

Lab CI/CD: deploy automático a S3

Creá un bucket con Terraform, configurá GitHub Actions, y tu sitio se despliega con cada push.

PARTE 1 — CREAR EL BUCKET CON TERRAFORM

1 Entrá a la carpeta del ejercicio

```
$ cd camino-devops-en-aws/sesion4/sesion4-lab-home-terraform/03-cicd-s3
```

2 Creá el bucket

```
$ terraform init
Terraform has been successfully initialized!

$ terraform apply -var="nombre=tunombre"
Plan: 4 to add, 0 to change, 0 to destroy.
Do you want to perform these actions? (yes)

Apply complete! Resources: 4 added.

Outputs:
bucket_name = "sesion4-tunombre-frontend"
website_url = "http://sesion4-tunombre-frontend.s3-website-us-east-1.amazonaws.com"
```

Anotá el `bucket_name` y la `website_url` — los vas a necesitar.



Tu bucket S3 con hosting web ya existe en AWS

PARTE 2 — CONFIGURAR SECRETS EN GITHUB

3 Abrí tu repo en GitHub

Andá a tu repo en GitHub y seguí este camino:

4 Agregá los 3 secrets

Creá cada uno con **New repository secret**:

```
# Secret 1
Name:  AWS_ACCESS_KEY_ID
Value: tu access key de AWS

# Secret 2
Name:  AWS_SECRET_ACCESS_KEY
Value: tu secret key de AWS

# Secret 3
Name:  S3_BUCKET_NAME
Value: sesion4-tunombre-frontend
```

Los Access Keys son los mismos que usaste en `aws configure` (sesión 2).

LAS CREDENCIALES QUEDAN ENCRYPTADAS

GitHub las guarda de forma segura. Nadie las puede ver después de guardarlas, ni siquiera vos. Solo las usa el pipeline.

PARTE 3 — CREAR EL WORKFLOW

5 Creá la carpeta y el archivo

```
# Desde la raíz de tu repo
$ mkdir -p .github/workflows
$ cp sesion4/sesion4-lab-home-terraform/03-cicd-s3/deploy-frontend.yml .github/workflows/
```

GitHub detecta automáticamente cualquier `.yaml` dentro de `.github/workflows/`.

6 Copiá el frontend a la raíz del repo

```
# Crear carpeta frontend en la raíz
$ mkdir -p frontend
$ cp sesion4/sesion4-lab-home-terraform/03-cicd-s3/frontend/index.html frontend/
```

El workflow busca archivos en `./frontend/` para subirlos a S3.

7

Comiteá y pushé

```
$ git add .  
$ git commit -m "agregar pipeline CI/CD"  
$ git push
```

8

Mirá el pipeline corriendo

Andá a tu repo en GitHub → pestaña **Actions**



Vas a ver "Deploy Frontend to S3" corriendo

Cuando termine, aparece un  verde

9

Visitá tu sitio web

```
# Abrió en el navegador:  
http://sesion4-tunombre-frontend.s3-website-us-east-1.amazonaws.com
```



Tu sitio está en vivo!

Desplegado automáticamente con GitHub Actions a S3. Cada push lo actualiza.

PARTE 5 — PROBÁ EL CICLO COMPLETO

10

Modificá el frontend y pushé

Abrió `frontend/index.html`, cambiá algo (el título, un color, un texto) y:

```
$ git add frontend/index.html  
$ git commit -m "cambiar titulo del frontend"
```

```
$ git push
```

Esperá 30 segundos, recargá la página, y vas a ver tu cambio en vivo.



Eso es CI/CD: cambiás código, pushés, y el cambio está en producción

LIMPIEZA

11 Destruir el bucket

```
# Primero vaciar el bucket
$ aws s3 rm s3://sesion4-tunombre-frontend --recursive

# Luego destruir con Terraform
$ cd sesion4/sesion4-lab-home-terraform/03-cicd-s3
$ terraform destroy -var="nombre=tunombre"
Destroy complete! Resources: 4 destroyed.
```

Hay que vaciar el bucket antes de destruirlo. S3 no deja borrar buckets con archivos adentro.

NO TE OLVIDES DE LIMPIAR

Si dejás el bucket, no te va a costar plata (S3 es muy barato), pero es buena práctica limpiar lo que ya no usás.



Bonus completado

Terraform creó la infraestructura, GitHub Actions desplegó el código. El camino DevOps completo.