DevOps

PIN FINAL

Grupo 5

Emilio Pascutti - emiliopascutti2164@gmail.com

Natalio Cardozo - nathaliocardozo@gmail.com

Julio Gonzalez - aresden113@gmail.com

Jonathan David Martino - jonamartino@gmail.com

Gonzalo Martín Montalvo - montalvog@gmail.com

Objetivos:

Terraform - EKS – AWS – Prometheus – Grafana

El objetivo de este PIN, será desarrollar una Infraestructura como Código (IaaC) de Terraform, para desplegar un clúster de kubernetes en AWS, monitoreado con Prometheus y Grafana.

Flujo de Trabajo

/PINFINAL

├── Creación de cuenta IAM con AmazonEC2FullAccess.

├── Configuración de Terraform para el Clúster EKS.

├── Deploy de Prometheus y Grafana con Helm automático.

├── Verificación de entornos

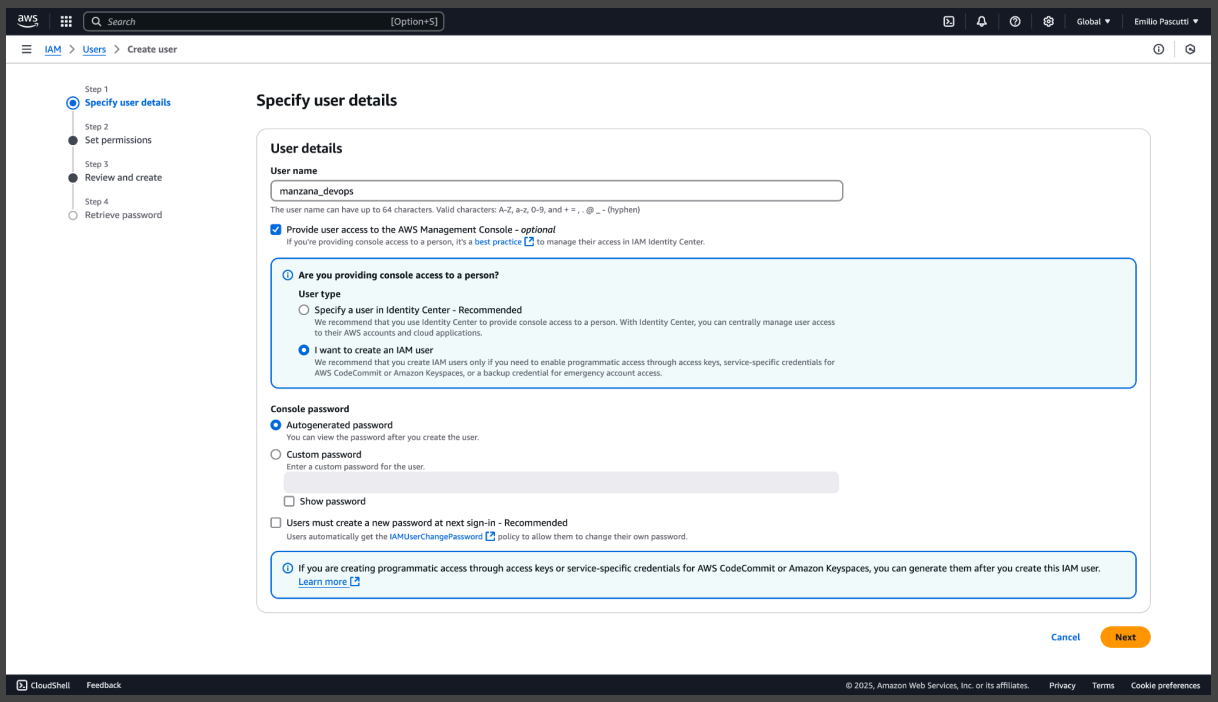
├── Configuración de Prometheus como source de Grafana y configuración de Dashboards.

├── Clean up

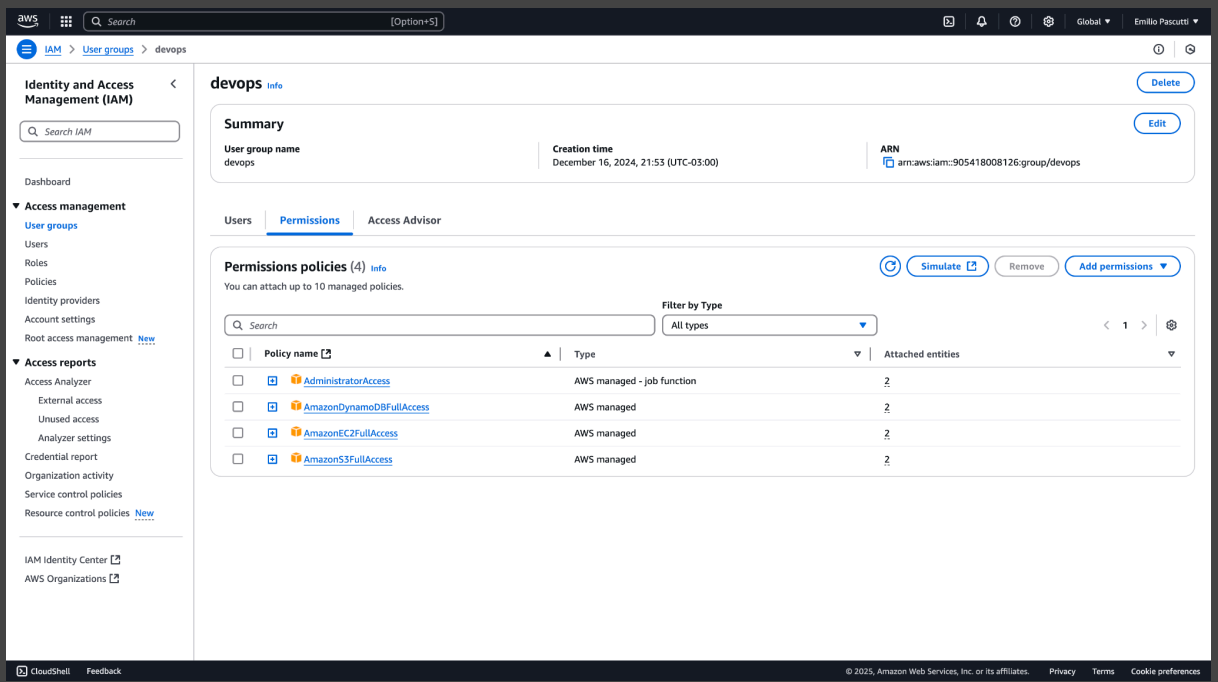
└── Análisis y conclusiones

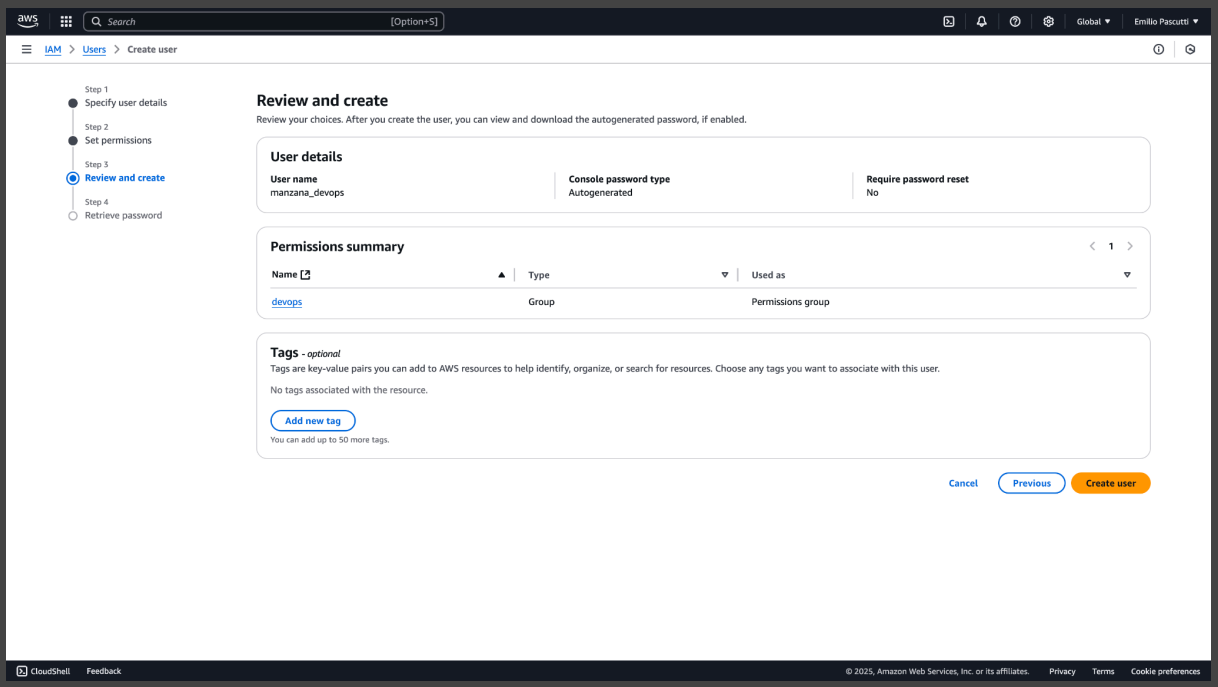
Creación de usuario IAM en AWS para EC2

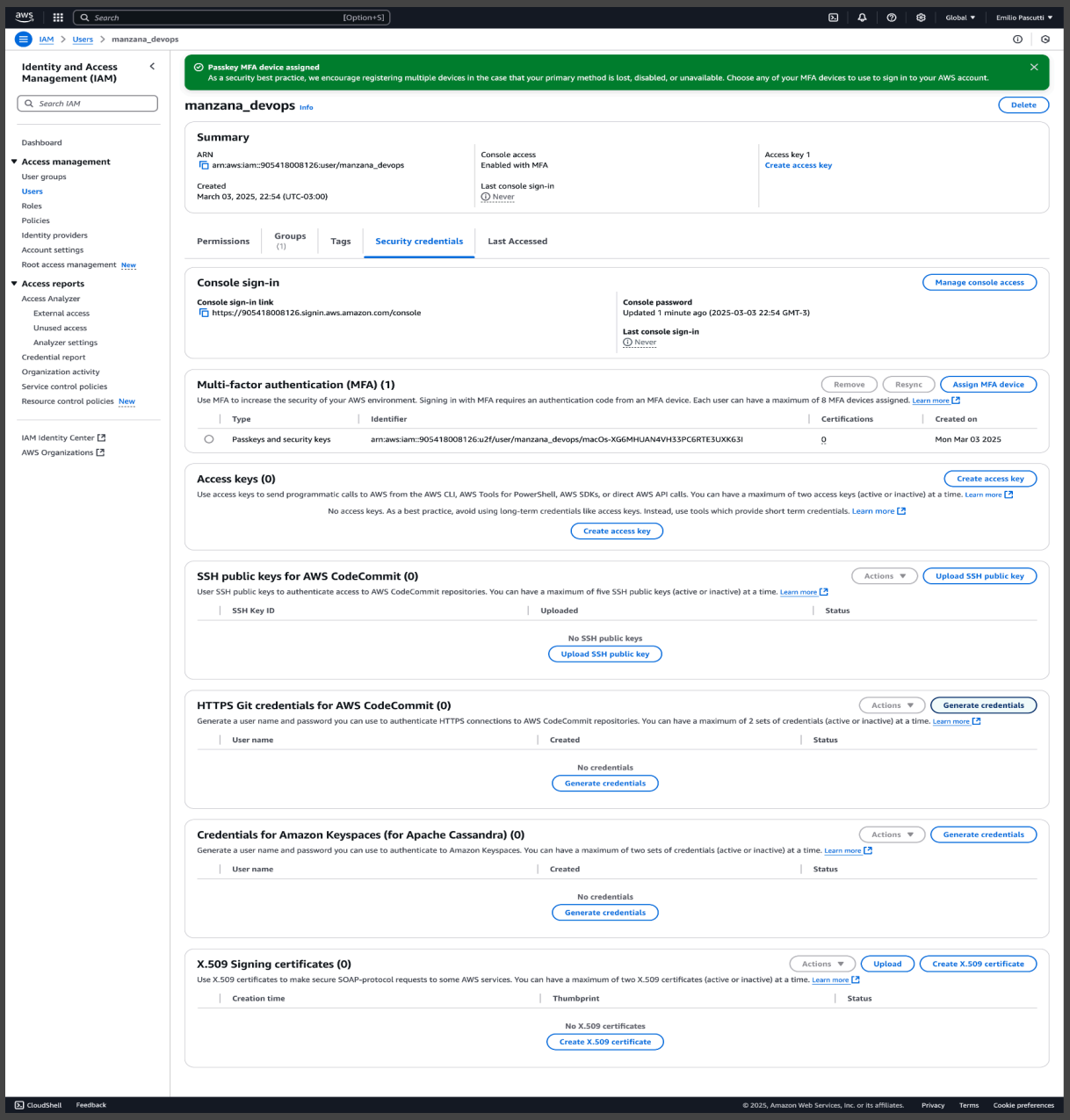
Rol IAM con policy de AmazonEC2FullAccess
Nombre de usuario IAM: manzana_devops



DevOps group:




Search [Option+S]
Global
Emilio Pascutti
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Review and create
Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.
User details
User name
manzana_devops
Console password type
Autogenerated
Require password reset
No
Permissions summary
Name
Type
Used as
devops
Group
Permissions group
Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Previous
Create user
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Option+S]
Global
Emilio Pascutti
IAM > Users > manzana_devops
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
Resource control policies New
IAM Identity Center
AWS Organizations
Passkey MFA device assigned
As a security best practice, we encourage registering multiple devices in the case that your primary method is lost, disabled, or unavailable. Choose any of your MFA devices to use to sign in to your AWS account.
manzana_devops Info
Delete
Summary
ARN
arn:aws:iam::905418008126:user/manzana_devops
Created
March 03, 2025, 22:54 (UTC-03:00)
Console access
Enabled with MFA
Last console sign-in
Never
Access key 1
Create access key
Permissions
Groups (1)
Tags
Security credentials
Last Accessed
Console sign-in
Manage console access
Console sign-in link
https://905418008126.signin.aws.amazon.com/console
Console password
Updated 1 minute ago (2025-03-03 22:54 GMT-3)
Last console sign-in
Never
Multi-factor authentication (MFA) (1)
Remove
Resync
Assign MFA device
Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more
Type
Identifier
Certifications
Created on
Passkeys and security keys
arn:aws:iam::905418008126:u2f/user/manzana_devops/macOs-XG6MHUAN4VH33PC6RTE3UXK63I
0
Mon Mar 03 2025
Access keys (0)
Create access key
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more
No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more
Create access key
SSH public keys for AWS CodeCommit (0)
Actions
Upload SSH public key
User SSH public keys to authenticate access to AWS CodeCommit repositories. You can have a maximum of five SSH public keys (active or inactive) at a time. Learn more
SSH Key ID
Uploaded
Status
No SSH public keys
Upload SSH public key
HTTPS Git credentials for AWS CodeCommit (0)
Actions
Generate credentials
Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can have a maximum of 2 sets of credentials (active or inactive) at a time. Learn more
User name
Created
Status
No credentials
Generate credentials
Credentials for Amazon Keyspaces (for Apache Cassandra) (0)
Actions
Generate credentials
Generate a user name and password you can use to authenticate to Amazon Keyspaces. You can have a maximum of two sets of credentials (active or inactive) at a time. Learn more
User name
Created
Status
No credentials
Generate credentials
X.509 Signing certificates (0)
Actions
Upload
Create X.509 certificate
Use X.509 certificates to make secure SOAP-protocol requests to some AWS services. You can have a maximum of two X.509 certificates (active or inactive) at a time. Learn more
Creation time
Thumbprint
Status
No X.509 certificates
Create X.509 certificate
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

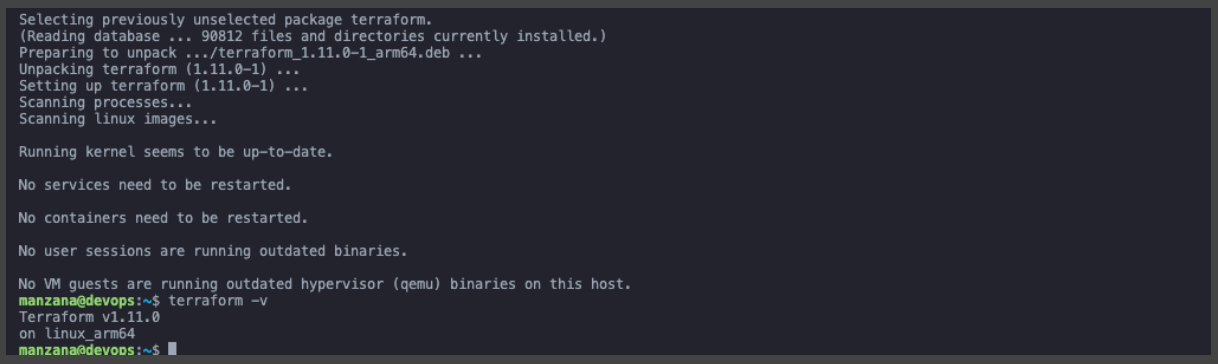
Configuración de Terraform para EKS:

Terraform:

```wget -O - https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) 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 install terraform
```



AWS CLI:

```curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

kubectl para ARM:

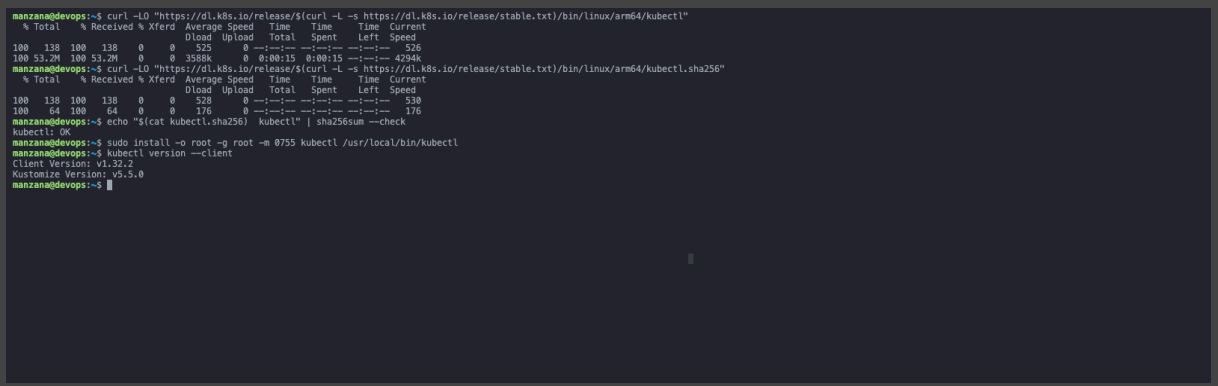
```curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/arm64/kubectl"

curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/arm64/kubectl.sha256"

echo "$(cat kubectl.sha256)  kubectl" | sha256sum --check

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

kubectl version --client
```



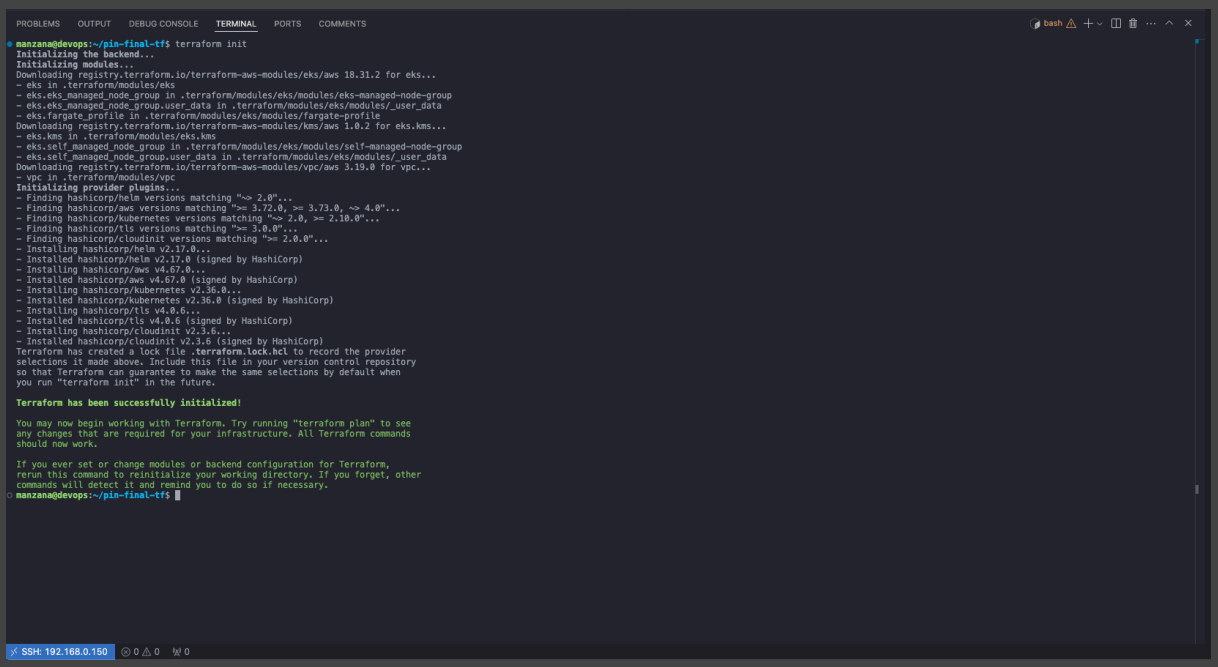
Repositorio de Github:

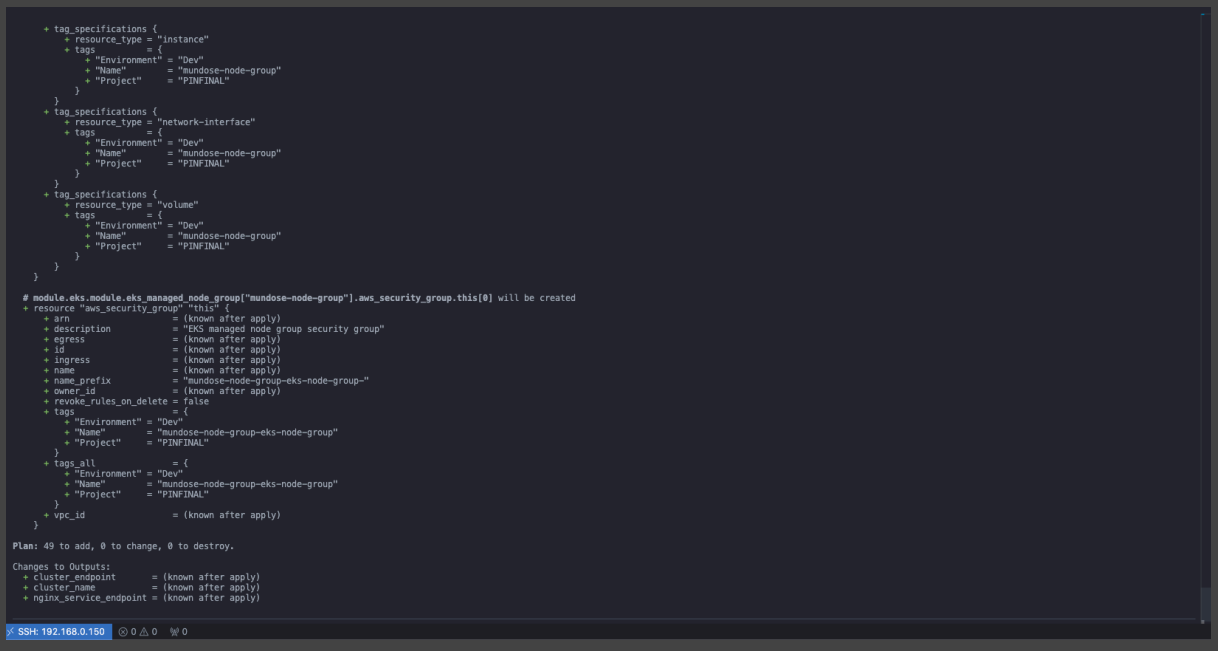
Creamos un repositorio con todas las configuraciones del IaaC donde detallamos los pasos para el despliegue del Clúster:

https://github.com/manzana2164/pin-final-tf.git

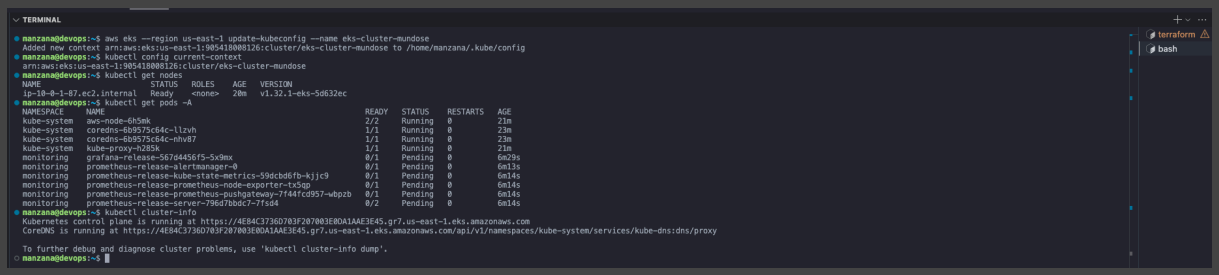
Inicialización y aplicación de Terraform:

```
terraform init
```





Después del terraform apply, chequeamos que es accesible el cluster:



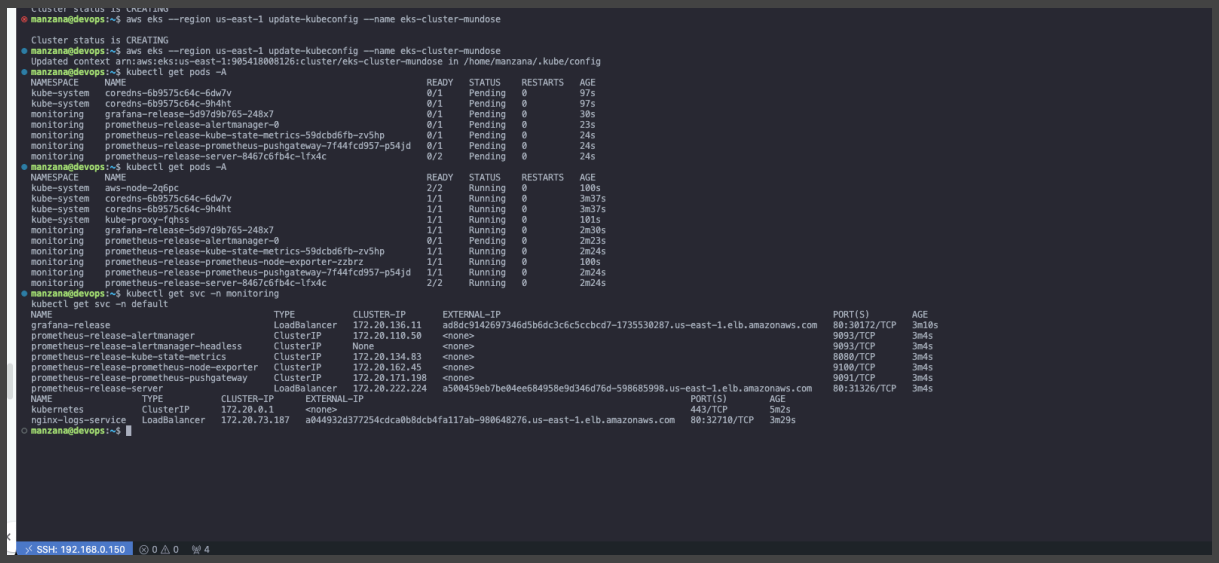
Accedemos al cluster:

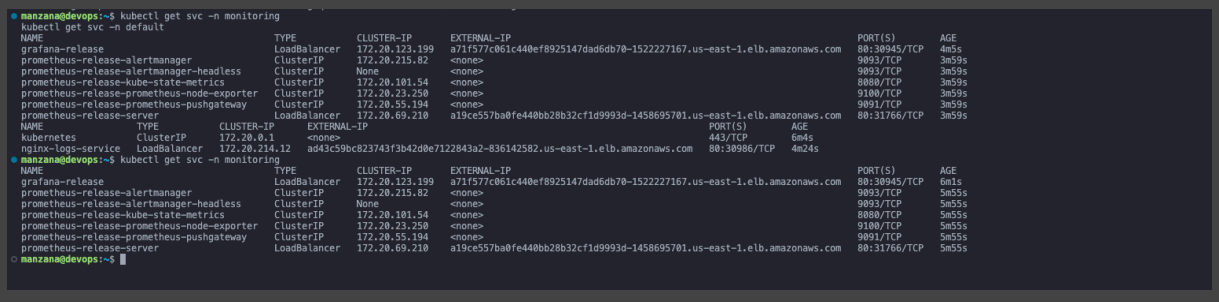
```
aws eks --region us-east-1 update-kubeconfig --name eks-cluster-mundose
```

```
kubectl get pods -A
```

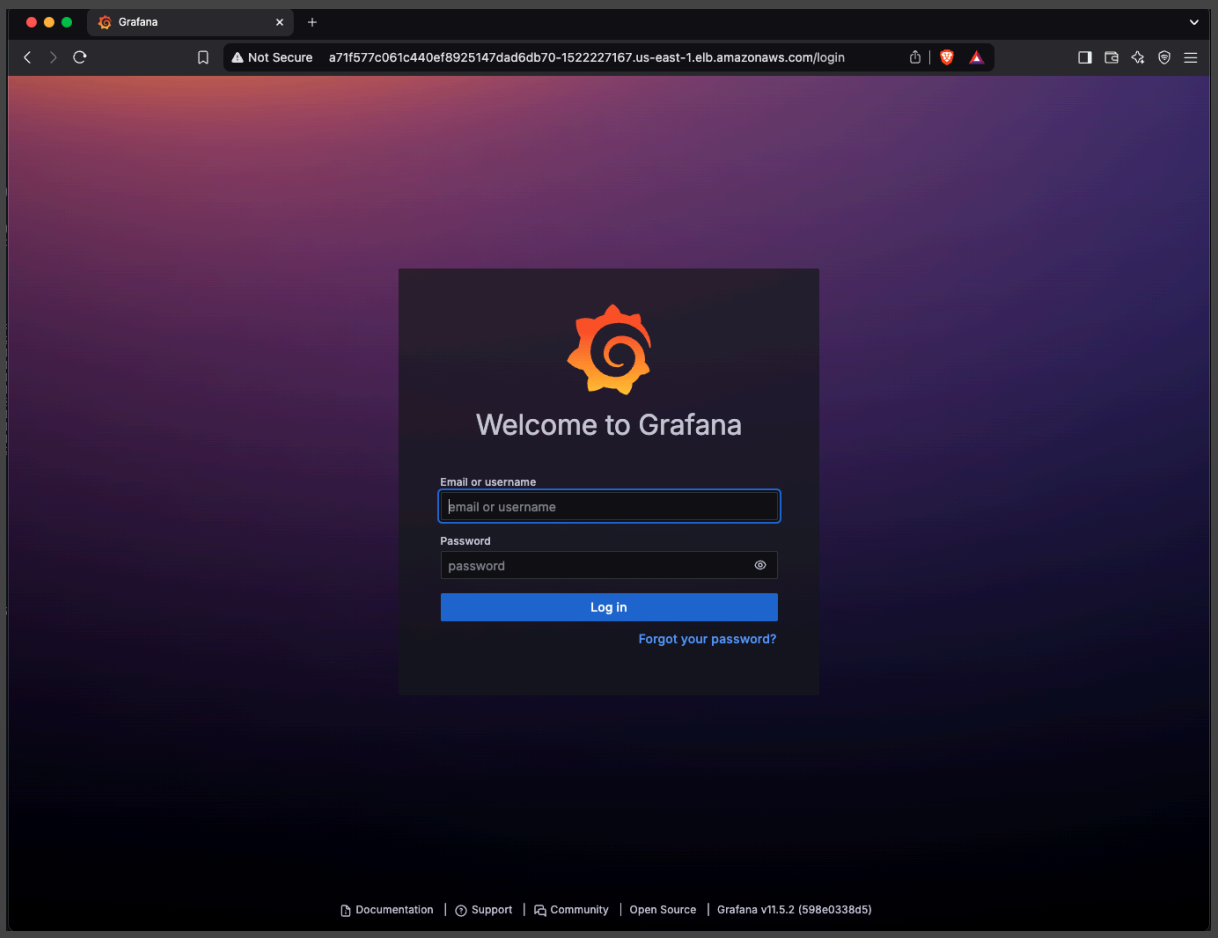
Chequeamos los LoadBalancers:

```
kubectl get svc -n monitoring
kubectl get svc -n default
```

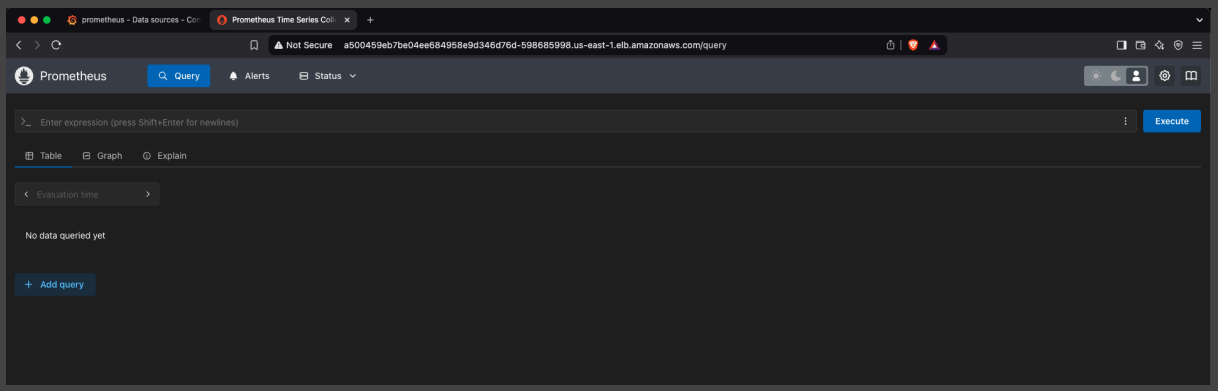




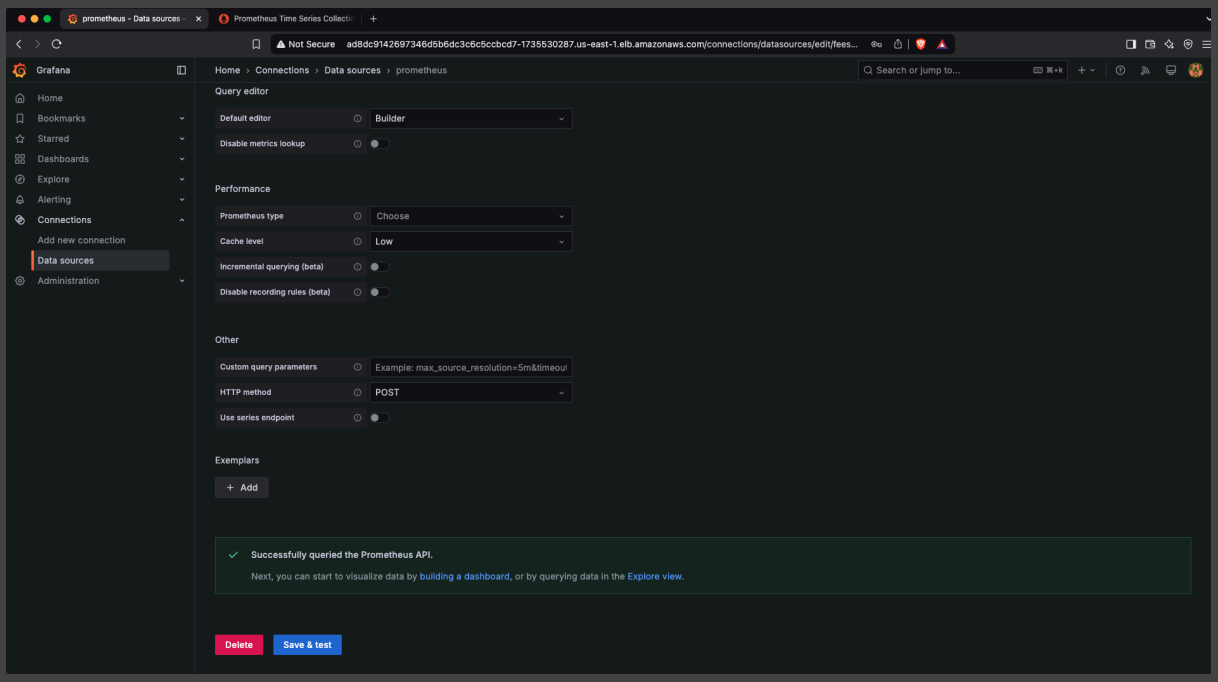
Accedemos a Grafana:
ad8dc9142697346d5b6dc3c6c5ccbcd7-1735530287.us-east-1.elb.amazonaws.com



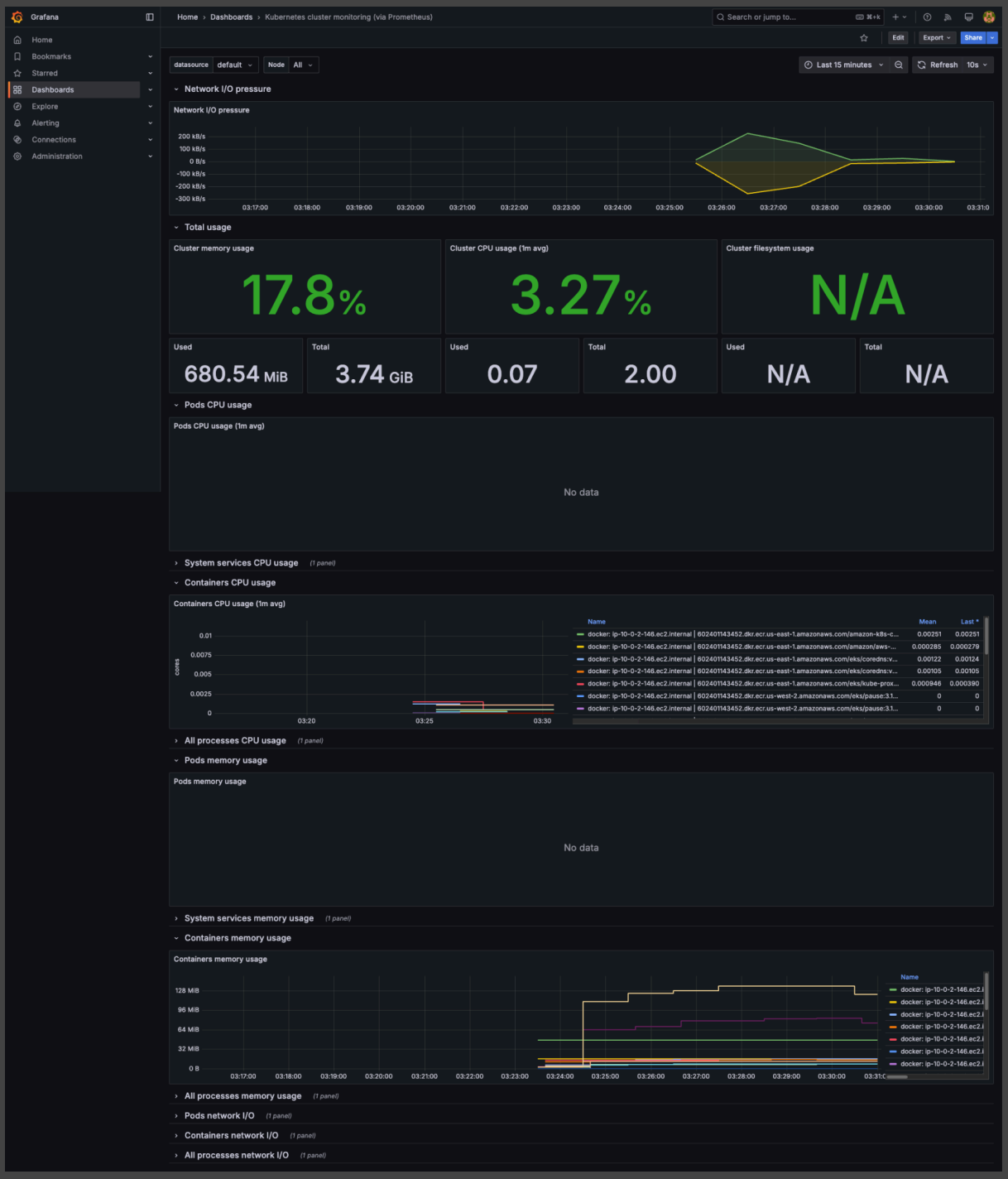
Accedemos a Prometheus:
a500459eb7be04ee684958e9d346d76d-598685998.us-east-1.elb.amazonaws.com

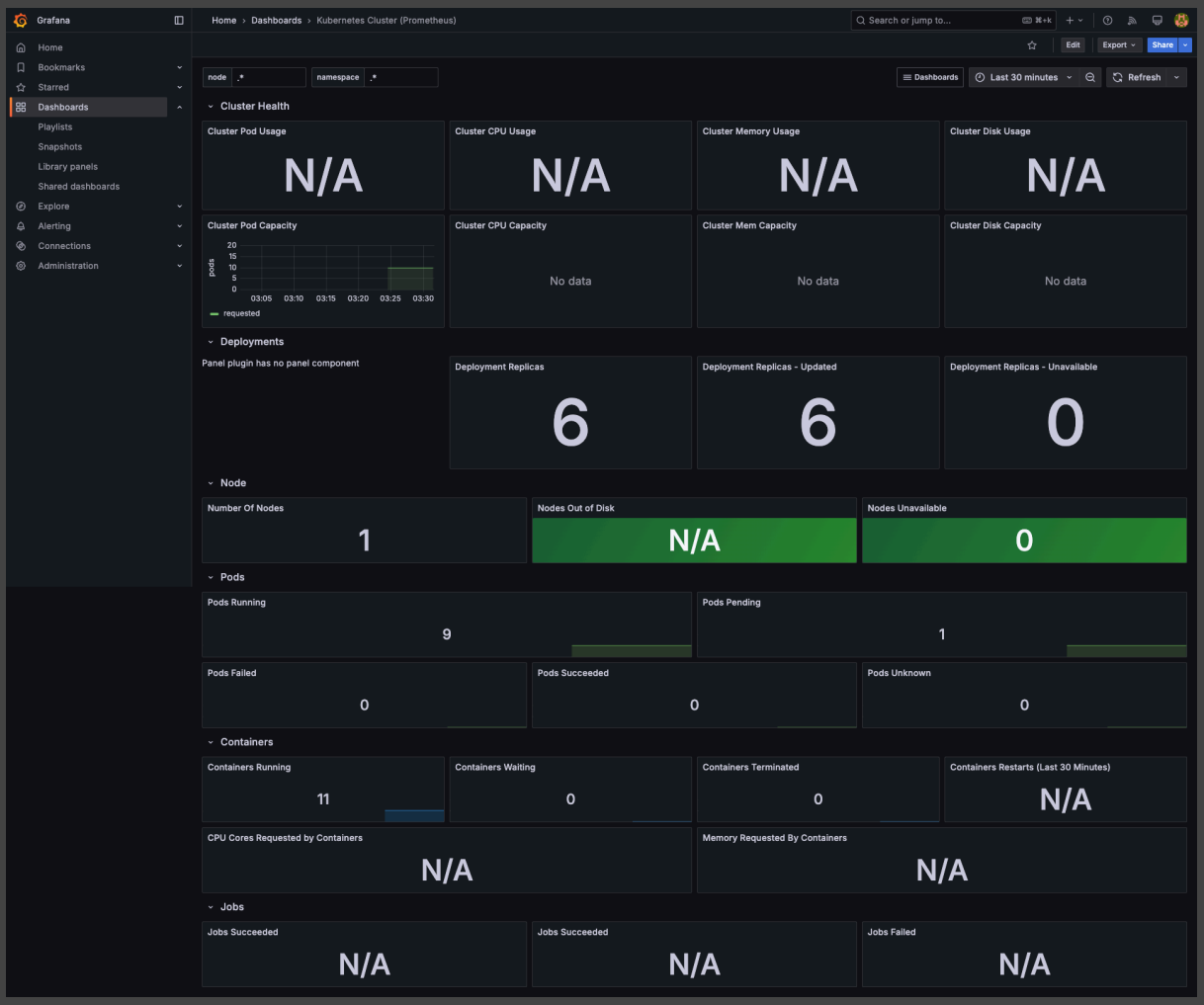


Agregamos la source de Prometheus:

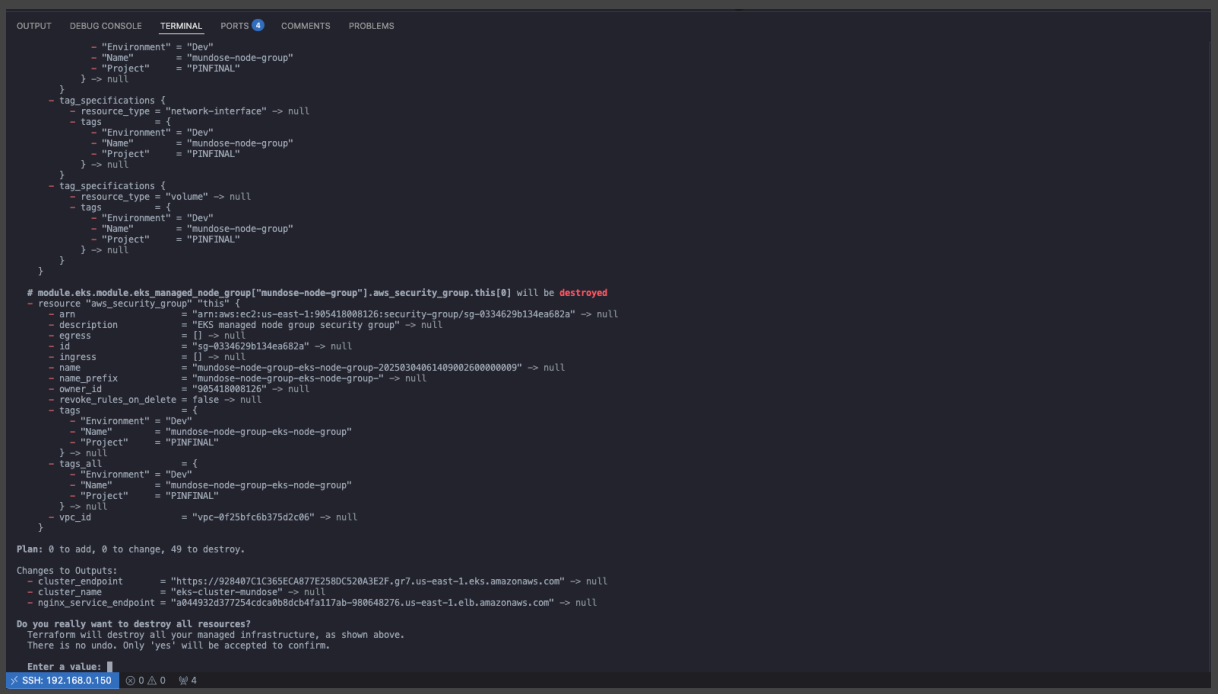


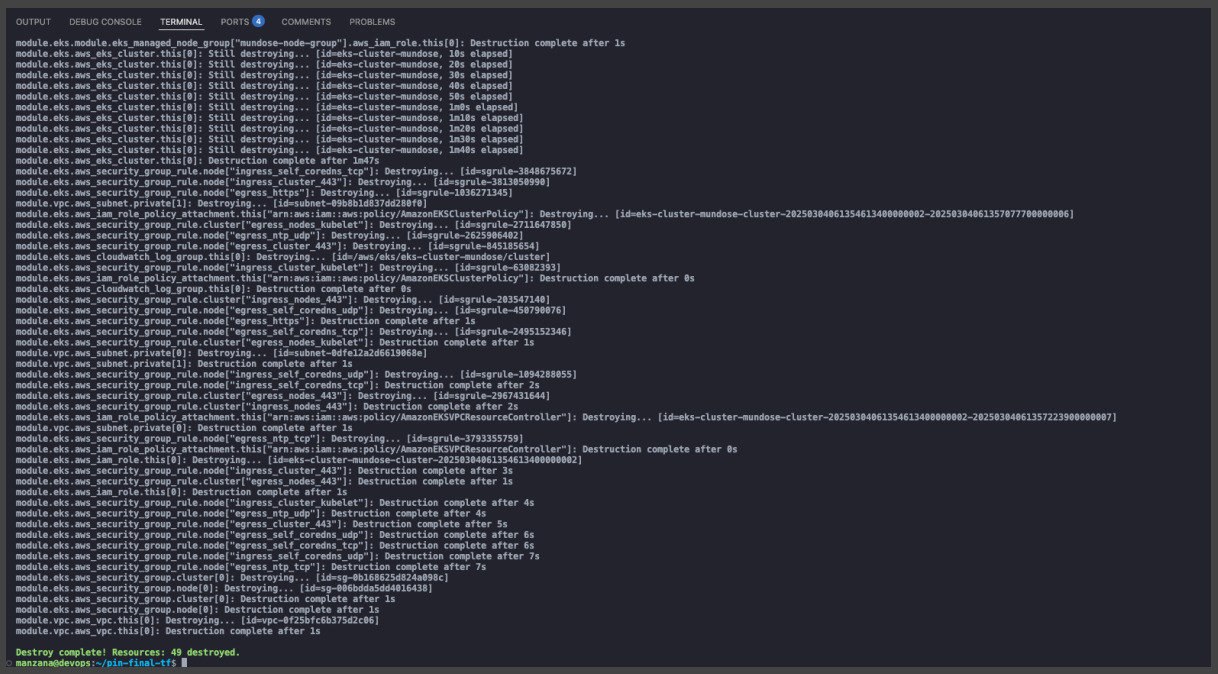
Creamos los dashboards por importación con los códigos 3119 y 6417


Grafana
Home
Bookmarks
Starred
Dashboards
Explore
Alerting
Connections
Administration
Home › Dashboards › Kubernetes cluster monitoring (via Prometheus)
Search or jump to...
Edit
Export
Share
datasource default
Node All
Last 15 minutes
Refresh 10s
Network I/O pressure
Network I/O pressure
200 kB/s
100 kB/s
0 B/s
-100 kB/s
-200 kB/s
-300 kB/s
03:17:00 03:18:00 03:19:00 03:20:00 03:21:00 03:22:00 03:23:00 03:24:00 03:25:00 03:26:00 03:27:00 03:28:00 03:29:00 03:30:00
Total usage
Cluster memory usage
17.8%
Cluster CPU usage (1m avg)
3.27%
Cluster filesystem usage
N/A
Used
680.54 MiB
Total
3.74 GiB
Used
0.07
Total
2.00
Used
N/A
Total
N/A
Pods CPU usage
Pods CPU usage (1m avg)
No data
System services CPU usage (1 panel)
Containers CPU usage
Containers CPU usage (1m avg)
cores
0.01
0.0075
0.005
0.0025
0
03:20
03:25
03:30
Name
Mean
Last *
docker: ip-10-0-2-146.ec2.internal | 602401143452.dkr.ecr.us-east-1.amazonaws.com/amazon-k8s-c... 0.00251 0.00251
docker: ip-10-0-2-146.ec2.internal | 602401143452.dkr.ecr.us-east-1.amazonaws.com/amazon/aws-... 0.000285 0.000279
docker: ip-10-0-2-146.ec2.internal | 602401143452.dkr.ecr.us-east-1.amazonaws.com/eks/coredns:v... 0.00122 0.00124
docker: ip-10-0-2-146.ec2.internal | 602401143452.dkr.ecr.us-east-1.amazonaws.com/eks/coredns:v... 0.00105 0.00105
docker: ip-10-0-2-146.ec2.internal | 602401143452.dkr.ecr.us-east-1.amazonaws.com/eks/kube-prox... 0.000946 0.000390
docker: ip-10-0-2-146.ec2.internal | 602401143452.dkr.ecr.us-west-2.amazonaws.com/eks/pause:3.1... 0 0
docker: ip-10-0-2-146.ec2.internal | 602401143452.dkr.ecr.us-west-2.amazonaws.com/eks/pause:3.1... 0 0
All processes CPU usage (1 panel)
Pods memory usage
Pods memory usage
No data
System services memory usage (1 panel)
Containers memory usage
Containers memory usage
128 MiB
96 MiB
64 MiB
32 MiB
0 B
03:17:00 03:18:00 03:19:00 03:20:00 03:21:00 03:22:00 03:23:00 03:24:00 03:25:00 03:26:00 03:27:00 03:28:00 03:29:00 03:30:00
Name
All processes memory usage (1 panel)
Pods network I/O (1 panel)
Containers network I/O (1 panel)
All processes network I/O (1 panel)



Luego de verificar los dashboards, procedemos con un `terraform destroy` a eliminar la instancia EC2





Análisis y Conclusiones:

Para la región **`us-east-1`**, utilizamos la versión de Kubernetes **`1.32`**, dado que la versión 1.22 no era compatible.

La instancia al ser t2.micro, le agregamos un timeout a Prometheus y Grafana de 600 segundos. Dado al tiempo de demora en el deploy Prometheus, modificamos el **`eks.tf`** para incorporar una instancia de **`T3.Medium`**, y, además, limitamos los recursos de los pods en cuanto a cpu y memoria, cpu = “500m” y memoria = "512Mi".