AYI NEDJIMI Consultants

# Sécurité AWS : Guide Complet Hardening Compte et Services

Catégorie : Cloud Security  Lecture : 10 min  Publié le : 12/03/2026  Auteur : Ayi NEDJIMI

*Guide complet de durcissement AWS : sécurisation du compte root, politiques IAM avancées, hardening S3 EC2 RDS, monitoring CloudTrail et GuardDuty.*

La migration vers Amazon Web Services représente une opportunité majeure pour les organisations qui cherchent à gagner en agilité et en scalabilité. Cependant, cette transition s'accompagne de responsabilités de sécurité considérables que trop d'équipes sous-estiment encore en 2026. Le modèle de responsabilité partagée d'AWS délimite clairement les frontières : AWS sécurise l'infrastructure physique, mais la configuration des services, la gestion des accès et la protection des données relèvent entièrement du client. Dans notre pratique de conseil en cybersécurité cloud, nous constatons que plus de soixante-dix pour cent des incidents AWS proviennent d'erreurs de configuration, et non de vulnérabilités intrinsèques à la plateforme. Ce guide exhaustif vous accompagne dans le durcissement méthodique de votre compte AWS, depuis la sécurisation du compte racine jusqu'au monitoring continu, en passant par les stratégies IAM avancées et la protection des services les plus exposés.

## Résumé exécutif

Guide complet de durcissement d'un compte AWS : configuration du compte racine, politiques IAM avancées, sécurisation des services critiques S3, EC2, RDS et monitoring continu avec CloudTrail et GuardDuty. Approche conforme aux CIS Benchmarks AWS. La migration vers le cloud transforme radicalement les paradigmes de sécurité : responsabilité partagée, identités éphémères, surfaces d'attaque distribuées et configurations complexes multiplient les vecteurs de compromission. Les équipes sécurité doivent adapter leurs compétences et leurs outils à ces nouveaux environnements tout en maintenant une visibilité complète sur les ressources déployées. Ce guide technique détaille les approches éprouvées en production, les pièges courants à éviter et les stratégies de durcissement prioritaires pour sécuriser efficacement vos workloads cloud en 2026. Chaque recommandation est issue de retours d'expérience concrets en environnement entreprise.

**Retour d'expérience :** lors d'un audit récent d'une entreprise du secteur bancaire, nous avons identifié 47 buckets S3 avec des ACL mal configurées, dont 12 accessibles publiquement contenant des données clients sensibles. Le durcissement complet du compte a réduit la surface d'attaque de 83 % en seulement trois semaines de travail. Les coûts de remédiation post-incident auraient été dix fois supérieurs à l'investissement préventif dans le hardening.

## Le modèle de responsabilité partagée AWS en pratique

Le modèle de responsabilité partagée constitue le fondement de toute stratégie de sécurité sur AWS. **AWS est responsable de la sécurité du cloud** (infrastructure physique, réseau backbone, hyperviseur), tandis que le client est responsable de la **sécurité dans le cloud** (configuration des services, gestion des identités, chiffrement des données, pare-feu applicatifs). Cette distinction est capitale car elle détermine qui doit agir en cas de compromission. Pour les services managés comme RDS ou Lambda, la frontière se déplace : AWS prend en charge une plus grande partie de la pile, mais le client reste responsable de la configuration, des accès et des données. Consultez CIS Benchmarks pour les détails officiels de ce modèle. L'erreur la plus courante est de considérer que le passage au cloud exonère de toute responsabilité sécuritaire, ce qui conduit à des configurations par défaut dangereusement permissives.

En complément, il est essentiel de comprendre que chaque service AWS dispose de ses propres mécanismes de sécurité. *Amazon VPC* gère l'isolation réseau, *AWS IAM* contrôle les accès, *AWS KMS* gère le chiffrement, et *Amazon CloudTrail* assure la traçabilité. L'orchestration cohérente de ces services forme la base d'une posture de sécurité robuste. Les équipes qui maîtrisent les interactions entre ces composants disposent d'un avantage considérable face aux menaces ciblant les environnements cloud. Une approche intégrée permet de détecter rapidement les anomalies et de répondre efficacement aux incidents, là où une configuration en silo laisse des angles morts exploitables par les attaquants.

## Sécurisation du compte racine et AWS Organizations

Le compte racine AWS possède un accès illimité à toutes les ressources et ne peut pas être restreint par des politiques IAM. Sa compromission est le scénario catastrophe par excellence. La première action de durcissement consiste à activer un **MFA matériel** (YubiKey ou clé FIDO2) sur ce compte. Les MFA virtuels (applications TOTP) sont acceptables mais offrent une résistance moindre aux attaques de phishing avancées. Ensuite, supprimez toutes les clés d'accès programmatiques associées au compte root. Ces clés sont rarement nécessaires et constituent un vecteur d'attaque privilégié. Configurez une adresse email dédiée et supervisée pour le compte root, distincte des adresses personnelles ou génériques de l'entreprise.

**AWS Organizations** permet de structurer plusieurs comptes AWS sous une hiérarchie d'unités organisationnelles (OU). Les *Service Control Policies* (SCP) appliquées aux OU restreignent les actions possibles dans les comptes membres, même pour les administrateurs locaux. Par exemple, une SCP peut interdire la désactivation de CloudTrail, empêcher l'utilisation de régions non autorisées ou bloquer la création de ressources sans chiffrement. Cette gouvernance centralisée est indispensable pour les organisations qui gèrent plus de cinq comptes AWS. Nous recommandons systématiquement la séparation des comptes par environnement (production, staging, développement) et par domaine fonctionnel (réseau, sécurité, applications). Retrouvez d'autres stratégies de sécurisation cloud dans notre article sur Container Security Docker Runtime Protection.

## Politiques IAM avancées et moindre privilège

La gestion des identités et des accès est le pilier central de la sécurité AWS. Le principe du moindre privilège exige que chaque entité (utilisateur, rôle, service) ne dispose que des permissions strictement nécessaires à ses fonctions. En pratique, cela implique d'abandonner les politiques managées larges comme `AdministratorAccess` ou `PowerUserAccess` au profit de politiques personnalisées granulaires. **IAM Access Analyzer** aide à identifier les permissions inutilisées et à générer des politiques basées sur l'activité réelle. Les conditions IAM permettent de restreindre les accès par IP source, plage horaire, MFA obligatoire ou tag de ressource.

Les **rôles IAM** sont préférables aux utilisateurs pour les applications et services. Ils fournissent des credentials temporaires via STS (Security Token Service), éliminant le risque de clés d'accès exposées dans le code ou les variables d'environnement. Pour les accès inter-comptes, les rôles avec relation de confiance croisée remplacent avantageusement le partage de clés d'accès. L'utilisation de *IAM Identity Center* (anciennement SSO) centralise l'authentification et simplifie la gestion des accès multi-comptes. Pour approfondir la gestion des accès cloud, consultez notre article dédié Securite Llm Agents Guide Pratique. La rotation automatique des credentials restants et l'audit régulier des permissions complètent cette stratégie. Les référentiels CIS recommandent des vérifications trimestrielles, mais un audit mensuel est préférable dans les environnements à haute sensibilité.

| Service AWS | Risque principal | Mesure de durcissement | Priorité |
|---|---|---|---|
| S3 | Exposition publique de données | Block Public Access, chiffrement SSE-KMS, bucket policies restrictives | Critique |
| EC2 | Security groups permissifs | Restreindre les ports, utiliser des NACL, activer VPC Flow Logs | Critique |
| RDS | Accès non chiffré | Chiffrement au repos et en transit, sous-réseaux privés, IAM auth | Haute |
| Lambda | Rôles trop permissifs | Moindre privilège par fonction, variables d'environnement chiffrées | Haute |
| IAM | Escalade de privilèges | SCP restrictives, Access Analyzer, MFA obligatoire | Critique |
| CloudTrail | Désactivation par attaquant | Trail multi-région, SCP anti-suppression, alarmes CloudWatch | Critique |

## Durcissement de S3, EC2 et des services critiques

Amazon S3 reste le service le plus fréquemment impliqué dans les fuites de données cloud. Le paramètre **S3 Block Public Access** doit être activé au niveau du compte et de chaque bucket. Les bucket policies doivent explicitement refuser les accès non authentifiés et imposer le chiffrement via `aws:SecureTransport`. Le chiffrement côté serveur avec des clés gérées par KMS (SSE-KMS) offre un contrôle fin sur les accès aux données via les politiques de clés. La

journalisation des accès S3 vers un bucket dédié dans un compte de sécurité séparé garantit l'intégrité des logs d'audit. L'activation du versioning protège contre la suppression accidentelle ou malveillante.

Pour EC2, les **security groups** doivent suivre une logique de liste blanche stricte. Aucune règle entrante ne devrait autoriser 0.0.0.0/0 sauf pour les ports HTTP/HTTPS sur les load balancers publics. L'utilisation de *Systems Manager Session Manager* remplace SSH avec authentification IAM et journalisation complète. Les AMI doivent être durcies selon les benchmarks du Azure Defender for Cloud, avec suppression des packages inutiles, désactivation des services non essentiels et configuration du pare-feu local. L'utilisation d'Instance Metadata Service v2 (IMDSv2) est obligatoire pour contrer les attaques SSRF qui exploitent le service de métadonnées pour récupérer des credentials. Pour en savoir plus sur les escalades de privilèges dans AWS, consultez notre analyse sur Gcp Security Bonnes Pratiques Audit 2026.

## Monitoring continu avec CloudTrail, GuardDuty et Security Hub

**AWS CloudTrail** enregistre toutes les actions API dans votre compte AWS. La configuration doit inclure un trail multi-région couvrant les événements de gestion et les événements de données pour S3 et Lambda. Les logs doivent être centralisés dans un bucket S3 dédié avec chiffrement KMS, validation d'intégrité des fichiers et politiques de rétention adaptées aux exigences réglementaires. L'envoi vers CloudWatch Logs permet de créer des alarmes en temps réel sur les événements critiques : connexion root, modification des trails, changement de politique IAM, création de clés d'accès.

**Amazon GuardDuty** utilise le machine learning et la threat intelligence pour détecter les comportements malveillants. Son activation ne nécessite aucune configuration complexe et couvre la détection de minage de cryptomonnaies, les communications avec des serveurs C2, les accès anormaux aux API et la compromission d'instances EC2. **AWS Security Hub** agrège les findings de GuardDuty, Inspector, Macie et des outils tiers en un tableau de bord unifié. Il vérifie automatiquement la conformité avec les CIS AWS Foundations Benchmark et les standards PCI DSS. L'intégration avec AWS Security renforce la couverture de détection avec des feeds de menaces supplémentaires. Les organisations matures intègrent ces alertes dans leur SIEM et leur processus de réponse aux incidents, comme décrit dans notre guide Serverless Security Lambda Functions Cloud.

**Mon avis :** la combinaison CloudTrail + GuardDuty + Security Hub constitue le trio minimal indispensable pour toute organisation utilisant AWS en production. Le coût est négligeable par rapport au budget cloud global, et la valeur apportée en termes de détection et de conformité est considérable. Les équipes qui négligent ces fondamentaux se retrouvent systématiquement en difficulté lors d'un incident. J'observe encore trop d'entreprises qui économisent sur le monitoring pour découvrir ensuite que le coût d'un incident non détecté est cent fois supérieur.

## Comment sécuriser un compte AWS racine efficacement ?

La sécurisation du compte root AWS repose sur une série de mesures complémentaires qui forment une défense en profondeur. Premièrement, activez un MFA matériel de type FIDO2 ou YubiKey, qui offre une résistance supérieure au phishing par rapport aux applications TOTP. Deuxièmement, supprimez toutes les clés d'accès associées au compte root via la console IAM, car ces clés sont rarement nécessaires et constituent un vecteur d'attaque privilégié. Troisièmement, configurez une alerte CloudWatch sur l'événement `ConsoleLogin` filtré sur le compte root pour être notifié immédiatement de toute connexion. Quatrièmement, utilisez AWS Organizations avec des SCP qui bloquent explicitement les actions destructrices même depuis le compte root des comptes membres. Consultez CIS Benchmarks pour les meilleures pratiques officielles. L'ensemble de ces mesures réduit drastiquement le risque de compromission du compte le plus sensible de votre infrastructure AWS. Un audit régulier des paramètres du compte root vérifie que ces mesures restent en place au fil du temps, car des dérives involontaires peuvent réintroduire des risques. Retrouvez aussi nos recommandations sur Oauth Oidc Abus Consent Securite pour une vue complète de la sécurisation cloud.

## Pourquoi le principe du moindre privilège est-il vital sur AWS ?

Le principe du moindre privilège constitue la pierre angulaire de toute architecture sécurisée sur AWS, et sa violation est responsable de la majorité des escalades de privilèges observées lors d'audits. Lorsqu'un utilisateur ou un service dispose de permissions excessives, une compromission de ses credentials donne à l'attaquant un accès disproportionné à l'infrastructure. Sur AWS, les politiques IAM permettent une granularité extrême, jusqu'au niveau de l'action individuelle et de la ressource spécifique, avec des conditions contextuelles. IAM Access Analyzer analyse les journaux CloudTrail pour identifier les permissions accordées mais jamais utilisées, permettant de resserrer les politiques sans impacter les opérations. Les permission boundaries ajoutent une couche de restriction supplémentaire en définissant les limites maximales de permissions qu'un rôle peut accorder. Pour comprendre les techniques d'escalade exploitant les permissions excessives, consultez notre article sur Gcp Security Bonnes Pratiques Audit 2026. La mise en oeuvre rigoureuse du moindre privilège demande un investissement initial en temps d'analyse mais réduit considérablement la surface d'attaque et simplifie la réponse aux incidents.

## Quelles sont les erreurs de configuration AWS les plus dangereuses ?

Les erreurs de configuration AWS les plus critiques forment un catalogue tristement récurrent dans nos rapports d'audit. Les **buckets S3 publics** restent en tête de liste malgré les avertissements répétés d'AWS. Les **security groups avec 0.0.0.0/0** sur les ports SSH (22) ou RDP (3389) exposent les instances à des attaques par force brute automatisées. L'absence de chiffrement sur RDS, EBS et S3 laisse les données vulnérables en cas d'accès non autorisé. Les clés d'accès IAM codées en dur dans les repositories Git constituent un autre vecteur classique, car les scanners automatiques détectent ces secrets en quelques minutes après un push public.

La désactivation ou l'absence de CloudTrail empêche toute investigation forensique post-incident. Le recours à IMDSv1 plutôt qu'IMDSv2 sur les instances EC2 permet les attaques SSRF exploitant le service de métadonnées pour voler des credentials. Enfin, l'utilisation de politiques IAM trop larges combinée à l'absence de MFA crée les conditions idéales pour une compromission complète du compte. Le Azure Defender for Cloud fournit des benchmarks détaillés pour évaluer et corriger ces configurations.

**À retenir :** le durcissement d'un compte AWS repose sur cinq piliers fondamentaux : sécurisation du compte root avec MFA matériel, politiques IAM granulaires suivant le moindre privilège, chiffrement systématique des données au repos et en transit, monitoring continu avec CloudTrail, GuardDuty et Security Hub, et gouvernance centralisée via AWS Organizations et SCP. La mise en oeuvre méthodique de ces mesures réduit la surface d'attaque de plus de quatre-vingts pour cent.

Votre compte AWS root est-il protégé par un MFA matériel, ou reposez-vous encore sur une application TOTP vulnérable au phishing avancé ?

**Sources et références :** CISA · Cloud Security Alliance

## Perspectives et prochaines étapes

Le durcissement d'un compte AWS est un processus continu qui doit évoluer avec la croissance de votre infrastructure et l'apparition de nouveaux services. Les organisations les plus matures automatisent la vérification de conformité via AWS Config Rules et des pipelines de remédiation automatique. L'adoption de l'Infrastructure as Code avec Terraform ou CloudFormation permet de garantir que chaque déploiement respecte les standards de sécurité définis. La prochaine étape logique consiste à intégrer ces contrôles dans votre pipeline CI/CD pour une approche DevSecOps complète, sujet que nous abordons en détail dans notre guide dédié. Le paysage des menaces cloud évolue rapidement, et seule une posture de sécurité proactive et automatisée peut maintenir un niveau de protection adéquat face à des attaquants de plus en plus sophistiqués.

**Ayi NEDJIMI Consultants** — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr