Principes de base Microsoft Azure

Cours complet sur le Cloud Computing et Microsoft Azure : présentation du Cloud (acteurs majeurs, IaaS/PaaS/SaaS, modèles public/privé/hybride), création et gestion de serveurs Azure (VM, App Service, Containers), gestion des données (stockage, bases SQL et NoSQL), réseaux virtuels (VNet, peering, VPN), montée en charge (Load Balancer, VMSS), sécurisation (Microsoft Entra ID, RBAC, MFA, Microsoft Defender for Cloud) et monitoring (Azure Monitor, Advisor). Aucun prérequis.

60 min de lecture **Niveau Intermédiaire**

Document généré le 11/07/2026 à 20h37 · nouvy.fr/wiki/principes-base-microsoft-azure

Sommaire

45 section(s) · 60 min de lecture

> ***Objectifs opérationnels*** *: créer et gérer des environnements virtuels, gérer les réseaux sous environnement Azure.*
> ***Objectifs d'apprentissage*** *: maîtriser le concept de Cloud Computing.*
> ***Prérequis*** *: aucun.*

Ce cours couvre l'ensemble des fondamentaux du Cloud Computing à travers la plateforme **Microsoft Azure**, depuis les concepts généraux (IaaS/PaaS/SaaS) jusqu'à la sécurisation et le monitoring d'une infrastructure cloud complète.

Sommaire du cours

---

1. Présentation du Cloud Computing

Qu'est-ce que le Cloud Computing ?

Le **Cloud Computing** (informatique en nuage) désigne la fourniture à la demande de ressources informatiques (serveurs, stockage, bases de données, réseaux, logiciels) via Internet, avec **paiement à l'usage**. Plutôt que de posséder une infrastructure physique dans un local technique, l'entreprise loue des ressources hébergées chez un fournisseur de cloud.

Les 5 caractéristiques du Cloud (selon le NIST)

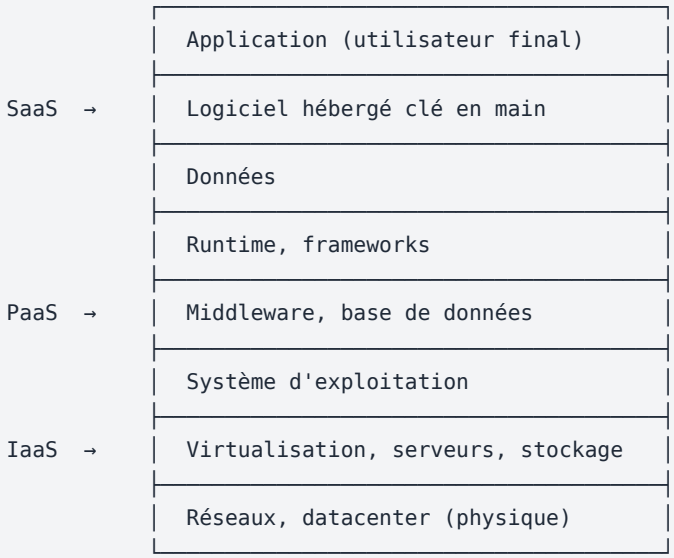
Caractéristique	Description
Libre-service à la demande	L'utilisateur provisionne les ressources sans intervention humaine du fournisseur
Accès réseau étendu	Disponible partout, depuis n'importe quel appareil connecté à Internet
Mise en commun des ressources	Le fournisseur mutualise ses ressources entre plusieurs clients (multi-tenant)
Élasticité rapide	Capacité d'augmenter/diminuer les ressources rapidement en fonction de la demande
Service mesuré	Facturation transparente basée sur la consommation réelle (CPU, Go, requêtes...)

Les acteurs majeurs du Cloud

Fournisseur	Solution	Parts de marché (estimées)	Spécificités
Amazon Web Services (AWS)	AWS	~31 %	Pionnier, catalogue le plus large, leader historique
Microsoft Azure	Azure	~25 %	Intégration native Microsoft 365, AD, hybride
Google Cloud Platform (GCP)	Google Cloud	~11 %	Excellence sur l'IA/ML, BigQuery, Kubernetes (GKE)
Alibaba Cloud	—	~4 %	Leader en Asie
OVHcloud	—	< 2 %	Cloud souverain européen, conforme RGPD
Scaleway	—	< 1 %	Cloud français, écologique
IBM Cloud, Oracle Cloud	—	< 5 % cumulés	Spécialisés enterprise legacy

> ***Pourquoi Azure pour ce cours ?*** *Microsoft Azure est très répandu en entreprise grâce à son intégration avec l'écosystème Microsoft (Active Directory, Office 365, Windows Server, .NET). C'est aussi le cloud public qui propose la meilleure approche* ***hybride*** *(cloud + on-premise).*

Les 3 types de services Cloud (modèles de service)



Copier

IaaS — Infrastructure as a Service

Le fournisseur cloud loue de l'**infrastructure brute** : machines virtuelles, stockage, réseaux. Le client gère tout le reste (OS, middleware, applications, données).

Critère	Description
Ce que le fournisseur gère	Datacenter, matériel, virtualisation, réseau physique
Ce que le client gère	OS, patches, middleware, runtime, données, applications
Exemples Azure	Azure Virtual Machines, Azure Virtual Network, Azure Storage
Cas d'usage	Migration "lift-and-shift", remplacement de serveurs physiques, environnements de test

PaaS — Platform as a Service

Le fournisseur fournit une **plateforme prête à recevoir du code** (runtime, base de données, middleware). Le client se concentre sur ses applications et données, sans gérer l'OS ni les patches.

Critère	Description
Ce que le fournisseur gère	Tout jusqu'au runtime/middleware
Ce que le client gère	Code applicatif et données
Exemples Azure	Azure App Service, Azure SQL Database, Azure Functions
Cas d'usage	Déploiement rapide d'applications web, API, microservices

SaaS — Software as a Service

Le fournisseur livre un **logiciel complet** accessible via navigateur, sans aucune installation. Le client est simple consommateur.

Critère	Description
Ce que le fournisseur gère	Tout, jusqu'à l'application
Ce que le client gère	Ses utilisateurs et ses données
Exemples Microsoft	Microsoft 365, Dynamics 365, Power BI, Teams
Exemples non-Microsoft	Salesforce, Slack, Dropbox, Gmail

Les 3 modèles de déploiement Cloud

Modèle	Description	Quand l'utiliser
Cloud public	Ressources mutualisées, accessibles par Internet (Azure, AWS, GCP)	Élasticité maximale, scénarios standards, pas de contrainte réglementaire forte
Cloud privé	Infrastructure dédiée à une seule organisation (Azure Stack chez le client, ou cloud OVH dédié)	Données très sensibles, conformité stricte, contrôle total
Cloud hybride	Combinaison cloud public + on-premise (ou public + privé), avec passerelle réseau	Modernisation progressive, bursting de capacité, données sensibles isolées + apps publiques

> ***Cas concret hybride** : une PME garde ses serveurs métier internes (ERP, AD) on-premise, mais déploie son site e-commerce et ses environnements de test sur Azure. Une **VPN Gateway** ou **ExpressRoute** relie les deux.*

Avantages et limites du Cloud

Avantages	Limites / risques
Pas d'investissement initial (CAPEX → OPEX)	Coûts variables et parfois imprévisibles
Élasticité : payer ce qu'on consomme	Dépendance au fournisseur (vendor lock-in)
Mise en service en quelques minutes	Connectivité Internet indispensable
Haute disponibilité par défaut (SLA 99,9 % et plus)	Conformité réglementaire (RGPD, hébergement EU)
Sécurité industrielle (certifications ISO 27001, SOC 2)	Données partagées sur infrastructure mutualisée
Innovation continue (services managés à jour)	Compétences cloud à acquérir/recruter

---

2. Créer et gérer des serveurs dans Azure

L'interface Azure Portal

L'accès à Azure se fait principalement via :

- **Azure Portal** : `https://portal.azure.com` — interface web graphique
- **Azure CLI** : `az` en ligne de commande (multiplate-forme)
- **Azure PowerShell** : module `Az` pour PowerShell
- **API REST** et **SDK** (Python, .NET, Java, Node.js...)

> ***Recommandation** : démarrer avec le portail pour visualiser, puis automatiser avec CLI ou PowerShell pour les déploiements répétitifs.*

Concepts de base à connaître avant de créer une VM

Concept	Rôle
Abonnement (*subscription*)	Conteneur de facturation — toutes les ressources créées sont rattachées à un abonnement
Groupe de ressources (*resource group*)	Conteneur logique regroupant des ressources liées (ex: une appli + sa BDD + son réseau). Sert à gérer les permissions et la suppression groupée
Région (*region*)	Localisation géographique du datacenter (France Central, West Europe, East US...)
Zone de disponibilité	3 datacenters indépendants dans une même région (haute disponibilité)
Tags	Étiquettes clé/valeur pour catégoriser les ressources (`environnement=prod`, `projet=site-web`)

Création d'une machine virtuelle Azure (depuis le portail)

1. Portail Azure → **Créer une ressource** → **Machine virtuelle**

2. Onglet **Informations de base** :
    - **Abonnement** : choisir l'abonnement de facturation (ex. *Azure for Students*)
    - **Groupe de ressources** : créer ou choisir (ex: `rg-nouvy-test`)
    - **Nom de la machine virtuelle** : `vm-web-01`
    - **Région** : `Poland Central` *(bon compromis pour un compte étudiant : quotas généreux, données EU, latence acceptable depuis la France)*
    - **Options de disponibilité** : `Aucune redondance d'infrastructure requise` *(lab/test — pour la prod : Zone de disponibilité)*
    - **Type de sécurité** : `Lancer des machines virtuelles approuvées` *(Trusted Launch VM — sécurité par défaut sur Azure 2024+, inclut Secure Boot et vTPM)*
    - **Image** : `Debian 13 "trixie" x64 Gen2` *(VM Linux légère, idéale pour un budget étudiant)*
    - **Architecture VM** : `x64`
    - **Taille** : `Standard_B2ats_v2` *(2 vCPU, 1 Go RAM, série B "burstable" — la SKU étudiante par excellence, ~5 €/mois en 24/7, ~1,50 €/mois avec auto-shutdown)*
    - **Compte administrateur** :
        - **Type d'authentification** : `Clé publique SSH` *(plus sécurisé que mot de passe)*
        - **Nom d'utilisateur** : `azureuser`
        - **Source de la clé publique SSH** : *Générer une nouvelle paire de clés* (Azure téléchargera le fichier `.pem`)
    - **Ports d'entrée publics** : *Autoriser les ports sélectionnés* → cocher **SSH (22)**

3. Onglet **Disques** :
    - **Type de disque OS** : `Standard SSD` *(meilleur rapport prix/perf pour un lab — éviter Premium SSD qui coûte 3× plus cher pour rien sur une B2ats_v2)*
    - **Disques de données** : aucun pour ce scénario

4. Onglet **Mise en réseau** :
    - **Réseau virtuel (VNet)** : créer ou choisir un VNet existant
    - **Sous-réseau** : ex. `192.168.1.0/24`
    - **Adresse IP publique** : créée automatiquement
    - **Groupe de sécurité réseau (NSG)** : règles de pare-feu

5. Onglets **Gestion**, **Surveillance**, **Avancé** : laisser les valeurs par défaut

6. **Vérifier + créer** → validation des paramètres → **Créer**

Durée du déploiement : environ 3 à 5 minutes.

> ***Pour une VM Windows Server 2025 à la place** : à l'onglet **Informations de base**, changer **Image** pour `Windows Server 2025 Datacenter`, choisir une **Taille** d'au moins 4 Go de RAM (`Standard_B2s`), et passer le **Type d'authentification** à `Mot de passe` (Windows ne prend pas en charge l'authentification SSH par défaut). Conserver le **Type de sécurité** `Lancer des machines virtuelles approuvées` (Trusted Launch).*

Connexion à une VM

VM Windows — RDP

1. Portail Azure → VM → bouton **Se connecter** → **RDP**
2. Télécharger le fichier `.rdp` pré-configuré
3. Ouvrir le fichier → saisir le compte admin → connexion

VM Linux — SSH

1. Portail Azure → **Machines virtuelles** → ouvrir la VM concernée
2. Dans la **Vue d'ensemble**, copier l'**Adresse IP publique** affichée à droite
3. Depuis un terminal local (PowerShell, Terminal macOS/Linux), se connecter avec la clé SSH téléchargée lors de la création :

```
ssh -i ~/.ssh/azure_key.pem azureuser@<IP-publique>
```

Copier

> *Sur Windows, la clé `.pem` doit être placée dans `C:\Users\<vous>\.ssh\`. PowerShell intègre OpenSSH depuis Windows 10/11 — la commande `ssh` fonctionne directement.*

Suppression d'une VM

> ***Attention** : supprimer une VM ne supprime pas automatiquement ses dépendances (disque OS, IP publique, NIC, NSG). Pour tout nettoyer d'un coup, supprimer **le groupe de ressources entier** :*

1. Portail Azure → **Groupes de ressources** → ouvrir `rg-nouvy-test`
2. Cliquer **Supprimer le groupe de ressources** (barre supérieure)
3. Saisir le nom du groupe pour confirmer → **Supprimer**

Toutes les ressources contenues (VM, disques, NIC, IP publique, NSG, VNet) sont supprimées en cascade.

> ***Astuce économies** : pour ne pas être facturé pendant les périodes d'inactivité, ouvrir la VM dans le portail → bouton **Arrêter** (barre supérieure). Le statut passe à Arrêté (libéré) — les disques restent facturés mais le **compute est gratuit**. Pour automatiser : VM → **Opérations** → **Arrêt automatique** → activer une heure quotidienne (ex. 19h00).*

Azure Marketplace

L'**Azure Marketplace** est un catalogue de plus de 20 000 images, applications et services prêts à l'emploi, publiés par Microsoft et des éditeurs tiers (Oracle, Red Hat, Cisco, SAP…).

Exemples de ressources disponibles :

- VM préconfigurées (LAMP, WordPress, Jenkins, GitLab, SQL Server)
- Solutions de sécurité (Fortinet, Palo Alto, Sophos)
- Bases de données managées (MongoDB Atlas, MariaDB)
- Solutions analytiques (Databricks, Tableau Server)

Accès : Portail Azure → **Créer une ressource** → barre de recherche.

Azure App Service — PaaS web

Azure App Service est le service PaaS de référence pour héberger des **applications web et des API** sans gérer d'OS ni de serveur.

Caractéristiques clés

Caractéristique	Détail
Langages supportés	.NET, Java, Node.js, Python, PHP, Ruby
Conteneurs	Possibilité de déployer des images Docker custom
CI/CD	Intégration native GitHub, Azure DevOps, Bitbucket
SSL/TLS	Certificat HTTPS gratuit via App Service Managed Certificate
Domaine personnalisé	Pointer son `monsite.fr` vers App Service
Slots de déploiement	Tester en staging avant de basculer en production (zero-downtime)
Auto-scaling	Mise à l'échelle automatique selon CPU/RAM/queue

Création d'une App Service via portail

1. Portail Azure → **Créer une ressource** → **Web App**
2. **Informations de base** :
    - **Nom** : `nouvy-site-web` (devient `https://nouvy-site-web.azurewebsites.net`)
    - **Publier** : `Code` (ou `Container`)
    - **Pile d'exécution** : `PHP 8.3`, `.NET 8`, `Node 20 LTS`…
    - **Région** : `France Central`
3. **Plan App Service** :
    - Niveau **F1 (Gratuit)** pour les tests (limité à 60 min CPU/jour)
    - Niveau **B1 (Basique)** pour démos/petits sites (~13 €/mois)
    - Niveau **P1v3 (Premium)** pour la prod
4. **Vérifier + créer**

Déploiement de code dans App Service

Option A — Depuis Visual Studio Code :

1. Installer l'extension **Azure App Service**
2. Clic droit sur le dossier du projet → **Deploy to Web App**
3. Sélectionner l'App Service → confirmer

Option B — Via GitHub Actions (CI/CD) :

1. Portail → App Service → **Centre de déploiement** → **GitHub**
2. Autoriser GitHub → choisir dépôt + branche
3. Azure crée automatiquement un workflow GitHub Actions
4. Chaque `git push` déclenche un déploiement

Option C — ZIP deploy depuis le portail :

1. Préparer une archive `.zip` du projet (build de production)
2. Portail → App Service `nouvy-site-web` → menu **Outils de développement** → **Centre de déploiement**
3. Onglet **Déploiement ZIP** → **Parcourir** → sélectionner le `.zip` → **Déployer**
4. Suivre la progression dans **Journaux** → page d'accueil de l'app rechargée après quelques secondes

Conteneurs dans Azure

Azure propose plusieurs services pour exécuter des **conteneurs Docker** :

Service	Cas d'usage
Azure Container Instances (ACI)	Conteneur unique ou petit groupe, démarré à la demande, pas d'orchestration
Azure App Service for Containers	Conteneur web simple avec scaling auto
Azure Kubernetes Service (AKS)	Cluster Kubernetes managé pour orchestrer des centaines de conteneurs
Azure Container Apps	Microservices serverless basés sur Kubernetes (sans gérer le cluster)

> ***Quand utiliser ACI vs AKS ?*** *ACI = conteneur ponctuel ou batch. AKS = production multi-conteneurs avec orchestration, montée en charge, mises à jour glissantes.*

---

3. Gérer les données sous Azure

Vue d'ensemble des services de stockage

Service	Type de données	Cas d'usage
Azure Storage Account — Blob	Objets non-structurés (fichiers, images, vidéos)	Sauvegardes, hébergement statique, archivage
Azure Storage Account — Files	Partages SMB/NFS	Lecteur réseau partagé entre VM ou postes
Azure Storage Account — Queue	Messages asynchrones	Communication entre microservices
Azure Storage Account — Table	Clé/valeur NoSQL bas de gamme	Logs, telemetrie simple
Azure Disks	Disques durs virtuels (VHD) attachés aux VM	Disques OS et données des VM
Azure SQL Database	Base relationnelle managée (basée sur SQL Server)	Applications transactionnelles classiques
Azure Database for MySQL/PostgreSQL	Bases relationnelles open source managées	Apps PHP/Python/Node.js
Azure Cosmos DB	Base NoSQL multimodèle (document, graphe, colonnes)	Apps mondiales, IoT, faible latence
Azure Data Lake Storage	Stockage massif optimisé Big Data	Data analytics, ML, BI

Storage Account — création

1. Portail → **Créer une ressource** → **Compte de stockage**
2. **Informations de base** :
    - **Nom** : `stnouvyprod01` *(uniquement minuscules + chiffres, unique mondialement)*
    - **Région** : `France Central`
    - **Performances** : `Standard` (HDD) ou `Premium` (SSD)
    - **Redondance** :
        - **LRS** (Locally Redundant) — 3 copies dans 1 datacenter
        - **ZRS** (Zone Redundant) — 3 copies dans 3 datacenters de la région
        - **GRS** (Geo Redundant) — réplication dans une région secondaire (recommandé en prod)
        - **RA-GRS** — GRS + lecture seule sur la région secondaire
3. Onglet **Avancé** : laisser par défaut (chiffrement activé)
4. **Vérifier + créer**

Blob Storage — utilisation depuis le portail

Le **Blob Storage** stocke des objets dans des **conteneurs** (équivalent dossiers).

Étape 1 — Créer un conteneur

1. Portail → Compte de stockage `stnouvyprod01` → menu **Stockage de données** → **Conteneurs**
2. Cliquer **+ Conteneur** (barre supérieure)
3. **Nom** : `documents`
4. **Niveau d'accès anonyme** : `Blob (accès en lecture anonyme pour les blobs uniquement)` *(pour un usage interne, garder `Privé`)*
5. **Créer**

Étape 2 — Uploader un fichier

1. Cliquer sur le conteneur `documents` → bouton **Charger** (barre supérieure)
2. Glisser-déposer le fichier (ex. `rapport-2026.pdf`) ou cliquer **Parcourir des fichiers**
3. Onglet **Avancé** *(optionnel)* : choisir le **Niveau d'accès** (Hot / Cool / Archive)
4. **Charger**

Étape 3 — Récupérer l'URL publique

1. Cliquer sur le fichier uploadé dans le conteneur
2. Copier le champ **URL** : `https://stnouvyprod01.blob.core.windows.net/documents/rapport-2026.pdf`

> ***Pour uploader en masse** : utiliser **Azure Storage Explorer** (application gratuite Windows/Mac/Linux) — interface drag-and-drop avec arborescence multi-comptes.*

Niveaux d'accès Blob

Tier	Coût stockage	Coût accès	Cas d'usage
Hot	Le plus cher	Le moins cher	Données fréquemment lues
Cool	Moins cher	Plus cher	Sauvegardes mensuelles, peu lues
Archive	Le moins cher	Le plus cher (réhydratation 1 à 15 h)	Conservation légale longue durée

Azure SQL Database

Service PaaS de base de données relationnelle, compatible SQL Server.

Modèles d'achat

Modèle	Description
DTU (Database Transaction Unit)	Tarif simple, ressources packagées (Basic, Standard, Premium)
vCore	Tarification par vCPU + mémoire, plus de flexibilité (recommandé en prod)
Serverless	Auto-pause / auto-resume, paiement à la seconde (idéal pour BD irrégulières)

Création via portail

1. Portail → **Créer une ressource** → **SQL Database**
2. Créer ou choisir un **serveur SQL logique** (FQDN du serveur, login admin, mot de passe)
3. **Calcul + stockage** : choisir un niveau (ex. `General Purpose - Serverless`)
4. **Mise en réseau** : autoriser l'IP du poste admin pour les tests
5. **Vérifier + créer**

Connexion depuis SSMS ou Visual Studio Code (extension MSSQL)

> ***Important** : **Azure Data Studio est retiré depuis février 2026**. Microsoft recommande désormais l'extension officielle **MSSQL pour Visual Studio Code**, qui reprend les principales fonctionnalités d'ADS (IntelliSense, exécution de requêtes, graphique de plans, gestion des connexions) avec une intégration native à VS Code et GitHub Copilot.*

Installer l'extension MSSQL dans Visual Studio Code :

1. Ouvrir Visual Studio Code → onglet **Extensions** *(Ctrl+Shift+X)*
2. Rechercher `mssql` → installer **SQL Server (mssql)** publiée par **Microsoft**
3. Une nouvelle icône **SQL Server** apparaît dans la barre d'activité (à gauche)
4. Cliquer **+ Add Connection** → choisir **Parameters** ou **Connection String**

Paramètres de connexion à Azure SQL :

Champ	Valeur
Serveur	`nouvy-sql-server.database.windows.net`
Authentification	`SQL Login` *(ou `Microsoft Entra ID – Universal with MFA` si fédéré)*
Login	`nouvyadmin`
Mot de passe	`******`
Base	`nouvy-db`
Encrypt	`True` (obligatoire sur Azure SQL)
Trust Server Certificate	`False` *(les certificats Azure sont signés par une CA publique)*

Une fois connecté : clic droit sur la base → **New Query** pour ouvrir un éditeur T-SQL avec IntelliSense.

> ***Sécurité** : par défaut, le pare-feu Azure SQL bloque toutes les IP. Il faut ajouter manuellement les IP autorisées : portail → serveur SQL → **Sécurité** → **Mise en réseau** → **Pare-feu et réseaux virtuels** → **+ Ajouter l'adresse IPv4 du client**.*

Azure Cosmos DB

Base NoSQL globale, multimodèle (document, graphe, key-value, colonnes) avec latence garantie < 10 ms et disponibilité 99,999 %.

API supportée	Compatibilité
NoSQL (natif)	Documents JSON
MongoDB	Apps existantes MongoDB sans modification
Cassandra	Apps existantes Cassandra
Gremlin	Bases de graphes
Table	Compatible Azure Table Storage

Cas d'usage typiques : applications globales (un déploiement, plusieurs régions), IoT, e-commerce mondial, jeux multijoueurs.

---

4. Gérer les réseaux sous Azure

Composants du réseau virtuel (VNet)

Composant	Rôle
Virtual Network (VNet)	Réseau privé isolé dans Azure (équivalent VLAN/sous-réseau on-premise)
Sous-réseau (Subnet)	Découpage logique d'un VNet
Network Security Group (NSG)	Pare-feu de niveau 4 (filtre IP/port) appliqué à un sous-réseau ou une NIC
Application Security Group (ASG)	Regroupement logique de NIC pour appliquer des règles NSG par "rôle"
Route Table	Routes personnalisées (par défaut, Azure route automatiquement)
Public IP	Adresse IP publique attachable à une NIC, Load Balancer, etc.
Network Interface (NIC)	Carte réseau virtuelle attachée à une VM

Création d'un VNet (depuis le portail)

1. Portail → **Créer une ressource** → rechercher **Réseau virtuel** → **Créer**
2. Onglet **Informations de base** :
    - **Abonnement** + **Groupe de ressources** : `rg-nouvy-test`
    - **Nom du réseau virtuel** : `vnet-nouvy`
    - **Région** : la même que les VM qui utiliseront ce VNet
3. Onglet **Adresses IP** :
    - **Espace d'adressage IPv4** : `192.168.0.0/16` *(plage CIDR du VNet — 65 536 adresses)*
    - **Sous-réseaux** : éditer le sous-réseau `default`
        - **Nom** : `subnet-web`

- **Plage d'adresses** : `192.168.1.0/24` *(256 adresses dont 5 réservées par Azure)*

4. Onglets **Sécurité** et **Étiquettes** : laisser par défaut
5. **Vérifier + créer** → **Créer**

Structure d'adressage proposée pour NOUVY :

```
VNet vnet-nouvy : 192.168.0.0/16
  ├── subnet-web      : 192.168.1.0/24    → 192.168.1.0   → 192.168.1.255
  ├── subnet-app      : 192.168.2.0/24    → 192.168.2.0   → 192.168.2.255
  ├── subnet-db       : 192.168.3.0/24    → 192.168.3.0   → 192.168.3.255
  └── GatewaySubnet   : 192.168.255.0/27  → 192.168.255.0 → 192.168.255.31
```

Copier

Pour ajouter les autres sous-réseaux (`subnet-app`, `subnet-db`, `GatewaySubnet`) après création : portail → VNet `vnet-nouvy` → menu **Paramètres** → **Sous-réseaux** → **+ Sous-réseau** (ou **+ Sous-réseau de passerelle** pour le `GatewaySubnet`).

> ***Plage 192.168.0.0/16*** *: RFC 1918, privée. Choisir une plage qui ne chevauche pas le réseau on-premise si une VPN est prévue. Si tes utilisateurs nomades sont susceptibles d'avoir une box internet en `192.168.1.x` (Livebox, Freebox, SFR...), préférer une plage moins exposée comme `192.168.100.0/22` pour éviter les conflits de routage en VPN P2S.*

Donner Internet à un VNet — NAT Gateway

> ***Depuis le 30 septembre 2025****, Azure a* ***retiré l'accès Internet sortant par défaut*** *sur les nouvelles VM. Sans configuration explicite (NAT Gateway, IP publique, Load Balancer), une VM Linux ne peut* ***plus*** *faire `apt update`, `curl`, etc. La* ***NAT Gateway*** *est la solution recommandée pour donner Internet sortant à un ou plusieurs subnets en production.*

Ce que fait — et ne fait pas — chaque "Gateway" Azure

⚠ **Le `GatewaySubnet` n'est pas une passerelle Internet** — il sert uniquement à héberger la VPN Gateway / ExpressRoute Gateway pour les tunnels chiffrés vers du on-premise ou des utilisateurs nomades.

Composant	Subnet associé	Rôle
VPN Gateway / ExpressRoute	`GatewaySubnet` *(nom réservé)*	Tunnel **chiffré** Azure ↔ on-premise / nomades — **PAS Internet général**
NAT Gateway	Subnet de VM *(association explicite)*	Donne **Internet sortant** au subnet entier
Application Gateway	Subnet dédié	Load Balancer L7 + WAF pour HTTPS entrant
Azure Bastion	`AzureBastionSubnet` *(nom réservé)*	SSH/RDP via portail sans exposer les VM

Création d'une NAT Gateway depuis le portail

Étape 1 — Créer la NAT Gateway

1. Portail → **Créer une ressource** → rechercher **Passerelle NAT** → **Créer**
2. Onglet **Informations de base** :
   - **Groupe de ressources** : `rg-nouvy-test`
   - **Nom** : `natgw-nouvy`
   - **Région** : même que le VNet
   - **Zone de disponibilité** : `Zone 1` *(redondance recommandée — la NAT Gateway est zonale)*
   - **Délai d'inactivité TCP** : `4 minutes` *(par défaut, peut être augmenté jusqu'à 120 min)*
3. Onglet **IP sortante** :
   - **Adresses IP publiques** : **Créer une nouvelle adresse IP publique**
     - **Nom** : `pip-natgw-nouvy`
     - **SKU** : `Standard` *(obligatoire pour NAT Gateway)*
     - **Attribution** : `Statique`
   - *(Optionnel)* **Préfixes d'adresses IP publiques** : pour réserver un bloc d'IP au lieu d'une seule
4. Onglet **Sous-réseau** :
   - **Réseau virtuel** : `vnet-nouvy`
   - Cocher les sous-réseaux qui auront Internet via cette NAT : `subnet-web`, `subnet-app`
   - *(Le `subnet-db` est souvent volontairement laissé sans Internet pour la sécurité de la base)*
5. Onglet **Étiquettes** *(optionnel)* : `environnement=prod`, `responsable=admin@nouvy.fr`
6. **Vérifier + créer** → **Créer**

> ***Création rapide** : la NAT Gateway est provisionnée en **1 à 2 minutes** (contrairement à une VPN Gateway qui prend 30-45 min).*

Étape 2 — Vérifier l'association au subnet

1. Portail → ouvrir le VNet `vnet-nouvy` → menu **Paramètres** → **Sous-réseaux**
2. Cliquer sur `subnet-web` → vérifier le champ **Passerelle NAT** : il doit afficher `natgw-nouvy`
3. Si non associé : éditer le subnet → champ **Passerelle NAT** → choisir `natgw-nouvy` → **Enregistrer**

Étape 3 — Tester depuis une VM du subnet

Depuis la VM Linux connectée en SSH *(via Azure Bastion ou IP publique temporaire le temps du test)* :

```
# Vérifier l'IP publique sortante (doit être celle de la NAT Gateway)
curl ifconfig.me

# Tester l'accès Internet
sudo apt update
curl -I https://www.google.com
```

Copier

Si `curl ifconfig.me` renvoie l'IP `pip-natgw-nouvy` créée à l'étape 1, la NAT Gateway fonctionne et toutes les VM du subnet sortent par cette IP.

Quand utiliser quoi pour l'accès Internet

Scénario	Solution recommandée
1 VM de test isolée	IP publique attachée à la VM + NSG restreint à ton IP source
Plusieurs VM, dev/prod	NAT Gateway pour la sortie + **Azure Bastion** pour le SSH/RDP admin (zéro VM exposée à Internet)
App publique avec back-ends privés	**Application Gateway** en frontal HTTPS + NAT Gateway pour les sorties des back-ends
BD critique	Aucune NAT, aucune IP publique — accès uniquement via VNet peering ou Private Endpoint

Coûts indicatifs

Composant	Coût France Central
NAT Gateway (heure)	~0,045 €/h soit **~32 €/mois**
Données traitées	~0,045 €/Go *(en plus du trafic Internet sortant standard)*
IP publique Standard statique	~3 €/mois

> ***Bonne pratique IP fixe** : la NAT Gateway donne une **IP sortante stable** — utile pour communiquer cette IP à un partenaire qui doit whitelister Azure (ex. webhook, API tierce, SMTP relai).*

Déployer une VM sur un VNet existant

Pourquoi suivre cette procédure

Quand on crée une VM Azure sans avoir préparé son VNet à l'avance, Azure auto-crée un VNet `<nom-VM>-vnet` et y soude la NIC. **Le champ "Réseau virtuel" d'une NIC existante est grisé à vie** dans le portail — Azure ne permet pas de migrer une NIC d'un VNet à un autre.

Opération	Possible ?
Changer le **sous-réseau** d'une NIC (dans le même VNet)	✅ Oui — NIC → Configurations IP → ipconfig1
Changer le **VNet** d'une NIC existante	❌ **Non** — limite Azure

Bonne pratique : créer le VNet en premier, puis créer chaque VM en sélectionnant explicitement le VNet existant `vnet-nouvy` au moment de la création — c'est ce qui suit.

Étape 1 — Créer la VM sur `vnet-nouvy`

1. Portail → **Créer une ressource** → **Machine virtuelle**

Onglet *Informations de base* :

Champ	Valeur
Abonnement	identique à `vnet-nouvy`
Groupe de ressources	`rg-nouvy-test`
Nom de la machine virtuelle	`vm-debian-01`
Région	⚠ **Identique à `vnet-nouvy`** (sinon le VNet ne sera pas proposé)
Options de disponibilité	`Aucune redondance d'infrastructure requise`
Type de sécurité	`Lancer des machines virtuelles approuvées` (Trusted Launch)
Image	`Debian 13 "trixie" x64 Gen2`
Taille	`Standard_B2ats_v2` *(suffisant + permet 2 NICs)*
Type d'authentification	`Clé publique SSH`
Nom d'utilisateur	`azureuser`
Source de la clé	*Générer une nouvelle paire de clés*
Ports d'entrée publics	`Aucun` *(si tu utilises Azure Bastion ou VPN P2S)* — sinon cocher **SSH (22)**

Onglet *Disques* : Type `Standard SSD`.

Onglet *Mise en réseau* — étape cruciale :

Champ	Valeur
Réseau virtuel	⚠ Ouvrir la **liste déroulante** → choisir **`vnet-nouvy`** *(et NON "Créer un nouveau réseau virtuel")*
Sous-réseau	`subnet-web (192.168.1.0/24)`
Adresse IP publique	`Aucun` *(si NAT Gateway ou Bastion)* — sinon laisser créer
Groupe de sécurité réseau de la carte réseau	`De base` ou existant
Ports d'entrée publics	mêmes choix qu'à l'onglet précédent
Supprimer la carte réseau et le disque public lors de la suppression de la VM	☑ Cocher — évite les ressources orphelines en cas de suppression future

> ***Si `vnet-nouvy` n'apparaît pas dans la liste** : c'est que la **région** choisie en haut ne correspond pas à celle du VNet. Revenir à l'onglet **Informations de base** et corriger la région.*

Onglets *Gestion*, *Surveillance*, *Avancé*, *Étiquettes* : laisser les valeurs par défaut ou ajouter tes tags `environnement=dev`, `projet=lab-azure`.

Onglet *Vérifier + créer* : validation Azure → **Créer**.

Étape 2 — Récupérer la clé SSH générée

Azure ne propose **qu'une seule fois** le téléchargement de la clé privée :

1. Pop-up post-création → cliquer **Télécharger la clé privée et créer la ressource**
2. Enregistrer le fichier `vm-debian-01_key.pem` dans un dossier sécurisé
3. Sur macOS/Linux, restreindre les permissions :

```
chmod 600 ~/Téléchargements/vm-debian-01_key.pem
```

Copier

> *Sur Windows, déplacer la clé dans `C:\Users\<vous>\.ssh\`. PowerShell intègre OpenSSH depuis Windows 10/11.*

Étape 3 — Vérifier l'attachement au bon VNet

1. Portail → la nouvelle VM `vm-debian-01` → **Vue d'ensemble**
2. Vérifier ces champs :

Champ	Valeur attendue
Réseau virtuel/sous-réseau	`vnet-nouvy/subnet-web`
IP privée	dans `192.168.1.x` (ex. `192.168.1.4`)
IP publique	présente si tu en as demandé une à l'étape 3
Emplacement	identique à celui de `vnet-nouvy`

Étape 4 — Tester l'accès Internet (NAT Gateway)

Si la NAT Gateway créée précédemment est bien associée à `subnet-web`, la VM accède à Internet en sortie sans IP publique :

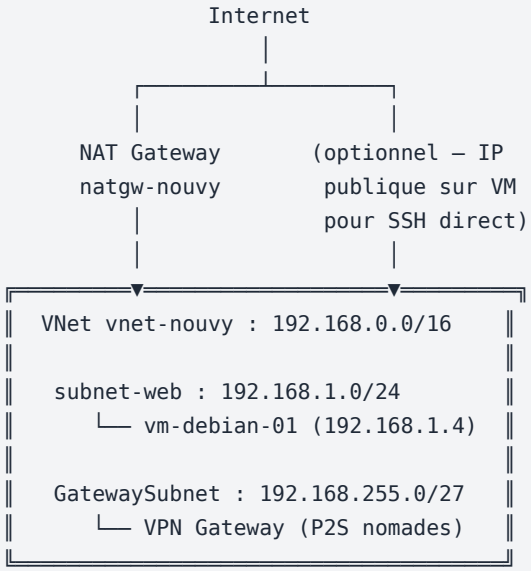
1. Se connecter à la VM *(via Azure Bastion, VPN P2S, ou IP publique temporaire)*
2. Tester :

```
# Vérifier l'IP sortante (doit être celle de la NAT Gateway pip-natgw-nouvy)
curl ifconfig.me

# Mettre à jour Debian
sudo apt update && sudo apt upgrade -y

# Tester l'accès web
curl -I https://www.google.com
```

Copier

Si `curl ifconfig.me` renvoie l'IP de la NAT Gateway et que `apt update` fonctionne → l'architecture est complète : **VNet + subnet + NAT Gateway + VM**.

Récapitulatif de l'architecture obtenue



Copier

Bonnes pratiques pour ne plus jamais retomber dans le piège

Règle	Pourquoi
Toujours créer le VNet AVANT la première VM	Évite l'auto-création de VNet "satellites" qui polluent l'abonnement
Garder la même région pour tout un projet	Pas de chevauchement de coût inter-régions, latence minimale, VNet réutilisable
Nommer les ressources avec le projet	`vnet-nouvy`, `vm-nouvy-debian-01`, `nic-nouvy-debian-01` — facilite la suppression groupée
Cocher "Supprimer en cascade" à la création VM	NIC, disque, IP publique supprimés avec la VM — évite les orphelins
Tagger systématiquement	`environnement=dev/prod`, `projet`, `responsable` — facilite l'audit Cost Management

Network Security Group (NSG)

Le **NSG** est un pare-feu stateful appliqué à un sous-réseau ou une NIC. Chaque règle a :

- Une **priorité** (100 à 4096, le plus petit gagne)
- Une **direction** (entrant / sortant)
- Une **source** et **destination** (IP, ASG, ou tag de service)
- Un **port et protocole**
- Une **action** (Allow / Deny)

Exemple de règles NSG

Priorité	Nom	Source	Destination	Port	Protocole	Action
100	Allow-RDP-Admin	IP-publique-admin	Any	3389	TCP	Allow
200	Allow-HTTPS	Any	Any	443	TCP	Allow
300	Allow-HTTP	Any	Any	80	TCP	Allow
4096	DenyAllInbound (par défaut)	Any	Any	Any	Any	Deny

Connectivité entre VNets — Peering

Le **peering** relie deux VNets (même région ou inter-région) via le backbone Azure (faible latence, bande passante élevée).

Création d'un peering depuis le portail :

1. Portail → ouvrir le 1er VNet (ex. `vnet-prod`) → menu **Paramètres** → **Peerings**
2. **+ Ajouter** : Azure crée les **deux** liens en une seule opération (prod→test et test→prod)
3. Section **Paramètres du réseau virtuel distant** :
    - **Nom du lien de peering depuis le réseau virtuel distant** : `peering-test-to-prod`
    - **Modèle de déploiement** : `Resource Manager`
    - **Réseau virtuel** : choisir `vnet-test` dans la liste (même tenant) ou coller l'**ID de ressource** (autre abonnement)
4. Section **Paramètres du réseau virtuel local** :
    - **Nom du lien de peering depuis le réseau virtuel local** : `peering-prod-to-test`
    - Cocher **Autoriser l'accès au réseau virtuel** sur les deux côtés (par défaut)
5. **Ajouter** → état des deux peerings passe à **Connecté** en quelques secondes

> *Un peering est facturé au Go transféré (entrant et sortant). Pour des volumes importants, considérer **Azure Virtual WAN**.*

Communication entre serveurs locaux et serveurs Azure

Trois solutions selon les besoins :

Option 1 — VPN Site-to-Site (S2S)

Tunnel IPsec/IKE entre le routeur du site on-premise et une **VPN Gateway** Azure.

Critère	Valeur
Bande passante	100 Mbps à 1,25 Gbps selon SKU
Coût	Modéré (~25 à 500 €/mois selon SKU)
Mise en œuvre	2 à 4 heures
Cas d'usage	PME, sites de taille moyenne

Option 2 — VPN Point-to-Site (P2S)

VPN client (OpenVPN ou IKEv2) installé sur les postes des utilisateurs nomades.

Cas d'usage : télétravail, accès admin ponctuel.

Critère	Valeur
Bande passante	100 Mbps à 1 Gbps partagés selon SKU
Coût	Faible (~25 à 100 €/mois selon SKU, pas de matériel)
Mise en œuvre	1 à 2 heures (passerelle + certificats + client)
Cas d'usage	Postes individuels, télétravailleurs, admins en déplacement

Étape 1 — Préparer le réseau virtuel (Gateway Subnet)

La VPN Gateway exige un sous-réseau **dédié** nommé `GatewaySubnet` dans le VNet cible.

1. Portail → ouvrir le VNet (ex. `vnet-nouvy`) → menu **Paramètres** → **Sous-réseaux**
2. **+ Sous-réseau de passerelle** *(et non `+ Sous-réseau` classique)*
3. **Plage d'adresses** : `192.168.255.0/27` *(au minimum /29 — /27 recommandé pour évolutivité ExpressRoute)*
4. **Enregistrer**

Étape 2 — Créer la VPN Gateway

1. Portail → **Créer une ressource** → rechercher **Passerelle de réseau virtuel** → **Créer**
2. Onglet **Informations de base** :
    - **Nom** : `vgw-nouvy-p2s`
    - **Région** : même que le VNet
    - **Type de passerelle** : `VPN`
    - **Type de VPN** : `Routé`
    - **SKU** : `VpnGw1` *(suffisant pour P2S — 250 connexions S2S, ~30 €/mois)*
    - **Génération** : `Generation1`
    - **Réseau virtuel** : `vnet-nouvy` *(le portail détecte automatiquement le `GatewaySubnet`)*
3. **Adresse IP publique** :
    - **Type** : `Créer une nouvelle adresse`
    - **Nom** : `pip-vgw-nouvy`
    - **SKU** : `Standard` *(obligatoire depuis 2024)*
    - **Attribution** : `Statique`

4. **Vérifier + créer**

> ***Patience** : la création d'une VPN Gateway prend **30 à 45 minutes** — c'est normal, Azure provisionne deux instances actives/passives en arrière-plan.*

Étape 3 — Configurer le Point-to-Site

1. Portail → ouvrir `vgw-nouvy-p2s` → menu **Paramètres** → **Configuration Point-à-site**
2. **Configurer maintenant**
3. **Pool d'adresses** : `172.16.10.0/24` *(plage attribuée aux clients VPN, ne doit chevaucher aucun sous-réseau du VNet ni du réseau on-premise)*
4. **Type de tunnel** :
   - `OpenVPN (SSL)` *(recommandé — fonctionne derrière la plupart des pare-feu via TCP 443)*
   - `IKEv2` *(client Windows natif, mais souvent bloqué par les pare-feu d'hôtel)*
   - `IKEv2 et OpenVPN` *(les deux activés simultanément)*
5. **Type d'authentification** : choisir une option *(voir étape 4)*

Étape 4 — Choisir le mode d'authentification

Méthode	Quand l'utiliser
Microsoft Entra ID *(recommandé)*	Tenant Entra ID existant — MFA + Accès conditionnel hérité, pas de certificats à gérer
Certificats Azure	Lab, petite équipe, pas d'Entra ID — gestion manuelle d'un certificat racine + certificats clients
Authentification RADIUS	Intégration avec un serveur RADIUS / NPS existant (souvent couplé AD on-premise)

Cas A — Authentification Microsoft Entra ID :

1. Dans **Configuration Point-à-site** → **Type d'authentification** : cocher **Azure Active Directory** *(libellé non encore mis à jour côté portail)*
2. **Locataire** : `https://login.microsoftonline.com/<tenant-id>/`
3. **Audience** : `c632b3df-fb67-4d84-bdcf-b95ad541b5c8` *(ID public de l'app Azure VPN — identique pour tous les tenants)*
4. **Émetteur** : `https://sts.windows.net/<tenant-id>/`
5. **Enregistrer**
6. Approuver l'app **Azure VPN** dans Entra ID : un admin du tenant doit visiter ce lien une seule fois :
   `https://login.microsoftonline.com/common/oauth2/authorize?client_id=41b23e61-6c1e-4545-b367-cd054e0ed4b4&response_type=code&redirect_uri=https://portal.azure.com&nonce=1234&prompt=admin_consent`

Cas B — Authentification par certificats *(recommandé pour Azure for Students)* :

> *Cette méthode ne dépend d'aucune licence Entra ID — tout est généré localement en PowerShell. C'est la voie la plus simple pour un lab.*

B.1 — Générer le certificat racine sur un poste Windows admin

1. Touche `Win` → taper `PowerShell` → clic droit → **Exécuter en tant qu'administrateur**
2. Coller la commande suivante (en une seule fois — les `` ` `` sont des sauts de ligne PowerShell) :

```
$cert = New-SelfSignedCertificate `
    -Type Custom -KeySpec Signature `
    -Subject "CN=P2SRootCert" `
    -KeyExportPolicy Exportable `
    -HashAlgorithm sha256 -KeyLength 2048 `
    -CertStoreLocation "Cert:\CurrentUser\My" `
    -KeyUsageProperty Sign -KeyUsage CertSign
```

Copier

Aucune sortie ne s'affiche si tout s'est bien passé. Le certificat racine `P2SRootCert` est créé dans le magasin **Personnel** de l'utilisateur courant.

B.2 — Générer un certificat client signé par la racine

Dans la **même** session PowerShell *(la variable `$cert` doit encore exister)* :

```
New-SelfSignedCertificate `
    -Type Custom `
    -DnsName P2SClientCert `
    -KeySpec Signature `
    -Subject "CN=P2SClientCert" `
    -KeyExportPolicy Exportable `
    -HashAlgorithm sha256 -KeyLength 2048 `
    -CertStoreLocation "Cert:\CurrentUser\My" `
    -Signer $cert `
    -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.2")
```

Copier

> *Génère un certificat **client** dérivé du racine. C'est lui qui sera installé sur les postes des utilisateurs nomades pour qu'ils puissent se connecter.*

B.3 — Exporter la clé publique du certificat racine

C'est cette clé publique (sans la privée) qu'on va coller dans Azure.

1. Touche `Win + R` → taper `certmgr.msc` → **Entrée**
2. Naviguer : **Certificats — Utilisateur actuel** → **Personnel** → **Certificats**
3. Repérer `P2SRootCert` → clic droit → **Toutes les tâches** → **Exporter…**
4. Assistant d'exportation :
    - **Suivant**
    - ⚠ **Non, ne pas exporter la clé privée** *(crucial — Azure ne doit JAMAIS recevoir la clé privée)*
    - **Suivant**
    - Cocher **X.509 codé en base 64 (.CER)** *(et NON le format DER binaire)*
    - **Suivant** → choisir un emplacement (ex. Bureau) + nom `P2SRootCert.cer`
    - **Suivant** → **Terminer**

B.4 — Récupérer le contenu base64 à coller dans Azure

1. Ouvrir `P2SRootCert.cer` avec **Bloc-notes** *(clic droit → Ouvrir avec → Bloc-notes)*
2. Le fichier contient :

```
-----BEGIN CERTIFICATE-----
MIIDazCCAlOgAwIBAgIQXk... (plusieurs lignes de caractères)
...XYZ12345=
-----END CERTIFICATE-----
```

Copier

3. **Sélectionner uniquement le contenu ENTRE** `-----BEGIN CERTIFICATE-----` et `-----END CERTIFICATE-----` *(sans ces 2 lignes elles-mêmes)*
4. **Ctrl+C** pour copier

> ⚠ *Azure attend uniquement le contenu base64. Si tu colles avec les en-têtes `BEGIN/END`, l'import échoue avec l'erreur* ***"au moins un certificat racine valide doit être configuré"****.*

B.5 — Coller la clé publique dans Azure

1. Portail → VPN Gateway → menu **Paramètres** → **Configuration Point-à-site**
2. **Type d'authentification** : cocher `Certificat Azure`
3. Section **Certificat racine** :
   - **Nom** : `P2SRootCert`
   - **Données publiques de certificat** : **coller** le contenu copié à l'étape B.4
4. **Enregistrer** *(barre supérieure)* — l'erreur de validation disparaît

B.6 — Exporter le certificat client pour distribution aux utilisateurs

Chaque utilisateur nomade doit installer le certificat client (avec sa clé privée) sur son poste.

1. Retour dans `certmgr.msc` → **Personnel** → **Certificats**
2. Repérer `P2SClientCert` → clic droit → **Toutes les tâches** → **Exporter…**
3. Cette fois :
   - **OUI, exporter la clé privée** *(on en a besoin côté client)*
   - Format **PFX** *(`.pfx` ou `.p12`)*
   - Cocher : ☑ **Inclure tous les certificats dans le chemin d'accès de certification** + ☑ **Exporter toutes les propriétés étendues**
   - Définir un **mot de passe** *(à communiquer à l'utilisateur par canal sécurisé — pas par email en clair)*
   - Enregistrer comme `P2SClientCert.pfx`
4. Distribuer le `.pfx` + le mot de passe à l'utilisateur

B.7 — Installation du certificat client chez l'utilisateur final

L'utilisateur (sur son poste Windows) :

1. Double-clic sur `P2SClientCert.pfx`
2. Choisir **Utilisateur actuel** → **Suivant**
3. Saisir le mot de passe fourni → **Suivant**
4. Laisser l'emplacement par défaut *(magasin automatique)* → **Suivant** → **Terminer**

Le certificat est installé. L'utilisateur peut maintenant télécharger le client VPN *(étape 5 ci-*

dessous) et se connecter.

Diagnostic rapide en cas d'erreur

Symptôme	Cause
"Au moins un certificat racine valide doit être configuré"	Le champ **Données publiques** est vide ou contient les en-têtes `-----BEGIN/END CERTIFICATE-----`
"Format incorrect" / "Données invalides"	Export en DER binaire au lieu de Base-64
`certmgr.msc` ne trouve pas le certificat	Vérifier dans **Utilisateur courant** → Personnel *(et non Ordinateur local)*
`New-SelfSignedCertificate` non reconnue	PowerShell pas lancé en admin, ou Windows trop ancien *(Win 10/11 ou Server 2016+ requis)*
L'utilisateur final n'arrive pas à se connecter	Le certificat client n'est pas installé, ou installé sur **Ordinateur local** au lieu d'**Utilisateur courant**

Étape 5 — Télécharger le client VPN

1. Toujours dans **Configuration Point-à-site** → bouton **Télécharger le client VPN** (en haut)
2. Azure génère une archive `.zip` contenant les profils de configuration pour Windows, macOS et Linux
3. Distribuer cette archive aux utilisateurs autorisés

Étape 6 — Installer et utiliser le client côté utilisateur

Sur Windows :

1. Télécharger et installer **Azure VPN Client** depuis le Microsoft Store *(obligatoire pour OpenVPN + Entra ID — IKEv2 utilise le client Windows natif)*
2. Ouvrir l'app → **+** (en bas à gauche) → **Importer**
3. Sélectionner le fichier `azurevpnconfig.xml` extrait de l'archive
4. **Enregistrer** → cliquer **Connecter**
5. **Authentification Entra ID** : popup de connexion Microsoft + MFA si exigé → connexion établie en quelques secondes

Sur macOS :

- Installer **Azure VPN Client** depuis l'App Store → importer le fichier `azurevpnconfig.xml` extrait de l'archive

Sur Linux :

- Installer **azure-vpn-client** via le dépôt Microsoft *(Ubuntu/Debian)* → importer le profil
- Alternative : configurer manuellement OpenVPN avec le fichier `.ovpn` fourni dans l'archive

Étape 7 — Vérifier la connexion

1. Sur le poste client connecté : ouvrir une invite de commande → `ipconfig` *(Windows)* ou `ifconfig` *(macOS/Linux)*

2. Une nouvelle interface affiche une IP dans le pool `172.16.10.x` défini à l'étape 3
3. Tester l'accès à une VM Azure : `ping 192.168.1.4` ou RDP/SSH sur une VM privée du VNet
4. Côté portail : **Configuration Point-à-site** → onglet **Allouer les adresses IP** → la liste affiche les utilisateurs actuellement connectés et leur IP attribuée

> ***Sécurité P2S Entra ID** : appliquer une stratégie **Accès conditionnel** ciblant l'app Azure VPN (App ID `41b23e61-6c1e-4545-b367-cd054e0ed4b4`) pour exiger MFA + conformité d'appareil Intune sur les connexions VPN.*

Option 3 — ExpressRoute

Liaison **privée dédiée** (fibre optique) entre l'entreprise et Azure, via un opérateur partenaire (Orange, Equinix, Colt...).

Critère	Valeur
Bande passante	50 Mbps à 100 Gbps
Coût	Élevé (à partir de ~200 €/mois + frais opérateur)
Latence	Très basse, garantie SLA
Cas d'usage	Grandes entreprises, charges critiques, conformité

> ***Recommandation hybride NOUVY** : commencer avec une **VPN S2S** (suffisante pour la majorité des PME), passer à ExpressRoute uniquement si la bande passante ou la latence pose problème.*

Montée en charge (Scaling) des applications

Azure propose plusieurs mécanismes pour absorber la montée en charge.

Scale-up vs Scale-out

Type	Description	Exemple Azure
Scale-up (vertical)	Augmenter la taille de la machine (plus de CPU/RAM)	Passer une VM de B2s à D4s
Scale-out (horizontal)	Ajouter plus d'instances identiques	Passer de 2 à 10 VM derrière un Load Balancer

Azure Load Balancer

Répartit le trafic réseau (couche 4 — TCP/UDP) entre plusieurs instances back-end.

Type	Cas d'usage
Public Load Balancer	Trafic Internet entrant vers un pool de VM
Internal Load Balancer	Équilibrage entre tiers d'application internes (ex. front → back)

Azure Application Gateway (couche 7)

Load Balancer **applicatif** (HTTP/HTTPS) avec :

- Routage par chemin d'URL (`/api` → backend1, `/static` → backend2)
- Affinité de session (cookies)
- Web Application Firewall (WAF) intégré
- SSL offloading

Virtual Machine Scale Sets (VMSS)

Un **VMSS** orchestre un groupe de VM identiques avec auto-scaling intégré.

Étape 1 — Créer le VMSS depuis le portail

1. Portail → **Créer une ressource** → rechercher **Virtual machine scale sets** → **Créer**
2. Onglet **Informations de base** :
    - **Groupe de ressources** : `rg-nouvy-test`
    - **Nom** : `vmss-web`
    - **Région** : même que vos autres ressources
    - **Mode d'orchestration** : `Flexible` *(recommandé depuis 2023)*
    - **Image** : `Ubuntu Server 24.04 LTS - x64 Gen2`
    - **Taille** : `Standard_B2s` *(ou B2ats_v2 pour budget étudiant)*
    - **Authentification** : Clé publique SSH + nom d'utilisateur `nouvyadmin`
3. Onglet **Mise en réseau** : choisir le VNet `vnet-nouvy` et activer un **Load Balancer** pour distribuer le trafic
4. Onglet **Mise à l'échelle** :
    - **Nombre d'instances initial** : `2`
    - **Stratégie de mise à l'échelle** : `Manuelle` pour l'instant (auto-scaling configuré à l'étape 2)
5. **Vérifier + créer**

Étape 2 — Configurer l'auto-scaling

1. Portail → VMSS `vmss-web` → menu **Paramètres** → **Mise à l'échelle**
2. Choisir **Mise à l'échelle automatique personnalisée**
3. **Nom** : `autoscale-web`
4. **Mode** : `Mettre à l'échelle en fonction d'une métrique`
5. **Limites d'instances** : Min `2`, Max `10`, Par défaut `2`
6. **Règles** : cliquer **+ Ajouter une règle**
    - **Source de la métrique** : la ressource VMSS courante
    - **Nom de la métrique** : `Percentage CPU`
    - **Opérateur** : `Supérieur à` `70` sur **10 minutes**
    - **Opération** : `Augmenter le nombre de` `1` instance
    - **Durée de refroidissement** : `5 minutes`
    - **Ajouter**
7. Ajouter une 2e règle symétrique : `Percentage CPU < 30` → `Diminuer de 1`
8. **Enregistrer**

App Service — auto-scaling intégré

Pour les App Services en plan **Standard** ou supérieur, l'auto-scaling se configure depuis le portail :

- Onglet **Mise à l'échelle horizontale** → définir min/max instances
- Règles basées sur métriques (CPU %, mémoire, requêtes/sec)

---

5. Sécurisation et protection

Microsoft Defender for Cloud (ex Azure Security Center)

Microsoft Defender for Cloud est le centre de sécurité unifié d'Azure. Il fournit :

Fonction	Description
Score de sécurisation	Note globale (0-100 %) basée sur des recommandations
Recommandations de sécurité	Liste priorisée d'actions (ex: "activer le chiffrement disque", "désactiver SSH ouvert")
Alertes de sécurité	Détection de menaces (force brute SSH, exfiltration, malware)
Inventaire des ressources	Vue d'ensemble des ressources et de leur posture de sécurité
Conformité réglementaire	Tableaux de bord ISO 27001, PCI-DSS, SOC 2, RGPD

Activation : Portail → **Microsoft Defender for Cloud** → **Vue d'ensemble**. Plan gratuit ou Standard (payant, ~13 €/serveur/mois).

Microsoft Entra ID (anciennement Azure Active Directory)

Microsoft Entra ID (renommé en 2023, anciennement Azure AD) est le service d'**identité et d'accès** d'Azure et Microsoft 365.

Concept	Rôle
Tenant	Annuaire unique pour une organisation (`nouvy.onmicrosoft.com`)
Utilisateur	Compte individuel (membre, invité B2B, ou compte de service)
Groupe	Regroupement d'utilisateurs (sécurité ou Microsoft 365)
Application	App enregistrée pour SSO ou OAuth
Service Principal	Identité d'application utilisée pour automatisation
Identité managée	Identité automatique attribuée à une ressource Azure (sans gérer de secrets)

Différences avec Active Directory on-premise

Critère	AD on-premise (AD DS)	Microsoft Entra ID
Protocoles	Kerberos, LDAP, NTLM	OAuth 2.0, OpenID Connect, SAML 2.0
Structure	Forêt → Domaines → OU	Plat (groupes, unités administratives)
GPO	Oui	Non (équivalent : Intune)
Hébergement	Serveurs sur site	Cloud Microsoft

> ***Hybride** : on peut **synchroniser** un AD on-premise avec Entra ID via **Microsoft Entra Connect** — les utilisateurs gardent un seul mot de passe et accèdent à Office 365/Azure avec leurs identifiants AD.*

Création d'un utilisateur dans Entra ID

1. Portail → **Microsoft Entra ID** → menu **Gérer** → **Utilisateurs**
2. Barre supérieure → **+ Nouvel utilisateur** → **Créer un nouvel utilisateur**
3. Onglet **Informations de base** :
    - **Nom d'utilisateur principal** : `marie.durand` (le suffixe `@nouvy.onmicrosoft.com` est ajouté automatiquement)
    - **Nom à afficher** : `Marie Durand`
    - **Mot de passe** : choisir **Générer automatiquement** *(à transmettre à l'utilisatrice)*, ou **Permettre de créer le mot de passe**
    - Cocher **Le compte est activé**
4. Onglet **Propriétés** *(optionnel)* : poste, service, manager, lieu d'utilisation
5. Onglet **Affectations** *(optionnel)* : ajouter à un groupe ou un rôle
6. **Vérifier + créer** → **Créer**

> *Le mot de passe affiché à la création doit être communiqué à l'utilisatrice — elle sera invitée à le changer à sa première connexion.*

Création d'un groupe et attribution de rôle (RBAC)

RBAC — Role-Based Access Control

Le contrôle d'accès dans Azure repose sur **3 éléments** :

- **Qui** (utilisateur, groupe, service principal)
- **Quoi** (rôle = ensemble de permissions)
- ****(scope = ressource concernée : abonnement, groupe de ressources, ressource individuelle)

Rôles intégrés Azure les plus utilisés

Rôle	Permissions
Owner (Propriétaire)	Tout, y compris attribution de rôles
Contributor (Contributeur)	Tout sauf attribution de rôles
Reader (Lecteur)	Lecture seule
User Access Administrator	Gérer les attributions de rôles uniquement
Virtual Machine Contributor	Gérer les VM mais pas le réseau ni le stockage
Storage Blob Data Contributor	Lire/écrire dans Blob Storage
Network Contributor	Gérer le réseau (VNet, NSG, IP publique)

Attribution d'un rôle à un groupe

1. Portail → **Microsoft Entra ID** → **Groupes** → **Nouveau groupe**
2. Type : `Sécurité`, Nom : `GRP_Azure_Admins`
3. Ajouter les membres → Créer
4. Portail → ressource cible (ex. groupe de ressources `rg-nouvy-prod`) → **Contrôle d'accès (IAM)**
5. **Ajouter une attribution de rôle** → choisir `Contributor`
6. **Membres** : sélectionner `GRP_Azure_Admins` → **Vérifier + attribuer**

> ***Bonne pratique** : ne jamais attribuer de rôle directement à un utilisateur. Toujours passer par un **groupe** — ça simplifie les départs/arrivées.*

Activation MFA (Multi-Factor Authentication)

Le **MFA** ajoute une 2e étape de vérification (SMS, app Authenticator, clé FIDO2) en plus du mot de passe. **Indispensable** pour tout compte admin.

Méthode 1 — Security Defaults (gratuit, simple)

1. Portail → **Microsoft Entra ID** → **Propriétés** (en bas)
2. **Gérer les paramètres de sécurité par défaut** → Activer
3. → MFA obligatoire pour tous les utilisateurs et admins

Méthode 2 — Conditional Access (licence P1/P2 requise, granulaire)

1. Portail → **Microsoft Entra ID** → **Sécurité** → **Accès conditionnel**
2. **Nouvelle stratégie** :
    - **Affectations** : utilisateurs ciblés (ex. `GRP_Azure_Admins`)
    - **Applications cloud** : `Toutes les applications cloud`
    - **Conditions** : `Hors zone de confiance` (en dehors du bureau)
    - **Octroi** : `Exiger l'authentification multifacteur`

Côté utilisateur — première authentification

À la prochaine connexion, l'utilisateur :

1. Saisit son login + mot de passe
2. Reçoit un message demandant de configurer MFA

3. Installe l'app **Microsoft Authenticator** (iOS/Android)
4. Scanne un QR code → l'app génère des codes à 6 chiffres
5. À chaque connexion, valide via l'app (notification push ou code)

> ***Recommandation forte** : activer MFA sur 100 % des comptes, pas seulement les admins. Microsoft impose progressivement le MFA sur les portails admin Azure depuis 2024.*

Chiffrement des données

Azure chiffre par défaut **toutes les données au repos** :

Service	Chiffrement par défaut
Disques VM	Azure Disk Encryption (BitLocker pour Windows, dm-crypt pour Linux)
Blob Storage	AES-256, géré par Azure
Azure SQL	Transparent Data Encryption (TDE) activé par défaut
Cosmos DB	AES-256 par défaut

Pour aller plus loin : utiliser **Azure Key Vault** pour stocker des secrets, certificats, et clés cryptographiques avec rotation automatique.

Étape 1 — Créer le Key Vault

1. Portail → **Créer une ressource** → rechercher **Key Vault** → **Créer**
2. Onglet **Informations de base** :
   - **Groupe de ressources** : `rg-nouvy-prod`
   - **Nom du coffre de clés** : `kv-nouvy-prod` *(globalement unique, 3-24 caractères)*
   - **Région** : `France Central`
   - **Niveau tarifaire** : `Standard` *(suffisant — Premium pour HSM)*
3. Onglet **Configuration de l'accès** :
   - **Modèle d'autorisation** : `Contrôle d'accès en fonction du rôle Azure` *(recommandé, gère via RBAC)*
4. Onglet **Mise en réseau** : `Activer l'accès public depuis tous les réseaux` pour les tests *(restreindre en prod via Private Endpoint)*
5. **Vérifier + créer**

Étape 2 — Ajouter un secret

1. Portail → ouvrir le Key Vault `kv-nouvy-prod`
2. Menu **Objets** → **Secrets** → **+ Générer/Importer**
3. **Options de chargement** : `Manuel`
4. **Nom** : `SqlAdminPassword`
5. **Valeur secrète** : `MotDePasseTresFort!2026`
6. *(Optionnel)* **Date d'activation** / **Date d'expiration**
7. **Créer**

> *Pour qu'une App Service ou une VM consomme ce secret, lui attribuer une **identité managée** + le rôle `Key Vault Secrets User` sur le coffre — plus de mot de passe dans le code.*

---

6. Monitoring et bonnes pratiques

Azure Monitor

Azure Monitor est la plateforme unifiée de surveillance des ressources Azure et hybrides.

Composants principaux

Composant	Rôle
Metrics	Métriques numériques temps réel (CPU, RAM, IOPS, requêtes/sec) — granularité 1 minute
Logs (Log Analytics)	Journaux applicatifs et système, requêtables en KQL
Alerts	Alertes basées sur métriques ou requêtes log → email, SMS, webhook, Teams
Application Insights	APM (monitoring applicatif) — traces, dépendances, exceptions
Workbooks	Tableaux de bord interactifs personnalisés

Création d'une alerte simple — CPU > 80 %

1. Portail → ressource (ex. VM) → **Alertes** → **Créer** → **Règle d'alerte**
2. **Étendue** : la VM cible
3. **Condition** : signal `Percentage CPU` → seuil `> 80 %` sur 5 minutes
4. **Action** : créer un groupe d'actions (email à `admin@nouvy.fr`, ou webhook Teams)
5. **Détails** : nom, gravité, description

Requêtes Log Analytics (KQL)

Exemple — top 10 des erreurs par application sur les 24 dernières heures :

```
AppExceptions
| where TimeGenerated > ago(24h)
| summarize Count = count() by AppRoleName, OuterMessage
| top 10 by Count desc
```

Copier

Azure Advisor

Azure Advisor analyse en continu les ressources et fournit des **recommandations personnalisées** dans 5 catégories :

Catégorie	Type de recommandation
Fiabilité	Activer la sauvegarde, redondance géographique, haute disponibilité
Sécurité	Lien avec Microsoft Defender for Cloud — règles NSG trop permissives, MFA non activé
Performances	Augmenter la taille d'une VM saturée, indexer une BD lente
Coût	VM sous-utilisées à redimensionner ou arrêter, instances réservées
Excellence opérationnelle	Conventions de nommage, tags manquants

Accès : Portail → **Advisor** → tableau de bord global.

Conventions de nommage et bonnes pratiques

Nommage des ressources

Format recommandé : `<type>-<projet>-<environnement>-<numéro>`

Type ressource	Préfixe	Exemple
Resource Group	`rg-`	`rg-nouvy-prod-fc`
Virtual Machine	`vm-`	`vm-web-prod-01`
Virtual Network	`vnet-`	`vnet-nouvy-prod`
Subnet	`snet-`	`snet-web`
NSG	`nsg-`	`nsg-web-prod`
Storage Account	`st` (sans tiret, minuscule)	`stnouvyprod01`
Key Vault	`kv-`	`kv-nouvy-prod`
App Service	`app-`	`app-nouvy-site`
SQL Server	`sql-`	`sql-nouvy-prod`

Tags obligatoires

Tagger systématiquement les ressources :

- `environnement` : `prod`, `staging`, `dev`, `test`
- `projet` : `site-web`, `erp`, `dataviz`
- `responsable` : email du propriétaire
- `centre-de-couts` : code comptable

Appliquer des tags depuis le portail :

1. Ouvrir la ressource cible (VM, groupe de ressources, App Service…) dans le portail
2. Menu **Vue d'ensemble** → cliquer **Modifier** à droite de **Étiquettes** *(ou utiliser le menu **Étiquettes** dans la barre latérale)*
3. Ajouter chaque paire clé/valeur :

    - `environnement` → `prod`
    - `projet` → `site-web`
    - `responsable` → `admin@nouvy.fr`
    - `centre-de-couts` → `CC-2026-001`
4. **Appliquer**

> ***Bulk tagging** : pour étiqueter plusieurs ressources d'un coup, ouvrir **Toutes les ressources** dans le portail → filtrer/cocher les ressources → barre supérieure → **Affecter des étiquettes**.*

Maîtrise des coûts

Action	Bénéfice
Arrêter (deallocate) les VM la nuit/week-end	Économie 60-70 % sur le compute
Auto-shutdown : portail → VM → Auto-shutdown	Évite l'oubli (planification 19h/jour)
Réserver les VM long terme (1 ou 3 ans)	Réduction 30-72 % vs pay-as-you-go
Niveau Cool/Archive pour les blobs anciens	Économie 40-90 % sur le stockage
Supprimer les disques orphelins (issus de VM supprimées)	Évite la facturation à vide
Budgets et alertes Cost Management	Notification email à 80 % du budget

Sauvegardes

Configurer **Azure Backup** pour protéger les VM et bases de données depuis le portail :

Étape 1 — Créer un coffre Recovery Services

1. Portail → **Créer une ressource** → rechercher **Coffre Recovery Services** → **Créer**
2. **Informations de base** :
    - **Groupe de ressources** : `rg-nouvy-prod`
    - **Nom du coffre** : `rsv-nouvy-prod`
    - **Région** : `France Central` *(doit être identique à la région des VM à protéger)*
3. **Vérifier + créer**

Étape 2 — Activer la sauvegarde d'une VM

1. Portail → ouvrir le coffre `rsv-nouvy-prod`
2. Menu **Démarrer** → **+ Sauvegarde**
3. **Type de source de données** : `Machine virtuelle Azure`
4. **Stratégie de sauvegarde** : `DefaultPolicy` (quotidienne, rétention 30 jours) — ou créer une stratégie personnalisée
5. **+ Ajouter** → cocher les VM à protéger (ex. `vm-web-01`) → **OK**
6. **Activer la sauvegarde**
7. Première sauvegarde : déclencher manuellement via VM → **Sauvegarde** → **Sauvegarder maintenant**

> ***Politique recommandée** : sauvegarde quotidienne, rétention 30 jours quotidienne + 12 mois mensuelle + 5 ans annuelle pour les données critiques. Personnaliser via **Coffre** → **Stratégies de sauvegarde** → **+ Ajouter**.*

Récapitulatif — les concepts clés à retenir

Notion	Définition courte
IaaS	Infrastructure brute louée (VM, réseau) — ex. Azure VMs
PaaS	Plateforme prête à recevoir du code — ex. App Service
SaaS	Logiciel clé en main accessible navigateur — ex. Microsoft 365
Cloud public	Mutualisé entre clients via Internet (Azure, AWS, GCP)
Cloud privé	Dédié à une seule organisation
Cloud hybride	Public + privé/on-premise reliés
Resource Group	Conteneur logique de ressources Azure (gestion + permissions)
VNet	Réseau virtuel privé dans Azure
NSG	Pare-feu IP/port appliqué à un sous-réseau ou NIC
VPN Gateway / ExpressRoute	Liaison entreprise ↔ Azure
VMSS / App Service auto-scale	Scaling horizontal automatique
Microsoft Entra ID	Annuaire d'identités cloud (ex Azure AD)
RBAC	Contrôle d'accès par rôle (Owner, Contributor, Reader…)
MFA	Authentification à 2 facteurs
Microsoft Defender for Cloud	Centre de sécurité Azure unifié
Azure Monitor	Plateforme unifiée de monitoring (métriques, logs, alertes)
Azure Advisor	Recommandations automatiques (fiabilité, sécurité, coût, perf, ops)

Pour aller plus loin

- **Certifications Microsoft** :

    - **AZ-900** *(Azure Fundamentals)* — niveau débutant, prouve les concepts de ce cours
    - **AZ-104** *(Azure Administrator Associate)* — administrateur cloud
    - **AZ-305** *(Azure Solutions Architect Expert)* — architecte cloud
    - **SC-300** *(Identity and Access Administrator)* — focus identité

- **Compte gratuit Azure** : `https://azure.microsoft.com/fr-fr/free` — 200 $ de crédit pendant 30 jours + plus de 25 services gratuits 12 mois

- **Microsoft Learn** : `https://learn.microsoft.com/fr-fr/training/azure/` — formations gratuites avec sandbox

- **Azure Architecture Center** : référentiels d'architectures de référence par scénario (web app, microservices, IoT, BI, etc.)