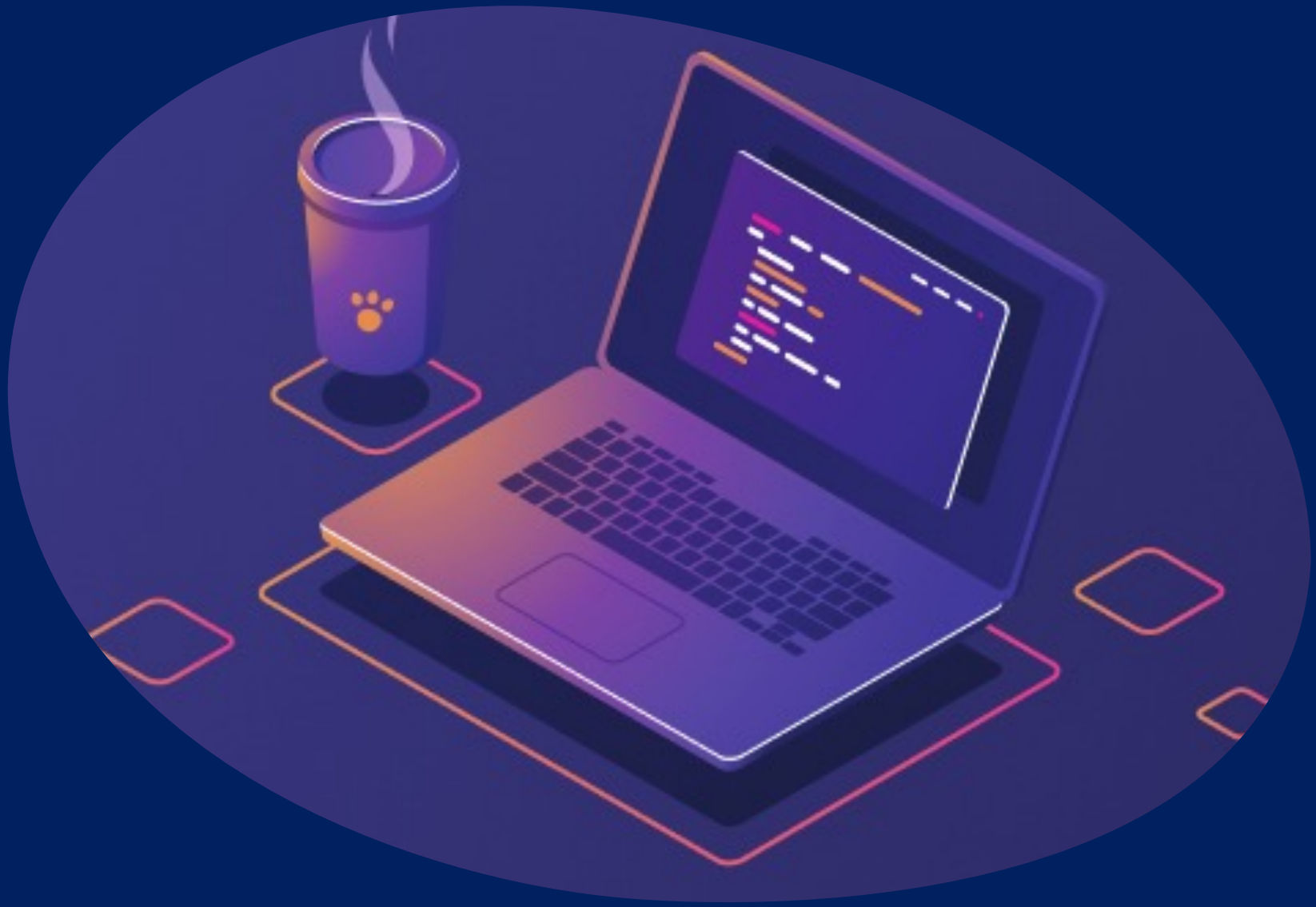
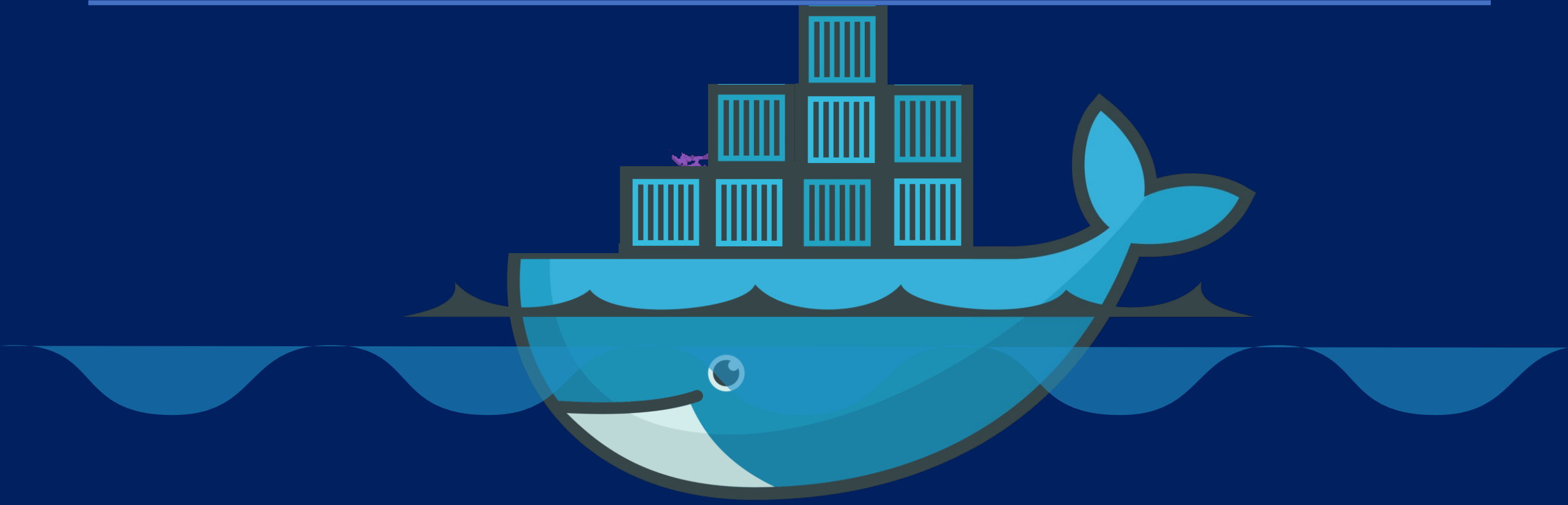


Week 8 : SOFTWARE DEVELOPMENT TOOLS AND ENVIRONMENTS

Docker Overview



Why do you need docker?

Web Server

Database

Messaging

Orchestration

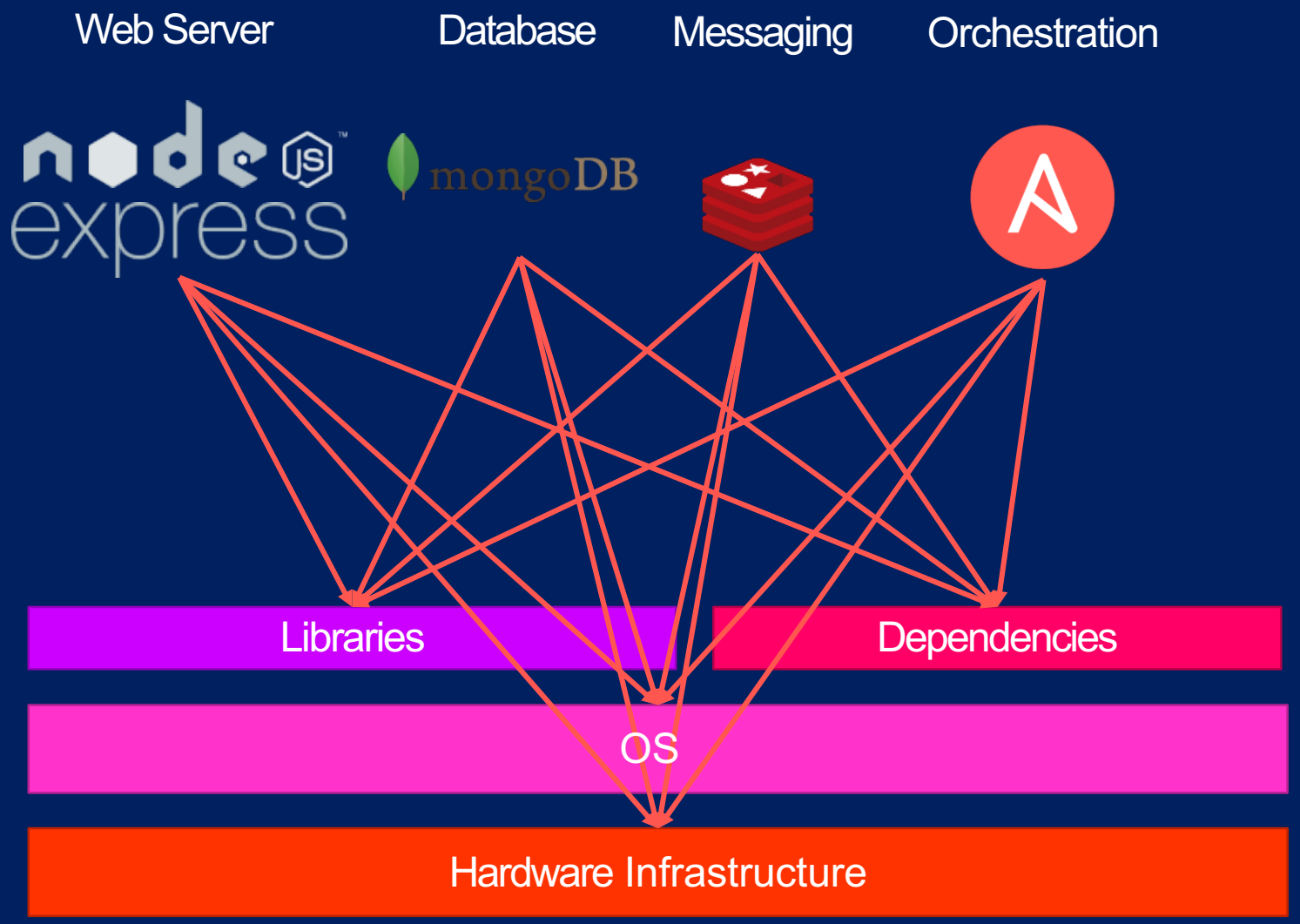
Libraries

Dependencies

OS

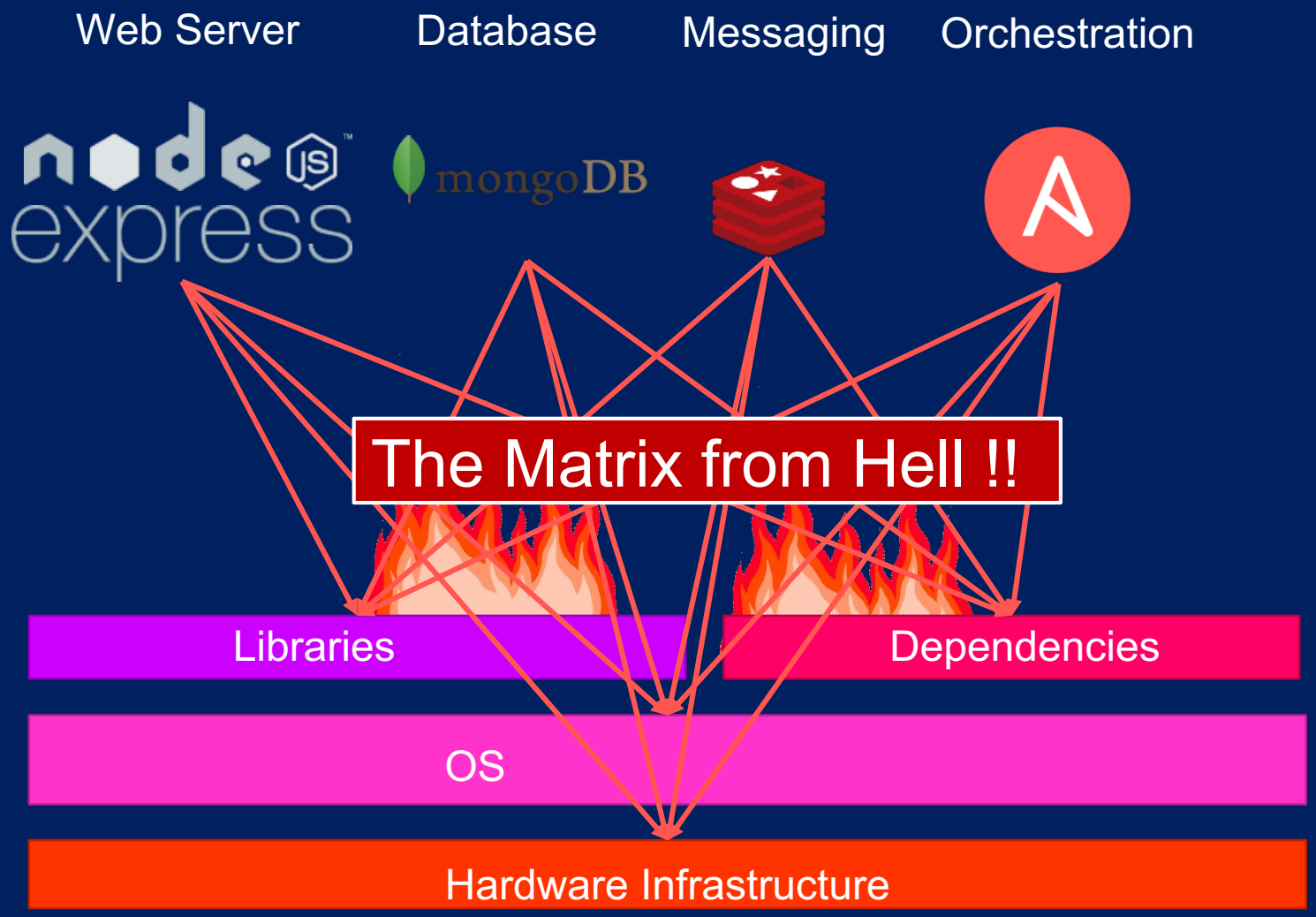
Hardware Infrastructure

Why do you need docker?



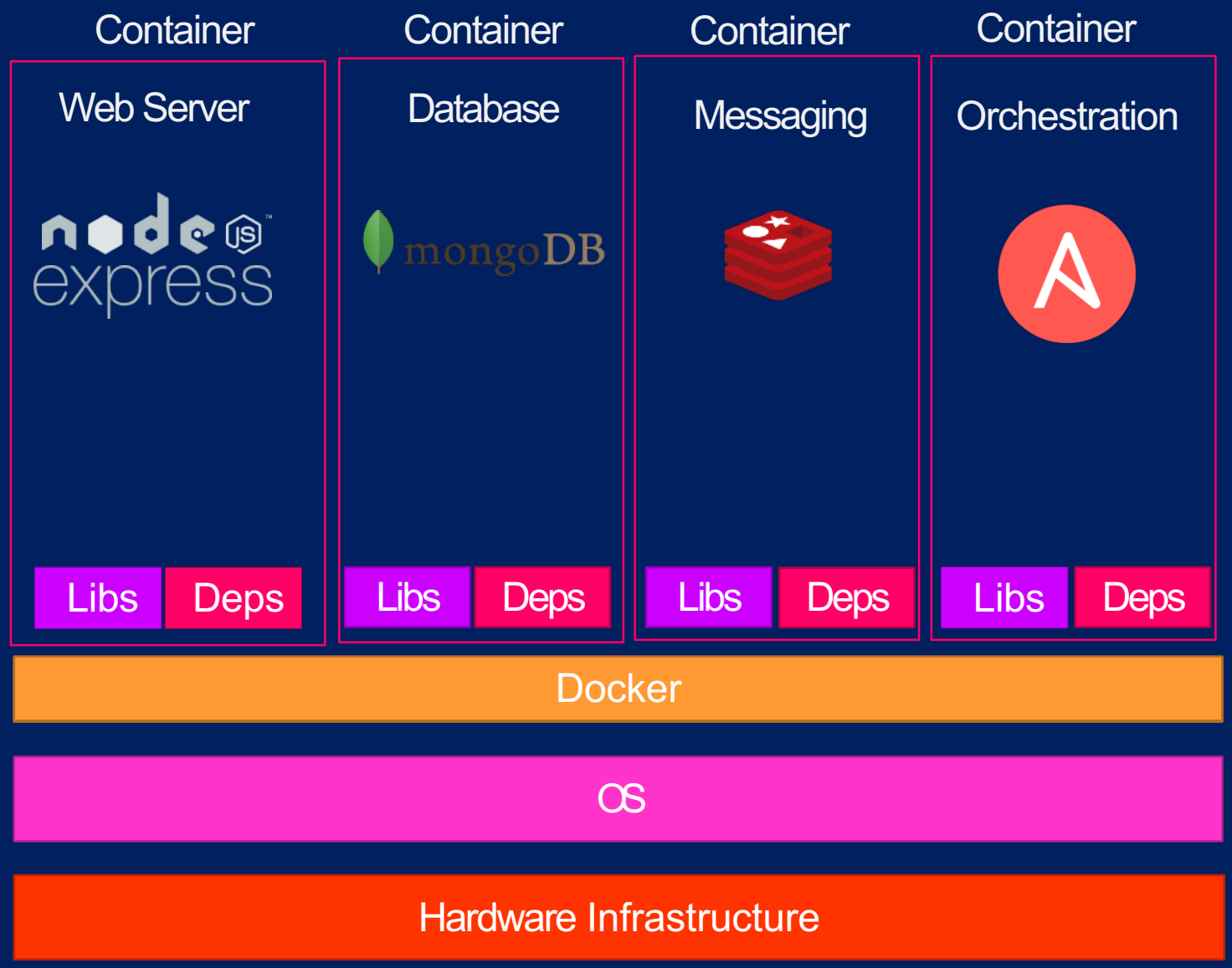
Why do you need docker?

- Compatibility/Dependency
- Long setup time
- Different Dev/Test/Prod environments

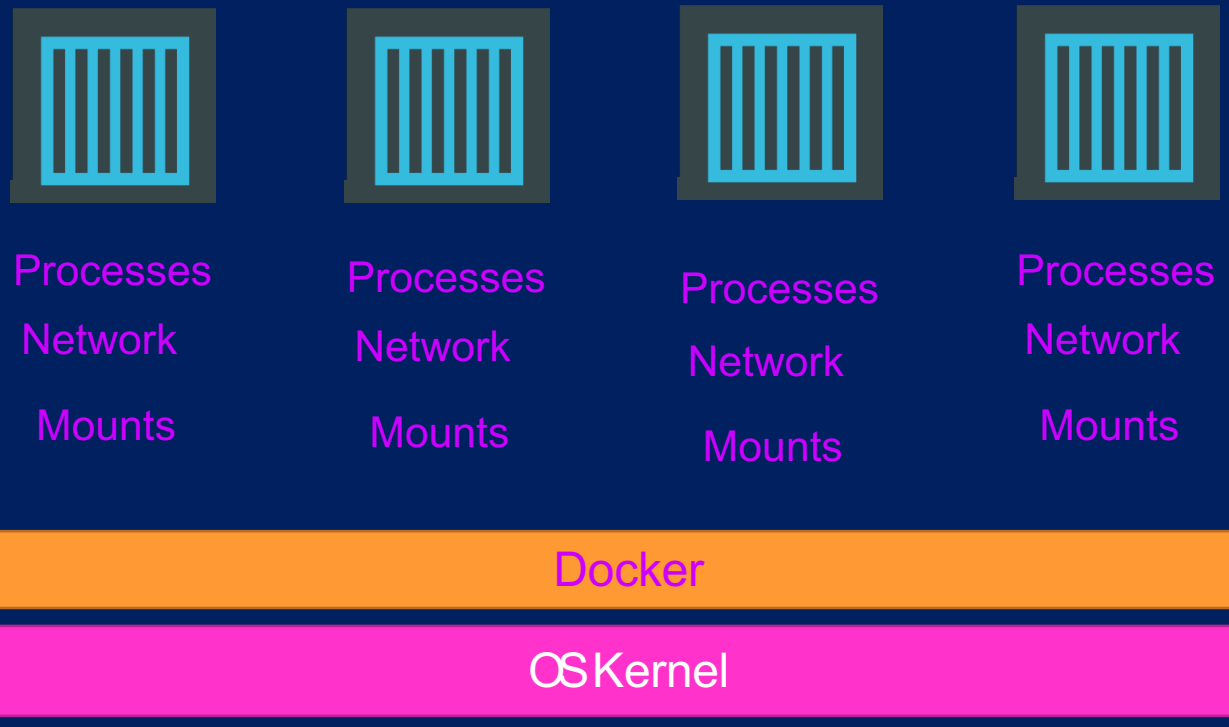


What can it do?

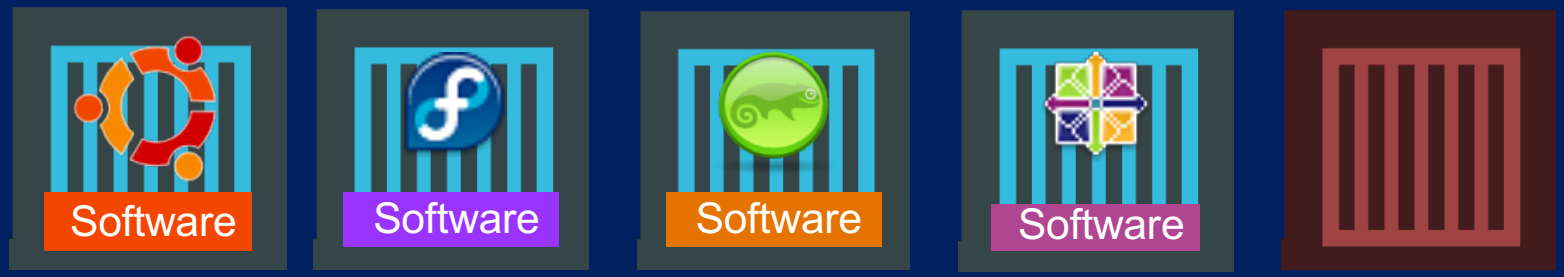
- Containerize Applications
- Run each service with its own dependencies in separate containers



What are containers?



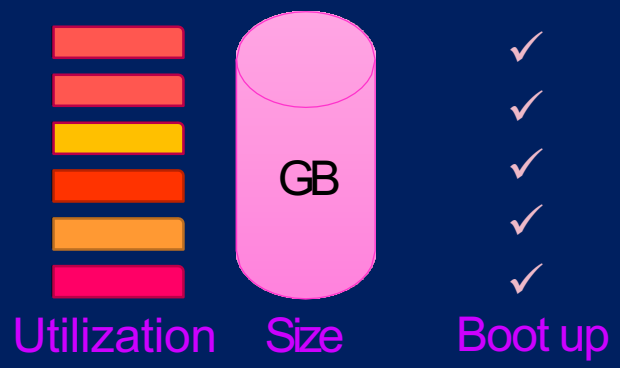
Sharing the kernel for Operating System

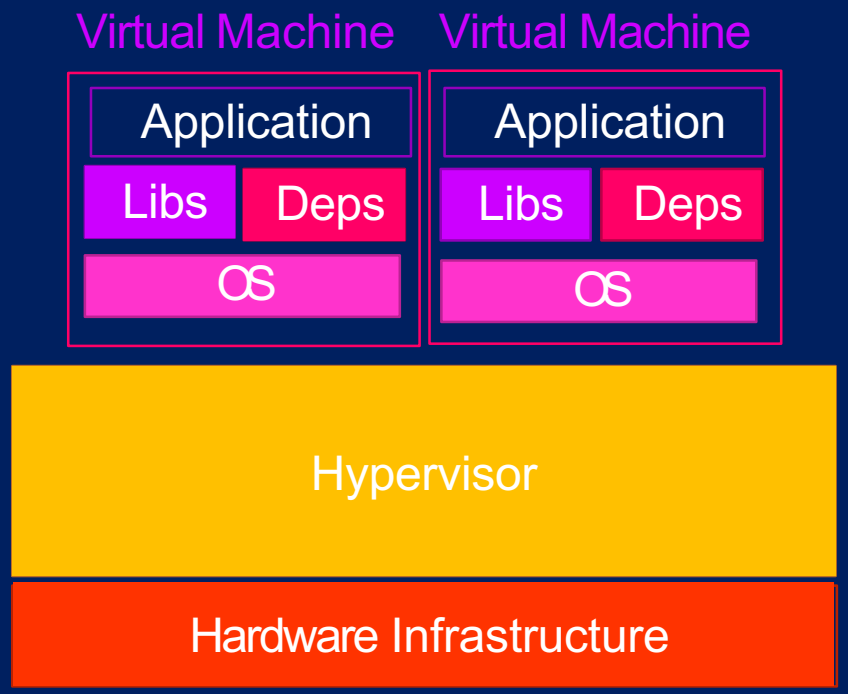


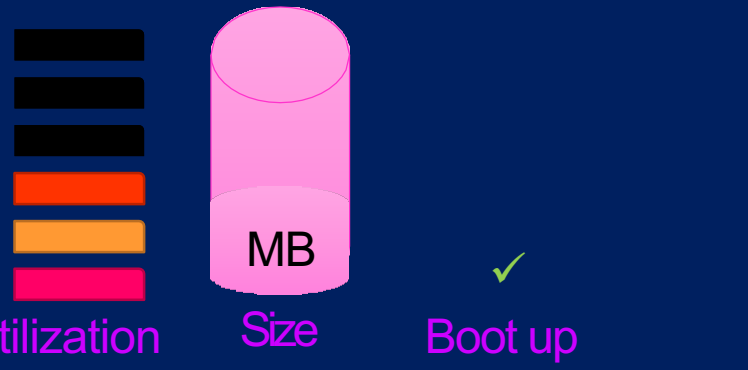
Docker

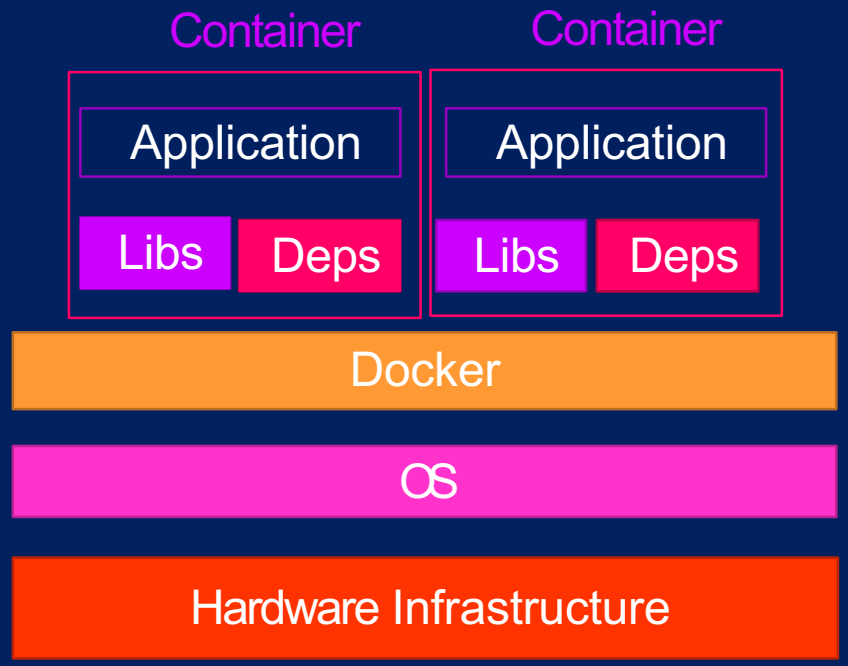
OS - Ubuntu

Containers vs Virtual Machines

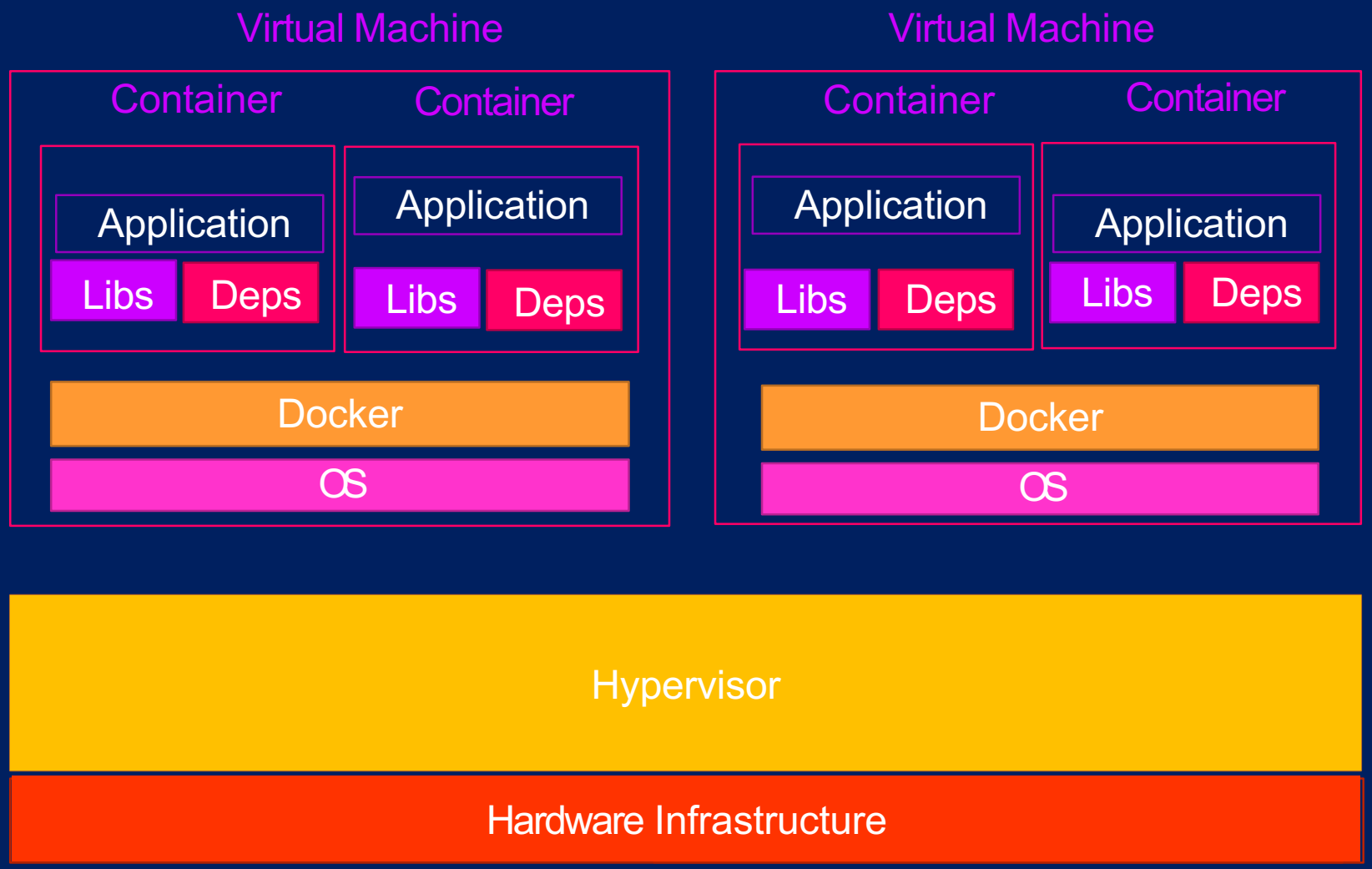






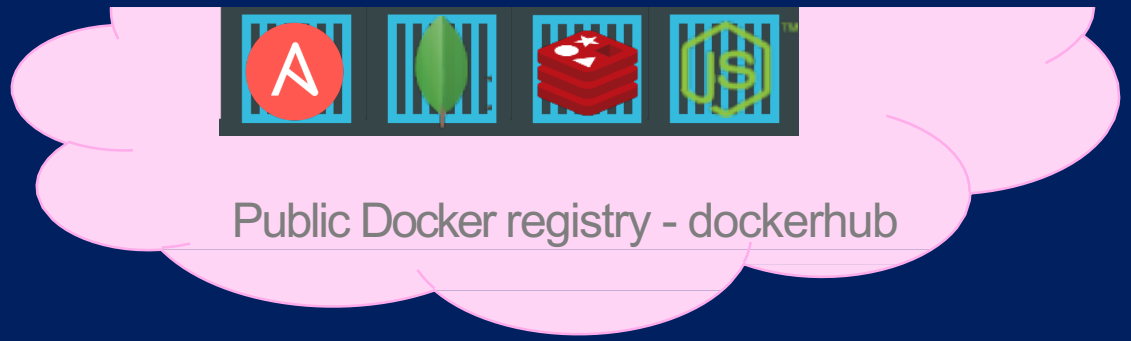


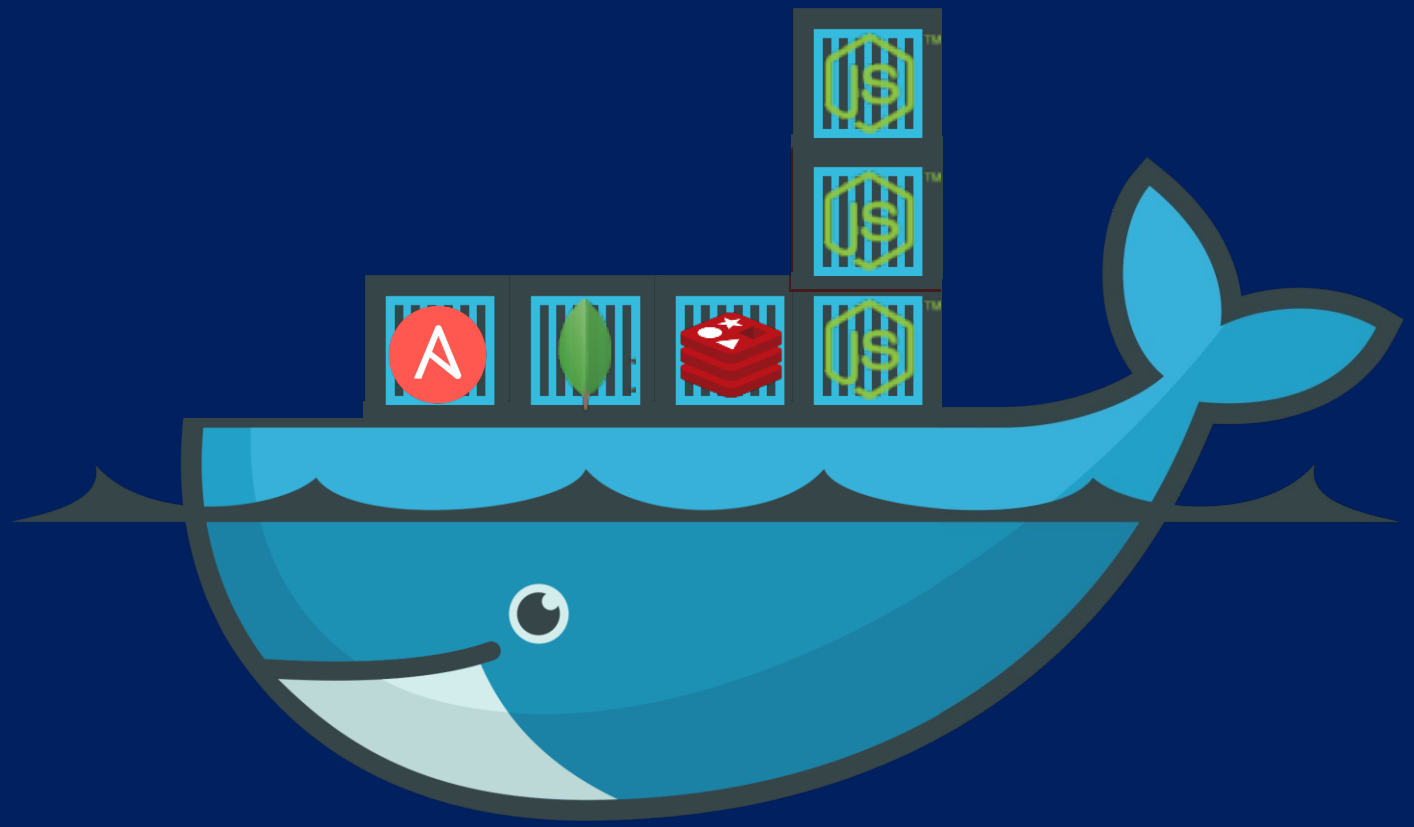
Containers & Virtual Machines



How is it done?

```
docker run ansible
docker run mongodb
docker run redis
docker run nodejs
docker run nodejs
docker run nodejs
```





Container vs image

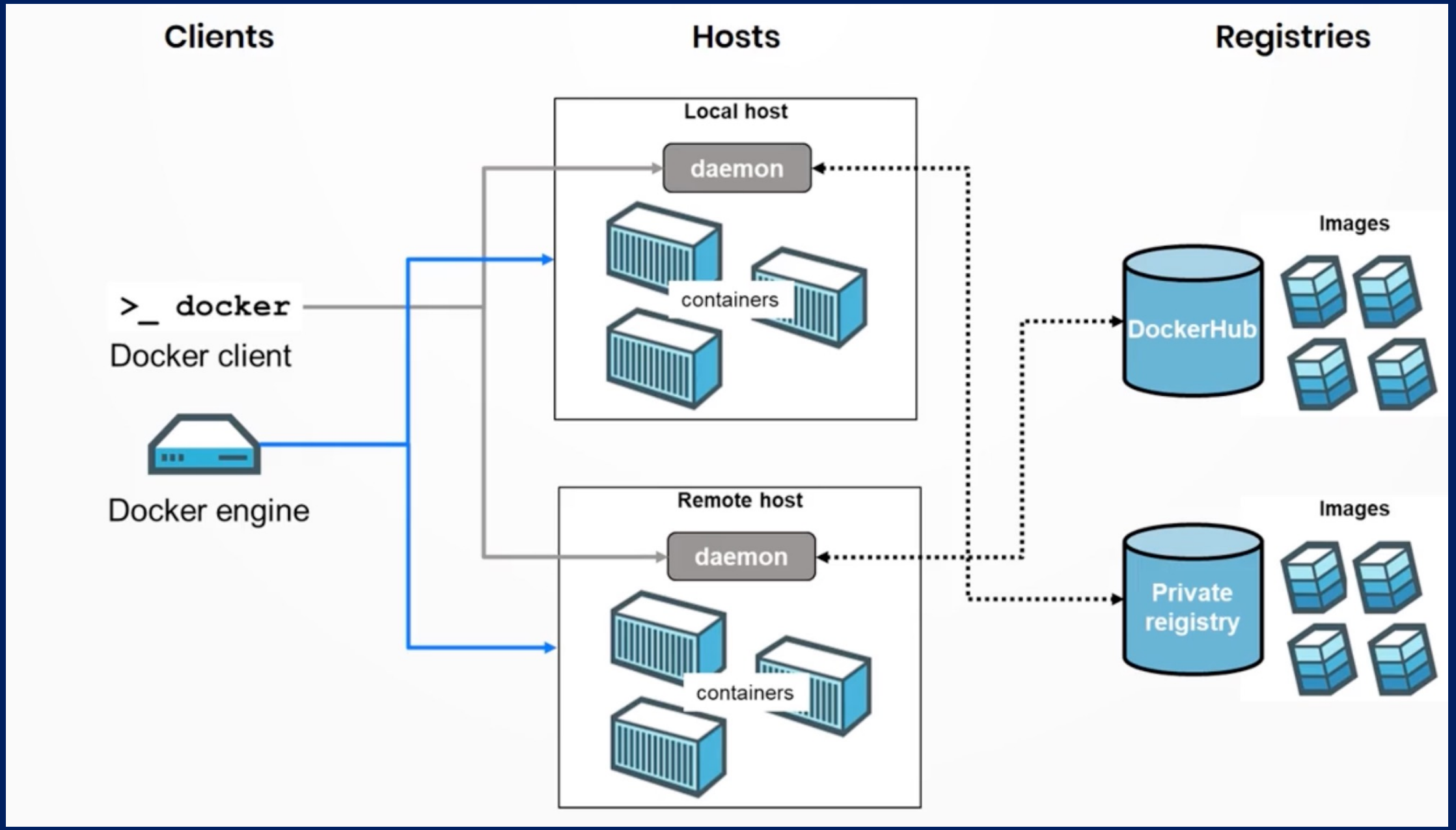
Docker Image

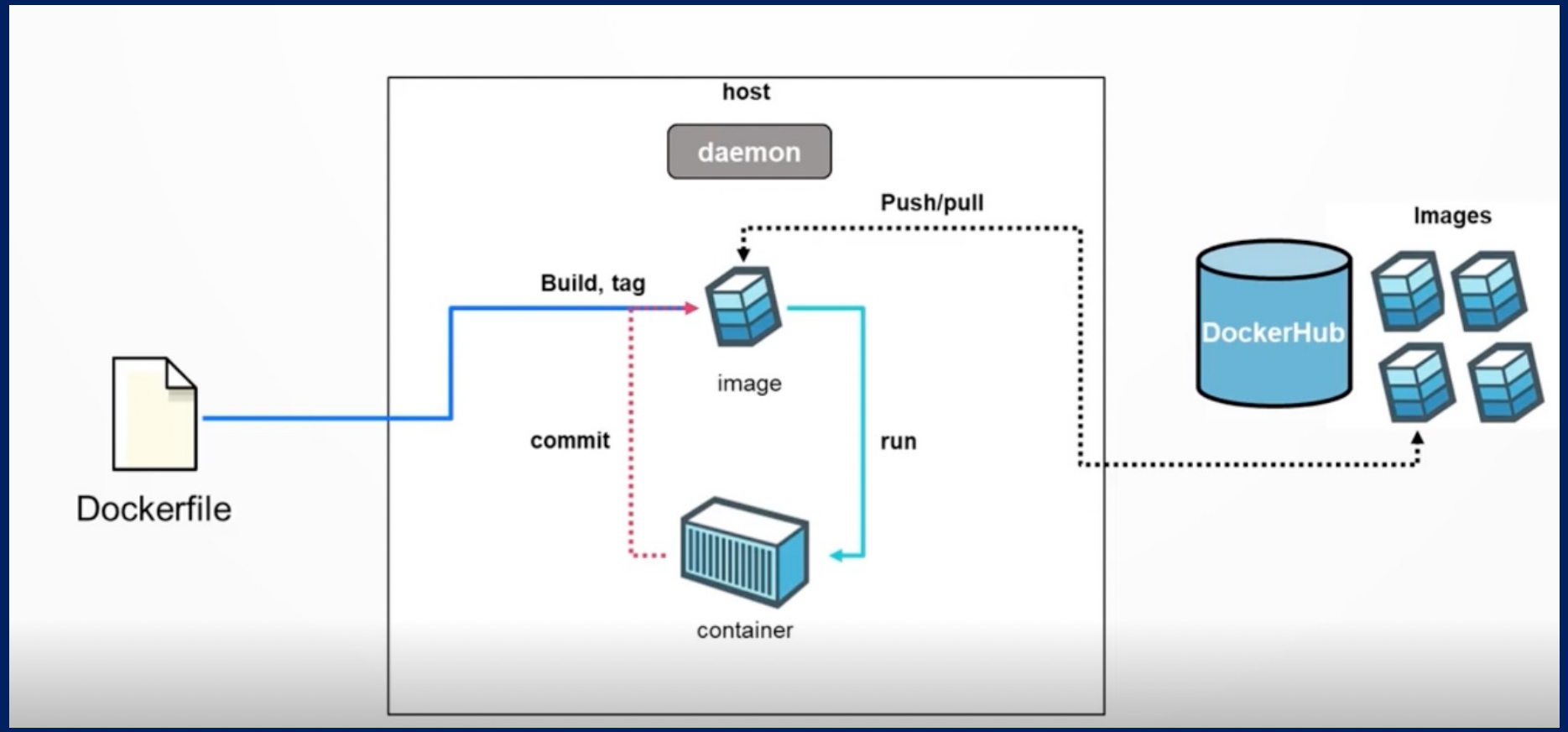
Package
Template Plan

Docker Container #1

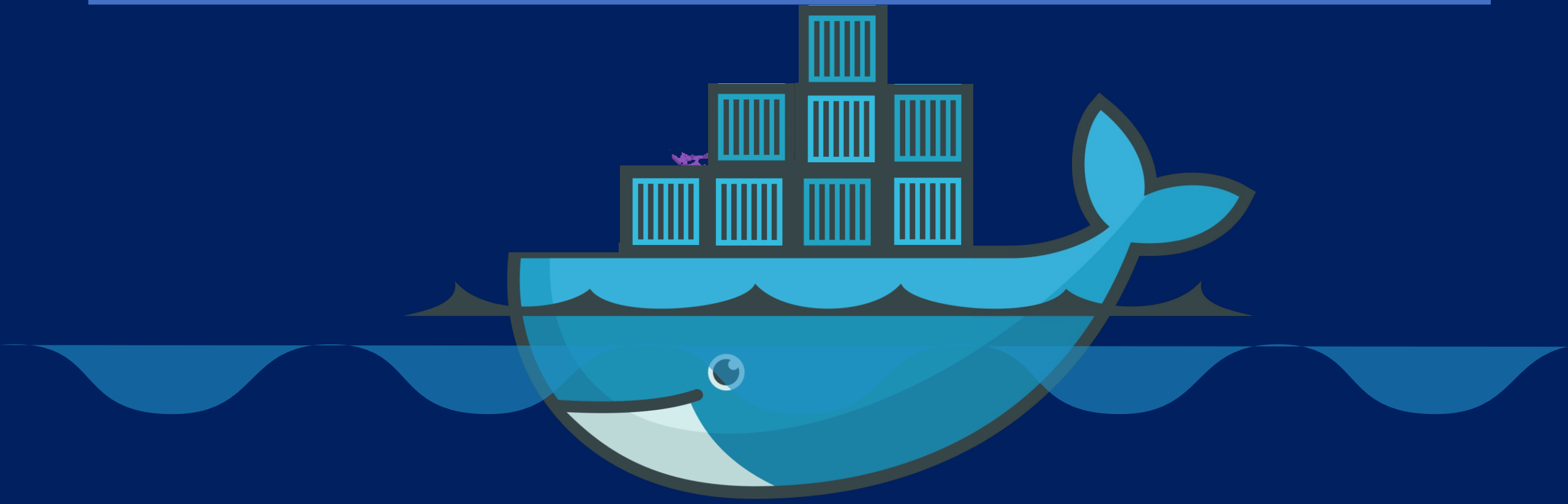
Docker Container #2

Docker Container #3


Clients
Hosts
Registries
Local host
daemon
containers
>_ docker
Docker client
Docker engine
Remote host
daemon
containers
Images
DockerHub
Images
Private reigistry


host
daemon
Push/pull
Images
Build, tag
DockerHub
image
commit
run
Dockerfile
container

Install Docker



Install Docker Engine on Ubuntu

To get started with Docker Engine on Ubuntu, make sure you meet the prerequisites, then install Docker.

Prerequisites

OS requirements

To install Docker Engine, you need the 64-bit version of one of these Ubuntu versions:

- Ubuntu Kinetic 22.10
- Ubuntu Jammy 22.04 (LTS)
- Ubuntu Focal 20.04 (LTS)
- Ubuntu Bionic 18.04 (LTS)

Docker Engine is compatible with `x86_64` (or `amd64`), `armhf`, `arm64`, and `s390x` architectures.

Uninstall old versions

Older versions of Docker went by the names of `docker`, `docker.io`, or `docker-engine`. Uninstall any such older versions before attempting to install a new version:

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
```

It's OK if `apt-get` reports that none of these packages are installed.

Images, containers, volumes, and networks stored in `/var/lib/docker/` aren't automatically removed when you uninstall Docker. If you want to start with a clean installation, and prefer to clean up any existing data, refer to the uninstall Docker Engine section.

Installation methods

You can install Docker Engine in different ways, depending on your needs:

- Docker Engine comes bundled with Docker Desktop for Linux. This is the easiest and quickest way to get started.
- You can also set up and install Docker Engine from Docker's `apt` repository.
- Install it manually and manage upgrades manually.
- Using a convenience scripts. Only recommended for testing and development environments.

https://docs.docker.com/engine/install/ubuntu/

How to install Docker on Amazon Linux 2

Author: Vivek Gite • Last updated: January 3, 2023 • 17 comments

How do I install docker and docker-compose using the yum command on Amazon Linux 2 running on the EC2 or Lightsail cloud instance?

This page explains how to install and test Docker on Amazon Linux 2 over ssh based session.

Tutorial details	
Difficulty level	Easy
Root privileges	Yes
Requirements	Linux terminal
Category	Package Manager
Prerequisites	yum command
OS compatibility	Amazon Linux • Linux
Est. reading time	6 minutes

https://www.cyberciti.biz/faq/how-to-install-docker-on-amazon-linux-2/

Once the command runs successfully, consider adding the currently logged-in user to the docker group. This allows you to run docker without invoking sudo.

```
$ sudo usermod -aG docker $USER
```

```
$ newgrp docker
```

sudo usermod -aG docker $USER
newgrp docker

By default, Docker autostarts upon installation. To verify this, run the command:

```
$ sudo systemctl status docker
```

```
james@james-PC:~$
james@james-PC:~$ sudo systemctl status docker
[sudo] password for james:
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2022-03-31 21:29:43 EAT; 2h 16min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 1112 (dockerd)
      Tasks: 14
     Memory: 63.7M
        CPU: 4.164s
```

If, for any reason, Docker is not running, simply execute the following command:

```
$ sudo systemctl start docker
```

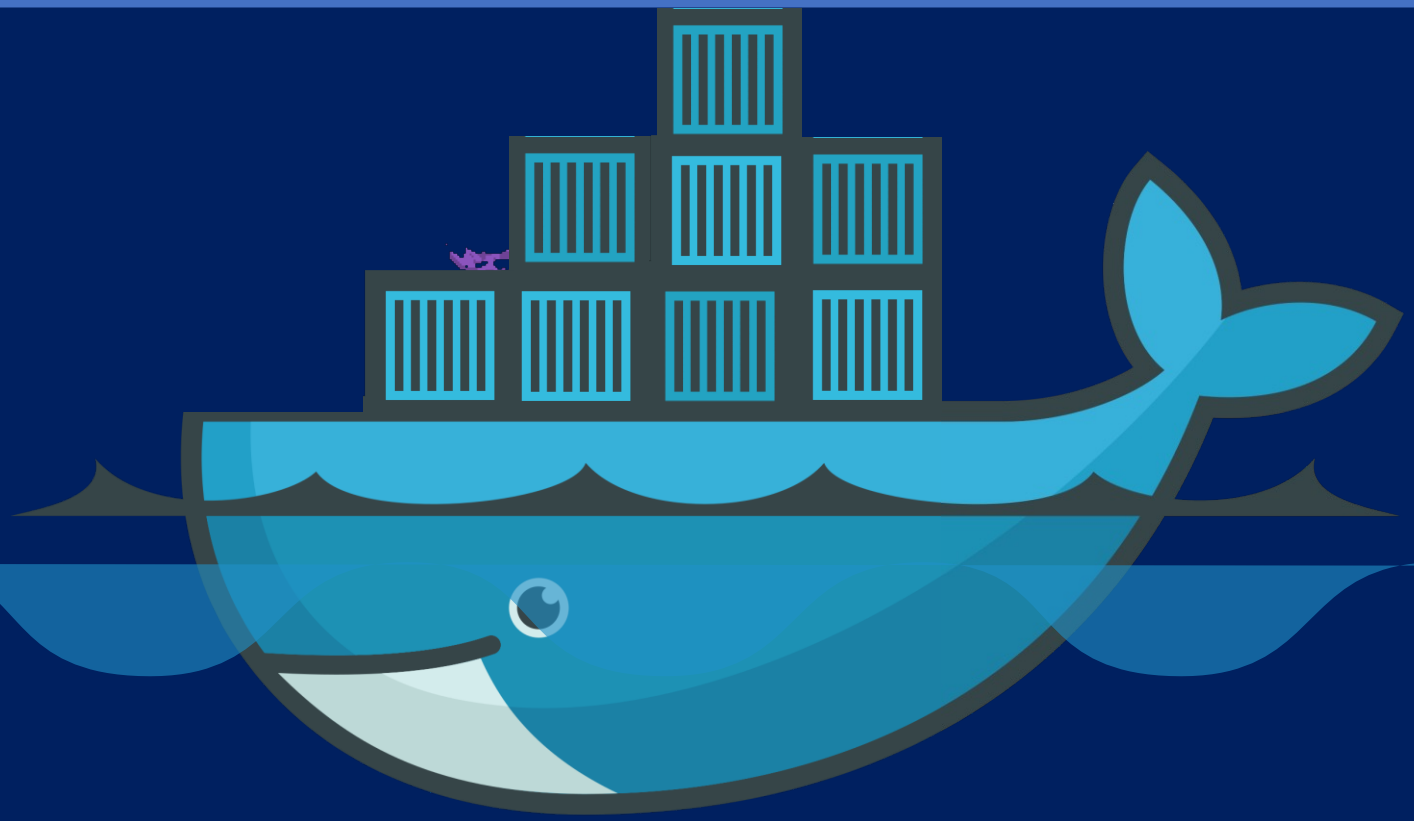
To enable Docker to start automatically every time on system startup, run the command:

```
$ sudo systemctl enable docker
```

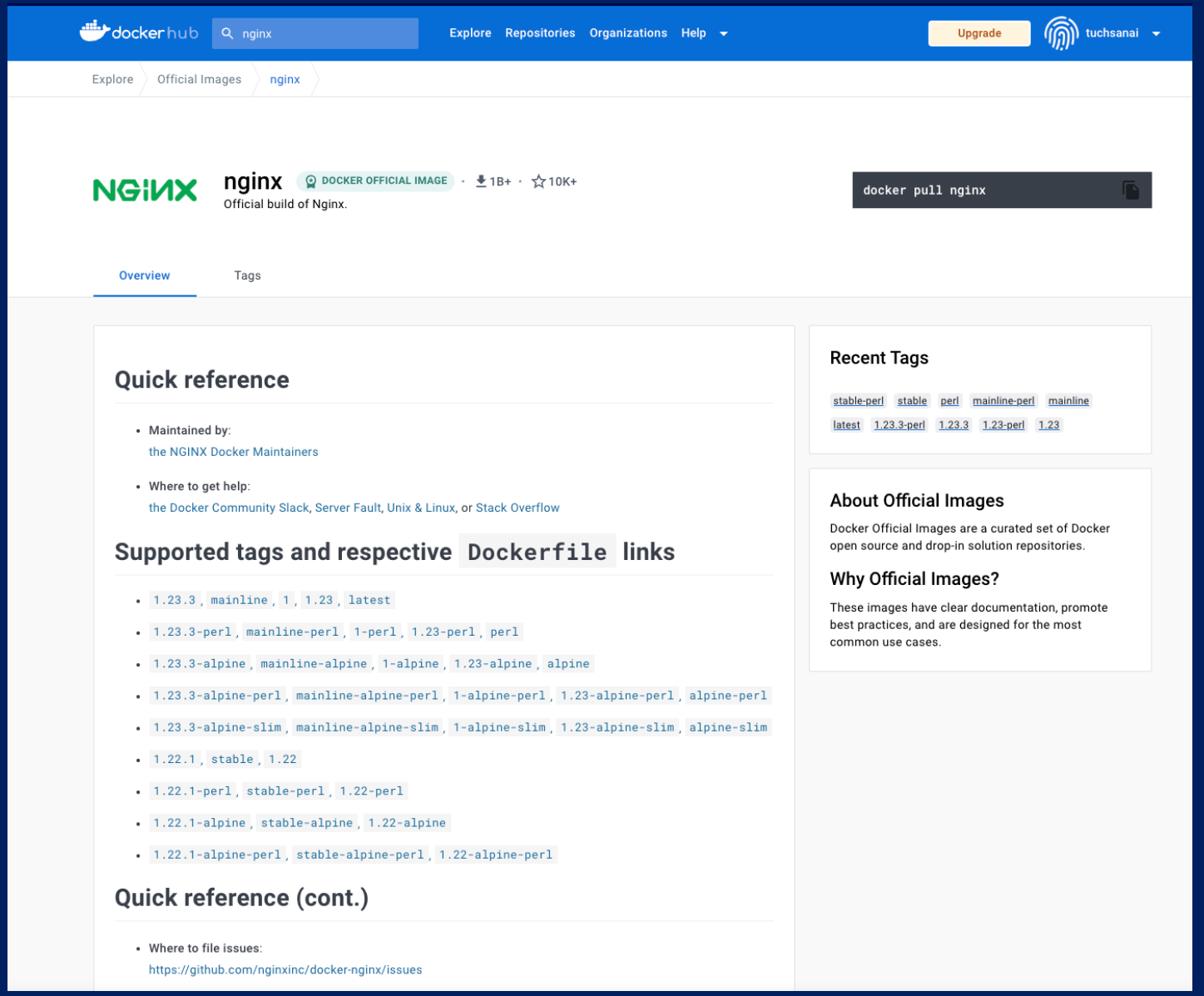
To restart Docker run:

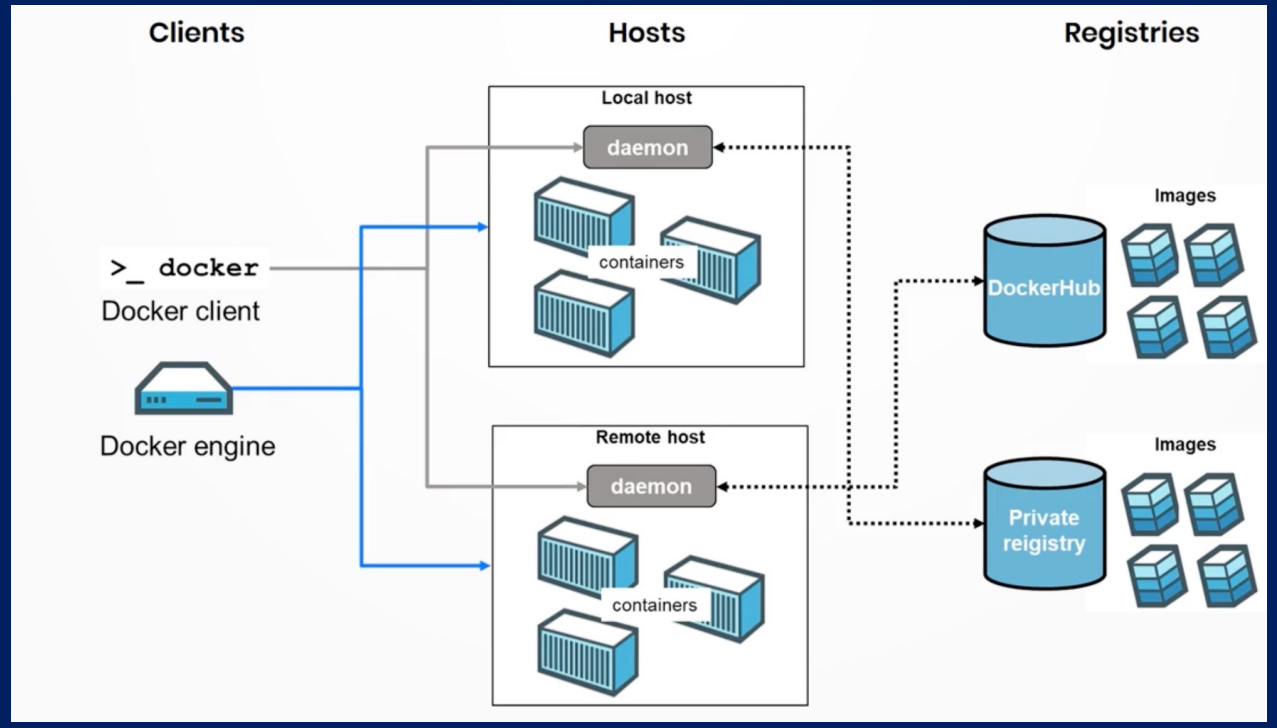
```
$ sudo systemctl restart docker
```

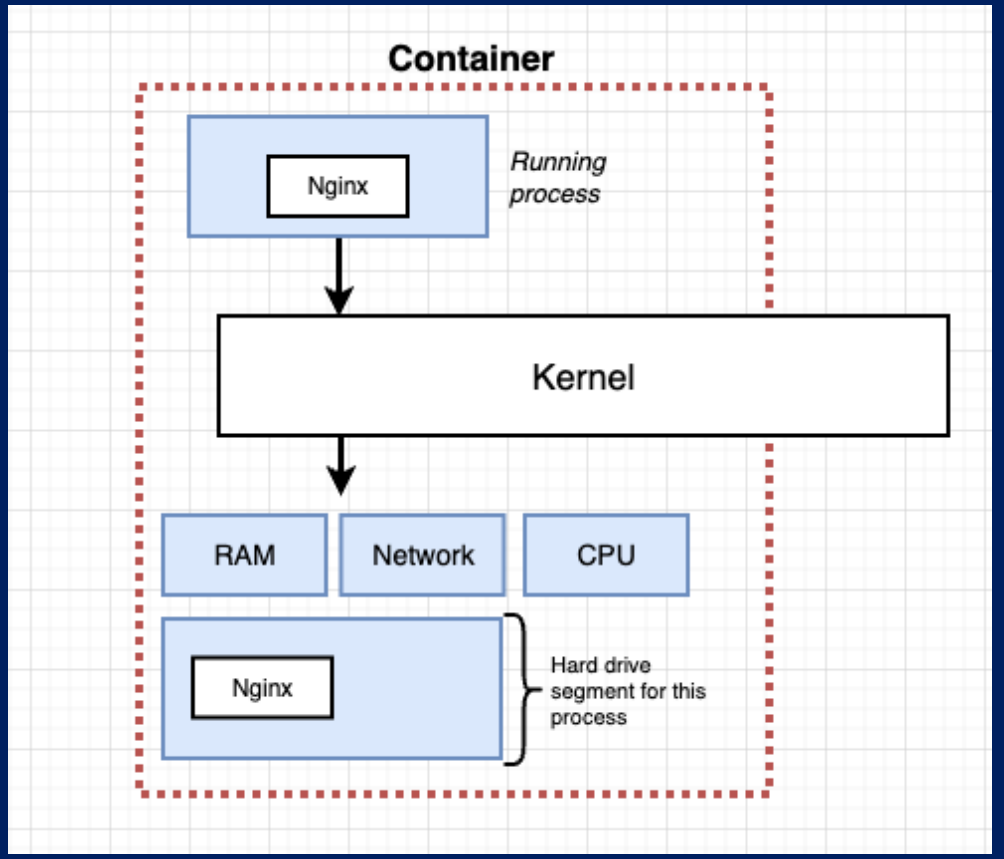
Docker Run



Docker Registry




Clients
Hosts
Registries
Local host
daemon
containers
>_ docker
Docker client
Docker engine
Remote host
daemon
containers
Images
DockerHub
Images
Private reigistry

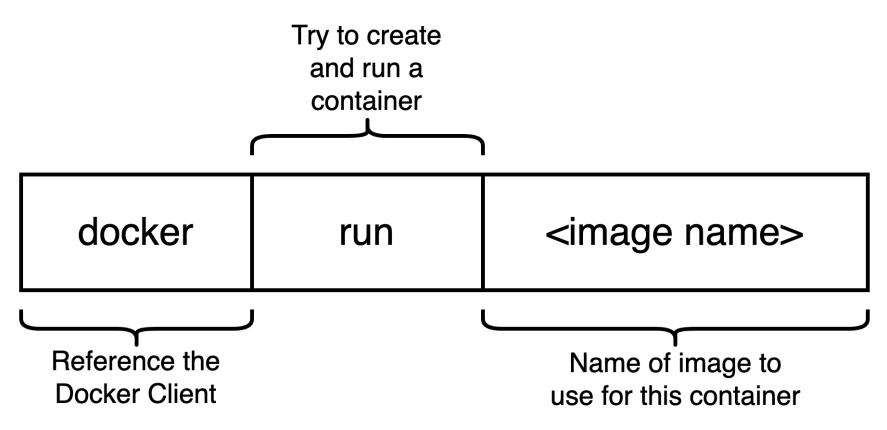

Container
Nginx
Running process
Kernel
RAM
Network
CPU
Nginx
Hard drive segment for this process

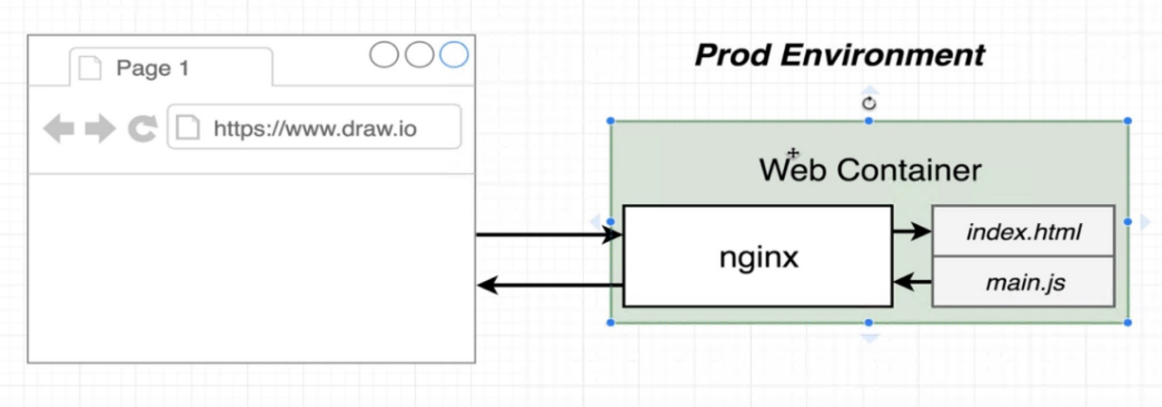
Run – start a container

```
docker run nginx
```

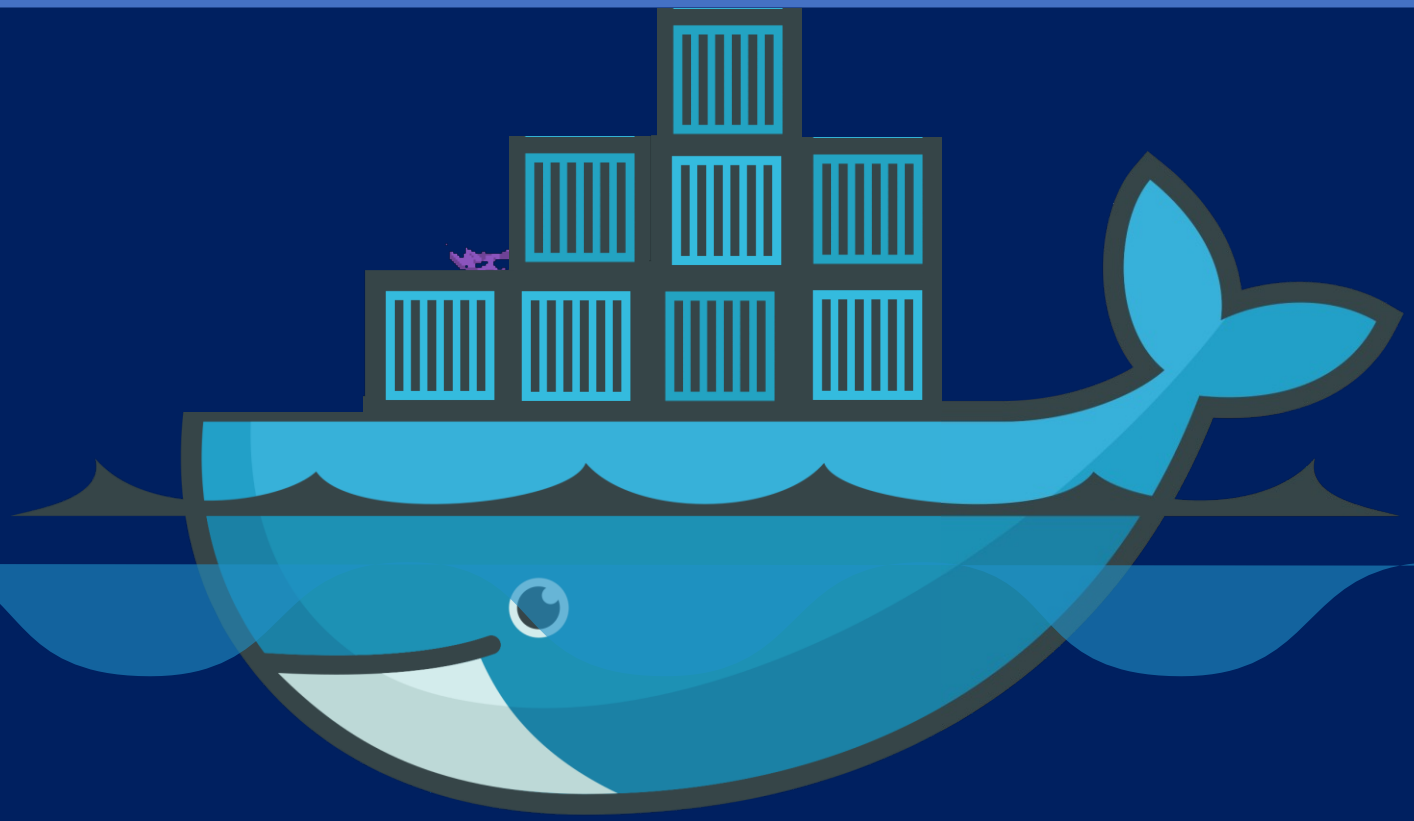
```
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
fc7181108d40: Already exists
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
```

```
Status: Downloaded newer image for nginx:latest
```





Docker Mapping

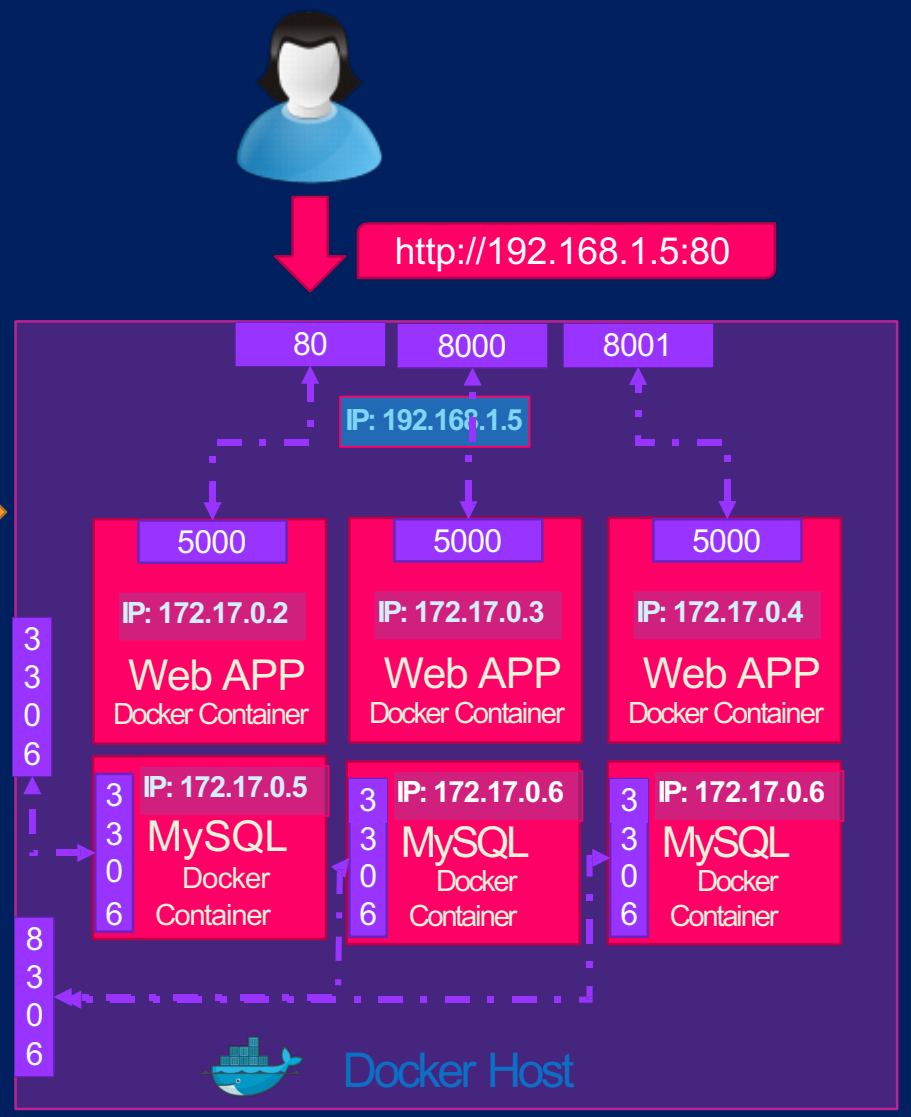


Run – PORT mapping

```
docker run myname/webapp
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

http://172.17.0.2:5000 — Internal IP

```
docker run -p 80:5000 myname/simple-webapp
docker run -p 8000:5000 myname/simple-webapp
docker run -p 8001:5000 myname/simple-webapp
docker run -p 3306:3306 mysql
docker run -p 8306:3306 mysql
docker run -p 8306:3306 mysql
```



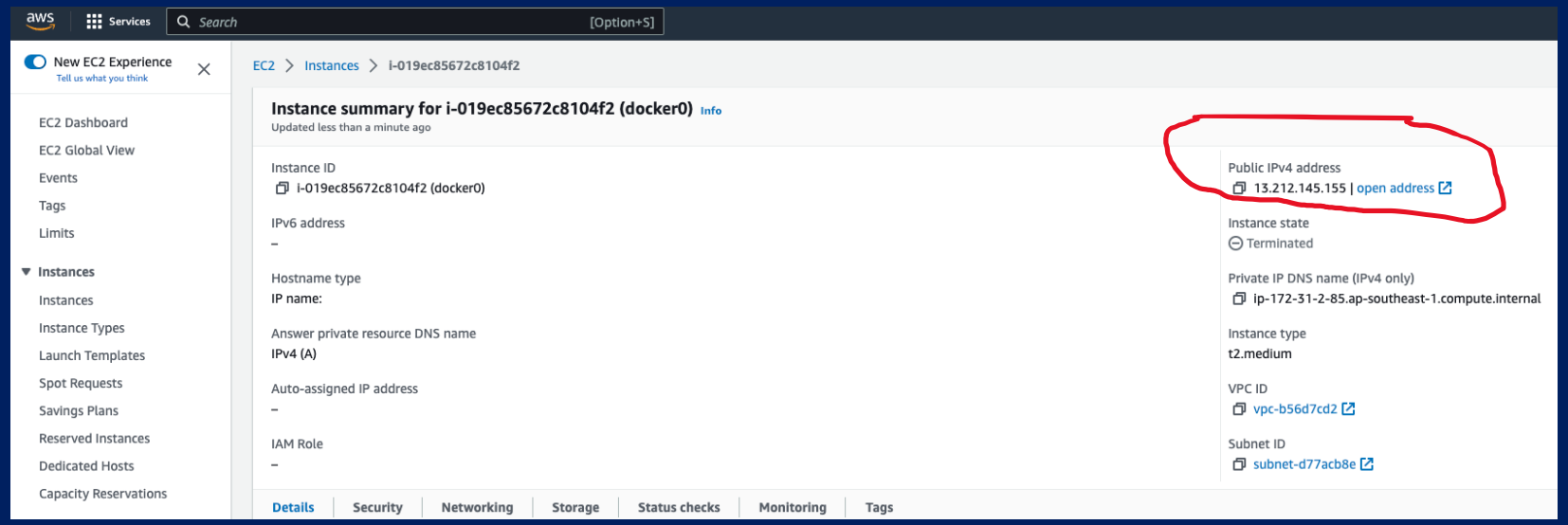

```
root@osboxes:/root # docker run -p 8306:3306 -e MYSQL_ROOT_PASSWORD=pass mysql
docker: Error response from daemon: driver failed programming external connectivity on endpoint boring_bhabha (
5079d342b7e8ee11c71d46): Bind for 0.0.0.0:8306 failed: port is already allocated.
```

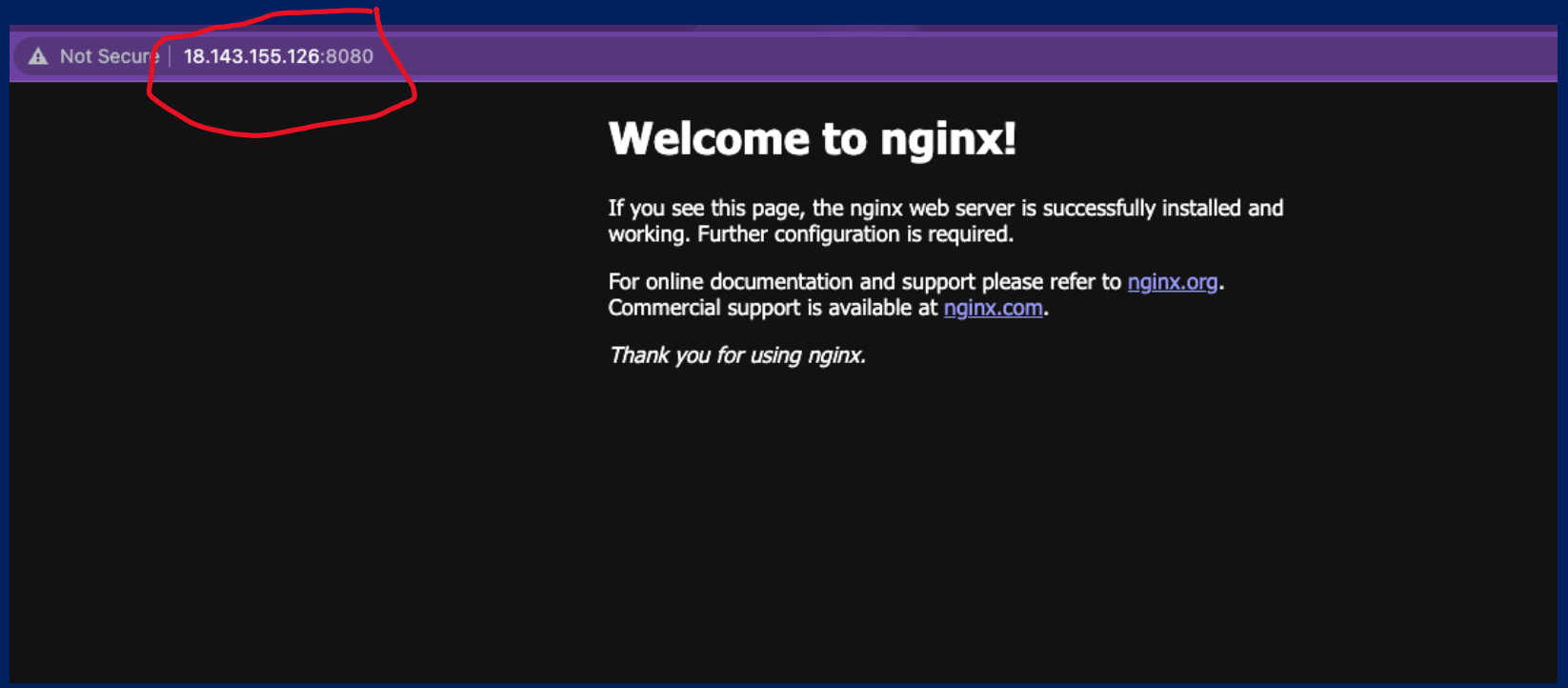
LAB 1 : Run Nginx with port mapping

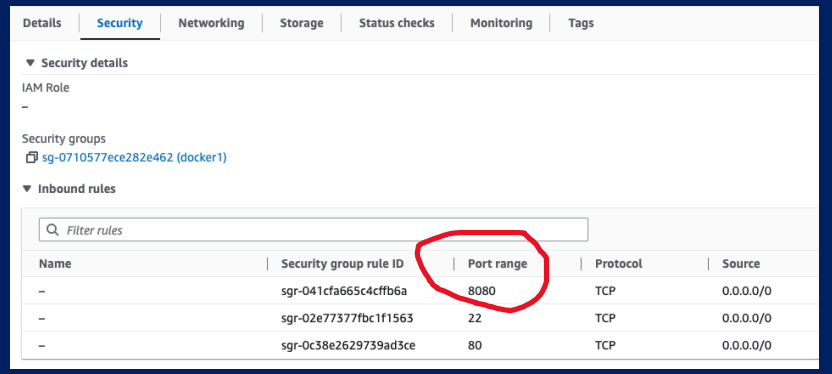
```
docker run -p 8080:80 nginx
```

```
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
fc7181108d40: Already exists
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
```

```
Status: Downloaded newer image for nginx:latest
```


aws
Services
Search
[Option+S]
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
EC2 > Instances > i-019ec85672c8104f2
Instance summary for i-019ec85672c8104f2 (docker0) Info
Updated less than a minute ago
Instance ID
i-019ec85672c8104f2 (docker0)
IPv6 address
–
Hostname type
IP name:
Answer private resource DNS name
IPv4 (A)
Auto-assigned IP address
–
IAM Role
–
Public IPv4 address
13.212.145.155 | open address
Instance state
Terminated
Private IP DNS name (IPv4 only)
ip-172-31-2-85.ap-southeast-1.compute.internal
Instance type
t2.medium
VPC ID
vpc-b56d7cd2
Subnet ID
subnet-d77acb8e
Details
Security
Networking
Storage
Status checks
Monitoring
Tags


Not Secure
18.143.155.126:8080
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.

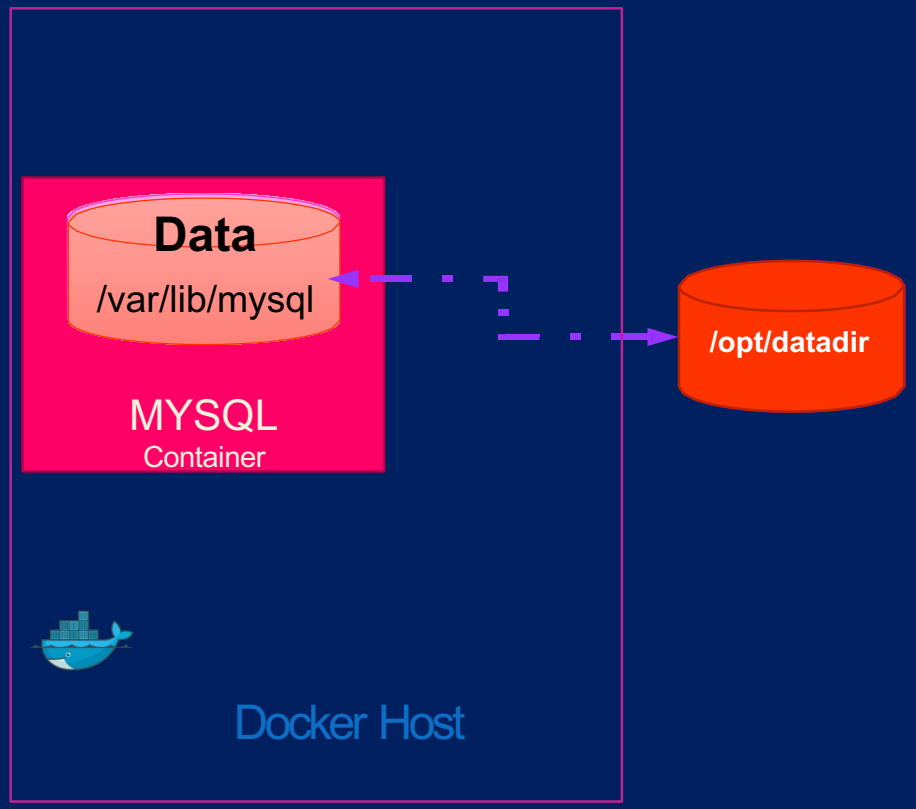

Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Security details
IAM Role
–
Security groups
sg-0710577ece282e462 (docker1)
Inbound rules
Filter rules
Name
Security group rule ID
Port range
Protocol
Source
–
sgr-041cfa665c4cffb6a
8080
TCP
0.0.0.0/0
–
sgr-02e77377fbc1f1563
22
TCP
0.0.0.0/0
–
sgr-0c38e2629739ad3ce
80
TCP
0.0.0.0/0

RUN – Volume mapping

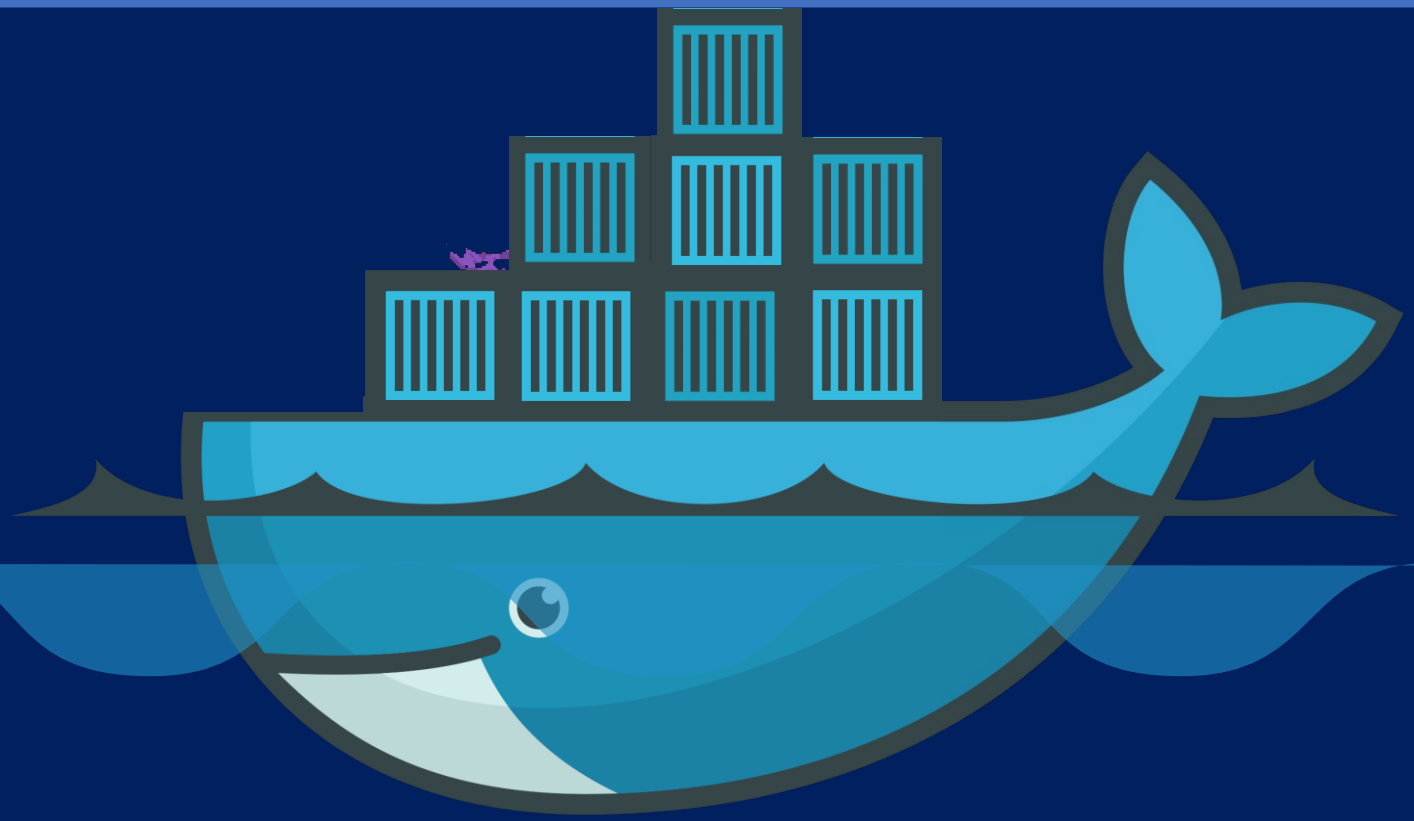
```
docker run mysql
```

```
docker stop mysql
docker rm mysql
```

```
docker run -v /opt/datadir:/var/lib/mysql mysql
```

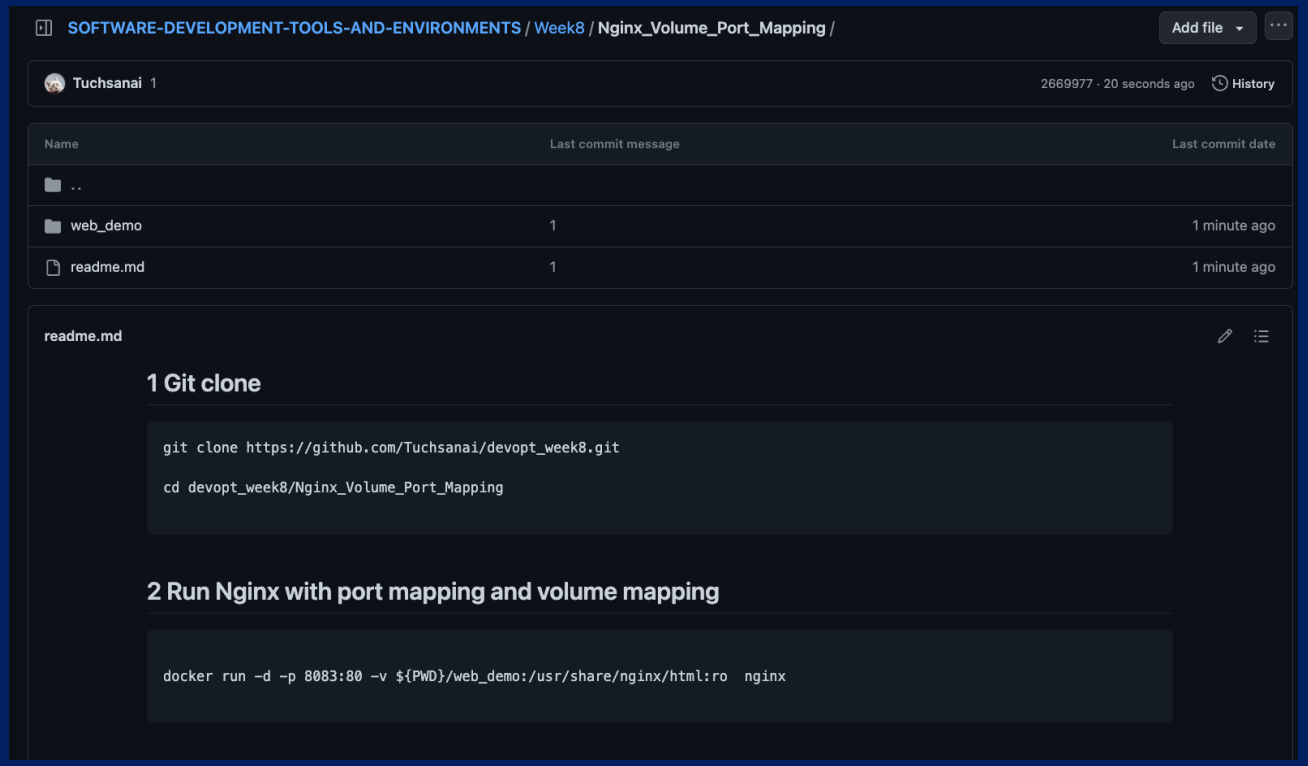


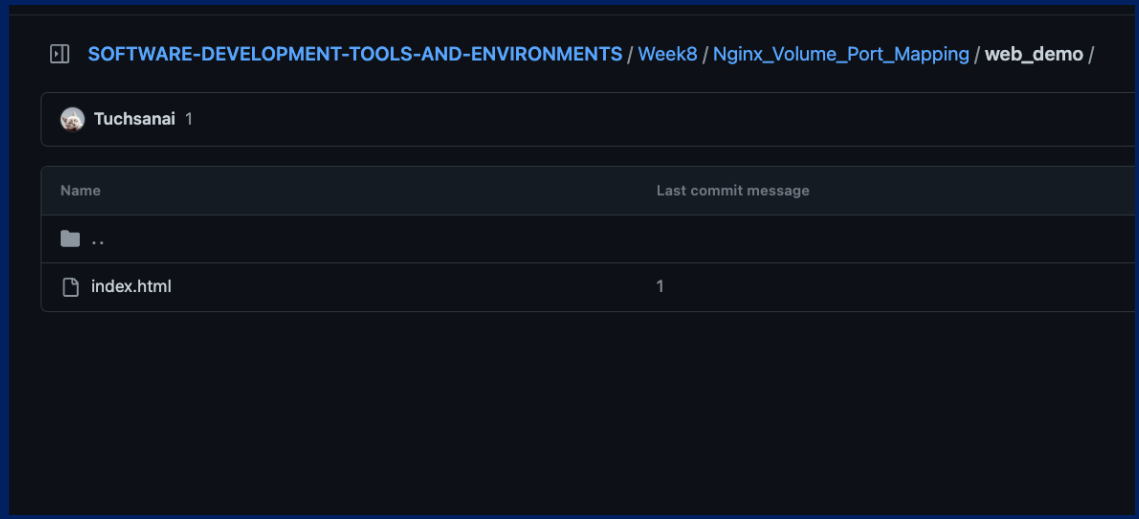
Docker run from Repository

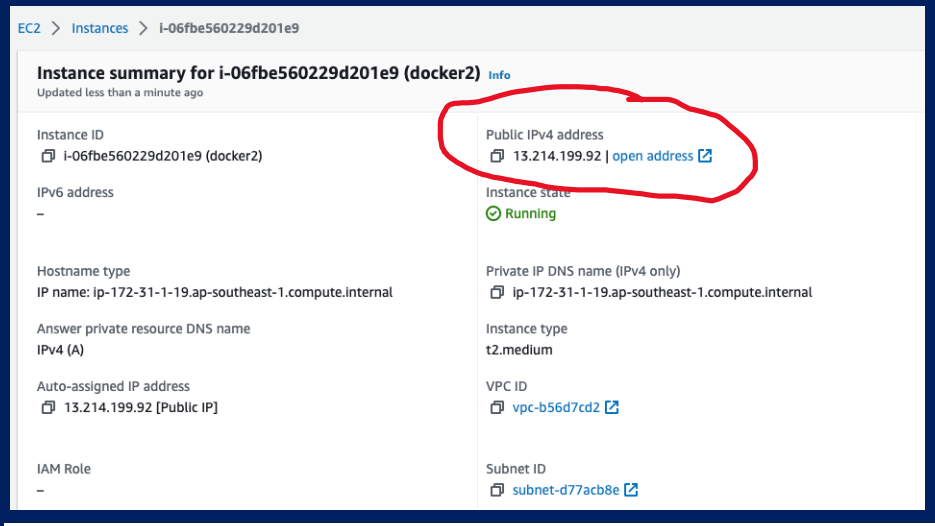


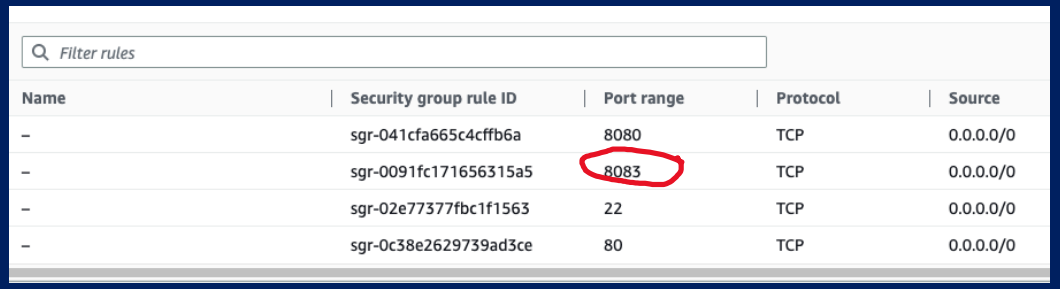
LAB2 :Run Nginx with Volume and Port Mapping

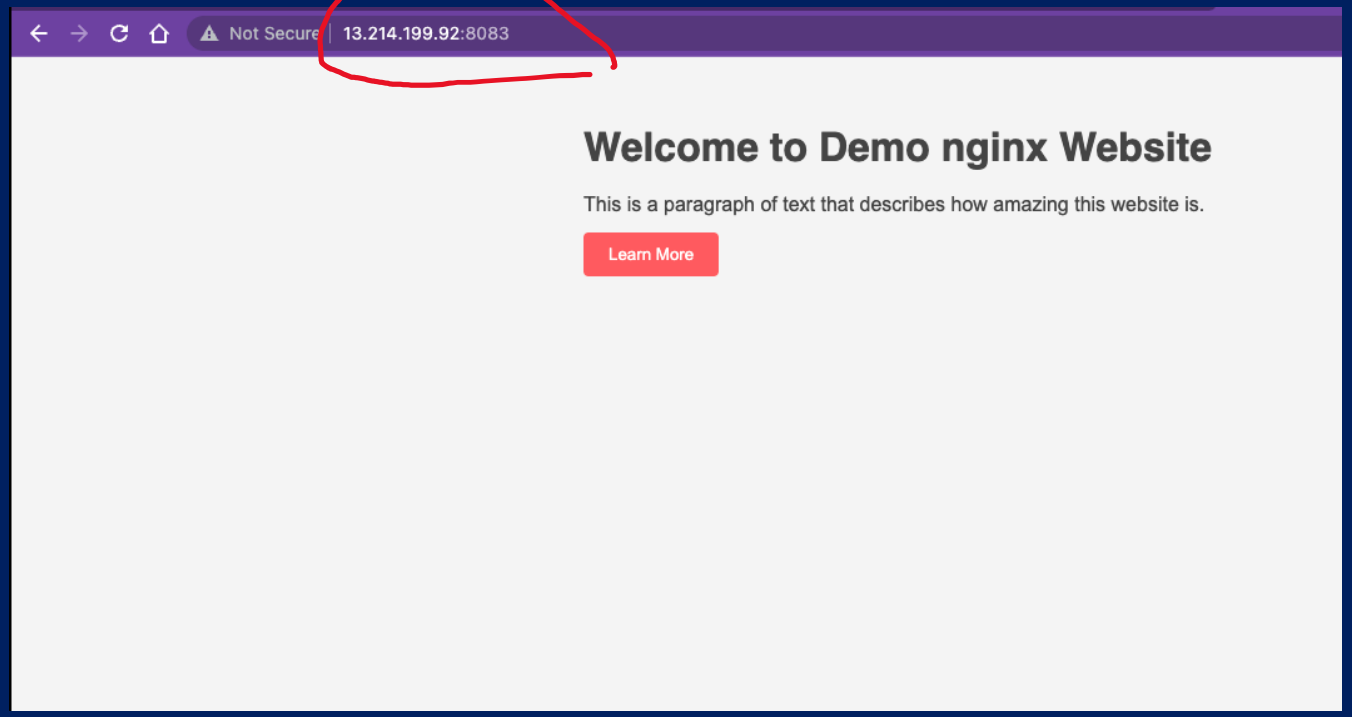
```
docker run -d -p 8080:80 -v ${PWD}/web_demo:/usr/share/nginx/html  nginx
```



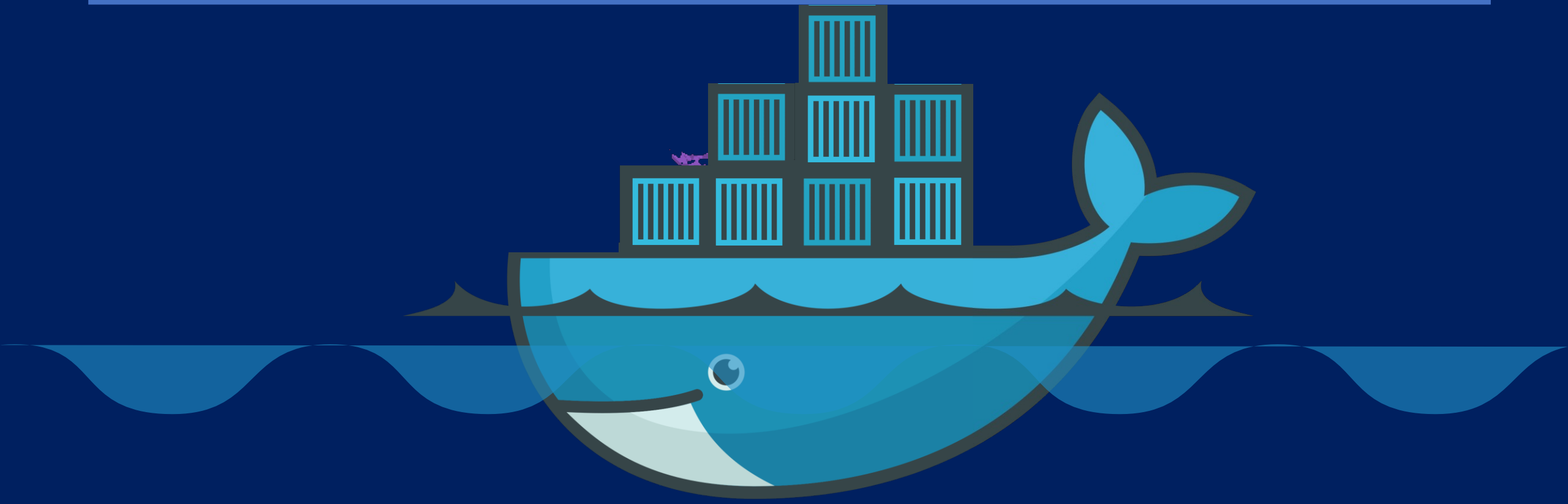



EC2 > Instances > i-06fbe560229d201e9
Instance summary for i-06fbe560229d201e9 (docker2) Info
Updated less than a minute ago
Instance ID
i-06fbe560229d201e9 (docker2)
IPv6 address
–
Hostname type
IP name: ip-172-31-1-19.ap-southeast-1.compute.internal
Answer private resource DNS name
IPv4 (A)
Auto-assigned IP address
13.214.199.92 [Public IP]
IAM Role
–
Public IPv4 address
13.214.199.92 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
ip-172-31-1-19.ap-southeast-1.compute.internal
Instance type
t2.medium
VPC ID
vpc-b56d7cd2
Subnet ID
subnet-d77acb8e


Filter rules
Name
Security group rule ID
Port range
Protocol
Source
– sgr-041cfa665c4cffb6a 8080 TCP 0.0.0.0/0
– sgr-0091fc171656315a5 8083 TCP 0.0.0.0/0
– sgr-02e77377fbc1f1563 22 TCP 0.0.0.0/0
– sgr-0c38e2629739ad3ce 80 TCP 0.0.0.0/0


Not Secure
13.214.199.92:8083
Welcome to Demo nginx Website
This is a paragraph of text that describes how amazing this website is.
Learn More

Docker build and run from Repository



LAB3 : Build and Run Docker Image

SOFTWARE-DEVELOPMENT-TOOLS-AND-ENVIRONMENTS / Week8 / node-bulletin-board-master /

Tuchsanai 1

Name	Last commit message
..	
bulletin-board-app	p
.gitignore	0
LICENSE	0
readme.md	1

readme.md

1 Git clone

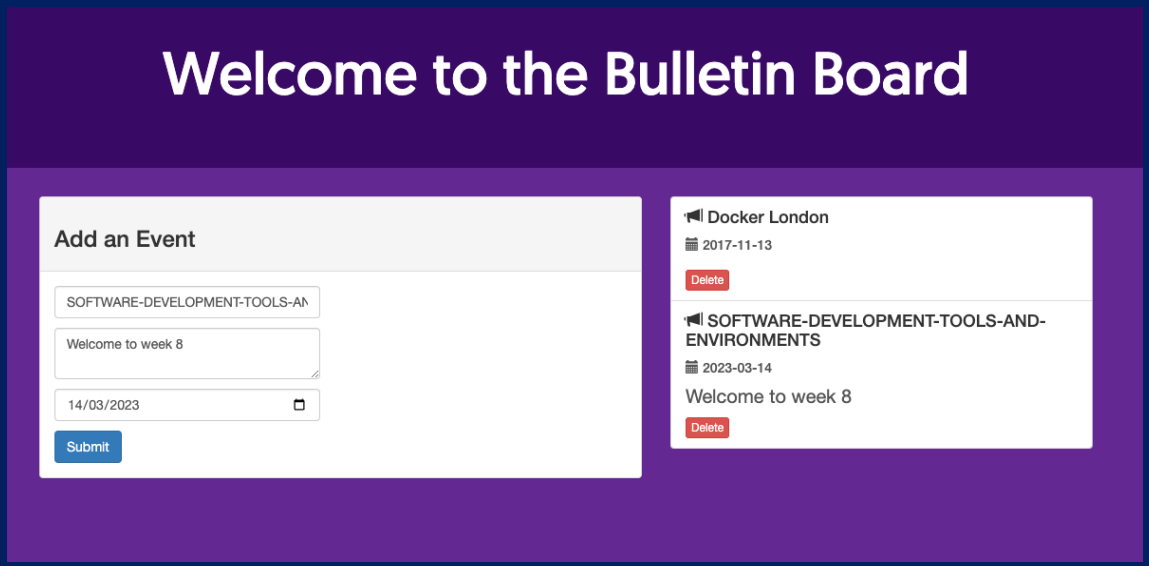
```
git clone https://github.com/Tuchsanai/devopt_week8.git

cd devopt_week8/node-bulletin-board-master/bulletin-board-app
```

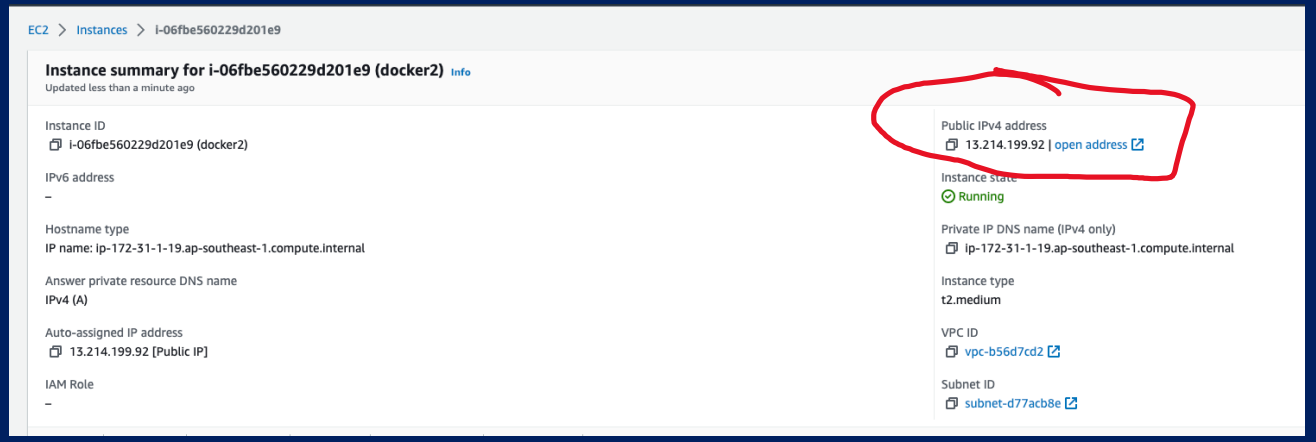
3 Build Docker image

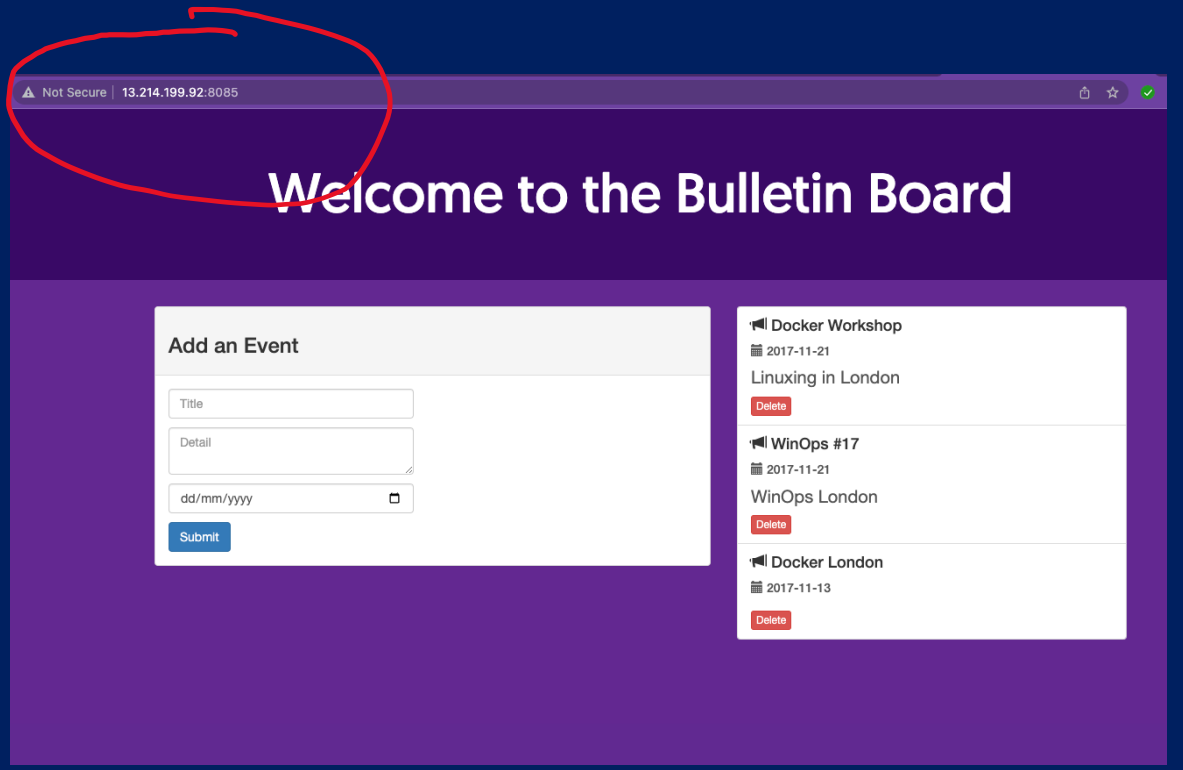
```
docker build -t bulletinboard:1.0 .
```

3 Run Nginx with port mapping and volume mapping

```
docker run -p 8085:8080 -d --name bb bulletinboard:1.0
```







▼ Inbound rules

Filter rules

Name	Security group rule ID	Port range	Protocol	Source
–	sgr-041cfa665c4cffb6a	8080	TCP	0.0.0.0/0
–	sgr-0f5e344cab838a992	8085	TCP	0.0.0.0/0
–	sgr-0091fc171656315a5	8083	TCP	0.0.0.0/0
–	sgr-02e77377fbc1f1563	22	TCP	0.0.0.0/0
–	sgr-0c38e2629739ad3ce	80	TCP	0.0.0.0/0

▼ Outbound rules