IAAS

Fondations

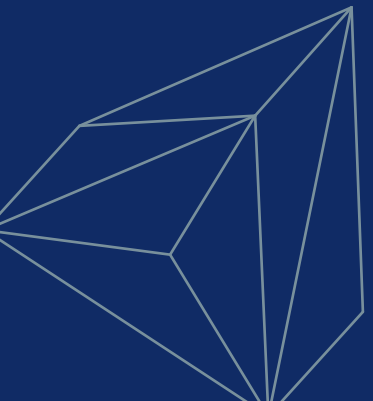
Loïc Rouquette

Sommaire

Objectifs de la séance

- Distinguer les caractéristiques fondamentales d'une solution d'hébergement locale (on-premises) et d'une solution mutualisée (cloud) (AAV 4).
- Différencier les principaux modèles de service cloud : IaaS, PaaS (Platform as a Service) et SaaS (Software as a Service).
- Identifier et décrire les rôles des composants majeurs d'une infrastructure IaaS : machines virtuelles (VM), stockage (Bloc et Objet), réseau virtuel et mécanismes de sécurité de base (AAV 1).
- Comprendre les principes fondamentaux de l'estimation des besoins en ressources pour un service (AAV 2).
- Connaître l'environnement technique qui sera utilisé lors des travaux pratiques (TP).

Introduction aux modèles d'hébergement



Qu'est que l'hébergement d'applications ?

L'hébergement cloud fournit des ressources sur Internet en utilisant l'infrastructure d'un fournisseur de cloud. Cette infrastructure exploite la virtualisation pour abstraire un serveur physique et répartir les ressources sur un réseau de serveurs virtuels et physiques, ceux-ci étant généralement répartis dans plusieurs datacenters. L'hébergement cloud diffère de l'hébergement traditionnel, dans la mesure où ce dernier repose uniquement sur des machines physiques sur site (ou « on-premises »).

— OVH.com

Qu'est que l'hébergement d'applications ? (i)

On-Premises (Locale)

Propriété et Contrôle Total :

- L'entreprise possède matériel (serveurs, réseau…) et licences logicielles.
- Maîtrise complète de la configuration, sécurité, données, technologies.
- Avantage pour secteurs réglementés ou à haute confidentialité.

Qu'est que l'hébergement d'applications ? (ii)

On-Premises (Locale)

Responsabilité Totale :

- Gestion intégrale : Achat, installation, configuration, maintenance, mises à jour, sécurité, surveillance, alimentation, refroidissement, pannes.
- *Nécessite une expertise technique interne significative.*

Qu'est que l'hébergement d'applications ? (iii)

On-Premises (Locale)

Modèle Financier - CapEx Dominant :

- Fortes dépenses d'investissement initiales (Capital Expenditures).
- Coûts amortis sur plusieurs années.
- OpEx (électricité, personnel) existent mais l'investissement initial prédomine.

Qu'est que l'hébergement d'applications ? (iv)

On-Premises (Locale)

Inconvénients Associés :

- Manque de Flexibilité / Agilité : Scaling lent (achat/installation de matériel).
- Planification de Capacité Complexe : Risque de sur-provisionnement (coûteux) ou sous-provisionnement (performance dégradée).
- Obsolescence Technologique : Risque à gérer par l'entreprise.
- Coûts Cachés : Pannes, maintenance imprévue, mises à niveau.

Qu'est que l'hébergement d'applications ? (v)

Cloud Computing (Mutualisée)

Location de Ressources :

- Location de ressources virtualisées (calcul, stockage, réseau…) hébergées dans les datacenters du fournisseur.
- Pas d'achat ni de possession de matériel physique par le client.

Qu'est que l'hébergement d'applications ? (vi)

Cloud Computing (Mutualisée)

Gestion Déléguée de l'Infrastructure Physique :

- Le fournisseur cloud gère, maintient et sécurise l'infrastructure physique sous-jacente.
- Le client interagit via des interfaces logicielles (consoles, API).

Qu'est que l'hébergement d'applications ? (vii)

Cloud Computing (Mutualisée)

Modèle Financier - OpEx Dominant :

- Dépenses d'exploitation basées sur la consommation réelle ("pay-as-you-go") ou abonnements.
- Élimine ou réduit fortement les investissements initiaux (CapEx).

Qu'est que l'hébergement d'applications ? (viii)

Cloud Computing (Mutualisée)

Élasticité et Scalabilité Rapides :

- Ajustement rapide (souvent en minutes) des ressources (↑↓) pour s'adapter à la demande.
- Possibilité d'auto-scaling pour gérer les pics de charge automatiquement.
- Évite le surprovisionnement.

Qu'est que l'hébergement d'applications ? (ix)

Cloud Computing (Mutualisée)

Accès Réseau Étendu & Self-Service :

- Ressources accessibles via le réseau (Internet) depuis n'importe où.
- Provisionnement des ressources par l'utilisateur via consoles web ou API.

Qu'est que l'hébergement d'applications ? (x)

Cloud Computing (Mutualisée)

Service Mesuré :

- Utilisation des ressources suivie et mesurée précisément.
- Facturation basée sur la consommation réelle.

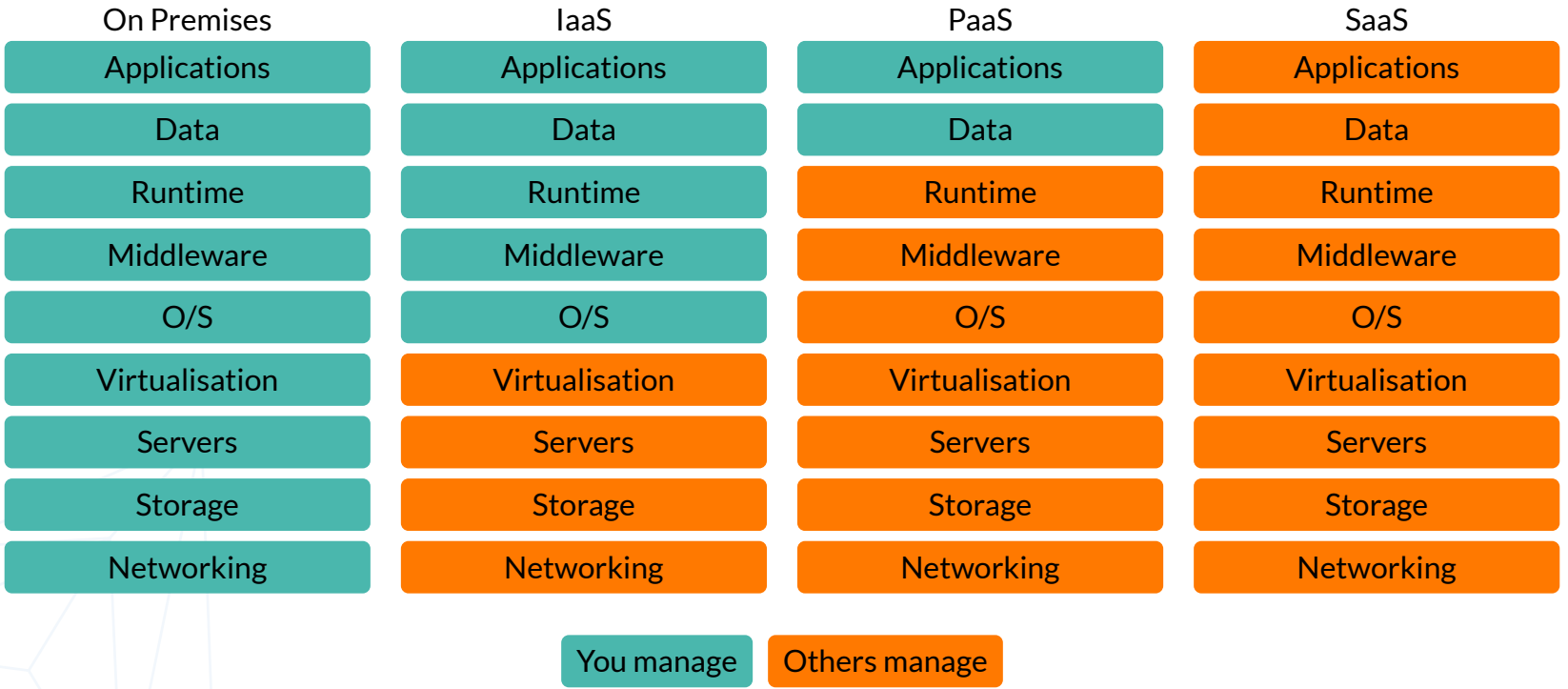
Modèle de Service Cloud : IaaS, PaaS, SaaS

Caractéristique	Infrastructure as a Service (IaaS)	Platform as a Service (Paas)	Software as a Service (SaaS)
Description	Blocs de construction d'infrastructure IT virtualisée (calcul, stockage, réseau) à la demande.	Plateforme pour développer, déployer et gérer des applications sans gérer l'infrastructure.	Logiciel complet prêt à l'emploi, accessible via Internet sur abonnement.
Ce que vous gérez	Applications, Données, Middleware, Système d'exploitation.	Applications, Données.	Rien.
Ce que le fournisseur gère	Virtualisation, Serveurs, Stockage, Réseau, Datacenter physique.	Tout ce que gère l'IaaS + Middleware, Système d'exploitation, Runtime.	Tout.
Niveau de contrôle	Élevé (sur l'OS et au-dessus).	Moyen (sur les applications et données).	Faible (limité aux fonctionnalités de l'application).

Modèle de Service Cloud : IaaS, PaaS, SaaS

Caractéristique	Infrastructure as a Service (IaaS)	Platform as a Service (Paas)	Software as a Service (SaaS)
Flexibilité	Très élevée (contrôle de l'infrastructure).	Moyenne (contraintes liées à la plateforme).	Faible (limité par les fonctionnalités du logiciel).
Cible utilisateur	Administrateurs système, Ingénieurs infrastructure, Architectes Cloud.	Développeurs d'applications, Équipes DevOps	Utilisateurs finaux, Entreprises (pour des fonctions métier spécifiques).
Exemples	AWS EC2, Azure Virtual Machines, Google Compute Engine, OVHcloud Public Cloud.	AWS Elastic Beanstalk, Azure App Service, Google App Engine, Heroku.	Microsoft 365, Google Workspace, Salesforce, Slack, Dropbox.

Récapitulatif



Focus IaaS : Pourquoi choisir une infrastructure mutalisée ?

Focus IaaS : Pourquoi choisir une infrastructure mutalisée ?

- **Modèle financier OpEx :**
  - Évite les dépenses d'investissement initiales (CapEx).
  - Coûts basés sur l'utilisation (dépenses opérationnelles - OpEx).

Focus IaaS : Pourquoi choisir une infrastructure mutalisée ?

- **Modèle financier OpEx :**
  - Évite les dépenses d'investissement initiales (CapEx).
  - Coûts basés sur l'utilisation (dépenses opérationnelles - OpEx).
- **Flexibilité et Scalabilité :**
  - Ajustement rapide des ressources (CPU, RAM, stockage) à la hausse ou à la baisse.
  - Adapte l'infrastructure aux besoins fluctuants.

Focus IaaS : Pourquoi choisir une infrastructure mutalisée ?

- **Modèle financier OpEx :**
  - Évite les dépenses d'investissement initiales (CapEx).
  - Coûts basés sur l'utilisation (dépenses opérationnelles - OpEx).
- **Flexibilité et Scalabilité :**
  - Ajustement rapide des ressources (CPU, RAM, stockage) à la hausse ou à la baisse.
  - Adapte l'infrastructure aux besoins fluctuants.
- **Déploiement rapide :**
  - Provisionnement d'infrastructure en minutes/heures (vs jours/semaines on-premises).
  - Accélère la mise sur le marché et l'innovation.

Focus IaaS : Pourquoi choisir une infrastructure mutalisée ?

- **Modèle financier OpEx :**
  - Évite les dépenses d'investissement initiales (CapEx).
  - Coûts basés sur l'utilisation (dépenses opérationnelles - OpEx).
- **Flexibilité et Scalabilité :**
  - Ajustement rapide des ressources (CPU, RAM, stockage) à la hausse ou à la baisse.
  - Adapte l'infrastructure aux besoins fluctuants.
- **Déploiement rapide :**
  - Provisionnement d'infrastructure en minutes/heures (vs jours/semaines on-premises).
  - Accélère la mise sur le marché et l'innovation.
- **Accès à une Infrastructure Globale :**
  - Accès aux datacenters mondiaux du fournisseur.
  - Permet de déployer au plus près des utilisateurs (faible latence).

Focus IaaS : Pourquoi choisir une infrastructure mutalisée ?

- **Modèle financier OpEx :**
  - Évite les dépenses d'investissement initiales (CapEx).
  - Coûts basés sur l'utilisation (dépenses opérationnelles - OpEx).
- **Flexibilité et Scalabilité :**
  - Ajustement rapide des ressources (CPU, RAM, stockage) à la hausse ou à la baisse.
  - Adapte l'infrastructure aux besoins fluctuants.
- **Déploiement rapide :**
  - Provisionnement d'infrastructure en minutes/heures (vs jours/semaines on-premises).
  - Accélère la mise sur le marché et l'innovation.
- **Accès à une Infrastructure Globale :**
  - Accès aux datacenters mondiaux du fournisseur.
  - Permet de déployer au plus près des utilisateurs (faible latence).

- **Réduction de la Charge de Maintenance :**
  - Le fournisseur gère le matériel physique, les pannes, les mises à niveau, le datacenter.
  - Libère les équipes IT pour des tâches à plus forte valeur ajoutée.

Focus IaaS : Pourquoi choisir une infrastructure mutalisée ?

- **Modèle financier OpEx :**
  - Évite les dépenses d'investissement initiales (CapEx).
  - Coûts basés sur l'utilisation (dépenses opérationnelles - OpEx).
- **Flexibilité et Scalabilité :**
  - Ajustement rapide des ressources (CPU, RAM, stockage) à la hausse ou à la baisse.
  - Adapte l'infrastructure aux besoins fluctuants.
- **Déploiement rapide :**
  - Provisionnement d'infrastructure en minutes/heures (vs jours/semaines on-premises).
  - Accélère la mise sur le marché et l'innovation.
- **Accès à une Infrastructure Globale :**
  - Accès aux datacenters mondiaux du fournisseur.
  - Permet de déployer au plus près des utilisateurs (faible latence).

- **Réduction de la Charge de Maintenance :**
  - Le fournisseur gère le matériel physique, les pannes, les mises à niveau, le datacenter.
  - Libère les équipes IT pour des tâches à plus forte valeur ajoutée.
- **Contrôle sur l'Environnement :**
  - Contrôle direct sur l'OS, les logiciels installés, la configuration réseau.
  - Idéal pour applications spécifiques ou migrations "legacy".

Cas typiques d'utilisation de l'IaaS

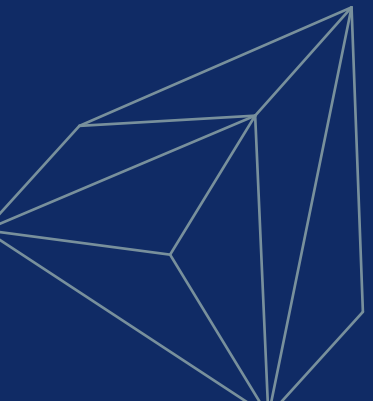
- Hébergement de sites web et d'applications : Des blogs personnels aux applications d'entreprise complexes.
- Environnements de Développement et de Test : Création rapide d'environnements isolés et jetables pour coder, tester et valider des logiciels.
- Stockage, Sauvegarde et Reprise après Sinistre (Disaster Recovery) : Utilisation du stockage cloud (souvent objet) pour des sauvegardes externalisées et mise en place de plans de reprise d'activité en cas de sinistre sur le site principal.
- Calcul Haute Performance (HPC) : Accès temporaire à une grande puissance de calcul pour des simulations scientifiques, de la modélisation financière, du rendu graphique, etc..

- Analyse de Big Data : Traitement et analyse de très grands volumes de données grâce à la scalabilité du calcul et du stockage cloud.
- Migration d'applications existantes ("Lift-and-Shift") : Déplacer des applications fonctionnant sur des serveurs physiques ou virtuels on-premises vers des VMs dans le cloud avec un minimum de modifications.

Comparaison simplifiée : On-Premises vs Cloud

	On-Primises	Cloud IaaS
Avantages	– Contrôle total sur l'infrastructure et les données ; – Sécurité perçue comme maîtrisée (car interne).	– Faibles coûts initiaux (OpEx dominant) ; – Grande flexibilité et scalabilité à la demande – Déploiement rapide ; – Réduction de la charge de gestion de l'infra physique.
Inconvénients	– Coûts initiaux (CapEx) très élevés ; – Rigidité et lenteur pour s'adapter aux changements ; – Responsabilité totale de la gestion, maintenance et sécurité, risque de sur/sous-provisionnement.	– Dépendance vis-à-vis du fournisseur et de la connectivité Internet ; – Responsabilité partagée de la sécurité (nécessite une bonne configuration par le client) ; – Potentiels coûts imprévus si la consommation n'est pas maîtrisée, moins de contrôle direct sur le matériel physique.

Composants fondamentaux d'IaaS - Compute & Stockage



Virtualisation

La virtualisation est une technologie permettant de créer des représentations virtuelles de ressources informatiques physiques. Plutôt que d'avoir un système d'exploitation directement installé sur un matériel serveur dédié, la virtualisation permet d'exécuter plusieurs systèmes d'exploitation et applications isolés les uns des autres sur une seule machine physique.

Virtualisation : Rôle de l'Hyperviseur

Au cœur de la virtualisation se trouve l'hyperviseur, également appelé Moniteur de Machine Virtuelle (VMM - Virtual Machine Monitor). Il s'agit d'une couche logicielle (ou parfois matérielle/firmware) qui crée, gère et alloue les ressources matérielles physiques (CPU, mémoire, stockage, réseau) aux différentes machines virtuelles. L'hyperviseur est responsable de l'abstraction du matériel sous-jacent, permettant aux VMs de fonctionner comme si elles disposaient de leur propre matériel dédié.

Les deux types d'Hyperviseur

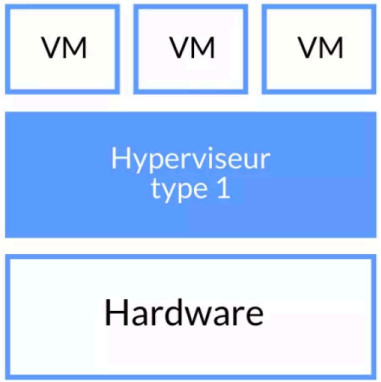


Figure 1: Représentation d'un hyperviseur de Type I.

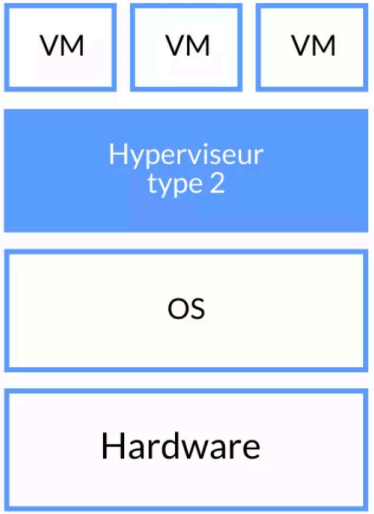


Figure 2: Réprésentation d'un hyperviseur de Type II.

Tableau comparatif : Hyperviseurs Type I et Type II

Caractéristique	Hyperviseur Type I (Baremetal)	Hyperviseur Type II (Hébergé)
Architecture	S'exécute directement sur le matériel physique	S'exécute sur un Système d'Exploitation (OS) hôte
Performance	Élevée (accès direct au matériel)	Moindre (passe par l'OS hôte)
Gestion	Complexe (niveau administrateur système requis)	Simple (comme une application standard)
Isolation	Forte (pas de couche OS hôte partagée)	Moins forte (dépend de l'isolation de l'OS hôte)
Cas d'usage	Datacenters, Cloud, Production, Serveurs critiques	Postes de travail, Développement, Tests, Multi-OS
Exemples	VMware ESXi, Hyper-V, KVM, Xen	VirtualBox, VMware Workstation/Fusion, Parallels

Machines Virtuelles (VMs)

Principe

La Machine Virtuelle (VM) est l'unité de calcul fondamentale dans un environnement IaaS. C'est une émulation logicielle complète d'un ordinateur physique, incluant un CPU virtuel, de la RAM virtuelle, un stockage virtuel (disque dur) et une interface réseau virtuelle. Chaque VM fonctionne comme un système invité isolé sur l'hyperviseur, capable d'exécuter son propre système d'exploitation (Windows, Linux…) et ses applications, indépendamment des autres VMs partageant le même matériel physique.

Machines Virtuelles (VMs) (ii)

Cycle de Vie

Le **cycle de vie** général d'une VM comprend les phases suivantes :

- **Création / Provisionnement / Préparation :**
  - Allocation des ressources (CPU, RAM, diques) ;
  - Préparation pour le premier démarrage ;
  - VM non encore exécutée.
- **En Cours d'Exécution (Running) :**
  - VM démarée, OS actif ;
  - Opérationnelle, traîte les requêtes ;
  - *Facturation des ressources de calcul active.*

Machines Virtuelles (VMs) (iii)

Cycle de Vie

- **Arrêtée (Stopped / Terminated [GCP[1]]) :**
  - VM et OS arrêtés ;
  - État RAM perdu ;
  - Peut être redémarrée ;
  - *Facturation stockage persistant & IP statiques continue.*

[1] Google Cloud Platform

Machines Virtuelles (VMs) (iv)

Cycle de Vie

- **Suspendue (Suspended [GCP]) :**
  - État de veille (RAM & CPU sauvegardés sur disque) ;
  - Reprise rapide à l'état exact prédédent ;
  - *Facturation stockage persistant & IP statiques continue.*

Machines Virtuelles (VMs) (v)

Cycle de Vie

- **Supprimée (Terminated [AWS[2]]): :**
  - Destruction définitive de la VM et des ressources associées (sauf exceptions) ;
  - Irrécupérable ;
  - Fin de la facturation pour cette VM.

[2] Amazon Web Services

Machines Virtuelles (VMs) (vi)

Cycle de Vie

- **Autres états transitoires :**
  - *Stopping/Suspending :* Transition vers l'arrêt/suspension.
  - *Repairing [GCP]:* Maintenance par le fournisseur (facturation souvent suspendue).

Machines Virtuelles (VMs) (vii)

Images (Templates)

Qu'est ce qu'une image ?

Une "copie maître" contenant un OS (et potentiellement logiciels/paramètres). Base pour démarrer de nouvelles VMs avec une configuration prédéfinie.

Machines Virtuelles (VMs) (viii)

Images (Templates)

Types d'Images

- Publiques
  - Fournies par le fournisseur cloud (AWS, Azure, GCP…) ou partenaires.
  - OS populaires (Linux, Windows Server) avec configuration de base.
- Personnalisées
  - Créées par l'utilisateur à partir d'une VM configurée (applis, sécurité…).
  - Nécessite souvent de "généraliser" la VM source (outils : Sysprep, waagent) pour supprimer les identifiants uniques.

Machines Virtuelles (VMs) (ix)

Images (Templates)

Avantages des Images Personnalisées :

- Standardisation des déploiements.
- Cohérence entre les environnements (dev, test, prod).
- Accélération de la mise en service de nouvelles instances.

Formats & Fonctionnalités Avancées :

- Standards comme OVA/OVF pour import/export inter-plateformes.
- Fonctionnalités spécifiques (ex: GCP Machine Images pour capture multi-disques).

Machines Virtuelles (VMs) (x)

Instances

Une Instance est simplement une VM spécifique qui a été créée (instanciée) à partir d'une image et qui est en cours d'exécution (ou dans un autre état de son cycle de vie).

Stockage IaaS : Distinction Stockage Bloc et Sockage Objet

Après le Calcul (VMs), le Stockage est l'autre service essentiel de l'IaaS. Les fournisseurs cloud offrent divers types de stockage pour répondre à différents besoins :

- Performance
- Mode d'accès
- Coût
- Cas d'usage

Deux paradigmes principaux : **Stockage Bloc** et **Stockage Objet**.

Stockage IaaS : Distinction Stockage Bloc et Sockage Objet

Stockage par blocs : principes & accès

Principe :

- Fournit des volumes (disques virtuels bruts) attachés à une VM.
- Similaire à un disque dur physique (HDD/SSD) local.
- Données organisées en blocs de taille fixe avec adresses.

Accès :

- Volume généralement attaché à une seule VM à la fois.
- L'OS de la VM crée un système de fichiers (NTFS, ext4…) sur le volume.
- Accès via les commandes standards du système de fichiers (comme un disque local).

Stockage IaaS : Distinction Stockage Bloc et Sockage Objet

Stockage par blocs : Performance & Cas d'usage

Performance :

- Optimisé pour faible latence et IOPS (Opérations Entrée/Sortie par seconde) élevés.
- Idéal pour lectures/écritures fréquentes et rapides.
- Varie selon type (SSD > HDD) et options (IOPS garantis).

Cas d'Usage Typiques :

- Disques de démarrage (OS) des VMs.
- Bases de données transactionnelles (SQL, NoSQL).
- Applications nécessitant accès via système de fichiers monté.
- Charges de travail exigeant des performances disque élevées.

Exemples :

- AWS EBS, Azure Managed Disks, GCP Persistent Disks.

Stockage IaaS : Distinction Stockage Bloc et Sockage Objet

Stockage Objet : Principe & Accès

Principe :

- Données stockées en "objets" = données + métadonnées + ID unique.
- Structure d'adressage plate dans un "bucket" (ou "conteneur").
- Pas de hiérarchie de dossiers/fichiers native.

Accès :

- Principalement via API RESTful (requêtes HTTP/HTTPS).
- Non monté comme un disque local sur une VM.
- Accessible via réseau (Internet/privé) avec son ID unique (si autorisé).

Stockage IaaS : Distinction Stockage Bloc et Sockage Objet

Stockage Objet : Performance & Cas d'Usage

Performance :

- Optimisé pour haute durabilité, disponibilité et débit (throughput).
- Scalabilité quasi illimitée (capacité totale, nombre d'objets).
- Latence pour un objet individuel généralement plus élevée que le stockage bloc.

Cas d'Usage Typiques :

- Stockage de fichiers statiques volumineux (images, vidéos, docs).
- Sauvegardes et Archivage long terme.
- Logs applicatifs / système.
- Distribution de contenu web statique.
- Data Lakes (Big Data).

Exemples :

- AWS S3, Azure Blob Storage, Google Cloud Storage (GCS).

Stockage IaaS : Tableau comparatif

Caractéristique	Stockage Bloc	Stockage Objet
Unité	Bloc sur volume (disque virtuel)	Objets (données + méta + ID) dans bucket/conteneur
Structure	Hiérarchique (système de fichiers)	Plate (pas de hiérarchie)
Accès	OS (disque local)	API RESTfull (HTTP/HTTPs)
Perf. clé	Faible latence, IOPS élevés	Haut débit, Haute Durabilité
Scalabilité	Limitée par taille volume	Quasi illimitée (capacité, nb objets)
Usage	OS, BDD transactionnelles, applis	Médias, Backups, Archives, Web, Data Lakes

Stockage IaaS : Importance du Bon Choix de Stockage

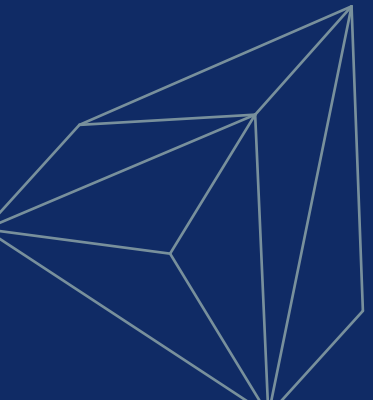
Le choix (Bloc vs Objet) impacte directement :

- L'architecture de l'application ;
- Les coûts de l'infrastructure ;
- Les performances.

Utiliser le mauvais type de stockage peut entraîner :

- Des dépenses inutiles (ex: Bloc cher pour les archives) ;
- Des performances catastrophiques (ex: Objet lent pour une BDD intensive).

Réseaux IaaS



Réseau IaaS : Le Troisième Pilier

Après le calcul et le stockage, le **réseau** est essentiel en IaaS.

Les fournisseurs de cloud offrent des **capacités de mise en réseau virtualisées** pour :

- Connecter les ressources (VMs, stockage, etc.) entre elles.
- Connecter les ressources avec l'extérieur (Internet, réseaux on-premises).
- Assurer la **sécurité** et l'**isolation**.

Concepts Clés du Réseau IaaS

- Réseau Virtuel Privé (VPC/VNet)
- Sous-réseaux (Subnets)
- Addressage IP (privé vs public)
- Notation CIDR

Réseau Privé Virtuel (VPC/VNet)

- Cœur de la mise en réseau IaaS.
- Réseau logique, défini par logiciel.
- Isolé des réseaux des autres clients sur l'infrastructure physique partagée.
- La "tranche privée" du cloud public.
- Contrôle granulaire :
  - Définir votre espace d'adressage IP privé(plage CIDR) ;
  - Créer des sous-réseaux ;
  - Configurer le routage ;
  - Génèer la connectivité externe (passerelles).
- L'isolation logique est la base de la sécurité dans le cloud.
- Peut être régional (AWS) ou global (GCP).

Sous-Réseaux (Subnets)

- Un VPC est divisé en un ou plusieurs **sous-réseaux** ;
- Chaque sous-réseau = une **plage d'adresses IP spécifique** dans le VPC.
- Permet d'**organiser logiquement** les ressources (VMs, etc.).
  - Ex: sous-réseaux distincs pour serveurs web, app, bases de données.
- Élément clé pour la **sécurité** (ACLs réseau) et le **routage** ;
- **Confiné à une seule Zone de Disponibilité (AZ)** ches la plupart des fournisseurs (ex: AWS).
  - **AZ** = centre de données physiquement distinct ;
  - Répartir les ressources dans différentes AZs => architecture résiliente.
- Chez GCP, les sous-réseaux sont **régionaux**, instances dans une zone spécifique.
- Les ressources sont **toujours** lancées à **l'intérieur d'un sous-réseau**.
- Distinction courante : **sous-réseaux publics** (route vers Internet) vs **privés** (pas d'accès direct à Internet).

Adressage IP : Privé vs Public

Adresses IP Privées :

- Pour la **communication interne au VPC** ;
- Non routables sur l'Internet public ;
- Attributées aux instances lancées dans un sous-réseau ;
- Soucent issues des plages RGC 1918 (`10.0.0.0/24`, `172.16.0.0/12`, `192.168.0.0/16`).

Adresses IP Publiques :

- Pour la **communication avec Internet** ;
- Routables sur **Internet** ;
- Attribuées de deux manières :

- **Dynamiquement (éphémère) :** : Attributée au lancement dans un sous-réseau public, temporaire ;
- **Statiquement (Réservée/Élastique) :** Allouée et associée manuellement, fixe, persiste à l'arrêt, peut être réassociée. Utile pour services à adresse stable (serveur web).

Adressage IP & Notation CIDR

- Lien IP privée / IP publique géré par NAT (Network Address Translation) à la passerelle Internet.
- Support IPv6 également courant (adresses globalement uniques).
- Notation CIDR (Classless Inter-Domain Routing) :
  - Ex: `10.0.0.0/16`
  - Permet de définir une plage d'adresses IP et son masque de sous-réseau.
  - Le nombre après le / = nombre de bits fixes pour la partie réseau.
  - Détermine la taille de la plage d'adresses pour les hôtes.

Conception Initiale du Réseau

- Étape architecturale critique (sécurité, scalabilité, connectivité).
- Choix judicieux des plages CIDR pour éviter les conflits futurs (connexion à d'autres réseaux).
- Segmentation en sous-réseaux (par couche applicative, public/privé) = pratique fondamentale de sécurité.
- Répartition stratégique des sous-réseaux entre différentes AZs = indispensable pour la haute disponibilité.
- L'isolation du VPC est la 1ère barrière, mais dépend d'une configuration réfléchie et correcte.

Connectivité : Passerelles Internet & Routage

Comment le trafic entre et sort-il du réseau virtuel ?

Connectivité : Passerelles Internet & Routage

Deux composantes clés :

- Passerelle Internet (Internet Gateway - IGW)
- Tables de Routage

Passerelle Internet (Internet Gateway - IGW)

- Composant géré par le fournisseur (hautement disponible, scalable).
- **Attachée à un VPC** pour la communication bidirectionnelle entre les **sous-réseaux publics** et l'Internet public.
- Sans IGW attachée et configurée, un VPC est isolé d'Internet.
- Sert de **cible dans les tables de routage** pour le trafic destiné à l'extérieur du VPC.
- Effectue le **NAT** pour le trafic sortant initié depuis une instance privée (utilise l'IP publique/élastique).
- Effectue le **NAT inverse** pour le trafic entrant destiné à une IP publique.

Tables de Routage

- Ensemble de règles (**routes**) qui déterminent la **destination du trafic quittant un sous-réseau ou passant par une passerelle**.
- Chaque sous-réseau doit être associé à une table de routage (sinon, table par défaut du VPC).
- Contient typiquement :
  - Une **route locale implicite** : communication interne au VPC (via IPs privées).
  - Des **routes explicites** : pour communiquer avec l'extérieur.
    - Spécifient une **destination** (plage CIDR) et une **cible** ("next hop").
    - Ex: Destination `0.0.0.0/0` ou `::/0` (tout trafic inconnu) avec pour cible l'**IGW** pour un sous-réseau public.
  - Autres cibles possibles : passerelle VPN/Direct Connect, passerelle d'appairage VPC, passerelle NAT, interface réseau spécifique.

Tables de Routage (ii)

- En cas de routes multiples pour une destination, la **route la plus spécifique** est utilisée (masque de sous-réseau le plus long).
- Si aucune route ne correspond, le trafic est généralement **abandonné ("blackholed")**.
- La connectivité n'est pas automatique : les **tables de routage définissent explicitement les chemins autorisés**.
- Une VM dans un sous-réseau privé sans route appropriée (vers IGW ou NAT Gateway) ne pourra pas initier de connexions Internet.
- Une configuration incorrecte peut soit **isoler** des ressources qui devraient communiquer, soit créer des chemins **non souhaités** (sécurité).
- Le routage est un outil fondamental pour la **segmentation logique** et le **contrôle précis des flux de données**.

Sécurité Réseau : Groupes de Sécurité / NSG / Pare-feu

Au-delà du routage, il faut contrôler quel trafic est autorisé à entrer/sortir d'une ressource spécifique (VM).

- Rôle des Groupes de Sécurité (SG - AWS/GCP) ou Network Security Groups (NSG - Azure).
- Agissent comme des pare-feux virtuels.
- Associés aux interfaces réseau (NICs) des VMs (ou sous-réseau pour NSG Azure).

Groupe de Sécurité / NSG : Rôle et Fonctionnement

- Fonction principale : **Filtrer le trafic** au niveau de l'instance/sous-réseau.
- N'autorisent que les communications **explicitement permises**.
- Mécanisme de sécurité fondamental en IaaS, granularité fine.
- Fonctionnement **Stateful (à état)** :
  - Le pare-feu suit les connexions actives.
  - Si le trafic sortant est autorisé, le trafic de réponse correspondant (entrant) est **automatiquement autorisé à revenir**.
  - Simplifie la configuration (pas besoin de règles symétriques pour le trafic de retour), contrairement aux pare-feux Stateless (Network ACLs AWS).

Groupe de Sécurité / NSG : Règles

- Configuration via des **règles d'autorisation**. Tout trafic non autorisé est **refusé par défaut**.
- Chaque règle définit :
  - **Direction :** Entrante (Inbound/Ingress) ou Sortante (Outbound/Egress).
  - **Protocole :** TCP, UDP, ICMP, etc.
  - **Plage de Ports :** Port spécifique ou plage.
  - **Source** (entrant) / **Destination** (sortant) :
    - IP spécifique, Plage CIDR (`0.0.0.0/0` = Internet).
    - **Un autre groupe de sécurité/NSG** (communication entre groupes de VMs).
    - Service Tags (Azure), Listes de Préfixes (AWS).
  - **Action :** Généralement "Autoriser" (Allow). NSG Azure supporte aussi "Refuser" (Deny) explicite.

- **Priorité** (NSG Azure) : Détermine l'ordre d'évaluation (bas = haute priorité). SG AWS/GCP : l'ordre n'importe pas, une règle d'autorisation suffit.

Groupe de Sécurité / NSG : Application & Bonnes Pratiques

Application :

- Directement au niveau de la **NIC de chaque VM**.
- Une VM peut être associée à un ou plusieurs groupes.
- NSG Azure peut aussi être associé à un **sous-réseau** (double filtrage si NSG aussi sur la NIC).

Règles par Défaut :

Souvent tout trafic sortant autorisé, tout trafic entrant non autorisé refusé. **Crucial de les vérifier et adapter.**

- Permettent la **micro-segmentation** au niveau de chaque instance.
- Outil principal pour implémenter le **principe du moindre privilège** : n'autoriser que les flux strictement nécessaires.

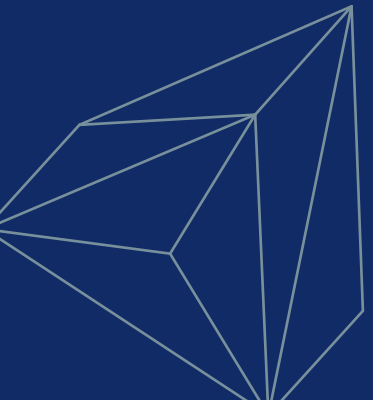
- Ex: serveur web : entrant 80/443 (Internet), 22/3389 (plage IP admin spécifique).

Bonne pratique :

Être aussi **spécifique que possible** dans les règles (ports exacts, sources/destinations précises).

- Essentiels pour **isoler les VMs** (même dans le même sous-réseau) et **protéger chaque composant**.

Introduction à la plannification



Comment déterminer les ressources cloud nécessaires ?

Principes d'évaluation des besoins en ressources

Planification de la capacité ("Capacity Planning") / Dimensionnement ("Sizing")

- Déterminer la quantité de ressources (CPU, RAM, Stockage, Réseau) nécessaire pour une application.
- **Importance double :**
  - **Assurer la Performance :** Éviter le sous-dimensionnement (lenteur, pannes).
  - **Optimiser les Coûts :** Éviter le surdimensionnement (capacité inutilisée, gaspillage en "pay-as-you-go").

Ressources Clés à Évaluer

- **CPU (Processeur) :** Puissance de calcul. Mesurée en vCPU, fréquence. Dépend de l'intensité de traitement.
- **RAM (Mémoire Vive) :** Espace mémoire. Mesurée en Go. Dépend des données en mémoire, utilisateurs simultanés, efficacité de l'app.
- **Stockage :**
  - **Capacité :** Quantité de données (Go/To).
  - **Performance :** Vitesse d'accès (IOPS, débit Mo/s).
  - Choix type (Bloc vs Objet) et caractéristiques (SSD vs HDD, IOPS provisionnés).
- **Réseau (Bande Passante / Débit) :** Quantité de données qui transite (Mbps/Gbps) entrant/sortant. Coûts de transfert sortant ("egress") importants.

Comment Estimer les Besoins ?

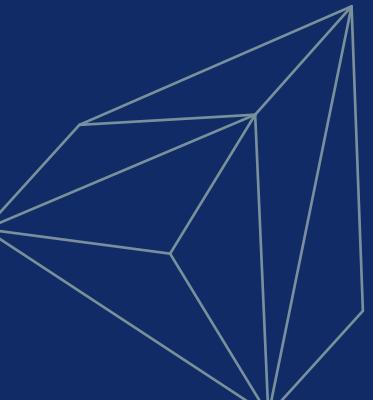
Méthodes combinées :

- **Analyse des Exigences Applicatives :** Comprendre le fonctionnement, les opérations coûteuses, les dépendances, le profil utilisateur.
- **Analyse Historique et Monitoring :** Si l'app existe, analyser les données de conso historiques (moyenne, **pics**). Outils de monitoring cloud essentiels.
- **Benchmarking :** Tester l'app sur différentes configs d'infra en simulant la charge. Établir des **baselines**, identifier les **goulots d'étranglement**, comprendre le **scaling**.
- **Prévisions de Charge (Forecasting) :** Anticiper l'évolution future de la demande (croissance utilisateurs, données, nouvelles features). Souvent difficile mais nécessaire.
- **Déploiement Pilote :** Déployer pour un groupe restreint ou charge limitée pour valider les hypothèses en environnement réel.

Plannification dans le Cloud : Itération et Optimisation

- **Spécificité du cloud : l'élasticité.**
- Moins de surprovisionnement massif à long terme comme on-premises.
- Approche plus **dynamique et itérative**.
- Commencer avec une estimation raisonnable, puis **monitorer** la performance et conso réelles.
- **Ajuster la taille des ressources ("right-sizing")** continuellement.
- Utilisation de mécanismes d'**auto-scaling** pour adapter automatiquement la capacité à la demande en temps réel.
- La planification devient un **processus continu d'optimisation** basé sur les données de monitoring, benchmarking, et automatisation.
- Objectif : Trouver le meilleur **équilibre** entre la **performance applicative** et le **coût** de l'infrastructure cloud.

Présentation de l’environnement TP



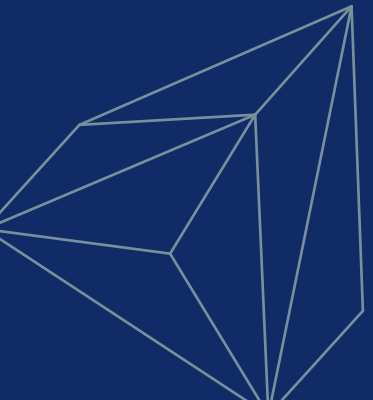
Présentation de la plateforme utilisée en TP et modalités d'accès/installation

(Cette diapositive est un placeholder. Le contenu doit être adapté à la plateforme spécifique utilisée pour le cours.)

- Plateforme utilisée : ScaleWay
- Permet de créer et gérer :
  - Machines Virtuelles (Compute)
  - Réseaux Virtuels (VPC/VNet, Sous-réseaux, Routage, Sécurité)
  - Stockage (Volumes, Buckets)
  - Règles de sécurité (Groupes de Sécurité/NSG)
- Modalités d'accès/installation :

  - [Instructions spécifiques : console web, CLI, compte fourni, environnement virtuel, etc.]
- **Action Requise :** Assurez-vous d'avoir accès à la plateforme avant la prochaine séance de TP. Les instructions détaillées seront fournies séparément.

Conclusion & Récapitulation



Fondations IaaS

- Contraste **On-Premises** (propriété, contrôle total, CapEx) vs **Cloud Computing** (location, gestion déléguée, flexibilité, OpEx).
- Modèles de service cloud : **IaaS**, **PaaS**, **SaaS** (niveaux de gestion).
- Composants fondamentaux de l'IaaS :
  - **Compute :** Hyperviseur (Type 1), VMs, images/templates.
  - **Stockage :** Bloc (OS, DB) vs Objet (fichiers, backups, médias).
  - **Réseau :**
    - VPC/VNet (isolation)
    - Sous-réseaux (organisation, HA)
    - Adressage IP (privé/public)
    - Passerelle Internet (connectivité externe)
    - Tables de Routage (diriger le trafic)
    - Groupes de Sécurité/NSG (pare-feux stateful, micro-segmentation)

Planification

- Importance de la **planification de la capacité (sizing)** pour équilibrer performance et coût.
- Ressources clés : CPU, RAM, Stockage, Réseau.
- Méthodes d'estimation : Analyse exigences/historique, monitoring, benchmarking, prévisions, déploiement pilote.
- Approche cloud : **Itérative et optimisée** grâce à l'élasticité, monitoring, auto-scaling.

À ce stage, vous devriez être capable de :

- Distinguer les caractéristiques fondamentales local vs. cloud.
- Différencier IaaS, PaaS, SaaS.
- Identifier et décrire les rôles des composants IaaS majeurs (VM, Stockage Bloc/Objet, Réseau Virtuel, Groupe de Sécurité).
- Comprendre les principes de base de l'estimation des besoins en ressources.
- Connaître l'environnement technique des TPs.