

Informations

Durée : 3 jours (21h.)

Tarif* : Nous consulter

Réf : BDAW

Niveau : Moyen

intra

Mise à jour le 16/09/25

*tarif valable jusqu'au 31/12/2025

Prochaines sessions

Contactez-nous pour connaître nos futures sessions.

Pré-requis

- Connaissances de base en administration systèmes et bases de données
- Notions réseau et cloud (VPC, IAM)
- Avoir suivi la formation AWS Cloud Practitioner Essentials (recommandé)

Objectifs

Objectifs pédagogiques :

- Comprendre les différentes solutions de stockage AWS et leurs cas d'usage
- Maîtriser le stockage objet (S3), bloc (EBS) et archivage (Glacier)
- Administrer une base relationnelle avec RDS
- Mettre en place une base NoSQL avec DynamoDB
- Appliquer les bonnes pratiques de sauvegarde, sécurité et optimisation des coûts

Objectifs opérationnels :

- Concevoir, déployer et administrer des solutions de stockage et bases de données AWS adaptées aux besoins (objet / bloc / relationnel / NoSQL), assurer la sécurité, la sauvegarde et la résilience des données, optimiser les coûts de stockage, garantir la performance et la fiabilité des bases de données dans un environnement AWS.

Programme

Jour 1 - Stockage sur AWS

Introduction aux solutions de stockage AWS.

Amazon S3 : concepts, classes de stockage, politiques, versioning, cycle de vie

Amazon EBS : volumes, snapshots, attachement aux instances EC2

Amazon Glacier : archivage à faible coût

Atelier : créer un bucket S3, configurer un cycle de vie et attacher un volume EBS

Jour 2 - Bases relationnelles avec RDS

Présentation d'Amazon RDS et Aurora

Déploiement d'une base relationnelle multi-AZ

Backups, snapshots et restauration automatique

Monitoring et optimisation des performances

Atelier : déployer une base MySQL/PostgreSQL sur RDS et connecter une application

Jour 3 - Bases NoSQL avec DynamoDB et bonnes pratiques

Introduction aux bases NoSQL et cas d'usage

Amazon DynamoDB : tables, clés primaires, throughput, index secondaires

Sécurité et IAM pour DynamoDB

Sauvegarde, restauration et intégration avec d'autres services (Lambda, API Gateway)

Étude de cas pratique : conception d'une application serverless avec DynamoDB

Bonnes pratiques de stockage et de gestion des coûts