

PIN DEVOPS 2401

Repositorio - <https://github.com/palacioea/Pin-2024>

Grupo 01
Daniel Bertoni
Emiliano Palacio

Contenido

Enfoque de la Solución.....	2
Creación de la Instancia EC2 en AWS	2
Acceso SSH a la Instancia.....	4
Instalación de Aplicaciones	5
Creación del Cluster EKS.....	5
Configuración de kubectl.....	7
Instalación del EBS Driver.....	8
Despliegue de Nginx	9
Configuración de Helm y Despliegue de Prometheus.....	10
Instalación de Grafana	10
Limpieza de Recursos	13

Enfoque de la Solución

Vamos a desplegar el EC2 desde un pc de salto vía Terraform, además desplegaremos todas las herramientas que se solicitan para poder montar el EKS con un script en Shell para automatizar el despliegue.

Una vez implementado el EC2, procederemos a levantar el EKS desde línea de comandos desde el bastión previamente desplegado.

Terminada todas las capturas de pantalla, se procederá a borrar el EKS desde el bastion y luego con Terraform se destruirá el bastión construido.

Además, utilizaremos una herramienta Cloud-Nuke para poder eliminar también todos los demás recursos que se hayan creado si fuese necesario por otras vías.

Creación de la Instancia EC2 en AWS

Utilizamos Terraform para aprovisionar una instancia EC2 en AWS. En el proceso, se generaron 16 recursos, incluyendo la creación del par de claves (KEY_PAIR) necesario para establecer la conexión SSH. Verificamos exitosamente la creación de la instancia a través de la consola de AWS.

```
+ tenancy                = (known after apply)
+ timeouts               = null
+ user_data              = (known after apply)
+ user_data_base64       = (known after apply)
+ user_data_replace_on_change = false
+ volume_tags            = null
+ vpc_security_group_ids = (known after apply)
}
tls_private_key.devops_key: Creating...
aws_iam_policy.my_policy: Creating...
aws_vpc.main: Creating...
aws_iam_role.my_role: Creating...
tls_private_key.devops_key: Creation complete after 2s [id=82cdcd39434b8e3871752978a6751adee99d5001]
aws_key_pair.devops: Creating...
local_file.private_key: Creating...
local_file.private_key: Creation complete after 0s [id=c6d3abd8dc6da41fe6bfc753e07948e707f4df12]
null_resource.set_permissions: Creating...
null_resource.set_permissions (local-exec): Provisioning with 'local-exec'...
null_resource.set_permissions (local-exec): Executing: ["/bin/sh" "-c" "chmod 400 ./devops.pem"]
null_resource.set_permissions: Creation complete after 0s [id=635613157551006064]
aws_iam_policy.my_policy: Creation complete after 1s [id=arn:aws:iam::654654500943:policy/MyEC2Policy]
aws_iam_role.my_role: Creation complete after 1s [id=role_admin]
aws_iam_role_policy_attachment.ebs_csi_driver_attachment: Creating...
aws_iam_role_policy_attachment.my_attachment: Creating...
aws_iam_instance_profile.ec2_profile: Creating...
aws_key_pair.devops: Creation complete after 0s [id=devops]
aws_iam_role_policy_attachment.ebs_csi_driver_attachment: Creation complete after 1s [id=role_admin-20241013025325456000000001]
aws_iam_role_policy_attachment.my_attachment: Creation complete after 1s [id=role_admin-202410130253254774000000002]
aws_iam_instance_profile.ec2_profile: Creation complete after 1s [id=ec2_instance_profile]
aws_vpc.main: Creation complete after 3s [id=vpc-0feb3810ed5a6155f]
aws_internet_gateway.main: Creating...
aws_subnet.main: Creating...
aws_security_group.allow_ssh: Creating...
aws_subnet.main: Creation complete after 1s [id=subnet-096b4c61d934cca0b]
aws_internet_gateway.main: Creation complete after 1s [id=igw-08977330d7267be88]
aws_route_table.main: Creating...
aws_security_group.allow_ssh: Creation complete after 3s [id=sg-043da7f3a8768a1e4]
aws_route_table.main: Creation complete after 2s [id=rtb-0d67579d3b66d4d64]
aws_instance.mi_instancia: Creating...
aws_route_table_association.a: Creating...
aws_route_table_association.a: Creation complete after 1s [id=rtbassoc-040f8deb9b595d01]
aws_instance.mi_instancia: Still creating... [10s elapsed]
aws_instance.mi_instancia: Creation complete after 17s [id=i-084f4a5b67454c87b]

Apply complete! Resources: 16 added, 0 changed, 0 destroyed.
```

```

# Generar una nueva clave privada
resource "tls_private_key" "devops_key" {
  algorithm = "RSA"
  rsa_bits  = 4096
}

# Crear un Key Pair en AWS usando la clave generada
resource "aws_key_pair" "devops" {
  key_name   = "devops"
  public_key = tls_private_key.devops_key.public_key_openssh
}

# Opcional: Guardar la clave privada en un archivo local
resource "local_file" "private_key" {
  content  = tls_private_key.devops_key.private_key_pem
  filename = "${path.module}/devops.pem"
}

resource "null_resource" "set_permissions" {
  depends_on = [local_file.private_key]

  provisioner "local-exec" {
    command = "chmod 400 ${local_file.private_key.filename}"
  }
}

```

DevOps-Project I-084f4a5b67454c87b En ejecución t2.micro 2/2 comprobador Ver alarmas + us-east-1c 98.81.156.45 disabled allow_ssh de

I-084f4a5b67454c87b (DevOps-Project)

Detalles Estado y alarmas Monitoreo Seguridad Redes Almacenamiento Etiquetas

Resumen de instancia Información

ID de la instancia
I-084f4a5b67454c87b (DevOps-Project)

Dirección IPv4
-

Tipo de nombre de anfitrión
Nombre de IP: ip-10-0-1-82.ec2.internal

Responder al nombre DNS de recurso privado
-

Dirección IP asignada automáticamente
98.81.156.45 [P pública]

Ro de IAM
role_admin

IMDSv2
Optional
EC2 recomienda configurar IMDSv2 como obligatorio Más información

Detalles de la instancia Información

Plataforma
Ubuntu (Inferido)

Detalles de la plataforma
Linux/UNIX

Detener la protección
desactivado

Recuperación automática de instancias
Predeterminada

Indice de lanzamiento de AMI
0

Especificación de crédito

Dirección IPv4 pública
98.81.156.45 | dirección abierta

Estado de la instancia
En ejecución

Nombre DNS de IP privada (solo IPv4)
ip-10-0-1-82.ec2.internal

Tipo de instancia
t2.micro

ID de VPC
vpc-9fab3810cd5a6155f (main-vpc)

ID de subred
subnet-696b4d81d934cca0b (main-subnet)

ARN de instancia
arn:aws-ec2:us-east-1:854654300943:instance/I-084f4a5b67454c87b

ID de AMI
ami-005fc0f236362e09f

Nombre de AMI
ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20240927

Hora de lanzamiento
Sat Oct 12 2024 23:53:53 GMT-0300 (Argentina Standard Time) (14 minutes)

Ciclo de vida
normal

Par de claves asignado en el lanzamiento
devops

ID de kernel

Direcciones IPv4 privadas
10.0.1.82

DNS de IPv4 pública
-

Direcciones IP elásticas
-

Hallazgo de AWS Compute Optimizer
Suscribirse a AWS Compute Optimizer para recibir recomendaciones | Más información

Nombre del grupo de Auto Scaling
-

Monitores
desactivado

Protección de terminación
desactivado

Ubicación de AMI
amazon/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20240927

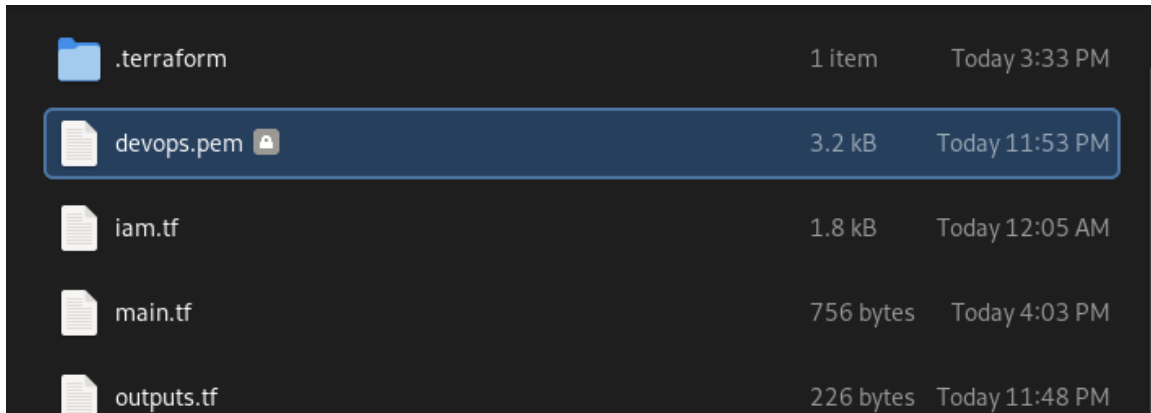
Comportamiento de detención de hibernación
desactivado

Motivo de transición de estado
-

Mensaje de transición de estado

Acceso SSH a la Instancia

La clave PEM se descargó oportunamente, se logró la conexión SSH a la instancia de forma exitosa.



```
ubuntu@ip-10-0-1-82: ~  
mhoes@local ..ia/Hdd/Containers/proyectos/Definitivo % ssh -i "devops.pem" ubuntu@98.81.156.45  
The authenticity of host '98.81.156.45 (98.81.156.45)' can't be established.  
ED25519 key fingerprint is SHA256:5dRz3AY4STaqrDaIampZ0VZpHTU3ctBxKe7/0hKFz3yI.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '98.81.156.45' (ED25519) to the list of known hosts.  
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1015-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/pro  
  
System information as of Sun Oct 13 03:09:59 UTC 2024  
  
System load:  0.0          Processes:            102  
Usage of /:   21.1% of 7.57GB Users logged in:      0  
Memory usage: 22%         IPv4 address for eth0: 10.0.1.82  
Swap usage:   0%  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-10-0-1-82:~$
```

Instalación de Aplicaciones

Se creó un script que permite la instalación de todas las aplicaciones requeridas que se aprovisiono directamente desde Terraform.

```
main.tf > resource "aws_iam_instance_profile" "ec2_profile" > role
1  # Crear el perfil de instancia IAM
2  resource "aws_iam_instance_profile" "ec2_profile" {
3      name = "ec2_instance_profile"
4      role = aws_iam_role.my_role.name
5  }
6
7  # Crear la instancia EC2
8  resource "aws_instance" "mi_instancia" {
9      ami = "ami-005fc0f236362e99f"
10     instance_type = var.ec2_instance_type
11     iam_instance_profile = aws_iam_instance_profile.ec2_profile.name
12     subnet_id = aws_subnet.main.id
13     key_name = aws_key_pair.devops.key_name
14     vpc_security_group_ids = [aws_security_group.allow_ssh.id]
15     associate_public_ip_address = true
16     user_data = file("script/apps_install.sh")
17
18     tags = {
19         Name = var.ec2_name
20     }
21 }
```

Creación del Cluster EKS

Iniciamos el proceso de creación del cluster EKS utilizando el siguiente comando:

```
eksctl create cluster \
--name eks-mundos-e \
--region us-east-1 \
--with-oidc \
--nodegroup-name testv3 \
--node-type t2.large \
--nodes 1 \
--nodes-min 1 \
--nodes-max 3 \
--node-volume-size 20 \
--ssh-access=false \
--managed \
--asg-access \
--external-dns-access \
--full-ecr-access \
--appmesh-access \
--alb-ingress-access
```

```

ubuntu@ip-10-0-1-33:~$ eksctl create cluster \
--name eks-mundos-e \
--region us-east-1 \
--with-oidc \
--nodegroup-name testv3 \
--node-type t2.large \
--nodes 1 \
--nodes-min 1 \
--nodes-max 3 \
--node-volume-size 20 \
--ssh-access=false \
--managed \
--asg-access \
--external-dns-access \
--full-ecr-access \
--appmesh-access \
--alb-ingress-access
2024-10-13 03:33:27 [i] eksctl version 0.192.0
2024-10-13 03:33:27 [i] using region us-east-1
2024-10-13 03:33:27 [i] setting availability zones to [us-east-1c us-east-1b]
2024-10-13 03:33:27 [i] subnets for us-east-1c - public:192.168.0.0/19 private:192.168.64.0/19
2024-10-13 03:33:27 [i] subnets for us-east-1b - public:192.168.32.0/19 private:192.168.96.0/19
2024-10-13 03:33:27 [i] nodegroup "testv3" will use "" [AmazonLinux2/1.30]
2024-10-13 03:33:27 [i] using Kubernetes version 1.30
2024-10-13 03:33:27 [i] creating EKS cluster "eks-mundos-e" in "us-east-1" region with managed nodes
2024-10-13 03:33:27 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2024-10-13 03:33:27 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stack --region=us-east-1 --cluster=eks-mundos-e'
2024-10-13 03:33:27 [i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "eks-mundos-e" in "us-east-1"
2024-10-13 03:33:27 [i] CloudWatch logging will not be enabled for cluster "eks-mundos-e" in "us-east-1"
2024-10-13 03:33:27 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=eks-mundos-e'
2024-10-13 03:33:27 [i] default addons vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2024-10-13 03:33:27 [i]
2 sequential tasks: { create cluster control plane "eks-mundos-e",
  2 sequential sub-tasks: {
    5 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
      associate IAM OIDC provider,
      no tasks,
      update VPC CNI to use IRSA if required,
    },
    create managed nodegroup "testv3",
  },
}
2024-10-13 03:33:27 [i] building cluster stack "eksctl-eks-mundos-e-cluster"
2024-10-13 03:33:28 [i] deploying stack "eksctl-eks-mundos-e-cluster"

```

```

2024-10-13 03:40:29 [i] creating addon
2024-10-13 03:40:29 [i] successfully created addon
2024-10-13 03:40:30 [i] creating addon
2024-10-13 03:40:30 [i] successfully created addon
2024-10-13 03:40:30 [i] creating addon
2024-10-13 03:40:31 [i] successfully created addon
2024-10-13 03:42:32 [i] deploying stack "eksctl-eks-mundos-e-addon-vpc-cni"
2024-10-13 03:42:32 [i] waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-vpc-cni"
2024-10-13 03:43:02 [i] waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-vpc-cni"
2024-10-13 03:43:02 [i] updating addon
2024-10-13 03:43:12 [i] addon "vpc-cni" active
2024-10-13 03:43:13 [i] building managed nodegroup stack "eksctl-eks-mundos-e-nodegroup-testv3"
2024-10-13 03:43:13 [i] deploying stack "eksctl-eks-mundos-e-nodegroup-testv3"
2024-10-13 03:43:13 [i] waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-testv3"
2024-10-13 03:43:43 [i] waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-testv3"
2024-10-13 03:44:25 [i] waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-testv3"
2024-10-13 03:45:55 [i] waiting for CloudFormation stack "eksctl-eks-mundos-e-nodegroup-testv3"
2024-10-13 03:45:55 [i] waiting for the control plane to become ready
2024-10-13 03:45:56 [✓] saved kubeconfig as "/home/ubuntu/.kube/config"
2024-10-13 03:45:56 [i] no tasks
2024-10-13 03:45:56 [✓] all EKS cluster resources for "eks-mundos-e" have been created
2024-10-13 03:45:56 [✓] created 0 nodegroup(s) in cluster "eks-mundos-e"
2024-10-13 03:45:56 [i] nodegroup "testv3" has 1 node(s)
2024-10-13 03:45:56 [i] node "ip-192-168-9-221.ec2.internal" is ready
2024-10-13 03:45:56 [i] waiting for at least 1 node(s) to become ready in "testv3"
2024-10-13 03:45:56 [i] nodegroup "testv3" has 1 node(s)
2024-10-13 03:45:56 [i] node "ip-192-168-9-221.ec2.internal" is ready
2024-10-13 03:45:56 [✓] created 1 managed nodegroup(s) in cluster "eks-mundos-e"
2024-10-13 03:45:57 [i] kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
2024-10-13 03:45:57 [✓] EKS cluster "eks-mundos-e" in "us-east-1" region is ready
ubuntu@ip-10-0-1-33:~$

```

Configuración de kubectl

Configuramos `kubectl` para interactuar con el cluster EKS recién creado utilizando el siguiente comando:

```
aws eks update-kubeconfig --name eks-mundos-e --region us-east-1
```

```
2024-10-13 03:45:56 [✓] created 1 managed nodegroup(s) in cluster "eks-mundos-e"
2024-10-13 03:45:57 [i] kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
2024-10-13 03:45:57 [✓] EKS cluster "eks-mundos-e" in "us-east-1" region is ready
ubuntu@ip-10-0-1-33:~$ aws eks update-kubeconfig --name eks-mundos-e --region us-east-1
Added new context arn:aws:eks:us-east-1:654654500943:cluster/eks-mundos-e to /home/ubuntu/.kube/config
ubuntu@ip-10-0-1-33:~$
```

Verificación de la conexión con el cluster

```
2024-10-13 03:45:58 [i] node ip-192-168-9-221.ec2.internal is ready
2024-10-13 03:45:56 [✓] created 1 managed nodegroup(s) in cluster "eks-mundos-e"
2024-10-13 03:45:57 [i] kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
2024-10-13 03:45:57 [✓] EKS cluster "eks-mundos-e" in "us-east-1" region is ready
ubuntu@ip-10-0-1-33:~$ aws eks update-kubeconfig --name eks-mundos-e --region us-east-1
Added new context arn:aws:eks:us-east-1:654654500943:cluster/eks-mundos-e to /home/ubuntu/.kube/config
ubuntu@ip-10-0-1-33:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE      VERSION
ip-192-168-9-221.ec2.internal      Ready    <none>    3m42s    v1.30.4-eks-a737599
ubuntu@ip-10-0-1-33:~$
```


Instalación del EBS Driver

Para habilitar el soporte de volúmenes EBS, aplicamos el controlador EBS driver con los siguientes comandos:

```
kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-1.35"
eksctl create iamserviceaccount \
  --name ebs-csi-controller-sa \
  --region us-east-1 \
  --namespace kube-system \
  --cluster eks-mundos-e \
  --attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \
  --approve \
  --role-only \
  --role-name AmazonEKS_EBS_CSI_DriverRole
eksctl create addon \
  --name aws-ebs-csi-driver
  --cluster eks-mundos-e
  --service-account-role-arn arn:aws:iam::xxxxxxx:role/AmazonEKS_EBS_CSI_DriverRole
  --force
```

```
ubuntu@ip-10-0-1-33:~$ kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-1.35"
serviceaccount/ebs-csi-controller-sa created
serviceaccount/ebs-csi-node-sa created
role.rbac.authorization.k8s.io/ebs-csi-leases-role created
clusterrole.rbac.authorization.k8s.io/ebs-csi-node-role created
clusterrole.rbac.authorization.k8s.io/ebs-external-attacher-role created
clusterrole.rbac.authorization.k8s.io/ebs-external-provisioner-role created
clusterrole.rbac.authorization.k8s.io/ebs-external-resizer-role created
clusterrole.rbac.authorization.k8s.io/ebs-external-snapshotter-role created
rolebinding.rbac.authorization.k8s.io/ebs-csi-leases-rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/ebs-csi-attacher-binding created
clusterrolebinding.rbac.authorization.k8s.io/ebs-csi-node-getter-binding created
clusterrolebinding.rbac.authorization.k8s.io/ebs-csi-provisioner-binding created
clusterrolebinding.rbac.authorization.k8s.io/ebs-csi-resizer-binding created
clusterrolebinding.rbac.authorization.k8s.io/ebs-csi-snapshotter-binding created
deployment.apps/ebs-csi-controller created
poddisruptionbudget.policy/ebs-csi-controller created
daemonset.apps/ebs-csi-node created
csidriver.storage.k8s.io/ebs.csi.aws.com created
ubuntu@ip-10-0-1-33:~$
```

```
ubuntu@ip-10-0-1-33:~$ eksctl create iamserviceaccount \
  --name ebs-csi-controller-sa \
  --region us-east-1 \
  --namespace kube-system \
  --cluster eks-mundos-e \
  --attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \
  --approve \
  --role-only \
  --role-name AmazonEKS_EBS_CSI_DriverRole
2024-10-13 03:59:26 [i] 1 iamserviceaccount (kube-system/ebs-csi-controller-sa) was included (based on the include/exclude rules)
2024-10-13 03:59:26 [!] serviceaccounts in Kubernetes will not be created or modified, since the option --role-only is used
2024-10-13 03:59:26 [i] 1 task: { create IAM role for serviceaccount "kube-system/ebs-csi-controller-sa" }
2024-10-13 03:59:26 [i] building iamserviceaccount stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2024-10-13 03:59:26 [i] deploying stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2024-10-13 03:59:26 [i] waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2024-10-13 03:59:57 [i] waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
ubuntu@ip-10-0-1-33:~$
```

```
ubuntu@ip-10-0-1-33:~$ eksctl create addon --name aws-ebs-csi-driver --region us-east-1 --cluster eks-mundos-e --service-account-role-arn arn:aws:iam::654654500943:role/AmazonEKS_EBS_CSI_DriverRole --force
2024-10-13 04:05:15 [i] Kubernetes version "1.30" in use by cluster "eks-mundos-e"
2024-10-13 04:05:15 [i] IRSA is set for "aws-ebs-csi-driver" addon; will use this to configure IAM permissions
2024-10-13 04:05:15 [!] IRSA has been deprecated; the recommended way to provide IAM permissions for "aws-ebs-csi-driver" addon is via pod identity associations; after addon creation is completed, run `eksctl utils migrate-to-pod-identity`
2024-10-13 04:05:15 [i] using provided ServiceAccountRoleARN "arn:aws:iam::654654500943:role/AmazonEKS_EBS_CSI_DriverRole"
2024-10-13 04:05:15 [i] creating addon
ubuntu@ip-10-0-1-33:~$
```

Despliegue de Nginx

Para desplegar Nginx, aplicamos el manifiesto `nginx.yaml` usando el comando:

```
kubectl apply -f nginx.yaml
```

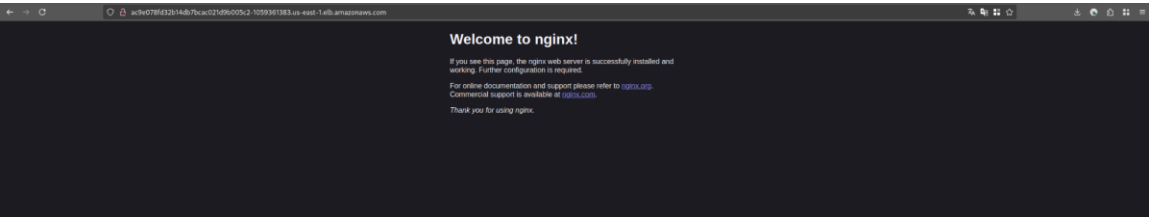
```
ubuntu@ip-10-0-1-33:~$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-576c6b7b6-6w57l 1/1     Running   0           2m9s
pod/nginx-deployment-576c6b7b6-tlcmh 1/1     Running   0           2m9s

NAME                                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/kubernetes                  ClusterIP     10.100.0.1      <none>            443/TCP          22m
service/nginx-service               LoadBalancer 10.100.43.237   ac9e078fd32b14db7bcac021d9b005c2-1059361383.us-east-1.elb.amazonaws.com 80:31797/TCP     2m9s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment    2/2     2             2           2m9s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-576c6b7b6 2         2         2       2m9s
```

Luego, verificamos el acceso a Nginx desde el navegador.



Configuración de Helm y Despliegue de Prometheus

Agregamos los repositorios de Prometheus utilizando Helm:

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
helm repo update
kubectl create namespace prometheus
helm install prometheus prometheus-community/prometheus \
  --namespace prometheus \
  --set alertmanager.persistentVolume.storageClass="gp2" \
  --set server.persistentVolume.storageClass="gp2"
kubectl patch svc prometheus-server -n prometheus -p '{"spec": {"type": "NodePort"}}'
```

```
replicaset.apps/prometheus-server-79cfff87cb      1      1      0      24s
ubuntu@ip-10-0-1-33:~$ kubectl get all -n prometheus
NAME                                     READY   STATUS    RESTARTS   AGE
pod/prometheus-kube-state-metrics-74cdb59bff-b2q2z  1/1     Running   0           80s
pod/prometheus-prometheus-node-exporter-ftkhr      1/1     Running   0           80s
pod/prometheus-prometheus-pushgateway-66fc55f8d-9mL4t  1/1     Running   0           80s
pod/prometheus-server-79cfff87cb-s9tj9            2/2     Running   0           80s

NAME                                     TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/prometheus-kube-state-metrics  ClusterIP     10.100.174.71    <none>            8080/TCP         80s
service/prometheus-prometheus-node-exporter  ClusterIP     10.100.146.48    <none>            9100/TCP         80s
service/prometheus-prometheus-pushgateway  ClusterIP     10.100.247.254   <none>            9091/TCP         80s
service/prometheus-server               ClusterIP     10.100.126.91    <none>            80/TCP           80s

NAME                                     DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/prometheus-prometheus-node-exporter  1          1          1          1             1             kubernetes.io/os=linux  80s

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-kube-state-metrics  1/1      1             1           80s
deployment.apps/prometheus-prometheus-pushgateway  1/1      1             1           80s
deployment.apps/prometheus-server               1/1      1             1           80s

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-kube-state-metrics-74cdb59bff  1          1          1           80s
replicaset.apps/prometheus-prometheus-pushgateway-66fc55f8d  1          1          1           80s
replicaset.apps/prometheus-server-79cfff87cb  1          1          1           80s
```

Instalación de Grafana

Instalamos Grafana utilizando Helm, creando previamente el namespace:

```
kubectl create namespace grafana
helm install grafana grafana/grafana \
  --namespace grafana \
  --set adminPassword='EKS!sAWSome' \
  --values grafana.yaml \
  --set service.type=LoadBalancer
```

```
ubuntu@ip-10-0-1-33:~$ kubectl get all -n grafana
NAME                                     READY   STATUS    RESTARTS   AGE
pod/grafana-77c6fb8d5d-bcvpq           1/1     Running   0           14m

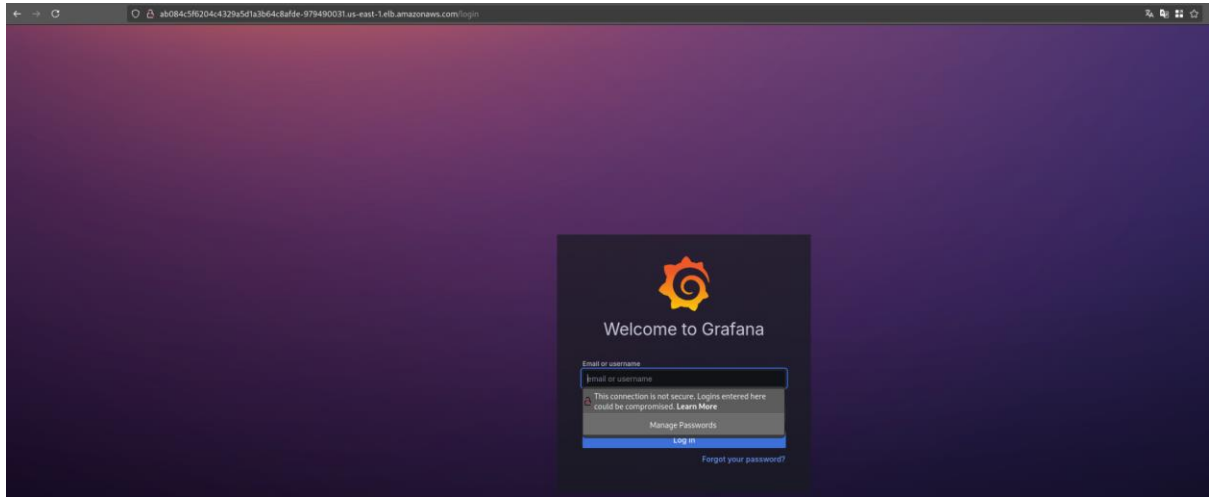
NAME                                     TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/grafana                         LoadBalancer  10.100.61.238    ab084c5f6204c4329a5d1a3b64c8afde-979490031.us-east-1.elb.amazonaws.com  80:31133/TCP  14m

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana                 1/1      1             1           14m

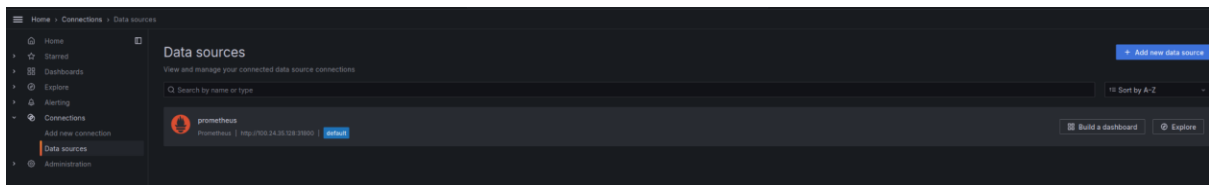
NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana-77c6fb8d5d      1          1          1           14m
```

Monitoreo

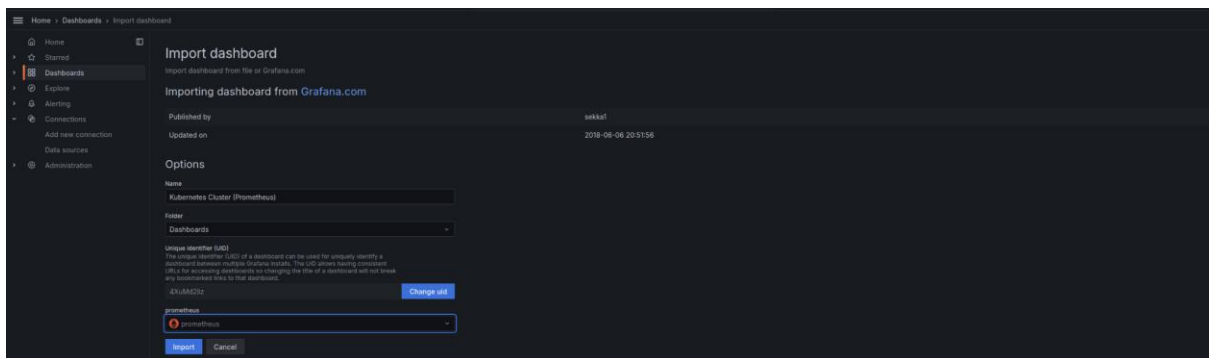
Accedemos al external-ip

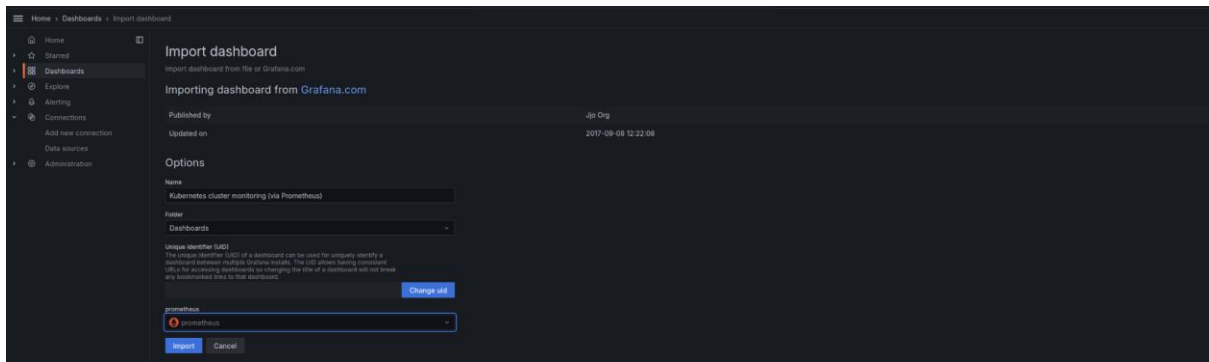


Agregamos Prometheus como datasource:

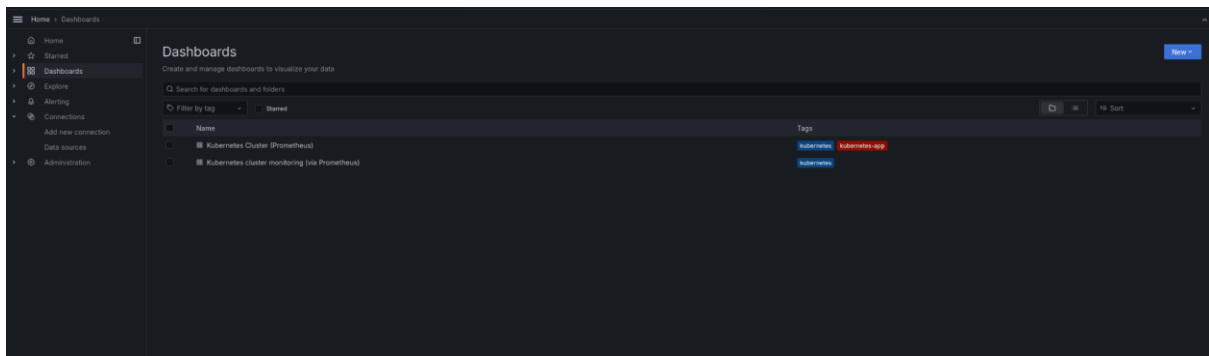


importamos los dashboards 6417 y 3119 en la carpeta Dashboard y configuramos el prometheus que creamos en el datasource.

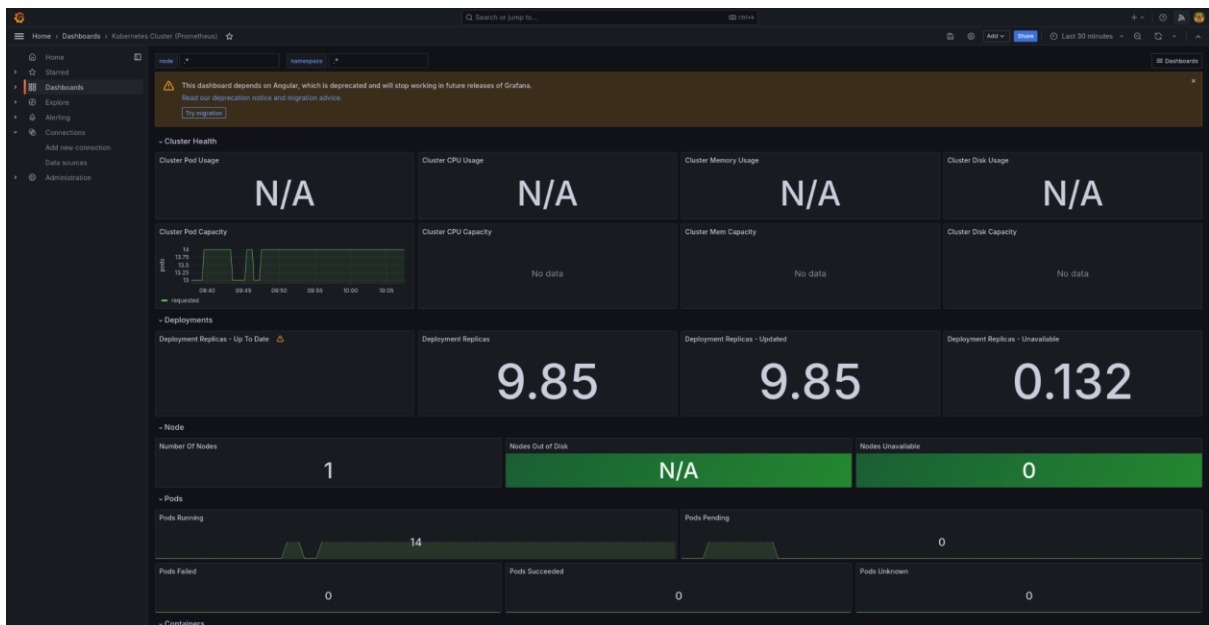




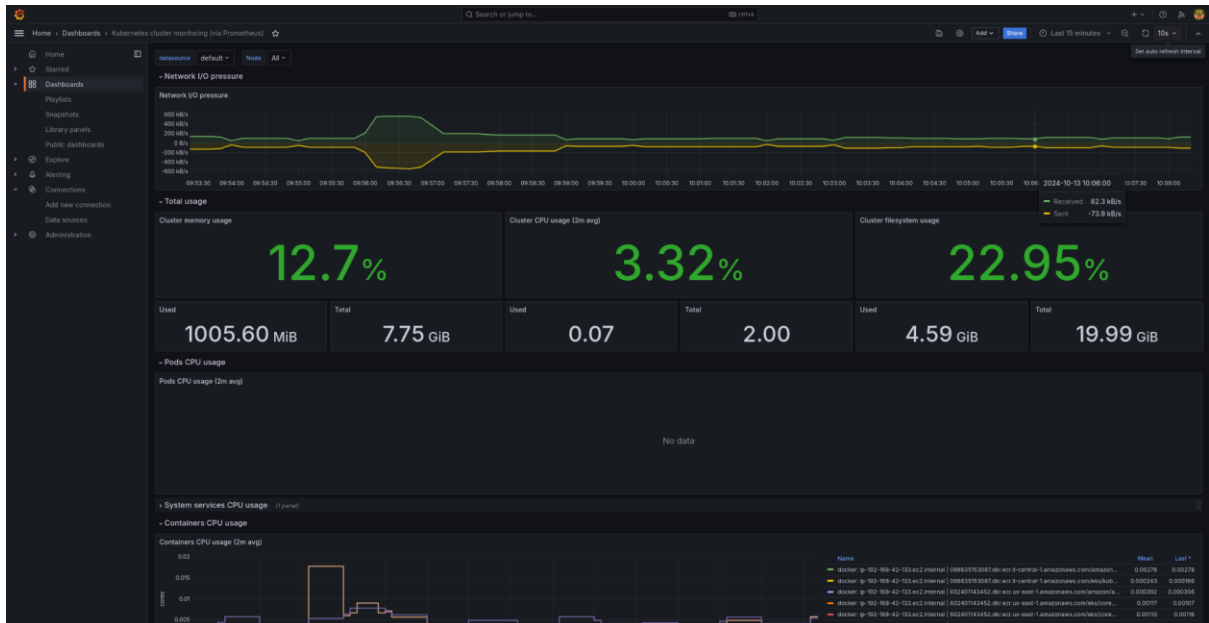
Verificamos importación de ambos



ID 6417:



ID 3119:



Limpieza de Recursos

Procedimos a limpiar todos los recursos desplegados para evitar costos innecesarios.

Desde el bastion (EC2)

```
helm uninstall prometheus --namespace prometheus
kubectl delete ns prometheus
helm uninstall grafana --namespace grafana
eksctl delete cluster --name mundos-e --region us-east-1
```

```
2024-10-13 13:14:27 [i]
4 sequential tasks: { delete nodegroup "mundose",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      delete IAM role for serviceaccount "kube-system/ebs-csi-controller-sa",
      delete serviceaccount "kube-system/ebs-csi-controller-sa",
    },
    delete IAM OIDC provider,
  }, delete addon IAM "eksctl-mundos-e-addon-vpc-cni", delete cluster control plane "mundos-e" [async]
}
2024-10-13 13:14:29 [i] will delete stack "eksctl-mundos-e-nodegroup-mundose"
2024-10-13 13:14:29 [i] waiting for stack "eksctl-mundos-e-nodegroup-mundose" to get deleted
2024-10-13 13:14:29 [i] waiting for CloudFormation stack "eksctl-mundos-e-nodegroup-mundose"
2024-10-13 13:14:59 [i] waiting for CloudFormation stack "eksctl-mundos-e-nodegroup-mundose"
2024-10-13 13:15:33 [i] waiting for CloudFormation stack "eksctl-mundos-e-nodegroup-mundose"
2024-10-13 13:16:37 [i] waiting for CloudFormation stack "eksctl-mundos-e-nodegroup-mundose"
2024-10-13 13:17:12 [i] waiting for CloudFormation stack "eksctl-mundos-e-nodegroup-mundose"
2024-10-13 13:18:23 [i] waiting for CloudFormation stack "eksctl-mundos-e-nodegroup-mundose"
2024-10-13 13:20:12 [i] waiting for CloudFormation stack "eksctl-mundos-e-nodegroup-mundose"
2024-10-13 13:21:03 [i] waiting for CloudFormation stack "eksctl-mundos-e-nodegroup-mundose"
2024-10-13 13:21:50 [i] waiting for CloudFormation stack "eksctl-mundos-e-nodegroup-mundose"
2024-10-13 13:21:51 [i] will delete stack "eksctl-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2024-10-13 13:21:51 [i] waiting for stack "eksctl-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa" to get deleted
2024-10-13 13:21:51 [i] waiting for CloudFormation stack "eksctl-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2024-10-13 13:22:21 [i] waiting for CloudFormation stack "eksctl-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2024-10-13 13:22:21 [i] serviceaccount "kube-system/ebs-csi-controller-sa" was not created by eksctl; will not be deleted
2024-10-13 13:22:21 [i] will delete stack "eksctl-mundos-e-addon-vpc-cni"
2024-10-13 13:22:21 [i] will delete stack "eksctl-mundos-e-cluster"
2024-10-13 13:22:21 [✓] all cluster resources were deleted
ubuntu@ip-10-0-1-33:~$
```

Desde el pc de salto:

```
terraform destroy -auto-approve
```

```
devops@devops:/media/Containers/proyectos/Definitivo$ terraform destroy -auto-approve
tls_private_key.devops_key: Refreshing state... [id=095c942864d7376c16a0104c31ae62c70dc3caa2]
local_file.private_key: Refreshing state... [id=4f3d84b57fea148435a3cd64adfeae2200d13afb]
aws_key_pair.devops: Refreshing state... [id=devops]
aws_iam_policy.my_policy: Refreshing state... [id=arn:aws:iam::654654500943:policy/MyEC2Policy]
aws_vpc.main: Refreshing state... [id=vpc-038f4765978c1869c]
aws_iam_role.my_role: Refreshing state... [id=role_admin]
aws_security_group.allow_ssh: Refreshing state... [id=sg-07b7efb01fe4e1e11]
aws_internet_gateway.main: Refreshing state... [id=igw-02cfd9279b9ef2ef2]
aws_subnet.main: Refreshing state... [id=subnet-091a4249188de41b5]
aws_iam_role_policy_attachment.ebs_csi_driver_attachment: Refreshing state... [id=role_admin-20241013065551885100000001]
aws_iam_role_policy_attachment.my_attachment: Refreshing state... [id=role_admin-20241013065551890400000002]
aws_iam_instance_profile.ec2_profile: Refreshing state... [id=ec2_instance_profile]
aws_route_table.main: Refreshing state... [id=rtb-0a6e8b71f9f310cc4]
aws_instance.mi_instancia: Refreshing state... [id=i-03cb1988d6ae8c8e1]
aws_route_table_association.a: Refreshing state... [id=rtbassoc-099674736b2048040]
```

```
local_file.private_key: Destroying... [id=4f3d84b57fea148435a3cd64adfeae2200d13afb]
local_file.private_key: Destruction complete after 0s
tls_private_key.devops_key: Destroying... [id=095c942864d7376c16a0104c31ae62c70dc3caa2]
tls_private_key.devops_key: Destruction complete after 0s
aws_iam_role_policy_attachment.my_attachment: Destroying... [id=role_admin-20241013065551890400000002]
aws_iam_role_policy_attachment.ebs_csi_driver_attachment: Destroying... [id=role_admin-20241013065551885100000001]
aws_iam_instance_profile.ec2_profile: Destroying... [id=ec2_instance_profile]
aws_iam_role_policy_attachment.my_attachment: Destruction complete after 0s
aws_iam_role_policy_attachment.ebs_csi_driver_attachment: Destruction complete after 0s
aws_iam_policy.my_policy: Destroying... [id=arn:aws:iam::654654500943:policy/MyEC2Policy]
aws_iam_instance_profile.ec2_profile: Destruction complete after 0s
aws_iam_role.my_role: Destroying... [id=role_admin]
aws_iam_policy.my_policy: Destruction complete after 1s
aws_iam_role.my_role: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
```

Anexo

CLOUD-NUKE - Eliminar otros servicios creado a mano.

```
cloud-nuke_linux_amd64 aws --region us-east-1
```

```
devops@devops:/media/Containers/proyectos$ ./cloud-nuke_linux_amd64 aws --region us-east-1
WARNING This program sends telemetry to Gruntwork. To disable, set DISABLE_TELEMETRY=true as an environment variable

# AWS Resource Query Parameters
-----
| Query Parameter | Value |
|-----|-----|
| Target Regions | us-east-1 |
| Target Resource Types | 108 resource types (too many to list all) |
| List Unaliased KMS Keys | false |
|-----|-----|

INFO Found 1 ec2 resources in us-east-1
INFO Found 1 ec2-keypairs resources in us-east-1
INFO Found 4 vpc resources in us-east-1
INFO Found 1 ec2_dhcp_option resources in us-east-1
INFO Found 3 ec2-subnet resources in us-east-1
INFO Found 1 internet-gateway resources in us-east-1
INFO Found 1 network-interface resources in us-east-1
INFO Found 1 security-group resources in us-east-1
INFO Done searching for resources
INFO Found total of 13 resources

# Found AWS Resources
-----
| Resource Type | Region | Identifier | Nukable |
|-----|-----|-----|-----|
| ec2 | us-east-1 | i-03cb1988d6ae8c8e1 | - |
| ec2-keypairs | us-east-1 | key-04c01435777da509e | ✓ |
| vpc | us-east-1 | vpc-0bb2406e0ae5c98f9 | ✓ |
| vpc | us-east-1 | vpc-038f4765978c1869c | ✓ |
| vpc | us-east-1 | vpc-0b99bf3d9ae6768e5 | ✓ |
| vpc | us-east-1 | vpc-049d416eabd615d94 | ✓ |
| ec2_dhcp_option | us-east-1 | dopt-07d8408366deb27f9 | ✓ |
| ec2-subnet | us-east-1 | subnet-0a5813d76e0be588f | ✓ |
| ec2-subnet | us-east-1 | subnet-091a4249188de41b5 | ✓ |
| ec2-subnet | us-east-1 | subnet-0fda860b51e5f98b7 | ✓ |
| internet-gateway | us-east-1 | igw-02cfd9279b9ef2ef2 | ✓ |
| network-interface | us-east-1 | eni-050e638ff370dc31d | ✓ |
| security-group | us-east-1 | sg-07b7efb01fe4e1e11 | ✓ |
|-----|-----|-----|-----|

WARNING THE NEXT STEPS ARE DESTRUCTIVE AND COMPLETELY IRREVERSIBLE, PROCEED WITH CAUTION!!!
Are you sure you want to nuke all listed resources? Enter 'nuke' to confirm (or exit with ^C) :
```

Confirmamos

```
WARNING THE NEXT STEPS ARE DESTRUCTIVE AND COMPLETELY IRREVERSIBLE, PROCEED WITH CAUTION!!!
Are you sure you want to nuke all listed resources? Enter 'nuke' to confirm (or exit with ^C) : nuke
Nuking batch of 1 ec2 resource(s) in us-east-1 [0/13] 0% | 14s

Nuking batch of 1 security-group resource(s) in us-east-1 [13/13] 100% | 1m44s2s
```