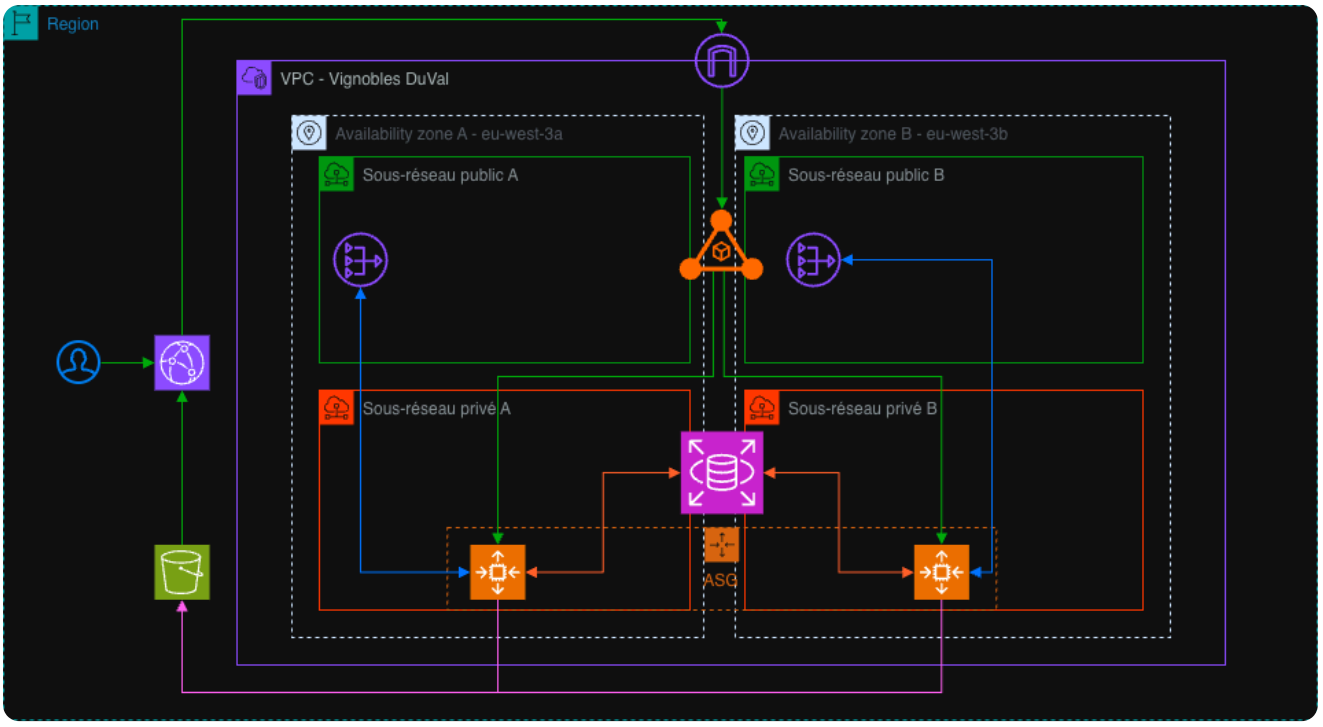
Labo AWS - Architecture Cloud de l'entreprise Vignes DuVal

Compte rendu rédigé par GADONNAUD EWEN, étudiant en 1ère année de BTS SIO option SISR, au lycée Paul-Louis Courier de Tours (2025-2026)

Sommaire



---

1. Introduction et Objectif

Ce document détaille ma conception d'une architecture cloud pour l'entreprise (fictive) **"Vignes DuVal"**, dont l'infrastructure applicative (site WordPress) est actuellement hébergée sur un unique serveur physique local (*on-premise*).

L'architecture actuelle présente des risques critiques de type **Single Point of Failure (SPOF)**, une performance limitée à l'international, et une posture de sécurité non optimale (co-localisation de la base de données et du serveur web).

L'objectif de ce laboratoire, pour moi, est de concevoir une architecture cible sur Amazon Web Services (AWS) répondant aux principes du "Well-Architected Framework", puis de mettre en œuvre une preuve de concept (PoC) fonctionnelle de ses fondations. L'architecture vise à être :

- **Hautement Disponible** : Résiliente aux pannes d'un composant ou d'un centre de données (Zone de Disponibilité).
- **Sécurisée** : Basée sur le principe de moindre privilège et l'isolation stricte des composants.
- **Scalable** : Capable de s'adapter automatiquement aux variations de trafic.
- **Performante** : Offrant une faible latence aux utilisateurs mondiaux.

---

2. Conception de l'Architecture Cible (Analyse du Schéma V2)

L'architecture cible (voir schéma présent à la première page, en haut) est une conception **3-tiers découplée** que j'ai réalisée, répartie sur deux Zones de Disponibilité (AZ) pour une redondance complète.

2.1. Justification des Services Réseau (VPC, Subnets, Gateways)

VPC (Virtual Private Cloud) : Crée un datacenter virtuel logiquement isolé pour "Vignes DuVal", offrant un contrôle granulaire sur l'adressage IP et le routage.

Multi-AZ (AZ A & AZ B) : L'architecture est dupliquée sur deux datacenters physiquement distincts. En cas de défaillance de l'AZ A, l'AZ B prend le relais sans interruption de service.

Subnets Publics vs Privés :

- Les **Subnets Publics** agissent comme une zone démilitarisée (DMZ). Ils n'hébergent que les services devant être joints depuis Internet (Load Balancer, NAT Gateways).
- Les **Subnets Privés** hébergent le cœur de l'application (serveurs EC2, base de données RDS). Ils sont inaccessibles depuis Internet, réduisant drastiquement la surface d'attaque.

Internet Gateway (IGW) : Permet la communication bidirectionnelle entre le VPC et Internet (Flux Ingress).

NAT Gateways : Déployées dans chaque AZ public, elles fournissent un accès Internet sortant (Egress) aux instances des subnets privés (ex: pour les mises à jour `dnf update`). Cette redondance évite qu'une panne d'AZ ne coupe l'accès aux mises à jour pour l'autre.

2.2. Justification des Services Applicatifs (ALB, ASG, EC2)

Application Load Balancer (ALB) : Positionné dans les subnets publics, il sert de point d'entrée unique. Il répartit le trafic HTTP/S sur les instances EC2 saines dans les deux AZ et gère le déchargement SSL.

Auto Scaling Group (ASG) : Garantit qu'un nombre défini d'instances EC2 (minimum 1 par AZ) est toujours en cours d'exécution. Il remplace automatiquement les instances défaillantes (auto-guérison) et peut ajouter/supprimer des instances en fonction de la charge (élasticité).

Instances EC2 (Subnets Privés) : Les serveurs web WordPress. Les placer dans des subnets privés est une bonne pratique de sécurité fondamentale : seul l'ALB est autorisé à communiquer avec eux.

2.3. Justification des Services de Données (RDS, S3)

Amazon RDS (Multi-AZ) : Fournit une base de données MySQL gérée. Le choix "Multi-AZ" provisionne une réplique *standby* synchrone dans une AZ distincte. En cas de défaillance de la base de données primaire, AWS bascule automatiquement le trafic sur la réplique.

Amazon S3 (Simple Storage Service) : Utilisé pour stocker les médias statiques (images, PDF) de WordPress. Cela découple les fichiers du cycle de vie des instances EC2 et offre une durabilité et une scalabilité quasi illimitées à faible coût.

2.4. Justification des Services de Périphérie (CloudFront, VPC Endpoint)

Amazon CloudFront : Un CDN (Content Delivery Network) qui met en cache le contenu (S3) et les pages (ALB) dans des points de présence mondiaux. Cela réduit la latence pour les clients internationaux de "Vignes DuVal" et absorbe une partie du trafic, réduisant la charge sur l'infrastructure.

VPC Endpoint (pour S3) : (Implicite dans le schéma) Un "Gateway Endpoint" sera configuré pour les subnets privés. Il permet aux instances EC2 de communiquer avec S3 (pour l'upload de médias) en utilisant le réseau privé d'AWS, sans passer par la NAT Gateway. C'est plus sécurisé et réduit les coûts de transfert de données.

---

3. Analyse des Flux de Données

L'architecture V2 gère quatre flux de trafic distincts :

1. **Flux Client (HTTP/S)** : L'utilisateur (bleu) contacte CloudFront (violet). Le CDN sert le contenu depuis son cache. Si le contenu n'est pas en cache, CloudFront entre dans le VPC via l'IGW (violet), atteint l'ALB (orange), qui relaie la requête à une instance EC2 (orange) disponible.
2. **Flux Applicatif (Base de Données)** : Les instances EC2 (orange) contactent l'endpoint RDS (violet) via le réseau privé (flux orange) pour lire/écrire les données (articles, commentaires).
3. **Flux de Mises à Jour (Egress)** : Les instances EC2 (orange) initient une connexion sortante vers leur NAT Gateway respective (violet/bleu), qui relaie la demande à l'IGW (gris) pour accéder à Internet (ex: dépôts de paquets).
4. **Flux Média (S3)** :
   - **Upload** : Une instance EC2 (orange) envoie un média (photo ou autre) directement à S3 (vert) via le VPC Endpoint (flux violet).
   - **Download** : CloudFront (violet) récupère les médias sur S3 (vert) et les sert à l'Utilisateur (bleu).

---

4. Périmètre d'Implémentation du Laboratoire (Fondation V1)

L'architecture cible V2 garantit la haute disponibilité mais inclut des services (ALB, Multi-AZ RDS, NAT Gateways, ASG multi-instance) qui ne sont pas inclus dans le "Free Tier" d'AWS.

Ce laboratoire se concentre donc sur l'implémentation de la **fondation V1** de cette architecture. Cette approche conserve le même *workflow* logique (découplage, isolation) tout en restant dans les limites du Free Tier.

4.1 Services implémentés (V1) :

- VPC (`10.0.0.0/16`)
- Internet Gateway (IGW)
- **Subnets** :
  - Subnet-Public-A (`10.0.1.0/24`) dans `eu-west-3a`
  - Subnet-Privé-A (`10.0.2.0/24`) dans `eu-west-3a`
  - Subnet-Privé-B (`10.0.3.0/24`) dans `eu-west-3b` (requis pour la validation RDS)
- **Route Tables** : RTB-Public et RTB-Private
- VPC Gateway Endpoint pour S3
- 1 Instance EC2 t2.micro (dans le Subnet-Public-A)
- 1 Instance RDS t2.micro Single-AZ (utilisant les subnets privés A et B)
- 1 Bucket S3
- Rôles IAM et Security Groups stricts
- 1 Distribution CloudFront (CDN)

---

5. Journal d'Implémentation et Analyse des Problèmes Rencontrés

L'implémentation de la fondation V1 a été réalisée manuellement via la console AWS pour un contrôle granulaire des composants. Ce journal détaille les étapes chronologiques et les obstacles dont j'ai fait face, ainsi que leurs solutions.

Phase 0 : Initialisation du Compte

Action : Création du compte AWS et validation.

Constat : Présence de 100$ de crédits promotionnels, agissant comme un filet de sécurité en cas de dépassement accidentel du Free Tier.

Action : Changement de la langue de la console (Français → Anglais) pour s'aligner sur la terminologie standard de l'industrie.

Phase 1 : Réseau (VPC)

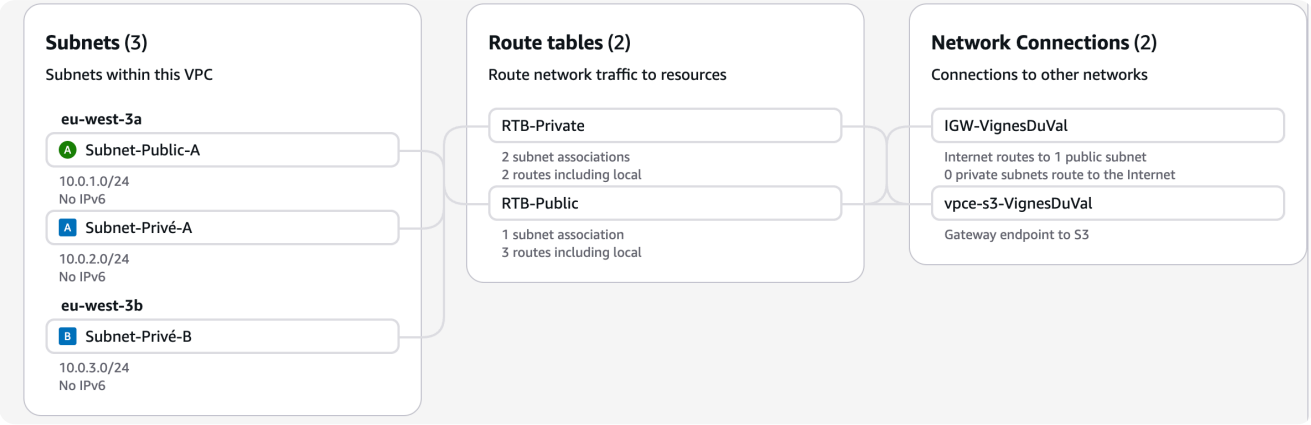
Action : L'assistant "VPC and more" a été évalué.

Obstacle 1 : Manque de contrôle CIDR. L'assistant impose des blocs CIDR en `/20` (4 096 IP), que j'ai jugés trop larges et rigides.

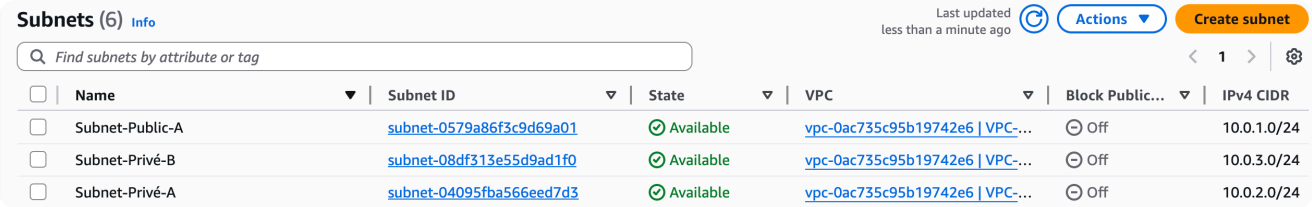
Résolution : L'assistant a été abandonné au profit d'une création 100% manuelle.

Actions :

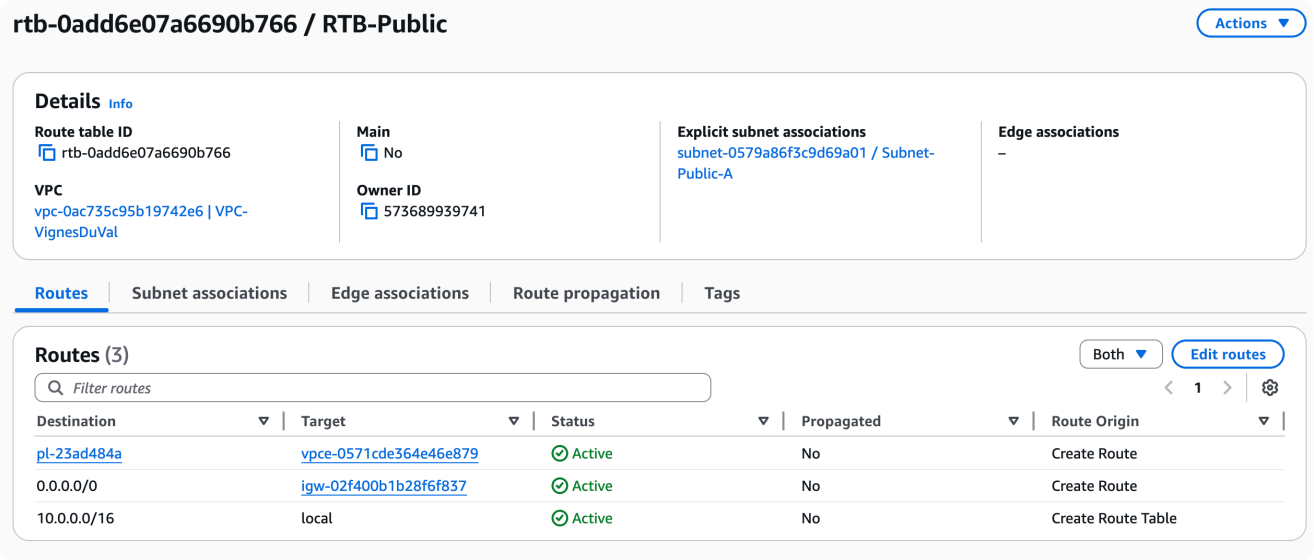
- Création du VPC-VignesDuVal (`10.0.0.0/16`)
- Création des subnets Subnet-Public-A (`10.0.1.0/24`) et Subnet-Privé-A (`10.0.2.0/24`), tous deux dans l'AZ `eu-west-3a`
- Création et attachement de l'IGW-VignesDuVal
- Création de RTB-Public, ajout d'une route `0.0.0.0/0` vers l'IGW, et association avec Subnet-Public-A
- Identification de la table de routage principale, renommage en RTB-Private, et association avec Subnet-Privé-A
- Création d'un VPC Gateway Endpoint pour S3, associé à la RTB-Private



Flux de sous-réseaux, tables de routage et connections aux autres réseaux pour le VPC-VignesDuVal



Configuration manuelle des sous-réseaux



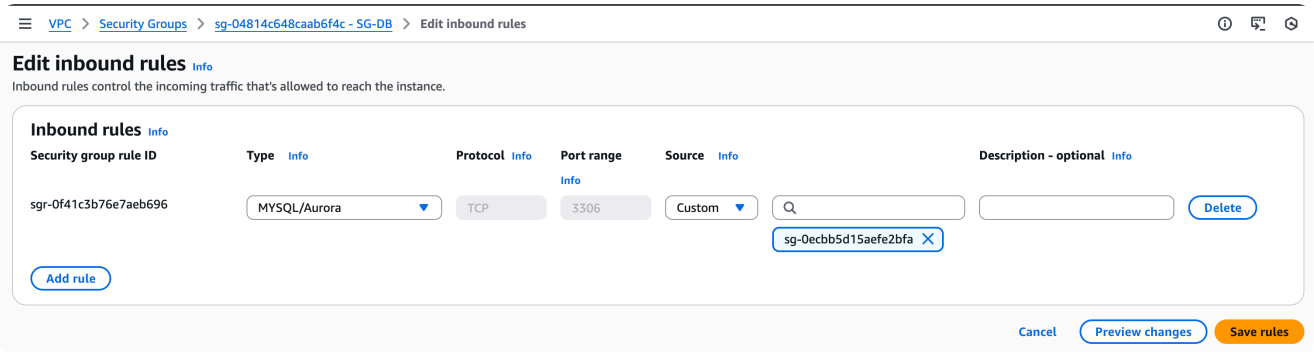
Association de RTB-Public à la passerelle IGW (Ingress)

Phase 2 : Sécurité (IAM & Security Groups)

Action (IAM) : Création d'un Rôle IAM `Role-EC2-S3-VignesDuVal` pour le service EC2, avec la policy managée `AmazonS3FullAccess`.

Action (SG) : Création de deux Security Groups (SG) dans le VPC-VignesDuVal :

- **SG-Web** : Autorise SSH (Port 22) depuis My IP et HTTP (Port 80) depuis Anywhere-IPv4
- **SG-DB** : Autorise MySQL/Aurora (Port 3306) avec pour source l'ID du SG-Web



Règle Inbound du SG-DB montrant la source SG-Web, étant la seule source pouvant accéder à la DB

Phase 3 : Services de Données (S3 & RDS)

Action (S3) : Création du bucket `vignes-duval-media-0001` en mode "Block all public access".

Action (RDS) : Lancement d'une instance RDS t2.micro (MySQL, Single-AZ), configurée pour utiliser le VPC-VignesDuVal et le SG-DB.

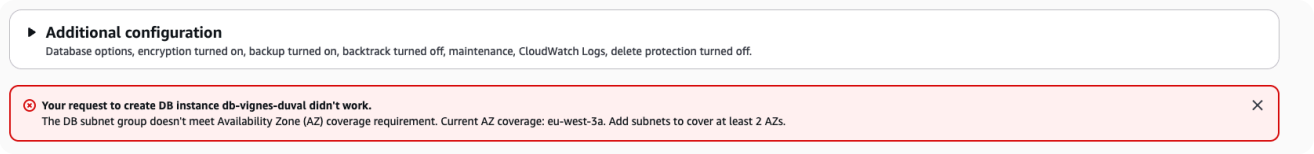
Obstacle 2 : Défaillance de création RDS (Couverture d'AZ).

Erreur :

```
The DB subnet group doesn't meet Availability Zone (AZ) coverage
requirement.
Current AZ coverage: eu-west-3a. Add subnets to cover at least 2 AZs.
```

Analyse : Le service RDS exige, pour des raisons de validation de haute disponibilité, que le "DB Subnet Group" (même pour une instance Single-AZ) soit composé de subnets dans au moins deux AZ différentes. Notre réseau ne contenait des subnets que dans `eu-west-3a`.

Résolution : Retour à la console VPC. Création d'un nouveau subnet Subnet-Privé-B (`10.0.3.0/24`) dans l'AZ `eu-west-3b`. Association de ce nouveau subnet à la RTB-Private. La création de l'instance RDS a ensuite été relancée avec succès.



Erreur de création de la DB à cause de la non présence de multiples AZ

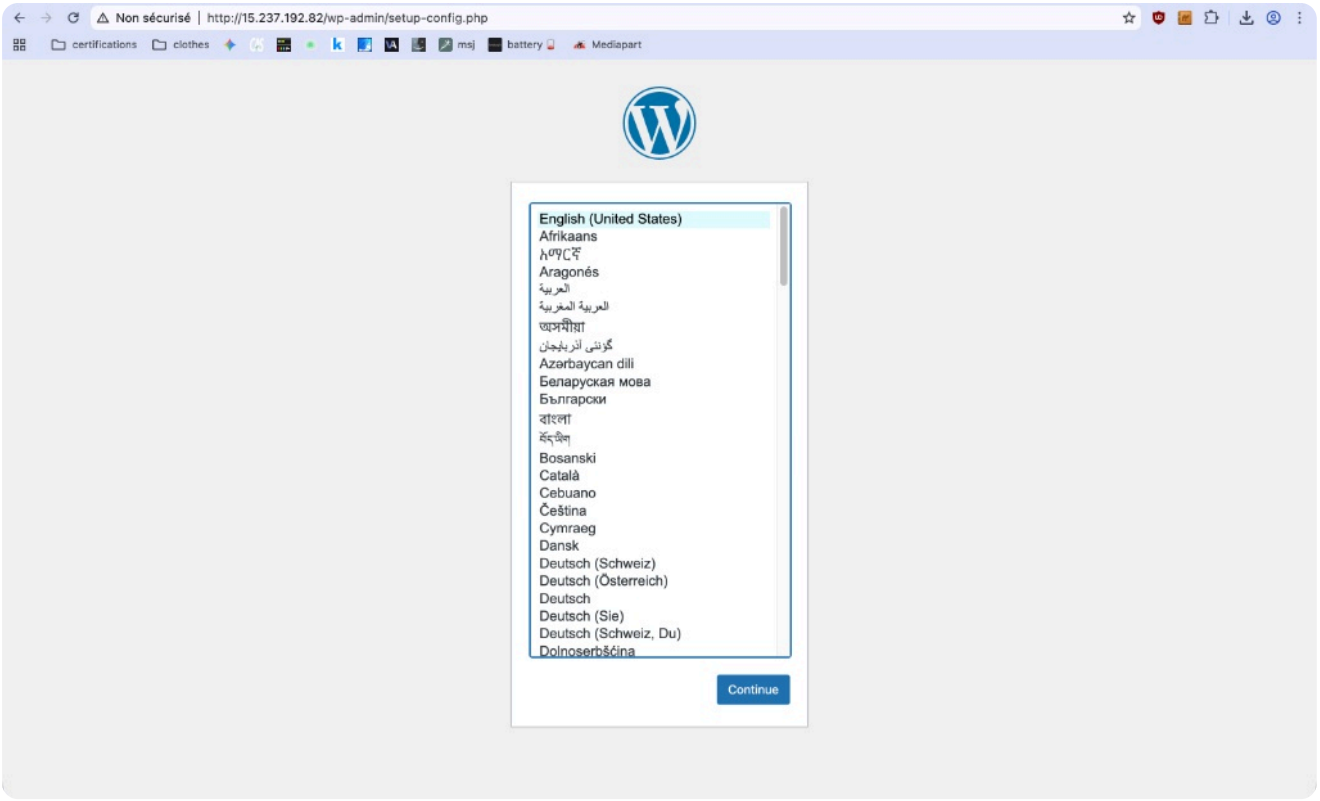
Phase 4 : Calcul et Applicatif (EC2 & WordPress)

Action (EC2) : Lancement d'une instance t2.micro (Amazon Linux 2023) nommée `Web-VignesDuVal` dans Subnet-Public-A avec Auto-assign public IP activé, le SG-Web et le Role-EC2-S3-VignesDuVal.

Action (Installation) : Connexion SSH, installation de `httpd` , `php-mysqlnd` , `php` via `dnf` , et déploiement des fichiers WordPress.

Action (Configuration Web - Tentative 1) :

- Accès à `http://[IP-Publique-EC2]` : Succès



Ecran d'installation de WordPress

Obstacle 3 : Échec de connexion WordPress (Sélection de BDD).

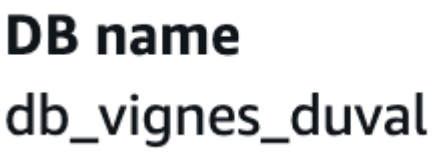
Erreur : `Impossible de sélectionner la base données db-vignes-duval`.

Analyse : Ce message est un faux négatif. Il indique que la connexion réseau (EC2 → RDS via SG-DB) et l'authentification ont réussi. L'échec se situe au niveau SQL : le serveur de base de données a été contacté, mais le schéma (base de données) nommé `db_vignes_duval` n'existait pas à l'intérieur.

Cause : Lors de la création de l'instance RDS, le champ "Initial database name" (caché dans "Additional configuration") n'a pas été renseigné.

Résolution : L'instance RDS a été supprimée. Une nouvelle instance `db-vignes-duval` a été créée, en spécifiant cette fois `db_vignes_duval` dans le champ "Initial database name".

Erreur rencontrée lors de l'interconnexion de WordPress et RDS



Nom de la base de donnée qu'il fallait remplir lors de la configuration initiale de RDS, même nom qu'il fallait saisir dans le formulaire de connexion WordPress à RDS

Action (Configuration Web - Tentative 2) :

- Récupération du nouvel Endpoint RDS (`db-vignes-duval.c9yi406qy74e.eu-west-3.rds.amazonaws.com`) et saisie dans l'installeur WordPress
- **Résultat** : Succès. L'installation s'est terminée.

Phase 5 : CDN (CloudFront) et Dépannage Post-Redémarrage

Action : Activation des "DNS hostnames" sur le VPC-VignesDuVal pour obtenir un nom DNS stable pour l'EC2.

Action : Redémarrage de l'instance EC2 pour appliquer le paramètre.

Obstacle 4 : Perte d'accès et redirection (IP vs DNS).

Problème : Le redémarrage de l'instance a entraîné l'assignation d'une nouvelle IP publique, rendant l'ancienne inaccessible. De plus, WordPress (ayant mémorisé l'ancienne IP) redirigeait en boucle vers l'adresse obsolète.

Résolution : Modification du fichier `wp-config.php` sur l'EC2 via SSH pour y définir statiquement les valeurs `WP_HOME` et `WP_SITEURL` avec le nouveau nom DNS public stable (ex: `ec2-...compute.amazonaws.com`).

```
define( 'WP_HOME', 'http://ec2-35-181-55-75.eu-west-3.compute.amazonaws.com' );
define( 'WP_SITEURL', 'http://ec2-35-181-55-75.eu-west-3.compute.amazonaws.com' );
```

Ajout de WP_HOME et WP_SITEURL dans wp-config.php

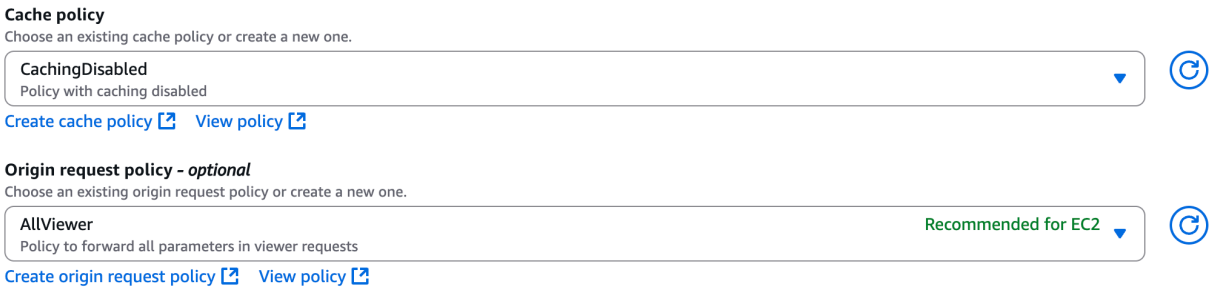
Obstacle 5 : `Operation timed out` en SSH.

Analyse : L'IP de la connexion 4G (partage de connexion) est dynamique et a changé. La règle My IP du SG-Web bloquait la nouvelle IP.

Résolution : Modification de la règle SSH (Port 22) du SG-Web de My IP à Anywhere-IPv4 (`0.0.0.0/0`). Cette configuration est acceptable pour un laboratoire car l'authentification est sécurisée par clé `.pem` (et non par mot de passe).

Action : Création de la distribution CloudFront, pointant vers le nom DNS public de l'EC2.

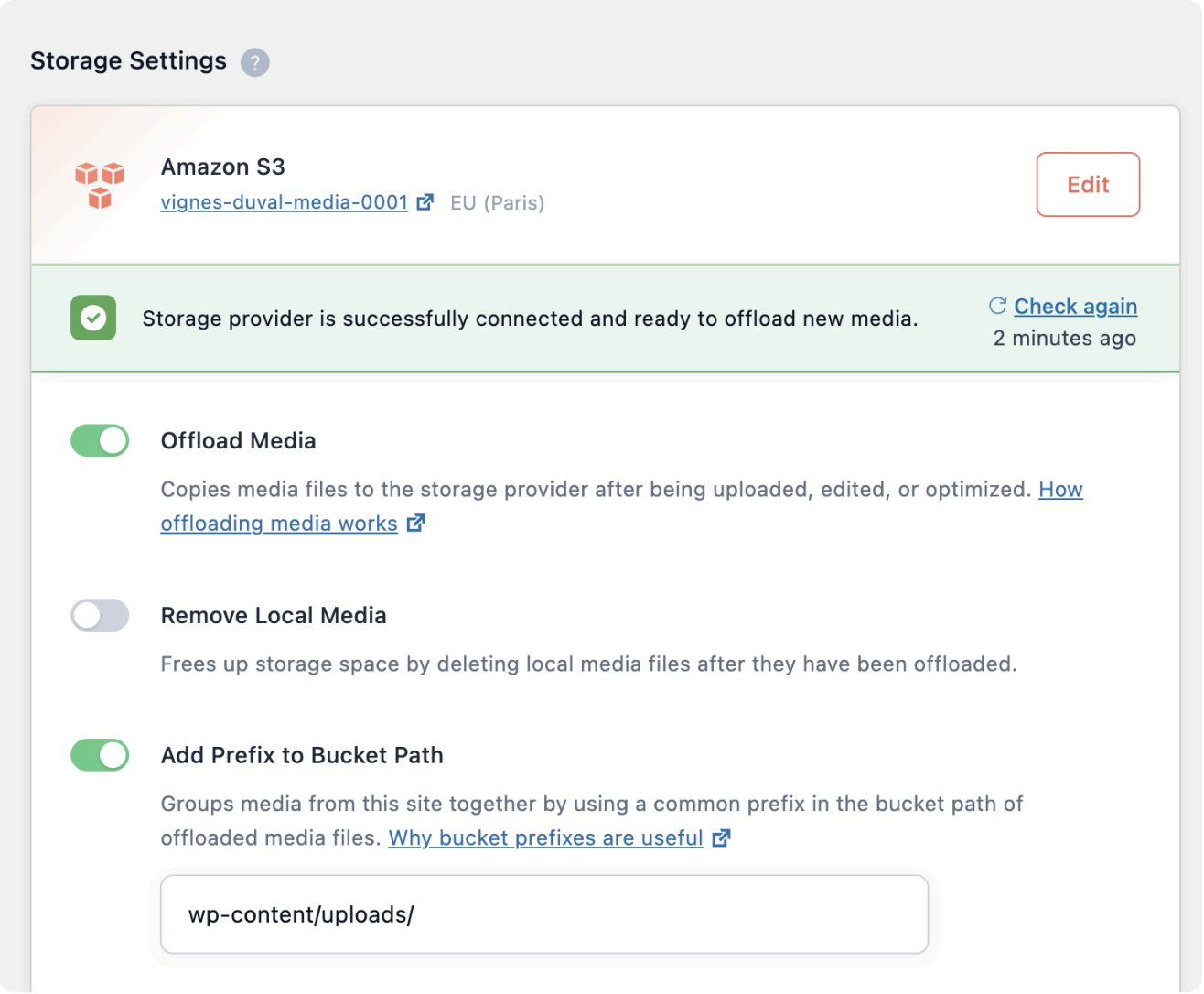
Configuration : Le comportement de cache par défaut a été modifié (après création) pour utiliser les policies `Managed-CachingDisabled` et `Managed-AllViewer` afin de garantir le bon fonctionnement du panneau `/wp-admin/` de WordPress.



Réglages Cache Behavior CloudFront pour WordPress

Phase 6 : Plugin S3 et Dépannage Final

Action : Installation du plugin "WP Offload Media Lite" sur WordPress afin de délivrer les médias uploadés sur WordPress, sur le bucket S3 dédié. Le plugin a correctement détecté le Rôle IAM (`Role-EC2-S3-VignesDuVal`) et le bucket S3.



Plugin WP Offload Media Lite opérationnel

Obstacle 6 : Échec de l'upload S3 (Routage).

Problème : L'upload d'images échouait silencieusement (le fichier n'apparaissait pas dans le bucket).

Analyse : L'instance EC2 se trouve dans le Subnet-Public-A (utilisant RTB-Public), mais le VPC Endpoint S3 n'était associé qu'à la RTB-Private. L'EC2 n'avait donc pas accès au "tunnel" privé vers S3.

Résolution : Modification de la configuration du VPC Endpoint S3 pour l'associer également à la table de routage RTB-Public.

Obstacle 7 : Échec persistant de l'upload S3 (Application).

Problème : L'upload échouait toujours.

Test 1 : Un upload manuel via le CLI (`aws s3 cp test-upload.txt s3:// ...`) depuis l'instance a réussi.

```
[ec2-user@ip-10-0-1-218 ~]$ echo "Ceci est un test" > test-upload.txt
[ec2-user@ip-10-0-1-218 ~]$ cat test-upload.txt
Ceci est un test
[ec2-user@ip-10-0-1-218 ~]$ aws s3 cp test-upload.txt s3://vignes-duval-media-0001/
Completed 17 Bytes/17 Bytes (238 Bytes/s) with 1 file(s) remainingupload: ./test-upload.txt to s3://vignes-duval-media-0001/test-upload.
txt
```

Test d'import de fichier vers S3 depuis le CLI pour écarter l'hypothèse d'un problème de permissions/réseau

Analyse (Test 1) : Ce succès a prouvé que l'infrastructure (IAM, Réseau, Endpoint) était correcte. Le problème était donc local au serveur, au niveau d'Apache/PHP.

Test 2 (Hypothèse) : SELinux. Une désactivation temporaire (`sudo setenforce 0`) n'a pas résolu le problème.

Test 3 (Diagnostic) : Les logs d'erreur `php-fpm` (trouvés via le panneau de diagnostic du plugin) ont révélé l'absence de bibliothèques de traitement d'image. Le diagnostic du plugin a confirmé : `PHP GD: Disabled` .

Analyse (Test 3) : L'échec n'était pas l'upload lui-même. WordPress échouait à créer les miniatures (resize) de l'image, ce qui interrompait tout le processus avant l'envoi vers S3.

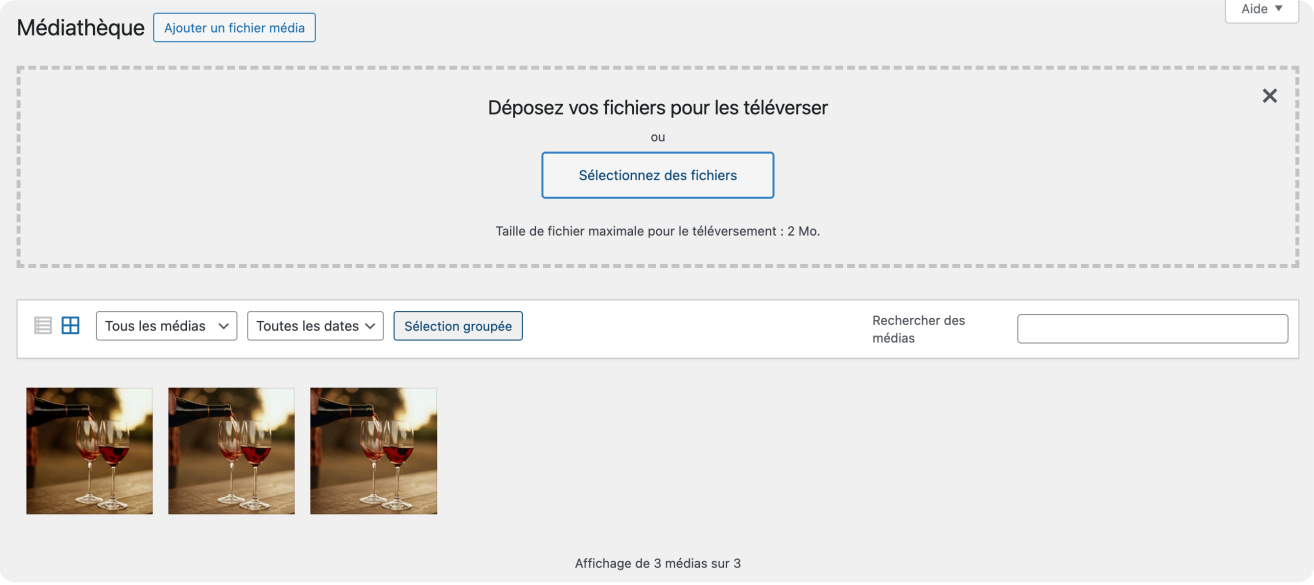
Résolution : Installation du module PHP manquant :

```
sudo dnf install -y php-gd
```

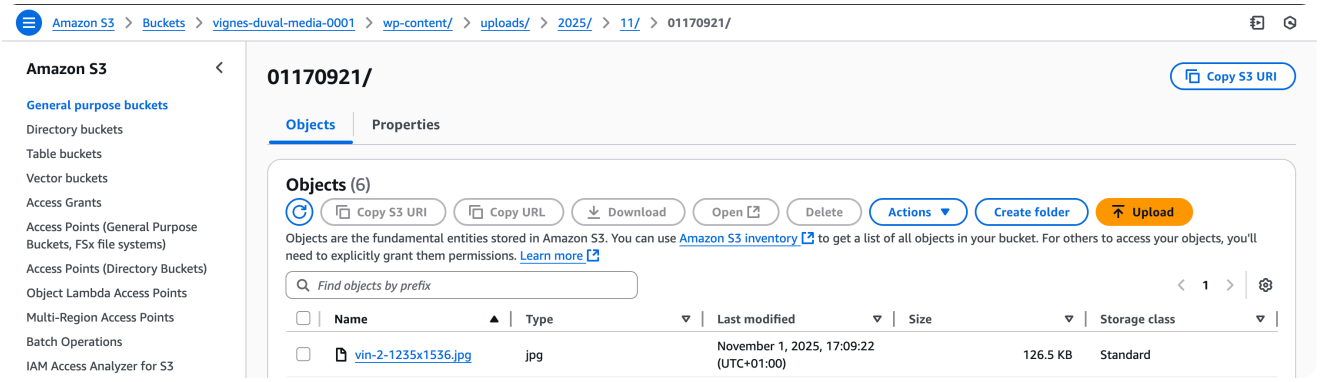
Puis redémarrage des services :

```
sudo systemctl restart httpd
sudo systemctl restart php-fpm
```

Résultat : Succès. L'upload d'une nouvelle image dans la médiathèque WordPress a entraîné sa copie immédiate dans le bucket S3.



Upload d'une ou plusieurs images dans la médiathèque WordPress



Fichier uploadé visible dans le bucket S3

Obstacle 8 : Limitation du plugin (Version "Lite").

Problème : L'onglet "Delivery Settings" du plugin ne permet pas de spécifier notre URL CloudFront pour la livraison (fonctionnalité "Pro" payante).

Résolution : L'architecture finale est un modèle hybride :

- Le plugin est configuré avec **Offload Media ON** (il agit comme un outil de backup/copie vers S3)
- Le plugin est configuré avec **Deliver Offloaded Media OFF**
- Le site continue de servir les images depuis l'EC2, qui sont ensuite mises en cache par CloudFront

---

6. Conclusion et Prochaines Étapes

Ce laboratoire a permis de valider avec succès le déploiement d'une architecture applicative WordPress **3-tiers découplée** sur AWS. Les fondations V1 sont fonctionnelles, sécurisées, et respectent les contraintes du Free Tier.

Plus important encore, le processus de dépannage a permis d'explorer en profondeur les interactions complexes entre les services AWS (VPC, RDS, EC2), la configuration du système d'exploitation (SELinux, modules PHP) et la configuration applicative (WordPress, plugins). Les obstacles rencontrés (exigences d'AZ pour RDS, routage des Endpoints, modules PHP manquants) ont été des points d'apprentissage critiques qui démontrent une compréhension allant au-delà de la simple implémentation.

L'architecture finale est fonctionnelle et robuste.

6.1. Nettoyage Impératif des Ressources

Pour éviter toute facturation, l'étape suivante est le décommissionnement complet de l'environnement dans l'ordre suivant :

1. **CloudFront** : Disable (Désactiver) la distribution. Attendre le déploiement. Delete (Supprimer) la distribution.

2. **EC2** : Terminate (Mettre fin) l'instance `Web-VignesDuVal`.
3. **RDS** : Delete (Supprimer) la base de données `db-vignes-duval`. (Nécessite de décocher "Delete protection" et de refuser l'instantané final).
4. **S3** : Empty (Vider) le bucket `vignes-duval-media-0001`, puis Delete (Supprimer) le bucket.
5. **VPC** :
   - Delete (Supprimer) le `vpce-s3-VignesDuVal`
   - Delete VPC (Supprimer le VPC) `VPC-VignesDuVal`. (Ceci supprimera les Subnets, Route Tables, et l'IGW associés)
6. **IAM** : Delete (Supprimer) le Rôle `Role-EC2-S3-VignesDuVal`.
7. **Key Pair** : Supprimer la clé `vignes-duval-key` de la console EC2.

Fin du rapport