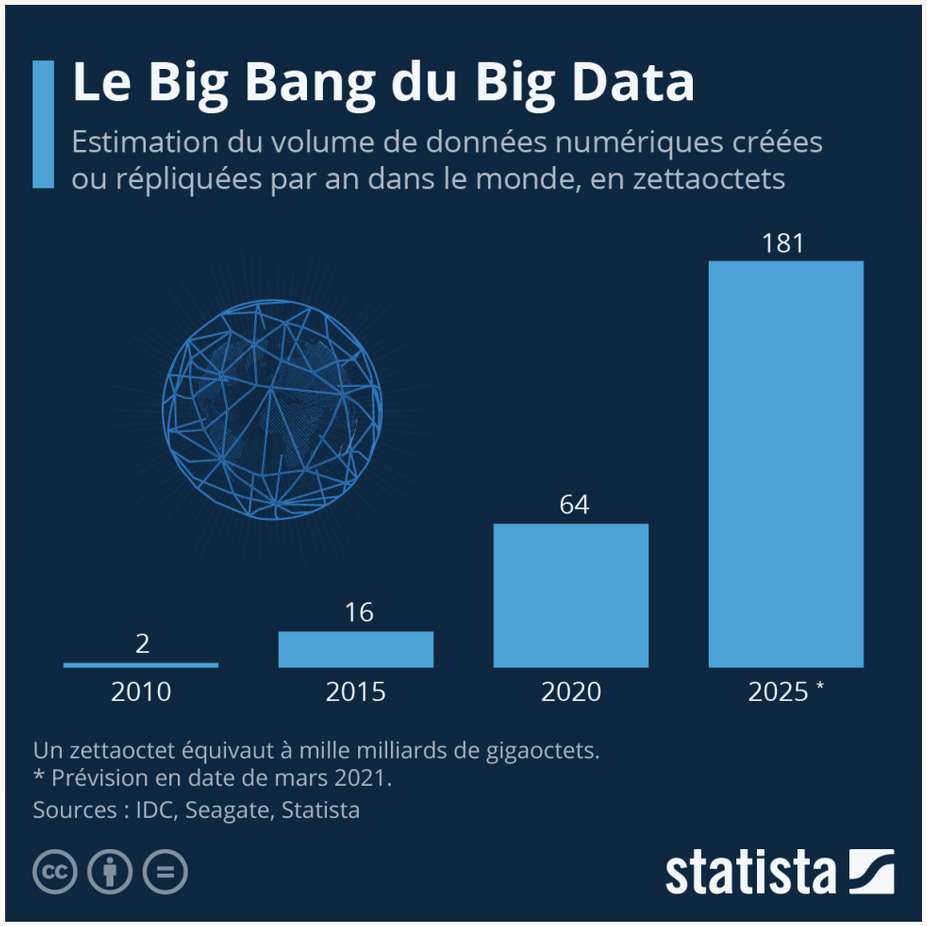


Fig 6.1 - Le Big Bang du Big Data

1 - Big Data - définition et caractéristiques

- Stockage et données
  - *Data store* ; Dépôt de données :
    - Terme **générique** qui désigne l'ensemble des bases de données, des systèmes de fichiers ou de répertoires.
  - *Data warehouse* ; entrepôt de données ;
    - Type particulier de base de données.
  - *Data lake* ; lac de données :
    - Stockage de données massives, en clusters, où les données sont gardées **dans leur formats natifs**, dans des base NoSQL par exemple.
  - *Dataviz* ; **visualisation** de données.

1 - Big Data - définition et caractéristiques

- Stockage et données, suite
  - *Data mining* ; exploration de données :
    - Consiste à rechercher des relations qui n'ont pas encore été identifiées.
    - Processus utilisé pour extraire des données utilisables d’un plus grand ensemble de données brutes.
  - *Open data* ; données ouvertes :
    - Données numériques dont l'accès et l'usage sont laissés libres aux usagers.
  - BI, *Business Intelligence* ; informatique décisionnelle :
    - Processus d'analyse des données qui vise à doper les performances métier en aidant à prendre des décisions plus avisées.
  - Voir aussi :
    - La virtualisation de stockage, chapitre 4.
    - www.ovhcloud.com/fr/learn/#big-data

1 - Big Data - définition et caractéristiques

- Applications sectorielles
  - Marketing : Personnalisation des campagnes via l'analyse comportementale
  - Santé : Médecine prédictive et gestion épidémiologique
  - Finance : Détection de fraude et évaluation de risque
  - Industrie : Maintenance prédictive des équipements
- Défis majeurs
  - Stockage : Coûts infrastructurels exponentiels
  - Traitement : Limites des architectures traditionnelles
  - Éthique : Confidentialité des données et RGPD

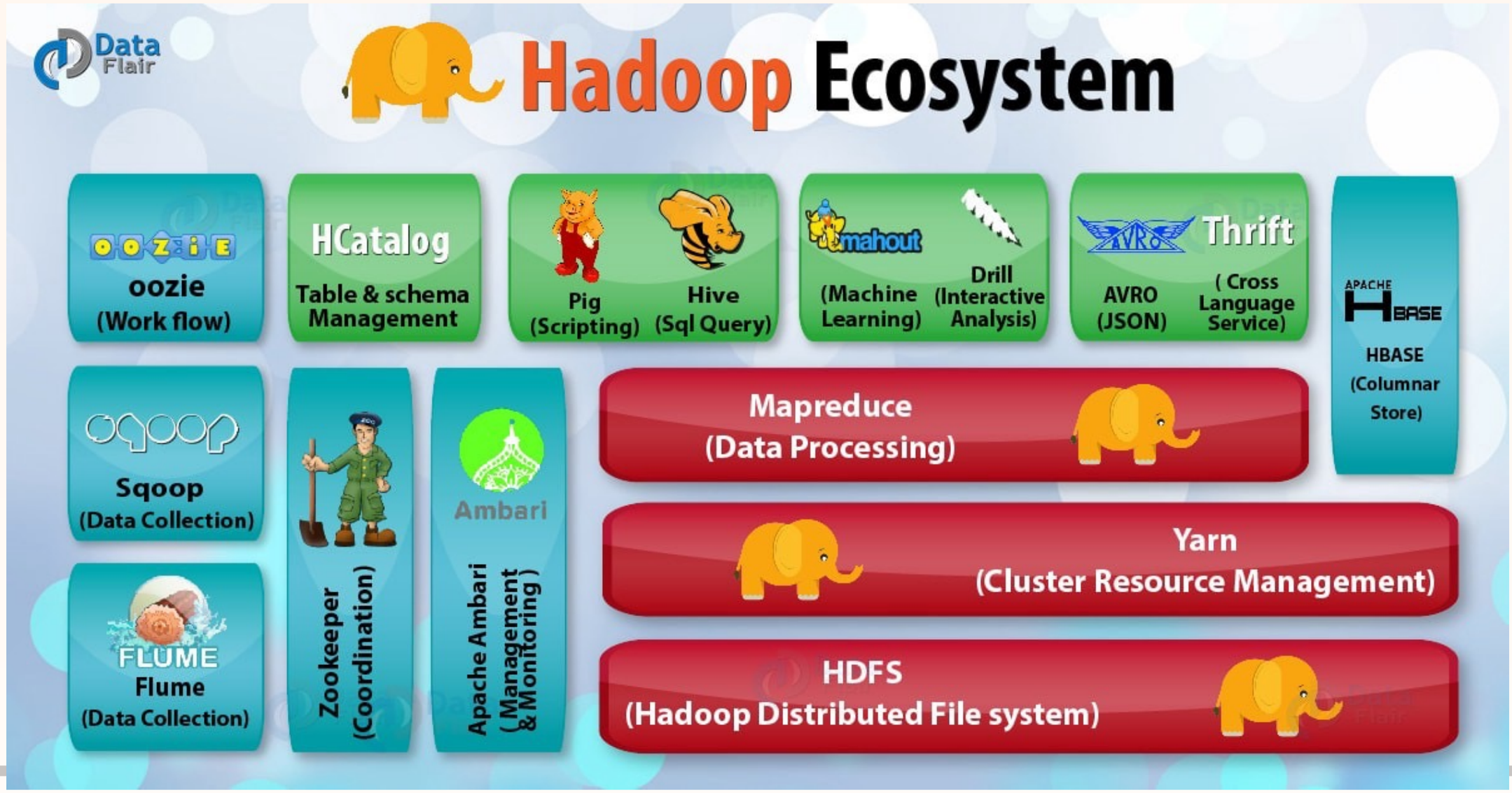
2 - Apache Hadoop : Architecture et Fonctionnement

Principes fondamentaux

- Apache Hadoop est un framework open-source conçu pour :
  - Stocker des données massives via HDFS, *Hadoop Distributed File System* ;
  - Rechercher et restituer des données ;
  - Traiter en parallèle avec *MapReduce* ;
  - Gérer les ressources via YARN, *Yet Another Resource Negociator.*
- Contexte
  - Hadoop a été créé par Doug Cutting et fait partie des projets de la fondation logicielle Apache depuis 2009.
  - Hadoop a été inspiré par la publication de MapReduce, GoogleFS et BigTable de Google.
  - La version actuele d'Apache Hadoop est 3.4.1 (octobre 2024).

2 - Hadoop : Architecture et Fonctionnement

- L'écosystème Apache Hadoop



2 - Hadoop : Architecture et Fonctionnement

- Principes fondamentaux, suite
  - Les principaux modules de Hadoop
    - **Hadoop Common** : un ensemble d'utilitaires et de bibliothèques permettant de piloter d'autres modules Hadoop.
    - **HDFS**, *Hadoop Distributed File System* : système de fichiers de Hadoop, conçu pour stocker d'importants volumes de données structurées ou non sur un ensemble de serveurs.
    - **Haddop MapReduce** permet d'exécuter en parallèle des calculs sur de grandes quantités de données. Il comprend deux fonctions Map et Reduce exécutées l'une après l'autre.
    - **YARN**, *Yet Another Resource Negociator* est le gestionnaire de ressources et de cluster de Hadoop. Appelé MapReduce 2.0, YARN est apparu dans la version 2 de Hadoop.

2 - Hadoop : Architecture et Fonctionnement

- Architecture technique
  -

Composant	Rôle	Caractéristiques
NameNode	Gère les métadonnées HDFS	Point unique de défaillance
DataNode	Stocke les blocs de données et gère l'accès disque des données	Réplication automatique
JobTracker	Orchestre les tâches MapReduce	Assignation aux TaskTrackers
NodeManager	Gère les ressources par nœud	Partie de YARN

2 - Hadoop : Architecture et Fonctionnement

- Processus MapReduce
  - Framework logiciel en Java permettant de développer des programmes exécutables de manière distribués grâce à l'utilisation de l'algorithme **MapReduce** développé par Google.
  - MapReduce ne permet de traiter que des problèmes qui peuvent se décomposer en de multiples tâches parallèles.
  - Phase **Map** :
    - Découpage des données en blocs distribués ;
    - Traitement parallèle sur les nœuds esclaves.
  - Phase **Reduce** :
    - Agrégation des résultats intermédiaires ;
    - Production du résultat final.

2 - Hadoop : Architecture et Fonctionnement

- MapReduce
  -

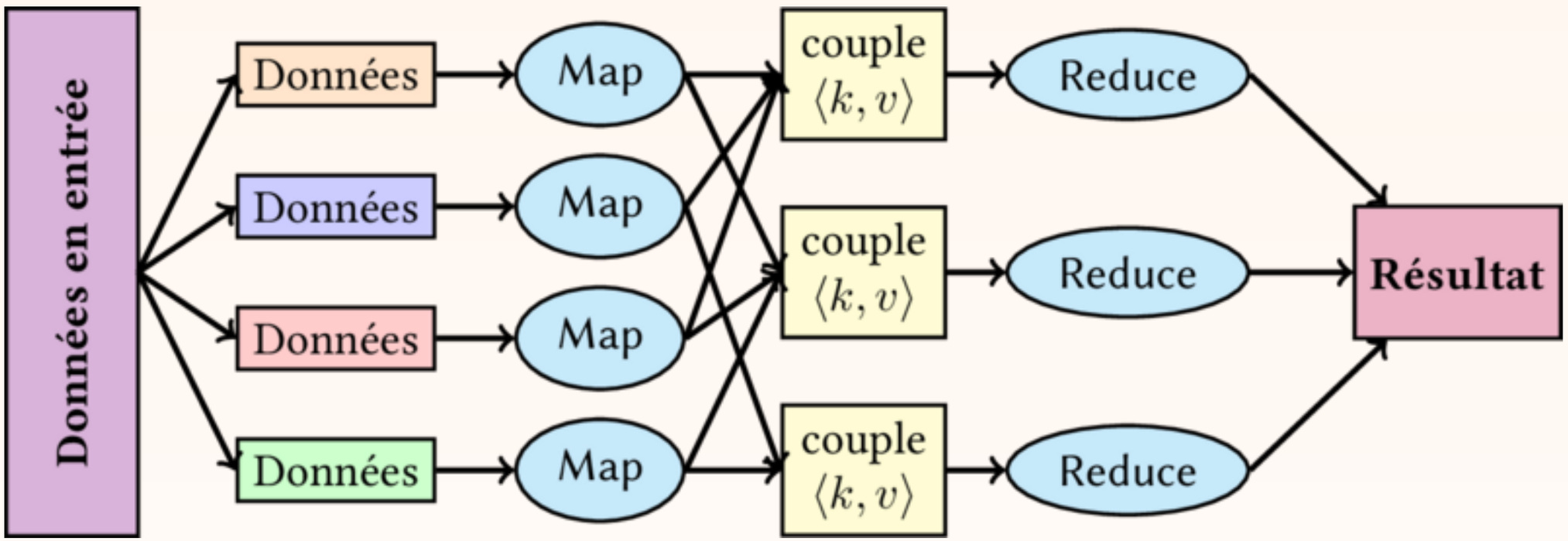


Fig 6.3 - Schéma du fonctionnement du MapReduce

- MapReduce - exemple 1
  - Compter les occurrences des mots d'une collection de documents
    - Entrée : < nom, contenu >
    - Map : pour chaque mot du contenu -> <mot, 1>
    - Shuffle : regroupe tous les <mot, 1> pour un mot donné
    - Reduce : <mot, nombre> -> <mot, total>
    - Sortie : collection de <mot, occurrences>

- MapReduce - exemple 2
  - Déterminer les documents pointant vers une URL
    - (Résolution inverse de liens)
  - Entrée : collection de < URL, contenu >
  - Map : pour chaque lien du contenu -> < URL lien, URL source>
  - Shuffle : regroupe tous les <URL lien, URL source> pour une URL donnée
  - Reduce : <URL lien, URL source> -> <URL lien, liste de sources>
  - Sortie : collection de <URL, liste de sources>

3 - Hadoop : Avantages et cas d'usage

- Atouts clés d'Hadoop
  - Scalabilité linéaire : Ajout de nœuds sans refonte
  - Tolérance aux pannes : Réplication 3x des données par défaut
  - Économique : Matériel standard x86
  - Polyvalence : Supporte tous formats de données

- Applications concrètes
  - Netflix : Analyse de 500M heures de streaming/jour pour les recommandations
  - Banques : Détection de transactions frauduleuses en <100ms
  - Recherche génomique : Traitement de séquences ADN massives

3 - Hadoop : Avantages et cas d'usage

- Apache Spark
  - Solution pour écrire simplement des applications distribuées.
  - Spark propose des bibliothèques de traitement classique.
  - Sa performance est remarquable ; il peut travailler sur des données sur disque ou des données chargées en RAM.
  - Il est plus jeune que MapReduce mais il dispose d'une communauté énorme.
  - C'est donc une solution qui s'avère être le successeur de MapReduce.
  - Voir : https://www.youtube.com/watch?v=ymtq8yjmD9I

3 - Hadoop : Avantages et cas d'usage

- Les acteurs pour Apache Hadoop :
  - **Fondation Apache** ;
  - **Cloudera**, start-up de la Silicon Valley ; développement de logiciels de Big Data basées sur le framework Hadoop ;
  - **Hortonworks**, société de logiciels informatique Californienne ; développement et soutien de Hadoop. Elle a fusionné avec Cloudera en 2018 ;
  - Les **GAFAM**.

4 - Conclusion

- Le Big Data transforme radicalement la prise de décision organisationnelle, tandis qu'Hadoop fournit l'infrastructure nécessaire pour exploiter ces données à grande échelle.
- Malgré des défis persistants en sécurité et gouvernance, leur synergie continue de driver l'innovation sectorielle.
- L'évolution vers des architectures cloud-native et l'intégration croissante de l'IA laissent présager de nouvelles avancées dans ce domaine.
- Voir :
  - Tutoriel d'introduction à Apache Hadoop - Mickael BARON, developpez.com
  - MapReduce et Hadoop - Alexandre Denis, Inria Bordeaux
  - Big data - Centre d’apprentissage, OVHcloud
  - Apache Hadoop - Installation, modules et projets d’Apache

4 - Conclusion

- Voir :
  - Apache Hadoop - Installation, modules et projets d'Apache
  - Apache Spark
  - Tutoriel d'introduction à Apache Hadoop - Mickael BARON, developpez.com
  - MapReduce et Hadoop - Alexandre Denis, Inria Bordeaux
  - Big data - Centre d'apprentissage, OVHcloud
  - Un déluge de données - Interstices
  - Définition : Qu'est-ce que le Big Data ? - Le Big Data
  - Hadoop – Tout savoir sur la principale plateforme Big Data - Le Big Data
  - Hadoop : qu'est-ce que c'est et comment apprendre à l'utiliser ? - DataScientest
  - Qu'est-ce qu'Apache Hadoop dans Azure HDInsight ?