Proyecto Integrador PIN2204

Integrantes:

Contenido

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.

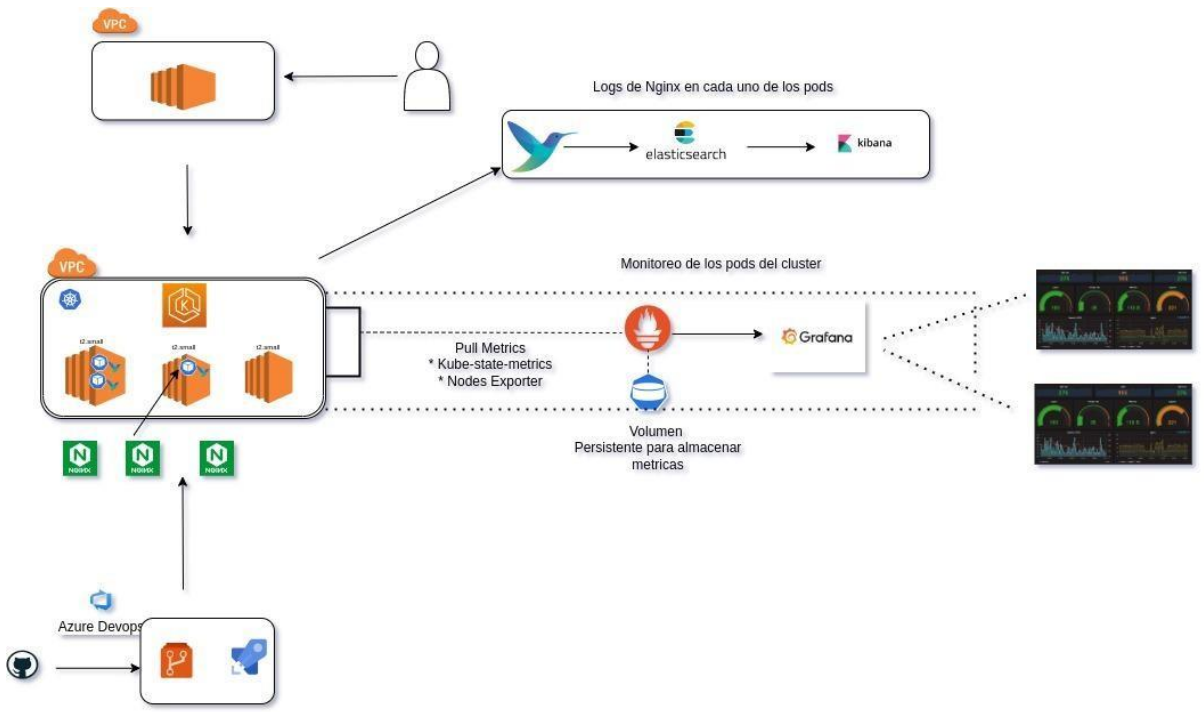
Durante la primera parte nos centramos en la creación de una instancia de EC2 en AWS para poder desde allí realizar todas las tareas necesarias.

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 el el cluster

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



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

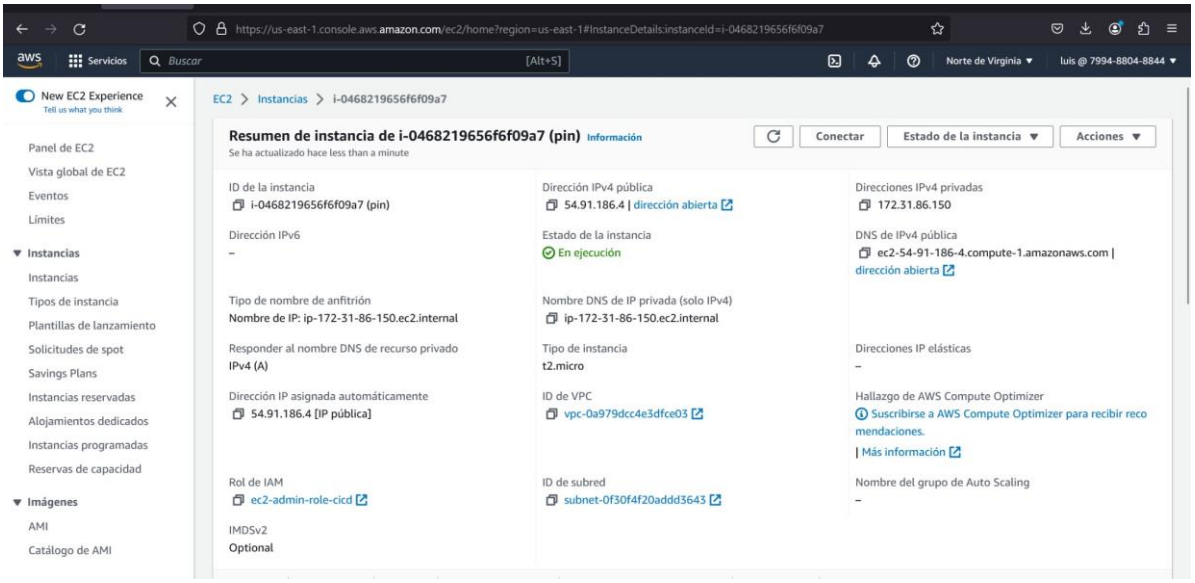
Crear instancia EC2

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)

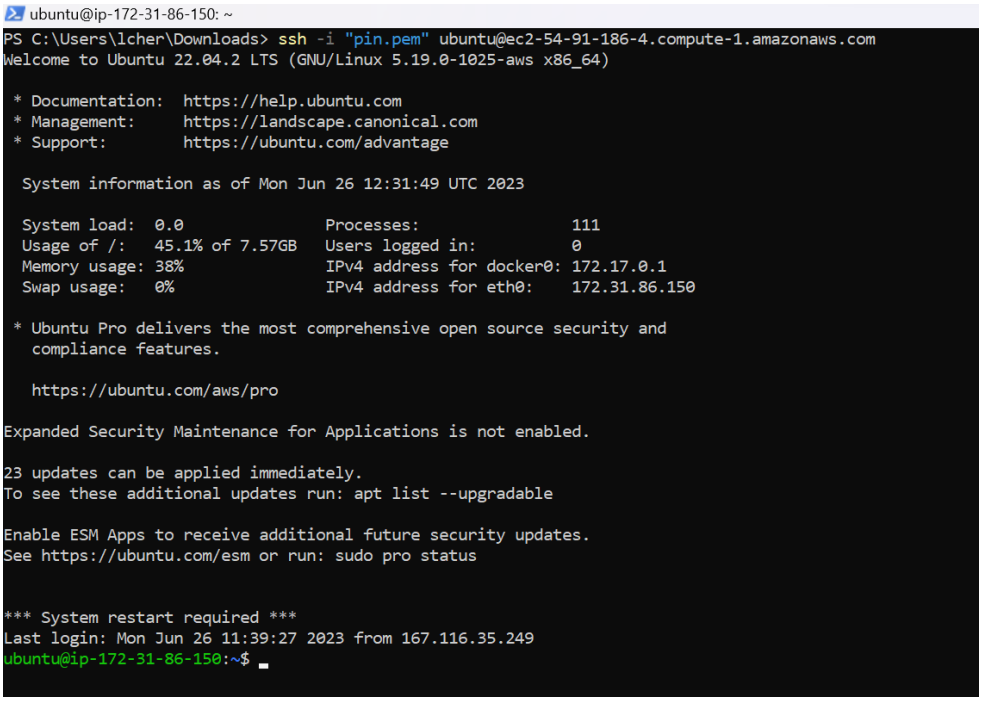
Se crea un par de claves para poder conectarse, llamadas “pin” en formato pem

Data de Instancia:



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

Conectar con instancia por SSH:



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:

Instalar herramientas de monitoreo de pods

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

Deploy driver

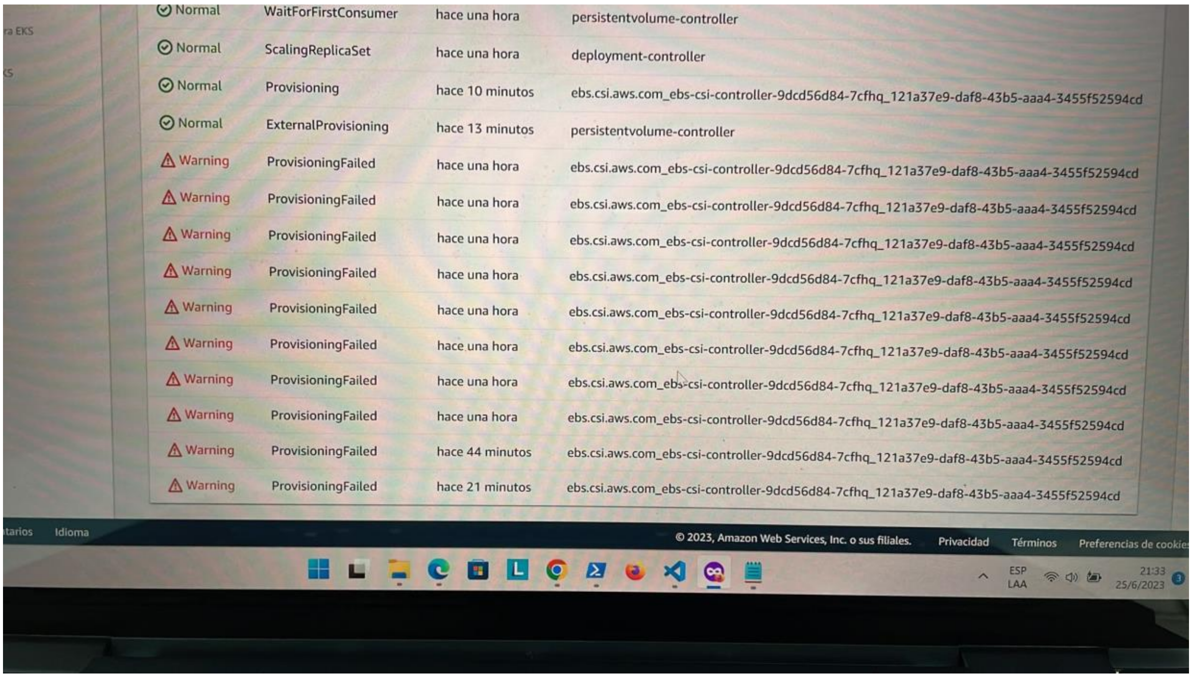
You may deploy the EBS CSI driver via Kustomize, Helm, or as an Amazon EKS managed add-on.

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

© 2023, Amazon Web Services, Inc. o sus filiales. Privacidad Términos Preferencias de cookies

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

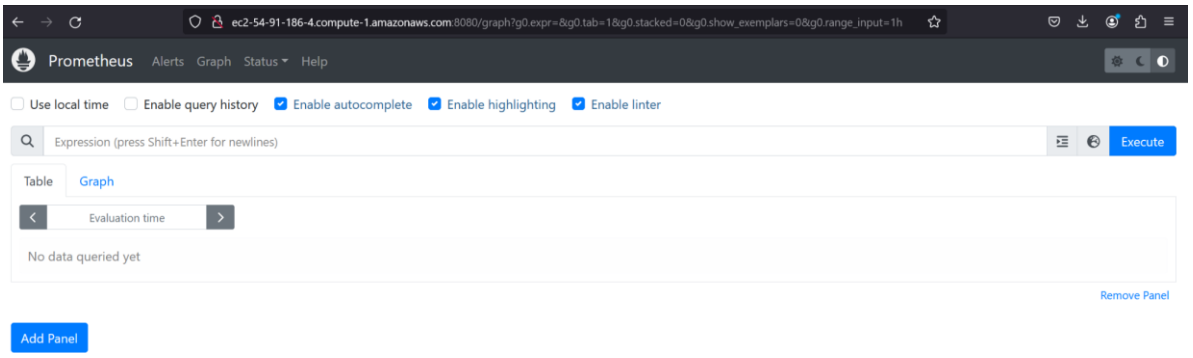
Prometheus corriendo:

```
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
```

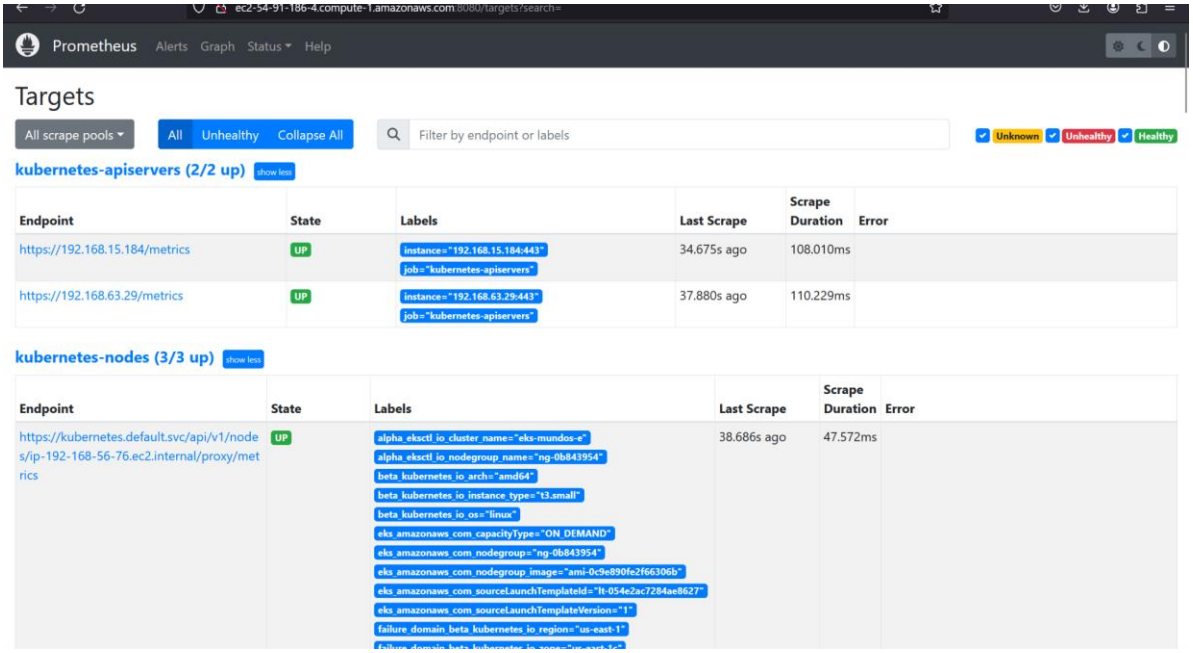
Port forward:

```
statefulset.apps/prometheus-alertmanager   1/1     93m
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:

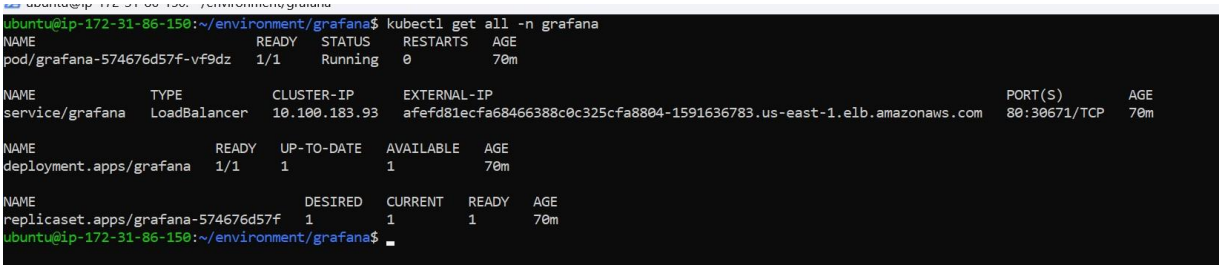

Prometheus
Alerts
Graph
Status
Help
Targets
All scrape pools
All
Unhealthy
Collapse All
Filter by endpoint or labels
Unknown
Unhealthy
Healthy
kubernetes-apiservers (2/2 up)
show less
Endpoint
State
Labels
Last Scrape
Scrape Duration
Error
https://192.168.15.184/metrics
UP
instance="192.168.15.184:443"
job="kubernetes-apiservers"
34.675s ago
108.010ms
https://192.168.63.29/metrics
UP
instance="192.168.63.29:443"
job="kubernetes-apiservers"
37.880s ago
110.229ms
kubernetes-nodes (3/3 up)
show less
Endpoint
State
Labels
Last Scrape
Scrape Duration
Error
https://kubernetes.default.svc/api/v1/nodes/ip-192-168-56-76.ec2.internal/proxy/metrics
UP
alpha_eksctl_io_cluster_name="eks-mundos-e"
alpha_eksctl_io_nodegroup_name="ng-0b843954"
beta_kubernetes_io_arch="amd64"
beta_kubernetes_io_instance_type="t3.small"
beta_kubernetes_io_os="linux"
eks_amazonaws_com_capacityType="ON_DEMAND"
eks_amazonaws_com_nodegroup="ng-0b843954"
eks_amazonaws_com_nodegroup_image="ami-0c9e890fe2f66306b"
eks_amazonaws_com_sourceLaunchTemplateId="lt-054e2ac7284ae8627"
eks_amazonaws_com_sourceLaunchTemplateVersion="1"
failure_domain_beta_kubernetes_io_region="us-east-1"
38.686s ago
47.572ms

Desplegar Grafana

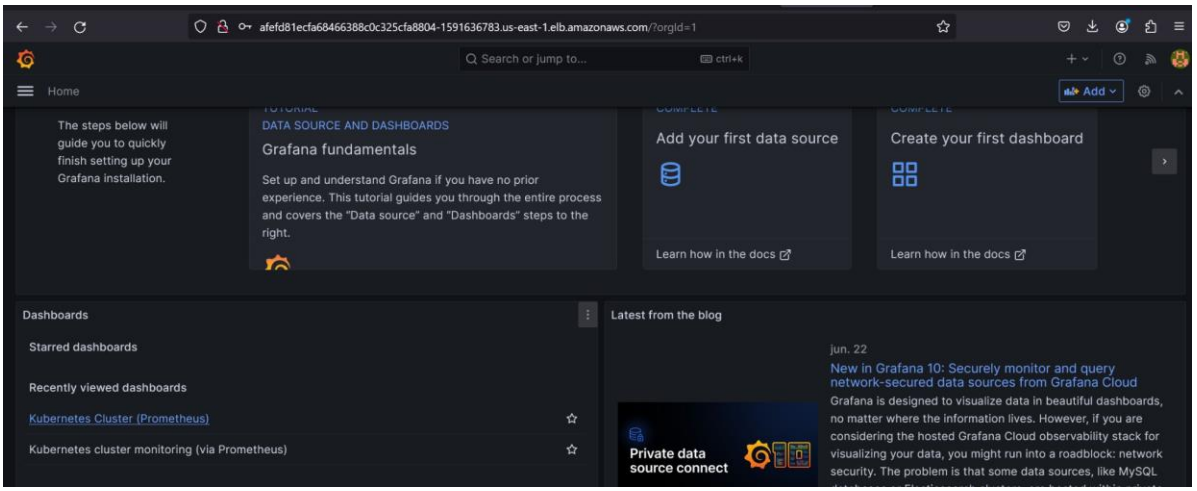
kubectl create namespace grafana

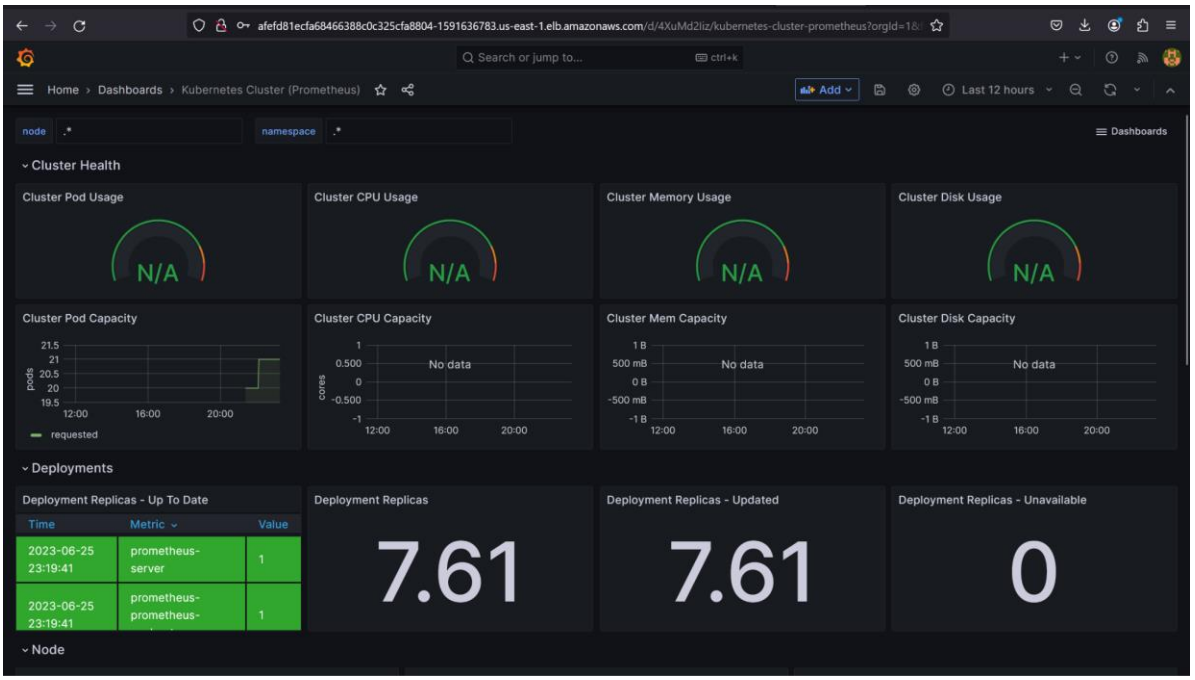
Crear el archivo yaml siguiendo ejemplo del PIN, en la ruta sugerida

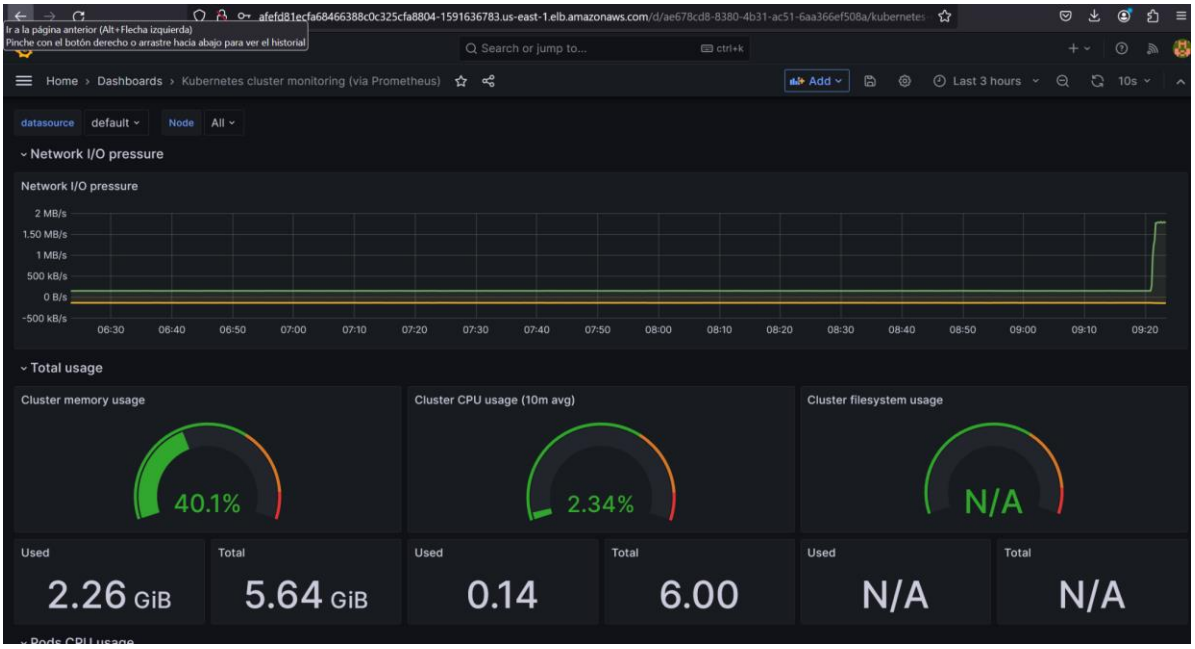
*helm install grafana grafana/grafana *
*--namespace grafana *
*--set persistence.storageClassName="gp2" *
*--set persistence.enabled=true *
*--set adminPassword='EKS!sAWSome' *
*--values ${HOME}/environment/grafana/grafana.yaml *
--set service.type=LoadBalancer



Acceso y adición de Dashboards (3119 y 6417):




Home › Dashboards › Kubernetes Cluster (Prometheus)
Add
Last 12 hours
node .*
namespace .*
Dashboards
Cluster Health
Cluster Pod Usage
N/A
Cluster CPU Usage
N/A
Cluster Memory Usage
N/A
Cluster Disk Usage
N/A
Cluster Pod Capacity
pods
requested
Cluster CPU Capacity
cores
No data
Cluster Mem Capacity
No data
Cluster Disk Capacity
No data
Deployments
Deployment Replicas - Up To Date
Time
Metric
Value
2023-06-25 23:19:41
prometheus-server
1
2023-06-25 23:19:41
prometheus-prometheus-
1
Deployment Replicas
7.61
Deployment Replicas - Updated
7.61
Deployment Replicas - Unavailable
0
Node


Ir a la página anterior (Alt+Flecha izquierda)
Pinche con el botón derecho o arrastre hacia abajo para ver el historial
Home › Dashboards › Kubernetes cluster monitoring (via Prometheus)
Add
Last 3 hours
10s
datasource default
Node All
Network I/O pressure
Network I/O pressure
Total usage
Cluster memory usage
40.1%
Cluster CPU usage (10m avg)
2.34%
Cluster filesystem usage
N/A
Used
2.26 GiB
Total
5.64 GiB
Used
0.14
Total
6.00
Used
N/A
Total
N/A

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:~$ 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:~$ eksctl delete cluster --name eks-mundos-e
2023-06-26 12:53:01 [ℹ]  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 [ℹ]  deleted 0 Fargate profile(s)
2023-06-26 12:53:02 [✔]  kubeconfig has been updated
2023-06-26 12:53:02 [ℹ]  cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2023-06-26 12:53:03 [ℹ]
2 sequential tasks: {
    2 sequential sub-tasks: {
        2 parallel sub-tasks: {
            2 sequential sub-tasks: {
                delete IAM role for serviceaccount "kube-system/ebs-csi-controller-irsa",
                delete serviceaccount "kube-system/ebs-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 [ℹ]  will delete stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-aws-node"
2023-06-26 12:53:03 [ℹ]  waiting for stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-aws-node" to get deleted
2023-06-26 12:53:03 [ℹ]  waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-aws-node"
2023-06-26 12:53:03 [ℹ]  will delete stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-irsa"
2023-06-26 12:53:03 [ℹ]  waiting for stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-irsa" to get deleted
2023-06-26 12:53:03 [ℹ]  waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-irsa"
2023-06-26 12:53:33 [ℹ]  waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-aws-node"
2023-06-26 12:53:33 [ℹ]  deleted serviceaccount "kube-system/aws-node"
2023-06-26 12:53:33 [ℹ]  waiting for CloudFormation stack "eksctl-eks-mundos-e-addon-iamserviceaccount-kube-system-ebs-csi-controller-irsa"
2023-06-26 12:53:33 [ℹ]  deleted serviceaccount "kube-system/ebs-csi-controller-irsa"
2023-06-26 12:53:34 [ℹ]  will delete stack "eksctl-eks-mundos-e-cluster"
2023-06-26 12:53:34 [!]  found the following undeleted stacks: eksctl-eks-mundos-e-nodegroup-ng-0b843954
Error: failed to delete all resources
ubuntu@ip-172-31-86-150:~$
```