

Desafío Número 8 - Bootcamp Devops Engineer

Alumno: Ignacio Peretti

Objetivo:

Este desafío tiene como objetivo integrar en una práctica lo que vimos de terraform, github actions y configurar un provider de AWS para efectivamente crear un recurso en este proveedor cloud.

Requisitos:

1. Realizar la PoC aplicando todos los pasos detallados en el post propuesto.
2. Configurar un repositorio propio para que pueda alojar el proyecto.
3. Crear unas credenciales en nuestra sandbox de AWS Academy
4. Crear una cuenta en Terraform Cloud (no requiere tarjeta de crédito)

Entregables:

Los entregables establecidos para este proyecto con:

1. Código fuente de todo lo producido para la PoC.
2. Genera la documentación necesaria para ser incorporada a la base de conocimientos del equipo.
3. Explica qué mejoras harías a este pipeline.
4. Evidencia de las pruebas con resultado exitoso.

Aplicamos los pasos siguiendo el post propuesto para este desafío.

<https://developer.hashicorp.com/terraform/tutorials/automation/github-actions>

Configurar HCP Terraform.

Creamos un espacio de trabajo llamado `learn-terraform-github-actions`.

Buscamos las credenciales del AWS que deseamos usar para el espacio de trabajo o creamos un nuevo par de claves, luego agregamos nuestras credenciales como variables de entorno para nuestro **Workspace**.

HCP Terraform utilizará estas credenciales para autenticarse en AWS.

Workspace variables (3)

Variables defined within a workspace always overwrite variables from variable sets that have the same type and the same key. Learn more about variable set [precedence](#).

Key	Value	Category	
AWS_ACCESS_KEY_ID	AKIAZQ3DNXC2MYKEUCHE	env	...
AWS_REGION	us-east-1	env	...
AWS_SECRET_ACCESS_KEY	Sensitive - write only	env	...

Sensitive

+ Add variable

Por último, nos dirigimos a la página Tokens en la configuración de usuario de Terraform HCP. Creamos el token con el nombre **Github-Actions**.

Tokens (1)



GitHub Actions Expires September 11th 2024



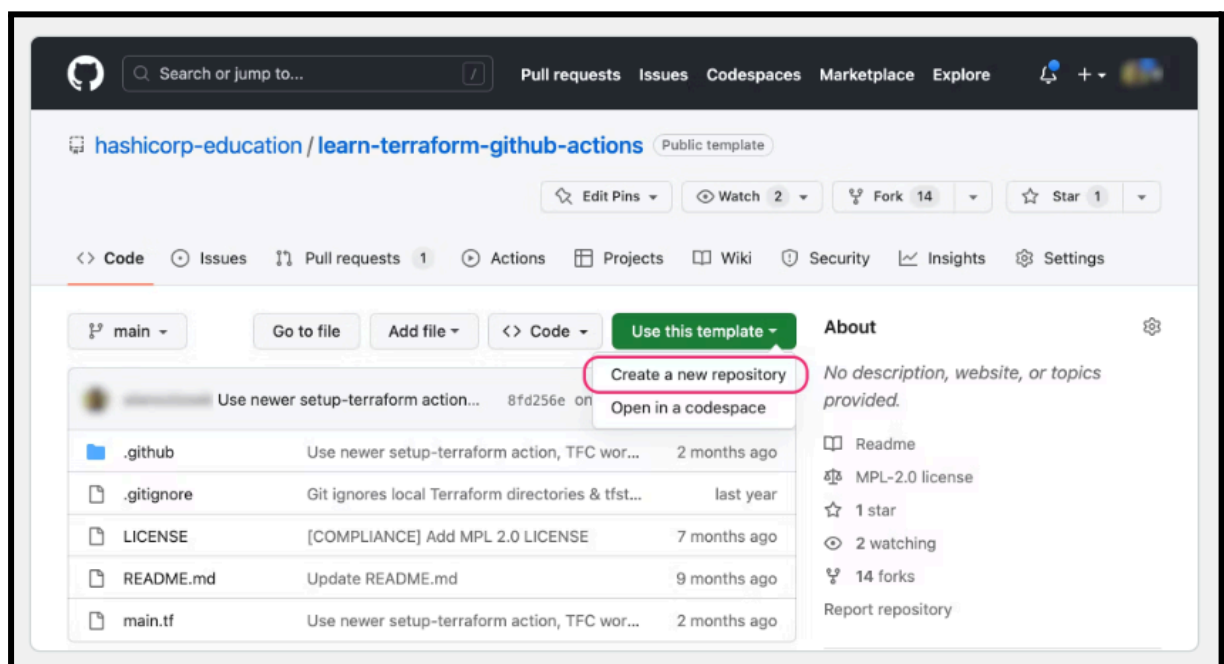
Created 2 hours ago by user ignacioperetti

Last used 33 minutes ago

Configurar un repositorio en GitHub.

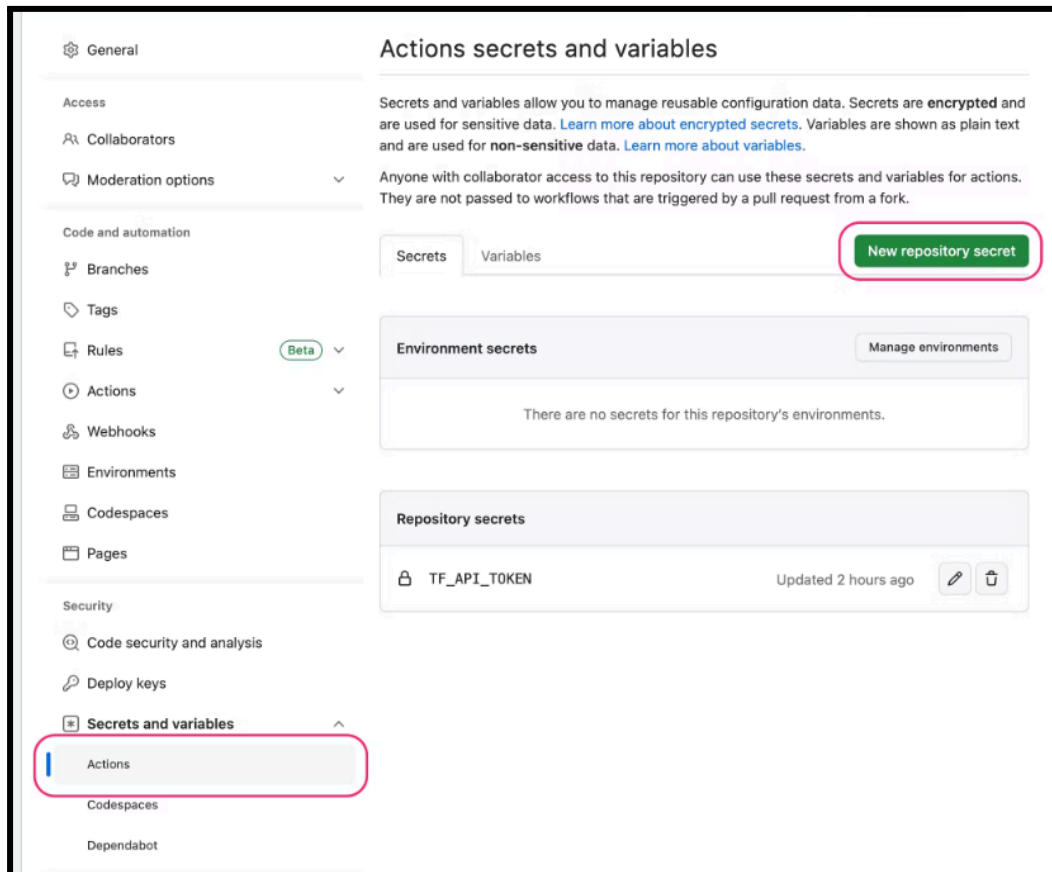
Navegaremos hasta el repositorio que nos indica en el post, seleccionamos usar esta plantilla y luego crearemos un nuevo repositorio.

<https://github.com/hashicorp-education/learn-terraform-github-actions>



Seleccionamos nuestra cuenta de Github e ingresamos learn-terraform-github-actions como nombre, seleccionamos público y hacemos click en crear repositorio.

En el nuevo repositorio nos dirigimos a configuración, secretos y variables y seleccionamos acciones.



Ahora seleccionamos la opción Nuevo repositorio secreto y creamos uno llamado **TF_API_TOKEN** y establecemos como valor el token de API terraform HCP que creamos anteriormente.

Luego, clonamos el repositorio.

```
git clone git@github.com:SU-NOMBRE-DE-USUARIO/learn-terraform-github-actions
```

Recuerde cambiar SU-NOMBRE-DE-USUARIO por el que usted haya elegido.

Revisamos el workspace clonado.

main.tf Contiene la configuración de Terraform para implementar una instancia EC2 de acceso público.

.github/workflows/terraform-plan.yml define el flujo de trabajo de acciones que ejecuta el plan Terraform.

.github/workflows/terraform-apply.yml define el flujo de trabajo de acciones que ejecuta Terraform apply.

A continuación necesitamos configurar los siguientes archivos.

.github/workflows/terraform-apply.yml

.github/workflows/terraform-plan.yml

```
.github/workflows/terraform-plan.yml

1  ## ...
2  on:
3    pull_request:
4
5  env:
6    TF_CLOUD_ORGANIZATION: "YOUR-ORGANIZATION-HERE"
7    TF_API_TOKEN: "${{ secrets.TF_API_TOKEN }}"
8    TF_WORKSPACE: "learn-terraform-github-actions"
9    CONFIG_DIRECTORY: "./"
10  ## ...
```

En este punto deberá reemplazar “YOUR-ORGANIZATION-HERE” con el nombre de su organización en ambos archivos.

Luego creamos un pull-request

Crea una nueva rama en tu repositorio bifurcado llamada update-tfc-org.

```
git checkout -b 'update-tfc-org'
```

Ahora confirme los cambios de nombre de la organización que realizó en los archivos de flujo de trabajo.

```
git add .github/workflows
```

Confirme estos cambios con un mensaje.

```
git commit -m 'Use our HCP Terraform organization'
```

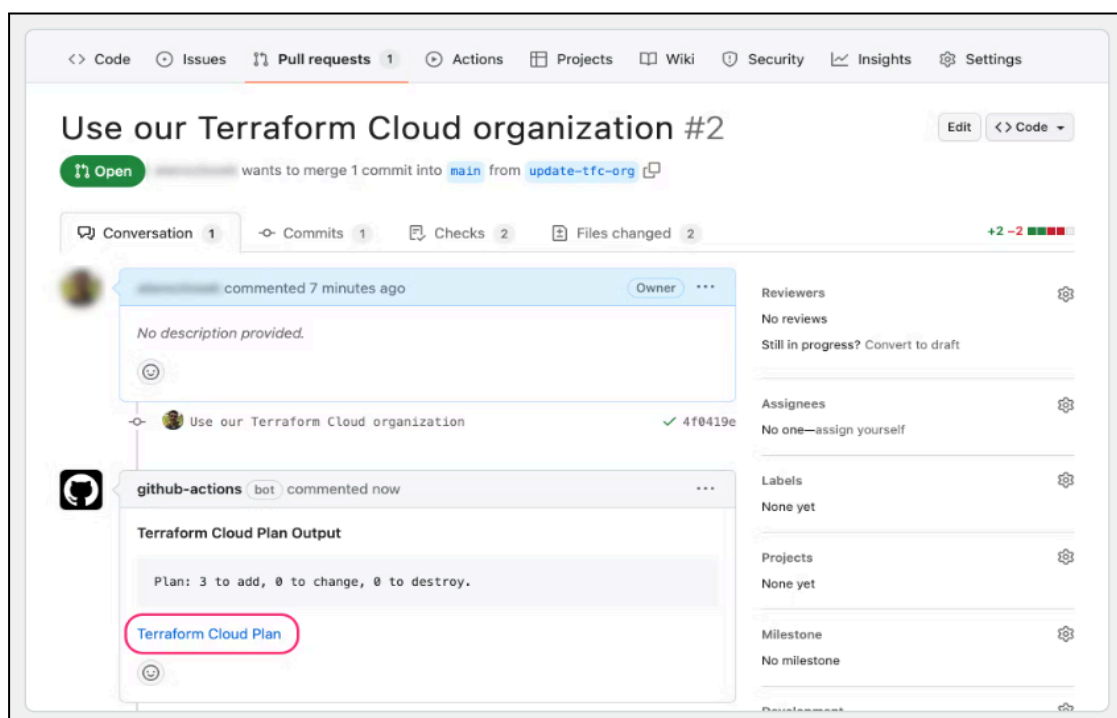
Impulsar estos cambios.

```
git push origin update-tfc-org
```

A continuación, abra un pull-request desde la rama update-tfc-org, seleccione la rama main.

Siguiente nos dirigimos hasta nuestra solicitud de pull-request y el trigger activará el Terraform Plan.

Cuando se complete el flujo de trabajo, se agregará un comentario con un enlace al plan especulativo.



Nos dirigimos al Plan de Terraform HCP y podemos ver su estado.

Current Run



Triggered from Terraform Cloud CI by Author (IgnacioPeretti) for SHA (b7...

CURRENT

✓ Applied

#run-J9ynn51TSKd73XRK | ignacioperetti triggered via API

an hour ago

Terraform planea crear tres recursos, que coincidan con el comentario de la solicitud de extracción.

Triggered from Terraform Cloud CI by Author (IgnacioPeretti) for SHA (b7665cd) CURRENT

✓ Applied

Plan & apply duration

1 minute

Resources changed

+3 ~0 -0



ignacioperetti triggered a run from API an hour ago

Run Details



Plan finished an hour ago

Resources: 3 to add, 0 to change, 0 to destroy

Started an hour ago > Finished an hour ago

+ 3 to create

Filter resources by address...

Filter by action

Show data sources

Terraform 1.9.4

Download raw log

> + aws aws_instance.web

> + aws aws_security_group.web-sg

> + random random_pet.sg

> Outputs 1 planned to change

Download Sentinel mocks

Sentinel mocks can be used for testing your Sentinel policies

En HCP Terraform, expanda la sección Apply Finished.

HCP Terraform muestra los recursos que creó y la dirección web de la instancia EC2.

✓ **Apply finished** an hour ago

Resources: 3 added, 0 changed, 0 destroyed ^

Started an hour ago > Finished an hour ago

+ 3 created

Filter by action ▾

 Terraform 1.9.4 [Download raw log](#)

> + aws_instance.web

✓ Created id=i-068bf23b919a8cff3

> + aws_security_group.web-sg

✓ Created id=sg-0e7a3f4ec3f7e87d5

> + random_pet.sg

✓ Created id=true-sunbeam

▼ **Outputs** 1 total

web-address : "ec2-44-243-117-56.us-west-2.compute.amazonaws.com:8080"

State versions created:

[ignacio-devops/learn-terraform-github-actions#sv-pw18Cjouh3WE4gP9](#) (Aug 12, 2024 19:24:47 pm)

Verificamos que la instancia EC2 este aprovisionada en Github Actions

Summary

Jobs

✓ Terraform Apply

Run details

Usage

Workflow file

Terraform Apply
succeeded 1 hour ago in 1m 26s

▼ ✓ Apply

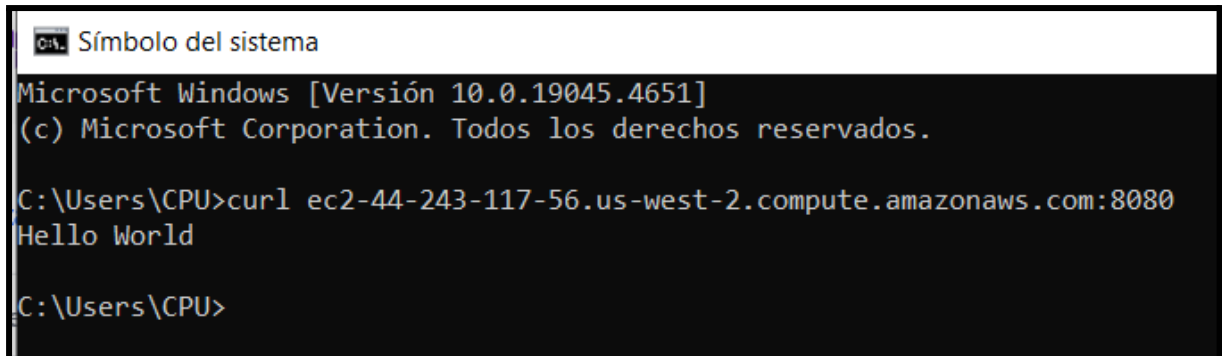
```
econds":30},"type":"apply_progress"}
38 {"@level":"info","@message":"aws_instance.web: Creation complete after 32s [id=i-068bf23b919a8cff3]","@module":"terraform.ui","@timestamp":"2024-08-12T22:24:47.181166Z","hook":{"resource":{"addr":"aws_instance.web","module":"","resource":"aws_instance.web","implied_provider":"aws","source_type":"aws_instance","resource_name":"web","resource_key":null},"action":"create","id_key":"id","id_value":"i-068bf23b919a8cff3","elapsed_seconds":32},"type":"apply_complete"}
39 {"@level":"info","@message":"Apply complete! Resources: 3 added, 0 changed, 0 destroyed.","@module":"terraform.ui","@timestamp":"2024-08-12T22:24:47.534951Z","changes":{"add":3,"change":0,"import":0,"remove":0,"operation":"apply"},"type":"change_summary"}
40 {"@level":"info","@message":"Outputs: 1","@module":"terraform.ui","@timestamp":"2024-08-12T22:24:47.535125Z","outputs":{"web-address":{"sensitive":false,"type":"string","value":"ec2-44-243-117-56.us-west-2.compute.amazonaws.com:8080"},"type":"outputs"}}
41
42 View Run in Terraform Cloud: https://app.terraform.io/app/ignacio-devops/workspaces/learn-terraform-github-actions/runs/run-J9ynn51TSKd73XRK
43 {
44   "run_id": "run-J9ynn51TSKd73XRK",
45   "run_link": "https://app.terraform.io/app/ignacio-devops/workspaces/learn-terraform-github-actions/runs/run-J9ynn51TSKd73XRK",
46   "run_status": "applied",
47   "status": "Success"
48 }
```

> ✓ Post Checkout

> ✓ Complete job

Por último, verifique que la instancia EC2 esté disponible públicamente.

Utilice el comando curl con el web-address valor de salida que podemos ver en el Apply finished.

A screenshot of a Windows command prompt window titled 'Símbolo del sistema'. The window shows the following text: 'Microsoft Windows [Versión 10.0.19045.4651]', '(c) Microsoft Corporation. Todos los derechos reservados.', 'C:\Users\CPU>curl ec2-44-243-117-56.us-west-2.compute.amazonaws.com:8080', 'Hello World', and 'C:\Users\CPU>'. The command prompt is running on a Windows system, and the output of the curl command is 'Hello World'.

Hemos configurado con éxito un flujo de trabajo completo de GitHub Actions para implementar un servidor web de acceso público dentro de un espacio de trabajo de HCP Terraform.

Como requisito tenemos que definir algunas mejoras que le haríamos a nuestro pipeline de terraform.

Las mejoras que incluiría en este pipeline de terraform son:

El uso de variables, podríamos incluirlas en parámetros que pueden variar como la región o el tipo de instancia.


El uso de módulos para encapsular algunos recursos que se pueden repetir, por ejemplo podríamos crear un módulo para la instancia de EC2 y otro para el grupo de seguridad.


Recuerda destruir los recursos y el espacio de trabajo HCP Terraform que creaste para este tutorial.

Vaya al **learn-terraform-github-actions**, ponga en cola un plan de destrucción y aplíquelo. Luego, elimine el espacio de trabajo de HCP Terraform.

Si necesita más información sobre cómo realizar un plan de destrucción puede seguir los pasos de esta guía.

<https://developer.hashicorp.com/terraform/tutorials/cloud-get-started/cloud-destroy>.

 **ignacioperetti** triggered a **destroy run** from UI a few seconds ago Run Details ▾

 **Plan finished** a few seconds ago

Resources: 0 to add, 0 to change, **3 to destroy** ^



Started 2 minutes ago > Finished 2 minutes ago



— 3 to destroy



Filter by action ▾

☐ Show data sources Terraform 1.9.4


Download raw log


> —  aws_instance.web 


> —  aws_security_group.web-sg 

> —  random_pet.sg 

> **Outputs** 1 planned to change

 Download Sentinel mocks

 Sentinel mocks can be used for [testing your Sentinel policies](#)

 **Apply finished** a few seconds ago

Resources: 0 added, 0 changed, **3 destroyed** ^



Started a minute ago > Finished a few seconds ago

— 3 destroyed



Filter by action ▾

 Terraform 1.9.4



Download raw log

> —  aws_instance.web 

✓ Deleted id=i-068bf23b919a8cff3

> —  aws_security_group.web-sg 

✓ Deleted id=sg-0e7a3f4ec3f7e87d5

> —  random_pet.sg 

✓ Deleted id=true-sunbeam

State versions created:

[ignacio-devops/learn-terraform-github-actions#sv-CcuJ9dUd5yxqx7tc](#) (Aug 12, 2024 20:48:05 pm)

Fin del documento.