



Cloud & Architecture

de la donnée



Par

Hamadi Camara

Ing. IA / Solution Architect / Founder



Objectifs du module



**Introduction au cloud
computing & conception
d'applications cloud natives**



**Gestion des données et
sécurité dans le cloud**



**Architectures avancées &
stratégie de migration
vers cloud**

Planning

01

02

03

Chap. 1.

1. Introduction à l'architecture cloud
2. Applications cloud natives

Chap. 2.

1. Gestion des données dans le cloud
2. Sécurité des données dans le cloud

Chap. 3.

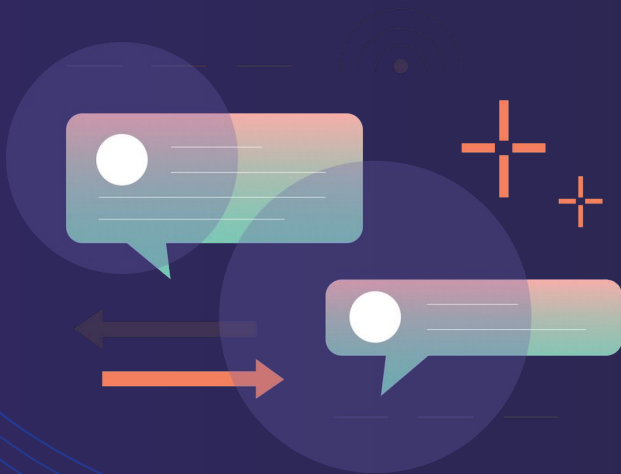
1. Analyse de données & Big Data
2. Architectures avancées & Migration

Cours • TD (Travaux Dirigés) • TP (Travaux Pratiques)

Et si on faisait connaissance
durant une partie de...**Quiz**



Morning Quiz 1/2



1. Qu'est ce qu'une application cloud native ?
2. Qu'est ce qu'un microservice ?
3. Qu'est que le design-pattern en architecture ?
4. Qu'est ce qu'un conteneur ?
5. Qu'appelle-t-on architecture client-serveur ?

Morning Quiz 2/2



6. Qu'est-ce que le CI/CD ?
7. Qu'est ce que le cloud ?
8. C'est quoi kubernetes ?
9. Qu'est ce que kafka ?
10. C'est quoi AWS ?

Au programme

01

Un peu
d'histoire

02

Quelques concepts
Bons à savoir

03

Introduction au **cloud**
computing

04

Modèles de
déploiements.

05

Discovery
Solution du jour

06

Travaux Pratiques
& Projets

Et si on remontait un
peu dans le **temps**.



Un peu d'histoire

1968

IBM lance son
OS virtualisé

1974

Fondation de **Microsoft**

1971

Envoi du **1er e-mail**

1976

Fondation d'**Apple Computer**

Un peu d'histoire

1989

Lancement d'internet avec
1M de machines connectés

2007

Lancement AWS,
solution cloud Amazon

1997

1ère mention du **Cloud
computing** par le professeur
Ramnath K. Chellappa

Maintenant

Domination du marché
du cloud par
GCP, AWS et AZURE

Quelques **concepts**
bons à savoir...



Bon à savoir...



On Premise

Consiste à **déployer** une application sur un **serveur informatique** appartenant à **l'entreprise**.

Bon à savoir...



Cloud Computing

Se compose des ressources, de l'infrastructure et des outils fournis **à la demande par les fournisseurs de cloud**

Bon à savoir...



Cloud Natif

Une approche qui crée et exécute des logiciels avec le modèle de **cloud computing**.

Bon à savoir...



Data Center

Désigne un ensemble de serveurs.

Bon à savoir...



Virtualisation lourde :

- Des machines virtuelles complètes
Ex : Virtualbox, VMware, Xen

Virtualisation légère :

- Ensemble de processus séparés des autres processus d'une machine
Ex : Docker, Rackspace

Le **cloud computing**
dans tous ses états...



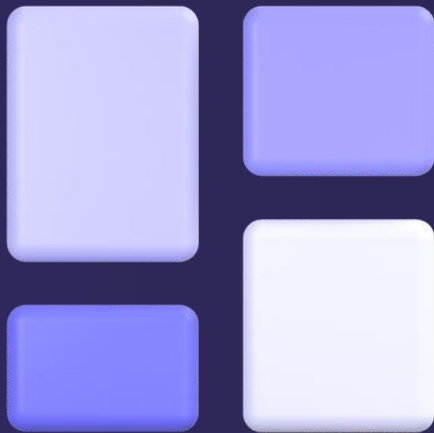
Une philosophie



L'usage de services dont vous ne gérez pas le matériel (= serveurs physiques)

- Un opérateur gère le matériel et multiplexe les utilisateurs
- Les utilisateurs sont des particuliers ou des entreprises

Software as a Service(SaaS)



Un **service** basé sur le **cloud** où vous accédez à une **application** directement via un **navigateur internet**.

Software as a Service(SaaS)

WHY?



- **Ne souhaitent pas acheter ou entretenir une infrastructure,**
- **Préfèrent une gestion plus simple des coûts par les coûts d'exploitation,**
- **Privilégient les modèles de souscription pour les logiciels.**

Exemples de solutions SaaS

Les éditeurs proposent une diversité de logiciels SaaS :

- **SAP**
- **ServiceNow**
- **(CRM) de Salesforce**
- **Slack**
- **Microsoft Office 365**
- **Dropbox**



Platform as a Service (PaaS)



Offre de cloud computing dans lequel un fournisseur de services fournit une plateforme à ses clients.

Platform as a Service (PaaS)

- **Utilisez les compétences et investissements** à votre disposition.
- **Réduisez les coûts.** Avec le PaaS, vous ne payez que ce que vous utilisez.
- **Accélérez les cycles de développement** des applications.
- **Améliorez l'efficacité des pratiques DevOps.**
- Assurez la **sécurité**.
- **Augmentez la productivité.** Les développeurs ont rapidement accès aux outils et ressources nécessaires grâce à des capacités en libre-service.

WHY?



Exemples de solutions PaaS

Les éditeurs proposent une diversité de logiciels PaaS :

- **Google App Engine**
- **OpenShift**
- **Heroku**
- **Force.com**
- **Windows Azure**
- **AWS Elastic Beanstalk**
- **Apache Stratos**
- **Magento Commerce**
- **AWS Lambda**
- **SAP Cloud**



Infrastructure as a Service(IaaS)



Information dotée **d'un sens** (dans un univers logique), permettant de **modéliser la réalité**, de **guider une action**.

Infrastructure as a Service (IaaS)

- **Flexibilité** : achetez uniquement les composants dont vous avez besoin pour votre cas d'utilisation et faites-les évoluer en fonction de vos besoins métier.
- **Prix** : l'IaaS est une solution économique puisqu'elle implique peu de frais et aucun coût de maintenance.
- **Contrôle** : l'utilisateur garde le contrôle de son infrastructure.
- **Sécurité** : le fournisseur est-il fiable et dispose-t-il des ressources nécessaires pour éviter et gérer les menaces pour la sécurité

WHY?



Exemples de solutions IaaS

Les éditeurs proposent une diversité de logiciels IaaS :

- **Google Cloud Platform (GCP)**
- **Amazon Web Services (AWS)**
- **IBM Cloud**
- **Microsoft Azure**
- **Rackspace**
- **Cisco Metacloud**
- **Oracle Cloud**



Vue d'ensemble

On-site	IaaS	PaaS	SaaS
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking



You manage

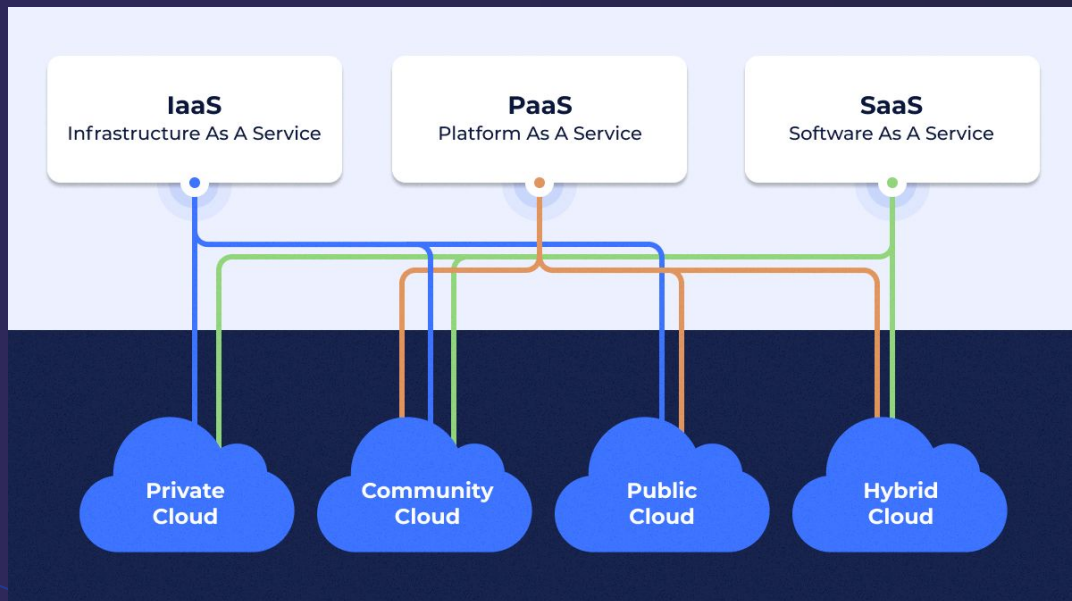


Service provider manages

Les **modèles de déploiements** dans le cloud.



Vue d'ensemble



Cloud Public



Usage collectif (bien que cloisonné),
déployé par fournisseur cloud tiers.
Il est généralement **ouvert au public et**
partagé par les utilisateurs.

Cloud Public

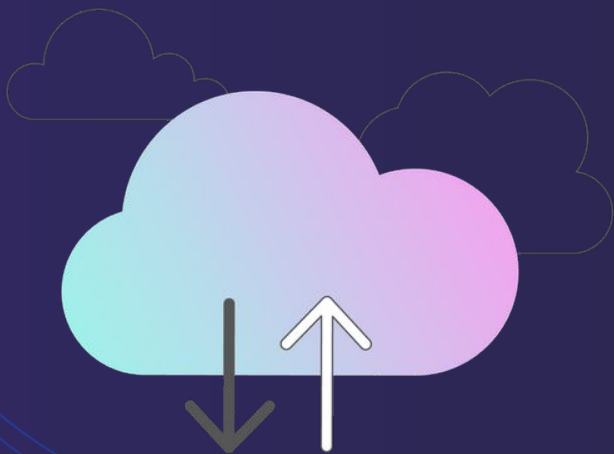
Avantages

1. **Pas nécessaire d'investir** dans du nouveau matériel pour faire évoluer l'infrastructure
2. Modèle de tarification "**Pay-as-you-go**",
3. **Aucune responsabilité en matière de maintenance** et de mise à jour du matériel.
4. Seules des **connaissances techniques minimales**
5. Les services sont **accessibles à tous via l'internet.**

Désavantages

1. **Il peut y avoir des politiques gouvernementales, des normes industrielles** ou des exigences légales auxquelles les plateformes d'informatique dématérialisée ne peuvent pas répondre.
2. Comme vous n'êtes pas propriétaire du matériel et des services, **il peut y avoir des restrictions d'utilisation.**
3. **Il arrive que des exigences commerciales** uniques ne puissent pas être satisfaites.

Cloud Privé



Usage privatif par un seul organisme,
Un déploiement généralement en On
Premise.

Cloud Privé

Avantages

1. **Configuration des application** ou scénario existant.
2. **Contrôle la sécurité.**
3. **Répondre à des exigences strictes en matière de sécurité,** de conformité ou de législation.

Désavantages

1. **Des dépenses initiales en capital sont nécessaires** pour l'achat et la maintenance du matériel.
2. L'utilisation et l'exploitation de l'informatique dématérialisée **nécessitent des compétences plus élevées.**
3. **Les possibilités de vente sont limitées** car vous devrez acheter et installer le nouveau matériel.

Cloud Hybride



Mélange de plusieurs modèles de cloud reliés entre eux offrant les **avantages des différents environnements.**

Cloud Hybride

Avantages

1. **Bénéficier d'une plus grande souplesse.**
2. **Profiter des économies d'échelle** de l'informatique dématérialisée.
3. **Utiliser ses équipements** pour répondre à des **scénarios de sécurité, de conformité ou d'héritage.**

Désavantages

1. **Complicé à mettre en place** et à gérer.
2. **Plus coûteux que l'utilisation d'un seul type de techniques** de déploiement.

Les applications cloud natives.



Application Cloud Native



Une application "cloud-native"
est un logiciel conçu pour
fonctionner dans un environnement
"cloud computing".

Avantages

- **Une évolutivité facilitée**, notamment face à la croissance des nouveaux utilisateurs de l'application.
- **La réduction des coûts liés à l'hébergement** de l'application.
- **Une automatisation accrue** qui libère du temps et des ressources aux développeurs.



Discovery

L'outil du jour

AWS



Overview

AWS Services

Deployment & Management

Application Services



Mobile Services



Enterprise Applications



Application Services

Administration & Security



Deployment & Management



Analytics



Foundation Services

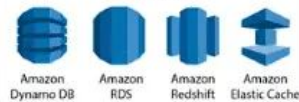
Compute



Storage & Content Delivery



Database



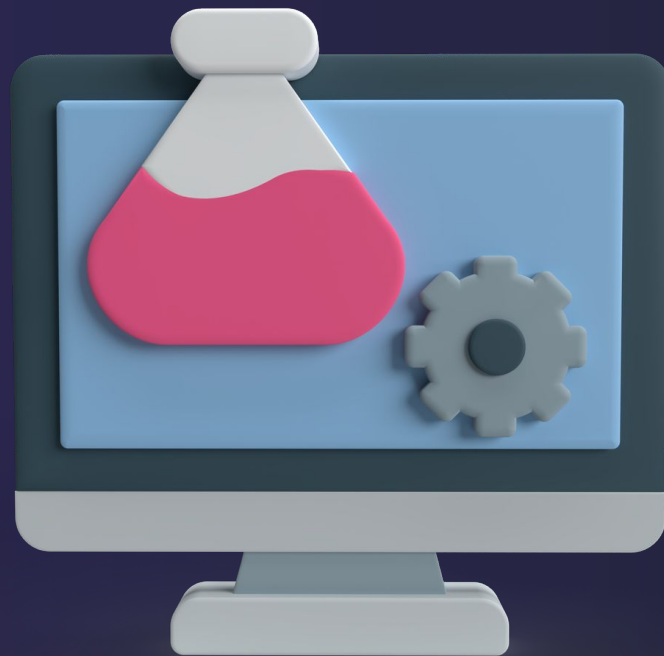
Networking





Travaux Pratiques

- Docker
- Docker-compose
- MySQL
- phpMyAdmin



Bon à savoir...



Les conteneurs sont similaires aux machines virtuelles, mais ils sont beaucoup plus légers et plus rapides à démarrer.

Ils partagent le même système d'exploitation que le système hôte sur lequel ils sont exécutés, ce qui permet de les utiliser de manière très efficace et de bénéficier de tous les avantages des conteneurs, **tels que la portabilité, l'isolation et la scalabilité.**

Docker

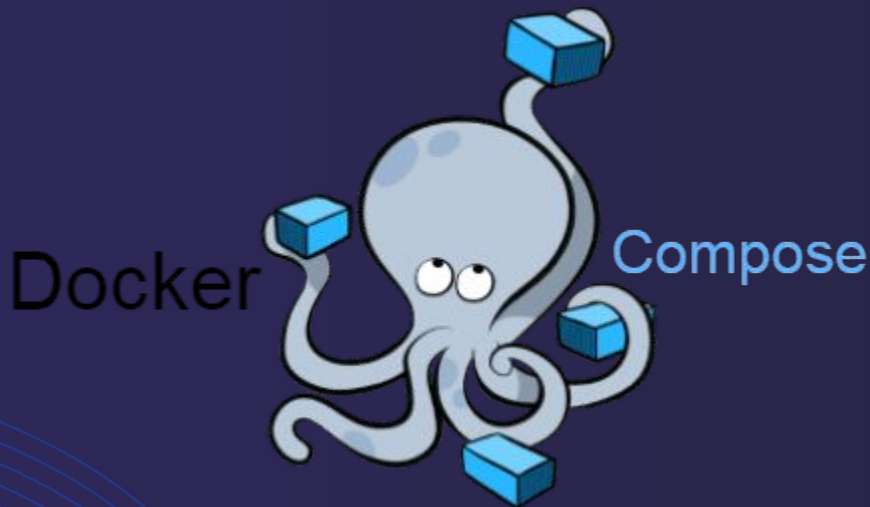


Docker est un logiciel de conteneurisation de code.

Il permet de **créer**, de **déployer** et **d'exécuter** des applications dans des **conteneurs logiciels**, qui sont des environnements isolés et légers.

Docker-compose

Un outil permettant de configurer et gérer des environnements à conteneurs multiples au travers d'un fichier yml.



Docker – Composant

- **Le Docker Hub** est le registre officiel de Docker. Il s'agit d'un répertoire SaaS permettant de gérer et de partager les conteneurs.
- **Les images Docker** est un modèle en lecture seule, utilisé pour créer des conteneurs Docker.
- **Les conteneurs Docker** est une instance d'image Docker exécutée sur un microservice individuel ou un stack d'application complet.

Etapes.

Commandes utiles :

\$> docker-compose build

\$> docker-compose up

- Création d'un fichier docker-compose.yml
- Définition du service MariaDB
- Définition du service PhpMyAdmin
- Définition du service WordPress

Création d'un fichier docker-compose.yml

```
version: "3"
```

```
Services:
```

```
  Service A:
```

```
    clé1: "valeur"
```

```
    clé2: "valeur"
```

```
  Service B:
```

```
    clé1: "valeur"
```

```
    clé2: "valeur"
```


Service

MariaDB

```
mariadb:
```

```
  image: "mariadb:latest"
```

```
  restart: 'always'
```

```
  volumes:
```

```
    - "./data:/var/lib/mysql/data"
```

```
    - "./logs:/var/lib/mysql/logs"
```

```
    - /var/docker/mariadb/conf:/etc/mysql
```

```
  Environment:
```

```
    MYSQL_ROOT_PASSWORD: "root"
```

```
    MYSQL_DATABASE: "heticdb"
```

```
    MYSQL_USER: "hetic"
```

```
    MYSQL_PASSWORD: "hetic"
```

Service

PhpMyAdmin

```
Phpmyadmin:
  image: phpmyadmin
  restart: always
  expose:
    - "40001"
  ports:
    - "40001:80"
  environment:
    - PMA_HOST=mariadb
    - PMA_PORT=3306
```

Service

WordPress

```
Wordpress :  
  image: wordpress  
  expose:  
    - 80  
  Ports:  
    - 80:80  
  environment:  
    WORDPRESS_DB_HOST : mariadb  
    WORDPRESS_DB_NAME : heticdb  
    WORDPRESS_DB_USER : hetic  
    WORDPRESS_DB_PASSWORD : hetic  
  volumes :  
    - ./wordpress:/var/www/html
```

Projets

à vos marques, prêts, partez.

