

---

---

**UNIVERSITÉ D'EL OUED**

Master I  
Option : Informatique

# **Cloud Computing**

Présentée par **YAGOUB Mohammed Amine**

# PLAN DE COURS

- **Partie I : Grid computing et Virtualisation**
- Introduction générale
- Grid computing
  - Introduction et Concepts de base
    - Paradigmes du calcul parallèle
    - Les systèmes distribués de calcul
  - Historique et Définitions
  - Caractéristiques des grilles de calcul
  - Principes de base du grid computing
  - Les différents types de grid Computing
  - Domaines d'application des grilles informatiques

# PLAN DE COURS

- **Partie I : Grid computing et Virtualisation**
  - Introduction générale
  - Grid computing
  - Virtualisation
    - Définition
    - Les domaines de la virtualisation
    - Les différents types de virtualisation
      - La virtualisation complète
      - La paravirtualisation
      - La virtualisation assistée par le matériel
      - Le cloisonnement
    - Les principales solutions
      - Xen
      - VMWare ...
    - Avantages & inconvénients de la virtualisation

# PLAN DE COURS

- **Partie 2 : Cloud computing**
- Définition de cloud
- Les principes de cloud
- Caractéristiques du Cloud Computing
  - Aspect technique
    - Service doit être accessible via un réseau
    - Ressources partagées
    - Service mesuré
    - ...
  - Aspect qualitatif
    - Élasticité rapide
    - Simplicité, Flexibilité, Fiabilité et tolérance aux pannes
    - ...
  - Aspect économique
    - Accès libre à la demande
    - Paiement à l'usage
    - ...

# PLAN DE COURS

- **Partie 2 : Cloud computing**
- Définition de cloud
- Les principes de cloud
- Caractéristiques du Cloud Computing
- Types de services du Cloud
- Modèles de déploiement du Cloud Computing
  - Cloud privé
  - Cloud public
  - ...
- Avantages et inconvénients du Cloud Computing
- Différences entre Grilles et Clouds
- Principaux fournisseurs de cloud
  - Amazon Web Services
  - Google Cloud Plateform
  - ...

# PLAN DE COURS

- **Partie 2 : Cloud computing**
- Définition de cloud
- Les principes de cloud
- Caractéristiques du Cloud Computing
- Types de services du Cloud
- Modèles de déploiement du Cloud Computing
- Avantages et inconvénients du Cloud Computing
- Différences entre Grilles et Clouds
- Les principaux fournisseurs de cloud
- Solutions libres de Cloud
  - Openstack
  - OpenNebula
  - ...
- Stockage dans le cloud
  - Stockage de fichier dans le cloud
  - Les bases de données dans les cloud

# PLAN DE COURS

- **Partie 2 : Cloud computing**
- Définition de cloud
- Les principes de cloud
- Caractéristiques du Cloud Computing
- Types de services du Cloud
- Modèles de déploiement du Cloud Computing
- Avantages et inconvénients du Cloud Computing
- Différences entre Grilles et Clouds
- Les principaux fournisseurs de cloud
- Solutions libres de Cloud
- Stockage dans le cloud
- Sécurité dans le Cloud Computing

## MODE D'ÉVALUATION

50 % Examen + 50 % Contrôle Continu

## INTRODUCTION GÉNÉRALE

Aujourd'hui, dans le domaine de l'information, les plateformes imposent des solutions de modélisation.

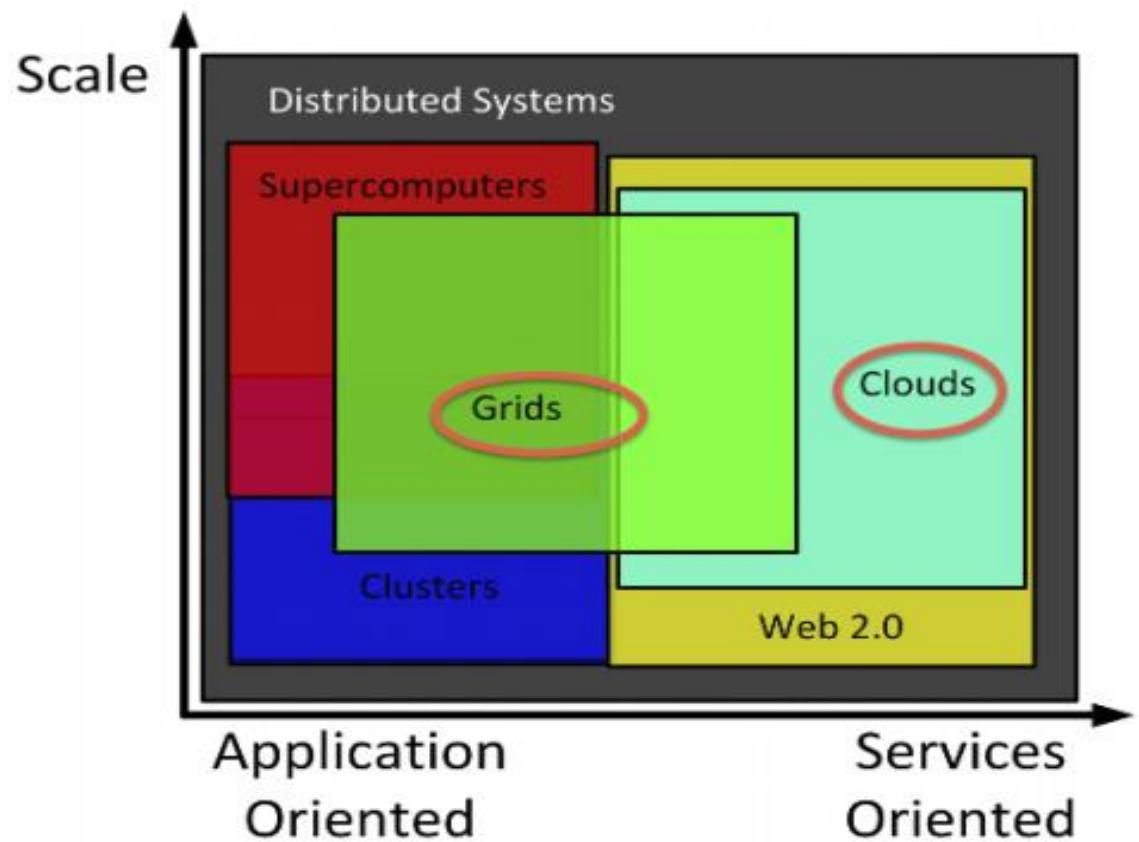
➤ Nous obligent à avoir :

- une énorme capacité de stockage
- une puissance de calcul énorme



# INTRODUCTION GÉNÉRALE

- Solutions Hardware
- Solutions Software



## INTRODUCTION GÉNÉRALE

### Solutions Hardware

- **Cluster**
  - Ensemble d'ordinateurs homogènes
  - Travailleraient en commun
  - Dans un même lieu

## INTRODUCTION GÉNÉRALE

### Solutions Hardware

- **Supercalculateur**
- Le calcul est de haute performance (High Performance Computing HPC)
- On mesure en Mips (Millions of Instructions Per Second) ou MFlops (Million of Floating point operation per second)
- Le contrôle y est centralisé et les processeurs sont reliés par un bus (ou autre type de réseau d'interconnexion).

## INTRODUCTION GÉNÉRALE

### Solutions Hardware

#### ➤ Supercalculateur

Exemples :

- 65 TFlops pour le NEC SX-8 en 2004,
- 839 TFlops pour le NEC SX-9 en 2011,
- 141 TFlops par noeud pour NEC SX ACE en 2014
- Le supercalculateur le plus puissant du monde est à nouveau Frontier. Ce système HPE Cray EX atteint désormais une puissance de calcul de 1,194 exaflop/s et totalise 8 730 112 cœurs. Il est équipé de processeurs AMD EPYC de 3e génération, d'accélérateurs AMD Instinct 250X et d'une interconnexion Slingshot-II. L'ensemble est optimisé pour le calcul intensif (HPC), l'intelligence artificielle (IA) et l'entraînement de ses modèles pour les besoins du laboratoire ORNL (Oak Ridge National Laboratory) du Département américain de l'énergie (DOE).

## INTRODUCTION GÉNÉ

### Solutions Hardware

#### ➤ Supercalculateur

Exemples :

- Le supercalculateur le plus puissant au monde possède désormais une puissance de calcul de 1 exaflop (10<sup>18</sup> opérations par seconde) et totalise 8 700 000 processeurs. Il est composé d'accélérateurs AMD Instinct et de processeurs Intel Xeon. Il est nommé Frontier et est installé au laboratoire ORNL (Oak Ridge National Laboratory) à Tennessee.



## INTRODUCTION GÉNÉRALE

### Solutions Hardware

- Internet
  - Pourquoi étions-nous obligés de tout localiser au même endroit ?
  - Pourquoi les machines devaient être homogènes ?

# INTRODUCTION GÉNÉRALE

## Solutions Hardware

- L'idée du « **Grid** » puis « **Cloud** »



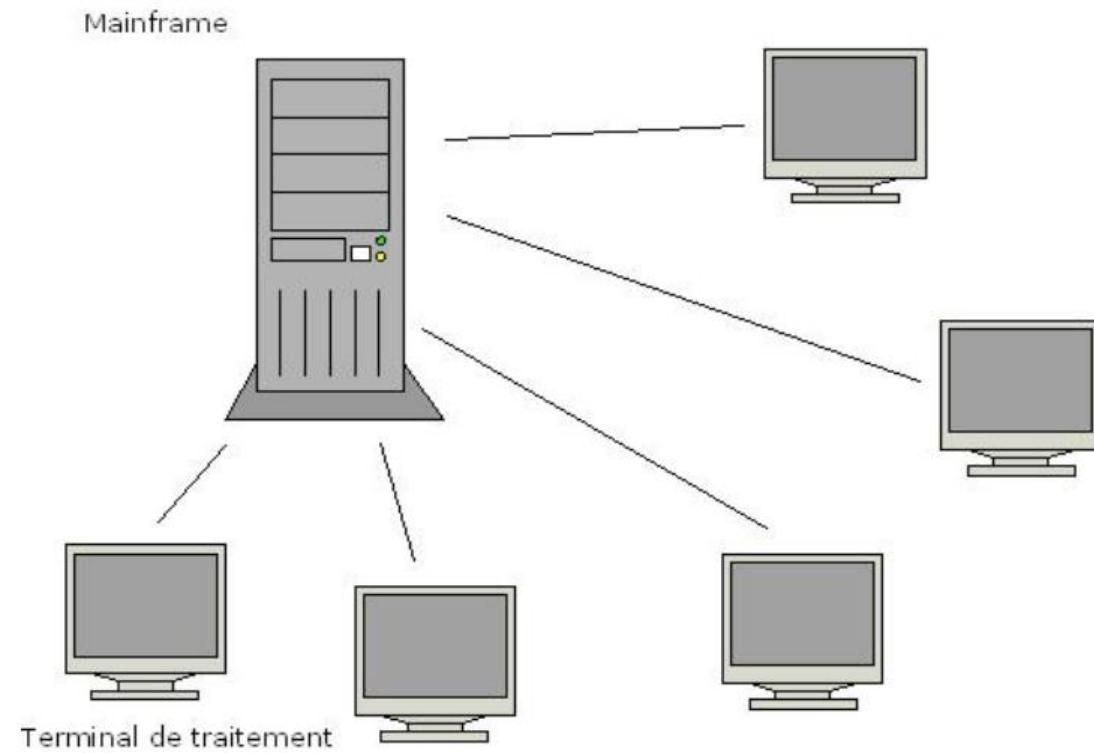
Google Data centers



# INTRODUCTION GÉNÉRALE

## Solutions Software

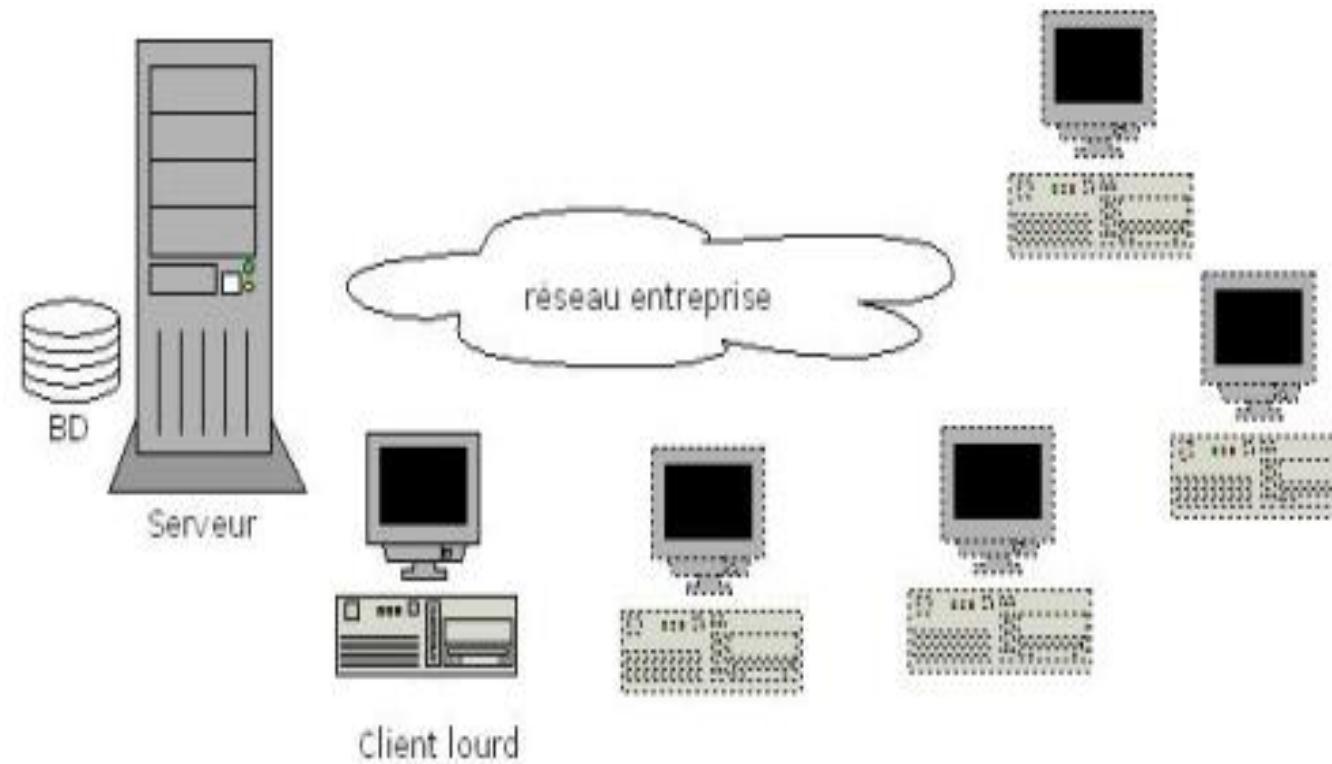
### ➤ **Architectures Mainframe**



# INTRODUCTION GÉNÉRALE

## Solutions Software

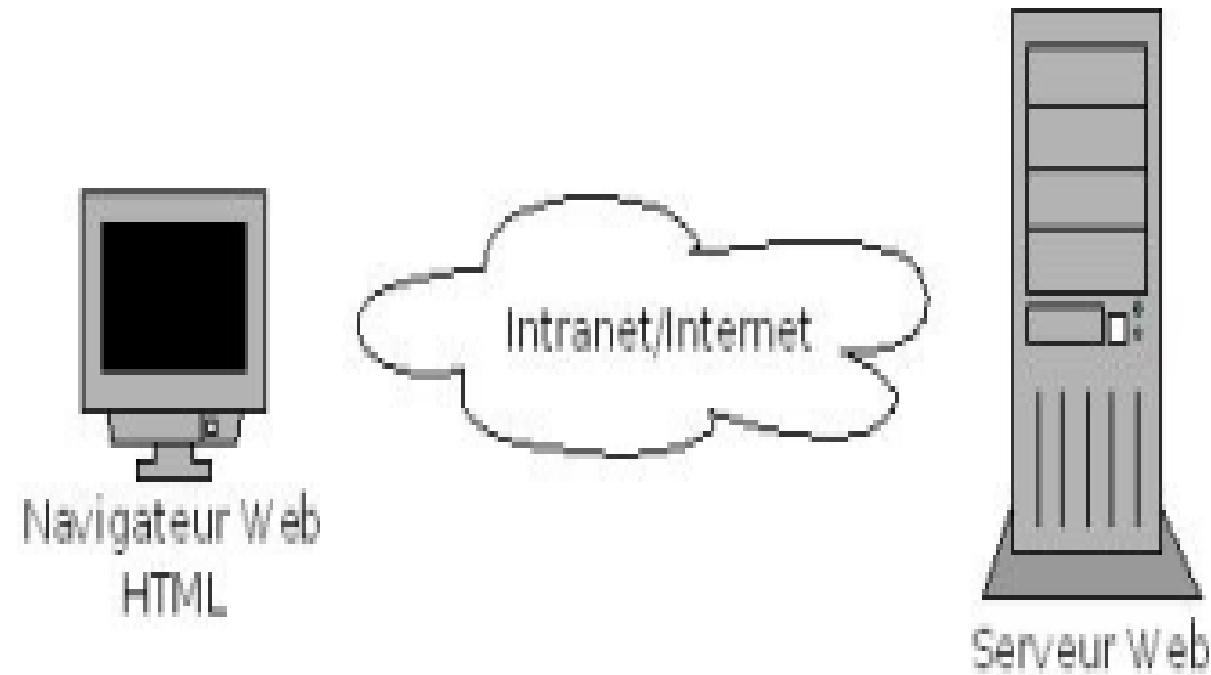
### ➤ **Architectures client-serveur**



# INTRODUCTION GÉNÉRALE

## Solutions Software

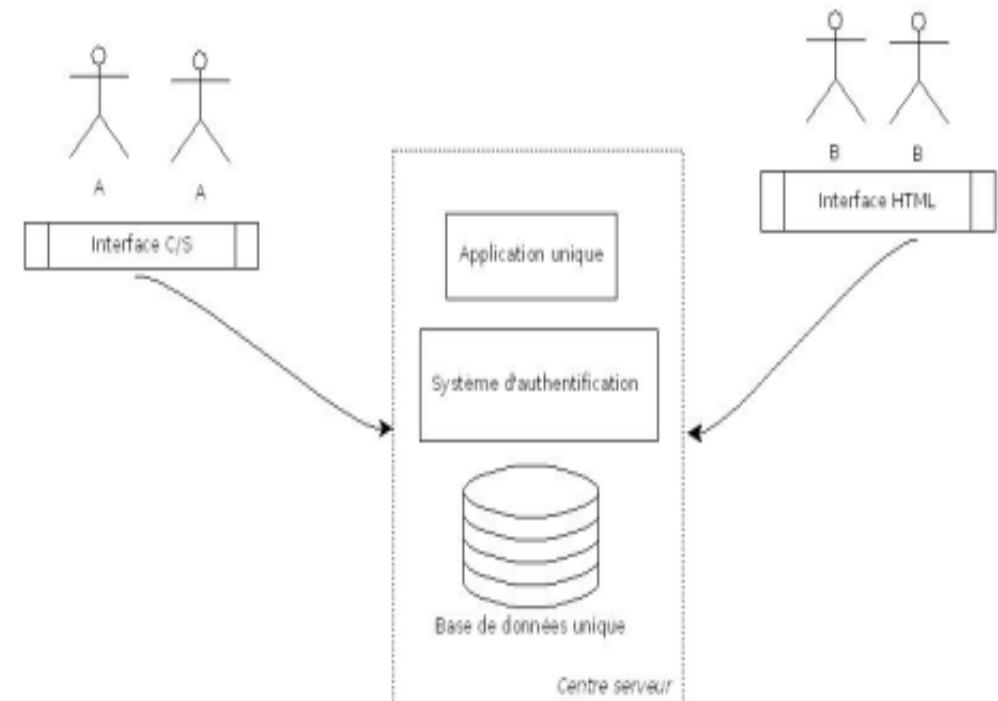
### ➤ **Architectures Web**



# INTRODUCTION GÉNÉRALE

## Solutions Software

- **Architecture service Web**
- **RIA, RDA, Web 2.0, Web 3.0 ...**



## INTRODUCTION GÉNÉRALE

### Solutions Software

#### ➤ La Plateforme collaborative

- Des outils informatiques.
- Des guides ou méthodes de travail en groupe, pour améliorer la communication, la production, la coordination.
- Un service de messagerie.
- Un système de partage des ressources et des fichiers.
- Des outils de type forum, pages de discussions
- Un trombinoscope, ou annuaire des profils des utilisateurs.
- Des groupes, par projet ou par thématique.
- Un calendrier

## GRID COMPUTING

### Problématique

- **Les temps de calcul : un problème majeur**

## GRID COMPUTING

➤ **Une solution : le calcul distribué**

**« Toute seconde de CPU inutilisée est PERDUE à jamais! »**

## GRID COMPUTING

### Systèmes distribués

Un système distribué est un ensemble composé d'éléments (machines) **indépendants, autonomes et reliés** par un système de communication équipées d'un logiciel dédié à la coordination des activités du système ainsi qu'au partage de ses ressources. Les éléments ont **des fonctions de traitement** (processeur) et **de stockage** (mémoire et disque). Ces éléments **collaborent** entre eux comme un système unique et cohérent pour répondre aux objectifs de l'utilisateur.

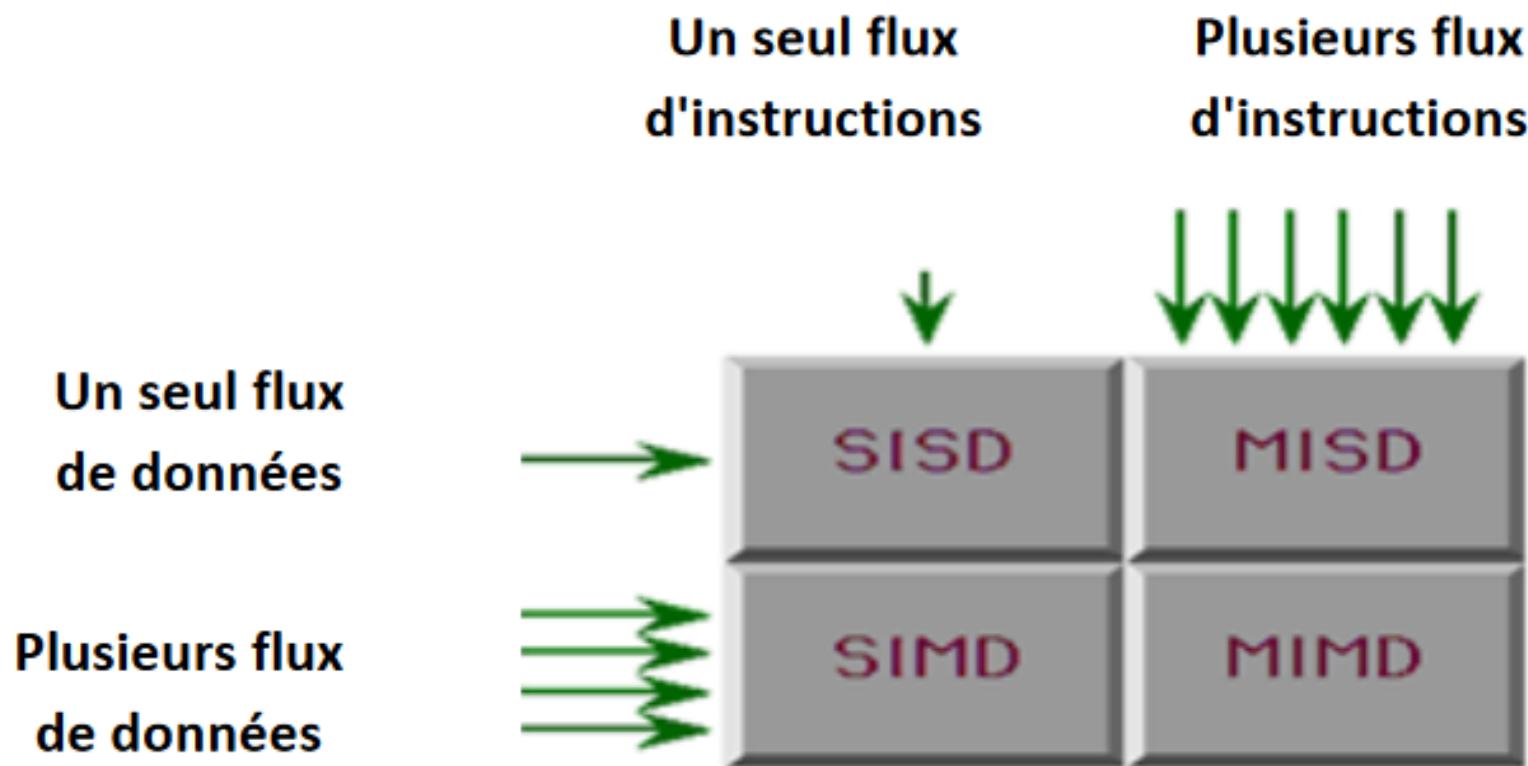
## GRID COMPUTING

### Caractéristique des systèmes distribués

- **Transparence**
- **Flexibilité**
- **Fiabilité**
- **Performance**

# GRID COMPUTING

## Architectures parallèles



## GRID COMPUTING

### **Avantages des systèmes distribués**

- Coût faible par rapport au prix de super ordinateur
- Puissance de calcul / stockage
- Performance de calcul / stockage
- Fiabilité et résistance aux pannes

## GRID COMPUTING

### Historique

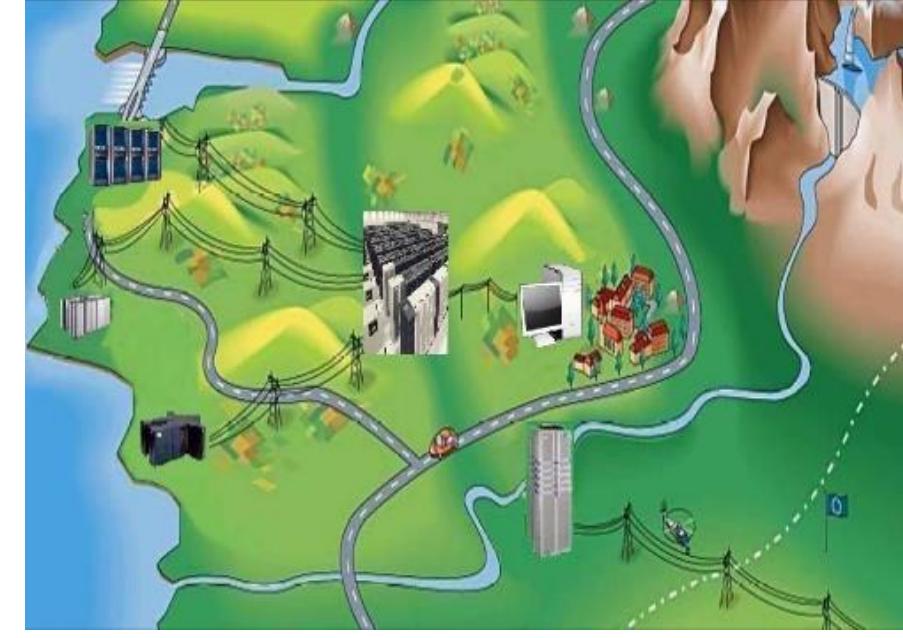
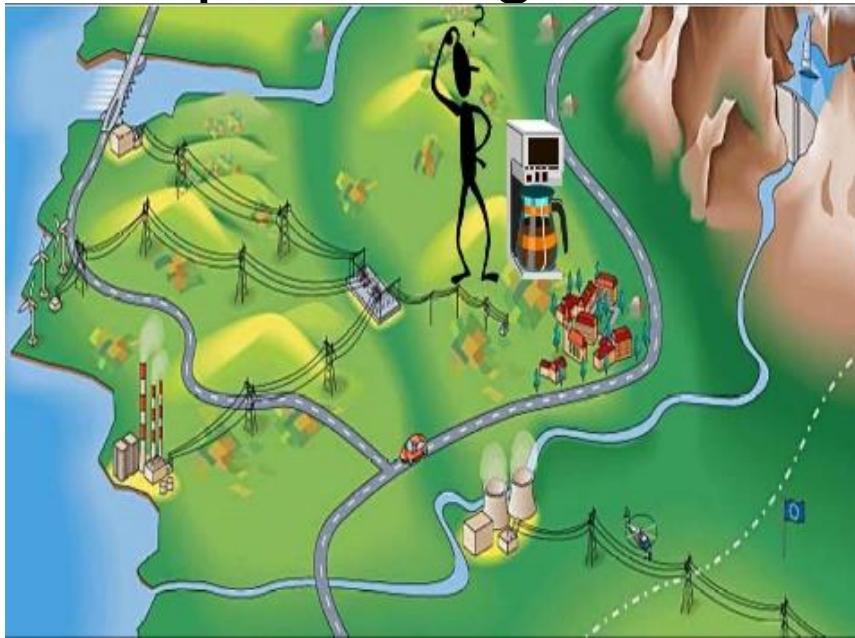
- Les origines du Grid Computing sont depuis les années 70.
- La société Apple, plus précisément l'entreprise NeXT avant qu'elle ne se fasse rachetée par Apple,
- Le programme de calcul parallèle distribué baptisé Zilla,



# GRID COMPUTING

## Historique

- Le terme de Grid Computing est apparu au année 1990
- S'inspire de la grille d'électricité (power grid)



## GRID COMPUTING

**Définitions : IBM, Fernando Corbató , Buyya, Plaszczak/Wellner, CERN**

**IBM** définit le grid computing comme une méthode utilisant une collection de protocoles standards pour avoir l'accès aux applications et aux données, à la puissance de calcul, à la capacité de stockage entre un vaste réseau de ressources informatiques à travers Internet.

# GRID COMPUTING

## Définitions

- Des ressources informatiques dont leur administration n'est pas centralisée,
- Des méthodes utilisées qui sont standardisées,
- Des ressources dont la qualité de service n'est pas assurée.

# GRIDE COMPUTING

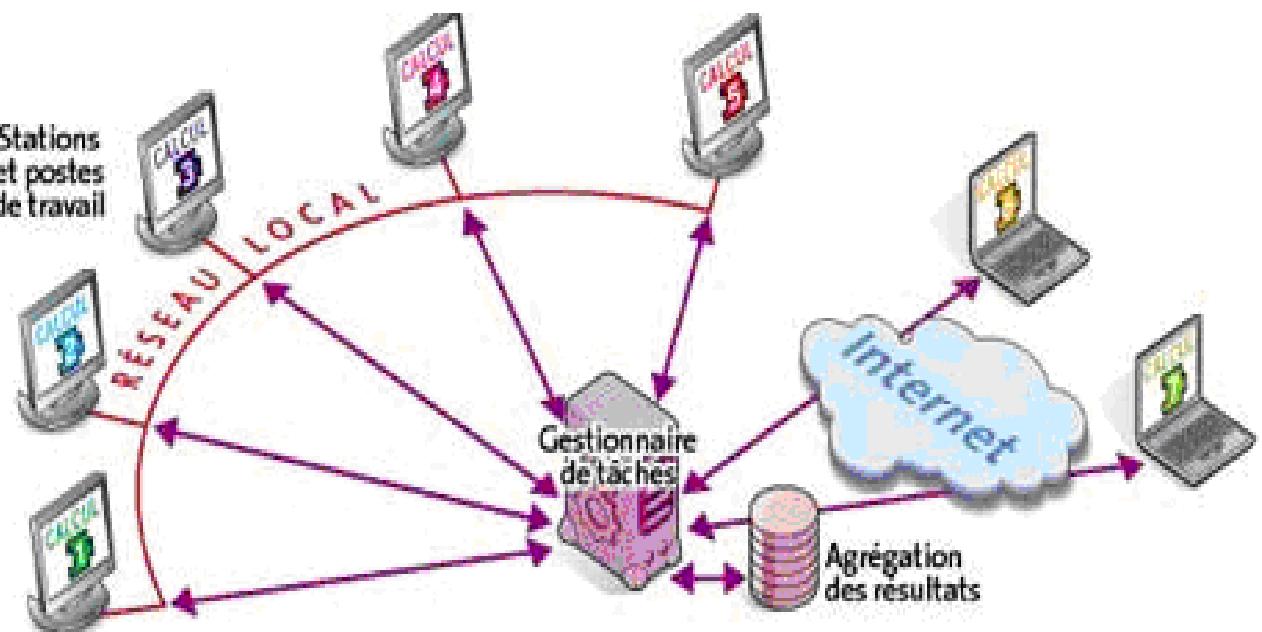
## Caractéristiques

- **Partagées**
- **Distribuées**
- **Hétérogènes**
- **Coordonnées**
- **Externalisées**
- **Non-contrôlées (ou autonomes)**
- **Environnement dynamique**

## GRID COMPUTING

### Constitution d'un Grid Computing

- **Un ou plusieurs serveurs,**
- **Un ou plusieurs clients.**



## GRID COMPUTING

### **Différents types de Grid informatique**

- Grid d'information
- Grid de stockage
- Grid de calcul (Computing)

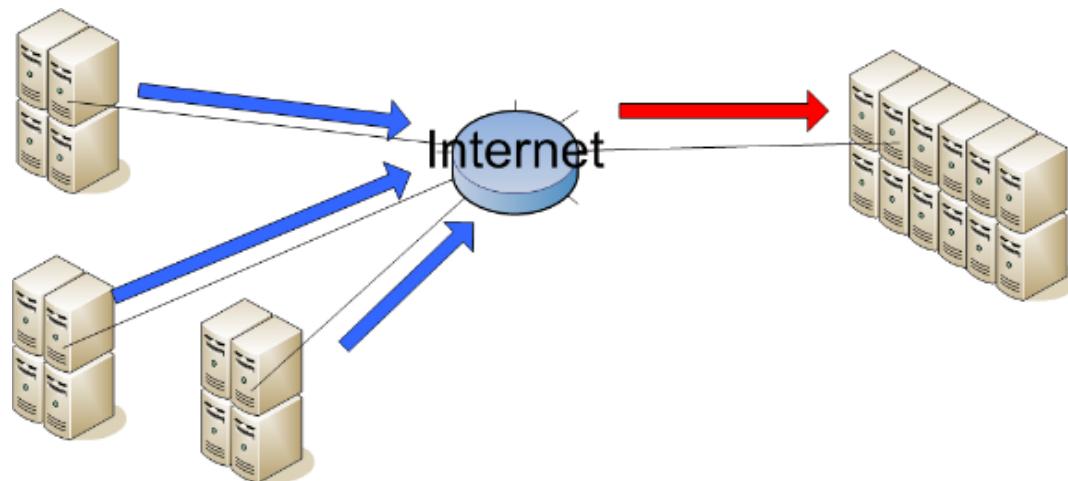
# GRID COMPUTING

## **Grid de calcul**

- Le Virtual SuperComputing
- Le MetaComputing
- L'Internet Computing

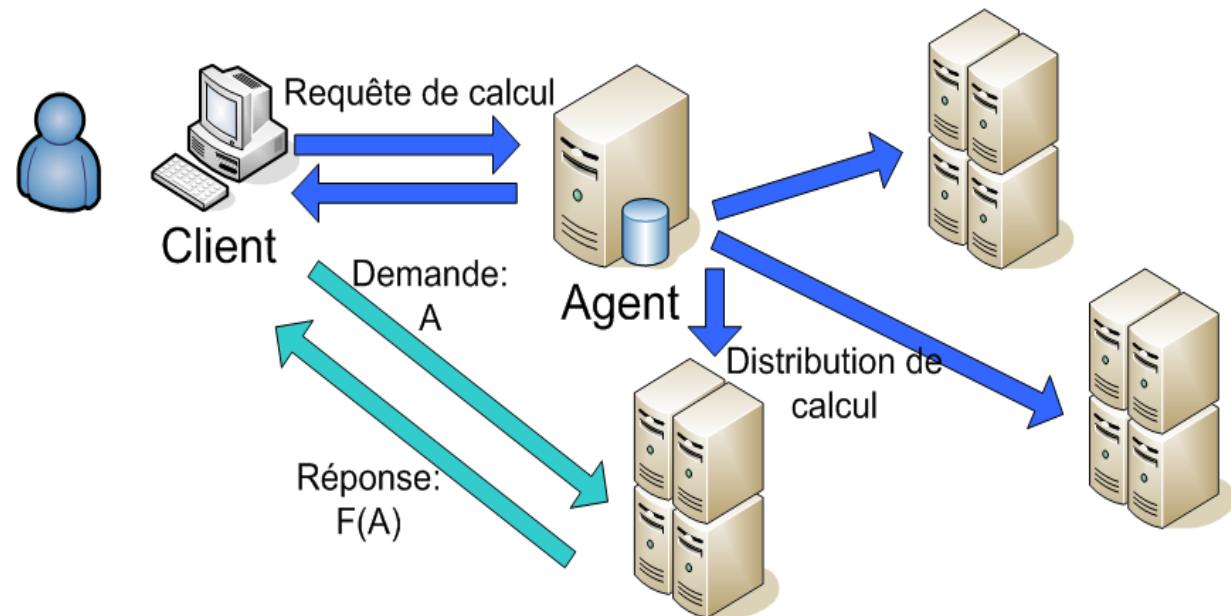
# GRID COMPUTING

## Le Virtual SuperComputing



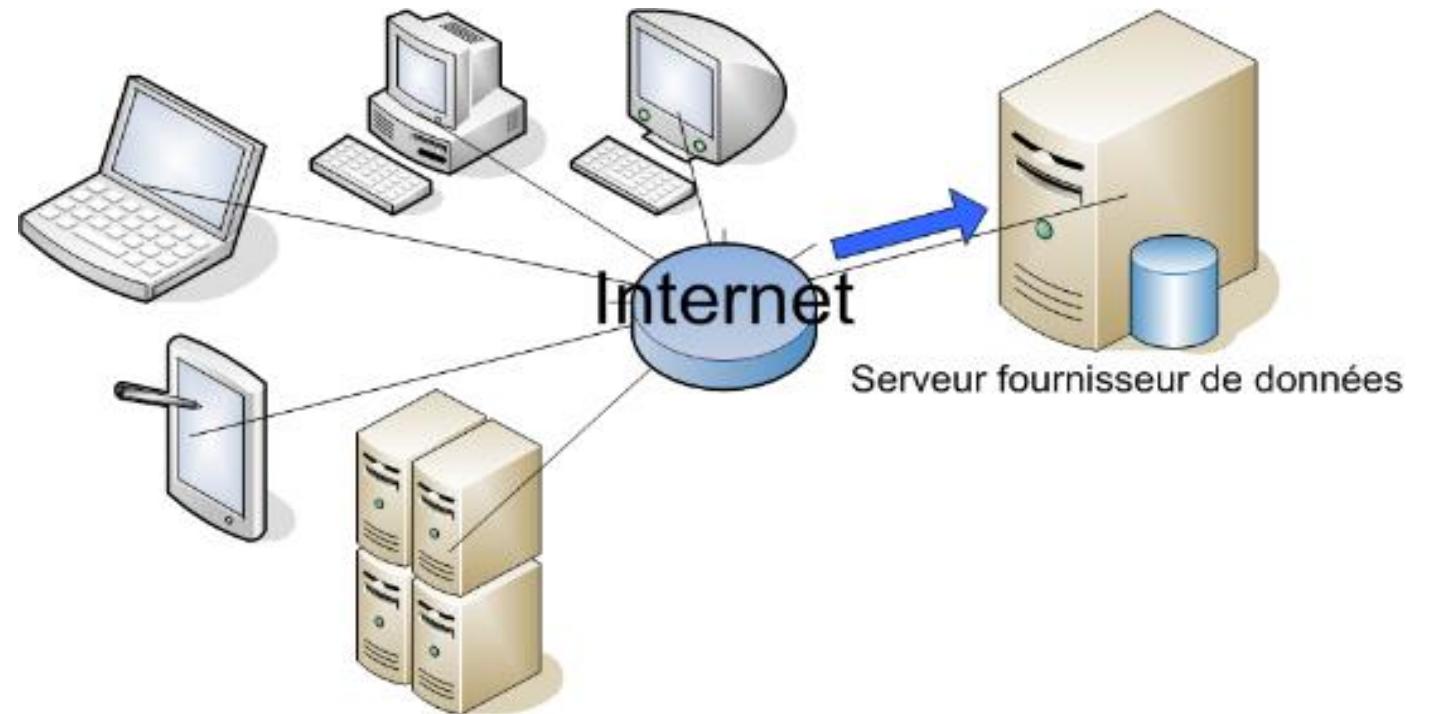
# GRID COMPUTING

## Le MetaComputing



# GRID COMPUTING

## L'Internet Computing



## GRID COMPUTING

### Domaines d'application de Grid Computing

- Le domaine industriel
  - Le calcul d'images (Films d'animation...),
  - La simulation d'utilisation de matériaux (Airbus),
  - La simulation d'utilisation de réseaux électriques,
  - La simulation d'activités pétrolières (Institut Français du Pétrole),
  - Les simulations bancaires (Banques, intermédiaires boursiers),
  - Les simulations militaires.

## GRID COMPUTING

### Domaines d'application de Grid Computing

- Le domaine scientifique



Par l'Université de Berkeley

La recherche d'intelligence extraterrestre

La société IBM

Le calcul dans le combat contre la maladie



## GRID COMPUTING

### **Domaines d'application de Grid Computing**

- Le domaine scientifique

- La Globus Alliance et le Globus Toolkit

- S'étend des universités ( Illinois, Californie, l'université de Chicago et NCSA...)

- Le Globus Toolkit est à la base de nombreux projets dans le monde scientifique...

## GRID COMPUTING

### Avantages

- Le calcul en grille permet le travail parallélisé de plusieurs machines sur un même problème.
- Les clients participant à la grille de calcul peuvent se situer dans des environnements complètement différents.
- Les clients ne sont pas forcément obligés d'être tout le temps connecté au serveur.
- Le principe d'externaliser/délocaliser la puissance de calcul permet de profiter d'une puissance de calcul exceptionnelle.

# GRID COMPUTING

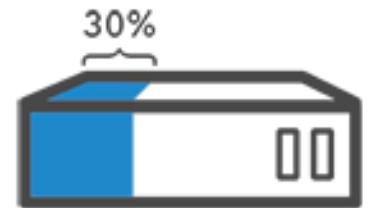
## Inconvénients

- Les clients ne possèdent pas une administration centralisée.
- L'impossibilité de prévoir à l'avance les ressources réelles qui seront présentes et disponibles à un instant T.
- Lenteur d'accès.
- Panne des ressources
- Environnement non fiable

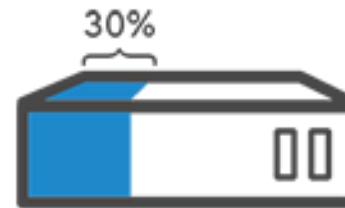
# VIRTUALISATION

## Problématique

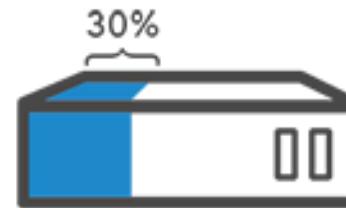
Les ordinateurs n'étaient capables de ne faire exécuter qu'un seul processus en même temps



MAIL

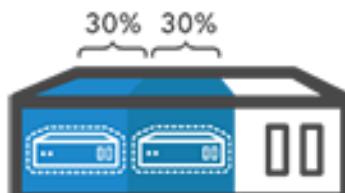


WEB

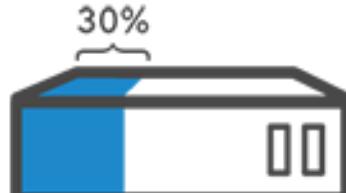


LEGACY APPS

- **La consolidation**



MAIL LEGACY APPS



WEB



# VIRTUALISATION

## Historique

- Apparaît dans les 1960's avec IBM de VM/CMS
- Répandu sur les mainframes avec des technologies propriétaires
- 1990's explosion de l'émulation sur x86 - VMWare popularise la machine virtuelle
- 2000's développement des solutions logicielles de virtualisation, aidées par les évolutions matérielles (Intel/AMD)

## VIRTUALISATION

### Définition

La virtualisation est une technologie qui permet de créer plusieurs environnements ou ressources à partir d'un seul système physique.

La virtualisation est une technologie qui vous permet de créer des services informatiques utiles à l'aide de ressources qui sont généralement liées au matériel.

## VIRTUALISATION

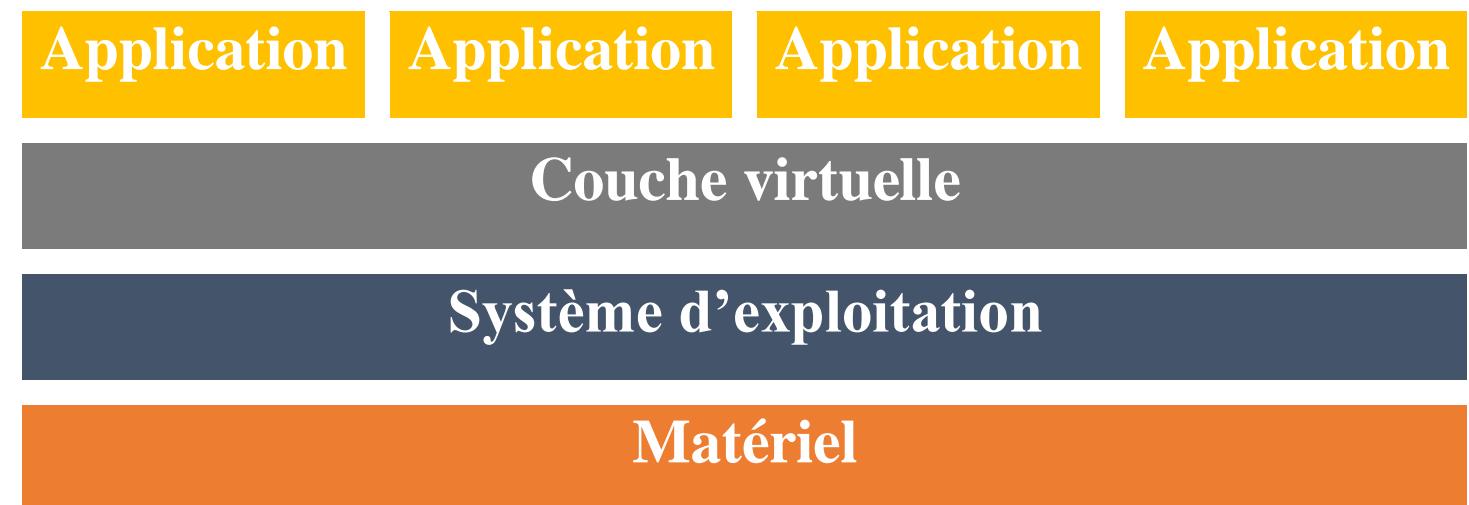
### **Les domaines de la virtualisation**

- La virtualisation d'applications
- La virtualisation de réseaux
- La virtualisation de stockage
- La virtualisation de serveurs

# VIRTUALISATION

## Les domaines de la virtualisation

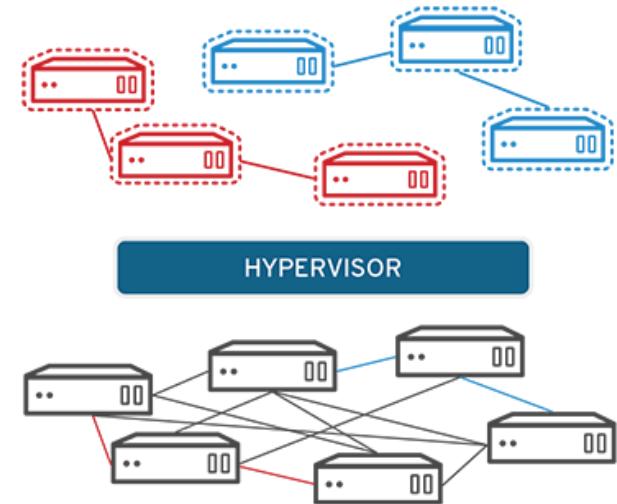
- La virtualisation d'applications



# VIRTUALISATION

## Les domaines de la virtualisation

- La virtualisation de réseaux
  - Une réduction du trafic de diffusion
  - Une sécurité accrue puisque l'information est encapsulée dans une couche supplémentaire
  - Une meilleure flexibilité



## VIRTUALISATION

### Les domaines de la virtualisation

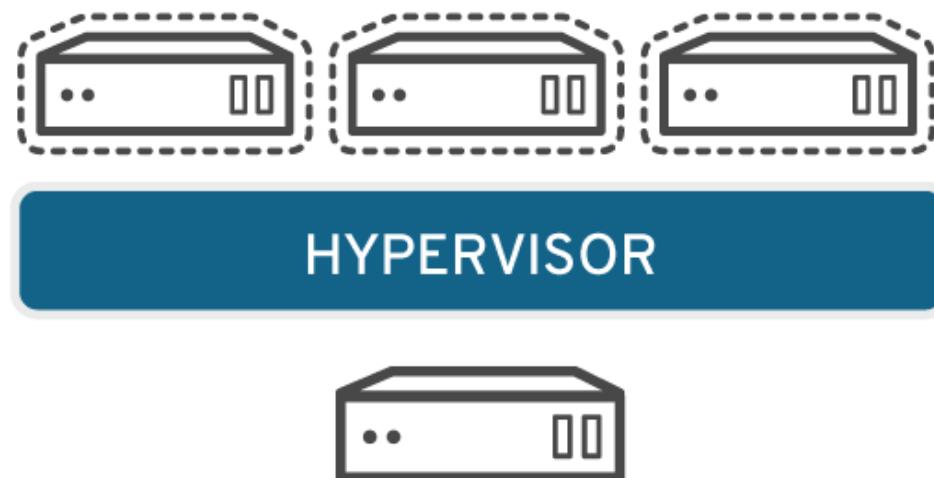
- La virtualisation de stockage (VDI, VHD, VMDK, OVF)
- Adjoindre un périphérique de stockage supplémentaire sans interruption des services
- Regrouper des unités de disques durs de différentes vitesses, de différentes tailles et de différents constructeurs
- Réallouer dynamiquement de l'espace de stockage

# VIRTUALISATION

## Les domaines de la virtualisation

- La virtualisation de serveurs

Fonctionner simultanément, sur un seul serveur physique, plusieurs serveurs virtuels.



# VIRTUALISATION

## Pourquoi virtualiser ?

- Le coût
- Criticité et performances
- La sécurité

# VIRTUALISATION

## Le coût

- Diminution du nombre de serveurs
- Diminution du matériel réseau
- Réduction de la consommation électrique
- Diminution des besoins en climatisation
- Diminution de la consommation d'espace
- Agrégation des charges d'inactivité
- Optimisation de la restauration
- Optimisation de la sauvegarde

## VIRTUALISATION

### Criticité et performances

- La simplicité du déploiement
- La simplicité de l'administration
- L'amélioration de la gestion du changement
- L'optimisation de l'évolution des capacités physiques
- L'amélioration de l'équilibre des charges
- La simplicité des tests de logiciels

## VIRTUALISATION

### La sécurité

La virtualisation va permettre une isolation des différents environnements logiciels au niveau des ressources physiques. La communication entre les différentes machines virtuelles sera uniquement possible via des connexions réseau de manière identique à la communication entre deux machines physiques.

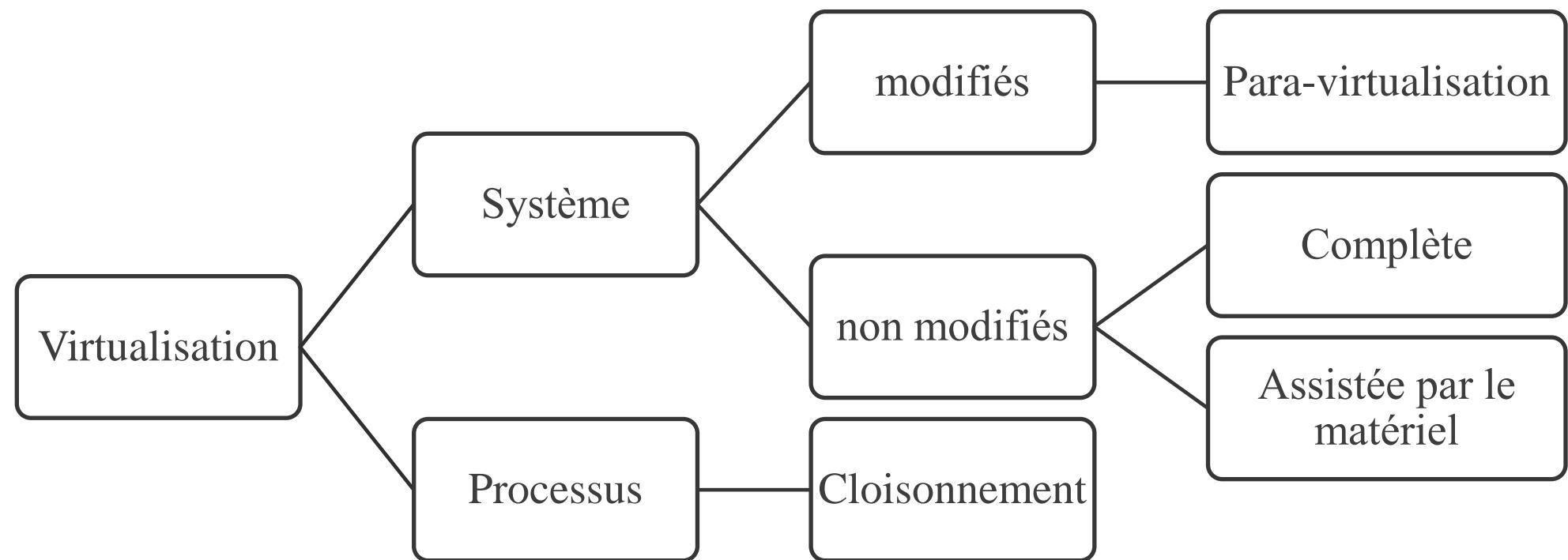
## VIRTUALISATION

### **Différents types de virtualisation**

- Virtualisation complète
- Para-virtualisation
- Virtualisation assistée par le matériel
- Cloisonnement

# VIRTUALISATION

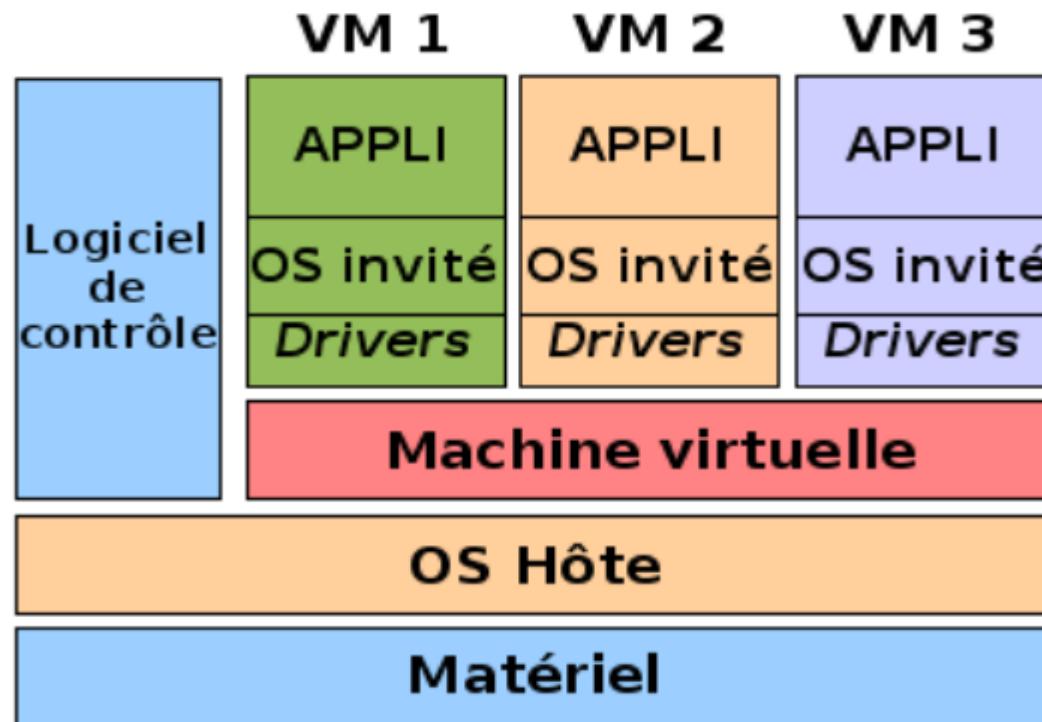
## Différents types de virtualisation



# VIRTUALISATION

## Différents types de virtualisation

- Virtualisation complète



## VIRTUALISATION

### **Différents types de virtualisation**

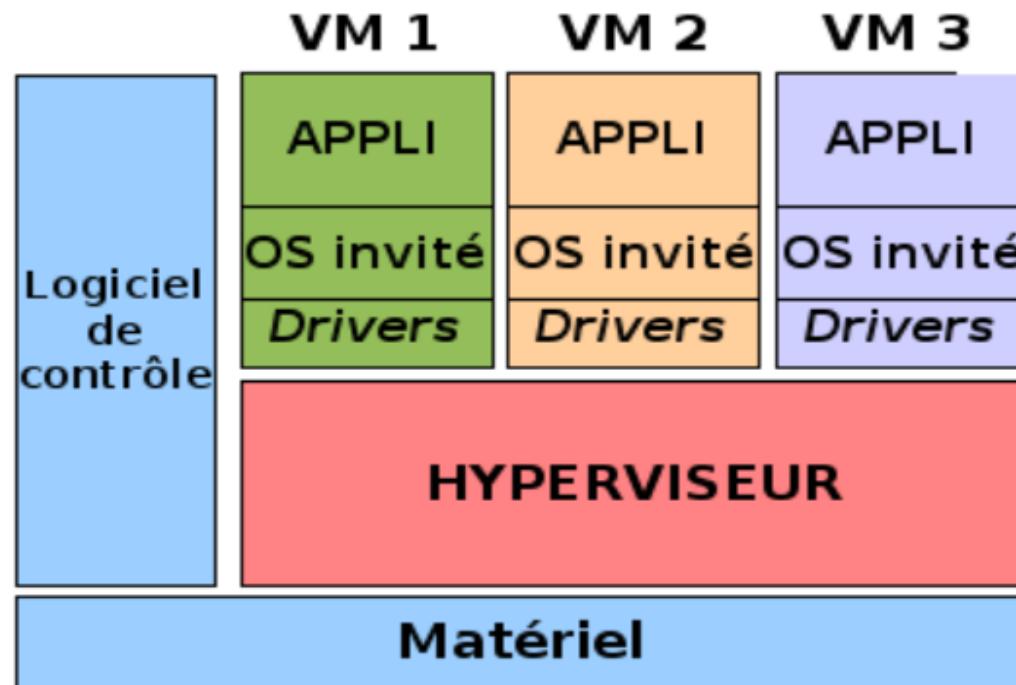
- Virtualisation complète

Exemples : Microsoft VirtualPC et Microsoft VirtualServer,  
VirtualBox, VMware ...

# VIRTUALISATION

## Différents types de virtualisation

### ➤ Para-virtualisation



# VIRTUALISATION

## Différents types de virtualisation

### ➤ Para-virtualisation

#### Exemples :

- Xen : projet OpenSource précurseur dans le monde du libre, version commercialisée par Citrix.
- KVM : projet hyperviseur intégré dans le noyau linux (Développé par Qumranet, racheté par RedHat)
- ESX/ESXi : hyperviseur leader de VMWare
- Hyper-V : hyperviseur de Microsoft

## VIRTUALISATION

### **Différents types de virtualisation**

#### ➤ Virtualisation assistée par le matériel

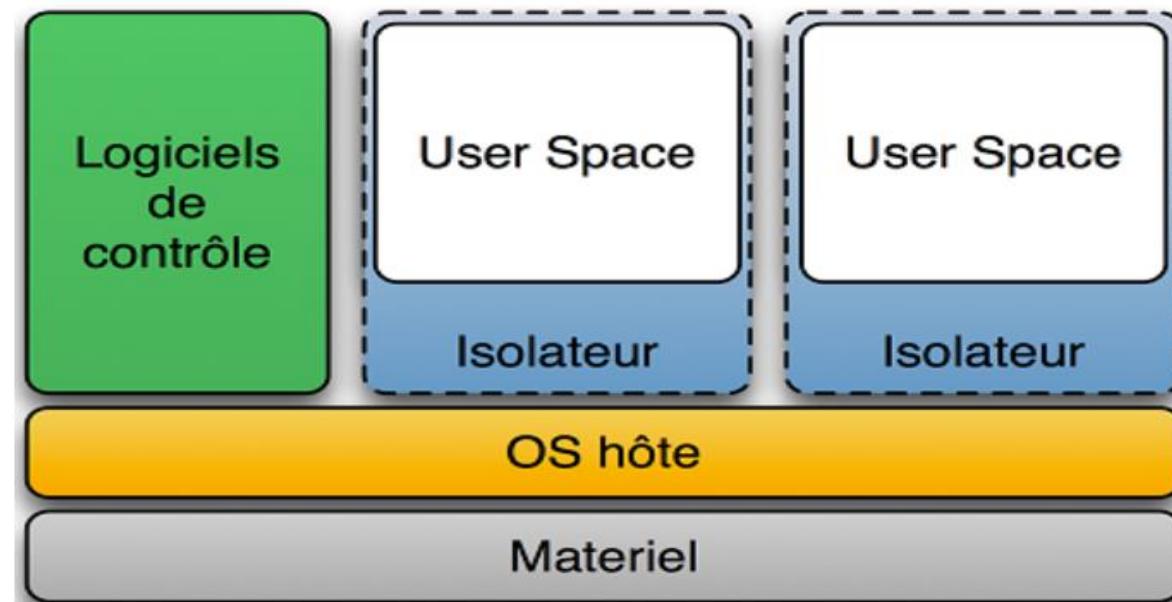
Le but est de faire fonctionner des systèmes invités dont les OS peuvent être différents mais non modifiés, mais on tire pleinement partie du matériel et de sa puissance.

Exemple : Intel-VTx (32 bits) et Intel VT-i (64 bits) pour Intel, AMD-V pour AMD, Advanced Power Virtualization pour IBM

# VIRTUALISATION

## Différents types de virtualisation

### ➤ Cloisonnement



# VIRTUALISATION

## Principales solutions

- XEN
- KVM
- VMware
- Hyper-V

## VIRTUALISATION

### Les principales solutions

- XEN
  - Hyperviseur géré par la société XenSource.
  - Le projet était à l'origine mené au sein de l'Université de Cambridge,
  - Le but était alors d'héberger 100 systèmes invités sur une seule machine physique, avec les meilleures performances possibles.

## VIRTUALISATION

### Les principales solutions

- KVM
  - Kernel-based Virtual Machine a été créé en 2006 par la société Qumranet.
  - Intégré au noyau Linux dès février 2007.
  - KVM un système optimisé pour la virtualisation de serveur.
  - Pour virtualiser des systèmes de type desktop, on peut lui préférer virtualbox.

## VIRTUALISATION

### Les principales solutions

- VMware vSphere
- Logiciel d'infrastructure de Cloud computing de l'éditeur VMware
- Hyperviseur basé sur l'architecture VMware ESXi

## VIRTUALISATION

### Les principales solutions

- Hyper-V
- Windows Server Virtualization, est un système de virtualisation basé sur un hyperviseur 64 bits

## VIRTUALISATION

### Avantages de la virtualisation

- Optimisation des ressources;
- Installation, sauvegarde, déploiement et migration faciles des machines virtuelles ;
- Economie sur le matériel par mutualisation ;
- Sécurisation et/ou isolation d'un réseau ;
- Diminution des risques liés au dimensionnement des serveurs
- Une reprise automatique lors des incidents.

## VIRTUALISATION

### Inconvénients de la virtualisation

- Si la machine physique tombe en panne plusieurs environnements virtuels sont interrompus.
- Nécessité des machines puissantes.
- Une dégradation des performances.

## VIRTUALISATION

### Travail Pratique I

Objectifs :

- Etudier et comparer des outils de virtualisations ;
- Créer et configurer des machines virtuelles ;

**Partie I : Virtualisation complet :** Créer, avec un outil de virtualisation, un laboratoire de piratage qui contient le système Kali Linux et ses outils de détection de piratage

**Partie II : Para-virtualisation :** Installer un serveur de virtualisation : Matériel requis : Un PC servant de serveur hôte (de préférence doté d'une RAM de 4GO) et un ordinateur portable client servant à l'administration des machines virtuelles.

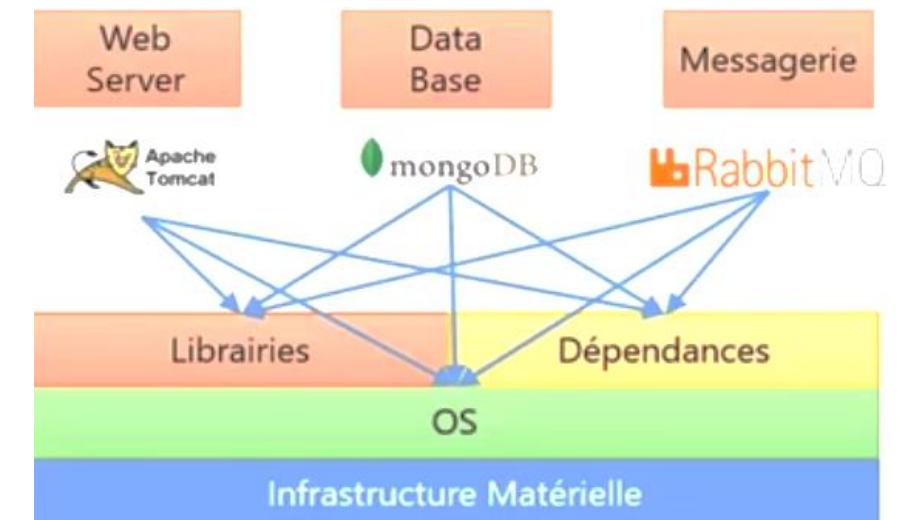
# VIRTUALISATION

- Suivez les étapes suivantes :
  1. Télécharger un hyperviseur de virtualisation
  2. Graver l'image iso
  3. Installer l'hyperviseur sur la machine serveur, en rebootant sur le CD-ROM
  4. Configurer les paramètres de « root » ainsi que les adresses IP pour les machines (serveur et l'adresse de la passerelle - gateway)
  5. Installer la machine client :
    - Télécharger la machine virtuelle client pour piloter à distance les machines
    - Lancer l'installateur. Puis configurer la machine virtuelle
    - Lancer la configuration d'une nouvelle machine virtuelle. Choisir une configuration typique pour la machine.
    - Préciser le système d'exploitation qui sera exécuté sur la machine virtuelle.

## VIRTUALISATION

# Problème de déploiement des applications

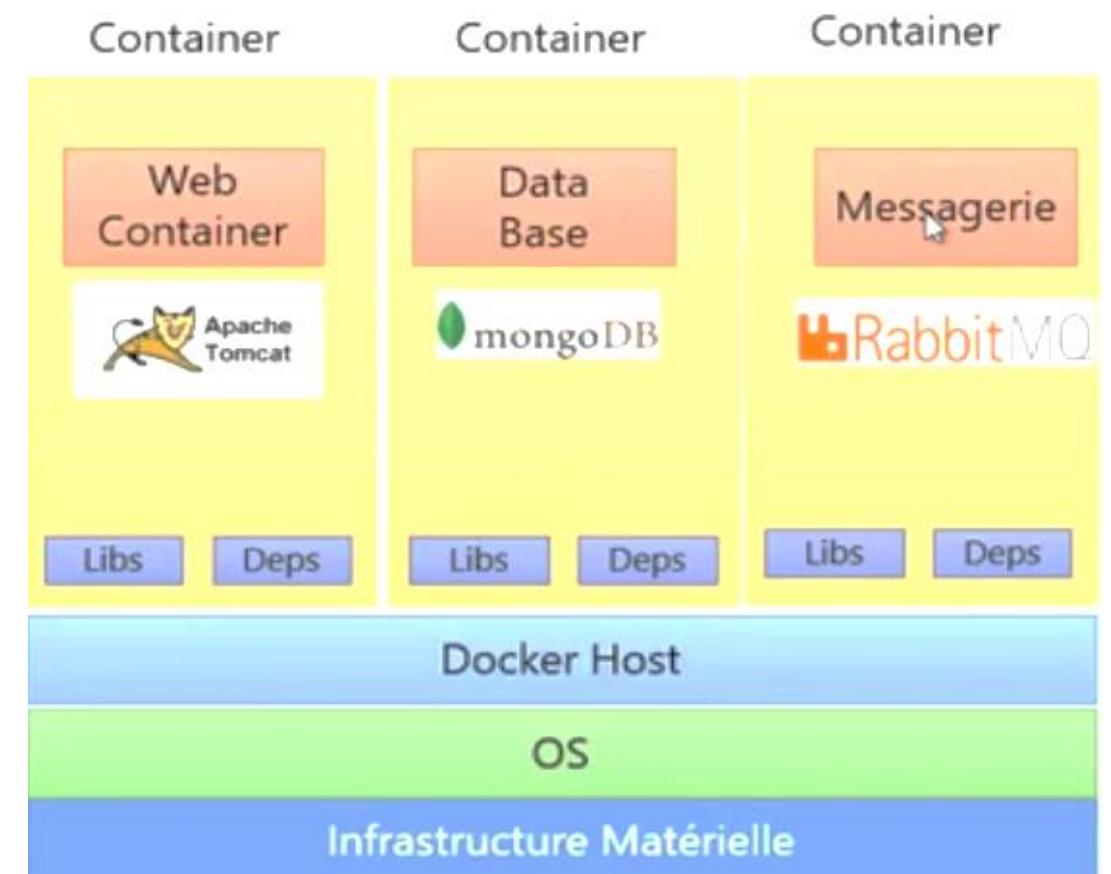
- Pour une application utilisant différentes technologies (services), des problèmes sont posés au moment du déploiement en production:
  - Compatibilité des applications avec les OS
  - Installer les dépendances et les librairies requises avec les bonnes versions pour chaque service.
  - Installer les différents environnements :
    - Dev
    - Test
    - Prod
- Ce qui prend beaucoup de temps pour déployer les applications.
- Avec des conflits entre les développeurs et les opérationnels (Administrateurs systèmes)



## VIRTUALISATION

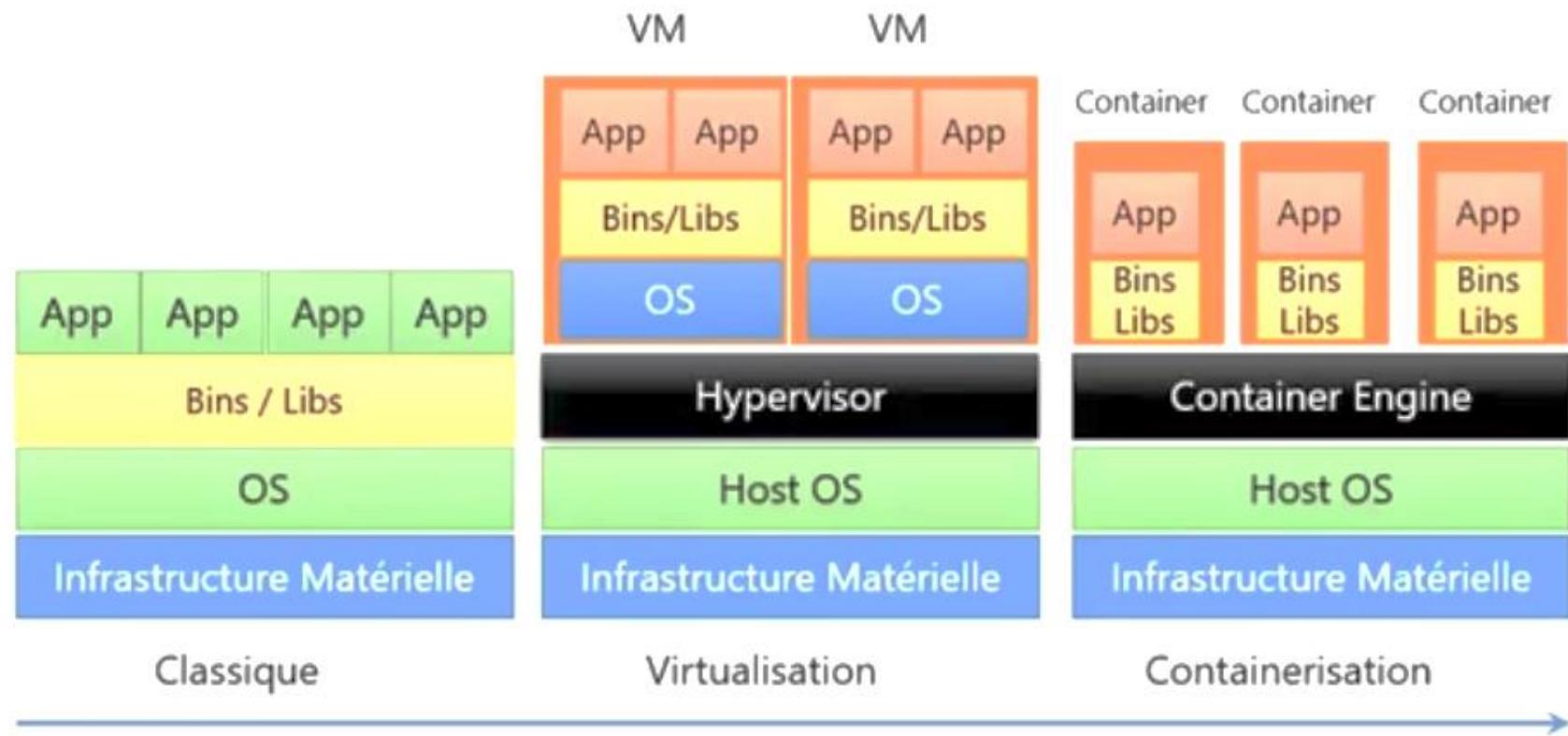
### Solution : Conteneurs d'application

- Embarquer les applications dans des conteneurs
- Exécuter chaque service avec ses propres dépendances dans des conteneurs séparés.



# VIRTUALISATION

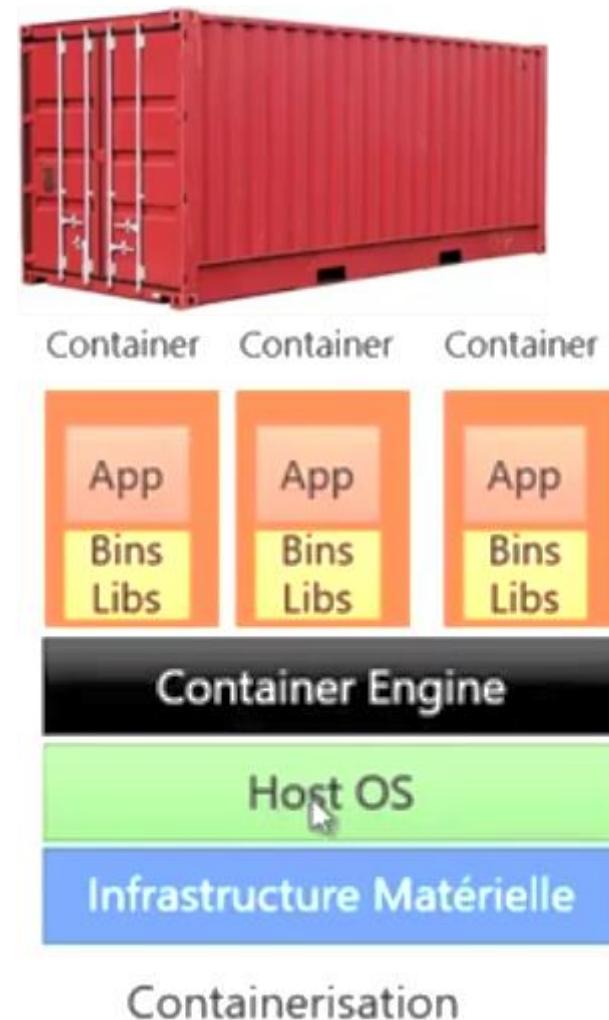
## Containerisation (Cloisonnement)



# VIRTUALISATION

## Container

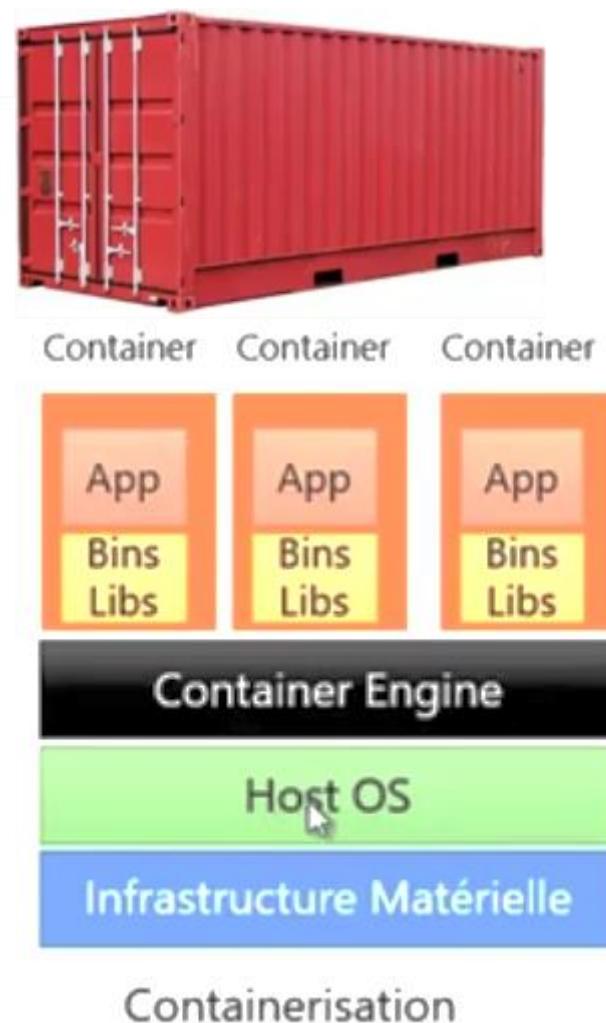
- Enveloppe permettant de packager une application avec juste ce dont elle a besoin pour fonctionner
- Peut être déployé tel quel dans n'importe quelle machine disposant d'un Container Engine avec différents environnement (Dev, Test, Prod)
- Utilise le Kernel de l'OS Hôte
- A son propre espace de processus et sa propre interface réseau.
- Isolé de l'hôte, mais exécutée directement dessus.
- Permet de décomposer l'infrastructure applicative en petits éléments légers facile à déployer et à réutiliser



## VIRTUALISATION

### Pourquoi utiliser les Containers

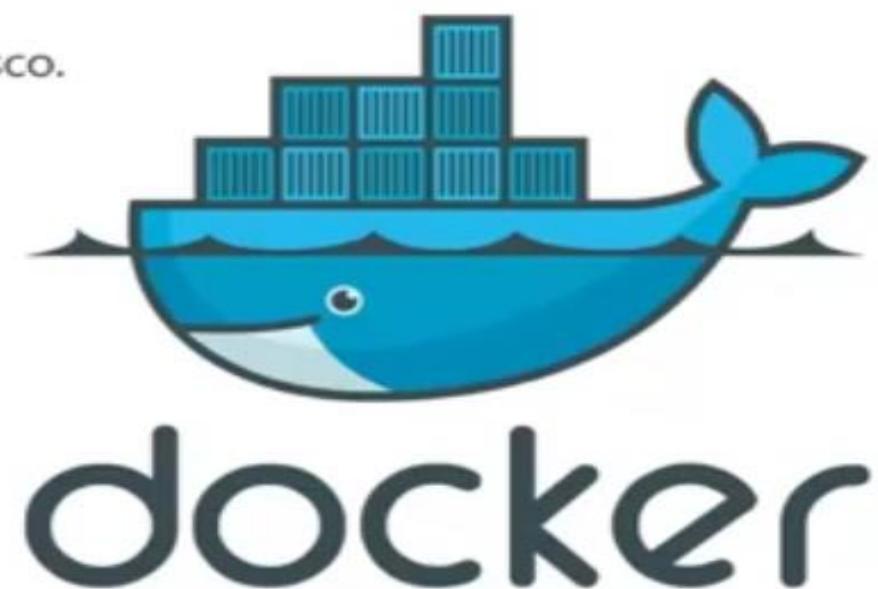
- Meilleures performances que les VM (Démarrage instantané )
- Portabilité d'un environnement à l'autre (Multi cloud)
- Cohérence entre les environnements Dev, test et prod
- Permet de modulariser facilement l'application
- Gérer l'héritage technique (Ancienne application)  
Grâce à l'isolation ↗



## VIRTUALISATION

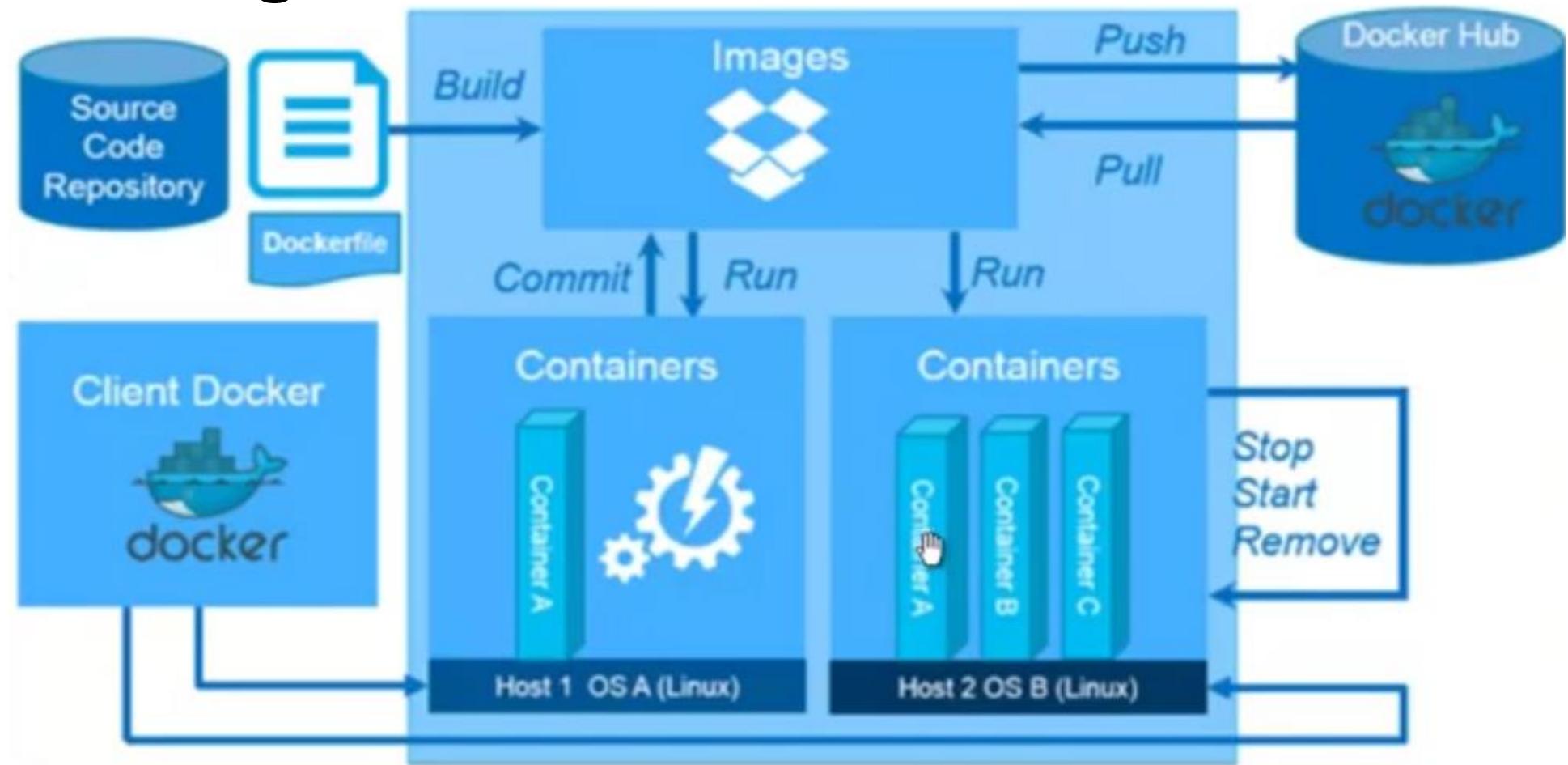
### C'est quoi Docker

- Docker permet de créer des environnements (appelées containers) de manière à isoler des applications.
- Il permet d'empaqueter une application ainsi que les dépendances nécessaire dans un conteneur virtuel isolé qui pourra être exécuté sur n'importe quelle machine supportant docker.
- Docker est un logiciel libre qui permet le déploiement d'applications sous la forme de conteneurs logiciels.
- L'origine de docker est :
  - Au départ, société française et maintenant basée à San Francisco.
  - dotCloud : un PaaS avec un container engine écrit en Python
  - En 2012 : Réécriture le langage Go.
  - En 2013, Première version Open source de docker
  - Réaction très positive de la communoté
  - dotCloud change de nom pour Docker.
  - En 2014, Levée de fonds : 40 millions \$
  - En 2015, Levée de fonds : 95 millions \$



## VIRTUALISATION

# Architecture globale de Docker



## VIRTUALISATION

# Architecture globale de Docker **2EFD0F4CFF142**

- Le développeur crée un fichier Dockerfile contenant les commandes que docker va exécuter pour construire une image docker de cette application.
  - **\$ docker build**
- L'image docker contient tout ce dont l'application a besoin pour s'exécuter correctement.
- Les images Docker peuvent être publiées dans un registre public (Docker hub) ou privé.
  - **\$ docker push image\_name**
- Pour télécharger une image docker d'une application dans un Host Docker, il suffit d'utiliser:
  - **\$ docker pull image\_name**
- La création et l'exécution d'un conteneur d'une application se fait par instantiation et exécution de l'image en utilisant :
  - **\$ docker run image\_name**

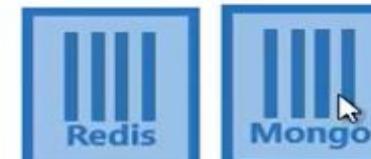
# VIRTUALISATION

Docker => Docker Hub

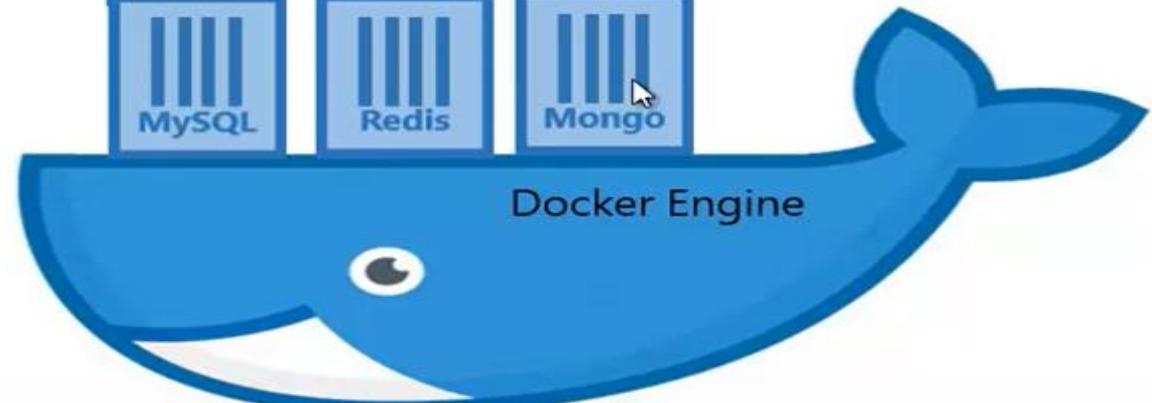
```
$ docker run mysql  
$ docker run redis  
$ docker run mongodb  
$ docker run mysql
```

<https://hub.docker.com/>

Docker Hub  
Public Docker Registry



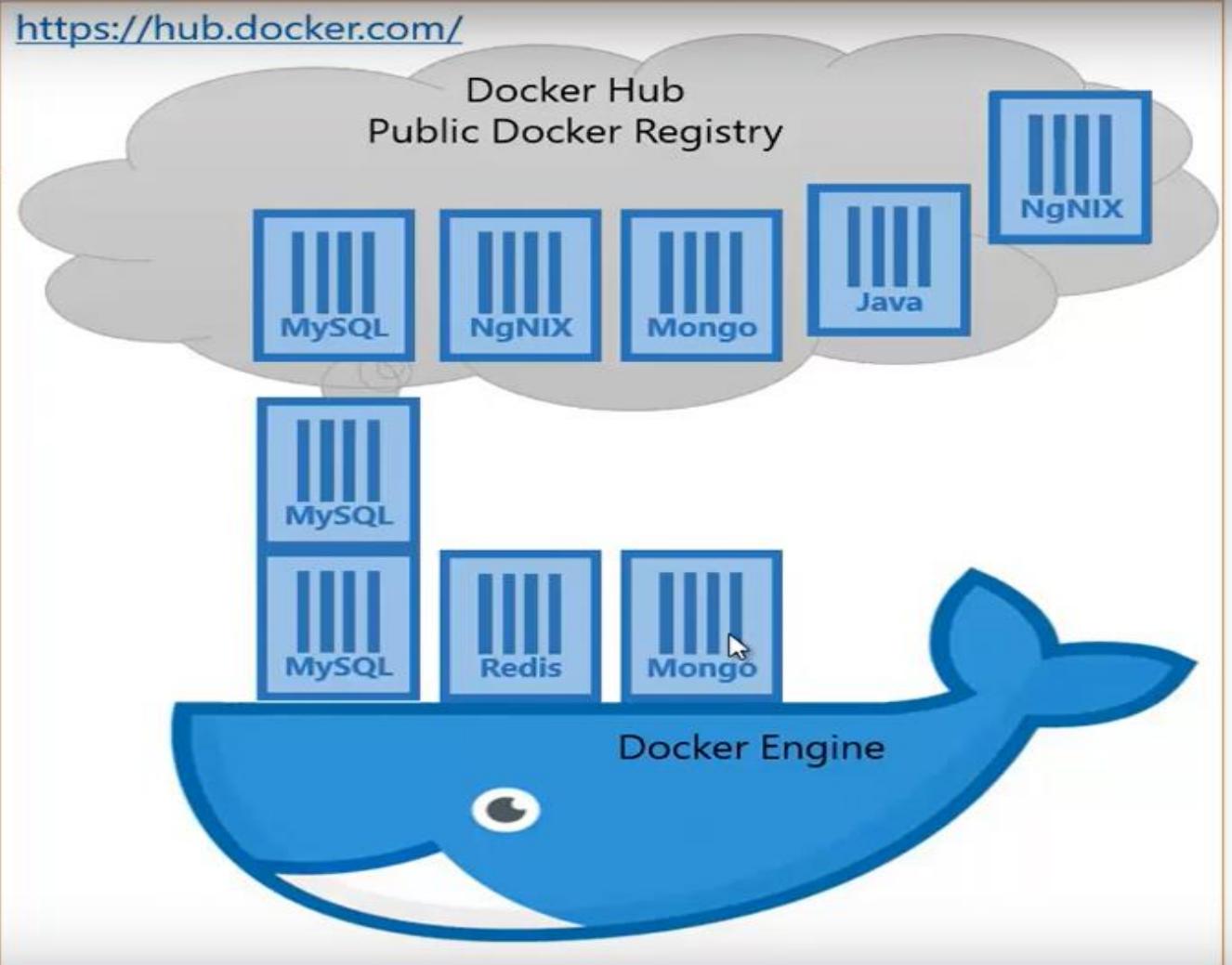
Docker Engine



# VIRTUALISATION

## Image vs Container

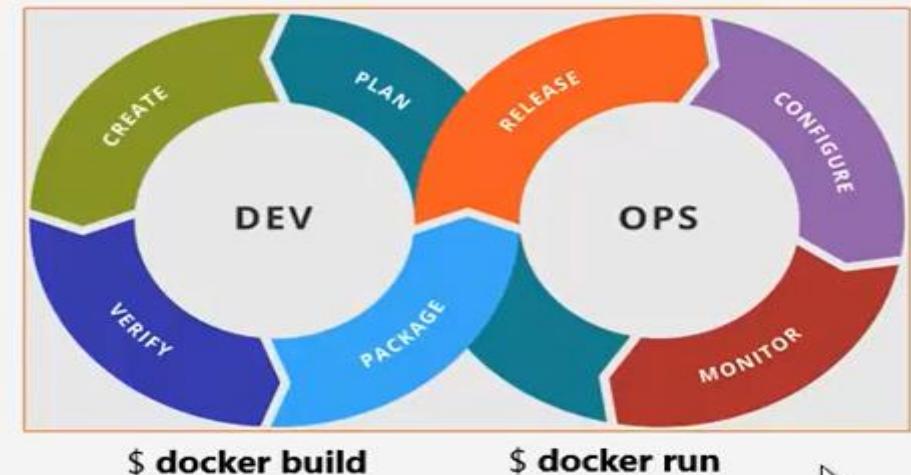
- Une image docker est juste un fichier package représentant la tempalte des conteneurs. Elle définit la structure du conteneur en englobe l'application containérisée et l'ensemble de ses dépendances.
- Un Conteneur représente une instance d'une image. Un conteneur est exécutée par le Docker Host. Ce qui implique l'exécution de l'application qu'il transporte dans un environnement isolé fourni par le conteneur.



# VIRTUALISATION

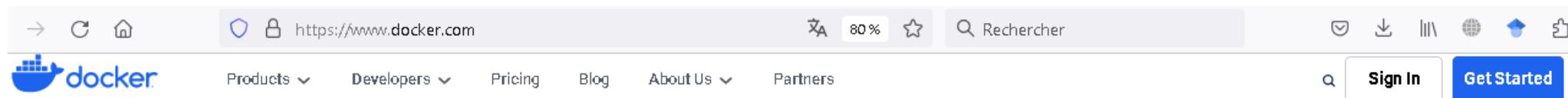
## Docker contribue à instaurer la culture DevOps

- **Sans Docker :**
  - Le développeur
    - développe l'application.
    - Génère le package de l'application à déployer (App.war)
    - Envoie à l'opérationnel (Administrateur système)
      - App.war
      - Un descriptif des dépendances qu'il faut installer et configurer pour que l'application s'exécute normalement.
  - L'opérationnel
    - doit se débrouiller pour satisfaire les exigences de l'application.
    - Pour chaque mise à jour, c'est toujours les mêmes histoires qui se répètent.
  - Ce qui rend la vie dur au administrateur systèmes (Opérationnels)
  - Ce qui crée beaucoup de conflits entre les développeurs qui tentent d'améliorer constamment les applications et les opérationnels qui doivent redéployer les mises à jour.
- **Avec Docker:**
  - Le développeur
    - développe l'application
    - Construit une image docker de son application contenant toutes les dépendances dont l'appli a besoin.
    - Publie l'image docker sur le registre docker
  - L'opérationnel déploie l'application en instantiant des conteneurs à partir de l'image docker récupérée à partir du repository docker.



## VIRTUALISATION

### Installation de Docker



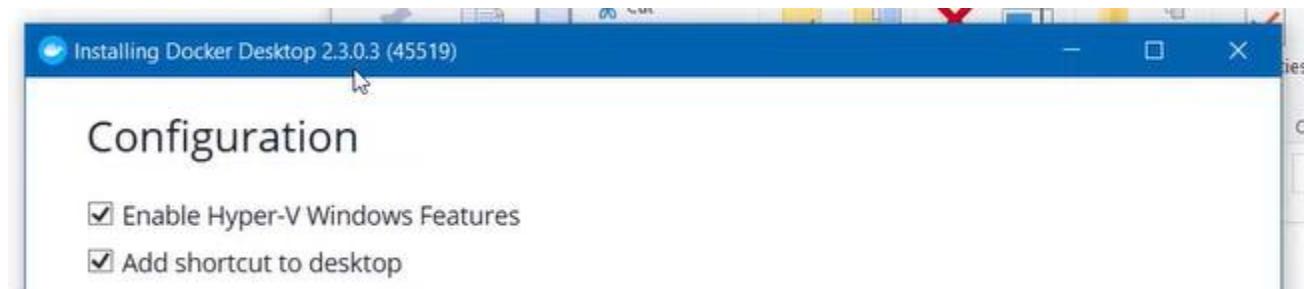
**Make better, secure  
software from the start**

Announcing Docker Scout general availability.

[Learn about Docker Scout](#)

# VIRTUALISATION

## Installation de Docker



```
PS C:\Users\Ahmad> docker -v
Docker version 19.03.8, build afacb8b
PS C:\Users\Ahmad> docker version
Client: Docker Engine - Community
  Version: 19.03.8
  API version: 1.40
  Go version:  go1.12.17
  Git commit:  afacb8b
  Built: Wed Mar 11 01:23:10 2020
  OS/Arch: windows/amd64
  Experimental: false
error during connect: Get http://%2F%2Fpipe%2Fdocker_engine/v1.40/version: open //./pipe/docker_engine: The system cannot find the file specified. In the default daemon configuration on Windows, the docker client must be run elevated to connect. This error may also indicate that the docker daemon is not running.
PS C:\Users\Ahmad>
```

## VIRTUALISATION

### Utilisation de Docker

The screenshot shows the Docker Hub interface for the MySQL official image. At the top, there's a blue header bar with the Docker Hub logo, a search bar, navigation links for Explore, Repositories, Organizations, Help, and a user profile section for 'dockertest39'. Below the header, the URL path 'Explore / Official Images / mysql' is visible. The main content area features the MySQL logo and the text 'mysql Docker Official Image · 1B+ · 10K+'. It describes MySQL as a widely used, open-source relational database management system (RDBMS). To the right, there's a button labeled 'docker pull mysql' with a copy icon. On the left, under the 'Quick reference' section, there's a note about maintenance by 'innovation oracle'. On the right, a 'Recent Tags' sidebar lists various Docker tags: 8.0, 8-oracle, 8, oracle, latest, innovation-oracle, innovation, 8.1.0-oracle, 8.1.0, and 8.1-oracle.

## VIRTUALISATION

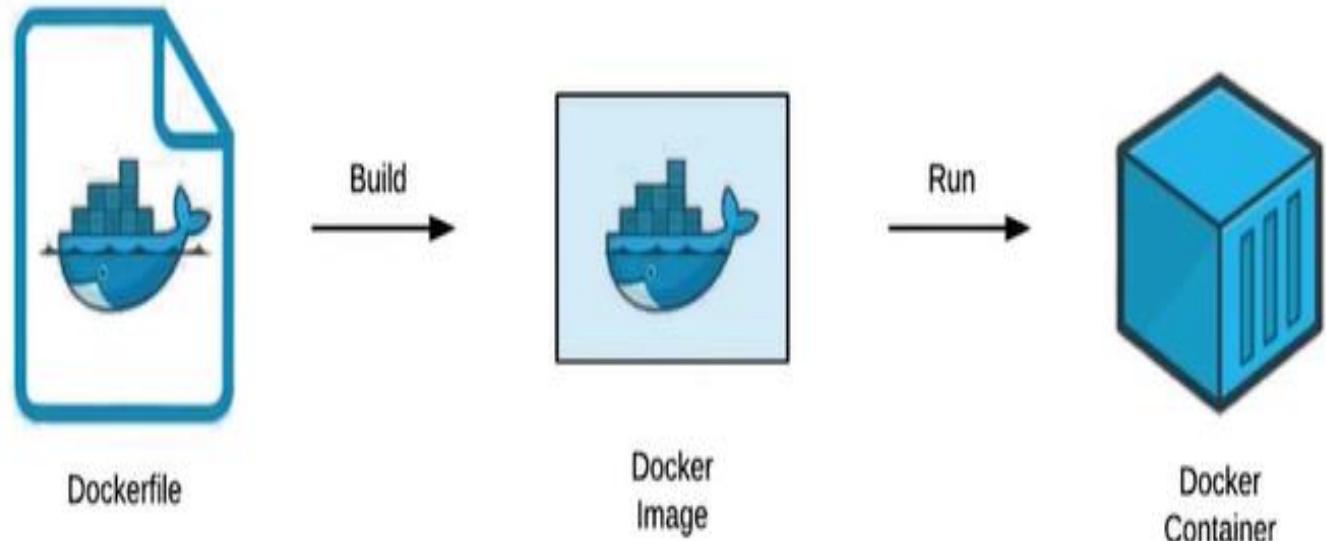
### Utilisation de Docker

```
# Sample contents of Dockerfile
# Stage 1
FROM microsoft/aspnetcore-build:2.0 AS build-env
WORKDIR /source

# caches restore result by copying csproj file separately
COPY *.csproj .
RUN dotnet restore

# copies the rest of your code
COPY ..
RUN dotnet publish --output /app/ --configuration Release

# Stage 2
FROM microsoft/aspnetcore
WORKDIR /app
COPY --from=builder /app .
ENTRYPOINT ["dotnet", "dockertest.dll"]
```



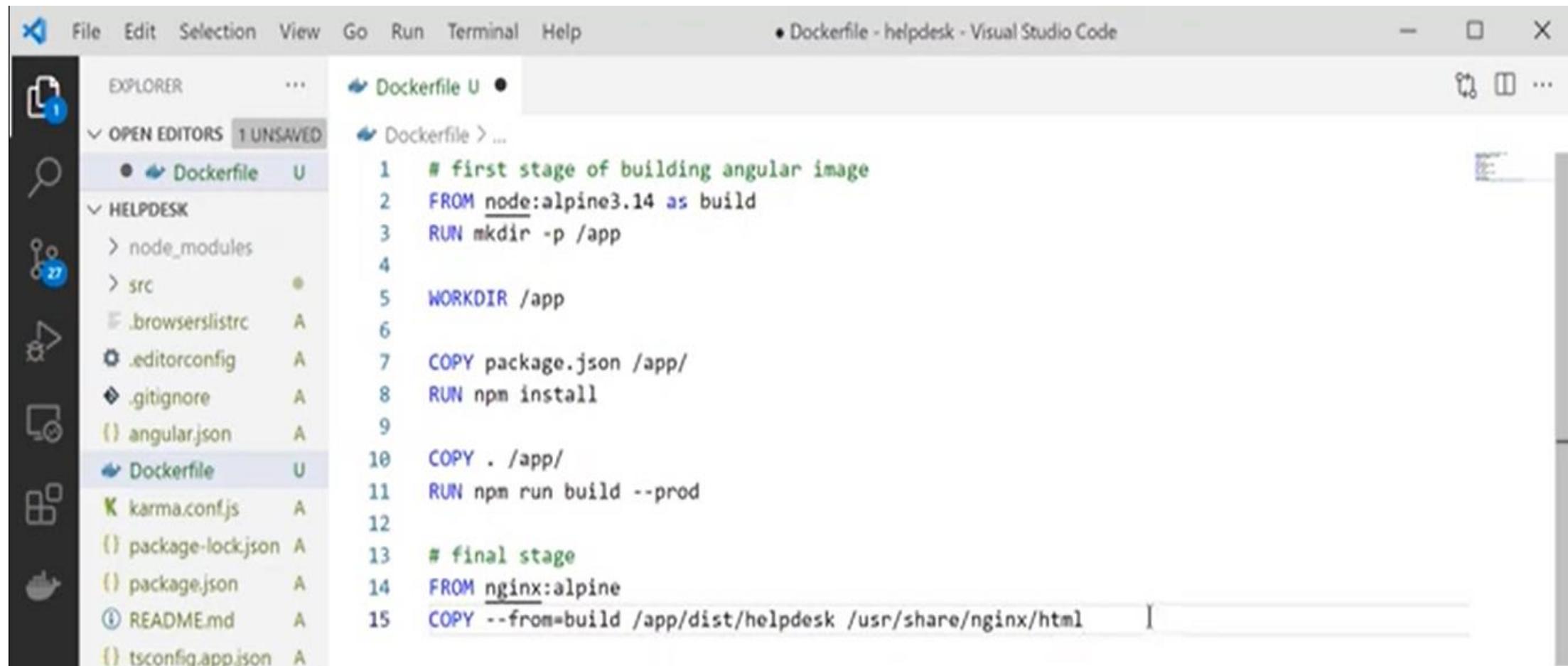
# VIRTUALISATION

## Instruction de Dockerfile

- A Dockerfile uses the following commands for building the images:
  - **ADD**: Copy files from a source on the host to the container's own filesystem at the set destination.
  - **CMD**: Execute a specific command within the container.
  - **ENTRYPOINT**: Set a default application to be used every time a container is created with the image.
  - **ENV**: Set environment variables.
  - **EXPOSE**: Expose a specific port to enable networking between the container and the outside world.
  - **FROM**: Define the base image used to start the build process.
  - **MAINTAINER**: Define the full name and email address of the image creator.
  - **RUN**: Central executing directive for Dockerfiles.
  - **USER**: Set the UID (the username) that will run the container.
  - **VOLUME**: Enable access from the container to a directory on the host machine.
  - **WORKDIR**: Set the path where the command, defined with CMD, is to be executed.

# VIRTUALISATION

## Utilisation de Docker



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Dockerfile - helpdesk - Visual Studio Code.
- Left Sidebar (Explorer):** Shows the project structure:
  - OPEN EDITORS: Dockerfile (1 UNSAVED)
  - HELPDESK:
    - node\_modules
    - src
    - .browserslistrc
    - .editorconfig
    - .gitignore
    - angular.json
    - Dockerfile (highlighted)
    - karma.conf.js
    - package-lock.json
    - package.json
    - README.md
    - tsconfig.app.json
- Right Panel (Dockerfile Content):**

```
1 # first stage of building angular image
2 FROM node:alpine3.14 as build
3 RUN mkdir -p /app
4
5 WORKDIR /app
6
7 COPY package.json /app/
8 RUN npm install
9
10 COPY . /app/
11 RUN npm run build --prod
12
13 # final stage
14 FROM nginx:alpine
15 COPY --from=build /app/dist/helpdesk /usr/share/nginx/html
```

## VIRTUALISATION

### Utilisation de Docker

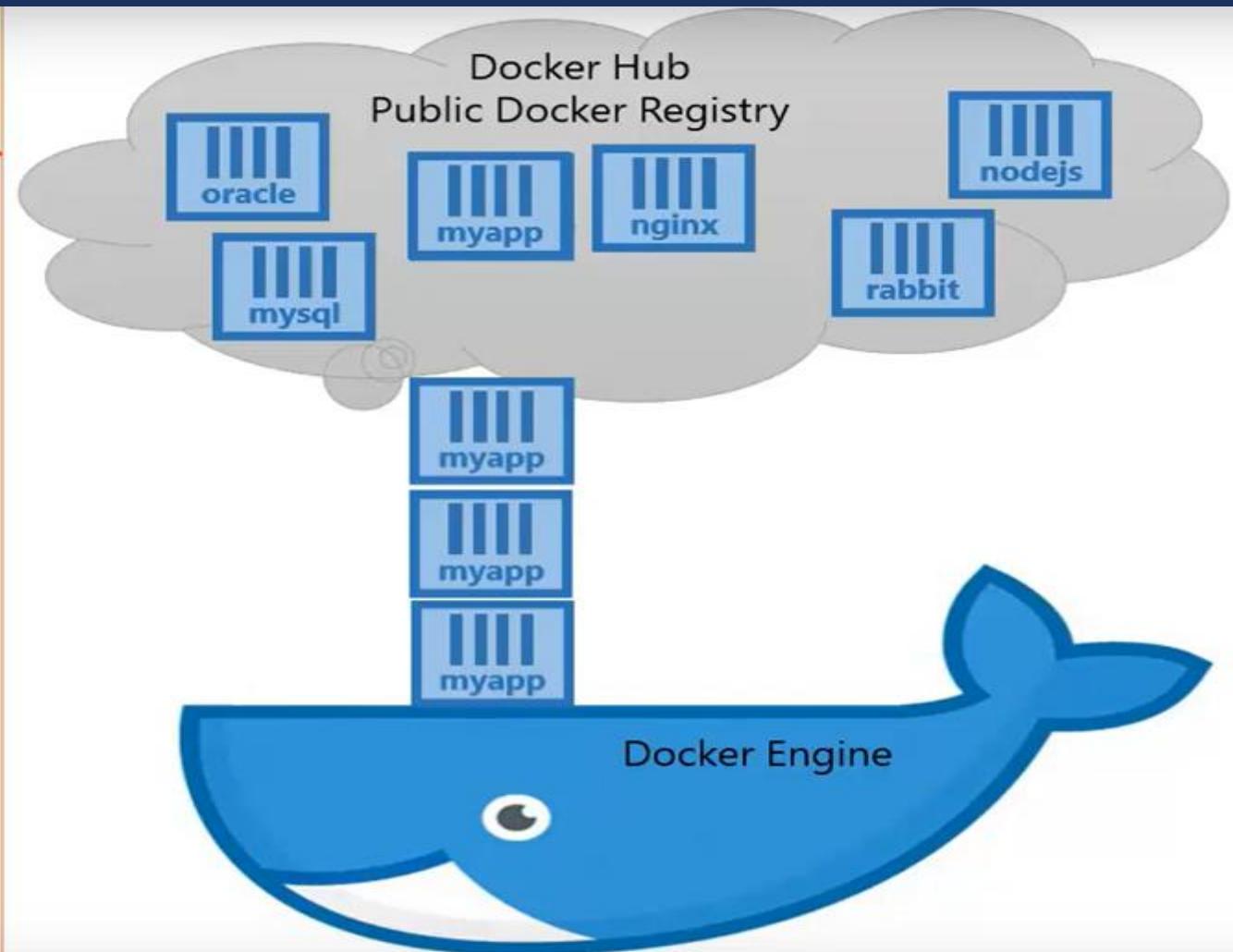
```
PS C:\Users\Ahmad> docker push ahmadmohey/redis
Using default tag: latest
The push refers to repository [docker.io/ahmadmohey/redis]
262de04acb7e: Preparing
45f6df634253: Preparing
e46136075591: Preparing
11f991845040: Preparing
dd1ebb1f5319: Preparing
814bff734324: Waiting
denied: requested access to the resource is denied
PS C:\Users\Ahmad> docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com
o create one.
Username: ahmadmohey
Password:
Login Succeeded
PS C:\Users\Ahmad> docker push ahmadmohey/redis
Using default tag: latest
The push refers to repository [docker.io/ahmadmohey/redis]
262de04acb7e: Mounted from library/redis
45f6df634253: Mounted from library/redis
e46136075591: Mounted from library/redis
11f991845040: Mounted from library/redis
dd1ebb1f5319: Mounted from library/redis
814bff734324: Mounted from ahmadmohey/nginx
latest: digest: sha256:1bd57e1a42b99ae53412b582784d0362fa8205243ce5f289cb4f76de2907cb97 size: 1574
```

# VIRTUALISATION

## Orchestration des conteneurs

```
$ docker run myapp  
$ docker run myapp  
$ docker run myapp
```

- Problème de Disponibilité et de monté en charge :
  - Il faut démarrer plusieurs instances en fonction de la charge
  - Si une instance tombe en panne, il faut la remplacer
  - Si Docker Engine tombe en panne, tous les conteneur vont cracher.
- L'orchestration des conteneurs est une solution pour ce problème

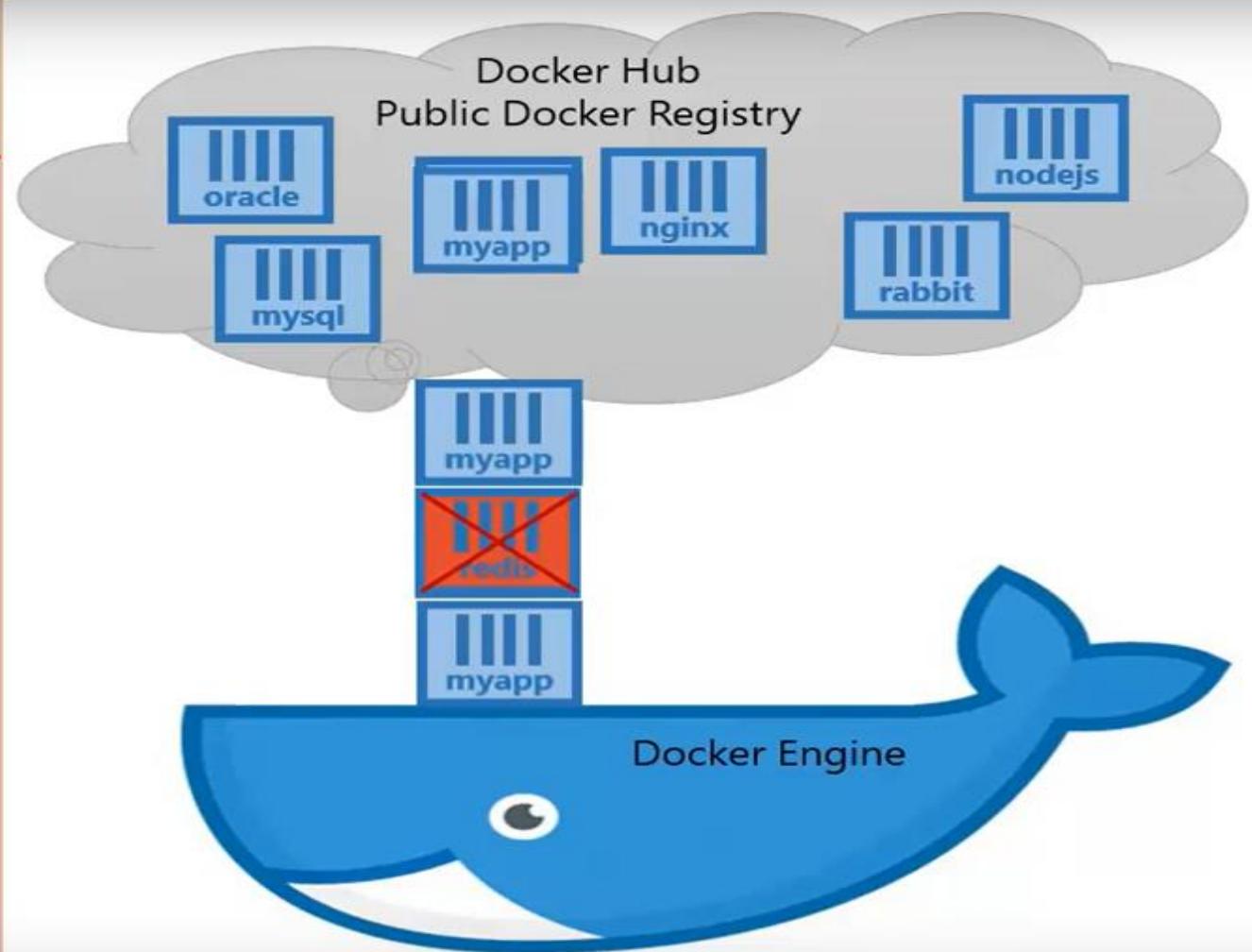


# VIRTUALISATION

## Orchestration des conteneurs

```
$ docker run myapp  
$ docker run myapp  
$ docker run myapp  
$ docker run myapp
```

- Problème de Disponibilité et de monté en charge :
  - Il faut démarrer plusieurs instances en fonction de la charge
  - Si une instance tombe en panne, il faut la remplacer
  - Si Docker Engine tombe en panne, tous les conteneur vont cracher.
- L'orchestration des conteneurs est une solution pour ce problème

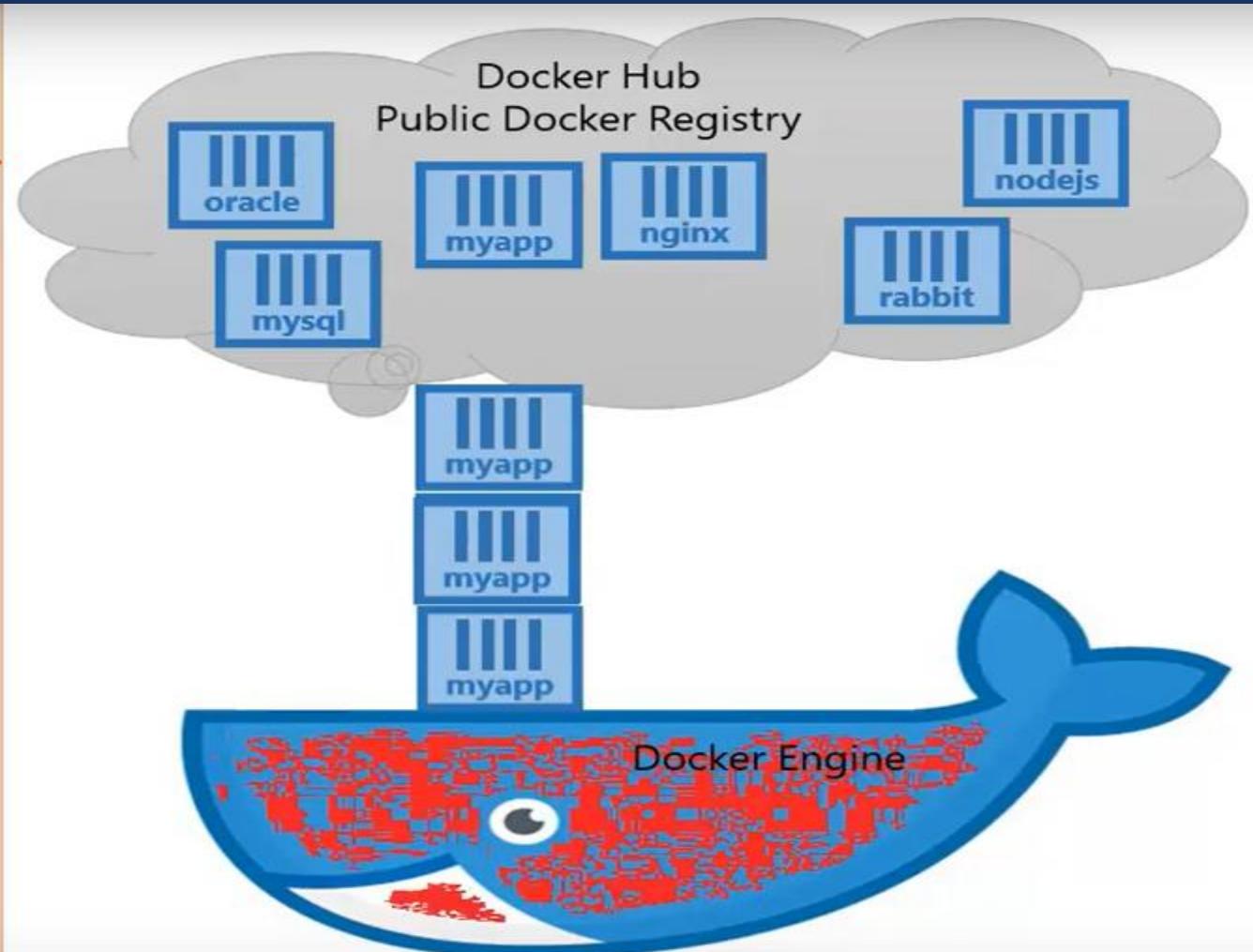


# VIRTUALISATION

## Orchestration des conteneurs

```
$ docker run myapp  
$ docker run myapp  
$ docker run myapp  
$ docker run myapp
```

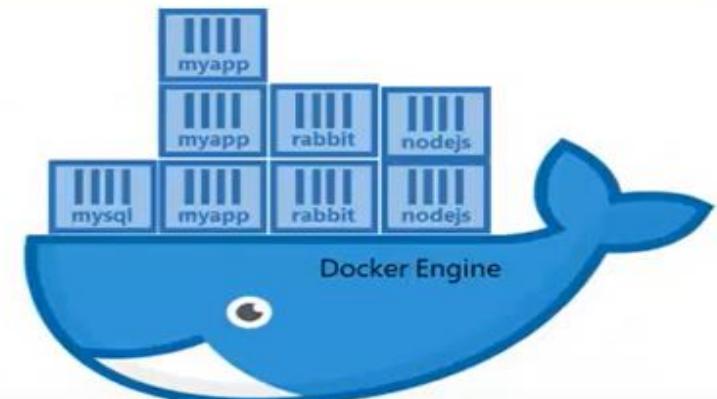
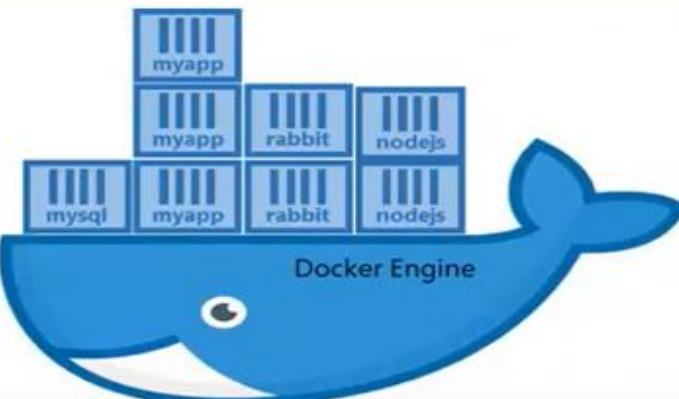
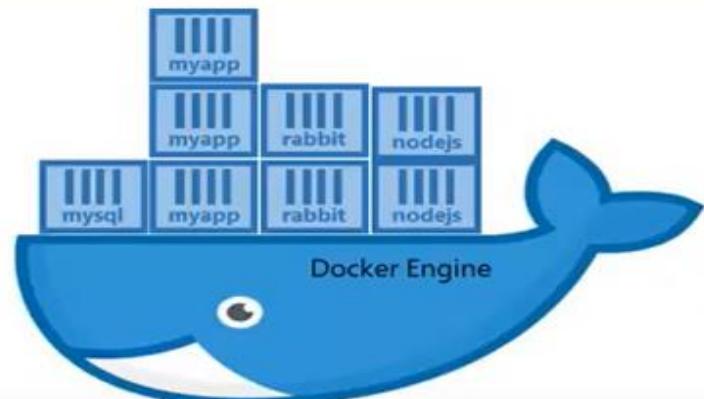
- Problème de Disponibilité et de monté en charge :
  - Il faut démarrer plusieurs instances en fonction de la charge
  - Si une instance tombe en panne, il faut la remplacer
  - Si Docker Engine tombe en panne, tous les conteneur vont cracher.
- L'orchestration des conteneurs est une solution pour ce problème



# VIRTUALISATION

## Orchestration des conteneurs

- Consiste en un ensemble d'outils et de scripts qui permettent l'automatisation des opérations :
  - Scalabilité des conteneurs, RéPLICATION des conteneurs, Haute disponibilité, bilan de santé des conteneurs etc.
- Exemples d'outils :
  - Docker Swarm
  - Kubernetes
  - MESOS



# VIRTUALISATION

## Utilisation de Docker : Kubernetes

### Container Orchestration

It's an open source container orchestration platform, to automate and manage containerized applications.

### Developed at Google

It was developed at Google and released as open source in 2014, So it combines the best practices of running containerized workloads at google for 15 years.

### Cloud Agnostic

you can run it anywhere on bare metal or in any cloud provider infrastructure.

### Zero Downtime Deployment

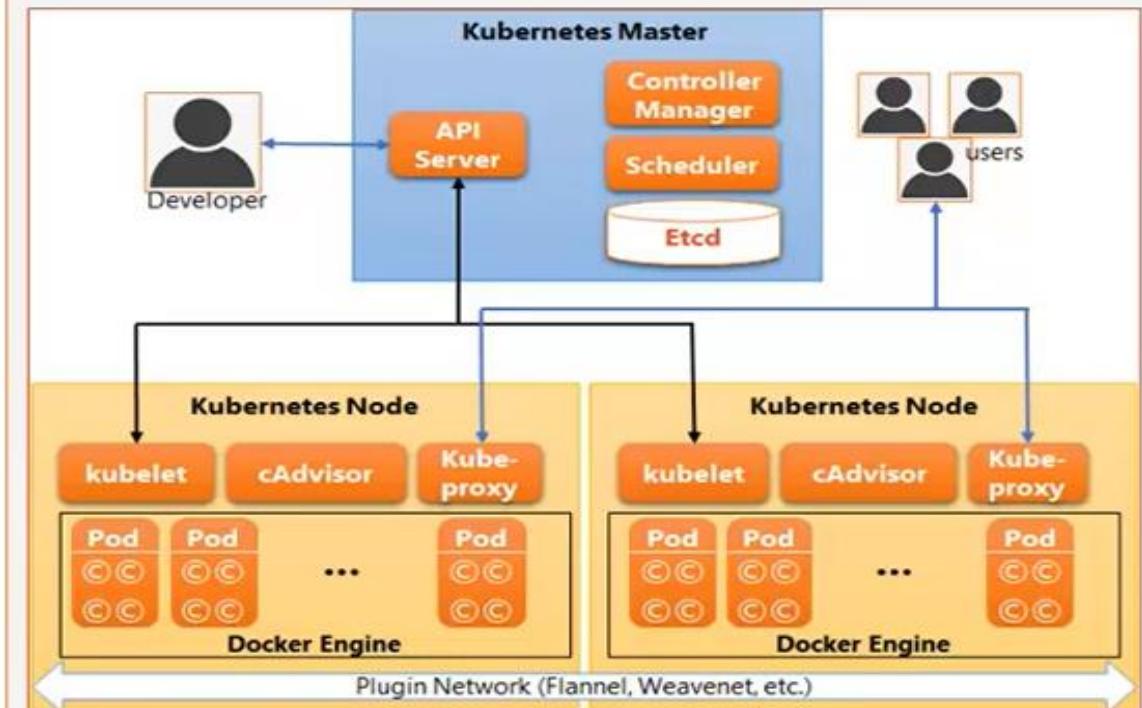
It became the industry standard for deploying containers in production



# VIRTUALISATION

## KUBERNETES

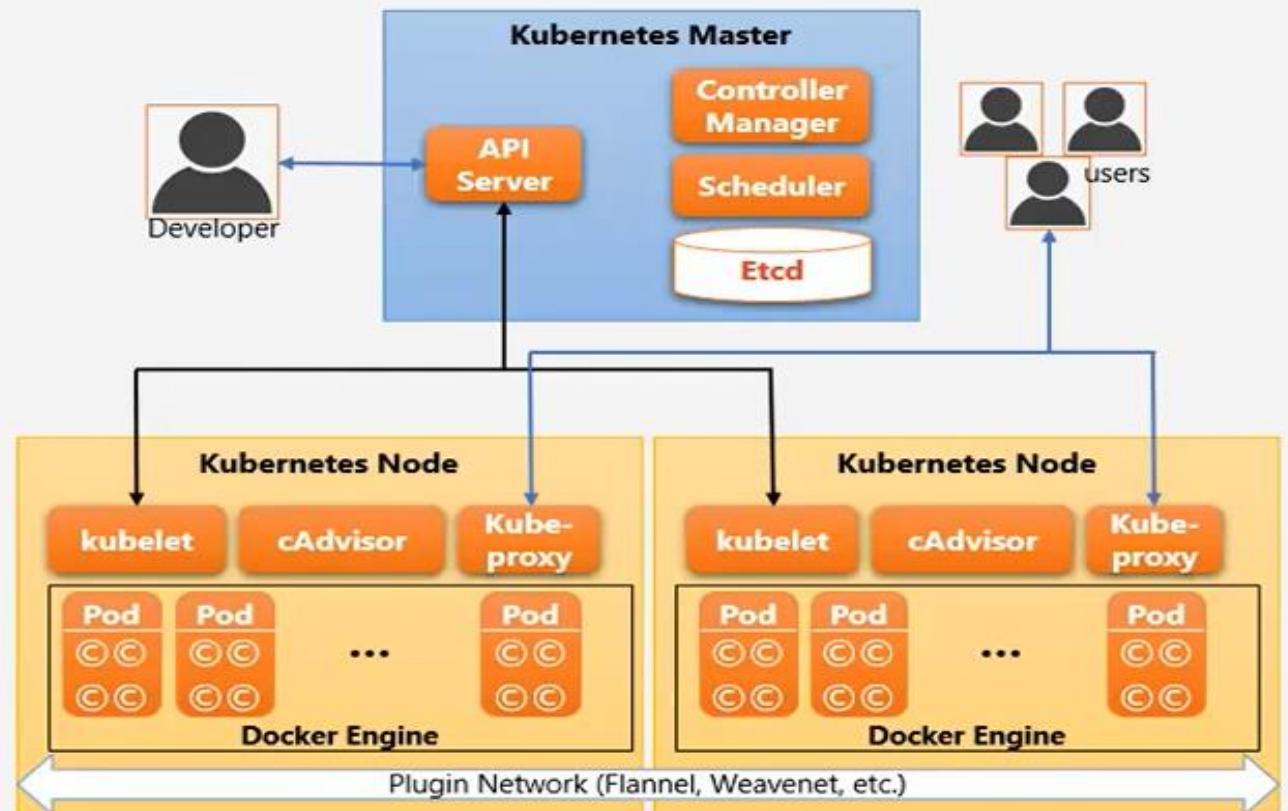
- Outils d'orchestration des conteneurs
- Mot d'origine Grèque signifiant « Timonier », ce qui gouverne la barre du bateau.
- Inspiré du projet Borg dont Google s'en sert depuis des années pour gérer ses conteneurs.
- Développé par Google avec ses partenaires : Redhat, CoreOS, IBM, Microsoft, Mesosphere, VMware, HP, etc...
- Ecrit en langage Go 
- Peut tourner en :
  - Local, Cloud (GCE, Azure, AWS, ..), Machines Physiques
- Kubernetes REST API pour exposer ses fonctionnalités.
- L'objectif est de gérer directement des applications et non pas de machines



# VIRTUALISATION

## Qu'est qu'on peut faire avec KUBERNETES

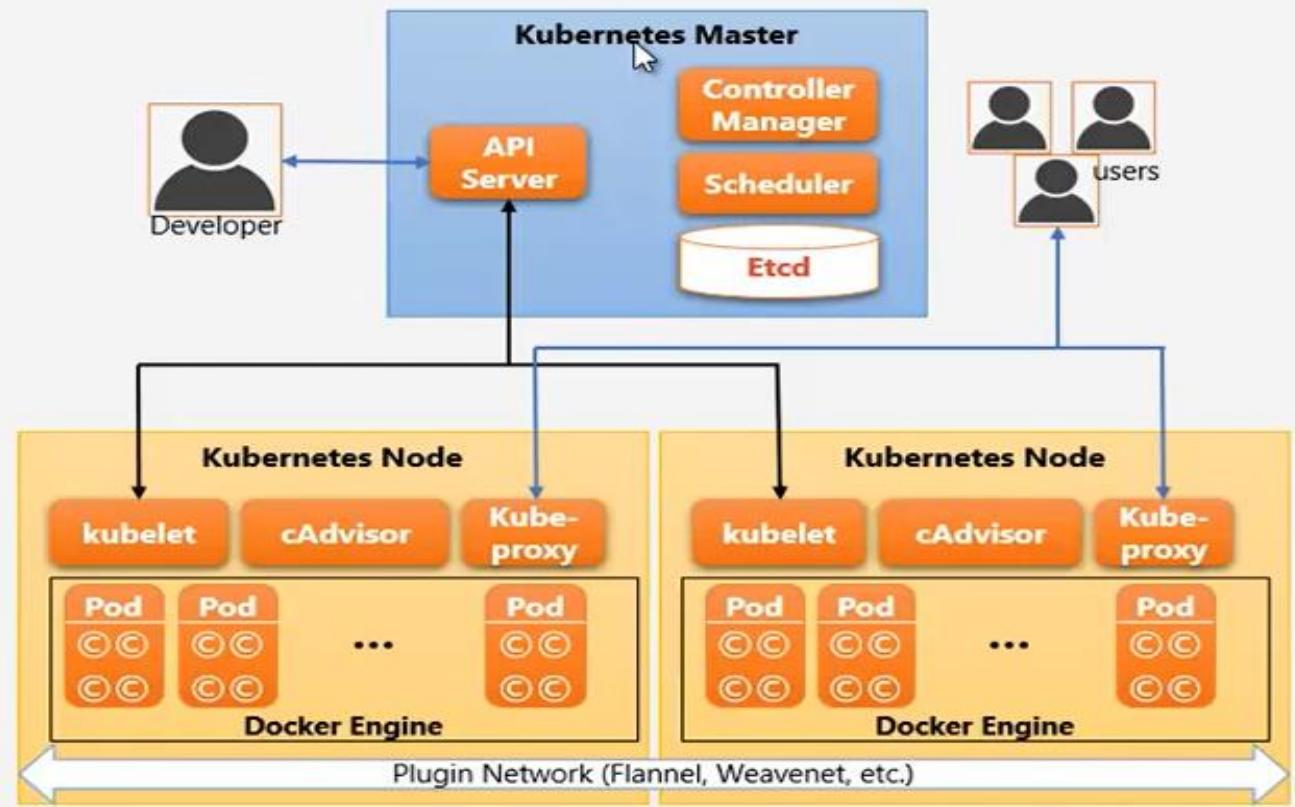
- Déployer une application sous forme de conteneurs d'une manière très rapide et prévisible
- Automatiser le déploiement et la réPLICATION des containers.
- Organiser les conteneurs en groupes et faire du Load banancing
- Déclarer l'architecture cible (Nombre de pods, contenu, stratégie de mise à jour, etc.) et laisser le système atteindre et à maintenir cette cible
- Faire des mises à jour sans interrompre le service
- Séparer l'application de l'architecture sous-jacente.
- DéTECTER les problèmes et les résoudre tout seul



# VIRTUALISATION

## Concepts de base

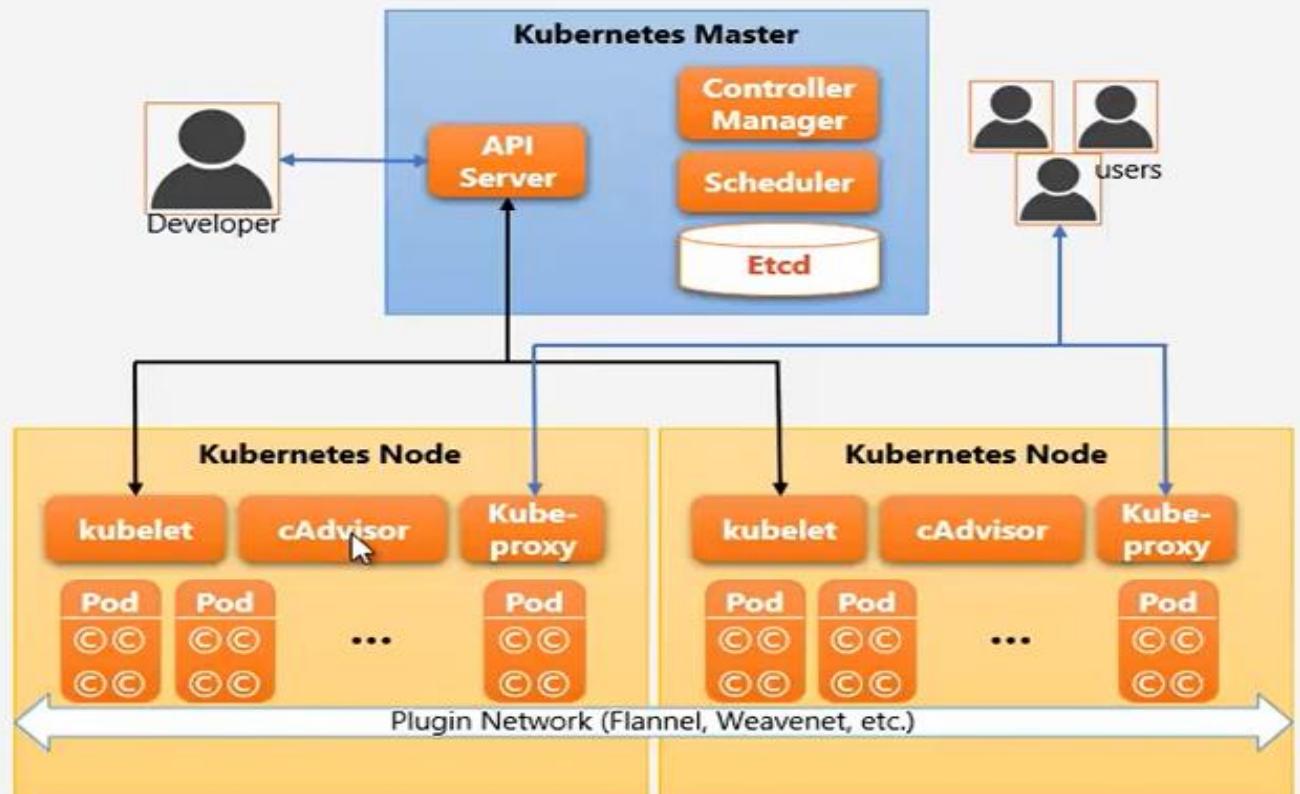
- Cluster : Ensemble de machines physique ou virtuelle utilisées par Kubernetes pour faire tourner des applications.
- Node : Une machine physique ou virtuelle dans laquelle seront déployés des Pods. Chaque Node doit exécuter un Container Engine comme Docker.
- Pod: Groupe de containers déployés ensemble
- Namespace : Segmenter Pod, Volume, Secrêts, ..
- Kubectl : Application en ligne de commande permettant d'intéragir avec Kubernetes.



# VIRTUALISATION

## Architecture de Kubernetes

- Kubernetes suit l'architecture maître/esclave.
- Les composants de Kubernetes peuvent être divisés en ceux qui gèrent un nœud individuel et ceux qui font partie du plan de contrôle.

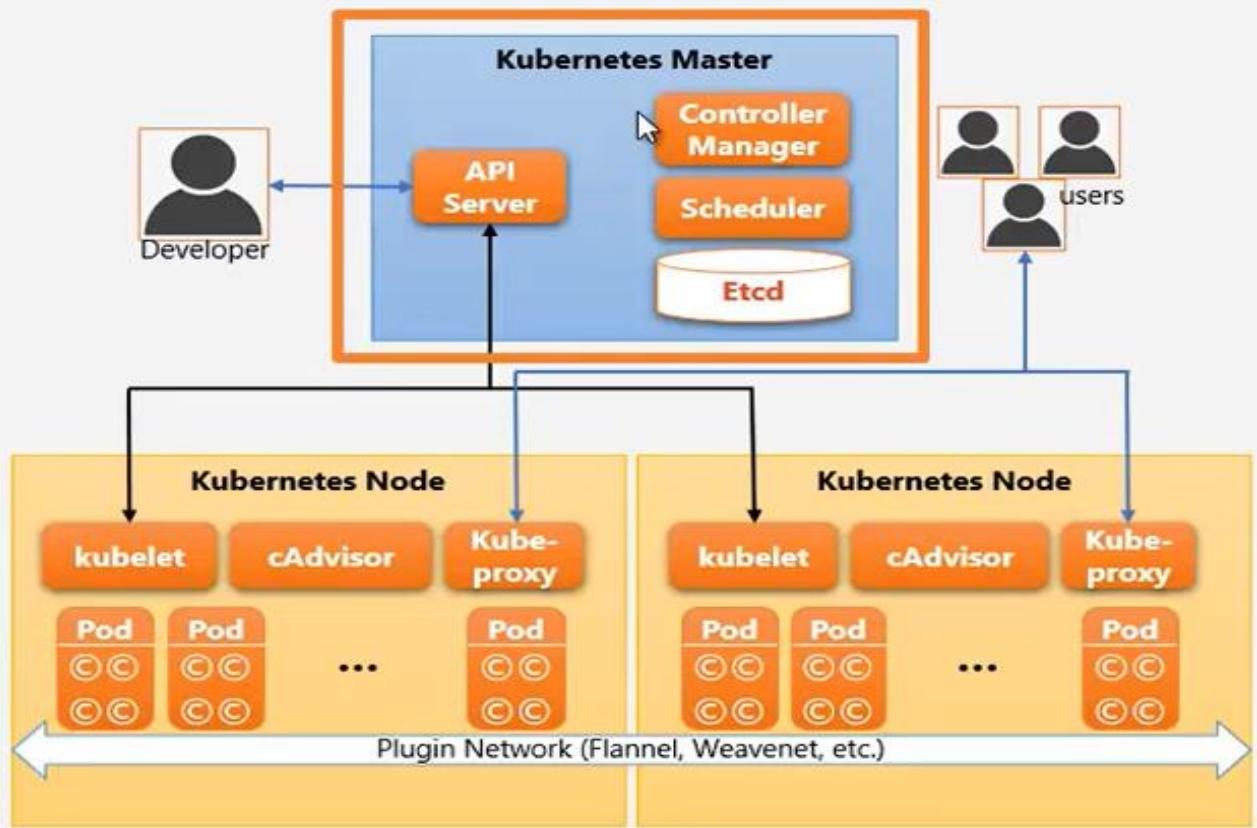


# VIRTUALISATION

## Kubernetes Master

- **Kubernetes Master :**

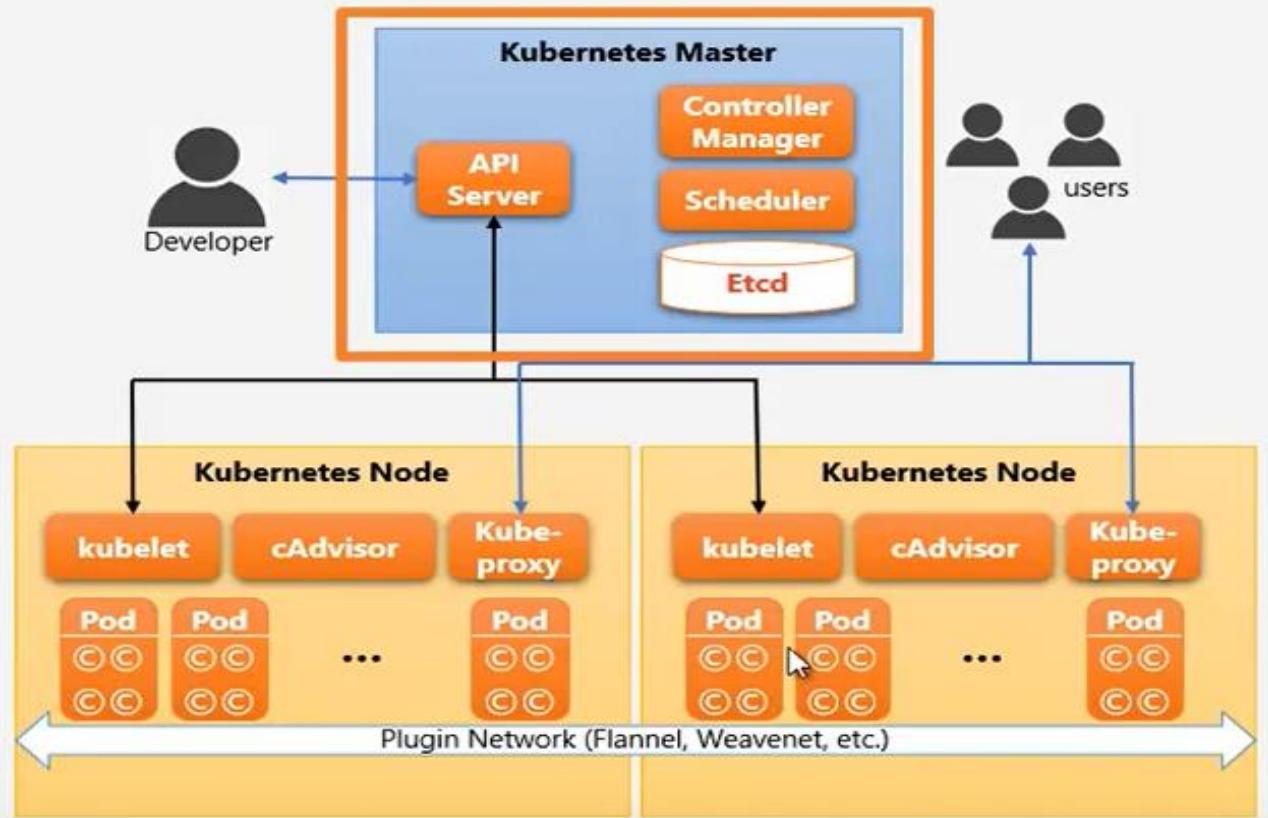
- Unité de contrôle principale qui gère la charge de travail et dirige les communications dans le système.
- Le plan de contrôle de Kubernetes consiste en plusieurs composants, chacun ayant son propre processus, qui peuvent s'exécuter sur un seul Node maître ou sur plusieurs maîtres permettant de créer des clusters haute disponibilité.



# VIRTUALISATION

## Kubernetes Master

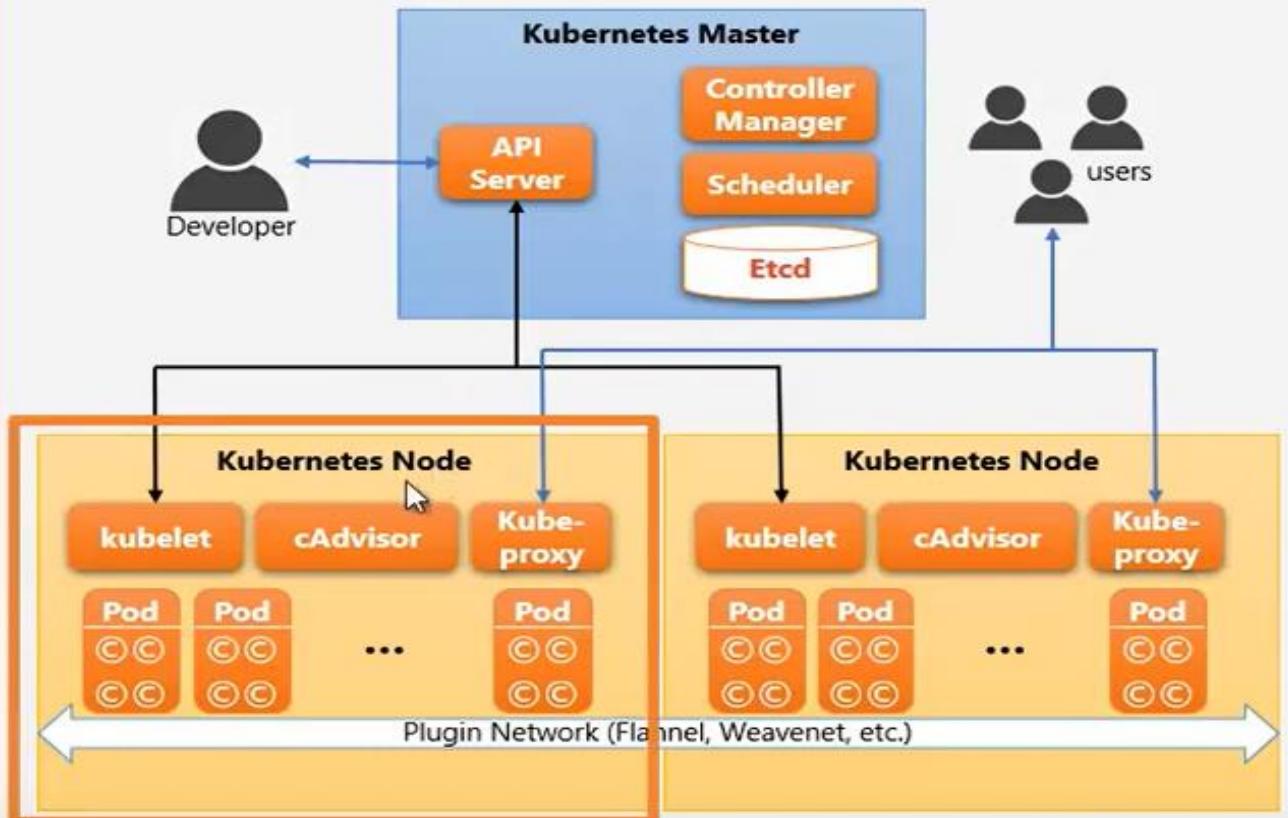
- Les différents composants du plan de contrôle de Kubernetes sont décrits ci-dessous:
  - Etcd : Unité de stockage Distribuée persistante et légère
  - API Server: API REST de communication avec les composants internes et externes
  - Scheduler: L'ordonnanceur qui permet de sélectionner quel Node devrait faire tourner un Pod
  - Controller manager : processus dans lequel s'exécutent les contrôleurs principaux de Kubernetes tels que DaemonSet Controller et le Replication Controller.



# VIRTUALISATION

## Kubernetes Node

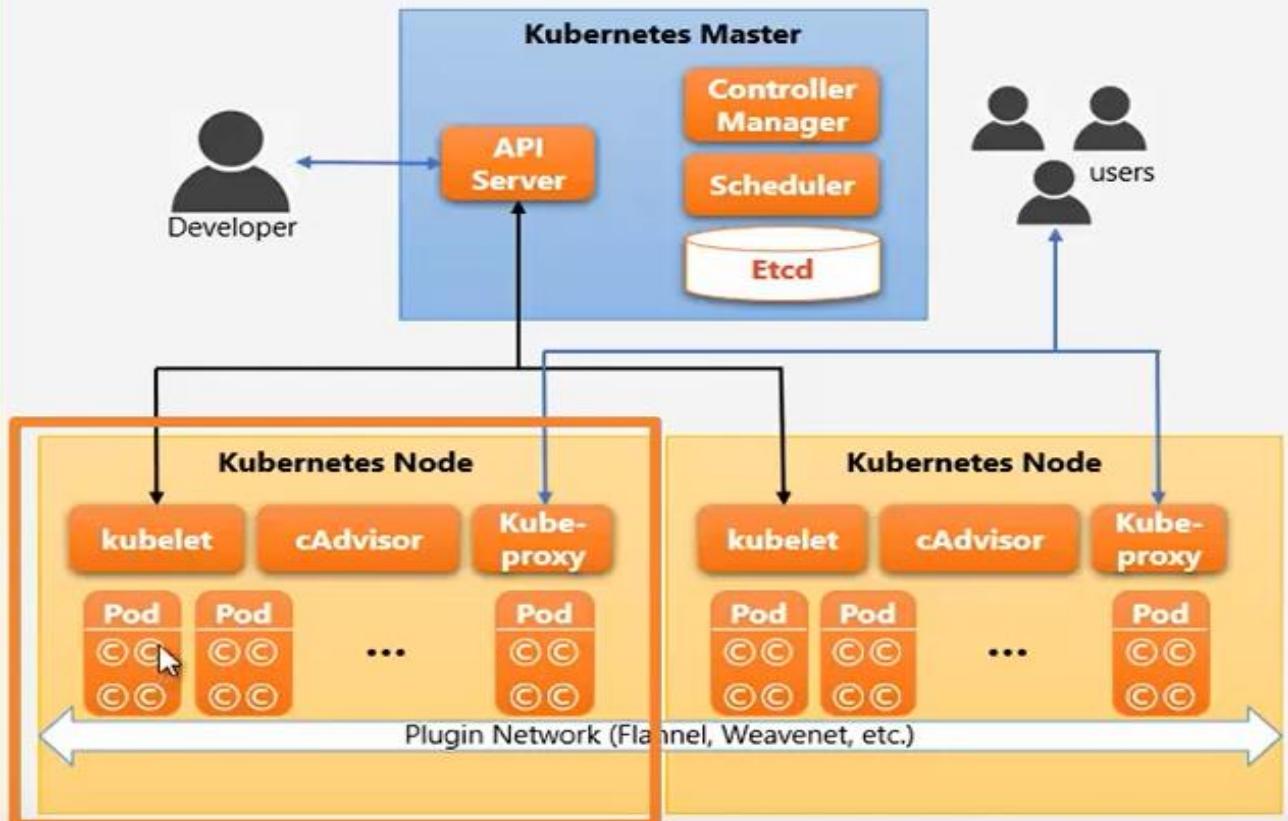
- Le Node aussi appelé Worker est une machine unique (ou une machine virtuelle) où des conteneurs (charges de travail) sont déployés.
- Chaque Node du cluster doit exécuter le programme de conteneurisation (par exemple Docker Engine), ainsi que les composants pour communiquer avec le composants du plan de contrôle du Master.



# VIRTUALISATION

## Kubernetes Node

- Les composants d'un Worker Node de Kubernetes sont:
  - Kubelet : Responsable de l'état d'exécution du nœud ( c'est-à-dire, d'assurer que tous les conteneurs sur un nœud sont en bonne santé organisés en Pods sous la direction du plan de contrôle)
  - Kube-proxy : Responsable d'effectuer le routage du trafic vers le conteneur approprié en se basant sur l'adresse IP et le numéro de port de la requête entrante.
  - cAdvisor : 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 du Node.



# VIRTUALISATION

## Pod

- Groupe de containers déployés ensemble
- Les **containers du Pod** :
  - Sont toujours démarrés, arrêtés et répliqués en groupe
  - Partagent le même Network, Namespace, IP et Ports
  - Peuvent communiquer ensemble en utilisant localhost
  - Peuvent partager des données via des shared volumes
- Le Pod est l'entité de base qu'on va répliquer et qu'on va mettre sur différentes nœuds du cluster.

```
// Créer un pod
$ kubectl create -f mypod.yml
//Lister tous les pods
$ kubectl get pods
//Supprimer un pod
$ kubectl delete Pod_ID
```

# VIRTUALISATION

## Déploiement du contrôleur

- Elément central de Kubernetes
- C'est le fichier dans lequel on décrit la cible que l'on veut atteindre.
  - Les conteneurs qu'on veut mettre dans le Pod
  - Les détails de réPLICATION
  - Les stratégies qu'on va utiliser pour les mises à jour
- C'est le déploiement du contrôleur de Kubernetes qui va se charger de maintenir cet état.
  - Par exemple, s'il y a un Pod qui plante, il va le désactiver et l'enlever du load balancer et créer un autre à côté pour démarrer sur un autre nœud pour que ça puisse continuer à fonctionner dans les mêmes conditions exigées dans le descripteur de déploiement.

myDeployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

```
$ kubectl create -f myDeployment.yaml
```

# VIRTUALISATION

## Déploiement du contrôleur

- Dans cet exemple :
  - Un déploiement nommé nginx est créé, indiqué par le champ metadata: name.
  - Le déploiement crée trois pods répliqués, indiqués par le champ replicas.
  - Le modèle de pod, ou le champ spec: template, indique que ses pods sont libellés app: nginx.
  - La spécification du modèle de pod, ou du champ template: spec, indique que les pods exécutent un conteneur, nginx, qui exécute l'image Docker Hub nginx dans sa version 1.7.9.
  - Le déploiement ouvre le port 80 pour une utilisation par les pods.
- En résumé, le modèle de pod contient les instructions suivantes pour les pods créés par ce déploiement :
  - Chaque pod est libellé app: nginx.
  - Créez un conteneur et nommez-le nginx.
  - Exécutez l'image nginx dans la version 1.7.9.
  - Ouvrez le port 80 pour envoyer et recevoir du trafic.

## Deployment.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

# VIRTUALISATION

## Installing Kubernetes with Minikube

<https://kubernetes.io/docs/tasks/tools/install-minikube/>

- **Kubectl** est un outil en ligne de commande de kubernetes, kubectl, vous permet d'exécuter des commandes dans les clusters Kubernetes. Vous pouvez utiliser kubectl pour déployer des applications, inspecter et gérer les ressources du cluster et consulter les logs, ....
- **Minikube**, qui est un outil qui fait tourner un cluster Kubernetes à un nœud unique dans une machine virtuelle sur votre machine. C'est une machine virtuelle qui avec l'essentiels des packages de:
  - Docker
  - Kubernetes

- Installer cubectl :
  - <https://kubernetes.io/docs/tasks/tools/install-kubectl/>
- Installer miniKube :
  - <https://kubernetes.io/docs/tasks/tools/install-minikube/>

## VIRTUALISATION

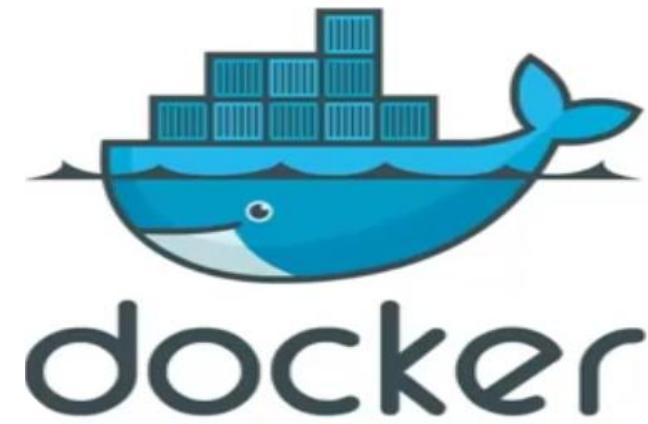
### Travail Pratique II

#### Objectifs :

- Etudier et pratiquer l'outils de containerisations Docker , Kubernetes

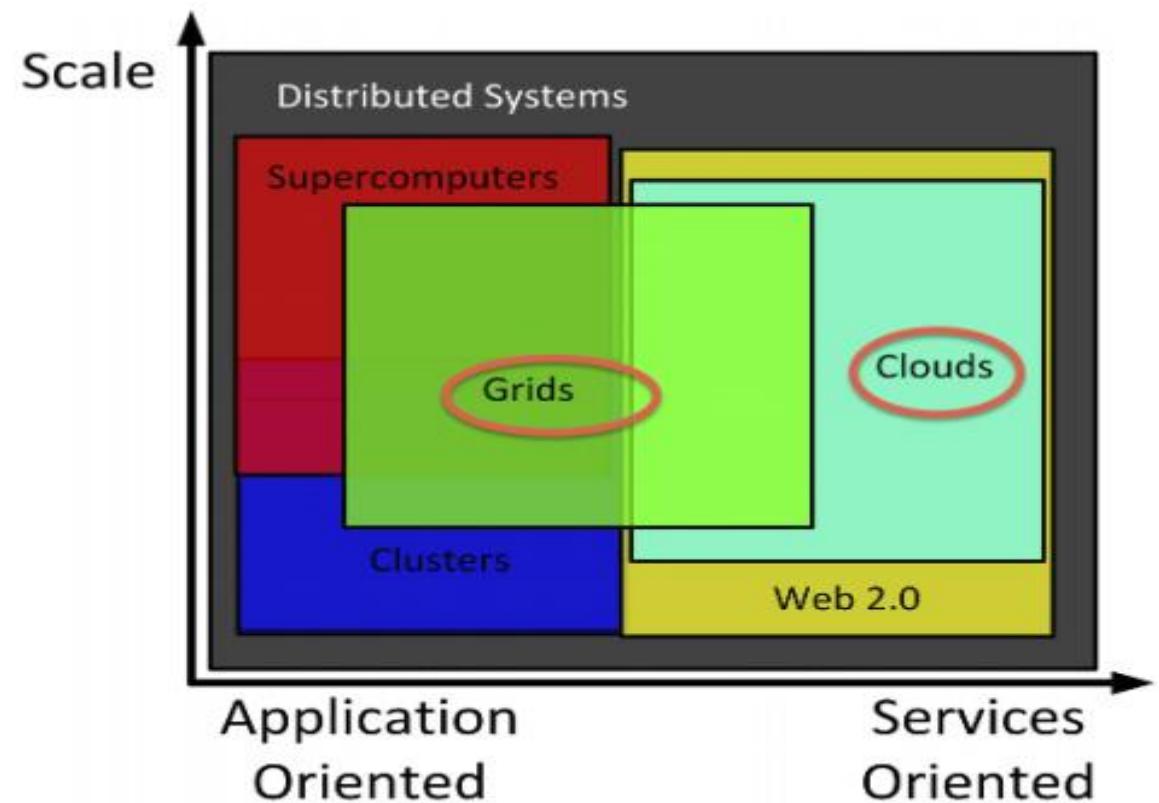
#### Travail à faire :

- I. Réaliser une application web sur une configuration locale (PHP, JSP, python ... et base de données relationnelle).
- II. Installer l'outil de containerisation Docker,
- III. Isoler l'application sur un container Docker.
- IV. Utiliser Kubernetes pour faire des réplications

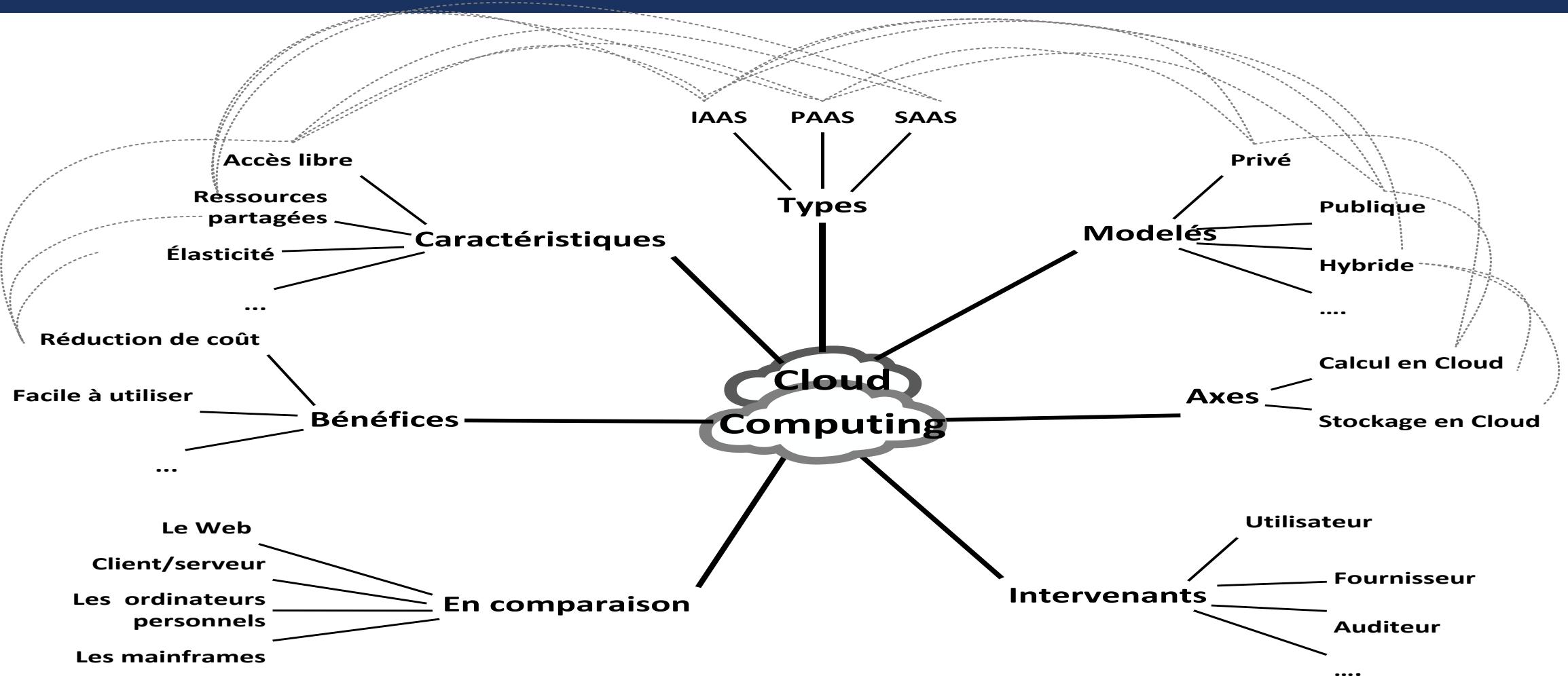


## INTRODUCTION GÉNÉRALE

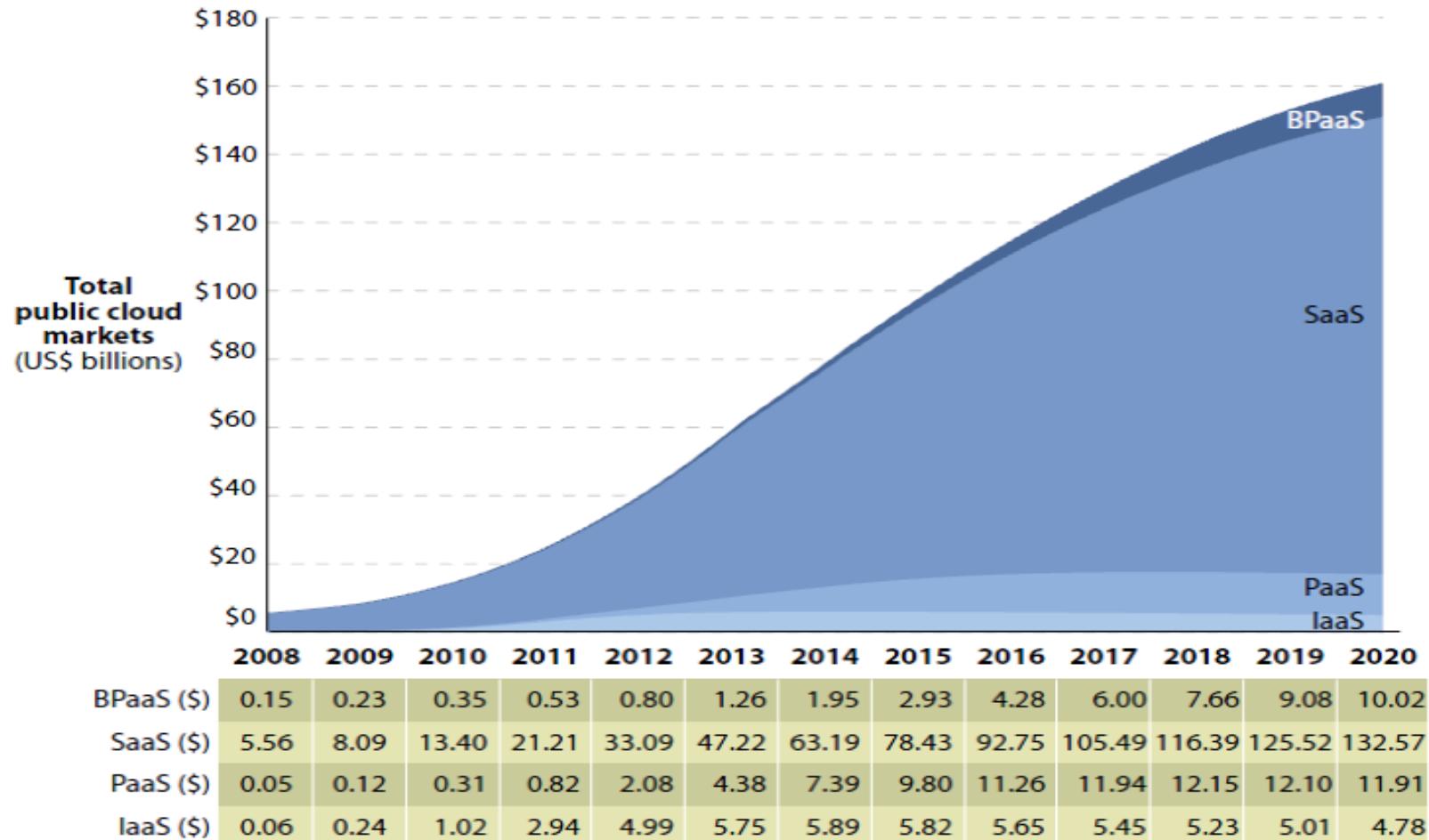
- Les années 70 ASP « Application Service Provider »
- Les années 90 Grid Computing
- L'évolution de la virtualisation,
- Les travaux de IBM
- L'année 2006 Amazon EC2
- 2009 la vrai explosion du Cloud
  - IBM Smart Business Service (de IBM),
  - Google App Engine (de Google)
  - Microsoft Azure (de Microsoft)
  - Sun Cloud (de Sun)



# INTRODUCTION GÉNÉRALE



# INTRODUCTION GÉNÉRALE



## DÉFINITION

### **Selon l’Institut national des normes et de la technologie (NIST)**

« Le Cloud Computing est un modèle qui permet d'accéder rapidement à un pool de ressources informatiques mutualisées et à la demande ».

**Selon IBM** « le Cloud Computing, souvent appelé simplement «nuage», est une fourniture de ressources informatiques à la demande, des applications aux centres de données, en passant par l'Internet et le paiement à l'utilisation » .

# CARACTÉRISTIQUES DU CLOUD COMPUTING

## Aspect technique

- Basé Virtualisation
- Ressources partagées
- Service doit être accessible via un réseau
- Service mesuré
- Basé SLA (Service Level agreement)
- Sécurité efficace

## Aspect qualitatif

- Disponibilité
- Agilité
- Élasticité rapide
- Fiabilité : Simplicité, flexibilité et tolérance aux pannes

## Aspect économique

- Dépenses opérationnelles
- Efficacité énergétique
- Paiement à l'usage
- Accès libre à la demande

## CARACTÉRISTIQUES DU CLOUD COMPUTING

### Aspect technique

- Basé Virtualisation
- Ressources partagées
- Service mesuré
- Service doit être accessible via un réseau
- Basé SLA (Service Level agreement)
- Sécurité efficace

## CARACTÉRISTIQUES DU CLOUD COMPUTING

### Aspect qualitatif

- Disponibilité
- Agilité
- Élasticité rapide
- Fiabilité : Simplicité, rapidité et tolérance aux pannes

## CARACTÉRISTIQUES DU CLOUD COMPUTING

### Aspect économique

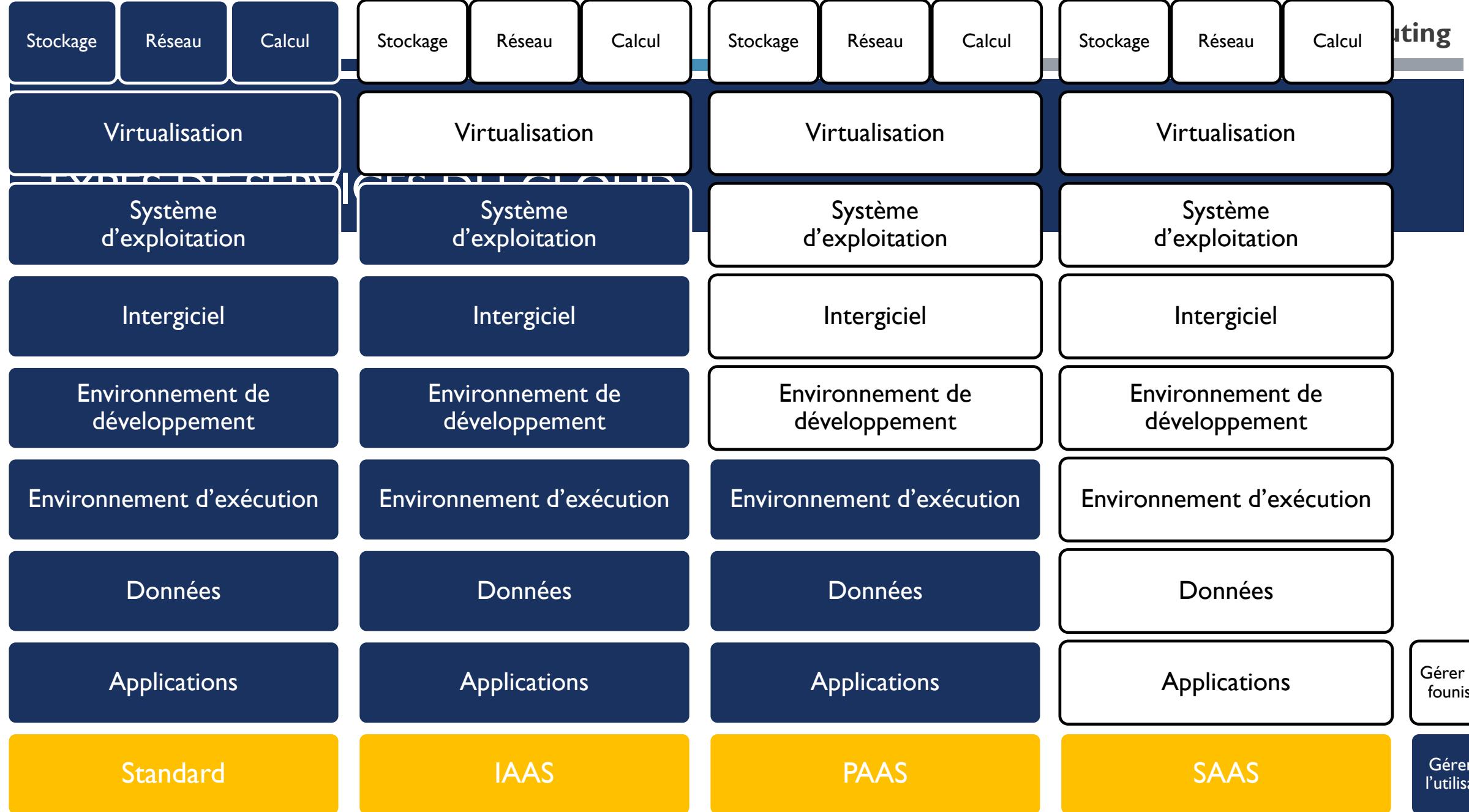
- Dépenses opérationnelles
- Paiement à l'usage
- Accès libre à la demande
- Efficacité énergétique

## LES INTERVENANTES DU CLOUD

- **Fournisseur du Cloud**
- **Utilisateur du Cloud**
- **Fournisseur du réseau**
- **Courtier Cloud**
- **Auditeur Cloud**

## TYPES DE SERVICES DU CLOUD

- **IaaS (Infrastructure as a Service)**
- **PaaS (Platform as a Service)**
- **SaaS (Software as a Service)**
- **XaaS (Anything-as-a-Service)**



## TYPES DE SERVICES DU CLOUD

### ➤ IaaS (Infrastructure as a Service)

- **Ec2, HDFS (Hadoop Distributed File System), S3 (Simple Storage Service), OpenStack ...**
- **Avantage :**
  - Administration,
  - Capacité de stockage infini,
  - Grande flexibilité d'utilisation,
  - Contrôle total des systèmes
- **Inconvénient :**
  - Sécurité ,
  - investissements très élevés,
  - besoin d'administrateurs système comme pour les solutions de serveurs classiques sur site.

## TYPES DE SERVICES DU CLOUD

### ➤ PaaS (Platform as a Service)

- ***Google App Engine, Elastic Beanstalk ...***
- **Avantage :**
  - le déploiement est automatisé,
  - pas de logiciel supplémentaire à acheter ou à installer,
  - Pas d'infrastructure nécessaire,
  - Environnement hétérogène
- **Inconvénient :**
  - Les cibles sont les développeurs.
  - limitation des technologies (ex. : Python ou Java pour Google AppEngine, .NET pour Microsoft).

## TYPES DE SERVICES DU CLOUD

### ➤ SaaS (Software as a Service)

- les outils ERP, serveurs web, outils collaboratifs, serveurs de messagerie (Yahoo Email, Gmail, ...), Google Apps, Dropbox, office 365 (outil collaboratif), Salesforce CRM, Google Documents, Facebook, Twitter, MobileMe...
- **Avantage :**
  - Pas d'installation et plus de licence,
  - Accessible via un abonnement,
  - plus de mise à jour (elles sont continues chez le fournisseur),
  - plus de migration de données etc.
  - Paiement à l'usage.
  - Test de nouveaux logiciels avec facilité.
- **Inconvénient :**
  - Logiciel limité
  - Sécurité
  - Dépendance au fournisseur

## TYPES DE SERVICES DU CLOUD

- **XaaS (Anything-as-a-Service)**
  - SecaaS Security as a Service, BaaS Backup as a Service, CaaS Communication as a Service, DBaaS Database as a Service, MaaS Monitoring as a Service...

## MODÈLES DE DÉPLOIEMENT DU CLOUD COMPUTING

- **Cloud privé**
- **Cloud public**
- **Cloud communautaire**
- **Cloud hybride**

## MODÈLES DE DÉPLOIEMENT DU CLOUD COMPUTING

### ➤ Cloud privé

- *Cloud privé typique*
- *Cloud privé géré*
- *Cloud privé hébergé*
- *Cloud privé virtuel*

## MODÈLES DE DÉPLOIEMENT DU CLOUD COMPUTING

### ➤ **Cloud public**

- Le consommateur et le fournisseur de service sont des organisations différentes
  - Amazon AWS, Google Cloud, Microsoft Azure, BlueCloud de IBM ...

## MODÈLES DE DÉPLOIEMENT DU CLOUD COMPUTING

### ➤ **Cloud communautaire**

- ou Cloud collectif est pour partager l'infrastructure par plusieurs organisations indépendantes ayant des intérêts communs.

➤ Le modèle fédéré

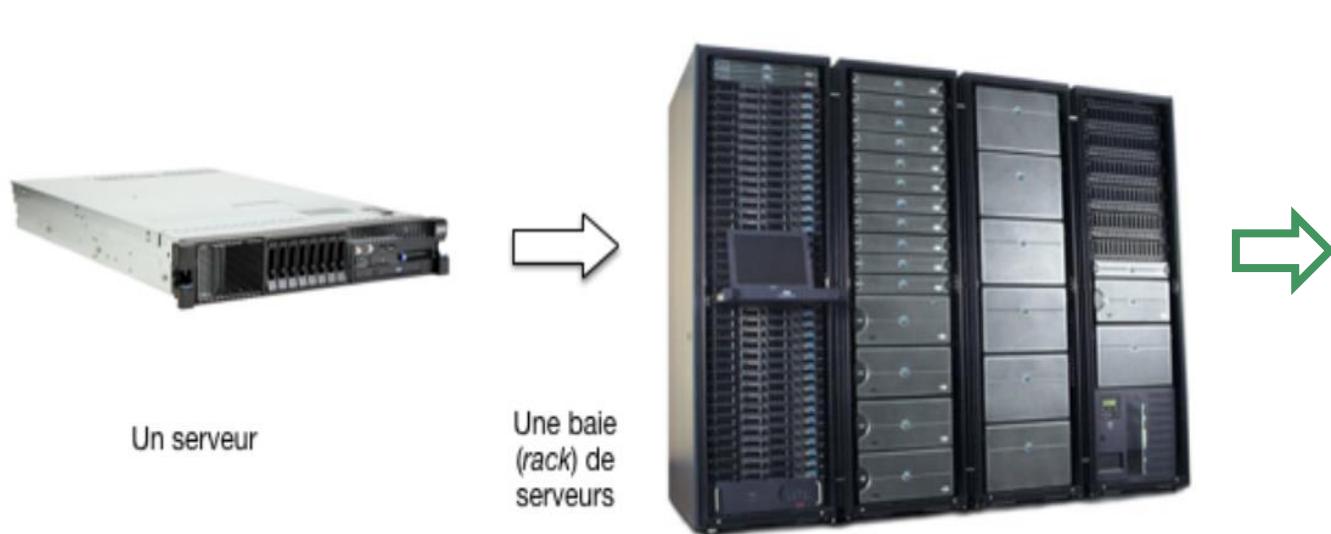
➤ Le tiers de confiance "broker"

## MODÈLES DE DÉPLOIEMENT DU CLOUD COMPUTING

### ➤ **Cloud hybride**

- Le Cloud hybride pourrait être constitué de n'importe quelle combinaison des trois types public, privé et communautaire.

# LA COUCHE MATÉRIELLE



Une grappe de serveurs

# DIFFÉRENCES ENTRE GRILLES ET CLOUDS

| Critère                   | Grid Computing   | Cloud Computing                                       |
|---------------------------|--|---|
| Objectif                  | Partage collaboratif des ressources                                | Utilisation des services (niveau d'abstraction élevé) |
| Axes de calcul            | Opérations de calcul intensif                                      | Instances standards et de haut niveau                 |
| Gestion du Workflow       | Sur un nœud physique   | En instance EC2 ( Amazon EC2 + S3 )                   |
| Découverte                | Indexation centralisée et des services d'information décentralisés | Services aux membres                                  |
| La gestion des ressources | Distribuée   | centralisée/distribuée                                |
| Allocation/Ordonnancement | Décentralisé   | centralisé/décentralisé                               |
| Interopérabilité          | Normes standards de grille   | Web Services (SOAP et REST)                           |
| Niveau d'abstraction      | Faible(plus de détails)  | Haut(éliminer les détails)                            |
| Degré de scalabilité      | Normale  | Haute   |
| Multitâche                | Oui  | Oui   |
| Transparence              | Bas  | Haut  |
| Temp d'exécution          | N'est pas en temps réel  | Services en temps réel                                |
| Unité d'allocation        | Petite tâche   | Toutes formes et tailles                              |
| Virtualisation            | N'est pas une matière première                                     | Vitale  |
| Portail accessible        | Via un système DNS   | Via IP uniquement                                     |
| Sécurité                  | Faible (service de certificat de grille)                           | Haute (virtualisation)                                |
|                           |  | Un hyperviseur (VM) sur lequel peut tourner           |

# DIFFÉRENCES ENTRE GRILLES ET CLOUDS

| Critère                | Grid Computing   | Cloud Computing  |
|------------------------|--|--|
| Système d'exploitation | N'importe quel standards   | Un hyperviseur (VM) sur lequel peut tourner plusieurs systèmes |
| Propriétaire           | Multiple   | Unique   |
| Tolérance aux pannes   | Limité   | Forte(La migration)  |
| Prix des services      | Dominé par le bien public ou privé                                       | Prix des services publics, prix réduit pour les gros clients   |
| Facilité d'utilisation | Faible   | Elevée   |
| Type de service        | CPU, réseau, mémoire, bande passante, dispositif, stockage,...           | IaaS, PaaS, SaaS, tout est service                             |
| Exemple du monde réel  | BOINC, XtremWeb, SETI,...  | AWS, Google app ...  |
| Temps de réponse       | ne peut pas être déterminé   | En temps réel  |
| Objet critique         | Ressources informatique  | Services   |
| Nombre d'utilisateurs  | peu  | Plus   |
| Ressources             | Limité (car le matériel est limité)                                      | Illimité   |
| Configuration          | Difficile car les utilisateurs n'ont pas des priviléges d'administrateur | Très facile à configurer                                       |
| Future                 | Cloud Computing  | Prochaine génération d'Internet Smart cities peut être,...     |

## LES PRINCIPAUX FOURNISSEURS DE CLOUD

- **Amazon Web Services (AWS)**
- **Google Cloud Platform**
- **Microsof Azure**
- **IBM**
- **SalesForce ...**

## LES PRINCIPAUX FOURNISSEURS DE CLOUD

### ➤ **Amazon Web Services (AWS)**



- En 2002, "Amazon Web Services".
- Le catalogue de services s'est enrichi avec le temps, et on compte des centaines d'offres de services
- « Leaders » de cloud pour toutes les années.
- Avec *430 milliards d'€ de chiffre d'affaires* ; *28,8 milliards d'€ de bénéfices* en 2024

## LES PRINCIPAUX FOURNISSEURS DE CLOUD

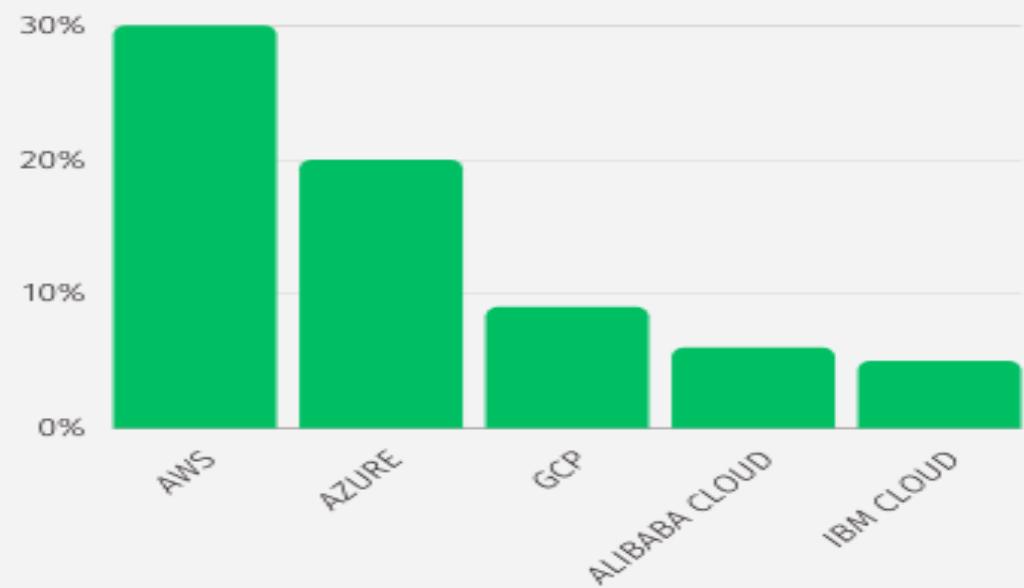
### ➤ Amazon Web Services (AWS)



**testprep**  
training

### Top 5 Cloud Market Leaders in 2023

according to their market share ->



# LES PRINCIPAUX FOURNISSEURS DE CLOUD

## ➤ Amazon Web Services (AWS)



The screenshot shows the AWS Management Console homepage. The left sidebar includes links for History, Console Home, EC2, IAM, Billing, and AWS Organizations. The main content area displays a grid of service icons and names. The services are categorized as follows:

- Compute:** Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts, EC2 Image Builder.
- Storage:** S3, EFS, FSx, S3 Glacier, Storage Gateway, AWS Backup.
- Blockchain:** Amazon Managed Blockchain.
- Satellite:** Ground Station.
- Quantum Technologies:** Amazon Braket.
- Management & Governance:** AWS Organizations, CloudWatch, AWS Auto Scaling, CloudFormation, CloudTrail, Config, AWS Lambda.
- Analytics:** Athena, EMR, CloudSearch, Elasticsearch Service, Kinesis, QuickSight.
- Business Applications:** Alexa for Business, Amazon Chime, WorkMail, Amazon Honeycode.
- End User Computing:** WorkSpaces, AppStream 2.0, WorkDocs, WorkLink.
- Security, Identity, & Compliance:** IAM, Resource Access Manager, Cognito, Secrets Manager.
- Internet Of Things:** IoT Core, FreeRTOS, IoT 1-Click, IoT Analytics.

The URL in the browser bar is <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1>.

## LES PRINCIPAUX FOURNISSEURS DE CLOUD

### ➤ Amazon Web Services (AWS)



Calcul



Stockage



Base de données



Migration



Mise en réseau et diffusion de contenu



Outils pour développeurs



Outils de gestion



Sécurité, identité et conformité



Analyses



Intelligence artificielle



Services mobiles



Services applicatifs



Messagerie



Productivité d'entreprise



Streaming de bureau et d'application



Internet des objets



Centre d'appels



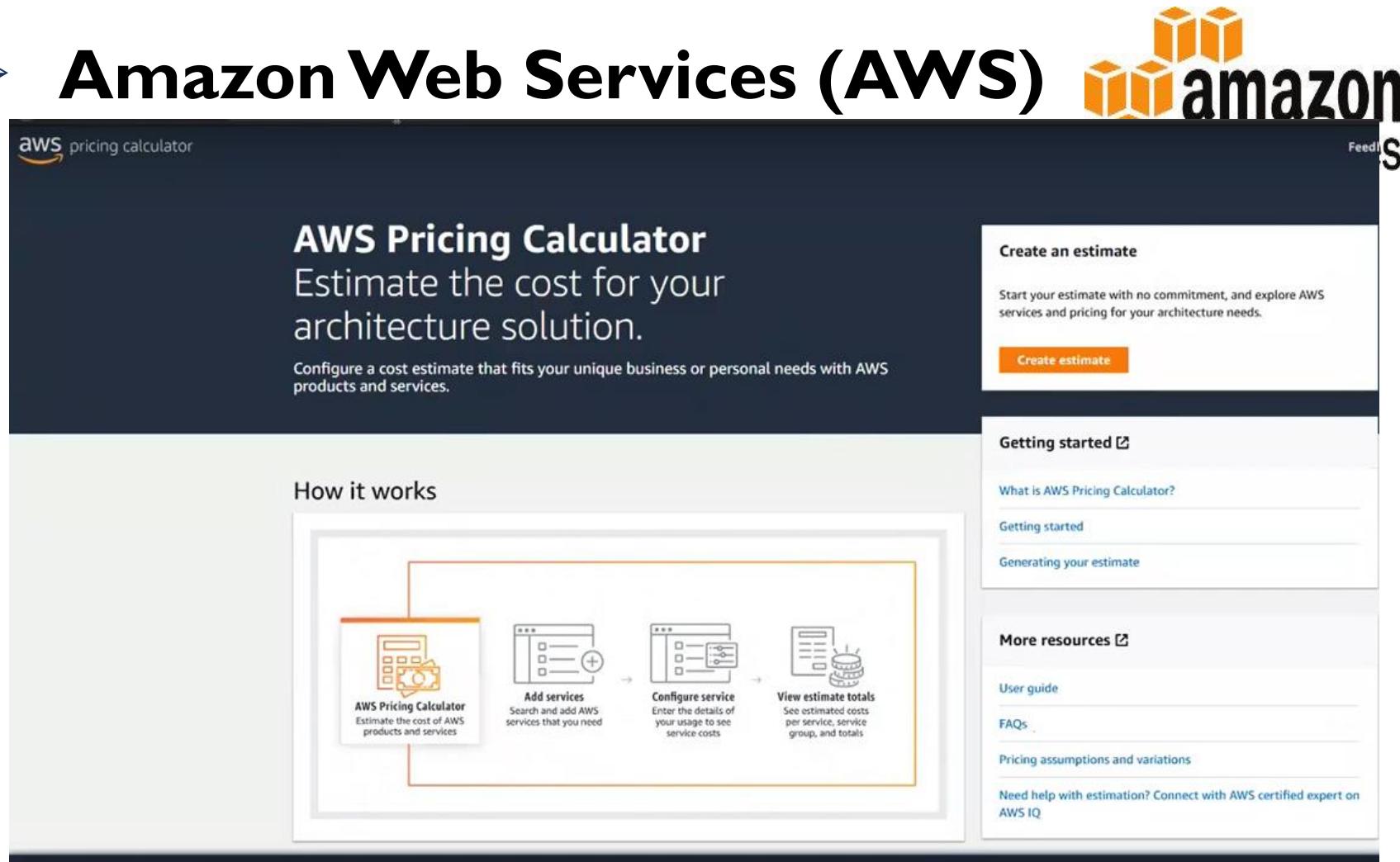
Développement de jeux



Afficher tous les produits

## LES PRINCIPAUX FOURNISSEURS DE CLOUD

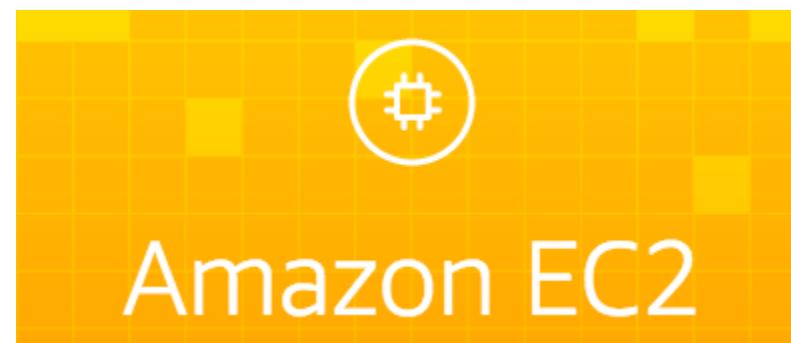
### ➤ Amazon Web Services (AWS)



The screenshot shows the AWS Pricing Calculator interface. At the top, there's a navigation bar with the AWS logo and a search bar. Below it, the main title is "AWS Pricing Calculator" followed by the subtitle "Estimate the cost for your architecture solution." A sub-instruction says "Configure a cost estimate that fits your unique business or personal needs with AWS products and services." On the left, a section titled "How it works" shows a four-step process: 1. AWS Pricing Calculator (Estimate the cost of AWS products and services), 2. Add services (Search and add AWS services that you need), 3. Configure service (Enter the details of your usage to see service costs), and 4. View estimate totals (See estimated costs per service, service group, and totals). To the right, there are three main sections: "Create an estimate" (with a "Create estimate" button), "Getting started" (with links to "What is AWS Pricing Calculator?", "Getting started", and "Generating your estimate"), and "More resources" (with links to "User guide", "FAQs", "Pricing assumptions and variations", and "Need help with estimation? Connect with AWS certified expert on AWS IQ").

## LES PRINCIPAUX FOURNISSEURS DE CLOUD

- **Amazon Web Services (AWS)** 
- **EC2 : Elastic Compute Cloud**

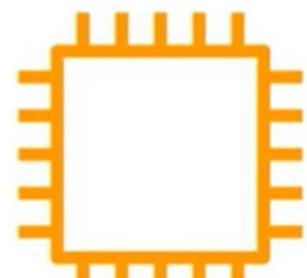


## LES PRINCIPAUX FOURNISSEURS DE CLOUD

### ➤ Amazon Web Services (AWS)



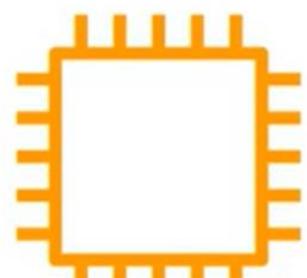
### EC2 Pricing Model



On-demand



Spot  
Instance



Reserved

## LES PRINCIPAUX FOURNISSEURS DE CLOUD

### ➤ Amazon Web Services (AWS)



#### On-demand Instances

Active



Stopped/Terminated

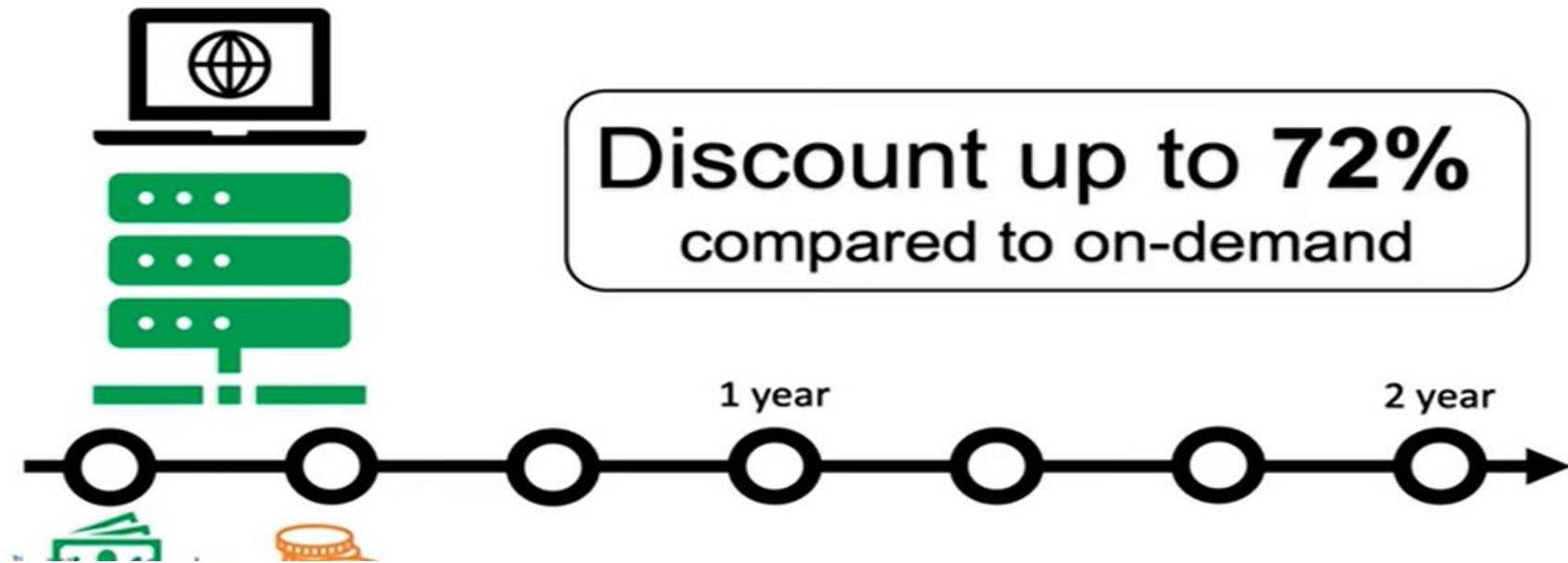


## LES PRINCIPAUX FOURNISSEURS DE CLOUD

### ➤ Amazon Web Services (AWS)



#### Reserved Instances



## LES PRINCIPAUX FOURNISSEURS DE CLOUD

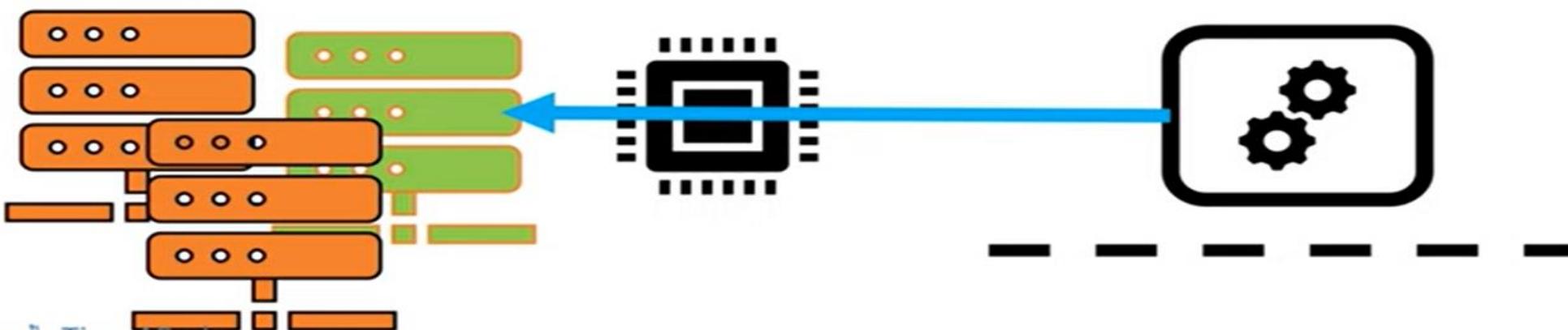
### ➤ Amazon Web Services (AWS)



#### ⌚ Spot Instances



Discount up to 90%



# LES PRINCIPAUX FOURNISSEURS DE CLOUD

## ➤ Amazon Web Services (AWS)



**Configure Amazon EC2** [Info](#)

|                                  |           |       |   |                 |          |        |     |
|----------------------------------|-----------|-------|---|-----------------|----------|--------|-----|
| <input type="radio"/>            | t3.micro  | 1 GiB | 2 | Up to 5 Gigabit | EBS only | 0.0104 | Yes |
| <input type="radio"/>            | t2.micro  | 1 GiB | 1 | Low to Moderate | EBS only | 0.0116 | Yes |
| <input type="radio"/>            | t4g.small | 2 GiB | 2 | Up to 5 Gigabit | EBS only | 0.0168 | Yes |
| <input checked="" type="radio"/> | t3a.small | 2 GiB | 2 | Up to 5 Gigabit | EBS only | 0.0188 | Yes |

**Pricing strategy** [Info](#)

Pricing model

- EC2 Instance Savings Plans**  
EC2 Instance Savings Plans provides a significant discount when you commit to an hourly spend on an instance family in a particular region. Estimate could be a mix of EC2 Instance Savings Plans and On-Demand pricing for the best value and performance.
- Compute Savings Plans**  
Compute Savings Plans provide a discount on any EC2 Instance when you commit to an hour of usage. Estimate could be a mix of Compute Savings Plans and On-Demand pricing for the best value and performance.
- On-Demand and Reserved**  
A mix of Reserved and On-Demand pricing for the best value and performance.
- All Reserved**  
Reserved Instances provide a significant discount (up to 75%) compared to on-Demand instances.

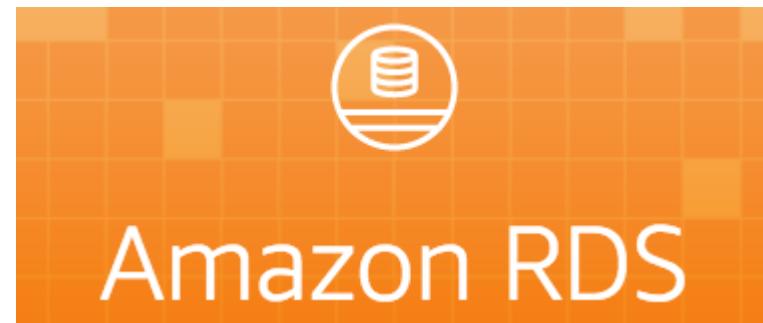
Total Upfront cost: 0.00 USD  
 Total Monthly cost: 17.60 USD

Show Details ▾

Save and view summary

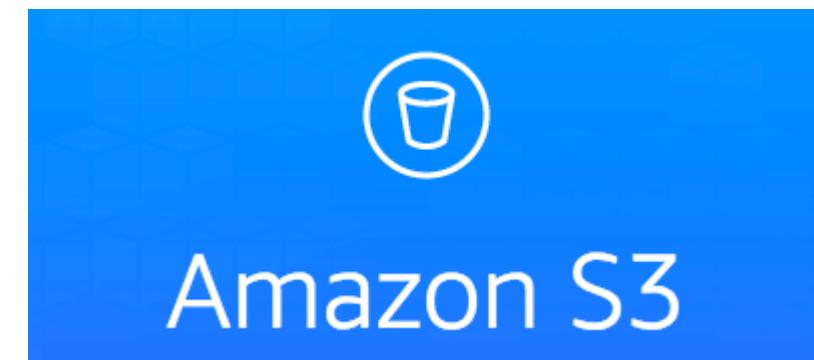
## LES PRINCIPAUX FOURNISSEURS DE CLOUD

- **Amazon Web Services (AWS)** 
- **RDS : Relational Database Service**



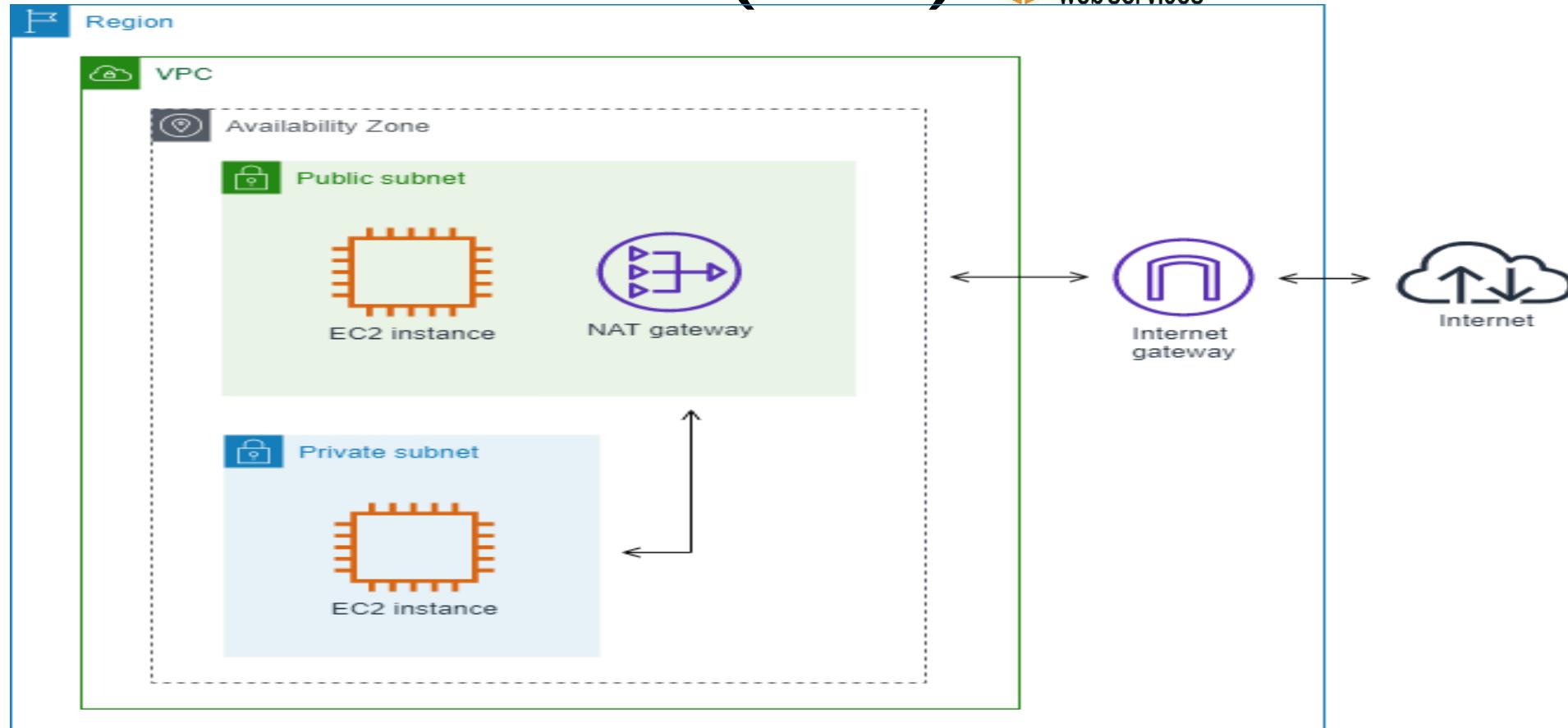
## LES PRINCIPAUX FOURNISSEURS DE CLOUD

- **Amazon Web Services (AWS)** 
- **S3 : Simple Storage Service**



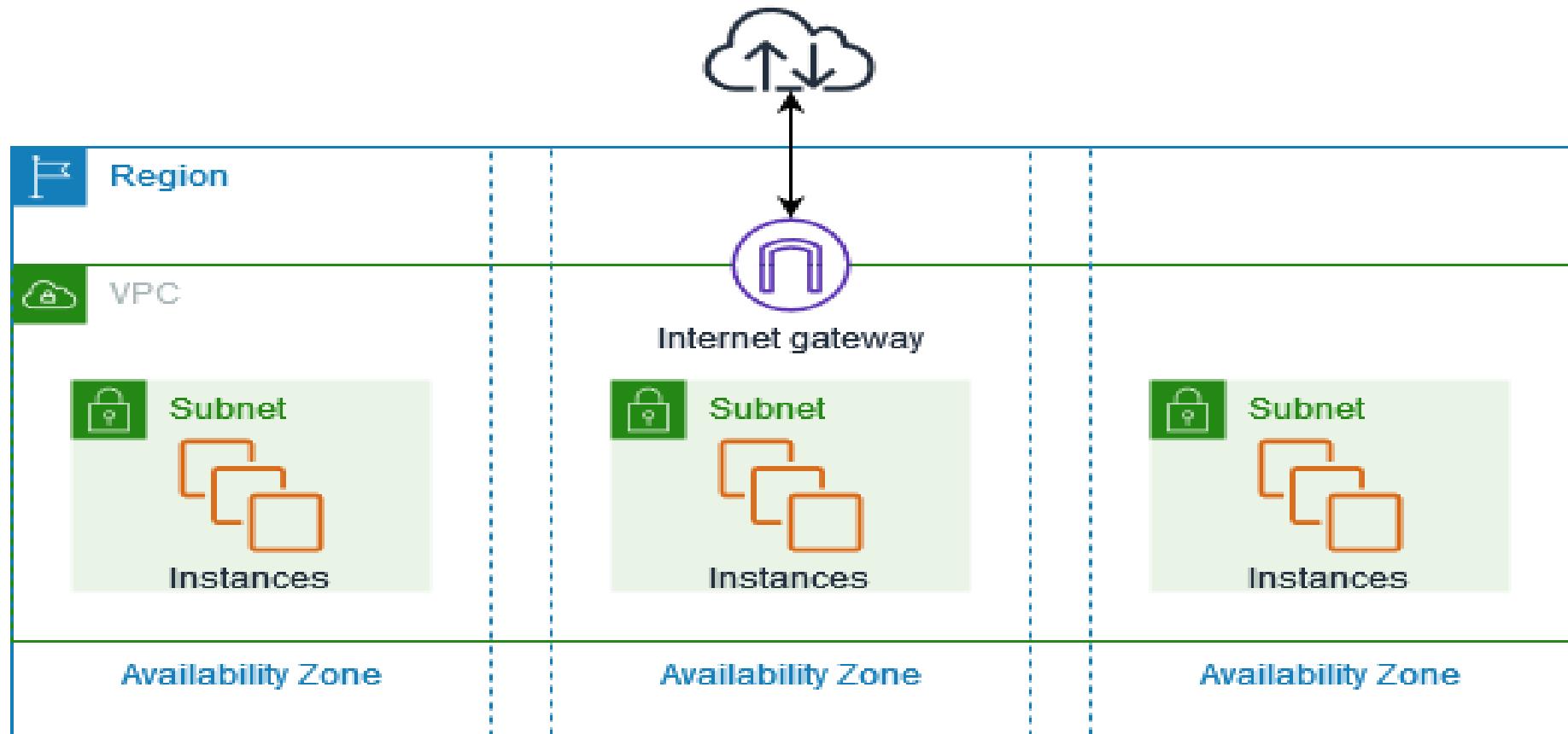
## LES PRINCIPAUX FOURNISSEURS DE CLOUD

### ➤ Amazon Web Services (AWS)



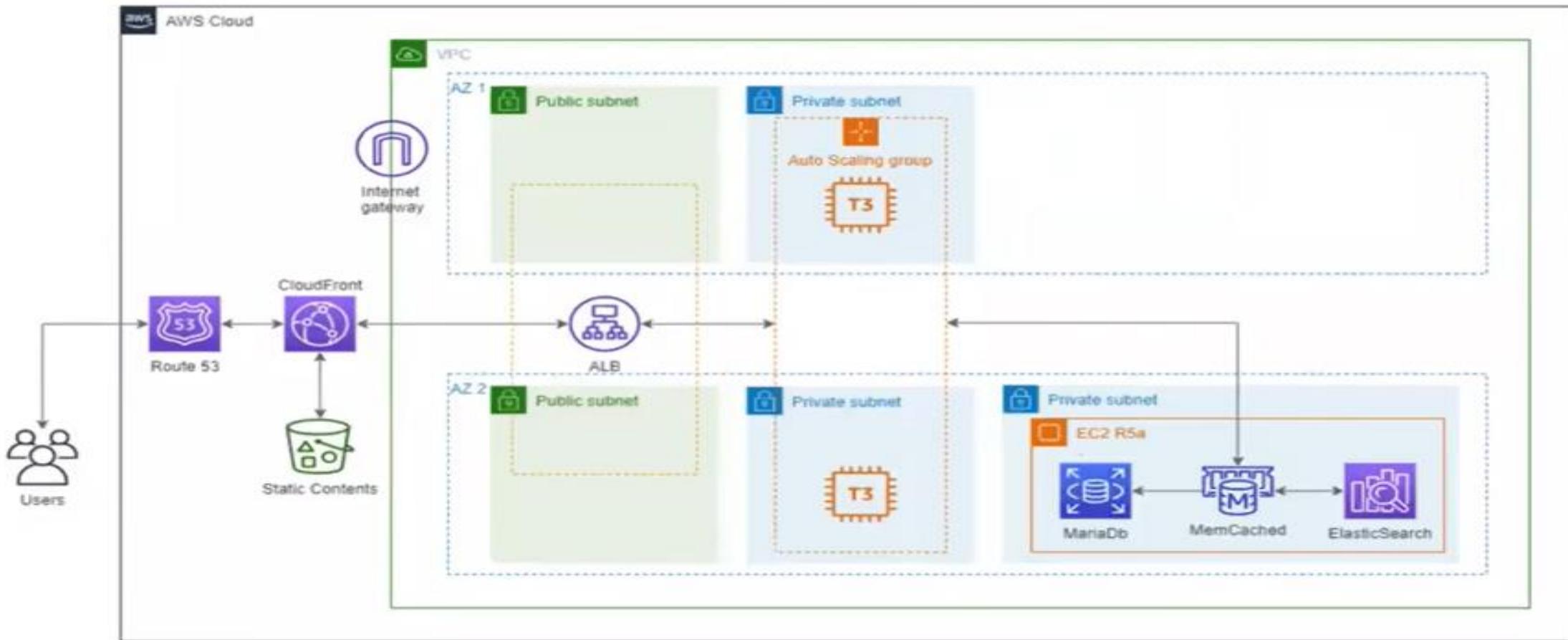
## LES PRINCIPAUX FOURNISSEURS DE CLOUD

### ➤ Amazon Web Services (AWS)



## LES PRINCIPAUX FOURNISSEURS DE CLOUD

### ➤ Amazon Web Services (AWS)

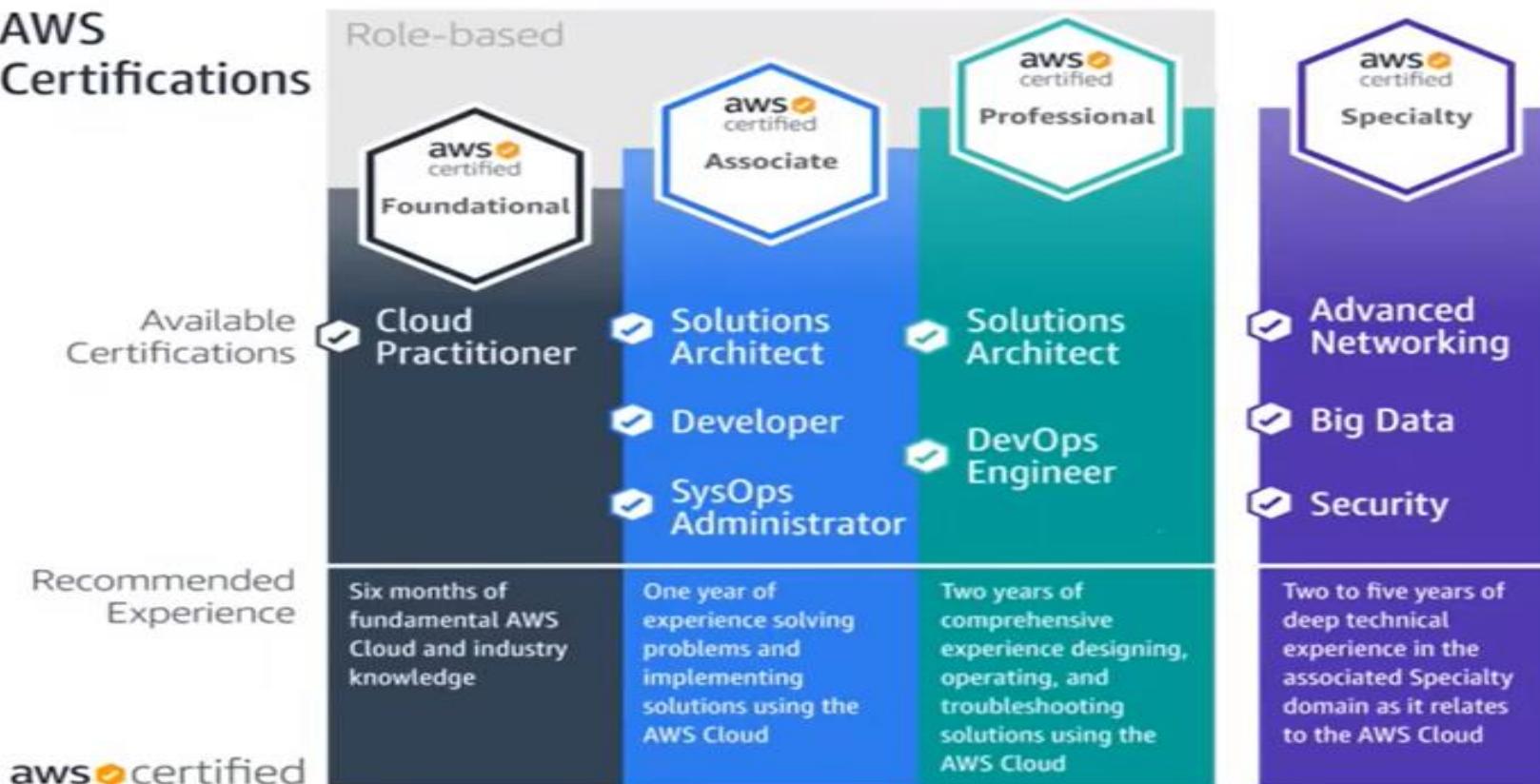


# LES PRINCIPAUX FOURNISSEURS DE CLOUD

# ➤ Amazon Web Services (AWS)



## AWS Certifications



## LES PRINCIPAUX FOURNISSEURS DE CLOUD

### ➤ Microsoft Azure



- A réalisé un chiffre d'affaires de 34 milliards de dollar en 2023, deuxième position derrière AWS depuis 2019.
- Propose un service très diversifié grâce à une plateforme familiarisée.
- Microsoft a créé son Cloud public Azure par-dessus Windows Server et Hyper-V.
- Microsoft a défini 17 régions situées un peu partout aux États-Unis, en Europe, en Asie, en Amérique du Sud et en Australie.

## LES PRINCIPAUX FOURNISSEURS DE CLOUD

### ➤ **Google Cloud Platform**

- Lancé en 2008, Google AppEngine est disponible en Cloud Public
- Sa grande force réside dans le fait qu'il soit justement gratuit, et propose des applications de qualité, malgré le très grand nombre de requête effectuées sur les serveurs.
- Avec un chiffre d'affaires de près 8,4 milliards de dollar en 2023
- Le géant a des datacenters en Europe (Belgique, Royaume-Uni, Allemagne, Pays-Bas et Finlande). Mais pas en France

## CLOUD COMPUTING

### Travail Pratique VI

#### Objectifs :

- Etudier et comparer des principaux fournisseurs de cloud;
- L'utilisation des services cloud;
- Créer et configurer un cloud privé ;

Il s'agit de réaliser une étude comparative entre deux fournisseurs de cloud en se basant sur les critères : - La description – les types de services fournis – la disponibilité - les prix - L'utilisation et la configuration - La sécurité ...

## CLOUD COMPUTING

### Travail Pratique V

- I. Réaliser un module d'authentification sur une configuration locale (PHP, JSP, ... et base de données relationnelle).
- II. Migrer la base de données sur un fournisseur cloud.
- III. Utiliser une service cloud pour créer un serveur web pour le module crée.
- IV. Réaliser le même travail avec une base de données NOSQL.

## SOLUTIONS LIBRES DE CLOUD

- **OpenStack**
- **OpenNebula**
- **CloudStack**
- **Eucalyptus ...**

## SOLUTIONS LIBRES DE CLOUD

- **OpenStack**
  - OpenStack est une plateforme développée par la NASA dédié aux infrastructures massives.
  - Tout le code est distribué sous licence Apache.
  - Elle a un grand potentiel en raison de son architecture et de la communauté et par le soutien de ses partenaires du secteur privé.
  - OpenStack est une solution récente, open source et en cours de développement.

## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : Définitions**

- Wikipédia  
OpenStack est un ensemble de logiciels open source permettant de déployer des infrastructures de cloud Computing (IaaS).
- Openstack.org  
OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.

## SOLUTIONS LIBRES DE CLOUD

### ➤ OpenStack : Historique



- 2010 : Rackspace Hosting + NASA lancent un projet communautaire et Open source
- Objectif : créer et offrir des services de Cloud Computing avec du matériel standard

## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : Summit**

- Aux USA jusqu'en 2013
- Aujourd'hui : alternance USA et Asie/Europe
- Quelques centaines participants au début, à 4500 aujourd'hui
- En parallèle : conférence et Design Summit
- Détermine le nom de la release : lieu/ville à proximité du Summit

# SOLUTIONS LIBRES DE CLOUD

## ➤ OpenStack : Historique - <https://releases.openstack.org/>

| Series                                  | Status  | Initial Release Date            | Next Phase                           | EOL Date                             |
|---|---|---------------------------------|--------------------------------------|--------------------------------------|
| <a href="#">2024.1 Caracal (SLURP)</a>  | <a href="#">Development</a>                           | 2024-04-03 estimated (schedule) | <a href="#">Maintained</a>           | <a href="#">estimated 2023-04-03</a> |
| <a href="#">2023.2 Bobcat</a>           | <a href="#">Maintained</a>                            | 2023-10-04                      | <a href="#">Extended Maintenance</a> | <a href="#">estimated 2025-04-04</a> |
| <a href="#">2023.1 Antelope (SLURP)</a> | <a href="#">Maintained</a>                            | 2023-03-22                      | <a href="#">Extended Maintenance</a> | <a href="#">estimated 2024-09-22</a> |
| <a href="#">zed</a>                     | <a href="#">Maintained</a>                            | 2022-10-05                      | <a href="#">Extended Maintenance</a> | <a href="#">estimated 2024-04-05</a> |
| <a href="#">Yoga</a>                    | <a href="#">Maintained</a>                            | 2022-03-30                      | <a href="#">Extended Maintenance</a> | <a href="#">estimated 2023-11-02</a> |
| <a href="#">Xena</a>                    | <a href="#">Extended Maintenance</a> (see note below) | 2021-10-06                      | <a href="#">Unmaintained</a>         | TBD                                  |
| <a href="#">Wallaby</a>                 | <a href="#">Extended Maintenance</a> (see note below) | 2021-04-14                      | <a href="#">Unmaintained</a>         | TBD                                  |
| <a href="#">Victoria</a>                | <a href="#">Extended Maintenance</a> (see note below) | 2020-10-14                      | <a href="#">Unmaintained</a>         | TBD                                  |
| <a href="#">Ussuri</a>                  | <a href="#">Extended Maintenance</a> (see note below) | 2020-05-13                      | <a href="#">Unmaintained</a>         | TBD                                  |
| <a href="#">Train</a>                   | <a href="#">Extended Maintenance</a> (see note below) | 2019-10-16                      | <a href="#">Unmaintained</a>         | TBD                                  |

## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : Gouvernance**

- Entité indépendante de gouvernance principale du projet
- Les membres du « board of directors » sont issus des entreprises sponsors et élus par les membres individuels
- Tout le monde peut devenir membre individuel (gratuitement)
- Plus de 850 organisations à travers le monde
- Plus de 19000 membres individuels dans 187 pays
- Supporte le projet par différents moyens :
  - Evénements : organisation (Summits) ou participation (OSCON, etc.)
  - Infrastructure de développement (serveurs)
  - Ressources humaines : marketing, release manager, quelques développeurs (principalement sur l'infrastructure)

# SOLUTIONS LIBRES DE CLOUD

## ➤ OpenStack : Communauté

### OpenInfra Member Benefits

|  | Silver   | Gold      | Platinum  |
|--|----------|-----------|-----------|
| Your company logo displayed on openinfra.dev   | ✓        | ✓         | ✓         |
| Ability to feature products and services in the <b>OpenStack Marketplace</b>                         | ✓        | ✓         | ✓         |
| Access to <b>commercial use trademark programs</b>   | ✓        | ✓         | ✓         |
| Opportunity to showcase <b>customer case studies</b> and contribute to <b>Superuser Magazine</b>     | ✓        | ✓         | ✓         |
| <b>Support for member announcements</b> through press release quotes & social media                  | ✓        | ✓         | ✓         |
| Premier placement on the <b>OpenInfra Jobs Board</b>   | ✓        | ✓         | ✓         |
| Access to <b>strategic marketing support and guidance</b>  | ✓        | ✓         | ✓         |
| <b>Early opportunities to purchase sponsorship packages</b> including for the OpenInfra Summit       | ✓        | ✓         | ✓         |
| Access to <b>OpenInfra Foundation leadership</b> including open source strategy and guidance         | ✓        | ✓         | ✓         |
| Opportunity to attend <b>OpenInfra Foundation press briefings and PR events</b>                      | ✓        | ✓         | ✓         |
| Participate in <b>Platinum &amp; Gold specific events and discussion forums</b>                      | ✓        | ✓         | ✓         |
| Access to <b>support for open source projects</b> your company cares about                           | ✓        | ✓         | ✓         |
| Opportunity to participate in designated <b>marketing emails</b> reaching audience of 65k+ people    | ✓        | ✓         | ✓         |
| Foundation-issued <b>press release announcing membership</b>   | ✓        | ✓         | ✓         |
| <b>Potential seat on the OpenInfra Board of Directors</b> if elected through Gold member elections   | ✓        |           |           |
| <b>Appoint one representative</b> to the OpenInfra Foundation Board of Directors                     |          | ✓         |           |
| Access to Membership Spotlight feature on openstack.org reaching millions of readers annually        |          | ✓         |           |
| <b>Guide OpenInfra Foundation strategy, goals, and finances</b> including approval of annual budgets |          | ✓         |           |
| Opportunity to participate in regional and global <b>OpenInfra press briefings</b>                   |          | ✓         |           |
| <b>Most prominent placement</b> in membership displays on websites, events and marketing materials   |          | ✓         |           |
| <b>Annual Membership Fee</b>   | \$15,000 | \$150,000 | \$350,000 |

## SOLUTIONS LIBRES DE CLOUD

### ➤ OpenStack : Communauté - PLATINUM MEMBERS



## SOLUTIONS LIBRES DE CLOUD

### ➤ OpenStack : Communauté - GOLD MEMBERS



Bloomberg  
Engineering



cleura



inspur 浪潮



NEC

H3C

N  
nipa.cloud

卓朗昆仑云

ubuntu®  
Delivered by Canonical

VEXXHOST

ZTE

## SOLUTIONS LIBRES DE CLOUD

### ➤ OpenStack : Communauté - SILVER MEMBERS



## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : Communauté - INFRASTRUCTURE DONORS**



# SOLUTIONS LIBRES DE CLOUD

# ➤ OpenStack : Communauté - SUPPORTING ORGANIZATIONS



## SOLUTIONS LIBRES DE CLOUD

### ➤ OpenStack : Communauté

COMPANIES

**850**

INDIVIDUAL MEMBERS

**+19 k**

TOTAL DEVELOPERS

**2130**

AVERAGE MONTHLY  
CONTRIBUTORS

**374**

COUNTRIES

**187**

TOP 10 COUNTRIES

United States, China, India,  
Great Britain, France, Russia, Australia, Canada,  
Japan, Germany

## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : Packaging**

- **OpenStack Distributions** : Red Hat - SUSE – Ubuntu
- **Packaged Deploys For Different Linux Distros**
  - Mirantis - <https://fuel.mirantis.com/>
  - Piston Cloud Computing - <http://www.pistoncloud.com/openstack-cloud-software/>
  - Rackspace - [http://www.rackspace.com/cloud/private/openstack\\_software/](http://www.rackspace.com/cloud/private/openstack_software/)
- **Configuration Management Tools**
  - Opscode Chef - <https://github.com/opscode/openstack-chef-repo/>
  - Puppet Labs Puppet -<http://puppetlabs.com/solutions/cloud-automation/compute/openstack>

## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : Packaging**

- **Devstack**

<http://docs.openstack.org/developer/devstack/>

“a series of extensible scripts used to quickly bring up a complete OpenStack environment”

Ubuntu, RedHat

- **Packstack**

<https://www.rdoproject.org/install/quickstart/>

“installation utility that lets you spin up a proof of concept cloud on one node”

RedHat

## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : Développement**

- Python
- Ouvert à tous (individuels et entreprises)
- Cycle de développement de 6 mois

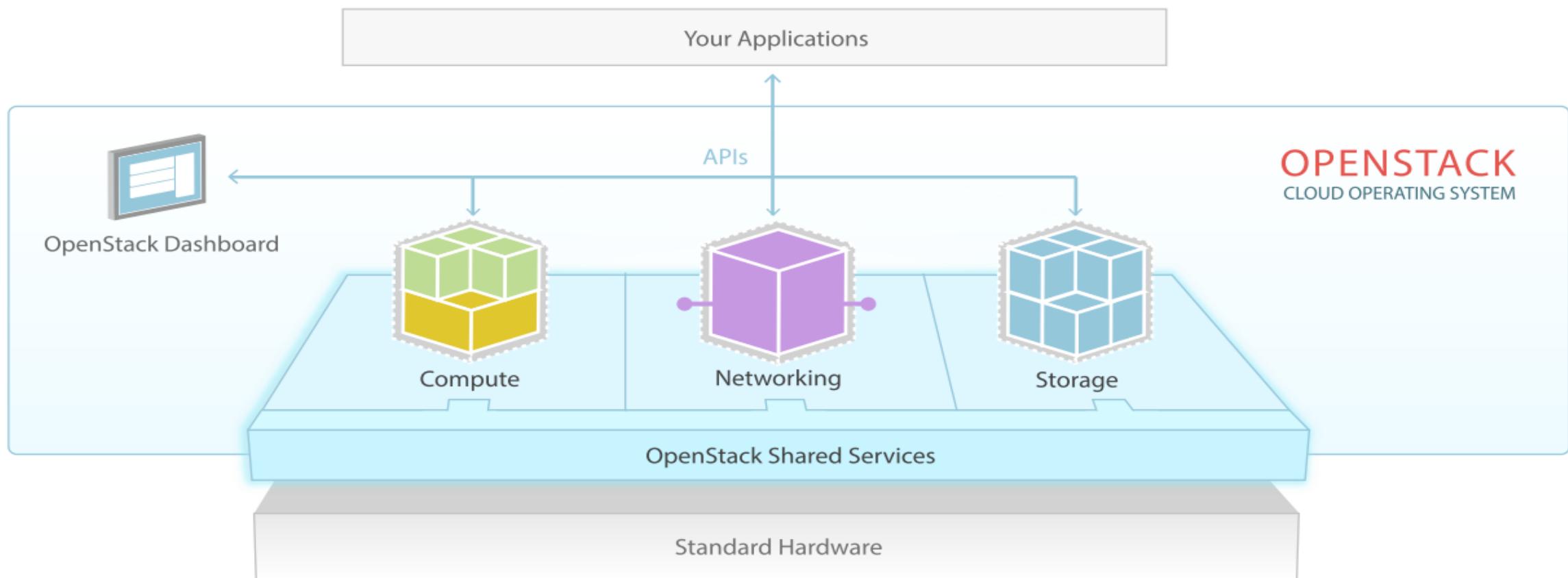
## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : L'architecture**

- Des composants indépendants
- Qui dialoguent entre eux via une file d'attente (RabbitMQ, protocole AMQP)
- Et qui peuvent échanger des informations via les API qu'ils exposent

## SOLUTIONS LIBRES DE CLOUD

### ➤ OpenStack : L'architecture



# SOLUTIONS LIBRES DE CLOUD

## ➤ OpenStack : L'architecture - Aperçu : pour un utilisateur

The screenshot shows the OpenStack Compute (Nova) dashboard under the 'Vue d'ensemble' (Overview) tab. The left sidebar lists various services: Projet, Compute, Vue d'ensemble (selected), Instances, Volumes, Images, Paires de clés, Accès API, Container Infra, Réseau, Orchestration, and Identité. The top right shows a user 'demo1'. The main area displays a summary of quota usage with pie charts:

- Instances:** Utilisé 2 sur 10
- VCPUs:** Utilisé 2 sur 20
- RAM:** Utilisé 512Mo sur 50Go
- Groupes de sécurité:** Utilisé 4 sur 10
- Volumes:** Utilisé 0 sur 10
- IP flottantes:** Utilisé 1 sur 50
- Stockage de volumes:** Utilisé 0Octet sur 1000Go

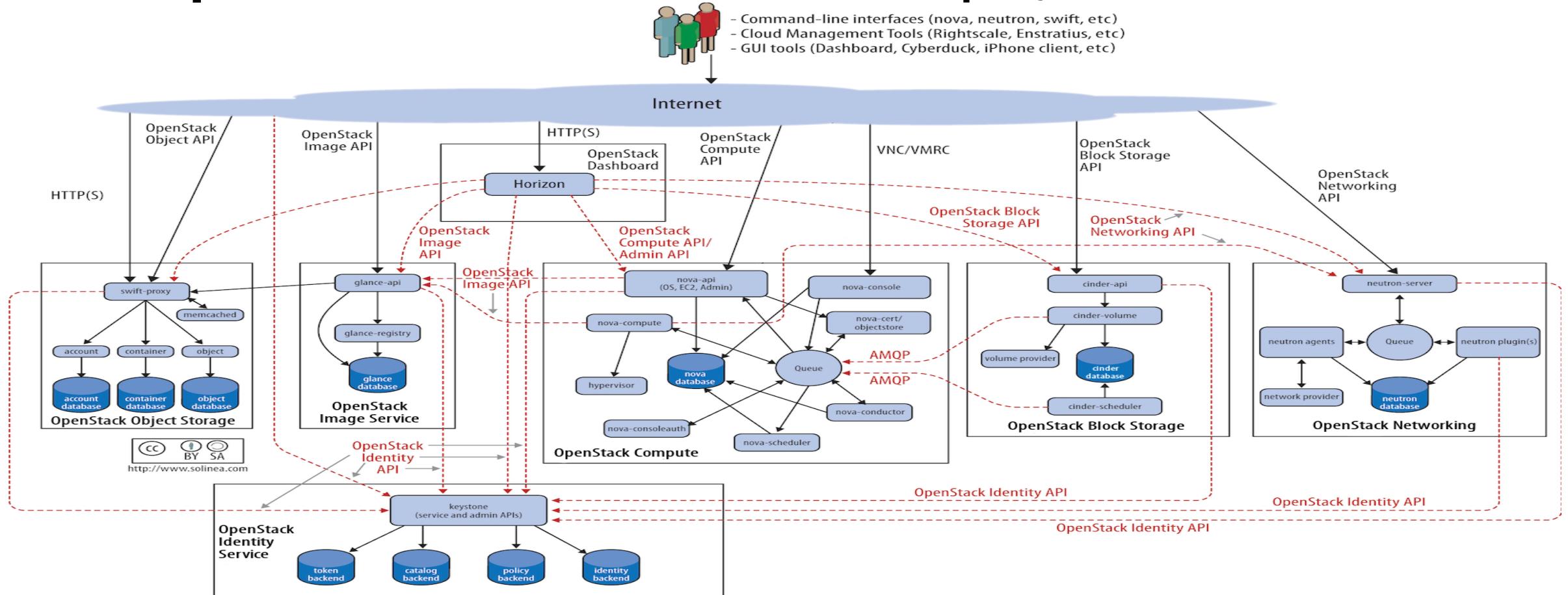
At the bottom, a summary table provides a quick overview of usage across all resources.

| Ressource           | Utilisé | Sur    |
|---------------------|---------|--------|
| Instances           | 2       | 10     |
| VCPUs               | 2       | 20     |
| RAM                 | 512Mo   | 50Go   |
| Groupes de sécurité | 4       | 10     |
| Volumes             | 0       | 10     |
| IP flottantes       | 1       | 50     |
| Stockage de volumes | 0Octet  | 1000Go |

**Résumé de l'Utilisation**

# SOLUTIONS LIBRES DE CLOUD

## ➤ OpenStack : L'architecture - Ce que ça cache ...



## SOLUTIONS LIBRES DE CLOUD

### ➤ OpenStack : L'architecture

- Indispensables :
  - Authentification, Autorisations (Identity Service) : **Keystone**
  - Catalogue de services : **Keystone**
  - Gestion/execution des instances (Compute) : **Nova**
  - Registre d'images (Image Service) : **Glance**
- Stockage :
  - Stockage block (Block Storage) : **Cinder**
  - Stockage objet (Object Storage ) : **Swift**
- Réseau :
  - Réseau (Networking) : **Neutron**

## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : L'architecture - Keystone (identity)**

- Service de renseignements (utilisateurs, groupes, projets...)
- Authentifie les utilisateurs en leur donnant des jetons
- Valide l'authenticité d'un jeton
- Dépositaire du catalogue de service

## SOLUTIONS LIBRES DE CLOUD

- **OpenStack : L'architecture - Nova (compute)**
  - Gère le catalogue des instances
  - Traite les demandes de création d'instance (en relation avec l'hyperviseur)
  - Gère le placement des instances sur les noeuds compute
  - Fournit les métadonnées (systèmes et utilisateurs) aux instances
  - Assure des fonctions réseaux (limitées : création de bridge, DHCP, firewall...), bientôt abandonnées.
  - Fournit un accès aux instances (console, VNC...)

## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : L'architecture - Glance (images)**

- Service « images» de l'infrastructure virtualisée
- Stocke et distribue les images d'instance
- Appelé par Compute (Nova) lors de l'instanciation d'une nouvelle Instance.
- Format des images : raw, vmdk, qcow , iso, ...
- Gère les métadonnées des images (type, OS, architecture... )
- Plusieurs backend de stockage :
  - Système de fichier local ou NFS (NAS/SAN)
  - Swift (OpenStack Object Service)
  - S3 (Object Service Amazon)

## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : L'architecture - Neutron (Network)**

- Virtualisation du réseau
- API Unifiée, pour piloter plusieurs architectures réseau :
  - Linux Bridge,
  - OpenVSwitch
  - SDN (Software-Defined Networking) : OpenContrail, OpenDaylight, Cisco Nexus, VmWare NSX, ...

## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenStack : L'architecture - Horizon (dashboard)**

- Interface web
- Fournit une interface graphique pour gérer les ressources
- Permet de surveiller l'utilisation des ressources

<https://manager.infomaniak.com/v3/747782/ng/public-cloud/4104/project/5338/users>  
<https://api.publ.infomaniak.cloud/horizon/auth/login/?next=/horizon/project/>



## SOLUTIONS LIBRES DE CLOUD

### ➤ **OpenNebula**

- On notera également qu'il s'agit d'un projet européen, développé par une société espagnole en 2008.
- Une solution de Cloud open source sous licence Apache
- Il est écrit en C++, Ruby et Shell
- C'est un projet qui fournit un ensemble de fonctionnalités organisant le fonctionnement d'un ensemble de serveurs physiques, fournissant des ressources à des machines dites « virtuelles ».

## SOLUTIONS LIBRES DE CLOUD

### ➤ **Eucalyptus**

- Eucalyptus est la plateforme utilisée par Amazon EC2.
- permet l'installation d'une infrastructure de Cloud privé et hybride
- Il est écrit en langage Java, C et Python

## CLOUD COMPUTING

### Travail Pratique IV

Objectifs :

- L'utilisation des services de Google cloud;
- I. Ouvrir le lien : <https://google.qwiklabs.com/>
  - II. Connecter
  - III. Taper ***BigQuery*** dans la zone de recherche
  - IV. Choisir [Using BigQuery in the GCP Console](#)
  - V. Réaliser le **Qwiklabs**

## CLOUD COMPUTING

### Travail Pratique V

Il s'agit de créer un cloud privé en utilisant **OpenStack**.

## STOCKAGE DANS LE CLOUD

- **Stockage d'objets**
- **Stockage de fichiers**
- **Stockage par blocs**

# STOCKAGE DANS LE CLOUD

- **Stockage d'objets**
  - Amazon Simple Storage Service (S3)
    - de stocker et protéger n'importe quelle quantité de données pour un large éventail de cas d'utilisation comme :
      - des site web
      - des applications mobiles,
      - la sauvegarde , la restauration, et d'archivage
      - des applications d'entreprise,
      - des appareils IoT,
    - importer des magasins de données existants à des fins d'analyse,
      - des analyses du Big Data...

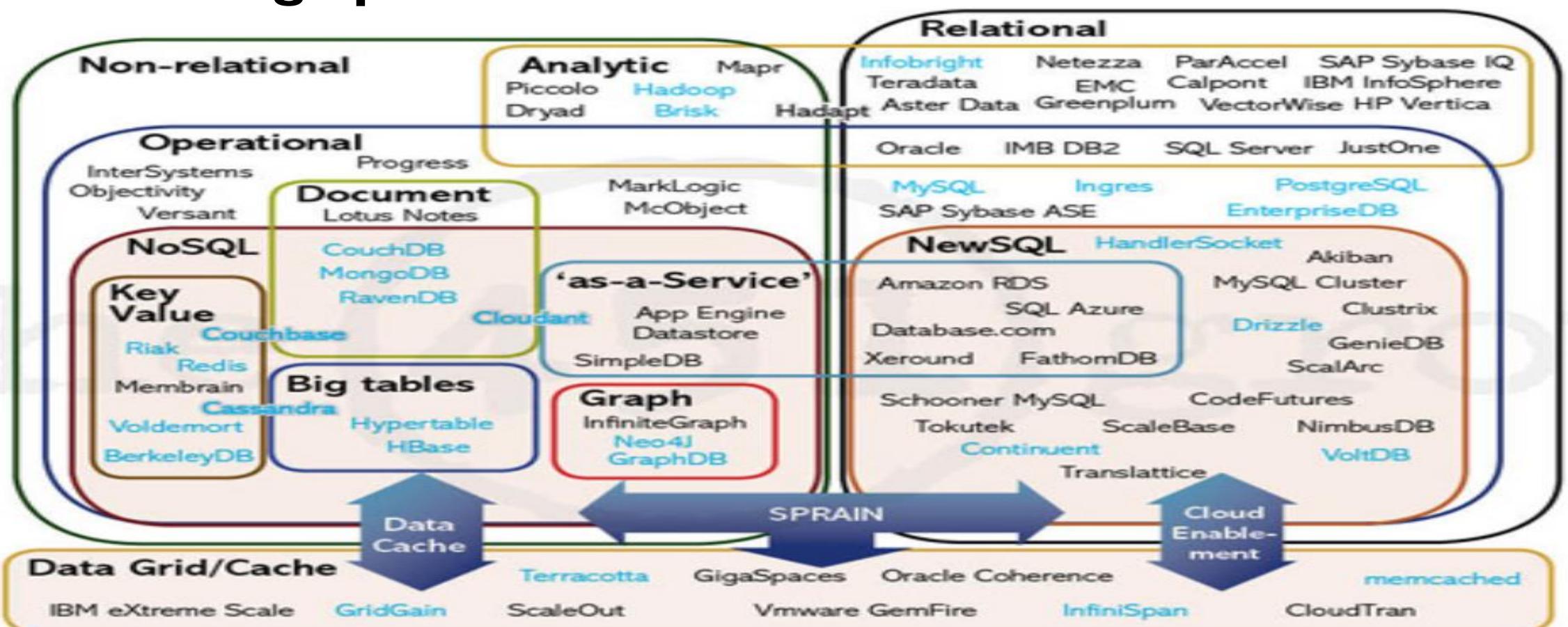
# STOCKAGE DANS LE CLOUD

## ➤ Stockage de fichiers

| Fournisseur         | Stockage<br>Gratuit | Plans Premium   | Limite Taille<br>du Fichier | Limite<br>Stockage | Restauratio<br>n Fichiers | Applis                              |
|---------------------|---------------------|---|-----------------------------|--------------------|---------------------------|-------------------------------------|
| <u>pCloud</u>       | 10 GB               | 2 TB à 9.99 € par mois<br>95 € par an<br>250 € pour toujours(!) | –                           | 10 TB              | ✓                         | Windows, Mac,iOS,<br>Android, Linux |
| <u>Google Drive</u> | 15 GB               | 1 TB à \$9.99/mois  | 5 TB                        | 2 TB               | ✓                         | Windows,Mac,iOS,<br>Android         |
| <u>One Drive</u>    | 5 GB                | 50 GB à €1/mois   | 10 GB                       | 1 TB               | ✓                         | Windows, iOS,Android                |
| <u>Mega</u>         | 50 GB               | 200GB à €4.99/mois<br>500 GB à €99/an                           | –                           | 8 TB               | –                         | Windows, iOS                        |
| <u>Box</u>          | 10GB                | 100 GB à €8/mois  | 2 GB                        | 100 GB             | –                         | Windows                             |
| <u>Media Fire</u>   | 10GB                | 1 TB à \$5/mois<br>\$45/an                                      | 20 GB                       | 1 TB               | –                         | Windows, iOS                        |
| <u>HubiC</u>        | 25 GB               | 100 GB à €1/mois<br>€10/an                                      | –                           | 10TB               | ✓                         | Windows, iOS,Android                |

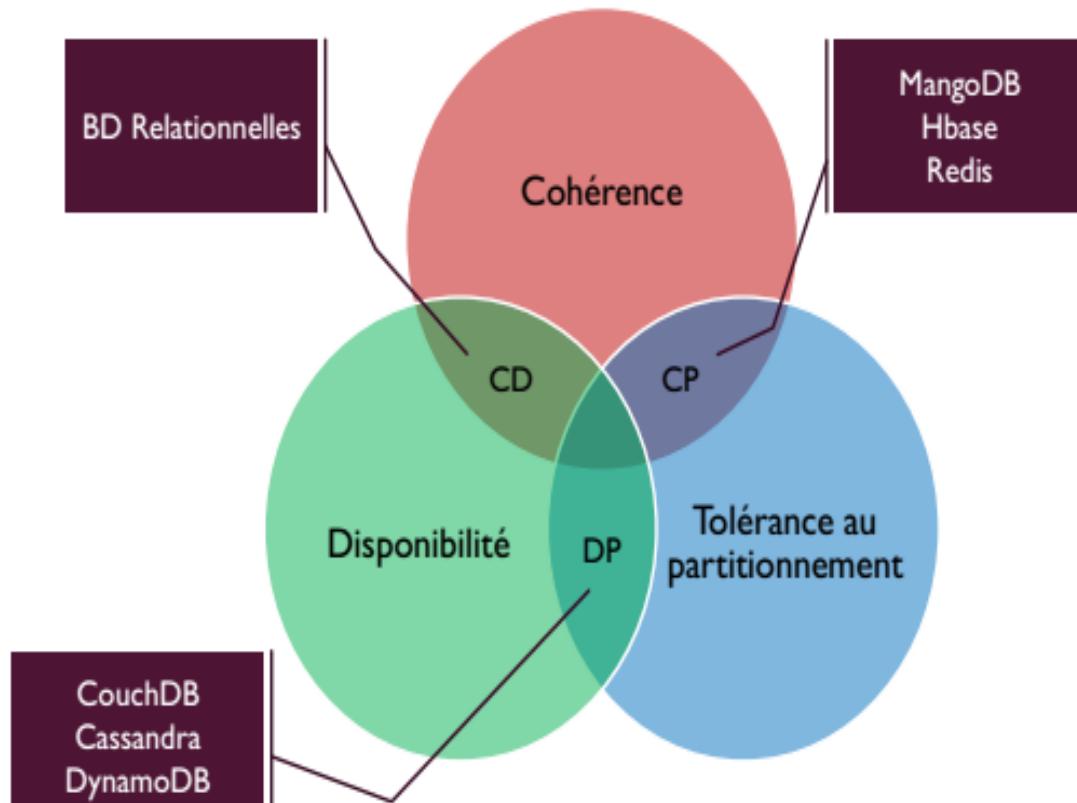
# STOCKAGE DANS LE CLOUD

## ➤ Stockage par blocs



## STOCKAGE DANS LE CLOUD

### ➤ Bases de données non-relationnel



## STOCKAGE DANS LE CLOUD

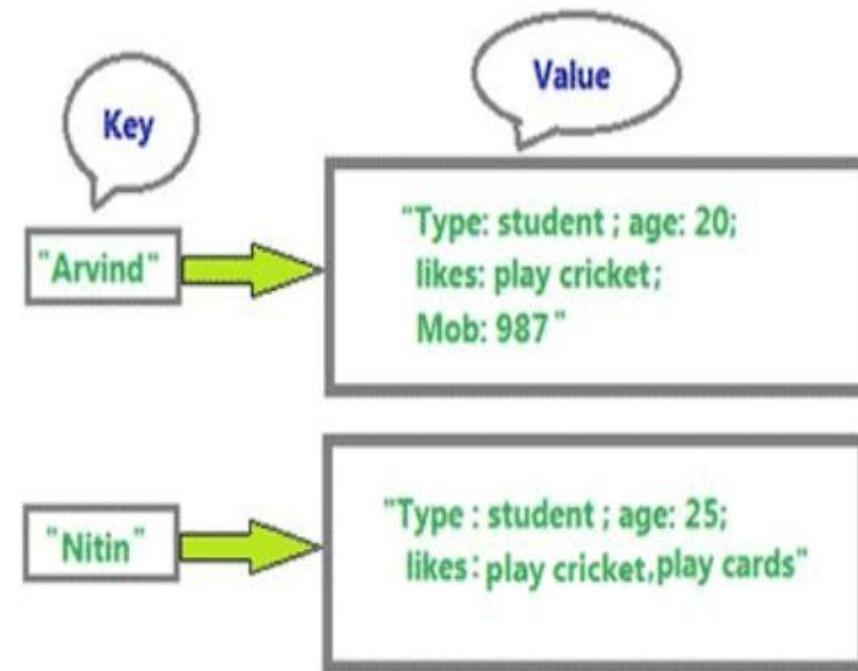
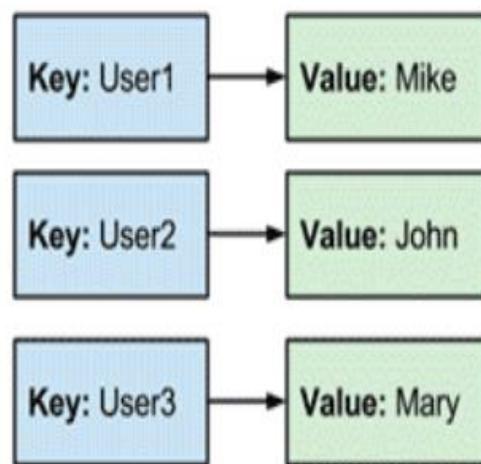
### ➤ **Bases de données non-relationnel**

- **Clé–valeur**
- **Orientée colonnes**
- **Orientée documents**
- **Orientée graphe**

## STOCKAGE DANS LE CLOUD

### ➤ Bases de données non-relationnel

#### ➤ Clé-valeur



## STOCKAGE DANS LE CLOUD

### ➤ Bases de données non-relationnel

#### ➤ Orientée colonnes

| Clé | Date début validité | Date fin Validité | Nom    | Prénom    | Csp      |   |  |  |
|-----|---------------------|-------------------|--------|-----------|----------|---|--|--|
| A   | 01/01/2010          | 02/02/2015        | Dupont | Null      | Cadre    | A | Nom : Dupont DATE: 01/01/2010<br>Nom: Cap DATE: 03/02/2015 | Csp: Cadre DATE: 01/01/2010                                    |
| A   | 03/02/2015          | Null              | Cap    | Null      | Cadre    | B | Prénom : Joséphine DATE: 01/01/2010                        |  |
| B   | 01/01/2010          | Null              | Null   | Joséphine | Null     |   |  |  |
| C   | 01/01/2010          | 03/03/2016        | Black  | George    | Cadre    | C | Nom : Black DATE: 01/01/2010                               | Prénom : George DATE: 01/01/2010                               |
| C   | 04/04/2016          | Null              | Black  | George    | Retraité |   |  | Csp : Cadre DATE: 01/01/2010<br>Csp: Retraité DATE: 04/04/2016 |

# STOCKAGE DANS LE CLOUD

## ➤ Bases de données non-relationnel

### ➤ Orientée documents

| CUSTOMER |       |
|----------|-------|
| ID       | NAME  |
| 1        | Guido |

| PRODUCT |              |
|---------|--------------|
| ID      | NAME         |
| 1000    | IPod Touch   |
| 1020    | Monster Beat |

| BILLING_ADDRESS |             |            |
|-----------------|-------------|------------|
| ID              | CUSTOMER_ID | ADDRESS_ID |
| 1               | 1           | 55         |

| ADDRESS |             |         |           |
|---------|-------------|---------|-----------|
| ID      | STREET      | CITY    | POST_CODE |
| 55      | Chaumontweg | Spiegel | 3095      |

| ORDER |             |                     |
|-------|-------------|---------------------|
| ID    | CUSTOMER_ID | SHIPPING_ADDRESS_ID |
| 90    | 1           | 55                  |

| ORDER_ITEM |          |            |        |
|------------|----------|------------|--------|
| ID         | ORDER_ID | PRODUCT_ID | PRICE  |
| 1          | 90       | 1000       | 250.55 |
| 1          | 90       | 1020       | 199.55 |

```
{
  "id":1,
  "name":"Guido",
  "billingAddress":[{"street":"Chaumontweg","city":"Spiegel","postCode":"3095"}]
}

{
  "id":90,
  "customerId":1,
  "orderItems":[
    {
      "productId":1000,"price": 250.55, „productName”: „iPod Touch”
    },
    {
      "productId":1020,"price": 199.55, „productName”: „Monster Beat”
    }],
  "shippingAddress":[{"street":"Chaumontweg","city":"Spiegel","postCode":"3095"}]
}
```

## STOCKAGE DANS LE CLOUD

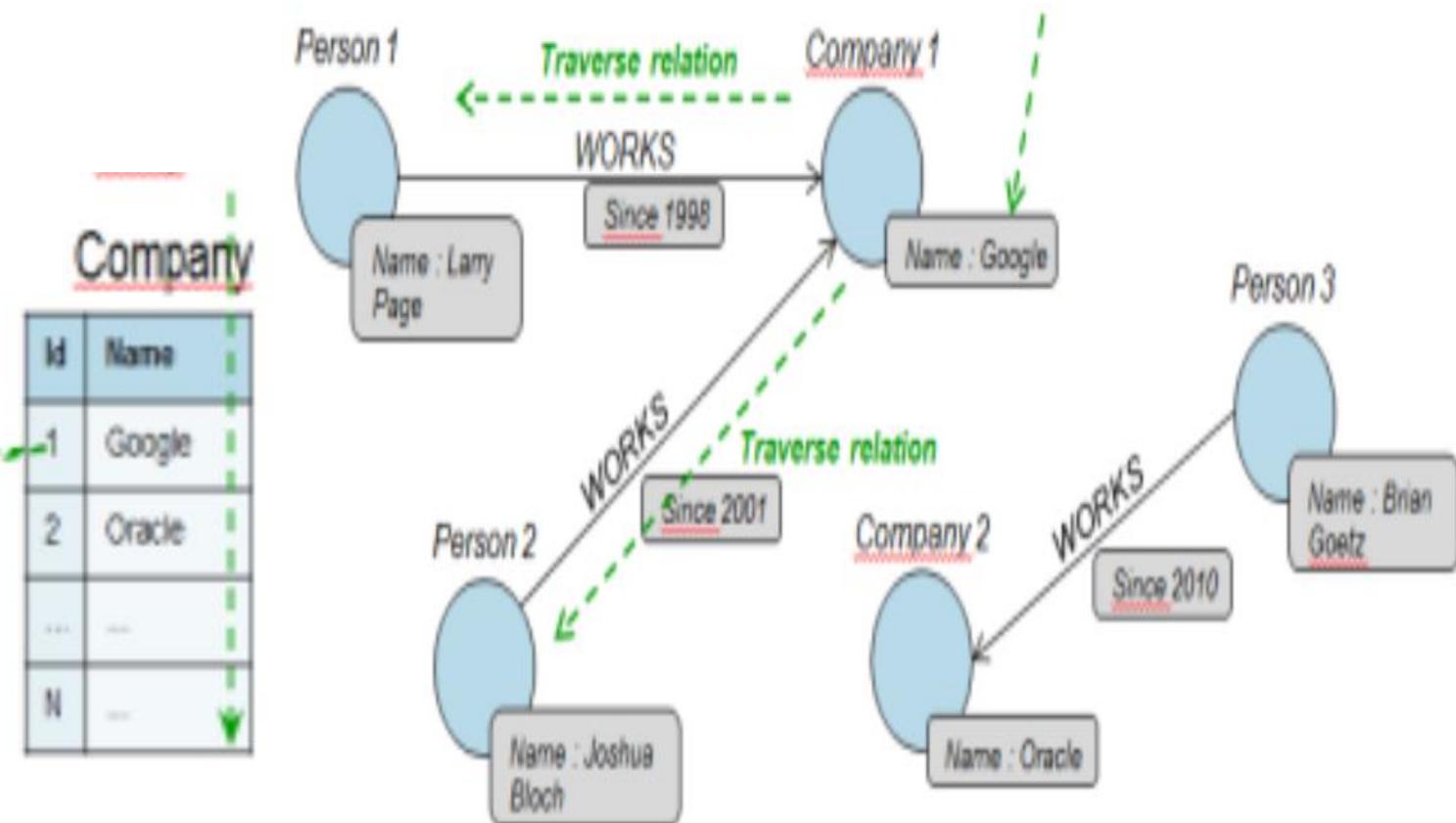
### ➤ Bases de données non-relationnel

#### ➤ Orientée graphe

| Person |              |
|--------|--------------|
| ID     | Name         |
| 1      | Larry Page   |
| 2      | Joshua Bloch |
| 3      | Brian Goetz  |
| N      | ...          |

| Works In |           |       |
|----------|-----------|-------|
| PersonId | CompanyId | Since |
| 1        | 1         | 1998  |
| 2        | 1         | 2001  |
| 3        | 2         | 2010  |



## STOCKAGE DANS LE CLOUD

### ➤ Relationnels Vs non-relationnels

| Fonctionnalité                    | Relationnels (SQL)<br>(ex : MySQL)  | non-relationnels (NoSQL)<br>(ex : Datastore)  | Avantage           |
|-----------------------------------|---|---|--------------------|
| <b>Scalabilité multi-serveurs</b> | Limité à un serveur principal pour les écritures.<br>Pour avoir plus de puissance en écriture il faut un plus gros serveur.                           | Multi-serveurs par défaut, pas de limitations. La charge est aisément répartie sur plusieurs serveurs.  | NoSQL              |
| <b>Structuration des données</b>  | Les données sont toujours très structurées dans des tables. Pratique quand on a besoin d'une structure précise.                                       | Adapté pour les données en structure arborescente (ex : XML, JSON...) dont la forme peut varier.  | Dépend des besoins |
| <b>Puissance du langage SQL</b>   | Permet d'effectuer des requêtes complexes et précises (parfois assez lentes).   | Permet d'effectuer seulement des requêtes simples, mais rapides.  | SQL                |
| <b>Schéma des données</b>         | Le schéma est contraignant : vous devez absolument faire "rentrer" les données dans les cases prévues. Avantage : tout est toujours proprement rangé. | Le schéma n'est pas contraignant : vous pouvez faire ce que vous voulez, stocker comme vous le voulez. Défaut : il faut vérifier manuellement si tout est bien rangé. | Dépend des besoins |
| <b>Transactions</b>               | Les transactions sont strictes : les données sont immédiatement enregistrées et à jour.   | Les transactions sont possibles mais cela demande plus de travail. Les données peuvent être enregistrées avec un léger retard.  | SQL                |

## BIG DATA



BIG DATA

HOW TO



Storage



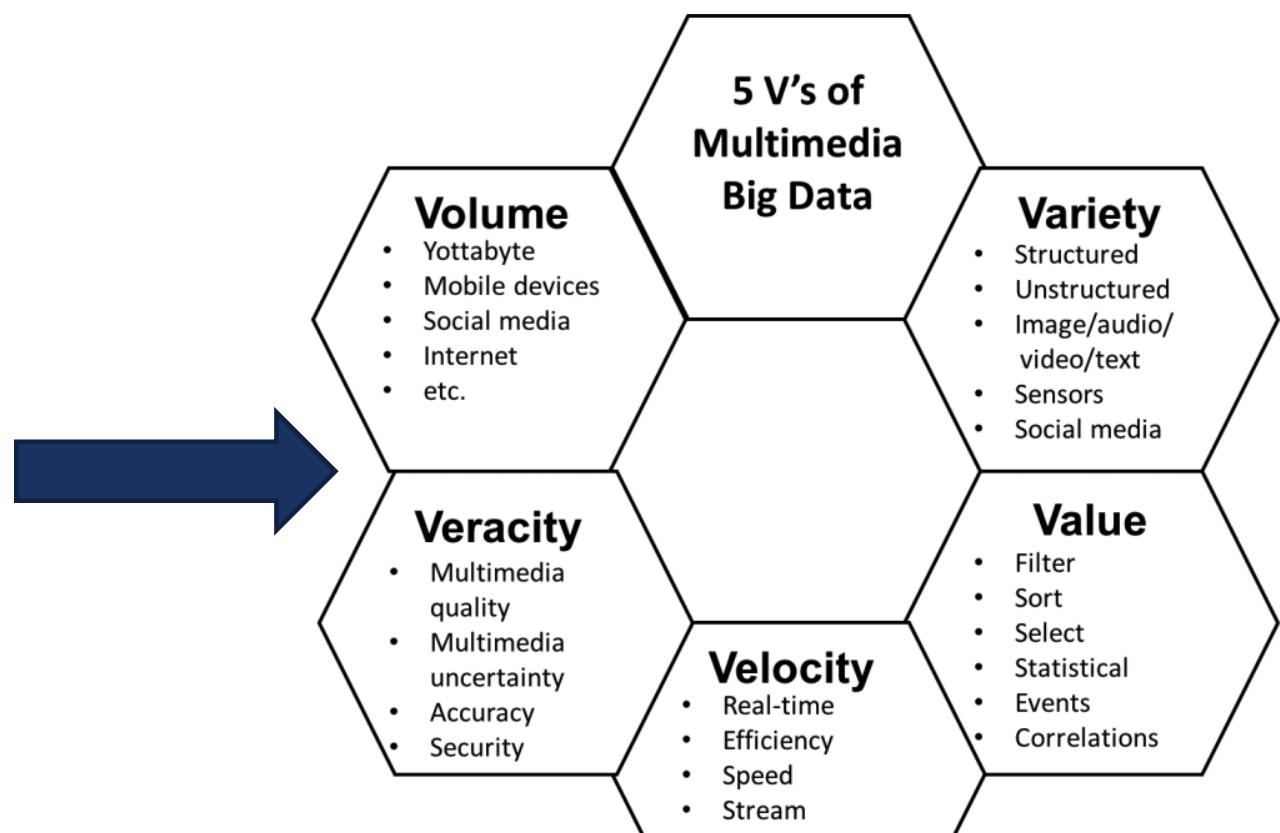
Process



Analysis

## BIG DATA

# THE SOLUTION: BIG DATA!



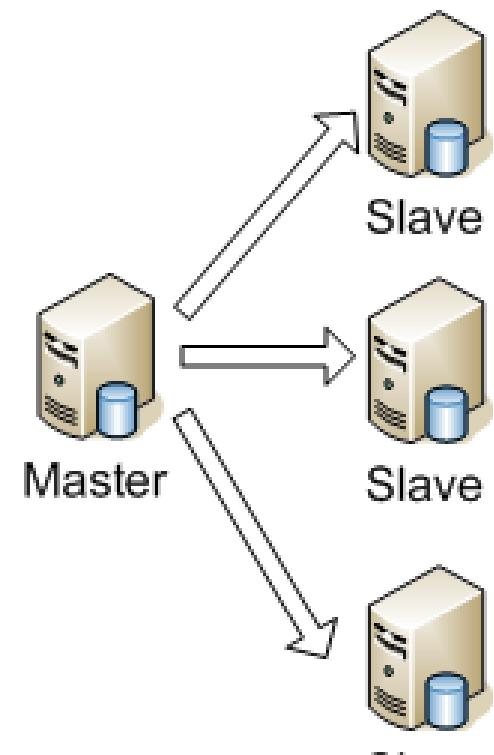
## BIG DATA

### HOW TO SOLVE THIS!!!

- **CLUSTER OF COMPUTERS**
- **DISTRIBUTED FILE SYSTEM**
- **MAPREDUCE PARADIGM**
- ...
- **NOSQL**
- **GPU**
- ...

BIG DATA

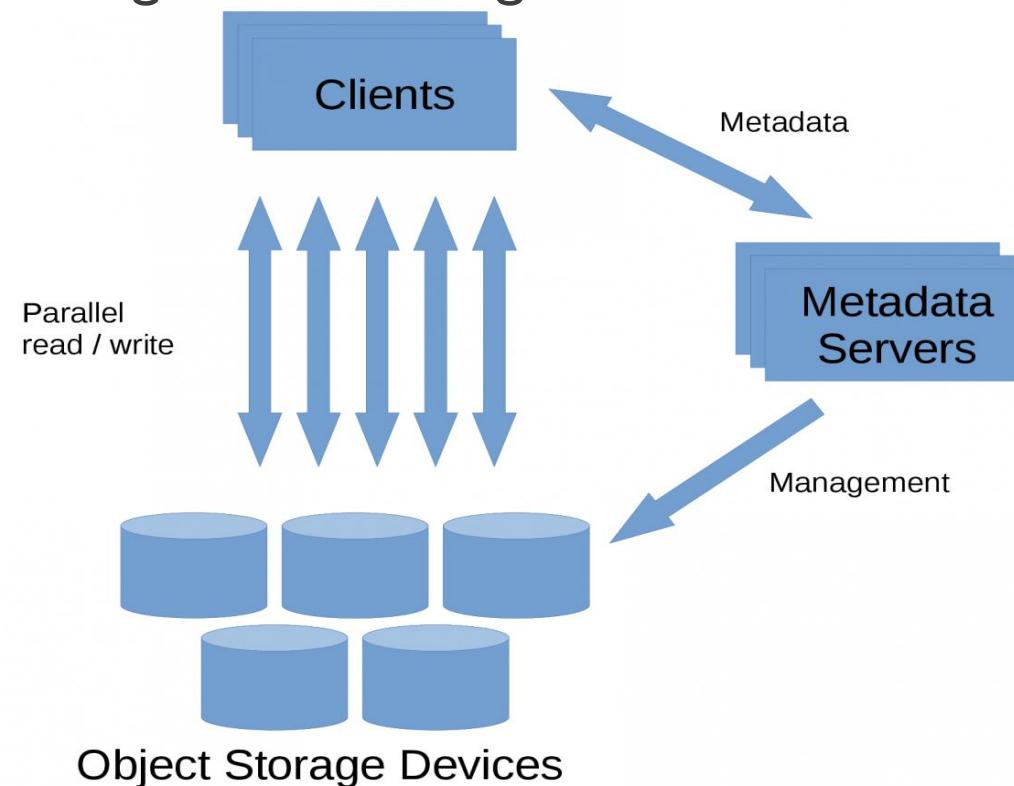
## CLUSTER OF COMPUTERS



# BIG DATA

## DISTRIBUTED FILE SYSTEM

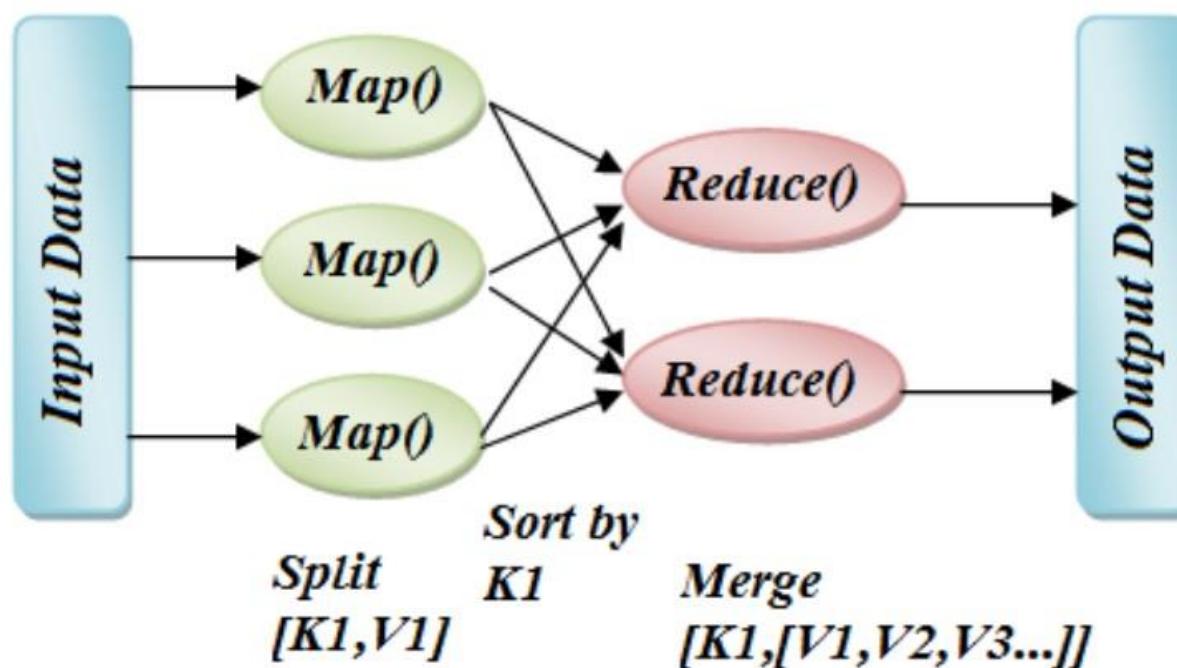
Is a method of storing and accessing files based in a client/server architecture.



# BIG DATA

## MAPREDUCE PARADIGM

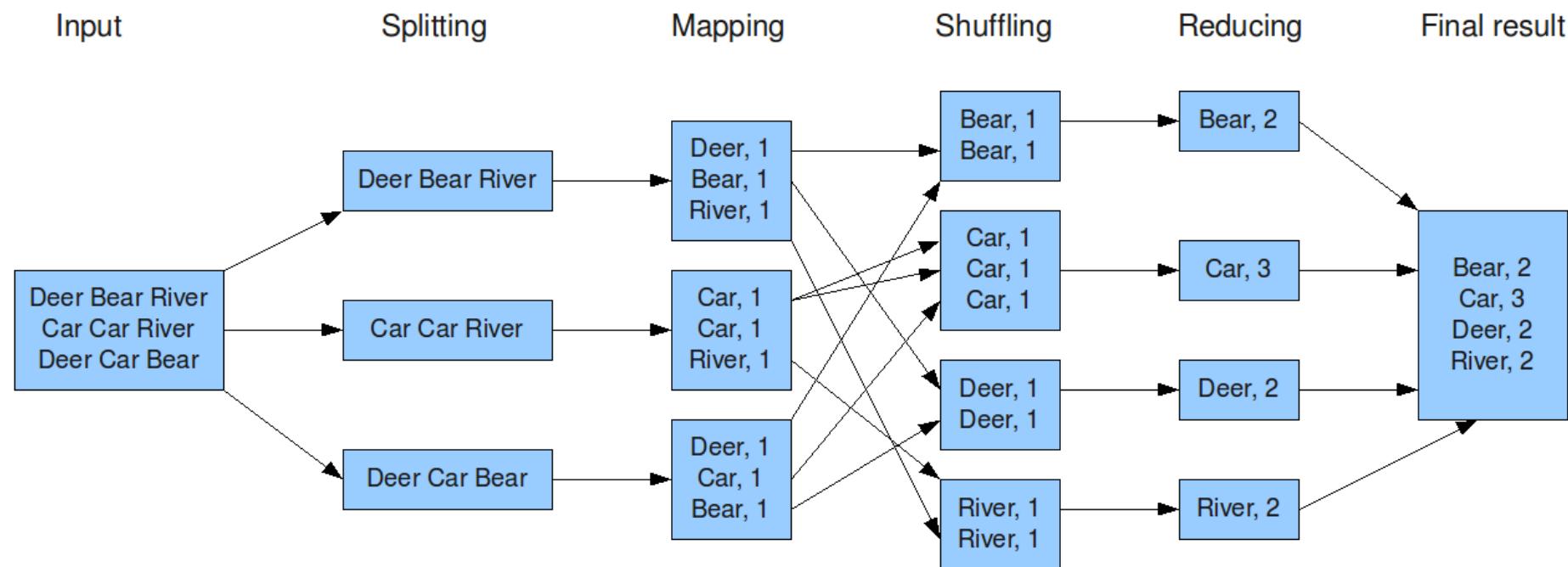
- **MapReduce** is a programming model in 1985
- processing and generating big data sets with a parallel, distributed algorithm on a cluster.



# BIG DATA

## MAPREDUCE -EXAMPLE- WORDCOUNT

The overall MapReduce word count process



# BIG DATA

## EXAMPLE2- COMMON FRIEND IN FACEBOOK

### Input:

A -> B,C,D  
B-> A,C,D  
C-> A,B  
D->A,B

### Map:

(A,B) -> B,C,D  
(A,C) -> B,C,D  
(A,D) -> B,C,D  
**(A,B) -> A,C,D**  
(B,C) -> A,C,D  
(B,D) -> A,C,D  
(A,C) -> A,B  
(B,C) -> A,B  
(A,D) -> A,B  
(B,D) -> A,B

### Reduce:

(A,B) -> C,D  
(A,C) -> B  
(A,D) -> B  
(B,C) -> A  
(B,D) -> A

*Suggest  
Friends ☺*



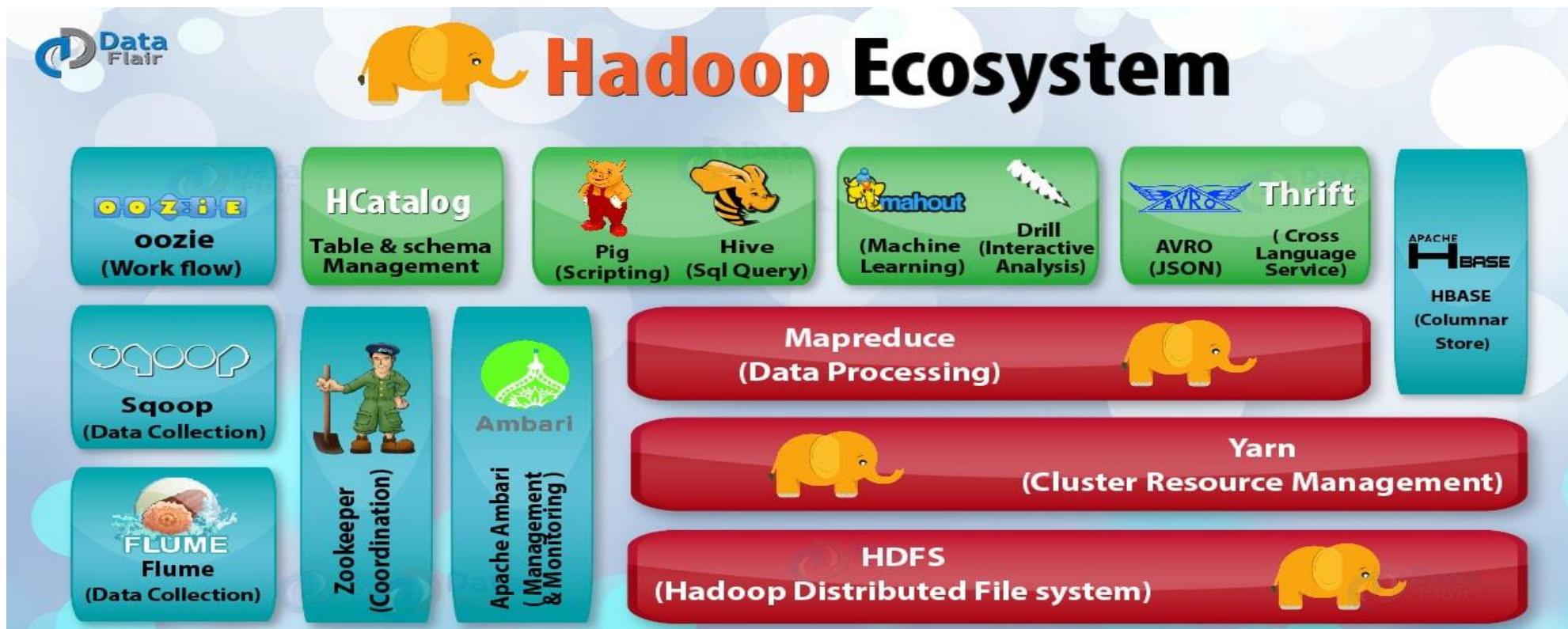
## MAIN TOOL :APACHE HADOOP

- Collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation.
- Developed with Java
- It Include:
  - **Hadoop Common:** The common utilities that support the other Hadoop modules.
  - **Hadoop Distributed File System (HDFS™):** A distributed file system that provides high-throughput access to application data.
  - **Hadoop YARN:** A framework for job scheduling and cluster resource management.
  - **Hadoop MapReduce:** A YARN-based system for parallel processing of large data sets.
  - **Hadoop Ozone:** An object store for Hadoop.
  - **Hadoop Submarine:** A machine learning engine for Hadoop.

# BIG DATA

## HADOOP ECOSYSTEM

- <https://data-flair.training/blogs/hadoop-ecosystem-components/>



## DÉVELOPPEMENT DU CLOUD : LA SIMULATION

### Pourquoi la Simulation?

- La gestion des ressources
- L'ordonnancement des applications
  - Compliquées
  - Il n'existe pas une solution optimale
  - Dans le Cloud, il est difficile d'effectuer les différents scénarios avec différent nombres de ressources et utilisateurs afin d'évaluer la performance
  - Lorsque on veut évaluer les scénarios de manière répétable et contrôlable, cela est parfois impossible car l'issue du coût et la gestion.

## DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

### Outils de simulation

- GridSim
- CloudSim
- Simgrid

## DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

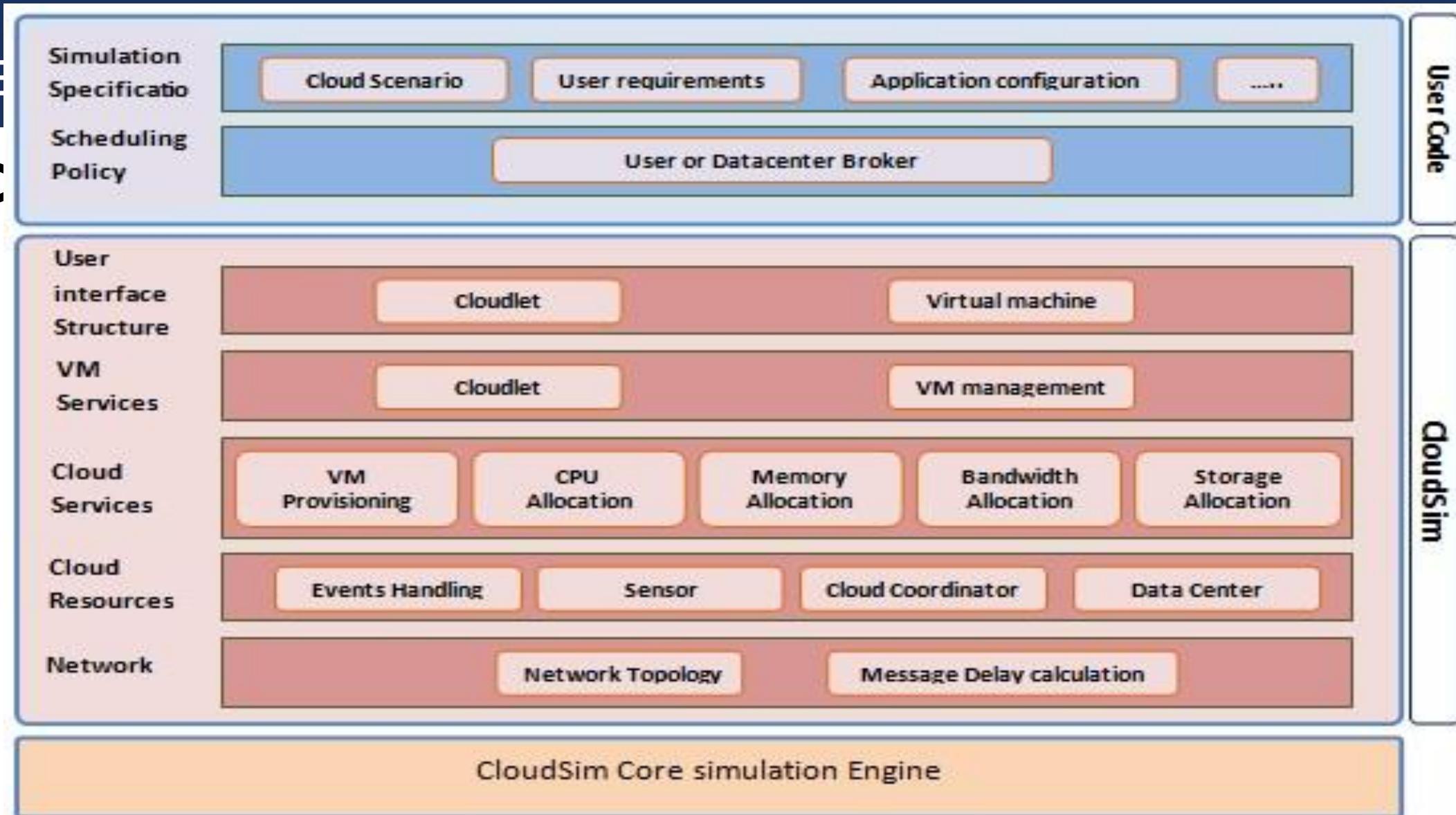
### Cloudsim : Essentials

- Framework for modeling and simulation of cloud
  - Latest version- CloudSim 4.0
- Use Integrated development [IDE]-
  - Eclipse Classic[version 3.5.1 and later]
  - Netbeans[Version 6.9.1 and later]
  - Requires JDK 1.6 or later
- Add External Jar commons-math3-3.x.jar

## DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

### Cloudsim : Structure du répertoire

- cloudsim/
  - top level CloudSim directory
- docs/
  - CloudSim Documentation
- examples/
  - CloudSim examples
- jars/
  - CloudSim jar archives
- sources/
  - CloudSim source code

DÉVE  
C

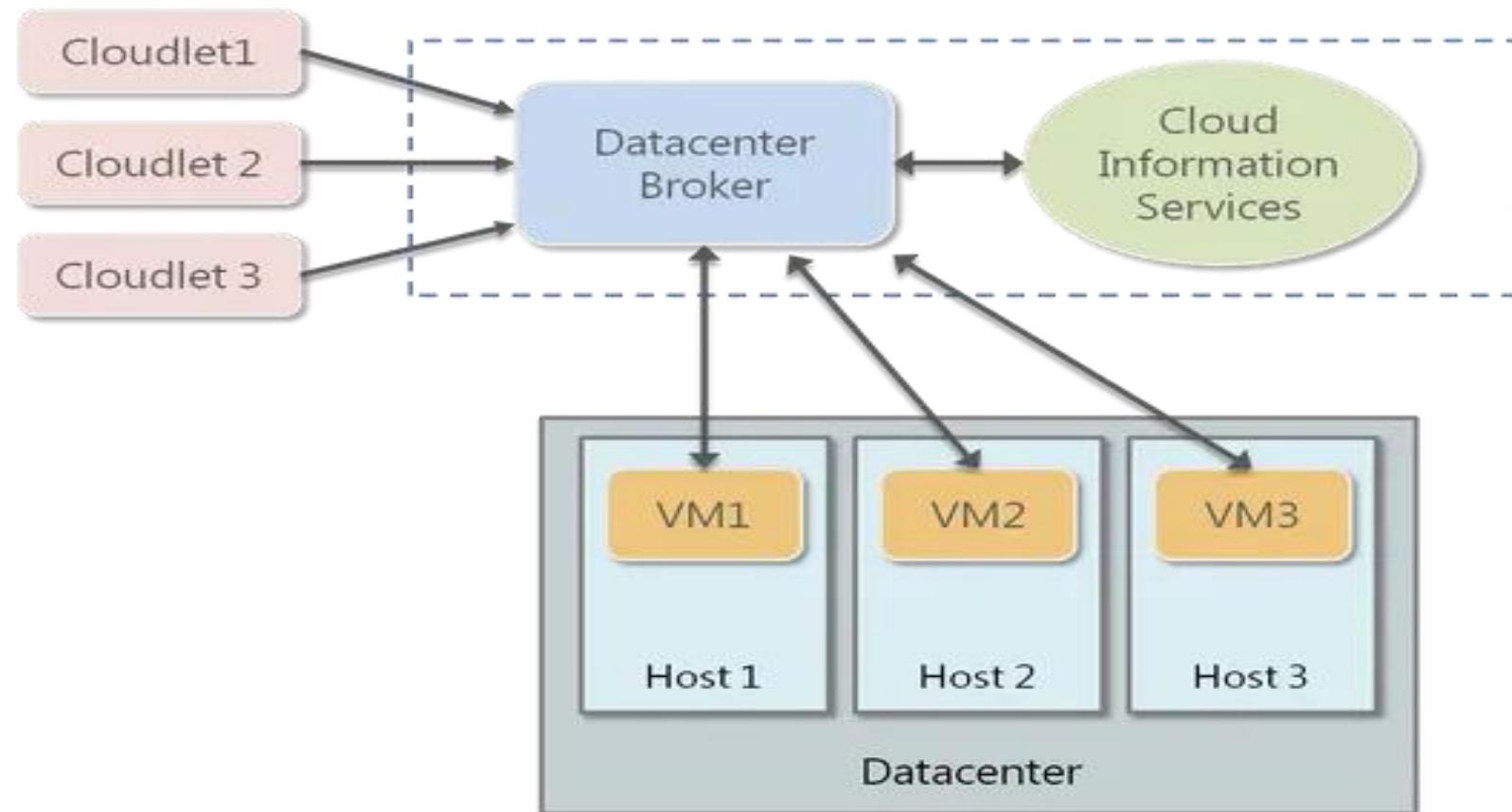
## DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

### **Cloudsim – composantes importants**

- Datacenter
- Host
- DatacenterBroker
- Vm
- VMAllocationpolicy
- VmScheduler
- Cloudlet
- CloudletScheduler
- CloudInformationService
- ...

# DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

## Cloudsim – composantes importants



## DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

### **Cloudsim – composantes importants**

- VM Allocation policy
- Vm Scheduler policy
- Cloudlet Scheduler policy

## DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

### **Cloudsim – Principaux algorithmes et techniques utilisés**

- First come first Serve Algorithm
- Round Robin algorithm
- Min-Min and Max-Min
- Most fit task scheduling
- Priority scheduling algorithm
- Genetic Algorithm
- Particle swarm optimization
- Resource aware scheduling
- Dynamic performance
- ...

## DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

### **Cloudsim – Outils et plugins utilisés**

- CloudAnalyst
- GreenCloud
- iCanCloud
- MDCSim
- NetworkCloudSim
- CloudMIG Xpress
- VirtualCloud
- CloudReports
- Cloud-Auction
- Dynamic CloudSim
- RealCloudSim
- WorkflowSim

## DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

### **CloudSim - Recherches portées**

- Load balancing
- Data centers
- Resource provisioning
- Creation and also execution of Cloudlets
- Task scheduling
- Cost and storage factors
- Energy optimization

## DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

### **CloudSim - Principaux domaines de recherche**

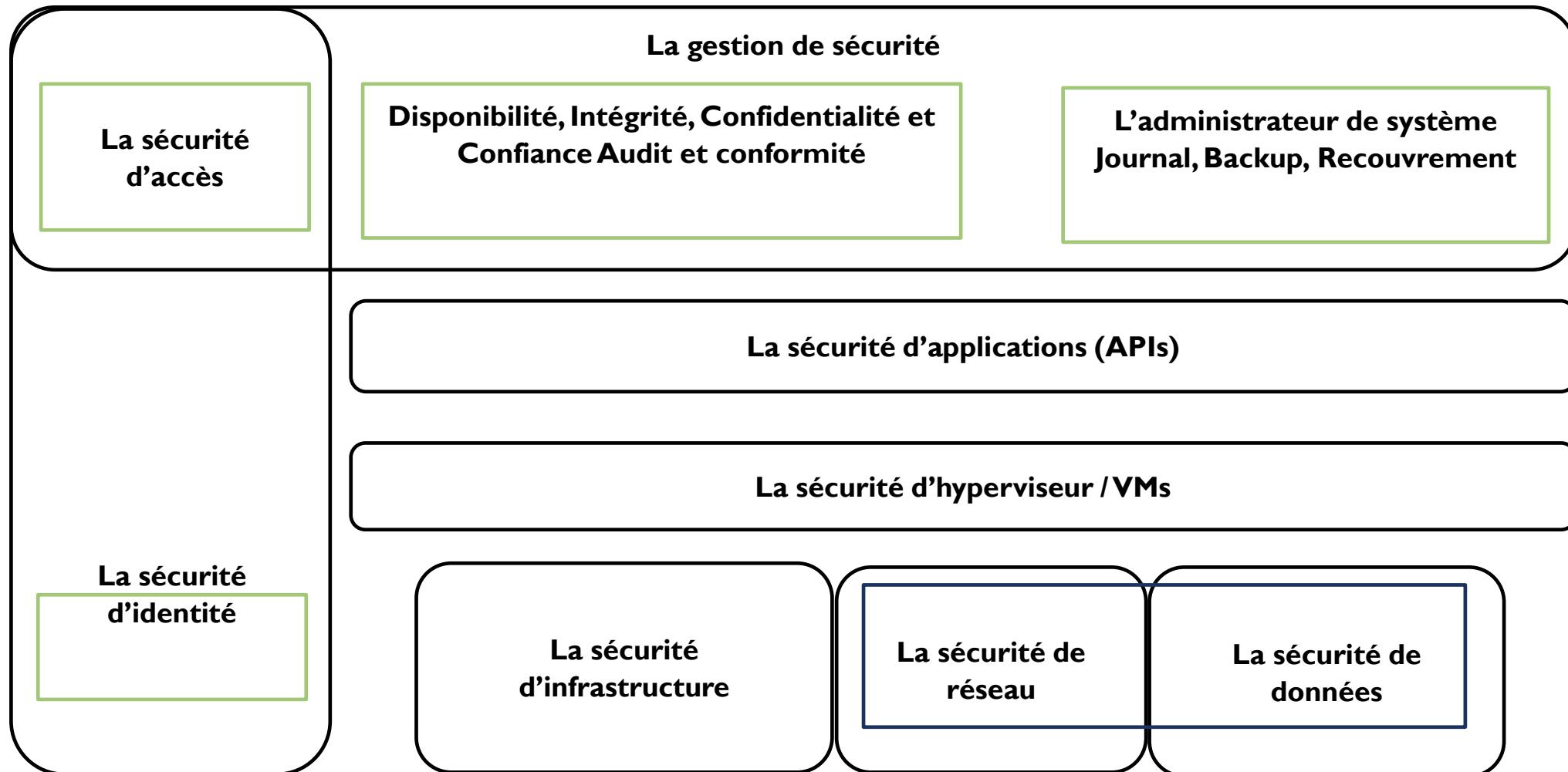
- Security and integrity
- Virtualization also in Cloud
- Privacy provisioning also in multi-tenancy cloud
- Data segregation and also recovery issue
- Securing cloud architecture
- Resource optimization scheduling
- Cloud cryptography and also verifiable computation

## DÉVELOPPEMENT DANS LE CLOUD : LA SIMULATION

### **CloudSim - Principaux domaines de recherche**

- Trusted computing technology and secure computation outsourcing
- Work with failure detection and prediction
- Secure management of virtualized resources and securing mobile cloud
- Key management and cloud access control
- Data segregation and software security
- Security in Data centre for data management
- Privacy aware in protocol design

# LA SÉCURITÉ DANS LE CLOUD



## PRINCIPAUX CHALLENGES DE RECHERCHE DANS LE CLOUD

- **Sécurité**
- **Performance**
- **Disponibilité**
- **Difficulté de l'intégration avec l'IT**
- **Manque de personnalisation**
- **Cout plus élevé pour les Clouds confus**
- **Difficulté d'intégration avec les systèmes internes**
- **Manque de fournisseurs confiants**
- **Problèmes de Migration**
- **Consolidation de serveurs**
- **Ordonnancement**
- **Interopérabilité**

---

---

---

MERCI DE VOTRE ATTENTION

QUESTIONS, REMARQUES ?

---

---

---

# *Cloud Computing*

Présentée par **YAGOUB Mohammed Amine**