

# PRÁCTICA 6: GESTIÓN DE SITIOS Y ARCHIVOS ESTÁTICOS EN AMAZON S3

---

## Objetivos

Al finalizar esta práctica, serás capaz de:

- Crear y configurar un bucket en Amazon S3 mediante CLI o consola web.
  - Subir y administrar archivos estáticos en S3.
  - Activar *Static Website Hosting* y acceder desde Internet.
  - Configurar permisos públicos y privados.
  - Habilitar el versionado de objetos y restaurar archivos antiguos.
  - Generar enlaces temporales con control de acceso.
  - Diferenciar los distintos niveles de administración en la nube (IaaS vs PaaS).
- 

## Requisitos previos

- Cuenta activa en **AWS Academy Learner Lab** (Free Tier).
- Máquina virtual **Ubuntu 22.04 o superior** con permisos root.
- Conexión a Internet **activa en la MV**.
- Editor de texto o terminal (nano, vim).
- Opcional: navegador **Firefox** dentro de la MV (ya instalado por defecto).

## PARTE 0: Preparación

```
sudo apt update  
sudo apt install awscli -y  
aws --version  
aws configure
```

He tenido que usar estos comandos ya que no me dejaba, asi que me meti a la web, descargué el archivo zip y lo descomprimi y lo instalé.

### COMANDOS:

```
sudo apt update  
sudo apt install unzip -y
```

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install
```

```
root@mail:/home/vboxuser# aws --version  
aws-cli/2.32.15 Python/3.13.11 Linux/6.14.0-36-generic exe/x86_64.ubuntu.24  
root@mail:/home/vboxuser# aws configure  
AWS Access Key ID [None]:
```

Introduce tus credenciales y región eu-west-1.

```
root@mail:/home/vboxuser# aws configure  
AWS Access Key ID [None]: ASIATCJKFBRIUQX7ZIG5  
AWS Secret Access Key [None]: oIC60d2dLtud15oFTUzjiuJD350jjY0oBwUdSST8  
AWS Session Token [None]: IQoJb3JpZ2luX2VjEDkaCXVzLXdIc3QtMiJIMEYCIQCIig2x0u3xi/e9n+d4DBhBB  
<BoLJ8LEhyRbrJxetC0ffFwqPfI6QGpnfQR9mgy83j0KVuzX18d8XqM2j07rPngaplxXwP35mTQ==  
Default region name [None]: us-east-1  
Default output format [None]: json  
root@mail:/home/vboxuser#
```

Crea un archivo de prueba:

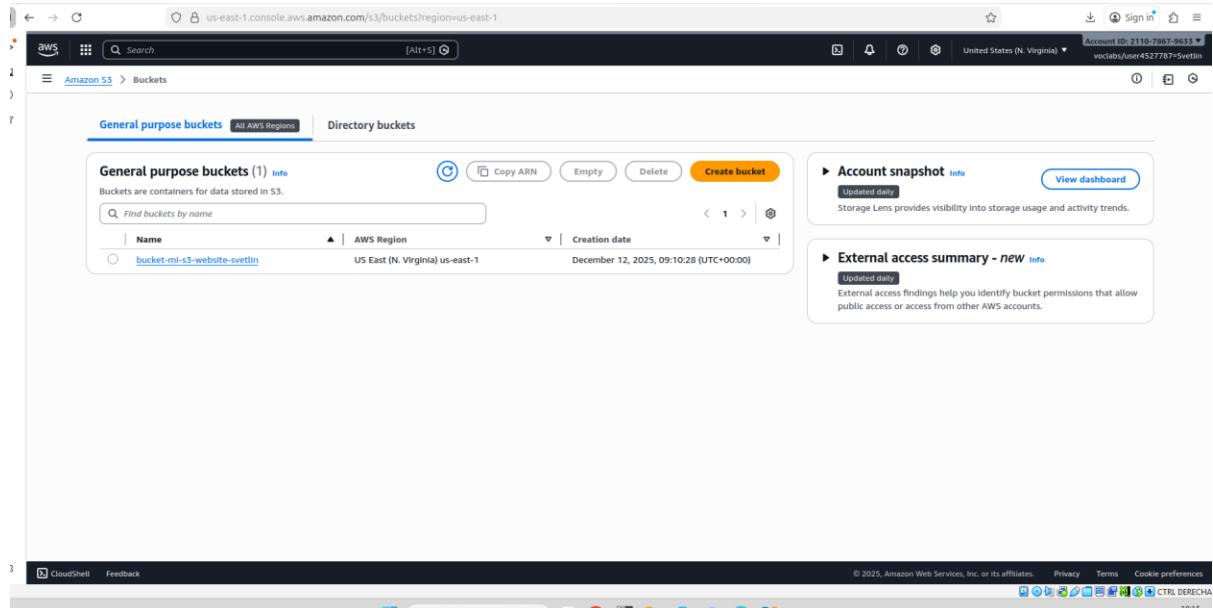
```
mkdir ~/s3-website  
echo "<h1>Bienvenido a mi web en Amazon S3</h1>" > ~/s3-website/index.html  
root@mail:/home/vboxuser# mkdir ~/s3-website  
root@mail:/home/vboxuser# echo "<h1>Bienvenido a mi web en Amazon S3</h1>" > ~/s3-website/i  
ndex.html  
root@mail:/home/vboxuser#
```

## PARTE 1: Crear el bucket

```
aws s3api create-bucket \
```

```
--bucket mi-s3-website-[tu-nombre] \
--region eu-west-1 \
--create-bucket-configuration LocationConstraint=eu-west-1
```

Como no tenemos permisos, no podemos crear el bucket desde la terminal de la MV por eso lo hemos creado desde el propio aws academy



Sustituye [tu-nombre] por tu nombre real.

Verifica:

```
aws s3 ls
```

```
root@mail:/home/vboxuser# aws s3 ls
2025-12-12 09:10:28 bucket-mi-s3-website-svetlin
root@mail:/home/vboxuser#
```

## PARTE 2: Subir archivos

```
aws s3 cp ~/s3-website/index.html s3://mi-s3-website-[tu-nombre] --acl public-read
aws s3 ls s3://mi-s3-website-[tu-nombre]
```

Con el comando descrito no funciona, hay que usar este en su lugar

```
root@mail:/home/vboxuser# aws s3 cp ~/s3-website/index.html s3://bucket-mi-s3-website-svetlin  
upload: ../../root/s3-website/index.html to s3://bucket-mi-s3-website-svetlin/index.html  
root@mail:/home/vboxuser#
```

## PARTE 3: Activar Static Website Hosting

Crea website.json:

```
cat > website.json <<EOF  
{  
    "IndexDocument": { "Suffix": "index.html" },  
    "ErrorDocument": { "Key": "error.html" }  
}  
EOF
```

```
root@mail:/home/vboxuser# cat > website.json <<EOF  
{  
    "IndexDocument": { "Suffix": "index.html" },  
    "ErrorDocument": { "Key": "error.html" }  
}  
EOF  
root@mail:/home/vboxuser#
```

Luego ejecuta el comando:

```
aws s3api put-bucket-website \  
--bucket mi-s3-website-[tu-nombre] \  
--website-configuration file://website.json
```

```

Unknown output type: JSON
root@mail:/home/vboxuser# aws s3api put-bucket-website \
--bucket bucket-mi-s3-website-svetlin \
--website-configuration file://website.json \
--output text
root@mail:/home/vboxuser# aws s3api get-bucket-website --bucket bucket-mi-s3-
website-svetlin

Unknown output type: JSON
root@mail:/home/vboxuser# aws s3api get-bucket-website --bucket bucket-mi-s3-
website-svetlin --output text
ERRORDOCUMENT    error.html
INDEXDOCUMENT   index.html
root@mail:/home/vboxuser#

```

La URL resultante seguirá el formato:

**http://mi-s3-website-[tu-nombre].s3-website-eu-west-1.amazonaws.com**

---

## PARTE 4: Política de acceso público

Crea policy.json:

```

cat > policy.json <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        { "Sid": "PublicRead",
          "Effect": "Allow",
          "Principal": "*",
          "Action": ["s3:GetObject"],
          "Resource": ["arn:aws:s3:::mi-s3-website-[tu-nombre]/*"]
        }
    ]
}
EOF

```

```
root@mail:/home/vboxuser#  
cat > policy.json <<EOF  
{  
"Version": "2012-10-17",  
"Statement": [  
{ "Sid": "PublicRead",  
"Effect": "Allow",  
"Principal": "*",  
"Action": ["s3:GetObject"],  
"Resource": ["arn:aws:s3:::mi-s3-website-[tu-nombre]/*"]  
}  
]  
}  
EOF  
root@mail:/home/vboxuser#
```

Luego ejecuta el comando:

La cuenta de academy no permite editar politicas en el bucket por lo que este paso a dado fallo;

```
root@mail:/home/vboxuser# aws s3api put-bucket-policy \  
> ^C  
root@mail:/home/vboxuser# aws s3api put-bucket-policy \  
--bucket bucket-mi-s3-website-svetlin \  
--policy file://policy.json  
  
An error occurred (MalformedPolicy) when calling the PutBucketPolicy operation: Policy has invalid resource  
root@mail:/home/vboxuser# █
```

aws s3api put-bucket-policy \  
--bucket mi-s3-website-[tu-nombre] \  
--policy file://policy.json

## PARTE 5: Versionado de archivos

Ejecuta:

```
aws s3api put-bucket-versioning \
--bucket mi-s3-website-[tu-nombre] \
--versioning-configuration Status=Enabled

root@mail:/home/vboxuser# aws s3api put-bucket-versioning \
- -bucket bucket-mi-s3-website-svetlin \
--versioning-configuration Status=Enabled \
--output text
root@mail:/home/vboxuser# █
```

Edita el archivo **index.html**, súbelo otra vez y lista versiones ejecutando el comando:

```
aws s3api list-object-versions --bucket mi-s3-website-[tu-nombre]
HAY QUE AÑADIRLE AL FINAL EL "--output text"
```

```
root@mail:/home/vboxuser# aws s3api list-object-versions --bucket bucket-mi-s
3-website-svetlin --output text
None    None        None
VERSIONS      FULL_OBJECT      "6bf9e1c453a1195244a836198c1d2a93"      True
          index.html      2025-12-12T09:23:01+00:00      42      STANDARD
          null
CHECKSUMALGORITHM      CRC64NVME
OWNER      e5ce4f403ac5b089515ba2cf816262f3dd86da03b1ccee8164cbef91acb6dde
root@mail:/home/vboxuser# █
```

## PARTE 6: Carpeta y backup

Ejecuta:

```
mkdir ~/s3-backups
echo "Backup de WordPress (ejemplo)" > ~/s3-backups/backup.txt
aws s3 cp ~/s3-backups/backup.txt s3://mi-s3-website-[tu-nombre]/backups/ --acl private
```

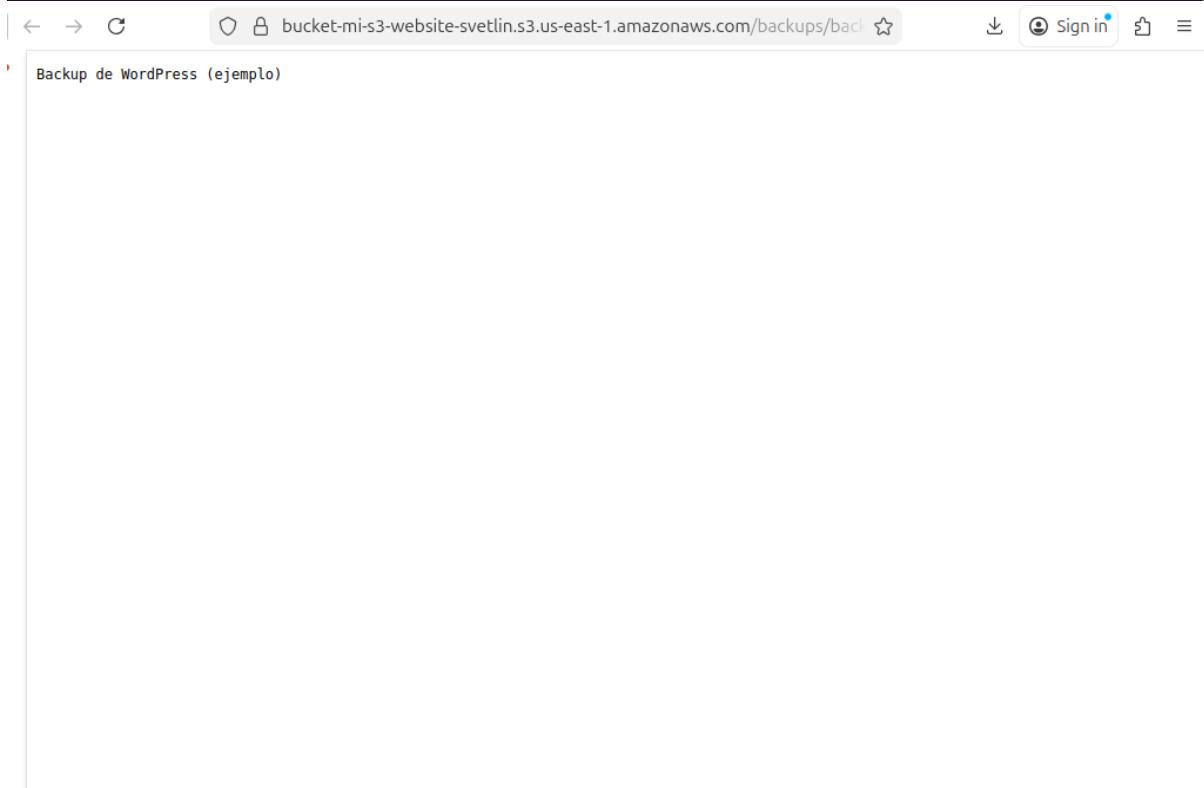
```
root@mail:/home/vboxuser# mkdir ~/s3-backups
root@mail:/home/vboxuser# echo "Backup de WordPress (ejemplo)" > ~/s3-backups
/backup.txt
root@mail:/home/vboxuser# aws s3 cp ~/s3-backups/backup.txt s3://bucket-mi-s3
-website-svetlin/backups/ --acl private
upload: ../../root/s3-backups/backup.txt to s3://bucket-mi-s3-website-svetlin
/backups/backup.txt
root@mail:/home/vboxuser#
```

## PARTE 7: Enlace temporal

```
aws s3 presign s3://mi-s3-website-[tu-nombre]/backups/backup.txt --expires-in 600
```

Abre la URL generada en el navegador para verificar.

```
root@mail:/home/vboxuser# aws s3 presign s3://bucket-mi-s3-website-svetlin/backups/backup.txt --expires-in 600
https://bucket-mi-s3-website-svetlin.s3.us-east-1.amazonaws.com/backups/backup.txt?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=ASIATCJKBRIQXAJETQ2%2F20251212%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20251212T095503Z&X-Amz-Expires=600&X-Amz-SignedHeaders=host&X-Amz-Security-Token=%27IQoJb3JpZ2luX2VjEDkaCXVzLXdIc3QtMiJIMEYCIQDDo5602ta9h%2B5cMC5KVTWGDx3HwFyrUb9%2Bawahy9ezbAIhAOK27lJ4M8Guul3FEza8SPBvnjMXA970pPh8pxzHqm%2FXKqECCAIQARoMMjExMDc4Njc5NjMzIgxXjkjTZk22yWoym3Mq%2FgGFVw8jehLwM936Dxs%2FWdbzxxJRaNf%2Bb80ctZK623FF7vZOAGJg0aTamqR0gzZrDmyrnTUCuD8qqDJ7gPn98xYfTV5wQ%2FlTHGZPqeXblMa1uKJ10jHtpYeF4siL8PwN9KThhGcWMDlq6q5VaxTwZVZk1kk3oaI%2BuWVM3vWQ6UKbMF7DjEM%2BN%2B%2Fh%2BnhtilDs1t9VT%2FvouIXvxlrW5q7XbdDb2tLMSBBJRLiCEdI8NneHPqcpexU0WtJIGliWgCSi9qIlji7haPT2bSz44aQh%2BS4vDcKf5GSBnCKLwzdxKMgReCyC22LmYqjKoQVS8%2B2IaRg%2BzKGWo%2FrkTVoa2kMDCqr%2B%2FJBjqcAUxzP40su69102pGiALTJsv0iLU8%2FZbIOmpebQGkFl1oGCR0w1nTaZDb0zYXsXN3FdxyH89F39%2Bfjpy9Dnzs31PVCPZgr56ncdXboicAXo1DW05GrojBEbYR%2FKxKXbcS4Fu0oFGcMBo2XPAJZe%2F8L8JbSij7R6N6E8KRPLh4qQ4txdsR9mjFVPoeJKMZwq40VWmFGJWwhwwuY7w%3D%3D%27&X-Amz-Signature=8e3c3295cff884d11310f2c2519a5083215e8095ffa3b3e41013c1124290a73d
root@mail:/home/vboxuser#
```



## PARTE 8 – Etiquetado y metadatos

```
aws s3api put-object-tagging \
--bucket mi-s3-website-[tu-nombre] \
--key index.html \
```

```
--tagging 'TagSet=[{Key=proyecto,Value=SMR2},{Key=profesor,Value=Miguel}]'
```

```
aws s3api get-object-tagging \
--bucket mi-s3-website-[tu-nombre] \
--key index.html
```

```
root@mail:/home/vboxuser# aws s3api put-object-tagging \
--bucket bucket-mi-s3-website-svetlin \
--key index.html \
--tagging 'TagSet=[{Key=proyecto,Value=SMR2},{Key=profesor,Value=Miguel}]'
aws s3api get-object-tagging \
--bucket bucket-mi-s3-website-svetlin \
--key index.html --output text

Unknown output type: JSON
null
TAGSET proyecto SMR2
TAGSET profesor Miguel
root@mail:/home/vboxuser#
```

Uso: gestión contable, filtrado y automatización.

## PARTE 9 – Clases de almacenamiento

Sube un archivo con clase alternativa:

```
aws s3 cp ~/s3-backups/backup.txt \
s3://mi-s3-website-[tu-nombre]/ \
--storage-class STANDARD_IA
```

```
root@mail:/home/vboxuser# aws s3 cp ~/s3-backups/backup.txt \
s3://bucket-mi-s3-website-svetlin/ \
--storage-class STANDARD_IA
upload: ../../root/s3-backups/backup.txt to s3://bucket-mi-s3-website-svetlin
/backup.txt
root@mail:/home/vboxuser#
```

Consulta la clase:

```
aws s3api list-objects-v2 \
```

```
--bucket mi-s3-website-[tu-nombre] \
--query "Contents[].[Key,StorageClass]"
```

```
root@mail:/home/vboxuser# aws s3api list-objects-v2 \
--bucket bucket-mi-s3-website-svetlin \
--query "Contents[].[Key,StorageClass]" --output text
backup.txt      STANDARD_IA
backups/backup.txt      STANDARD
index.html      STANDARD
root@mail:/home/vboxuser#
```

## PARTE 10 – Reglas de ciclo de vida

Crea lifecycle.json:

```
cat > lifecycle.json <<EOF
{
    "Rules": [
        {
            "ID": "MoveToIA",
            "Prefix": "",
            "Status": "Enabled",
            "Transitions": [
                {"Days": 30, "StorageClass": "STANDARD_IA"}
            ],
            "Expiration": {"Days": 365}
        }
    ]
}
EOF
```

```
root@mail:/home/vboxuser# cat > lifecycle.json <<EOF
{
  "Rules": [
    {
      "ID": "MoveToIA",
      "Prefix": "",
      "Status": "Enabled",
      "Transitions": [
        {"Days": 30, "StorageClass": "STANDARD_IA"}
      ],
      "Expiration": {"Days": 365}
    }
  ]
}
EOF
root@mail:/home/vboxuser#
```

Aplica y consulta:

```
aws s3api put-bucket-lifecycle-configuration \
--bucket mi-s3-website-[tu-nombre] \
--lifecycle-configuration file://lifecycle.json
```

```
aws s3api get-bucket-lifecycle-configuration \
--bucket mi-s3-website-[tu-nombre]
```

```
root@mail:/home/vboxuser# aws s3api put-bucket-lifecycle-configuration \
--bucket bucket-mi-s3-website-svetlin \
--lifecycle-configuration file://lifecycle.json
aws s3api get-bucket-lifecycle-configuration \
--bucket bucket-mi-s3-website-svetlin --output text

Unknown output type: JSON
all_storage_classes_128K
RULES      MoveToIA          Enabled
EXPIRATION    365
TRANSITIONS   30      STANDARD_IA
root@mail:/home/vboxuser#
```

## PARTE 11 – Script de despliegue automatizado

Genera deploy-s3.sh:

```
cat > deploy-s3.sh <<'EOF'  
  
#!/bin/bash  
  
set -e  
  
BUCKET="mi-s3-website-$(whoami)"  
  
aws s3api create-bucket --bucket $BUCKET --region eu-west-1 \  
    --create-bucket-configuration LocationConstraint=eu-west-1  
  
aws s3 cp index.html s3://$BUCKET --acl public-read  
  
aws s3api put-bucket-website \  
    --bucket $BUCKET \  
    --website-configuration '{"IndexDocument":{"Suffix":"index.html"}}'  
  
cat > policy.json <<POL  
  
{"Version":"2012-10-17", "Statement": [{"Sid":"PublicRead",  
    "Effect":"Allow", "Principal":"*",  
    "Action":["s3:GetObject"],  
    "Resource":["arn:aws:s3:::$BUCKET/*"]}]}  
  
POL  
  
aws s3api put-bucket-policy --bucket $BUCKET --policy file://policy.json  
  
echo "Implementación completada. Sitio disponible en:"  
  
echo "http://\$BUCKET.s3-website-eu-west-1.amazonaws.com"  
  
EOF
```

```

root@mail:/home/vboxuser# cat > deploy-s3.sh <<'EOF'
#!/bin/bash
set -e
BUCKET="mi-s3-website-$(whoami)"
aws s3api create-bucket --bucket $BUCKET --region eu-west-1 \
--create-bucket-configuration LocationConstraint=eu-west-1
aws s3 cp index.html s3://$BUCKET --acl public-read
aws s3api put-bucket-website \
--bucket $BUCKET \
--website-configuration '{"IndexDocument":{"Suffix":"index.html"}}'
cat > policy.json <<POL
{"Version":"2012-10-17","Statement":[{"Sid":"PublicRead",
"Effect":"Allow","Principal":"*",
>Action":["s3:GetObject"],
"Resource":["arn:aws:s3:::$BUCKET/*"]}]}
```

POL

- aws s3api put-bucket-policy --bucket \$BUCKET --policy file://policy.json
- echo "Implementación completada. Sitio disponible en:"
- echo "http://\$BUCKET.s3-website-eu-west-1.amazonaws.com"

EOF

```

root@mail:/home/vboxuser#

```

Luego ejecuta los siguientes comandos para dar permisos y ejecutar el script.

**chmod +x deploy-s3.sh**

**./deploy-s3.sh**

Como no tenemos permisos no me deja crear el bucket

```

root@mail:/home/vboxuser# chmod +x deploy-s3.sh
root@mail:/home/vboxuser# ./deploy-s3.sh

An error occurred (AccessDenied) when calling the CreateBucket operation: Use
r: arn:aws:sts::211078679633:assumed-role/voclabs/user4527787=Svetlin is not
authorized to perform: s3:CreateBucket on resource: "arn:aws:s3:::mi-s3-websi
te-root" because no identity-based policy allows the s3:CreateBucket action
root@mail:/home/vboxuser#
```

## PARTE 12 – Monitorización y métricas (visualización básica)

Ejecuta:

**aws cloudwatch list-metrics --namespace AWS/S3 | head -15**

```
root@mail:/home/vboxuser# aws cloudwatch list-metrics --namespace AWS/S3 --output table | head -15
-----
|ListMetrics|
+-----+
root@mail:/home/vboxuser#
```

Comenta que tipos de métricas (almacenamiento, peticiones, latencia) devuelve.

## ENTREGABLES

1. Captura de instalación y aws configure.
2. Creación y listado del bucket.
3. Subida y url de index.html.
4. Política JSON aplicada.
5. Versionado con list-object-versions.
6. Etiquetado y listar tags.
7. Ejemplo de Storage Class.
8. Regla lifecycle configurada.
9. Ejecución del script deploy-s3.sh.
10. Evidencia de URL temporal.

## Notas importantes

- **Seguridad:** no subir archivos con datos personales.
- **Coste:** las operaciones S3 empleadas son Free Tier.
- **Nombres únicos:** el nombre del bucket es global.
- **Cierre:** elimina todo al finalizar.
- **Comparativa:**
  - EC2 → control total del servidor.
  - S3 → servicio gestionado, ideal para web estática y backups.

NO TENEMOS PERMISOS EN LA  
CUENTA DE ACADEMY POR LO QUE A  
LO LARGO  
DE LA PRACTICA HEMOS TENIDO  
MUCHOS ERRORES, PERO LOS  
HEMOS SOLUCIONADO