

Manual: Configuración Personalizada de MongoDB con Docker

Este manual describe los pasos necesarios para montar un contenedor de MongoDB personalizado utilizando Docker y Docker Compose. Incluirá autenticación, la configuración de archivos `.env`, `Dockerfile` y `docker-compose.yml`, y cómo interactuar con MongoDB desde el contenedor.

1. Montar una Imagen de MongoDB Genérica

1. Descargar la imagen oficial de MongoDB:

```
docker pull mongo
```

2. Ejecutar un contenedor con MongoDB:

```
docker run --name mi_mongo -d -p 27017:27017 mongo
```

3. Ingresar al contenedor y ejecutar `mongo`:

```
docker exec -it mi_mongo bash
mongo
```

Si se presenta el error:

```
/bin/sh: 1: mongo: not found
```

Significa que el cliente `mongo` no está disponible dentro del contenedor. A continuación, una alternativa para conectarse.

2. Alternativa: Cliente MongoDB Compass

Si no puedes acceder al cliente `mongo` dentro del contenedor, puedes utilizar **MongoDB Compass** para conectarte a la base de datos:

1. Descargar MongoDB Compass desde [aquí](#).

2. Iniciar la aplicación y utilizar la siguiente cadena de conexión:

```
mongodb://localhost:27017/
```

Nota: No es necesario configurar una clave en este punto.

3. Crear una Configuración Personalizada del Contenedor MongoDB

Vamos a crear un contenedor personalizado con autenticación y una estructura más flexible para el desarrollo. Sigue estos pasos:

3.1. Crear Archivos de Configuración

1. **Archivo .env:** Este archivo almacena las variables de entorno para MongoDB.

```
MONGO_INITDB_ROOT_USERNAME=admin
MONGO_INITDB_ROOT_PASSWORD=abcd1234
MONGO_PORT=27017
```

2. **Archivo Dockerfile:** Este archivo define cómo se construirá la imagen personalizada.

```
# Utiliza la imagen oficial de MongoDB 4.4
FROM mongo:4.4

# Exponer el puerto de MongoDB
EXPOSE 27017

# Comando de inicio para MongoDB
CMD ["mongod", "--bind_ip_all"]
```

3. **Archivo docker-compose.yml:** Este archivo configura el contenedor con la autenticación habilitada y el mapeo de volúmenes para persistencia de datos.

```
version: '3.8'

services:
  mongodb:
    container_name: serve-mongodb
    build:
      context: .
      dockerfile: Dockerfile # Utiliza el Dockerfile anterior
    restart: always
    environment:
      MONGO_INITDB_ROOT_USERNAME: ${MONGO_INITDB_ROOT_USERNAME}
      MONGO_INITDB_ROOT_PASSWORD: ${MONGO_INITDB_ROOT_PASSWORD}
    ports:
      - "27017:27017"
    volumes:
      - mongodb_data:/data/db
```

```

- "/home/ubuntu/db_backups:/backups/mongodb"
networks:
- network_local_server
command: ["mongod", "--bind_ip_all"]
healthcheck:
  test: ["CMD-SHELL", "mongo --eval 'db.runCommand({ ping: 1 })'"]
  interval: 30s
  timeout: 10s
  retries: 5
labels:
- com.corhuila.group=databases

volumes:
  mongodb_data:
    driver: local

networks:
  network_local_server:
    external: true

```

3.2. Montar el Contenedor Personalizado

1. Accede a la carpeta donde se encuentran los archivos `.env`, `Dockerfile` y `docker-compose.yml` (e.g., **sesión 2 - mongo**).
2. Ejecuta los siguientes comandos para construir y levantar el contenedor:

```

docker-compose down
docker-compose up -d --build

```

3.3. Ingresar al Contenedor y Conectarse a MongoDB

1. Accede al contenedor usando el comando `exec`:

```

docker exec -it serve-mongodb bash

```

2. Auténticate con las credenciales configuradas:

```

mongo -u "admin" -p "abcd1234" --authenticationDatabase "admin"

```

4. Probar la Base de Datos MongoDB

Una vez conectado al contenedor, puedes realizar las siguientes operaciones:

1. Crear una base de datos y una colección:

```
use Curso
db.createCollection("Aprendices")
```

2. Insertar un documento en la colección:

```
db.Aprendices.insert({
  "nombre": "María Gomez",
  "edad": 20,
  "email": "maria.gomez@example.com",
  "curso": "Programación",
  "fechaRegistro": "2024-09-18"
})
```

3. Consultar los datos insertados:

```
db.Aprendices.find().pretty()
```

5. Notas Finales

Este manual te permite configurar y ejecutar una instancia de MongoDB personalizada en Docker. Asegúrate de ajustar las configuraciones según las necesidades de tu proyecto.