DESPLIEGUE DE APLICACIONES CON MÚLTIPLES CONTENEDORES USANDO DOCKER

PARTE 1: Instalar Docker y verificar si existe

```
PS C:\Users\sofia> docker --version
Docker version 27.5.1, build 9f9e405
PS C:\Users\sofia>
```

PARTE 2: Crear la aplicación

1. Crear una red de Docker

```
PS C:\Users\sofia> docker network create my-network
b76e2ba56c3b85b6af3edd257fbe1b6bb64708e8972d9b2190b4e51a0458601f
PS C:\Users\sofia> docker network ls
NETWORK ID     NAME                                                      DRIVER    SCOPE
bc8e00e0d7df   bridge                                                    bridge    local
6552adcbe2eb   host                                                      host      local
1b0e0a6c8433   localstack_localstack-docker-desktop-desktop-extension_default   bridge    local
b76e2ba56c3b   my-network                                                bridge    local
3e2c33625c1e   none                                                      null      local
a8f6b3dba0c0   wordpress-mariadb-sofia_default                           bridge    local
a865d2f39bb2   wordpress_mariadb_default                                 bridge    local
9c3394a866c5   wordpressmariadb_default                                  bridge    local
6f2b1b2661a5   wordpressmariadb_wp_net                                   bridge    local
PS C:\Users\sofia>
```

2: Lanzar la base de datos (MySQL)

```
PS C:\Users\sofia> docker run --name mysql-container --network my-network -e MYSQL_ROOT_PASSWORD=admin -e MYSQL_DATABASE
=testdb -d mysql:latest
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
cea172a6e83b: Pull complete
fa811e9a869e: Pull complete
47a2982daa21: Pull complete
634d7076afe3: Pull complete
aa8a3958f09f: Pull complete
84e4e5ea3754: Pull complete
2275c0ff11a0: Pull complete
2792ea2d4e0e: Pull complete
f488b2cd8494: Pull complete
9451290759df: Pull complete
Digest: sha256:7839322bd6c3174a699586c3ea36314c59b61b4ce01b4146951818b94aef5fd7
Status: Downloaded newer image for mysql:latest
52046c1010b0c2dc2926dfba41faabe84cafed1173c7da9802a845e8457af272
PS C:\Users\sofia> docker ps
CONTAINER ID   IMAGE                                       COMMAND                  CREATED        STATUS
    PORTS                    NAMES
52046c1010b0   mysql:latest                                "docker-entrypoint.s…"   1 second ago   Up Less than a secon
d   3306/tcp, 33060/tcp      mysql-container
4c1f2cca2e9d   localstack/localstack-docker-desktop:0.5.3   "/bin/sh -c '/servic…"   25 hours ago   Up 2 minutes
                            localstack_localstack-docker-desktop-desktop-extension-service
ac410305a8dd   wordpress:latest                            "docker-entrypoint.s…"   2 weeks ago    Up 2 minutes
    0.0.0.0:8081->80/tcp     wordpress_sofia
27d4819e9c37   mariadb:latest                              "docker-entrypoint.s…"   2 weeks ago    Up 2 minutes
    3306/tcp                 mariadb_sofia
PS C:\Users\sofia>
```

3: Backend en Node.js

3.1. Crea una carpeta llamada backend y dentro crear un archivo server.js

```
C: > Users > sofia > Desktop > backend > JS server.js > ...
const express = require("express") ;
const mysql = require("mysql2");
const app = express () ;

const db = mysql.createConnection({
    host: "db",
    user: "root",
    password: "rootpassword",
    database: "testdb"
});

app.get("/", (req, res) => {
    db.query("SELECT 'Hello from MySQL' AS message", (err, result) => {
        if (err) throw err;
        res.json(result[0]);
    });

});

app.listen(3000, () => console.log("Backend running on port 3000"));
```

3.2. En la misma carpeta, crear un Dockerfile:

```
C: > Users > sofia > Desktop > backend > Dockerfile
FROM node:18
WORKDIR /app
COPY . .
RUN npm install express mysql2
CMD ["node", "server. js"]
```

3.3. Construir la imagen y ejecutar el contenedor:

```
PS C:\Users\sofia> cd 'C:\Users\sofia\Desktop\backend'
PS C:\Users\sofia\Desktop\backend> docker build -t backend-image .
[+] Building 111.5s (10/10) FINISHED                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                   0.0s
 => => transferring dockerfile: 133B                                                                   0.0s
 => [internal] load metadata for docker.io/library/node:18                                             1.9s
 => [auth] library/node:pull token for registry-1.docker.io                                            0.0s
 => [internal] load .dockerignore                                                                      0.0s
 => => transferring context: 2B                                                                        0.0s
 => [1/4] FROM docker.io/library/node:18@sha256:df9fa4e0e39c9b97e30240b5bb1d99bdb861573a82002b2c52ac7d6b8d6d773  101.1s
 => => resolve docker.io/library/node:18@sha256:df9fa4e0e39c9b97e30240b5bb1d99bdb861573a82002b2c52ac7d6b8d6d773e   0.0s
 => => sha256:07d1b5af933d2dfc3d0dd509d6e20534825e4a537f7b006a6cb5b8e5a1f20905 24.01MB / 24.01MB                7.4s
 => => sha256:df9fa4e0e39c9b97e30240b5bb1d99bdb861573a82002b2c52ac7d6b8d6d773e 6.41kB / 6.41kB                 0.0s
 => => sha256:aa6c239d30ee04dede270729f9502389b1a9546687ce656872536340ee0a9e03 2.49kB / 2.49kB                 0.0s
 => => sha256:de20d623379fc7c7ccf845a22c3153b920d57446ba7c8e64ba25d21a60b48ad6 6.39kB / 6.39kB                 0.0s
 => => sha256:23b7d26ef1d294256da0d70ce374277b9aab5ca683015073316005cb63d33849 48.49MB / 48.49MB              35.4s
 => => sha256:1eb98adba0eb44a2e4facf9ca3626a4a66feedd0dd56d159cca90a35205744e7 64.40MB / 64.40MB               9.9s
 => => sha256:b617a119f8a27982374d94ec6eb3738ae3d38d6fc2c34c865813926cf596a621 211.33MB / 211.33MB            54.3s
 => => sha256:ee496386c5de1ce84096ca486e1aabcdf7cb8f0afd5a9b4863dbf870b340744f 3.32kB / 3.32kB               10.1s
 => => sha256:058db40e534297246fb14bc4e107d2f6ddc140494a26793e1e90696cd6b2507a 45.68MB / 45.68MB              34.4s
 => => sha256:04deb1529fda049f44f9be8d16ca833a53961813ce07eda3cefd50cd3fd74880 1.25MB / 1.25MB                35.0s
 => => sha256:3b3ca5178f3ece5f7e96b38b9c4b9c3a101a9d6777cf0f5c320f869512c80024 448B / 448B                   35.5s
 => => extracting sha256:23b7d26ef1d294256da0d70ce374277b9aab5ca683015073316005cb63d33849                     9.0s
 => => extracting sha256:07d1b5af933d2dfc3d0dd509d6e20534825e4a537f7b006a6cb5b8e5a1f20905                     2.7s
 => => extracting sha256:1eb98adba0eb44a2e4facf9ca3626a4a66feedd0dd56d159cca90a35205744e7                    11.7s
```

```
PS C:\Users\sofia\Desktop\backend> docker run --name backend-container --network my-network -p 3000:3000 -d backend-imag
e
d6a8a51e915a1f325216d63dd3bdc218453b0ceee5f2a0f59f857dfceb6a127b
PS C:\Users\sofia\Desktop\backend>
```

3.3.1. Ver las imágenes creadas:

```
PS C:\Users\sofia\Desktop\backend> docker images
REPOSITORY                          TAG             IMAGE ID       CREATED          SIZE
backend-image                       latest          01c12f9c1250   2 minutes ago    1.1GB
localstack/localstack               latest          33f282dd715c   26 hours ago     1.2GB
public.ecr.aws/lambda/python        3.9             18f051ae776a   27 hours ago     551MB
mysql                               latest          4b2d796bebc2   9 days ago       859MB
nextcloud                           30.0.8-apache   e76c48e387cc   5 weeks ago      1.28GB
redmine                             latest          9c6122d9ed64   6 weeks ago      611MB
postgres                            15              bda3cb97199d   8 weeks ago      430MB
mariadb                             latest          9f3d79eba61e   2 months ago     328MB
wordpress                           latest          29e1d310b5c6   2 months ago     701MB
httpd                               latest          83d938198316   3 months ago     148MB
hello-world                         latest          74cc54e27dc4   3 months ago     10.1kB
python                              3.11-slim       de3a6c124050   4 months ago     130MB
localstack/localstack-docker-desktop  0.5.3          70565af3b5d9   15 months ago    159MB
fauria/vsftpd                       latest          9bfb39139661   2 years ago      394MB
PS C:\Users\sofia\Desktop\backend>
```

3.3.2. Ver los contenedores en ejecución:

```
PS C:\Users\sofia\Desktop\backend> docker ps
CONTAINER ID   IMAGE                                        COMMAND                  CREATED          STATUS          PO
RTS                    NAMES
52046c1010b0   mysql:latest                                 "docker-entrypoint.s…"   16 minutes ago   Up 16 minutes   33
06/tcp, 33060/tcp      mysql-container
4c1f2cca2e9d   localstack/localstack-docker-desktop:0.5.3   "/bin/sh -c '/servic…"   26 hours ago     Up 18 minutes
                       localstack_localstack-docker-desktop-desktop-extension-service
ac410305a8dd   wordpress:latest                             "docker-entrypoint.s…"   2 weeks ago      Up 18 minutes   0.
0.0.0:8081->80/tcp     wordpress_sofia
27d4819e9c37   mariadb:latest                               "docker-entrypoint.s…"   2 weeks ago      Up 18 minutes   33
06/tcp                 mariadb_sofia
PS C:\Users\sofia\Desktop\backend>
```

4: Servidor web con Nginx

4.1. Crear una carpeta nginx y dentro un archivo default.conf:

C: > Users > sofia > Desktop > nginx > default.conf

```
server {
    listen 80;
    location / {
        proxy_pass http://backend:3000;
    }
}
```

4.2. En la carpeta nginx, crear un Dockerfile:

C: > Users > sofia > Desktop > nginx > Dockerfile

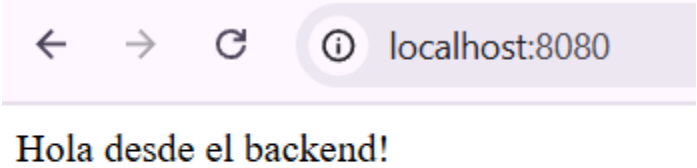
```
FROM nginx:latest
COPY default.conf /etc/nginx/conf.d/default.conf
```

4.3. Construir y ejecutar el contenedor:

```
PS C:\Users\sofia\desktop\nginx> docker build -t nginx-image .
[+] Building 1.3s (8/8) FINISHED                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                      0.0s
 => => transferring dockerfile: 106B                                      0.0s
 => [internal] load metadata for docker.io/library/nginx:latest           1.1s
 => [auth] library/nginx:pull token for registry-1.docker.io              0.0s
 => [internal] load .dockerignore                                         0.0s
 => => transferring context: 2B                                           0.0s
 => [internal] load build context                                         0.0s
 => => transferring context: 33B                                          0.0s
 => [1/2] FROM docker.io/library/nginx:latest@sha256:5ed8fcc66f4ed123     0.0s
 => CACHED [2/2] COPY default.conf /etc/nginx/conf.d/default.conf         0.0s
 => exporting to image                                                    0.0s
 => => exporting layers                                                   0.0s
 => => writing image sha256:ff4ac3687eac1cf7ce725304a141d5f5b104bad0d     0.0s
 => => naming to docker.io/library/nginx-image                            0.0s
```

```
PS C:\Users\sofia\desktop\nginx> docker run --name nginx-container --network
 my_network -p 8080:80 -d nginx-image
2ba186704fa0a4ff1d39454acb67bd8d1682cd6c88c5ac39d8b3b14191586327
PS C:\Users\sofia\desktop\nginx>
```

5. Buscar en el navegador http://localhost:8080 con el mensaje del backend



PARTE 3: Gestión y verificación

1. Verificar que todos estén corriendo:

```
PS C:\Users\sofia\desktop\nginx> docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED          STA
TUS          PORTS                    NAMES
48aa80b05e9d   nginx-image     "/docker-entrypoint.…"   6 seconds ago    Up
6 seconds    0.0.0.0:8080->80/tcp     nginx-container
bf279ae1c1aa   backend-image   "docker-entrypoint.s…"   27 seconds ago   Up
27 seconds   0.0.0.0:3000->3000/tcp   backend-container
86995dff5da8   mysql:latest    "docker-entrypoint.s…"   57 seconds ago   Up
57 seconds   3306/tcp, 33060/tcp      mysql-container
PS C:\Users\sofia\desktop\nginx>
```

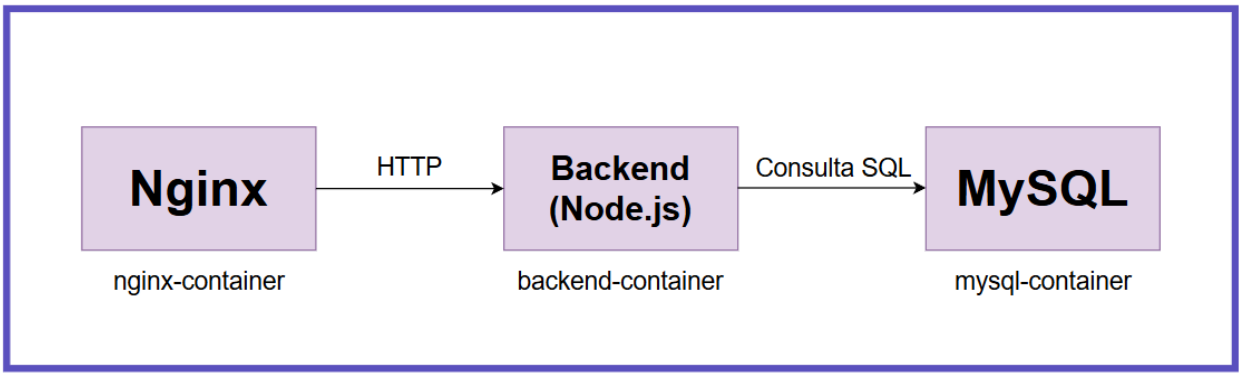
Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
mysql-container	86995dff5da8	mysql:lates		1.01%	6 minutes ago	
backend-contai	bf279ae1c1aa	backend-im	3000:3000	0%	5 minutes ago	
nginx-container	48aa80b05e9d	nginx-imag	8080:80	0%	5 minutes ago	

2. Inspecciona la red:

```
PS C:\Users\sofia\desktop\nginx> docker network inspect my-network
```


```
        "Containers": {
            "48aa80b05e9db8db4ddaf43259f578fe2ef3aece23fca2e19aaee8b4f932617
d": {
                "Name": "nginx-container",
                "EndpointID": "b3b844aca32838a87dfc867972f153d82879290395a9d
7c7160255f424be2b87",
                "MacAddress": "02:42:ac:16:00:04",
                "IPv4Address": "172.22.0.4/16",
                "IPv6Address": ""
            },
            "86995dff5da8222a894c4103394dd8110a6c0e12ba67cc68956ed4bb319904b
9": {
                "Name": "mysql-container",
                "EndpointID": "550842d45f4a42b574216ab3909e1531e06662f471bfc
f5150bc2ca6216a7212",
                "MacAddress": "02:42:ac:16:00:02",
                "IPv4Address": "172.22.0.2/16",
                "IPv6Address": ""
            },
            "bf279ae1c1aa5dc56a1874c6b776e18a303e497f914e39e9a2c7401a66db973
7": {
                "Name": "backend-container",
                "EndpointID": "b649c147ef65ae274b3a56dfecc3c7f93c948d0fb274d
45242b44ca994234cfb",
                "MacAddress": "02:42:ac:16:00:03",
                "IPv4Address": "172.22.0.3/16",
                "IPv6Address": ""
```

PARTE 4: Demostración



En este proyecto se han creado tres contenedores con Docker.

- Uno de ellos contiene una base de datos **MySQL**, donde se podrían guardar datos.
- El segundo contenedor es un **backend** hecho en Node.js, que responde con un mensaje simple.
- El tercero es un servidor **Nginx**, actúa como intermediario entre el navegador y el backend.

Los tres contenedores están conectados a la misma red, lo que permite que se comuniquen entre sí.

Cuando se accede a **localhost:8080** desde el navegador, Nginx redirige la petición al backend, y este devuelve su respuesta.