

Documentación de trabajo PIN final - GRUPO 15

Github: <https://github.com/alejandrotoledoweb/PIN-FINAL-DEVOPS-2403>

Creación de cluster con microK8s localmente

Creación del namespace `pin-devops`

```
Kubectl create namespace pin-devops
```

Comando a correr

```
kubectl apply -f nginx-deploy.yaml
```

Archivo: `nginx-deploy.yaml`

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

```
---  
  
apiVersion: v1  
kind: Service  
metadata:  
  name: nginx-service  
spec:  
  type: LoadBalancer  
  selector:  
    app: nginx  
  ports:  
    - port: 80  
      targetPort: 80
```

Revisamos que el nginx este corriendo

Obtengo los servicios que estan asignados al namespaces `pin-devops`

Comando:

```
kubectl get all -n pin-devops
```

O

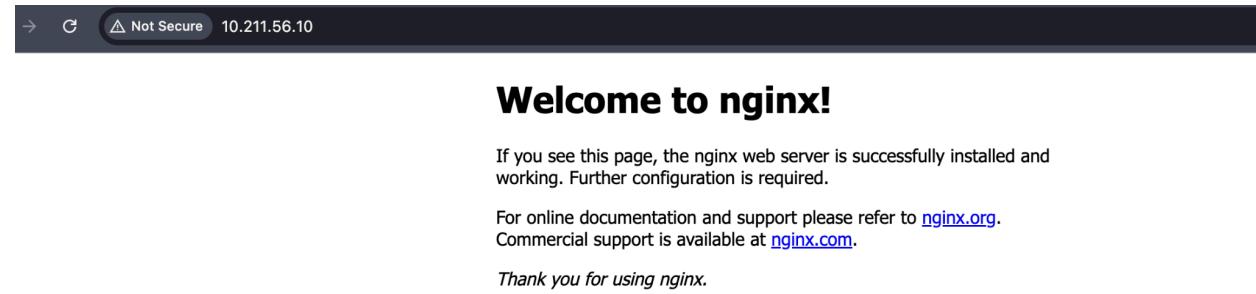
```
kubectl get svc -n pin-devops
```

```
test-nginx ~ % kubectl get all -n pin-devops
NAME                                     READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-647677fc66-mx2fh   1/1     Running   0          31h

NAME              TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/nginx-service   LoadBalancer   10.152.183.158  10.211.56.10  80:30322/TCP  31h

NAME              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment   1/1     1           1           31h

NAME             DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-647677fc66   1         1         1         31h
test-nginx ~ %
```



Install prometheus pods

Creación del script para usar helm para instalar prometheus

Archivo: prometheus.sh

```
#!/bin/bash

# Create the namespace for Prometheus
kubectl create namespace prometheus

# Add the Prometheus Helm repository
helm repo add prometheus-community
https://prometheus-community.github.io/helm-charts
helm repo update

# Install Prometheus using Helm
helm install prometheus prometheus-community/prometheus --namespace
prometheus \
--set alertmanager.persistentVolume.storageClass="microk8s-hostpath" \
--set server.persistentVolume.storageClass="microk8s-hostpath"

# Apply a PersistentVolumeClaim (updated for MicroK8s)
kubectl apply -f pv-microk8s.yaml

echo "Finishing installation..."
sleep 630

# Check Prometheus pods
kubectl get pods -n prometheus
```

Archivo: pv-microk8s.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  labels:
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/name: alertmanager
  name: storage-prometheus-alertmanager-0
  namespace: prometheus
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
  volumeMode: Filesystem
  storageClassName: microk8s-hostpath
```

Resultado de correr el script

```
* test-nginx /prometheus/prometheus.sh
"prometheus-community" already exists with the same configuration, skipping
Hang tight while we grab the latest from your chart repositories...
  - Successfully got an update from the "prometheus-community" chart repository
Update Complete. *Happy Helming!*
NAME: prometheus
LAST DEPLOYED: Wed Mar  5 14:29:22 2025
NAMESPACE: prometheus
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-server.prometheus.svc.cluster.local

Get the Prometheus server URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app.kubernetes.io/name=prometheus,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9090

The Prometheus alertmanager can be accessed via port 9093 on the following DNS name from within your cluster:
prometheus-alertmanager.prometheus.svc.cluster.local

Get the Alertmanager URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app.kubernetes.io/name=alertmanager,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].metadata.name}")
#####
##### WARNING: Pod Security Policy has been disabled by default since #####
#####   it deprecated after k8s 1.25+. use #####
#####   (index.Values "prometheus-node-exporter" "rbac" #####
##### . "pspEnabled") with (index.Values #####
##### "prometheus-node-exporter" "rbac" "pspAnnotations") #####
#####   in case you still need it. #####
#####

The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
prometheus-prometheus-pushgateway.prometheus.svc.cluster.local

Get the PushGateway URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus-pushgateway,component=pushgateway" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9091

For more information on running Prometheus, visit:
https://prometheus.io/
Warning: resource persistentvolumeclaims/storage-prometheus-alertmanager-0 is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
persistentvolumeclaim/storage-prometheus-alertmanager-0 configured
Finishing installation...
NAME          READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-0   0/1     Pending   0          1s
```

```

#####
# it deprecated after k8s 1.25+. use
# (indoor .Values "prometheus-node-exporter" "rbac"
#####
# "pspEnabled") with (indoor .Values
#####
# "prometheus-node-exporter" "rbac" "pspAnnotations")
#####
# in case you still need it.
#####

The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
prometheus-prometheus-pushgateway.prometheus.svc.cluster.local

Get the PushGateway URL by running these commands in the same shell:
export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus-pushgateway,component=pushgateway" -o jsonpath="{.items[0].metadata.name}")
kubectl --namespace prometheus port-forward $POD_NAME 9091

For more information on running Prometheus, visit:
https://prometheus.io/
Warning: resource persistentvolumeclaims/storage-prometheus-alertmanager-0 is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. Kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
persistentvolumeclaim/storage-prometheus-alertmanager-0 configured
Finishing installation...
NAME          READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-0   0/1     Pending   0          1s
prometheus-kube-state-metrics-5bd466f7f6-mj7st   0/1     ContainerCreating   0          1s
prometheus-prometheus-node-exporter-9c87p        0/1     ContainerCreating   0          1s
prometheus-prometheus-pushgateway-544579d549-f69t2  0/1     ContainerCreating   0          1s
prometheus-server-596945876b-zjrgn                0/2     Pending   0          1s

```

Revisar los pods y servicios del namespace prometheus

Al revisar los pods, hay dos pods que estan en pending, para resolver esto necesitamos habilitar **metallb** y **storage** con microk8s

```

→ test-nginx kubectl get all -n prometheus
          NAME          READY   STATUS    RESTARTS   AGE
pod/prometheus-alertmanager-0   0/1     Pending   0          3m10s
pod/prometheus-kube-state-metrics-5bd466f7f6-mj7st   1/1     Running  0          3m10s
pod/prometheus-prometheus-node-exporter-9c87p        1/1     Running  0          3m10s
pod/prometheus-prometheus-pushgateway-544579d549-f69t2  1/1     Running  0          3m10s
pod/prometheus-server-596945876b-zjrgn                0/2     Pending   0          3m10s

          NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/prometheus-alertmanager   ClusterIP  10.152.183.214 <none>       9093/TCP   3m10s
service/prometheus-alertmanager-headless   ClusterIP  None           <none>       9093/TCP   3m10s
service/prometheus-kube-state-metrics   ClusterIP  10.152.183.168 <none>       8080/TCP   3m10s
service/prometheus-prometheus-node-exporter   ClusterIP  10.152.183.156 <none>       9100/TCP   3m10s
service/prometheus-prometheus-pushgateway   ClusterIP  10.152.183.56  <none>       9091/TCP   3m10s
service/prometheus-server   ClusterIP  10.152.183.161 <none>       80/TCP     3m10s

          NAME          DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/prometheus-prometheus-node-exporter   1         1         1         1           1           kubernetes.io/os=linux   3m10s

          NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-kube-state-metrics   1/1     1           1           3m10s
deployment.apps/prometheus-prometheus-pushgateway 1/1     1           1           3m10s
deployment.apps/prometheus-server   0/1     1           0           3m10s

          NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-kube-state-metrics-5bd466f7f6  1         1         1         3m10s
replicaset.apps/prometheus-prometheus-pushgateway-544579d549 1         1         1         3m10s
replicaset.apps/prometheus-server-596945876b               1         1         0         3m10s

          NAME          READY   AGE
statefulset.apps/prometheus-alertmanager   0/1     3m10s

```

```

prometheus-server-596945876b-zjrgn          0/2      Pending        0       1s
▶ test-nginx kubectl get pods -n prometheus
NAME                                         READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-0                   0/1     Pending   0          66s
prometheus-kube-state-metrics-5bd466f7f6-mj7st 1/1     Running  0          66s
prometheus-prometheus-node-exporter-9c87p    1/1     Running  0          66s
prometheus-prometheus-pushgateway-544579d549-f69t2 1/1     Running  0          66s
prometheus-server-596945876b-zjrgn          0/2     Pending   0          66s
▶ test-nginx kubectl get svc -n prometheus
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)   AGE
prometheus-alertmanager   ClusterIP  10.152.183.214 <none>     9093/TCP  73s
prometheus-alertmanager-headless   ClusterIP  None         <none>     9093/TCP  73s
prometheus-kube-state-metrics   ClusterIP  10.152.183.168 <none>     8080/TCP  73s
prometheus-prometheus-node-exporter   ClusterIP  10.152.183.156 <none>     9100/TCP  73s
prometheus-prometheus-pushgateway   ClusterIP  10.152.183.56  <none>     9091/TCP  73s
prometheus-server            ClusterIP  10.152.183.161 <none>     80/TCP    73s
▶ test-nginx multipass list
Name          State     IPv4           Image
microk8s-vm   Running  10.211.56.3  Ubuntu 22.04 LTS
10.1.254.64
▶ test-nginx

```

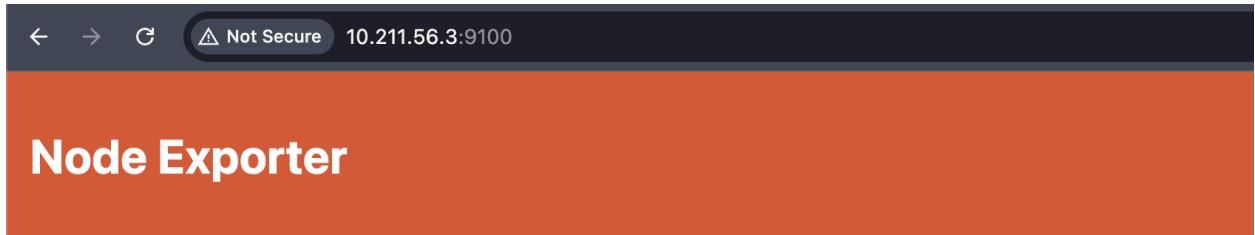
Al revisar que servicios de prometheus estan expuestos y funcionan se uso el

ip: 10.211.56.3

```

statefulset.apps/prometheus-alertmanager  0/1      3m10s
▶ test-nginx multipass list
Name          State     IPv4           Image
microk8s-vm   Running  10.211.56.3  Ubuntu 22.04 LTS
10.1.254.64

```



Prometheus Node Exporter

Version: (version=1.9.0, branch=HEAD, revision=02afa5c53c36123611533f2defea6ccd4546a9bb)

- [Metrics](#)

Download a detailed report of resource usage (pprof format, from the Go runtime):

- [heap usage \(memory\)](#)
- [CPU usage \(60 second profile\)](#)

To visualize and share profiles you can upload to [pprof.me](#)

Habilitar metallb and storage en microk8s

```
↳ test-nginx microk8s enable metallb
Infer repository core for addon metallb
Enabling MetalLB
Enter each IP address range delimited by comma (e.g. '10.64.140.43-10.64.140.49,192.168.0.105-192.168.0.111'): 10.211.56.10-10.211.56.20
Applying MetalLB manifest
customresourcedefinition.apirextensions.k8s.io/addresspools.metallb.io created
customresourcedefinition.apirextensions.k8s.io/bfdprofiles.metallb.io created
customresourcedefinition.apirextensions.k8s.io/bgpadvertisements.metallb.io created
customresourcedefinition.apirextensions.k8s.io/bgppeers.metallb.io created
customresourcedefinition.apirextensions.k8s.io/ipaddresses.metallb.io created
customresourcedefinition.apirextensions.k8s.io/ipaddresspools.metallb.io created
customresourcedefinition.apirextensions.k8s.io/l2advertisements.metallb.io created
namespace/metallb-system created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metalLB-system:controller created
clusterrole.rbac.authorization.k8s.io/metalLB-system:speaker created
role.rbac.authorization.k8s.io/controller created
role.rbac.authorization.k8s.io/pod-lister created
clusterrolebinding.rbac.authorization.k8s.io/metalLB-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metalLB-system:speaker created
rolebinding.rbac.authorization.k8s.io/controller created
secret/webhook-server-cert created
service/webhook-service created
rolebinding.rbac.authorization.k8s.io/pod-lister created
daemonset.apps/speaker created
deployment.apps/controller created
validatingwebhookconfiguration.admissionregistration.k8s.io/validating-webhook-configuration created
Waiting for MetalLB controller to be ready
'ideployment.apps/controller' condition met
Error from server (InternalError): error when creating "STDIN": Internal error occurred: failed calling webhook "ipaddresspoolvalidationwebhook.metallb.io": failed to call webhook: Post "https://webhook-service.metalLB-system.svc:443/validate-metallb-ip-vbmeta-ipaddresspool?timeout=10s": dial tcp 10.152.183.57:443: connect: connection refused
Error from server (InternalError): error when creating "STDIN": Internal error occurred: failed calling webhook "l2advertisementsvalidationwebhook.metallb.io": failed to call webhook: Post "https://webhook-service.metalLB-system.svc:443/validate-metallb-ip-vbmeta-l2advertisements?timeout=10s": dial tcp 10.152.183.57:443: connect: connection refused
Failed to create default address pool, will retry
Error from server (InternalError): error when creating "STDIN": Internal error occurred: failed calling webhook "ipaddresspoolvalidationwebhook.metallb.io": failed to call webhook: Post "https://webhook-service.metalLB-system.svc:443/validate-metallb-ip-vbmeta-ipaddresspool?timeout=10s": dial tcp 10.152.183.57:443: connect: connection refused
Error from server (InternalError): error when creating "STDIN": Internal error occurred: failed calling webhook "l2advertisementsvalidationwebhook.metallb.io": failed to call webhook: Post "https://webhook-service.metalLB-system.svc:443/validate-metallb-ip-vbmeta-l2advertisements?timeout=10s": dial tcp 10.152.183.57:443: connect: connection refused
Failed to create default address pool, will retry
Error from server (InternalError): error when creating "STDIN": Internal error occurred: failed calling webhook "ipaddresspoolvalidationwebhook.metallb.io": failed to call webhook: Post "https://webhook-service.metalLB-system.svc:443/validate-metallb-ip-vbmeta-ipaddresspool?timeout=10s": dial tcp 10.152.183.57:443: connect: connection refused
Error from server (InternalError): error when creating "STDIN": Internal error occurred: failed calling webhook "l2advertisementsvalidationwebhook.metallb.io": failed to call webhook: Post "https://webhook-service.metalLB-system.svc:443/validate-metallb-ip-vbmeta-l2advertisements?timeout=10s": dial tcp 10.152.183.57:443: connect: connection refused
Failed to create default address pool, will retry
inaddresspool.metallb.io/default-addresspool created
l2advertisements.metallb.io/default-advertise-all-pools created
MetalLB is enabled
```

```
Waiting for provisioning, dated 2023-01-10T20:20:00Z, over 20m0s persistencyvolume controller storageclass.storage.k8s.io microk8s hostpath
↳ test-nginx kubectl get storageclass
No resources found
↳ test-nginx microk8s enable storage
Infer repository core for addon storage
DEPRECATION WARNING: 'storage' is deprecated and will soon be removed. Please use 'hostpath-storage' instead.

Infer repository core for addon hostpath-storage
Enabling default storage class.
WARNING: Hostpath storage is not suitable for production environments.
A hostpath volume can grow beyond the size limit set in the volume claim manifest.

deployment.apps/hostpath-provisioner created
storageclass.storage.k8s.io/microk8s-hostpath created
serviceaccount/microk8s-hostpath created
clusterrole.rbac.authorization.k8s.io/microk8s-hostpath created
clusterrolebinding.rbac.authorization.k8s.io/microk8s-hostpath created
Storage will be available soon.
↳ test-nginx kubectl get storageclass
NAME          PROVISIONER      RECLAIMPOLICY  VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
microk8s-hostpath (default)  microk8s.io/hostpath  Delete          WaitForFirstConsumer  false                6s
↳ test-nginx kubectl delete pvc prometheus-server -n prometheus
```

Actualizar el servicio para usar un LoadBalancer

```
netcat is enabled
▶ test-nginx kubectl patch svc prometheus-server -n prometheus -p '{"spec": {"type": "LoadBalancer"}}'
service/prometheus-server patched
▶ test-nginx
```

Recrear el pvc de prometheus

Archivo: prometheus.sh

```
#!/bin/bash

# Ensure namespace exists
kubectl get namespace prometheus >/dev/null 2>&1 || kubectl create namespace prometheus

# Add the Prometheus Helm repository
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
helm repo update

# Install Prometheus using Helm
helm install prometheus prometheus-community/prometheus --namespace prometheus \
--set alertmanager.persistentVolume.storageClass="micr0k8s-hostpath" \
--set server.persistentVolume.storageClass="micr0k8s-hostpath"

# Apply PersistentVolumeClaim if the file exists
if [ -f "./prometheus/pv.yaml" ]; then
    kubectl apply -f ./prometheus/pv.yaml --namespace prometheus
else
    echo "Warning: ./prometheus/pv.yaml not found, skipping PVC creation"
fi

echo "Finishing installation..."
sleep 1

# Check pod status
kubectl get pods -n prometheus
```

Archivo: pv-microk8s.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: prometheus-server-pv
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  hostPath:
    path: "/mnt/data/prometheus"
```

Revisar que todos los pods de prometheus ahora estén corriendo y no en pending

```
▶ test-nginx kubectl get pods -n prometheus
NAME                               READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-0          1/1     Running   0          46m
prometheus-kube-state-metrics-5bd466f7f6-mj7st  1/1     Running   0          46m
prometheus-prometheus-node-exporter-9c87p        1/1     Running   0          46m
prometheus-prometheus-pushgateway-544579d549-f69t2 1/1     Running   0          46m
prometheus-server-69854fd789-2rbvq              2/2     Running   0          19m
▶ test-nginx
▶ test-nginx kubectl get all -n prometheus
NAME                                         READY   STATUS    RESTARTS   AGE
pod/prometheus-alertmanager-0                1/1     Running   0          48m
pod/prometheus-kube-state-metrics-5bd466f7f6-mj7st  1/1     Running   0          48m
pod/prometheus-prometheus-node-exporter-9c87p        1/1     Running   0          48m
pod/prometheus-prometheus-pushgateway-544579d549-f69t2 1/1     Running   0          48m
pod/prometheus-server-69854fd789-2rbvq              2/2     Running   0          21m

NAME                           TYPE      CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
service/prometheus-alertmanager   ClusterIP  10.152.183.214 <none>        9093/TCP  48m
service/prometheus-alertmanager-headless  ClusterIP  None           <none>        9093/TCP  48m
service/prometheus-kube-state-metrics   ClusterIP  10.152.183.168 <none>        8080/TCP  48m
service/prometheus-prometheus-node-exporter  ClusterIP  10.152.183.156 <none>        9100/TCP  48m
service/prometheus-prometheus-pushgateway  ClusterIP  10.152.183.56  <none>        9091/TCP  48m
service/prometheus-server           LoadBalancer  10.152.183.161  10.211.56.11  80:30186/TCP  48m

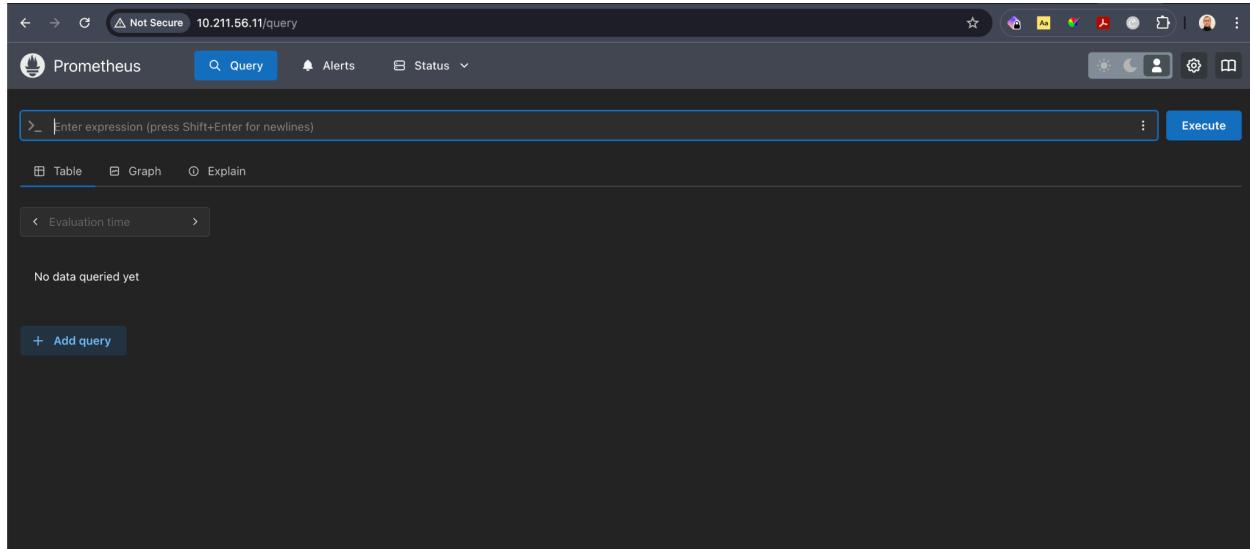
NAME                         DESIRED  CURRENT  READY   UP-TO-DATE  AVAILABLE  NODE SELECTOR   AGE
daemonset.apps/prometheus-prometheus-node-exporter  1         1         1         1         1           kubernetes.io/os=linux  48m

NAME                         READY   UP-TO-DATE  AVAILABLE  AGE
deployment.apps/prometheus-kube-state-metrics  1/1     1           1           48m
deployment.apps/prometheus-prometheus-pushgateway 1/1     1           1           48m
deployment.apps/prometheus-server               1/1     1           1           48m

NAME                         DESIRED  CURRENT  READY   AGE
replicaset.apps/prometheus-kube-state-metrics-5bd466f7f6  1         1         1         48m
replicaset.apps/prometheus-prometheus-pushgateway-544579d549 1         1         1         48m
replicaset.apps/prometheus-server-596945876b            0         0         0         48m
replicaset.apps/prometheus-server-5ffc5844b9            0         0         0         25m
replicaset.apps/prometheus-server-69854fd789             1         1         1         21m

NAME                         READY   AGE
statefulset.apps/prometheus-alertmanager  1/1     48m
```

Revisar que el dashboard de prometheus esta levantado y expuesto



Se revisa la ruta /targets para ver que todo este en estado UP

A screenshot of the Prometheus targets interface at the URL 10.211.56.11/targets. The interface shows three groups of targets: 'kubernetes-apiservers', 'kubernetes-nodes', and 'kubernetes-nodes-cadvisor'. Each group has a table with columns: Endpoint, Labels, Last scrape, and State. The 'kubernetes-apiservers' group has one endpoint: https://10.211.56.3:16443/metrics. The 'kubernetes-nodes' group has one endpoint: https://kubernetes.default.svc/api/v1/nodes/microk8s-vm/proxy/metrics. The 'kubernetes-nodes-cadvisor' group has one endpoint: https://kubernetes.default.svc/api/v1/nodes/microk8s-vm/proxy/metrics/cadvisor. All targets are marked as '1 / 1 up' and 'UP'.

Prometheus [Query](#) [Alerts](#) [Status > Target health](#)

Not Secure 10.211.56.11/targets

kubernetes-pods

Endpoint	Labels	Last scrape	State
http://10.211.56.3:7472/metrics	app="metallb" component="speaker" controller_revision_hash="79f47fd97" instance="10.211.56.3:7472" job="kubernetes-pods" namespace="metallb-system" node="microk8s-vn" pod="speaker-99r4v" pod_template_generation="1"	53.615s ago 14ms	UP
http://10.1.254.72:7472/metrics	app="metallb" component="controller" instance="10.1.254.72:7472" job="kubernetes-pods" namespace="metallb-system" nodes="microk8s-vn" pod="controller-7ffc454778-mp9zx" pod_template_hash="7ffc454778"	32.189s ago 1ms	UP

kubernetes-pods-slow

No targets

No active targets in this scrape pool. [Hide empty pools](#)

kubernetes-service-endpoints

Endpoint	Labels	Last scrape	State
http://10.1.254.65:9153/metrics	addonmanager_kubernetes_io_mode="Reconcile" instance="10.1.254.65:9153" jobs="kubernetes-service-endpoints" k8s_app="kube-dns" kubernetes_io_cluster_service="true" kubernetes_io_name="CoreDNS" namespace="kube-system" node="microk8s-vn" service="kube-dns"	10.854s ago 29ms	UP
http://10.1.254.71:8080/metrics	app_kubernetes_io_component="metrics" app_kubernetes_io_instance="prometheus" app_kubernetes_io_managed_by="Helm" app_kubernetes_io_name="kube-state-metrics"	27.287s ago 11ms	UP

Prometheus [Query](#) [Alerts](#) [Status > Target health](#)

Not Secure 10.211.56.11/targets

kubernetes-service-endpoints

Endpoint	Labels	Last scrape	State
http://10.211.56.3:9100/metrics	services="prometheus-kube-state-metrics" app_kubernetes_io_component="metrics" app_kubernetes_io_instance="prometheus" app_kubernetes_io_managed_by="Helm" app_kubernetes_io_name="prometheus-node-exporter" app_kubernetes_io_part_of="prometheus-node-exporter" app_kubernetes_io_version="1.9.0" helm_sh_charts="prometheus-node-exporter-4.44.0" instance="10.211.56.3:9100" job="kubernetes-service-endpoints" namespace="prometheus" node="microk8s-vn" service="prometheus-prometheus-node-exporter"	56.479s ago 43ms	UP

kubernetes-service-endpoints-slow

No targets

No active targets in this scrape pool. [Hide empty pools](#)

kubernetes-services

No targets

No active targets in this scrape pool. [Hide empty pools](#)

prometheus

Endpoint	Labels	Last scrape	State
http://localhost:9090/metrics	instance="localhost:9090" job="prometheus"	29.659s ago 7ms	UP

Not Secure 10.211.56.11/targets

Prometheus

Query Alerts Status > Target health

service="prometheus-prometheus-node-exporter"

kubernetes-service-endpoints-slow 0 / 0 up

No targets
No active targets in this scrape pool. Hide empty pools

kubernetes-services 0 / 0 up

No targets
No active targets in this scrape pool. Hide empty pools

prometheus 1 / 1 up

Endpoint	Labels	Last scrape	State
http://localhost:9090/metrics	instance="localhost:9090" job="prometheus"	29.659s ago	7ms UP

prometheus-pushgateway 1 / 1 up

Endpoint	Labels	Last scrape	State
http://prometheus-prometheus-pushgateway.prometheus.svc:9091/metrics	instance="prometheus-prometheus-pushgateway.prometheus.svc:9091" job="prometheus-pushgateway"	12.862s ago	9ms UP

The screenshot shows the Prometheus web interface at the URL 10.211.56.11/targets. The top navigation bar includes links for Not Secure, Query, Alerts, and Status > Target health. Below the navigation is a search bar with the query 'service="prometheus-prometheus-node-exporter"'. The main content area is divided into three sections: 'kubernetes-service-endpoints-slow' (0 / 0 up), 'kubernetes-services' (0 / 0 up), and two tables for 'prometheus' and 'prometheus-pushgateway'. Each table has columns for Endpoint, Labels, Last scrape, and State. The 'prometheus' table shows one endpoint at localhost:9090 with labels instance="localhost:9090" and job="prometheus", last scraped 29.659s ago, and a 7ms response time, marked as UP. The 'prometheus-pushgateway' table shows one endpoint at prometheus-prometheus-pushgateway.prometheus.svc:9091 with labels instance="prometheus-prometheus-pushgateway.prometheus.svc:9091" and job="prometheus-pushgateway", last scraped 12.862s ago, and a 9ms response time, also marked as UP.

Agregar Grafana

Archivo: grafana.sh

```
#!/bin/bash
# Generate a random admin password
ADMIN_PASSWORD=$(openssl rand -base64 12)

# Add the Grafana Helm chart repository
helm repo add grafana https://grafana.github.io/helm-charts

# Update Helm repository cache
helm repo update

# Create the 'grafana' namespace if it does not exist
kubectl create namespace grafana --dry-run=client -o yaml | kubectl apply -f
./grafana/grafana.yaml

# Install Grafana with Helm
helm install grafana grafana/grafana \
--namespace grafana \
--values ./grafana/grafana.yaml \
--set persistence.enabled=true \
--set persistence.storageClassName=microk8s-hostpath \
--set adminPassword="$ADMIN_PASSWORD" \
--set service.type=LoadBalancer

echo "Finishing the installation..."
sleep 60

# Get all Grafana resources
kubectl get all -n grafana

# Obtain the URL for the Grafana page
grafana_ip=$(kubectl get svc -n grafana grafana -o
jsonpath='{.status.loadBalancer.ingress[0].ip}')
echo "-----"
echo "You can view the Grafana page in your browser with this URL: http://$grafana_ip:80"
echo "Grafana admin password: $ADMIN_PASSWORD"
echo "-----"
```

Archivo: grafana.yaml

```
datasources:
  datasources.yaml:
    apiVersion: 1
    datasources:
      - name: Prometheus
        type: prometheus
        url: http://prometheus-server.prometheus.svc.cluster.local
        access: proxy
        isDefault: true
```

```
* test=nginx ./grafana/grafana.sh
"grafana" already exists with the same configuration, skipping
Hang tight while we grab the latest from your chart repositories...
  ... Successfully got an update from the "grafana" chart repository
  ... Successfully got an update from the "prometheus-community" chart repository
Update Complete. *Happy Helming!*
error: error validating "./grafana/grafana.yaml": error validating data: [apiVersion not set, kind not set]; if you choose to ignore these errors, turn validation off with --validate=false
NAME: grafana
LAST DEPLOYED: Wed Mar  5 16:38:44 2025
NAMESPACE: grafana
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:
   kubectl get secret --namespace grafana grafana -o jsonpath='{.data.admin-password}' | base64 --decode ; echo

2. The Grafana server can be accessed via port 80 on the following DNS name from within your cluster:
   grafana.grafana.svc.cluster.local

   Get the Grafana URL to visit by running these commands in the same shell:
   NOTE: It may take a few minutes for the LoadBalancer IP to be available.
   You can watch the status of by running 'kubectl get svc --namespace grafana -w grafana'
   export SERVICE_IP=$(kubectl get svc --namespace grafana grafana -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
   http://$SERVICE_IP:80

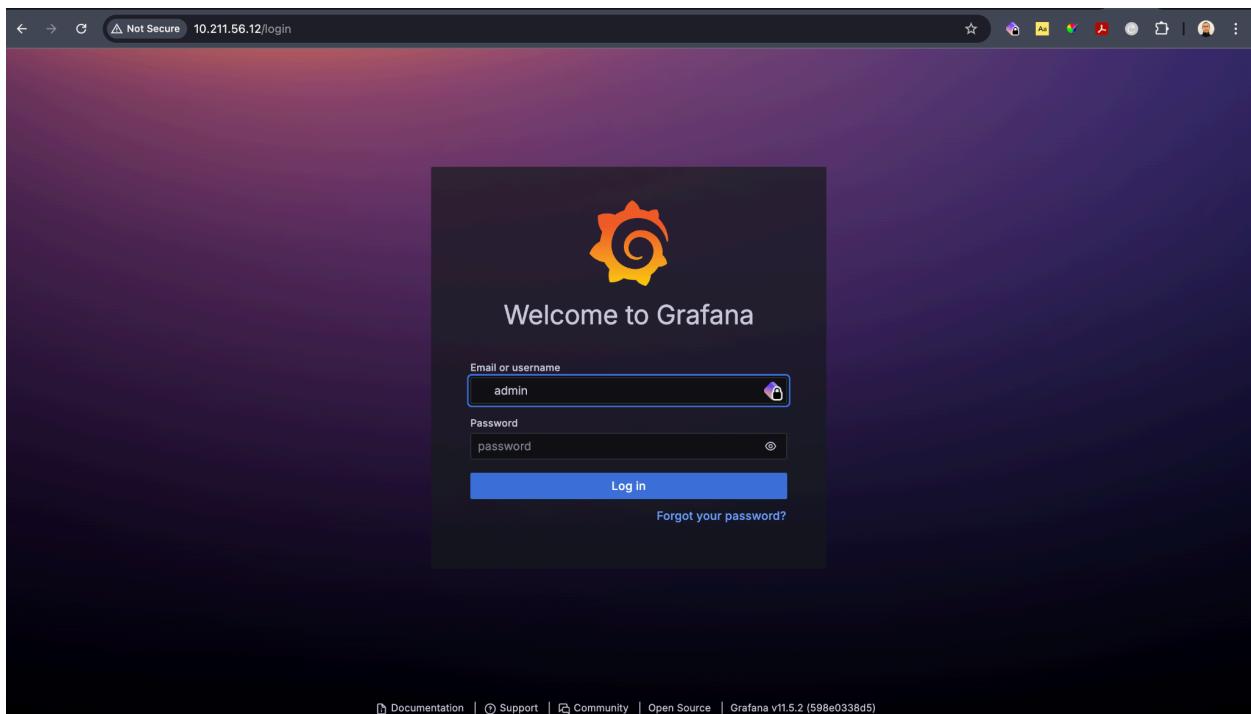
3. Login with the password from step 1 and the username: admin
Finishing the installation...
NAME          READY   STATUS    RESTARTS   AGE
pod/grafana-6c6f09ffca-xx5pf  1/1     Running   0          60s

NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/grafana  LoadBalancer  10.152.183.230  10.211.56.12  80:31023/TCP  60s

NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana  1/1     1           1           60s
NAME           DESIRED  CURRENT    READY   AGE
replicaset.apps/grafana-6c6f09ffca  1       1         1      60s
-----
You can view the Grafana page in your browser with this URL: http://10.211.56.12:80
Grafana admin password: dxWrRuJFwaXn187
-----
```

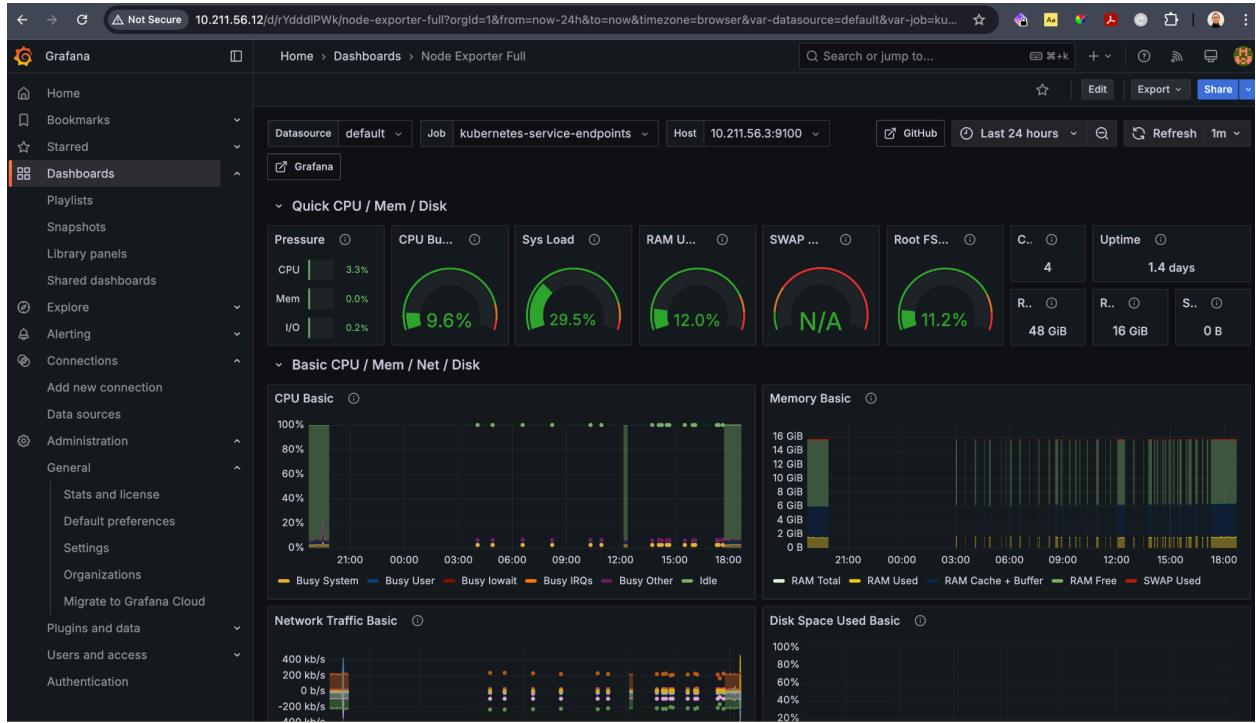
Revisar que el dashboard funciona en el host y puerto mencionado e ingreso con la clave creada

http://10.211.56.12



Agrego los dashboards para ver las métricas

Node Exporter Full (1860) – Estado del servidor y nodos



Kubernetes Cluster Metrics (6417) – Visión general del clúster

The screenshot shows a Grafana dashboard titled "Kubernetes Cluster (Prometheus)". The left sidebar is the Grafana navigation menu. The main area contains several panels:

- Cluster Health:** Four panels: Cluster Pod Usage, Cluster CPU Usage, Cluster Memory Usage, Cluster Disk Usage, all showing "N/A".
- Cluster Metrics:** Four panels: Cluster Pod Capacity, Cluster CPU Capacity, Cluster Mem Capacity, Cluster Disk Capacity. The first panel has a chart showing pods requested over time from 16:40 to 17:00, with a single bar at 10.
- Deployments:** Three panels: Deployment Replicas (9), Deployment Replicas - Updated (9), Deployment Replicas - Unavailable (0).
- Node:** Three panels: Number Of Nodes, Nodes Out of Disk, Nodes Unavailable.

Below the dashboard is a terminal window displaying the following commands and their outputs:

```
statefulset.apps/prometheus-alertmanager 1/1 4h8m
└─* test-nginx curl http://10.211.56.11/api/v1/status/runtimeinfo
{"status":"success","data":[{"startTime":"2025-03-05T19:59:00.787886156Z","CWD":"/prometheus","hostname":"prometheus-server-69854fd789-2rbvq","serverTime":"2025-03-06T23:06:25.3406940812Z","reloadConfigSuccess":true,"lastConfigTime":"2025-03-05T20:01:52Z","corruptionCount":0,"goroutineCount":145,"GOMAXPROCS":4,"GOMEMLIMIT":15065985024,"GOGC":"75","GODEBUG":"","storageRetention":"15d"}]
└─* test-nginx kubectl get pods -n prometheus | grep node-exporter
prometheus-prometheus-node-exporter-9c87p 1/1 Running 0 27h
└─* test-nginx
```

Kubernetes Cluster monitoring (3119) – Visión general del clúster

