

Concevoir et déployer sur AWS une application web de gestion (métier au choix : tâches, notes, produits, réservations, etc.)

- Stockage principal : Amazon RDS MySQL.
- Fichiers (images/vidéos/documents) : Amazon S3 (bucket privé).

Exigences techniques

- Infrastructure as Code : Terraform
- Réseau : VPC en 2 zones de disponibilité (us-east-1a et us-east-1b) avec :
 - 2 subnets publics (un par AZ) et 2 subnets privés (un par AZ).
 - L'EC2 est déployée dans le subnet public de us-east-1a (IP publique, route vers IGW).
 - Les 2 subnets privés sont utilisés par RDS Multi-AZ.
- Compute : application conteneurisée Docker déployée sur EC2 (port 80 ou 8080).
- Données (RDS) : instance sans IP publique dans les subnets privés, accessible uniquement depuis l'EC2 via Security Groups (3306).
- Fichiers (S3) : bucket privé (Block Public Access ON), accès via SDK et/ou URLs présignées.
- Sécurité : IAM Instance Profile = LabInstanceProfile sur l'EC2 (aucune clé dans le code) ; règles SG minimales (HTTP depuis votre IP ; SSH 22 optionnel ; DB autorisée uniquement depuis le SG web).

Livrables

- Code source : Terraform (main.tf, variables.tf, outputs.tf, ...), code de l'application, Dockerfile, README (déploiement, tests, destruction).
- Schéma d'architecture : VPC, subnets (public/privé), IGW, routes, Security Groups, EC2, RDS (Multi-AZ), S3.
- Démonstration : navigation web montrant les fonctionnalités clés (ex. CRUD métier, upload vers S3, téléchargement, etc.).

Critères d'évaluation (résumé)

- Infra & réseau : VPC, subnets, routes, SG corrects + reproductibilité Terraform.
- RDS & S3 privés : RDS sans IP publique (Multi-AZ 2 zones, DB Subnet Group privé) ; S3 non public.
- Fonctionnel & conteneurisation : app web accessible (80/8080), besoins métiers pertinents, image Docker propre.
- Sécurité / coûts : pas de secrets en dur, tailles raisonnables, ports minimaux ouverts.
- Qualité du README & tests : clarté, commandes de vérification, etc...

Contraintes Lab : région us-east-1, aucun bucket S3 public, destruction obligatoire des ressources à la fin (vider S3 puis terraform destroy).