Amazon CloudFront

---

1. Amazon CloudFront (Le concept de base)

Le contexte et le problème

Imaginez que votre site web et vos vidéos soient hébergés dans un bucket S3 à Paris. Un utilisateur à Tokyo clique sur votre site. Sa requête doit traverser la Terre entière via des dizaines de câbles sous-marins pour récupérer les fichiers à Paris, puis faire le chemin inverse. Résultat : le site est extrêmement lent pour lui (**latence élevée**).

À quoi ça sert ?

CloudFront est le **CDN (Content Delivery Network)** d'AWS. C'est un réseau mondial de serveurs de cache (appelés **Edge Locations** ou emplacements réseaux) répartis partout dans le monde.

Lorsqu'un utilisateur à Tokyo demande une image, CloudFront va la chercher une première fois à Paris (l'**Origine**), la livre à l'utilisateur, et en garde une copie (en **cache**) à Tokyo. Le prochain utilisateur japonais obtiendra l'image instantanément depuis le serveur local de Tokyo.

---

2. CloudFront Distribution (La Distribution)

Le contexte et le problème

Lorsque vous décidez d'utiliser CloudFront, AWS ne sait pas automatiquement quel site ou quel bucket il doit mettre en cache, ni comment il doit se comporter (faut-il forcer le HTTPS ? combien de temps garder les fichiers en mémoire ?).

À quoi ça sert ?

Une **Distribution** est tout simplement la configuration de votre service CloudFront pour un site ou une application donnée. C'est l'entité que vous créez dans la console AWS pour activer le CDN.

Ce qu'il faut retenir pour l'examen

Lorsque vous créez une distribution, vous devez lui définir :

- **L'Origine :** L'adresse du serveur d'où proviennent les données originales (ex: un bucket S3, un Load Balancer, ou un serveur externe).
- **Les Behaviors (Comportements) :** Les règles de mise en cache. Par exemple : "*Pour tous les fichiers dans* `/images/*`*, garde le cache pendant 5 jours et force le protocole HTTPS*".

- **Un nom de domaine :** CloudFront vous génère un lien (ex: `d1234.cloudfront.net`) que vous pouvez masquer derrière votre propre nom de domaine (ex: `cdn.mon-site.com`).

---

3. Origin Access Control - OAC (Contrôle d'accès à l'origine)

Le contexte et le problème

Par définition, vous voulez que vos utilisateurs passent **uniquement** par CloudFront pour bénéficier du cache et des protections de sécurité (comme AWS WAF).

Si votre bucket S3 est complètement public, un utilisateur malin pourrait contourner CloudFront, récupérer l'URL directe de S3, et télécharger vos fichiers en masse. Cela saturerait votre bucket S3 et ferait grimper votre facture, tout en contournant vos règles de sécurité.

À quoi ça sert ?

OAC (Origin Access Control) est le mécanisme de sécurité moderne qui permet de **verrouiller votre bucket S3** pour qu'il n'accepte *que* les requêtes provenant de votre distribution CloudFront spécifique. Le bucket S3 reste 100 % privé pour le reste du monde.

Ce qu'il faut retenir pour l'examen

- **Signature AWS (SigV4) :** À chaque fois qu'un utilisateur demande un fichier, CloudFront va signer la requête en tâche de fond en utilisant le protocole de sécurité d'AWS (Signature Version 4) avant de l'envoyer à S3. S3 vérifie cette signature et sait que la demande est légitime.
- **La configuration en deux étapes :** 1. Vous activez OAC dans la configuration de votre distribution CloudFront. 2. Vous devez modifier la **Bucket Policy** de votre S3 pour y ajouter une règle qui dit : "*J'autorise uniquement l'effet* `s3:GetObject` *si la requête provient de la distribution CloudFront XYZ*".
- **Le piège de l'historique (OAI vs OAC) :** Dans le passé, AWS utilisait **OAI** (Origin Access Identity). L'examen SAA-C03 privilégie désormais **OAC** car il est plus sécurisé, prend en charge toutes les régions AWS, gère nativement le chiffrement SSE-KMS, et supporte les requêtes HTTP `POST` ou `PUT`.

---

4. CloudFront VPC Origins

Le contexte et le problème

Pendant longtemps, pour que CloudFront puisse aller chercher du contenu sur une application située dans votre réseau privé (derrière un Application Load Balancer ou sur une instance EC2), cette application devait obligatoirement avoir une adresse IP publique ou être exposée sur Internet. C'était un problème pour les architectures d'entreprise ultra-sécurisées.

À quoi ça sert ? (*What for?*)

Les **VPC Origins** permettent à CloudFront de se connecter **directement et de manière privée** à des ressources situées à l'intérieur de votre VPC, sans qu'elles aient besoin d'être exposées sur l'Internet public.

Ce qu'il faut retenir pour l'examen

- **Plus besoin d'IP publique :** Vos ALB (Application Load Balancers) ou vos instances EC2 peuvent rester totalement privés dans votre VPC.
- **Sécurité renforcée :** Le trafic entre les serveurs CloudFront et votre VPC transite via des interfaces réseau gérées par AWS éliminant le besoin de configurer des règles de pare-feu complexes sur Internet.

---

5. CloudFront Geo-restrictions (Géo-restrictions)

Le contexte et le problème

Vous lancez une plateforme de streaming vidéo, mais vous n'avez les droits de diffusion pour un film qu'en France. Si un utilisateur américain essaie de le regarder, vous êtes dans l'illégalité. À l'inverse, vous subissez une attaque informatique (DDoS) massive provenant d'un pays spécifique et vous voulez couper l'accès immédiatement.

À quoi ça sert ?

CloudFront regarde l'adresse IP de l'utilisateur, en déduit son pays, et bloque ou autorise l'accès avant même que la requête n'arrive à votre serveur.

Ce qu'il faut retenir pour l'examen

- **Allow List (Liste blanche) :** Vous spécifiez les seuls pays autorisés à voir votre contenu. Tous les autres sont bloqués.
- **Block List (Liste noire) :** Tout le monde a accès, sauf les pays spécifiques que vous listez.

Si vous avez besoin d'un contrôle plus fin (bloquer par ville ou par code postal, ou selon d'autres critères avancés), CloudFront seul ne suffit pas : l'examen attendra que vous coupliez CloudFront avec **AWS WAF** (Web Application Firewall).

---

6. Cache Invalidation (Invalidation du cache)

Le contexte et le problème

Par défaut, CloudFront garde vos fichiers en mémoire pendant une certaine durée (définie par le **TTL** - *Time To Live*, généralement 24 heures). Mais imaginez que vous veniez de corriger un bug critique dans votre fichier `script.js` ou que vous ayez modifié le logo de votre entreprise. Si vous attendez que le TTL expire, vos utilisateurs continueront de voir l'ancienne version buggée pendant 24 heures. C'est *indésirable*.

À quoi ça sert ?

L'**invalidation** est une commande d'urgence qui dit à toutes les Edge Locations de CloudFront dans le monde : "*Supprimez immédiatement votre copie locale de ce fichier, elle n'est plus bonne. Allez chercher la nouvelle version sur l'origine au prochain clic.*"

Ce qu'il faut retenir pour l'examen

- **Forcer le rafraîchissement (Force cache refresh) :** C'est exactement ce que fait l'invalidation.
- **L'utilisation des chemins (Paths) :** Vous n'êtes pas obligé d'invalider tout votre site. Vous pouvez cibler précisément :
  - Un fichier spécifique : `/images/logo.png`
  - Tout un dossier via l'astérisque : `/images/*`
  - Tout le site d'un coup (très lourd et potentiellement payant si abusé) : `/*`

Revision #3
Created 2026-06-03 20:33:27 UTC by Victor
Updated 2026-06-03 20:57:42 UTC by Victor