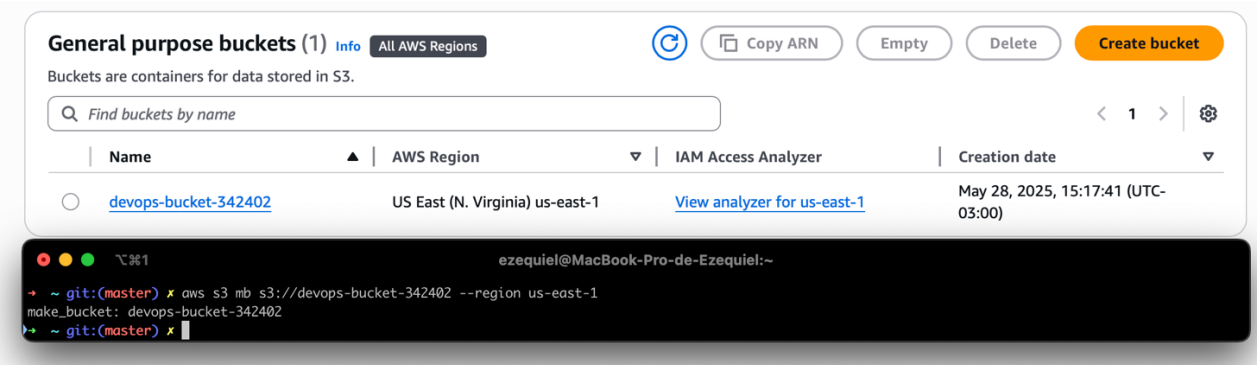
Desafio 3 AWS Uso de roles

Primer paso: Crear bucket s3



Segundo paso: Crear permisos IAM de escritura

Creo en mi computadora un archivo…trust-policy.json y luego creo el rol S3WriteRole

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Principal": {
    "Service": "s3.amazonaws.com"
   },
   "Action": "sts:AssumeRole"
```

```
    }
  ]
}
```

aws iam create-role --role-name S3WriteRole --assume-role-policy-document file://trust-policy.json

Salida CLI

Tercer paso: Asocio una política para Permitir Escritura en S3.

Creo una política que permita la acción PutObject en el bucket. Luego creo el json s3-write-policy.json:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "s3:PutObject"
   ],
   "Resource": "arn:aws:s3::: devops-bucket-342402/*"
  }
 ]
}
```

CLI para crear y asociar la politica

aws iam put-role-policy --role-name S3WriteRole --policy-name S3WritePolicy --policy-document file://s3-write-policy.json

Cuarto paso: Crear un Usuario IAM (s3-support) con Acceso Programático

Creo el usuario s3-support y genero credenciales de acceso programático.

CLI para creacion de usuario y generacion de clave de acceso

aws iam create-user --user-name s3-support

aws iam create-access-key --user-name s3-support

Salida CLI

Quinto paso: Actualizo la política de confianza del rol para permitir a s3-support asumirlo

Actualizo la política de confianza para permitir que el usuario s3-support asuma el rol S3WriteRole. Guardo el siguiente JSON como updated-trust-policy.json:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::ID_CUENTA:user/s3-support"
```

```
    },
    "Action": "sts:AssumeRole"
  }
 ]
}
```

CLI para actualizar la politica de confianza

aws iam update-assume-role-policy --role-name S3WriteRole --policy-document file://updated-trust-policy.json

Sexto paso: Configuro AWS CLI con las credenciales de s3-support

CLI

aws configure --profile s3-support

Entradas solicitadas:

AWS Access Key ID [None]: AKIAXXXXXXXXXXXXXXXX

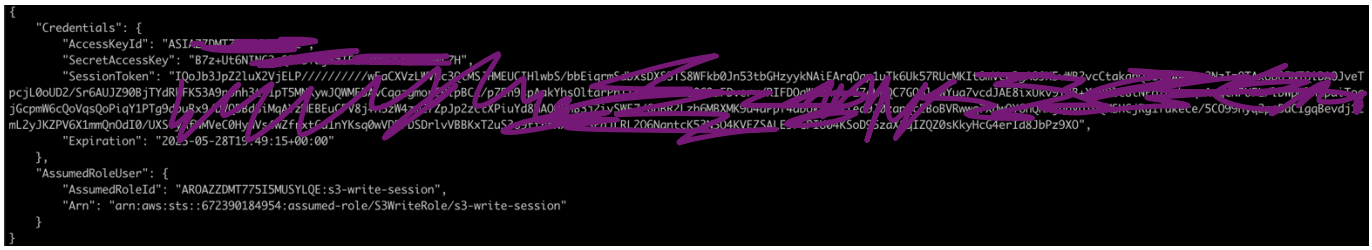
AWS Secret Access Key [None]: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Default region name [None]: us-east-1

Default output format [None]: json

Septimos paso: Asumir el rol y escribir en el bucket S3

aws sts assume-role --role-arn arn:aws:iam::ID_CUENTA:role/S3WriteRole --role-session-name s3-write-session --profile s3-support



Configuramo AWS CLI con las credenciales temporales

export AWS_ACCESS_KEY_ID=ASIAXXXXXXXXXXXXXXX

export AWS_SECRET_ACCESS_KEY=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

export AWS_SESSION_TOKEN=TOKEN_TEMPORAL

Finalmente pruebo subir un archivo test al bucket s3

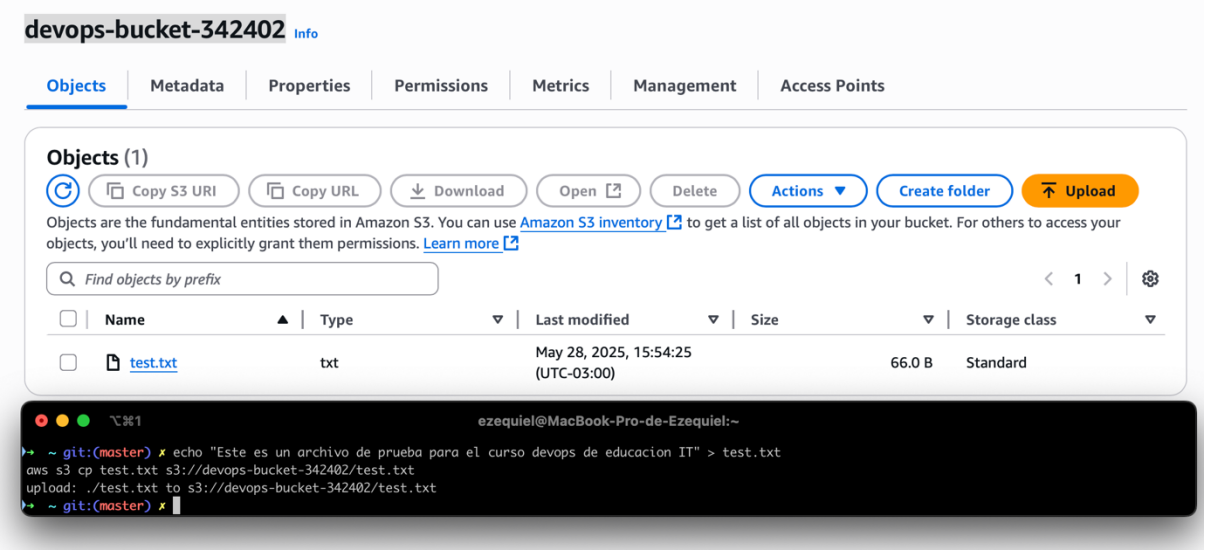


Diagrama: Interacción de Objetos IAM

Descripción:

El diagrama muestra la relación entre los objetos IAM y el bucket de S3:

Usuario IAM (s3-support): Configurado con credenciales programáticas (AccessKeyId y SecretAccessKey).

Rol IAM (S3WriteRole): Tiene una política de confianza que permite al usuario s3-support asumir el rol mediante sts:AssumeRole. La política asociada permite s3:PutObject en el bucket.

Servicio STS: Procesa la solicitud assume-role y emite credenciales temporales (incluyendo un SessionToken).

Bucket de S3 (devops-bucket-342402): Recibe la subida del archivo test.txt a través de AWS CLI con las credenciales temporales.

Flujo:

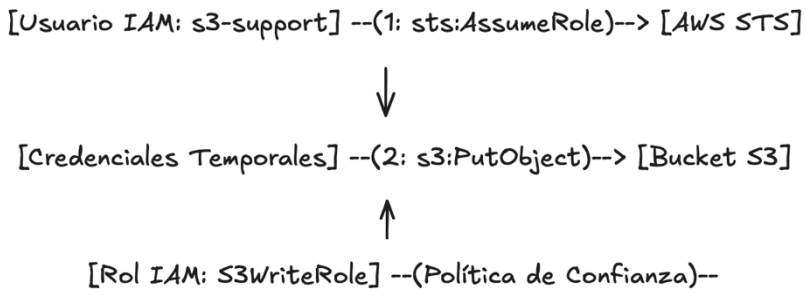
El usuario s3-support se autentica en la CLI.

El usuario ejecuta sts:AssumeRole para asumir el rol S3WriteRole.

AWS STS devuelve credenciales temporales.

La CLI usa estas credenciales para subir un archivo al bucket de S3.

Diagrama representativo:



-El usuario s3-support interactúa con AWS STS para asumir el rol.

-La política de confianza del rol autoriza al usuario a asumirlo.

-La política de permisos del rol permite escribir en el bucket de S3.

-Las credenciales temporales habilitan un acceso seguro y limitado al bucket.