

# Cloud

Lucas Nussbaum

lucas.nussbaum@univ-lorraine.fr

Licence professionnelle ASRALL

*Administration de systèmes, réseaux et applications à base de logiciels libres*



UNIVERSITÉ  
DE LORRAINE



nancy Charlemagne  
Département Informatique

# Introduction

- ▶ Principe général : dématérialisation des ressources informatiques  
« l'informatique comme un service »  
~> Ressources à la demande, paiement à l'usage.
- ▶ Bref historique :
  - ◆ 2006 : Amazon Web Services (AWS)
  - ◆ 2006 : Google Apps (SaaS)
  - ◆ 2010 : Microsoft Azure
  - ◆ 2012 : Google Compute Engine

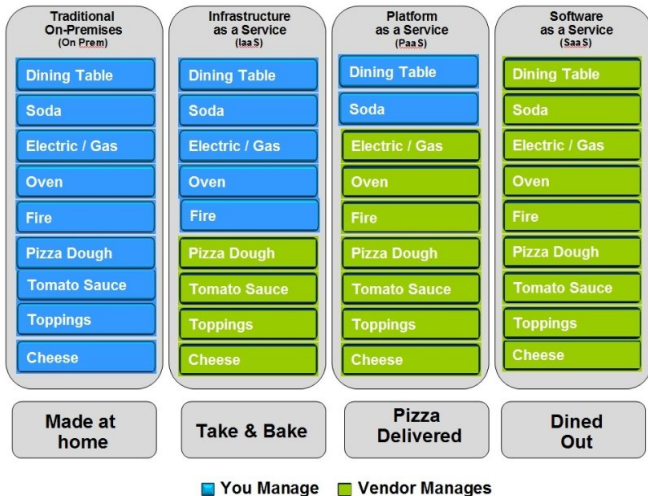
# Types de cloud

Différents types de Cloud :

- ▶ **IaaS** (Infrastructure as a Service) : machines virtuelles, stockage
  - ◆ Amazon Web Services, Google Cloud Platform, Microsoft Azure
  - ◆ Anciennement : Dedibox, Linode
- ▶ **PaaS** (Platform as a Service) : déploiement facilité d'applications pour une certaine plate-forme. intégration avec serveur de base de données.
  - ◆ Google AppEngine (Python), Heroku (Rails), Azure (ASP.NET)
  - ◆ Anciennement : mygale.org 😊, pages perso de Free ?
- ▶ **SaaS** (Software as a Service) : application hébergée
  - ◆ Google Apps (Docs), Salesforce.com
  - ◆ Anciennement : Wordpress.com ?

# Types de clouds – Pizzas

## Pizza as a Service



source : <https://www.linkedin.com/pulse/20140730172610-9679881-pizza-as-a-service>

# Besoins

- ▶ Contexte :
  - ◆ Grand nombre de ressources. Dimensionnement qui varie au cours du temps pour optimiser les coûts. « *pet vs cattle* »
- ▶ Automatisation
  - ◆ Provisioning (par exemple avec *Terraform*)
  - ◆ Gestion de configuration (par exemple avec *Ansible*, *Puppet*, *Salt*, *Chef*, ...)
  - ◆ Déploiement des applications (conteneurs, outils spécifiques au langage, souvent intégré à la gestion de projet (Gitlab, GitHub))
  - ◆ Orchestration (= prendre automatiquement les décisions qui sont habituellement prises par des humains, par exemple « vu la charge, dois-je démarrer une nouvelle VM ? en arrêter une ? »)
- ↳ *Infrastructure as Code*
- ▶ Monitoring (élastique), par exemple avec *Prometheus*
  - ◆ Notamment pour prendre les décisions d'orchestration

# Avantages et inconvénients

- ▶ Avantages :
  - ◆ Consommation au plus proche des besoins (si correctement géré)
  - ◆ Capacité élastique
  - ◆ Gestion de l'infrastructure transférée au fournisseur
  - ◆ Haute disponibilité et redondance facilités
  - ◆ Encouragement de bonnes pratiques
- ▶ Inconvénients :
  - ◆ Nouvelles méthodes, nouveaux outils. Plus difficile qu'une infrastructure gérée de manière artisanale, à l'ancienne !
  - ◆ Coût : pas forcément moins cher
  - ◆ Juridique : localisation, nationalité (↪ *cloud souverain* : Numergy, Cloudwatt (...)) ↪ OVH, 3DS Outscale, Scaleway, OBS)
  - ◆ Sécurité : « The cloud is just someone else's computer »
  - ◆ Vendor lock-in

# TP Cloud : Utilisation de Google Cloud Platform

- ▶ Objectif : manipuler une vraie infrastructure de cloud :  
Google Cloud Platform (GCP)
  - ◆ Grâce à une *Google Cloud Platform Education Grant*
  - ◆ Vous n'avez pas à payer.  
Vous n'avez pas à entrer votre numéro de carte bancaire.  
Ne vous inscrivez pas au *Free Trial* dans le cadre du TP.

# Google Cloud Platform (GCP) : services

- ▶ **Google Compute Engine** – IaaS : machines virtuelles  
Équivalent à Amazon EC2, Azure Virtual Machines
- ▶ **Google App Engine** – PaaS (hébergement d'applications)  
Équivalent à AWS Elastic Beanstalk, Azure Cloud Services
- ▶ **Google Kubernetes Engine** – orchestration de containers  
Équivalent à Amazon EC2 Container Service, Azure Container Service
- ▶ **Google Cloud Bigtable** – IaaS : BDD NoSQL  
Équivalent à Amazon DynamoDB, Azure Cosmos DB
- ▶ **Google Cloud SQL** – IaaS : BDD SQL  
Équivalent à Amazon RDS, Azure SQL Database
- ▶ **Google Storage** – IaaS : stockage objet  
Équivalent à Amazon S3, Azure Blob Storage
- ▶ et d'autres



# Accès à Google Cloud Platform

- ▶ Suivez le lien fourni par mail.
- ▶ Allez sur `https://console.cloud.google.com/start`

## Tutoriels à suivre

- ▶ *Build a to-do app with MongoDB*

<https://cloud.google.com/community/tutorials/compute-quickstart>

- ◆ Au lieu de l'application NodeJS/MongoDB fournie dans le TP, vous pouvez utiliser un couple base de données / application que vous maitrisez (par exemple le Wordpress/MariaDB du module Supervision, un simple couple PHPMyAdmin/MariaDB, ou un site web développé pour un autre module). Si vous utilisez Wordpress/MariaDB, regardez le Vagrantfile pour savoir comment permettre à une machine distance de se connecter à la base de données.

- ▶ *Apprendre à utiliser Cloud Storage*

<https://console.cloud.google.com/start>

- ▶ *Votre première application (App Engine)*

<https://console.cloud.google.com/appengine/start>

## Tutoriels à suivre (2)

- ▶ Modifiez l'exemple du premier tutoriel pour utiliser une base de données créée avec Google Cloud SQL.
- ▶ Puis ajoutez de la répartition de charge avec *Google Cloud Load Balancing*.
- ▶ Explorez les autres services disponibles sur GCP
  - ◆ Tutoriels supplémentaires :  
<https://codelabs.developers.google.com/?cat=Cloud>
- ▶ Essayez d'automatiser certaines opérations (création de VM, de base de données) avec la console