MundosE - *https://mundose.com/index.html*

Diplomatura en DevOps

Edición 2403

Informe Práctico Integrador FINAL (PIN-F)

Tutor:

- Guazzardo, Marcelo

Grupo 6:

- Hector Barrios hdbarrios@gmail.com
- Juan Pablo Heyda juanpabloh.123@gmail.com
- Renzo Carletti renzocarletti@hotmail.com / pipito1498@gmail.com
- Johanna Dominguez johisd9@hotmail.com
- Lucas Bufano lucas.bufano2@gmail.com

Repositorio de GitHub Público:

- https://github.com/hdbarrios/devops-g6-pin-final

MundosE - https://mundose.com/index.html

Objetivos:

- Instalar terraform
- Configurar usuario programático en AWS IAM
- Desarrollar código terraform para crear una instancia EC2 y un EKS con 3 nodos
- Utilizar github actions para desplegar nginx, grafana y prometheus en EKS, usando helm
    - Plus - destruir toda la infra desde github actions.

Anexos:

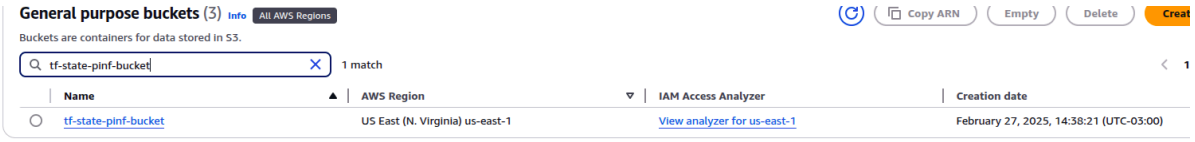
- Costos relacionados a la implementación en AWS.
- Instalar terraform en estación de trabajo.
- Evaluar código terraform y credenciales aws:
- Imagen de “validation_infra” Job que crea lo solicitado sobre el eks

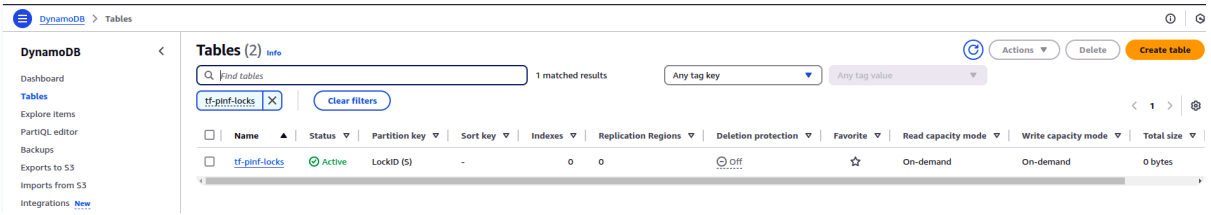
Crear el backend:

Usando **`create_backen.sh`**, si no tienes el archivo ~/.aws/config creado, ejecuta **`aws configure –profile terraform-admin`** (se aconseja si tienes más de una cuenta por administrar usar profiles)

MundosE - *https://mundose.com/index.html*

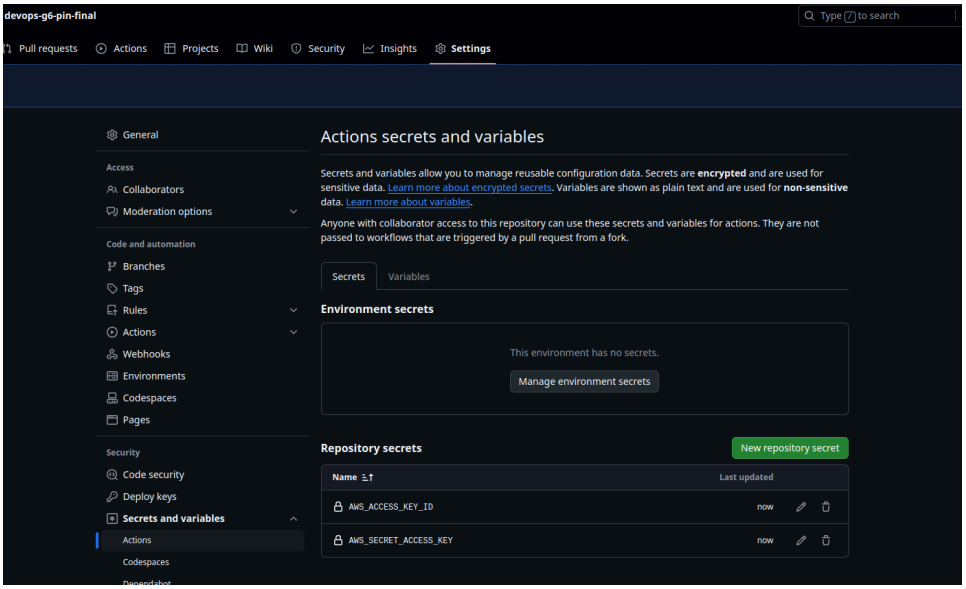

```
{
    "Table": {
        "AttributeDefinitions": [
            {
                "AttributeName": "LockID",
                "AttributeType": "S"
            }
        ],
        "TableName": "tf-pinf-locks",
        "KeySchema": [
            {
                "AttributeName": "LockID",
                "KeyType": "HASH"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2025-02-27T14:38:22.653000-03:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 0,
            "WriteCapacityUnits": 0
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-east-1:536697232168:table/tf-pinf-locks",
        "TableId": "eba11c73-f500-42a3-9598-26e62e8f1545",
        "BillingModeSummary": {
            "BillingMode": "PAY_PER_REQUEST"
        }
    }
}
```



GitHub Actions:

Para la construcción de la infraestructura se requiere acceso a aws. Desde el repositorio de github vamos settings/secrets/actions y agregamos dos New Repository Secret:



MundosE - *https://mundose.com/index.html*

El Workflows diseñados fueron:

.github/workflows/

- apply.yml (dos Jobs con un total de 13 steps)
- destroy.yml (un Job con 6 steps)

Terraform Apply, solo se ejecuta en condiciones;

```
on:
  push:
    branches:
      - master
  pull_request:
    branches:
      - master
  workflow_dispatch:
    branches:
      - master
```

Los Jobs:

terraform_apply, tiene 7 steps

actions/checkout@v4
Install Terraform
Terraform version
Terraform ini
Terraform validation
Terraform plan, *solo se ejecuta para push y pull_request event*
Terrafom apply, *solo se ejecuta en push event*

validacion_infra, tiene 4 steps, se ejecuta sobre el mismo worker que el job terraform_apply, y si y solo si se cumple: que terraform_apply este completo y se ejecuta solo en un push event

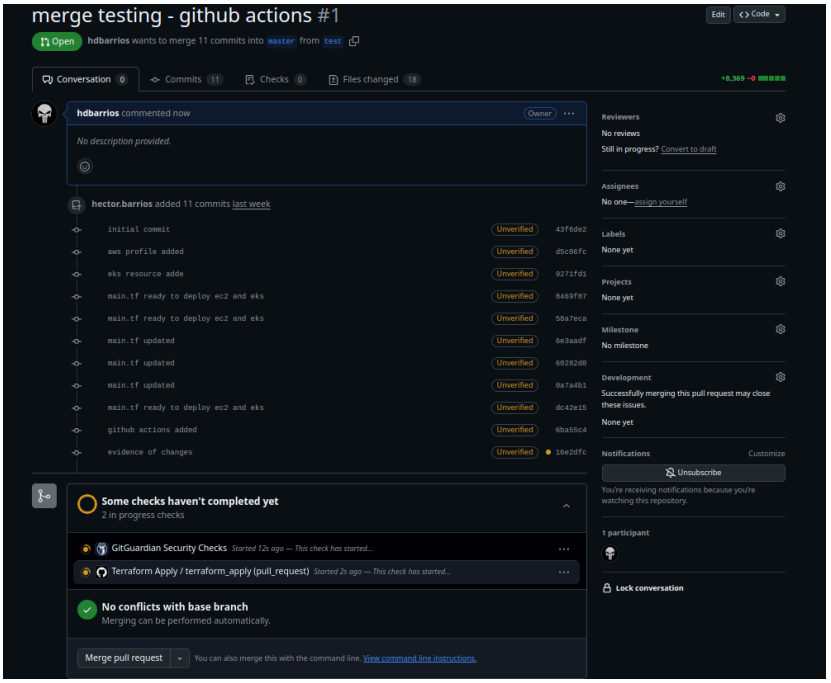
actions/checkout@v4
Install kubectl
Install Helm
Install AWS CLI
Verificando el cluster EKS, este step ejecuta la instalacion de nginx, grafana y prometheus
Port-forward prometheus-service, es un jstep para ejecutar un port forward desde el worker del actions.

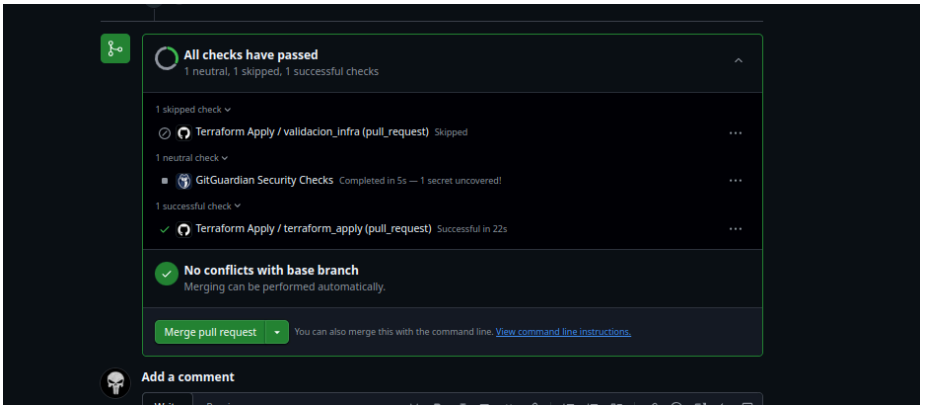
Terraform Destroy, solo se ejecuta con la condición;

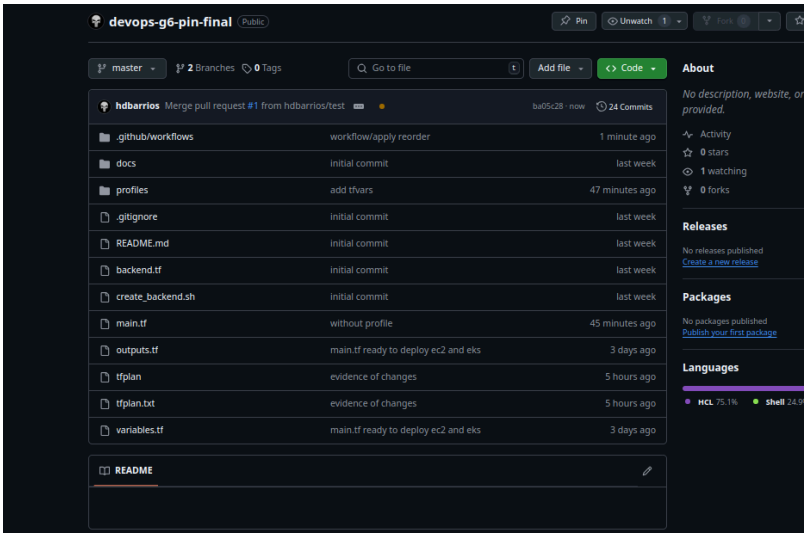
```
on:
  workflow_dispatch:
    branches:
      - master
```

MundosE - *https://mundose.com/index.html*

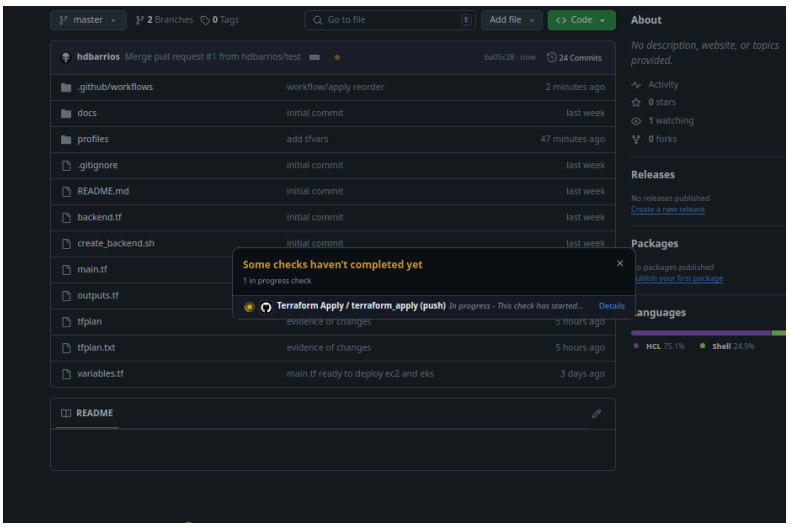
Ejemplo de ejecución con PR: https://github.com/hdbarrios/devops-g6-pin-final/pull/1



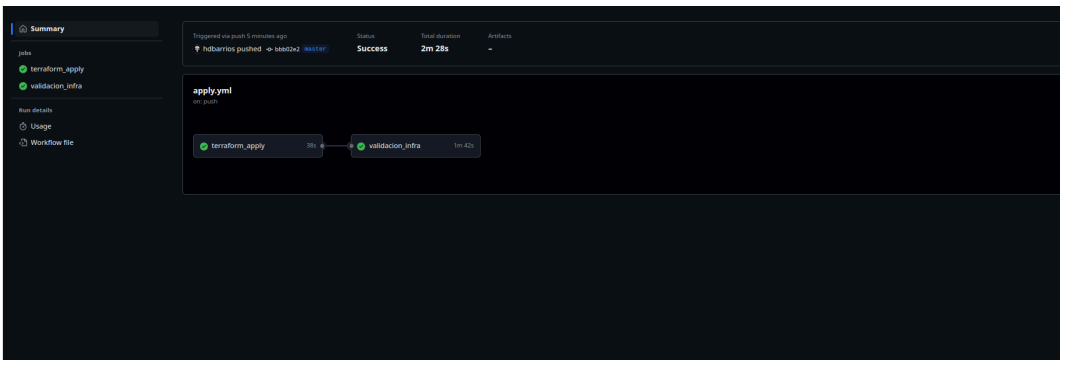


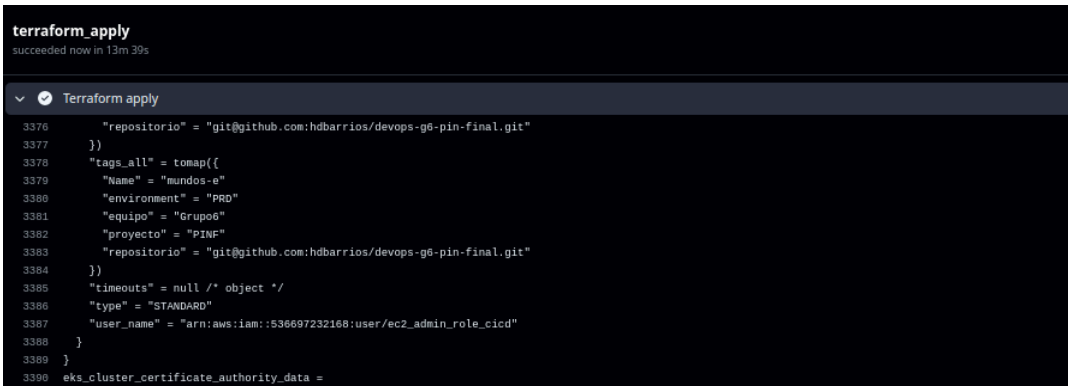


MundosE - *https://mundose.com/index.html*

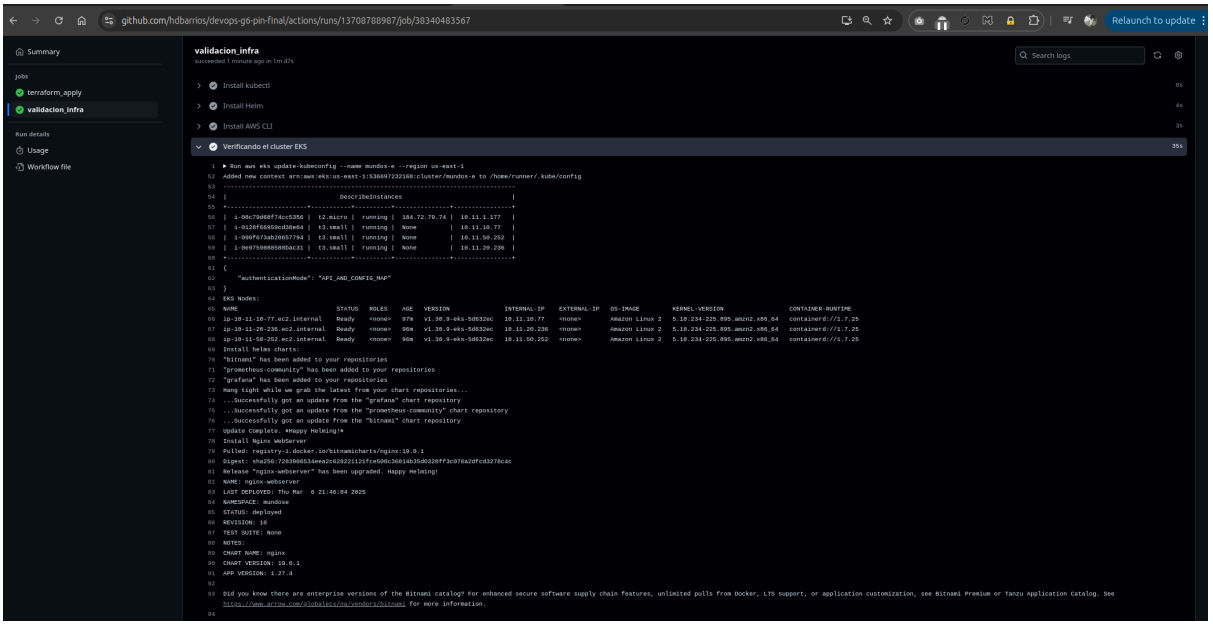


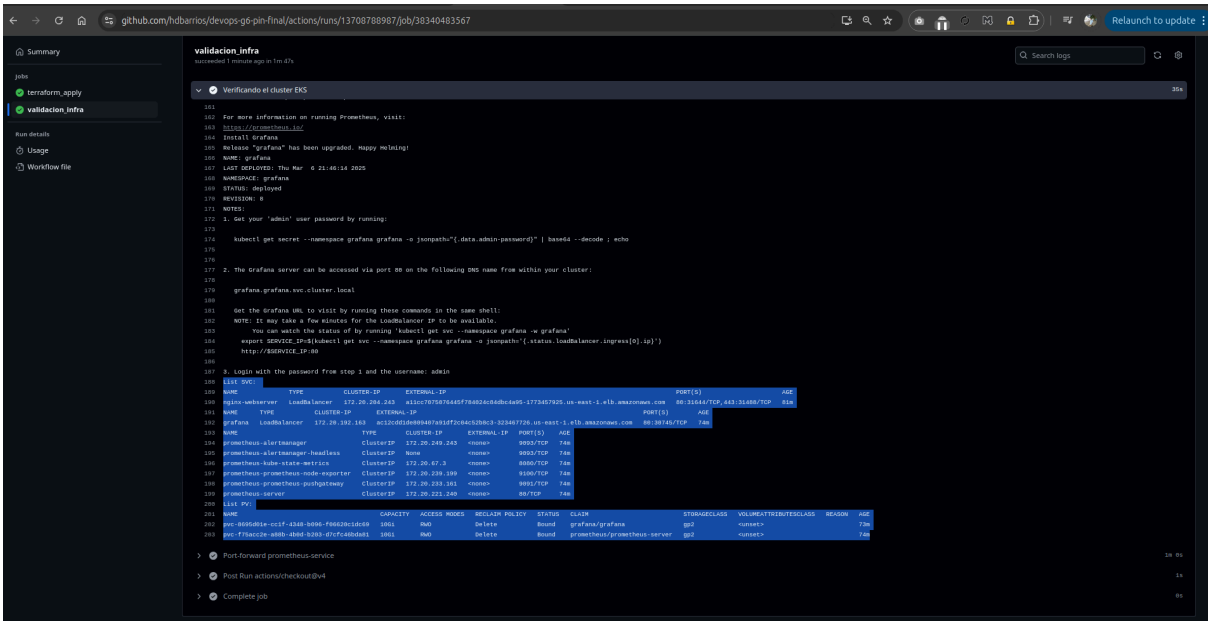
Details: https://github.com/hdbarrios/devops-g6-pin-final/actions/runs/13710001184

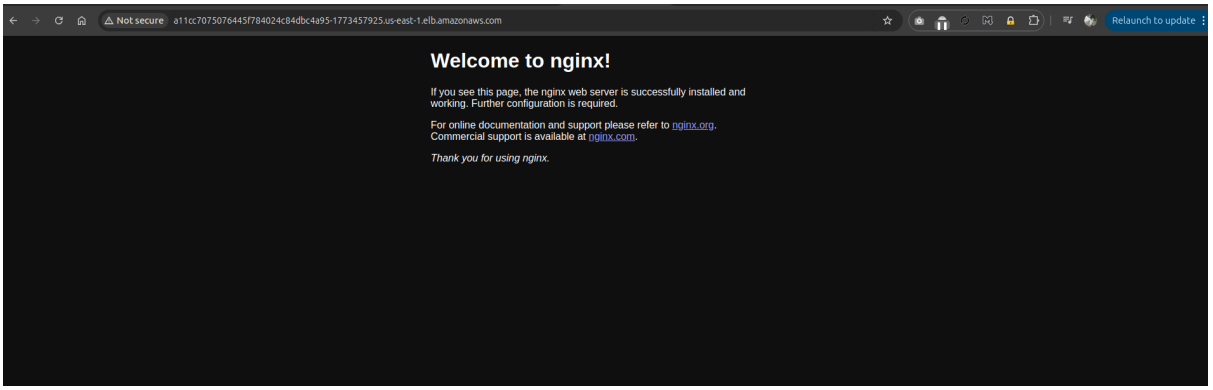




MundosE - https://mundose.com/index.html



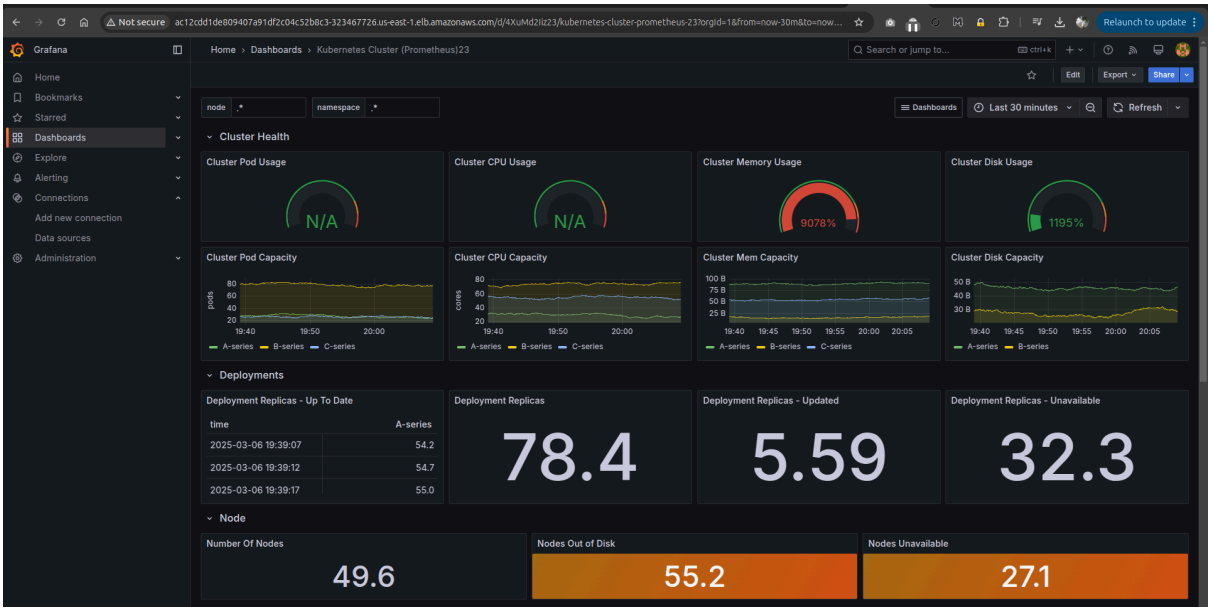




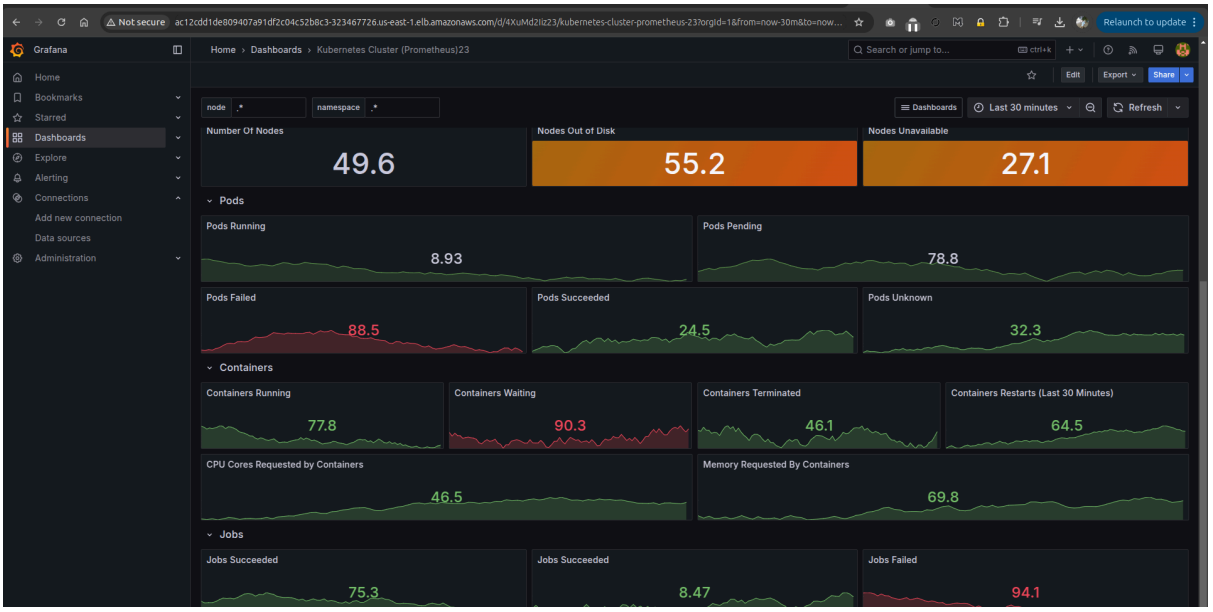
MundosE - *https://mundose.com/index.html*

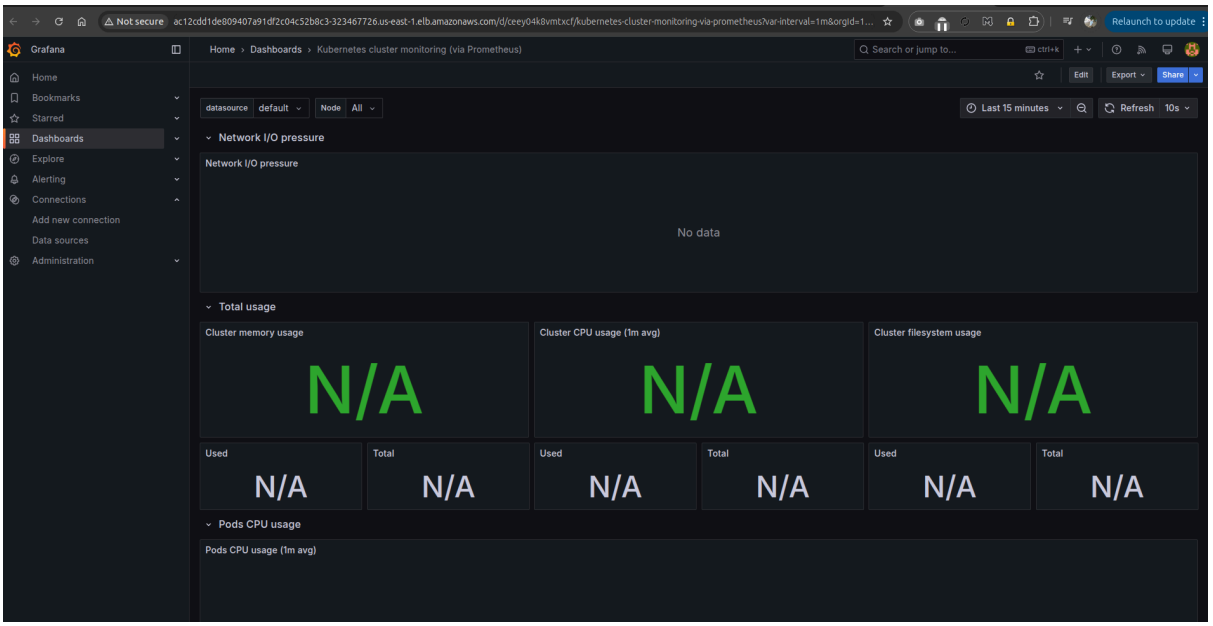
Adicional:(elb)/pinfg6/grupo6.html descripcion del equipo 6

Grafana:

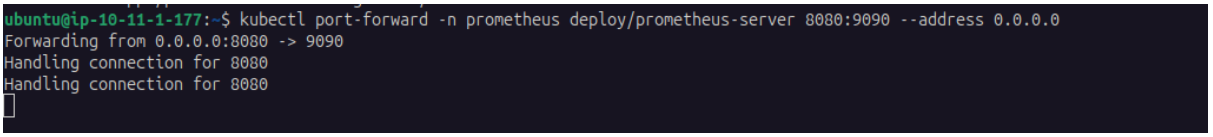


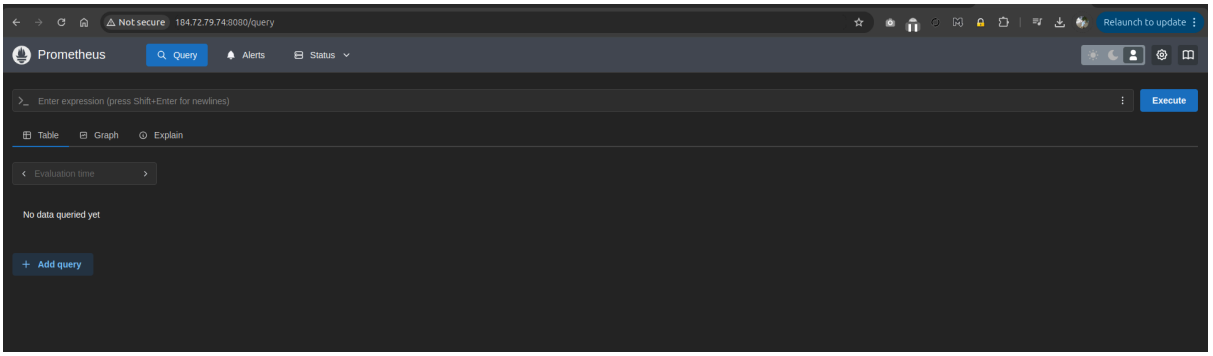
MundosE - https://mundose.com/index.html





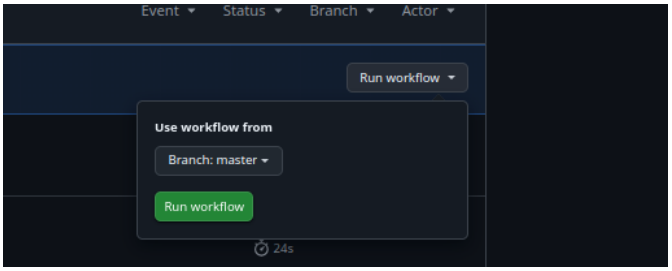
Port-Froward Manual: Desde la instancia EC2 conexión SSH.

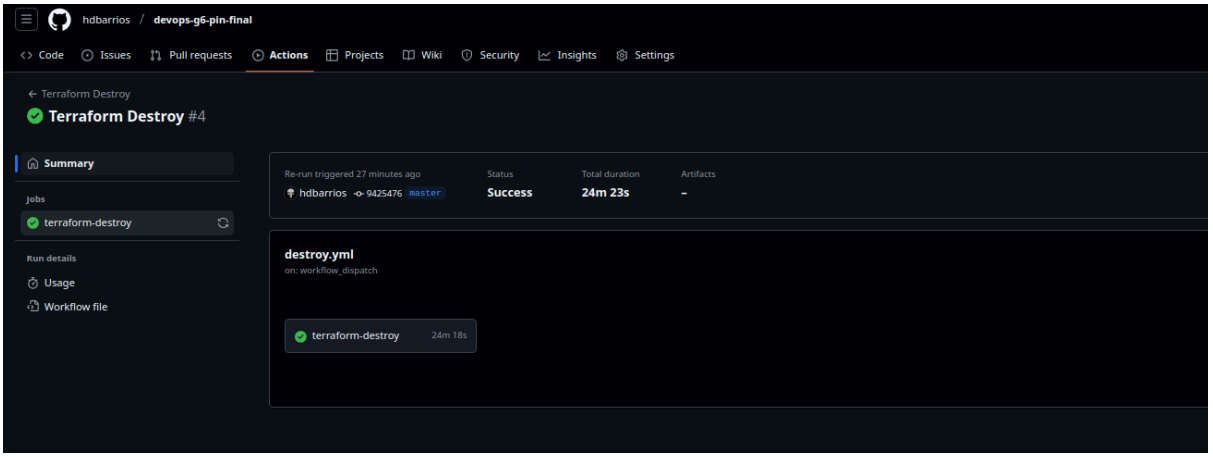




MundosE - *https://mundose.com/index.html*

Para el Destroy: Ejecucion de Actinons con Master:

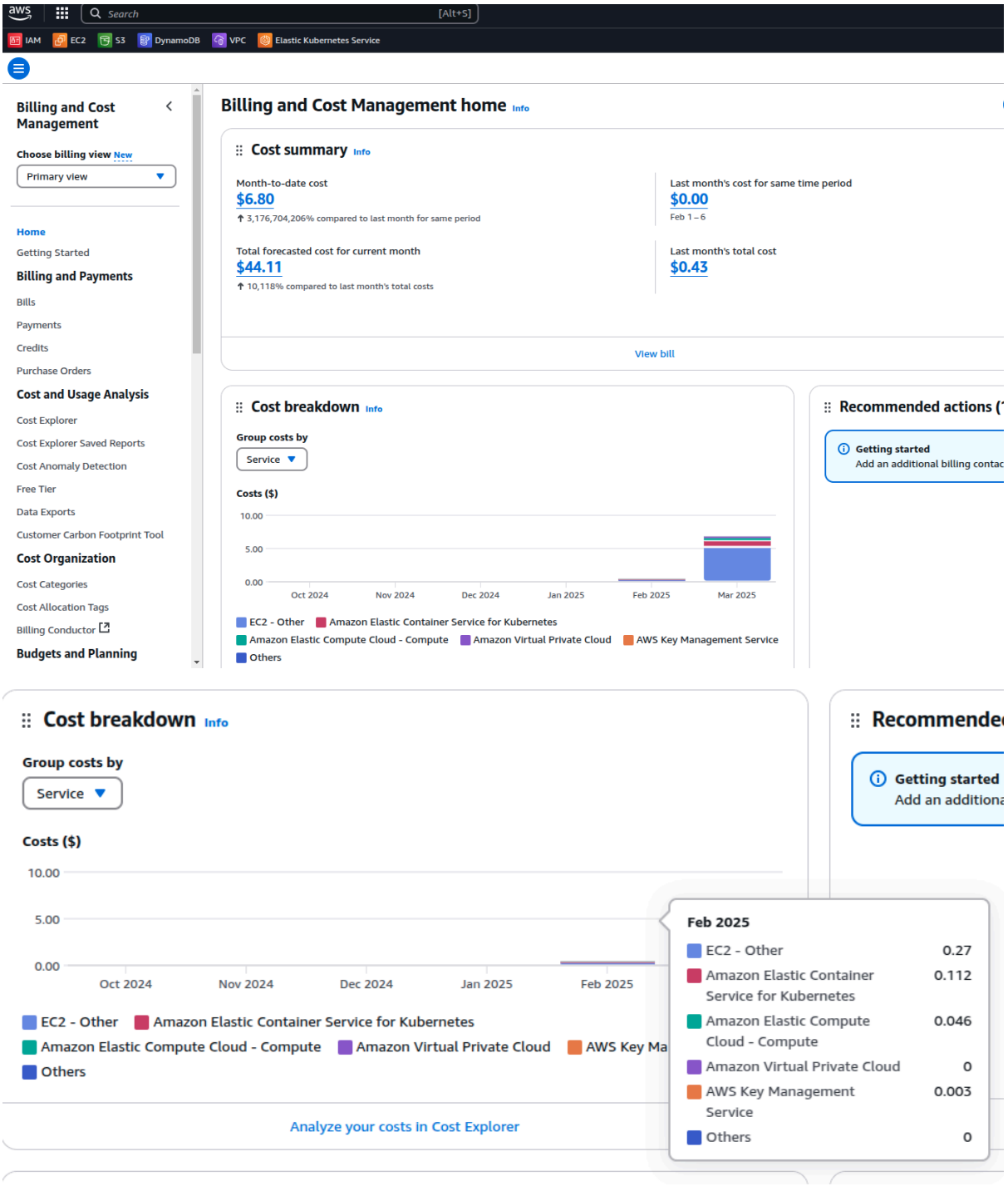




MundosE - *https://mundose.com/index.html*

ANEXOS:

Costos relacionados:



MundosE - *https://mundose.com/index.html*

Pre- Requisitos:

Instalar terraform

https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli

```
$ sudo apt-get update && sudo apt-get install -y gnupg software-properties-common

$ wget -O- https://apt.releases.hashicorp.com/gpg | \
gpg --dearmor | \
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null

$ gpg --no-default-keyring \
--keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
--fingerprint

$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \
sudo tee /etc/apt/sources.list.d/hashicorp.list

$sudo apt update && sudo apt-get install terraform -y

$ terraform -version
  Terraform v1.10.2
  on linux_amd64
```

Adicional puedes instalar `tfenv` para tener varias versiones de Terraform:

https://github.com/tfutils/tfenv

```
$ export PATH="$HOME/.tfenv/bin:$PATH"   #INCLUIR EN ~/.bash_profile
$ tfenv
tfenv 3.0.0
Usage: tfenv <command> [<options>]

Commands:
   install       Install a specific version of Terraform
   use           Switch a version to use
   uninstall     Uninstall a specific version of Terraform
   list          List all installed versions
   list-remote   List all installable versions
   version-name  Print current version
   init          Update environment to use tfenv correctly.
   pin           Write the current active version to ./.terraform-version

$ tfenv use 1.10.2
No installed versions of terraform matched '1.10.2:^1.10.2$'. Trying to install a
matching version since TFENV_AUTO_INSTALL=true
Installing Terraform v1.10.2
Downloading                release                tarball                from
https://releases.hashicorp.com/terraform/1.10.2/terraform_1.10.2_linux_amd64.zip
######################################################################################
######################################################################################
################## 100.0%
Downloading             SHA             hash             file             from
https://releases.hashicorp.com/terraform/1.10.2/terraform_1.10.2_SHA256SUMS
```

```
Not instructed to use Local PGP (/home/hbarrios/.tfenv/use-{gpgv,gnupg}) & No keybase install found, skipping OpenPGP signature verification
Archive:  /tmp/tfenv_download.Yfp59d/terraform_1.10.2_linux_amd64.zip
  inflating: /home/hbarrios/.tfenv/versions/1.10.2/LICENSE.txt
  inflating: /home/hbarrios/.tfenv/versions/1.10.2/terraform
Installation of terraform v1.10.2 successful. To make this your default version, run 'tfenv use 1.10.2'
Switching default version to v1.10.2
Default version (when not overridden by .terraform-version or TFENV_TERRAFORM_VERSION) is now: 1.10.2
```

Evaluar código terraform y credenciales aws:

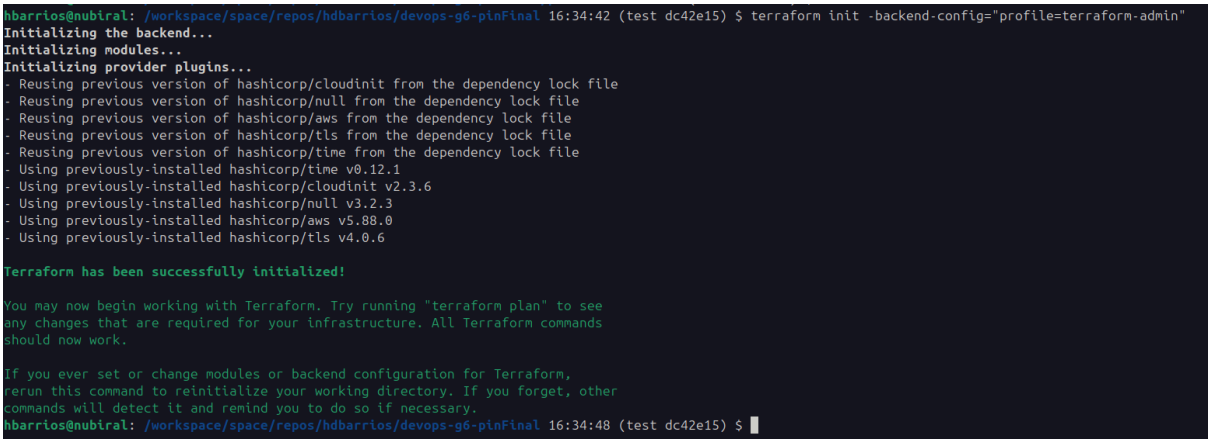
Lista de comandos de Terraform que puedes usar para probar y luego aplicar tu configuración en los archivos de Terraform:

1. Inicializar Terraform

Primero, se debe inicializar el entorno de Terraform. Esto instalará los proveedores necesarios y configurará tu backend (si lo estás usando).

```
terraform init -backend-config="profile=terraform-admin"
```

Este comando se ejecuta una sola vez cuando configuras un nuevo proyecto Terraform o cuando haces cambios en los proveedores y módulos.



2. Verificar la configuración (Planificación)

Para revisar qué cambios realizará Terraform en tu infraestructura, se puede ejecutar el comando `terraform plan`. Este comando no realizará ningún cambio, solo mostrará una descripción detallada de lo que se va a hacer.

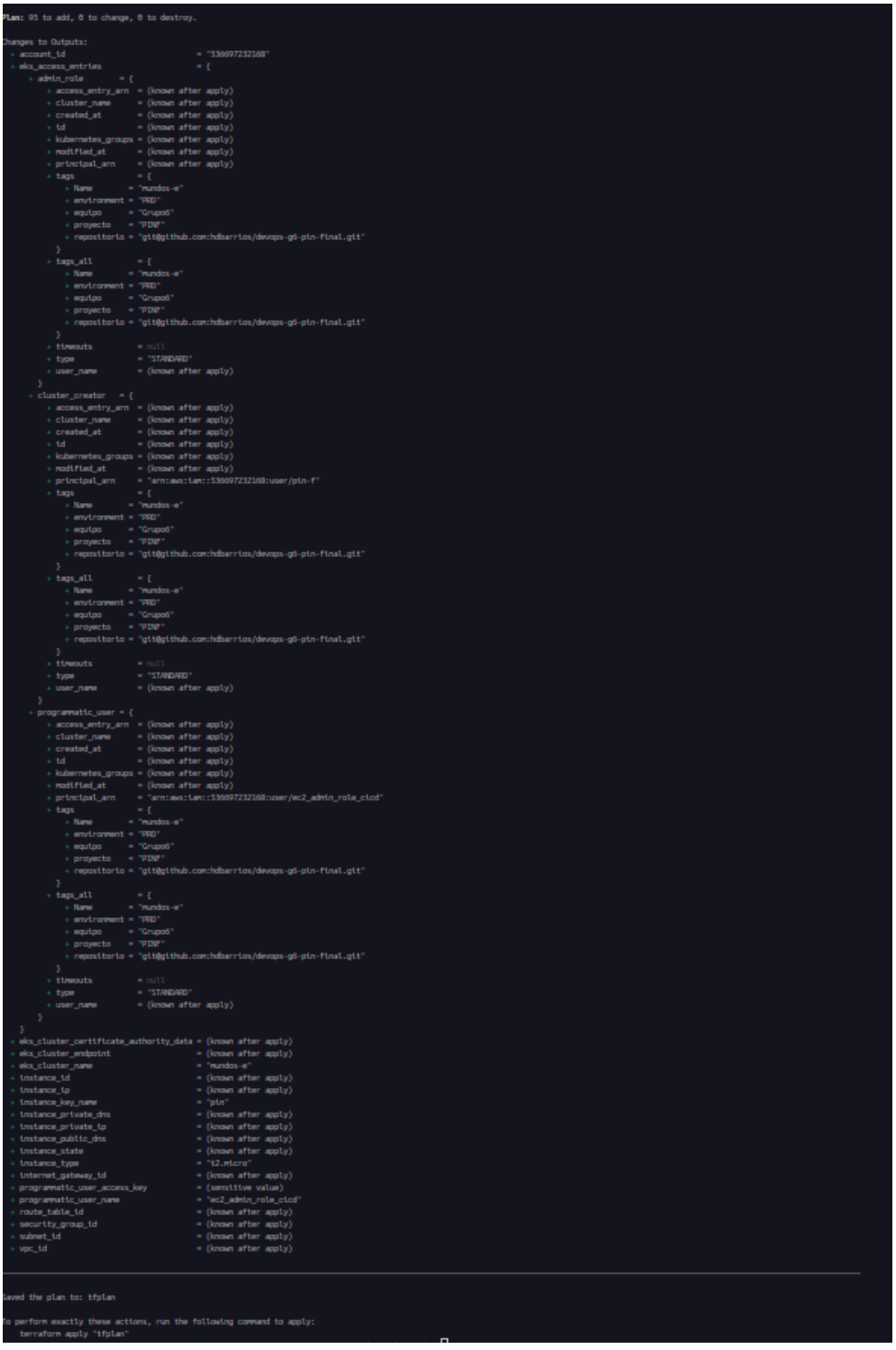
```
terraform plan -var-file=profiles/pinf.tfvars -out=tfplan
```

MundosE - *https://mundose.com/index.html*

```
hbarrios@nubiral: /workspace/space/repos/hdbarrios/devops-g6-pinFinal 09:26:12 (test ed4d2f7) $ terraform plan -var-file=profiles/pinf.tfvars -out=tfplan
Acquiring state lock. This may take a few moments...
data.aws_availability_zones.azs: Reading...
data.aws_availability_zones.azs: Read complete after 1s [id=us-east-1]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:
```



```
Plan: 95 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + account_id                          = "536697232168"
  + eks_access_entries                  = {
      + admin_role       = {
          + access_entry_arn  = (known after apply)
          + cluster_name      = (known after apply)
          + created_at        = (known after apply)
          + id                = (known after apply)
          + kubernetes_groups = (known after apply)
          + modified_at       = (known after apply)
          + principal_arn     = (known after apply)
          + tags              = {
              + Name        = "mundos-e"
              + environment = "PRD"
              + equipo      = "Grupo6"
              + proyecto    = "PINF"
              + repositorio = "git@github.com:hdbarrios/devops-g6-pin-final.git"
            }
          + tags_all          = {
              + Name        = "mundos-e"
              + environment = "PRD"
              + equipo      = "Grupo6"
              + proyecto    = "PINF"
              + repositorio = "git@github.com:hdbarrios/devops-g6-pin-final.git"
            }
          + timeouts          = null
          + type              = "STANDARD"
          + user_name         = (known after apply)
        }
      + cluster_creator  = {
          + access_entry_arn  = (known after apply)
          + cluster_name      = (known after apply)
          + created_at        = (known after apply)
          + id                = (known after apply)
          + kubernetes_groups = (known after apply)
          + modified_at       = (known after apply)
          + principal_arn     = "arn:aws:iam::536697232168:user/pin-f"
          + tags              = {
              + Name        = "mundos-e"
              + environment = "PRD"
              + equipo      = "Grupo6"
              + proyecto    = "PINF"
              + repositorio = "git@github.com:hdbarrios/devops-g6-pin-final.git"
            }
          + tags_all          = {
              + Name        = "mundos-e"
              + environment = "PRD"
              + equipo      = "Grupo6"
              + proyecto    = "PINF"
              + repositorio = "git@github.com:hdbarrios/devops-g6-pin-final.git"
            }
          + timeouts          = null
          + type              = "STANDARD"
          + user_name         = (known after apply)
        }
      + programmatic_user = {
          + access_entry_arn  = (known after apply)
          + cluster_name      = (known after apply)
          + created_at        = (known after apply)
          + id                = (known after apply)
          + kubernetes_groups = (known after apply)
          + modified_at       = (known after apply)
          + principal_arn     = "arn:aws:iam::536697232168:user/ec2_admin_role_cicd"
          + tags              = {
              + Name        = "mundos-e"
              + environment = "PRD"
              + equipo      = "Grupo6"
              + proyecto    = "PINF"
              + repositorio = "git@github.com:hdbarrios/devops-g6-pin-final.git"
            }
          + tags_all          = {
              + Name        = "mundos-e"
              + environment = "PRD"
              + equipo      = "Grupo6"
              + proyecto    = "PINF"
              + repositorio = "git@github.com:hdbarrios/devops-g6-pin-final.git"
            }
          + timeouts          = null
          + type              = "STANDARD"
          + user_name         = (known after apply)
        }
    }
  + eks_cluster_certificate_authority_data = (known after apply)
  + eks_cluster_endpoint                   = (known after apply)
  + eks_cluster_name                       = "mundos-e"
  + instance_id                            = (known after apply)
  + instance_ip                            = (known after apply)
  + instance_key_name                      = "pin"
  + instance_private_dns                   = (known after apply)
  + instance_private_ip                    = (known after apply)
  + instance_public_dns                    = (known after apply)
  + instance_state                         = (known after apply)
  + instance_type                          = "t2.micro"
  + internet_gateway_id                    = (known after apply)
  + programmatic_user_access_key           = (sensitive value)
  + programmatic_user_name                 = "ec2_admin_role_cicd"
  + route_table_id                         = (known after apply)
  + security_group_id                      = (known after apply)
  + subnet_id                              = (known after apply)
  + vpc_id                                 = (known after apply)

Saved the plan to: tfplan

To perform exactly these actions, run the following command to apply:
    terraform apply "tfplan"
```

Para este proyecto se debe especificar el archivo de variables `pinf.tfvarss` con la opción `-var-file` para que Terraform use las configuraciones definidas en ese archivo.

Salida esperada: Terraform mostrará un resumen de los recursos que se van a crear, modificar o destruir.

Nota:

- *se puede ejecutar para validar sintaxis:* `terraform validate`

Permite tener código de salida si se quiere implementar CI/CD:

```
terraform plan -var-file=profiles/pinf.tfvarss -out=tfplan
-detailed-exitcode && echo $?
```

Código de salida 0: No hay cambios
Código de salida 1: Ocurrió un error
Código de salida 2: Se detectaron cambios

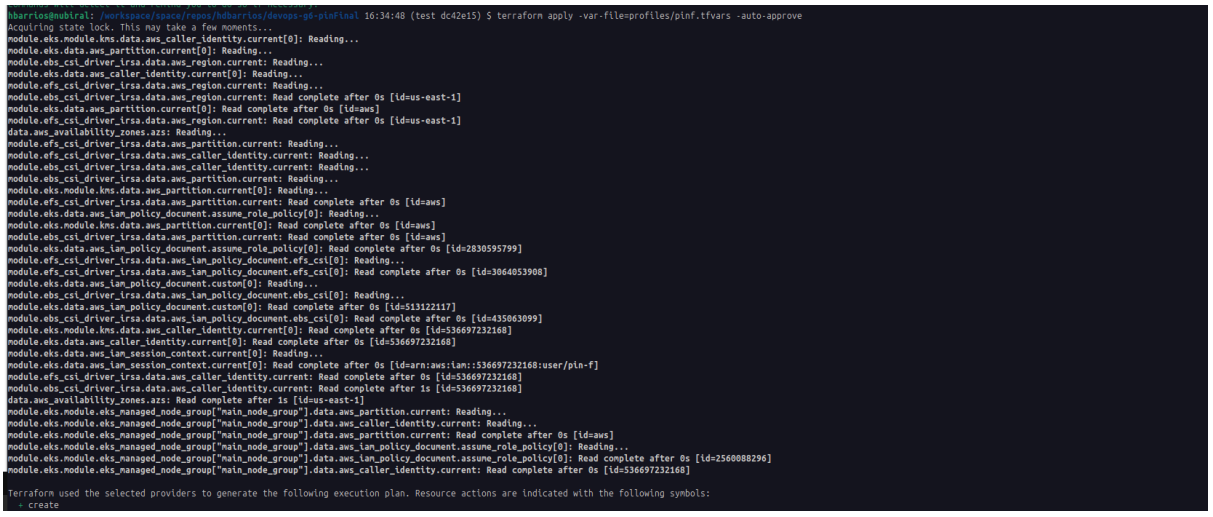
3. Aplicar la configuración (Ejecutar cambios)

Si todo está bien con el plan y se puede aplicar los cambios, ejecuta:

```
terraform apply -var-file=profiles/pinf.tfvarss -auto-approve
```

Terraform pedirá confirmación antes de proceder. Al estar seguro de que los cambios son correctos, se escribe `yes` *para confirmar. con el flag* `-auto-approve` *se evita la interacción, mismo proceso que se usa en github-actions.*

```
- name: Terraform apply
  if: github.event_name == 'push'  # Solo en push a master (es decir, cuando el PR se mergea)
  working-directory: terraform-pinf
  run: |
    terraform apply -var-file=profiles/pinf.tfvars -auto-approve -input=false
```



…

Nota: en la instancia, para validar la aplicación del provider.sh, basta con hacer un cat /var/log/cloud-init-output.log

…

MundosE - *https://mundose.com/index.html*

```
Apply complete! Resources: 95 added, 0 changed, 0 destroyed.

Outputs:

account_id = "536697232168"
eks_access_entries = {
  "admin_role" = {
    "access_entry_arn" = "arn:aws:eks:us-east-1:536697232168:access-entry/mundos-e/role/536697232168/ec2_admin_role_cicd/4ccab3ec-f30c-4ed5-8470-e14b58b3141d"
    "cluster_name" = "mundos-e"
    "created_at" = "2025-03-05T19:48:09Z"
    "id" = "mundos-e:arn:aws:iam::536697232168:role/ec2_admin_role_cicd"
    "kubernetes_groups" = toset([])
    "modified_at" = "2025-03-05T19:48:09Z"
    "principal_arn" = "arn:aws:iam::536697232168:role/ec2_admin_role_cicd"
    "tags" = tomap({
      "Name" = "mundos-e"
      "environment" = "PRD"
      "equipo" = "Grupo6"
      "proyecto" = "PINF"
      "repositorio" = "git@github.com:hdbarrios/devops-g6-pin-final.git"
    })
    "tags_all" = tomap({
      "Name" = "mundos-e"
      "environment" = "PRD"
      "equipo" = "Grupo6"
      "proyecto" = "PINF"
      "repositorio" = "git@github.com:hdbarrios/devops-g6-pin-final.git"
    })
    "timeouts" = null /* object */
    "type" = "STANDARD"
    "user_name" = "arn:aws:sts::536697232168:assumed-role/ec2_admin_role_cicd/{{SessionName}}"
  }
  "cluster_creator" = {
    "access_entry_arn" = "arn:aws:eks:us-east-1:536697232168:access-entry/mundos-e/user/536697232168/pin-f/eccab3ec-f30a-7ac5-dc48-eb3e685bdb80"
    "cluster_name" = "mundos-e"
    "created_at" = "2025-03-05T19:48:08Z"
    "id" = "mundos-e:arn:aws:iam::536697232168:user/pin-f"
    "kubernetes_groups" = toset([])
    "modified_at" = "2025-03-05T19:48:08Z"
    "principal_arn" = "arn:aws:iam::536697232168:user/pin-f"
    "tags" = tomap({
      "Name" = "mundos-e"
      "environment" = "PRD"
      "equipo" = "Grupo6"
      "proyecto" = "PINF"
      "repositorio" = "git@github.com:hdbarrios/devops-g6-pin-final.git"
    })
    "tags_all" = tomap({
      "Name" = "mundos-e"
      "environment" = "PRD"
      "equipo" = "Grupo6"
      "proyecto" = "PINF"
      "repositorio" = "git@github.com:hdbarrios/devops-g6-pin-final.git"
    })
    "timeouts" = null /* object */
    "type" = "STANDARD"
    "user_name" = "arn:aws:iam::536697232168:user/pin-f"
  }
  "programmatic_user" = {
    "access_entry_arn" = "arn:aws:eks:us-east-1:536697232168:access-entry/mundos-e/user/536697232168/ec2_admin_role_cicd/00cab3ec-f308-ac05-383f-b134d50c0272"
    "cluster_name" = "mundos-e"
    "created_at" = "2025-03-05T19:48:08Z"
    "id" = "mundos-e:arn:aws:iam::536697232168:user/ec2_admin_role_cicd"
    "kubernetes_groups" = toset([])
    "modified_at" = "2025-03-05T19:48:08Z"
    "principal_arn" = "arn:aws:iam::536697232168:user/ec2_admin_role_cicd"
    "tags" = tomap({
      "Name" = "mundos-e"
      "environment" = "PRD"
      "equipo" = "Grupo6"
      "proyecto" = "PINF"
      "repositorio" = "git@github.com:hdbarrios/devops-g6-pin-final.git"
    })
    "tags_all" = tomap({
      "Name" = "mundos-e"
      "environment" = "PRD"
      "equipo" = "Grupo6"
      "proyecto" = "PINF"
      "repositorio" = "git@github.com:hdbarrios/devops-g6-pin-final.git"
    })
    "timeouts" = null /* object */
    "type" = "STANDARD"
    "user_name" = "arn:aws:iam::536697232168:user/ec2_admin_role_cicd"
  }
}
eks_cluster_certificate_authority_data = "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJUlp6TEN5aE5icjh3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1Wlh5bGN6QWVGd
QxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHUGliM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUURQQWNqZFNCNkJCc2laUlFHQWZ6WGVuYjE5OWNTT1R0ZTFoNUg2akpSL2NJSC9uQmIwZk1yZHFNZy8KMFVUNEZPU3FCQnJkVStEaWk1OFJ6RlJUDd1VBQ
lMNndsOUVSVFhLNW52M3hvZzpNpRnlOSzIxZlBTTmhMVHk2YklncG5jCnRNTXA2Um9tOEZrNS91VURzbXpSZlA3VjF3KzAvOC9GN0RiZmNzRDZKeFNpaTZ0RjFSRzNZaHQ3aVhxNElodDUKQ3dyMjJYRm92WWkwcWJTaGo1NG9kMU1oVWN6T0xEbFhORXJ1MkJ0
FBR2pXVEJYTUE0R0ExVWREd0VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJRQUg5Z2ZRsV2tEdHlxQ2hsR0M5QklkcFRQeFlEQVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDd1VB
VkZDRGbmlubDRpelRvWmp6b1VUNndWCnl3MnpiajVCa0l0VVQ2ZGkSMkRzMGp5LzNPS2ZYMHJxaVFFK2xkR21ETFBuNWE0RkVham1sWk15RXRZM2kyUysKd25FN01NVDFhRytWU2V1MS9DL3dOVXFFL0VWcUdHR2RCZFNzemhtWC9GNSszTjI1NlkyKzY0U0Js
9qbk5haENZK3JueWtSClIwZ3FlcXIwSjZ0b0lZU0ZtR0IvYmkvL1R4N0wxeUgwSkJXdDJPait5VWxsbmhrSFA1SHlOWnNWaXdaNm5pTFAKdGVSQjNMWmREMUdmCi0tLS0tRU5EIENFUlRJRklDQVRFL50tLS0K"
eks_cluster_endpoint = "https://22336D28003CEAE2C70387AA364EC7AA.gr7.us-east-1.eks.amazonaws.com"
eks_cluster_name = "mundos-e"
instance_id = "i-0df37b5eedc09e523"
instance_ip = "18.212.94.255"
instance_key_name = "pin"
instance_private_dns = "ip-10-11-1-200.ec2.internal"
instance_private_ip = "10.11.1.200"
instance_public_dns = "ec2-18-212-94-255.compute-1.amazonaws.com"
instance_state = "running"
instance_type = "t2.micro"
internet_gateway_id = "igw-08d36c15c7ef0498a"
programmatic_user_access_key = <sensitive>
programmatic_user_name = "ec2_admin_role_cicd"
route_table_id = "rtb-072fe5b2a33a64f66"
security_group_id = "sg-0847f6c76e18cb38b"
subnet_id = "subnet-07b6ae6bd85b57f7f"
vpc_id = "vpc-03d00d484b1cd1678"
hbarrios@nubiral: /workspace/space/repos/hdbarrios/devops-g6-pinFinal 16:51:38 (test dc42e15) $
```

4. Ver los resultados de la aplicación

Después de aplicar la configuración, Terraform mostrará la salida definida en tu archivo `outputs.tf`, si has configurado algún bloque `output`.

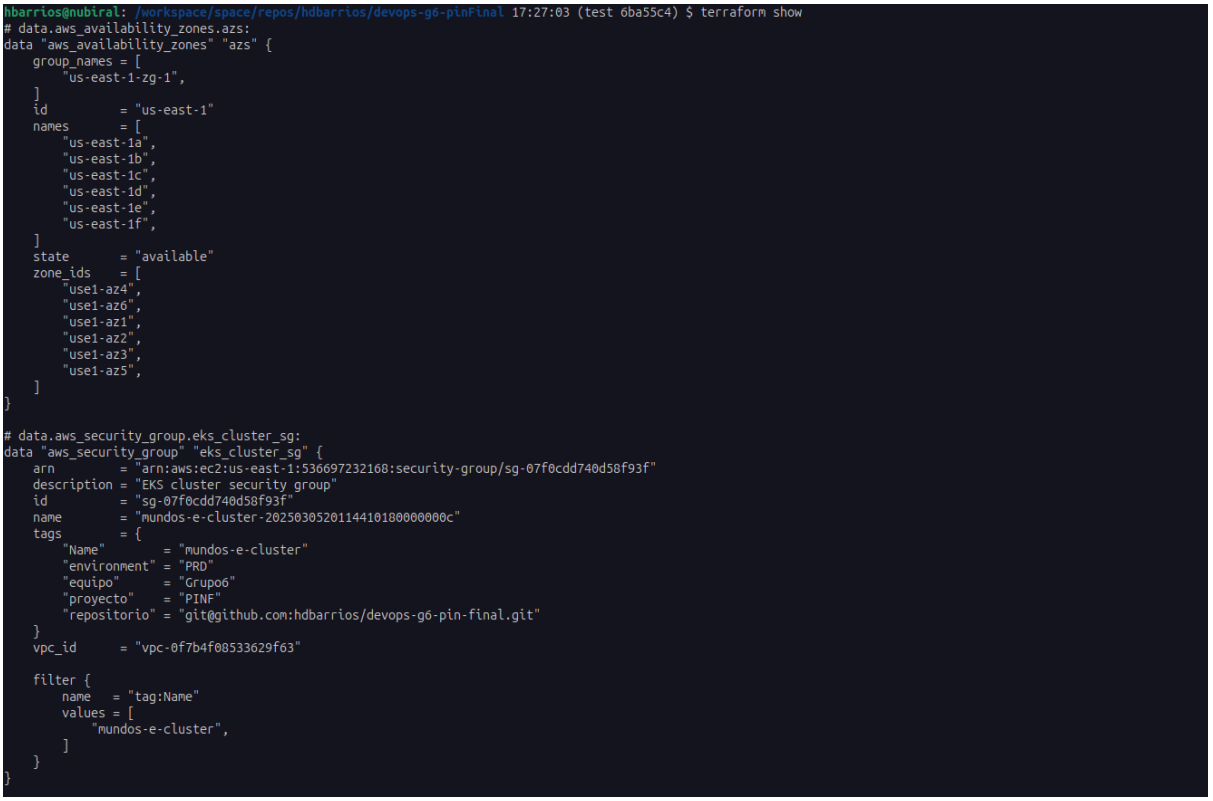
Para obtener información adicional sobre los recursos creados (por ejemplo, la dirección IP pública de una instancia EC2), se puede usar:

```
terraform refresh -var-file=profiles/pinf.tfvars
```

5. Comprobar el estado actual de Terraform

Si se requiere ver el estado actual de la infraestructura gestionada por Terraform, ejecutar:

```
terraform show
```



Salida en yml

(

instalación:

```
export VERSION=v4.2.0 && export BINARY=yq_linux_amd64 && wget https://github.com/mikefarah/yq/releases/download/${VERSION}/${BINARY}.tar.gz -O - | tar xz && sudo mv ${BINARY} /usr/bin/yq
```

)

```
terraform show -json | jq . | yq eval -P
```

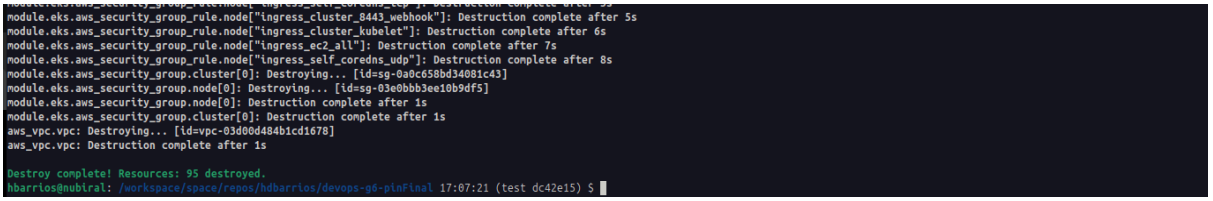
6. Destruir la infraestructura (opcional)

Si se necesita destruir todos los recursos que has creado (por ejemplo, para probar la limpieza), puedes usar:

```
terraform destroy -var-file=profiles/pinf.tfvarss
```

Terraform solicitara confirmación. Escribe `yes` para proceder.

> *Terraform pedirá confirmación antes de proceder. Al estar seguro de que los cambios son correctos, se escribe `yes` para confirmar. Con el flag `-auto-approve` se evita la interacción, mismo proceso que se usa en github-actions.*




hdbarrios / devops-g6-pin-final
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Terraform Apply
Merge pull request #19 from hdbarrios/test #51
Re-run all jobs
Summary
Jobs
terraform_apply
validacion_infra
Run details
Usage
Workflow file
validacion_infra
succeeded 8 minutes ago in 2m 4s
Search logs
Set up job
Run actions/checkout@v4
Install kubectl
Install Helm
Install AWS CLI
Verificando el cluster EKS
Port-forward prometheus-service
Post Run actions/checkout@v4
Complete job

Mundos