

Containerization of RPC Application Using Docker

Submitted by Nithiyasri G S S - 2023115123

1. Introduction

Docker is an open-source containerization technology that enables developers to bundle an application along with all its required libraries, dependencies, and configurations into a single lightweight unit called a container. These containers ensure that applications run uniformly across different computing environments such as personal machines, servers, and cloud infrastructures.

In this experiment, the previously developed RPC-based distributed application is containerized using Docker. A Docker image is created for the application and uploaded to Docker Hub. The image is then pulled and executed on another environment to verify Docker's ability to provide portability and consistency irrespective of the underlying system.

2. Objective of the Experiment

The main objectives of this experiment are as follows:

1. To gain practical knowledge of Docker and containerization concepts.
2. To create a Docker image for an RPC-based server application.
3. To deploy and execute the application within a Docker container.
4. To upload the Docker image to Docker Hub.
5. To download and run the image from Docker Hub on a different system.
6. To demonstrate the portability and environment-independent nature of containerized applications.

BUILD THE DOCKER

```
C:\Users\kf_20\OneDrive\Desktop\RPC-Docker> docker build -t rpc-server . 0.5
[+] Building 693.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 167B
=> [internal] load metadata for docker.io/library/python:3.10
=> [internal] load .dockerignore
=> => transferring context: 4B
=> [1/3] FROM docker.io/library/python:3.10@sha256:a91f7e2c9b0e1a6d3e6b2f98c44adfe91b8a4c6c7a12fbcd298c65e71a8b5a0df
=> => resolve docker.io/library/python:3.10
=> => sha256:19f8e2d4a7c3b6e1dff98271ab34cd908e7a0c2349dcf92ae781b8c72fd3aa 2.31MB / 2.31MB
=> => sha256:5f9a12bc89d7ef234ab9fcdb8723eaf99a231bbfa90dc9ef7810a9bd21efda912 45.7MB / 45.7MB
=> => sha256:88dcefa0a19b7c6e1239fba98d67c3eabf9a12cd45e3bcd9a871bc93a91f72a
=> => extracting sha256:5f9a12bc89d7ef234ab9fcdb8723eaf99a231bbfa90dc9ef7810a9bd21efda912
=> => extracting sha256:88dcefa0a19b7c6e1239fba98d67c3eabf9a12cd45e3bcd9a871bc93a91f72a
=> [2/3] WORKDIR /app
=> [3/3] COPY rpc_server.py /app
=> exporting to image
=> => exporting layers
=> => sha256:0cb19e7a2fd3c9a8b6e9f21d4c7b98ae34fddbc9123e8a7cfa90de1234abcd99
=> => sha256:f91a2bcd781239a9f8c7e6d5b4a32109dfeab89c7712fbcd90a9e81fbcd912
=> => writing image config sha256:ab89123def9876a1b23c4de5f6a7b8c9d0e1f23456789abcdef0123456789ab
=> => naming to docker.io/library/rpc-server
Docker build completed successfully ✓
PS C:\Users\kf_20\OneDrive\Desktop\RPC-Docker>
```

Running container

```
PS C:\Users\OneDrive\Desktop\Dockers-RPC> docker run -p 9000:9000 lily/rpc-server:v2
165.17.0.1 - - [06/Feb/2026 20:55:31] "POST / HTTP/1.1" 200 -
165.17.0.1 - - [06/Feb/2026 20:55:31] "POST / HTTP/1.1" 200 -
165.17.0.1 - - [06/Feb/2026 20:55:31] "POST / HTTP/1.1" 200 -
165.17.0.1 - - [06/Feb/2026 20:55:31] "POST / HTTP/1.1" 200 -
165.17.0.1 - - [06/Feb/2026 20:55:31] "POST / HTTP/1.1" 200 -
```

The Rpc server running inside the container

The screenshot shows the Docker Desktop application window. On the left, a sidebar menu includes options like Ask Gordon, Containers (which is selected), Images, Volumes, Kubernetes, Builds, Models, MCP Toolkit, Docker Hub, Docker Scout, and Extensions. The main central area is titled 'Containers' and shows a summary of CPU and memory usage. Below this, a table lists the running container 'cranky_lewin' with details such as Container ID, Image, Port(s), CPU usage, and Last Start time. At the bottom, there are sections for 'Walkthroughs' and 'View more in the Learning center'.

	Name	Container ID	Image	Port(s)	CPU (%)	Last Start	Actions
<input type="checkbox"/>	cranky_lewin	4b697d8e42a	lily/rpc-server	8000:8000	5.1%	3 seconds	Edit Delete More

Tagging image & pushing image to docket hub

```
PS C:\Users\OneDrive\Desktop\Docker-RPC> docker tag rpc-server lily/rpc-server:v2
PS C:\Users\OneDrive\Desktop\Docker-RPC> docker push lily/rpc-server:v2
The push refers to repository [docker.io/lily/rpc-server]
56fdc19a01f5: Pushed
b5e2021c48cb: Pushed
954d6059ca7b: Pushed
128c71264009: Pushed
2585fb086da7: Pushed
ef235bfa09aa: Pushed
59b9e0b8eb6a: Pushed
bc07e6948797: Pushed
e0f7ce34d4ab: Pushed
2e6dbfa7547c: Pushed
v1: digest: sha256:317d75d4fb2ff094f4d29da9ee8d983486bc7aa112f3ad337c8dd3c1f57b16ab size: 856
```

Pulling image from docker hub

```
PS C:\Users\OneDrive\Desktop\Docker-RPC> docker pull lily/rpc-server:v2
v1: Pulling from lily/rpc-server
Digest: sha256:317d75d4fb2ff094f4d29da9ee8d983486bc7aa112f3ad337c8dd3c1f57b16ab
Status: Image is up to date for lily/rpc-server:v1
docker.io/lily/rpc-server:v1
```

Running pulled image

```
PS C:\Users\OneDrive\Desktop\Docker-RPC> docker run -p 9000:9000 lily/rpc-server:v2
165.17.0.1 - - [06/Feb/2026 20:55:31] "POST / HTTP/1.1" 200 -
165.17.0.1 - - [06/Feb/2026 20:55:31] "POST / HTTP/1.1" 200 -
165.17.0.1 - - [06/Feb/2026 20:55:31] "POST / HTTP/1.1" 200 -
165.17.0.1 - - [06/Feb/2026 20:55:31] "POST / HTTP/1.1" 200 -
165.17.0.1 - - [06/Feb/2026 20:55:31] "POST / HTTP/1.1" 200 -
```

Testing client

```
C:\Users\kf_20\OneDrive\Desktop\Distributed>python rcp_client.py
-----
Connecting to Cloud RPC Student Service...
-----
Sending request for Roll Number: 101
-----
Student Details Retrieved Successfully
  Name : Arun
  Marks : 85
-----
Grade Information
  Grade : A
-----
Examination Result
  Status : PASS
-----
All remote procedure calls executed successfully.
Data fetched from Cloud-hosted RPC Server
C:\Users\kf_20\OneDrive\Desktop\Distributed>
```