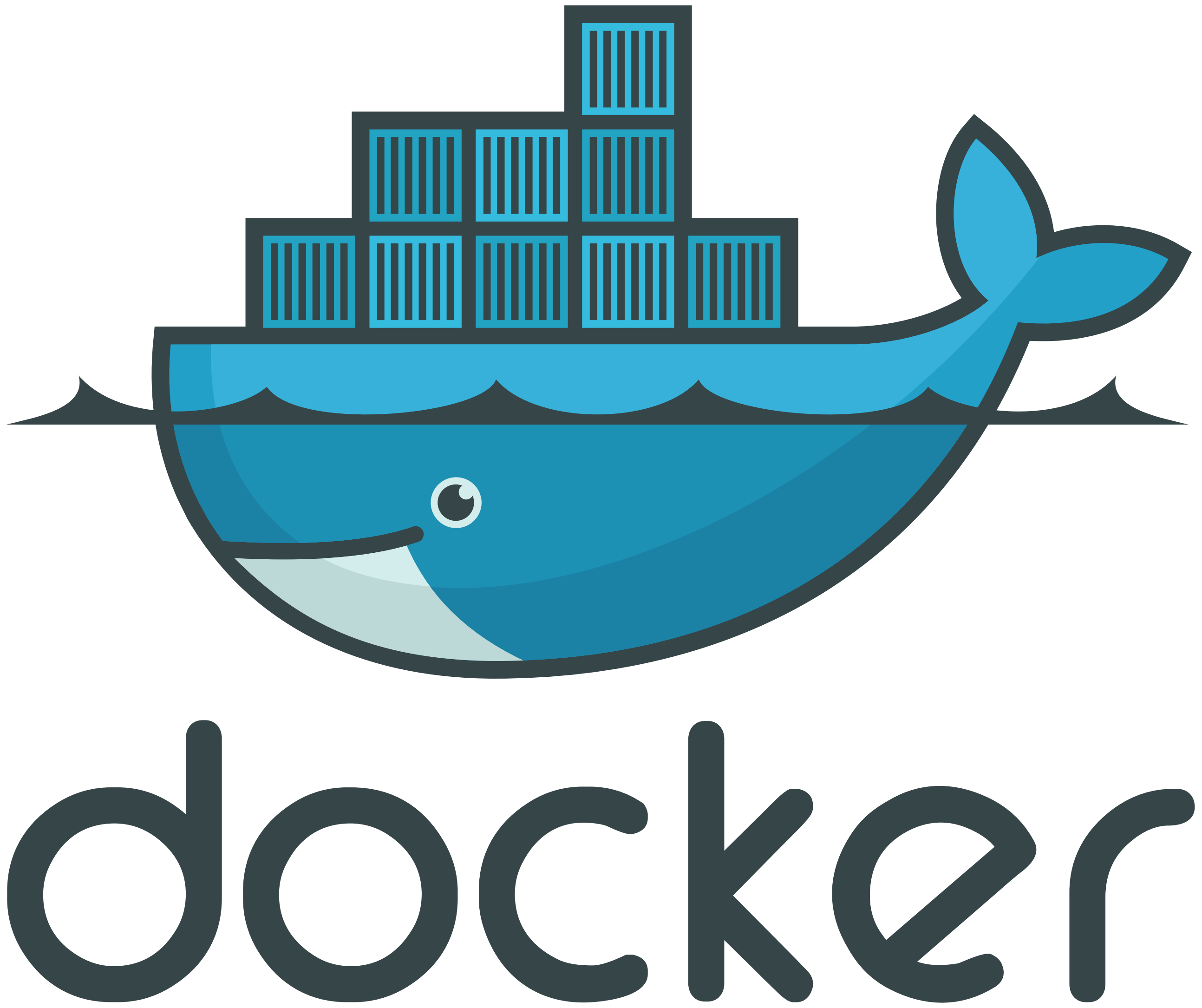

docker

- Docker Installation on Mac, Windows & Linux
- Creating Demo Project on Node and Python
- Creating DockerFile
- Creating Docker Image
- Running Containers
- Pre-defined Images
- DockerHub
- Docker Volumes and Network
- Docker Compose

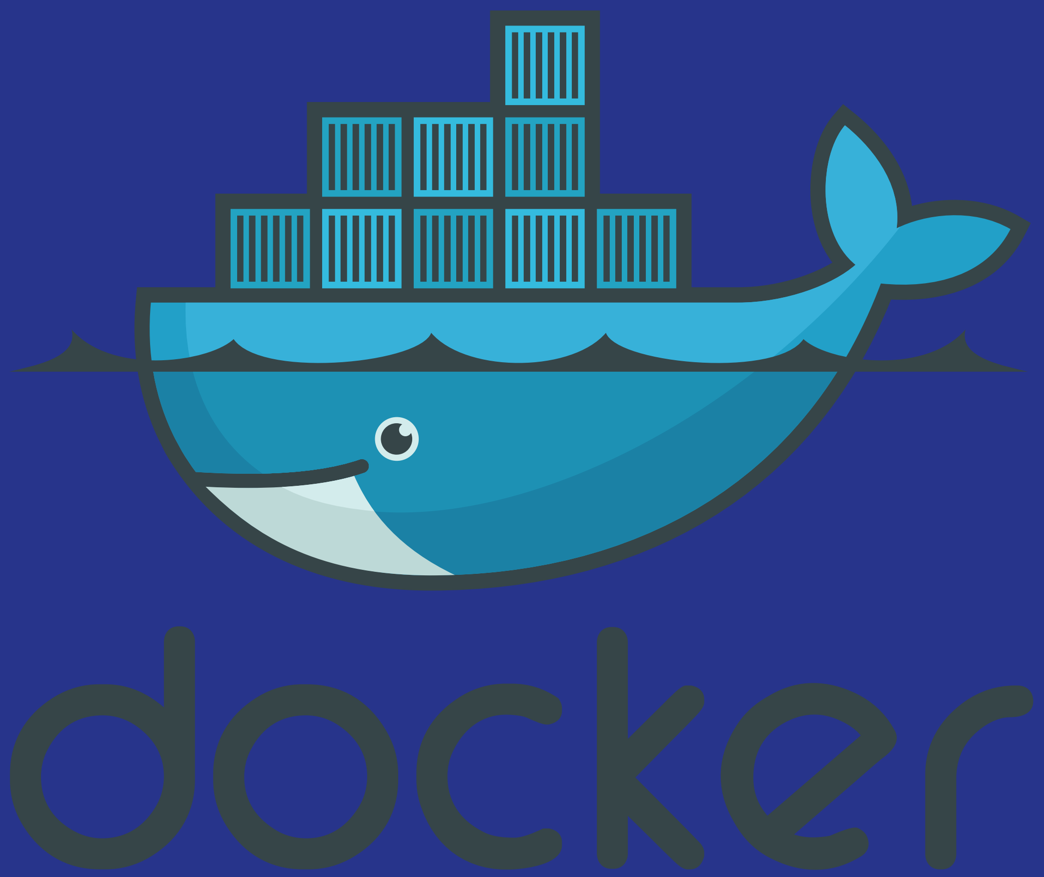
Overview

What is Docker and Why?

What are Containers?

How Container Works?

What is a Docker?



What is a Docker?

- Docker is a containerization platform for developing, packaging, shipping, and running applications.

- It provides the ability to run an application in an isolated environment called a container.
- Makes deployment and development efficient.

Why do we need Docker?

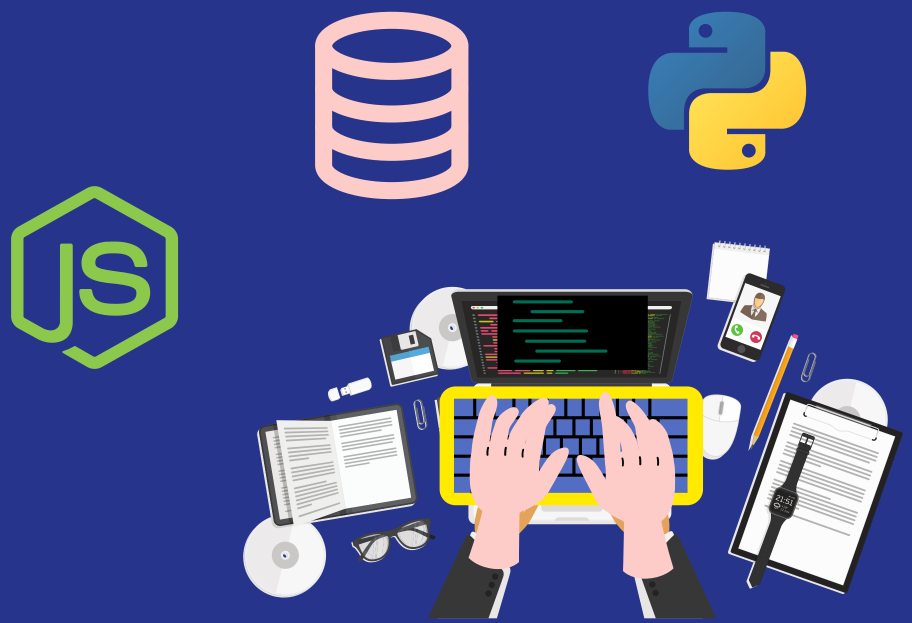
Developer

```
                                                    header0_initialPadding + 'px');
if ($(window).scrollTop() > header1_initialDistance) {
    if (parseInt(header1.css('padding-top'), 10) >= header1_initialPadding
        header1.css('padding-top', '' + $(window).scrollTop() - header1_initial
    }
} else {
    header1.css('padding-top', '' + header1_initialPadding + 'px');
}

if ($(window).scrollTop() > header2_initialDistance) {
    if (parseInt(header2.css('padding-top'), 10) >= header2_initialPadding
        header2.css('padding-top', '' + $(window).scrollTop() - header2_initial
    }
} else {
    header2.css('padding-top', '' + header2_initialPadding + 'px');
```

Working Successfully

```
                                              header0_initialPadding + 'px');
if ($(window).scrollTop() > header1_initialDistance) {
    if (parseInt(header1.css('padding-top'), 10) >= header1_initialPadding
        header1.css('padding-top', '' + $(window).scrollTop() - header1_initial
    }
} else {
    header1.css('padding-top', '' + header1_initialPadding + 'px');
}

if ($(window).scrollTop() > header2_initialDistance) {
    if (parseInt(header2.css('padding-top'), 10) >= header2_initialPadding
        header2.css('padding-top', '' + $(window).scrollTop() - header2_initial
    }
} else {
    header2.css('padding-top', '' + header2_initialPadding + 'px');
```

Working Successfully



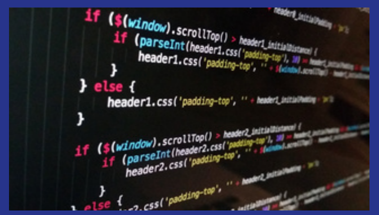
```
if ($(window).scrollTop() > header1_initialDistance) {
    if (parseInt(header1.css('padding-top'), 10) >= header1_initialPadding
        header1.css('padding-top', '' + $(window).scrollTop() - header1_initialDistance
    }
} else {
    header1.css('padding-top', '' + header1_initialPadding + 'px');
}

if ($(window).scrollTop() > header2_initialDistance) {
    if (parseInt(header2.css('padding-top'), 10) >= header2_initialPadding
        header2.css('padding-top', '' + $(window).scrollTop() - header2_initialDistance
    }
} else {
    header2.css('padding-top', '' + header2_initialPadding + 'px');
```

Let me test on my machine
Tester

```
if ($(window).scrollTop() > header1_initialDistance) {
    if (parseInt(header1.css('padding-top'), 10) >= header1_initialPadding
        header1.css('padding-top', '' + $(window).scrollTop() - header1_initialDistance
    }
} else {
    header1.css('padding-top', '' + header1_initialPadding + 'px');
}

if ($(window).scrollTop() > header2_initialDistance) {
    if (parseInt(header2.css('padding-top'), 10) >= header2_initialPadding
        header2.css('padding-top', '' + $(window).scrollTop()
    }
} else {
    header2.css('padding-top', '' + header2_initialPadding + 'px');
```

Developer

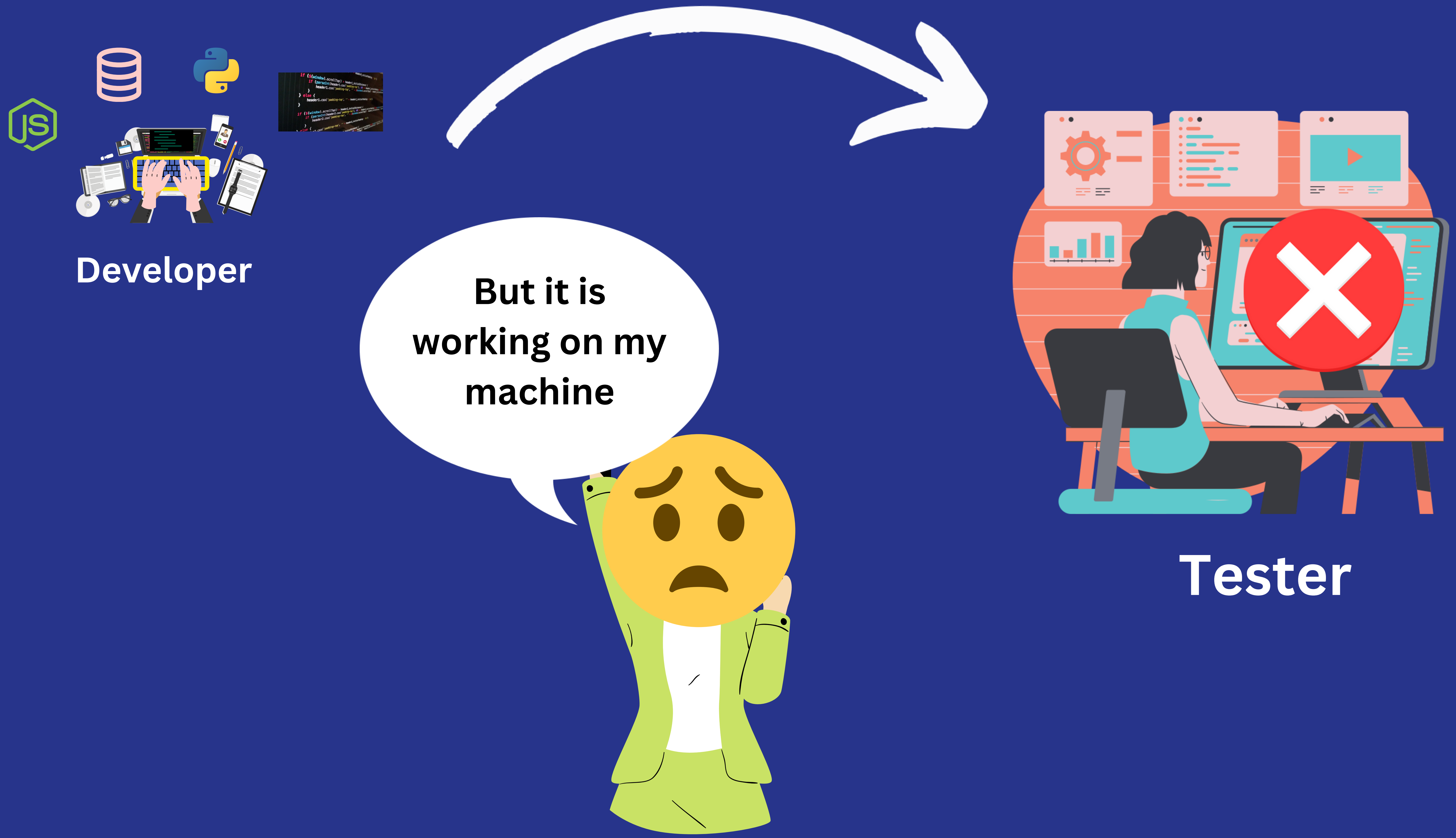
Tester



```
if ($(window).scrollTop() > header1_initialDistance) {
    if (parseInt(header1.css('padding-top'), 10) == header1_initialPadding
        header1.css('padding-top', '' + $(window).scrollTop() - header1_initialDistance
    }
} else {
    header1.css('padding-top', '' + header1_initialPadding + 'px');
}

if ($(window).scrollTop() > header2_initialDistance) {
    if (parseInt(header2.css('padding-top'), 10) == header2_initialPadding
        header2.css('padding-top', '' + $(window).scrollTop()
    }
} else {
    header2.css('padding-top', '' + header2_initialPadding + 'px');
```

Developer

Tester



Developer

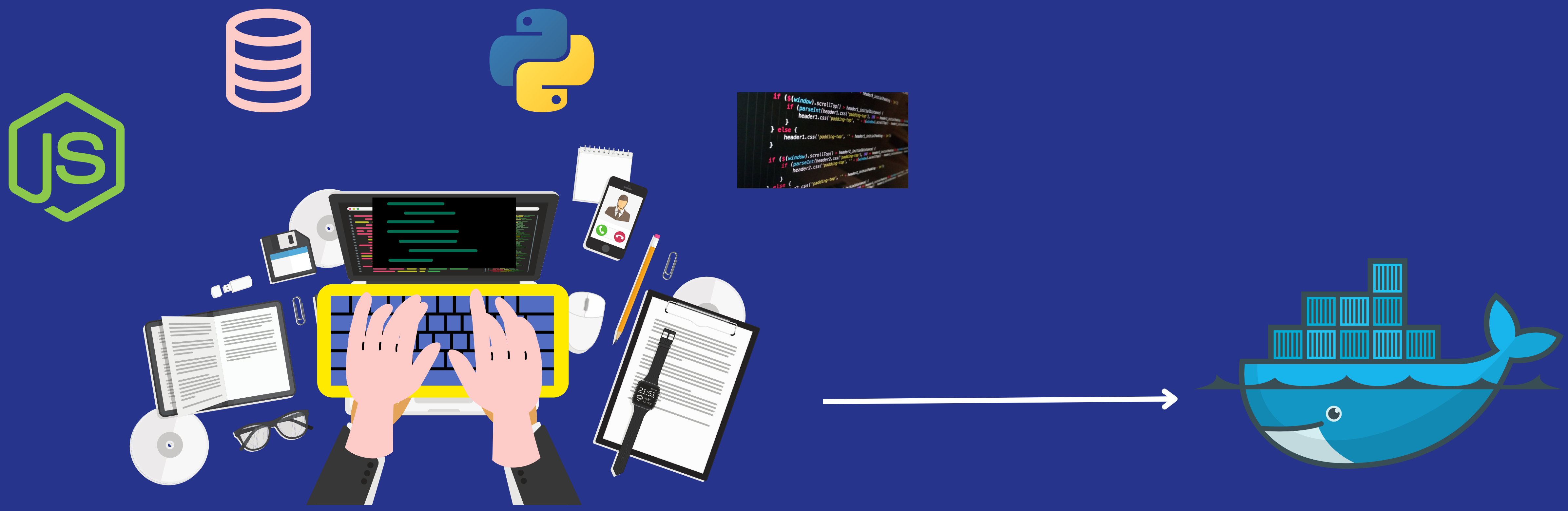
But it is working on my machine

Tester

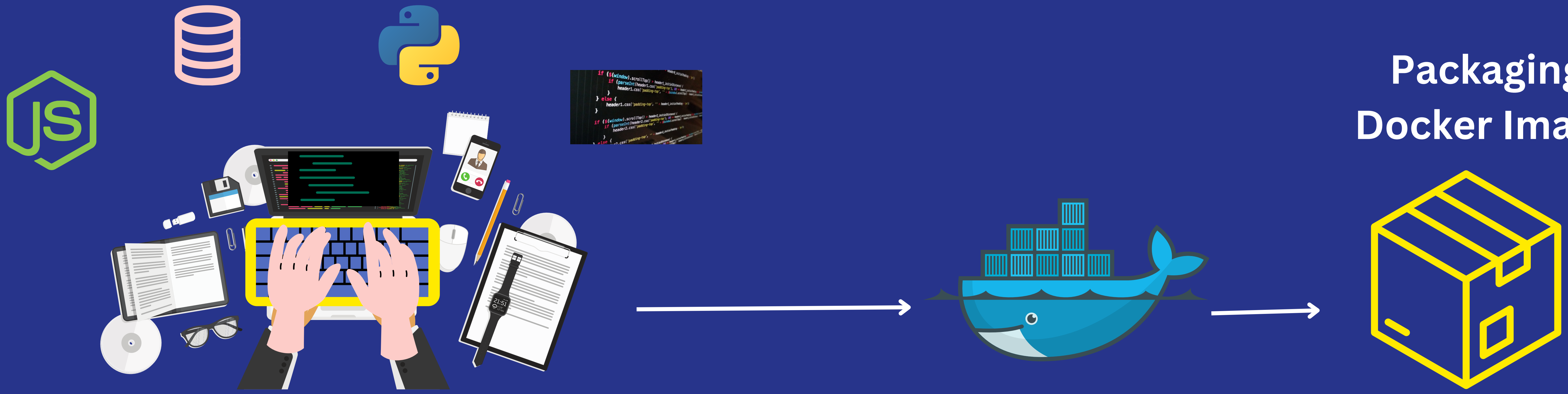
What is the solution now?

No worries
We have

Docker

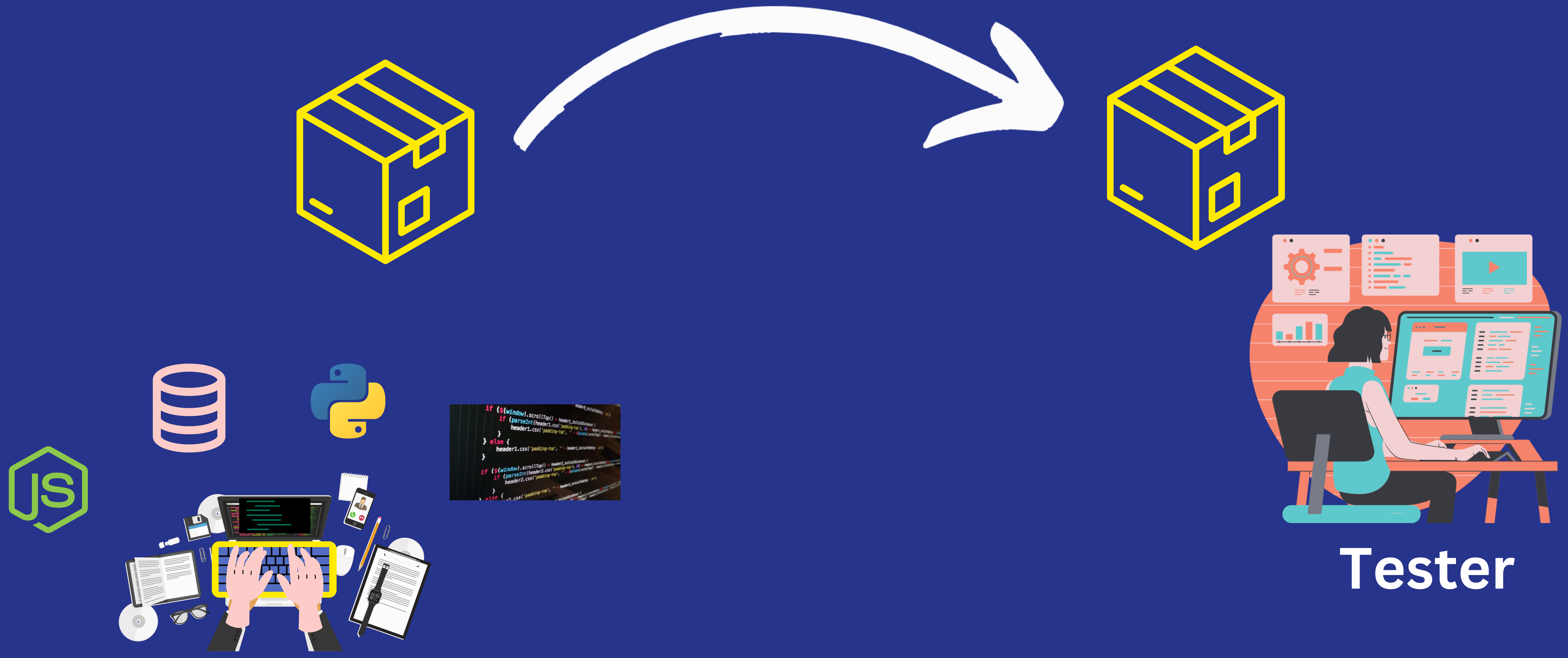


Developer



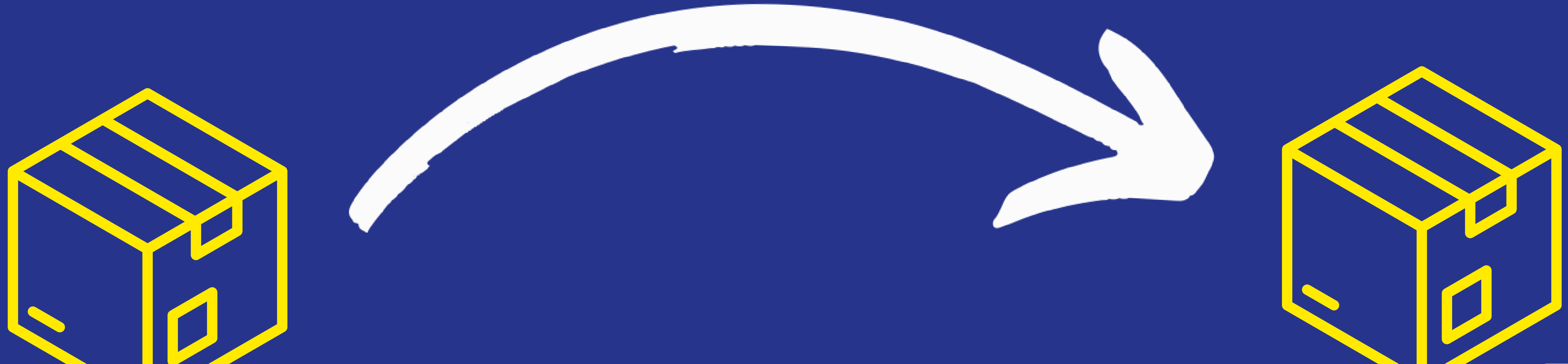
Packaging
Docker Image

Developer



Developer

Tester



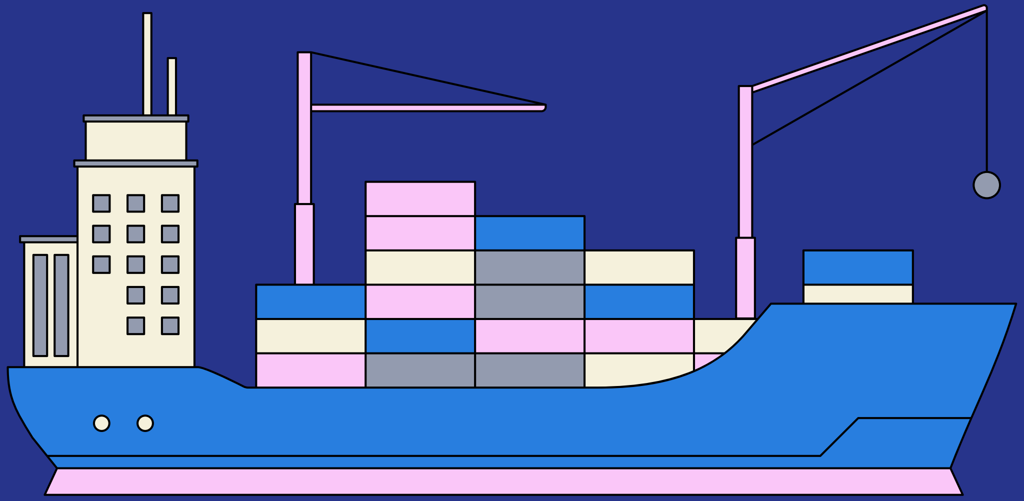
```
if ($(window).scrollTop() > header1_initialDistance) {
    if (parseInt(header1.css('padding-top'), 10) <= header1_initialPadding
        header1.css('padding-top', '' + $(window).scrollTop() - header1_initialDistance
    }
} else {
    header1.css('padding-top', '' + header1_initialPadding + 'px');
}

if ($(window).scrollTop() > header2_initialDistance) {
    if (parseInt(header2.css('padding-top'), 10) <= header2_initialPadding
        header2.css('padding-top', '' + $(window).scrollTop()
    }
} else {
```

Developer

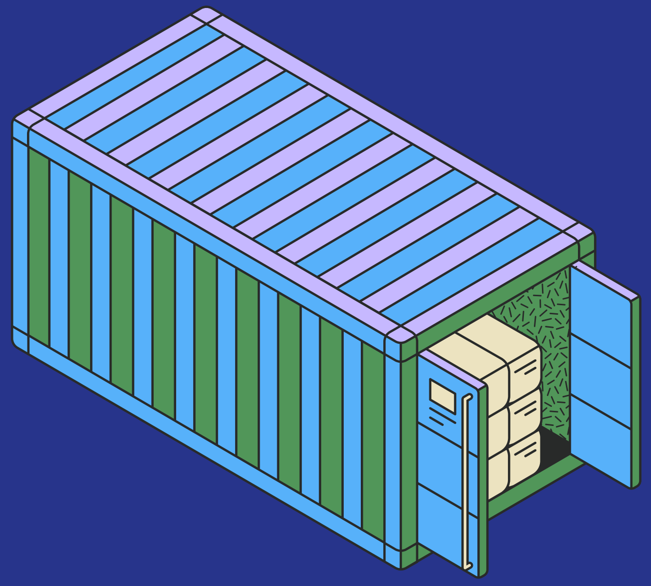
Tester

How did it work?

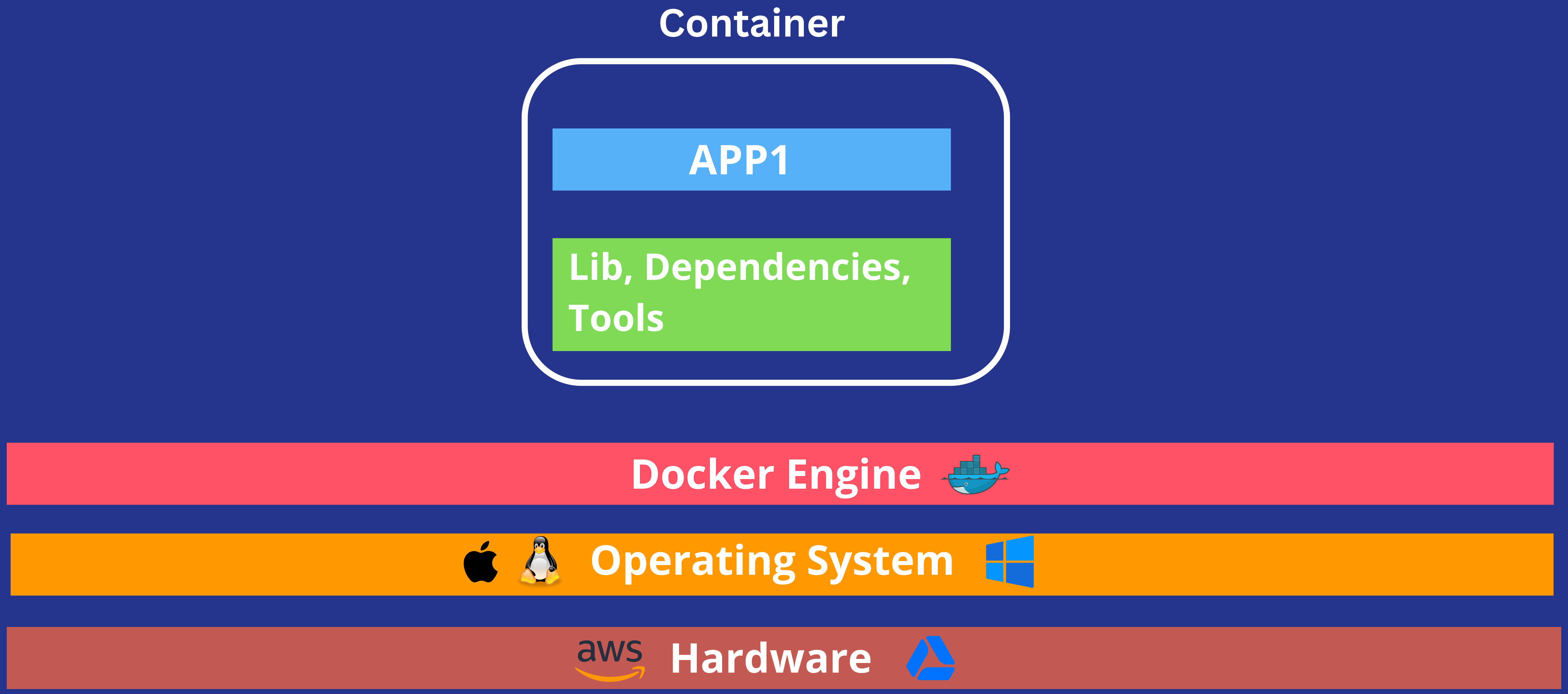
What is a Container?



What is a Container?



- A way to package an application with all the necessary dependencies and configuration.
- It can be easily shared
- Makes deployment and development efficient.



Container

APP1

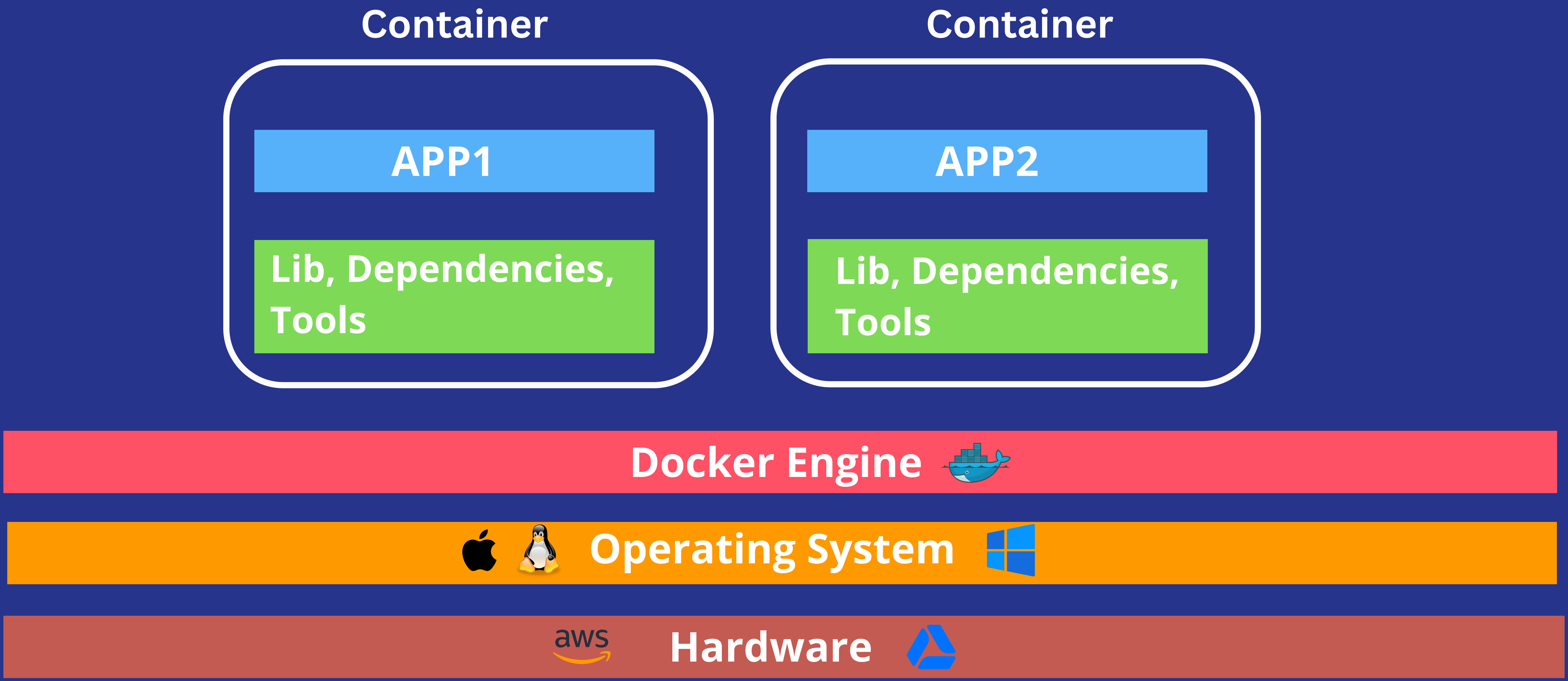
Lib, Dependencies, Tools

Docker Engine

Operating System

aws

Hardware



Container

APP1

Lib, Dependencies, Tools

Container

APP2

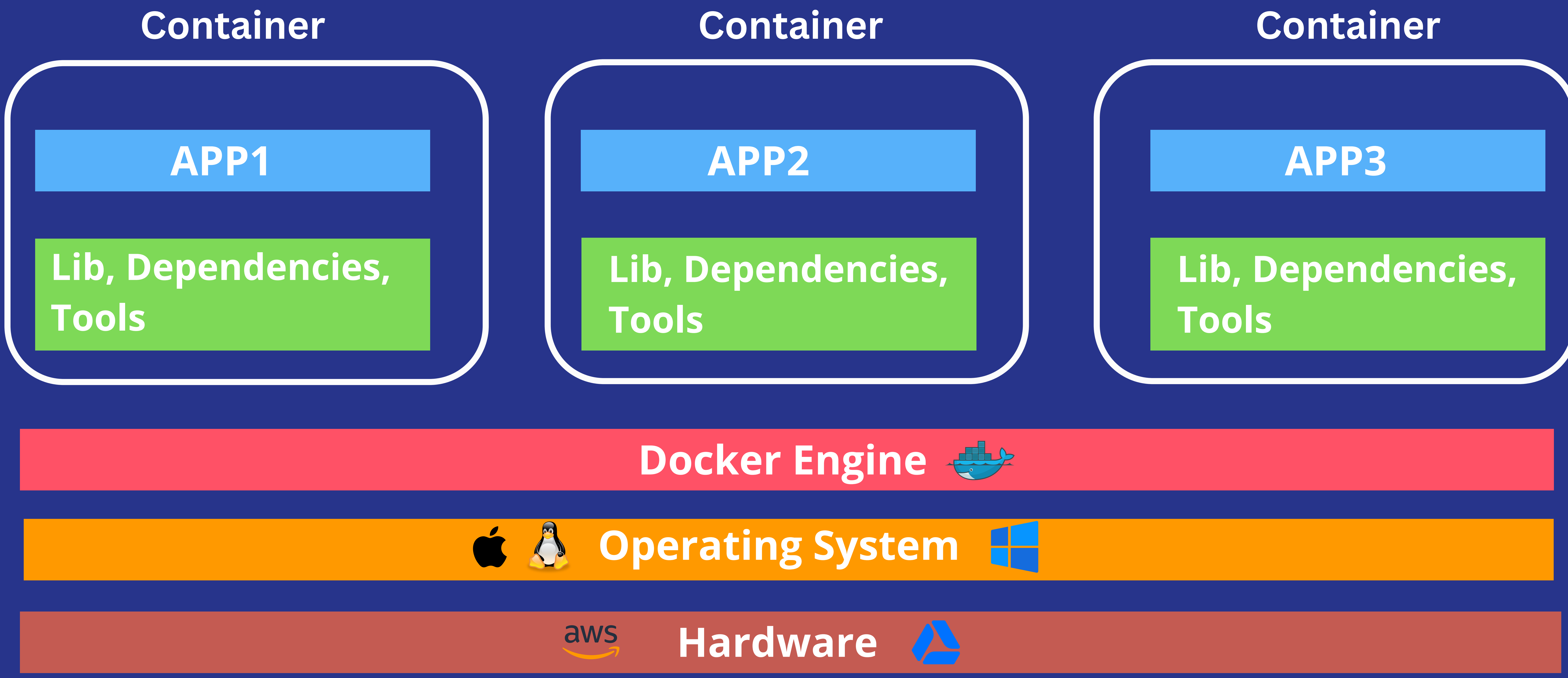
Lib, Dependencies, Tools

Docker Engine

Operating System

aws

Hardware



Container

APP1

Lib, Dependencies, Tools

Container

APP2

Lib, Dependencies, Tools

Container

APP3

Lib, Dependencies, Tools

Docker Engine

Operating System

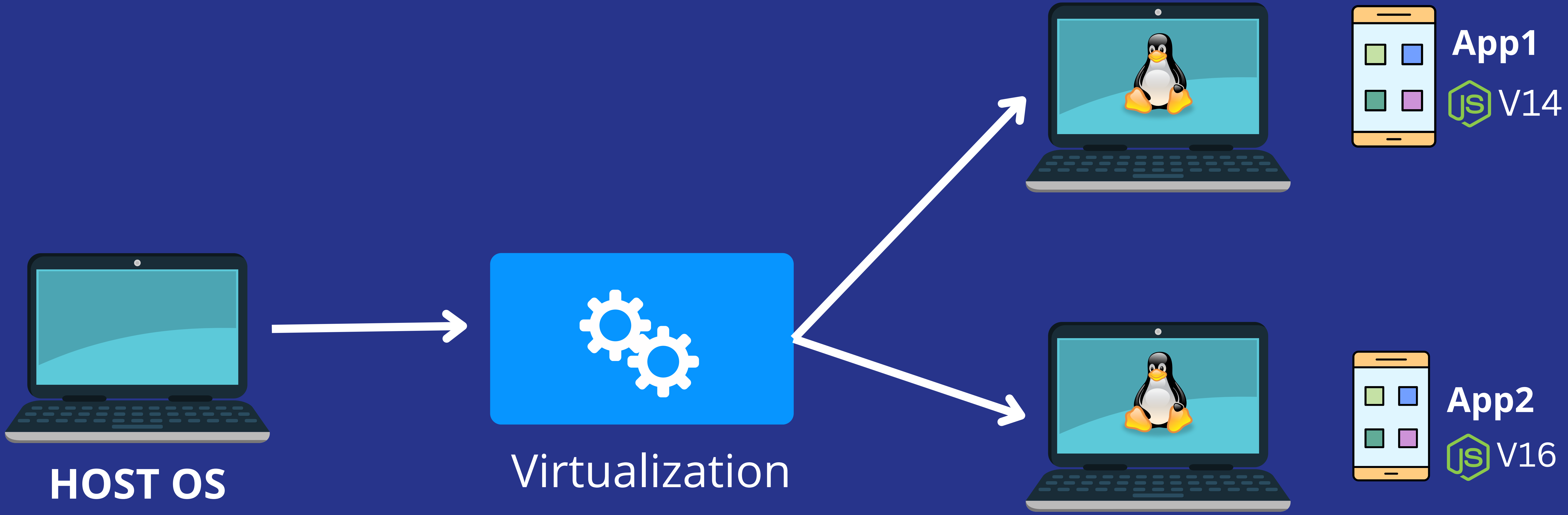
Hardware

App1

V14

App2

V16



HOST OS

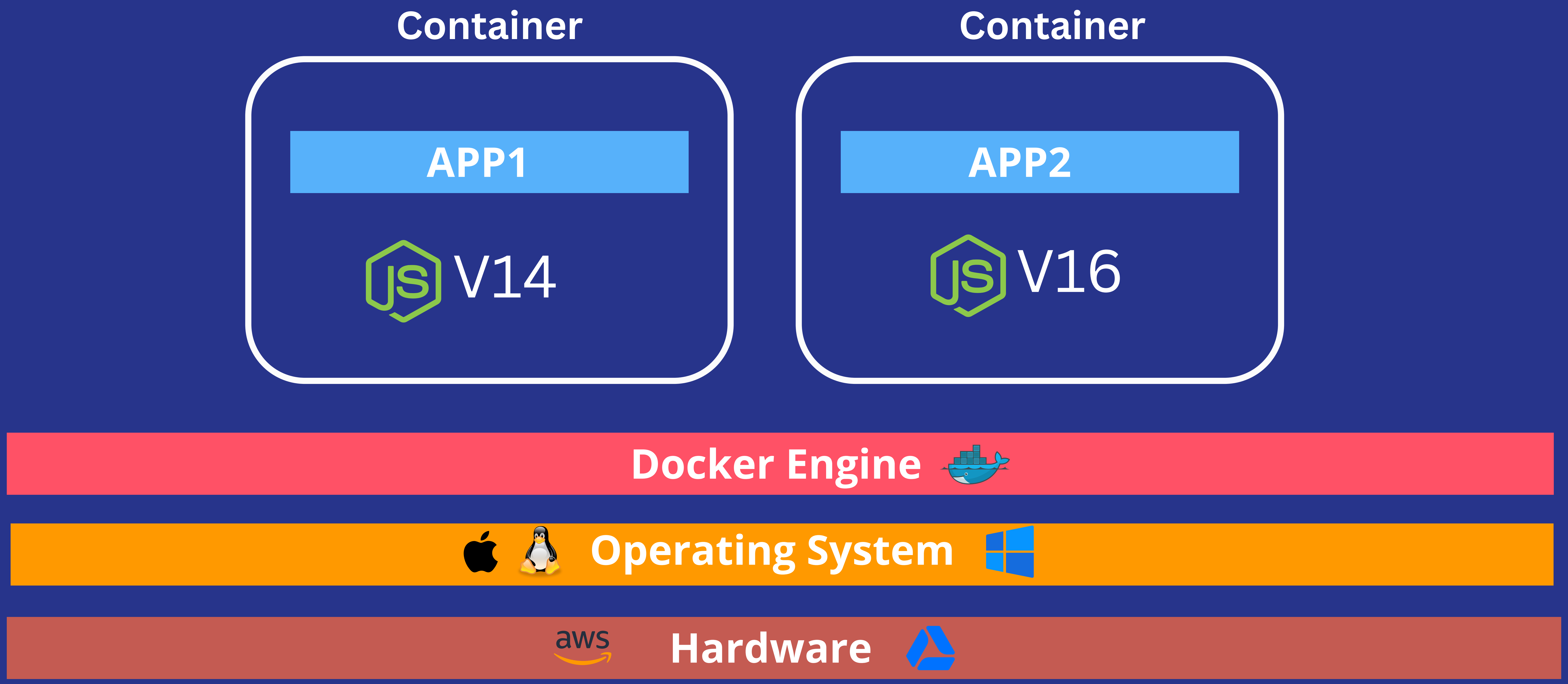
Virtualization

App1

V14

App2

V16



Container

APP1

V14

Container

APP2

V16

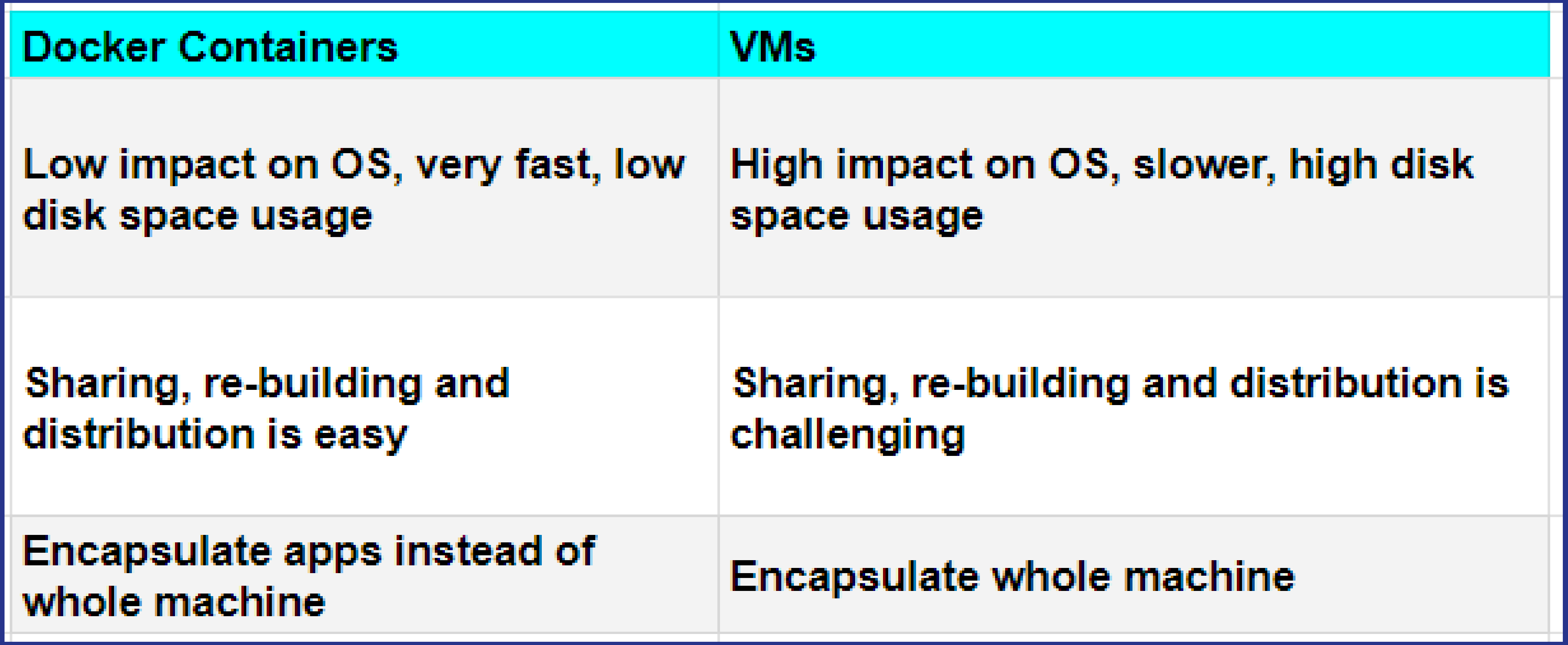
Docker Engine

Operating System

aws

Hardware

Docker vs VMs



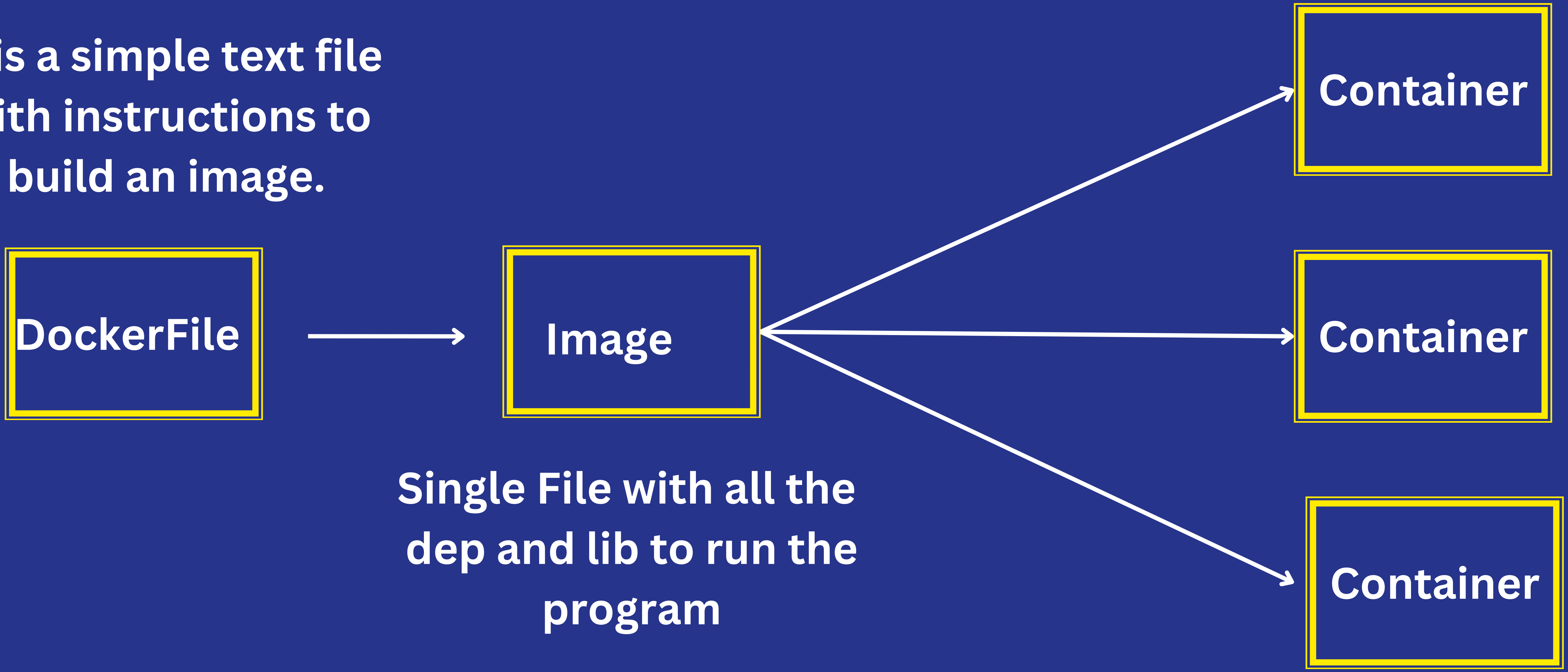
Docker Containers	VMs
Low impact on OS, very fast, low disk space usage	High impact on OS, slower, high disk space usage
Sharing, re-building and distribution is easy	Sharing, re-building and distribution is challenging
Encapsulate apps instead of whole machine	Encapsulate whole machine

Main components of Docker

- DockerFile
- Docker Image
- Docker Container
- Docker Registry

Instance of an Image

It is a simple text file with instructions to build an image.



DockerFile

Image

Container

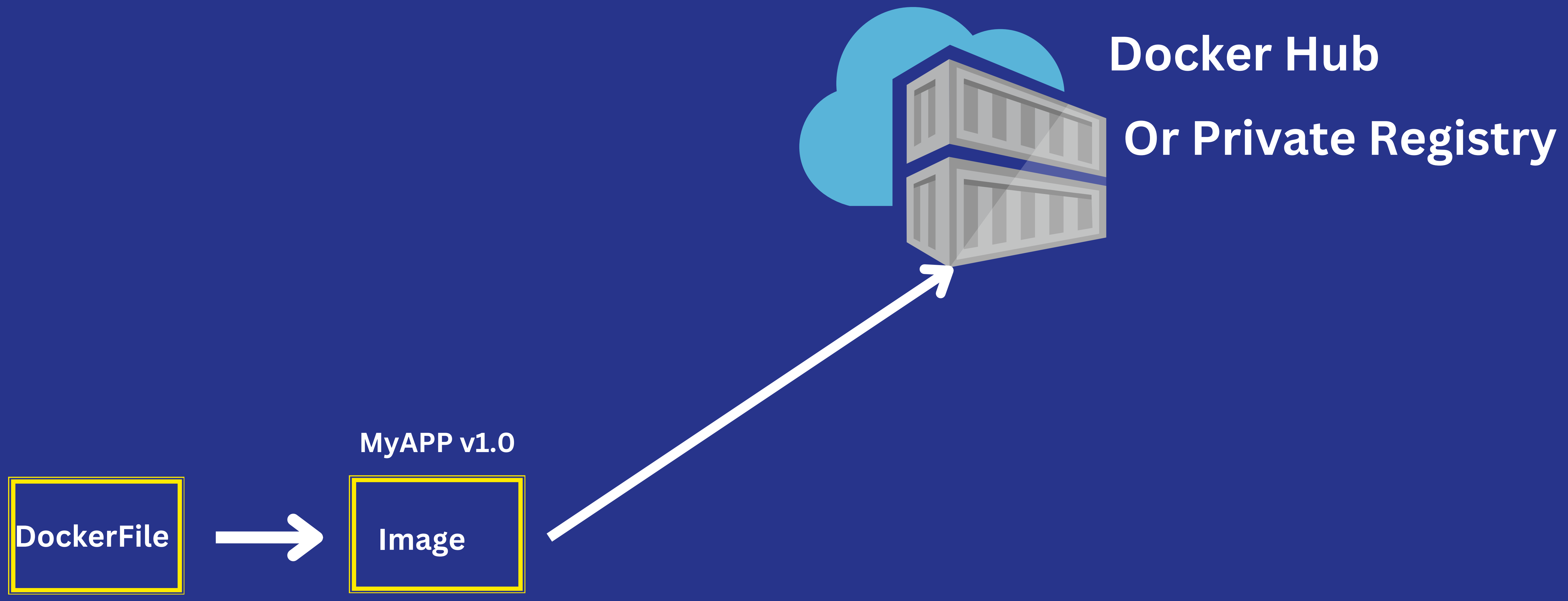
Container

Container

Single File with all the dep and lib to run the program

Docker Registry

A Docker registry is a central repository for storing and distributing Docker images.



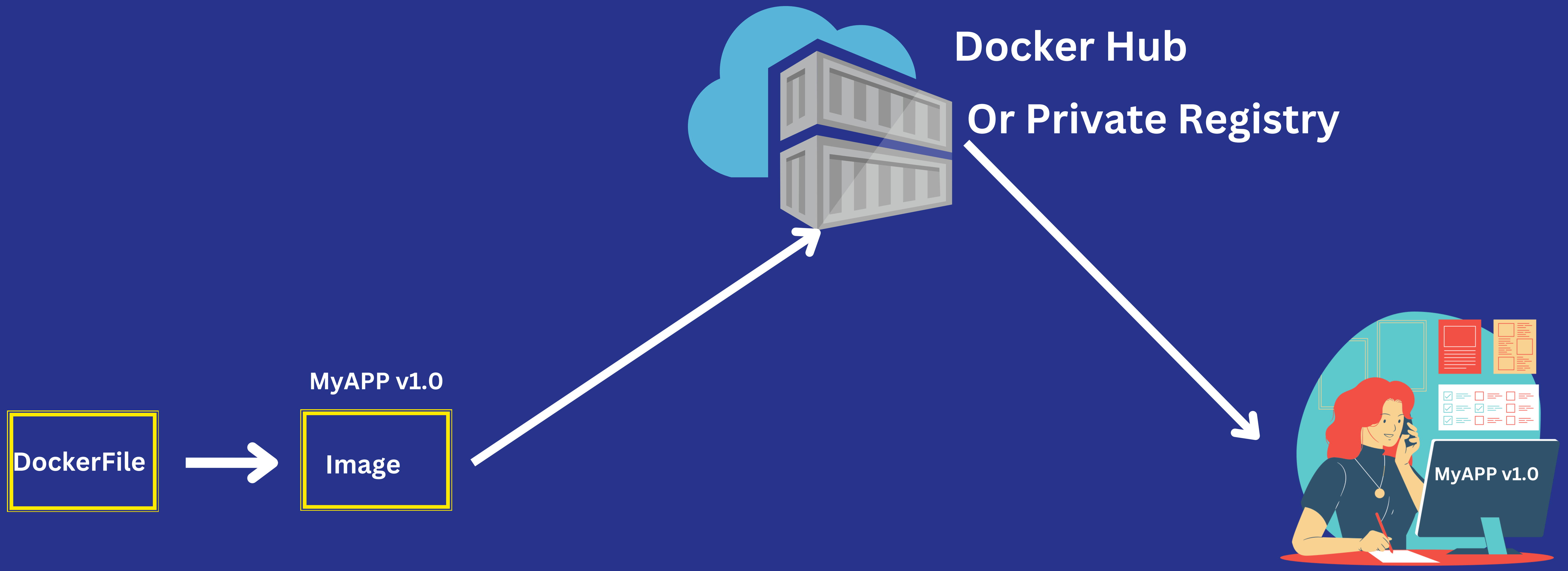
Docker Hub

Or Private Registry

MyAPP v1.0

DockerFile

Image



DockerFile

MyAPP v1.0

Image

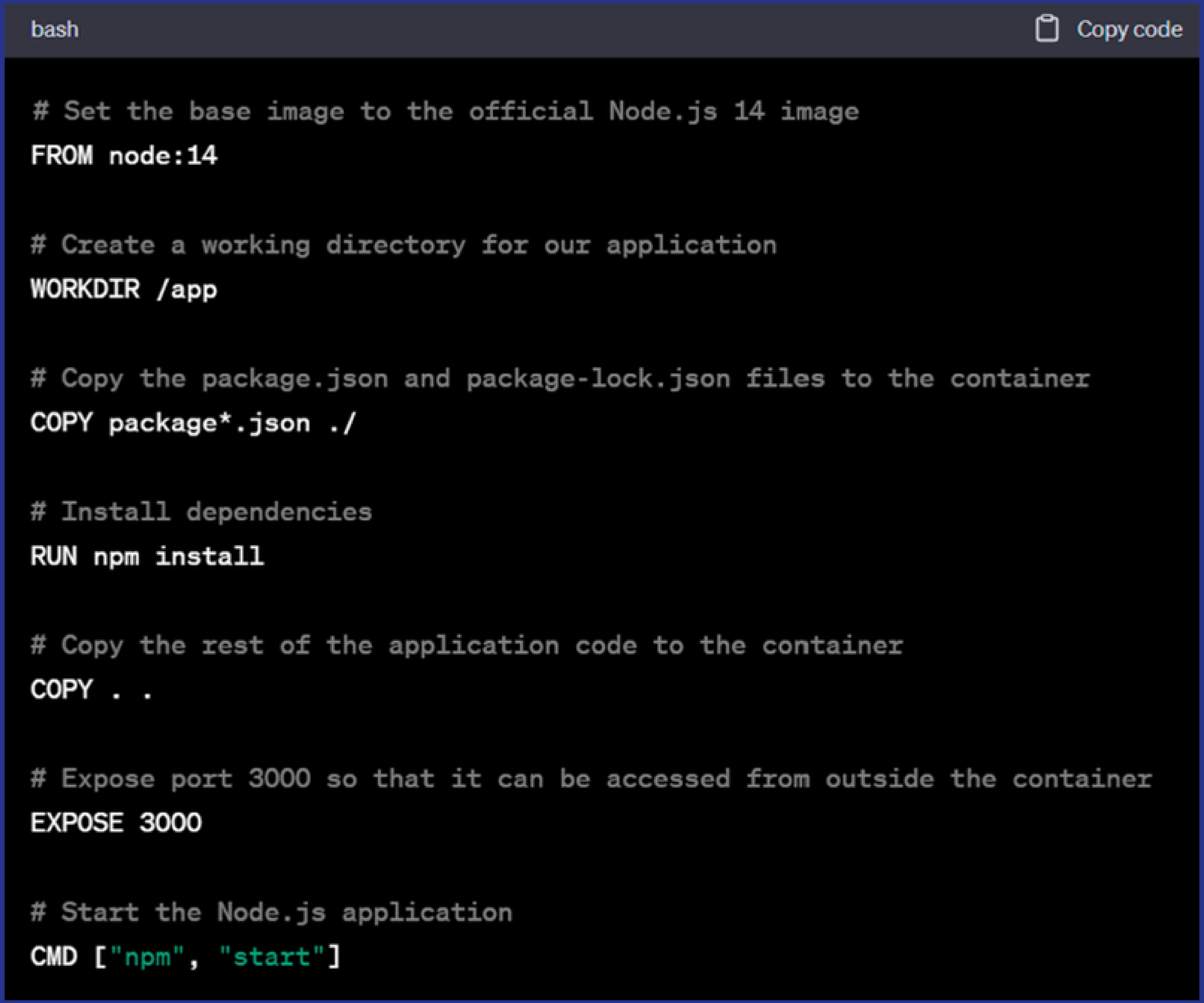
Docker Hub

Or Private Registry

MyAPP v1.0

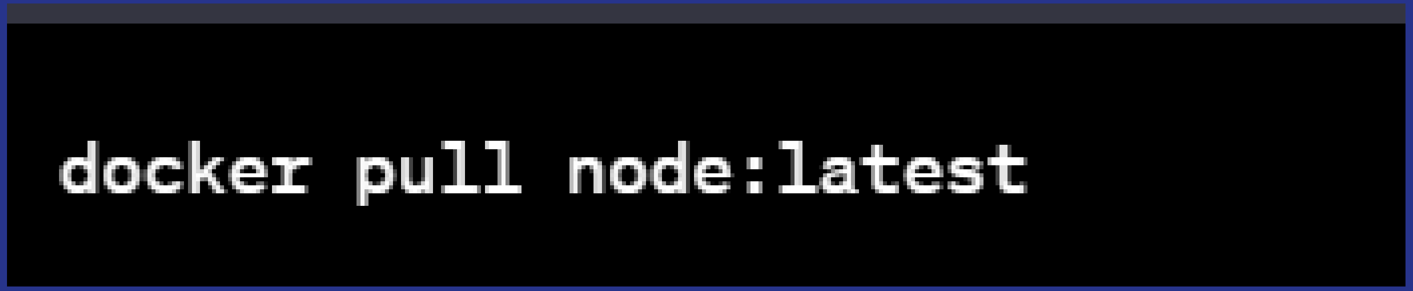
Conclusion:

- Docker is a powerful technology that allows developers to create, package, and deploy applications in containers.
- It provides a consistent environment for development, testing, and deployment, and it's compatible with any platform that supports Docker.
- By using Docker, developers can focus on building great applications instead of worrying about infrastructure and compatibility issues.

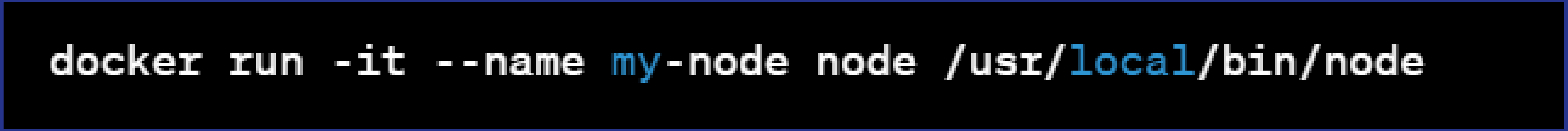


```
# Set the base image to the official Node.js 14 image
FROM node:14

# Create a working directory for our application
WORKDIR /app

# Copy the package.json and package-lock.json files to the container
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application code to the container
COPY . .

# Expose port 3000 so that it can be accessed from outside the container
EXPOSE 3000

# Start the Node.js application
CMD ["npm", "start"]
```

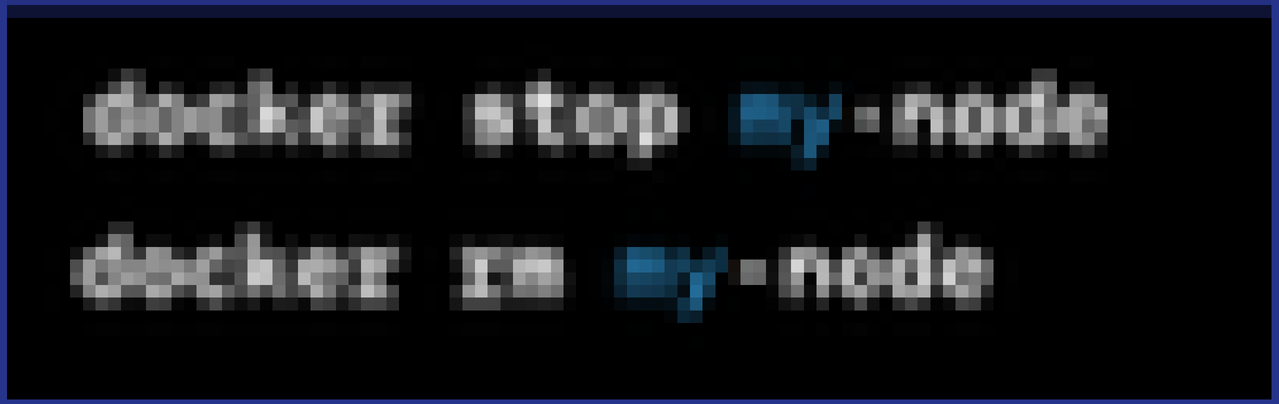
```
docker build -t my-image
```



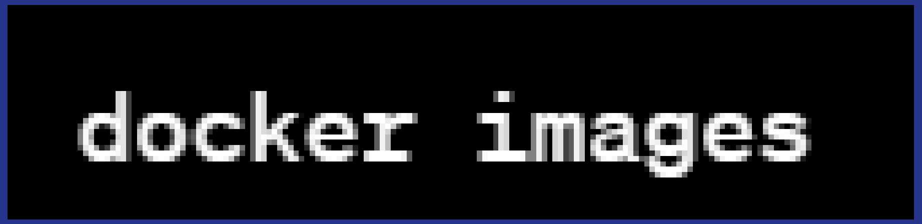
```
docker pull node:latest
```



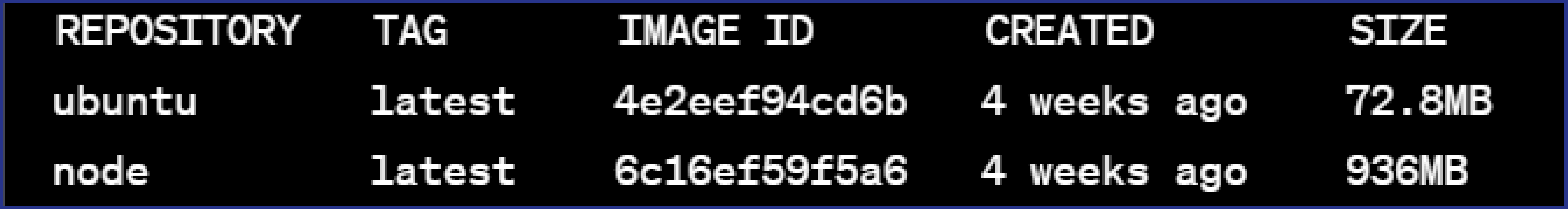
```
docker run -it --name my-node node /usr/local/bin/node
```



```
docker stop my-node
docker rm my-node
```

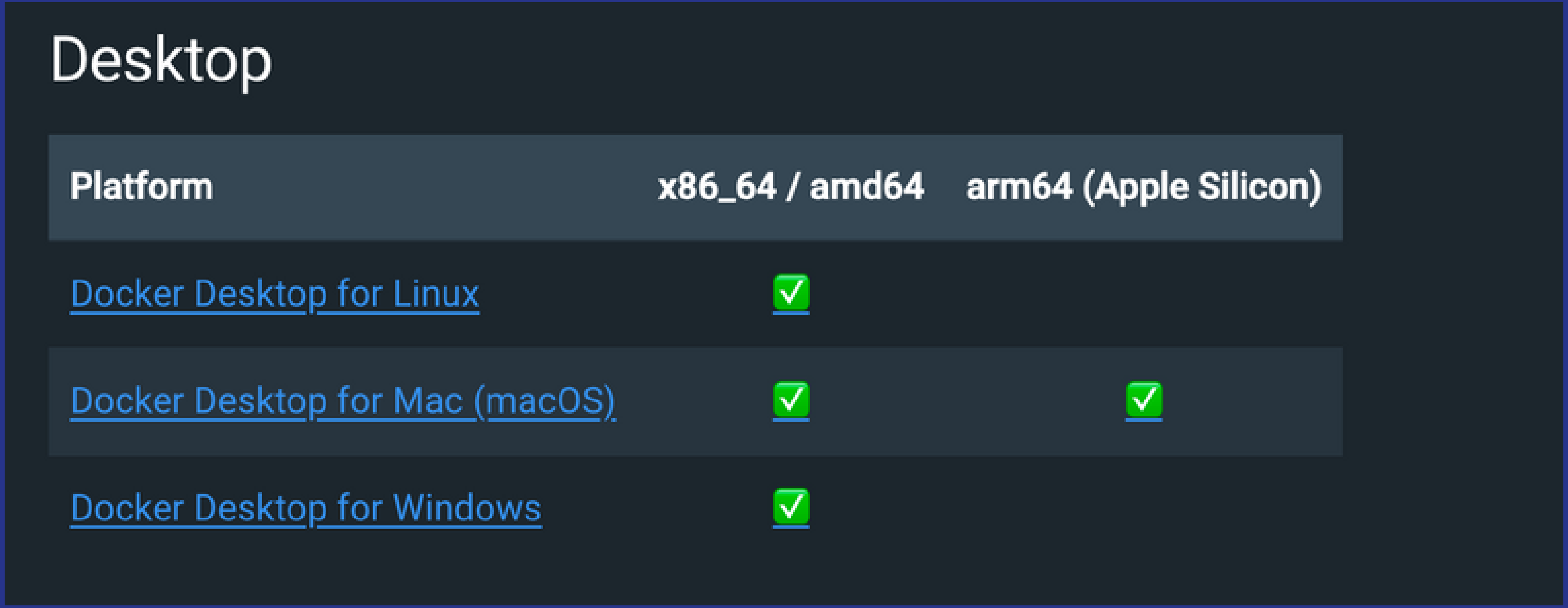


```
docker images
```



```
REPOSITORY   TAG      IMAGE ID       CREATED       SIZE
ubuntu       latest   4e2eef94cd6b   4 weeks ago   72.8MB
node         latest   6c16ef59f5a6   4 weeks ago   936MB
```

Install Docker Engine

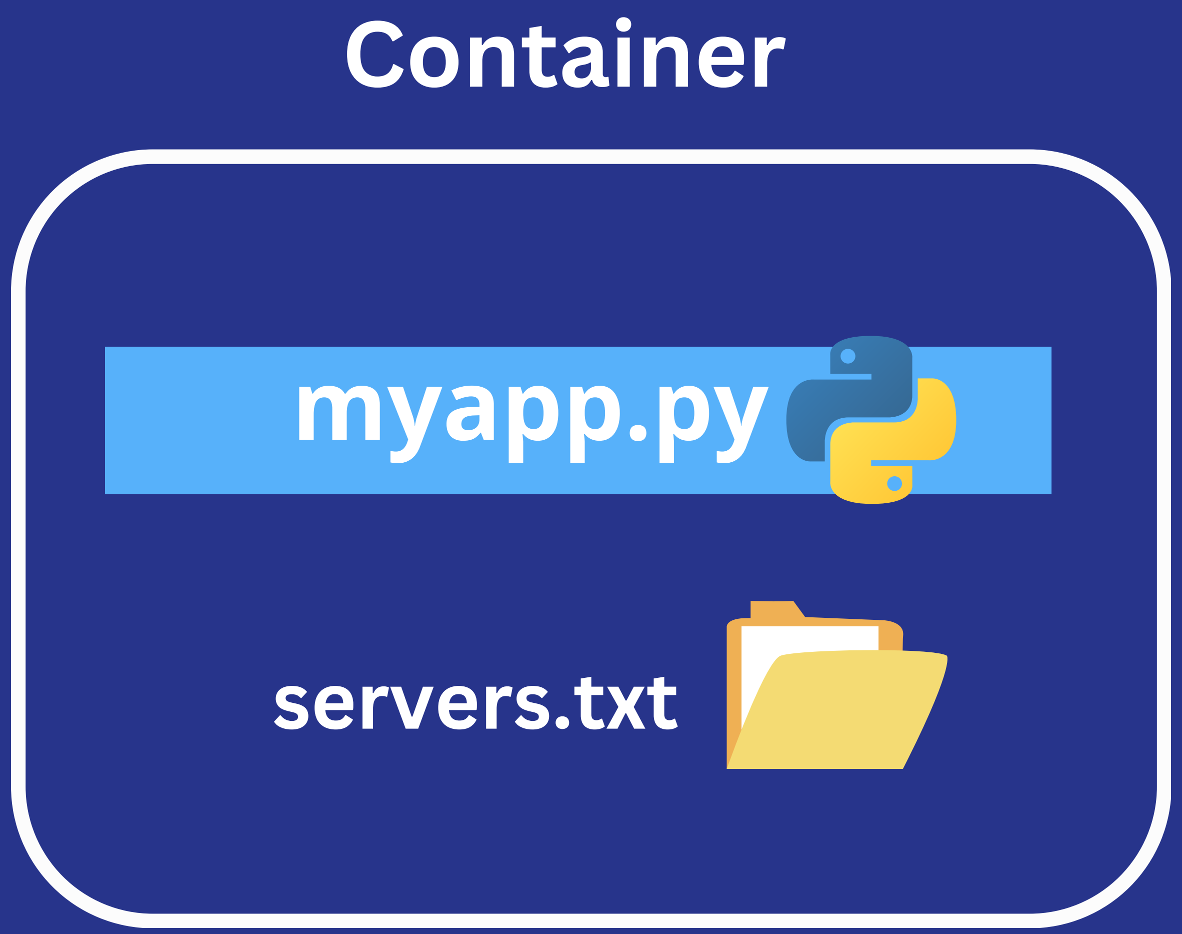


Desktop

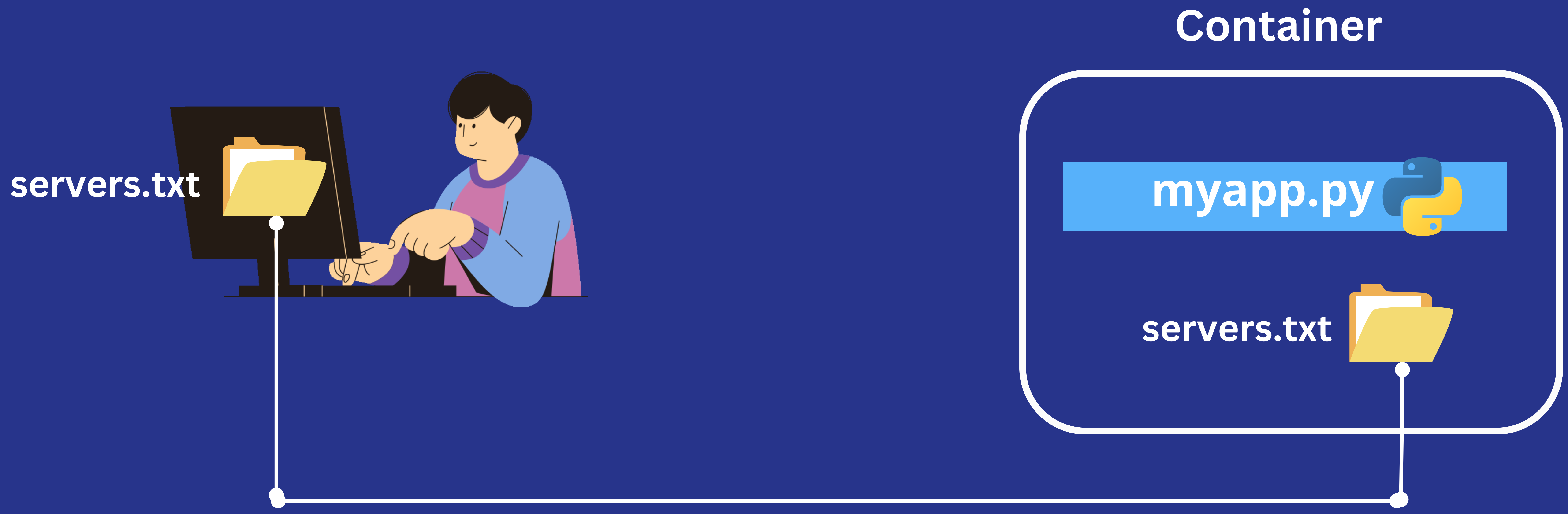
Platform	x86_64 / amd64	arm64 (Apple Silicon)
Docker Desktop for Linux	✅	
Docker Desktop for Mac (macOS)	✅	✅
Docker Desktop for Windows	✅	

Docker Volumes

servers.txt



Docker Volumes (bind mount)



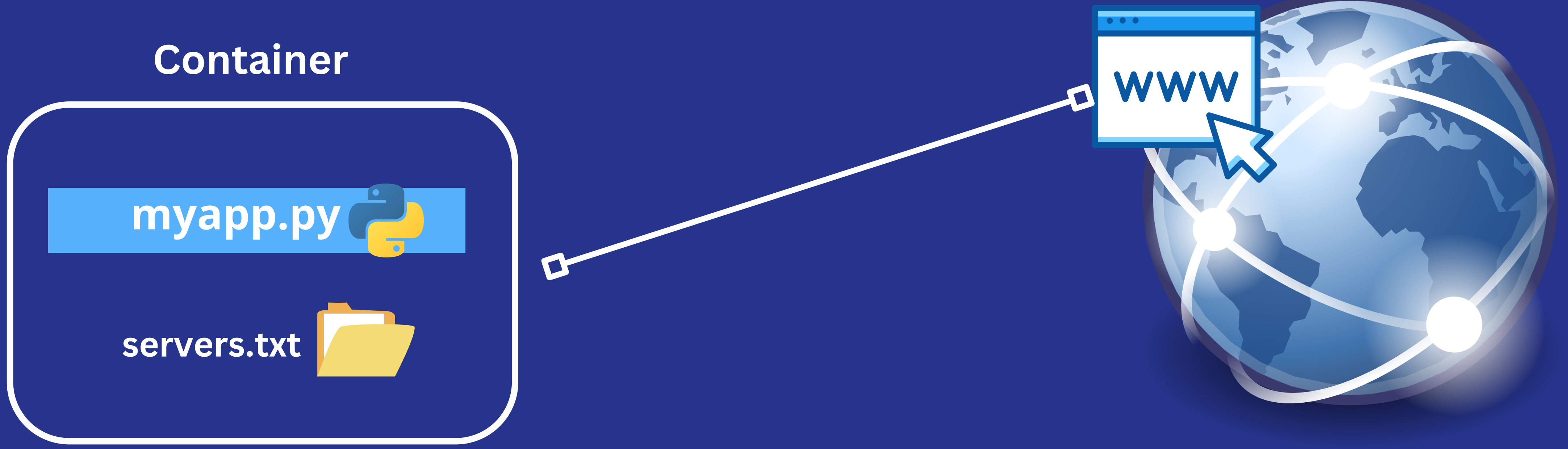
servers.txt

Container

myapp.py

servers.txt

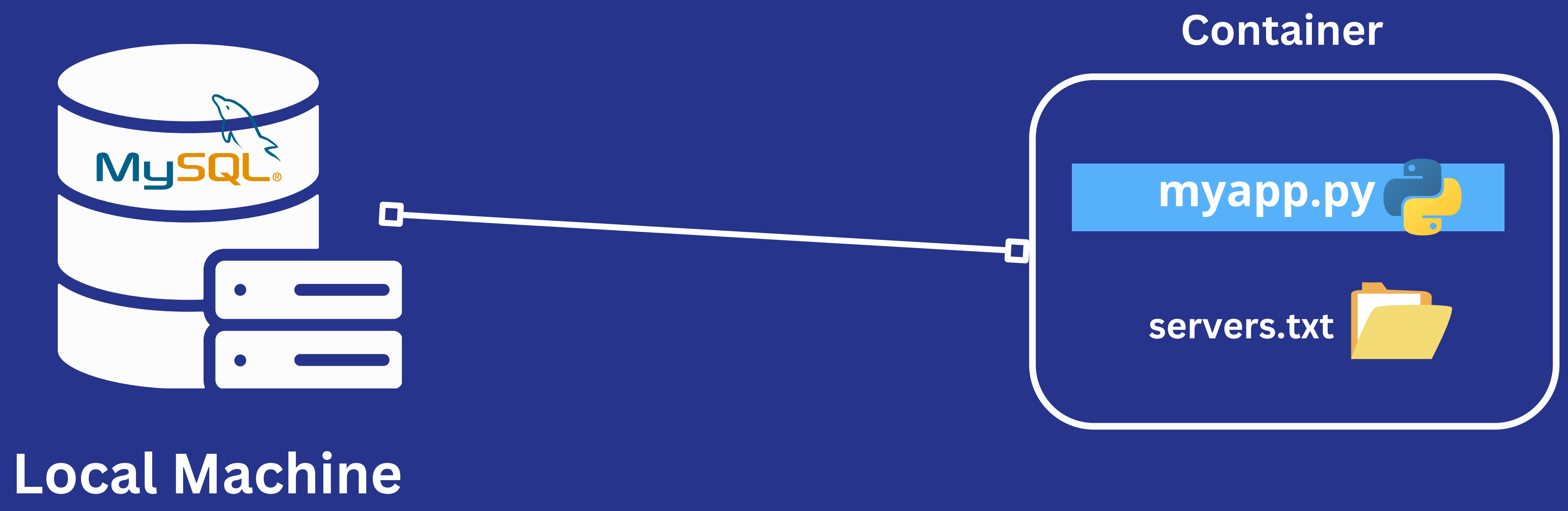
Communication From/To Containers

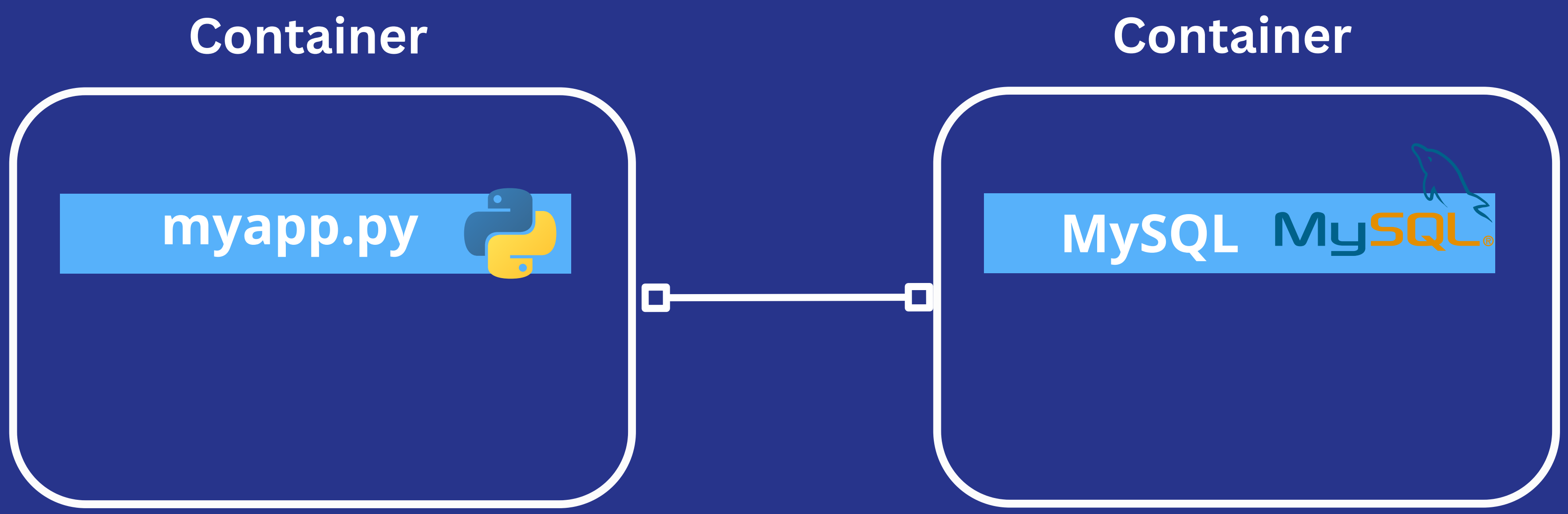


Container

myapp.py

servers.txt


Container
myapp.py
servers.txt
MySQL
Local Machine



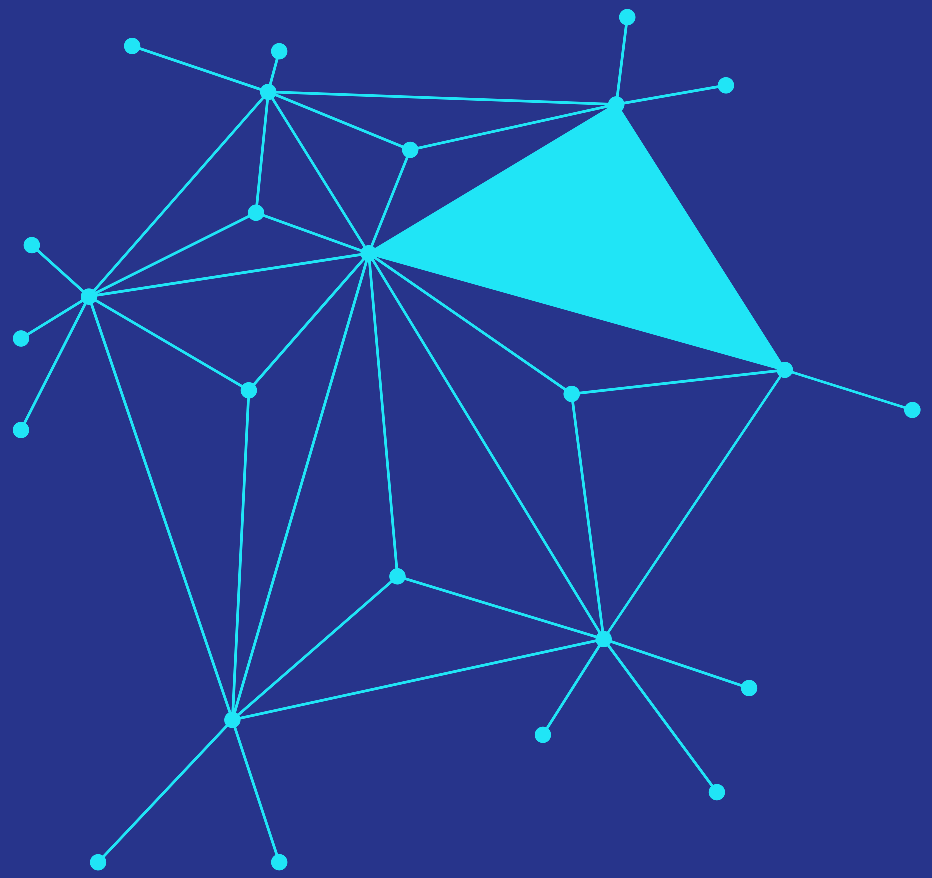
Container

myapp.py

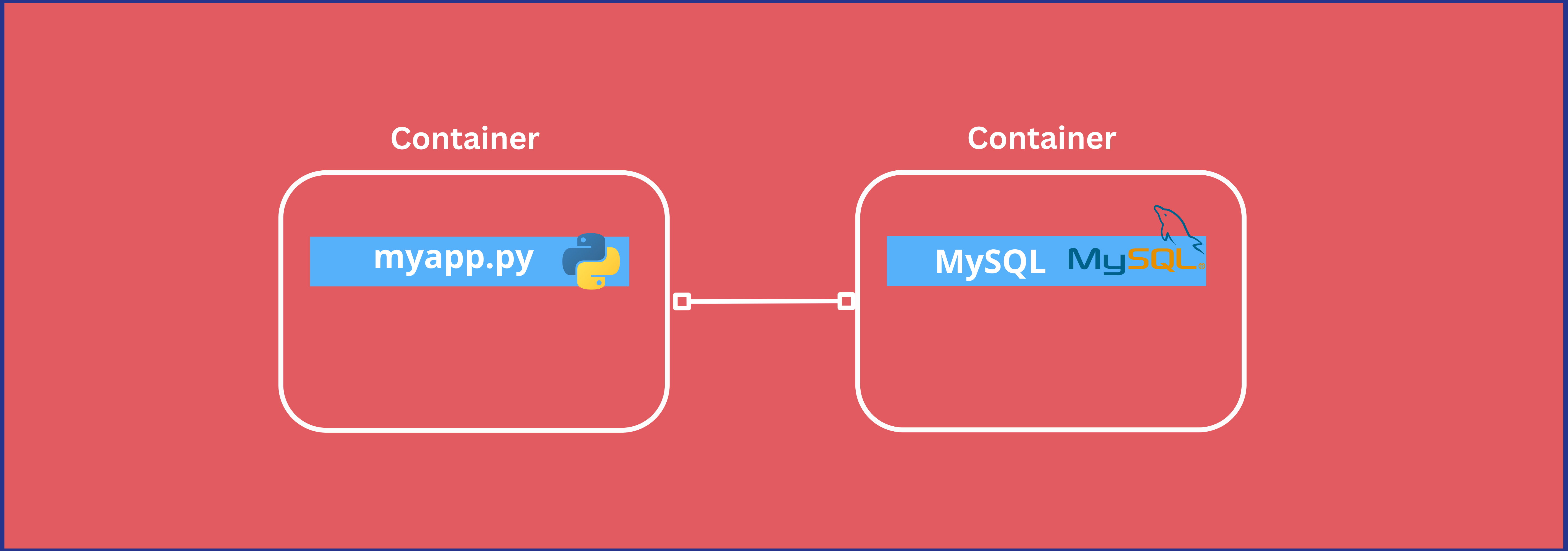
Container

MySQL

Docker Network



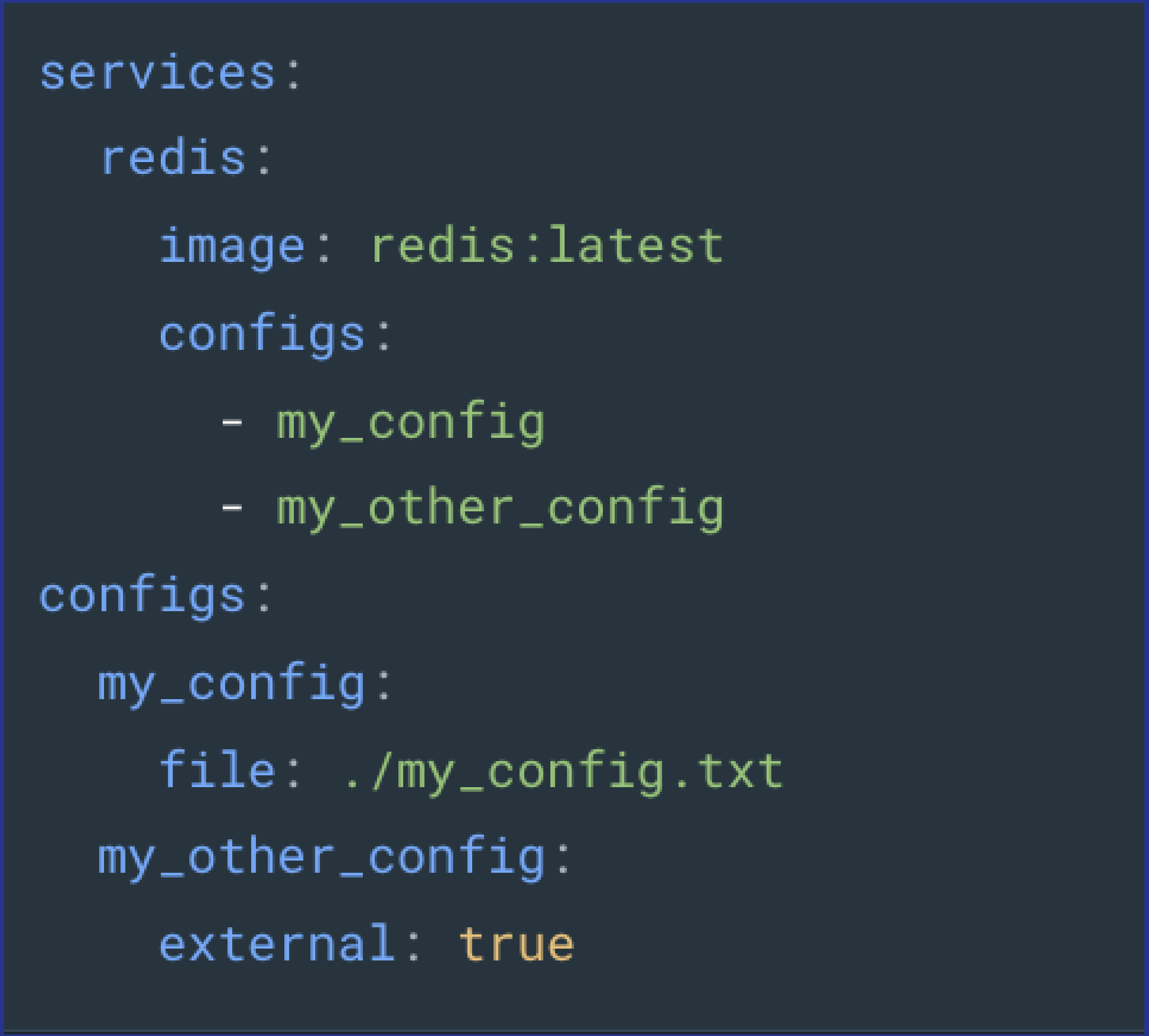
Network: my-net



--network my-net

Docker Compose

Configuration file to manage multiple containers running on same machine..



```yaml
services:
  redis:
    image: redis:latest
    configs:
      - my_config
      - my_other_config
configs:
  my_config:
    file: ./my_config.txt
  my_other_config:
    external: true
```

Thank you!

Please Like & SUBSCRIBE