Mise en place de garde-fous et surveillance des URL présignées

AWS Conseils prescriptifs

AWS Conseils prescriptifs: Mise en place de garde-fous et surveillance des URL présignées

Table of Contents

Mise en place de garde-corps et surveillance des systèmes présignés URLs

Ryan Baker, Amazon Web Services (AWS)

Août 2025 (historique du document)

La sécurité est une préoccupation essentielle pour toutes les entreprises et un pilier essentiel de l'AWS Well-Architected Framework. En tant qu'ingénieur en sécurité, vous souhaiterez mettre en place des garde-fous administratifs conformes aux exigences de contrôle de l'organisation. Dans le AWS Well-Architected Framework, les barrières de sécurité définissent les limites qui limitent l'activité.

Ce guide fournit des informations générales et les meilleures pratiques relatives à l'utilisation des objets présignés URLs, qui sont utilisés avec les objets Amazon Simple Storage Service (Amazon S3). Le présigné URLs permet aux utilisateurs ou aux applications ayant accès à des informations d'identification valides de générer des demandes qui sont signées à l'avance et acceptées jusqu'à une date d'expiration définie. Un cas d'utilisation courant de la présignature URLs consiste à étendre l'accès à des objets ou à des ressources en partageant ces demandes. Les demandes présignées partagées sont générées par des systèmes ou des utilisateurs autorisés à exécuter une demande spécifique, puis peuvent être envoyées à d'autres systèmes ou utilisateurs pour étendre la capacité d'exécuter cette même demande.

Dans ce guide, vous découvrirez :

- Les concepts de présigné URLs
- Cas d'utilisation pour le pré-signé URLs
- Rambardes recommandées et optionnelles
- Options de surveillance
- Exemples d' Services AWS utilisation de presigned URLs

Public visé

Ce guide est destiné aux architectes et aux ingénieurs de sécurité chargés d'implémenter les contrôles de sécurité dans le AWS Cloud.

Objectifs

En tant qu'ingénieur en sécurité, vous devez savoir comment les créateurs de solutions mettent en œuvre la sécurité et connaître le type d'accès dont disposent vos utilisateurs finaux. Ce guide couvre un type d'accès, présigné URLs, qui est souvent utilisé avec Amazon S3. Presigned URLs fournit aux constructeurs des options pour relier efficacement les mécanismes d'authentification.

Dans Amazon S3, les requêtes présignées URLs représentent une catégorie unique de demandes. Les ingénieurs de sécurité peuvent surveiller et gérer ces demandes pour s'assurer qu'elles ne sont utilisées que lorsque cela est approprié et nécessaire. L'objectif de ce guide est d'aider les ingénieurs de sécurité à fournir ce type de supervision de haut niveau.

Après avoir lu ce guide, vous devriez comprendre ce qu'est une URL présignée, à quel moment elle est généralement utilisée et quelles sont les motivations de son utilisation.

Conditions préalables

Si votre entreprise n'a pas défini de politique de sécurité, d'objectifs de contrôle ou de normes, comme décrit dans le guide Implémentation des contrôles de sécurité sur AWS, nous vous recommandons d'effectuer ces tâches de gouvernance avant de poursuivre ce guide.

Avant de commencer, vous devez également connaître les meilleures pratiques recommandées et facultatives en matière de contrôle et de surveillance. Pour en savoir plus, consultez :

- Politiques de contrôle des services (AWS Organizations documentation)
- Politiques de contrôle des ressources (AWS Organizations documentation)
- Politiques relatives aux compartiments pour Amazon S3 (documentation Amazon S3)
- Journalisation des demandes avec journalisation des accès au serveur (documentation Amazon S3)
- Journalisation des appels d'API Amazon S3 à l'aide de AWS CloudTrail(documentation Amazon S3)

Vue d'ensemble de la version présignée URLs

Une URL présignée est un type de requête HTTP reconnu par le service Gestion des identités et des accès AWS (IAM). Ce qui différencie ce type de demande de toutes les autres AWS demandes, c'est le paramètre de X-Amz-Expires requête. Comme pour les autres demandes authentifiées, les demandes d'URL présignées incluent une signature. Pour les demandes d'URL présignées, cette signature est transmise dans`X-Amz-Signature`. La signature utilise les opérations cryptographiques Signature Version 4 pour coder tous les autres paramètres de demande.

> **Remarques**
>
> - La version 2 de Signature est actuellement en cours de dépréciation, mais elle est toujours prise en charge dans certains cas. Régions AWS Ce guide s'applique à la signature de la version 4 de Signature.
> - Le service de réception peut traiter les en-têtes non signés, mais le support pour cette option est limité et ciblé, conformément aux meilleures pratiques. Sauf indication contraire, supposons que tous les en-têtes doivent être signés pour qu'une demande soit acceptée.

Le `X-Amz-Expires` paramètre permet de traiter une signature comme valide avec un écart plus important par rapport à la date et à l'heure codées. D'autres aspects de la validité des signatures sont toujours évalués. Les identifiants de signature, s'ils sont temporaires, ne doivent pas être expirés au moment du traitement de la signature. Les informations d'identification de signature doivent être attachées à un directeur IAM disposant des autorisations suffisantes au moment du traitement.

Les demandes présignées URLs sont un sous-ensemble des demandes présignées

Une URL présignée n'est pas la seule méthode pour signer une demande pour une date future. Amazon S3 prend également en charge les requêtes POST, qui sont également généralement présignées. Une signature POST présignée autorise les téléchargements conformes à une politique signée et dont la date d'expiration est intégrée à cette politique.

Les signatures des demandes peuvent être datées du futur, bien que cela soit rare. Tant que les informations d'identification sous-jacentes sont valides, l'algorithme de signature n'interdit pas les rencontres futures. Cependant, ces demandes ne peuvent pas être traitées avec succès avant leur période de validité, ce qui rend les rencontres futures peu pratiques pour la plupart des cas d'utilisation.

Que permet une demande présignée ?

Une demande présignée ne peut autoriser que les actions autorisées par les informations d'identification utilisées pour signer la demande. Si les informations d'identification refusent implicitement ou explicitement l'action spécifiée par la demande présignée, la demande présignée est refusée lors de son envoi. Cela s'applique aux éléments suivants :

- Politiques de session associées aux informations d'identification
- Politiques basées sur l'identité associées au principal associé aux informations d'identification
- Politiques relatives aux ressources qui affectent la session ou le principal
- Politiques de contrôle des services qui affectent la session ou le principal
- Politiques de contrôle des ressources qui affectent la session ou le principal

Motivations motivant l'utilisation de demandes présignées

En tant qu'ingénieur en sécurité, vous devez savoir ce qui pousse les créateurs de solutions à utiliser URLs Presigned. Comprendre ce qui est nécessaire et ce qui est facultatif vous aidera à communiquer avec les créateurs de solutions. Les motivations peuvent inclure les suivantes :

- Pour prendre en charge un mécanisme d'authentification non IAM tout en bénéficiant de l'évolutivité d'Amazon S3. L'une des principales motivations est de communiquer directement avec Amazon S3 afin de bénéficier de l'évolutivité intégrée fournie par ce service. Sans cette communication directe, une solution devrait prendre en charge la charge de retransmission des octets envoyés PutObjectet des GetObjectappels. En fonction de la charge totale, cette exigence ajoute des difficultés de mise à l'échelle qu'un concepteur de solutions pourrait vouloir éviter.

  D'autres moyens de communication directe avec Amazon S3, tels que l'utilisation d'informations d'identification temporaires dans AWS Security Token Service (AWS STS) ou de signatures Signature Version 4 externes URLs, peuvent ne pas être adaptés à votre cas d'utilisation. Amazon S3 identifie les utilisateurs à l'aide AWS d'informations d'identification, tandis que les demandes présignées supposent une identification par des mécanismes autres que AWS les informations d'identification. Il est possible de combler cette différence tout en maintenant la communication directe pour les données grâce à des demandes présignées.
- Pour bénéficier de la compréhension native d'un navigateur en matière de URLs. URLs sont compris par les navigateurs, alors que les AWS STS informations d'identification et les signatures de la version 4 de Signature ne le sont pas. Cela est avantageux lors de l'intégration à des solutions basées sur un navigateur. Les solutions alternatives nécessitent plus de code, utilisent

davantage de mémoire pour les fichiers volumineux et peuvent être traitées différemment par des extensions telles que les scanners de logiciels malveillants et de virus.

Comparaison avec les AWS STS informations d'identification temporaires

Les informations d'identification temporaires sont similaires aux demandes présignées. Ils expirent tous les deux, permettent de délimiter l'accès et sont couramment utilisés pour relier les informations d'identification non IAM à une utilisation nécessitant des informations d'identification AWS.

Vous pouvez limiter une AWS STS identification temporaire à un seul objet et à une seule action S3, mais cela peut entraîner des problèmes de dimensionnement en raison AWS STS APIs de ses limites. (Pour plus d'informations, consultez l'article Comment puis-je résoudre les erreurs de limitation des API ou de « débit dépassé » pour IAM et sur AWS STS le site Web AWS Re:Post.) En outre, chaque identifiant généré nécessite un appel d' AWS STS API, ce qui ajoute de la latence et crée une nouvelle dépendance susceptible d'affecter la résilience. Un AWS STS identifiant temporaire a également un délai d'expiration minimum de 15 minutes, tandis qu'une demande présignée peut prendre en charge des durées plus courtes (60 secondes, c'est pratique dans les bonnes conditions).

Comparaison avec les solutions basées uniquement sur les signatures

Le seul élément intrinsèquement secret d'une demande présignée est sa signature Signature Version 4. Si un client connaît les autres détails d'une demande et reçoit une signature valide correspondant à ces détails, il peut envoyer une demande valide. Sans signature valide, ce n'est pas possible.

Les solutions présignées URLs et les solutions utilisant uniquement des signatures sont cryptographiquement similaires. Cependant, les solutions basées uniquement sur les signatures présentent des avantages pratiques tels que la possibilité d'utiliser un en-tête HTTP au lieu d'un paramètre de chaîne de requête pour transmettre la signature (voir la section Journalisation des interactions et des mesures d'atténuation). Les administrateurs doivent également tenir compte du fait que les chaînes de requête sont généralement traitées comme des métadonnées, alors que les en-têtes sont moins souvent traités comme tels.

D'autre part, AWS SDKs offrez moins de support pour la génération et l'utilisation directes de signatures. La création d'une solution basée uniquement sur les signatures nécessite davantage de code personnalisé. D'un point de vue pratique, l'utilisation de bibliothèques plutôt que de code personnalisé pour des raisons de sécurité est une bonne pratique générale. Le code des solutions utilisant uniquement des signatures doit donc faire l'objet d'une attention particulière.

Les solutions utilisant uniquement des signatures n'utilisent pas la chaîne de `X-Amz-Expires` requête et ne fournissent aucune période de validité explicite. IAM gère les périodes de validité implicites des signatures qui n'ont pas de date d'expiration explicite. Ces périodes implicites ne sont pas publiées. Ils ne changent généralement pas, mais ils sont gérés dans un souci de sécurité. Vous ne devez donc pas vous fier aux périodes de validité. Il y a un compromis entre le contrôle explicite de la date d'expiration et la gestion de l'expiration par IAM.

En tant qu'administrateur, vous préférerez peut-être une solution basée uniquement sur les signatures. Cependant, d'un point de vue pratique, vous devrez soutenir les solutions telles qu'elles ont été conçues.

Identifier les demandes présignées

Identification des demandes utilisant une URL présignée

Amazon S3 fournit deux mécanismes intégrés pour surveiller l'utilisation au niveau de la demande : les journaux d'accès au serveur Amazon S3 et les événements AWS CloudTrail liés aux données. Les deux mécanismes peuvent identifier l'utilisation des URL présignées.

Pour filtrer les journaux en fonction de l'utilisation des URL présignées, vous pouvez utiliser le type d'authentification. Pour les journaux d'accès au serveur, examinez le champ Type d'authentification, généralement nommé authtype lorsqu'il est défini dans une table Amazon Athena. Pour CloudTrail, examinez AuthenticationMethod`additionalEventData`sur le terrain. Dans les deux cas, la valeur du champ pour les demandes utilisant des URL présignées est`QueryString`, alors que c'`AuthHeader`est la valeur pour la plupart des autres demandes.

`QueryString`l'utilisation n'est pas toujours associée aux URL présignées. Pour limiter votre recherche à l'utilisation d'URL présignées uniquement, recherchez les demandes contenant le paramètre `X-Amz-Expires` de chaîne de requête. Pour les journaux d'accès au serveur, examinez Request-URI et recherchez les demandes dont la chaîne de requête contient un `X-Amz-Expires` paramètre. Pour CloudTrail, examinez l'`requestParameters`élément pour un `X-Amz-Expires` élément.

```
{"Records": [{…, "requestParameters": {…, "X-Amz-Expires": "300"}}, …]}
```

La requête Athena suivante applique ce filtre :

```
SELECT * FROM {athena-table} WHERE
  authtype = 'QueryString' AND
  request_uri LIKE '%X-Amz-Expires=%';
```

Pour AWS CloudTrail Lake, la requête suivante applique ce filtre :

```
SELECT * FROM {data-store-event-id} WHERE
  additionalEventData['AuthenticationMethod'] = 'QueryString' AND
  requestParameters['X-Amz-Expires'] IS NOT NULL
```

Identifier d'autres types de demandes présignées

La requête POST possède également un type d'authentification unique`HtmlForm`, dans les journaux d'accès au serveur Amazon S3 et CloudTrail. Ce type d'authentification étant moins courant, il est possible que vous ne trouviez pas ces demandes dans votre environnement.

La requête Athena suivante applique le filtre pour : `HtmlForm`

```
SELECT * FROM {athena-table} WHERE
  authtype = 'HtmlForm';
```

Pour CloudTrail Lake, la requête suivante applique le filtre :

```
SELECT * FROM {data-store-event-id} WHERE
  additionalEventData['AuthenticationMethod'] = 'HtmlForm'
```

Identification des modèles de demandes

Vous pouvez trouver des demandes présignées en utilisant les techniques décrites dans la section précédente. Toutefois, pour rendre ces données utiles, vous devez trouver des modèles. Les `TOP 10` résultats simples de votre requête peuvent fournir un aperçu, mais si cela ne suffit pas, utilisez les options de regroupement du tableau suivant.

Option de regroupement	Journaux d'accès au serveur	CloudTraillac	Description
Agent utilisateur	`GROUP BY useragent`	`GROUP BY userAgent`	Cette option de regroupement vous permet de trouver la source et le but des demandes. L'agent utilisateur est fourni par l'utilisateur et n'est pas fiable en tant que mécanisme d'authentification ou d'autorisation. Cependant

Option de regroupement	Journaux d'accès au serveur	CloudTraillac	Description
			, cela peut révéler beaucoup de choses si vous recherchez des modèles, car la plupart des clients utilisent une chaîne unique qui est au moins partiellement lisible par l'homme.
Demandeur	`GROUP BY requester`	`GROUP BY userIdentity['arn']`	Cette option de regroupement permet de trouver les principaux IAM qui ont signé les demandes. Si votre objectif est de bloquer ces demandes ou de créer une exception pour les demandes existantes, ces requêtes fournissent suffisamment d'informations à cette fin. Lorsque vous utilisez des rôles conformément aux meilleures pratiques IAM, le propriétaire du rôle est clairement identifié, et vous pouvez utiliser ces informations pour en savoir plus.

Option de regroupement	Journaux d'accès au serveur	CloudTraillac	Description
Adresse IP source	`GROUP BY remoteip`	`GROUP BY sourceIPAddress`	Cette option est groupée en fonction du dernier saut de traduction réseau avant d'atteindre Amazon S3. • Si le trafic passe par une passerelle NAT, il s'agira de l'adresse de la passerelle NAT. • Si le trafic passe par une passerelle Internet, il s'agira de l'adresse IP publique qui a envoyé le trafic vers la passerelle Internet. • Si le trafic provient de l'extérieur AWS, il s'agira de l'adresse Internet publique associée à l'origine. • S'il passe par un point de terminaison de cloud privé virtuel (VPC) de

Option de regroupement	Journaux d'accès au serveur	CloudTraillac	Description
			passerelle, il s'agira de l'adresse IP de l'instance dans le VPC. • S'il passe par une interface virtuelle publique (VIF), il s'agira de l'adresse IP locale du demandeur ou de tout intermédiaire tel qu'un serveur proxy ou un pare-feu qui n'expose que son adresse IP. • S'il passe par un point de terminaison VPC d'interface, il peut s'agir de l'adresse IP d'une instance du VPC. Il peut également s'agir d'une adresse IP provenant d'un autre VPC ou d'un réseau sur site. Comme pour les VIF publiques, il peut s'agir de l'adresse IP de

Option de regroupement	Journaux d'accès au serveur	CloudTraillac	Description
			n'importe quel intermédiaire. Ces données sont utiles si votre objectif est d'imposer des contrôles réseau. Vous devrez peut-être combiner cette option avec des données telles que `endpoint` (pour les journaux d'accès au serveur) ou `vpcEndpointId` (pour CloudTrail Lake) pour clarifier la source, car différents réseaux peuvent dupliquer des adresses IP privées.
Nom du compartiment S3	`GROUP BY bucket_name`	`GROUP BY requestParameters['bucketName']`	Cette option de regroupement permet de trouver les compartiments ayant reçu des demandes. Cela vous permet d'identifier le besoin d'exceptions.

Bonnes pratiques d'utilisation des demandes présignées

Cette section décrit les meilleures pratiques d'utilisation des demandes présignées qu'un ingénieur en sécurité doit prendre en compte. Les directives incluent :

- Les meilleures pratiques fondamentales, qui sont des pratiques que chaque organisation devrait suivre.
- Des garde-fous supplémentaires, qui sont des pratiques que vous devriez envisager, mais que vous pourriez décider de mettre en œuvre partiellement ou à quelques exceptions près. Ils visent à fournir un contrôle et une défense supplémentaires en profondeur, mais doivent être mis en balance avec la complexité globale.
- Enregistrement des interactions, qui peuvent résulter d'appareils ou de services relevant de votre responsabilité ou de celle de votre client dans le cadre du modèle de responsabilité partagée. Cette section inclut des précautions visant à limiter les informations accessibles par le biais des journaux.

Bonnes pratiques fondamentales

Les meilleures pratiques générales qui constituent des contrôles efficaces pour les autres demandes d' AWS API s'appliquent également aux demandes présignées. Cette section passe en revue deux des pratiques les plus pertinentes : le moindre privilège et les périmètres de données. Ces pratiques créent une profondeur de contrôle que d'autres pratiques étendent.

Application du principe du moindre privilège

La première étape pour limiter l'utilisation des demandes présignées consiste à limiter l'accès à Amazon S3 en général. Une URL présignée ne peut pas fournir d'accès à des ressources qui n'ont pas été accordées au principal qui a généré la signature de l'URL présignée. Il ne peut pas non plus donner accès à une ressource d'une manière qui n'a pas été accordée à ce mandant. En tant que tel, l'application des meilleures pratiques pour accorder le moins de privilèges à ces principaux constitue un garde-fou efficace.

Le processus de création d'une URL présignée est une opération algorithmique basée sur une norme publiée (Signature Version 4) pour la génération de signatures. Il n'est donc pas possible de limiter la génération d'URL présignées. Toutefois, pour être pertinente, une URL présignée doit être valide et fournir un accès aux ressources. La validité d'une URL présignée constitue donc également un garde-fou efficace.

Pour plus d'informations sur le moindre privilège, voir Accorder l'accès au moindre privilège dans le AWS Well-Architected cadre, pilier Sécurité.

Implémenter un périmètre de données

L'extension du moindre privilège consiste à maintenir un périmètre de données conforme aux besoins de votre entreprise. Les URL présignées sont compatibles avec les périmètres de données. Comme pour les autres demandes, la validité d'une demande d'URL présignée est évaluée au moment de la demande. Si les propriétés du réseau, de la ressource, de la session de rôle et du principal changent, elles sont évaluées au moment et en utilisant la méthode par laquelle une demande est reçue.

Supposons, par exemple, qu'un service exécuté dans un conteneur Amazon Elastic Kubernetes Service (Amazon EKS) signe une demande. La demande est ensuite envoyée depuis le système informatique personnel d'un utilisateur connecté à Internet. Dans ce cas, la SourceIp condition aws : évalue l'adresse IP publique visible de la demande depuis le système personnel de l'utilisateur, et non l'adresse IP du service dans le conteneur Amazon EKS.

De même, si les balises du principal ou de la ressource changent avant l'envoi de la demande, les valeurs mises à jour, et non les valeurs d'origine, s'appliqueront à la demande via les ResourceTag/ tag-key conditions aws : PrincipalTag/tag-key et aws :.

Rambardes supplémentaires

Lorsque les demandes présignées sont utilisées de manière appropriée par les créateurs de solutions et les utilisateurs, elles fournissent un mécanisme sécurisé permettant aux utilisateurs d'accéder aux données. En outre, la possibilité de générer des demandes présignées ne fournit pas aux donneurs d'ordre un accès qu'ils n'avaient pas déjà.

Dans ce contexte, des contrôles supplémentaires sont-ils nécessaires ? La justification des contrôles supplémentaires ne repose pas sur la nécessité de refuser l'accès, mais sur la capacité de surveiller, d'approuver l'utilisation et de fixer des limites, et de réduire les risques d'erreurs des utilisateurs. De cette façon, vous pouvez contribuer à garantir que l'utilisation est appropriée et nécessaire.

Les rambardes suivantes vous aideront à atteindre cet objectif. Avant d'activer ces contrôles, vous souhaiterez peut-être déterminer l'utilisation existante en identifiant les demandes présignées. Cette identification vous aide à vous préparer à l'impact du garde-corps sur l'utilisation existante ou à planifier des exceptions si nécessaire.

Rambarde pour S3 : SignatureAge

L'une des caractéristiques déterminantes des demandes présignées est qu'elles décrivent un délai d'expiration. La signature de la demande contient une date. Cette date est transmise sous forme de paramètre de chaîne de `X-Amz-Date` requête pour les URL présignées, et sous forme d'en-tête Date ou x-amz-date pour un POST présigné.

Amazon S3 fournit une clé de condition, s3:SignatureAge, que vous pouvez utiliser pour limiter le délai maximum entre la date de signature et l'expiration effective de la demande. Cette condition ne peut jamais augmenter la durée de validité, mais elle peut la réduire.

Dans la politique suivante, la clé de `s3:signatureAge` condition limite les demandes présignées à 15 minutes de validité. Les exemples suivants utilisent tous 15 minutes pour limiter la validité à une période similaire à celle prise en charge par la signature standard.

La deuxième déclaration de la politique refuse tout accès à Signature Version 2. Cette version du protocole de signature est obsolète, mais elle est toujours prise en charge dans certains d'entre eux. Régions AWS Nous vous recommandons de le bloquer explicitement avant qu'il ne soit totalement obsolète.

Vous pouvez appliquer la politique suivante en tant que politique AWS Organizations de contrôle des services (SCP). Les utilisateurs peuvent toujours utiliser des demandes présignées et déployer des solutions qui dépendent de ces demandes, à condition que le délai entre la génération des signatures et leur utilisation soit inférieur à 15 minutes. Selon la mise en œuvre, cette limitation peut n'avoir aucun impact, rendre une solution inutilisable ou provoquer des échecs occasionnels qui peuvent être réessayés.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPresignedOver15Minutes",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        }
      }
```

```
    },
    {
      "Sid": "DenySignatureVersion2",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
           "s3:signatureversion": "AWS"
        }
      }
    }
  ]
}
```

Exceptions

Si une solution nécessite un délai plus long avant son expiration et est donc affectée par la politique précédente, nous vous recommandons de fournir une méthode pour approuver les exceptions. Pour éviter d'énumérer les exceptions dans un SCP, utilisez aws : PrincipalTag, comme dans la politique suivante, pour gérer les exceptions de manière évolutive. D'autres AWS exemples, tels que les exemples de politique de périmètre des données AWS, utilisent cette stratégie.

Si vous implémentez une politique d'exception en utilisant`aws:PrincipalTag`, vous devez contrôler l'accès aux balises de configuration sur les principes. Les balises de ce type peuvent provenir directement des principaux et peuvent être contrôlées par un SCP, comme dans cet exemple de contrôle des valeurs de balise pouvant être définies. Une balise de ce type peut également provenir de balises de session, définies par un fournisseur d'identité (IdP) ou lors de l'utilisation. AWS STS Le contrôle de l'accès à `aws:PrincipalTag` est un sujet complexe. Toutefois, une organisation expérimentée dans l'utilisation du contrôle d'accès basé sur les attributs (ABAC) aura l'expérience et les contrôles nécessaires pour permettre une utilisation appropriée `aws:PrincipalTag` pour ce cas d'utilisation.

Dans l'exemple suivant, la `aws:PrincipalTag` condition crée une exception qui autorise tout principal auquel le nom tag (`long-presigned-allowed`) est attribué et défini sur`true`. À cette exception près, la restriction relative à l'âge de signature n'est pas appliquée.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Sid": "DenyPresignedOver15Minutes",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        },
        "StringNotEquals": {
          "aws:PrincipalTag/long-presigned-allowed": "true"
        }
      }
    }
  ]
}
```

Politiques de compartiment

Vous pouvez appliquer des politiques de compartiment à tous les compartiments ou à certains d'entre eux en utilisant une politique, comme dans l'exemple suivant. Contrairement à un SCP, une politique de compartiment cible également l'utilisation principale du service. L'annexe A ne documente aucune utilisation prévue du principal service des demandes présignées, mais si vous souhaitez mettre en œuvre un contrôle afin de prouver cette limite, la politique suivante vous fournira ce contrôle. De plus, contrairement à un SCP, une politique de compartiment peut s'appliquer aux principaux de votre compte de gestion.

ABAC-based les exceptions fonctionnent dans les politiques de compartiment de la même manière qu'un SCP. L'un des objectifs d'une politique de compartiment peut être de s'appliquer aux directeurs extérieurs à l'organisation. Les exceptions ABAC doivent donc être limitées aux principes auxquels les contrôles ABAC s'appliquent.

Dans l'exemple suivant, la `aws:PrincipalTag` condition de la première instruction crée une exception pour un principal auquel le nom tag (`long-presigned-allowed`) est attribué et défini sur`true`. À cette exception près, la restriction relative à l'âge de signature n'est pas appliquée. La deuxième déclaration applique cette restriction à tous les directeurs extérieurs à l' AWS organisation propriétaire du bucket. La portée de cette deuxième instruction doit correspondre aux contrôles ABAC pour définir la balise nommée pour les principaux.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPresignedOver15MinWithExceptions",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::{bucket-name}/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        },
        "StringNotEquals": {
          "aws:PrincipalTag/long-presigned-allowed": "true"
        }
      }
    },
    {
      "Sid": "DenyPresignedOver15MinutesOutsideOrg",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::{bucket-name}/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        },
        "StringNotEquals": {
          "aws:PrincipalOrgID": "${aws:ResourceOrgID}"
        }
      }
    }
  ]
}
```

Politiques de contrôle des ressources

Vous pouvez appliquer une politique aux compartiments à grande échelle en utilisant des politiques de contrôle des ressources (RCP). À l'instar des SCP et contrairement aux politiques relatives aux compartiments, les RCP ne ciblent pas l'utilisation principale du service. Les RCP affectent les principaux non liés au service, quel que soit le compte, mais ils n'affectent pas les ressources du compte de gestion. Pour plus d’informations, consultez la documentation AWS Organizations.

Comme pour les politiques relatives `aws:PrincipalTags` aux compartiments, si vous créez des exceptions pour les principaux, gardez à l'esprit l'étendue des contrôles ABAC sur le balisage des principaux.

Le RCP suivant limite l'utilisation des URL présignées dans tous les compartiments S3 d'une organisation en limitant l'âge des signatures à 15 minutes.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPresignedOver15MinWithExceptions",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::*/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        },
        "StringNotEquals": {
          "aws:PrincipalTag/long-presigned-allowed": "true",
        }
      }
    },
    {
      "Sid": "DenyPresignedOver15MinutesOutsideOrg",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::*/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        },
        "StringNotEquals": {
          "aws:PrincipalOrgID": "${aws:ResourceOrgID}"
        }
      }
    }
  ]
}
```

Rambarde pour S3:AuthType

Les URL présignées utilisent l'authentification par chaîne de requête, et les POST présignés utilisent toujours l'authentification POST. Amazon S3 prend en charge le refus des demandes en fonction du type d'authentification via la clé de condition s3:AuthType. `REST-QUERY-STRING`est la `s3:authType` valeur des chaînes de requête et `POST` la `s3:authType` valeur de POST.

Vous pouvez appliquer la politique suivante en tant que SCP. La politique permet `s3:authType` d'autoriser uniquement l'authentification basée sur les en-têtes. Il configure également une méthode pour fournir des exceptions à des utilisateurs ou à des rôles individuels.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyNonHeaderAuth",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "s3:authType": "REST-HEADER",
          "aws:PrincipalTag/non-header-auth-allowed": "true"
        }
      }
    }
  ]
}
```

Le refus des demandes en fonction du type d'authentification affecte toute solution ou fonctionnalité utilisant le type d'authentification refusé. Par exemple, le refus `REST-QUERY-STRING` empêche les utilisateurs d'effectuer des chargements ou des téléchargements depuis la console Amazon S3. Si vous souhaitez que les utilisateurs utilisent la console Amazon S3, n'utilisez pas ce garde-fou et ne faites pas d'exception pour les utilisateurs. D'autre part, si vous ne souhaitez pas que les utilisateurs utilisent la console Amazon S3, vous pouvez refuser l'accès `REST-QUERY-STRING` aux utilisateurs.

Vous refusez peut-être déjà aux utilisateurs l'accès direct aux ressources Amazon S3. Dans ce cas, un garde-fou pour le type d'authentification est redondant. Cependant, une instruction de `s3:authType` refus fournit un utilitaire de défense approfondie, car les implémentations visant à

refuser l'accès direct couvrent généralement de nombreuses instructions de contrôle, certaines avec des exceptions.

Les rôles utilisés pour les charges de travail n'ont généralement pas besoin d'accéder à une chaîne de requête ou `POST` d'authentification. Les exceptions sont les rôles qui prennent en charge les services conçus pour utiliser des demandes présignées. Vous pouvez créer des exceptions spécifiques pour ces rôles.

Vous pouvez également appliquer une politique de compartiment à tous les compartiments ou à certains d'entre eux en utilisant une stratégie telle que la suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyNonHeaderAuth",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::{bucket-name}/*",
      "Condition": {
        "StringNotEquals": {
          "s3:authType": "REST-HEADER",
          "aws:PrincipalTag/non-header-auth-allowed": "true"
        }
      }
    }
  ]
}
```

Cette politique de compartiment a pour effet de refuser l'utilisation des UploadPartCopyAPI CopyObjectet pour effectuer des copies entre régions. Amazon S3 Replication n'est pas affectée car elle ne repose pas sur ces API.

Si vous souhaitez utiliser une politique de compartiment telle que la politique précédente tout en prenant en charge l'interrégion CopyObjectou UploadPartCopyl'API, ajoutez une condition `aws:ViaAWSService` similaire à la suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Sid": "DenyNonHeaderAuth",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::{bucket-name}/*",
      "Condition": {
        "StringNotEquals": {
          "s3:authType": "REST-HEADER",
          "aws:PrincipalTag/non-header-auth-allowed": "true"
        },
        "Bool": {
          "aws:ViaAWSService": "false"
        },
      }
    }
  ]
}
```

Combinaison de garde-corps présignés et d'exceptions à d'autres garde-corps

Si vous ne prévoyez pas d'appliquer de garde-fou de manière générale à vos utilisateurs et à vos rôles, vous souhaiterez peut-être l'appliquer aux exceptions des autres barrières de sécurité courantes, afin que ces exceptions ne prennent pas en charge les demandes présignées.

Si vous avez des restrictions réseau mais que vous autorisez des exceptions pour des partenaires externes ou des cas d'utilisation particuliers, vous devez bloquer la chaîne de requête ou `POST` l'authentification lorsque ces exceptions sont appliquées, sauf si elles sont spécifiquement identifiées comme étant obligatoires.

Limitations de S3 : SignatureAge

Les administrateurs trouveront utile de mieux comprendre les implications des`s3:signatureAge`. Chaque demande signée inclut`X-Amz-Date`, qui doit indiquer l'heure actuelle. Cette valeur est renseignée par le client et le signataire de la demande. AWS rejette les demandes dont les heures sont considérées comme non valides. Cependant, un signataire peut générer des signatures à l'avance pour une date future. Amazon S3 rejette les demandes qui spécifient une date future si elles sont envoyées trop longtemps à l'avance. Toutefois, si la demande n'est envoyée qu'au moment de la connexion à la signature, celle-ci peut être générée plus tôt et envoyée ultérieurement.

`s3:signatureAge`limite l'âge maximum d'`X-Amz-Date`une signature uniquement pour les demandes présignées. Les demandes dont l'âge est supérieur à l'âge spécifié sont refusées, même si leur date d'expiration `X-Amz-Expires` ou si une `POST` politique les aurait déclarées valides. `s3:signatureAge`ne modifie pas la période de validité pour les demandes qui n'incluent pas d'expiration explicite. Il ne contrôle pas non plus la valeur de `X-Amz-Date` ce qu'un client utilise pour une signature.

Si l'horloge système est incorrecte ou si un client reporte intentionnellement ses demandes, il est possible que l'heure de signature ne corresponde pas à l'heure à laquelle la signature a été générée. Cela limite la mesure dans laquelle les solutions `s3:signatureAge` peuvent être contrôlées. Une solution qui utilise l'heure actuelle à laquelle elle génère des signatures est limitée comme prévu : les signatures restent valides pendant le nombre de millisecondes spécifié dans. `s3:signatureAge` Une solution qui n'utilise pas l'heure actuelle aura des limites différentes. L'une des restrictions est que les informations d'identification utilisées pour signer la signature doivent toujours être valides. En tant qu'administrateur, vous pouvez contrôler la validité maximale des informations d'identification temporaires émises. Vous pouvez autoriser la validité des informations d'identification jusqu'à 36 heures ou limiter leur validité à 15 minutes. L'expiration des informations d'identification temporaires ne dépend pas de la valeur de`X-Amz-Date`.

Les informations d'identification permanentes ne sont pas soumises à cette restriction. Il est recommandé d'utiliser uniquement des informations d'identification temporaires, et vous pouvez révoquer explicitement toute information d'identification permanente, ce qui invaliderait également toute signature basée sur cette identification.

Bien qu'il `s3:signatureAge` soit mesuré en millisecondes, il n'est pas pratique de le régler à moins de 60 secondes, même si votre horloge est bien synchronisée et que vous utilisez une faible latence. Les paramètres inférieurs à 60 secondes risquent de rejeter des demandes valides. Si vous vous attendez à des retards entre la génération des signatures et la soumission de la demande, ou à des problèmes de synchronisation des horloges, vous devez en tenir compte dans votre gestion des`s3:signatureAge`.

Cibler les seaux à grande échelle

Les SCP et RCP peuvent être utilisés `aws:PrincipalTag` pour faire des exceptions pour les utilisateurs. Vous ne pouvez pas utiliser de balises sur un bucket pour contrôler l'accès par ce biais`aws:ResourceTag`. Seules les balises d'objet sont utilisées pour le contrôle d'accès. Il n'est généralement pas évolutif d'ajouter une balise à chaque objet auquel vous souhaitez appliquer ce contrôle.

Une solution adaptée à de nombreux cas d'utilisation consiste à appliquer la politique et l'exception au niveau du compte, soit en modifiant les comptes auxquels s'applique le SCP ou le RCP, soit en utilisant aws : ResourceAccount, aws : ResourceOrgPaths ou aws : ResourceOrg ID. Par exemple, un SCP ou un RCP peut être appliqué à un ensemble de comptes de production.

Une autre solution consiste à utiliser une AWS Config règle personnalisée pour implémenter un contrôle de détection ou un contrôle réactif. L'objectif serait que chaque compartiment contienne une politique de compartiment avec le garde-corps approprié. En plus de tester le contenu d'une politique de compartiment, la AWS Config règle personnalisée peut récupérer les balises du compartiment et exclure le compartiment de la règle si le compartiment est étiqueté avec une valeur spécifique. Si cette règle échoue à son contrôle de conformité, elle peut soit marquer le compartiment comme non conforme, soit invoquer une correction pour ajouter le garde-fou à la politique du compartiment.

> **Note**
>
> Vous ne pouvez pas restreindre le contenu des balises des demandes à PutBucketTagging. Pour garder le contrôle sur la façon dont un bucket est étiqueté, vous devez limiter l'accès à `PutBucketTagging` et DeleteBucketTagging.

Journalisation des interactions et mesures d'atténuation

Une URL présignée contient une signature et peut être utilisée, pendant la période précédant son expiration, pour effectuer l'opération d'API spécifique pour laquelle elle a été signée. Il doit être traité comme un identifiant d'accès temporaire. La signature doit rester confidentielle uniquement pour les parties qui ont besoin de la connaître. Dans la plupart des environnements, il s'agit du client qui envoie la demande et du serveur qui la reçoit. L'envoi de la signature dans le cadre d'une session HTTPS directe conserve son caractère privé, car seul un participant à la session HTTPS a une visibilité sur l'URI qui transmet la signature.

Pour les URL présignées, la signature est transmise en tant que paramètre de chaîne de `X-Amz-Signature` requête. Les paramètres de chaîne de requête font partie d'un URI. Le risque est que les clients puissent enregistrer l'URI et la signature avec celui-ci. Les clients ont accès à l'intégralité de la requête HTTP et peuvent enregistrer n'importe quelle partie de la demande, des données et des en-têtes (y compris les en-têtes d'authentification). Cependant, cela est par convention moins courant. La journalisation des URI est plus courante et est requise dans des cas tels que la journalisation des accès. Les clients doivent utiliser la rédaction ou le masquage pour supprimer la signature avant de consigner les URI.

Dans certains environnements, les utilisateurs autorisent les intermédiaires (proxies) à obtenir de la visibilité sur leurs sessions HTTPS. L'activation des proxys nécessite un niveau élevé d'accès privilégié aux systèmes clients, car ils nécessitent une configuration et des certificats fiables. L'installation de la configuration du proxy et de certificats sécurisés, dans le contexte local de l'environnement intermédiaire du client, permet un niveau de privilège très élevé. C'est pourquoi l'accès à ces intermédiaires doit être étroitement contrôlé.

Le but d'un intermédiaire est généralement de bloquer les sorties indésirables et de suivre les autres sorties. Il est donc courant que ces intermédiaires enregistrent les demandes. Bien que les intermédiaires puissent, comme les clients, enregistrer n'importe quel contenu, en-tête et donnée (tous ces éléments étant très sensibles), il est plus courant qu'ils enregistrent des URI, tels que ceux qui incluent le paramètre de chaîne de `X-Amz-Signature` requête.

Atténuations

Nous recommandons que la journalisation des URI supprime le paramètre de la chaîne de `X-Amz-Signature` requête, expédie la chaîne de requête dans son intégralité ou traite les informations de manière hautement confidentielle, comme dans le cas d'un accès direct au serveur intermédiaire. Bien que ces protections soient fortement recommandées, le fait que les URL présignées expirent atténue les risques d'exposition des logs, à condition que l'exposition soit retardée suffisamment longtemps pour que les signatures expirent.

Amazon S3 voit également les signatures et doit les gérer de manière appropriée. Les journaux d'accès au serveur Amazon S3 incluent l'URI de la demande, mais le biffent`X-Amz-Signature`, comme recommandé. Il en va de même lorsque CloudTrail des événements de données sont enregistrés pour Amazon S3. Vous pouvez configurer Amazon CloudWatch Logs pour masquer les données à l'aide d'identifiants de données personnalisés.

L'expression régulière suivante correspond à `X-Amz-Signature` celle qui apparaît dans un URI :

```
X-Amz-Signature=[a-f0-9]{64}
```

L'expression régulière suivante ajoute des modèles de regroupement pour identifier plus spécifiquement le texte à remplacer :

```
(?:X-Amz-Signature=)([a-f0-9]{64})
```

Si vous disposez d'une entrée du journal d'accès telle que la suivante :

```
X-Amz-Signature=733255ef022bec3f2a8701cd61d4b371f3f28c9f193a1f02279211d48d5193d7
```

La première expression régulière traduit l'entrée du journal d'accès en :

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

La deuxième expression régulière, sur les systèmes qui prennent en charge les groupes non capturants, traduit l'entrée du journal d'accès en :

```
X-Amz-Signature=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

FAQ

Une demande présignée peut-elle être utilisée plusieurs fois ? Est-ce un risque pour la sécurité ?

Oui, une signature figurant dans une demande présignée peut être utilisée plusieurs fois. La question de savoir s'il s'agit d'un risque de sécurité est une question contextuelle. D'autres méthodes d'accès aux services AWS autorisent également la répétition. Un utilisateur ou une charge de travail disposant AWS d'informations d'identification peut envoyer de nombreuses demandes Services AWS, et chacune de ces demandes peut être dupliquée.

Si votre cas d'utilisation ne nécessite qu'une seule exécution, vous devez mettre en œuvre d'autres mécanismes pour appliquer l'usage unique. L'usage unique n'est pas une fonctionnalité des demandes présignées. En tant qu'ingénieur en sécurité, vous devez examiner les cas d'utilisation et les implémentations, mais dans de nombreux cas, une utilisation multiple convient à une utilisation acceptable.

Une autre personne que l'utilisateur prévu peut-elle utiliser une demande présignée ?

La signature d'une demande présignée peut être envoyée par toute personne en possession de cette signature. Il ne sera accepté que s'il passe d'autres formes de validation, telles que les contrôles du périmètre des données. Si la signature a expiré, si les informations d'identification ont expiré ou si les informations de signature n'ont pas accès aux ressources demandées, la demande sera refusée.

Il en va de même pour les autres méthodes d'authentification avec Services AWS. Les informations d'identification partagées de manière inappropriée autorisent un accès inapproprié. La meilleure pratique de base consiste à partager les informations d'identification et les signatures uniquement avec le public cible. Si vous ne pouvez pas faire confiance à votre public cible pour protéger les données privées et ne pas les partager avec d'autres, cela compromettra toute forme d'authentification.

Un utilisateur autorisé peut-il utiliser une demande présignée pour exfiltrer des données ?

La sécurisation des données nécessite des mesures énergiques. Permettre l'accès aux fins prévues tout en maintenant un périmètre de données nécessite une approche globale. L'accès au moindre privilège, le contrôle du périmètre des données et l'utilisation d'identifiants d'accès temporaires uniquement sont les meilleures pratiques générales applicables à la sécurisation des données. L'utilisation appropriée de ces contrôles limite également la capacité des utilisateurs à effectuer des actions par le biais des demandes présignées qu'ils génèrent.

Cela est dû au fait que l'accès fourni par une demande présignée est un sous-ensemble de l'accès accordé aux informations d'identification utilisées pour signer la demande. Dans ce contexte, les meilleures pratiques applicables à l'accès aux données s'appliquent généralement aux demandes présignées, mais les demandes présignées ne créent aucun nouvel accès aux données.

- L'expiration maximale est limitée à l'expiration des informations de signature. Si les informations de signature sont révoquées, les signatures basées sur les informations d'identification ne sont plus valides.
- Si les autorisations accordées au principal IAM associées aux informations d'identification de signature n'incluent pas l'exécution de l'action associée à la demande présignée, l'appel d'une demande présignée entraîne une réponse « accès refusé ». La réponse dépend de l'état actuel des autorisations au moment de l'invocation, qui n'a aucun rapport avec le moment où la signature de la demande présignée a été générée.
- Les propriétés du principal sont évaluées en fonction du principal associé aux informations de signature.
- Les propriétés d'une session de rôle sont évaluées en fonction de la session de rôle associée aux informations d'identification de signature.
- Les propriétés du réseau sont évaluées en fonction de la façon dont la demande a été reçue, comme pour les demandes normales.

Dans ce contexte, l'examen des risques associés aux demandes présignées est limité aux domaines dans lesquels elles sont signées avec des informations d'identification différentes de celles de l'utilisateur et fournissent un accès qui ne faisait pas partie du principal de l'utilisateur. Cet examen doit être appliqué à la conception du service, à la charge de travail ou à la solution qui génère des signatures au nom d'un utilisateur, plutôt qu'à la capacité de demande présignée elle-même.

Puis-je refuser l'accès à partir d'une URL présignée si je pense qu'elle a été partagée de manière non autorisée ?

Oui. Cela nécessite d'invalider les informations d'identification avec lesquelles l'URL a été signée. Il existe plusieurs moyens d'y parvenir :

- Supprimez les autorisations du principal IAM auquel appartiennent les informations d'identification. Si ce principal IAM n'a plus accès à la ressource et à l'opération pour lesquelles l'URL est signée, l'URL ne peut pas exécuter cette opération. Cela affecte toutes les utilisations correspondantes provenant de ce principal IAM.
- Si les informations d'identification utilisées pour signer l'URL sont des AWS STS informations d'identification temporaires, vous pouvez révoquer les autorisations de session pour les informations d'identification temporaires émises avant une heure précise pour le principal IAM. Selon le cas d'utilisation, d'autres sessions valides peuvent être invalidées avant leur date d'expiration normale, mais les nouvelles sessions ne seront pas affectées. La révocation des autorisations de session invalide également celles URLs qui ont été signées à l'aide des informations d'identification associées à ces sessions, mais les nouvelles autorisations URLs associées aux nouvelles sessions ne seront pas affectées.
- Si les informations d'identification utilisées pour signer l'URL sont des informations d'identification permanentes, désactivez la clé d'accès. Cela affecte toutes les utilisations liées à ces informations d'identification.

Ressources

Documentation Amazon S3

- Authentification des demandes (AWS Signature version 4)
- Authentification des demandes : utilisation des paramètres de requête (AWS Signature version 4)
- Authentification des demandes : téléchargements depuis un navigateur à l'aide de POST (AWS Signature Version 4)
- Clés de politique spécifiques à l'authentification Amazon S3 Signature version 4
- Utilisation d'URL présignées

Autres références

- Création d'un périmètre de données sur AWS (AWS livre blanc)
- SEC03-BP02 Accorder le moindre privilège d'accès (cadre AWS bien conçu, pilier de sécurité)
- SEC03-BP05 Définissez des barrières d'autorisation pour votre organisation (AWS Well Architected Framework, Security Pilier)

Annexe A : Comment Services AWS utiliser le présigné URLs

Cette annexe fournit des informations Services AWS et des fonctionnalités relatives à l'utilisation de la présignature URLs. Ces informations ont deux objectifs :

- Fournir aux ingénieurs de sécurité qui mettent en œuvre des contrôles des informations sur les impacts possibles de ces contrôles.
- Sensibiliser aux situations dans lesquelles ce risque peut être pertinent pour les interactions de journalisation des URL.

> **Important**
>
> Cette annexe ne fournit pas une liste complète des présignés Services AWS URLs ni leur utilisation. Il ne couvre pas non plus les solutions personnalisées ou tierces.

Console Amazon S3

Principal : utilisateur de la console

Expiration par défaut : 5 minutes

> **Exclusion de responsabilité**
>
> Cette section décrit le comportement actuel de la console Amazon S3. AWS les comportements de la console peuvent être modifiés sans préavis.

La console Amazon S3 prend en charge le téléchargement et le chargement d'objets. Les téléchargements utilisent une URL présignée dont le délai d'expiration est de 300 secondes (5 minutes). L'URL est générée par une demande adressée à`https://<bucket-region>.console.aws.amazon.com/s3/batchOpsServlet-proxy`.

Cette demande est lancée lorsque l'utilisateur clique sur un bouton de téléchargement, de sorte que l'URL n'est pas générée à l'avance ou envoyée au client tant que la demande explicite de téléchargement n'a pas été formulée.

Les téléchargements sont similaires, sauf que la console envoie deux demandes : `OPTIONS` sous forme de contrôle CORS avant le vol, et. `PUT` Les deux demandes utilisent la même signature.

Les informations d'identification utilisées pour la signature sont des informations d'identification temporaires associées à l'utilisateur actuellement connecté. Les détails relatifs à la méthode d'obtention de ces informations d'identification temporaires ne sont pas inclus dans ce guide.

Amazon S3 Object Lambda

Principal : appelant du point d'accès

Expiration par défaut : 61 secondes

> **Note**
>
> Depuis le 7 novembre 2025, S3 Object Lambda n'est disponible que pour les clients existants qui utilisent actuellement le service, ainsi que pour certains partenaires AWS Partner Network (APN). Pour des fonctionnalités similaires à celles de S3 Object Lambda, cliquez ici : modification de la disponibilité d'Amazon S3 Object Lambda.

Amazon S3 Object Lambda utilise des AWS Lambda fonctions pour traiter et transformer automatiquement les données lorsqu'elles sont extraites d'Amazon S3. Lorsque S3 Object Lambda invoque une fonction, celle-ci reçoit une URL présignée (`inputS3Url`) qu'elle peut utiliser pour télécharger l'objet d'origine depuis le point d'accès compatible.

Ils URLs sont présignés pour le point d'accès Amazon S3 compatible, qui est fourni lorsque vous configurez S3 Object Lambda. (Ce n'est pas la même chose que le point d'accès Object Lambda.) Au lieu d'utiliser un rôle lié à la fonction Lambda, l'URL est signée en utilisant l'identité de l'appelant d'origine, et les autorisations de cet utilisateur s'appliquent lorsque l'URL est utilisée. Si l'URL contient des en-têtes signés, la fonction Lambda doit inclure ces en-têtes dans l'appel à Amazon S3.

L'URL présignée renvoyée a un délai d'expiration de 61 secondes (une seconde de plus que la durée maximale d'une fonction Lambda d'un objet S3). L'URL générée ne peut être utilisée qu'avec le point d'accès compatible. L'appelant du point d'accès S3 Object Lambda doit avoir accès à ce point d'accès. Vous pouvez limiter cet accès au contexte de S3 Object Lambda en utilisant la condition. `"aws:CalledVia": ["s3-object-lambda.amazonaws.com"]` Lorsque cette condition est attachée à un point d'accès ou à un bucket de support, un utilisateur ne peut pas accéder directement au point d'accès ou au bucket de support.

L'avantage de cette approche est qu'il n'est pas nécessaire d'accorder à la fonction Lambda l'accès à votre compartiment ou point d'accès S3. Le rôle associé à la fonction Lambda aura besoin d'autorisations pour WriteGetObjectResponse, mais pas d'autorisations pour. GetObject

Lorsque S3 Object Lambda génère une version présignée URLs, il n'ajoute aucune restriction réseau. Une URL peut donc être utilisée en dehors de la fonction Lambda. Cependant, toutes les restrictions imposées à l'appelant de S3 Object Lambda s'appliquent toujours. Par exemple, si votre fonction Lambda s'exécute dans un VPC et que vous limitez l'appelant à utiliser un point de terminaison VPC, toute personne en possession de l'URL présignée devra être en mesure de l'envoyer via ce point de terminaison VPC. Cette restriction s'applique également à SourceIpet VpcSourceIp.

> **Note**
>
> Pour utiliser une fonction Lambda d'un objet S3 dans un VPC, celui-ci doit disposer d'une route vers les points de terminaison S3 publics à appeler. WriteGetObjectResponse Cela n'indique pas que les exigences relatives à l'utilisation d'un point de terminaison VPC ne s'appliqueraient pas aux demandes de récupération de données depuis le compartiment.

AWS Lambda Interrégional CopyObject

Principal :AWS interne

Expiration par défaut : 3600 secondes

Lorsque vous utilisez l'UploadPartCopyAPI CopyObjector pour effectuer une copie Régions AWS, Amazon S3 utilise une version présignée URLs en interne. Ils APIs peuvent être appelés directement depuis SDKs ou depuis les AWS CLI commandes `aws s3api copy-object` et`aws s3api upload-part`. Ils APIs ne sont pas utilisés pour la réplication Amazon S3, mais ils sont utilisés par les `aws s3 sync` commandes AWS CLI `aws s3 cp` et lorsque la source et la destination sont des compartiments S3. Ils sont également pris en charge par `TransferManager` des implémentations dans divers AWS SDKs domaines.

AWS Lambda GetFunction

Principal : AWS interne

Expiration par défaut : 10 minutes

AWS Lambda stocke la version utilisateur dans un compartiment S3 appartenant à l'équipe Lambda, avant de générer les actifs déployés sur des conteneurs Lambda. Lorsque vous souhaitez accéder au code de votre fonction, vous appelez l'GetFunctionAPI. Cette API répond par `Code.Location` une URL présignée valide pendant 10 minutes (ce délai d'expiration correspond au comportement actuel et non à un contrat publié). Si vous ne voulez pas le code, vous pouvez utiliser une combinaison de GetFunctionConfigurationGetFunctionConcurrency, et ListTagspour récupérer les autres données renvoyées par`GetFunction`.

L'URL renvoyée n'est pas signée avec les informations d'identification de l'utilisateur actuellement connecté, mais au nom de l'utilisateur par Lambda. Pour cette raison, les clés de condition (telles que`aws:SourceIP`) appliquées à l'utilisateur actuellement connecté ou aux informations d'identification de session temporaires de l'utilisateur ne s'appliquent pas à l'URL générée. Cela est vrai que les clés de condition soient appliquées GetFunctionuniquement ou qu'elles soient appliquées à toutes les utilisations de l'API AWS pour l'utilisateur ou la session.

La console Lambda utilise GetFunctionégalement l'URL présignée qu'elle renvoie. La console utilise les informations d'identification temporaires associées à l'utilisateur actuellement connecté pour appeler GetFunction. Les détails relatifs à l'obtention de ces informations d'identification temporaires ne sont pas couverts par ce document.

Amazon ECR

Principal : AWS interne

Expiration par défaut : 1 heure

Amazon Elastic Container Registry (Amazon ECR) fournit GetDownloadUrlForLayerl'API, qui renvoie une URL présignée valide pendant une heure et prend en charge le téléchargement d'une seule couche à partir d'une image Amazon ECR. Cependant, cette opération est utilisée par le proxy Amazon ECR et n'est généralement pas utilisée par les utilisateurs pour extraire et envoyer des images.

Amazon Redshift Spectrum

Principal : rôle transmis à CREATE EXTERNAL SCHEMA via `IAM_ROLE`

Expiration par défaut : 1 heure

Amazon Redshift Spectrum utilise le URLs présigné en interne et interdit les restrictions sur la combinaison du compartiment et du rôle Amazon Redshift qui limiteraient le présigné. URLs Vous pouvez utiliser une `s3:signatureAge` valeur de 16 minutes, mais les valeurs très faibles ne sont pas fiables. La valeur minimale que vous pouvez utiliser dépend du moment et de la taille de votre requête. Bien qu'une valeur inférieure à 16 minutes fonctionne pour de nombreux scénarios, elle doit être testée. Le rôle peut et doit être limité à l'utilisation exclusive de Redshift Spectrum, qui ne divulgue pas ce URLs qu'il génère, atténuant ainsi la justification typique des valeurs d'expiration inférieures.

Amazon SageMaker AI Studio

Amazon SageMaker AI Studio prend en charge deux actions d'API : CreatePresignedDomainUrlet CreatePresignedNotebookInstanceUrl. Toutefois, elles APIs ne sont pas liées à la fonctionnalité d'URL présignée de Signature version 4. Ils APIs créent une URL qui utilise un `authToken` paramètre, mais ils ne prennent en charge aucun des paramètres de requête standard de Signature Version 4.

`authToken`est un mécanisme différent mais présente des similitudes avec le pré-signé URLs. Il est envoyé sous forme de paramètre de chaîne de requête et prend en charge un délai d'expiration de 5 minutes.

SageMaker L'IA prend en charge les restrictions du réseau. Si vous limitez l'`sagemaker:CreatePresignedDomainUrl`action, celle-ci s'applique à la fois à l'appel CreatePresignedDomainUrlet à l'utilisation de l'URL générée. Si une URL est générée à partir d'un réseau valide puis envoyée par un réseau non valide, l'appel d'API pour générer l'URL aboutit, mais la demande d'envoi de l'URL échoue. Il en va de CreatePresignedNotebookInstanceUrlmême pour l'`sagemaker:CreatePresignedNotebookInstanceUrl`action.

Pour plus d'informations, consultez la documentation sur l'SageMaker IA.

Annexe B : Comment les contrôles relatifs à la présignature URLs s'appliquent Services AWS

Cette annexe décrit les interactions entre les commandes Services AWS présignées par l' URLsutilisateur, comme décrit dans l'annexe A, et les commandes décrites plus haut dans ce guide.

Rambarde pour S3 : SignatureAge

La console Amazon S3 n'est pas perturbée par l'expiration maximale de 5 minutes définie par la clé de `s3:signatureAge` condition. La console Amazon S3 génère une signature présignée URLs lorsque vous cliquez sur le bouton Télécharger et applique son propre délai d'expiration de 5 minutes. Une durée maximale inférieure à 2 minutes peut créer des défaillances aléatoires en fonction de la synchronisation de l'horloge et des latences.

Amazon S3 Object Lambda utilise un délai d'expiration de 61 secondes. Par conséquent, le fait de définir des conditions sur une `s3:signatureAge` valeur de 61 secondes ou plus ne provoquera aucune interruption. Des durées plus courtes peuvent être moins fiables et provoquer des pannes intermittentes.

Amazon S3 Cross-region CopyObjectn'est pas perturbé par une expiration maximale de 5 minutes. Cependant, des durées plus courtes peuvent créer des défaillances aléatoires en fonction de la synchronisation des horloges et des latences.

In AWS Lambda, GetFunctionfournit une URL vers des objets extérieurs au compte client, de sorte que les politiques du client n'affectent pas les objets générés URLs.

Amazon Redshift Spectrum a été testé avec `s3:signatureAge` une condition de 16 minutes. Cependant, des durées plus courtes peuvent entraîner des perturbations.

Garde-corps pour S3:AuthType lorsque les restrictions du réseau ne sont pas utilisées

La console Amazon S3 est généralement affectée par le `s3:authType` garde-corps. La console est acheminée vers Amazon S3 en fonction de la configuration du réseau local. Si le réseau local est acheminé vers Amazon S3 conformément aux restrictions du réseau, la console Amazon S3 fonctionnera toujours. Toutefois, s'il est acheminé via un proxy ou via Internet public d'une manière

non autorisée, son utilisation sera bloquée. Cependant, le blocage de l'utilisation est probablement l'objectif de cette politique.

Amazon S3 Object Lambda est affecté si la fonction Lambda n'est pas connectée à un VPC approprié. Dans cette configuration, le VPC doit disposer d'une passerelle NAT, non pas pour accéder au compartiment S3, mais pour appeler. WriteGetObjectResponse

Amazon S3 Interregion CopyObjectest perturbé si ce garde-fou est appliqué à une politique de compartiment sans l'exception recommandée concernant le moment où elle **`aws:viaAWSService`** est vraie.

Amazon Redshift Spectrum est affecté par `s3:authType` le garde-corps sauf si un routage VPC amélioré est utilisé. À l'heure actuelle, Redshift Spectrum prend en charge le routage VPC amélioré uniquement avec les clusters sans serveur, et non avec les clusters provisionnés.

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Pour être averti des mises à jour à venir, abonnez-vous à un fil RSS.

Modification	Description	Date
Mises à jour et clarifications	Dans la section Garde-corps supplémentaires, des informations ont été ajoutées sur les exceptions RCPs et ont été clarifiées.	8 août 2025
Publication initiale	—	23 juillet 2024

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.