

# Proyecto Integrador PIN2204

Integrantes:

## Contenido

|   |           |
|---|-----------|
| <b>Introducción</b>                               | <b>3</b>  |
| <b>Crear instancia EC2</b>                        | <b>6</b>  |
| <b>Conectar con instancia por SSH:</b>            | <b>7</b>  |
| <b>Crear cluster con eksctl</b>                   | <b>7</b>  |
| <b>Instalar herramientas de monitoreo de pods</b> | <b>15</b> |
| <b>Desplegar Grafana</b>                          | <b>18</b> |
| <b>Cleanup de recursos</b>                        | <b>20</b> |

# Introducción

Este proyecto tiene como idea principal el aprendizaje sobre distintos temas y la puesta en práctica mediante un laboratorio que permita integrar diferentes herramientas y tecnologías.

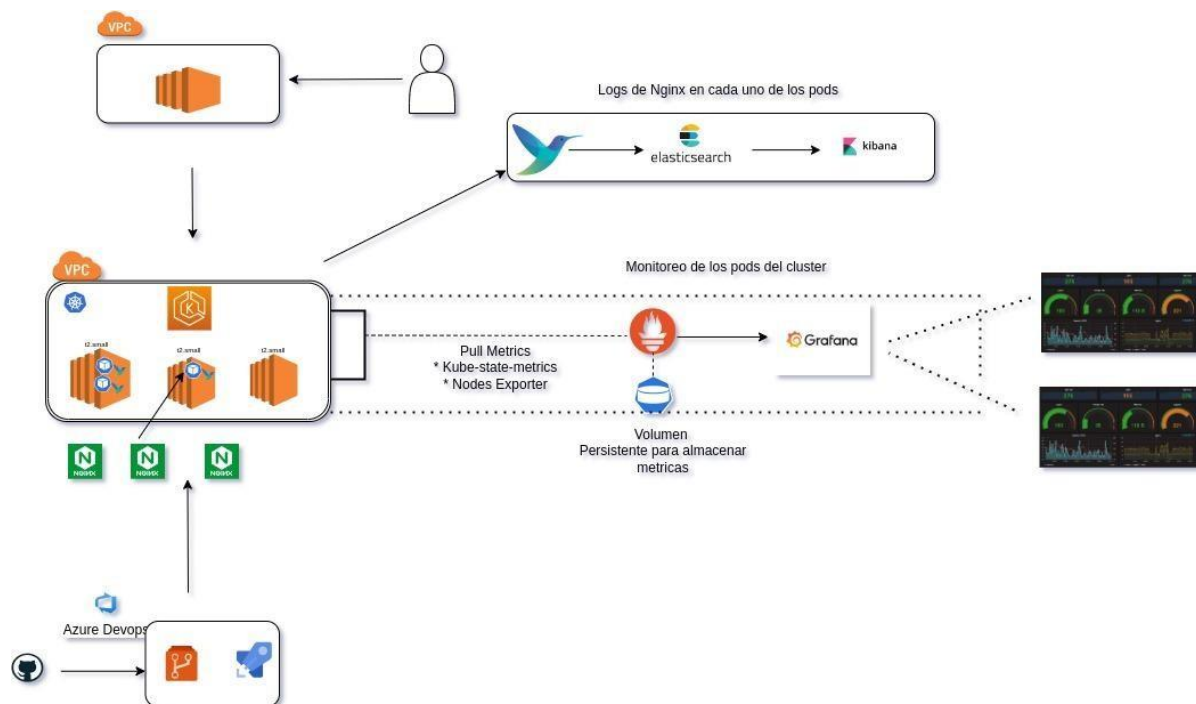
Durante la primera parte nos centramos en la creación de una instancia de EC2 en AWS para poder desde allí realizar todas las tareas necesarias.

Luego comenzamos con el despliegue de un cluster de Kubernetes con con la opción de ekcli

Una vez configurado el cluster, integramos un repositorio de Git en un proyecto de Azure Devops, mediante el cual se puede demostrar la integración continua y el despliegue continuo (CI/CD) desplegando en el cluster 3 réplicas de nginx.

En la segunda parte, configuraremos el monitoreo de log con el stack de Elastic, FluentBit y Kibana, el cual nos permitió ver los logs de nginx corriendo en el cluster

Por último, configuramos la parte de monitoreo de pods con el stack de Prometheus y Grafana.



<https://github.com/andres-dcic/mundoes>

# Crear instancia EC2

## Crear instancia siguiendo ej. del PIN:

Region: us-east-1

Sistema Operativo : Ubuntu Server 22.04

Family (Tipo): t2.micro

En la sección de “user data” se proceden a cargar todos los scripts para instalar las herramientas necesarias, como AWS CLI, KUBECTL, Docker, Helm, etc. y que listamos en el archivo (aqui link al archivo .sh del repo que compartiremos)

Se crea un par de claves para poder conectarse, llamadas “pin” en formato pem

## Data de Instancia:

The screenshot displays the AWS Management Console interface for an EC2 instance. The left sidebar shows navigation options like 'Panel de EC2', 'Vista global de EC2', 'Eventos', 'Límites', 'Instancias', 'Tipos de instancia', 'Plantillas de lanzamiento', 'Solicitudes de spot', 'Savings Plans', 'Instancias reservadas', 'Alojamientos dedicados', 'Instancias programadas', 'Reservas de capacidad', 'Imágenes', 'AMI', and 'Catálogo de AMI'. The main content area is titled 'Resumen de instancia de i-0468219656f6f09a7 (pin)' and includes a 'Conectar' button and a dropdown for 'Estado de la instancia'. The instance details are organized into three columns:

| Column 1 (Left)  | Column 2 (Middle)   | Column 3 (Right)   |
|--|---|--|
| ID de la instancia<br>i-0468219656f6f09a7 (pin)                            | Dirección IPv4 pública<br>54.91.186.4   <a href="#">dirección abierta</a> | Direcciones IPv4 privadas<br>172.31.86.150   |
| Dirección IPv6<br>-  | Estado de la instancia<br>En ejecución                                    | DNS de IPv4 pública<br>ec2-54-91-186-4.compute-1.amazonaws.com   <a href="#">dirección abierta</a>                                       |
| Tipo de nombre de anfitrión<br>Nombre de IP: ip-172-31-86-150.ec2.internal | Nombre DNS de IP privada (solo IPv4)<br>ip-172-31-86-150.ec2.internal     | Direcciones IP elásticas<br>-  |
| Responder al nombre DNS de recurso privado IPv4 (A)                        | Tipo de instancia<br>t2.micro   | Hallazgo de AWS Compute Optimizer<br>Suscribirse a AWS Compute Optimizer para recibir recomendaciones.   <a href="#">Más información</a> |
| Dirección IP asignada automáticamente<br>54.91.186.4 [IP pública]          | ID de VPC<br>vpc-0a979dce4e3dfce03  | Nombre del grupo de Auto Scaling<br>-  |
| Rol de IAM<br>ec2-admin-role-cicd  | ID de subred<br>subnet-0f30f4f20addd3643                                  |  |
| IMDSv2<br>Optional   |   |  |

Se le agrega a la instancia el rol ec2-admin, previamente creado

## Conectar con instancia por SSH:

```
ubuntu@ip-172-31-86-150: ~  
PS C:\Users\lcher\Downloads> ssh -i "pin.pem" ubuntu@ec2-54-91-186-4.compute-1.amazonaws.com  
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Mon Jun 26 12:31:49 UTC 2023  
  
System load:  0.0          Processes:            111  
Usage of /:   45.1% of 7.57GB Users logged in:      0  
Memory usage: 38%         IPv4 address for docker0: 172.17.0.1  
Swap usage:   0%          IPv4 address for eth0:   172.31.86.150  
  
* Ubuntu Pro delivers the most comprehensive open source security and  
  compliance features.  
  
https://ubuntu.com/aws/pro  
  
Expanded Security Maintenance for Applications is not enabled.  
  
23 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
*** System restart required ***  
Last login: Mon Jun 26 11:39:27 2023 from 167.116.35.249  
ubuntu@ip-172-31-86-150:~$
```

## Crear cluster con eksctl

```
eksctl create cluster \  
--name eks-mundos-e \  
--region us-east-1 \  
--node-type t3.small \  
--nodes 3 \  
--with-oidc \  
--ssh-access \  
--ssh-public-key pin \  
--managed \  
--full-ecr-access \  
--zones us-east-1a,us-east-1b,us-east-1c
```

El objetivo en esta instancia, es desplegar un pod de nginx, utilizando cualquier método válido, hasta la misma consola de aws.

Verificar NGINX en hostname:

← → ↻ 🏠 🔒 No seguro | a7b3929d76ab34732b9b67053e13f0a7-434065905.us-east-1.elb.amazonaws.com 🔗 ☆ ⏪ ⏩ ⚙️ 🖨️ 👤 ⋮

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

# Instalar herramientas de monitoreo de pods

## Instalación del driver EBS [acorde a este documento oficial de AWS](#)

### Deploy driver

You may deploy the EBS CSI driver via Kustomize, Helm, or as an [Amazon EKS managed add-on](#).

### Kustomize

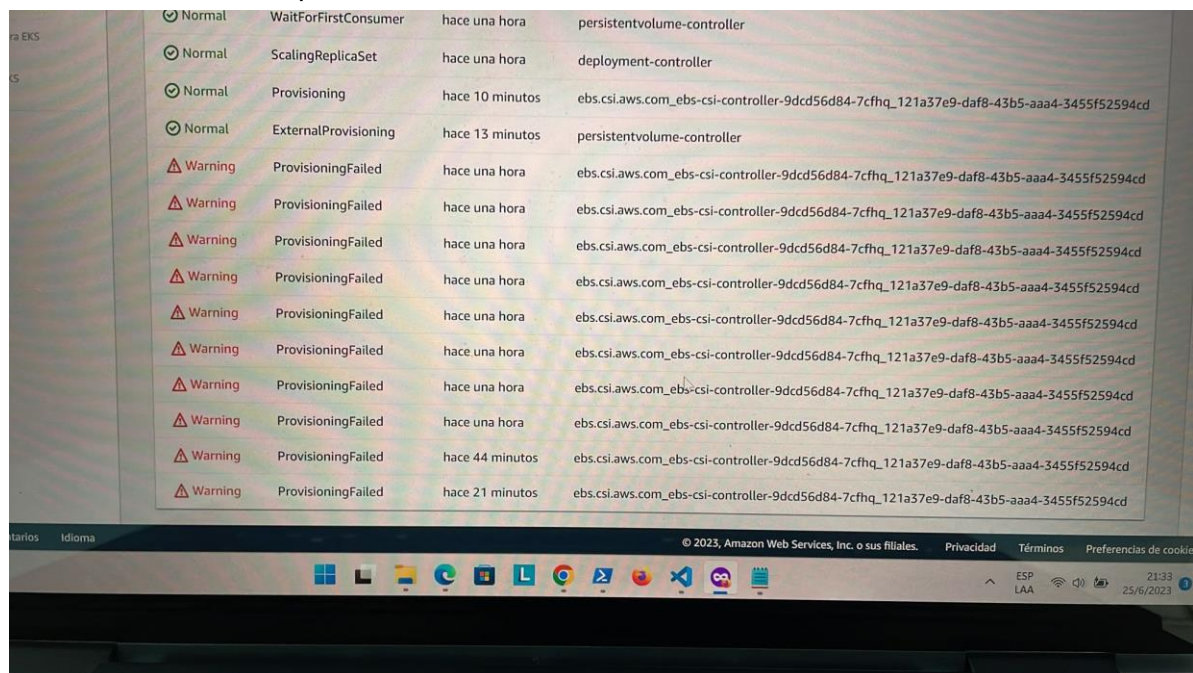
```
kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-1.20"
```

y con el siguiente comando:

```
kubectl apply -k
```

```
"github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-1.20"
```

Hecho eso, se verifica que no inician los servicios afines a AWS EBS y se revisa en el dashboard de AWS la posible causa:



|         |                      |                 |   |
|---------|----------------------|-----------------|---|
| Normal  | WaitForFirstConsumer | hace una hora   | persistentvolume-controller   |
| Normal  | ScalingReplicaSet    | hace una hora   | deployment-controller   |
| Normal  | Provisioning         | hace 10 minutos | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |
| Normal  | ExternalProvisioning | hace 13 minutos | persistentvolume-controller   |
| Warning | ProvisioningFailed   | hace una hora   | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |
| Warning | ProvisioningFailed   | hace una hora   | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |
| Warning | ProvisioningFailed   | hace una hora   | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |
| Warning | ProvisioningFailed   | hace una hora   | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |
| Warning | ProvisioningFailed   | hace una hora   | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |
| Warning | ProvisioningFailed   | hace una hora   | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |
| Warning | ProvisioningFailed   | hace una hora   | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |
| Warning | ProvisioningFailed   | hace una hora   | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |
| Warning | ProvisioningFailed   | hace 44 minutos | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |
| Warning | ProvisioningFailed   | hace 21 minutos | ebs.csi.aws.com_ebs-csi-controller-9dcd56d84-7cfhq_121a37e9-daf8-43b5-aaa4-3455f52594cd |

Y se corrobora que hay un problema de aprovisionamiento.

Procedemos a hacer troubleshooting del error, para lo cual [investigando encontramos que debemos decodificar el mensaje que AWS comparte](#) a fin de identificar el componente que genera el issue. Hallamos que la causa, es que esos recursos no pueden iniciar por falta de almacenamiento.

Por ende, identificamos el nodegroup que debió generarlos, y le asociamos la **política de administración de almacenamiento (EBS)** para poder administrar volúmenes. Eso resuelve el incidente, y nos permite avanzar a lo siguiente.



**Instalación Prometheus siguiendo la secuencia siguiente:**

```
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"
```

**Prometheus corriendo:**

```
EKS!sAWSome
ubuntu@ip-172-31-86-150:~/environment/grafana$ kubectl get all -n prometheus
NAME                                READY    STATUS    RESTARTS   AGE
pod/prometheus-alertmanager-0       1/1      Running   0           93m
pod/prometheus-kube-state-metrics-5fb6fbbf78-jxhmd  1/1      Running   0           93m
pod/prometheus-prometheus-node-exporter-5nnmx      1/1      Running   0           93m
pod/prometheus-prometheus-node-exporter-cqfgr      1/1      Running   0           93m
pod/prometheus-prometheus-node-exporter-zpqn2      1/1      Running   0           93m
pod/prometheus-prometheus-pushgateway-7d55869d46-nwbgs  1/1      Running   0           93m
pod/prometheus-server-78c8b85bf7-hcm8b            2/2      Running   0           93m

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
service/prometheus-alertmanager     ClusterIP      10.100.227.31  <none>         9093/TCP         93m
service/prometheus-alertmanager-headless ClusterIP      None          <none>         9093/TCP         93m
service/prometheus-kube-state-metrics ClusterIP      10.100.200.195 <none>         8080/TCP         93m
service/prometheus-prometheus-node-exporter ClusterIP      10.100.115.67  <none>         9100/TCP         93m
service/prometheus-prometheus-pushgateway ClusterIP      10.100.135.108 <none>         9091/TCP         93m
service/prometheus-server            ClusterIP      10.100.166.163 <none>         80/TCP           93m

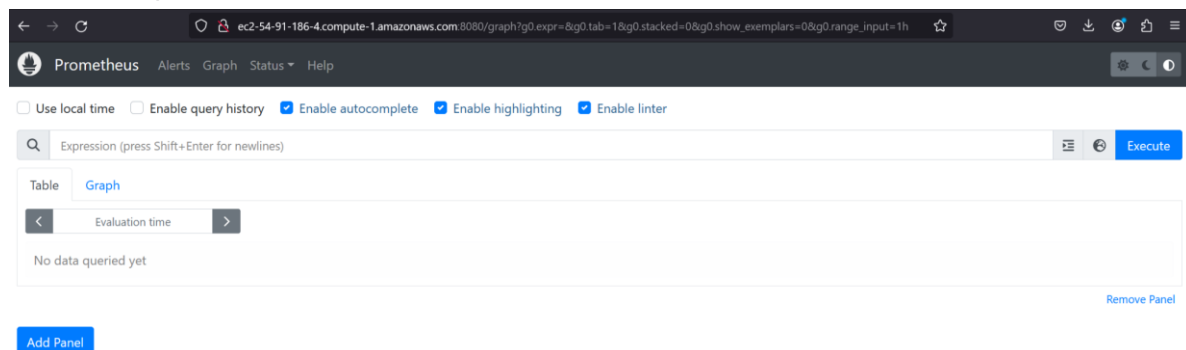
NAME                                DESIRED    CURRENT    READY    UP-TO-DATE    AVAILABLE    NODE SELECTOR    AGE
daemonset.apps/prometheus-prometheus-node-exporter 3           3           3         3              3            <none>           93m

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/prometheus-kube-state-metrics      1/1      1              1           93m
deployment.apps/prometheus-prometheus-pushgateway 1/1      1              1           93m
deployment.apps/prometheus-server                  1/1      1              1           93m

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/prometheus-kube-state-metrics-5fb6fbbf78 1           1           1         93m
replicaset.apps/prometheus-prometheus-pushgateway-7d55869d46 1           1           1         93m
replicaset.apps/prometheus-server-78c8b85bf7             1           1           1         93m
```

**Port forward:**

```
ubuntu@ip-172-31-86-150:~/environment/grafana$ kubectl port-forward -n prometheus deploy/prometheus-server 8080:9090 --address 0.0.0.0
Forwarding from 0.0.0.0:8080 -> 9090
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
```

**Visita en el público:**

Ir a targets:

The screenshot shows the Prometheus web interface at the URL `ec2-54-91-186-4.compute-1.amazonaws.com:9090/targets/search=`. The page title is "Targets". Below the title, there are filters: "All scrape pools", "All", "Unhealthy", "Collapse All", and a search bar "Filter by endpoint or labels". On the right, there are status filters: "Unknown", "Unhealthy", and "Healthy".

The first target group is "kubernetes-apiservers (2/2 up)". It contains two targets:

| Endpoint  | State | Labels   | Last Scrape | Scrape Duration | Error |
|---|-------|--|-------------|-----------------|-------|
| <a href="https://192.168.15.184/metrics">https://192.168.15.184/metrics</a> | UP    | <code>instance="192.168.15.184-443"</code><br><code>job="kubernetes-apiservers"</code> | 34.675s ago | 108.010ms       |       |
| <a href="https://192.168.63.29/metrics">https://192.168.63.29/metrics</a>   | UP    | <code>instance="192.168.63.29-443"</code><br><code>job="kubernetes-apiservers"</code>  | 37.880s ago | 110.229ms       |       |

The second target group is "kubernetes-nodes (3/3 up)". It contains one target:

| Endpoint  | State | Labels  | Last Scrape | Scrape Duration | Error |
|---|-------|---|-------------|-----------------|-------|
| <a href="https://kubernetes.default.svc/api/v1/nodes/ip-192-168-56-76.ec2.internal/proxy/metrics">https://kubernetes.default.svc/api/v1/nodes/ip-192-168-56-76.ec2.internal/proxy/metrics</a> | UP    | <code>alpha_eksctl_io_cluster_name="eks-mundos-a"</code><br><code>alpha_eksctl_io_nodegroup_name="ng-0b643954"</code><br><code>beta_kubernetes_io_arch="amd64"</code><br><code>beta_kubernetes_io_instance_type="t3.small"</code><br><code>beta_kubernetes_io_os="linux"</code><br><code>eks.amazonaws.com_capacityType="ON_DEMAND"</code><br><code>eks.amazonaws.com_nodegroup="ng-0b643954"</code><br><code>eks.amazonaws.com_nodegroup_image="ami-0c9e890fe2f66306b"</code><br><code>eks.amazonaws.com_sourceLaunchTemplateId="lt-054e2ac7284ae8627"</code><br><code>eks.amazonaws.com_sourceLaunchTemplateVersion="1"</code><br><code>failure_domain_beta_kubernetes_io_region="us-east-1"</code><br><code>failure_domain_beta_kubernetes_io_zone="us-east-1a"</code> | 38.686s ago | 47.572ms        |       |

# Desplegar Grafana

*kubectl create namespace grafana*

*Crear el archivo yaml siguiendo ejemplo del PIN, en la ruta sugerida*

```
helm install grafana grafana/grafana \
  --namespace grafana \
  --set persistence.storageClassName="gp2" \
  --set persistence.enabled=true \
  --set adminPassword='EKS!sAWSome' \
  --values ${HOME}/environment/grafana/grafana.yaml \
  --set service.type=LoadBalancer
```

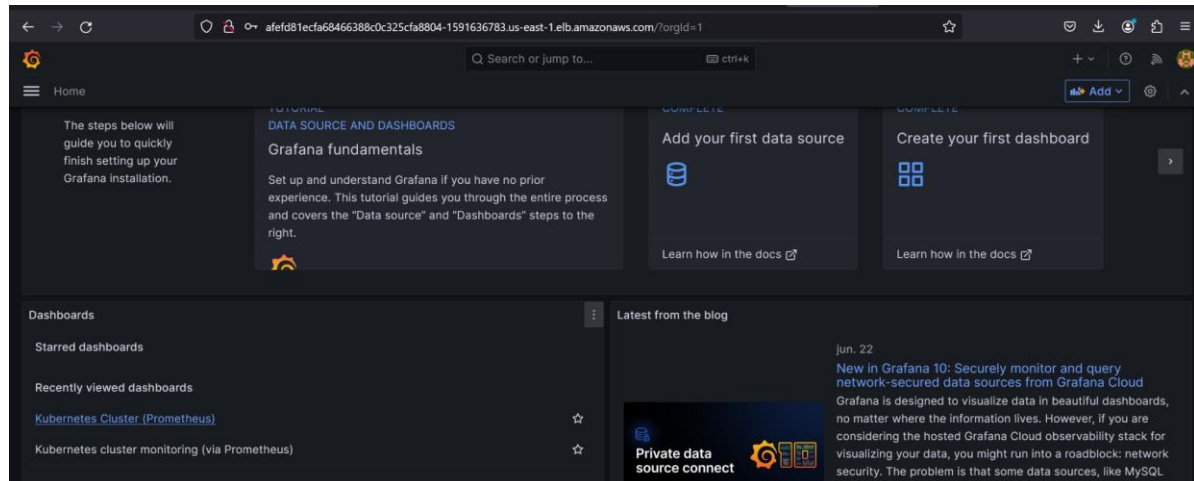
```
ubuntu@ip-172-31-86-150:~/environment/grafana$ kubectl get all -n grafana
NAME                                READY   STATUS    RESTARTS   AGE
pod/grafana-574676d57f-vf9dz        1/1     Running   0           70m

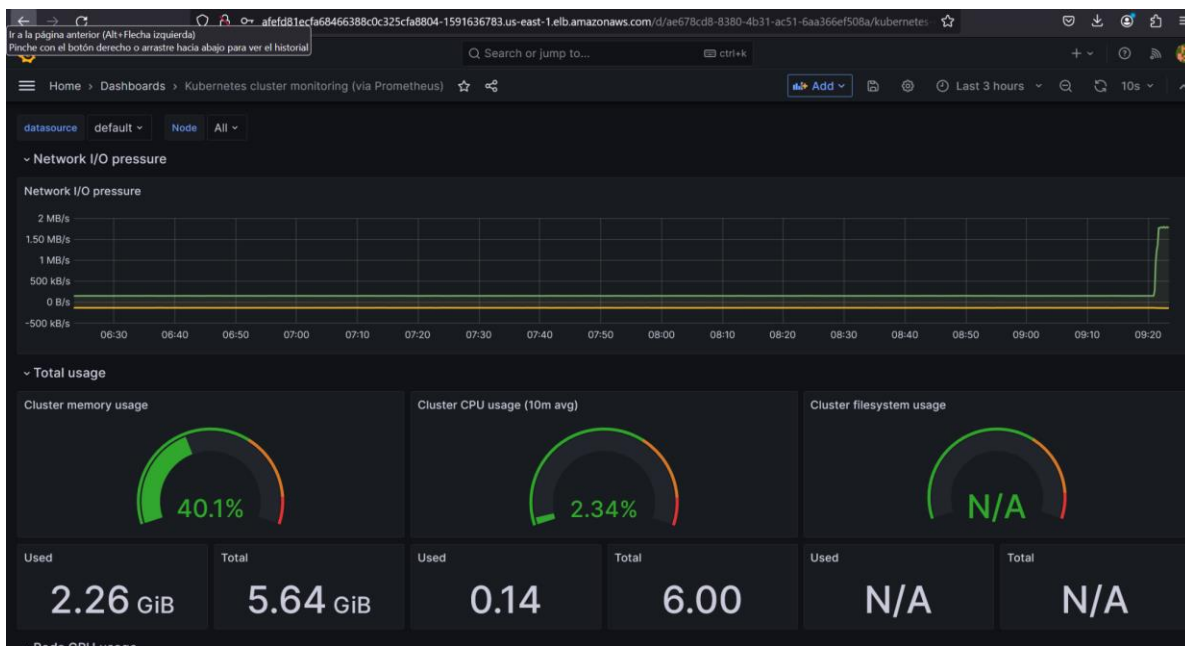
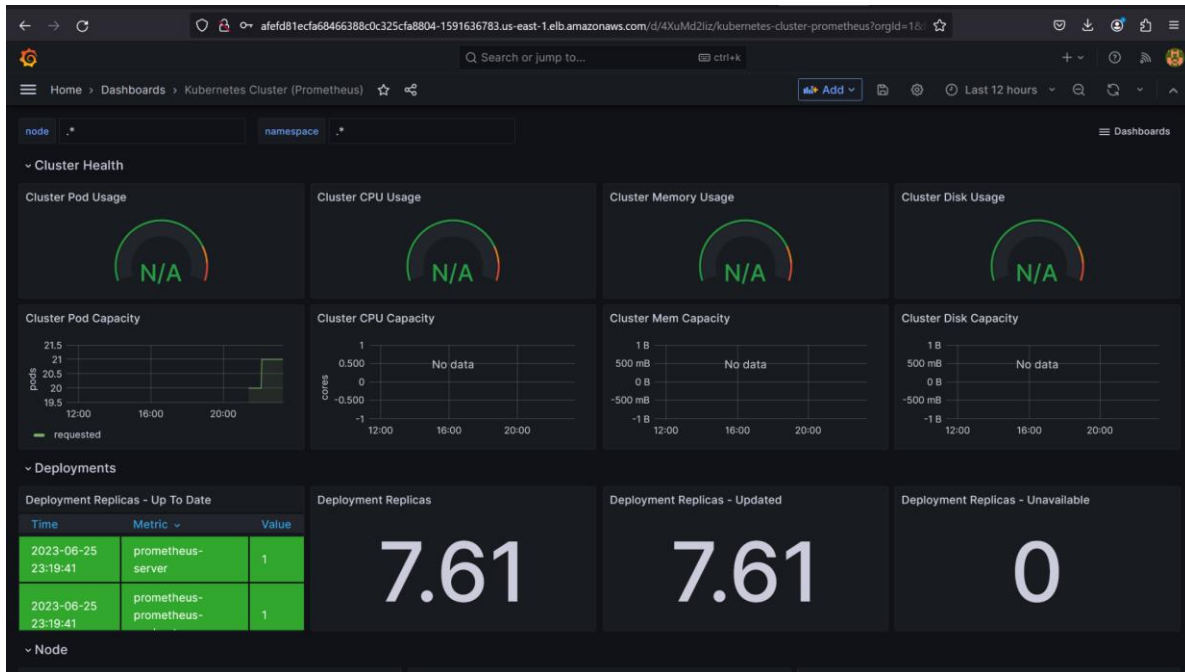
NAME                                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/grafana                     LoadBalancer  10.100.183.93    afefd81ecfa68466388c0c325cfa8804-1591636783.us-east-1.elb.amazonaws.com  80:30671/TCP     70m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana             1/1     1             1           70m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana-574676d57f  1         1         1       70m
ubuntu@ip-172-31-86-150:~/environment/grafana$
```

Acceso y adición de Dashboards (3119 y 6417):





# Cleanup de recursos

```
helm uninstall prometheus --namespace prometheus
```

```
kubectl delete ns prometheus
```

```
helm uninstall grafana --namespace grafana
```

```
kubectl delete ns grafana
```

```
rm -rf ${HOME}/environment/grafana
```

```
ubuntu@ip-172-31-86-150: ~
```

```
ubuntu@ip-172-31-86-150:~$ kubectl delete ns prometheus
namespace "prometheus" deleted
ubuntu@ip-172-31-86-150:~$ helm uninstall grafana --namespace grafana
release "grafana" uninstalled
ubuntu@ip-172-31-86-150:~$ kubectl delete ns grafana
namespace "grafana" deleted
ubuntu@ip-172-31-86-150:~$ rm -rf ${HOME}/environment/grafana
ubuntu@ip-172-31-86-150:~$
```

## Borrar Cluster EKS

```
eksctl delete cluster --name eks-mundos-e
```

```
ubuntu@ip-172-31-86-150: ~
```

```
ubuntu@ip-172-31-86-150:~$ eksctl delete cluster --name eks-mundos-e
2023-06-26 12:53:01 [!] deleting EKS cluster "eks-mundos-e"
2023-06-26 12:53:01 [!] stack's status of nodegroup named eksctl-eks-mundos-e-nodegroup-ng-0b843954 is DELETE_FAILED
2023-06-26 12:53:01 [!] deleted 0 Fargate profile(s)
2023-06-26 12:53:02 [✓] kubeconfig has been updated
2023-06-26 12:53:02 [!] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2023-06-26 12:53:03 [!]
2 sequential tasks: {
  2 sequential sub-tasks: {
    2 parallel sub-tasks: {
      2 sequential sub-tasks: {
        delete IAM role for serviceaccount "kube-system/eks-csi-controller-irsa",
        delete serviceaccount "kube-system/eks-csi-controller-irsa",
      },
      2 sequential sub-tasks: {
        delete IAM role for serviceaccount "kube-system/aws-node",
        delete serviceaccount "kube-system/aws-node",
      },
    },
    delete IAM OIDC provider,
  }, delete cluster control plane "eks-mundos-e" [async]
}
2023-06-26 12:53:03 [!] will delete stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-aws-node"
2023-06-26 12:53:03 [!] waiting for stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-aws-node" to get deleted
2023-06-26 12:53:03 [!] waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-aws-node"
2023-06-26 12:53:03 [!] will delete stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-eks-csi-controller-irsa"
2023-06-26 12:53:03 [!] waiting for stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-eks-csi-controller-irsa" to get deleted
2023-06-26 12:53:03 [!] waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-eks-csi-controller-irsa"
2023-06-26 12:53:03 [!] waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-aws-node"
2023-06-26 12:53:03 [!] deleted serviceaccount "kube-system/aws-node"
2023-06-26 12:53:03 [!] waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-eks-csi-controller-irsa"
2023-06-26 12:53:03 [!] deleted serviceaccount "kube-system/eks-csi-controller-irsa"
2023-06-26 12:53:03 [!] will delete stack "eksctl-eks-mundos-e-cluster"
2023-06-26 12:53:04 [!] found the following undeleted stacks: eksctl-eks-mundos-e-nodegroup-ng-0b843954
Error: failed to delete all resources
ubuntu@ip-172-31-86-150:~$
```



