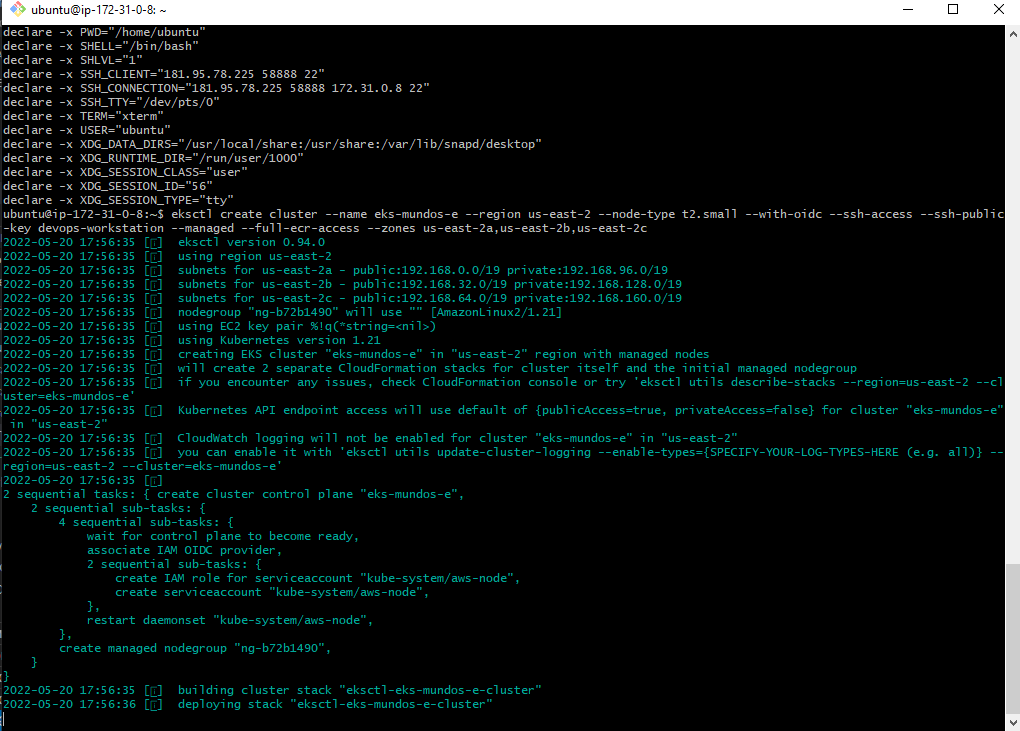
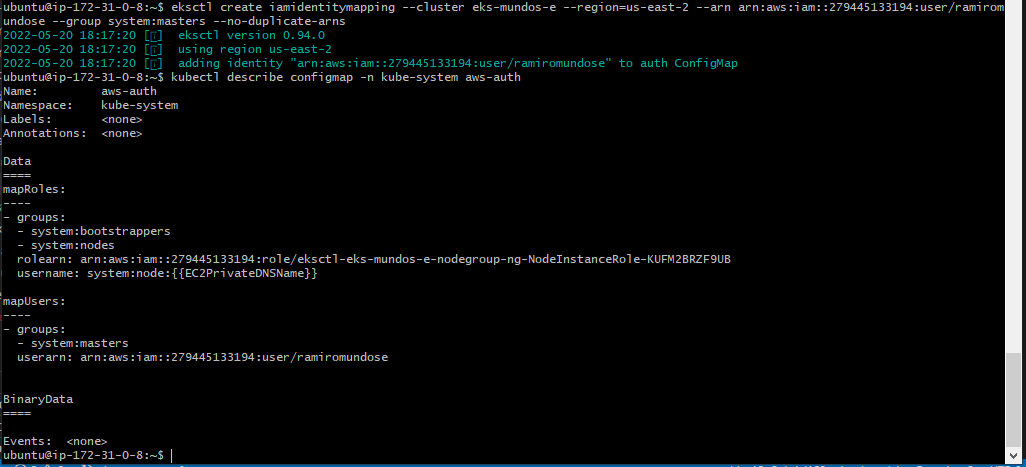
MundosE DevOps

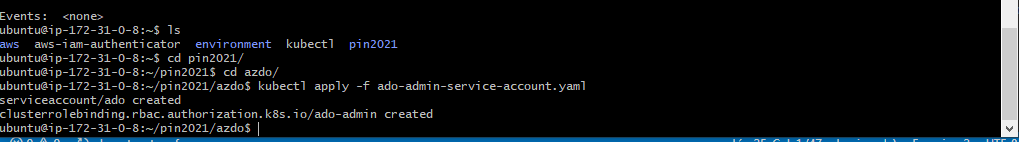
Creacion de Cluster EKS



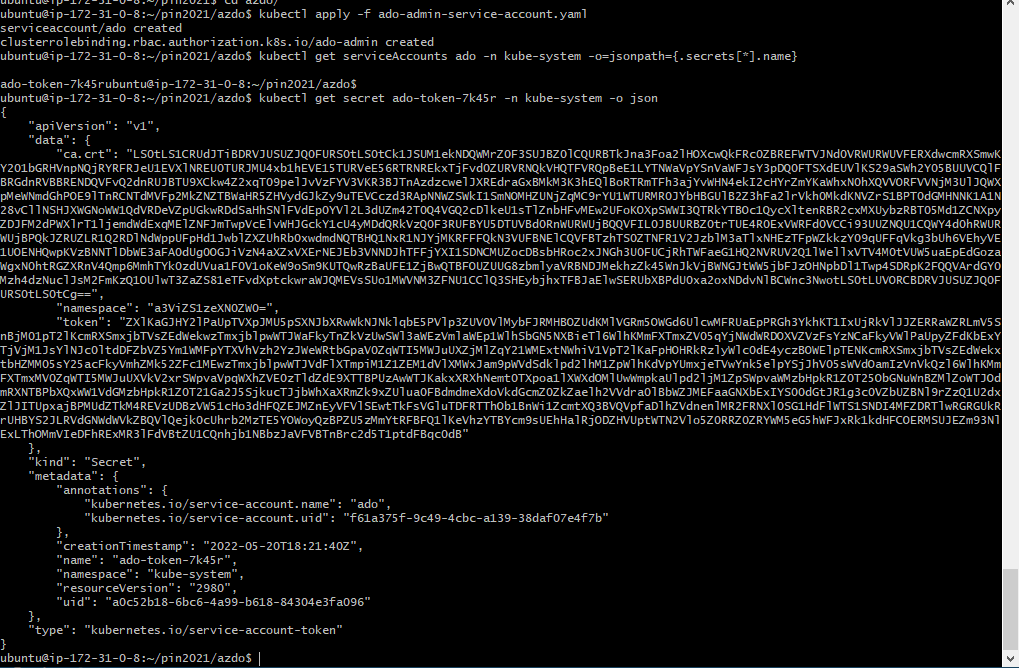
Se mapea la identidad



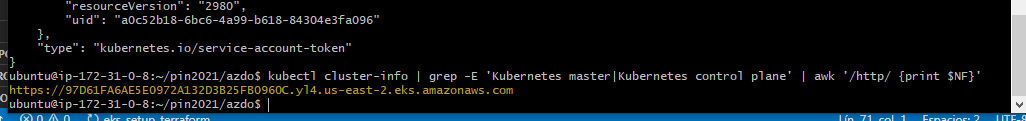
Ahora se trabaja con AzDO, creamos una service account para poder conectarnos al cluster de aws



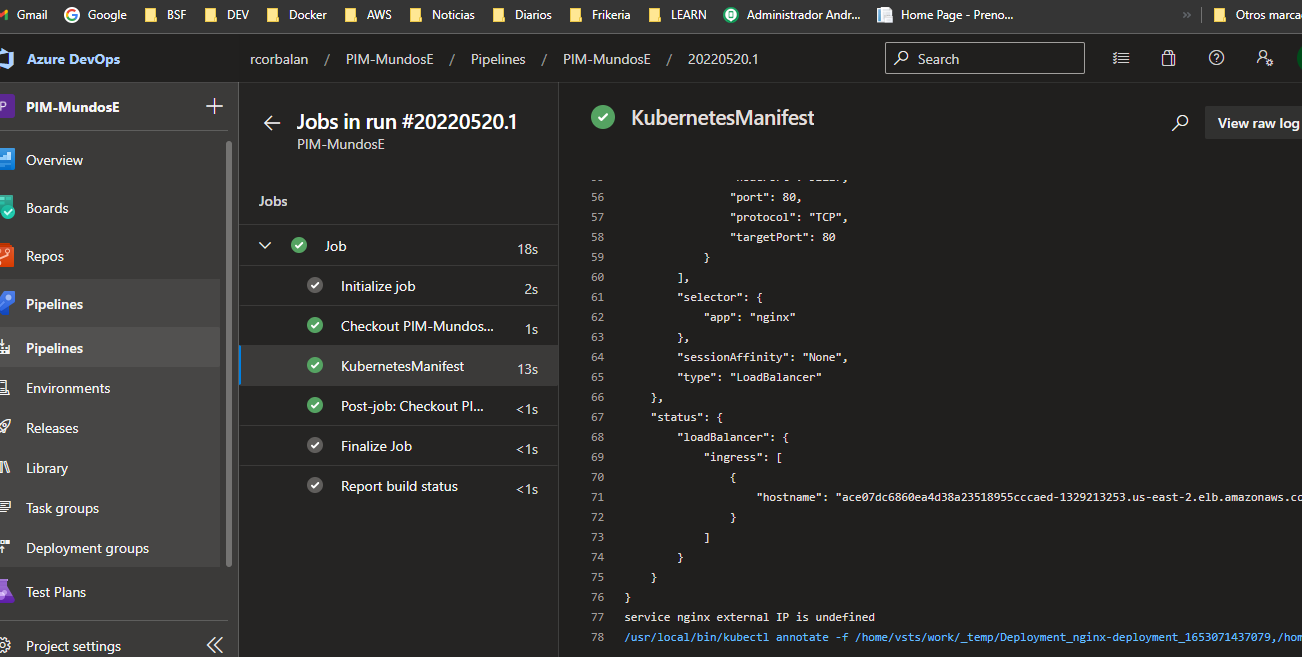
Obtenemos el secret y el json de configuracion



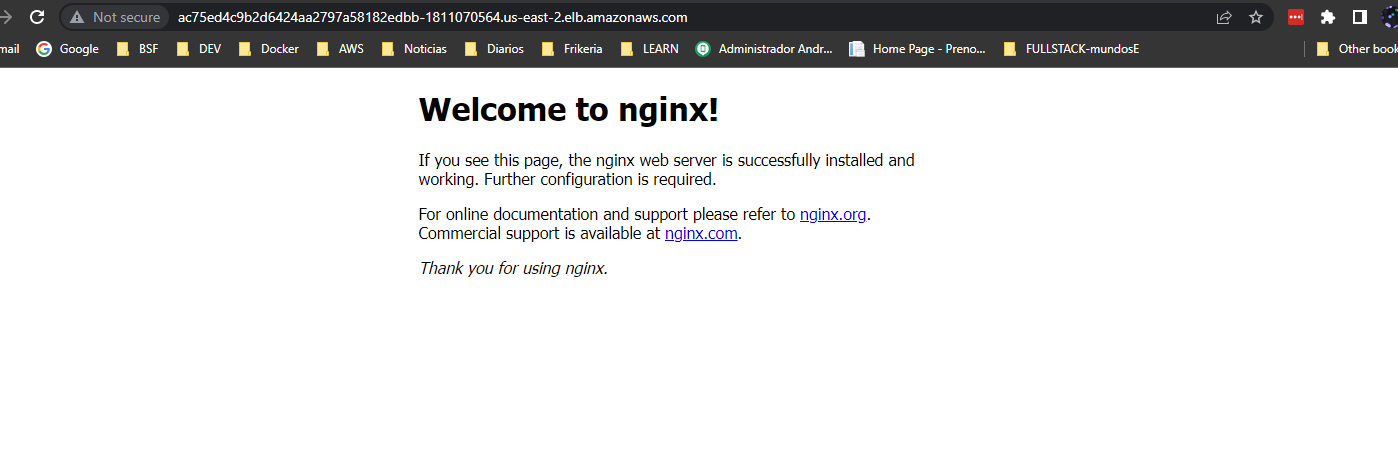
Obtenemos la url de la service account



Con estos datos configuramos la service connection en AzDO y ejecutamos el pipeline



Observamos que se creo en contenedor de NginX



Creamos variables de entorno para proseguir con la instalacion de las herramientas de monitoreo (ElasticSearch, Fluentbit y Kibana)

export AWS\_REGION='us-east-2'

export ACCOUNT\_ID=279445133194

export ES\_DOMAIN\_NAME="eksworkshop-logging"

**# Elasticsearch version**

export ES\_VERSION="7.4"

**# kibana admin user**

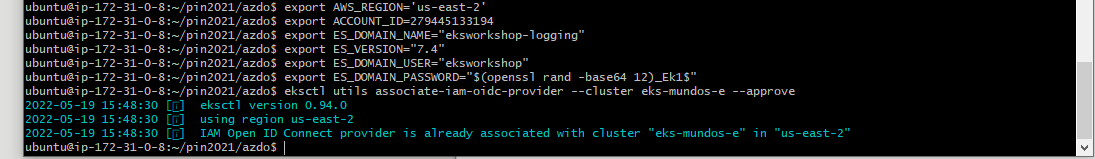
export ES\_DOMAIN\_USER="eksworkshop"

**# kibana admin password**

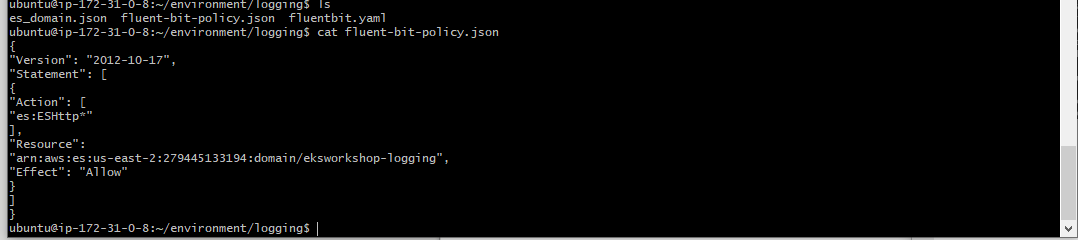
export ES\_DOMAIN\_PASSWORD="$(openssl rand -base64 12)\_Ek1$"

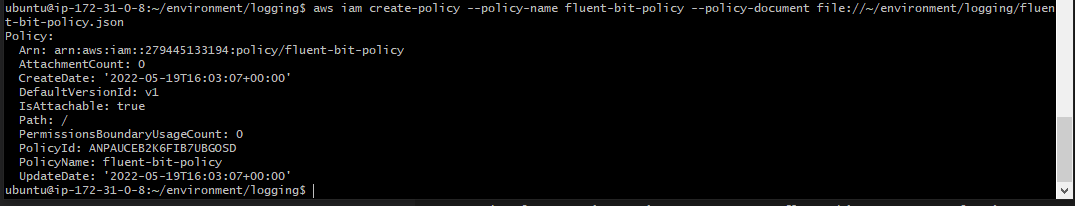
**# Configurar OpenID Connect**

eksctl utils associate-iam-oidc-provider --cluster eks-mundos-e --approve



Crear IAM policy con AWS Cli

aws iam create-policy --policy-name fluent-bit-policy --policy-document file://~/environment/logging/fluent-bit-policy.json



Creamos namespace de logging

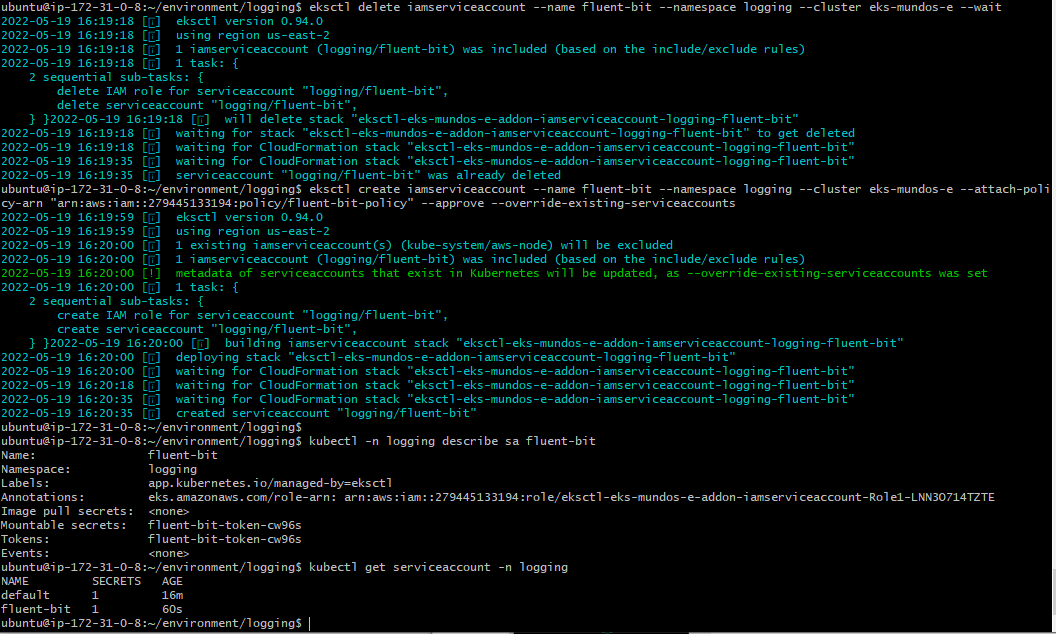
kubectl create namespace logging



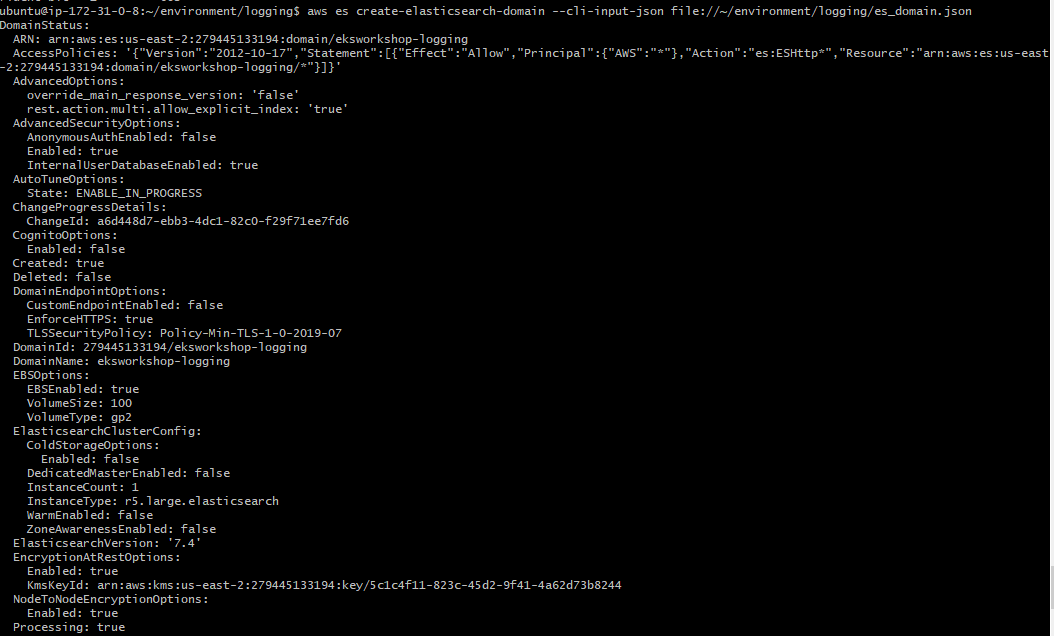
Creamos cuenta de servicio

eksctl create iamserviceaccount --name fluent-bit --namespace logging --cluster eks-mundos-e --attach-policy-arn "arn:aws:iam::279445133194:policy/fluent-bit-policy" --approve --override-existing-serviceaccounts

En este punto tuve que borrar la cuenta ya estaba creada de un laboratorio anterior y no nos dejaba asignarla al namespace logging



Creamos dominio de elasticsearch (opensearch). Encontramos que algunos tags habian cambiado con el cambio de version a OpenSearch



Para chequear que este creado el dominio. En este punto observo un problema con usar variables de entorno en sentencias de linea de comando, no nos tomaba la variable ${ES\_DOMAIN\_NAME}

if [ $(aws es describe-elasticsearch-domain --domain-name eksworkshop-logging --query 'DomainStatus.Processing') == "false" ]

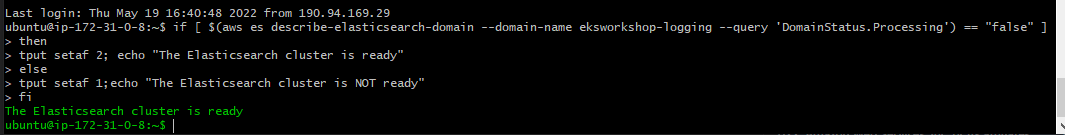
then

tput setaf 2; echo "The Elasticsearch cluster is ready"

else

tput setaf 1;echo "The Elasticsearch cluster is NOT ready"

fi



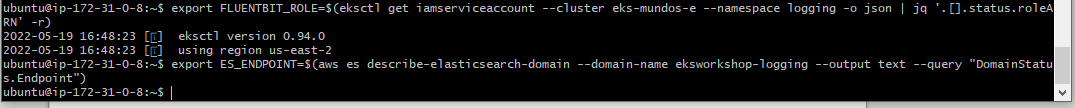
Configuramos acceso a elastic

**# We need to retrieve the Fluent Bit Role ARN**

export FLUENTBIT\_ROLE=$(eksctl get iamserviceaccount --cluster eks-mundos-e --namespace logging -o json | jq '.[].status.roleARN' -r)

**# Get the Elasticsearch Endpoint**

export ES\_ENDPOINT=$(aws es describe-elasticsearch-domain --domain-name eksworkshop-logging --output text --query "DomainStatus.Endpoint")



**# Update the Elasticsearch internal database**

curl -sS -u "eksworkshop:7KR20Z6kUEHE2F96\_Ek1\$" \

    -X PATCH \

    https://search-eksworkshop-logging-wxy7apc3i252g2chpzmfuuzr6q.us-east-2.es.amazonaws.com/\_opendistro/\_security/api/rolesmapping/all\_access?pretty \

    -H 'Content-Type: application/json' \

    -d'

[

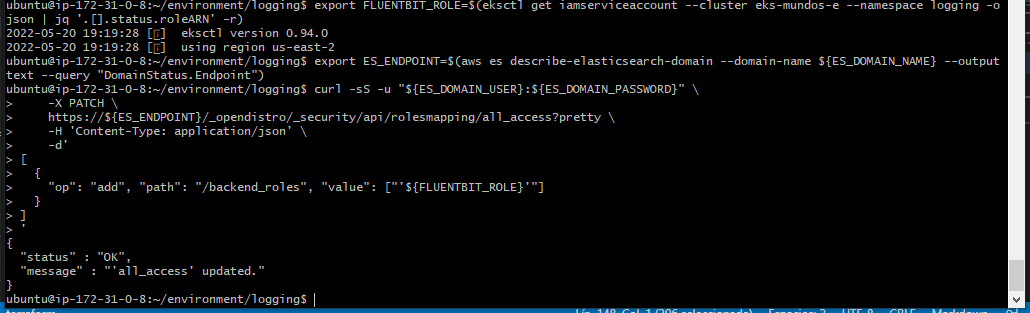
  {

    "op": "add", "path": "/backend\_roles", "value": ["'arn:aws:iam::279445133194:role/eksctl-eks-mundos-e-addon-iamserviceaccount-Role1-LNN3O714TZTE'"]

  }

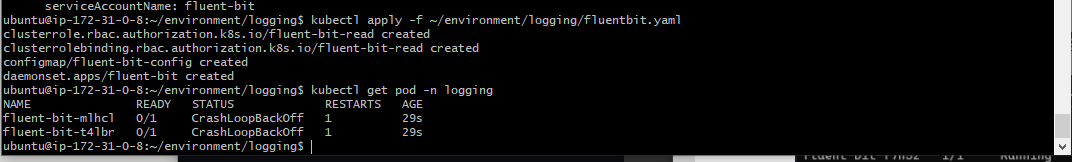
]

'



Desplegamos contenedor de fluentbit

kubectl apply -f ~/environment/logging/fluentbit.yaml

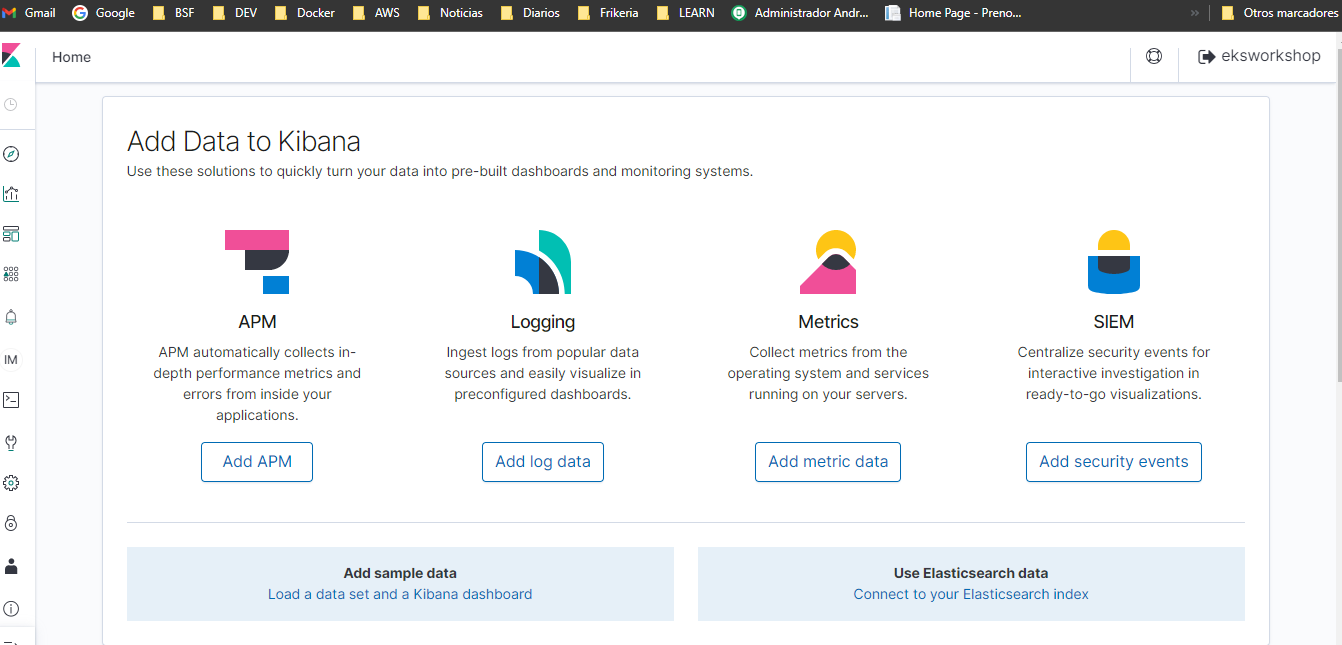


Obtenemos la informacion de kibana.

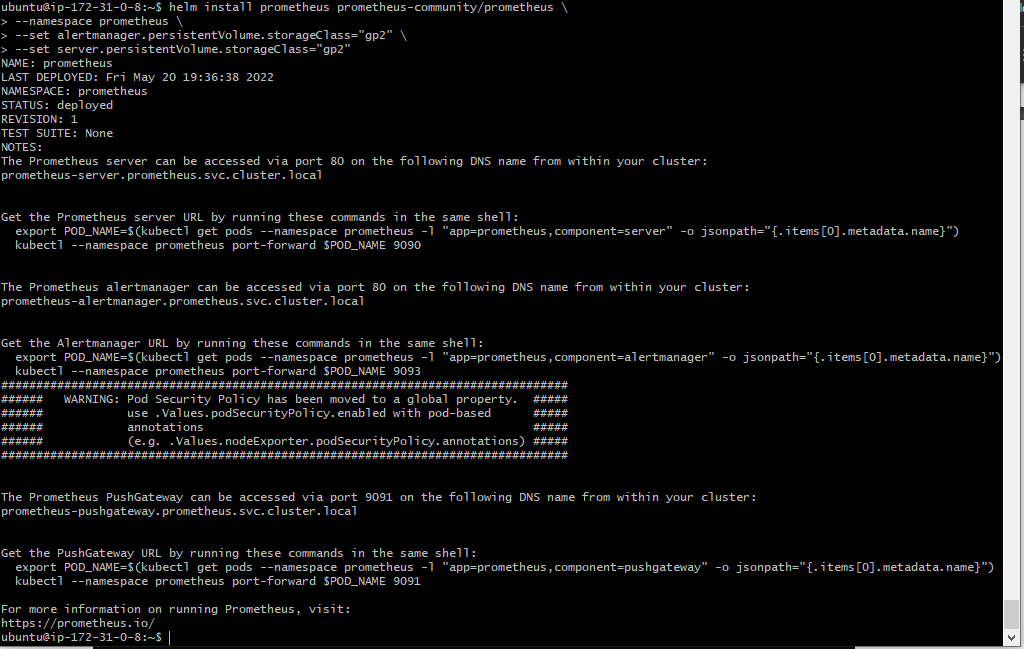
echo "Kibana URL: https://${ES\_ENDPOINT}/\_plugin/kibana/

Kibana user: ${ES\_DOMAIN\_USER}

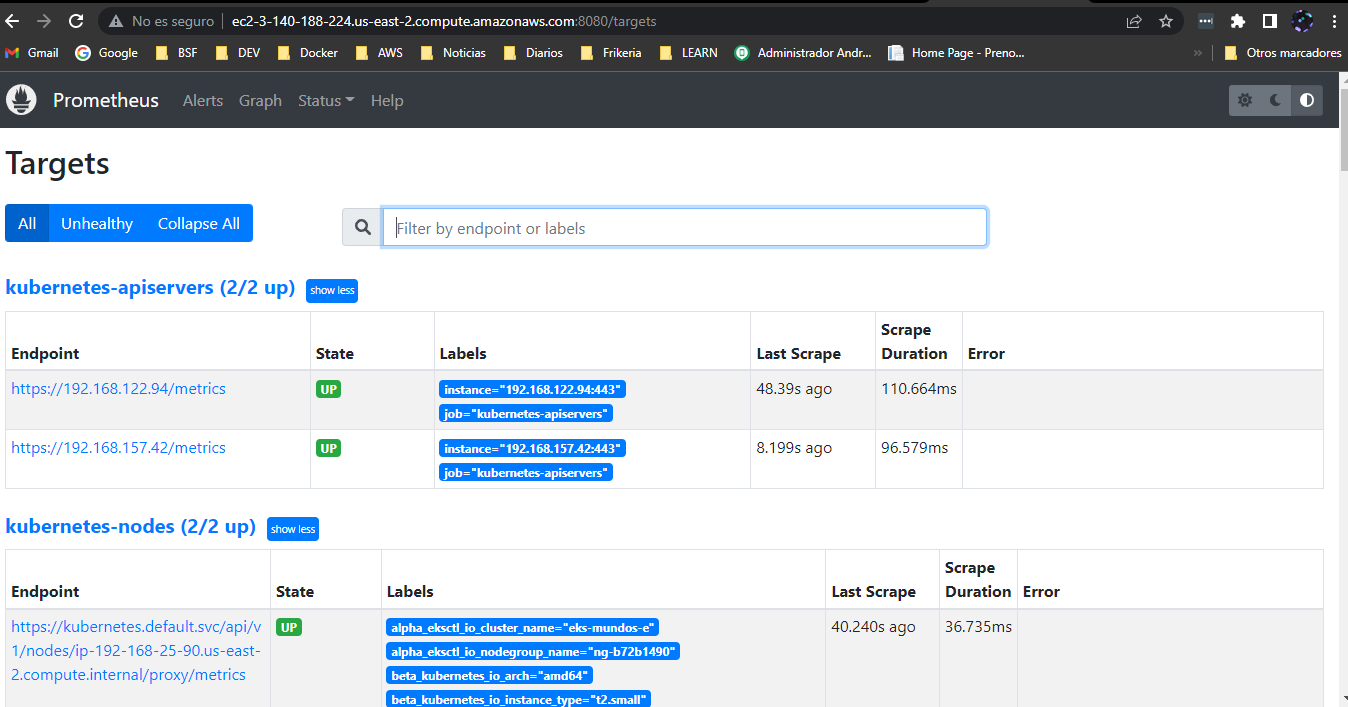
Kibana password: ${ES\_DOMAIN\_PASSWORD}"



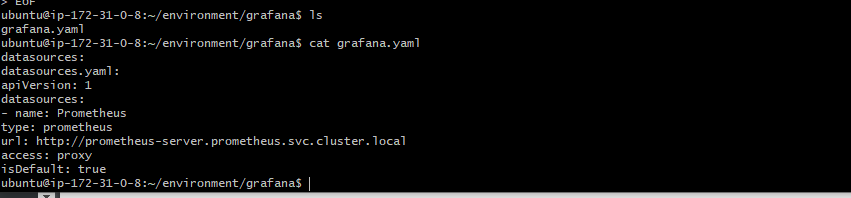
Desplegamos Prometheus

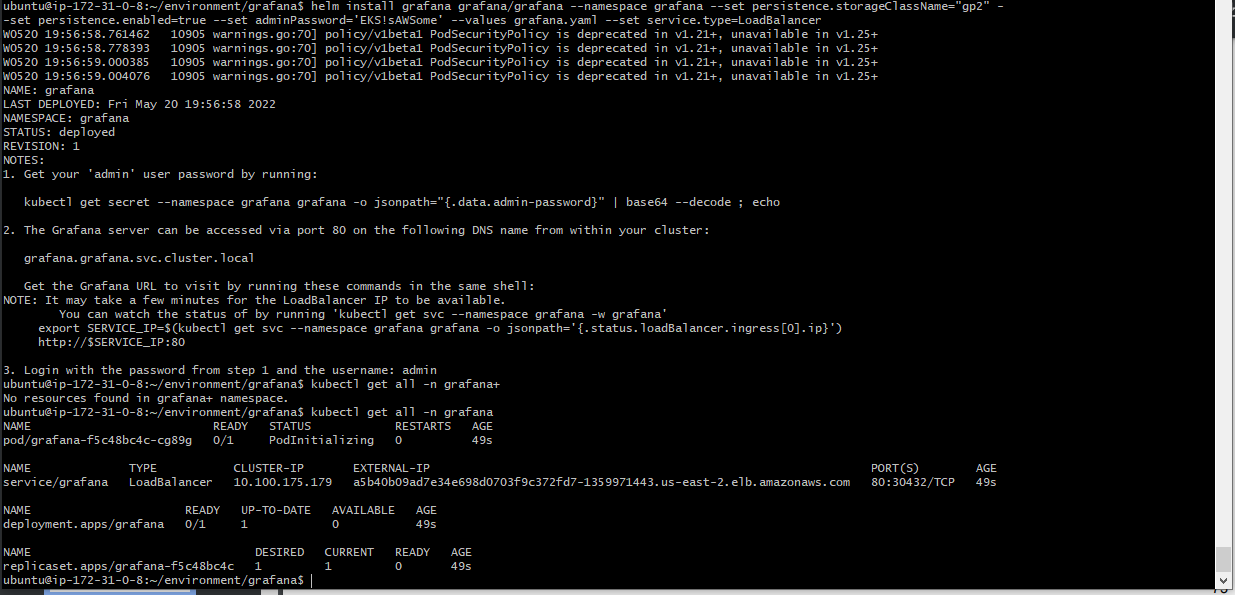


Hay que editar las rutas del security group de la instancia y permitir el 8080 inbound

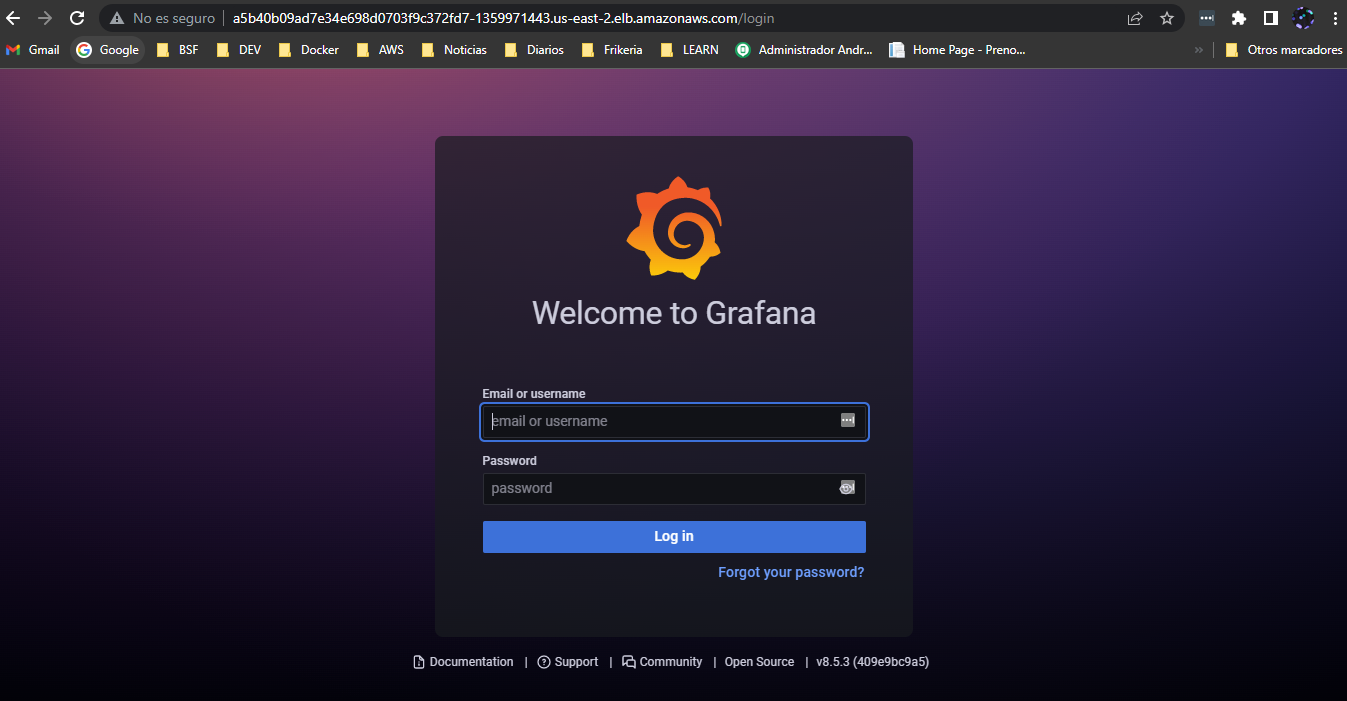


Desplegamos Grafana

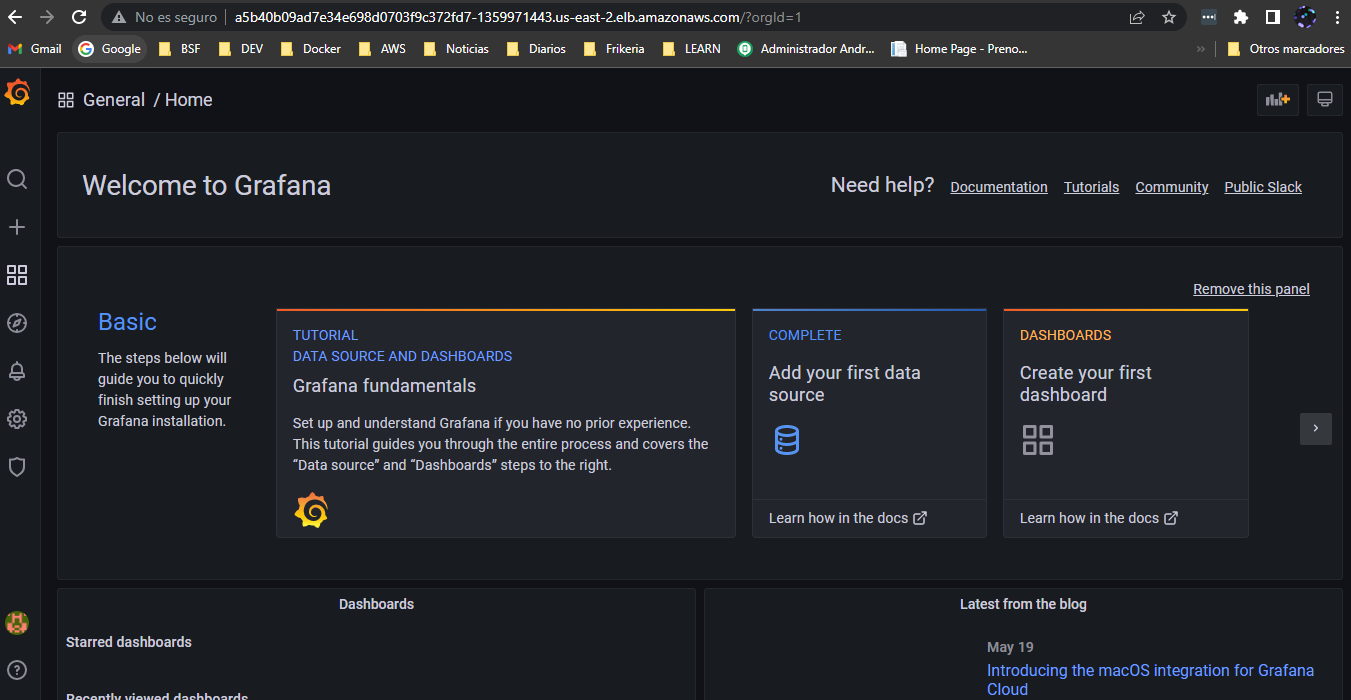




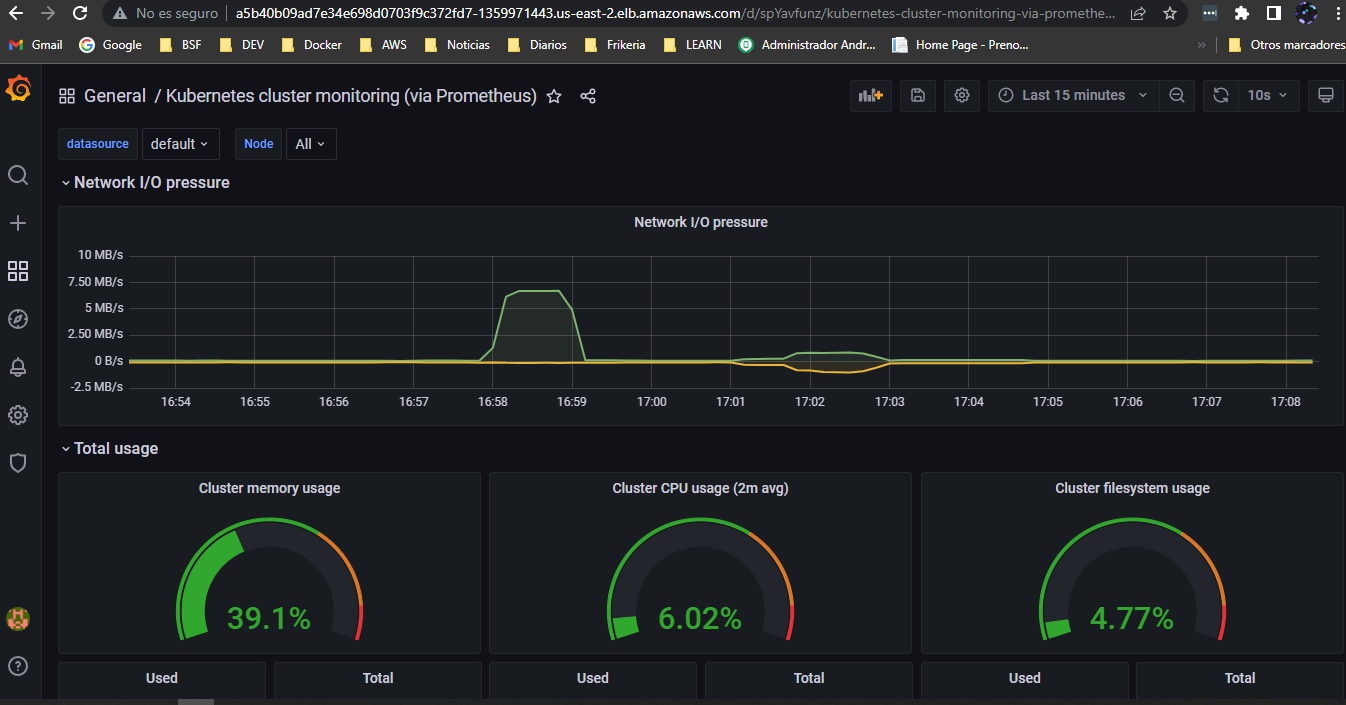
Url Grafana



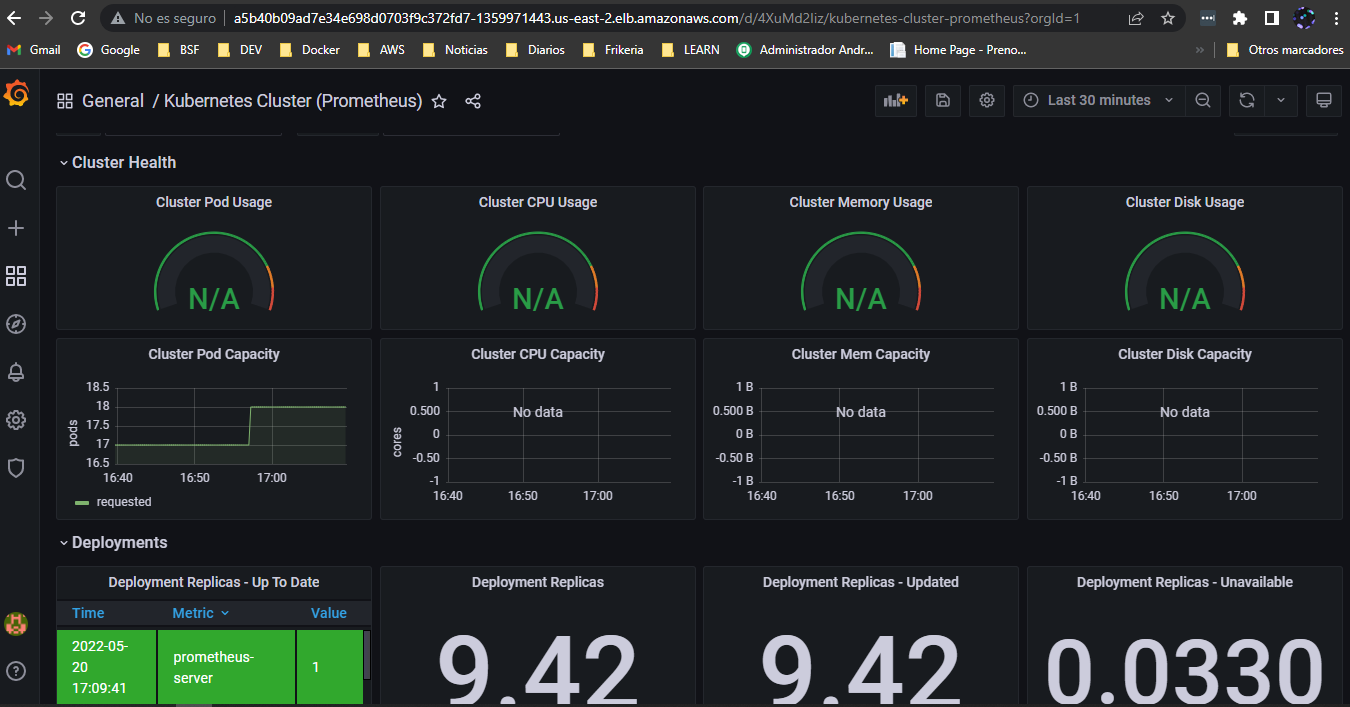
Ingresamos con admin y con la password que se le configure antes

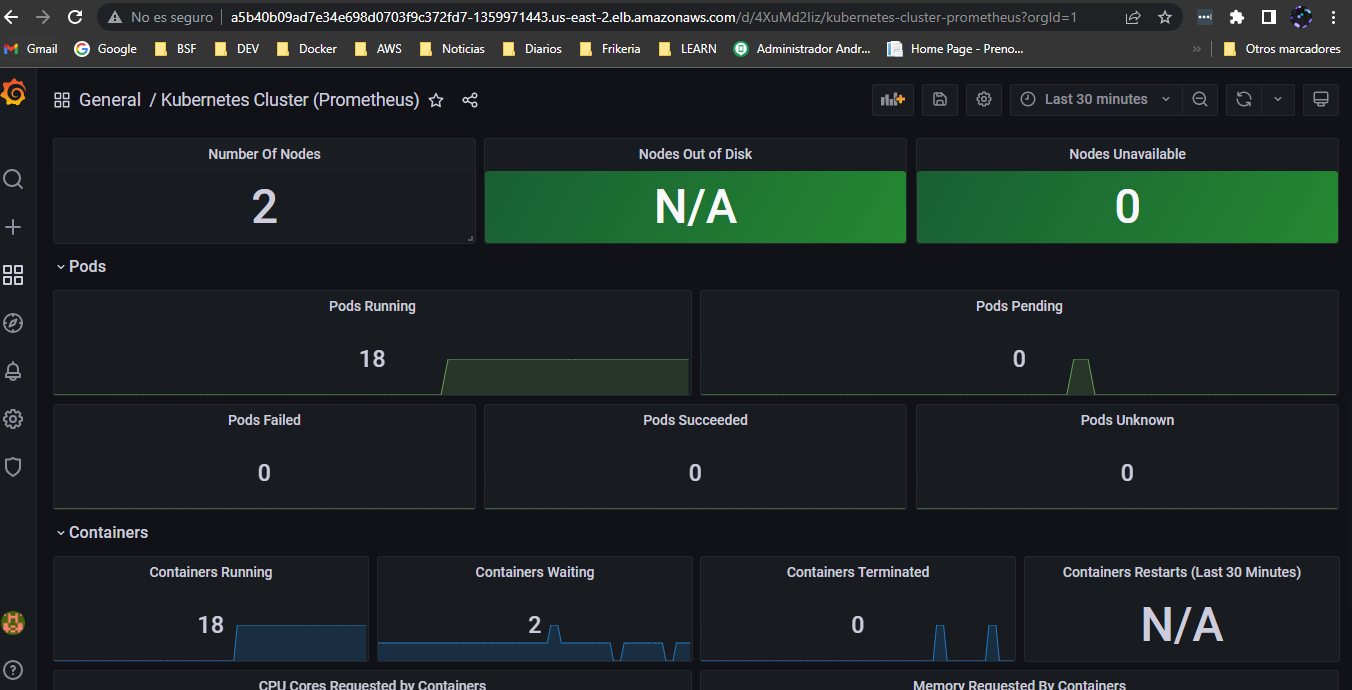


Importamos el dashboard id 3119



Importamos dashboard kubernetes cluster





BORRAMOS TODO