

AWS Academy Cloud Security

Foundations

Lab 3.1 Report: Using Resource -Based Policies to Secure an S3 Bucket

hamza taki
3ACI Info

1. Présentation du laboratoire et objectifs

Ce TP a pour vocation d'éclaircir la distinction entre les stratégies basées sur l'identité (IAM) et celles basées sur les ressources (spécifiquement pour les buckets S3). L'étude porte également sur l'interaction entre ces deux types de politiques afin de mettre en place un contrôle d'accès précis.

Objectifs clés :

- Assimiler la différence fondamentale entre les politiques liées aux identités et celles liées aux ressources.
- Illustrer le mécanisme d'emprunt de rôle (*Assume Role*) permettant à un utilisateur IAM d'ajuster ou d'élever ses privilèges.
- Observer les conséquences concrètes de ces configurations sur l'affichage et les opérations possibles au sein de la console AWS.

Architecture :

- **Architecture initiale :**

Le point de départ est l'utilisateur devuser, membre du groupe DeveloperGroup.

- **Architecture cible :**

La finalité est l'exploitation de rôles IAM dédiés (BucketAccessRole et

OtherBucketAccessRole) pour interagir avec des buckets S3 aux accès restreints.

2. Description détaillée des tâches

Tâche 1 : Accès à la console

- **Action :** Authentifiez-vous sur le portail de gestion AWS à l'aide des identifiants attribués à l'utilisateur devuser.

The screenshot shows the AWS Lab interface. On the left, a list of instructions for logging in as the IAM user 'devuser'. On the right, a panel displays the accumulated lab time (00:02:00) and a table of IAM user details.

3. Log in as the IAM user named *devuser*:

- Choose the **AWS Details** link at the top of the page.
- Copy the **IAMUserLoginURL** value, and load it in a new browser tab.
- For **IAM user name**, enter **devuser**
- For **Password**, enter the **IAMUserPassword** value from the AWS Details panel on the lab instructions page.
- Choose **Sign in**.

The AWS Management Console displays.

Warning: To avoid issues, do NOT change the Region during this lab unless instructed.

AWS Details	
IAMUserPassword	igw-0eafb12d095d6f7ef
AccountID	851725200325
IAMUserLoginURL	https://851725200325.signin.aws.a
Region	us-east-1

The screenshot shows the Amazon Lightsail sign-in page. The left sidebar contains the 'IAM user sign in' form with fields for Account ID, IAM username, and Password. The main content area features the 'Amazon Lightsail' logo and a 'Learn more' button.

← Previous

Next →

Provide feedback Multi-session disabled English

aws

IAM user sign in

Account ID or alias ([Don't have?](#))

851725200325

☒ Remember this account

IAM username

devuser

Password

igw-0eafb12d095d6f7ef

☒ Show Password [Having trouble?](#)

Sign in

Sign in using root user email

[Create a new AWS account](#)

Amazon Lightsail

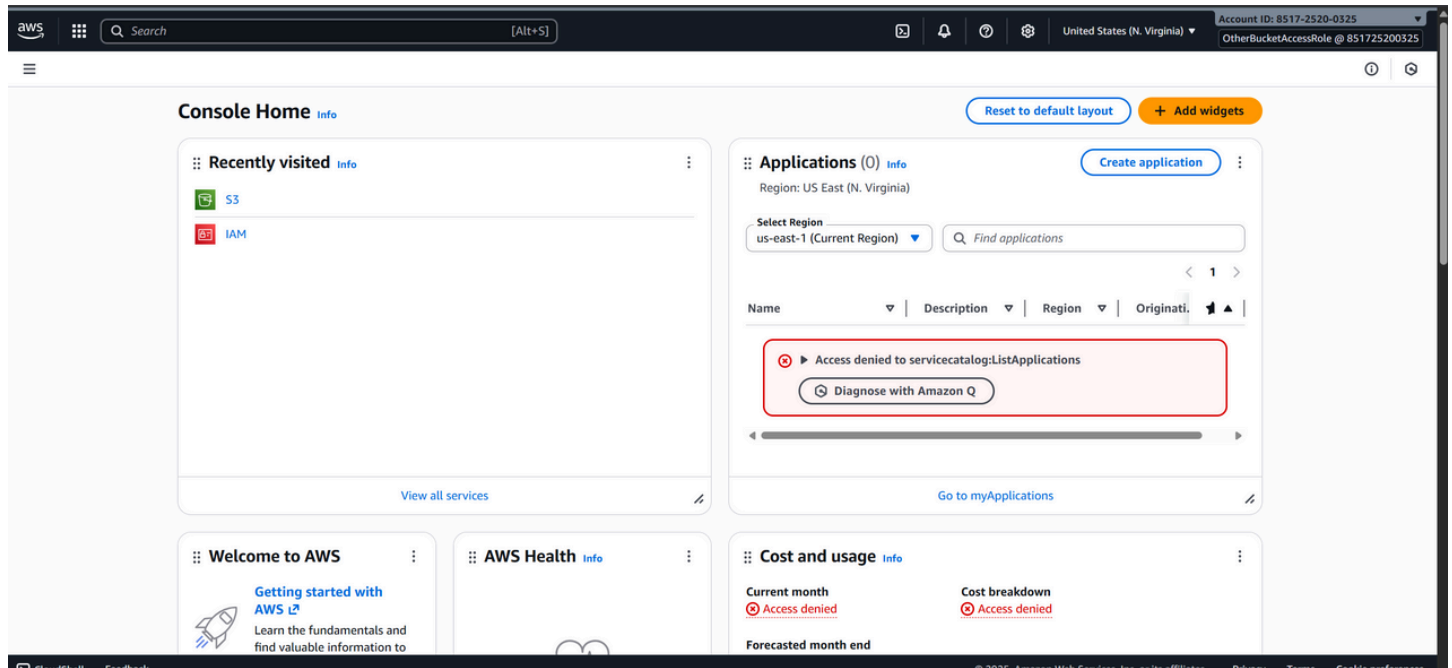
Lightsail is the easiest way to get started on AWS

[Learn more »](#)

By continuing, you agree to [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

Observation :

La connexion s'est effectuée avec succès, nous amenant directement sur la page d'accueil de la console AWS.



Tâche 2 : Tentative d'accès en lecture aux services AWS

Action 1 : Accéder à Amazon EC2

- **Étape** : Rendez-vous sur le tableau de bord EC2 pour tenter de visualiser les instances existantes.
- **Observation** : Rendez-vous sur le tableau de bord EC2 pour tenter de visualiser les instances existantes.
- **Raison** :
Ce blocage est dû au fait que l'utilisateur devuser ne dispose pas des autorisations IAM requises (telles que ec2:DescribeInstances) pour interroger les services EC2.

Action 2 : Accès à Amazon S3

- **Étape** : Navigation vers la console Amazon S3.

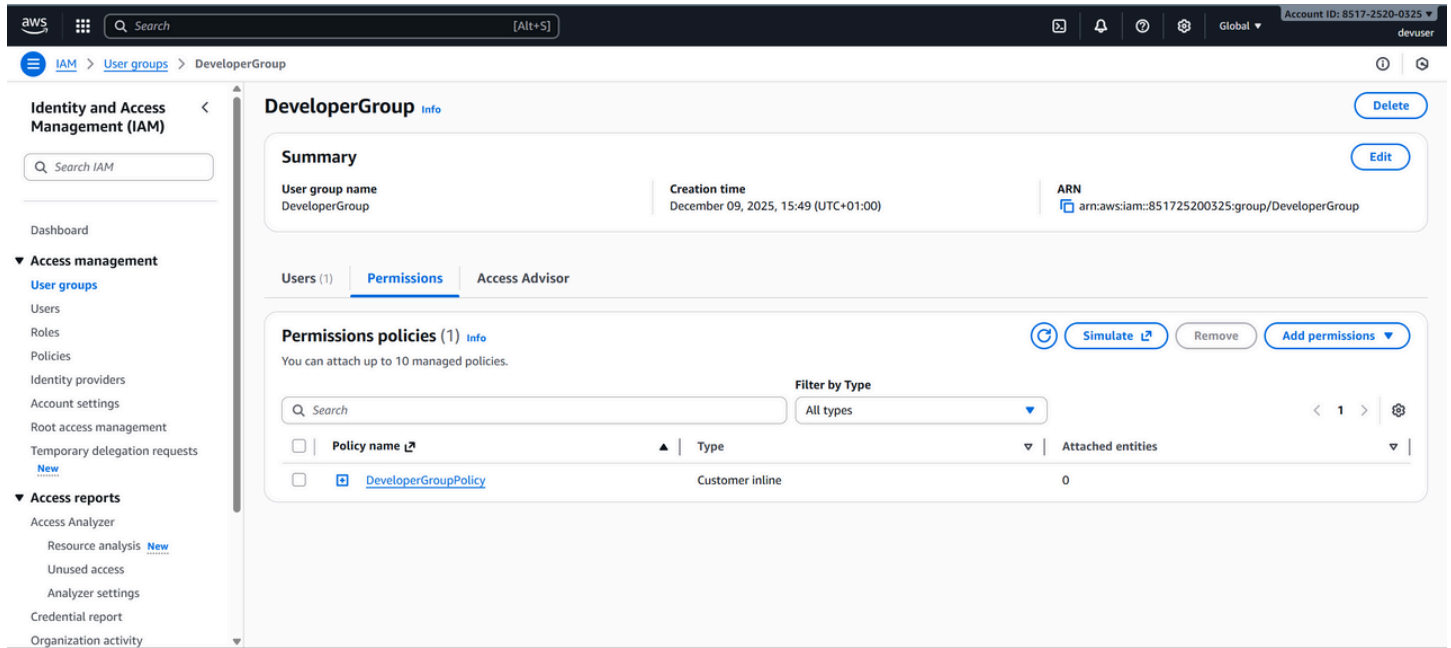
Name	AWS Region	Creation date
c185581a4806215113068836t1w851725200325-bucket1-hyyatx8ivrs	US East (N. Virginia) us-east-1	December 9, 2025, 15:49:29 (UTC+01:00)
c185581a4806215113068836t1w851725200325-bucket2-vegud3zjklj	US East (N. Virginia) us-east-1	December 9, 2025, 15:49:29 (UTC+01:00)
c185581a4806215113068836t1w851725200325-bucket3-z2mwxfu61yl	US East (N. Virginia) us-east-1	December 9, 2025, 15:49:29 (UTC+01:00)

- **Observation** : Bien que la liste des buckets (bucket1, bucket2, bucket3) se soit affichée, la colonne « Access » indiquait « Insufficient permissions » ou « Error ».
- **Raison** : L'utilisateur devuser bénéficie de l'autorisation ListAllMyBuckets, ce qui rend les noms des buckets visibles. Toutefois, l'absence de permissions supplémentaires l'empêche de lire les configurations spécifiques ou de déterminer le statut d'accès public de ces buckets.

Tâche 3 : Analyse de la politique basée sur l'identité

Action :

- Accès à la console IAM → *Users* → *devuser* → *Groups* → *DeveloperGroup* → *Permissions*.



- **Ouverture de la politique DeveloperGroupPolicy afin d'examiner le fichier JSON.**

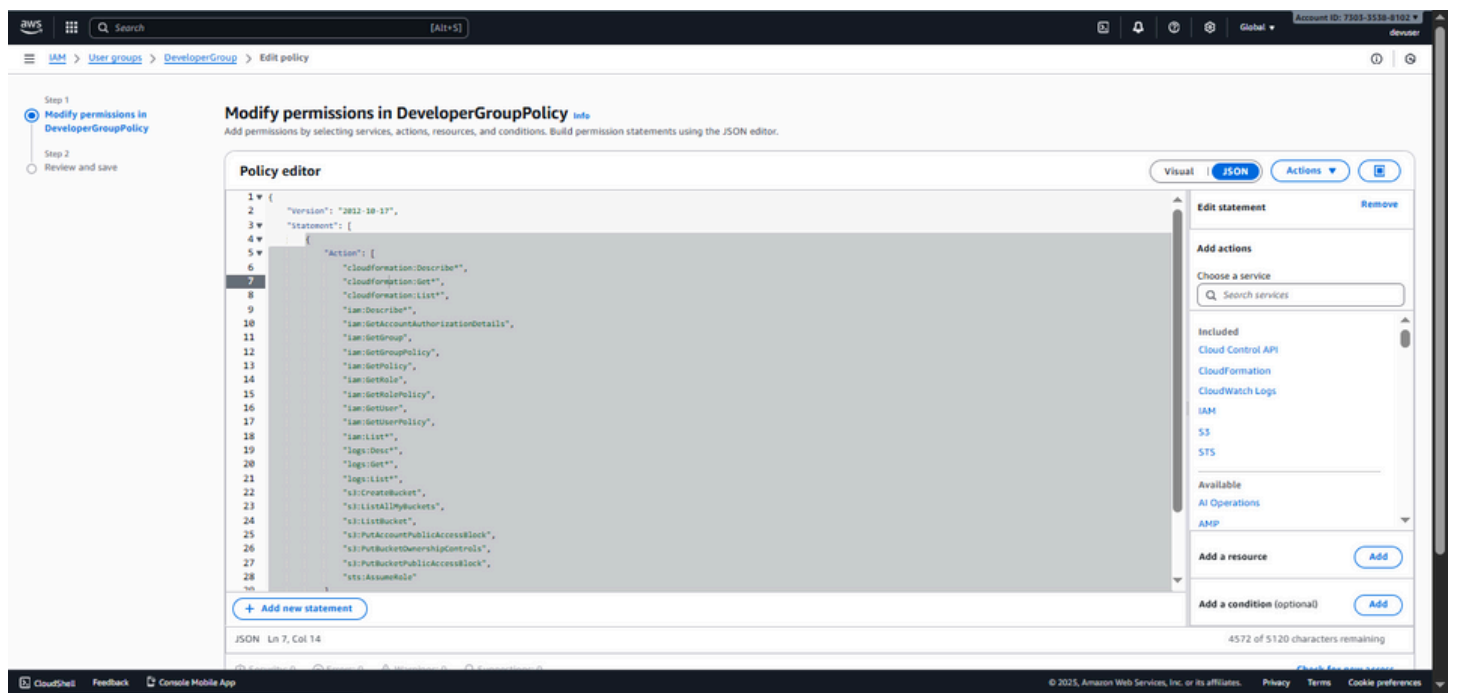
Analyse de la politique :

Il s'agit d'une politique basée sur l'identité (*Identity-Based Policy*) appliquée au groupe.

Actions autorisées : *Le document autorise les actions `s3:ListAllMyBuckets`, `s3:CreateBucket` et `s3:ListBucket`.*

- **Actions manquantes ou non autorisées :** L'analyse révèle l'absence totale de permissions `ec2`: (ce qui justifie les erreurs d'accès rencontrées sur le service EC2) ainsi que l'absence de la permission `s3:PutObject` au niveau global.

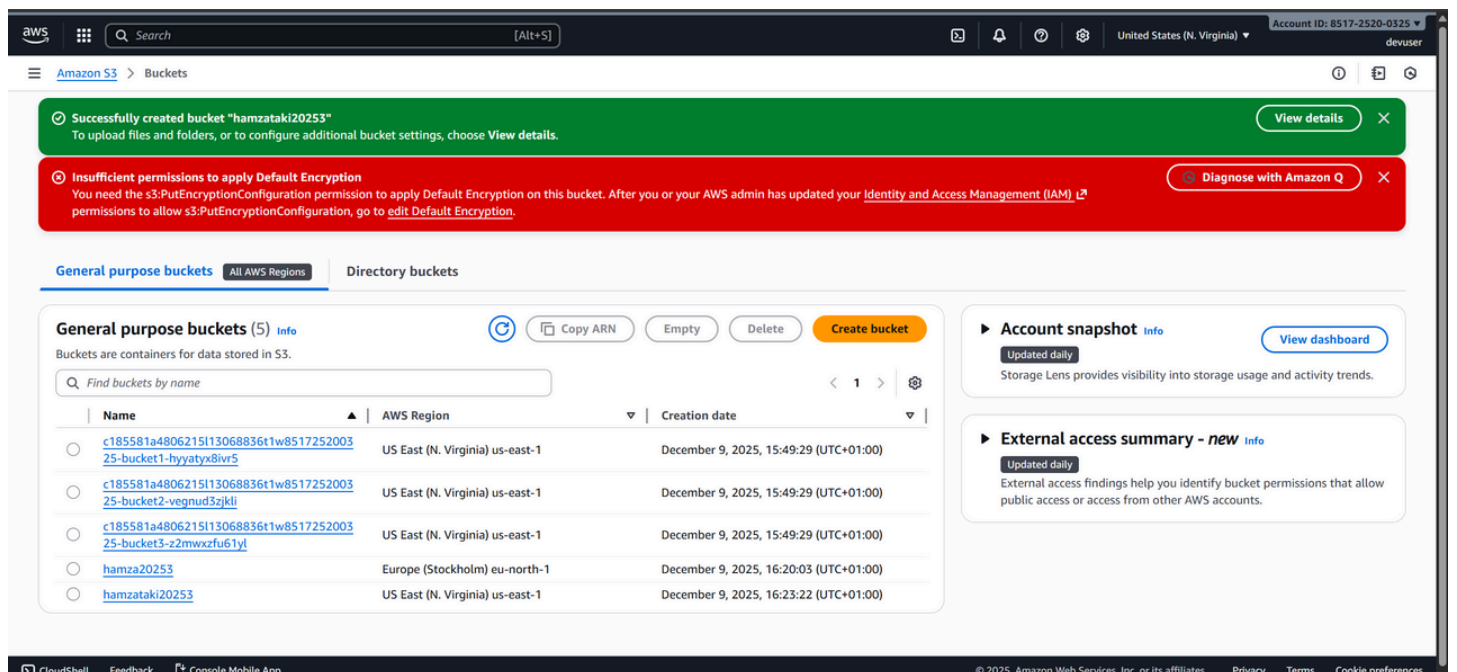
Cette configuration explique le comportement observé précédemment : l'utilisateur a le droit de voir la liste des buckets, mais il est bloqué pour toute autre opération administrative ou modification.



Tâche 4 : Tentative d'accès en écriture aux services AWS

Action 1 : Création d'un bucket

- **Étape** : Clic sur « Create bucket » puis création d'un bucket nommé *to4973*.



- **Résultat** : Succès.
- Raison : La politique *DeveloperGroupPolicy* autorise explicitement l'action *s3:CreateBucket*.

Action 2 : Téléversement d'un objet

- **Étape** : Ouverture du bucket nouvellement créé, puis tentative de téléverser *Image1.jpg*.
- **Résultat** : Échec (message *Upload failed*).
- **Raison** : Bien que nous soyons propriétaires du bucket que nous venons de créer, la politique IAM associée à *devuser* ne contient pas l'autorisation *s3:PutObject*. Dans AWS, la permission de créer une ressource n'implique pas automatiquement la permission de modifier son contenu si la politique reste restrictive.

Upload failed
For more information, see the Error column in the Files and folders table. [Diagnose with Amazon Q](#)

Upload: status [Close](#)

After you navigate away from this page, the following information is no longer available.

Summary

Destination	Succeeded	Failed
s3://hamzataki20253	0 files, 0 B (0%)	1 file, 730.0 B (100.00%)

[Files and folders](#) [Configuration](#)

Files and folders (1 total, 730.0 B)

Find by name

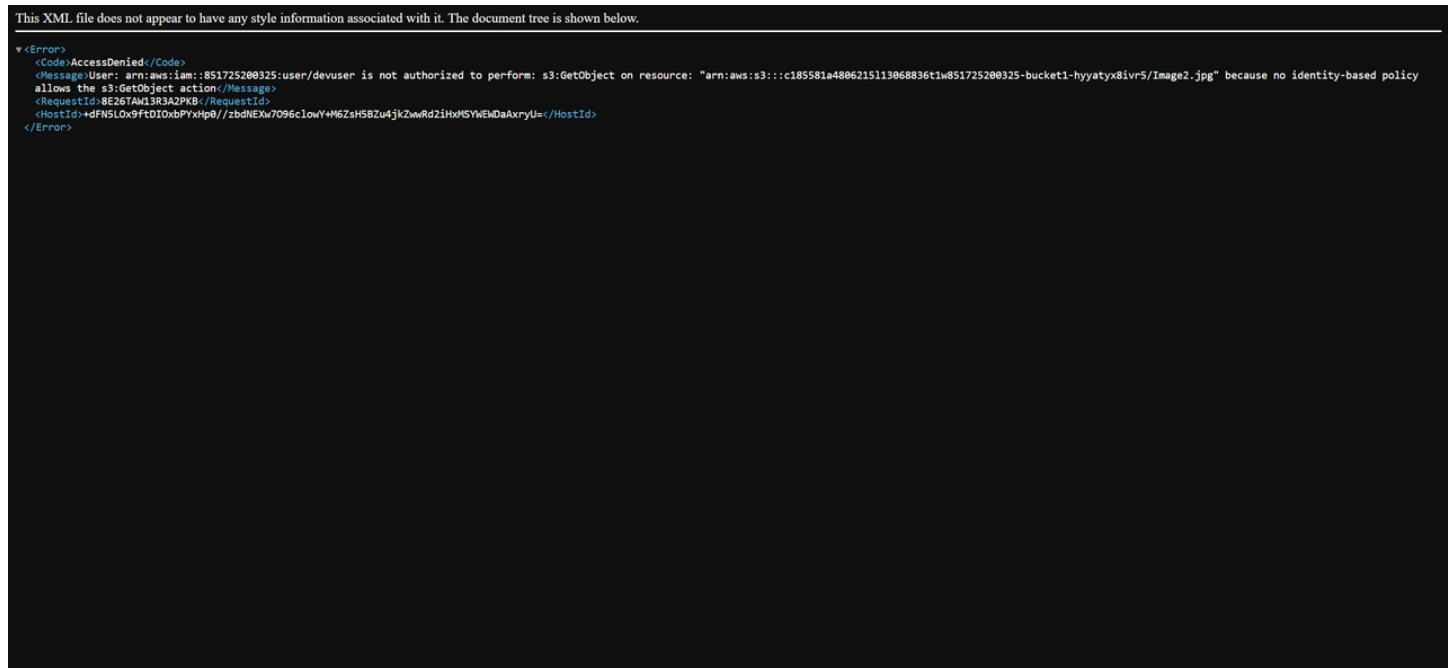
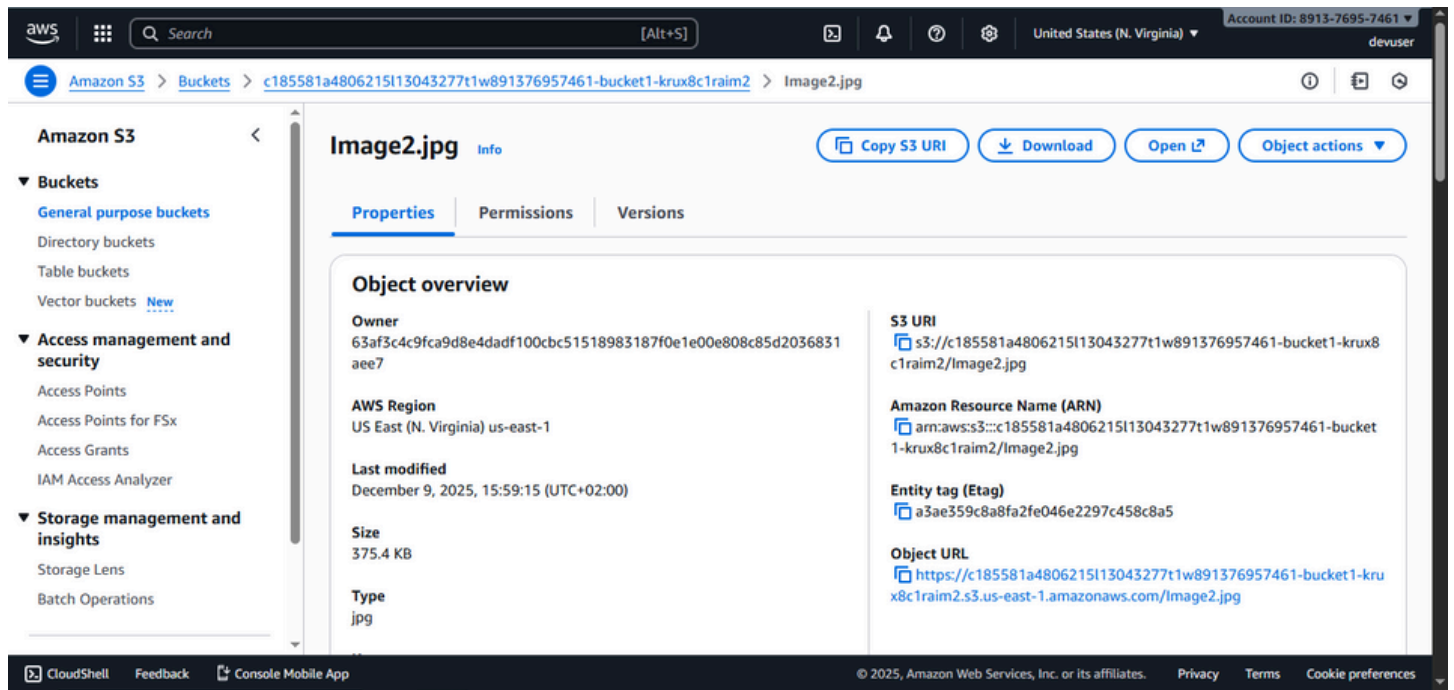
Name	Folder	Type	Size	Status	Error
DeveloperGroupPolicy.json	-	application/json	730.0 B	Failed	Access denied

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Tâche 5 : Assumer un rôle IAM

Test préalable : tentative de téléchargement avant d'assumer le rôle

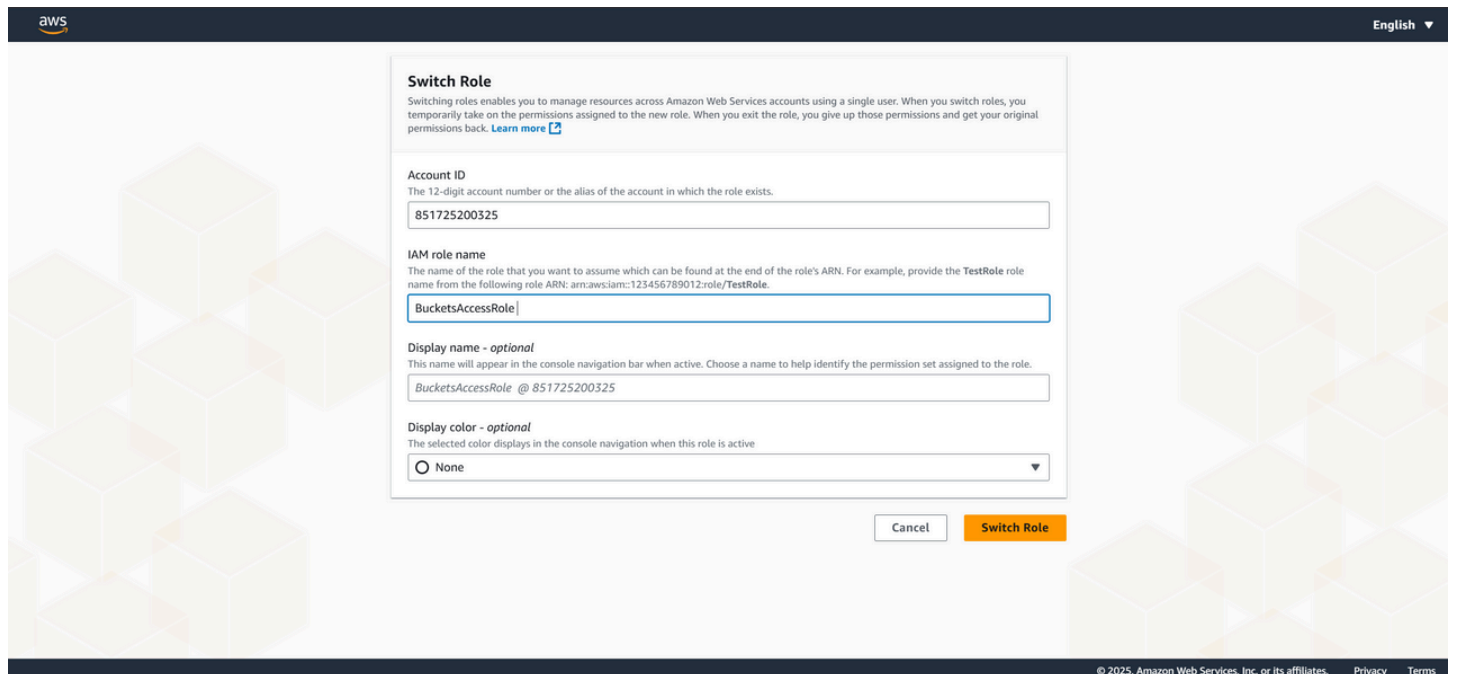
- **Action** : Navigation vers *bucket1* puis tentative de téléchargement d'un fichier existant.
- **Résultat** : Échec (erreur *Access Denied*).
-



Raison :

La politique basée sur l'identité de *devuser* (*DeveloperGroupPolicy*) autorise *s3:ListAllMyBuckets* et *s3:ListBucket*, mais ne contient pas la permission *s3:GetObject*, indispensable pour télécharger un objet depuis un bucket. Cela confirme que les permissions par défaut de l'utilisateur limitent l'accès en lecture au contenu des objets, même s'il peut voir la liste des buckets.

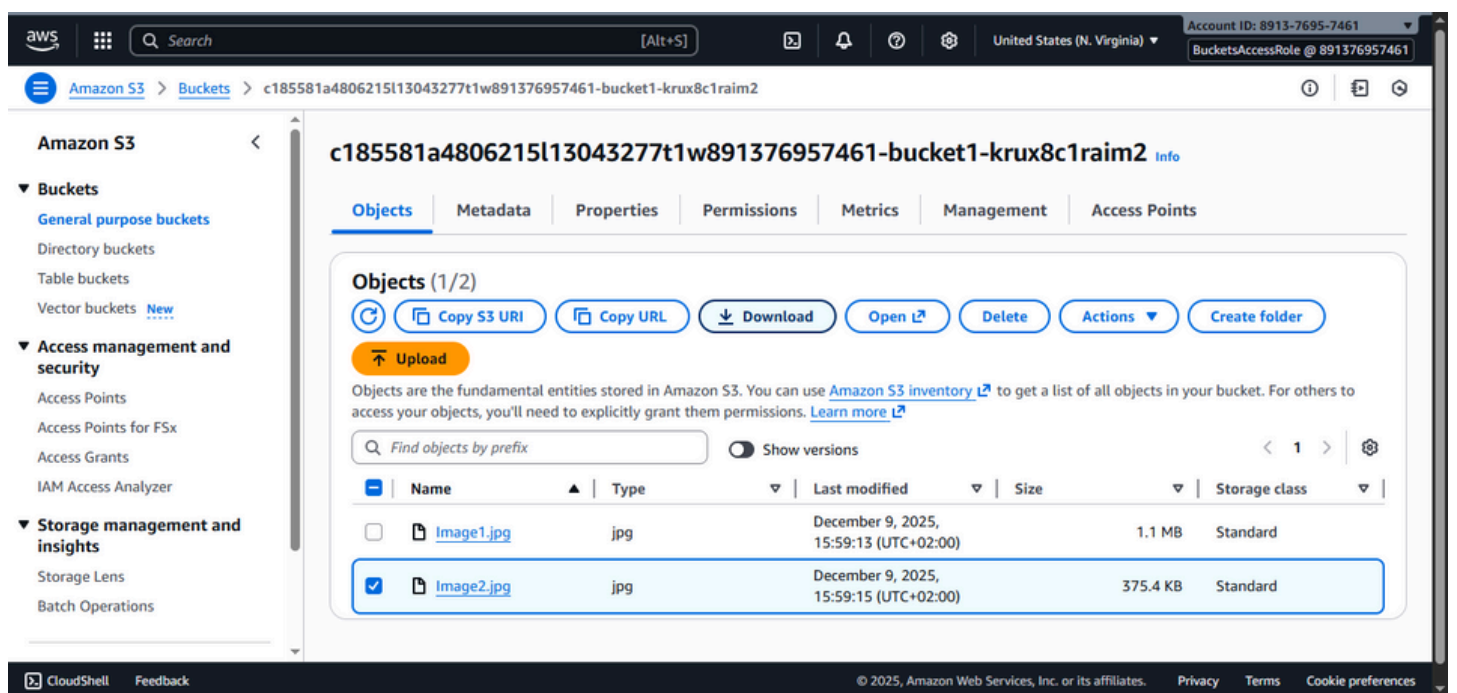
Action : (tu peux m'envoyer la suite pour la reformuler)



Action :

- Ouverture du menu utilisateur en haut à droite puis clic sur Switch Role.
- Saisie de l'ID du compte ainsi que du nom du rôle : `BucketsAccessRole`.
- Changement visible : L'identité affichée en haut à droite a été remplacée par *BucketsAccessRole*, confirmant que le rôle a bien été assumé.

Test d'accès : (envoie-moi la suite et je te la reformule)



The screenshot shows the AWS IAM console with the 'Roles' page selected. A red box highlights an 'Access denied' error message:

Access denied to iam:ListRoles
 You don't have permission to `iam:ListRoles`. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)

User: arn:aws:sts::891376957461:assumed-role/BucketsAccessRole/devuser
Action: iam:ListRoles
On resource(s): arn:aws:iam::891376957461:role/
Context: no identity-based policy allows the action

The right sidebar shows the user is 'Currently active as' `BucketsAccessRole` with Account ID `8913-7695-7461`. The role history shows `BucketsAccessRole @ 891376957461`.

The screenshot shows the AWS IAM console with the 'BucketsAccessRole' page selected. The 'ListAllBucketsPolicy' is highlighted in the table of attached policies.

Policy name	Type	Attached entities
GetBucketPolicy	Customer inline	0
GrantBucket1Access	Customer inline	0
ListAllBucketsPolicy	Customer inline	0

The 'ListAllBucketsPolicy' JSON is displayed below the table:

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Action": [
6         "s3:ListAllMyBuckets"
7       ],
8       "Resource": "*",
9       "Effect": "Allow"
10    }
11  ]
12 }
```

1. **Bucket 1** : Le téléchargement des objets a fonctionné sans problème.
2. **Bucket 2** : L'upload d'objets était autorisé, car le rôle disposait de la permission `s3:PutObject` spécifiquement sur `bucket2`.

The screenshot shows the AWS S3 console interface. At the top, there's a green banner indicating 'Upload succeeded' with a message: 'For more information, see the Files and folders table.' Below this, the 'Upload: status' section shows a summary of the upload. The destination is 's3://c185581a48062151130432771w449039445877-bucket2-ylsfa5cpevnb'. The upload was successful, with 1 file (375.4 KB) uploaded at 100.00%. There are no failed uploads. Below the summary, there are tabs for 'Files and folders' and 'Configuration'. The 'Files and folders' tab is active, showing a table with 1 total file (375.4 KB). The table has columns for Name, Folder, Type, Size, Status, and Error. The single file listed is 'image2.jpg', which is an image/jpeg file, 375.4 KB in size, and its status is 'Succeeded'.

Name	Folder	Type	Size	Status	Error
image2.jpg	-	image/jpeg	375.4 KB	Succeeded	-

Analyse :

En assumant le rôle, nous avons temporairement abandonné les permissions de *devuser* pour adopter celles de *BucketsAccessRole*. Ce rôle possédait une politique dédiée autorisant la lecture et/ou l'écriture sur certains buckets, des permissions que *devuser* ne détenait pas initialement.

Voici la reformulation :

Tâche 6 : Comprendre les politiques basées sur les ressources

Action :

- En étant connecté avec le rôle *bucket-access-role*, nous avons navigué vers *Bucket 2* puis ouvert l'onglet **Permissions**.
- Nous avons consulté la **Bucket Policy**.

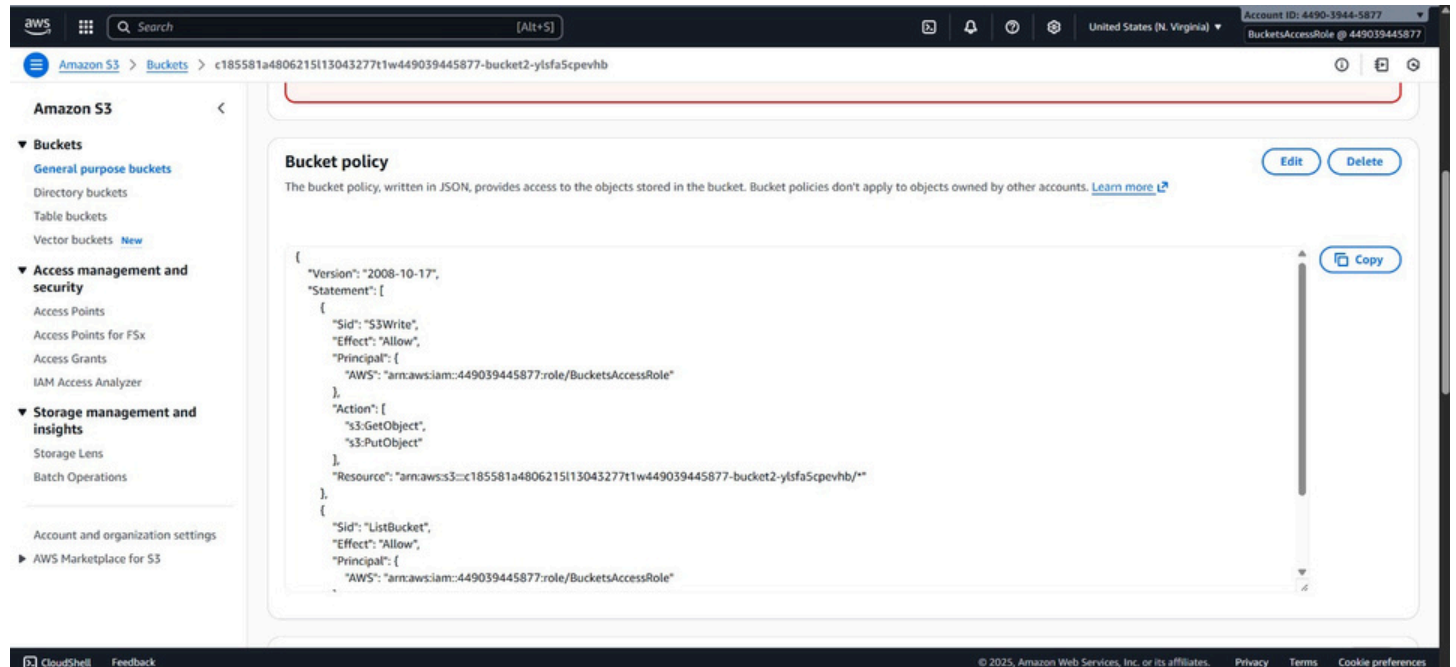
Analyse de la politique :

Il s'agit d'une **Resource-Based Policy**, c'est-à-dire une politique attachée directement à la ressource (le bucket S3), contrairement aux politiques basées sur l'identité qui sont liées à un utilisateur ou à un rôle IAM.

- La politique mentionnait explicitement *BucketsAccessRole* comme **Principal** autorisé.
- Cela confirme que l'accès à une ressource peut être accordé soit par la politique de l'utilisateur (Identity-Based Policy), soit par la politique de la ressource (Resource-Based Policy) lorsqu'il s'agit du même compte.
- Dans un contexte inter-comptes (cross-account), les deux types de politiques sont souvent nécessaires, mais dans ce laboratoire l'objectif principal était

d'observer comment le rôle obtenait l'autorisation via la bucket policy.

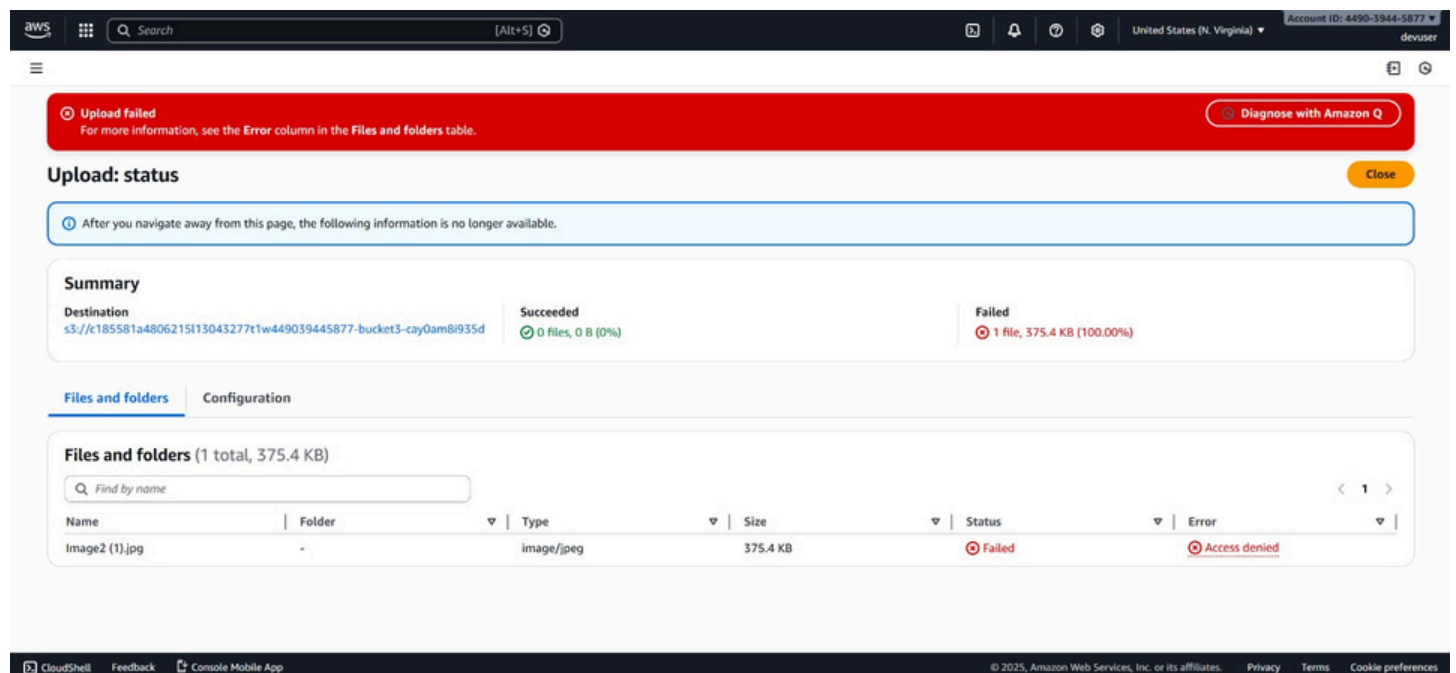
Si tu veux, envoie-moi la dernière partie (conclusion, résumé, etc.) et je la reformule aussi.

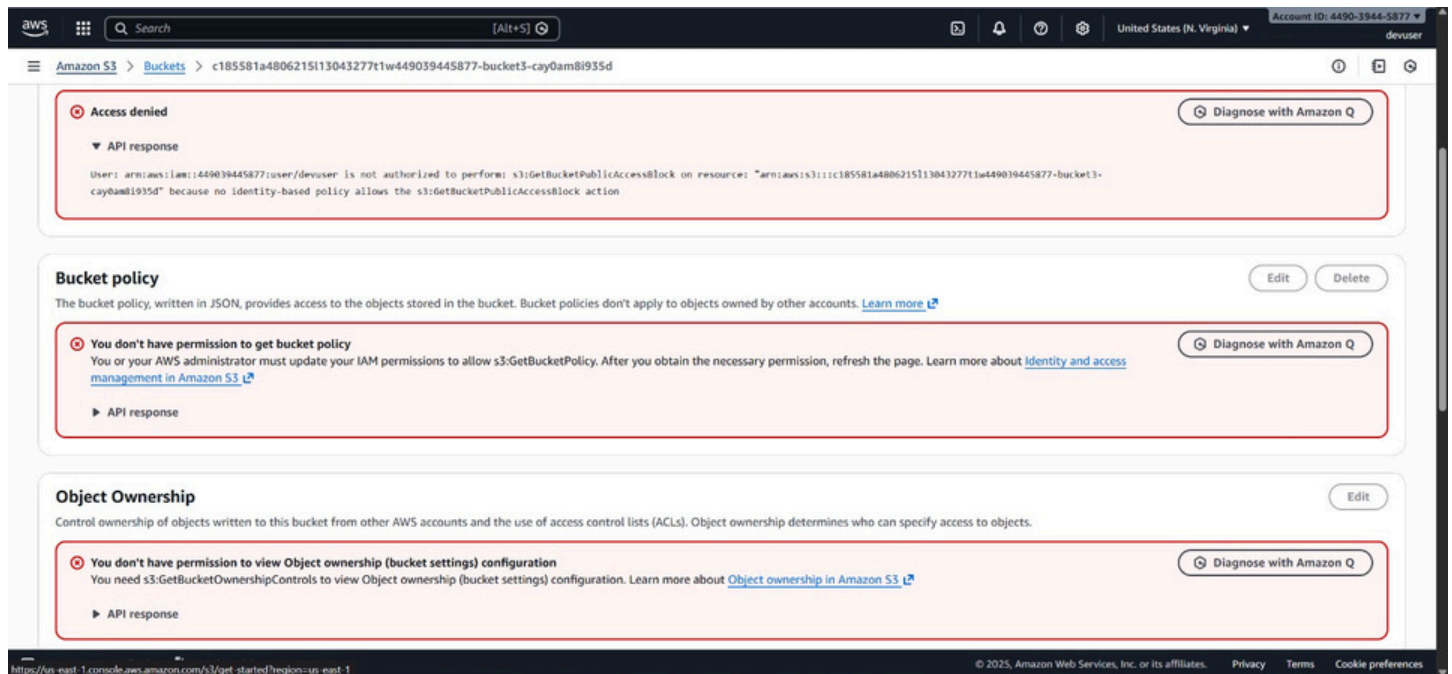


3. Challenge Task: : Téléversement dans Bucket 3

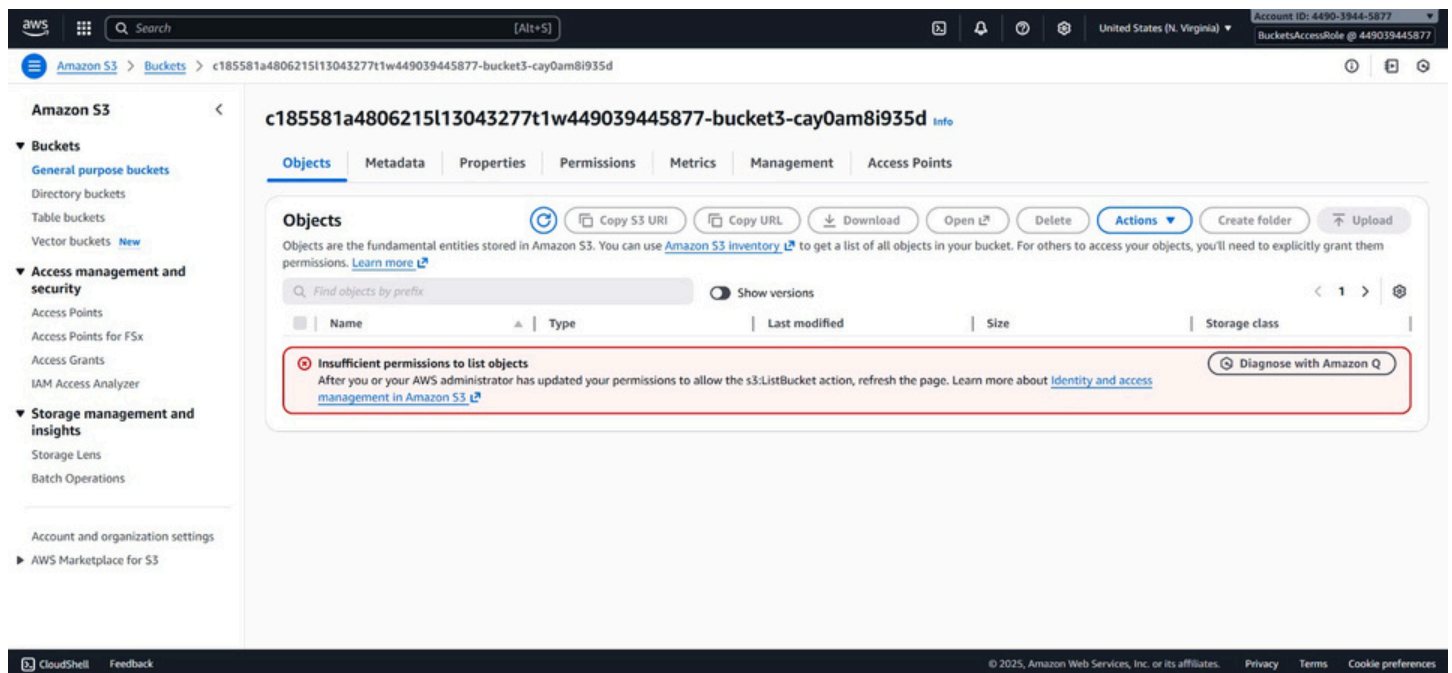
Objectif : Envoyer *Image2.jpg* dans *bucket3*.

Tentative 1 : En tant que *devuser*

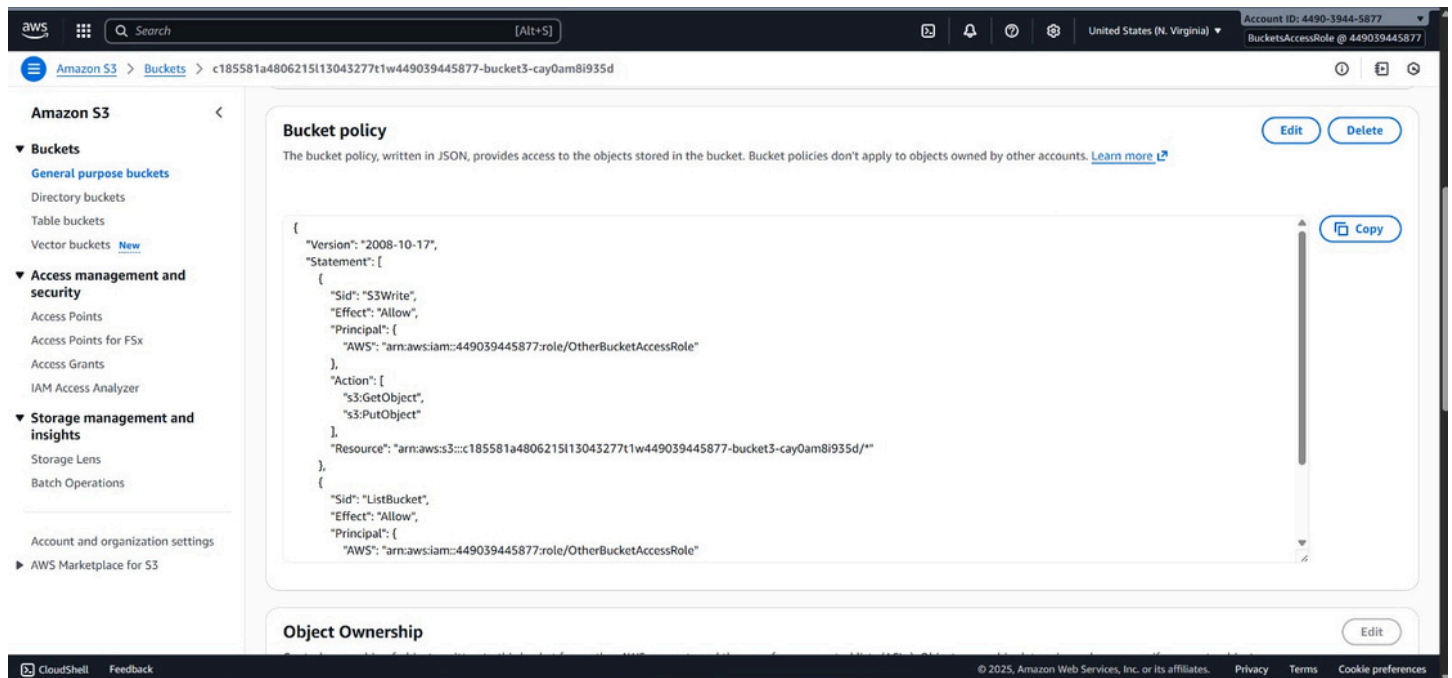




Tentative 2 : En utilisant le rôle *BucketAccessRole*

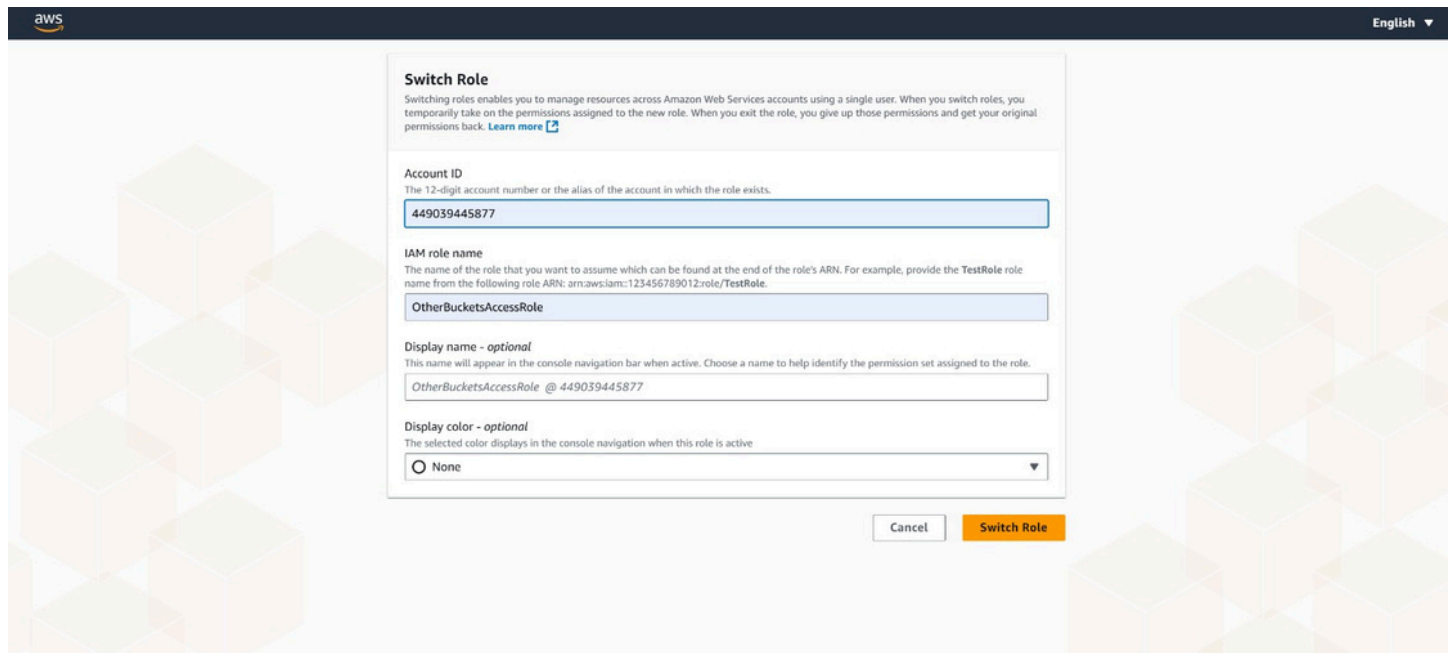


Résultat : Échec. La politique associée au rôle ne contenait aucune permission permettant d'interagir avec *bucket3*. Toutefois, nous pouvions consulter la bucket policy.



Solution :

1. Nous avons identifié un second rôle disponible dans le compte : **OtherBucketAccessRole**.
2. Nous avons de nouveau utilisé la fonctionnalité **Switch Role** pour passer du rôle *BucketsAccessRole* au rôle *OtherBucketAccessRole*.



1. Nous avons navigué vers *bucket3*.
2. Nous avons cliqué sur **Upload**, sélectionné *Image2.jpg*, puis validé l'envoi.
3. **Résultat** : Succès.

Explication :

Le rôle **OtherBucketAccessRole** possédait une politique autorisant

explicitement l'action `s3:PutObject` sur la ressource `bucket3`. En assumant ce rôle spécifique, nous avons obtenu les permissions nécessaires pour réaliser la tâche.

aws

Search

[Alt+S]

United States (N. Virginia)

Account ID: 4490-3944-5877

OtherBucketAccessRole @ 449039445877

Upload succeeded

For more information, see the Files and folders table.

Close

Upload: status

After you navigate away from this page, the following information is no longer available.

Summary

Destination

s3://t185581a4806215113043277t1w449039445877-bucket3-cay0am8i935d

Succeeded

1 file, 375.4 KB (100.00%)

Failed

0 files, 0 B (0%)

Files and folders

Configuration

Files and folders (1 total, 375.4 KB)

Find by name

< 1 >

Name	Folder	Type	Size	Status	Error
Image2.jpg	-	image/jpeg	375.4 KB	Succeeded	-

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Submit

Submission Report

Grades

Total score	15/15
-------------	-------

TASK 4 - Create bucket	5/5
------------------------	-----

TASK 5 - Uploaded object	5/5
--------------------------	-----

CHALLENGE TASK - Uploaded object	5/5
----------------------------------	-----

Conclusion

Ce laboratoire a clairement démontré la flexibilité et la granularité du contrôle d'accès dans AWS :

1. **Les politiques d'identité** associées à *devuser* offraient un accès de base, mais limitaient les actions sensibles, comme la création d'instances EC2 ou l'écriture dans des buckets non autorisés.
2. **Les rôles IAM** nous ont permis d'élever temporairement nos permissions afin d'accomplir des tâches spécifiques (gestion ciblée de certains buckets) sans avoir à attribuer ces privilèges de manière permanente à l'utilisateur.
3. **Les politiques basées sur les ressources**, appliquées directement aux buckets S3, ont ajouté une couche supplémentaire de contrôle, en définissant précisément quels *principals* (utilisateurs ou rôles) étaient autorisés à interagir avec les données du bucket.