Introduction à la sécurité AWS

Importance : La sécurité dans le cloud est cruciale en raison de sa complexité, de sa dynamique (ressources provisionnées/désallouées à la demande) et de son échelle mondiale (25 régions, 100 zones de disponibilité).

Défi : Gérer la sécurité dans un environnement en constante évolution, avec des réglementations variables selon les pays et les secteurs (ex : RGPD, HIPAA, PCI DSS).

1. Modèle de responsabilité partagée (Shared Responsibility Model)

Principe : AWS et le client se partagent les responsabilités de sécurité, mais la responsabilité ultime (accountability) reste toujours au client, même en cas de faille côté AWS.

- AWS : Sécurise l'infrastructure du cloud (matériel, réseau, virtualisation, centres de données).
- Le Client : Sécurise ce qui est dans le cloud (données, applications, configurations, accès utilisateurs).

Variations selon les modèles de service :

- IaaS (ex : EC2) : Le client gère l'OS, les applications, les données et la configuration réseau.
- PaaS (ex : RDS, Lambda) : AWS gère l'OS/runtime ; le client se concentre sur les applications et les données.
- SaaS (ex : Amazon Connect) : AWS gère toute la pile ; le client ne gère que ses données et l'accès.

Exemple concret avec les services de calcul :

2. Infrastructure globale AWS et considérations de sécurité

Régions :

Critères de choix :

1. Souveraineté des données (ex : données santé aux États-Unis → région conforme HIPAA).
2. Latence (proximité des utilisateurs finaux).
3. Disponibilité des services (certains services ne sont pas disponibles partout).
4. Coût (variations tarifaires selon les régions).
5. Résilience (architecture multi-régions pour les catastrophes naturelles).

Zones de disponibilité (AZ) :

Sites isolés dans une région, avec redondance (alimentation, réseau, refroidissement).

Bonnes pratiques :

- Répartir les ressources sur plusieurs AZ pour éviter les pannes localisées.
- Éviter une dispersion excessive (coût et complexité accrus).
- Privilégier les services gérés (ex : Lambda) qui offrent une résilience multi-AZ native.

Edge Locations :

Points de présence pour CloudFront (CDN) et AWS Shield (protection DDoS).

Avantages :

- Réduction de la latence (cache de contenu près des utilisateurs).
- Absorption des attaques DDoS (trafic filtré avant d'atteindre l'application).

3. Visibilité et contrôle

Défi : Dans le cloud, l'environnement est dynamique (ressources éphémères, échelle massive).

Solutions :

Outils AWS :

- AWS Config : Audit des configurations des ressources.
- CloudTrail : Journalisation des activités (qui a fait quoi ?).
- CloudWatch : Surveillance des métriques et logs en temps réel.
- Security Hub : Tableau de bord centralisé pour les alertes de sécurité.
- GuardDuty : Détection des menaces via IA et analyse des comportements.
- Detective : Investigation des incidents avec visualisations interactives.

- Automatisation : Intégrer la sécurité dans les pipelines DevOps (DevSecOps).

4. Conformité (Compliance)

Enjeux : Respecter les réglementations sectorielles (ex : PCI DSS pour les paiements, HIPAA pour la santé) et les bonnes pratiques (ex : AWS Well-Architected Framework). Responsabilité client : AWS fournit des outils (ex : AWS Artifact pour les rapports de conformité), mais c'est au client de configurer ses ressources de manière conforme.

Exemple : Même si AWS est certifié SOC2, vos applications ne le sont pas automatiquement !

5. Vitesse d'innovation et sécurité

Avantages : Nouveaux services AWS (ex : IA, IoT) offrent des fonctionnalités de sécurité avancées.

Risques : Adoption trop rapide sans évaluation sécurité → vulnérabilités. Complexité accrue avec chaque nouveau service (configurations à maîtriser).

Bonnes pratiques : Évaluer les risques avant d'adopter un nouveau service. Automatiser les tests de sécurité (ex : scans de vulnérabilités dans les pipelines CI/CD).

6. Bonnes pratiques générales de sécurité AWS

Comprendre le modèle de responsabilité partagée : Savoir ce qui relève d'AWS et ce qui vous incombe. Utiliser les services de sécurité AWS :

- Chiffrement : KMS pour les données, TLS pour les communications.
- Gestion des accès : IAM (principe du moindre privilège), MFA.
- Protection des données : Sauvegardes automatiques, classification des données.

Automatiser la sécurité :

- Intégrer des outils comme AWS Inspector (scans de vulnérabilités) dans les processus DevOps.

Surveiller et auditer :

- Centraliser les logs avec CloudTrail + CloudWatch.
- Configurer des alertes pour les activités suspectes.

Former les équipes :

- Culture de sécurité partagée (développeurs, ops, sécurité).