



Universidad de Guadalajara.
Centro Universitario de Ciencias Exactas e Ingenierías.



Ejercicio 9

Microservicios con Docker

Computación tolerante a fallas NRC 179961

Nombre del alumno: Loredó Padilla Orlando Javier

Código: 217560328

Departamento: Departamento de ciencias computacionales

Nombre del maestro: López Franco Michel Emanuel

Carrera: Ingeniería en computación

Sección: D06

Ciclo: 23B

Fecha: 6/11/2023.

Introducción

Genera una aplicación utilizando microservicios

Programa

En esta práctica se intentó usar Docker para crear microservicios, en sí, pequeñas aplicaciones que se deriven de una principal. Para ello se intentó usar un archivo de Docker compose en el cual se especificaría la pagina principal y dos que se derivan de esta en los puertos 3000 y 30001.

```
🐳 docker-compose.yml
1  version: "3.9"
2  services:
3    app1:
4      build: .
5      ports:
6        - "3000:3000"
7      links:
8        - mongo1
9    app2:
10     build: .
11     ports:
12       - "3001:3001"
13     links:
14       - mongo1
15   mongo1:
16     image: mongo
17     ports:
18       - "27017:27017"
19     environment:
20       - MONGO_INITDB_ROOT_USERNAME=orla
21       - MONGO_INITDB_ROOT_PASSWORD=password
22     volumes:
23       - mongo-data:/data/db
24       # mysql -> /var/lib/mysql
25       # postgres -> /var/lib/postgresql/data
26
```

La idea es que en estas dos instancias se ejecutará el mismo archivo index y de esta manera se pudiera estar reduciendo la cantidad de gente al dividir la estancia de estos en dos paginas iguales pero separadas.

```
JS index.js > ...
1  import express from 'express'
2  import mongoose from 'mongoose'
3
4  const Animal = mongoose.model('Animal', new mongoose.Schema({
5    tipo: String,
6    estado: String,
7  }))
8
9  const app = express()
10
11  mongoose.connect('mongodb://orla:password@mongo1:27017/miapp?authSource=admin')
12
13  app.get('/', async (_req, res) => {
14    res.send("Saludos!");
15  })
16
17  app.listen(3000, () => console.log('listening...'))
18
```

Ejecutamos el comando de Docker compose up para cargar la imagen y esta se empezará a descargar y compilar automáticamente. Podemos apreciar como se cargan los logs. Además si ejecutamos el comando Docker ps se ve que se cargó.

```
src-mongo1-1 | {"t":{"$date":"2023-11-14T01:15:21.316+00:00"},"s":"I", "c":"REPL", "id":6015317, "ctx":"initandlisten","msg":"Setting new configuration state","attr":{"newState":"ConfigReplicationDisabled","oldState":"ConfigPreStart"}}
src-mongo1-1 | {"t":{"$date":"2023-11-14T01:15:21.317+00:00"},"s":"I", "c":"STORAGE", "id":22262, "ctx":"initandlisten","msg":"Timestamp monitor starting"}
src-mongo1-1 | {"t":{"$date":"2023-11-14T01:15:21.352+00:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on","attr":{"address":"/tmp/mongodb-27017.sock"}}
src-mongo1-1 | {"t":{"$date":"2023-11-14T01:15:21.352+00:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on","attr":{"address":"0.0.0.0"}}
src-mongo1-1 | {"t":{"$date":"2023-11-14T01:15:21.352+00:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Waiting for connections","attr":{"port":27017,"ssl":"off"}}
src-mongo1-1 | {"t":{"$date":"2023-11-14T01:16:21.772+00:00"},"s":"I", "c":"INDEX", "id":5479200, "ctx":"TTLMonitor","msg":"Deleted expired documents using index","attr":{"namespace":"config.system.sessions","index":"_lsidTTLIndex","numDeleted":0,"durationMillis":128}}
[]
```

Seleccionar Símbolo del sistema

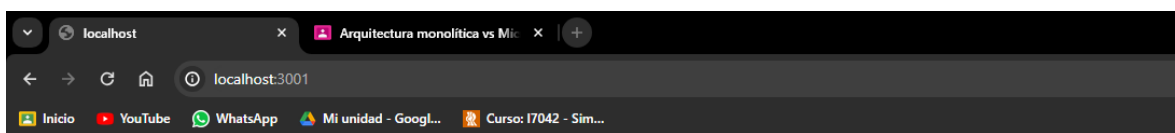
```
C:\Users\orlan> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
fa4d11d2b3fc   mongo    "docker-entrypoint.s..."  9 minutes ago Up 9 minutes   0.0.0.0:27017->27017/tcp           src-mongo1-1

C:\Users\orlan>
```

Además, en Docker se muestran como se crearon los dos contenedores para cada una de las aplicaciones.

<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	src-app1 ef5012a626df	latest	In use	16 seconds ago	1.09 GB	
<input type="checkbox"/>	src-app2 4fd559a1fc93	latest	In use	16 seconds ago	1.09 GB	

Sin embargo, al querer cargar la página con el localhost este no respondía, aunque en la consola del visual code los logs se actualizaban cada vez que se solicitaba entrar a la página.



No se puede acceder a este sitio

localhost rechazó la conexión.

Intenta:

- Comprobar la conexión.
- [Comprobar el proxy y el firewall.](#)

ERR_CONNECTION_REFUSED

Volver a cargar

Detalles

```
msg":"Slow query","attr":{"type":"command","ns":"config.system.sessions","command":{"listIndexes":"system.sessions","cursor":{"$db":"c
onfig"},"numYields":0,"reslen":245,"locks":{"FeatureCompatibilityVersion":{"acquireCount":{"r":1}},"Global":{"acquireCount":{"r":1}},"s
torage":{"cpuNanos":166798300,"protocol":"op_msg","durationMillis":415}}
src-mongo1-1 | {"t":{"$date":"2023-11-14T01:35:24.599+00:00"},"s":"I", "c":"COMMAND", "id":51803, "ctx":"LogicalSessionCacheReap","
msg":"Slow query","attr":{"type":"command","ns":"config.transactions","command":{"find":"transactions","filter":{"lastWriteDate":{"$lt":
{"$date":"2023-11-14T01:05:23.827Z"}}},"projection":{"_id":1,"sort":{"_id":1},"readConcern":{"$db":"config"},"planSummary":"EOF","pla
nningTimeMicros":104059,"keysExamined":0,"docsExamined":0,"nBatches":1,"cursorExhausted":true,"numYields":0,"nreturned":0,"queryFramewor
k":"sbe","reslen":108,"locks":{"FeatureCompatibilityVersion":{"acquireCount":{"r":1}},"Global":{"acquireCount":{"r":1}},"readConcern":{"
"provenance":"implicitDefault"},"storage":{"cpuNanos":136054300,"protocol":"op_msg","durationMillis":288}}
src-mongo1-1 | {"t":{"$date":"2023-11-14T01:36:11.451+00:00"},"s":"I", "c":"INDEX", "id":5479200, "ctx":"TTLMonitor","msg":"Deleted
expired documents using index","attr":{"namespace":"config.system.sessions","index":"lsidTTLIndex","numDeleted":0,"durationMillis":129}
}
src-mongo1-1 | {"t":{"$date":"2023-11-14T01:39:13.482+00:00"},"s":"I", "c":"INDEX", "id":5479200, "ctx":"TTLMonitor","msg":"Deleted
expired documents using index","attr":{"namespace":"config.system.sessions","index":"lsidTTLIndex","numDeleted":0,"durationMillis":167}
}
```

Conclusión

Ya habíamos utilizado Docker para crear una página anteriormente, ahora lo usamos para crear microservicios, esta es una práctica que se relaciona con el tema de tolerante a fallas ya que nos permitirá crear diferentes páginas pequeñas la cuales servirán como un respaldo de sus compañeras, explicándolo, de 3 páginas si la primera falla las otras dos seguirán funcionando.

Hacer servicios es una práctica muy útil si queremos tener mayor seguridad en nuestros sistemas por eso este trabajo sirvió como una buena introducción para saber como crearlos y poder sacarles provecho.

Referencias

- HolaMundo (2022). Aprende Docker ahora! curso completo gratis desde cero!. Recuperado el 18/10/2023 en: <https://youtu.be/4Dko5W96WHg?si=rq-slhaXuYqZCkMF>