Run a container from an existing image and manage it using Docker commands

Table of Contents

Description

In this section, we'll explain how to run a **Java-based TodoApp** container from a pre-existing Docker image and manage it using common Docker commands. This includes how to start, stop, remove, and inspect running containers, as well as monitor logs.

Problem Statement

Once you've built a Docker image for the Java-based TodoApp (or any other image), you need to manage the container lifecycle: starting, stopping, inspecting, and removing containers efficiently. Docker provides several commands to manage these tasks.

Prerequisites

Completion of all previous lab guides (up to Lab Guide-02) is required before proceeding with Lab Guide-03.

Software Requirement

- **Docker Desktop**: Installed and running on a Windows system.
- **TodoApp Docker Image**: Make sure `Docker image` is present for todoapp.

Hardware Requirement

- **CPU**: 64-bit processor with virtualization support.
- **RAM**: 4 GB minimum (8 GB recommended).
- **Disk Space**: 500 MB or more for Docker images and containers.

Implementation Steps

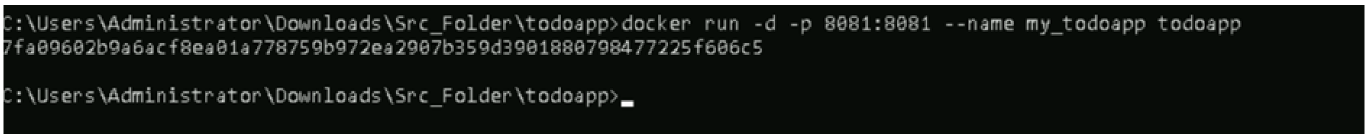
Step-1: Run a Container from an Existing Image

You can run a Docker container from the existing **Java-based TodoApp** image (**todoapp**) using the **docker run** command.

1. **Run the Container**:

   ```
   docker run -d -p 8081:8081 --name my_todoapp todoapp
   ```

   - **-d**: Runs the container in **detached mode** (in the background).
   - **-p 8081:8081**: Maps port 8081 of the host to port 8081 of the container.
   - **--name my_todoapp**: Names the container **my_todoapp** for easier management.
   - **todoapp**: The name and tag of the image you built.



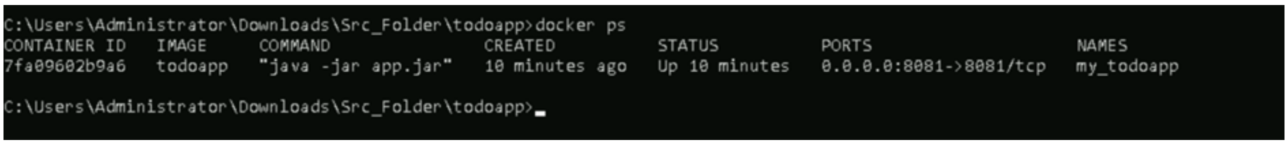
```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker run -d -p 8081:8081 --name my_todoapp todoapp
7fa09602b9a6acf8ea01a778759b972ea2907b359d39018807984772 25f606c5

C:\Users\Administrator\Downloads\Src_Folder\todoapp>
```

2. **Verify the Container is Running**:

   You can list all running containers with the following command:

   ```
   docker ps
   ```

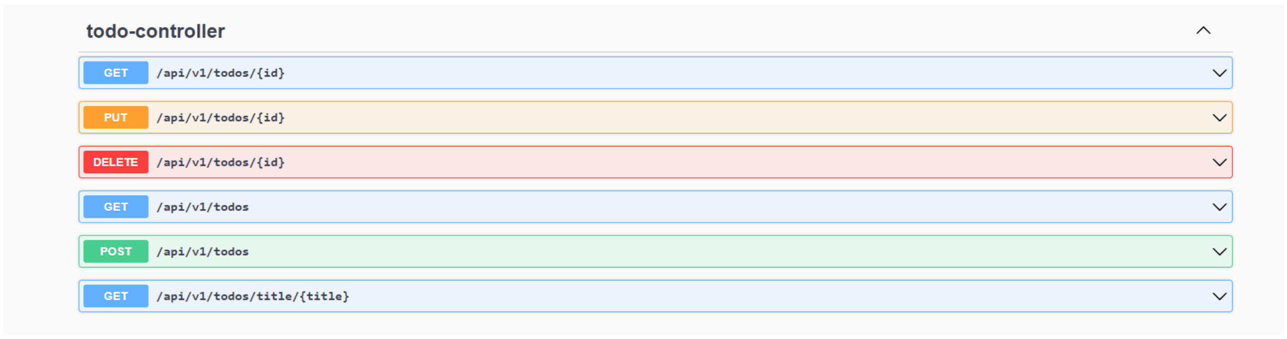
   You should see output similar to:



```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker ps
CONTAINER ID   IMAGE     COMMAND                 CREATED          STATUS          PORTS                    NAMES
7fa09602b9a6   todoapp   "java -jar app.jar"     10 minutes ago   Up 10 minutes   0.0.0.0:8081->8081/tcp   my_todoapp

C:\Users\Administrator\Downloads\Src_Folder\todoapp>
```

3. **Access the Application**:

   Open a browser and navigate to **http://localhost:8081/swagger-ui/index.html** to see your TodoApp running inside the container.



todo-controller

GET /api/v1/todos/{id}
PUT /api/v1/todos/{id}
DELETE /api/v1/todos/{id}
GET /api/v1/todos
POST /api/v1/todos
GET /api/v1/todos/title/{title}

Step-2: Managing Running Containers

Once your container is up and running, Docker provides several commands to manage the containers.

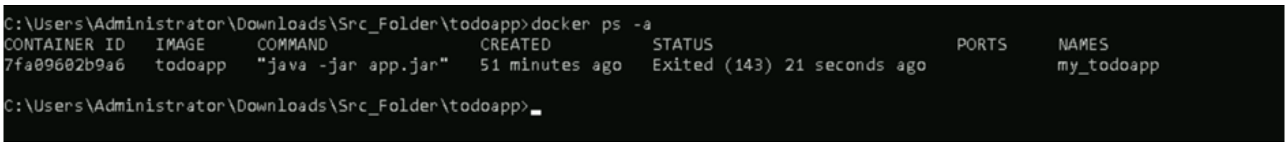
1. **List All Running Containers**:

   ```
   docker ps
   ```

2. **List All Containers (Including Stopped Containers)**:

   ```
   docker ps -a
   ```

   This will list all containers, including those that have stopped.



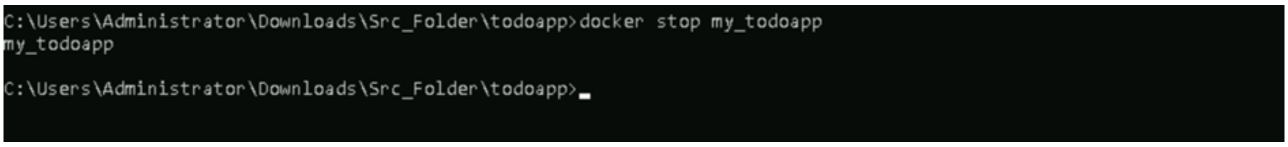
```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker ps -a
CONTAINER ID   IMAGE     COMMAND                CREATED          STATUS                       PORTS     NAMES
7fa09602b9a6   todoapp   "java -jar app.jar"    51 minutes ago   Exited (143) 21 seconds ago            my_todoapp

C:\Users\Administrator\Downloads\Src_Folder\todoapp>
```

Step-3: Stopping, Restarting, and Removing Containers

1. **Stop a Running Container**:

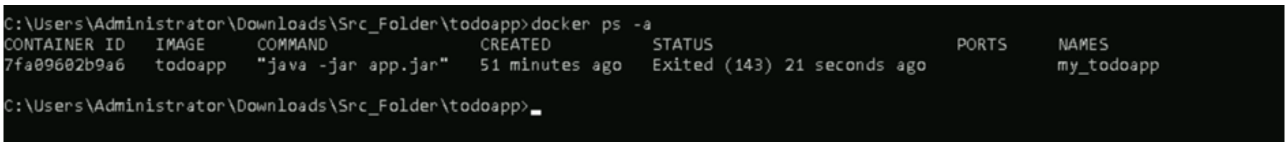
   Use the **docker stop** command to stop a running container:

   ```
   docker stop my_todoapp
   ```



```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker stop my_todoapp
my_todoapp

C:\Users\Administrator\Downloads\Src_Folder\todoapp>
```

   This will gracefully stop the **my_todoapp** container.

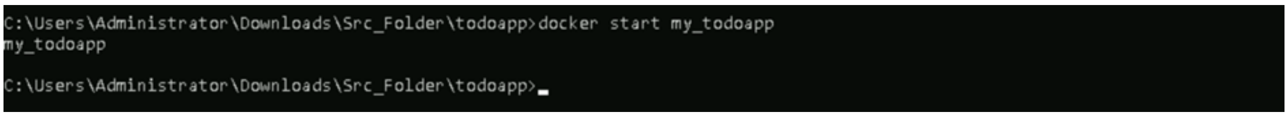


```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker ps -a
CONTAINER ID   IMAGE     COMMAND                CREATED          STATUS                       PORTS     NAMES
7fa09602b9a6   todoapp   "java -jar app.jar"    51 minutes ago   Exited (143) 21 seconds ago            my_todoapp

C:\Users\Administrator\Downloads\Src_Folder\todoapp>
```

2. **Restart a Stopped Container**:

   Use the **docker start** command to restart the container:

   ```
   docker start my_todoapp
   ```



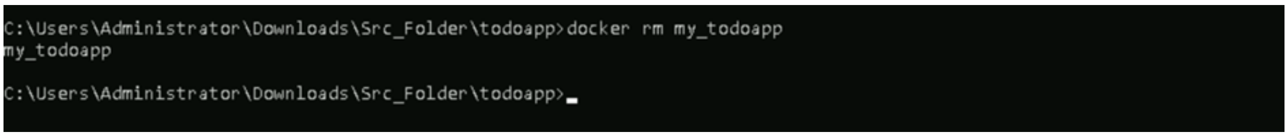
```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker start my_todoapp
my_todoapp

C:\Users\Administrator\Downloads\Src_Folder\todoapp>
```

This starts the container with the previous configuration (i.e., running the same image).

3. **Remove a Stopped Container**:

   To remove a stopped container, use the **docker rm** command:

   ```
   docker rm my_todoapp
   ```



```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker rm my_todoapp
my_todoapp

C:\Users\Administrator\Downloads\Src_Folder\todoapp>
```

   Note: A running container cannot be removed. You need to stop it first using **docker stop**.

4. **Force Remove a Running Container**:

   If you need to remove a running container, use the **-f** flag:

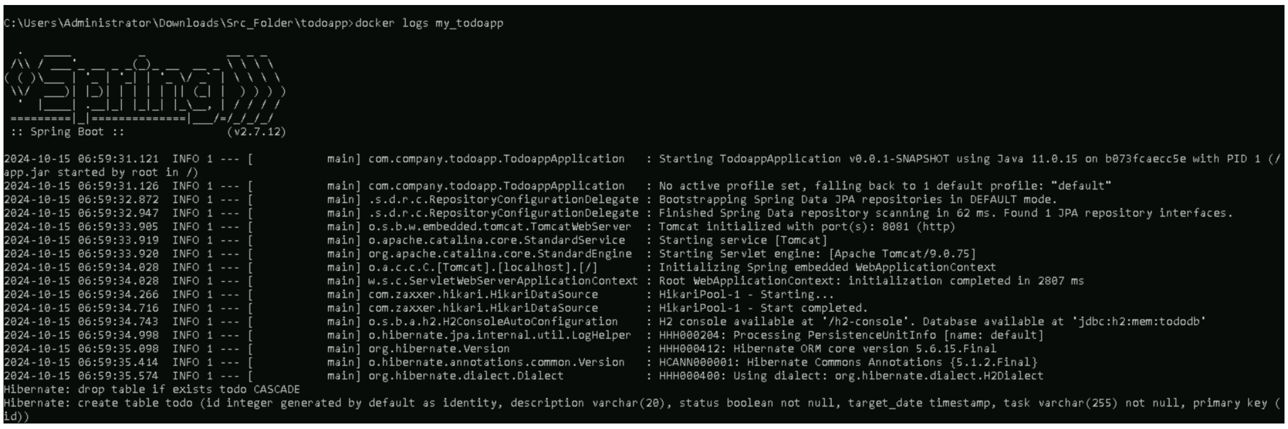
   ```
   docker rm -f my_todoapp
   ```

Step-4: Accessing Logs and Container Status

1. **View Container Logs**:

   To see the logs from a running container, use the **docker logs** command:

   ```
   docker logs my_todoapp
   ```



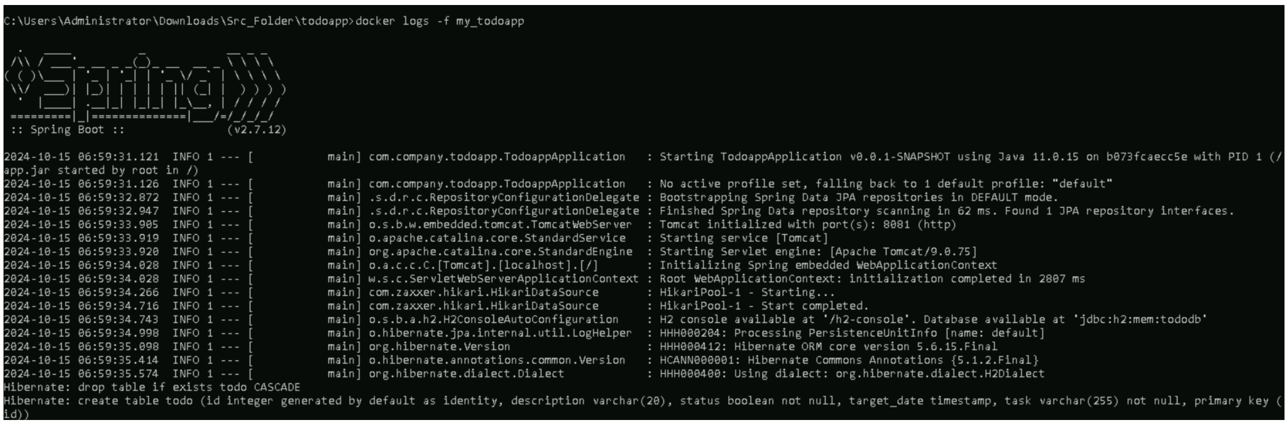
```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker logs my_todoapp

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::               (v2.7.12)

2024-10-15 06:59:31.121  INFO 1 --- [           main] com.company.todoapp.TodoappApplication   : Starting TodoappApplication v0.0.1-SNAPSHOT using Java 11.0.15 on b073fcaecc5e with PID 1 (/
app.jar started by root in /)
2024-10-15 06:59:31.126  INFO 1 --- [           main] com.company.todoapp.TodoappApplication   : No active profile set, falling back to 1 default profile: "default"
2024-10-15 06:59:32.872  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2024-10-15 06:59:32.947  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 62 ms. Found 1 JPA repository interfaces.
2024-10-15 06:59:33.905  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8081 (http)
2024-10-15 06:59:33.919  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-10-15 06:59:33.920  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.75]
2024-10-15 06:59:34.028  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-10-15 06:59:34.028  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2807 ms
2024-10-15 06:59:34.266  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2024-10-15 06:59:34.716  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2024-10-15 06:59:34.743  INFO 1 --- [           main] o.s.b.a.h2.H2ConsoleAutoConfiguration    : H2 console available at '/h2-console'. Database available at 'jdbc:h2:mem:tododb'
2024-10-15 06:59:34.998  INFO 1 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2024-10-15 06:59:35.098  INFO 1 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.6.15.Final
2024-10-15 06:59:35.414  INFO 1 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2024-10-15 06:59:35.574  INFO 1 --- [           main] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
Hibernate: drop table if exists todo CASCADE
Hibernate: create table todo (id integer generated by default as identity, description varchar(20), status boolean not null, target_date timestamp, task varchar(255) not null, primary key (
id))
```

   This shows the standard output logs of the running application.

2. **Follow Logs in Real-Time**:

If you want to stream the logs in real-time, add the **-f** flag:

```
docker logs -f my_todoapp
```



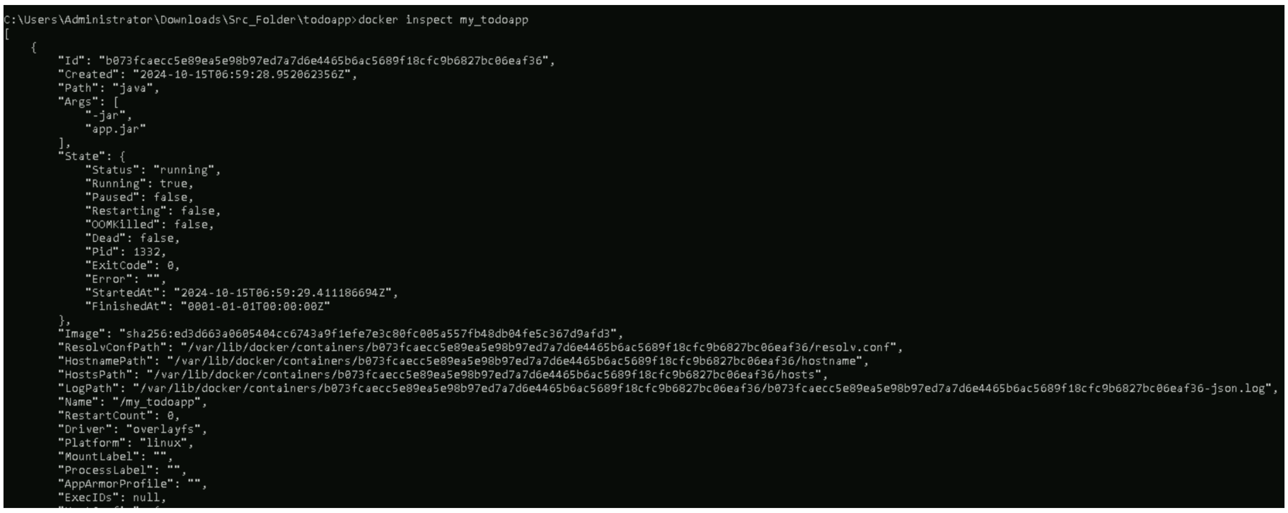
```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker logs -f my_todoapp

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::               (v2.7.12)

2024-10-15 06:59:31.121  INFO 1 --- [           main] com.company.todoapp.TodoappApplication   : Starting TodoappApplication v0.0.1-SNAPSHOT using Java 11.0.15 on b073fcaecc5e with PID 1 (/
app.jar started by root in /)
2024-10-15 06:59:31.126  INFO 1 --- [           main] com.company.todoapp.TodoappApplication   : No active profile set, falling back to 1 default profile: "default"
2024-10-15 06:59:32.872  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2024-10-15 06:59:32.947  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 62 ms. Found 1 JPA repository interfaces.
2024-10-15 06:59:33.905  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8081 (http)
2024-10-15 06:59:33.919  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-10-15 06:59:33.920  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.75]
2024-10-15 06:59:34.028  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-10-15 06:59:34.028  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2807 ms
2024-10-15 06:59:34.266  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2024-10-15 06:59:34.716  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2024-10-15 06:59:34.743  INFO 1 --- [           main] o.s.b.a.h2.H2ConsoleAutoConfiguration    : H2 console available at '/h2-console'. Database available at 'jdbc:h2:mem:tododb'
2024-10-15 06:59:34.998  INFO 1 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2024-10-15 06:59:35.098  INFO 1 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.6.15.Final
2024-10-15 06:59:35.414  INFO 1 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2024-10-15 06:59:35.574  INFO 1 --- [           main] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
Hibernate: drop table if exists todo CASCADE
Hibernate: create table todo (id integer generated by default as identity, description varchar(20), status boolean not null, target_date timestamp, task varchar(255) not null, primary key (
id))
```

This will continue displaying logs as they are written.

3. **Inspect a Container's Details**:

   To get detailed information about the container, use the **docker inspect** command:

```
docker inspect my_todoapp
```



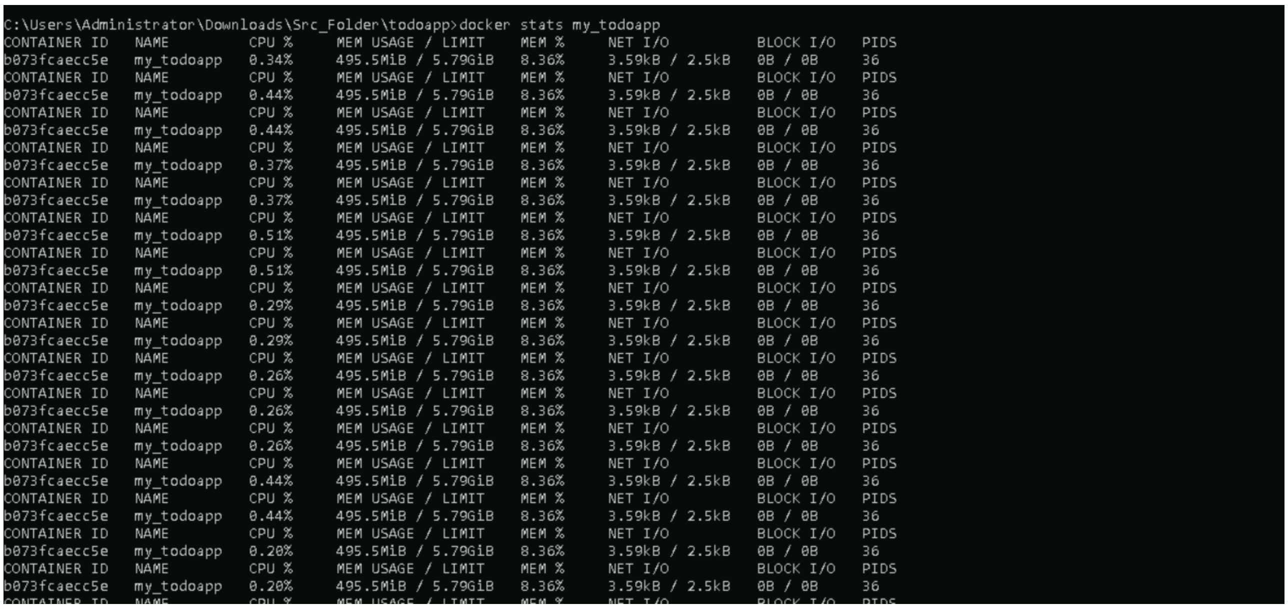
```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker inspect my_todoapp
[
    {
        "Id": "b073fcaecc5e89ea5e98b97ed7a7d6e4465b6ac5689f18cfc9b6827bc06eaf36",
        "Created": "2024-10-15T06:59:28.952062356Z",
        "Path": "java",
        "Args": [
            "-jar",
            "app.jar"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 1332,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2024-10-15T06:59:29.411186694Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:ed3d663a0605404cc6743a9f1efe7e3c80fc005a557fb48db04fe5c367d9afd3",
        "ResolvConfPath": "/var/lib/docker/containers/b073fcaecc5e89ea5e98b97ed7a7d6e4465b6ac5689f18cfc9b6827bc06eaf36/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/b073fcaecc5e89ea5e98b97ed7a7d6e4465b6ac5689f18cfc9b6827bc06eaf36/hostname",
        "HostsPath": "/var/lib/docker/containers/b073fcaecc5e89ea5e98b97ed7a7d6e4465b6ac5689f18cfc9b6827bc06eaf36/hosts",
        "LogPath": "/var/lib/docker/containers/b073fcaecc5e89ea5e98b97ed7a7d6e4465b6ac5689f18cfc9b6827bc06eaf36/b073fcaecc5e89ea5e98b97ed7a7d6e4465b6ac5689f18cfc9b6827bc06eaf36-json.log",
        "Name": "/my_todoapp",
        "RestartCount": 0,
        "Driver": "overlayfs",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "",
        "ExecIDs": null,
```

   This will return JSON-formatted data that contains all the configuration and status details of the container.

4. **View the Running Container's Status**:

   Use **docker stats** to monitor resource usage like CPU, memory, network, etc.

```
docker stats my_todoapp
```



```
C:\Users\Administrator\Downloads\Src_Folder\todoapp>docker stats my_todoapp
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.34%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.44%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.44%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.37%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.37%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.51%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.51%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.29%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.29%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.26%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.26%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.26%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.44%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.44%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.20%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
b073fcaecc5e   my_todoapp   0.20%     495.5MiB / 5.79GiB    8.36%     3.59kB / 2.5kB    0B / 0B     36
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O   PIDS
```

This command continuously updates with live statistics of the container's resource consumption.

References

For more detailed usage and documentation of Docker commands:

- Docker Official Documentation: https://docs.docker.com/
- Docker Command Reference: https://docs.docker.com/engine/reference/commandline/docker/