

# Proyecto Integrador Final - PIN

## Grupo 24 - Curso Devops 2404

### Integrantes

Denis Medel  
Eric Unzue Madueña  
Ignacio Montero  
Oscar Avendaño

### Herramientas Utilizadas

- AWS
- Terraform
- GitHub
- Grafana
- Prometheus
- Helm

### Proceso de Implementación

Se utilizó una VM local para la instalación de AWS-CLI, KUBECTL y TERRAFORM.

# Herramientas utilizadas:

## Instalación aws-cli:

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

```
administrador@denis-test:/opt/dockerfiles/curso-devops-pin3/kubernetes/terraform$ aws --version
aws-cli/2.23.10 Python/3.12.6 Linux/5.4.0-208-generic exe/x86_64.ubuntu.20
```

Configurar AWS CLI usando el comando **aws configure**:

1. Ejecuta el comando:  
`aws configure`
2. Se te pedirá que ingreses la siguiente información:
  - **AWS Access Key ID**: Ingresa tu clave de acceso
  - **AWS Secret Access Key**: Ingresa tu clave secreta
  - **Default region name**: Ingresa tu región preferida (por ejemplo: `us-east-1`, `us-west-2`, `eu-west-1`)
  - **Default output format**: Ingresa tu formato preferido (opciones: `json`, `yaml`, `text`, `table`)

Ejemplo:

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]:
wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-east-1
Default output format [None]: json
```

Para obtener tus claves de acceso, necesitarás:

1. Iniciar sesión en la consola de AWS
2. Ir al recurso IAM y crear un usuario para aws-cli
3. Seleccionar el usuario y configurar **permissions policies**
4. Crear una accesskey a partir de la pestaña Access Key del usuario

## Users (2) [Info](#)

An IAM user is an Identity with long-term credentials that is used to interact with AWS in an account.

[Delete](#)[Create user](#)[<](#) 1 [>](#) [Settings](#)

<input type="checkbox"/>	User name	Path	Group	Last activity	MFA	Password age
<input type="checkbox"/>	<a href="#">Curso-Devops</a>	/	0	<span>✓ Yesterday</span>	Virtual	<span>✓ 72 days</span>

### Access keys (1)

[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](#)

<div><div></div><div><b>Description</b> aws-cli-curso-devops</div><div><b>Last used</b> Yesterday</div><div><b>Last used region</b> us-east-1</div></div> <div><div><b>Status</b> <span>✓ Active</span></div><div><b>Created</b> 72 days ago</div><div><b>Last used service</b> s3</div></div> <div><div>Actions</div></div>
--

## Instalación kubectl:

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
chmod +x kubectl
sudo mv kubectl /usr/local/bin/
kubectl version --client
```

```
administrador@denis-test:/opt/dockerfiles/curso-devops-pin3/kubernetes/terraform$ kubectl version --client
Client Version: v1.32.3
Kustomize Version: v5.5.0
```

## Instalación de Terraform

```
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  
apt update && apt install -y terraform
```

## Creación dynamodb a través de **aws-cli**:

```
aws dynamodb create-table \  
--table-name terraformstatelock \  
--billing-mode PAY_PER_REQUEST \  
--attribute-definitions AttributeName=LockID,AttributeType=S \  
--key-schema AttributeName=LockID,KeyType=HASH \  
--tags Key=Nombre,Value=terraform_state_lock_curso_test
```

## Creación del Bucket S3 para Terraform a través de **aws-cli**:

```
aws s3api create-bucket \  
--bucket s3-tfstate-curso-test \  
--region us-east-1
```

## Despliegue de la infraestructura con terraform:

### **Inicialización proyecto:**

```
terraform init
```

### **Visualiza el plan de lo que Terraform creará:**

```
terraform plan
```

### **Aplica la configuración para crear la infraestructura:**

```
terraform apply
```

# Verificamos que la infraestructura esté corriendo:

### Clusters (1) [Info](#)

Cluster name

▲

Status

Kubernetes version

Support period

Upgrade policy

monitoreo-curso-g-24

Active

1.29 Upgrade now

Extended support until March 22, 2026

Extended

### Instances (3) [Info](#)

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

All states

☐

Name

▼

Instance ID

Instance state

Instance type

Status check

Alarm status

Ava

☐

default

i-0d1837ffd1940e571

Running

t3.medium

3/3 checks passed

View alarms +

us-e

☐

default

i-0189eca15445c5a75

Terminated

t3.small

-

View alarms +

us-e

☐

default

i-03c994f423f7eb2c0

Terminated

t3.small

-

View alarms +

us-e

## Lista servicios

```
administrador@denis-test:/opt/dockerfiles/curso-devops-pin3/kubernetes/terraform$ kubectl get svc -n curso-monitoreo
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
alertmanager-operated	ClusterIP	None	<none>	9093/TCP,9094/TCP,9094/UDP	10m
prometheus-grafana	NodePort	172.20.217.4	<none>	80:31000/TCP	10m
prometheus-kube-prometheus-alertmanager	ClusterIP	172.20.77.199	<none>	9093/TCP,8080/TCP	10m
prometheus-kube-prometheus-operator	ClusterIP	172.20.169.50	<none>	443/TCP	10m
prometheus-kube-prometheus-prometheus	NodePort	172.20.29.159	<none>	9090:31001/TCP,8080:31032/TCP	10m
prometheus-kube-state-metrics	ClusterIP	172.20.113.43	<none>	8080/TCP	10m
prometheus-operated	ClusterIP	None	<none>	9090/TCP	10m
prometheus-prometheus-node-exporter	ClusterIP	172.20.210.33	<none>	9100/TCP	10m

```
administrador@denis-test:/opt/dockerfiles/curso-devops-pin3/kubernetes/terraform$
administrador@denis-test:/opt/dockerfiles/curso-devops-pin3/kubernetes/terraform$
```

## Lista los Pods

```
administrador@denis-test:/opt/dockerfiles/curso-devops-pin3/kubernetes/terraform$ kubectl get pods -n curso-monitoreo
```

NAME	READY	STATUS	RESTARTS	AGE
alertmanager-prometheus-kube-prometheus-alertmanager-0	2/2	Running	0	51m
prometheus-grafana-79d9bf8cbd-hw6hv	3/3	Running	0	51m
prometheus-kube-prometheus-operator-6b65c566bd-vdgsq	1/1	Running	0	51m
prometheus-kube-state-metrics-5d84f96dc4-62q94	1/1	Running	0	51m
prometheus-prometheus-kube-prometheus-prometheus-0	2/2	Running	0	51m
prometheus-prometheus-node-exporter-brzb8	1/1	Running	0	51m
prometheus-prometheus-node-exporter-hgm6h	1/1	Running	0	2m4s

```
administrador@denis-test:/opt/dockerfiles/curso-devops-pin3/kubernetes/terraform$
```

## Lista la IP Pública del Nodo

```
administrador@denis-test:/opt/dockerfiles/curso-devops-pin3/kubernetes/terraform$ kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERN
ip-10-0-1-110.ec2.internal	Ready	<none>	145m	v1.29.13-eks-5d632ec	10.0.1.110	98.80.170.131	Amazon Linux 2	5.10
234-225.921.amzn2.x86_64	containerd://1.7.27							

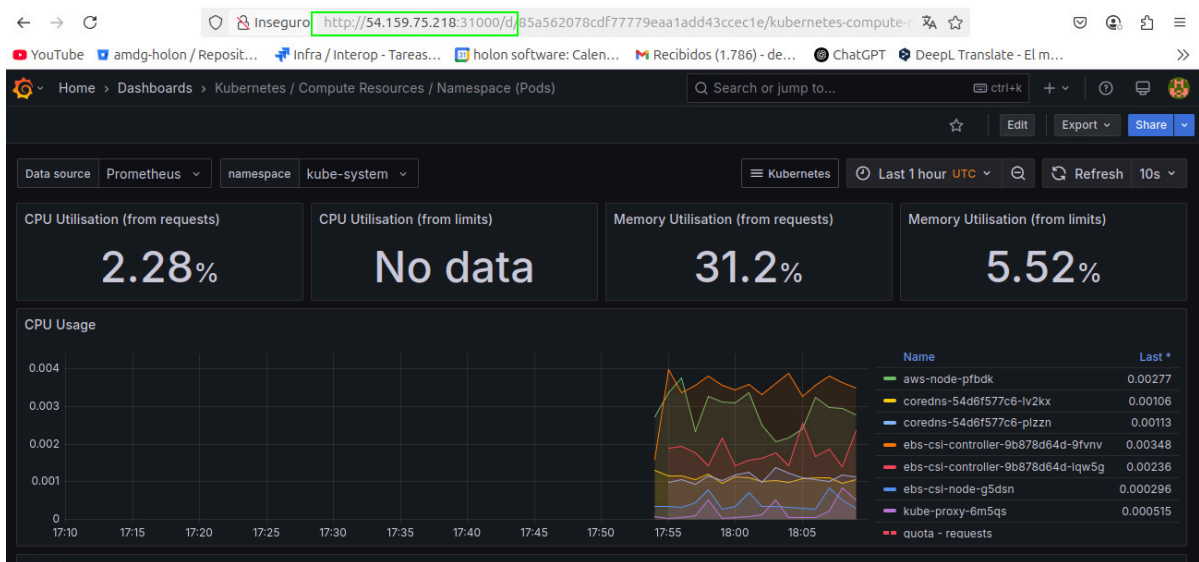
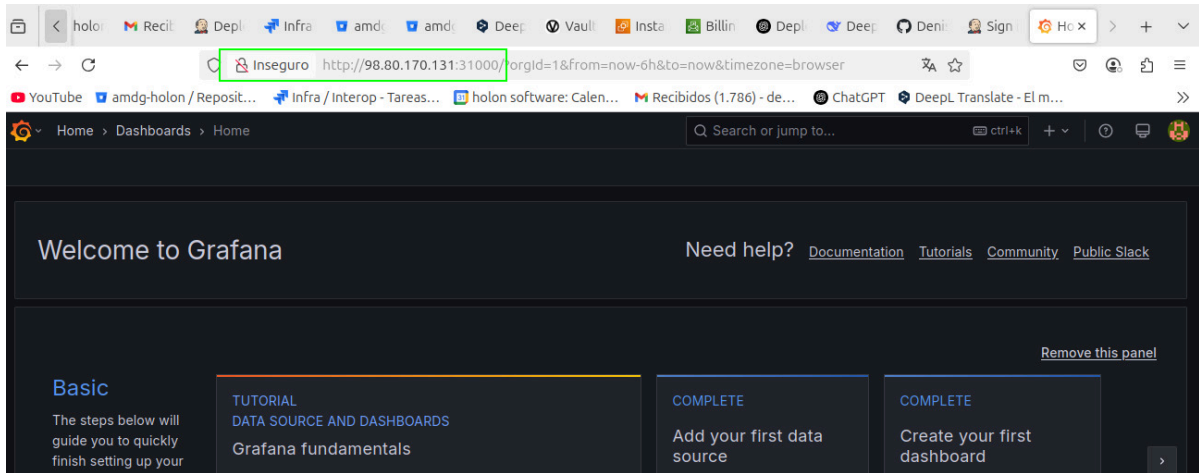
```
administrador@denis-test:/opt/dockerfiles/curso-devops-pin3/kubernetes/terraform$ kubectl get node -o wide
```

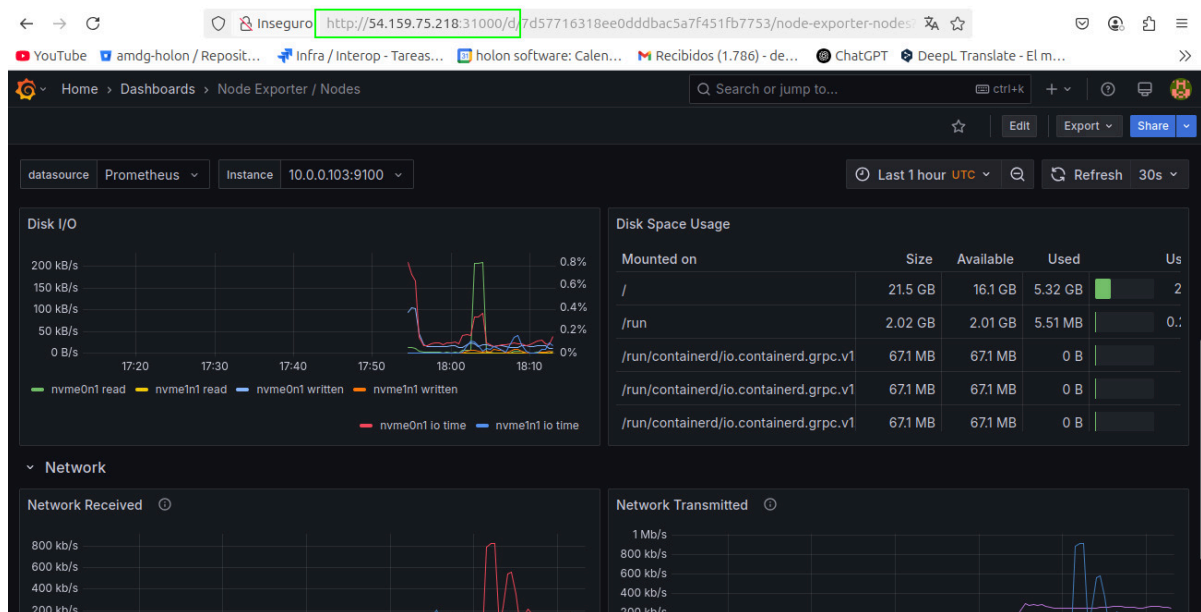
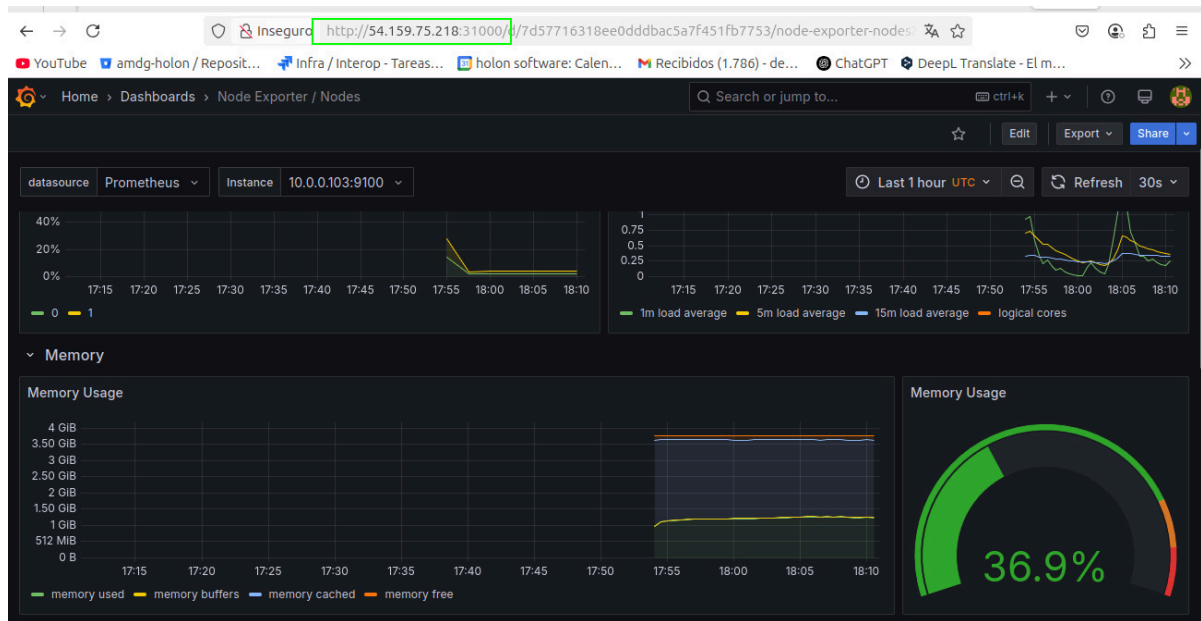
NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNE
ip-10-0-0-103.ec2.internal	Ready	<none>	11m	v1.29.13-eks-5d632ec	10.0.0.103	54.159.75.218	Amazon Linux 2	5.10
234-225.921.amzn2.x86_64	containerd://1.7.27							

## Monitoreo y Visualización

Se utiliza Grafana para visualización de Dashboards y en cuanto a recopilación de métricas son tomados por Prometheus ( agentes ) quien es la fuente de datos para este proyecto.

Accedemos a Grafana a través la ip externa recuperada anteriormente.





# Errores y Soluciones

Permisos del usuario creado anteriormente que interactúa con el cluster



## Error al Conectar Helm con Cluster de EKS

```
Error: Kubernetes cluster unreachable: the server has asked for the client to provide credentials

with helm_release.Stack-Grafana,
on helm.tf line 10, in resource "helm_release" "Stack-Grafana":
10: resource "helm_release" "Stack-Grafana" {
```

## Configurar Políticas de Acceso para el usuario AWS-CLI

**Access policies (2)** [Info](#) Remove Edit Add access policy

The access policies associated to the access entry and Kubernetes namespaces that you've scoped the access policies to.

	Policy name	Kubernetes namespaces
<input type="radio"/>	AmazonEKSAAdminPolicy	
<input type="radio"/>	AmazonEKSClusterAdminPolicy	

## Actualizar el .kube/config

aws eks update-kubeconfig --name nombre-del-cluster --region us-east-1

## Error al levantar el chart Helm debido al límite de PODS en la instancia worker

```
Events:
  Type     Reason             Age          From          Message
  ----     -
  Warning   FailedScheduling   25s (x9 over 40m)  default-scheduler  0/1 nodes are available: 1 Too many pods.
preemption: 0/1 nodes are available: 1 No
preemption victims found for incoming pod.
administrador@denis-test: /opt/dockerfiles/curso-devops-pin3/kubernetes/terraform$
```