Proyecto Final: Microservicios con Docker y Kubernetes

Índice

1. Introducción

Este proyecto tiene como finalidad demostrar el despliegue de microservicios utilizando Docker y Kubernetes sobre una instancia EC2 de AWS. Se desarrollaron dos servicios: una API en Node.js y una base de datos simple en Flask/Python. Los servicios se integran y se comunican a través de una arquitectura de contenedores.

2. Objetivo del Proyecto

Aplicar conocimientos prácticos sobre contenedores.

Construir imágenes Docker personalizadas.

Publicar imágenes en Docker Hub.

Desplegar los servicios en Kubernetes usando k3s.

Documentar y versionar el proyecto en GitHub.

Mostrar pruebas de funcionamiento para uso profesional (por ejemplo, en LinkedIn o portafolio).

3. Estructura del Proyecto

```
microservices-docker-k8s/
├── api-service/
│   ├── Dockerfile
│   ├── package.json
│   └── server.js
├── db-service/
│   ├── Dockerfile
│   └── app.py
├── k8s/
│   ├── api-deployment.yaml
│   ├── db-deployment.yaml
│   ├── nginx-deployment.yaml (opcional)
│   └── nginx-service.yaml (opcional)
├── docker-compose.yml
└── README.md
```

4. Tecnologías Utilizadas

Docker, Docker Compose, K3s, EC2 de AWS, Git y GitHub.

5. Desarrollo Paso a Paso

Docker y Docker Compose

Se crearon dos Dockerfile, uno por cada microservicio.

Se instalaron los paquetes necesarios como express y axios para el servicio API.

Se construyeron las imágenes localmente y luego se subieron a Docker Hub:

- marialix/api-service:latest

- marialix/db-service:latest

Se usó docker-compose.yml para levantar ambos servicios juntos para pruebas.

Kubernetes con K3s

Se instaló K3s en la instancia EC2 con bajo consumo de recursos.

Se desplegaron los microservicios con YAML: api-deployment.yaml, db-deployment.yaml.

Kubernetes toma las imágenes desde Docker Hub.

6. Publicación en Docker Hub

Las imágenes se subieron a Docker Hub desde máquina local:

docker build -t marialix/api-service:latest .

docker push marialix/api-service:latest

7. Pruebas de Funcionamiento

curl http://localhost:3000/api/data

[{"id":1,"name":"María"},{"id":2,"name":"Juan"},{"id":3,"name":"Lucía"}]

También se accedió desde el pod con kubectl exec y curl hacia db-service.

8. Conclusiones

Docker y Kubernetes permiten aislar, desplegar y escalar aplicaciones de forma eficiente.

Subir imágenes a Docker Hub facilita el despliegue remoto.

K3s es ideal para practicar en entornos con pocos recursos.

9. Lecciones Aprendidas

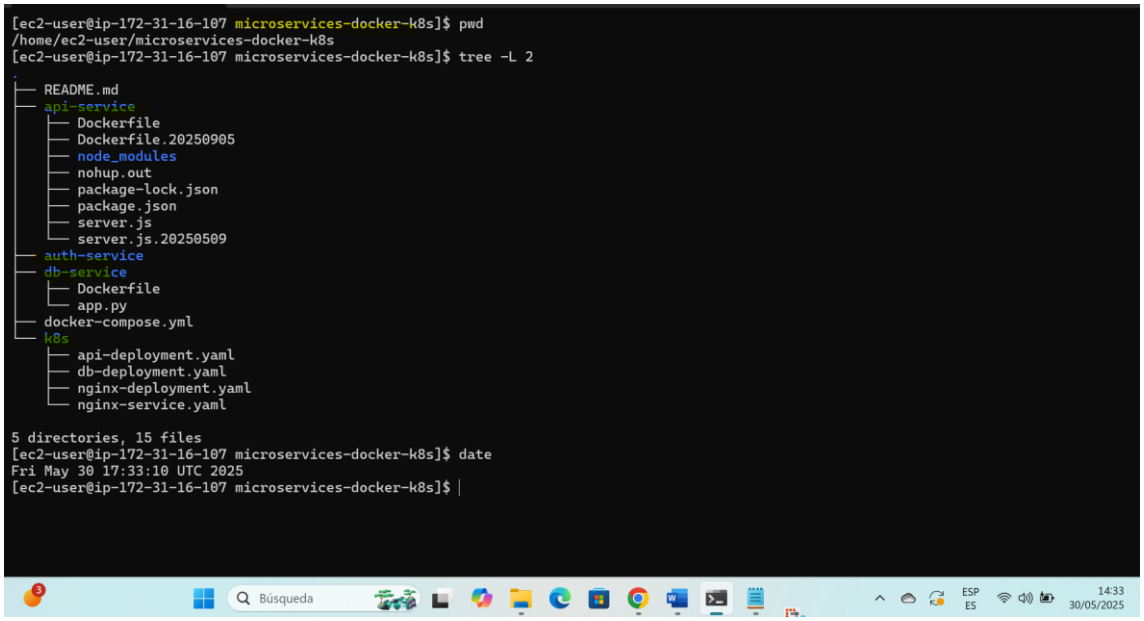
El manejo de recursos es clave en servidores como EC2 Free Tier.

Limpiar espacio periódicamente es necesario (docker system prune -a).

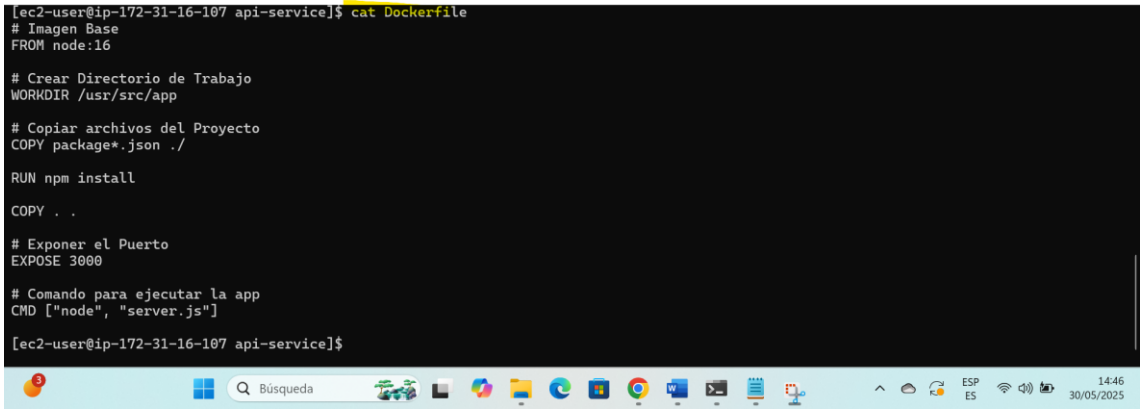
Kubernetes requiere más preparación, pero es más potente.

Documentar todo ayuda a entender y explicar el proyecto.

10. Evidencias



Docker



```
[ec2-user@ip-172-31-16-107 api-service]$ cat server.js
const express = require('express');
const axios = require('axios');
const app = express();

const dbServiceUrl = 'http://db-service:5000/data'; // URL del servicio de datos

app.get('/api/data', async (req, res) => {
  try {
    // Realizar solicitud al servicio db-service
    const response = await axios.get(dbServiceUrl);
    res.json(response.data); // Enviar los datos recibidos como respuesta
  } catch (error) {
    res.status(500).json({ error: 'No se pudo obtener los datos' });
  }
});

app.listen(3000, () => {
  console.log('Servidor corriendo en http://0.0.0.0:3000');
});

[ec2-user@ip-172-31-16-107 api-service]$
```


```
[ec2-user@ip-172-31-16-107 api-service]$ cat package.json
{
  "name": "api-service",
  "version": "1.0.0",
  "description": "Microservicio API en Node.js con Express",
  "main": "server.js",
  "scripts": {
    "start": "node server.js"
  },
  "dependencies": {
    "axios": "^1.9.0",
    "express": "^5.1.0"
  },
  "author": "marialix87",
  "license": "MIT"
}
[ec2-user@ip-172-31-16-107 api-service]$
```


```
[ec2-user@ip-172-31-16-107 microservices-docker-k8s]$ cd db-service/
[ec2-user@ip-172-31-16-107 db-service]$ ll
total 8
-rw-r--r--. 1 ec2-user ec2-user 277 May  5 17:23 Dockerfile
-rw-r--r--. 1 ec2-user ec2-user 374 May  5 17:22 app.py
[ec2-user@ip-172-31-16-107 db-service]$
```

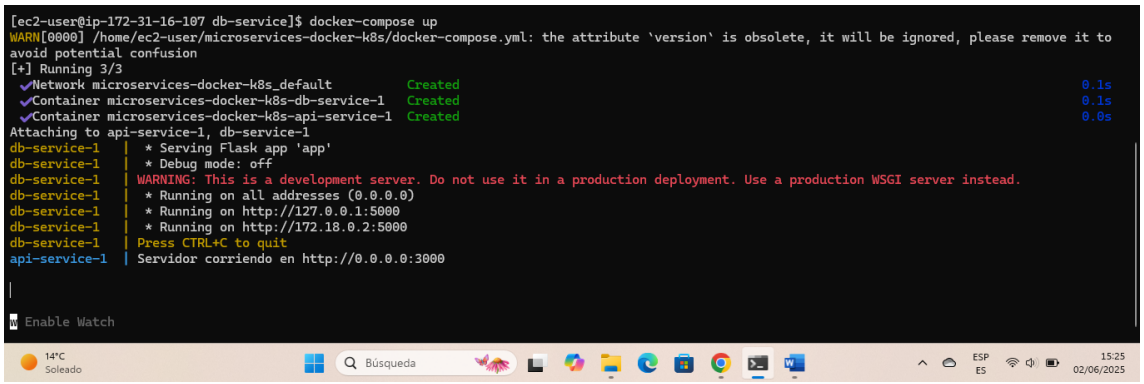

```
[ec2-user@ip-172-31-16-107 db-service]$ cat Dockerfile
# Usa imagen oficial de Python
FROM python:3.10-slim

# Establece el directorio de trabajo
WORKDIR /app

# Copia los archivos
COPY app.py /app

# Instala Flask
RUN pip install flask

# Expone el puerto 5000
EXPOSE 5000

# Comando para ejecutar la app
CMD ["python", "app.py"]

[ec2-user@ip-172-31-16-107 db-service]$
```

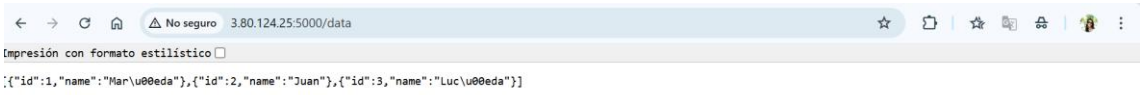
```
[ec2-user@ip-172-31-16-107 db-service]$ cat app.py
from flask import Flask, jsonify

app = Flask(__name__)

# Simula datos como si vinieran de una base de datos
fake_data = [
    {"id": 1, "name": "María"},
    {"id": 2, "name": "Juan"},
    {"id": 3, "name": "Lucía"}
]

@app.route("/data", methods=["GET"])
def get_data():
    return jsonify(fake_data)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)

[ec2-user@ip-172-31-16-107 db-service]$
```

Docker Compose

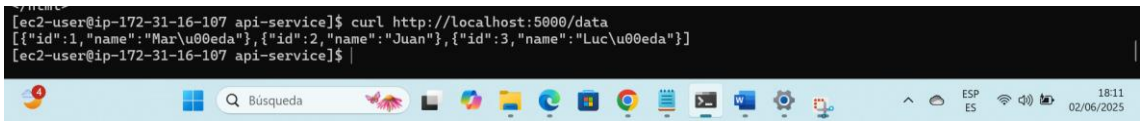
```
[ec2-user@ip-172-31-16-107 microservices-docker-k8s]$ docker-compose build
WARN[0000] /home/ec2-user/microservices-docker-k8s/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to
avoid potential confusion
Compose can now delegate builds to bake for better performance.
 To do so, set COMPOSE_BAKE=true.
[+] Building 11.9s (23/23) FINISHED                                                                                                  docker:default
 => [db-service internal] load build definition from Dockerfile                                                                                0.1s
 => => transferring dockerfile: 375B                                                                                                           0.0s
 => [db-service internal] load metadata for docker.io/library/python:3.10-slim                                                                 0.3s
 => [db-service auth] library/python:pull token for registry-1.docker.io                                                                       0.0s
 => [db-service internal] load .dockerignore                                                                                                   0.0s
 => => transferring context: 2B                                                                                                                0.0s
 => [db-service 1/4] FROM docker.io/library/python:3.10-slim@sha256:49454d2bf78a48f217eb25ecbcb4b5face313fea6a6e82706465a6990303ada2           0.0s
 => [db-service internal] load build context                                                                                                   0.0s
 => => transferring context: 468B                                                                                                              0.0s
 => CACHED [db-service 2/4] WORKDIR /app                                                                                                       0.0s
 => CACHED [db-service 3/4] COPY app.py /app                                                                                                   0.0s
 => [db-service 4/4] RUN pip install flask                                                                                                     4.4s
 => [db-service] exporting to image                                                                                                            0.2s
 => => exporting layers                                                                                                                        0.2s
 => => writing image sha256:395b309ff2166253d935ec57758c8071066b46cf10fa0dde3a53f1f7086f5831                                                   0.0s
 => => naming to docker.io/library/microservices-docker-k8s-db-service                                                                         0.0s
 => [db-service] resolving provenance for metadata file                                                                                        0.0s
 => [api-service internal] load build definition from Dockerfile                                                                               0.0s
 => => transferring dockerfile: 357B                                                                                                           0.0s
 => [api-service internal] load metadata for docker.io/library/node:18-alpine                                                                  0.3s
 => [api-service auth] library/node:pull token for registry-1.docker.io                                                                        0.0s
 => [api-service internal] load .dockerignore                                                                                                  0.0s
 => => transferring context: 2B                                                                                                                0.0s
```

```
 => [db-service] resolving provenance for metadata file                                                                                        0.0s
 => [api-service internal] load build definition from Dockerfile                                                                               0.0s
 => => transferring dockerfile: 357B                                                                                                           0.0s
 => [api-service internal] load metadata for docker.io/library/node:18-alpine                                                                  0.3s
 => [api-service auth] library/node:pull token for registry-1.docker.io                                                                        0.0s
 => [api-service internal] load .dockerignore                                                                                                  0.0s
 => => transferring context: 2B                                                                                                                0.0s
 => [api-service 1/5] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e           3.0s
 => => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e                        0.0s
 => => sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e 7.67kB / 7.67kB                                                 0.0s
 => => sha256:929b04d7c782f04f615cf785488fed452b6569f87c73ff666ad553a7554f0006 1.72kB / 1.72kB                                                 0.0s
 => => sha256:ee77c6cd7c1886ecc802ad6cedef3a8ec1ea27d1fb96162bf03dd3710839b8da 6.18kB / 6.18kB                                                 0.0s
 => => sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 3.64MB / 3.64MB                                                 0.2s
 => => sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4efea161b2b5a3d0e 40.01MB / 40.01MB                                               0.8s
 => => sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702d6e3 1.26MB / 1.26MB                                                 0.3s
 => => extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870                                                      0.6s
 => => sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e0549 446B / 446B                                                     0.3s
 => => extracting sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4efea161b2b5a3d0e                                                      1.9s
 => => extracting sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702d6e3                                                      0.1s
 => => extracting sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e0549                                                      0.0s
 => [api-service internal] load build context                                                                                                  0.8s
 => => transferring context: 4.97MB                                                                                                            0.8s
 => [api-service 2/5] WORKDIR /usr/src/app                                                                                                     0.1s
 => [api-service 3/5] COPY package*.json ./                                                                                                    0.0s
 => [api-service 4/5] RUN npm install                                                                                                          2.6s
 => [api-service 5/5] COPY . .                                                                                                                 0.2s
 => [api-service] exporting to image                                                                                                           0.3s
 => => exporting layers                                                                                                                        0.3s
 => => writing image sha256:0a9ddae1f9ea4680a00be8589308d349af5271c030cef3847575b6985479f701                                                   0.0s
 => => naming to docker.io/library/microservices-docker-k8s-api-service                                                                        0.0s
 => [api-service] resolving provenance for metadata file                                                                                       0.0s
[+] Building 2/2
 ✔ api-service  Built                                                                                                                          0.0s
 ✔ db-service   Built                                                                                                                          0.0s
[ec2-user@ip-172-31-16-107 microservices-docker-k8s]$
```


[ec2-user@ip-172-31-16-107 db-service]$ docker-compose up
WARN[0000] /home/ec2-user/microservices-docker-k8s/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 3/3
✔Network microservices-docker-k8s_default Created 0.1s
✔Container microservices-docker-k8s-db-service-1 Created 0.1s
✔Container microservices-docker-k8s-api-service-1 Created 0.0s
Attaching to api-service-1, db-service-1
db-service-1 | * Serving Flask app 'app'
db-service-1 | * Debug mode: off
db-service-1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
db-service-1 | * Running on all addresses (0.0.0.0)
db-service-1 | * Running on http://127.0.0.1:5000
db-service-1 | * Running on http://172.18.0.2:5000
db-service-1 | Press CTRL+C to quit
api-service-1 | Servidor corriendo en http://0.0.0.0:3000
Enable Watch
14°C Soleado
Búsqueda
ESP ES
15:25 02/06/2025


No seguro
3.80.124.25:5000/data
Impresión con formato estilístico
:{"id":1,"name":"Mar\u00eda"},{"id":2,"name":"Juan"},{"id":3,"name":"Luc\u00eda"}]

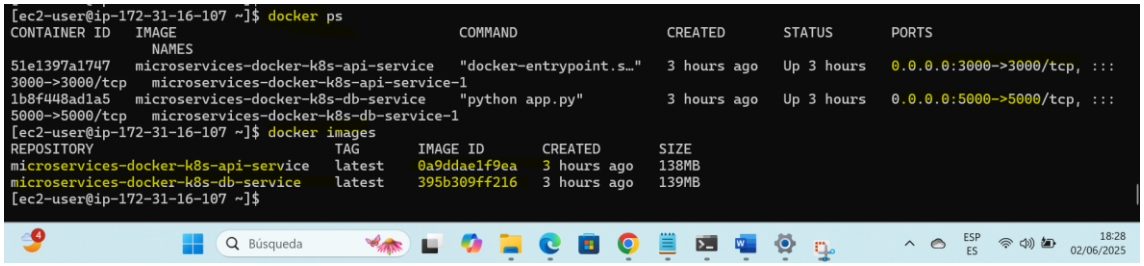

Búsqueda
ESP ES
18:07 02/06/2025


[ec2-user@ip-172-31-16-107 api-service]$ curl http://localhost:5000/data
[{"id":1,"name":"Mar\u00eda"},{"id":2,"name":"Juan"},{"id":3,"name":"Luc\u00eda"}]
[ec2-user@ip-172-31-16-107 api-service]$
Búsqueda
ESP ES
18:11 02/06/2025

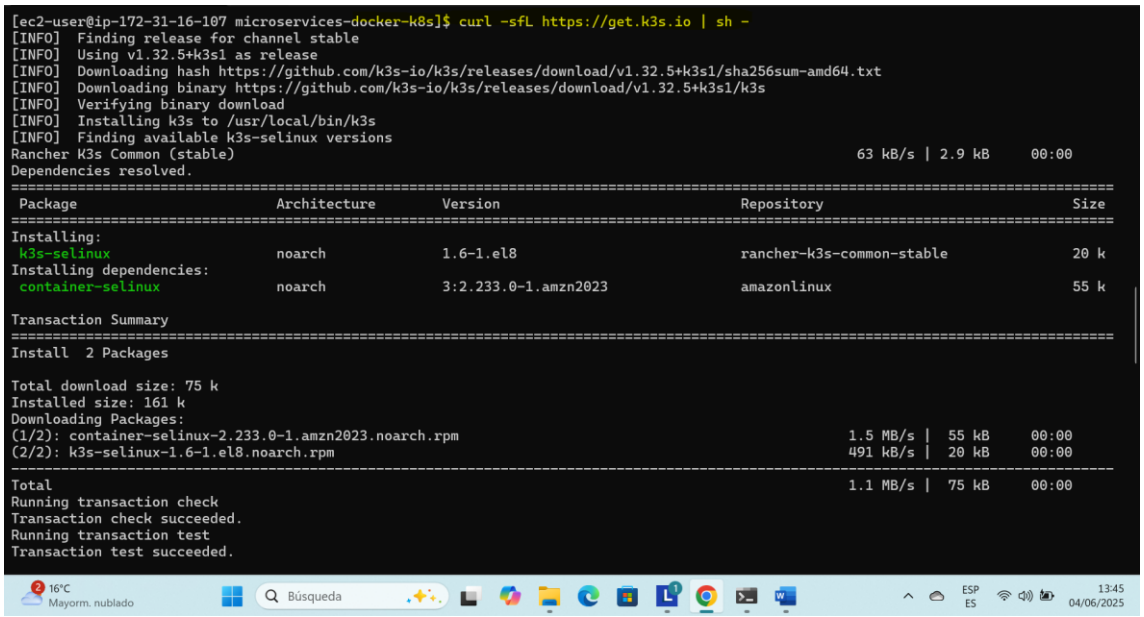

[ec2-user@ip-172-31-16-107 ~]$ curl http://localhost:3000/api/data
[{"id":1,"name":"Maria"},{"id":2,"name":"Juan"},{"id":3,"name":"Lucía"}][ec2-user@ip-172-31-16-107 ~]$
Búsqueda
ESP ES
18:25 02/06/2025


No seguro
3.80.124.25:3000/api/data
Impresión con formato estilístico
[{"id":1,"nombre":"María"},{"id":2,"nombre":"Juan"},{"id":3,"nombre":"Lucía"}]


Búsqueda
ESP
ES
18:26
02/06/2025


[ec2-user@ip-172-31-16-107 ~]$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
51e1397a1747 microservices-docker-k8s-api-service "docker-entrypoint.s…" 3 hours ago Up 3 hours 0.0.0.0:3000->3000/tcp, :::
3000->3000/tcp microservices-docker-k8s-api-service-1
1b8f448ad1a5 microservices-docker-k8s-db-service "python app.py" 3 hours ago Up 3 hours 0.0.0.0:5000->5000/tcp, :::
5000->5000/tcp microservices-docker-k8s-db-service-1
[ec2-user@ip-172-31-16-107 ~]$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
microservices-docker-k8s-api-service latest 0a9ddae1f9ea 3 hours ago 138MB
microservices-docker-k8s-db-service latest 395b309ff216 3 hours ago 139MB
[ec2-user@ip-172-31-16-107 ~]$
Búsqueda
ESP
ES
18:28
02/06/2025

Kubernetes


[ec2-user@ip-172-31-16-107 microservices-docker-k8s]$ curl -sfL https://get.k3s.io | sh -
[INFO] Finding release for channel stable
[INFO] Using v1.32.5+k3s1 as release
[INFO] Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.32.5+k3s1/sha256sum-amd64.txt
[INFO] Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.32.5+k3s1/k3s
[INFO] Verifying binary download
[INFO] Installing k3s to /usr/local/bin/k3s
[INFO] Finding available k3s-selinux versions
Rancher K3s Common (stable) 63 kB/s | 2.9 kB 00:00
Dependencies resolved.
Package Architecture Version Repository Size
Installing:
k3s-selinux noarch 1.6-1.el8 rancher-k3s-common-stable 20 k
Installing dependencies:
container-selinux noarch 3:2.233.0-1.amzn2023 amazonlinux 55 k
Transaction Summary
Install 2 Packages
Total download size: 75 k
Installed size: 161 k
Downloading Packages:
(1/2): container-selinux-2.233.0-1.amzn2023.noarch.rpm 1.5 MB/s | 55 kB 00:00
(2/2): k3s-selinux-1.6-1.el8.noarch.rpm 491 kB/s | 20 kB 00:00
Total 1.1 MB/s | 75 kB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
16°C
Mayorm. nublado
Búsqueda
ESP
ES
13:45
04/06/2025

```
[ec2-user@ip-172-31-16-107 k8s]$ cat api-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-service
spec:
  replicas: 1
  selector:
    matchLabels:
      app: api-service
  template:
    metadata:
      labels:
        app: api-service
    spec:
      containers:
        - name: api-service
          image: marialix/api-service:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 3000
          env:
            - name: DB_SERVICE_URL
              value: http://db-service:5000/data
---
apiVersion: v1
```

```
---
apiVersion: v1
kind: Service
metadata:
  name: api-service
spec:
  selector:
    app: api-service
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
  type: LoadBalancer
[ec2-user@ip-172-31-16-107 k8s]$
```

```
[ec2-user@ip-172-31-16-107 k8s]$ cat db-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: db-service
spec:
  replicas: 1
  selector:
    matchLabels:
      app: db-service
  template:
    metadata:
      labels:
        app: db-service
    spec:
      containers:
        - name: db-service
          image: marialix/db-service:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 5000
---
apiVersion: v1
```

```
---
apiVersion: v1
kind: Service
metadata:
  name: db-service
spec:
  selector:
    app: db-service
  ports:
    - protocol: TCP
      port: 5000
      targetPort: 5000
  type: ClusterIP
[ec2-user@ip-172-31-16-107 k8s]$
```

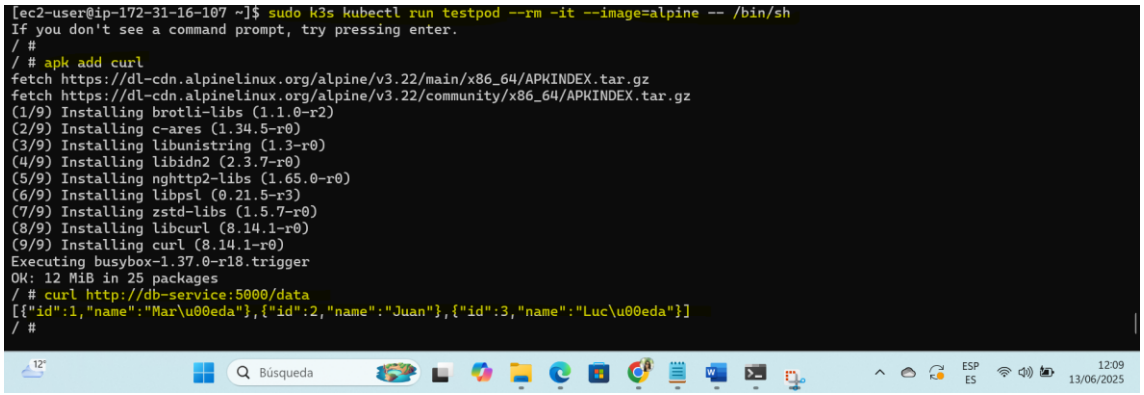
```
[ec2-user@ip-172-31-16-107 k8s]$ ll
total 16
-rw-r--r--. 1 ec2-user ec2-user 681 May 20 16:54 api-deployment.yaml
-rw-r--r--. 1 ec2-user ec2-user 572 May 20 17:03 db-deployment.yaml
-rw-r--r--. 1 ec2-user ec2-user 369 May  9 20:17 nginx-deployment.yaml
-rw-r--r--. 1 ec2-user ec2-user 174 May  9 20:17 nginx-service.yaml
[ec2-user@ip-172-31-16-107 k8s]$ sudo k3s kubectl apply -f db-deployment.yaml
deployment.apps/db-service created
service/db-service created
[ec2-user@ip-172-31-16-107 k8s]$ sudo k3s kubectl apply -f api-deployment.yaml
deployment.apps/api-service created
service/api-service created
[ec2-user@ip-172-31-16-107 k8s]$
```

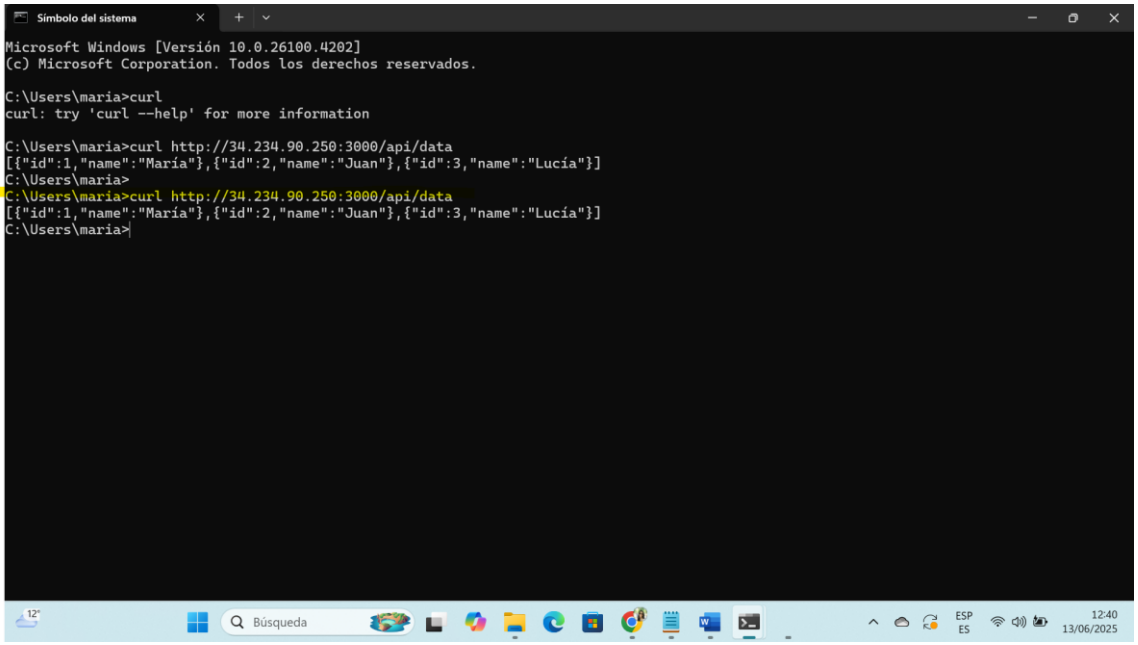

```
[ec2-user@ip-172-31-16-107 k8s]$ sudo k3s kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
api-service-6b99cf8dc8-78tmf   1/1     Running   0          8m12s
db-service-799959d8b4-6rnvk    1/1     Running   0          8m37s
[ec2-user@ip-172-31-16-107 k8s]$ sudo k3s kubectl get nodes
NAME                           STATUS   ROLES                  AGE   VERSION
ip-172-31-16-107.ec2.internal   Ready    control-plane,master   25h   v1.32.5+k3s1
[ec2-user@ip-172-31-16-107 k8s]$ sudo k3s kubectl get svc
NAME          TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)          AGE
api-service   LoadBalancer   10.43.22.31    172.31.16.107   3000:32346/TCP   8m57s
db-service    ClusterIP      10.43.81.105   <none>          5000/TCP         9m22s
kubernetes    ClusterIP      10.43.0.1      <none>          443/TCP          25h
[ec2-user@ip-172-31-16-107 k8s]$
```


```
[ec2-user@ip-172-31-16-107 k8s]$ curl http://localhost:3000/api/data
[{"id":1,"name":"Maria"},{"id":2,"name":"Juan"},{"id":3,"name":"Lucia"}][ec2-user@ip-172-31-16-107 k8s]$
[ec2-user@ip-172-31-16-107 k8s]$ date
Fri Jun 13 14:47:44 UTC 2025
[ec2-user@ip-172-31-16-107 k8s]$
```

Impresión con formato estilístico

[{"id":1,"name":"María"},{"id":2,"name":"Juan"},{"id":3,"name":"Lucía"}]





11. Autor

Nombre: María Alixandra

GitHub: https://github.com/marialix87

Proyecto: https://github.com/marialix87/microservices-docker-k8s