Cloud Computing Pour le Big Data

Ibrahima SOUMARE


LET’
S
GO

Plan

Module 1 :

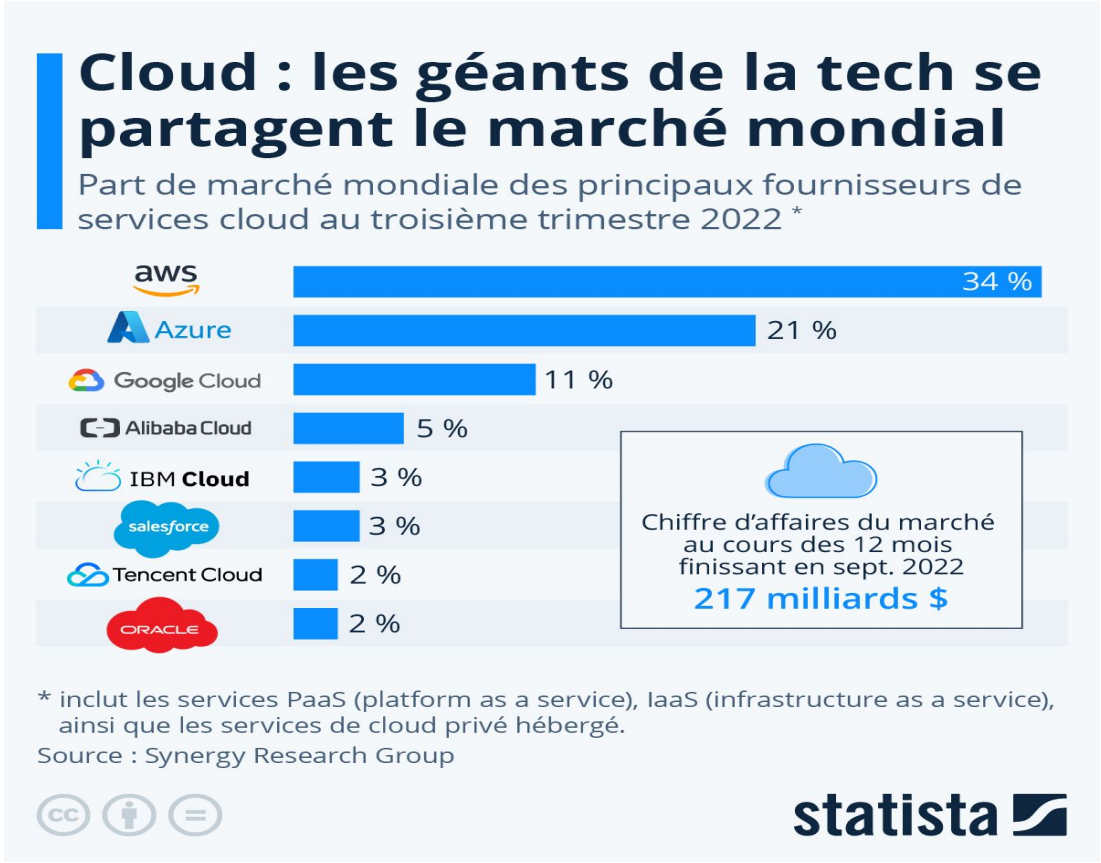
Définition et historique du Cloud Computing.
Avantages et inconvénients du Cloud.
Concepts clés : virtualisation, élasticité, multi-location, scalabilité.
Acteurs majeurs du Cloud (AWS, Azure, Google Cloud, etc.).

Module 2 :
IaaS (Infrastructure as a Service) : Machines virtuelles, stockage.
PaaS (Platform as a Service) : Outils pour développeurs, frameworks.
SaaS (Software as a Service) : Applications cloud (Gmail, Salesforce).
Différences entre les modèles et cas d'utilisation

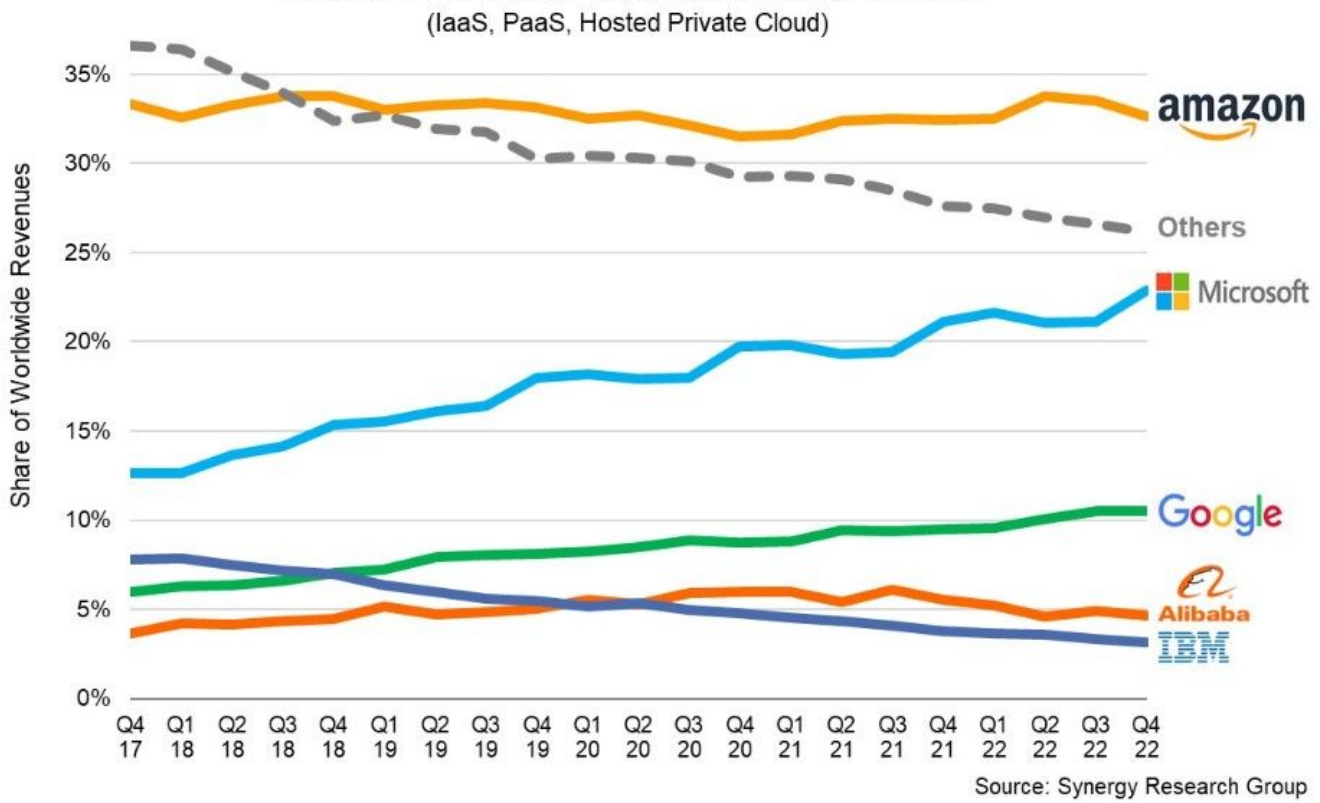
Module 3 :
IAM (Identity Access Management)
Services de stockage (S3, Google Cloud Storage, etc.).
Bases de données dans le cloud AWS (**RDS, DynamoDB**).
Bases de données dans le cloud GCP (S**QL, Bigquery**).

Module 4 :
Services de calcul AWS (EC2, Lambda, Glue, etc.).
Services de calcul GCP (VM, Functions, DataProc.).
Services IA et Machine Learning dans le cloud (AWS SageMaker, Google AI Platform).

Acteurs majeurs du Cloud (AWS, Azure, Google Cloud, etc.)

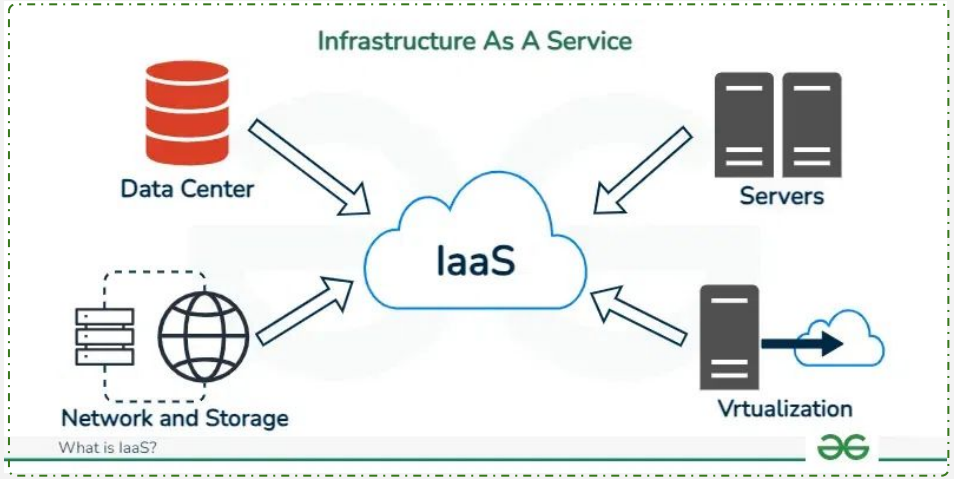


Acteurs majeurs du Cloud (AWS, Azure, Google Cloud, etc.)



Module 2

IaaS (Infrastructure as a Service) : Machines virtuelles, stockage.



Caractéristiques :

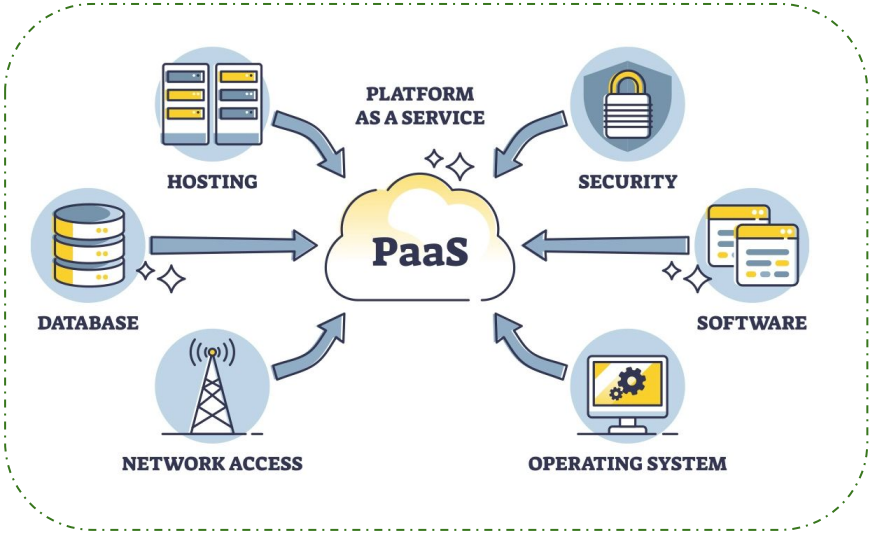
- **Responsabilité utilisateur :** Gérer le système d'exploitation, les applications, les données, les configurations réseau.
- **Flexibilité :** Ajout ou suppression de ressources en fonction des besoins.
- **Principaux fournisseurs :** AWS EC2, Azure Virtual Machines, Google Compute Engine.

L'IaaS fournit des infrastructures informatiques virtualisées telles que des serveurs, des réseaux, du stockage et des systèmes d'exploitation, accessibles via Internet. Les utilisateurs gèrent ces ressources comme s'il s'agissait d'un datacenter physique.

Cas d'utilisation :

1. Hébergement de sites web
2. Développement et tests
3. Stockage massif

PaaS (Platform as a Service)



Le PaaS fournit une plateforme complète pour le développement, le test et le déploiement d'applications. Il élimine la nécessité de gérer l'infrastructure sous-jacente.

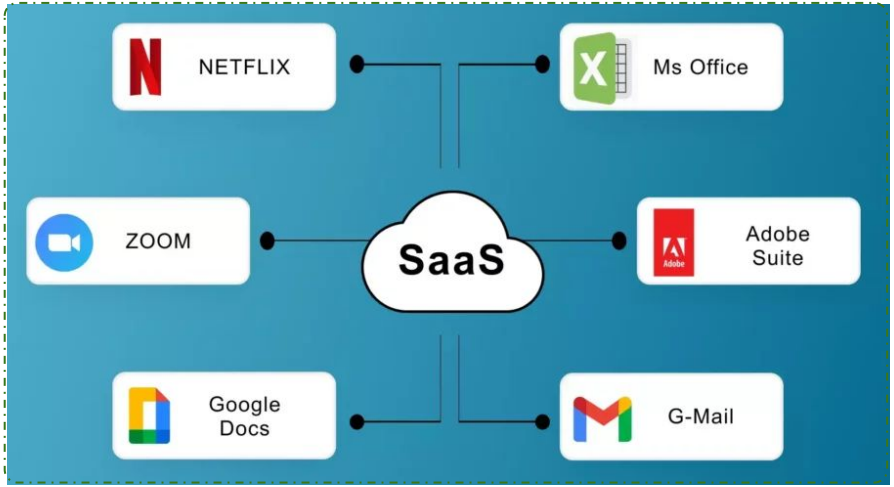
Caractéristiques :

- **Responsabilité utilisateur :** Se concentrer sur le développement d'applications (le fournisseur gère les serveurs, le stockage, le réseau et les mises à jour).
- **Environnement intégré :** Outils préconfigurés pour le codage, les bases de données et les analyses.
- **Principaux fournisseurs :** Google App Engine, Microsoft Azure App Services, AWS Elastic Beanstalk.

Cas d'utilisation :

1. Développement d'applications mobiles/web
2. Automatisation des pipelines CI/CD
3. Intégration avec l'IA et les Big Data

SaaS (Software as a Service)



Le SaaS propose des logiciels accessibles via Internet, souvent par abonnement. L'utilisateur n'a rien à installer ou gérer localement.

Caractéristiques :

- **Responsabilité utilisateur :** Accès et utilisation de l'application.
- **Maintenance et mises à jour :** Gérées par le fournisseur.
- **Principaux exemples :** Gmail, Salesforce, Microsoft Office 365, Dropbox.

Cas d'utilisation :

1. **Collaboration en équipe :** Utilisation d'outils comme Google Workspace.
2. **Gestion des relations client :** CRM avec Salesforce.
3. **Stockage en ligne :** Services comme Dropbox ou OneDrive.

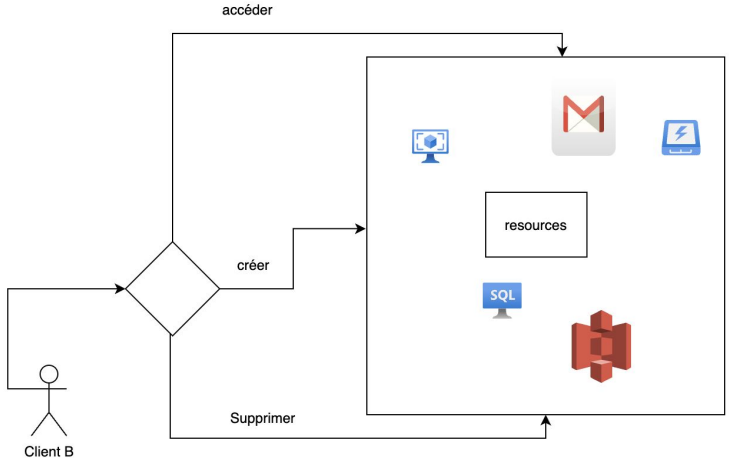
Résumé des Cas d'Utilisation

- **IaaS** : Utilisé par les entreprises ayant besoin de contrôle total sur l'infrastructure, comme les startups technologiques ou les grandes organisations.
- **PaaS** : Idéal pour les développeurs souhaitant se concentrer sur le code sans gérer l'infrastructure.
- **SaaS** : Parfait pour les entreprises et particuliers cherchant des solutions prêtes à l'emploi pour la collaboration, la productivité et la gestion des données.

En fonction des besoins d'une entreprise ou d'un projet, ces modèles peuvent être combinés pour offrir une solution cloud hybride adaptée.

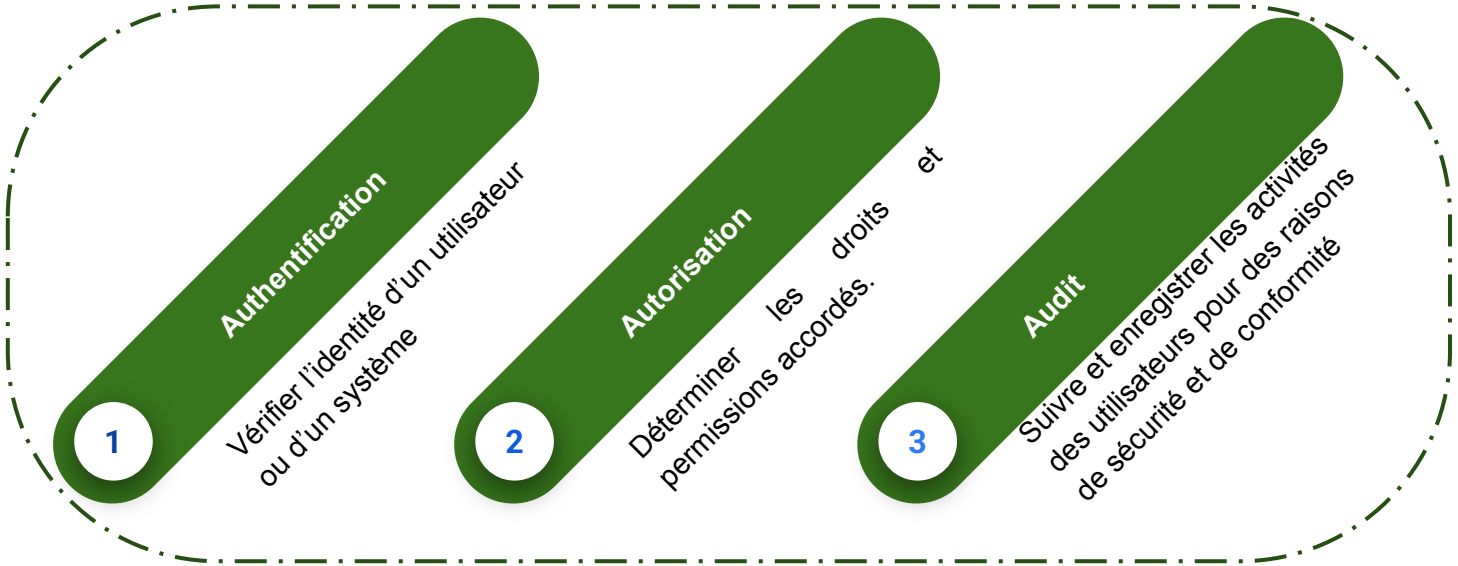
Aspect	IaaS	PaaS	SaaS
Cible principale	Administrateurs systèmes	Développeurs	Utilisateurs finaux
Responsabilité	OS, middleware, données	Applications, données	Utilisation uniquement
Exemples de services	AWS EC2, Azure VMs	Google App Engine	Gmail, Salesforce
Complexité technique	Élevée	Moyenne	Faible
Flexibilité	Très haute	Moyenne	Limitée

Introduction à l'Identity Access Management (IAM)



L'**Identity and Access Management (IAM)** est un domaine essentiel dans la gestion de la sécurité informatique. Il permet de contrôler qui a accès à quoi dans un système, de façon à protéger les ressources, les données sensibles et les systèmes critiques.

Objectifs d’un système IAM



Concepts fondamentaux d'IAM

1. **Utilisateur** :
   - Un individu ou un système qui interagit avec le système.
   - Exemple : employé, client, application.
2. **Identité** :
   - Une entité unique permettant d’identifier un utilisateur dans le système.
   - Elle peut inclure des identifiants comme le nom d’utilisateur, le mot de passe, ou des attributs comme l’adresse email ou l’ID employé.
3. **Rôle** :
   - Un ensemble de permissions regroupées selon une fonction donnée.
   - Exemple : "Admin", "Développeur", "Consultant".
4. **Principes du Moindre Privilège** :
   - Un utilisateur ou une application doit disposer uniquement des permissions nécessaires pour accomplir sa tâche.

Architecture IAM

Un système IAM repose généralement sur les composantes suivantes :

1. **Gestion des identités** :
    - Création, mise à jour et suppression des identités.
    - Exemples : bases d’utilisateurs, annuaires LDAP, Active Directory.
2. **Contrôle d'accès** :
    - Définir et appliquer les politiques d'accès aux ressources.
    - Méthodes courantes :
        - **RBAC (Role-Based Access Control)** : Contrôle basé sur les rôles.
        - **ABAC (Attribute-Based Access Control)** : Contrôle basé sur des attributs (ex. localisation, heure, etc.).
3. **Authentification** :
    - Vérification de l’identité.
    - Méthodes :
        - **Facteur unique** : Mot de passe.
        - **Multi-Facteurs (MFA)** : Mot de passe + un autre facteur (SMS, biométrie).
4. **Autorisation** :
    - Processus de décision pour savoir si un utilisateur ou une entité peut accéder à une ressource.
5. **Audit et Reporting** :
    - Journaux d'activités et rapports pour surveiller l’utilisation des permissions.

Cycle de vie des identités

1. **Provisionnement** :
    - Création de l’identité lors de l’embauche ou de l’inscription.
2. **Gestion** :
    - Mise à jour des droits en fonction des évolutions des rôles ou responsabilités.
3. **Désactivation** :
    - Suppression ou désactivation de l’accès après une démission, un licenciement ou une fin de contrat.

Politiques et standards d’IAM

1. **Mots de passe robustes** :
   - Longueur minimale, combinaison de caractères, renouvellement périodique.
2. **MFA (Authentication Multi-Facteurs)** :
   - Réduit le risque de compromission par vol de mot de passe.
3. **Revue des permissions périodiques** :
   - Vérifier régulièrement si les permissions sont toujours appropriées.
4. **Conformité aux réglementations** :
   - GDPR, HIPAA, PCI-DSS imposent des normes strictes pour la gestion des accès.

IAM dans le Cloud

Avec l’émergence des services Cloud, IAM a évolué pour répondre à de nouveaux besoins.

1. **IAM dans AWS** :
    - Gestion des utilisateurs et des rôles pour accéder aux ressources AWS.
    - Exemples : Politiques IAM, groupes, MFA.
2. **IAM dans Azure** :
    - Contrôles RBAC pour accéder aux ressources Azure.
    - Azure AD pour la gestion des identités.
3. **IAM dans Google Cloud Platform (GCP)** :
    - Gestion des permissions via des rôles prédéfinis ou personnalisés.

Meilleures pratiques

1. **Implémenter le principe du moindre privilège**.
2. **Activer le MFA pour tous les utilisateurs**.
3. **Automatiser le provisionnement et la désactivation des identités**.
4. **Utiliser des outils de gestion centralisée des identités**.
5. **Effectuer des audits réguliers des accès et permissions**.