

# Cloud Computing

## Définitions et concepts<sup>1</sup>

Danilo Carastan dos Santos

`danilo.carastan-dos-santos@univ-grenoble-alpes.fr`

2024

---

<sup>1</sup>Adapté du support développé par Thomas ROPARS et Renaud LACHAIZE

## Question de départ

**Selon vous, comment peut-on qualifier le Cloud Computing ?**

# Une définition de Cloud Computing

Définie par le NIST : *National Institute of Standards and Technology (USA)*<sup>1</sup>

## Définition

*“Cloud computing is a model for enabling ubiquitous, convenient, **on-demand** network access to a **shared pool of configurable computing resources** (e.g., networks, servers, storage, applications, and services) that can be **rapidly provisioned and released** with minimal management effort or service provider interaction.*

---

<sup>1</sup><http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

# Shared pool of configurable computing resources

Ensemble partagé de ressources informatiques configurables

**Idée** : louer un certain montant de capacité de calcul d'un data-centre physique.

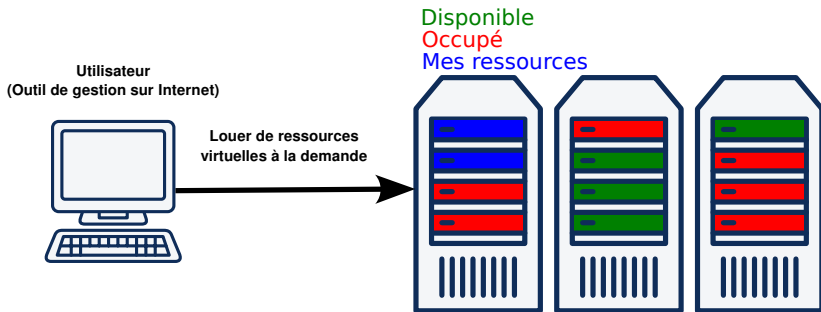
Disponible  
Occupé



# On-demand access

## Accès à la demande

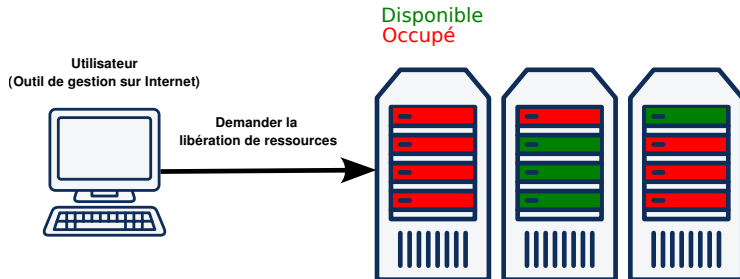
**Idée :** J'ai besoin de ressources informatique (serveurs). J'accède un site web pour louer de serveurs (virtuelles). Je paye le temps d'utilisation. Tout est automatique.



# Rapidly provisioned and released

Rapidement approvisionné et libéré


**Idée :** Je n'ai plus besoin de ressources informatique (serveurs). Je demande la libération de ressources. Les ressources sont libérées.



# Concept clé : Virtualisation

Largeur des rectangles → demande de ressource (nombre de CPUs, quantité de mémoire, etc.)

## Applications à lancer

App 1 

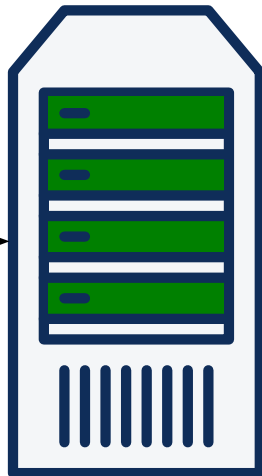
App 2 

App 3 

⋮ 







# Concept clé : Virtualisation

**Sans virtualisation** → seulement une app par machine →  
ressources sous-utilisées

## Applications à lancer

App 1 

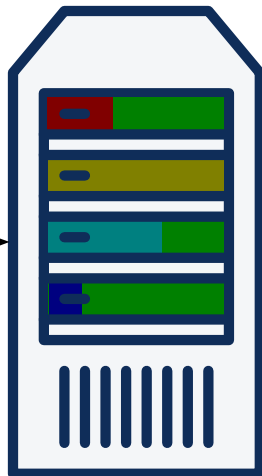
App 2 

App 3 

⋮ 









# Concept clé : Virtualisation

**Avec virtualisation** → plusieurs apps par machine → meilleure utilisation de ressources

## Applications à lancer

App 1 

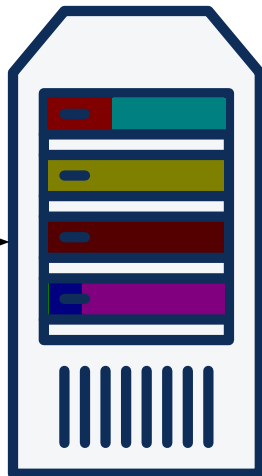
App 2 

App 3 

⋮ 







# Types d'infrastructures Cloud

## Cloud Privé

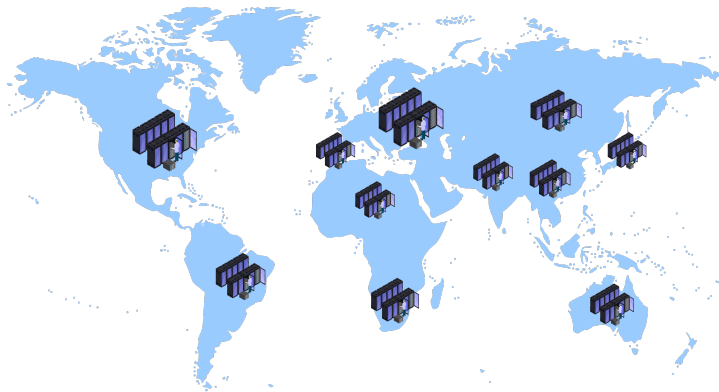
- Infrastructure utilisée **exclusivement** par une organisation
- Mode d'opération typique avant le cloud public
- Toujours pertinent (exemple : données sensibles)



# Types d'infrastructures Cloud

## Cloud Public

- Infrastructure disponible au **grand public** (qui veut payer pour le service) par des fournisseurs de services cloud
- Chaque client a un sous-ensemble isolé de ressources
- Infrastructure multilocataire



# Types d'infrastructures Cloud

## Cloud Hybride

- Un mélange entre le Cloud privé et le Cloud public
- Intérêt :
  - ▶ Augmenter la capacité du Cloud privé avec le Cloud public
  - ▶ Flexibilité en fonction de chaque application (exemple : données sensibles → Cloud privé, sinon → Cloud public)

## Multi Cloud

- Utiliser plusieurs fournisseurs de Cloud public
- Intérêt : Exploiter les avantages de chaque fournisseur en fonction du contexte de chaque application

# Les Hyperscalers

- Fournisseurs Cloud à très grande échelle
- **Amazon AWS, Google Cloud, Microsoft Azure**
- Des infrastructures titanesques<sup>1 2</sup>



---

<sup>1</sup>[https://youtu.be/80aK2\\_iwM0s](https://youtu.be/80aK2_iwM0s)

<sup>2</sup><https://youtu.be/j07V-P7-MBo>

## *“X as a service”*

X en tant que service

- Un certain service (X) qui est géré par un fournisseur Cloud
- Une centaine de services par chaque Hyperscaler. Chaque service a un nom différent. . .
- Liste de services : <sup>1</sup>

**Modèles de services Cloud** : Hiérarchie croissante de sous-traitance de contrôle et responsabilité.

- *Infrastructure as a service* : Infrastructure en tant que service
- *Platform as a service* : Plate-forme en tant que service
- *Function as a service* : Fonction en tant que service
- *Software as a service* : Logiciel en tant que service

---

<sup>1</sup><https://www.techtarget.com/searchcloudcomputing/feature/A-cloud-services-cheat-sheet-for-AWS-Azure-and-Google-Cloud>

# Infrastructure as a service (IaaS)

## Infrastructure en tant que service

- Louer une infrastructure virtuelle
  - ▶ processeur, mémoire, stockage, réseau
- Le fournisseur gère l'Infrastructure physique. L'utilisateur gère l'infrastructure virtuelle
- Utilisateurs sont administrateurs système
  - ▶ Installer/configurer/gérer les mises à jour de sécurité : le système d'exploitation, serveur web, base de données, etc
- Exemple : Google Compute Engine<sup>1</sup>

---

<sup>1</sup><https://cloud.google.com/compute>

# Platform as a service (PaaS)

Plate-forme en tant que service

- Louer une infrastructure virtuelle plus la maintenance des logiciels nécessaires pour l'application
  - ▶ processeur, mémoire, stockage, réseau, serveur web, système de base de données
- Utilisateurs sont développeurs
  - ▶ Le fournisseur Cloud gère l'infrastructure (exemple : mise à jour du système d'exploitation, serveur web, système de base de données)
- Exemples : Google Cloud Run<sup>1</sup> et App Engine<sup>2</sup>

---

<sup>1</sup><https://cloud.google.com/run>

<sup>2</sup><https://cloud.google.com/appengine>



# Software as a service (SaaS)

Logiciel en tant que service

- Logiciel (application) complètement développé et accessible pour les utilisateurs à partir d'un navigateur
- Il suffit d'avoir un navigateur connecté à l'Internet pour utiliser l'application
- Presque tout est transparent à l'utilisateur
  - ▶ Gestion de l'infrastructure, logiciels nécessaires (serveur web, etc) et développement de l'application
- Les utilisateurs ont très peu de contrôle (exemple : localisation des données, contrôle de latence de communication, etc)
- Exemples : Microsoft Office 365, Google Drive, Google Meet

# Function as a service (FaaS)

## Fonction en tant que service

- Aussi mentionné comme *Serverless Computing*
- L'application est développée comme plusieurs fonctions
  - ▶ Chaque fonction attend qu'un événement se déclenche
  - ▶ Une fois déclenchée, la fonction s'exécute. L'infrastructure et la plate-forme sont gérées par le fournisseur
  - ▶ L'utilisateur ne paye que le temps d'exécution de la fonction
- Nécessite une façon spécifique de développer les applications (Microservices)
- Exemple : Amazon AWS Lambda<sup>1</sup>

---

<sup>1</sup><https://aws.amazon.com/lambda/>

# An avertissement sur les technologies Cloud modernes

- Nous sommes actuellement submergés par des technologies
- Il faut choisir les technologies avec parcimonie, pour éviter de développer d'applications trop complexes sans nécessité
  - ▶ Exemple : *A majority of real-world analytic jobs process less than 100GB of input data. For many of such jobs, a single “scale-up” server (i.e., a mid-range multicore server) can do as well or better than a cluster in terms of performance, cost, power and server density.*<sup>1</sup>
  - ▶ Plus d'exemples: <https://blog.bradfieldcs.com/you-are-not-google-84912cf44afb>

---

<sup>1</sup>[R. Appuswamy et al. “Nobody ever got fired for buying a cluster”. 2013.]

# An avertissement sur les technologies du numérique

La sur-numérisation peut aussi desservir

- Exemple : Commentaire de Yassine Lakhnech sur l'allongement des délais de traitement de titres de séjour d'étudiants à l'UGA<sup>1</sup>
- "Ça fonctionnait très bien à Grenoble, mais la dématérialisation a fait que la situation se détériore. La disparition des accueils physiques a provoqué une forme de rigidité. On peut blablater sur l'attractivité internationale, mais si les étudiants et les chercheurs se retrouvent sur le coup d'une OQTF, et ne peuvent plus rentrer chez eux à Noël sous peine de ne pas pouvoir revenir, ça n'a pas de sens".

---

<sup>1</sup><https://mesinfos.fr/38000-grenoble/universite-grenoble-alpes-si-la-securite-n-est-plus-garantie-on-sera-obliges-de-repartir.html>