



+ New



Share

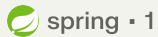


wrochow

Prove me with the steps required to create, install, and run a bare-bones REST Web application using Spring Boot and deploying to Docker on a Windows environment.

Sources

Getting Started | Spring Boot Docker



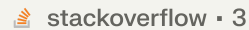
spring • 1

Tutorial: Run Spring Boot web service in Windows Docker container - LinkedIn



linkedin • 2

Spring Boot Application Host on Docker Desktop on Windows - Stack Overflow



stackoverflow • 3



View 2 more

Answer

To create, install, and run a bare-bones REST Web application using Spring Boot and deploy it to Docker on a Windows environment, you can follow the steps below:

1. Create a Spring Boot Web Application:

- Generate a Spring Boot web project by visiting <http://start.spring.io>, selecting the Web dependency, and clicking the "Generate Project" button. Extract the downloaded source code to your IDE's workspace and import the project as an existing Maven project.
- Add a new Java class (e.g., Controller.java) to create a REST endpoint. For example, you can map the root path and return a simple message like "Hello Docker World" when triggered.

2. Build the Spring Boot Application:

- Run Maven build to compile and package the Spring Boot application. You can test the application by running it in your IDE using the command `mvn spring-boot:run` and accessing it in a web browser at <http://localhost:8080>.

Home

Discover

Library

3. Create a Dockerfile:

- Create a Dockerfile in the root folder of your project. The Dockerfile should include instructions for building the Docker image, such as defining the base image, adding the application JAR file, exposing the necessary ports, and specifying the entry point for running the application.

4. Build the Docker Image:

- Open a PowerShell window, navigate to the project root folder, and build the Docker image using the ``docker build`` command. This command will create a Docker image based on the instructions provided in the Dockerfile.

5. Run the Docker Container:

- Once the Docker image is built, you can run a Docker container based on the image using the ``docker run`` command. Make sure to map the exposed port of the application within the container to a port on the host machine using the ``-p`` flag.

6. Access the Application:

- After running the Docker container, you should be able to access the Spring Boot application deployed in the Docker container by visiting the specified port on your Windows environment.

It's important to note that the specific commands and configurations in the Dockerfile may vary based on the requirements of your Spring Boot application and the Docker environment. Additionally, you may need to adjust the Docker networking settings to expose the application to remote servers within the same network [1](#) [2](#) [3](#) [4](#) [5](#).

Rewrite      

Please provide me with a basic Dockerfile that will launch a project called my-test-project on port 8080.

Sources

Using Dockerfile to Expose Ports - Linux Hint

 linuxhint • 1

docker run - Docker Docs

 docker • 2

 Home

 Discover

 Library

Answer

To create a basic Dockerfile that will launch a project called my-test-project on port 8080, you can follow the example below:

```
Dockerfile
# Use a base image
FROM openjdk:11

# Set the working directory in the container
WORKDIR /app

# Copy the JAR file into the container
COPY target/my-test-project.jar /app/my-test-project.jar

# Expose port 8080
EXPOSE 8080

# Command to run the application
CMD ["java", "-jar", "my-test-project.jar"]
```

In this Dockerfile:

- We use the `openjdk:11` as the base image, which provides the Java runtime environment.
- Set the working directory to `/app` in the container.
- Copy the JAR file of the project into the container.
- Expose port 8080 using the `EXPOSE` instruction.
- Specify the command to run the application using `CMD`.

After creating this Dockerfile, you can build the Docker image using the `docker build` command and then run a container based on this image using the `docker run` command. The application should be accessible on port 8080 of the host machine ¹ ² ³.

Rewrite ▾ 🚩 ⋮ 📄 ✎ ↻


Can I specify a container name with a command

Sources

Docker run reference

 docker • 1

How to name a container with docker run - Stack Overflow

 stackoverflow • 2

How to Name or Rename Docker Containers - Tecmint

 tecmint • 3



View 2 more

Answer

Yes, you can specify a container name with a command line argument when using the `docker run` command. To assign a name to a container, you can use the `--name` option followed by the desired name. Here's an example of how to do this:

```
bash
docker run -d --name my_container_name -p 8080:8080 my_image_name
```



In this example:

- `-d` runs the container in detached mode.
- `--name my_container_name` assigns the name "my_container_name" to the container.
- `-p 8080:8080` maps port 8080 on the host to port 8080 in the container.
- `my_image_name` specifies the name of the Docker image to be used for creating the container.

By providing the `--name` option followed by the desired name, you can specify a container name with a command line argument when using `docker run` ¹ ² ³.

Rewrite      

if I use `-p 8080:8088`, which is the port that the browser should connect to?

Sources

docker - How do I "point" container to port 8080? - Stack Overflow

Howto expose port in docker container - Server Fault

Using Dockerfile to Expose Ports - Linux Hint



{👤} linuxhint • 3

View 2 more

≡ Answer

If you use `-p 8080:8088`, the port that the browser should connect to is port 8080. When using the `-p` option in the `docker run` command, the format is `[host port]:[container port]`. In this case, the host port is 8080, so you should connect to port 8080 in your browser to access the service running inside the Docker container ⁴.

Therefore, when you use `-p 8080:8088`, the browser should connect to port 8080 on the host machine to access the service running on port 8088 inside the Docker container ⁴.

Rewrite ▾ ...

📁 Related

[how to check if a docker container is running on a specific port](#)[how to map a docker container port to a host machine port](#)[how to expose a docker container port to the internet](#)