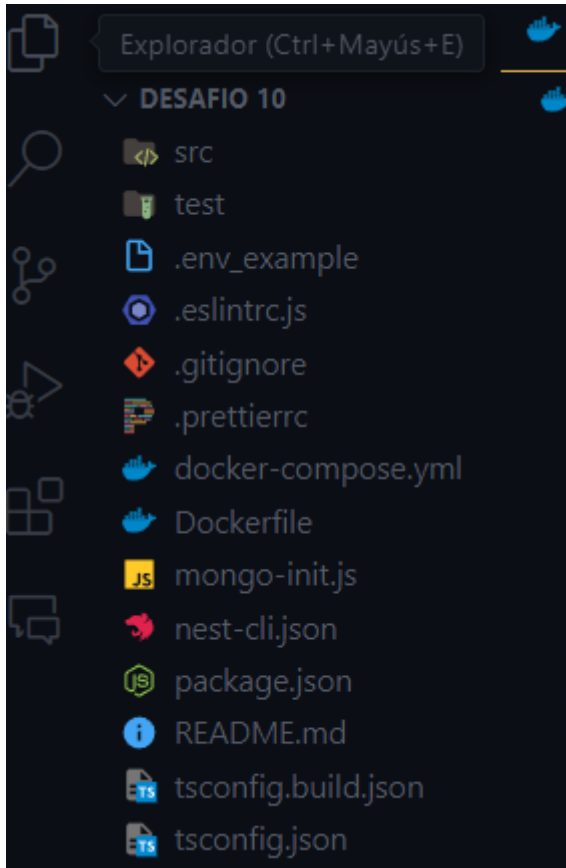


Desafío 10

1- Comenzamos el desafío 10 creando un **Dockerfile** específico para la aplicación. Luego, integraremos esa imagen en un archivo **Docker Compose**, junto con la imagen de nuestra base de datos. La estructura que utilizaremos es la siguiente



src/: Directorio que contiene el código fuente de la aplicación.

test/: Directorio que contiene los archivos de pruebas automatizadas.

.env_example: Archivo de ejemplo que muestra las variables de entorno necesarias para el proyecto.

docker-compose.yml: Archivo que define los servicios **Docker** (aplicación y base de datos).

Dockerfile: Archivo que da instrucciones para construir la imagen, **Docker** de la aplicación.

mongo-init.js: Script que se utiliza para inicializar **MongoDB** con datos iniciales.

package.json: Archivo que define las dependencias del proyecto y scripts de npm.

tsconfig.json: Archivo de configuración del compilador **TypeScript**.

2- Seguidamente seguimos con la creación de nuestro archivo **Dockerfile**, al cual le daremos las instrucciones para la creación de nuestra aplicación.

```
Dockerfile
Dockerfile
1  # imagen Node.js
2  FROM node:18-alpine AS builder
3
4  # directorio de trabajo
5  WORKDIR /app
6
7  # Copia archivos package.json package-lock.json
8  COPY package*.json ./
9
10 # Instala las dependencias
11 RUN npm install
12
13 #Instala typescript globalmente en la etapa de construccion
14 RUN npm install -g typescript
15
16 # Copia archivos de la aplicación
17 COPY . .
18
19 #compila typescript a javascript
20 RUN tsc
21
22 #Etapa 2: imagen final
23 FROM node:18-alpine
24
25 #Establece directorio de trabajo
26 WORKDIR /app
27
28 #Copia los archivos necesarios en etapa de construccion
29 COPY --from=builder /app/dist ./dist
30 COPY --from=builder /app/node_modules ./node_modules
31 COPY --from=builder /app/package*.json ./
32
33 # Expone el puerto en el que corre la aplicación
34 EXPOSE 3000
35
36 # Comando para iniciar la aplicación
37 CMD ["npm", "run", "start:dev"]
```

Etapas del Dockerfile

1. **builder:** instala Node.js, configura las dependencias y compila el código **TypeScript**.
2. **runtime:** Utiliza una imagen ligera con **Node.js** y los archivos que necesita para ejecutarla.

3- El siguiente paso es la creación de nuestro archivo **docker-compose.yml**

```
docker-compose.yml X
docker-compose.yml
1  version: '3'
2
3  services:
4    app:
5      build:
6        context: .
7        dockerfile: Dockerfile
8      ports:
9        - "3000:3000"
10     depends_on:
11       - mongodb
12     environment:
13       MONGO_DB_URI: mongodb://myuser:mypassword@mongodb:27017/mydatabase
14       MONGO_DB_NAME: mydatabase
15       MONGO_DB_USER: myuser
16       MONGO_DB_PASS: mypassword
17     volumes:
18       - ../app
19       - /app/node_modules
20
21     mongodb:
22       image: mongo:6.0
23       ports:
24         - "27017:27017"
25       environment:
26         MONGO_INITDB_ROOT_USERNAME: root
27         MONGO_INITDB_ROOT_PASSWORD: rootpassword
28       volumes:
29         - mongo-data:/data/db
30         - ./mongo-init.js:/docker-entrypoint-initdb.d/mongo-init.js:ro
31
32  volumes:
33    mongo-data:
```

Version '3': Version del formato del archivo de Docker compose

app: Servicio que define la aplicación Node.js.

- **depends_on:** Garantiza que MongoDB se ejecute antes de iniciar la aplicación.
- **MONGO_URI:** La URI de conexión para MongoDB.

mongodb: Servicio que define la base de datos MongoDB.

- **volumes:** Persisten los datos de MongoDB y permiten la inicialización mediante mongo-init.js.

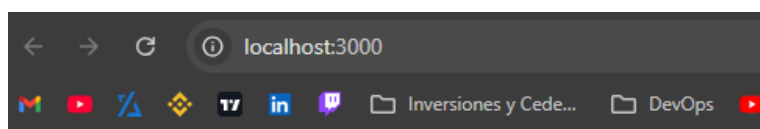
4- Dentro del docker compose agregamos un archivo js para iniciar la base de datos con un usuario y pass, llamado **mongo-init.js**

```
JS mongo-init.js X
JS mongo-init.js > ...
1 db = db.getSiblingDB('mydatabase');
2 db.createUser({
3   user: "myuser",
4   pwd: "mypassword",
5   roles: [{ role: "readWrite", db: "mydatabase" }]
6 });
```

5- Al tener nuestro **docker.compose.yml** finalmente configurado, junto con **Dockerfile** de nuestra app y el archivo para iniciar nuestra base de datos, seguimos con el levantamiento de nuestra aplicación usando docker compose y lo hacemos con el comando: **docker-compose up --build**. Al ejecutar ese comando en la terminal de Powershell crea las imágenes e inicia los contenedores.

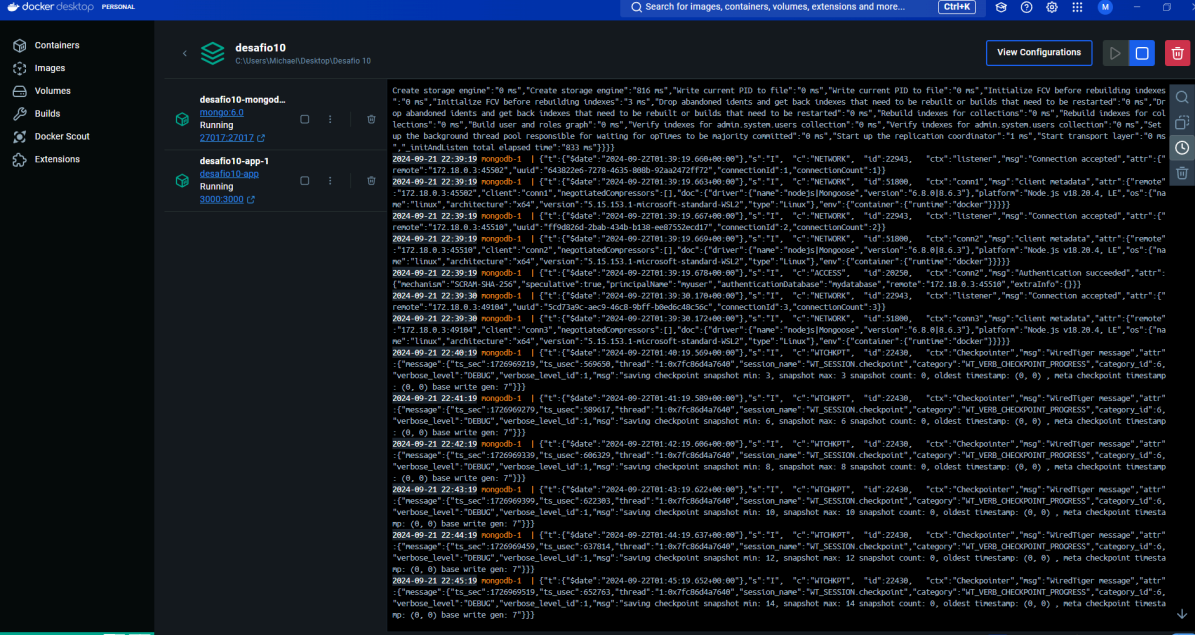
```
Simbolo del sistema - docker
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.571+00:00"},"s":"I", "c":"REPL", "id":"5853300", "ctx":"","msg":"current featureCompatibilityVersion value", "attr":{"featureCompatibilityVersion":
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.571+00:00"},"s":"I", "c":"STORAGE", "id":"5071100", "ctx":"","msg":"Clearing temp directory"}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.572+00:00"},"s":"I", "c":"CONTROL", "id":"20836", "ctx":"","msg":"Flow Control is enabled on this deployment"}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.572+00:00"},"s":"I", "c":"FTDC", "id":"20625", "ctx":"","msg":"Initializing full-time diagnostic data capture", "attr":{"dataDirectory":"/data/d
b/diagnostic.data"}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.574+00:00"},"s":"I", "c":"REPL", "id":"6015317", "ctx":"","msg":"Setting new configuration state", "attr":{"newState":"ConfigReplicationDisabled",
"oldState":"ConfigPreStart"}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.575+00:00"},"s":"I", "c":"STORAGE", "id":"22262", "ctx":"","msg":"Timestamp monitor starting"}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.576+00:00"},"s":"I", "c":"NETWORK", "id":"23015", "ctx":"","msg":"Listening on", "attr":{"address":"/tmp/mongodb-27017.sock"}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.576+00:00"},"s":"I", "c":"NETWORK", "id":"23015", "ctx":"","msg":"Listening on", "attr":{"address":"0.0.0.0"}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.576+00:00"},"s":"I", "c":"NETWORK", "id":"23016", "ctx":"","msg":"Waiting for connections", "attr":{"port":27017,"ssl":"off"}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.576+00:00"},"s":"I", "c":"CONTROL", "id":"8023003", "ctx":"","msg":"mongod startup complete", "attr":{"summaryOfTimeElapsed":{"Startup from clean
shutdown?":true,"Statistics":{"Transport layer setup":"0 ms","Run initial syncer crash recovery":"0 ms","Create storage engine lock file in the data directory":"0 ms","Create storage engine lock file in the da
ta directory":"0 ms","Get metadata describing storage engine":"0 ms","Get metadata describing storage engine":"0 ms","Validate options in metadata against current startup options":"0 ms","Validate options in m
etadata against current startup options":"0 ms","Initialize FCV before rebuilding indexes":"3 ms","Drop abandoned indexes and get back indexes that need to be rebuilt or builds that need to be restarted":"0 ms","Rebuild indexes for collections":"0 ms","Rebuild indexes for collections":"0 ms","Build user and roles graph":"0 ms","Ve
rify indexes for admin.system.users collection":"0 ms","Verify indexes for admin.system.users collection":"0 ms","Set up the background thread pool responsible for waiting for opfiles to be majority committed
":"0 ms","Start up the replication coordinator":"1 ms","Start transport layer":"0 ms","_initAndListen total elapsed time":"833 ms"}}}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.669+00:00"},"s":"I", "c":"NETWORK", "id":"22993", "ctx":"","msg":"Connection accepted", "attr":{"remote":"172.18.0.3:45502","uid":"643822e6-7278-4635-8
08b-92aa2472f72","connectionId":1,"connectionCount":1}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.663+00:00"},"s":"I", "c":"NETWORK", "id":"51800", "ctx":"","msg":"client metadata", "attr":{"remote":"172.18.0.3:45502","client":"conn1","negotiatedCompress
sors":[],"doc":{"driver":{"name":"nodejs/mongoose","version":"6.8.0@6.6.3"},"platform":"Node.js v18.20.4, LE","os":{"name":"Linux","architecture":"x64","version":"5.15.153.1-microsoft-standard-WSL2","type":"Li
nux"},"env":{"container":{"runtime":"docker"}}}}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.667+00:00"},"s":"I", "c":"NETWORK", "id":"22993", "ctx":"","msg":"Connection accepted", "attr":{"remote":"172.18.0.3:45510","uid":"fff9d826d-2bab-43db-b
138-ee7552ced17","connectionId":2,"connectionCount":2}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.669+00:00"},"s":"I", "c":"NETWORK", "id":"51800", "ctx":"","msg":"client metadata", "attr":{"remote":"172.18.0.3:45510","client":"conn2","negotiatedCompress
sors":[],"doc":{"driver":{"name":"nodejs/mongoose","version":"6.8.0@6.6.3"},"platform":"Node.js v18.20.4, LE","os":{"name":"Linux","architecture":"x64","version":"5.15.153.1-microsoft-standard-WSL2","type":"Li
nux"},"env":{"container":{"runtime":"docker"}}}}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:19.678+00:00"},"s":"I", "c":"ACCESS", "id":"20260", "ctx":"","msg":"Authentication succeeded", "attr":{"mechanism":"SCRAM-SHA-256","speculative":true,"princi
palName":"myuser","authenticationDatabase":"mydatabase","remote":"172.18.0.3:45502","userInfo":{"}}}
app-1 | [Next] 29 - 09/22/2024, 1:39:19 AM LOG [InstanceLoader] MongooseCoreModule dependencies initialized +527ms
app-1 | [Next] 29 - 09/22/2024, 1:39:19 AM LOG [RoutesResolver] ApplicationController [/]: +4ms
app-1 | [Next] 29 - 09/22/2024, 1:39:19 AM LOG [RouterExplorer] Mapped 1/ GET route +2ms
app-1 | [Next] 29 - 09/22/2024, 1:39:19 AM LOG [NextApplication] Next application successfully started +2ms
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:30.179+00:00"},"s":"I", "c":"NETWORK", "id":"22993", "ctx":"","msg":"Connection accepted", "attr":{"remote":"172.18.0.3:49104","uid":"5cd73a9c-aec9-46c8-9
bff-b0ed6cd48c56c","connectionId":3,"connectionCount":3}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:39:30.172+00:00"},"s":"I", "c":"NETWORK", "id":"51800", "ctx":"","msg":"client metadata", "attr":{"remote":"172.18.0.3:49104","client":"conn3","negotiatedCompress
sors":[],"doc":{"driver":{"name":"nodejs/mongoose","version":"6.8.0@6.6.3"},"platform":"Node.js v18.20.4, LE","os":{"name":"Linux","architecture":"x64","version":"5.15.153.1-microsoft-standard-WSL2","type":"Li
nux"},"env":{"container":{"runtime":"docker"}}}}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:40:19.569+00:00"},"s":"I", "c":"WTCHECKPT", "id":"22430", "ctx":"","msg":"Checkpoint message", "attr":{"message":{"ts_sec":"1726969219","ts_usec":"569650","thre
ad":"110x7fc6d4a76408","session_name":"WT_SESSION_checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":"6","verbose_level":"DEBUG","verbose_level_id":"1","msg":"saving checkpoint snapshot min: 3, sna
pshot max: 3 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 7"}}}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:41:19.589+00:00"},"s":"I", "c":"WTCHECKPT", "id":"22430", "ctx":"","msg":"Checkpoint message", "attr":{"message":{"ts_sec":"1726969279","ts_usec":"589617","thre
ad":"110x7fc6d4a76408","session_name":"WT_SESSION_checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":"6","verbose_level":"DEBUG","verbose_level_id":"1","msg":"saving checkpoint snapshot min: 6, sna
pshot max: 6 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 7"}}}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:42:19.606+00:00"},"s":"I", "c":"WTCHECKPT", "id":"22430", "ctx":"","msg":"Checkpoint message", "attr":{"message":{"ts_sec":"1726969339","ts_usec":"606329","thre
ad":"110x7fc6d4a76408","session_name":"WT_SESSION_checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":"6","verbose_level":"DEBUG","verbose_level_id":"1","msg":"saving checkpoint snapshot min: 8, sna
pshot max: 8 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 7"}}}}
mongo-d-1 | {"t":{"$date":"2024-09-22T01:43:19.622+00:00"},"s":"I", "c":"WTCHECKPT", "id":"22430", "ctx":"","msg":"Checkpoint message", "attr":{"message":{"ts_sec":"1726969399","ts_usec":"622303","thre
```

6- cuando termine de cargar todo en nuestra terminal de powershell nos dirigimos hacia nuestra dirección ip seguida del puerto 3000 <http://localhost:3000>



Hello World!

7- Dentro de nuestra pc local en la aplicación de Docker Vemos que los contenedores están en estado de Running.



The screenshot shows the Docker Desktop application window. On the left sidebar, the 'Containers' tab is selected. The main area displays a list of containers for the 'desafio10' project. Two containers are listed: 'desafio10-mongo...' and 'desafio10-app-1'. Both are in the 'Running' state. The 'desafio10-app-1' container is highlighted, showing its details on the right. The details pane includes a 'View Configurations' button, a play button, and a stop button. Below these, the container's logs are visible, showing a large amount of text, including system boot logs and application startup messages. The bottom status bar indicates 'Engine running', 'RAM 3.94 GB', 'CPU 0.33%', and 'Disk ~ GB avail. of ~ GB'.

Container Name	State	Image	Created	Updated	Restarted	Paused	Stopped
desafio10-mongo...	Running	mongo:6.0	22:17:27017				
desafio10-app-1	Running	desafio10-app:3609.3600					

```
2024-09-21 22:39:19 mongod-1 | {"t":{"date":"2024-09-22T01:39:19.669+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":{"listener","msg":"Connection accepted","attr":{"remote":"172.18.0.3:45582","aud":"643d226c-7278-4635-888b-92a02472f772","connectionId":1,"connectionCount":1}}
2024-09-21 22:39:19 mongod-1 | {"t":{"date":"2024-09-22T01:39:19.669+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":{"conn","msg":"client metadata","attr":{"remote":"172.18.0.3:45582","client":"conn1","negotiatedCompressors":[],"doc":{"driver":{"name":"nodejs/mongoose","version":"6.8.0@6.3"},"platform":"Node.js v18.20.4, LE","os":{"na me":"linux","architecture":"x64","version":"5.15.153.1,microsoft-standard-mls2","type":"linux"},"env":{"container":{"runtime":"docker"}}}}}}}
2024-09-21 22:39:19 mongod-1 | {"t":{"date":"2024-09-22T01:39:19.669+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":{"listener","msg":"Connection accepted","attr":{"remote":"172.18.0.3:45588","aud":"f49026d-2b4b-434b-b138-e07553cd17","connectionId":2,"connectionCount":2}}
2024-09-21 22:39:19 mongod-1 | {"t":{"date":"2024-09-22T01:39:19.669+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":{"conn2","msg":"client metadata","attr":{"remote":"172.18.0.3:45588","client":"conn2","negotiatedCompressors":[],"doc":{"driver":{"name":"nodejs/mongoose","version":"6.8.0@6.3"},"platform":"Node.js v18.20.4, LE","os":{"na me":"linux","architecture":"x64","version":"5.15.153.1,microsoft-standard-mls2","type":"linux"},"env":{"container":{"runtime":"docker"}}}}}}}
2024-09-21 22:39:19 mongod-1 | {"t":{"date":"2024-09-22T01:39:19.678+00:00"},"s":"I", "c":"ACCESS", "id":20250, "ctx":{"conn2","msg":"Authentication succeeded","attr":{"mechanism":"SCRAM-SHA-256","speculative":true,"principalName":"myuser","authenticationDatabase":"nydatabase","remote":"172.18.0.3:45588","extraInfo":{}}}}
2024-09-21 22:39:19 mongod-1 | {"t":{"date":"2024-09-22T01:39:19.678+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":{"listener","msg":"Connection accepted","attr":{"remote":"172.18.0.3:49104","aud":"5cd7a9c-ac9-46c8-90ff-bbedc48c56c","connectionId":3,"connectionCount":3}}
2024-09-21 22:39:19 mongod-1 | {"t":{"date":"2024-09-22T01:39:19.678+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":{"conn3","msg":"client metadata","attr":{"remote":"172.18.0.3:49104","client":"conn3","negotiatedCompressors":[],"doc":{"driver":{"name":"nodejs/mongoose","version":"6.8.0@6.3"},"platform":"Node.js v18.20.4, LE","os":{"na me":"linux","architecture":"x64","version":"5.15.153.1,microsoft-standard-mls2","type":"linux"},"env":{"container":{"runtime":"docker"}}}}}}}
2024-09-21 22:40:19 mongod-1 | {"t":{"date":"2024-09-22T01:40:19.569+00:00"},"s":"I", "c":"WTOHPT", "id":22430, "ctx":{"checkpoint","msg":"WireTiger message","attr":{"message":{"ts_sec":1726969219,"ts_usec":569650,"thread":"1:0x7fc86d4a7640","session_name":"WT_SESSION_checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 3, snapshot max: 3 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp : (0, 0) base write gen: 7}}}}}
2024-09-21 22:41:19 mongod-1 | {"t":{"date":"2024-09-22T01:41:19.589+00:00"},"s":"I", "c":"WTOHPT", "id":22430, "ctx":{"checkpoint","msg":"WireTiger message","attr":{"message":{"ts_sec":1726969219,"ts_usec":589617,"thread":"1:0x7fc86d4a7640","session_name":"WT_SESSION_checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 6, snapshot max: 6 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp : (0, 0) base write gen: 7}}}}}
2024-09-21 22:42:19 mongod-1 | {"t":{"date":"2024-09-22T01:42:19.684+00:00"},"s":"I", "c":"WTOHPT", "id":22430, "ctx":{"checkpoint","msg":"WireTiger message","attr":{"message":{"ts_sec":1726969339,"ts_usec":682363,"thread":"1:0x7fc86d4a7640","session_name":"WT_SESSION_checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 8, snapshot max: 8 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp : (0, 0) base write gen: 7}}}}}
2024-09-21 22:43:19 mongod-1 | {"t":{"date":"2024-09-22T01:43:19.652+00:00"},"s":"I", "c":"WTOHPT", "id":22430, "ctx":{"checkpoint","msg":"WireTiger message","attr":{"message":{"ts_sec":1726969339,"ts_usec":652763,"thread":"1:0x7fc86d4a7640","session_name":"WT_SESSION_checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 10, snapshot max: 10 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp : (0, 0) base write gen: 7}}}}}
2024-09-21 22:44:19 mongod-1 | {"t":{"date":"2024-09-22T01:44:19.637+00:00"},"s":"I", "c":"WTOHPT", "id":22430, "ctx":{"checkpoint","msg":"WireTiger message","attr":{"message":{"ts_sec":1726969459,"ts_usec":637814,"thread":"1:0x7fc86d4a7640","session_name":"WT_SESSION_checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 12, snapshot max: 12 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp : (0, 0) base write gen: 7}}}}}
2024-09-21 22:45:19 mongod-1 | {"t":{"date":"2024-09-22T01:45:19.652+00:00"},"s":"I", "c":"WTOHPT", "id":22430, "ctx":{"checkpoint","msg":"WireTiger message","attr":{"message":{"ts_sec":1726969519,"ts_usec":652763,"thread":"1:0x7fc86d4a7640","session_name":"WT_SESSION_checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 14, snapshot max: 14 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp : (0, 0) base write gen: 7}}}}}
```