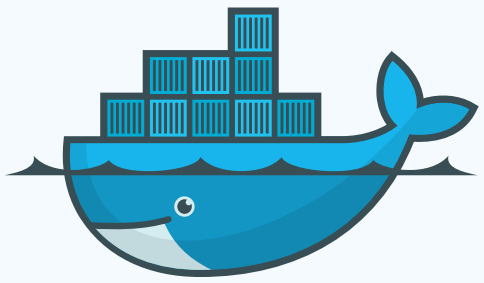
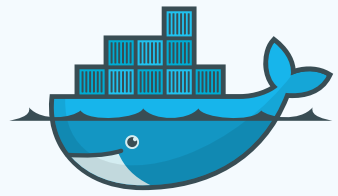


How to Deploy a Spring Boot Application in Docker

If you're new to Docker and want to know how to containerize and deploy your Spring Boot app, this guide will walk you through the process. Let's start by understanding some basic Docker concepts before jumping into the deployment.

Read More

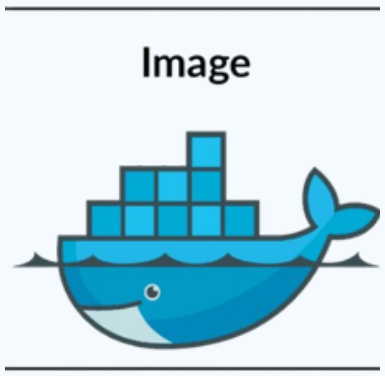
@hariharanr18



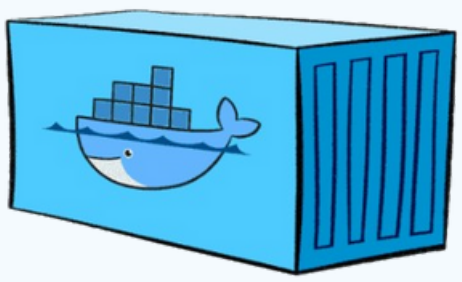
🚢 What is Docker?

Docker is a platform that allows you to package applications into containers. Containers are lightweight, stand-alone, and executable software packages that include everything needed to run the application, including code, runtime, libraries, and system settings.

🔍 Basic Docker Terms

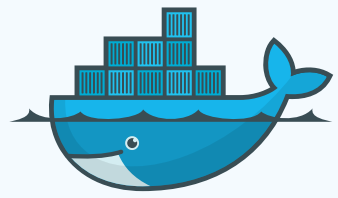


Docker Image: *A read-only template used to create Docker containers. It includes the app, its dependencies, and runtime.*



Docker Container: *A running instance of a Docker image. It's like a lightweight virtual machine.*

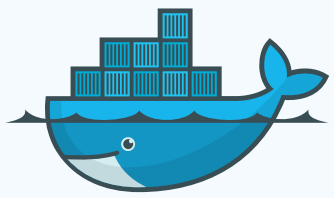
@hariharanr18



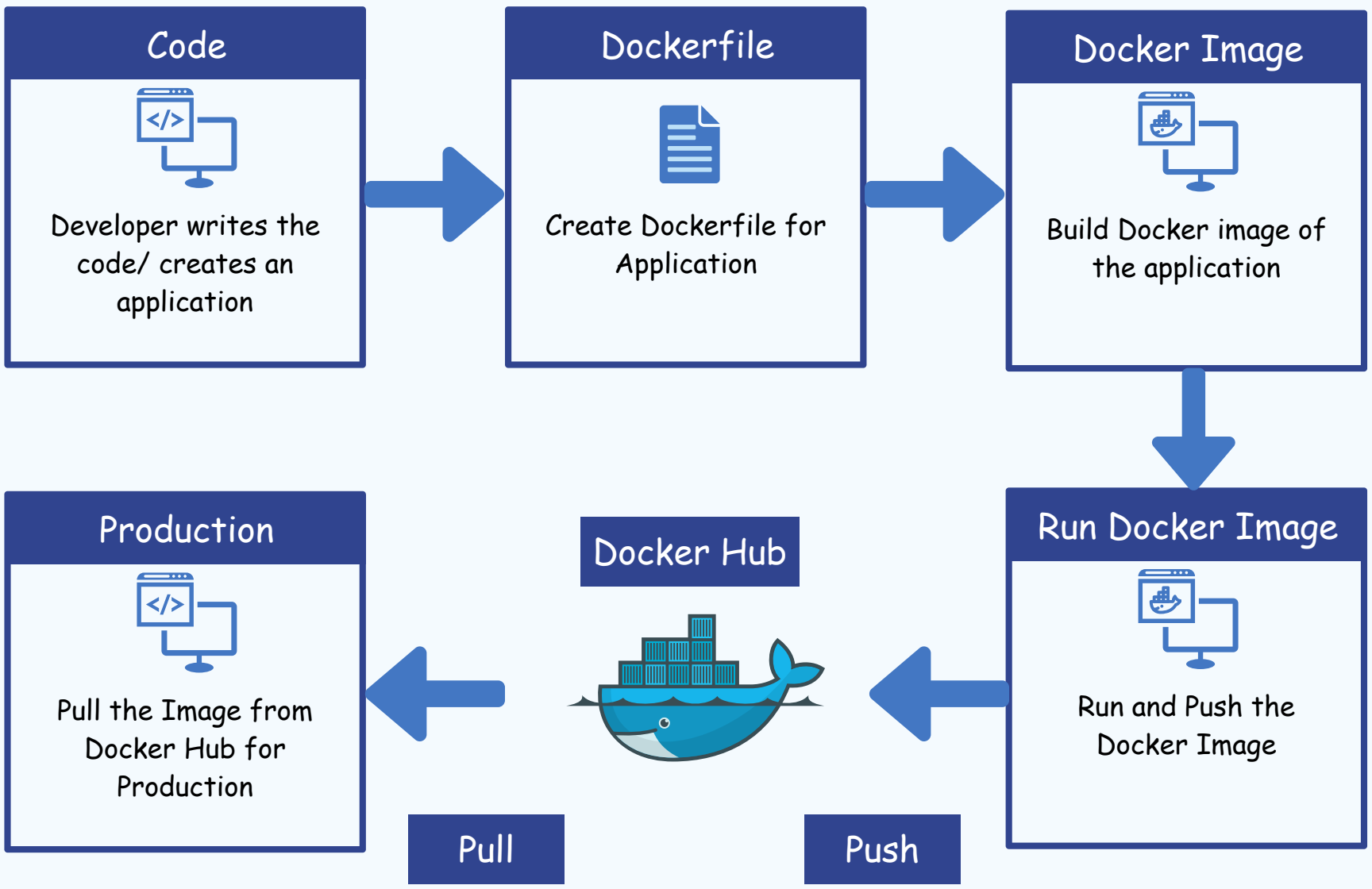
🛠️ Common Docker Commands

- docker images: Lists all Docker images on your system.
- docker run: Runs a container based on a Docker image.
- docker pull: Downloads a Docker image from Docker Hub.
- docker push: Uploads a Docker image to Docker Hub.
- docker tag: Tags an image with a new name or version.
- docker build: Builds a Docker image from a Dockerfile.
- docker ps: Shows running Docker containers.

@hariharanr18



Docker Workflow



@hariharanr18

🚀 Create a Spring Boot App and Package It as a JAR in Eclipse

1️⃣ Create a Spring Boot Project

- Go to File > New > Spring Starter Project.
- Fill in project details (Maven, JAR packaging, Java 21).
- Add dependencies like Spring Web.

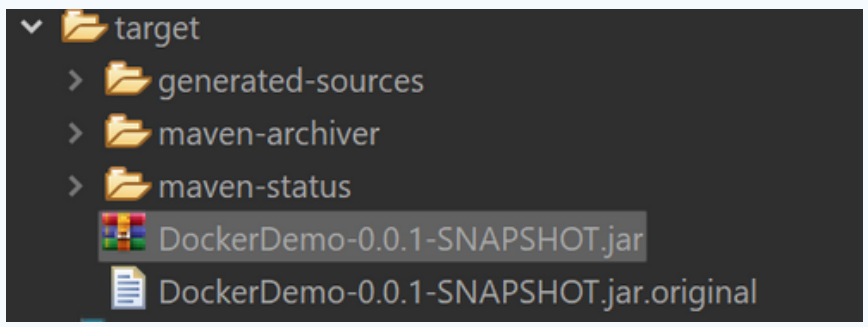
2️⃣ Add a REST Controller

Create HelloController.java:

```java
@RestController
@RequestMapping("/api")
public class HomeController {
	@GetMapping("/home")
	public String homeGreet() {
		return "Hello this application is deployed in docker";
	}
}
```

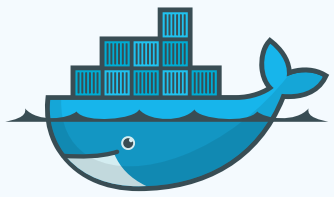
3️⃣ Build the JAR

- Right-click your project, select Run As > Maven Build.
- Enter clean package in Goals and hit Run.
- Your JAR will be generated in the target/ folder!



@hariharanr18

🔨 Creating the Dockerfile



In your Spring Boot project, create a Dockerfile that instructs Docker how to build and run your application:

Key Dockerfile Instructions:

- FROM: Defines the base image (in this case, amazoncorretto).
- LABEL: Adds metadata about the image.
- EXPOSE: Specifies the port on which the containerized app will run.
- WORKDIR: Sets the working directory inside the container.
- COPY: Copies files from your local system into the container.
- ENTRYPOINT: Defines the command that runs when the container starts.

Dockerfile

```
# 1. Define the base image
FROM amazoncorretto:21
# 2. Add metadata using LABEL (optional)
LABEL version="1.0"
# 3. Expose the port your app runs on
EXPOSE 8080:8080
# 4. Set the working directory inside the container
WORKDIR /app
# 5. Copy the jar file into the container
COPY target/DockerDemo-0.0.1-SNAPSHOT.jar /app/DockerDemo1.jar
# 6. Define the entry point for the app
ENTRYPOINT [ "java", "-jar", "DockerDemo1.jar" ]

```

@hariharanr18

🔨 Building Image and Running Docker Container

Build the Docker Image

- In the terminal, navigate to your project's root directory and run:

```
docker build -t docker-demo-image1:doctag .
```

View Docker Image Details

- After building your Docker image, you can list all the images and see their details (like image ID, size, and creation date) with the following command:

```
docker image ls
```

Run the Docker Container

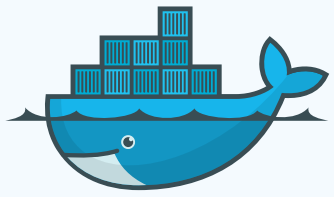
- Run the following command to start the container:

```
docker run -p 8080:8080 docker-demo-image1:doctag
```

- The -p option maps the container's internal port (8080) to your local machine's port (8080).
- Open your browser and go to http://localhost:8080 to see your Spring Boot app running inside the Docker container.

@hariharanr18

📤 Push to Docker Hub



Once your image is running successfully, you can push it to Docker Hub for easy sharing.

Tag the Image

- Before pushing, tag your image with your Docker Hub username:

```
docker tag springboot-docker-app yourusername/springboot-docker-app:v1
```

- For Example

```
docker tag fef9752937cd hariharanr1893/docker-spring-demo:doctag
```

Push the Image

- Push the image to Docker Hub

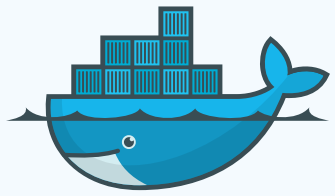
```
docker push yourusername/springboot-docker-app:v1
```

- For Example

```
docker push hariharanr1893/docker-spring-demo:doctag
```

@hariharanr18

Pull and Run from Docker Hub



To pull and run the image from Docker Hub on any machine:

Pull the Image

- Run the following command to download the image:

```
docker pull yourusername/springboot-docker-app:v1
```

- For Example

```
docker pull hariharanr1893/docker-spring-demo:doctag
```

Run the Image

- Start the container:

```
docker run -p 8080:8080 yourusername/springboot-docker-app:v1
```

- For Example

```
docker run -p 8080:8080 hariharanr1893/docker-spring-demo:doctag
```

@hariharanr18

Thank you !!!

Stay tuned for more tech-related content, tips, and deep dives into development practices.

@hariharanr18