

TRABAJO FINAL CURSO DOCKER KUBERNETES BÁSICO



Integrantes: GRUPO 05

- López Uribe Roosevelt Vladimir
- Quispe Paño Jeason Sergio
- Rodriguez Machaca Moises Alberto
- Velasquez Montero Javier Andres

Proyecto final: Gestión de Productos en Kubernetes

1. Ingresar al clúster de Kubernetes y ejecutar los siguientes comandos en el orden que se indica:

```
kubectl get pods -n ingress-nginx
kubectl apply -f namespace.yaml
kubectl apply -f resource-quota.yaml
kubectl apply -f configmap.yaml
kubectl apply -f deployment.yaml
kubectl apply -f service.yaml
kubectl apply -f ingress.yaml
```

```
mnadmin@Master: ~/wfinal/Demo01/archivos_yaml
-rw-rw-r-- 1 mnadmin mnadmin 260 Mar 21 15:32 configmap.yaml
-rw-rw-r-- 1 mnadmin mnadmin 737 Mar 21 15:32 deployment.yaml
-rw-rw-r-- 1 mnadmin mnadmin 433 Mar 21 15:32 ingress.yaml
-rw-rw-r-- 1 mnadmin mnadmin 60 Mar 21 15:32 namespace.yaml
-rw-rw-r-- 1 mnadmin mnadmin 212 Mar 21 15:32 resource-quota.yaml
-rw-rw-r-- 1 mnadmin mnadmin 332 Mar 21 15:32 service.yaml
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl get pods -n ingress-nginx
NAME                                READY   STATUS    RESTARTS   AGE
ingress-nginx-admission-create-7qmc 0/1     Completed 0           18h
ingress-nginx-admission-patch-7hxdk 0/1     Completed 0           18h
ingress-nginx-controller-cbb88bdbc-82sqh 1/1     Running   0           18h
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl apply -f namespace.yaml
namespace/pf-app-ns created
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl apply -f resource-quota.yaml
resourcequota/pf-app-quota created
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl apply -f configmap.yaml
configmap/pf-app-config created
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl apply -f deployment.yaml
deployment.apps/pf-app created
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl apply -f service.yaml
service/pf-service created
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/pf-ingress created
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$
```

2. Verificar funcionamiento, ejecutando los comandos:

- 2.1. funcionamiento de ingress:

```
kubectl get ingress --all-namespaces
```

```
mnadmin@Master: ~/wfinal/Demo01/archivos_yaml
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl get ingress --all-namespaces
NAMESPACE   NAME          CLASS   HOSTS          ADDRESS          PORTS   AGE
pf-app-ns   pf-ingress    nginx   empresa.com    10.99.82.122     80      6m37s
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$
```

2.2. Funcionamiento de ingress-controller:

```
kubectl get pods -n ingress-nginx
```

```
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl get pods -n ingress-nginx
NAME                                READY   STATUS    RESTARTS   AGE
ingress-nginx-admission-create-7qmc 0/1     Completed 0           19h
ingress-nginx-admission-patch-7hxdk 0/1     Completed 0           19h
ingress-nginx-controller-cbb88bdbc-82sqh 1/1     Running   0           19h
```

2.3. Modificación de type en controller:

```
kubectl edit svc ingress-nginx-controller -n ingress-nginx
```

```
selector:
  app.kubernetes.io/component: controller
  app.kubernetes.io/instance: ingress-nginx
  app.kubernetes.io/name: ingress-nginx
sessionAffinity: None
type: NodePort
status:
  loadBalancer: {}
```

2.4. Funcionamiento de ingress-nginx-controller (TYPE: NodePort):

```
kubectl get svc -n ingress-nginx
```

```
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl get svc -n ingress-nginx
NAME                                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)                                AGE
ingress-nginx-controller            NodePort    10.99.82.122   <none>         80:31048/TCP,443:30630/TCP           19h
ingress-nginx-controller-admission ClusterIP    10.100.208.244 <none>         443/TCP                               19h
```

2.5. Funcionamiento de PODs:

```
kubectl get pods -n pf-app-ns
```

```
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl get pods -n pf-app-ns
NAME                                READY   STATUS    RESTARTS   AGE
pf-app-6f957b6487-f5sz8             1/1     Running   0           9s
pf-app-6f957b6487-pp9d4             1/1     Running   0           9s
```

2.6. Funcionamiento de DEPLOYMENT:

```
kubectl get deployment -n pf-app-ns -o wide
```

```
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl get deployment -n pf-app-ns -o wide
NAME    READY   UP-TO-DATE   AVAILABLE   AGE    CONTAINERS   IMAGES                                     SELECTOR
pf-app  2/2     2            2           84s    pf-app       mrodriguezcursos0602/pf_imagen03:latest app=pf-app
```

2.7. Funcionamiento de SERVICE:

```
kubectl get svc -n pf-app-ns
```

```
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ kubectl get svc -n pf-app-ns
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
pf-service   NodePort    10.107.182.232 <none>        80:30435/TCP   22m
```

3. Realizar la prueba en un nodo master con los siguientes comandos:

Nota: Antes de realizar las pruebas en el nodo master, modificar /etc/hosts, para agregar el dominio empresa.com

```
sudo nano /etc/hosts
```

```
mnadmin@Master:~$ cat /etc/hosts
127.0.0.1      localhost empresa.com
127.0.1.1      Master

# The following lines are desirable for IPv6 capable hosts
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
```

- Verificar conectividad a empresa.com

```
ping empresa.com
```

```
mnadmin@Master:~$ ping empresa.com
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.068 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.040 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.063 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.036 ms
^C
```

3.1. Obtener listado de productos (GET)

```
curl http://empresa.com:31048/api/products
```

```
mnadmin@Master: ~/wfinal/Demo01/archivos_yaml
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ curl http://empresa.com:31048/api/products
[{"id":1,"nombre":"MACBOOK AIR","descripcion":"LAPTOP MACBOOK ","precio":3600.0}, {"id":2,"nombre":"HP 7I","descripcion":"LAPTOP HP 7I ","precio":2500.0}, {"id":3,"nombre":"DELL 5I","descripcion":"LAPTOP DELL 5I ","precio":1500.0}]mnadmin@Master:~/wfinal/Demo01/archivos_yaml$
```

3.2. Agregar un producto (POST)

- Crear un archivo json

```
sudo nano producto_post.json
```

- agregarle el siguiente contenido

```
{
    "id": null,
    "nombre" : "EPSON impresora - NUEVO",
    "descripcion" : "Tinta laser",
    "precio" : 900.55
}
```

- Ejecutar el comando:

```
curl -X POST http://empresa.com:31048/api/products \-H
"Content-Type: application/json" \--data @producto_post.json
```

```
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ curl -X POST http://empresa.com:31048/api/products \-H "Content-Type: applica
tion/json" \--data @producto_post.json
{"id":4,"nombre":"EPSON impresora - NUEVO","descripcion":"Tinta laser","precio":900.55}mnadmin@Master:~/wfinal/Demo01/archi
vos_yaml$
```

- Verificar producto registrado

```
curl http://empresa.com:31048/api/products
```

```
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ curl http://empresa.com:31048/api/products
[{"id":1,"nombre":"MACBOOK AIR","descripcion":"LAPTOP MACBOOK ","precio":3600.0}, {"id":2,"nombre":"HP 7I","descripcion":"LAPTOP HP 7I ","precio":2500.
0}, {"id":3,"nombre":"DELL 5I","descripcion":"LAPTOP DELL 5I ","precio":1500.0}, {"id":4,"nombre":"EPSON impresora - NUEVO","descripcion":"Tinta laser",
"precio":900.55}, {"id":5,"nombre":"EPSON impresora - NUEVO","descripcion":"Tinta laser","precio":900.55}]mnadmin@Master:~/wfinal/Demo01/archivos_yaml$
```

3.3. Modificar un producto (PUT)

- Crear un archivo json

```
sudo nano producto_put.json
```

- agregarle el siguiente contenido

```
{
    "id":5,
    "nombre":"TOSHIBA drive externo - SEGUNDA MANO",
    "descripcion":"Capacidad de 1 tera con puerto C",
    "precio":100.88
}
```

- Ejecutar el comando:

```
curl -X PUT http://empresa.com:31048/api/products/5 \-H
"Content-Type: application/json" \--data @producto_put.json
```

```
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ curl -X PUT http://empresa.com:31048/api/products/5 \-H "Content-Type: application/json" \--data @producto_put.json
{"id":5,"nombre":"TOSHIBA drive externo - SEGUNDA MANO","descripcion":"Capacidad de 1 tera con puerto C","precio":100.88}mnadmin@Master:~/wfinal/Demo01/archivos_yaml$
```

- Verificar producto registrado

```
curl http://empresa.com:31048/api/products
```

```
0},{id:3,"nombre":"DELL 5I","descripcion":"LAPTOP DELL 5I ","precio":1500.0}]mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ curl http://empresa.com:31048/api/products
[{"id":1,"nombre":"MACBOOK AIR","descripcion":"LAPTOP MACBOOK ","precio":3600.0},{id:2,"nombre":"HP 7I","descripcion":"LAPTOP HP 7I ","precio":2500.0},{id:3,"nombre":"DELL 5I","descripcion":"LAPTOP DELL 5I ","precio":1500.0},{id:4,"nombre":"EPSON impresora - NUEVO","descripcion":"Tinta laser","precio":900.55},{id:5,"nombre":"TOSHIBA drive externo - SEGUNDA MANO","descripcion":"Capacidad de 1 tera con puerto C","precio":100.88}]mnadmin@Master:~/wfinal/Demo01/archivos_yaml$
```

3.4. Eliminar un producto (DELETE)

- Antes de eliminar:

```
curl http://empresa.com:31048/api/products
```

```
0},{id:3,"nombre":"DELL 5I","descripcion":"LAPTOP DELL 5I ","precio":1500.0}]mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ curl http://empresa.com:31048/api/products
[{"id":1,"nombre":"MACBOOK AIR","descripcion":"LAPTOP MACBOOK ","precio":3600.0},{id:2,"nombre":"HP 7I","descripcion":"LAPTOP HP 7I ","precio":2500.0},{id:3,"nombre":"DELL 5I","descripcion":"LAPTOP DELL 5I ","precio":1500.0},{id:4,"nombre":"EPSON impresora - NUEVO","descripcion":"Tinta laser","precio":900.55},{id:5,"nombre":"TOSHIBA drive externo - SEGUNDA MANO","descripcion":"Capacidad de 1 tera con puerto C","precio":100.88}]mnadmin@Master:~/wfinal/Demo01/archivos_yaml$
```

```
curl -X DELETE http://empresa.com:31048/api/products/4
curl http://empresa.com:31048/api/products
```

```
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ curl -X DELETE http://empresa.com:31048/api/products/4
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ curl http://empresa.com:31048/api/products
[{"id":1,"nombre":"MACBOOK AIR","descripcion":"LAPTOP MACBOOK ","precio":3600.0},{id:2,"nombre":"HP 7I","descripcion":"LAPTOP HP 7I ","precio":2500.0},{id:3,"nombre":"DELL 5I","descripcion":"LAPTOP DELL 5I ","precio":1500.0},{id:5,"nombre":"TOSHIBA drive externo - SEGUNDA MANO","descripcion":"Capacidad de 1 tera con puerto C","precio":100.88}]mnadmin@Master:~/wfinal/Demo01/archivos_yaml$
```

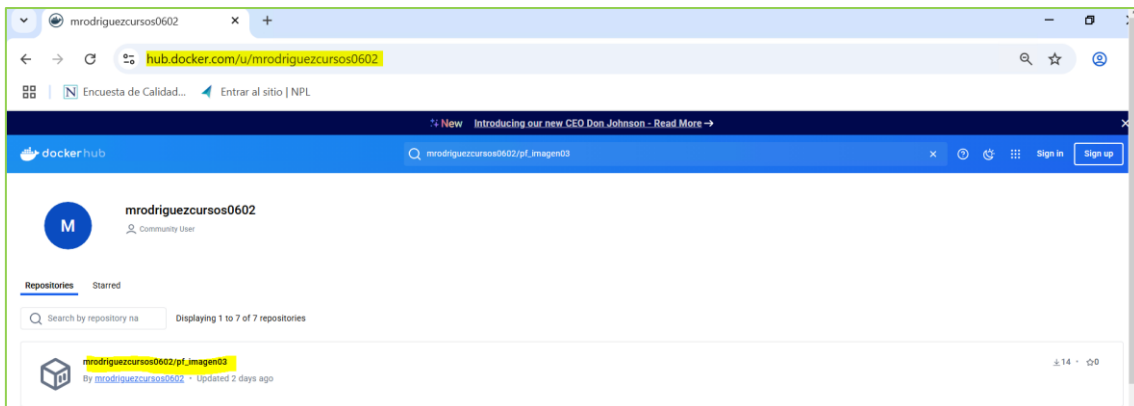
4. Otros

4.1. Pantalla ingress.yaml

```
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$ cat ingress.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: pf-ingress
  namespace: pf-app-ns
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  ingressClassName: nginx # Especifica la clase de Ingress
  rules:
  - host: empresa.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: pf-service
            port:
              number: 80
mnadmin@Master:~/wfinal/Demo01/archivos_yaml$
```

4.2. Imagen Docker

https://hub.docker.com/r/mrodriguezcursos0602/pf_imagen03



4.3. Archivo Dockerfile

```
1.. Dockerfile
0      10      20      30      40      50      60      70
# Utilizar una imagen base de Java (ajusta la versión si es necesario).
FROM openjdk:21-jdk-slim

# Crear un directorio para la aplicación.
WORKDIR /app

# Copiar el archivo JAR generado al contenedor.
ADD ./target/ms-products-0.0.1-SNAPSHOT.jar app.jar
# Exponer el puerto que usa la aplicación.
EXPOSE 7654

# Comando para ejecutar la aplicación.
ENTRYPOINT ["java", "-jar", "app.jar"]
```

Anexo:






El contenido del trabajo está subido en la ruta del github:

RUTA GITHUB: <https://github.com/javichomt/repo-equipo5>

Tipo: **PUBLICO**

Contenido:

- archivos_yaml
- ms-products.zip (código fuente comprimido)
- URL-dockerimage.txt

 archivos_yaml	Add files via upload
 ms-products	Add files via upload
 README.md	Initial commit
 URL-dockerimage.txt.txt	Add files via upload
 ms-products.zip	Add files via upload