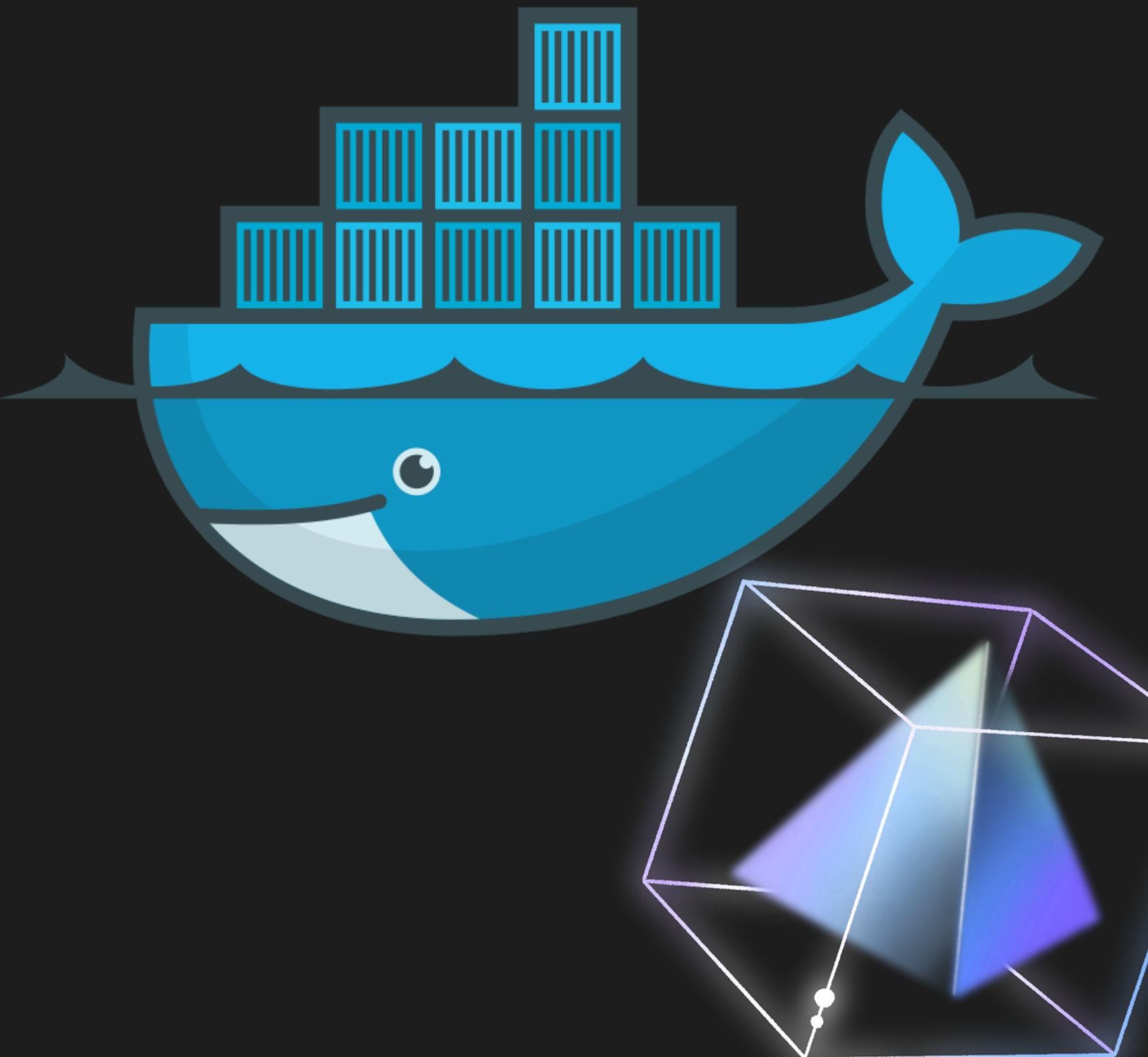


Present Title: Docker

What is -----  
**Docker?**



Present by Tar

# Content for Present:

Overview

Installation

How to use

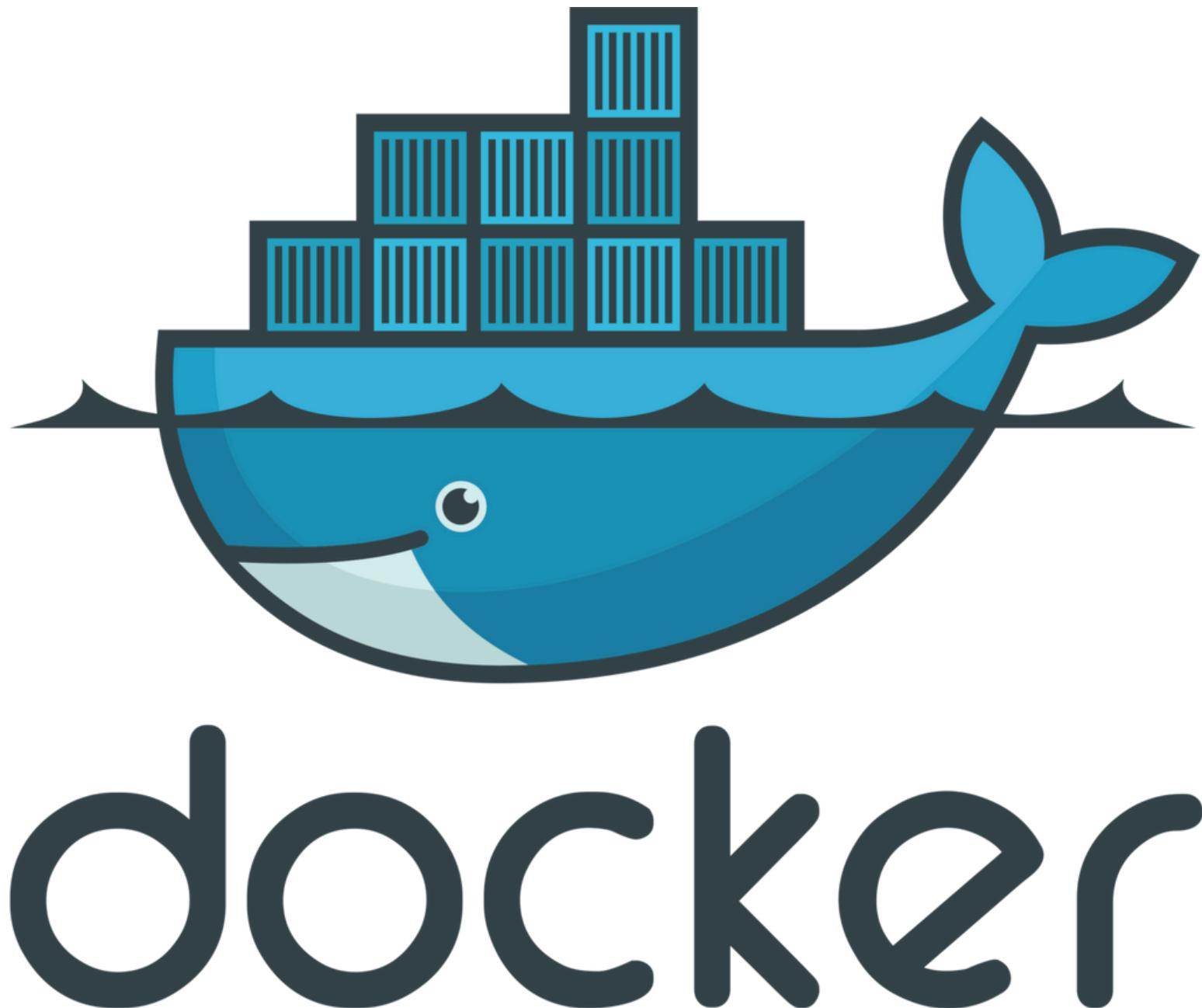
Example

Summarizes



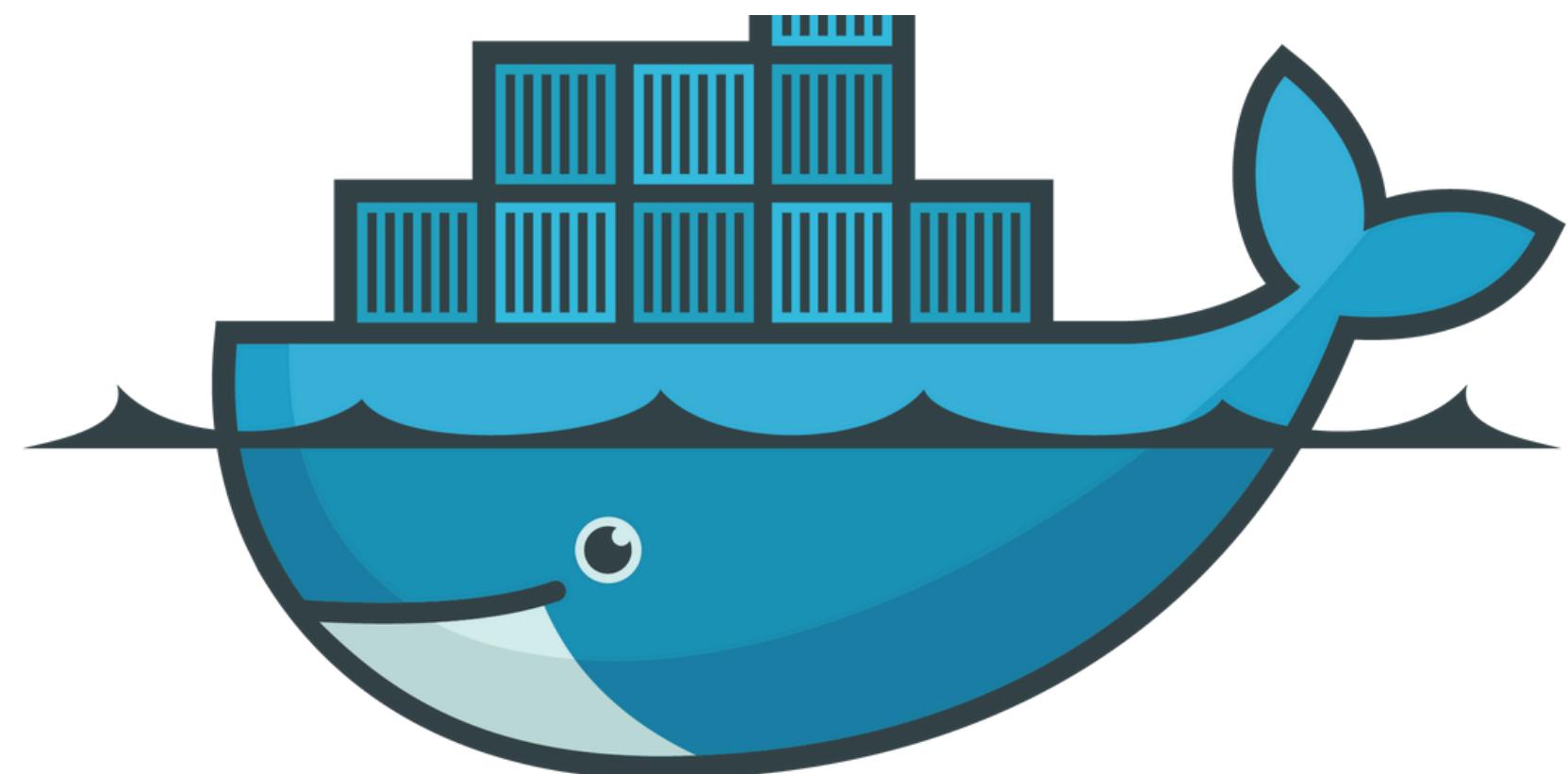
# Overview

- Docker is a platform for developing, shipping, and running applications inside lightweight, portable containers.
- Initially developed by Solomon Hykes at DotCloud, publicly introduced in 2013, and quickly open-sourced.
- Revolutionized software development with its innovative containerization technology.



# What is docker?

Docker is a tool that is used to automate the deployment of application in lightweight containers so that applications can work efficiently in different environments in isolation



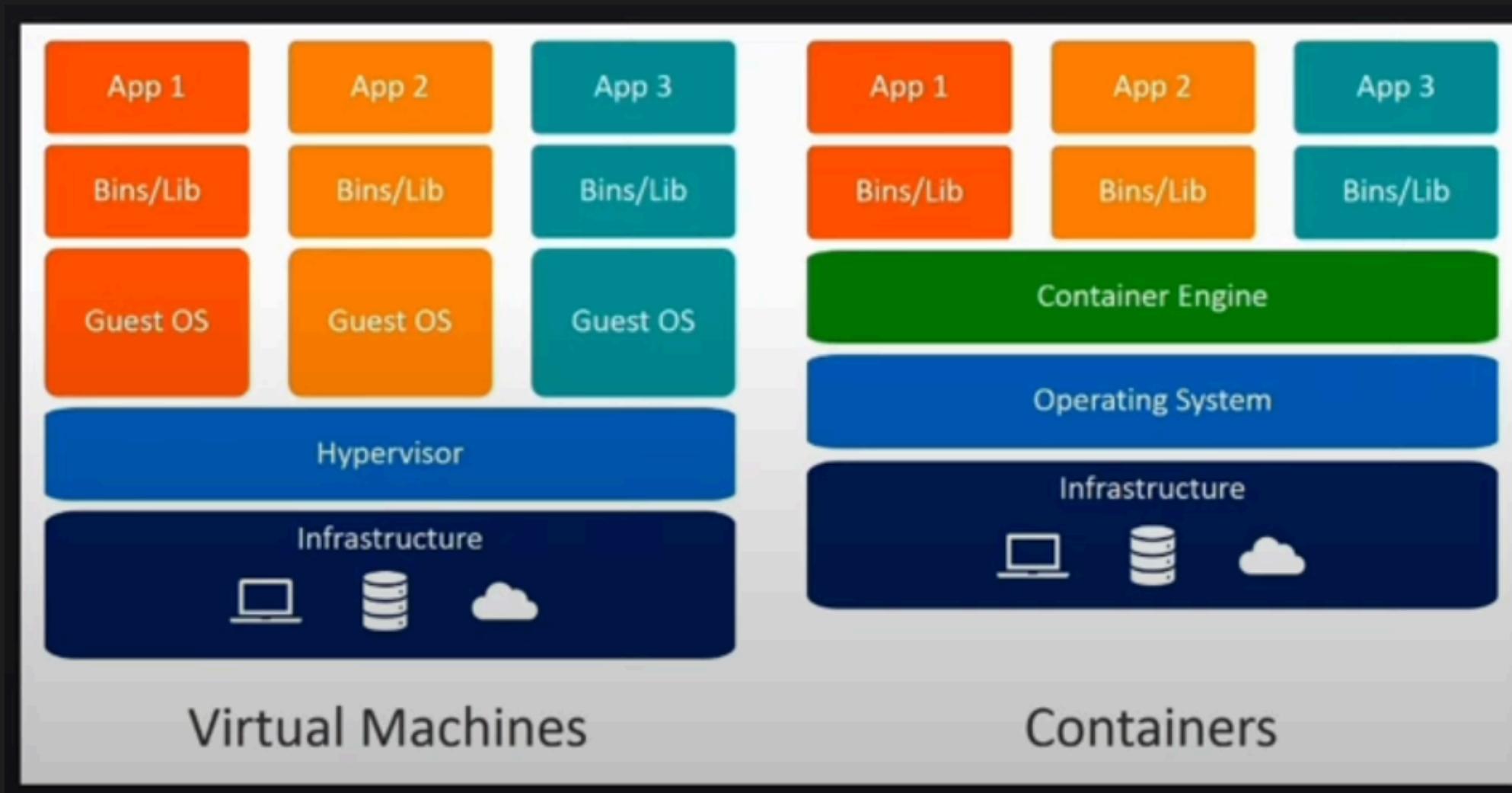
# docker

# How does Docker different from virtual machines (VM)?

Lightweight: Containers share the host OS kernel and do not require a full OS for each instance. This makes them much lighter and faster to start compared to VMs.

Isolation: Containers provide process and filesystem isolation using OS-level virtualization.

Efficiency: Containers use less disk space and memory, and they have less overhead because they don't need to run a separate OS for each instance.

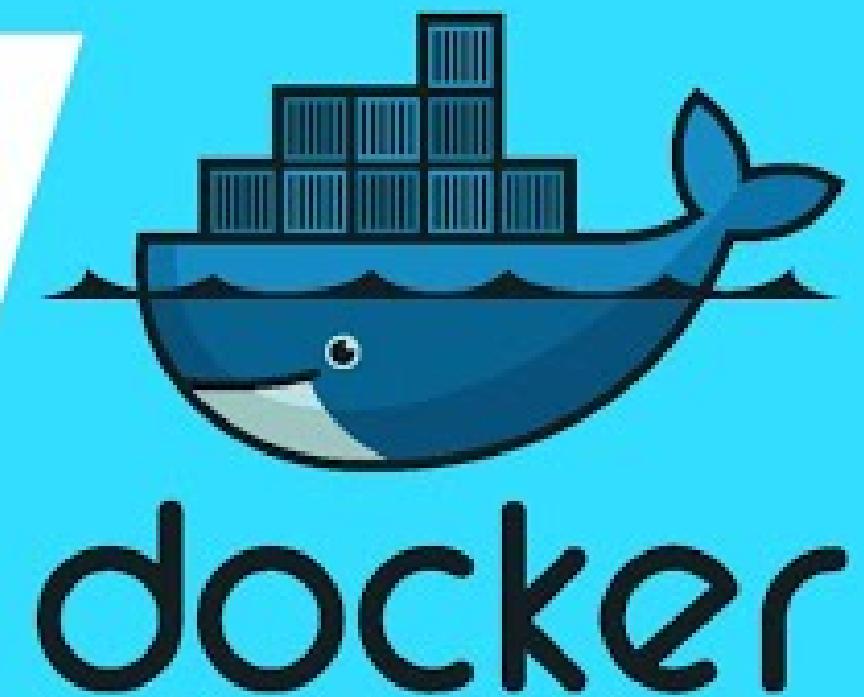


# The main elements of docker

1 Docker Images

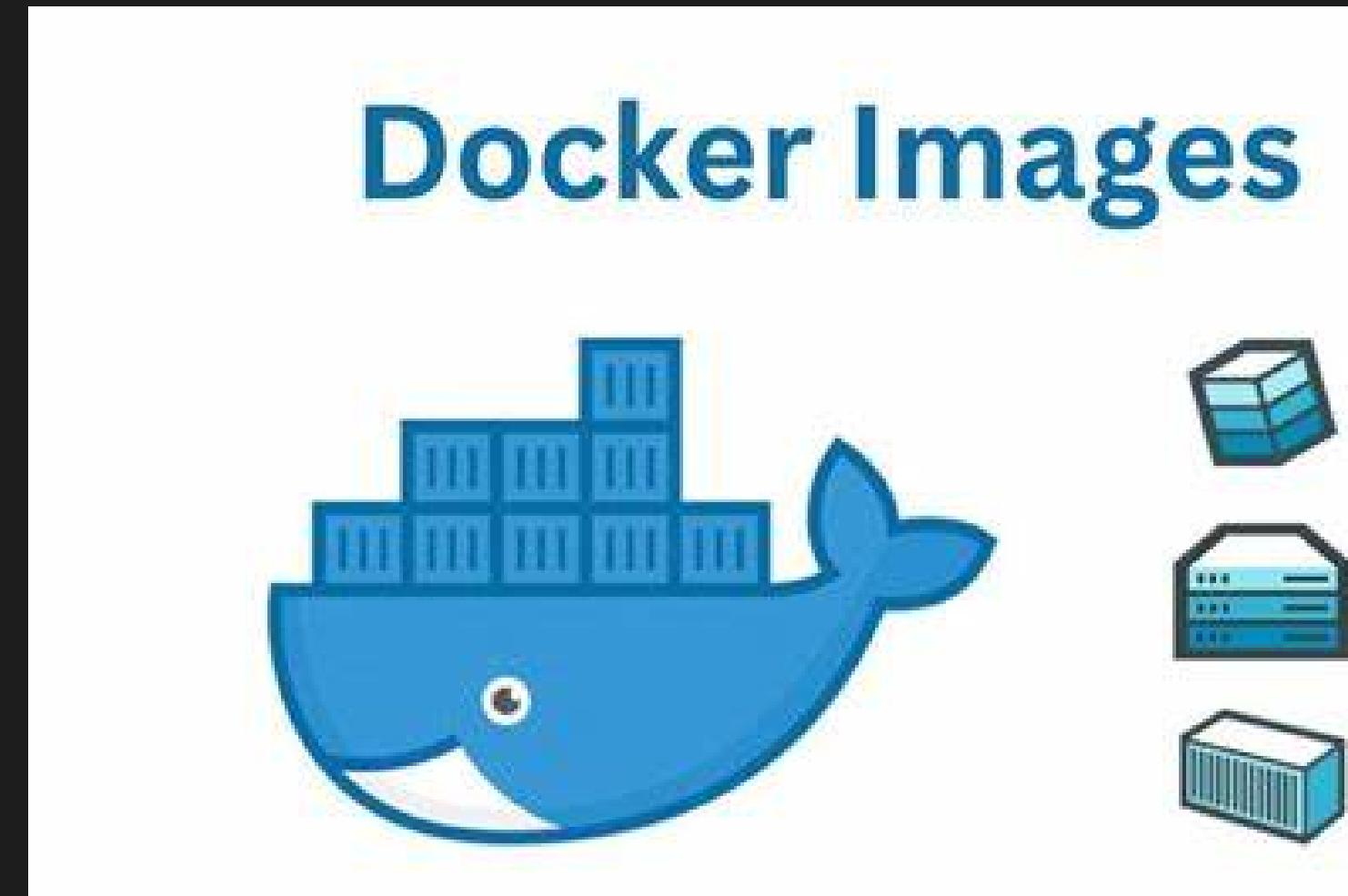
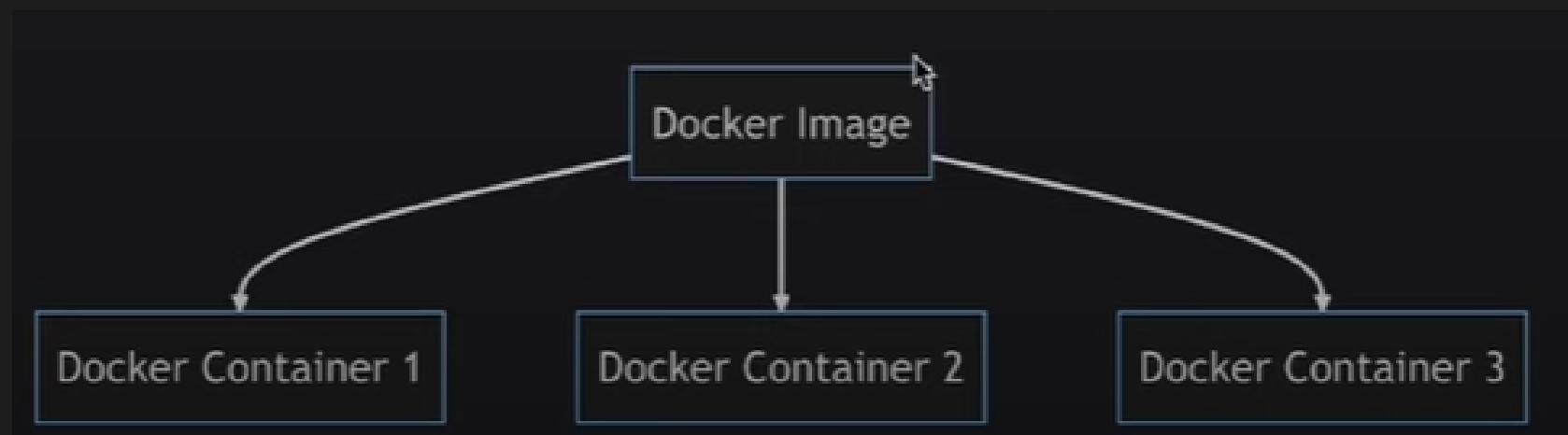
2 Docker Container

DOCKER  
IMAGES &  
CONTAINERS



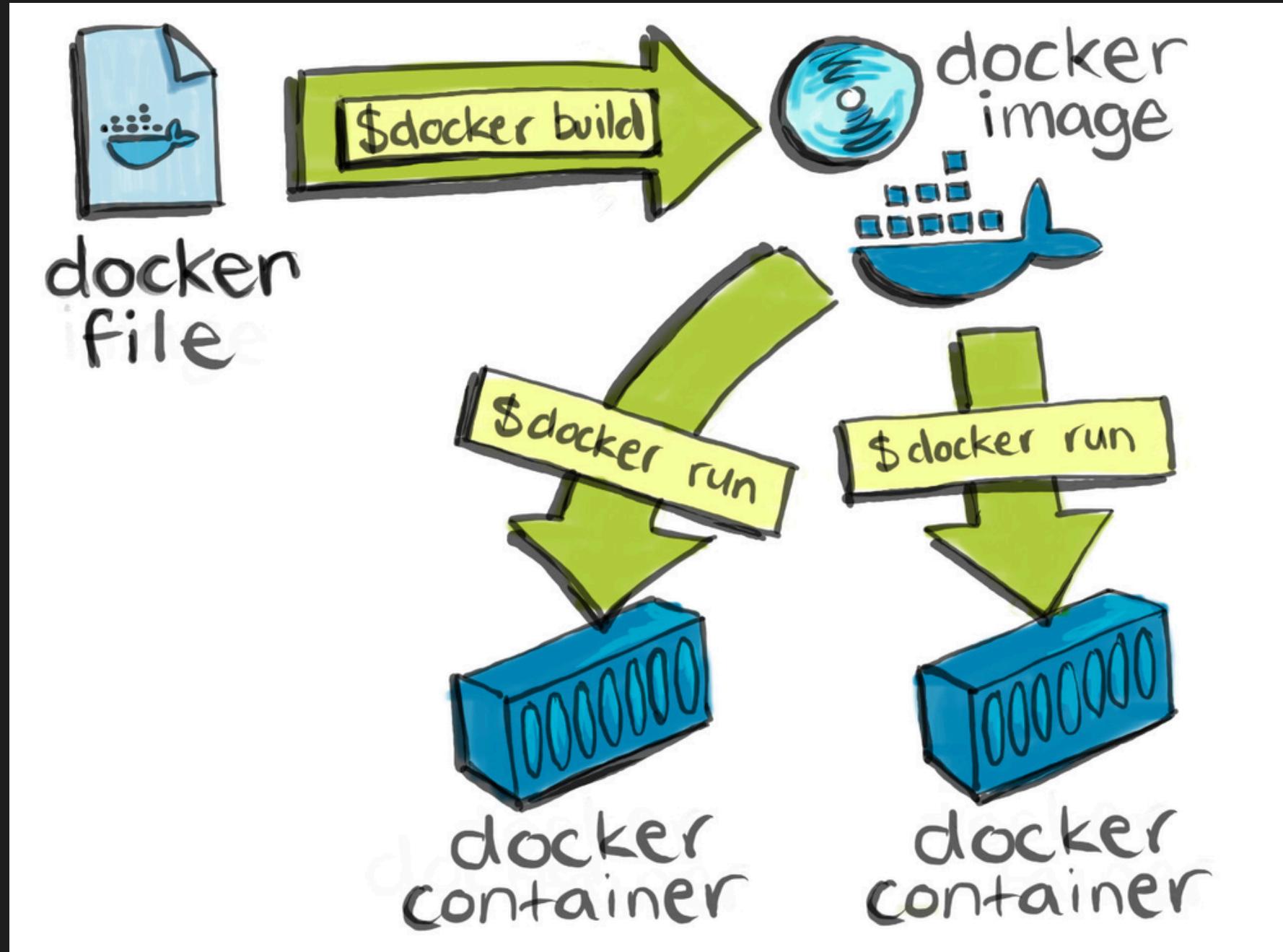
# what is docker images?

Docker images are read-only templates that contain a set of instructions for creating a container that can run applications. They encapsulate everything needed to run an application: code, runtime, libraries, environment variables, and configuration files.



# What is docker container?

Docker containers are runnable instances of Docker images. They encapsulate an application and its dependencies, providing a consistent environment across different stages of development, testing, and production.



# Docker helps solve problems

It works on my machine  
but crashes on Production



# Advantages of Docker

1. Easy to move apps to run on different machines.
2. Suitable for making apps that can SCL to support a large number of users.
3. Container will be light.
4. Container allows each app to work independently.
5. Reduce redundant work

# Disadvantages of docker

1. Security Vulnerabilities
2. Performance Overhead
3. Complex Networking
4. Data Persistence Challenges
5. Learning Curve and Complexity



# Installation

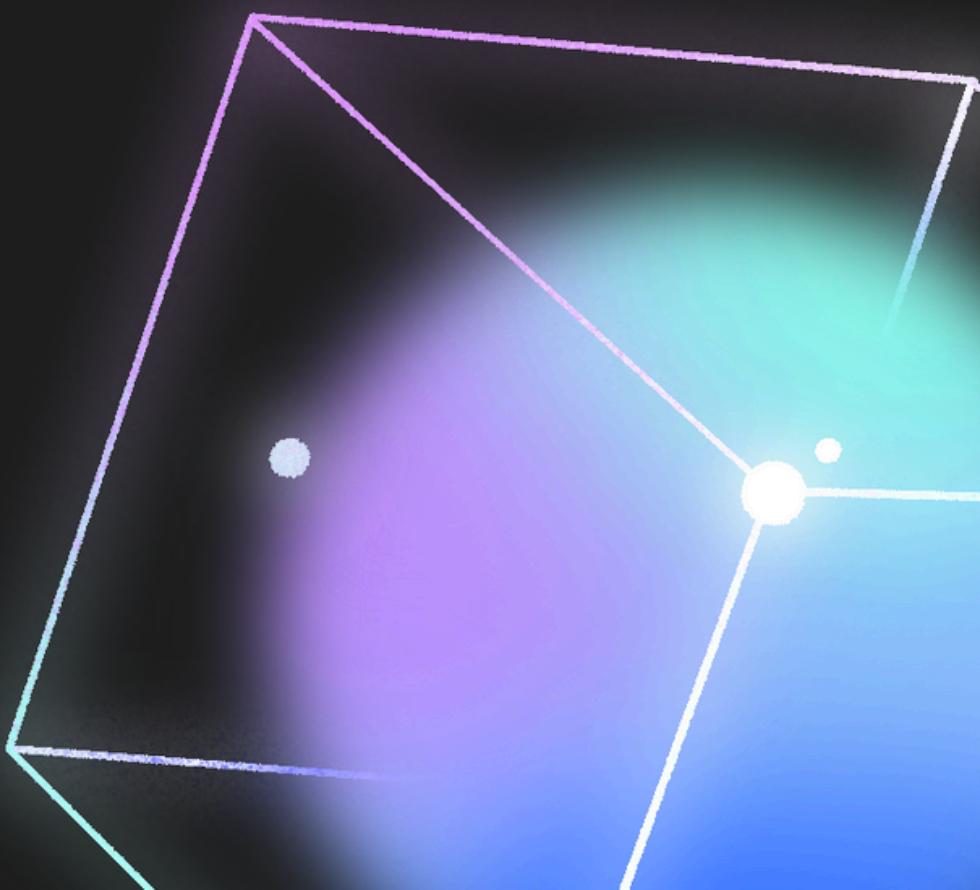
**step1:**Download Docker desktop from the website  
:<https://www.docker.com/>

**step2:**Run the installer and follow the on-screen  
instructions.



**Supported operating systems:** Windows, macOS, and Linux.

**Hardware requirements:** Modern 64-bit processors and at least 4GB of RAM.



## Check docker when download success

```
C:\Users\ASUS>docker -v  
Docker version 26.1.4, build 5650f9b
```

```
C:\Users\ASUS>|
```



# Run the following command to pull and run the "hello-world" image

```
C:\Users\ASUS>docker run hello-world
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

```
https://hub.docker.com/
```

For more examples and ideas, visit:

```
https://docs.docker.com/get-started/
```

Search

Only show running containers

<input type="checkbox"/>	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<a href="#">nostalgic_wiles</a> ddac0e8becae	<a href="#">hello-world</a>	Exited		N/A	2 minutes ago	 

# 1. create folder

```
C:\Users\ASUS>mkdir my-node-app
```

# 2. cd to my-node-app

```
C:\Users\ASUS\my-node-app>|
```

# 3. open vs code

# 4. create package.json

```
PS C:\Users\ASUS\my-node-app> npm init -y
>>
Wrote to C:\Users\ASUS\my-node-app\package.json:

{
  "name": "my-node-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

# 5. install express

```
npm install express
```

```
● >> C:\Users\ASUS\my-node-app>
```

```
added 64 packages, and audited 65 packages in 5s
```

```
12 packages are looking for funding  
run `npm fund` for details
```

```
found 0 vulnerabilities
```

```
○ PS C:\Users\ASUS\my-node-app> █
```

## 6. create file app.js

```
● ● ●  
1 const express = require('express');  
2 const app = express();  
3 const port = 3000;  
4  
5 app.get('/', (req, res) => {  
6   res.send('Hello, World!');  
7 });  
8  
9 app.listen(port, () => {  
10   console.log(`App running at http://localhost:${port}`);  
11 });
```

# 6. create Dockerfile

```
● ● ●  
1 FROM node:14  
2 WORKDIR /app  
3 COPY package*.json ./  
4 RUN npm install  
5 COPY . .  
6 EXPOSE 3000  
7 CMD ["node", "app.js"]
```



7. use command `docker build -t my-node-app .`

```
[+] Building 125.9s (4/9)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 150B
=> [internal] load metadata for docker.io/library/node:14
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa
=> => resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa
=> => sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa 776B / 776B
=> => sha256:2cafa3fbb0b6529ee4726b4f599ec27ee557ea3dea7019182323b3779959927f 2.21kB / 2.21kB
=> => sha256:1d12470fa662a2a5cb50378dc8ea228c1735747db410bbefb8e2d9144b5452 7.51kB / 7.51kB
=> => sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845ebc747976c408 23.07MB / 50.45MB
=> => sha256:b253aeafeaa7e0671bb60008df01de101a38a045ff7bc656e3b0fbfc7c05cca5 7.86MB / 7.86MB
=> => sha256:3d2201bd995cccf12851a50820de03d34a17011dcbb9ac9fdf3a50c952ccb131 10.00MB / 10.00MB
=> => sha256:1de76e268b103d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f5f9ee2 14.68MB / 51.88MB
=> => sha256:d9a8df5894511ce28a05e2925a75e8a4acbd0634c39ad734fdfba8e23d1b1569 2.10MB / 191.85MB
=> [internal] load build context
=> => transferring context: 2.21MB
```

8. docker run -d -p 3000:3000 my-node-  
app

<http://localhost:3000>

Hello, World!

# Summary

Docker's containerization technology has significantly impacted software development, enabling developers to create, deploy, and run applications consistently across various environments.

Its lightweight, fast, and efficient nature makes it ideal for modern DevOps practices and cloud-native applications.

With a robust ecosystem and active community, Docker continues to innovate and support the evolving needs of developers and enterprises worldwide.

Thank you

