

Informe de Actividades TalentoTech – Arquitectura En la Nube

Elaborado por: Carlos Andrés Piedrahita Velásquez – 8026998 – Grupo 7

Repositorio: <https://github.com/capiedrav/actividades-talentotech2>

Primera actividad – Despliegue de aplicación en Kubernetes

1. Aplicar manifiesto del despliegue y verificar que esté corriendo:

```
! deployment.yml M X ! service.yml
primera_actividad > ! deployment.yml > ...
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: web-app-deployment
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        app: web-app
10   template:
11     metadata:
12       labels:
13         app: web-app
14     spec:
15       containers:
16         - name: web-app
17           image: nigelpoulton/k8sbook:1.0 # imagen a usar en este ejercicio
18           ports:
19             - containerPort: 8080
20

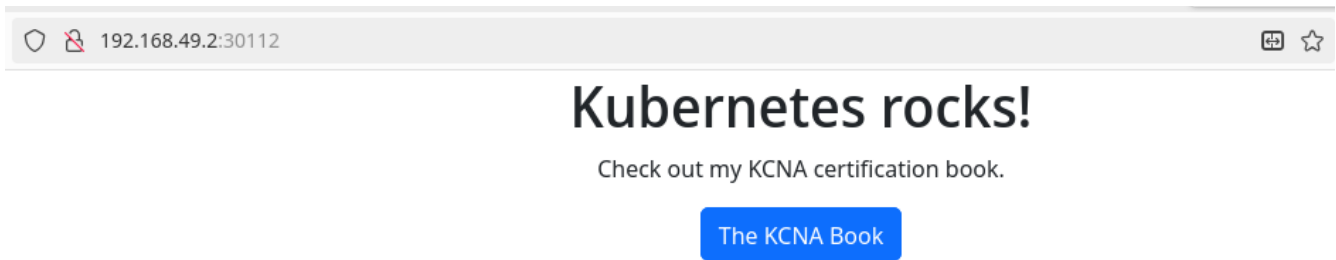
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
• capiedrav@suse-desktop:~/my_projects/talento-tech/primera_actividad> kubectl apply -f deployment.yml
deployment.apps/web-app-deployment created
• capiedrav@suse-desktop:~/my_projects/talento-tech/primera_actividad> kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
web-app-deployment  3/3     3            3           2m3s
• capiedrav@suse-desktop:~/my_projects/talento-tech/primera_actividad> kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
web-app-deployment-54f48b6b6f-csgln  2/2     Running   0          2m16s
web-app-deployment-54f48b6b6f-q8mq4  2/2     Running   0          2m16s
web-app-deployment-54f48b6b6f-zt2gk  2/2     Running   0          2m16s
• capiedrav@suse-desktop:~/my_projects/talento-tech/primera_actividad>
```

2. Aplicar manifiesto del servicio y verificar que esté corriendo:

```
! deployment.yml M ! service.yml M X
primera_actividad > ! service.yml > {} spec > [ ] ports > {} 0 > # port
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: web-app-service
5  spec:
6    selector:
7      app: web-app
8    type: LoadBalancer
9    ports:
10     - port: 8080
11

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
• capiedrav@suse-desktop:~/my_projects/talento-tech/primera_actividad> kubectl apply -f service.yml
service/web-app-service created
• capiedrav@suse-desktop:~/my_projects/talento-tech/primera_actividad> kubectl get svc
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes    ClusterIP     10.96.0.1        <none>           443/TCP          2d7h
web-app-service LoadBalancer  10.110.230.8     <pending>        8080:30112/TCP   9s
• capiedrav@suse-desktop:~/my_projects/talento-tech/primera_actividad> minikube service web-app-service --url
http://192.168.49.2:30112
• capiedrav@suse-desktop:~/my_projects/talento-tech/primera_actividad>
```

3. Probar la aplicación:



4. Escalar el despliegue:

```
capiedrav@suse-desktop:~/my_projects/talento-tech/primer_actividad> kubectl scale deployment web-app-deployment --replicas=5
deployment.apps/web-app-deployment scaled
capiedrav@suse-desktop:~/my_projects/talento-tech/primer_actividad> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
web-app-deployment-54f48b6b6f-csgln 2/2     Running   0          27m
web-app-deployment-54f48b6b6f-gnkgx 2/2     Running   0          15s
web-app-deployment-54f48b6b6f-q8mq4 2/2     Running   0          27m
web-app-deployment-54f48b6b6f-snsqg 2/2     Running   0          15s
web-app-deployment-54f48b6b6f-zt2gk 2/2     Running   0          27m
capiedrav@suse-desktop:~/my_projects/talento-tech/primer_actividad> █
```

Segunda actividad – Escalado en Kubernetes

1. Aplicar manifiestos de deployment, servicio, hpa y verificar que estén funcionando:

```
! deployment.yml ! service.yml ! hpa.yml x
segunda_actividad > ! hpa.yml > {} spec > {} scaleTargetRef > name
1  apiVersion: autoscaling/v2
2  kind: HorizontalPodAutoscaler
3  metadata:
4    name: web-app-hpa
5  spec:
6    scaleTargetRef:
7      apiVersion: apps/v1
8      kind: Deployment
9      name: web-app-deployment
10 minReplicas: 2
11 maxReplicas: 10
12 metrics:
13 - type: Resource
14   resource:
15     name: cpu
16     target:
17       type: Utilization
18       averageUtilization: 50

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS bash - se
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
web-app-deployment-7d6c69b54c-4vw7j 2/2     Running   0          115s
web-app-deployment-7d6c69b54c-bdj8x 2/2     Running   0          115s
web-app-deployment-7d6c69b54c-kdsxn 2/2     Running   0          115s
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl get svc
NAME            TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes      ClusterIP     10.96.0.1     <none>         443/TCP          2d8h
web-app-service  LoadBalancer 10.111.16.74  <pending>      8080:30511/TCP   2m8s
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl get hpa
NAME            REFERENCE                TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
web-app-hpa     Deployment/web-app-deployment  cpu: 2%/50%    2         10        3          2m1s
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> █
```

2. Generar tráfico para la aplicación para activar el escalado automático:

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> minikube service web-app-service --url http://192.168.49.2:30511
```

```
kubectl run -i --tty load-generator --image=busybox -- /bin/sh -c "while true; do wget -q -O- http://192.168.49.2:30511; done"
```

3.

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
load-generator	2/2	Running	0	10m
web-app-deployment-7d6c69b54c-2xwn6	2/2	Running	0	5m5s
web-app-deployment-7d6c69b54c-4vw7j	2/2	Running	0	34m
web-app-deployment-7d6c69b54c-9ft9w	2/2	Running	0	7m5s
web-app-deployment-7d6c69b54c-bdj8x	2/2	Running	0	34m
web-app-deployment-7d6c69b54c-kbf7z	2/2	Running	0	7m5s
web-app-deployment-7d6c69b54c-sbjw5	2/2	Running	0	7m5s
web-app-deployment-7d6c69b54c-smpkz	2/2	Running	0	9m5s

Verificar el estado del hpa:

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl get hpa
```

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
web-app-hpa	Deployment/web-app-deployment	cpu: 50%/50%	2	10	7	43m

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> □
```

4. Eliminar un pod y verificar que se cree nuevamente:

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
load-generator	2/2	Running	0	22m
web-app-deployment-7d6c69b54c-2xwn6	2/2	Running	0	16m
web-app-deployment-7d6c69b54c-4vw7j	2/2	Running	0	45m
web-app-deployment-7d6c69b54c-9ft9w	2/2	Running	0	18m
web-app-deployment-7d6c69b54c-bdj8x	2/2	Running	0	45m
web-app-deployment-7d6c69b54c-kbf7z	2/2	Running	0	18m
web-app-deployment-7d6c69b54c-sbjw5	2/2	Running	0	18m
web-app-deployment-7d6c69b54c-smpkz	2/2	Running	0	20m

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl delete pod web-app-deployment-7d6c69b54c-2xwn6
```

pod "web-app-deployment-7d6c69b54c-2xwn6" deleted

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
load-generator	2/2	Running	0	23m
web-app-deployment-7d6c69b54c-4vw7j	2/2	Running	0	46m
web-app-deployment-7d6c69b54c-9ft9w	2/2	Running	0	19m
web-app-deployment-7d6c69b54c-bdj8x	2/2	Running	0	46m
web-app-deployment-7d6c69b54c-kbf7z	2/2	Running	0	19m
web-app-deployment-7d6c69b54c-sbjw5	2/2	Running	0	19m
web-app-deployment-7d6c69b54c-smpkz	2/2	Running	0	21m
web-app-deployment-7d6c69b54c-zf6nm	2/2	Running	0	54s

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> □
```

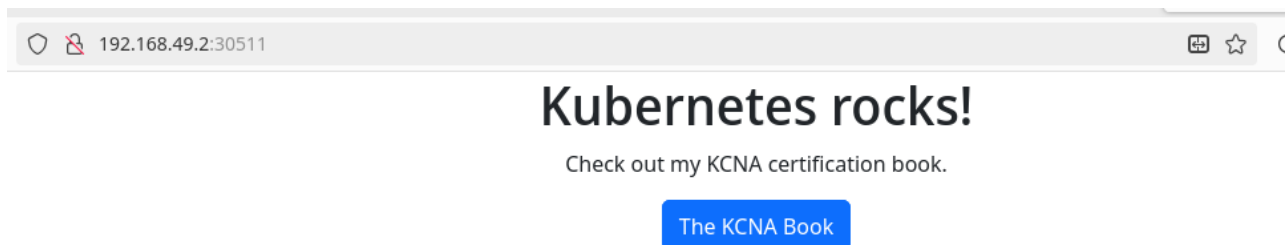
5. Escalar manualmente el despliegue:

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl scale deployment web-app-deployment --replicas=10
deployment.apps/web-app-deployment scaled
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
load-generator	2/2	Running	0	27m
web-app-deployment-7d6c69b54c-4vw7j	2/2	Running	0	50m
web-app-deployment-7d6c69b54c-92lt7	2/2	Running	0	5s
web-app-deployment-7d6c69b54c-9ft9w	2/2	Running	0	23m
web-app-deployment-7d6c69b54c-bdj8x	2/2	Running	0	50m
web-app-deployment-7d6c69b54c-hpps4	2/2	Running	0	5s
web-app-deployment-7d6c69b54c-j5glj	2/2	Running	0	5s
web-app-deployment-7d6c69b54c-kbf7z	2/2	Running	0	23m
web-app-deployment-7d6c69b54c-sbjw5	2/2	Running	0	23m
web-app-deployment-7d6c69b54c-smpkz	2/2	Running	0	25m
web-app-deployment-7d6c69b54c-zf6nm	2/2	Running	0	4m50s

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad>
```

6. Acceder a la aplicación y verificar que sigue funcionando:



7. Después de que la carga de trabajo disminuye, verificar que los pods en exceso se eliminan:

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl delete pod load-generator
pod "load-generator" deleted

capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl get hpa
```

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
web-app-hpa	Deployment/web-app-deployment	cpu: 2%/50%	2	10	2	76m

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
web-app-deployment-7d6c69b54c-4vw7j	2/2	Running	0	77m
web-app-deployment-7d6c69b54c-bdj8x	2/2	Running	0	77m

```
capiedrav@suse-desktop:~/my_projects/talento-tech/segunda_actividad>
```

Tercera Actividad – Seguridad en Contenedores Docker

Actividad basada en el video [Docker Security Best Practices](#)

En el video se propone como buenas prácticas de seguridad usar una imagen liviana, actualizar los paquetes de la imagen, crear un usuario sin permisos de administrador y limitar la cantidad de recursos que puede usar el contenedor una vez esté ejecutandose.

1. Usar una imagen liviana, actualizar los paquetes de la imagen, crear un usuario sin permisos de administrador:

```
1 FROM python:3.11-slim
2
3 ENV PYTHONDONTWRITEBYTECODE=1
4 ENV PYTHONUNBUFFERED=1
5
6 # update container packages
7 RUN apt update -y && apt upgrade -y
8
9 # create user app_user
10 RUN addgroup --gid 1000 app_user && \
11     adduser --uid 1000 --gid 1000 --disabled-password --gecos "" app_user --home /home/app_user
12
13 EXPOSE 8000
14
15 WORKDIR /home/app_user/drif-quickstart
16
17 # install requirements
18 COPY ./requirements.txt .
19 RUN pip install --no-cache-dir -r requirements.txt
20
21 # copy source code
22 COPY . .
23
24 # change ownership and permissions
25 RUN chown -R app_user:app_user . && chmod -R 755 .
26
27 # set app_user as default user
28 USER app_user
29
30 # start the app
31 CMD ["python", "/home/app_user/drif-quickstart/manage.py", "runserver", "0.0.0.0:8000"]
```

```
capiedrav@suse-desktop:~/my_projects/talento-tech/tercera_actividad> docker image build -t tercera_actividad_img .
[+] Building 0.1s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 758B
=> [internal] load metadata for docker.io/library/python:3.11-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/8] FROM docker.io/library/python:3.11-slim
=> [internal] load build context
=> => transferring context: 922B
=> CACHED [2/8] RUN apt update -y && apt upgrade -y
```

2. Verificar que el usuario no tiene permisos de administrador:

```
capiedrav@suse-desktop:~/my_projects/talento-tech/tercera_actividad> docker container exec -it tercera_actividad_ctr /bin/bash
app_user@2f0b48b3fa72:~/drif-quickstart$ apt update
Reading package lists... Done
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
E: Unable to lock directory /var/lib/apt/lists/
app_user@2f0b48b3fa72:~/drif-quickstart$ whoami
app_user
app_user@2f0b48b3fa72:~/drif-quickstart$
```

3. Limitar la cantidad de recursos que puede usar el contenedor una vez esté ejecutandose:

```
capiedrav@suse-desktop:~/my_projects/talento-tech/tercera_actividad> docker container run -d -p 8000:8000 --memory=256m --cpus=0.5 --name tercera_actividad_ctr tercera_actividad_img
2f0b48b3fa7203336d6cc1a454a30dfdc5de96bc8ccfbdbd1a4dfb88dc03afb82
```

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
2f0b48b3fa72	tercera_actividad_ctr	1.11%	79.66MiB / 256MiB	31.12%	2.74kB / 0B	0B / 0B	3

Cuarta Actividad - Monitoreo de Aplicaciones en Kubernetes con Istio, Kiali y Prometheus

1. Instale Istio, Kiali y Prometheus:

```
capiedrav@suse-desktop:~/my_projects/talento-tech/tercera_actividad> kubectl get pods -n istio-system
```

NAME	READY	STATUS	RESTARTS	AGE
istio-egressgateway-7985f6b7f9-shtzp	1/1	Running	4 (40h ago)	3d15h
istio-ingressgateway-f77976786-rv7f6	1/1	Running	3 (40h ago)	2d14h
istiod-55c88cf8cd-9nlm4	1/1	Running	3 (2m33s ago)	2d14h
kiali-845949f44d-nch9k	1/1	Running	4 (40h ago)	3d14h
prometheus-alertmanager-0	1/1	Running	4 (40h ago)	3d15h
prometheus-kube-state-metrics-75b5bb4bf8-9dttt	1/1	Running	7 (106s ago)	3d15h
prometheus-prometheus-node-exporter-6mzbs	1/1	Running	4 (40h ago)	3d15h
prometheus-prometheus-pushgateway-84557d6c79-f2gnm	1/1	Running	4 (40h ago)	3d15h
prometheus-server-644d686bc6-xwpfm	2/2	Running	8 (40h ago)	3d15h

2. Lanzar aplicaciones:

```
capiedrav@suse-desktop:~/my_projects/talento-tech/cuarta_actividad> kubectl apply -f deployment.yml -f
service.yml -f load-generator-pod.yml
deployment.apps/web-app-deployment created
service/web-app-service created
pod/load-generator-pod created
capiedrav@suse-desktop:~/my_projects/talento-tech/cuarta_actividad> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
load-generator-pod	2/2	Running	0	36s
web-app-deployment-7d6c69b54c-2rbxb	2/2	Running	0	36s
web-app-deployment-7d6c69b54c-7wvt8	2/2	Running	0	36s
web-app-deployment-7d6c69b54c-twdhw	2/2	Running	0	36s

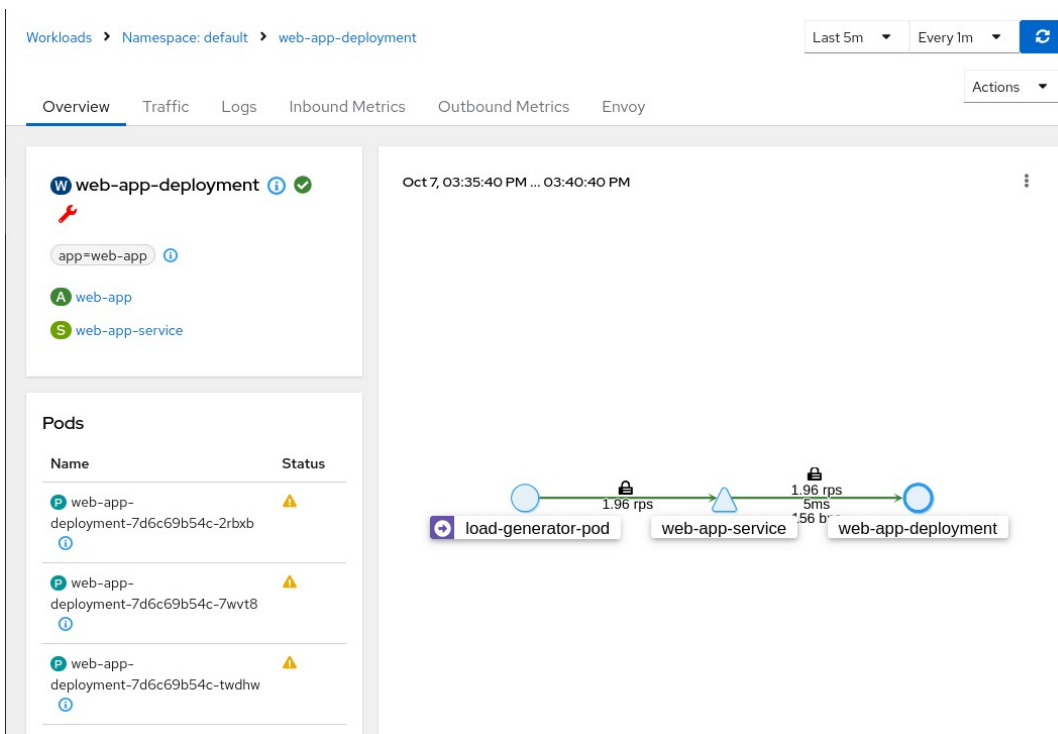
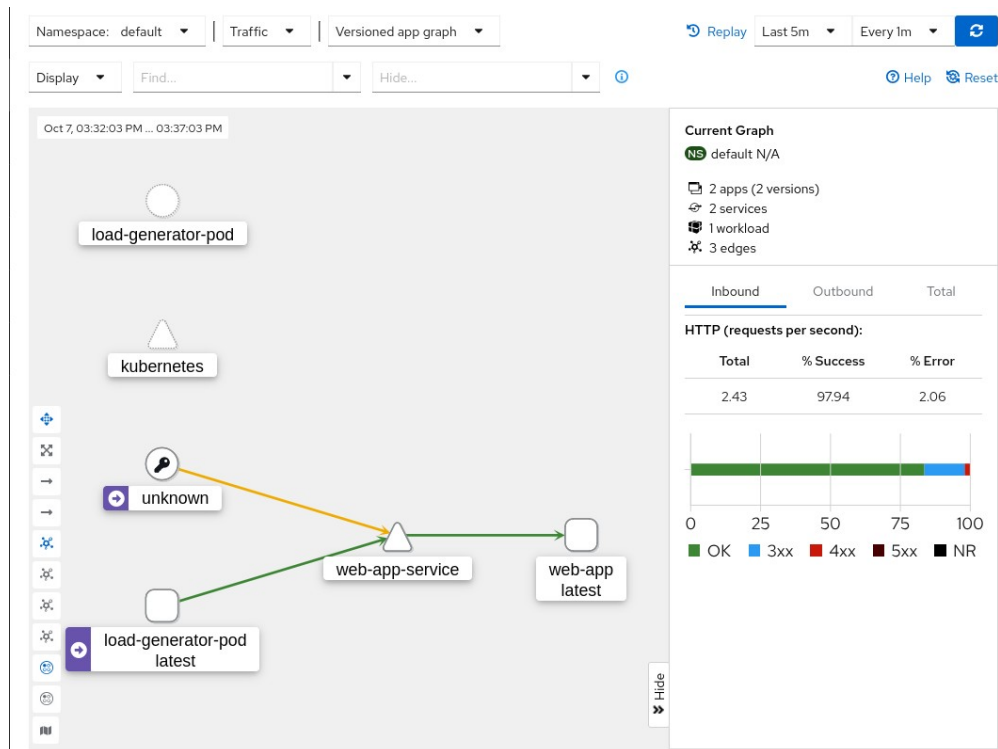
```
capiedrav@suse-desktop:~/my_projects/talento-tech/cuarta_actividad> █
```

3. Lanzar el dashboard de Kiali:

```
capiedrav@suse-desktop:~/my_projects/talento-tech> istioctl dashboard kiali
http://localhost:20001/kiali
█
```

Continúa en la siguiente página

4. Monitorear el cluster desde Kiali:



Cuarta Actividad Parte 2 – Monitorización de un clúster Kubernetes mediante Prometheus y Grafana

1. Configuración de la aplicación para pruebas con Prometheus y Grafana, primero se debe crear la imagen y subirla a Docker-hub

```
PS C:\Users\leina\OneDrive\Documentos\GitHub\metricas> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
metricas	latest	7660fa143470	2 minutes ago	1.11GB
registry.k8s.io/kube-apiserver	v1.31.0	604f5db92eaa	7 weeks ago	94.2MB
registry.k8s.io/kube-scheduler	v1.31.0	1766f54c897f	7 weeks ago	67.4MB
registry.k8s.io/kube-controller-manager	v1.31.0	045733566833	7 weeks ago	88.4MB

2. Ingresamos al folder de deployments compartido en kubernetes (osmergit) en github y cargamos los manifiestos:

```
PS C:\Users\leina\OneDrive\Documentos\GitHub\kubernetes\deployments> kubectl apply -f .\deployprometheus.yml
deployment.apps/metricas-deploy created
PS C:\Users\leina\OneDrive\Documentos\GitHub\kubernetes\deployments> kubectl apply -f .\lbprot.yml
service/svc-metricas created
PS C:\Users\leina\OneDrive\Documentos\GitHub\kubernetes\deployments> kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
metricas-deploy	1/1	1	1	12m
svc-test	5/5	5	5	119m

```
PS C:\Users\leina\OneDrive\Documentos\GitHub\kubernetes\deployments> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
metricas-deploy-5f4f58f9fc-j6qdp	1/1	Running	0	22m
svc-test-5dd69fb8bf-2s4ld	1/1	Running	0	129m
svc-test-5dd69fb8bf-6wr6h	1/1	Running	0	129m
svc-test-5dd69fb8bf-krrms	1/1	Running	0	129m
svc-test-5dd69fb8bf-nvjnd	1/1	Running	0	129m
svc-test-5dd69fb8bf-q9p9f	1/1	Running	0	129m

```
PS C:\Users\leina\OneDrive\Documentos\GitHub\kubernetes\deployments> kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
cloud-lb	LoadBalancer	10.98.177.65	<pending>	9000:32246/TCP	130m
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	133m
svc-metricas	LoadBalancer	10.109.73.99	<pending>	3002:30314/TCP	11m

3. Realizamos la instalación de Helm, descargando y actualizando los repositorio:

```
PS C:\Users\leina\OneDrive\Documentos\GitHub\kubernetes\deployments> helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "istio" chart repository
...Successfully got an update from the "grafana" chart repository
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. #Happy Helming!#
```


4. Ingresamos al folder de prometheus compartido en kubernetes (osmergit) en github y cargamos el deployment, el service y el configmap:

```
PS C:\Users\leina\OneDrive\Documentos\GitHub\kubernetes\prometheus> kubectl apply -f .\final.yml
deployment.apps/app-prometheus created
PS C:\Users\leina\OneDrive\Documentos\GitHub\kubernetes\prometheus> kubectl apply -f .\servicio.yml
service/prometheus-service created
PS C:\Users\leina\OneDrive\Documentos\GitHub\kubernetes\prometheus> kubectl apply -f .\prom-svc.yml
configmap/prometheus-config created
```

5. Lanzamos Prometheus mediante la ejecución de su servicio (port-forward):

```
PS C:\Users\leina\OneDrive\Documentos\GitHub\kubernetes\prometheus> kubectl port-forward svc/prometheus-service 9090:9090
Forwarding from 127.0.0.1:9090 -> 9090
Forwarding from [::1]:9090 -> 9090
```

6. Verificamos en que puerto se esta ejecutando prometheus internamente:

```
PS C:\Users\leina\OneDrive\Documents\GitHub\kubernetes> kubectl get svc prometheus-service
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
prometheus-service  NodePort    10.98.177.95   <none>          9090:32527/TCP   4m54s

PS C:\Users\leina\OneDrive\Documents\GitHub\kubernetes> kubectl get nodes -o wide
NAME                STATUS    ROLES          AGE      VERSION    INTERNAL-IP    EXTERNAL-IP    OS-IMAGE             KERNEL-VERSION    CONTAINER-RUNTIME
minikube            Ready     control-plane  143m     v1.31.0    172.20.135.193 <none>         Buildroot 2023.02.9    5.10.207          docker://27.2.0
```

7. Con el objetivo de ver las métricas cargamos tráfico a la aplicación, en un pod específico (hello-v2-.... Fue el elegido, según la documentación):

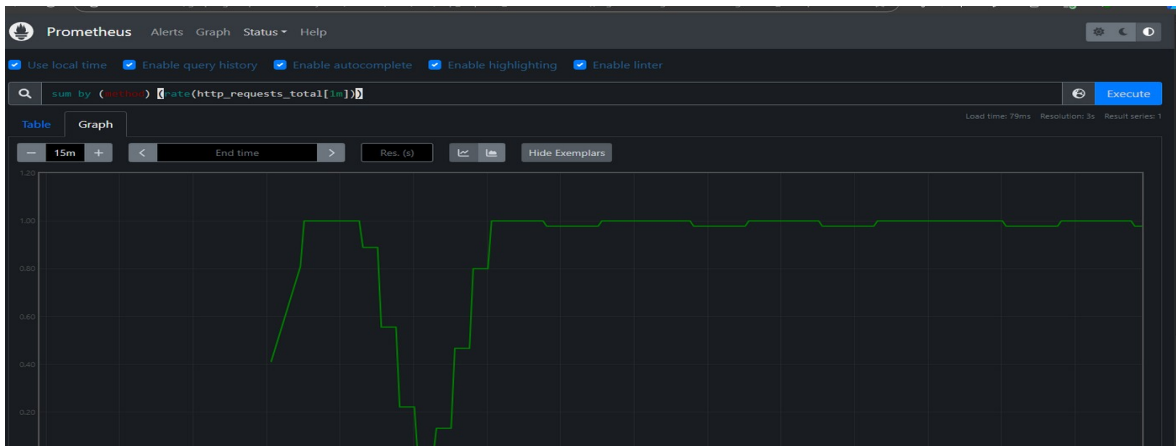
```
labeled:  
6   app: metrics-backend  
    osmergit, 6 hours ago + prometheus y Grafana  
7 spec:  
8   type: LoadBalancer # Cambiado de LoadBalancer a ClusterIP  
9   ports:  
10    - protocol: TCP  
11      port: 3002  
12      targetPort: 3002  
13 selector:  
14   app: metrics-backend
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

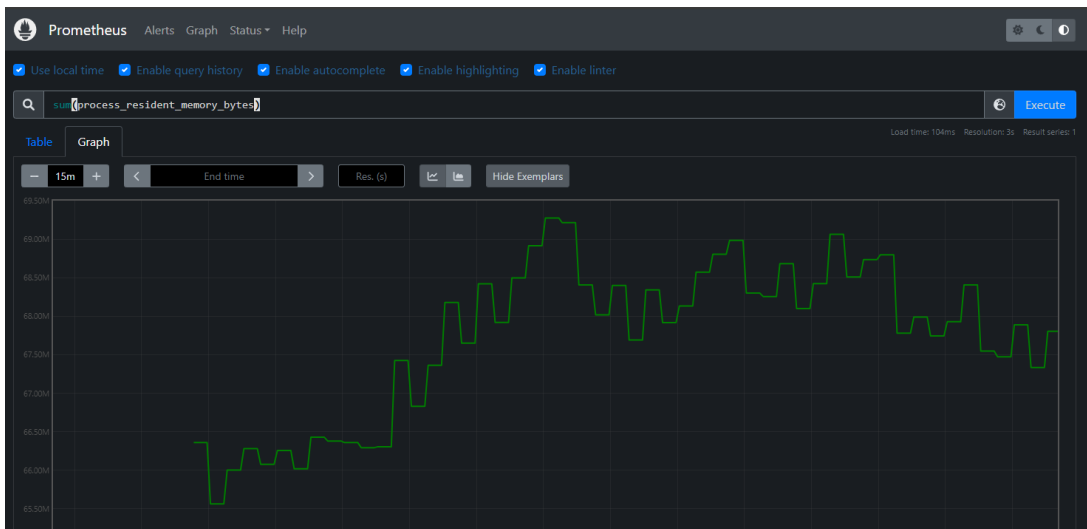
	TERMINAL	PORTS	GITLENS
hello-v2-6d96499dc-9-lhcvc	3/3 Running 0	124m	
hello-v2-6d96499dc-9-mthld	3/3 Running 0	124m	
metrics-deploy-5f4f589fc-r8sgf	2/2 Running 0	39m	
svc-test-5dd69fb8bf-8hlbb	1/1 Running 0	153m	
svc-test-5dd69fb8bf-crjkg	1/1 Running 0	153m	
svc-test-5dd69fb8bf-f25z6	1/1 Running 0	153m	
svc-test-5dd69fb8bf-g5fv5	1/1 Running 0	153m	
svc-test-5dd69fb8bf-jshpk	1/1 Running 0	153m	
svc-test-5dd69fb8bf-k26zv	1/1 Running 0	153m	
svc-test-5dd69fb8bf-mts8z	1/1 Running 0	153m	
svc-test-5dd69fb8bf-pk6qg	1/1 Running 0	153m	
svc-test-5dd69fb8bf-srbs8	1/1 Running 0	153m	
svc-test-5dd69fb8bf-wl949	1/1 Running 0	153m	

```
P5 C:\Users\leina\OneDrive\Documentos\Github\kubernetes> kubectl port-forward svc/prometheus-service 9090:9090  
Forwarding from 127.0.0.1:9090 -> 9090  
Forwarding from [::1]:9090 -> 9090
```

8. Después, una vez en Prometheus, podemos diferentes Querys, cómo se puede ver a continuación:
Métrica de peticiones http por método:



Métrica de procesos activos en cpu por segundos



Métrica de peticiones por código de estado:



Métrica de peticiones http por método y código de estado:

