

Tarea 5: Validación automática con GitHub Actions para Terraform

En primer lugar, vamos a añadir la configuración de Terraform con módulos utilizada en el ejercicio anterior a un repositorio que crearemos en nuestro GitHub.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  AZURE

root@daniel-MS-7E28:/home/terraform-localstack# aws --endpoint-url=http://localhost:4566 s3 ls
2025-07-02 07:51:20 bucket-desde-modulo
root@daniel-MS-7E28:/home/terraform-localstack# git remote add origin git@github.com:Danimartinez-1997/Terraform.git
fatal: no es un repositorio git (ni ninguno de los directorios superiores): .git
root@daniel-MS-7E28:/home/terraform-localstack# git init
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama predeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial para usar en todos
ayuda: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
ayuda:
ayuda: git config --global init.defaultBranch <nombre>
ayuda:
ayuda: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk' y
ayuda: 'development'. Se puede cambiar el nombre de la rama recién creada mediante este comando:
ayuda:
ayuda: git branch -m <nombre>
Iniciado repositorio Git vacío en /home/terraform-localstack/.git/
root@daniel-MS-7E28:/home/terraform-localstack# git add README.md
fatal: ruta especificada 'README.md' no concordó con ningún archivo
root@daniel-MS-7E28:/home/terraform-localstack# git commit -m "first commit"
En la rama master

Confirmación inicial

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que será confirmado)
.terraform.lock.hcl
.terraform/
main.tf
modules/
terraform.tfstate
terraform.tfstate.backup

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
root@daniel-MS-7E28:/home/terraform-localstack# git add *
root@daniel-MS-7E28:/home/terraform-localstack# git commit -m "first commit"
[master (commit-raiz) 0903fcf] first commit
6 files changed, 213 insertions(+)
create mode 100644 main.tf
create mode 100644 modules/mi-recurso/main.tf
create mode 100644 modules/mi-recurso/outputs.tf
create mode 100644 modules/mi-recurso/variables.tf
create mode 100644 terraform.tfstate
create mode 100644 terraform.tfstate.backup
root@daniel-MS-7E28:/home/terraform-localstack# git branch -M main
root@daniel-MS-7E28:/home/terraform-localstack# git remote add origin git@github.com:Danimartinez-1997/Terraform.git
root@daniel-MS-7E28:/home/terraform-localstack# git push -u origin main
Enter passphrase for key '/root/.ssh/id_ed25519':
Enumerando objetos: 10, listo.
Contando objetos: 100% (10/10), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (9/9), listo.
Escribiendo objetos: 100% (10/10), 2.14 KiB | 2.14 MiB/s, listo.
Total 10 (delta 1), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:Danimartinez-1997/Terraform.git
 * [new branch]      main -> main
rama 'main' configurada para rastrear 'origin/main'.
root@daniel-MS-7E28:/home/terraform-localstack#
```

Añadimos un workflow de GitHub Actions que ejecute terraform validate tras cada push.


```
! terraform-validate.yml x
terraform-localstack > workflows > ! terraform-validate.yml > name
1  name: Terraform Validate
2
3  on:
4    push:
5      branches:
6        - main
7    pull_request:
8      branches:
9        - main
10
11 jobs:
12   validate:
13     runs-on: ubuntu-latest
14
15     steps:
16     - name: Checkout repo
17       uses: actions/checkout@v3
18
19     - name: Setup Terraform
20       uses: hashicorp/setup-terraform@v2
21       with:
22         terraform_version: 1.5.6 # Cambia a la versión que uses
23
24     - name: Terraform Init
25       run: terraform init
26
27     - name: Terraform Validate
28       run: terraform validate
29
```

Verificamos que el workflow funciona correctamente.

The screenshot shows the GitHub Actions interface for a repository named 'Terraform'. The 'Actions' tab is selected, displaying a list of workflows. The 'Terraform Validate' workflow is highlighted. Below the workflow list, a table shows the execution history. A single run, labeled 'validate2', is shown with a green checkmark indicating success. The run was triggered by a push to the 'main' branch by user 'Danimartinez-1997'. The interface also includes a search bar for workflow runs and a sidebar with navigation links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.


Event	Status	Branch	Actor
validate2	Success	main	Danimartinez-1997







Por último, añadimos un badge de estado al Readme.md



 **Terraform** Public

Pin Watch 0


main 1 Branch 0 Tags Add file Code

 **Danimartinez-1997** Update README.md e712f51 · now 6 Commits

 .github/workflows	validate2	5 days ago
 modules/mi-recurso	first commit	5 days ago
 README.md	Update README.md	now
 main.tf	first commit	5 days ago
 terraform.tfstate	first commit	5 days ago
 terraform.tfstate.backup	first commit	5 days ago

 **README** 

Terraform + LocalStack

 Terraform Validate passing

Tarea 6: CI/CD básico con Terraform y LocalStack

Instrucciones

1. Subir el proyecto de Terraform a GitHub

Crea un repositorio y sube todos los archivos de tu configuración de Terraform.

Importante: Añade un archivo `.gitignore` para evitar subir archivos pesados o temporales, como la carpeta `.terraform/`, archivos `.tfstate` y otros archivos sensibles.

2. Configurar un workflow de GitHub Actions

En el repositorio, crea un archivo en `.github/workflows/deploy.yml` con la siguiente configuración básica:

- El workflow se debe ejecutar en cada push a la rama `main`.
- Debe levantar el servicio LocalStack en un contenedor dentro del runner.
- Debe ejecutar, en orden, los comandos:
 - `terraform init`
 - `terraform plan`
 - `terraform apply` (con la opción `-auto-approve` para evitar confirmaciones manuales).
- El workflow debe usar la imagen oficial de LocalStack y exponer los puertos necesarios para simular los servicios de AWS.

3. Verificar el despliegue automático

Cada vez que hagas un push a la rama `main`, el workflow debe ejecutarse automáticamente y desplegar la infraestructura localmente en LocalStack, sin errores.

Workflow file for this run

.github/workflows/deploy.yml at d76e8b3

```
1  name: Terraform Validate
2
3  on:
4    push:
5      branches:
6        - main
7    pull_request:
8      branches:
9        - main
10
11 jobs:
12   validate:
13     runs-on: ubuntu-latest
14
15     steps:
16     - name: Checkout repo
17       uses: actions/checkout@v3
18
19     - name: Setup Terraform
20       uses: hashicorp/setup-terraform@v2
21       with:
22         terraform_version: 1.5.6 # Cambia a la versión que uses
23
24     - name: Terraform Init
25       run: terraform init
26
27     - name: Terraform Validate
28       run: terraform validate
```

Tarea 7: Limpieza de commits en Git (Repositorio Terraform)

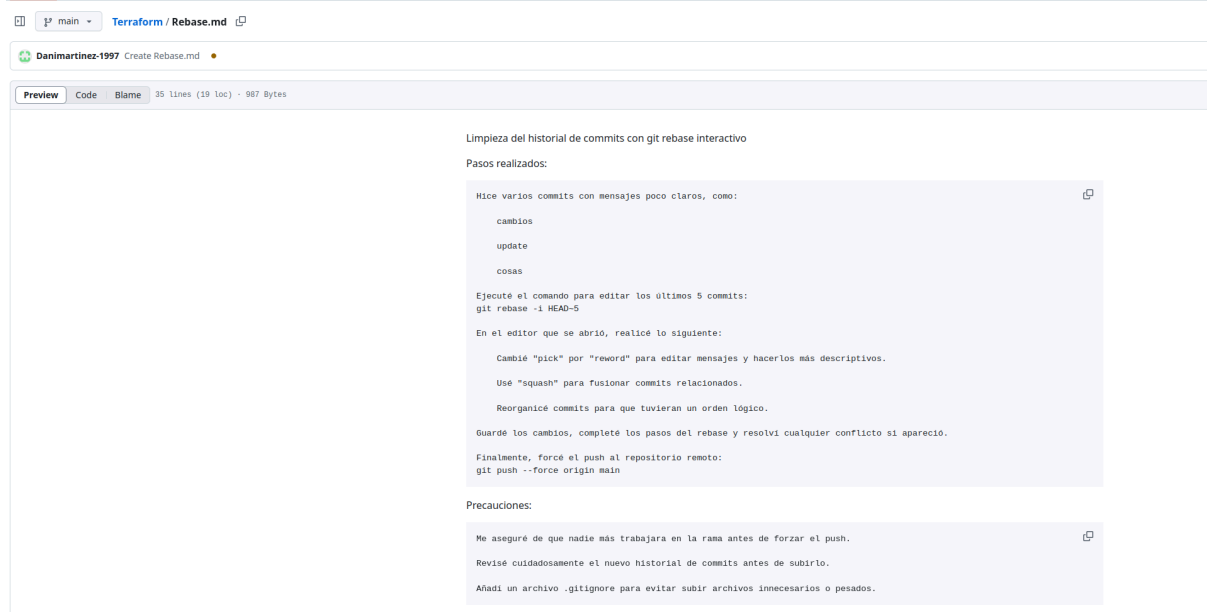
Ejecutamos el comando `git rebase -i HEAD~5` para editar los últimos 5 commits que se hayan subido.

```
GNU nano 7.2
reword e712f51 Update README.md
reword cecfac6 Commit después del desbloqueo forzado
reword 5d553da añade .gitignore
pick 5e68e07 Eliminar carpeta .terraform del repositorio
pick c344e0d Eliminar carpeta .terraform/ y añadir .gitignore

# Rebase 74612e7..c344e0d en 74612e7 (5 comandos)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                               commit's log message, unless -C is used, in which case
#                               keep only this commit's message; -c is same as -C but
#                               opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
#       create a merge commit using the original merge commit's
#       message (or the oneline, if no original merge commit was
#       specified); use -c <commit> to reword the commit message
# u, update-ref <ref> = track a placeholder for the <ref> to be updated
#                       to this position in the new commits. The <ref> is
#                       updated at the end of the rebase
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# Si eliminas una línea aquí EL COMMIT SE PERDERÁ.
#
# Como sea, si quieres borrar todo, el rebase será abortado.
#
```

Y hacemos un `git push` para subir los nuevos commits al repositorio de GitHub.

Por último, añadimos un [Rebase.md](#) con los cambios que hayamos realizado.



Tarea 8: Versionado con tags y releases en GitHub

Creamos dos tags en git simulando dos versiones diferentes, y las pusheamos a GitHub.

```
root@daniel-MS-7E28:/home/terraform-localstack# git tag -a v1.0 -m "Primera versión funcional"
root@daniel-MS-7E28:/home/terraform-localstack# git tag -a v1.1 -m "Mejoras en la infraestructura"
root@daniel-MS-7E28:/home/terraform-localstack# git tag
v1.0
v1.1
root@daniel-MS-7E28:/home/terraform-localstack# git push origin --tags
Enter passphrase for key '/root/.ssh/id_ed25519':
Enumerando objetos: 28, listo.
Contando objetos: 100% (28/28), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (26/26), listo.
Escribiendo objetos: 100% (28/28), 5.25 KiB | 1.75 MiB/s, listo.
Total 28 (delta 8), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (8/8), done.
To github.com:Danimartinez-1997/Terraform.git
 * [new tag]          v1.0 -> v1.0
 * [new tag]          v1.1 -> v1.1
```

En nuestro repositorio de GitHub, creamos una Release asociada a cada uno de los tags creados anteriormente.


Releases

Tags

Draft a new release

Find a release

now

 Danimartinez-1997

v1.1

7c9b54c

Compare


v1.1

Latest


Mejoras en la infraestructura

Assets


2

 Source code (zip)


8 minutes ago

 Source code (tar.gz)

8 minutes ago



now

 Danimartinez-1997

v1.0

7c9b54c


Compare

v1.0

Primera versión funcional

Assets

2



Creamos un archivo [Changelog.md](#) en el que detallamos los cambios de cada versión.

The screenshot shows a GitHub repository page for the file `Terraform / Changelog.md`. The file is 13 lines (10 loc) and 308 Bytes. The content of the file is as follows:

Changelog

[v1.1] - 2025-07-07

Añadido

- Se añadieron etiquetas a los recursos
- Se creó una salida (`output`) para mostrar el nombre del bucket

Cambiado

- Se actualizó la versión del proveedor AWS

[v1.0] - 2025-07-06

Añadido

- Infraestructura inicial: bucket S3 y tabla DynamoDB