

# 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, there's a sidebar with a dropdown for 'EN-US'. The main area contains instructions for logging in as the IAM user 'devuser' with the provided details. On the right, the 'AWS Details' panel is open, showing accumulated lab time (00:02:00), SSH key options, and a table with account information:

IAMUserPassword	igw-0eafb12d095d6f7ef
AccountID	851725200325
IAMUserLoginURL	<a href="https://851725200325.signin.aws.amazon.com">https://851725200325.signin.aws.amazon.com</a>
Region	us-east-1

The screenshot shows two side-by-side browser windows. The left window is the 'IAM user sign in' page, where the Account ID is '851725200325', the IAM username is 'devuser', and the Password is 'igw-0eafb12d095d6f7ef'. The right window is the 'Amazon Lightsail' landing page, featuring a cartoon robot and the text 'Lightsail is the easiest way to get started on AWS'.

### Observation :

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

The screenshot shows the AWS Console Home page. At the top, there's a search bar and navigation links for United States (N. Virginia) and Account ID: 8517-2520-0325. Below the header, there are several sections:

- Recently visited:** Shows S3 and IAM.
- Applications (0):** Region: US East (N. Virginia). A red box highlights an error message: "Access denied to servicecatalog>ListApplications".
- Welcome to AWS:** Includes a "Getting started with AWS" link and a "CloudWatch Feedback" button.
- AWS Health:** Shows a green status icon.
- Cost and usage:** Current month and Forecasted month end sections, both showing "Access denied".

## 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.

The screenshot shows the AWS EC2 console with the following details:

- Left sidebar:** Shows navigation links for EC2, Dashboard, Instances, Images, Elastic Block Store, and Network & Security.
- Main Content Area:**
  - Resources:** A grid of EC2 resources: Instances (running) 0, Auto Scaling Groups (API Error), Capacity Reservations (API Error), Dedicated Hosts (API Error), Elastic IPs (API Error), Instances (API Error), Key pairs (API Error), Load balancers (API Error), Placement groups (API Error), Security groups (API Error), Snapshots (API Error), and Volumes (API Error).
  - Launch instance:** Buttons for "Launch instance" and "Migrate a server". Note: Your instances will launch in the Europe (Stockholm) Region.
  - Instance alarms:** A note: "User: arn:aws:ia".
  - Service health:** An error occurred: "An error occurred retrieving service health information".
  - Zones:** A table with columns "Zone name" and "Zone ID", showing an error message: "An error occurred".
- Right Sidebar:** Account attributes (with an error message about a default VPC), Settings (with options like Data protection and security, Allowed AMIs, Zones, EC2 Serial Console, Default credit specification, EC2 console preferences), and Explore AWS (with a note about AWS Graviton2).

## Action 2 : Accès à Amazon S3

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

The screenshot shows the AWS S3 console with the following details:

- Left sidebar:** Shows navigation links for Amazon S3, Buckets, Access management and security, and Storage management and insights.
- Main Content Area:**
  - General purpose buckets:** A list of buckets:
 

Name	AWS Region	Creation date
c185581a4806215l13068836t1w851 725200325-bucket1-hyyatx8ivr5	US East (N. Virginia) us-east-1	December 9, 2025, 15:49:29 (UTC+01:00)
c185581a4806215l13068836t1w851 725200325-bucket2-vegnud3zjkl	US East (N. Virginia) us-east-1	December 9, 2025, 15:49:29 (UTC+01:00)
c185581a4806215l13068836t1w851 725200325-bucket3-z2mwxzfu61yl	US East (N. Virginia) us-east-1	December 9, 2025, 15:49:29 (UTC+01:00)
  - Account snapshot:** Updated daily, provides visibility into storage usage and activity trends.
  - External access summary - new:** Updated daily, helps identify bucket permissions that allow public access or access from other AWS accounts.

- **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*.

The screenshot shows the AWS IAM console. On the left, the navigation pane is open with 'User groups' selected under 'Access management'. The main content area is titled 'DeveloperGroup' and shows a summary with the user group name 'DeveloperGroup' and creation time 'December 09, 2025, 15:49 (UTC+01:00)'. An ARN is listed as 'arn:aws:iam::851725200325:group/DeveloperGroup'. Below the summary, there are tabs for 'Users (1)', 'Permissions' (which is selected), and 'Access Advisor'. Under 'Permissions policies (1)', it shows a single policy named 'DeveloperGroupPolicy' which is a 'Customer inline' policy. There are buttons for 'Edit', 'Delete', 'Simulate', 'Remove', and 'Add 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.

The screenshot shows the AWS IAM Policy editor interface. On the left, there are two tabs: "Step 1: Modify permissions in DeveloperGroupPolicy" (selected) and "Step 2: Review and save". The main area is titled "Modify permissions in DeveloperGroupPolicy" with a sub-instruction "Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor." Below this is the "Policy editor" section containing a JSON code block:

```

1 {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudformation:Describe",
        "cloudformation:Get",
        "cloudformation>List",
        "iam:Describe",
        "iam:GetAccountAuthorizationDetails",
        "iam:GetGroup",
        "iam:GetPolicy",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:GetUser",
        "iam GetUserPolicy",
        "iam:List",
        "logs:BatchGetLogEvents",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "s3:CreateBucket",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:PutAccountPublicAccessBlock",
        "s3:PutBucketOwnershipControls",
        "s3:PutBucketPublicAccessBlock",
        "sts:AssumeRole"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}

```

Below the JSON code, there are three buttons: "+ Add new statement", "JSON - Ln 7, Col 14", and "View details". To the right of the JSON editor is a sidebar with tabs: "Visual", "JSON" (selected), and "Actions". The "Actions" tab has a "Edit statement" button, a "Remove" button, and a "Actions" dropdown menu. The "Actions" dropdown lists several AWS services: Cloud Control API, Cloudformation, CloudWatch Logs, IAM, S3, STS, AI Operations, and AMP. There are also "Add a resource" and "Add a condition (optional)" buttons.

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

### Action 1 : Crédation d'un bucket

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

The screenshot shows the AWS S3 Buckets page. At the top, there is a green success message: "Successfully created bucket 'hamzataki20253'. To upload files and folders, or to configure additional bucket settings, choose View details." Below this, there is a red error message: "Insufficient permissions to apply Default Encryption. You need the s3:PutEncryptionConfiguration permission to apply Default Encryption on this bucket. After you or your AWS admin has updated your Identity and Access Management (IAM) permissions to allow s3:PutEncryptionConfiguration, go to edit Default Encryption." To the right of these messages are "View details" and "Diagnose with Amazon Q" buttons.

The main content area displays a table of General purpose buckets. The table has columns: Name, AWS Region, and Creation date. The table shows five buckets:

Name	AWS Region	Creation date
c185581a480621513068836t1w8517252003 25-bucket1-hyaytx8iv5	US East (N. Virginia) us-east-1	December 9, 2025, 15:49:29 (UTC+01:00)
c185581a480621513068836t1w8517252003 25-bucket2-vegnud5jkl	US East (N. Virginia) us-east-1	December 9, 2025, 15:49:29 (UTC+01:00)
c185581a480621513068836t1w8517252003 25-bucket3-z2mxzxfu61yj	US East (N. Virginia) us-east-1	December 9, 2025, 15:49:29 (UTC+01:00)
hamza20253	Europe (Stockholm) eu-north-1	December 9, 2025, 16:20:03 (UTC+01:00)
hamzataki20253	US East (N. Virginia) us-east-1	December 9, 2025, 16:23:22 (UTC+01:00)

On the right side of the page, there are three cards: "Account snapshot" (Updated daily), "External access summary - new" (Updated daily), and "Storage Lens provides visibility into storage usage and activity trends".

- 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.

The screenshot shows the AWS S3 console interface. At the top, there's a red banner with the message "Upload failed" and a link to "Diagnose with Amazon Q". Below the banner, the "Upload: status" section indicates "Succeeded" (0 files, 0 B (0%)) and "Failed" (1 file, 730.0 B (100.00%) with an "Access denied" error). The "Files and folders" tab is selected, showing a table with one item: "DeveloperGroupPolicy.json" (application/json, 730.0 B, Failed, Access denied). The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App, along with copyright and legal information.

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	

## 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*).
-

The screenshot shows the AWS S3 console interface. On the left, there's a navigation sidebar with sections like 'Amazon S3', 'Buckets', 'Access management and security', and 'Storage management and insights'. The main area displays the properties of an object named 'Image2.jpg' from a bucket. The 'Properties' tab is selected, showing details such as Owner (63af3c4c9fca9d8e4dadf100cbc51518983187f0e1e00e808c85d2036831aee7), AWS Region (US East (N. Virginia) us-east-1), Last modified (December 9, 2025, 15:59:15 (UTC+02:00)), Size (375.4 KB), and Type (jpg). To the right, there are links for 'S3 URI', 'Amazon Resource Name (ARN)', 'Entity tag (Etag)', and 'Object URL'. At the bottom of the main content area, there are buttons for 'Copy S3 URI', 'Download', 'Open in new tab', and 'Object actions'. The footer contains links for 'CloudShell', 'Feedback', 'Console Mobile App', and copyright information.

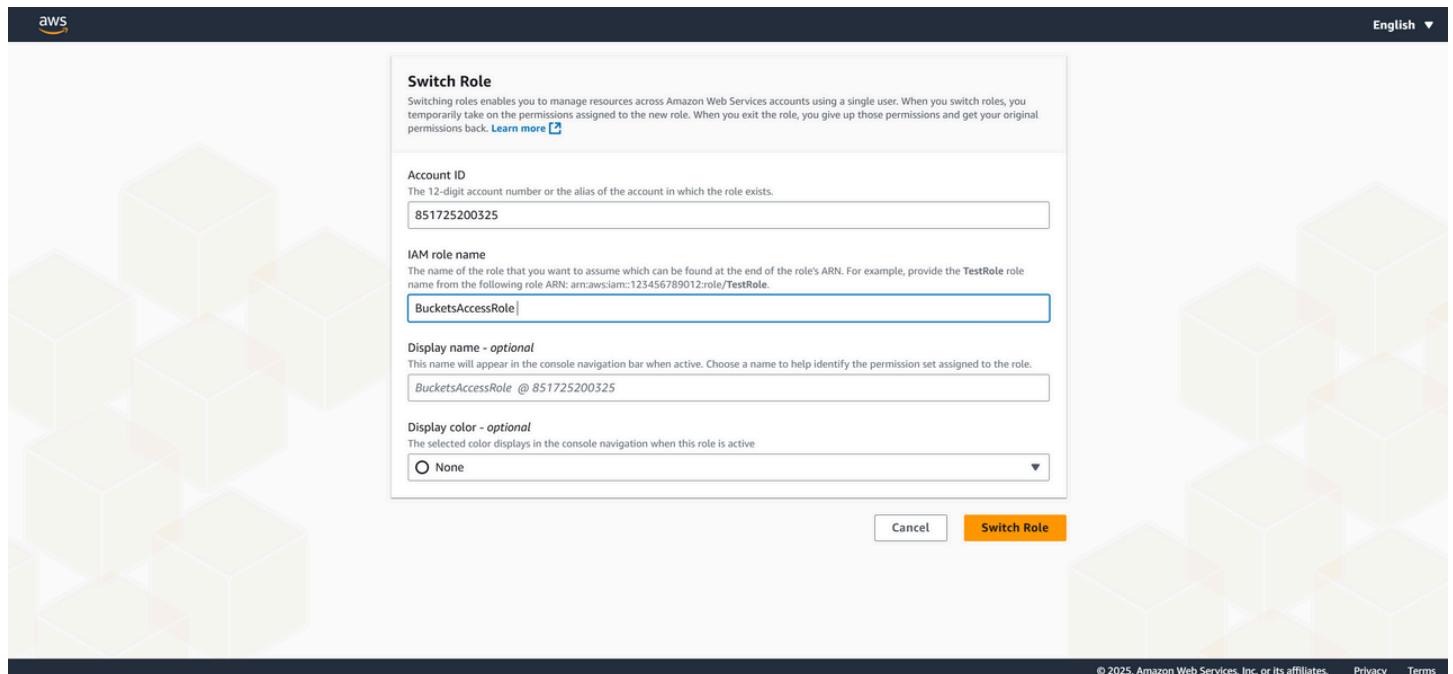
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<Error>
<Code>AccessDenied</Code>
<Message>User: arn:aws:iam::851725200325:user/devuser is not authorized to perform: s3:GetObject on resource: "arn:aws:s3:::c185581a4806215l13043277t1w891376957461-bucket1-krux8c1raim2/Image2.jpg" because no identity-based policy allows the s3:GetObject action</Message>
<RequestId>8E267A813R3A2PKB</RequestId>
<HostId>dFN5Lo9fTDIOxbPYxhP0//zbdNExw7096clowY4M6ZsHSBZu4jkZwwRd2iHxMSYxEwDaAxryU=</HostId>
</Error>
```

## Raison :

La politique basée sur l'identité de *devuser* (*DeveloperGroupPolicy*) autorise *s3>ListAllMyBuckets* et *s3>ListBucket*, mais ne contient pas la permission *s3GetObject*, 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)

Name	Type	Last modified	Size	Storage class
Image1.jpg	jpg	December 9, 2025, 15:59:13 (UTC+02:00)	1.1 MB	Standard
Image2.jpg	jpg	December 9, 2025, 15:59:15 (UTC+02:00)	375.4 KB	Standard

The screenshot shows the AWS IAM Roles page. The left sidebar includes sections for Identity and Access Management (IAM), Access management, and Access reports. The main content area displays a table for Roles (0). A red box highlights an 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." Below this, a detailed error message is shown: "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 account details: Account ID 8913-7695-7461, Account color Access denied, and Signed in as devuser.

The screenshot shows the AWS IAM Roles page for the 'BucketsAccessRole'. The left sidebar is identical to the previous screenshot. The main content area shows a table of attached policies: GetBucketPolicy, GrantBucket1Access, and ListAllBucketsPolicy. The 'ListAllBucketsPolicy' is expanded to show its JSON content:

```

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

```

- Bucket 1 :** Le téléchargement des objets a fonctionné sans problème.
- 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 success message: "Upload succeeded. For more information, see the Files and folders table." Below it, a status bar says "Upload: status". A note at the bottom says "After you navigate away from this page, the following information is no longer available." The main area is titled "Summary" and shows two sections: "Succeeded" (1 file, 375.4 KB (100.00%)) and "Failed" (0 files, 0 B (0%)). Below this, there are tabs for "Files and folders" (selected) and "Configuration". Under "Files and folders", it says "(1 total, 375.4 KB)" and lists "Image2.jpg" with details: Name, Folder, Type (image/jpeg), Size (375.4 KB), Status (Succeeded), and Error. The AWS footer at the bottom includes links for CloudShell, Feedback, and various legal notices.

## 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.

The screenshot shows the AWS S3 Bucket Policy configuration page. The left sidebar includes sections for Buckets, Access management and security, Storage management and insights, and Account and organization settings. The main content area displays a JSON-based bucket policy:

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "S3Write",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::449039445877:role/BucketsAccessRole"
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::c185581a4806215l13043277t1w449039445877-bucket2-ylsfa5cpervhb/*"
        },
        {
            "Sid": "ListBucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::449039445877:role/BucketsAccessRole"
            }
        }
    ]
}
```

### 3. Challenge Task: : Téléversement dans Bucket 3

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

Tentative 1 : En tant que *devuser*

The screenshot shows the AWS S3 Transfer Manager upload status page. A red banner at the top indicates an 'Upload failed' error, with a link to 'Diagnose with Amazon Q'. Below this, the 'Upload: status' section shows the destination as 's3://c185581a4806215l13043277t1w449039445877-bucket3-cayOam8i935d'. The summary table shows one succeeded file (0 files, 0 B) and one failed file (1 file, 375.4 KB). The 'Files and folders' tab is selected, showing a table with one entry: 'Image2 (1).jpg' (image/jpeg, 375.4 KB, Failed, Access denied).

Name	Folder	Type	Size	Status	Error
Image2 (1).jpg	-	image/jpeg	375.4 KB	Failed	Access denied

aws Search [Alt+S] United States (N. Virginia) Account ID: 4490-3944-5877 devuser

Amazon S3 > Buckets > c185581a4806215l13043277t1w449039445877-bucket3-cay0am8i935d

**Access denied**

API response

User: arn:aws:iam::449039445877:user/devuser is not authorized to perform: s3:GetBucketPublicAccessBlock on resource: "arn:aws:s3:::c185581a4806215l13043277t1w449039445877-bucket3-cay0am8i935d" because no identity-based policy allows the s3:GetBucketPublicAccessBlock action

**Bucket policy**

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

**You don't have permission to get bucket policy**

You or your AWS administrator must update your IAM permissions to allow s3:GetBucketPolicy. After you obtain the necessary permission, refresh the page. Learn more about [Identity and access management in Amazon S3](#)

API response

**Object Ownership**

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**You don't have permission to view Object ownership (bucket settings) configuration**

You need s3:GetBucketOwnershipControls to view Object ownership (bucket settings) configuration. Learn more about [Object ownership in Amazon S3](#)

API response

https://us-east-1.console.aws.amazon.com/s3/get-started?region=us-east-1 © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Tentative 2 : En utilisant le rôle *BucketAccessRole*

aws Search [Alt+S] United States (N. Virginia) Account ID: 4490-3944-5877 BucketsAccessRole @ 449039445877

Amazon S3 > Buckets > c185581a4806215l13043277t1w449039445877-bucket3-cay0am8i935d

**Objects**

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

**Insufficient permissions to list objects**

After you or your AWS administrator has updated your permissions to allow the s3>ListBucket action, refresh the page. Learn more about [Identity and access management in Amazon S3](#)

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

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

```

{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "S3Write",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::449039445877:role/OtherBucketAccessRole"
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::c185581a4806215113043277t1w449039445877-bucket3-cay0am8i935d/*"
        },
        {
            "Sid": "ListBucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::449039445877:role/OtherBucketAccessRole"
            }
        }
    ]
}

```

**Object Ownership**

## 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*.

**Switch Role**

Switching roles enables you to manage resources across Amazon Web Services accounts using a single user. When you switch roles, you temporarily take on the permissions assigned to the new role. When you exit the role, you give up those permissions and get your original permissions back. [Learn more](#)

**Account ID**  
The 12-digit account number or the alias of the account in which the role exists.

**IAM role name**  
The name of the role that you want to assume which can be found at the end of the role's ARN. For example, provide the `TestRole` role name from the following role ARN: `arn:aws:iam::123456789012:role/TestRole`.

**Display name - optional**  
This name will appear in the console navigation bar when active. Choose a name to help identify the permission set assigned to the role.

**Display color - optional**  
The selected color displays in the console navigation when this role is active.  
 None

**Cancel** **Switch Role**

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.

The screenshot shows the AWS S3 console interface. At the top, there's a green success message: "Upload succeeded. For more information, see the Files and folders table." Below this, a header bar includes the AWS logo, a search bar, and navigation links. The main content area has a title "Upload: status". A note says "After you navigate away from this page, the following information is no longer available." Under "Summary", there are two sections: "Succeeded" (1 file, 375.4 KB (100.00%)) and "Failed" (0 files, 0 B (0%)). Below this, there are tabs for "Files and folders" and "Configuration", with "Files and folders" selected. A table lists one file: "Image2.jpg" (image/jpeg, 375.4 KB, Status: Succeeded). The bottom of the screen shows standard AWS footer links like CloudShell, Feedback, and Copyright information.

[▶ Start Lab](#)[■ End Lab](#)[ℹ AWS Details](#)[ℹ Details](#)[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 priviléges 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.