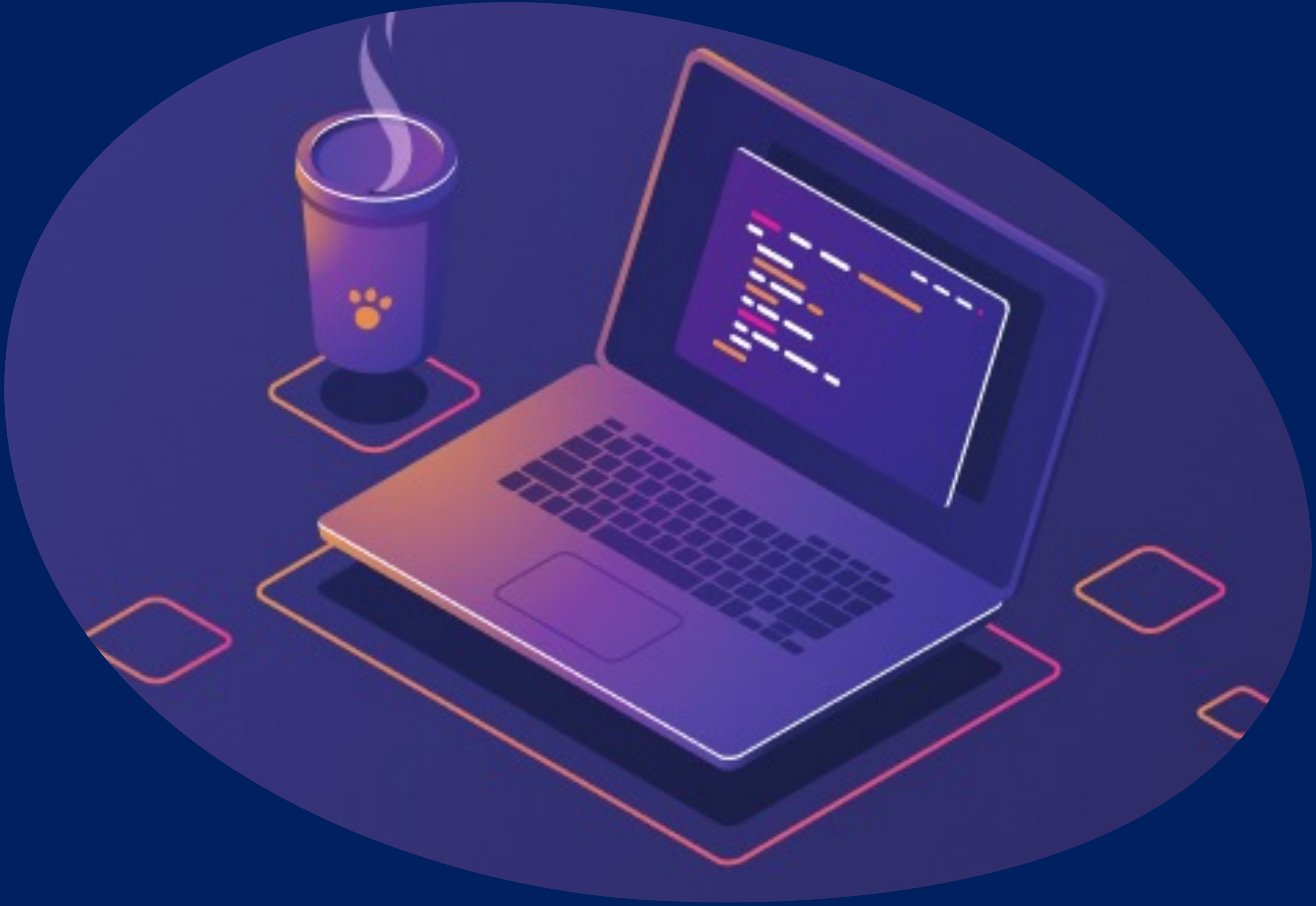
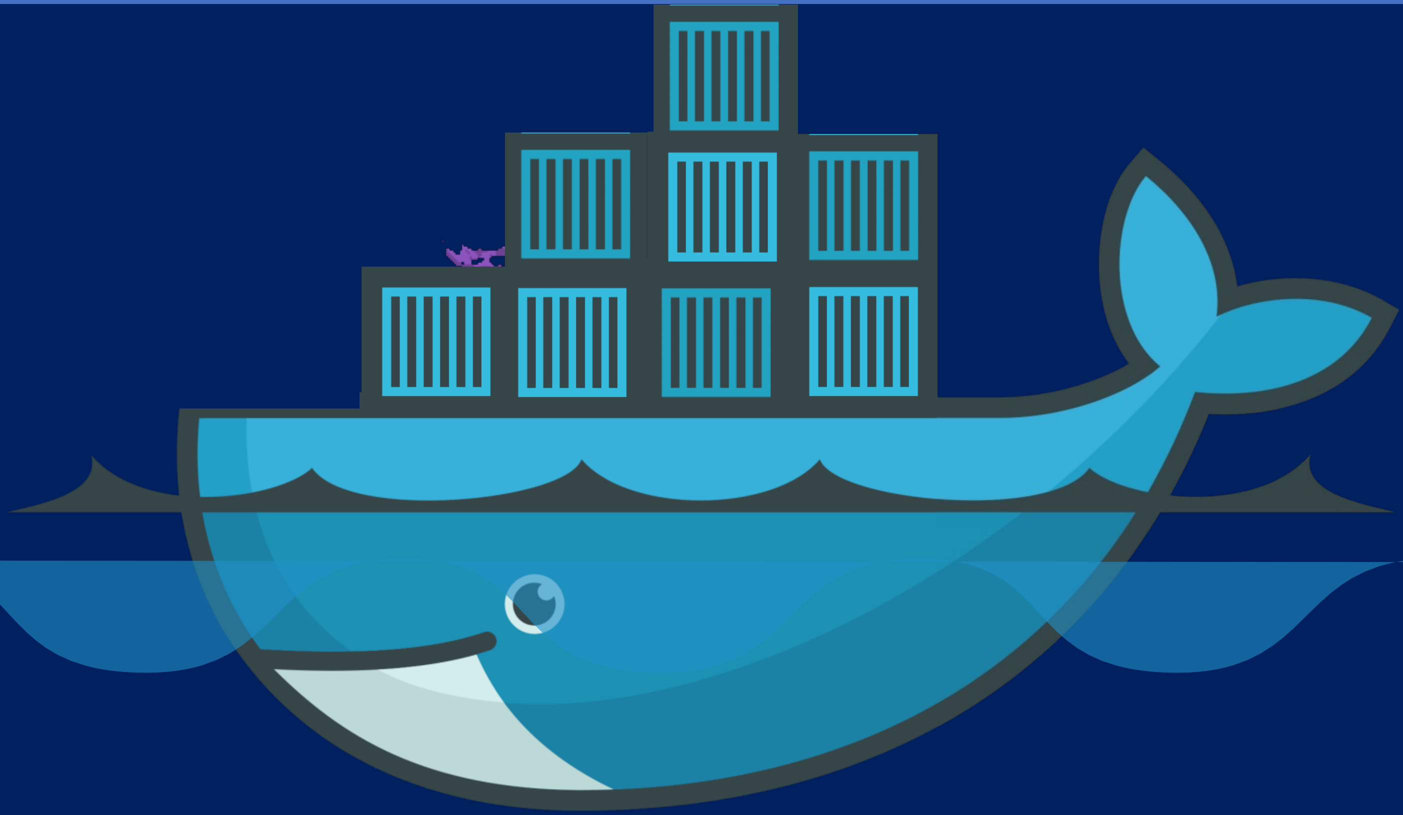
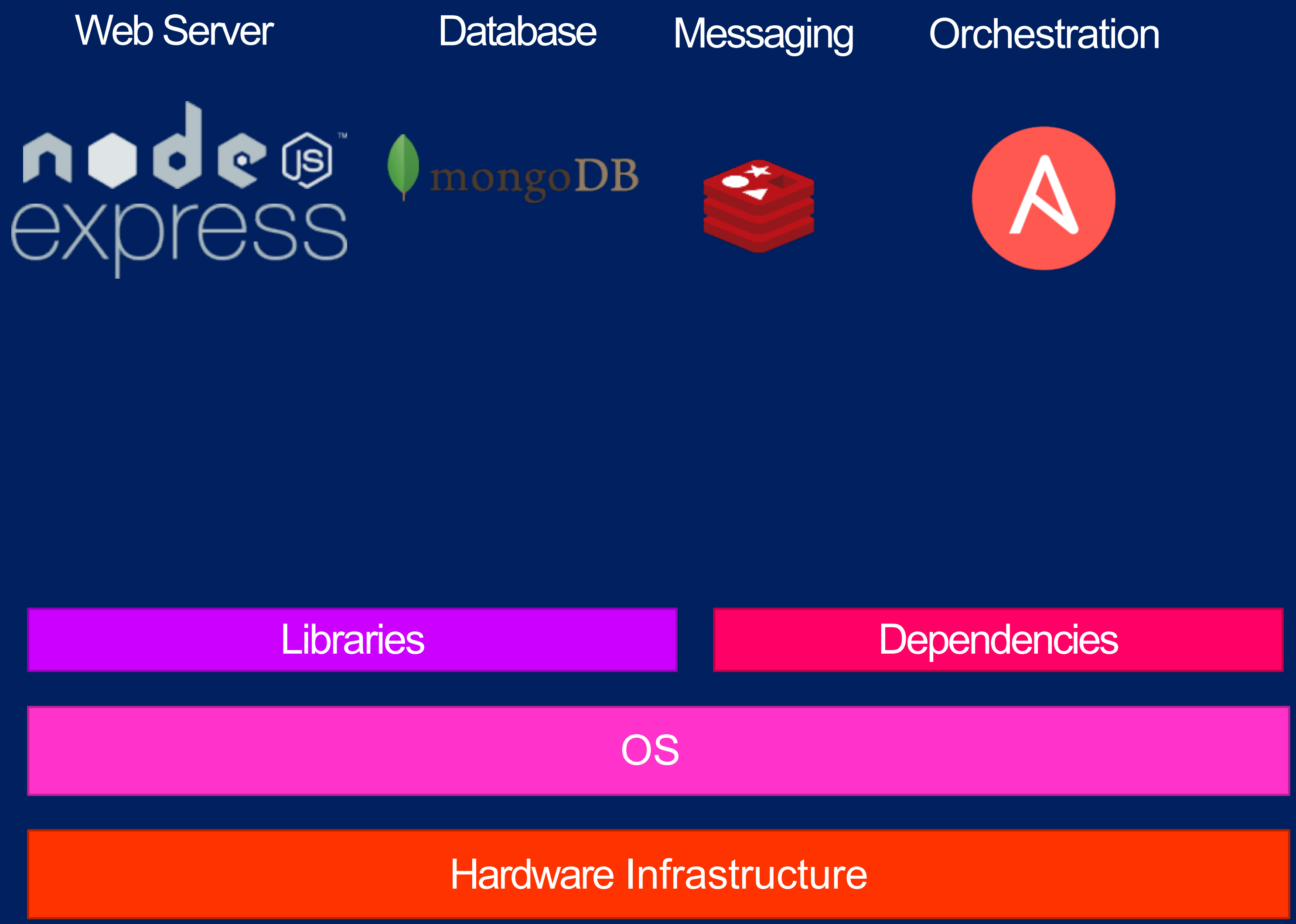


Week 8 : SOFTWARE DEVELOPMENT TOOLS AND ENVIRONMENTS

Docker Overview



Why do you need docker?



Web Server

Database

Messaging

Orchestration

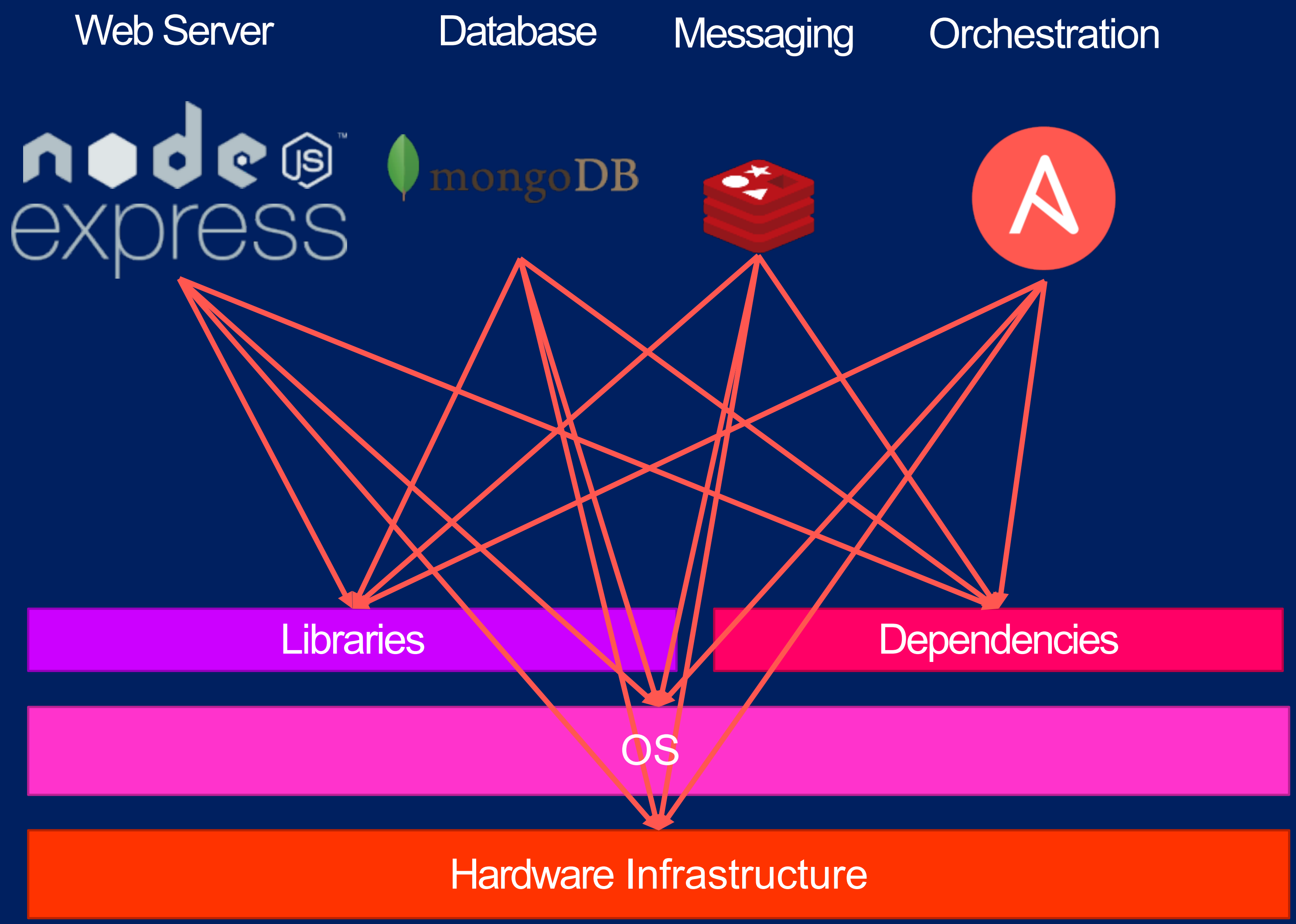
Libraries

Dependencies

OS

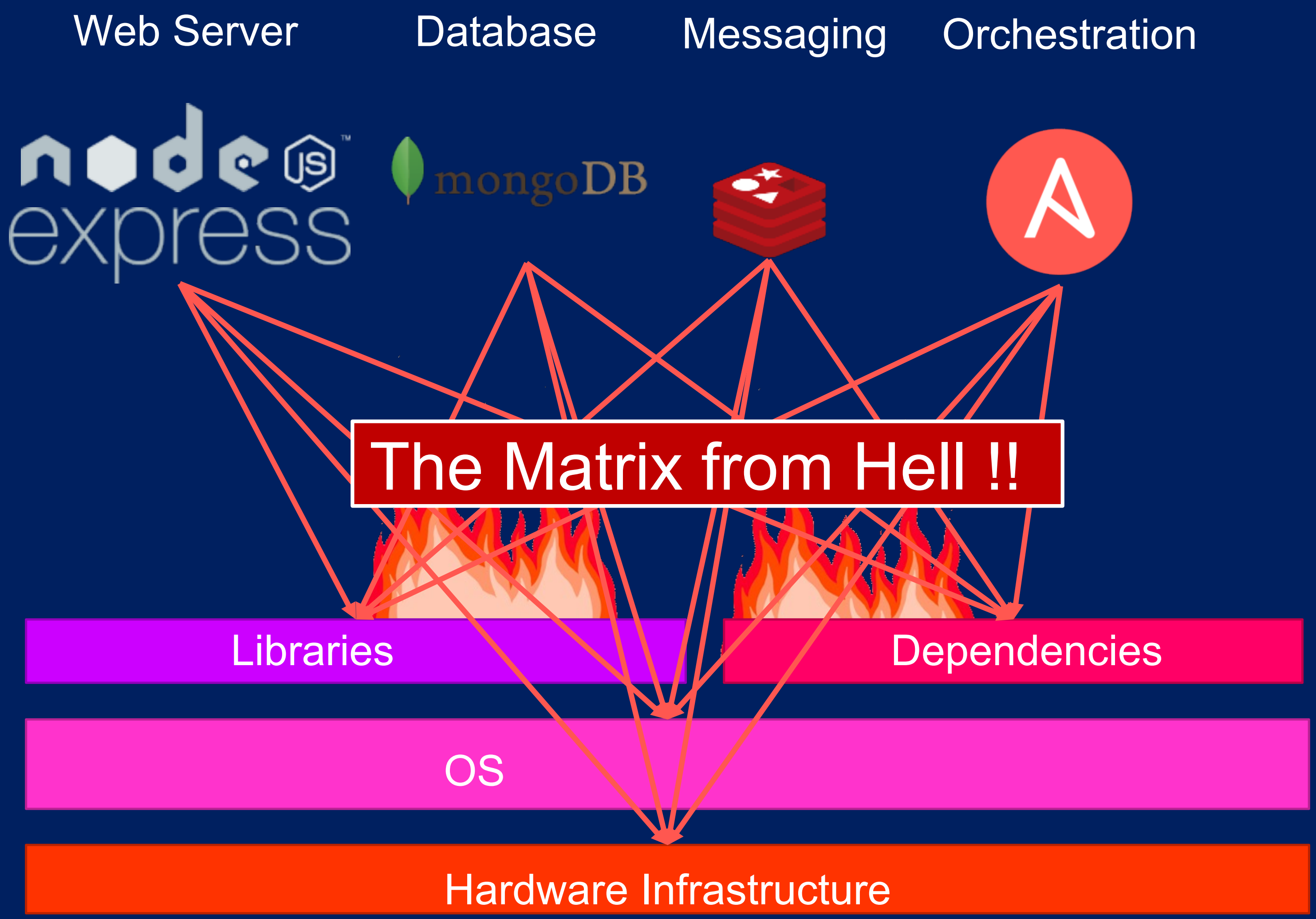
Hardware Infrastructure

Why do you need docker?



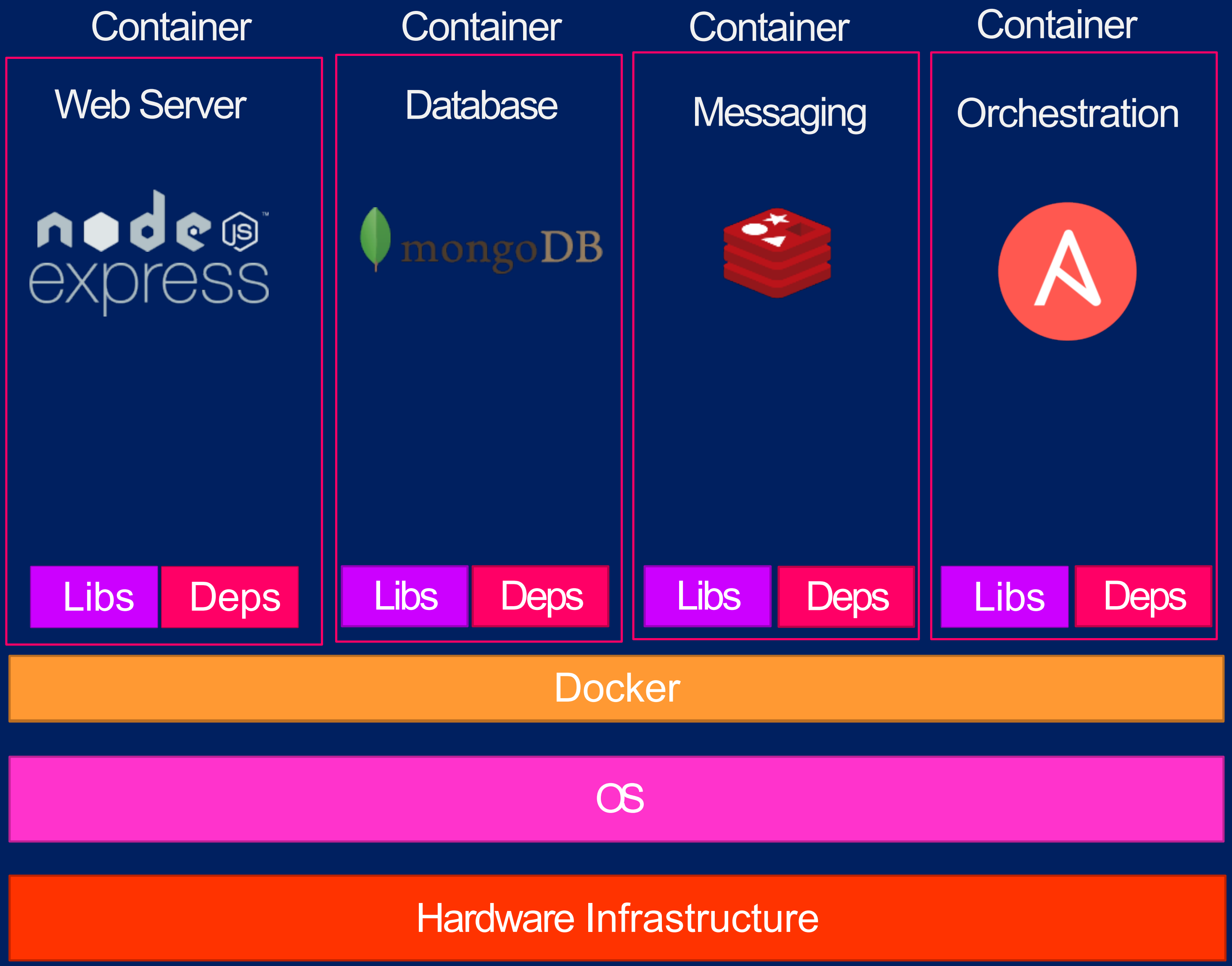
Why do you need docker?

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

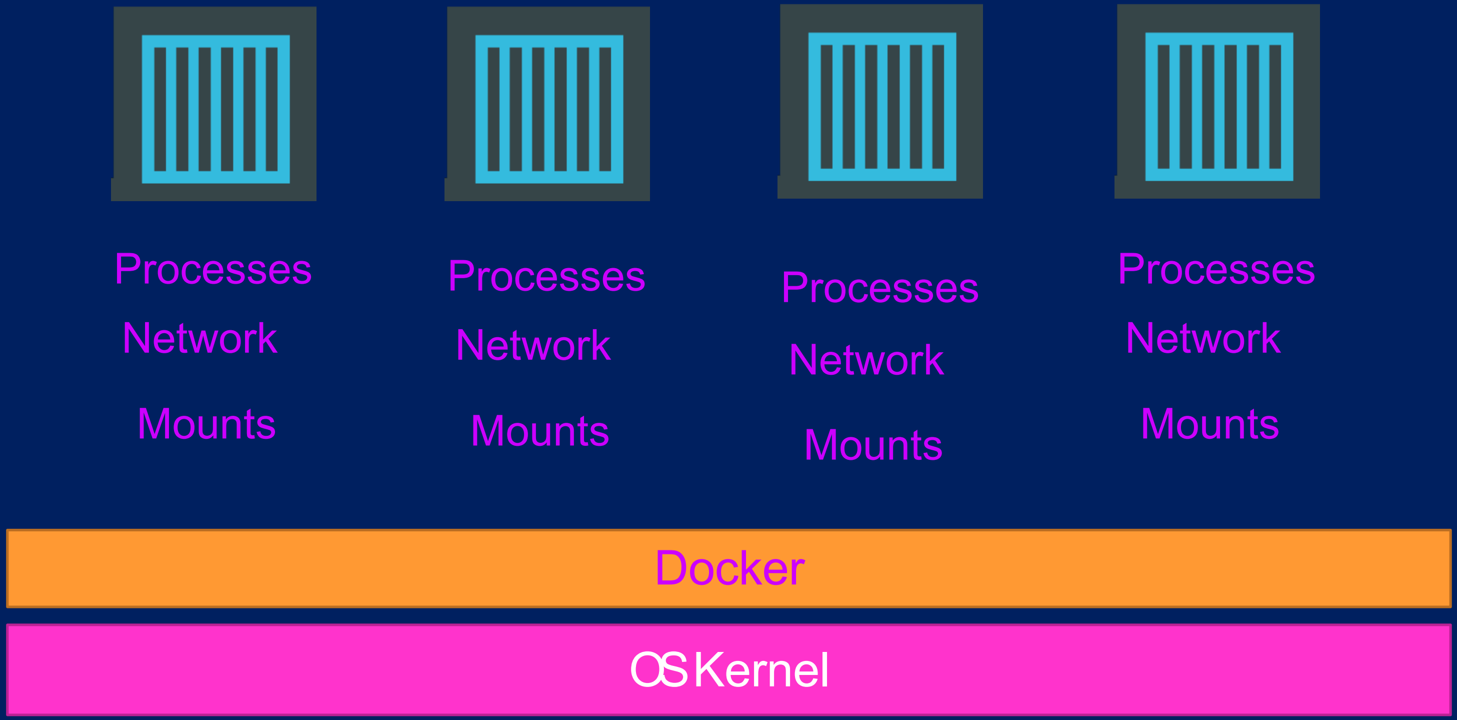


What can it do?

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



What are containers?



Processes

Network

Mounts

Processes

Network

Mounts

Processes

Network

Mounts

Processes

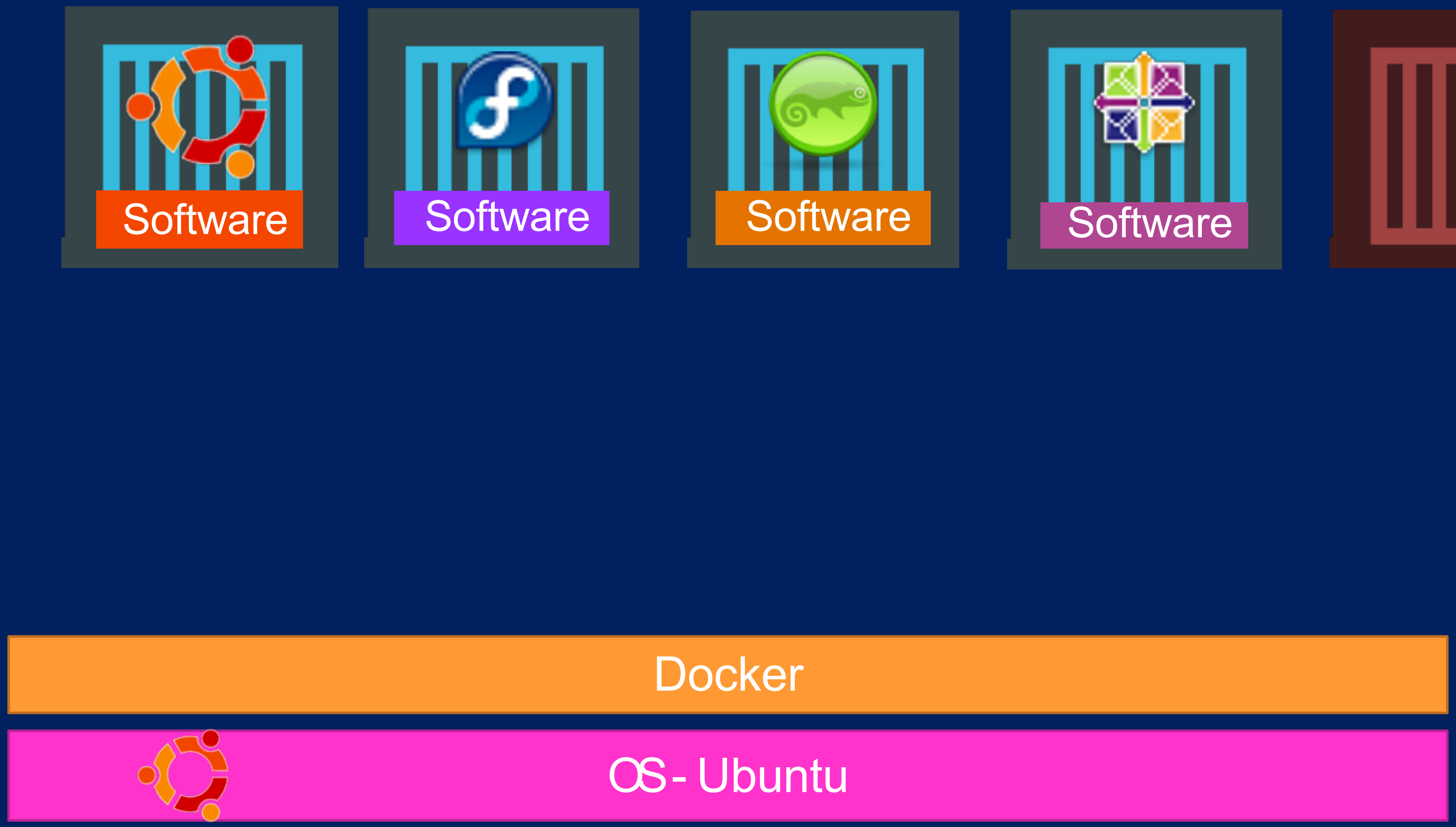
Network

Mounts

Docker

OS Kernel

Sharing the kernel for Operating System



Software

Software

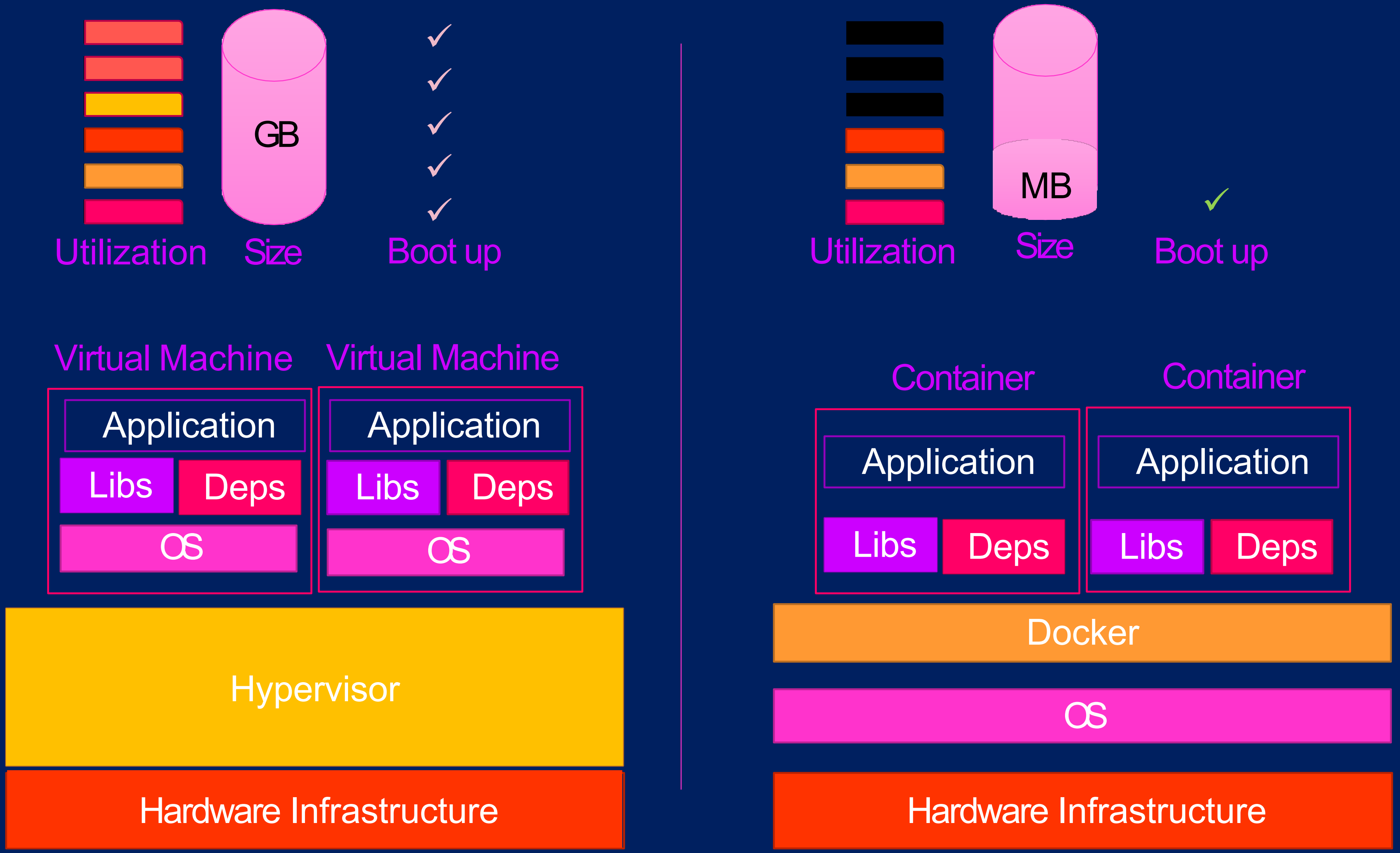
Software

Software

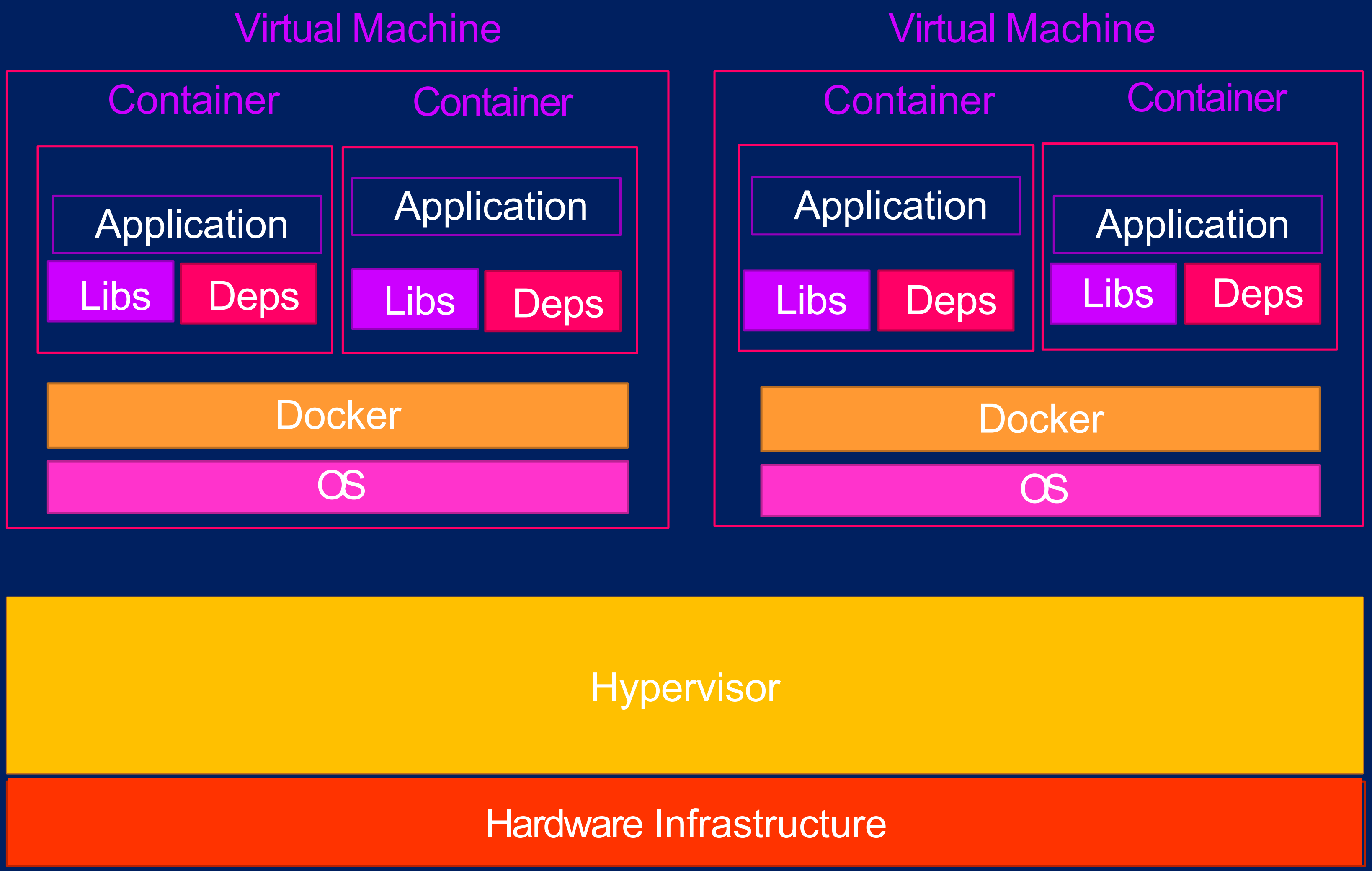
Docker

OS - Ubuntu

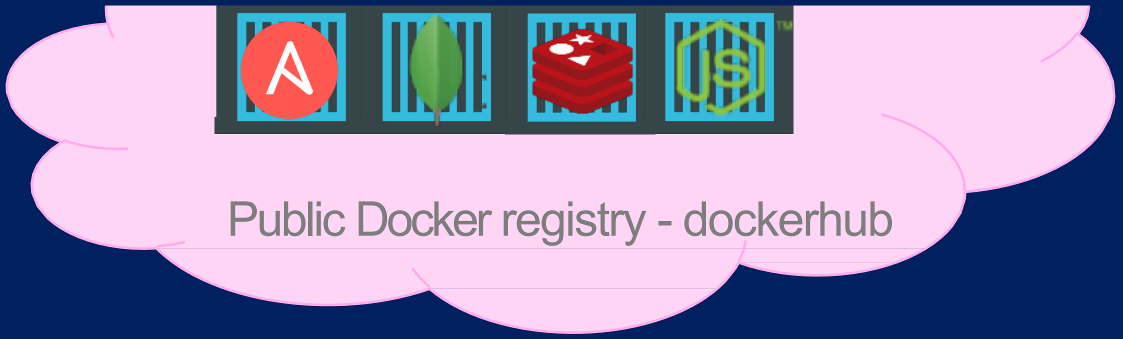
Containers vs Virtual Machines



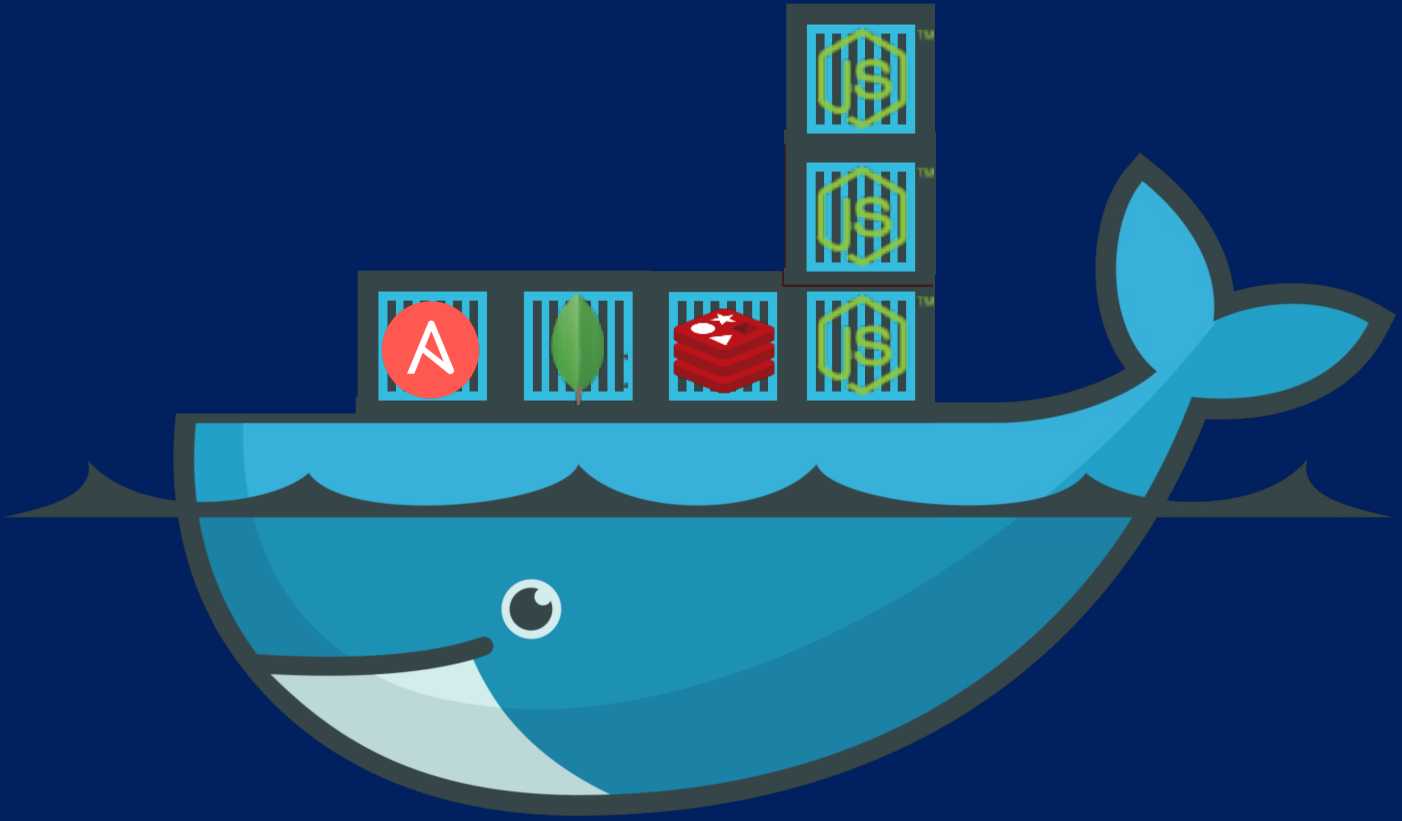
Containers & Virtual Machines



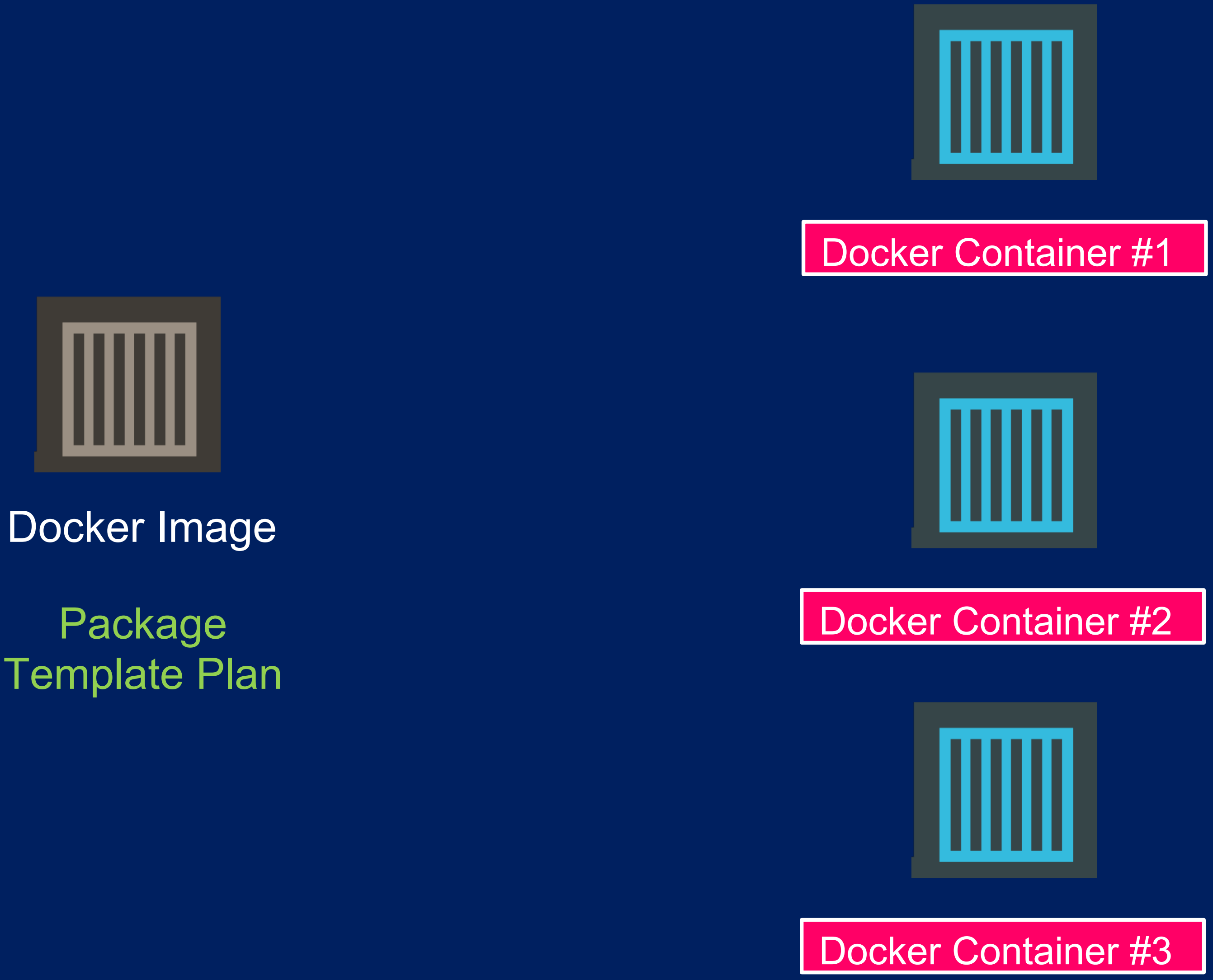
How is it done?



```
docker run ansible

docker run mongodb

docker run redis

docker run nodejs

docker run nodejs

docker run nodejs
```



Container vs image



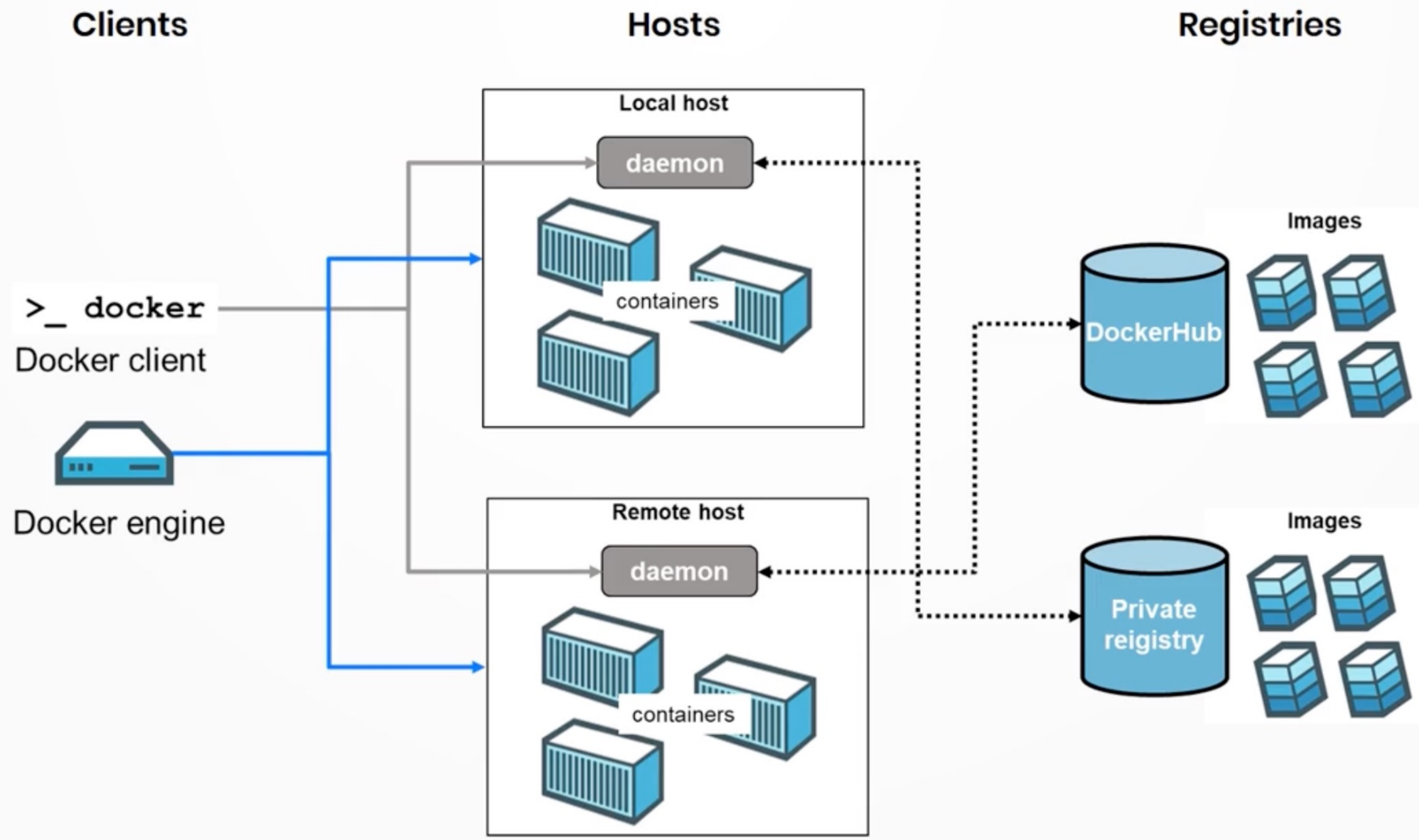
Docker Image

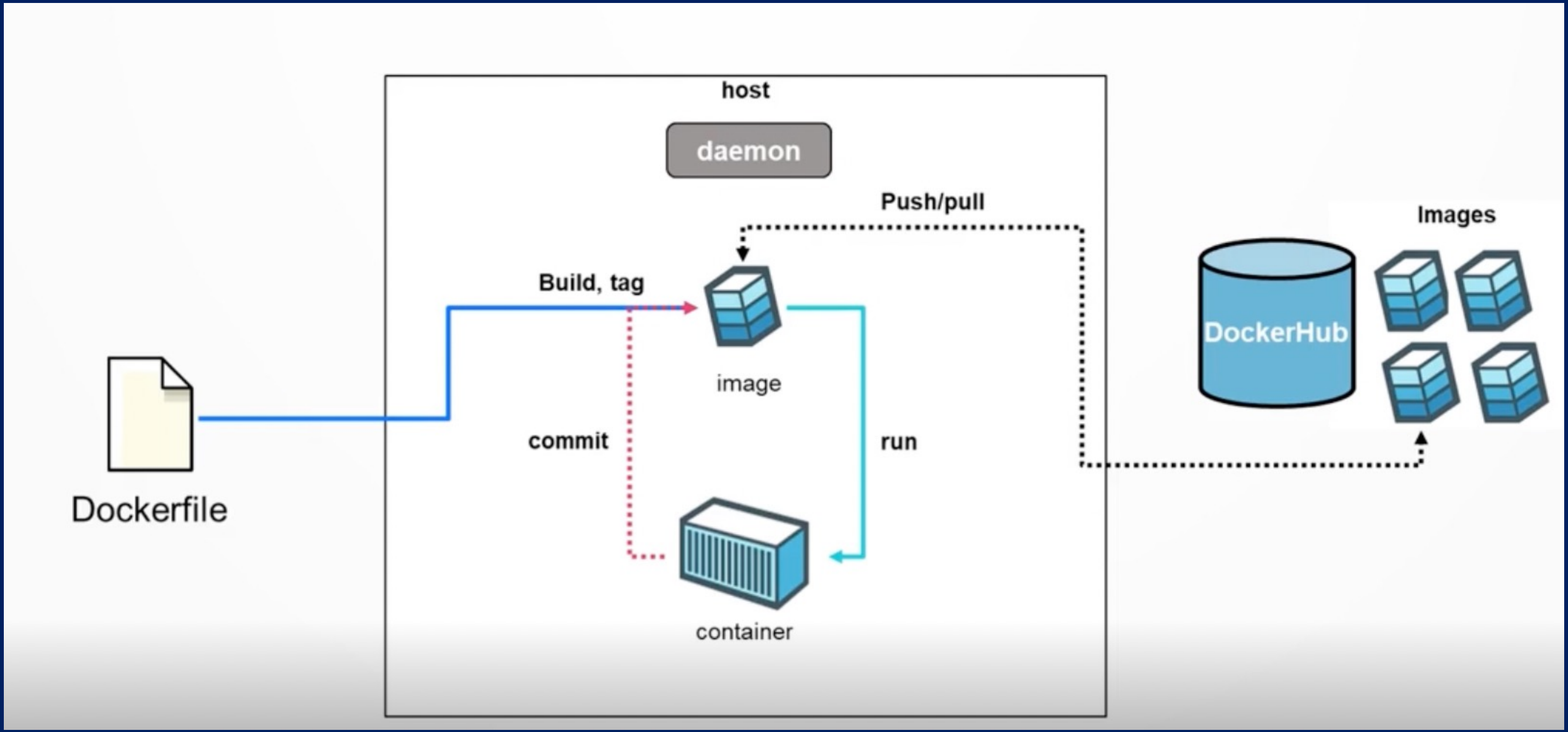
Package
Template Plan

Docker Container #1

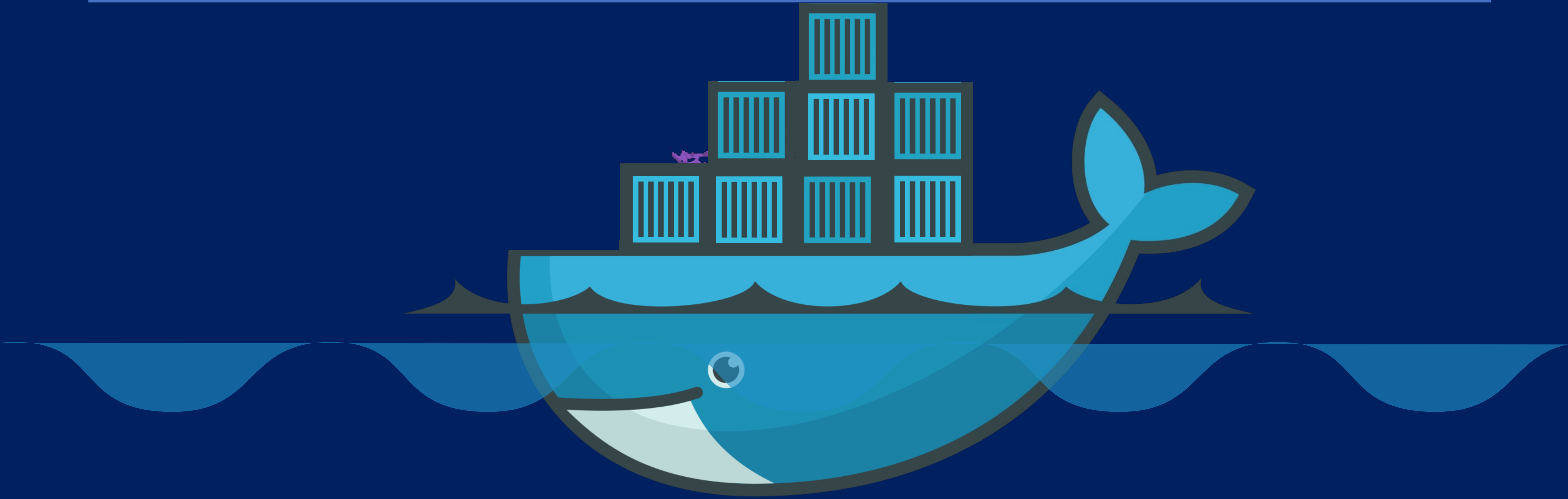
Docker Container #2

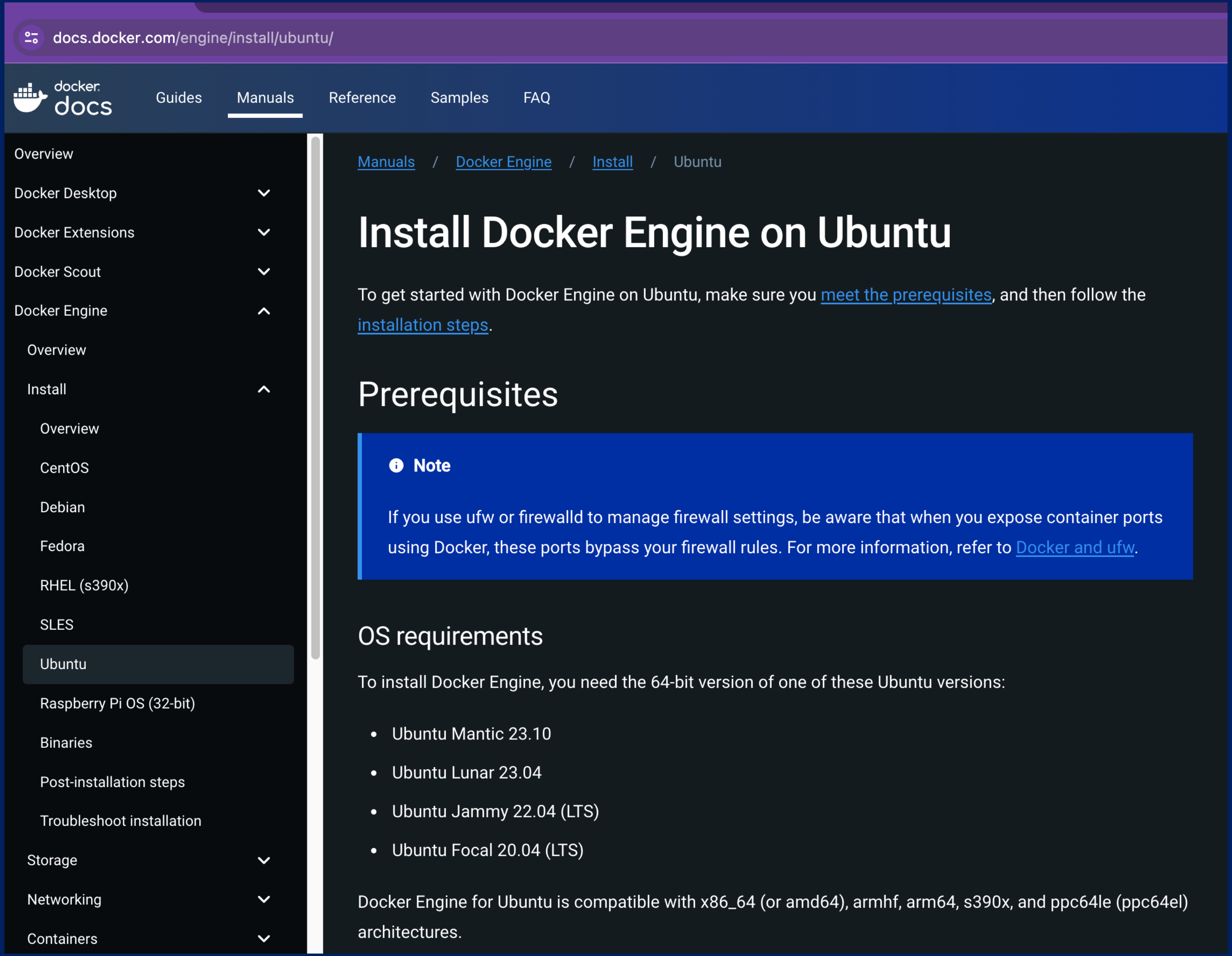
Docker Container #3


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


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

Lab1: Install Docker





Manuals / Docker Engine / Install / Ubuntu

Install Docker Engine on Ubuntu

To get started with Docker Engine on Ubuntu, make sure you meet the prerequisites, and then follow the installation steps.

Prerequisites

Note

If you use ufw or firewalld to manage firewall settings, be aware that when you expose container ports using Docker, these ports bypass your firewall rules. For more information, refer to Docker and ufw.

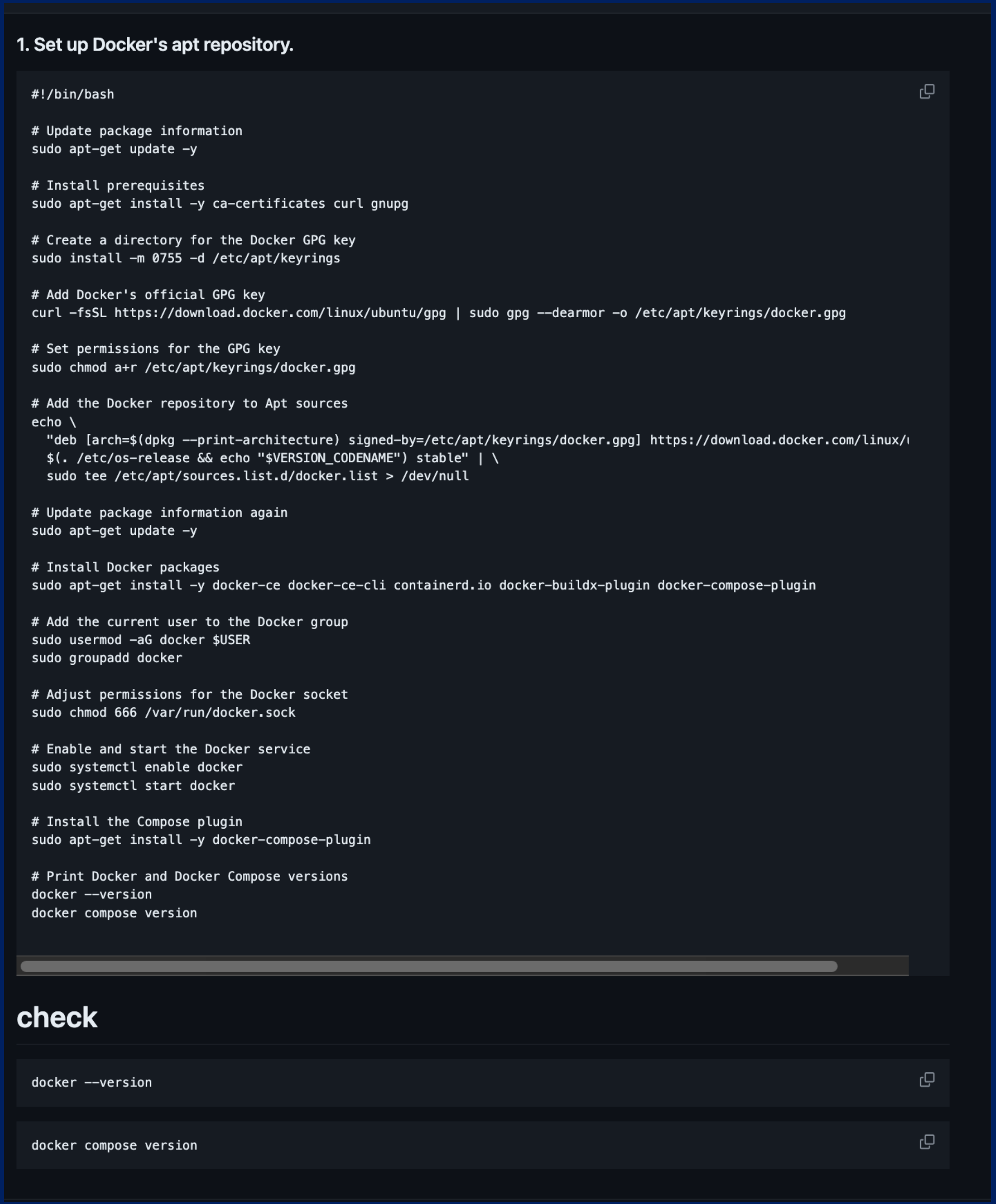
OS requirements

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

- Ubuntu Mantic 23.10
- Ubuntu Lunar 23.04
- Ubuntu Jammy 22.04 (LTS)
- Ubuntu Focal 20.04 (LTS)

Docker Engine for Ubuntu is compatible with x86_64 (or amd64), armhf, arm64, s390x, and ppc64le (ppc64el) architectures.

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



1. Set up Docker's apt repository.

```
#!/bin/bash

# Update package information
sudo apt-get update -y

# Install prerequisites
sudo apt-get install -y ca-certificates curl gnupg

# Create a directory for the Docker GPG key
sudo install -m 0755 -d /etc/apt/keyrings

# Add Docker's official GPG key
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

# Set permissions for the GPG key
sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Add the Docker repository to Apt sources
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Update package information again
sudo apt-get update -y

# Install Docker packages
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

# Add the current user to the Docker group
sudo usermod -aG docker $USER
sudo groupadd docker

# Adjust permissions for the Docker socket
sudo chmod 666 /var/run/docker.sock

# Enable and start the Docker service
sudo systemctl enable docker
sudo systemctl start docker

# Install the Compose plugin
sudo apt-get install -y docker-compose-plugin

# Print Docker and Docker Compose versions
docker --version
docker compose version
```

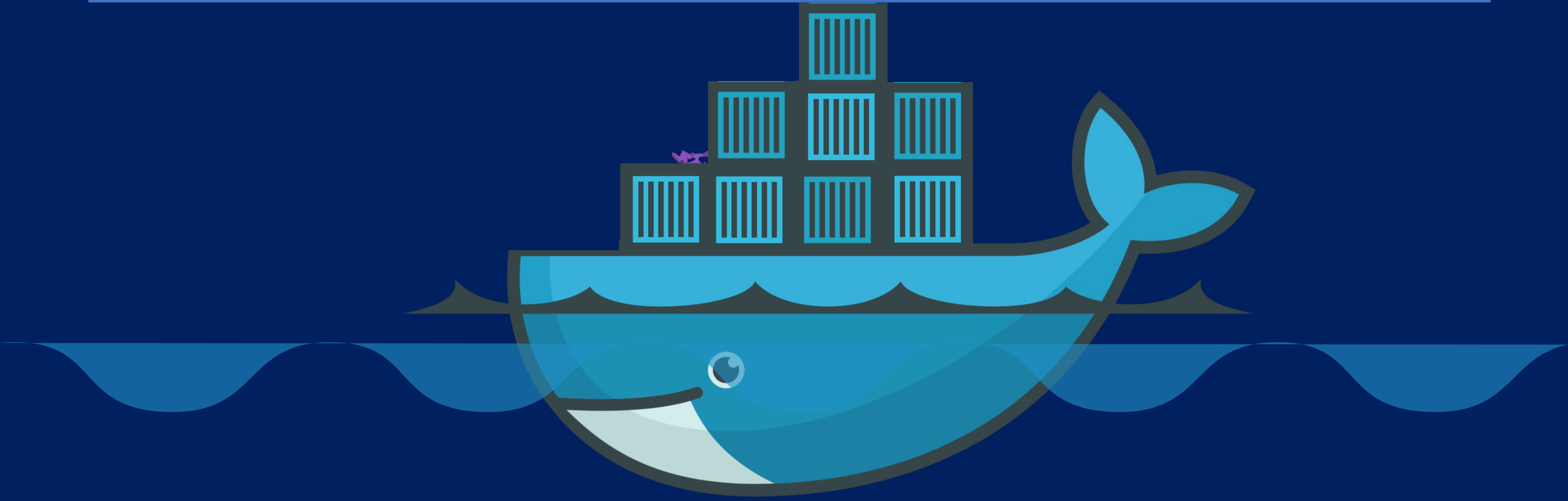
check

```
docker --version
```

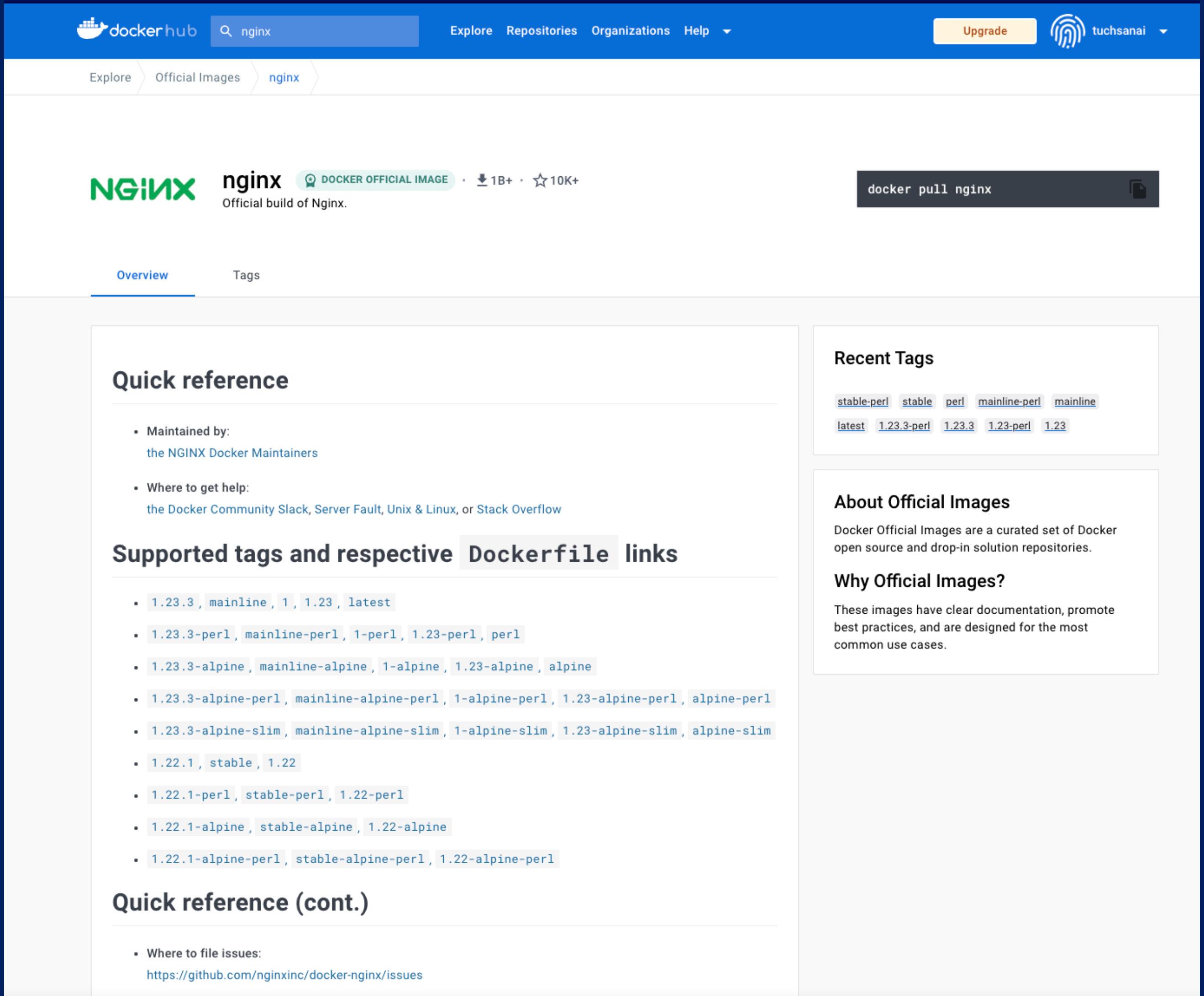
```
docker compose version
```

https://github.com/Tuchsanai/DevTools/tree/main/02_Docker

Docker Run



Docker Registry



Explore › Official Images › nginx

nginx DOCKER OFFICIAL IMAGE · 1B+ · ☆10K+

Official build of Nginx.

```
docker pull nginx
```

Overview | Tags

Quick reference

- Maintained by:
  the NGINX Docker Maintainers
- Where to get help:
  the Docker Community Slack, Server Fault, Unix & Linux, or Stack Overflow

Supported tags and respective `Dockerfile` links

- `1.23.3`, `mainline`, `1`, `1.23`, `latest`
- `1.23.3-perl`, `mainline-perl`, `1-perl`, `1.23-perl`, `perl`
- `1.23.3-alpine`, `mainline-alpine`, `1-alpine`, `1.23-alpine`, `alpine`
- `1.23.3-alpine-perl`, `mainline-alpine-perl`, `1-alpine-perl`, `1.23-alpine-perl`, `alpine-perl`
- `1.23.3-alpine-slim`, `mainline-alpine-slim`, `1-alpine-slim`, `1.23-alpine-slim`, `alpine-slim`
- `1.22.1`, `stable`, `1.22`
- `1.22.1-perl`, `stable-perl`, `1.22-perl`
- `1.22.1-alpine`, `stable-alpine`, `1.22-alpine`
- `1.22.1-alpine-perl`, `stable-alpine-perl`, `1.22-alpine-perl`

Quick reference (cont.)

- Where to file issues:
  https://github.com/nginxinc/docker-nginx/issues

Recent Tags

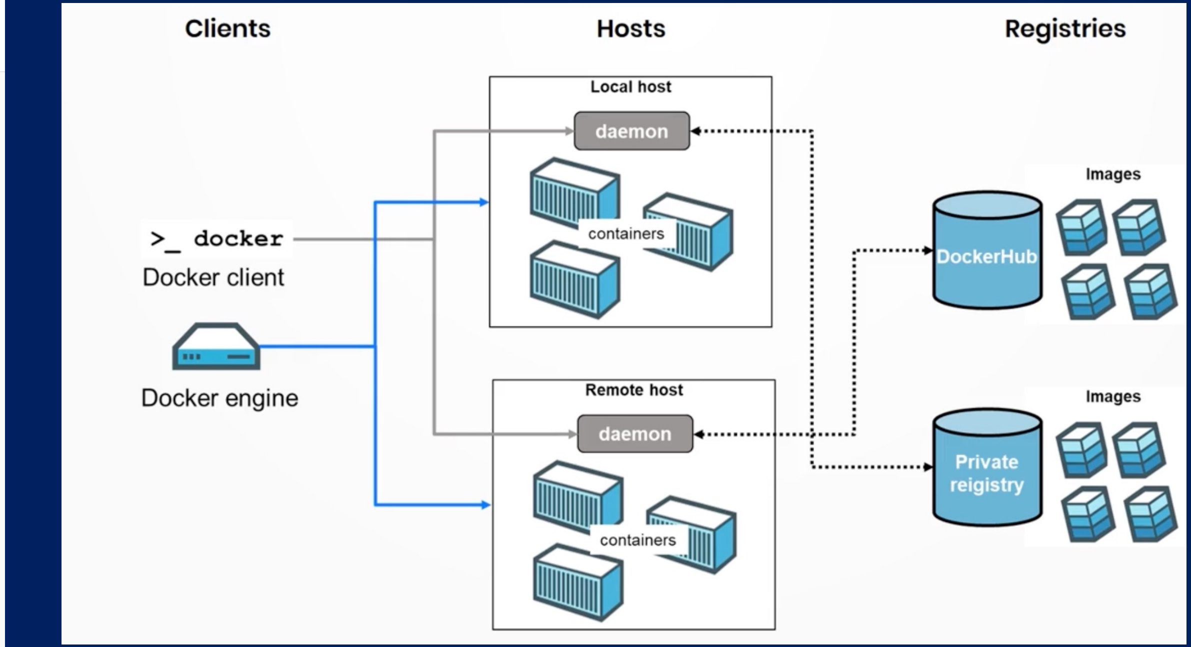
stable-perl stable perl mainline-perl mainline
latest 1.23.3-perl 1.23.3 1.23-perl 1.23

About Official Images

Docker Official Images are a curated set of Docker open source and drop-in solution repositories.

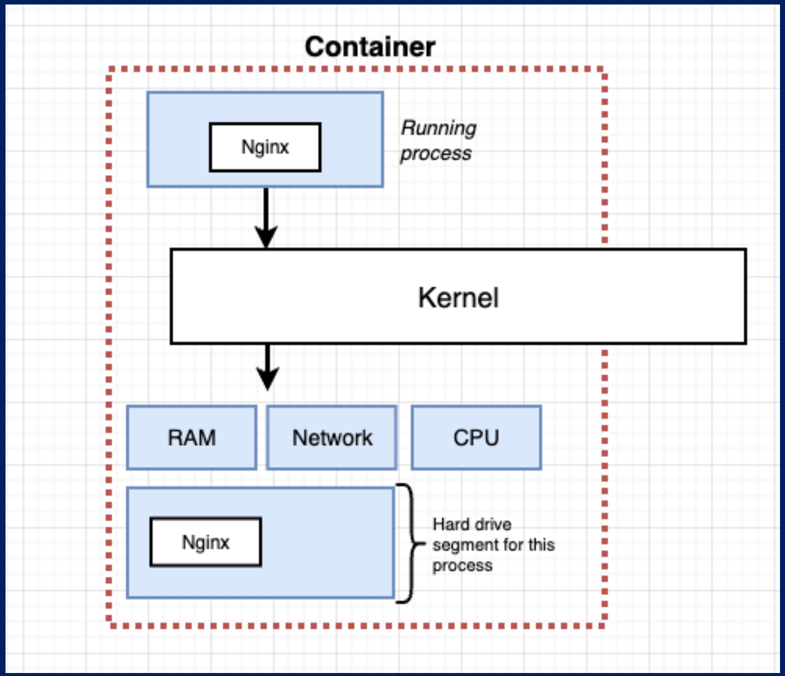
Why Official Images?

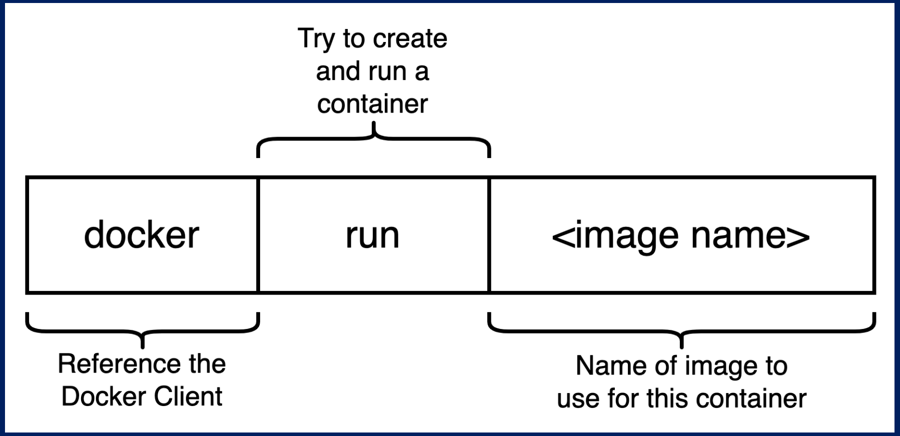
These images have clear documentation, promote best practices, and are designed for the most common use cases.

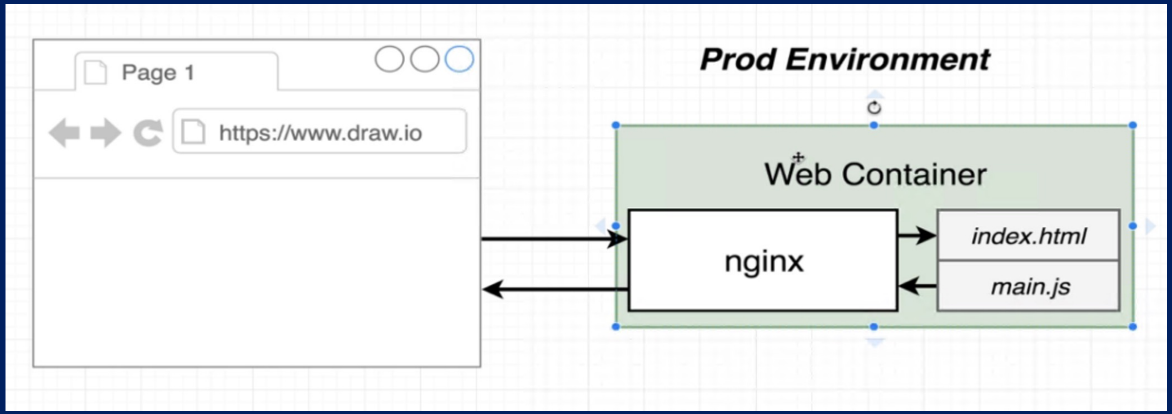


Run – start a container

```
▶ 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
```




Try to create and run a container
docker
run
<image name>
Reference the Docker Client
Name of image to use for this container



```
docker run ubuntu
```

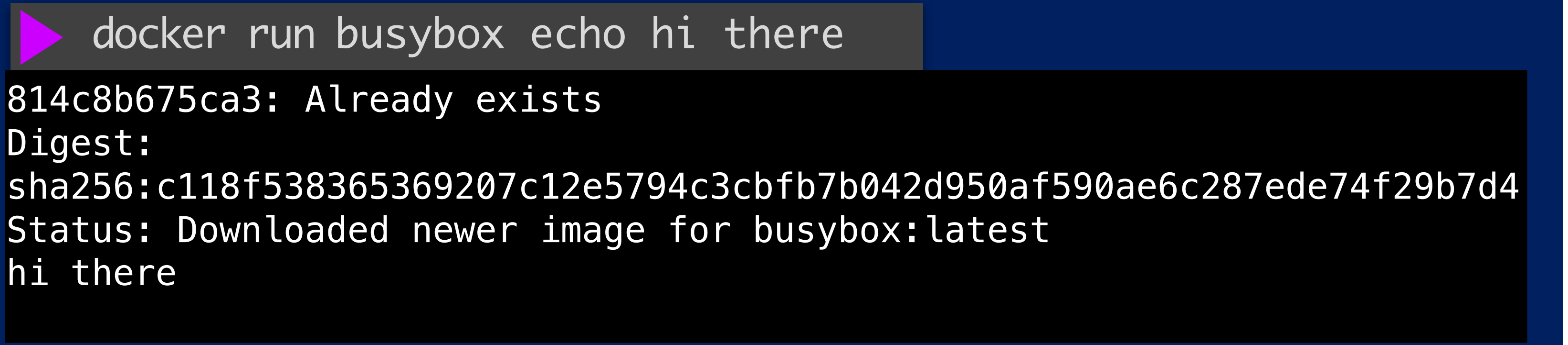
```
docker ps
```

```
CONTAINER ID       IMAGE              COMMAND            CREATED            STATUS             PORTS
```

```
docker ps -a
```

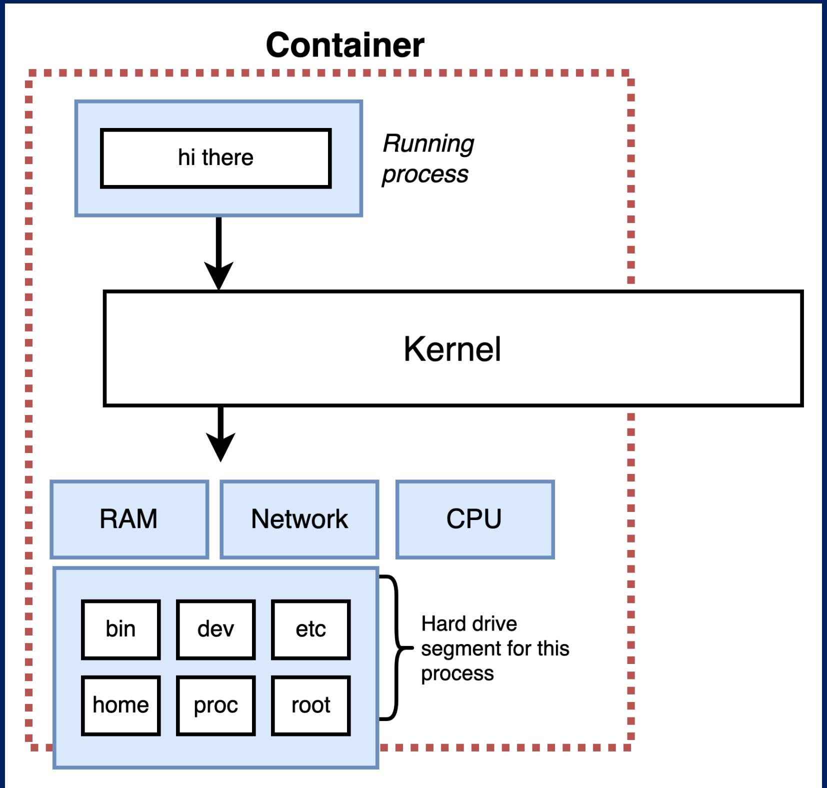
```
CONTAINER ID       IMAGE              COMMAND            CREATED             STATUS                        PORTS
45aacca36850       ubuntu             "/bin/bash"        43 seconds ago      Exited (0) 41 seconds ago
```

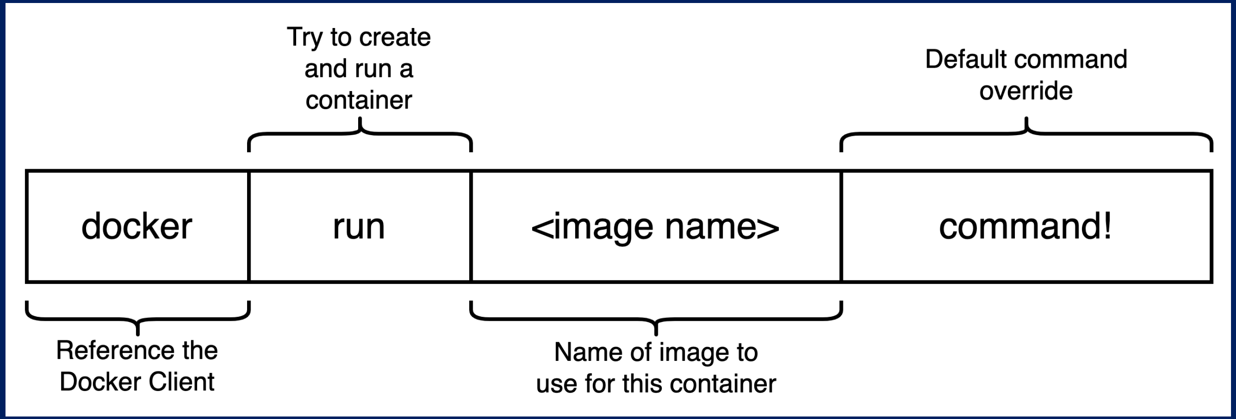
Run – with command

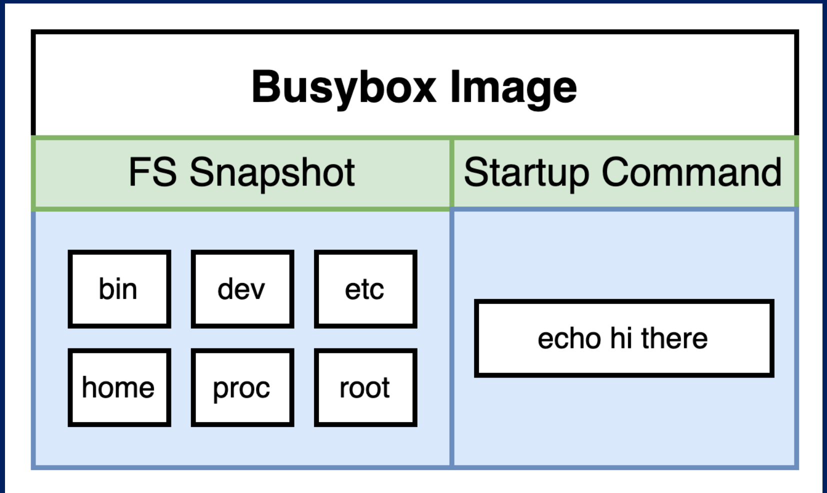


```
docker run busybox echo hi there
```

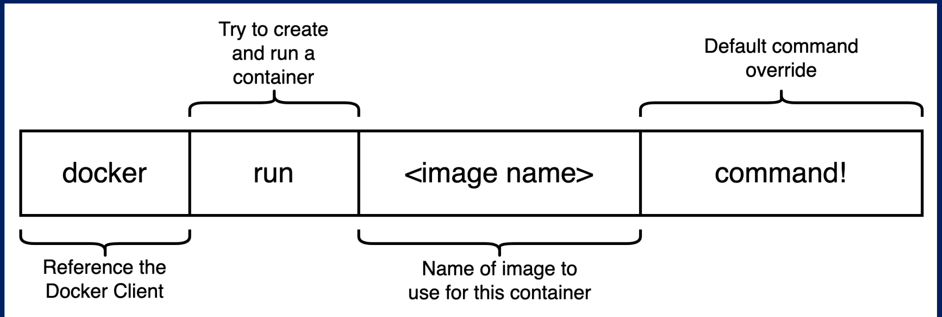
```
814c8b675ca3: Already exists
Digest:
sha256:c118f538365369207c12e5794c3cbfb7b042d950af590ae6c287ede74f29b7d4
Status: Downloaded newer image for busybox:latest
hi there
```




Try to create and run a container
Default command override
docker
run
<image name>
command!
Reference the Docker Client
Name of image to use for this container



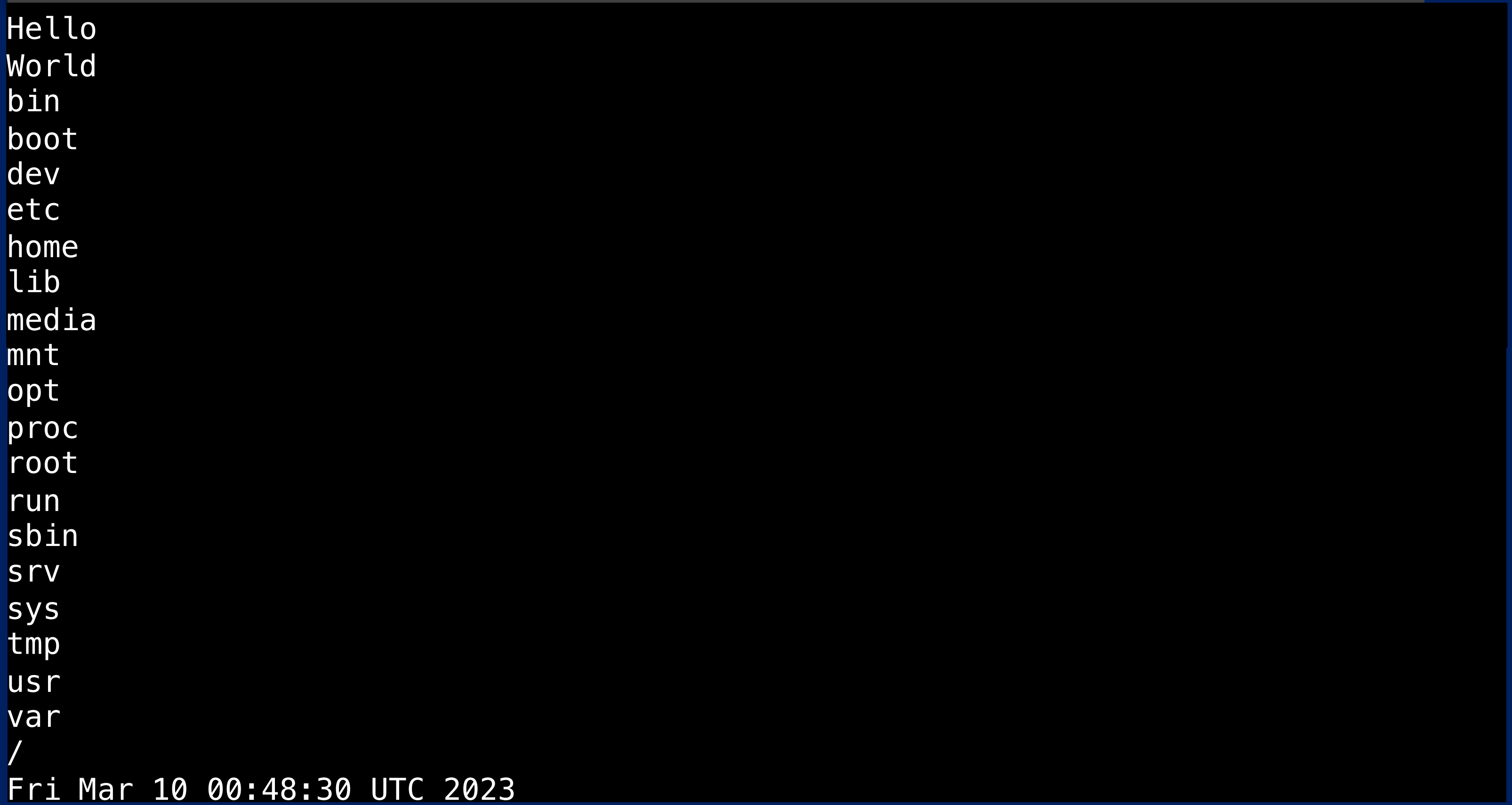
Append a command


Try to create and run a container
Default command override
docker
run
<image name>
command!
Reference the Docker Client
Name of image to use for this container

```
docker run ubuntu
```

```
docker run ubuntu sleep 5
```

```
docker run  ubuntu sh -c "echo 'Hello' && echo 'World' && ls && pwd && date"
```



```
Hello
World
bin
boot
dev
etc
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
/
Fri Mar 10 00:48:30 UTC 2023
```

Pull – download an image

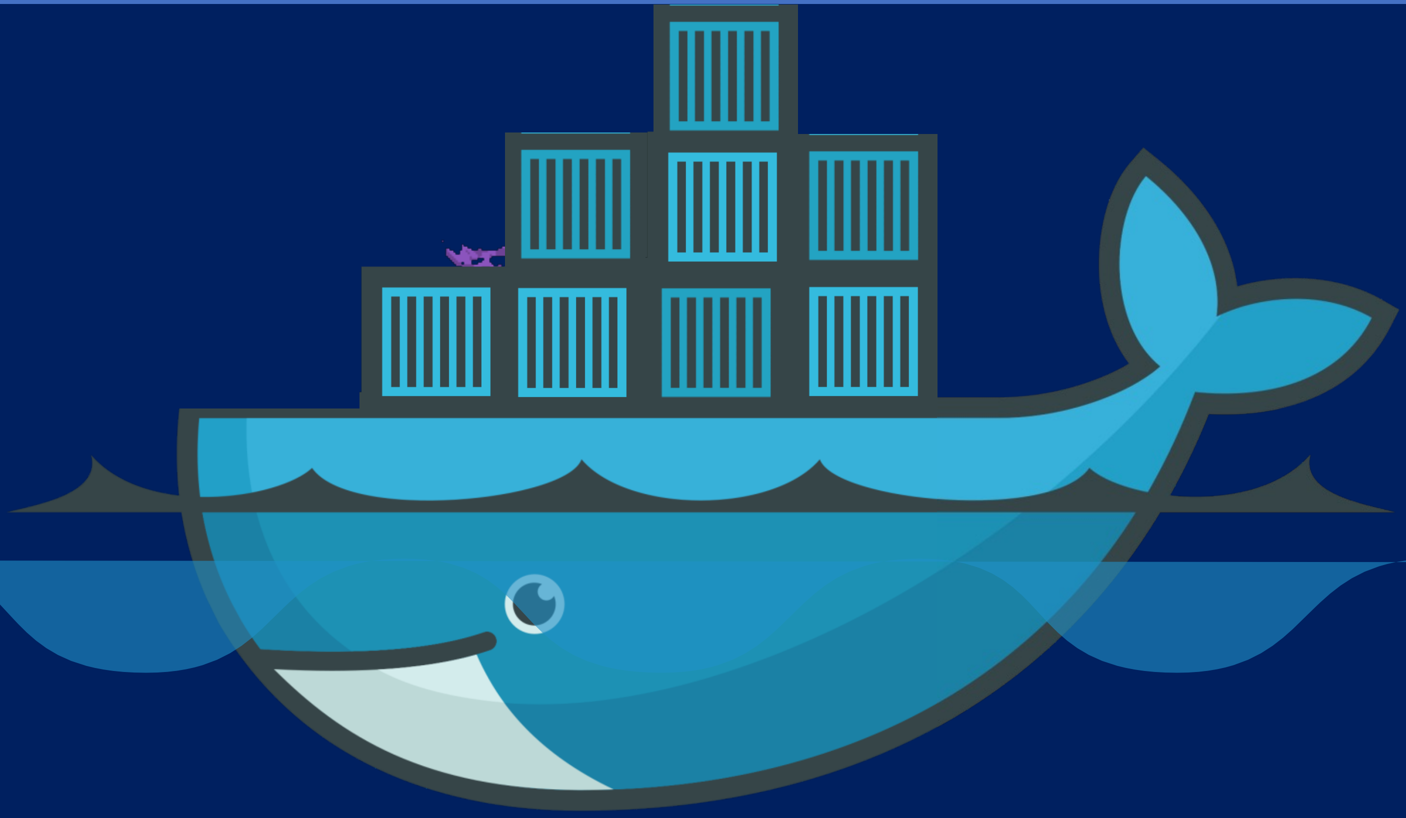
```
docker run nginx
```

```
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 pull nginx
```

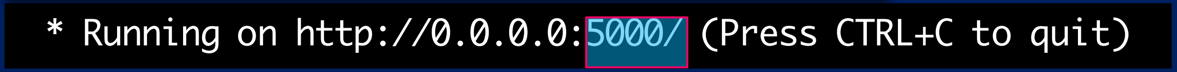
```
Using default tag: latest
latest: Pulling from library/nginx
fc7181108d40: Pull complete
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
Status: Downloaded newer image for nginx:latest
```

Docker Mapping



Run – PORT mapping

```
docker run myname/webapp
```

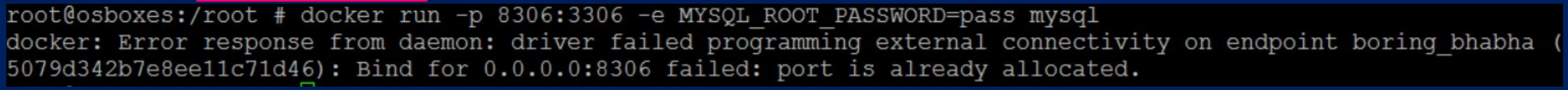


```
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

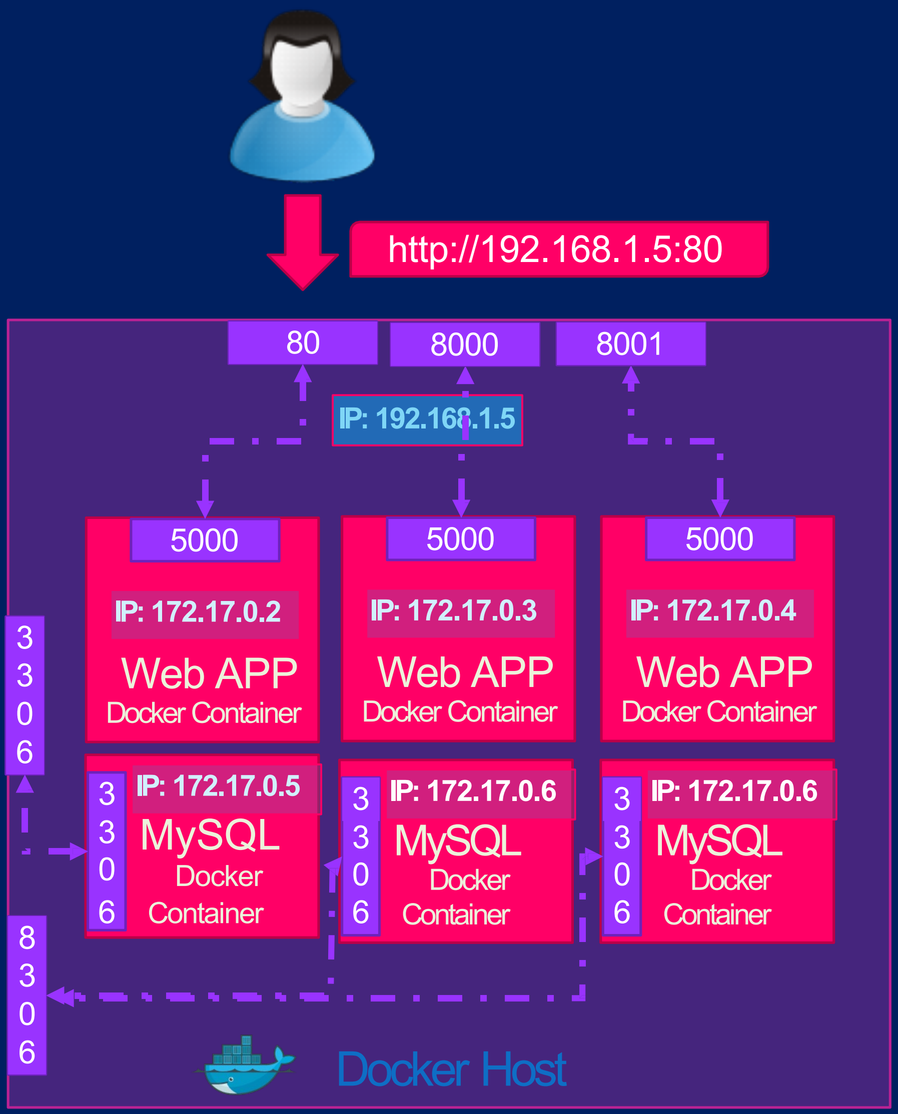
http://172.17.0.2:5000

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.
```

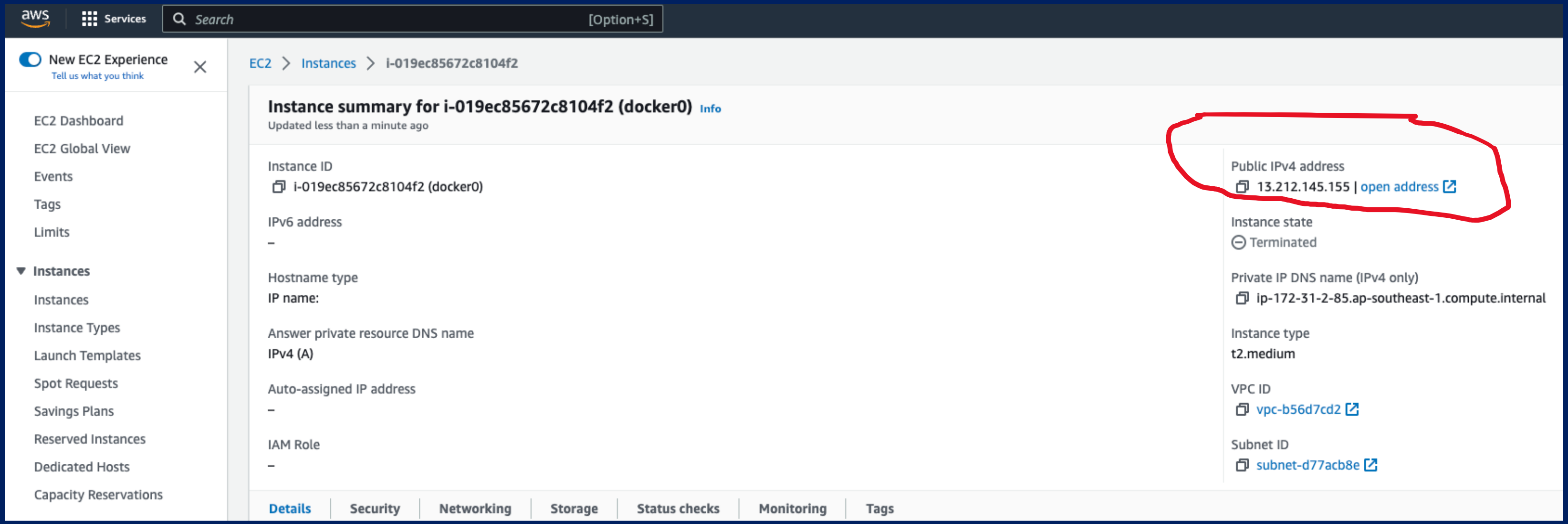


LAB 1 : Run Nginx with port mapping

```
▶ 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
```



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

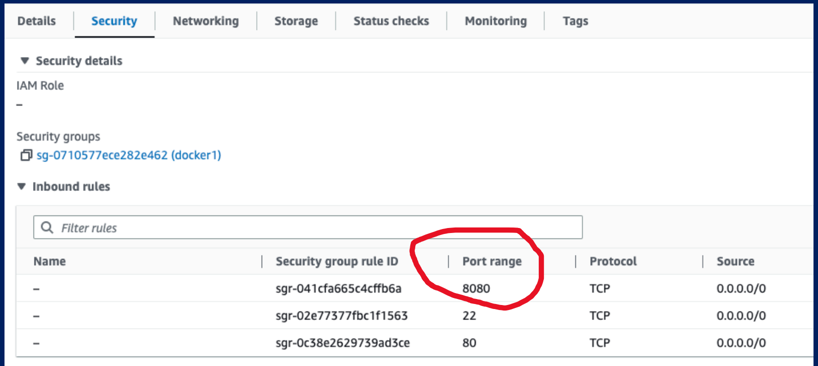
Instance state
Terminated

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



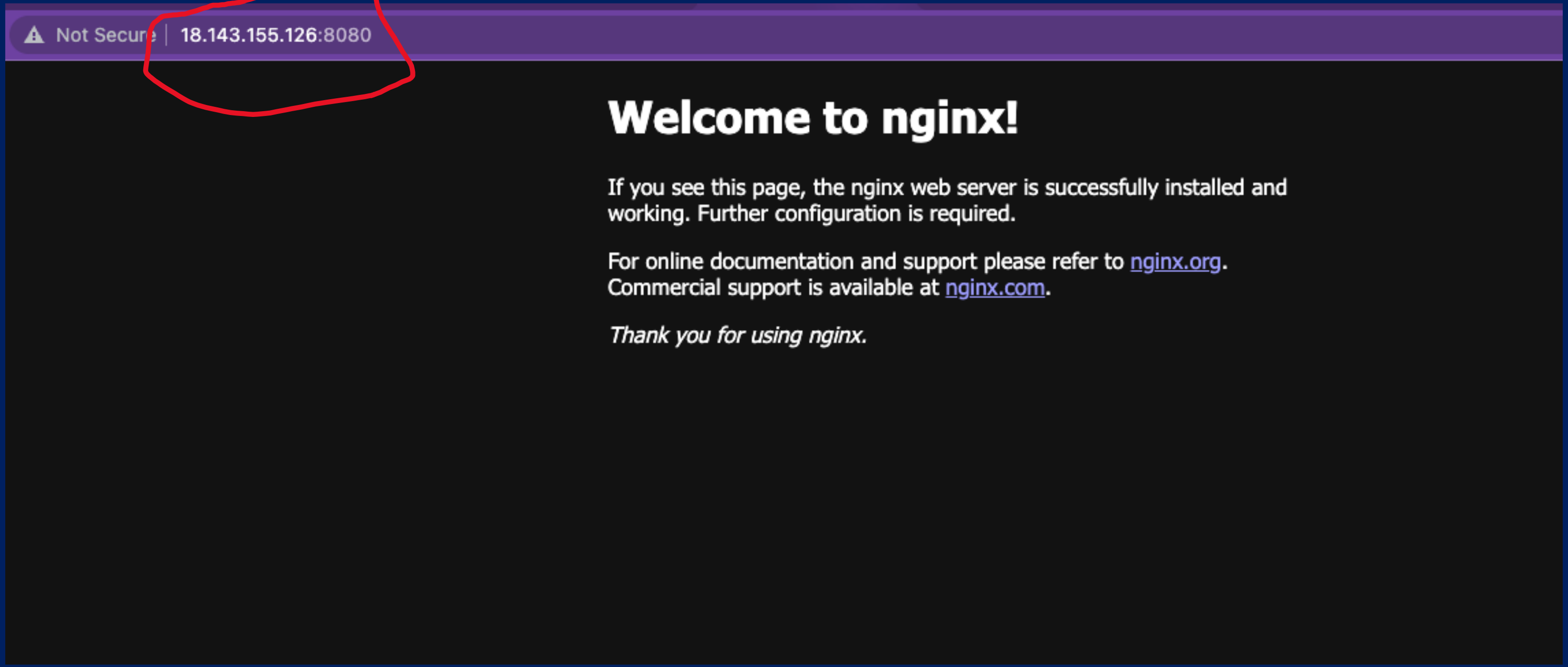
Security details

IAM Role
–

Security groups
sg-0710577ece282e462 (docker1)

Inbound 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



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.

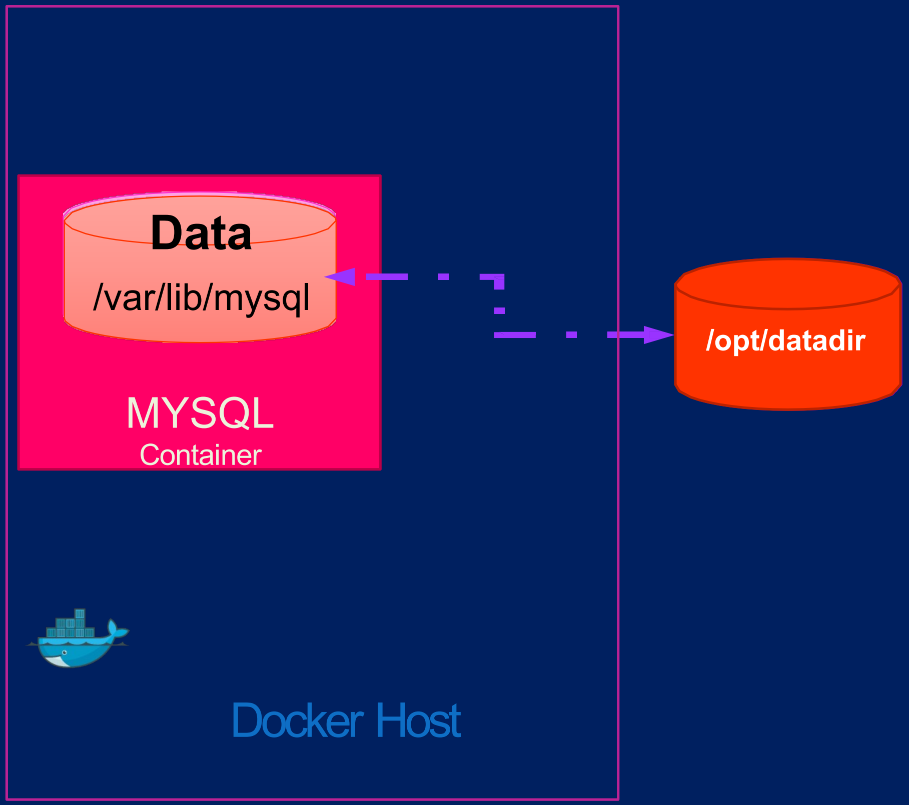
Thank you for using nginx.

RUN – Volume mapping

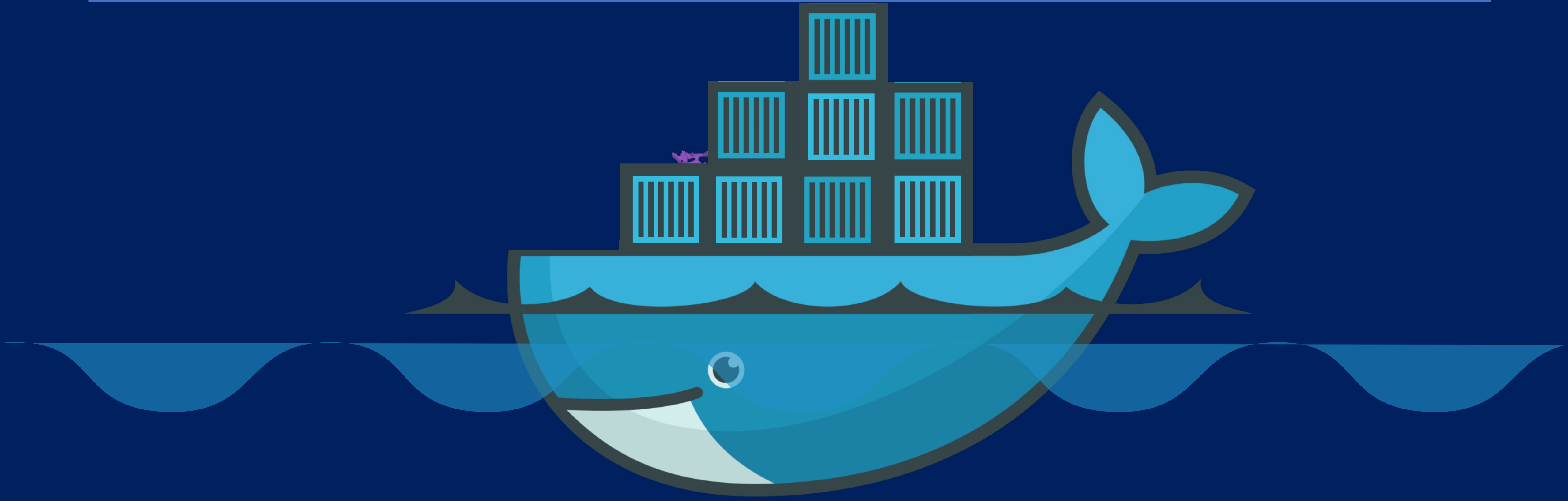
```
docker run mysql
```

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

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

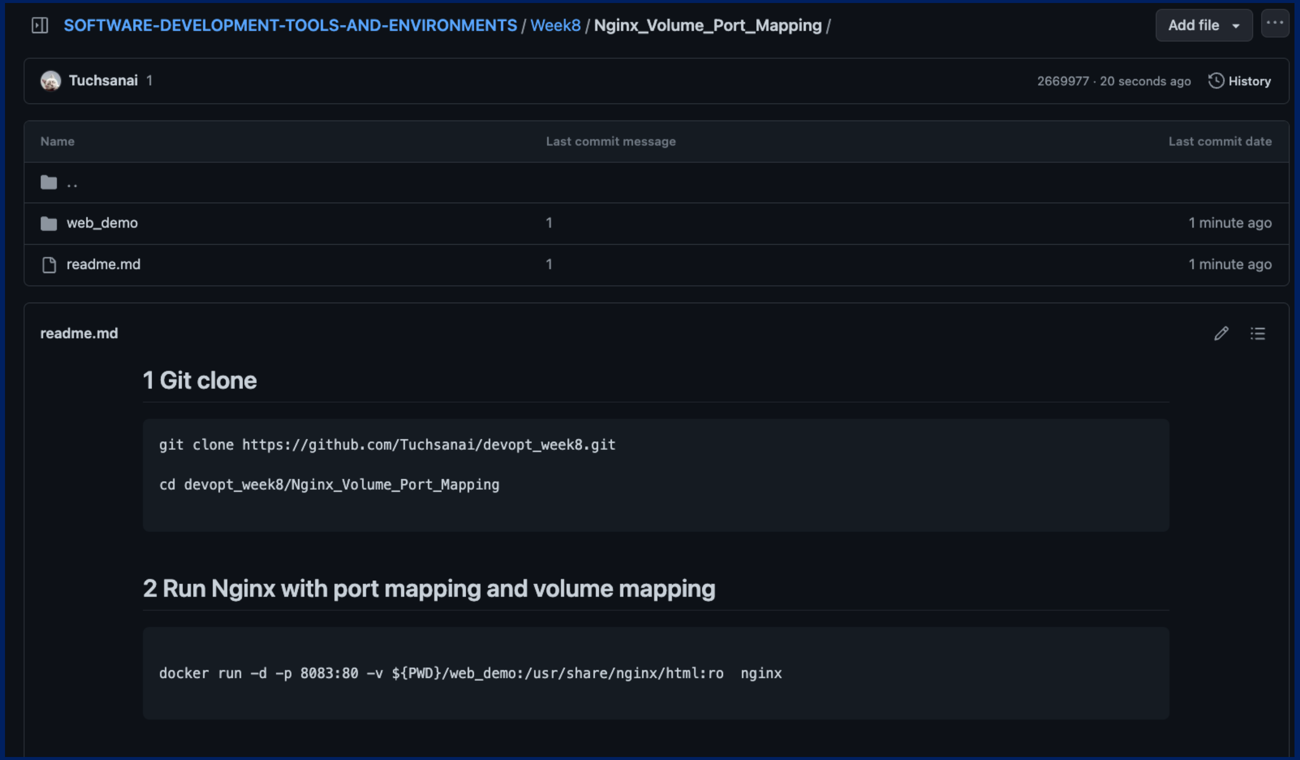


Docker run from Repository



LAB2 :Run Nginx with Volume and Port Mapping

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



SOFTWARE-DEVELOPMENT-TOOLS-AND-ENVIRONMENTS / Week8 / Nginx_Volume_Port_Mapping /

Tuchsanai 1 2669977 · 20 seconds ago History

Name	Last commit message	Last commit date
..		
web_demo	1	1 minute ago
readme.md	1	1 minute ago

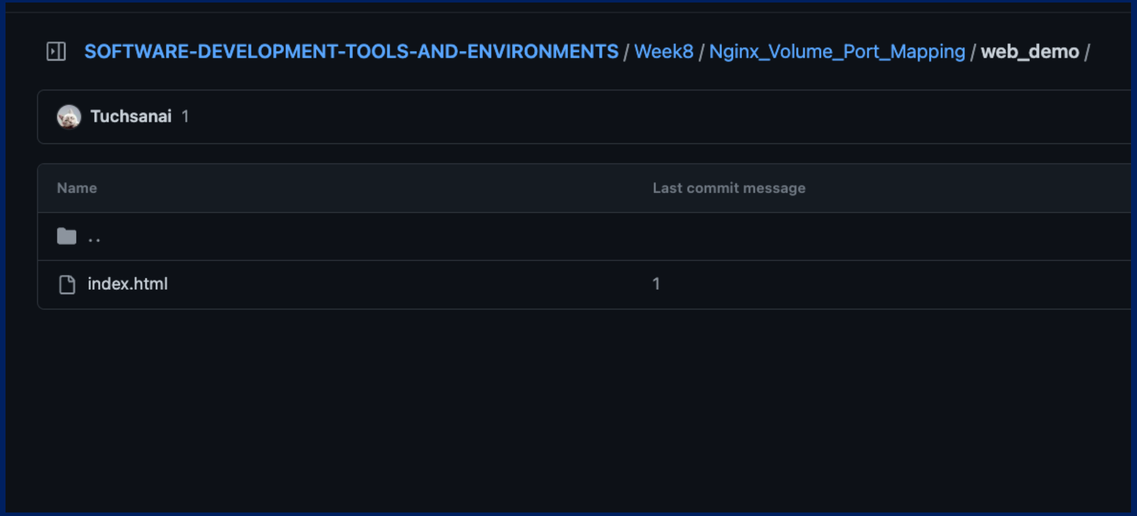
readme.md

1 Git clone

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

cd devopt_week8/Nginx_Volume_Port_Mapping
```

2 Run Nginx with port mapping and volume mapping

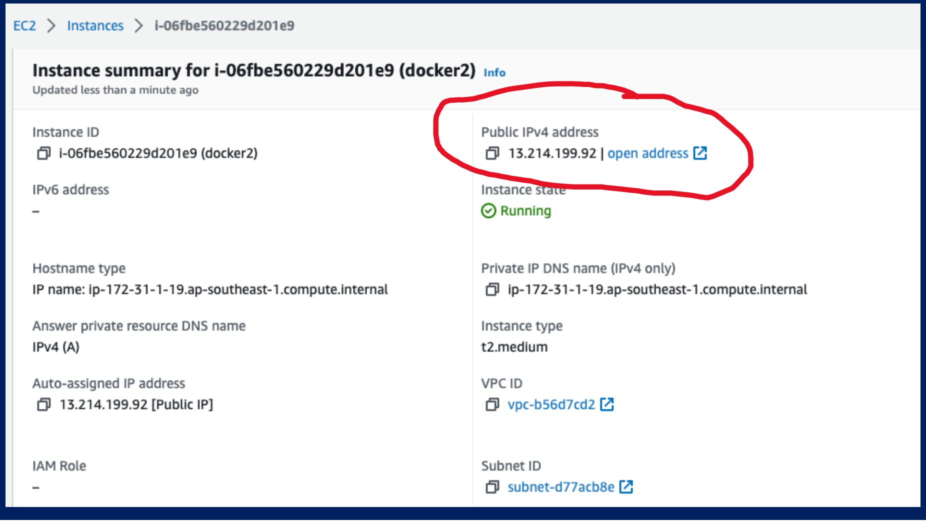
```
docker run -d -p 8083:80 -v ${PWD}/web_demo:/usr/share/nginx/html:ro  nginx
```



SOFTWARE-DEVELOPMENT-TOOLS-AND-ENVIRONMENTS / Week8 / Nginx_Volume_Port_Mapping / web_demo /

Tuchsanai 1

Name	Last commit message
..	
index.html	1

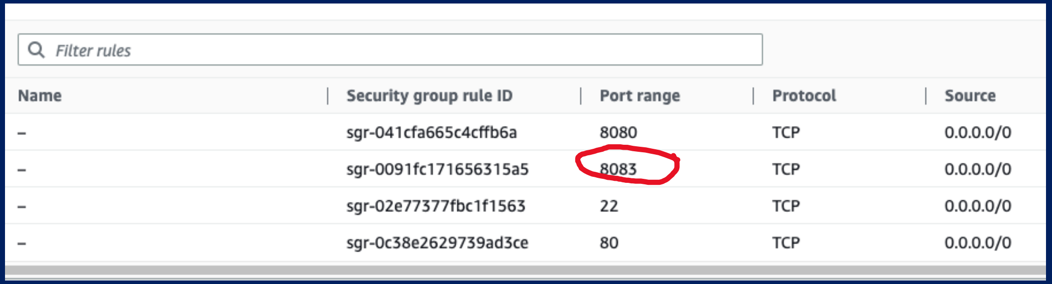


EC2 > Instances > i-06fbe560229d201e9

Instance summary for i-06fbe560229d201e9 (docker2) Info

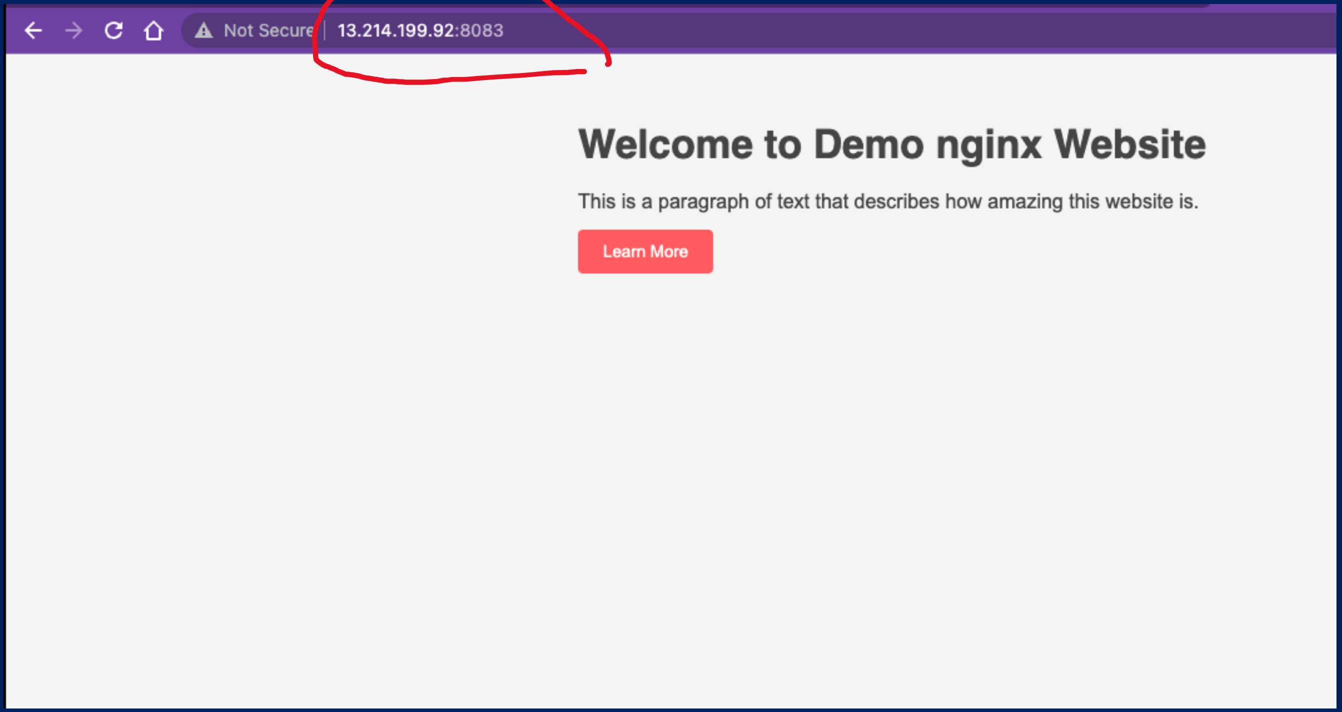
Updated less than a minute ago

Instance ID i-06fbe560229d201e9 (docker2)	Public IPv4 address 13.214.199.92 | open address
IPv6 address –	Instance state Running
Hostname type IP name: ip-172-31-1-19.ap-southeast-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-1-19.ap-southeast-1.compute.internal
Answer private resource DNS name IPv4 (A)	Instance type t2.medium
Auto-assigned IP address 13.214.199.92 [Public IP]	VPC ID vpc-b56d7cd2
IAM Role –	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

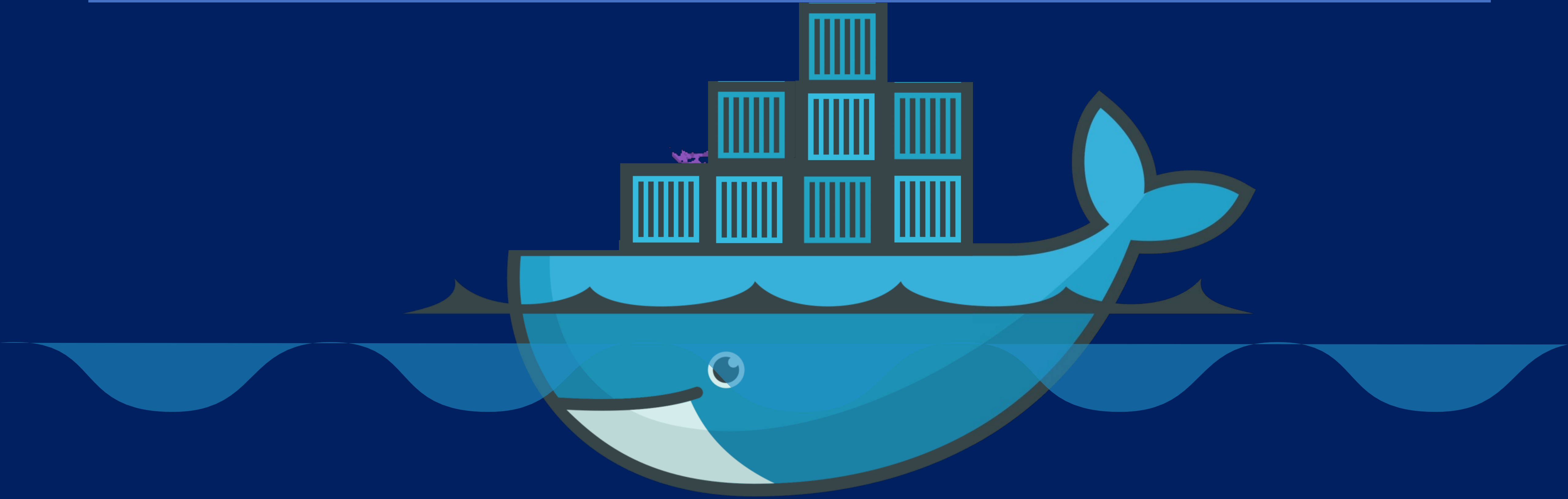


Not Secure | 13.214.199.92:8083

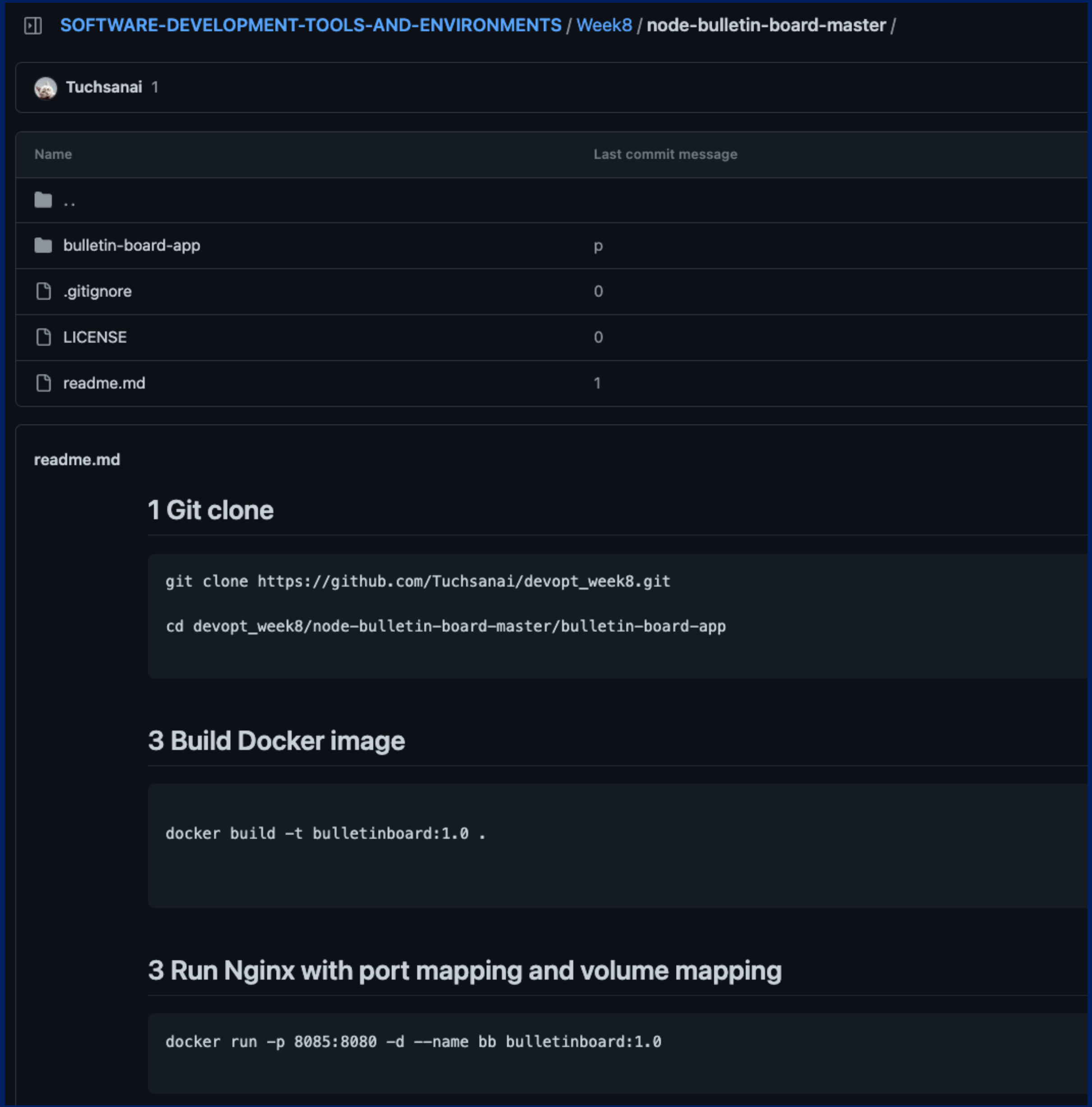
Welcome to Demo nginx Website

This is a paragraph of text that describes how amazing this website is.

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

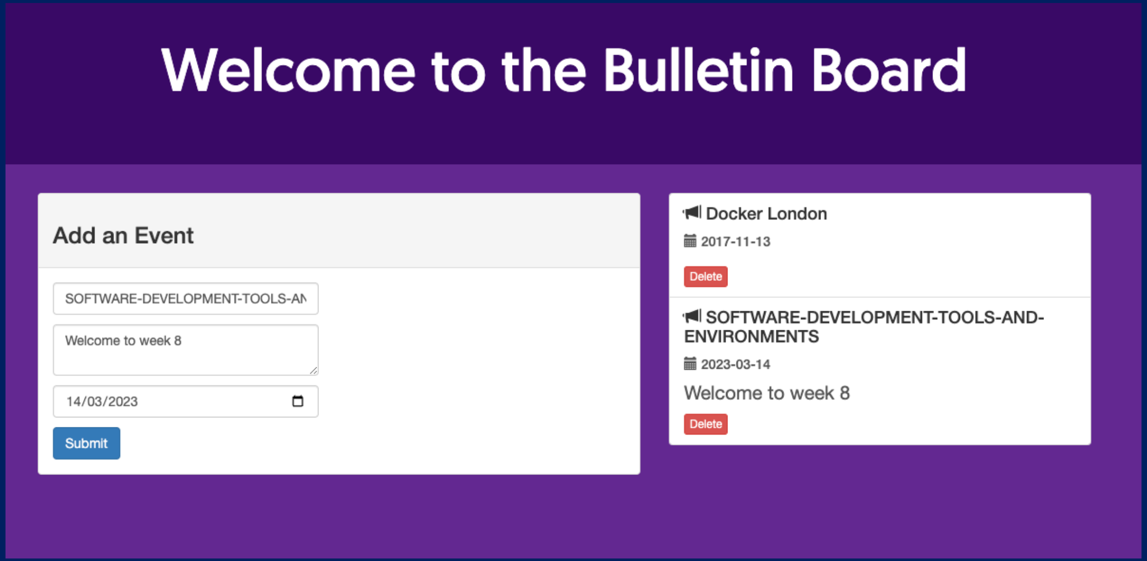
```
git clone https://github.com/Tuchsanai/devopt_week8.git

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
```



Welcome to the Bulletin Board

Add an Event

SOFTWARE-DEVELOPMENT-TOOLS-AN

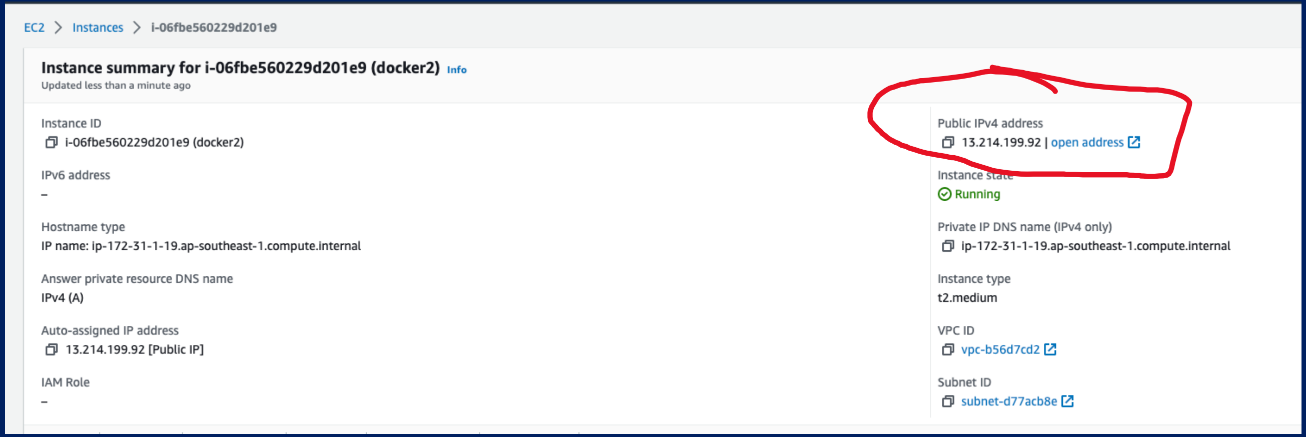
Welcome to week 8

14/03/2023

Submit

Docker London
2017-11-13
Delete

SOFTWARE-DEVELOPMENT-TOOLS-AND-ENVIRONMENTS
2023-03-14
Welcome to week 8
Delete



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

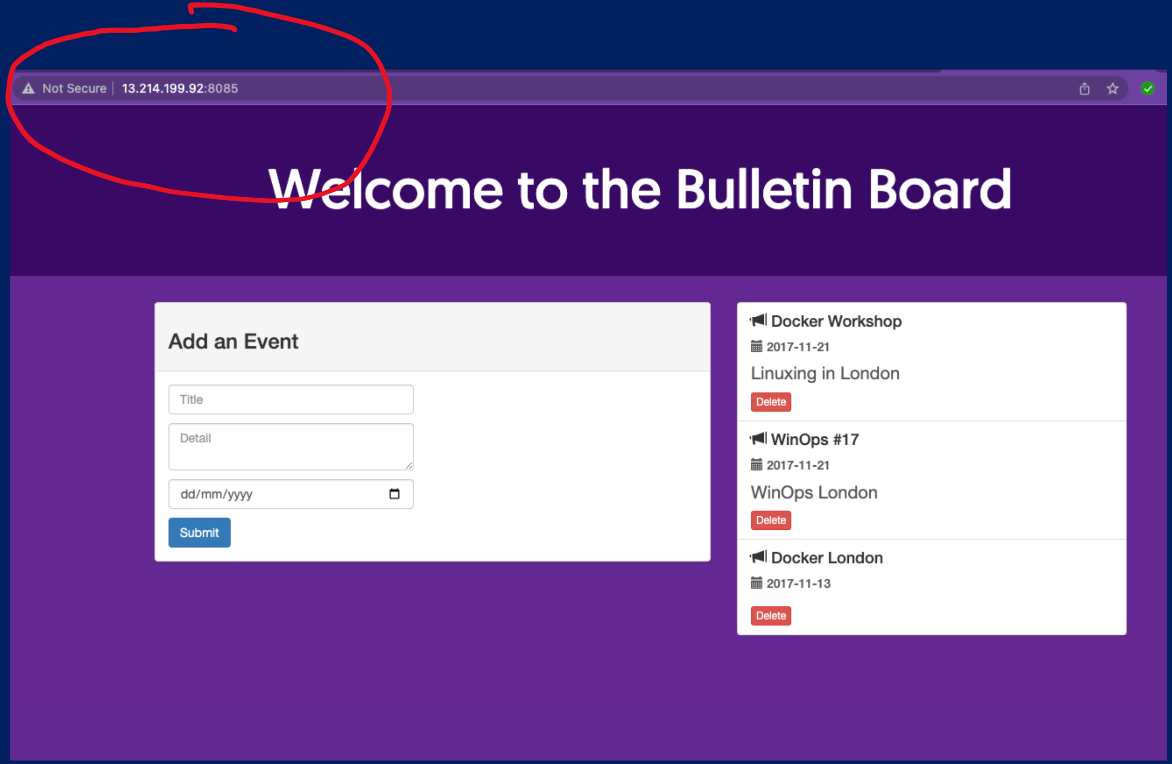
Instance state
Running

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



Not Secure | 13.214.199.92:8085

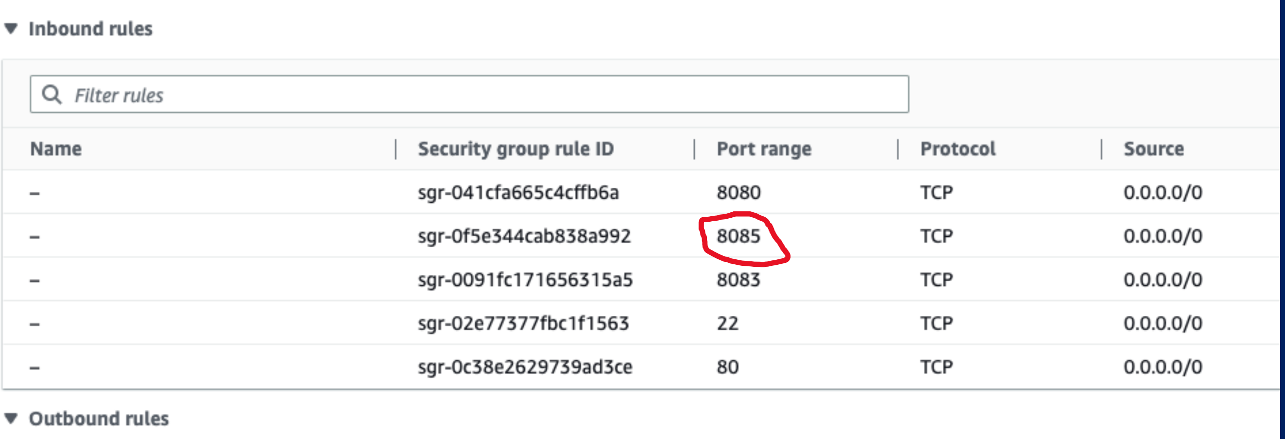
Welcome to the Bulletin Board

Add an Event

Docker Workshop
2017-11-21
Linuxing in London

WinOps #17
2017-11-21
WinOps London

Docker London
2017-11-13



▼ **Inbound 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**