

# DEEP LEARNING FOR COMPUTER VISION : Special

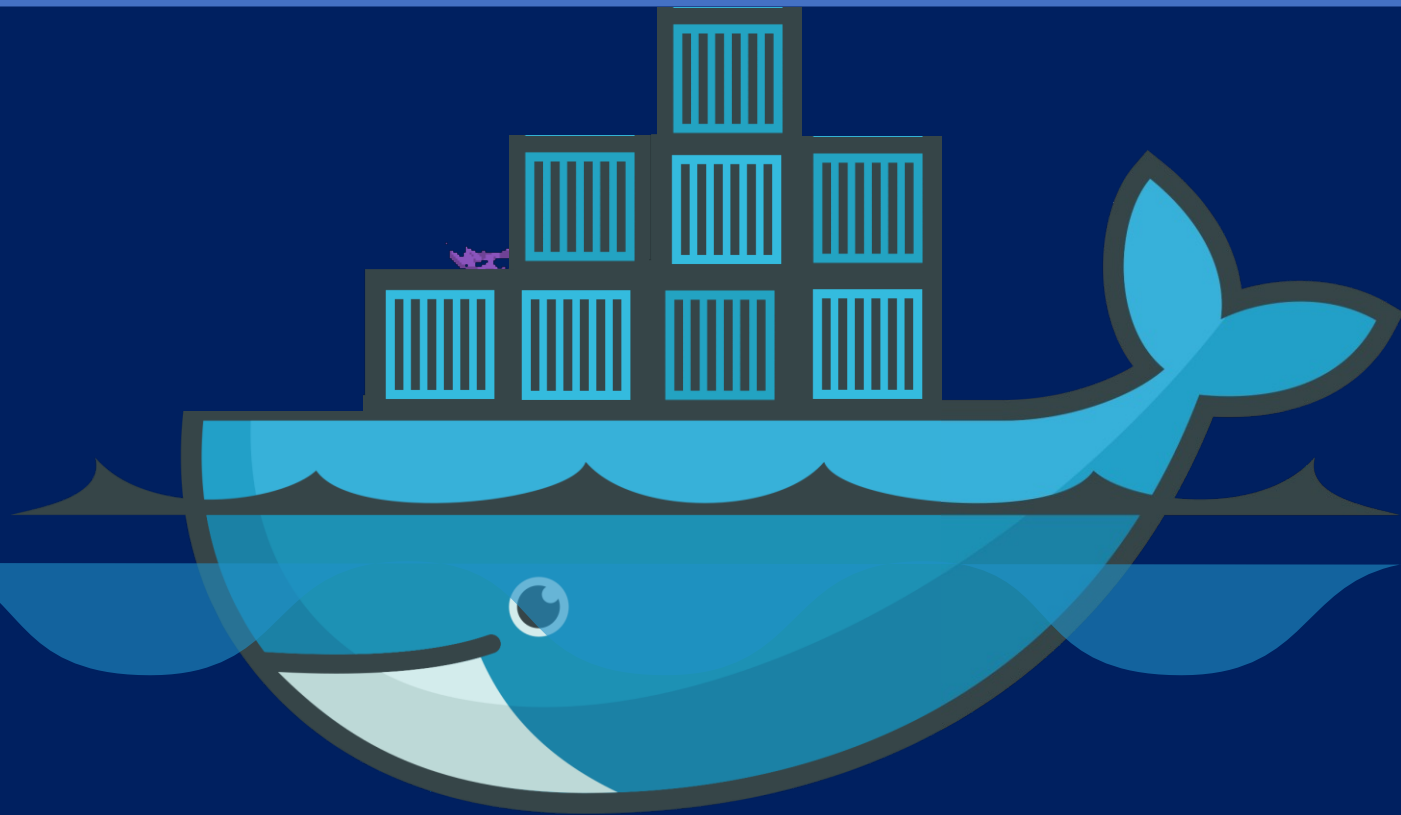
## Devops 1



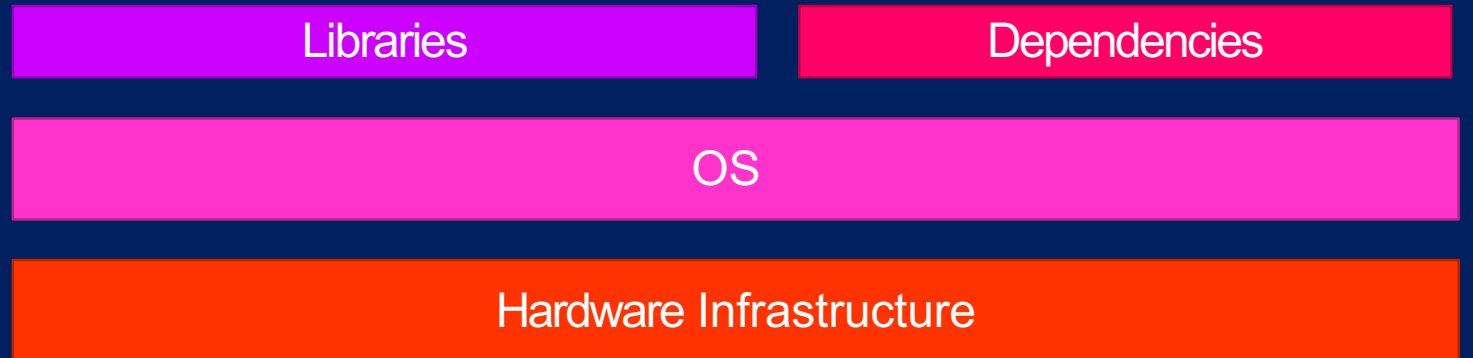
Dr. Tuchsanai. PloySuwan

# Docker Overview

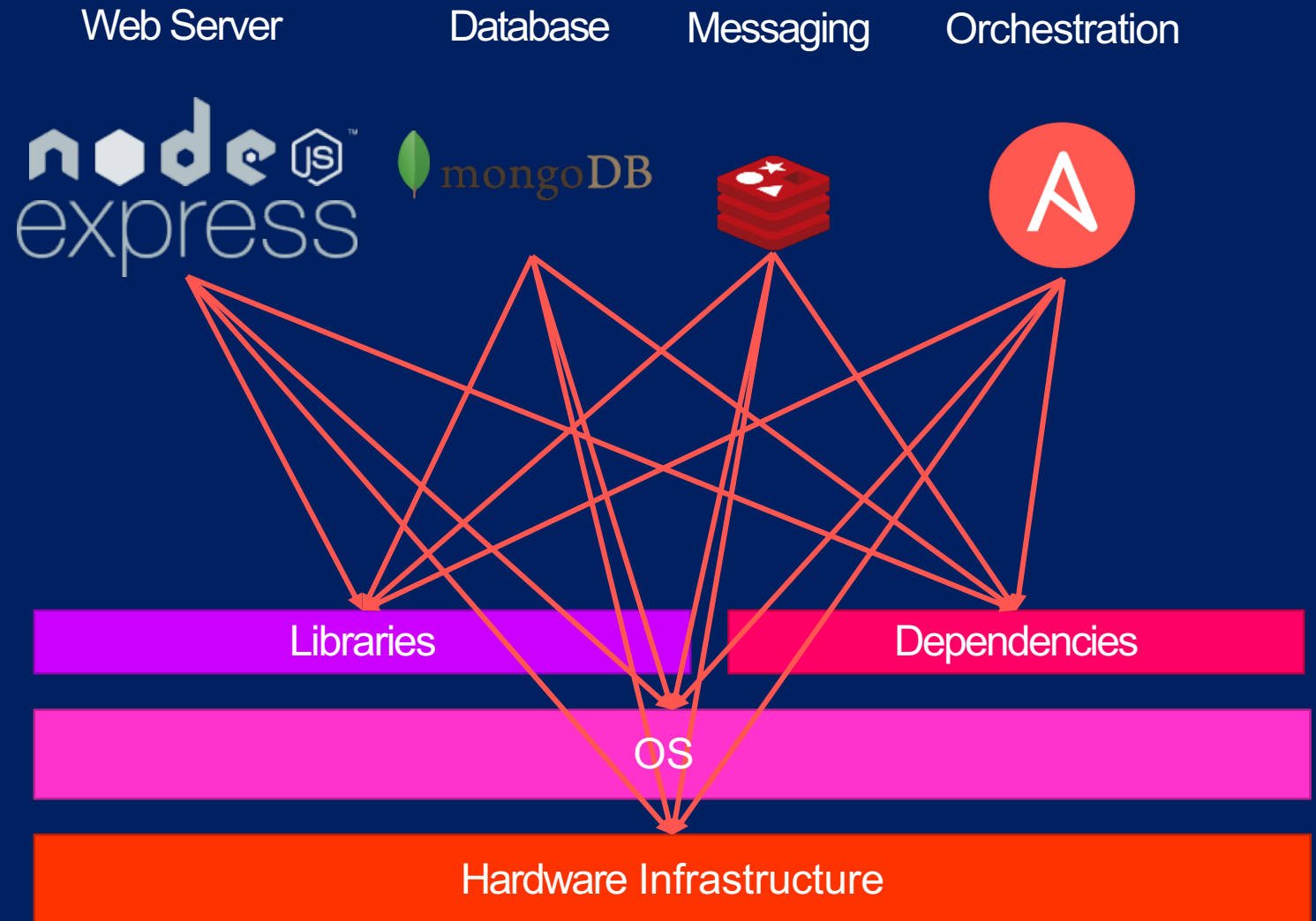
---



# Why do you need docker?

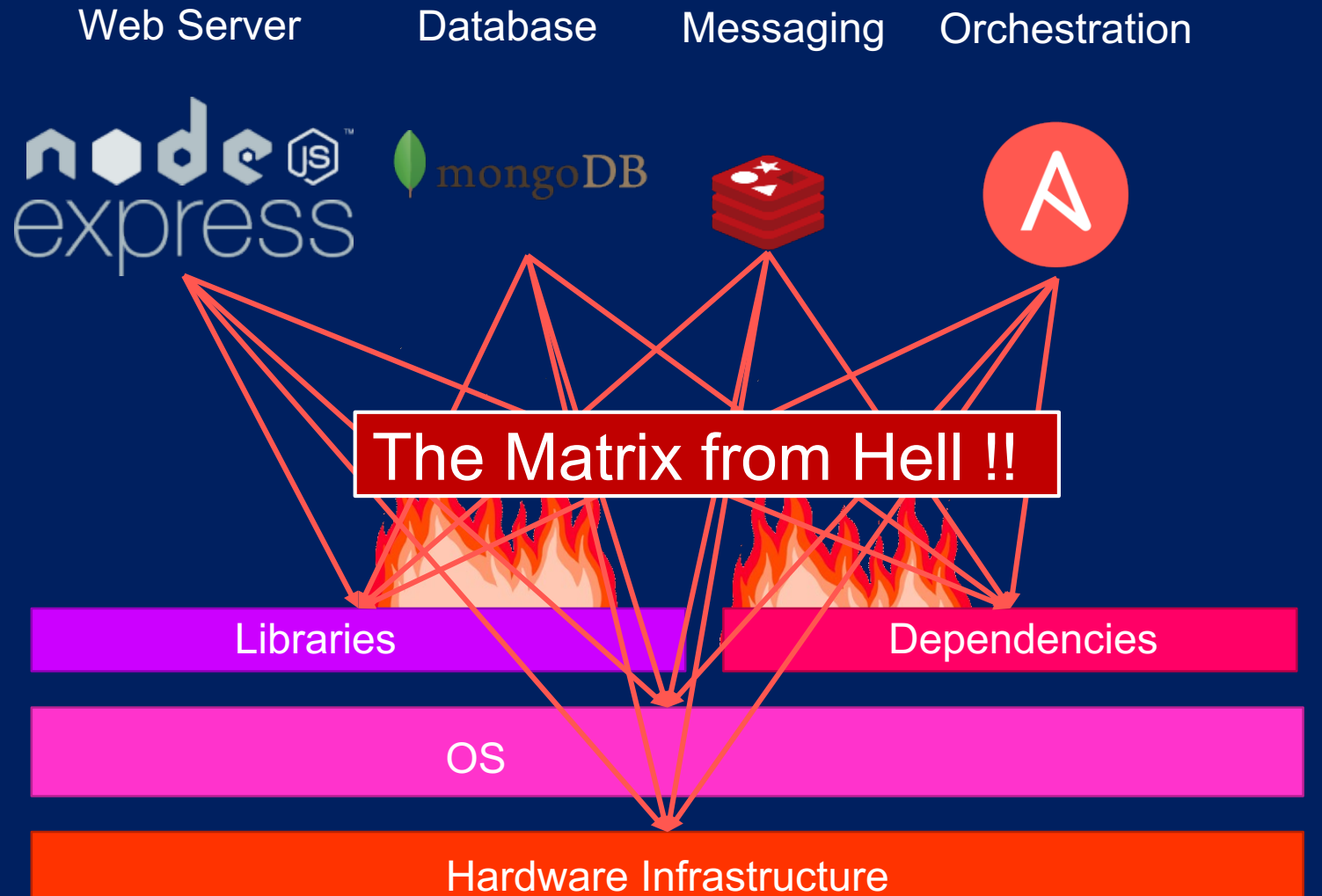


# Why do you need docker?



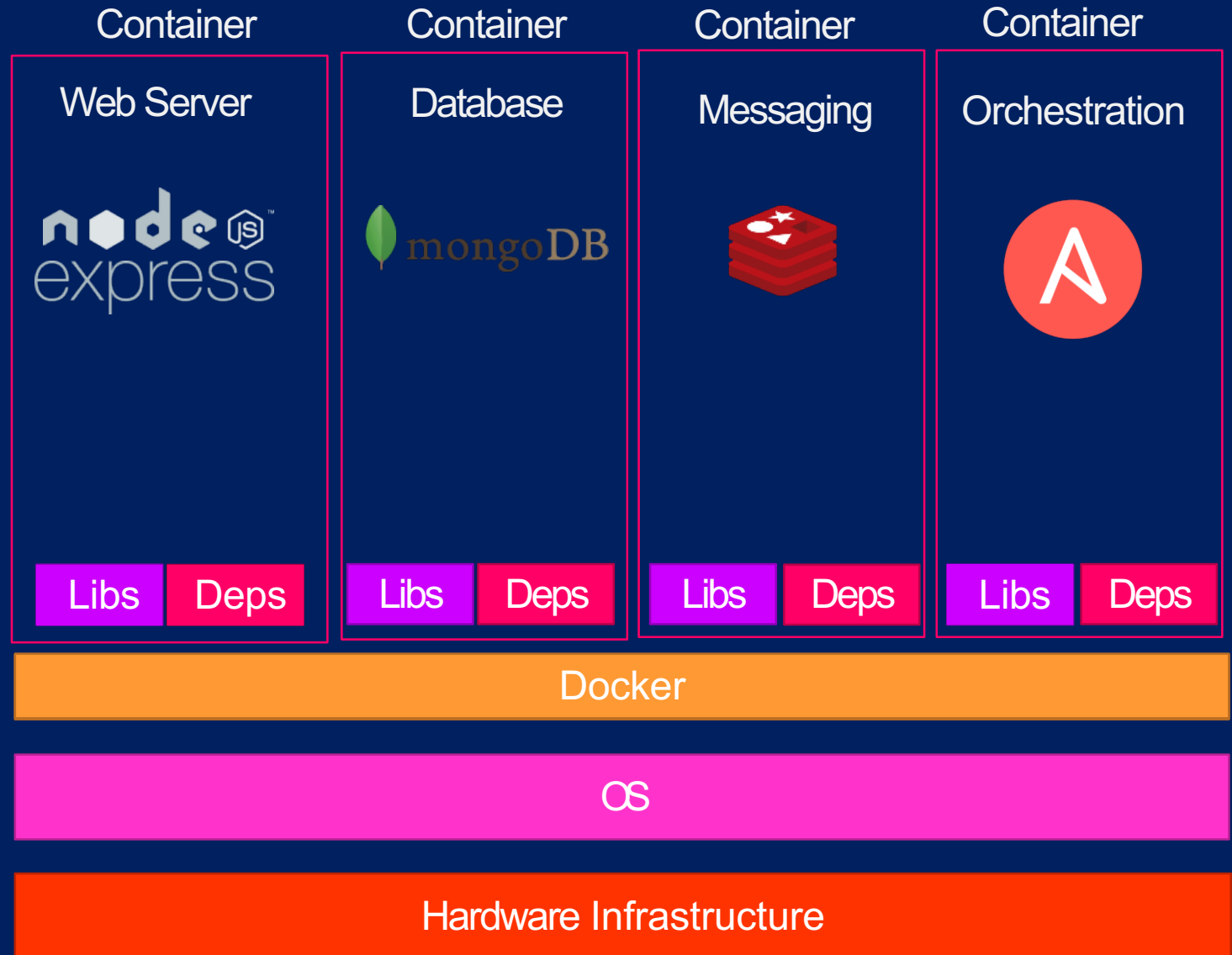
# Why do you need docker?

- 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



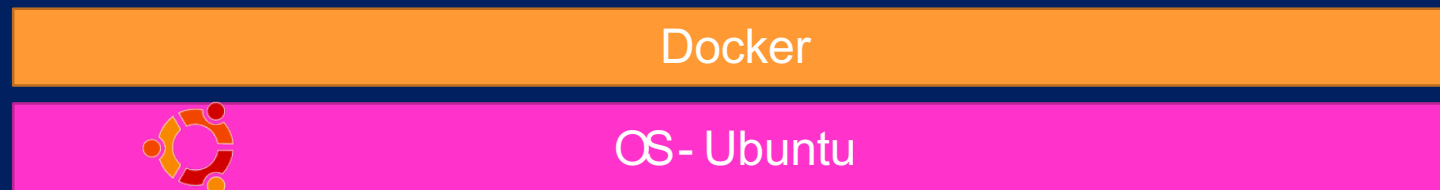
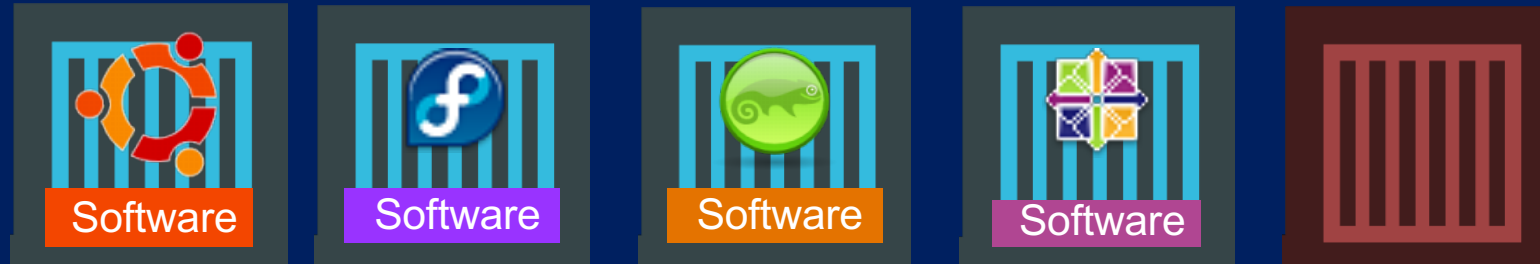
Processes  
Network  
Mounts

Docker

OS Kernel

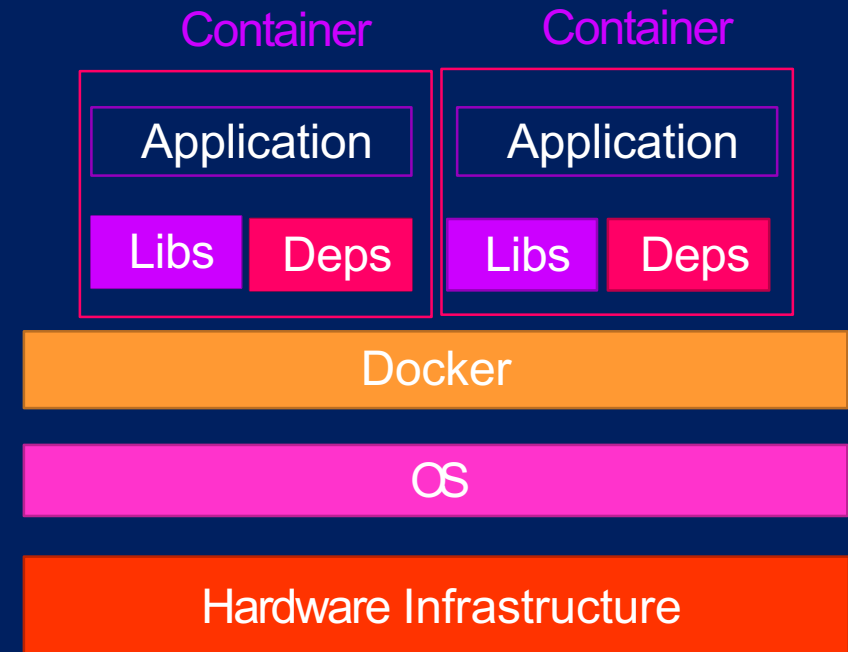
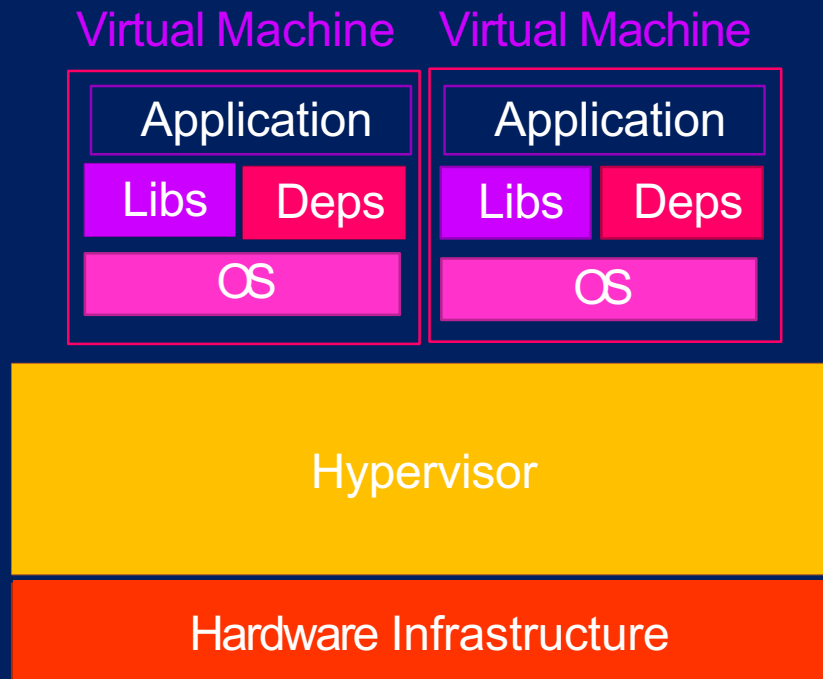
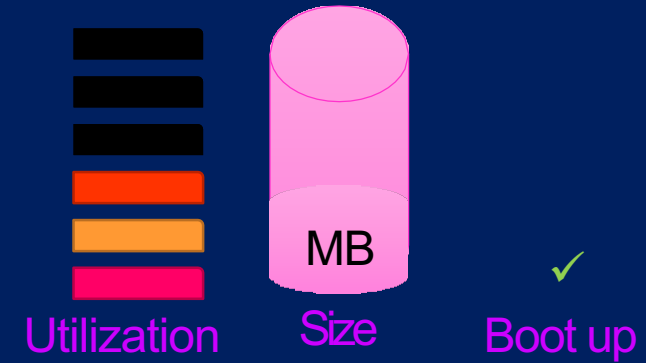
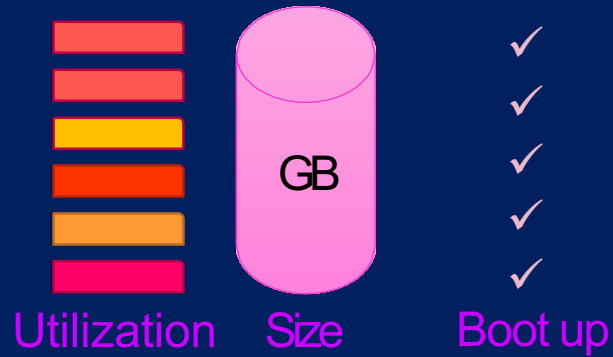


# Sharing the kernel for Operating System

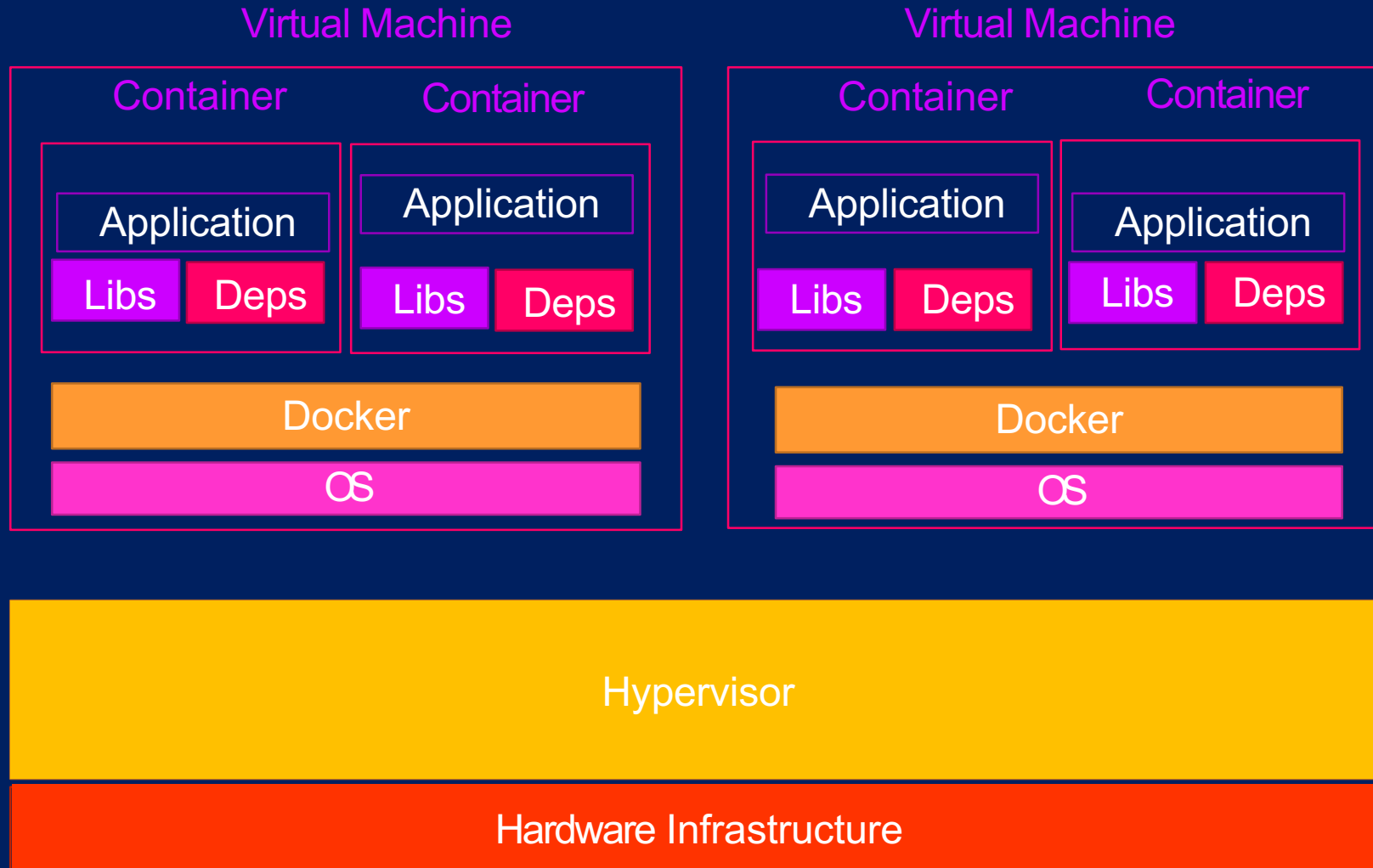




# Containers vs Virtual Machines

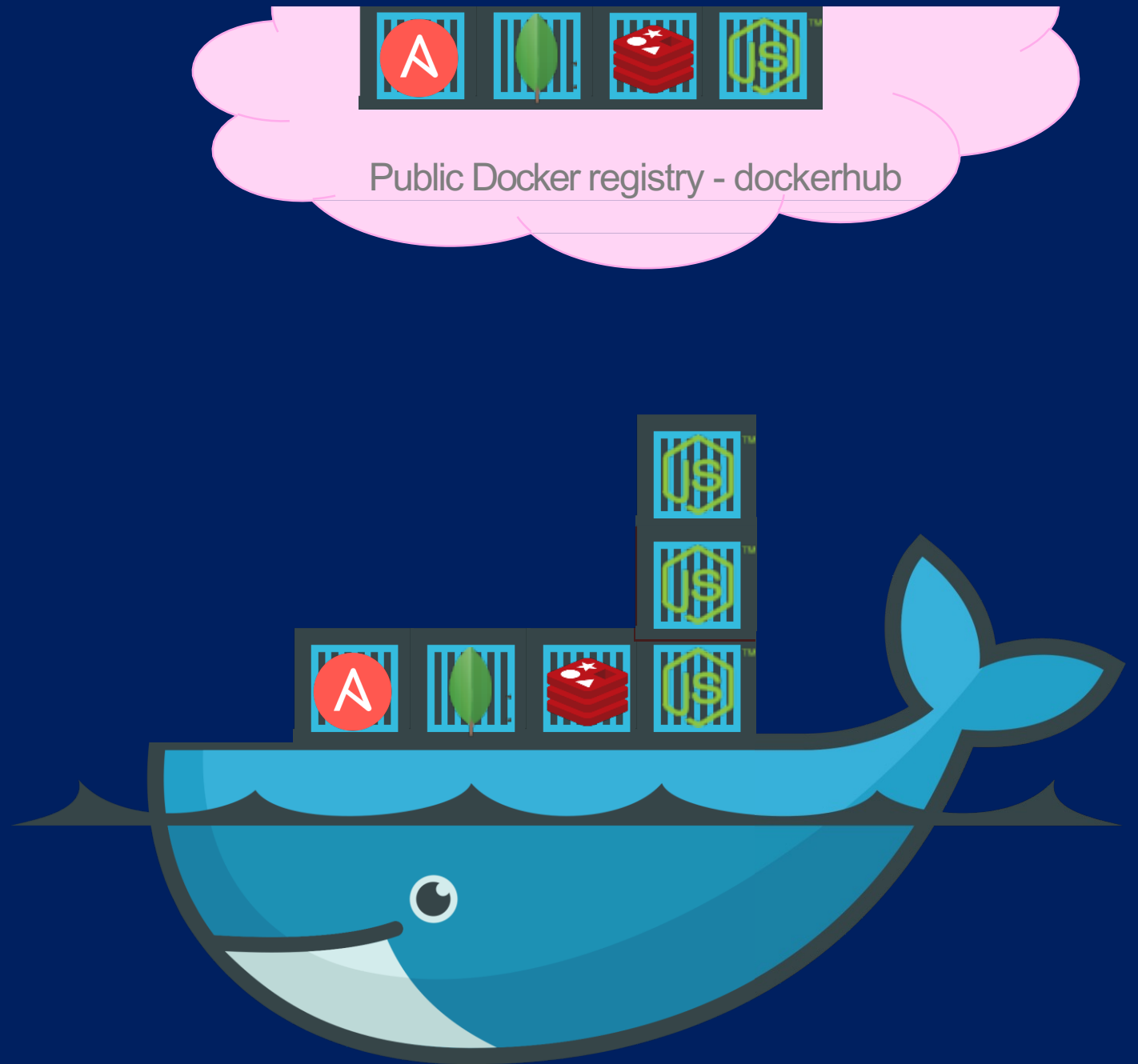


# Containers & Virtual Machines



# 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

`>_ docker`  
Docker client



Docker engine

## Hosts

Local host

daemon

containers

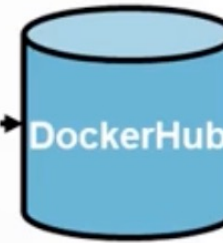
Remote host

daemon

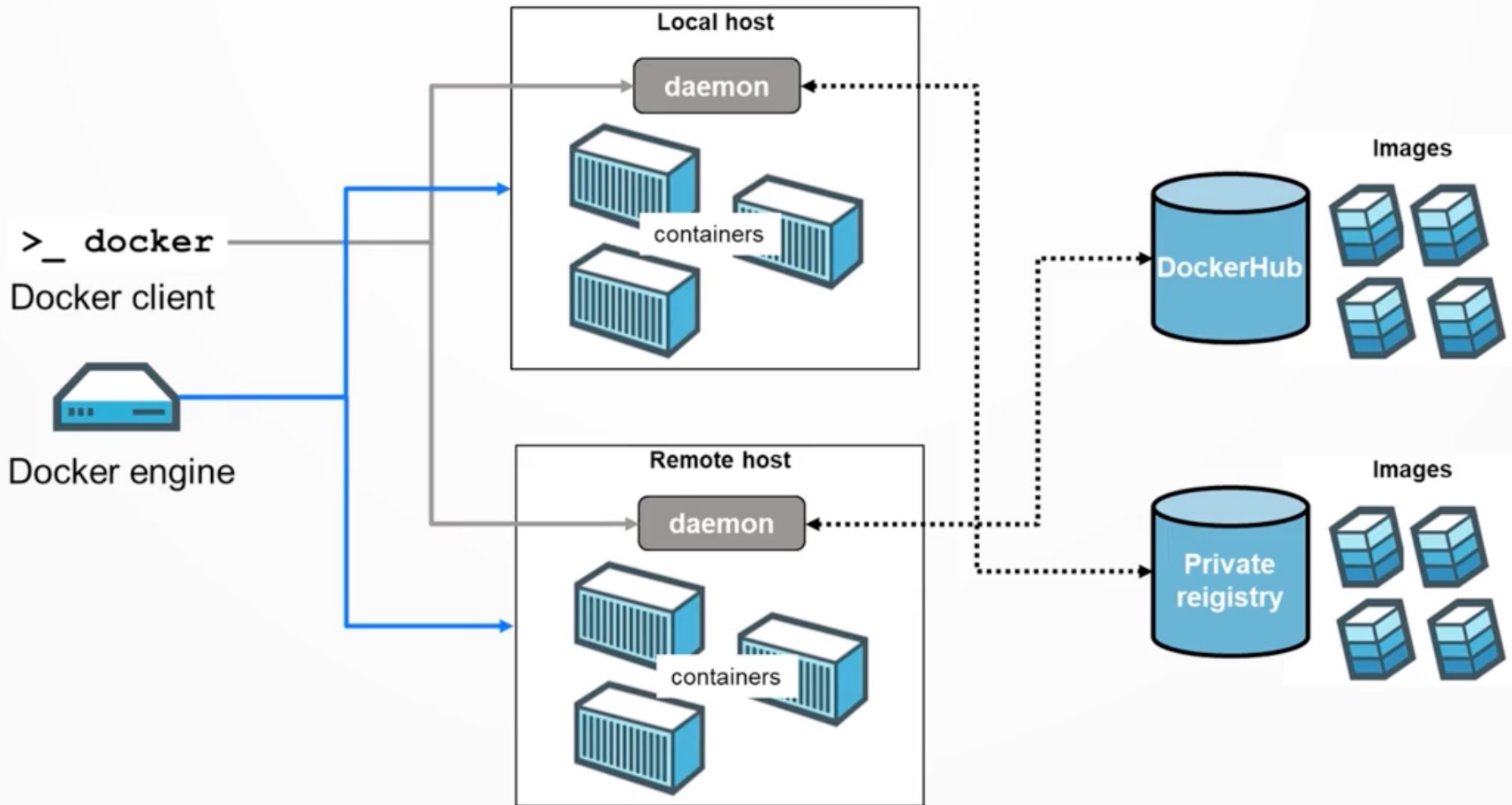
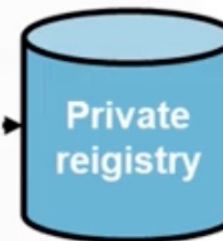
containers

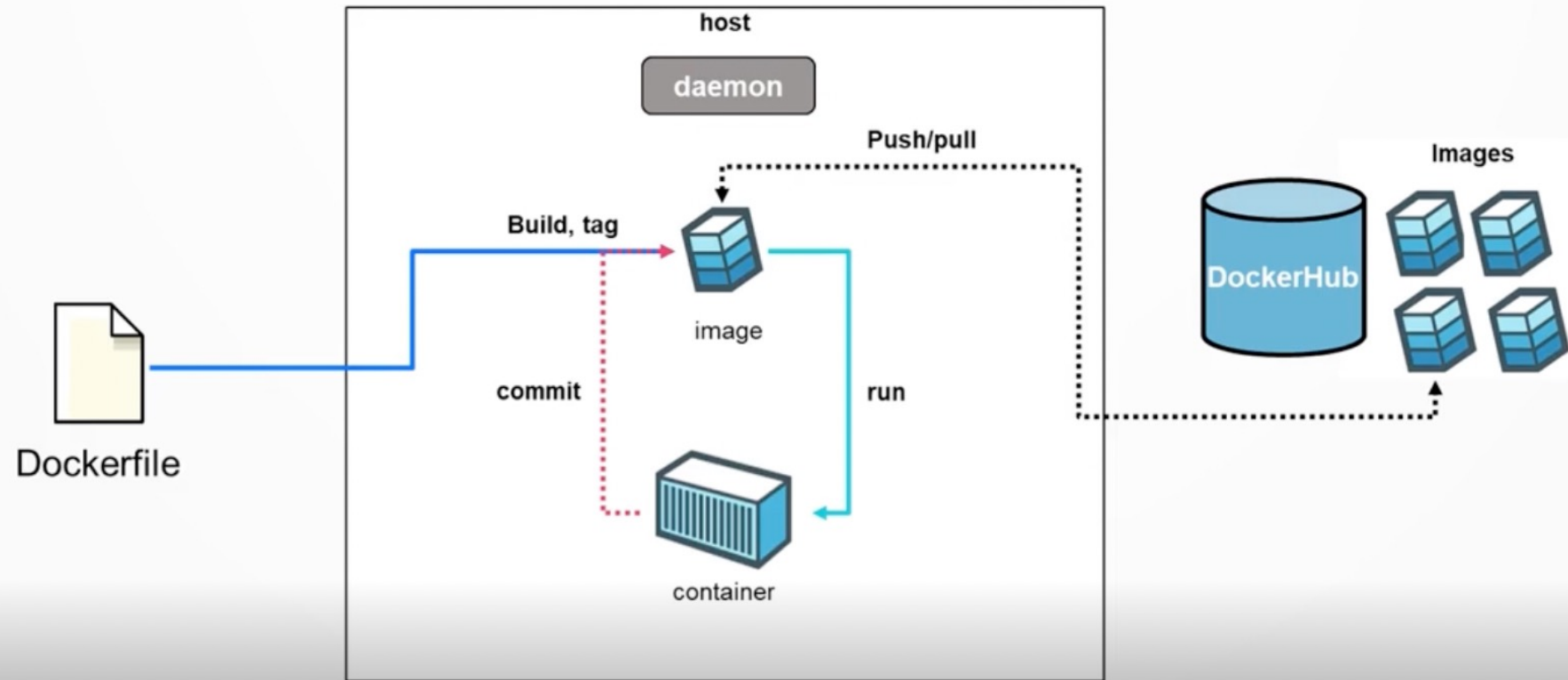
## Registries

Images



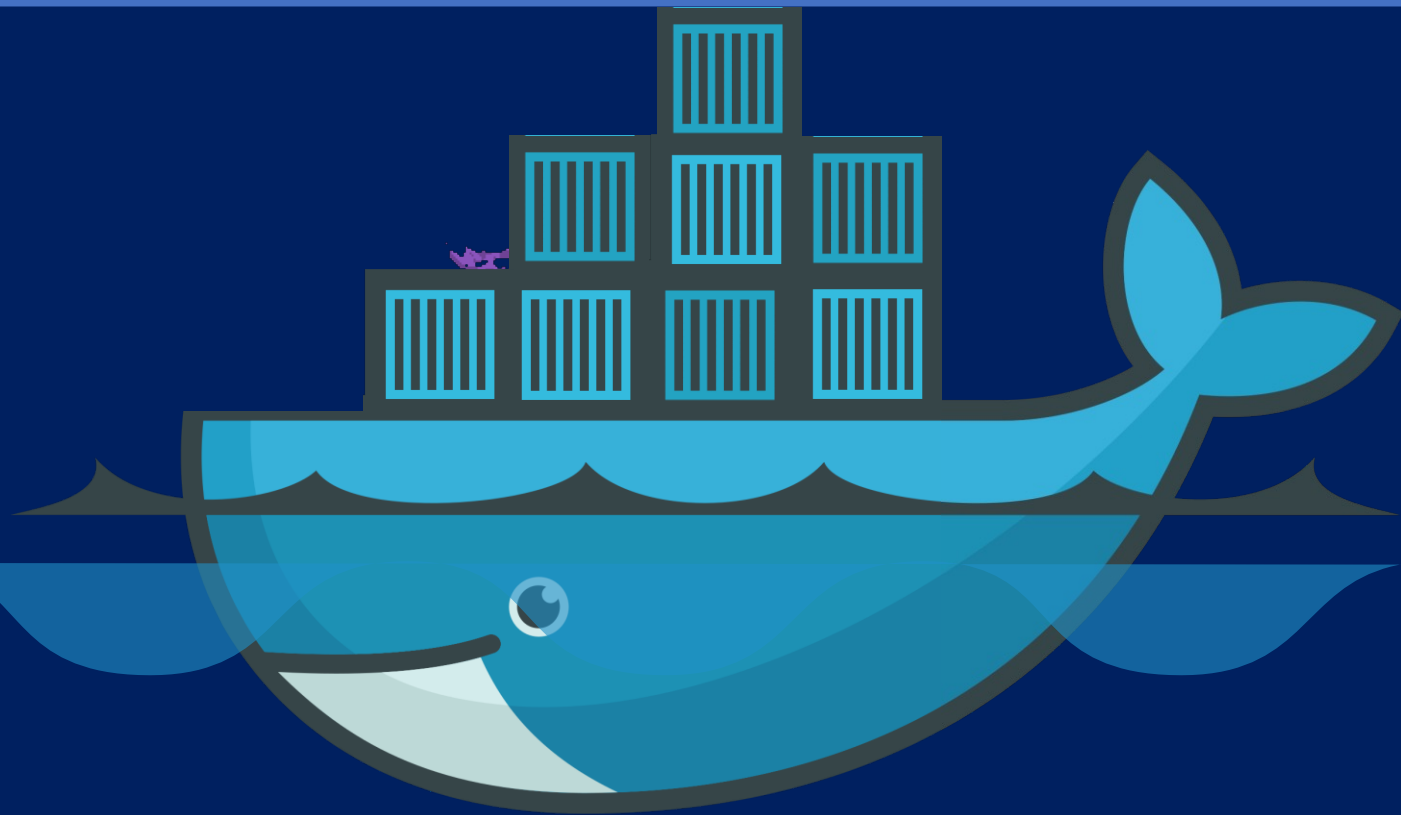
Images





# 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	<a href="#">Easy</a>
Root privileges	<a href="#">Yes</a>
Requirements	Linux terminal
Category	<a href="#">Package Manager</a>
Prerequisites	yum command
OS compatibility	Amazon Linux • <a href="#">Linux</a>
Est. reading time	6 minutes

ADVERTISEMENT

<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:

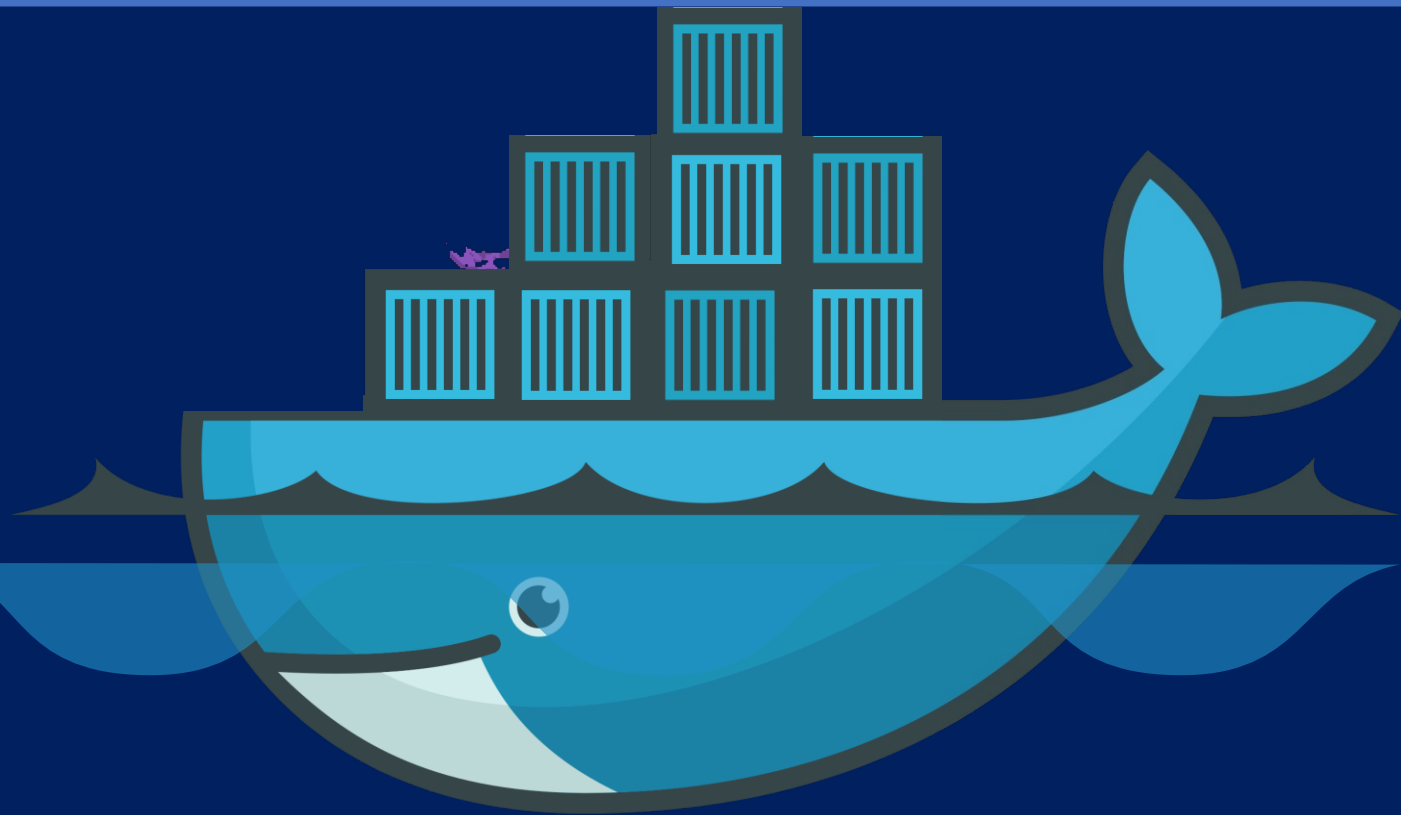
```
$ sudo systemctl enable docker
```

To restart Docker run:


```
$ sudo systemctl restart docker
```

# Docker Run

---



# Docker Registry



The screenshot shows the Docker Hub interface for the 'nginx' repository. The top navigation bar includes 'docker hub', a search bar with 'nginx', and links to 'Explore', 'Repositories', 'Organizations', and 'Help'. A user profile 'tuchsanai' is visible. The repository page for 'nginx' (Official build of Nginx) shows a 'docker pull nginx' button and a 'Tags' tab. The 'Tags' tab lists various image versions under 'Recent Tags', including 'stable-perl', 'stable', 'perl', 'mainline-perl', 'mainline', 'latest', '1.23.3-perl', '1.23.3', '1.23-perl', and '1.23'. A 'Quick reference' section on the left provides links to maintainers, help resources, and supported tags. The 'Supported tags and respective Dockerfile links' section lists various image variants like '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', and '1.22.1-alpine-perl', 'stable-alpine-perl', '1.22-alpine-perl'. A 'Quick reference (cont.)' section at the bottom provides a link to file issues.

**Quick reference**

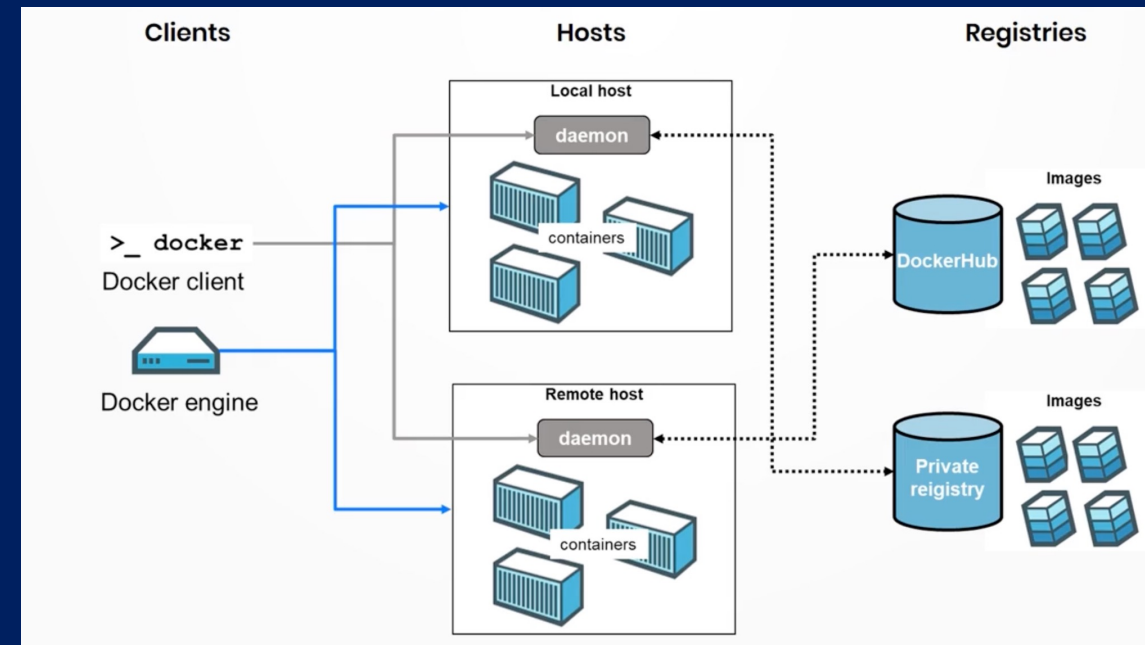
- Maintained by:  
the NGINX Docker Maintainers
- Where to get help:  
the Docker Community Slack, Server Fault, Unix & Linux, or Stack Overflow

**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>

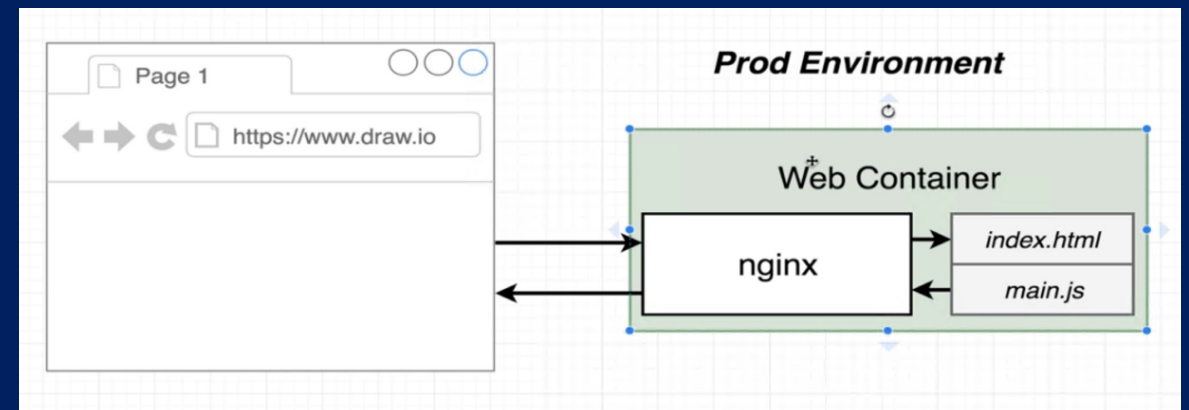
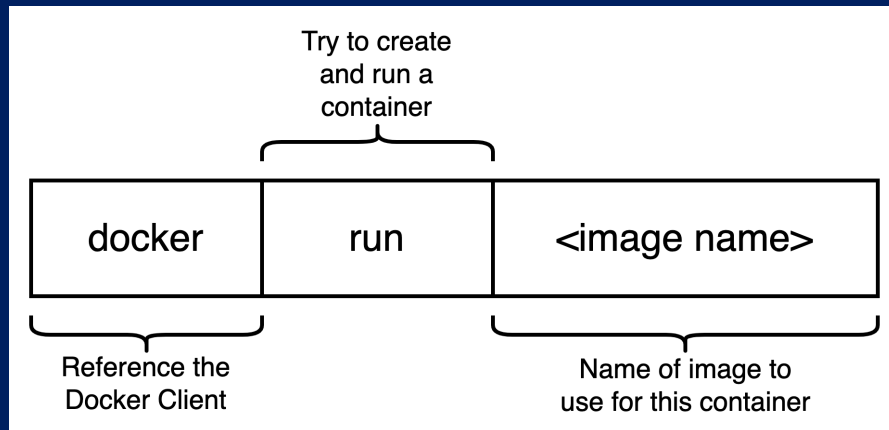
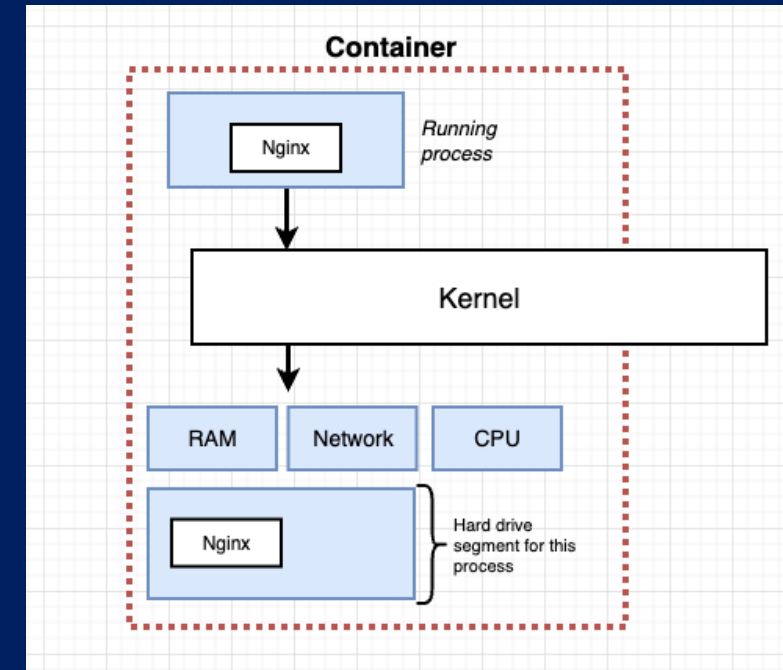


# Run – start a container

▶ `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



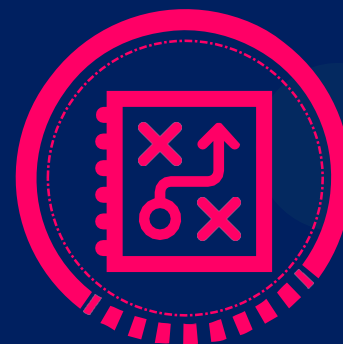
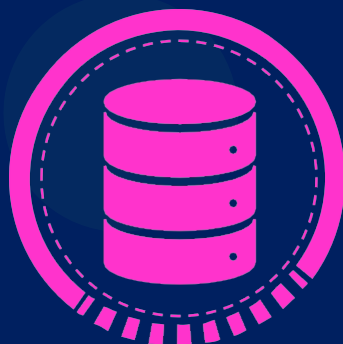
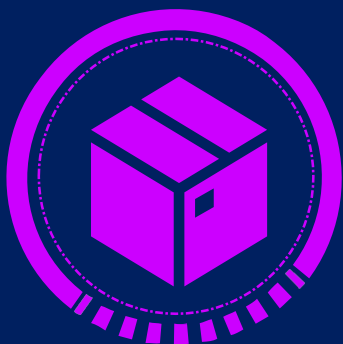
▶ docker run ubuntu

▶ 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
```

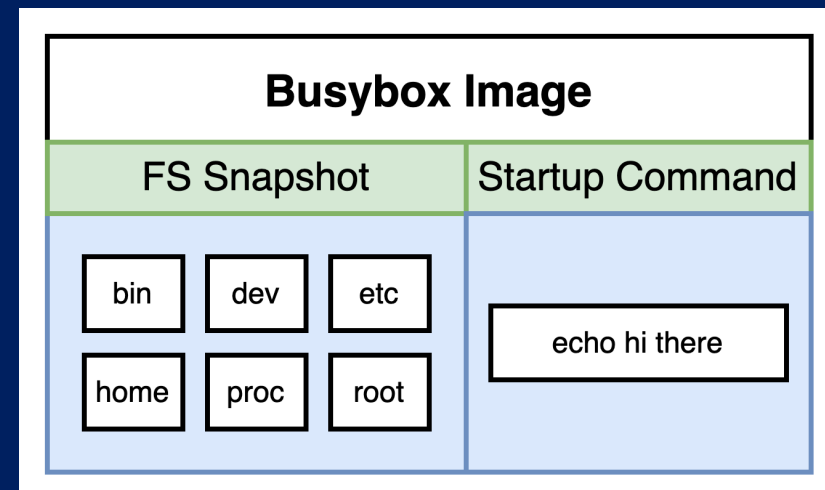
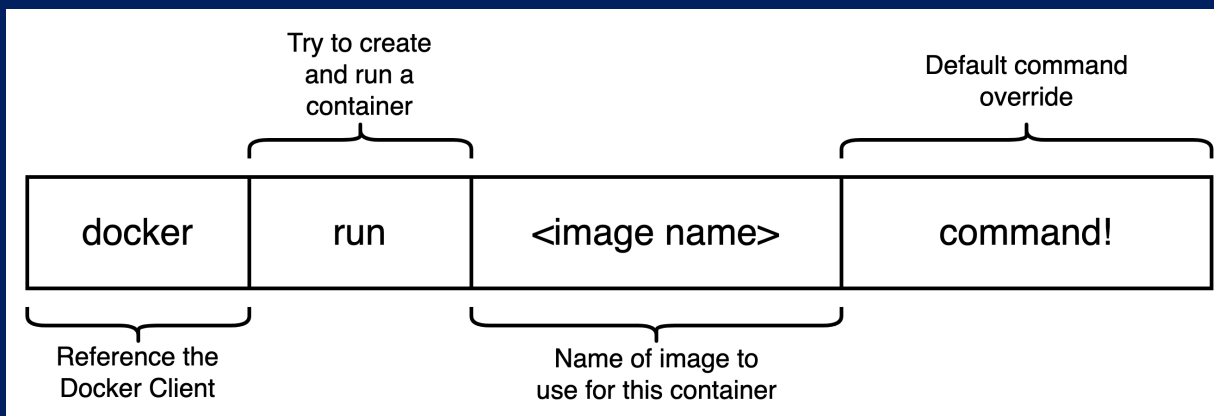
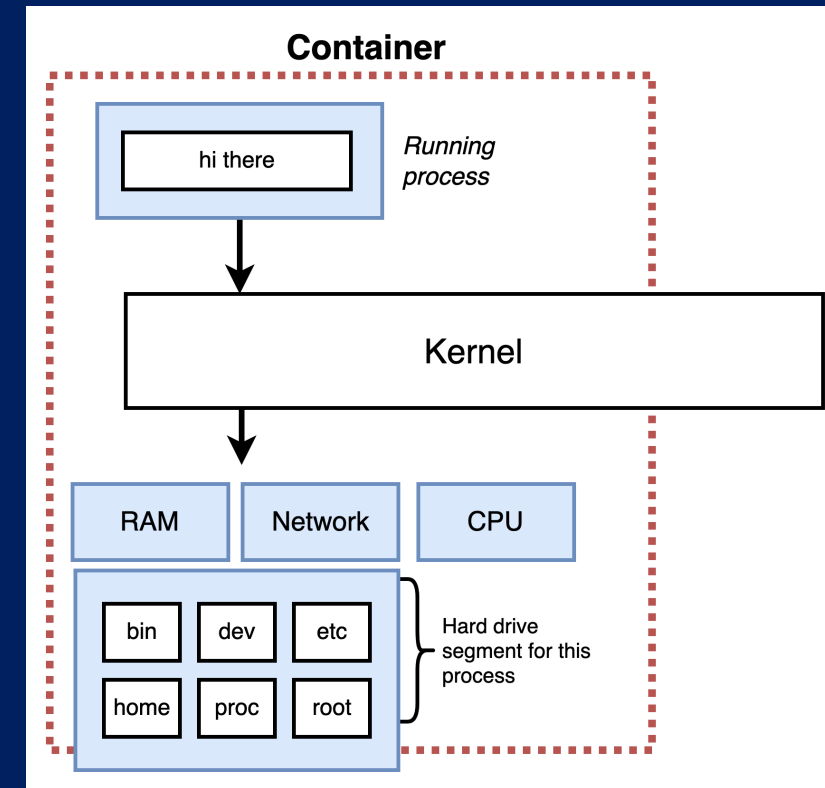
```
814c8b675ca3: Already exists
```

```
Digest:
```

```
sha256:c118f538365369207c12e5794c3cbfb7b042d950af590ae6c287ede74f29b7d4
```

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

```
hi there
```





# Append a command

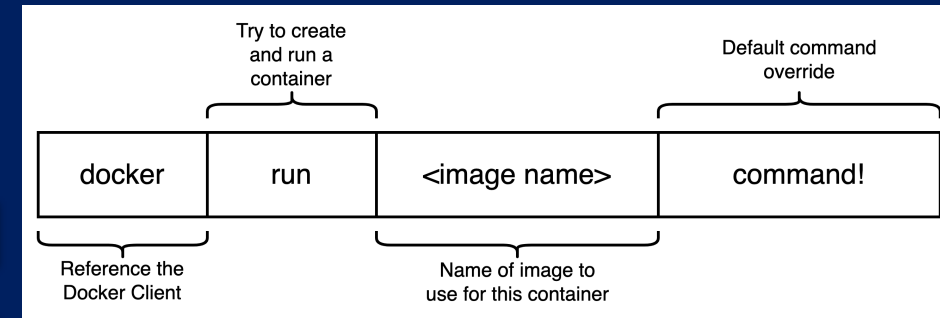
```
▶ docker run ubuntu
```

```
▶ 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
```

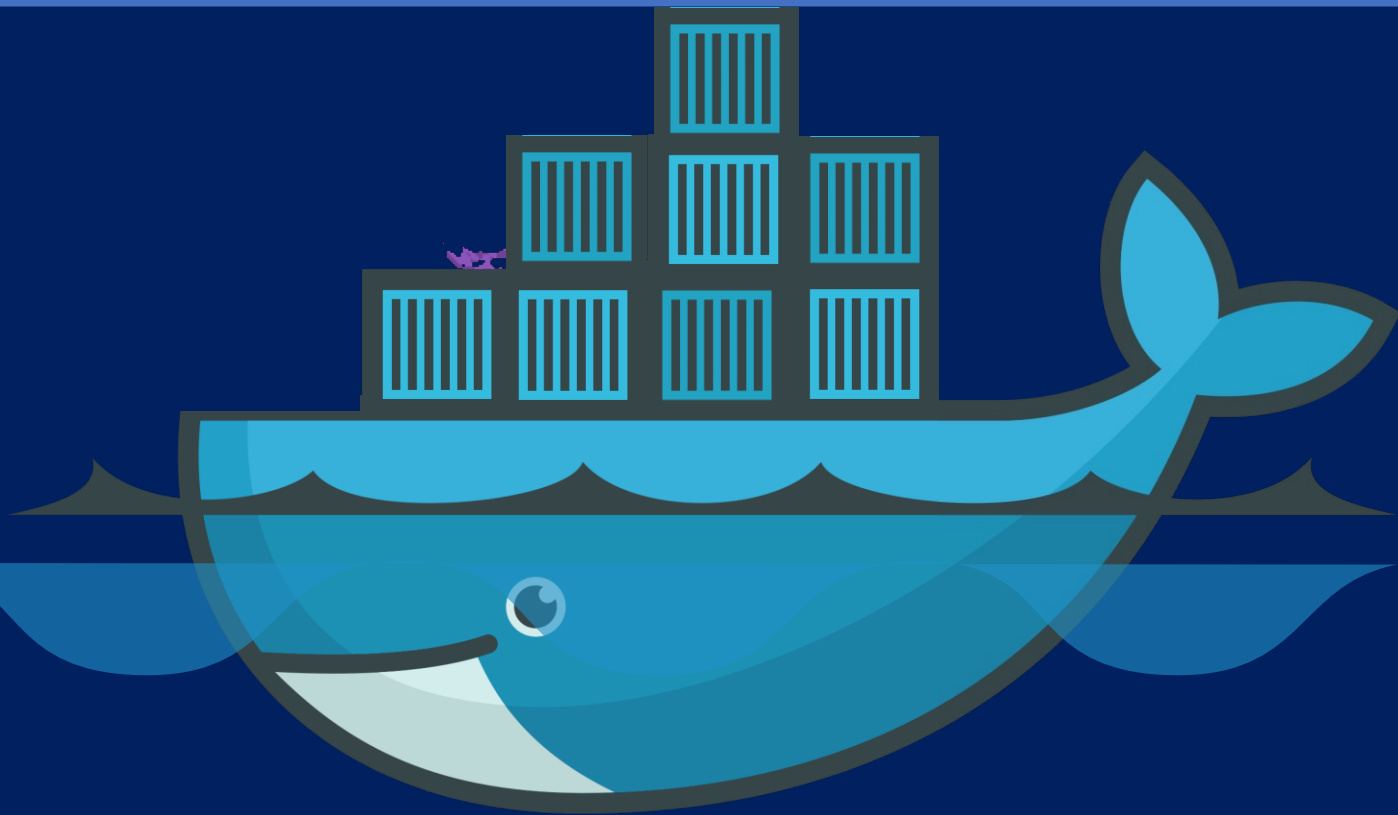
```
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
fc7181108d40: Already exists
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
Status: Downloaded newer image for nginx:latest
```

```
▶ 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
```

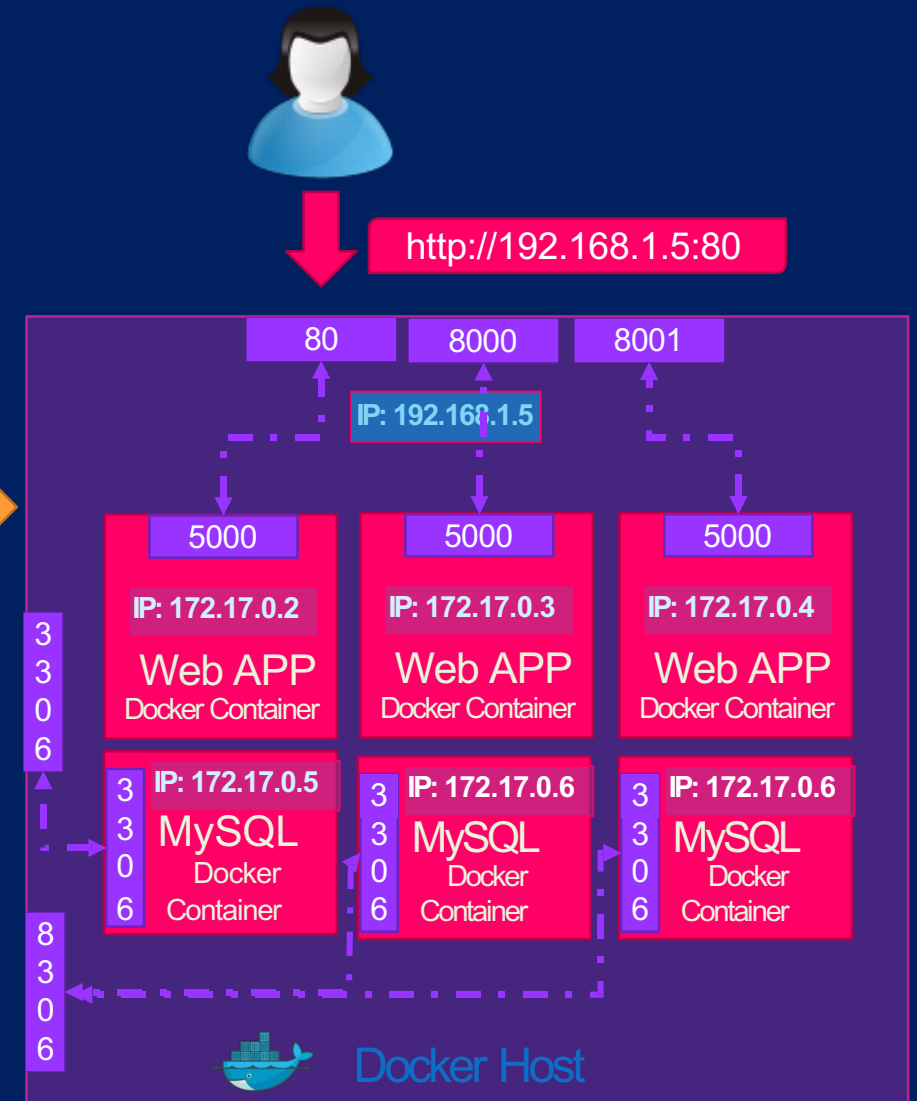
```
docker run -p 8001:5000 myname/simple-webapp
```

```
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
```

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)

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

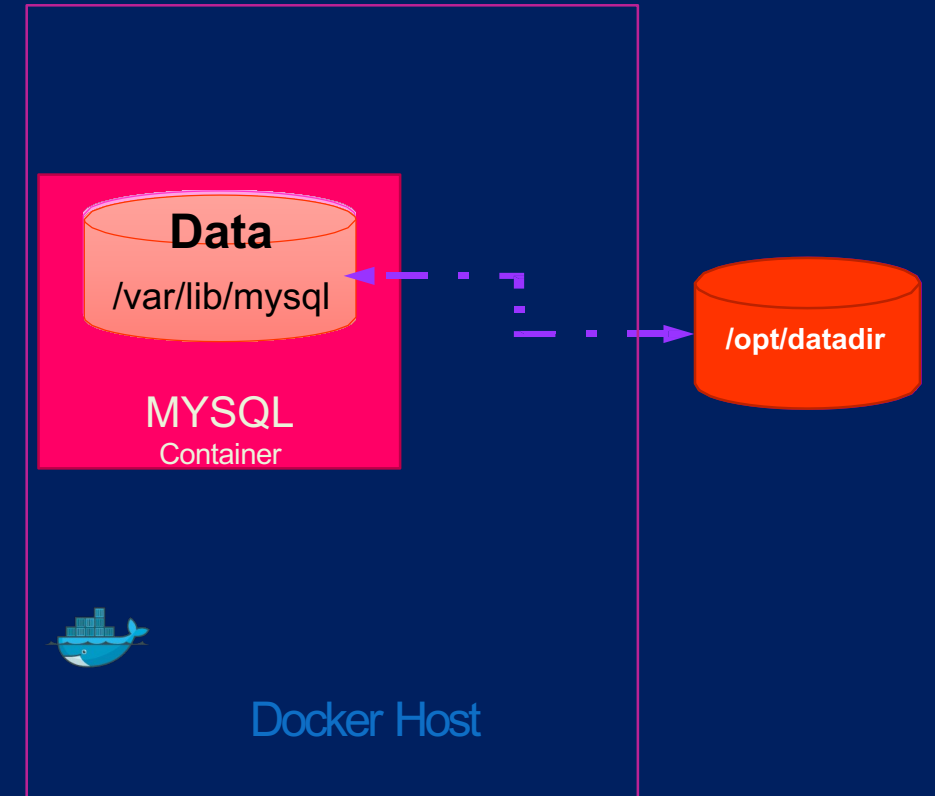
Name	Security group rule ID	Port range	Protocol	Source
-	sgr-041cfa665c4cffb6a	8080	TCP	0.0.0.0/0
-	sgr-02e77377fbc1f1563	22	TCP	0.0.0.0/0
-	sgr-0c38e2629739ad3ce	80	TCP	0.0.0.0/0

# RUN – Volume mapping

```
docker run mysql
```

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

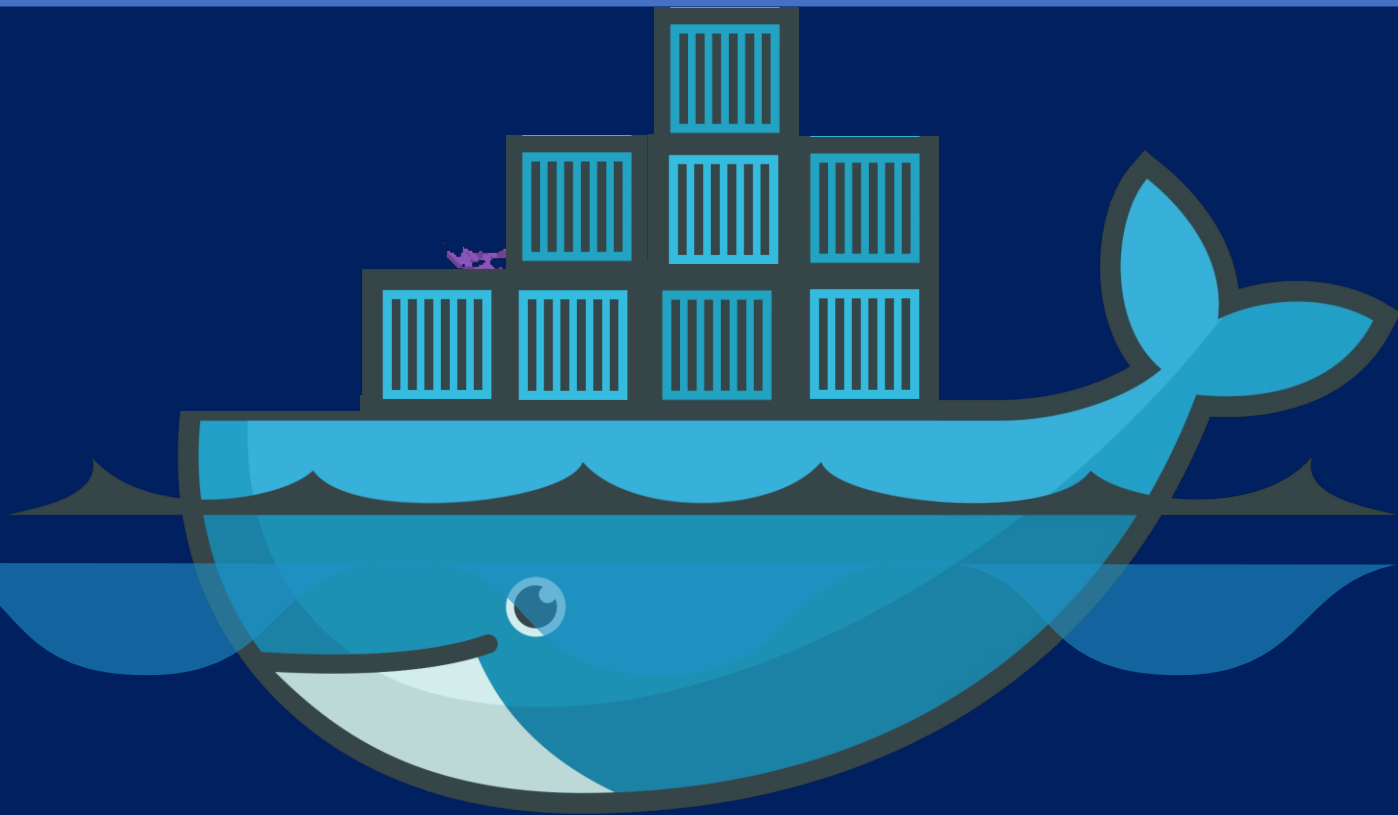
```
docker run -v /opt/datadir:/var/lib/mysql mysql
```






# 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
```



 SOFTWARE-DEVELOPMENT-TOOLS-AND-ENVIRONMENTS / Week8 / Nginx\_Volume\_Port\_Mapping /

 Tuchsanaï 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/Tuchsanaï/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 /

 Tuchsanaï 1

Name	Last commit message
..	
index.html	1

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   <a href="#">open address</a>
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

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

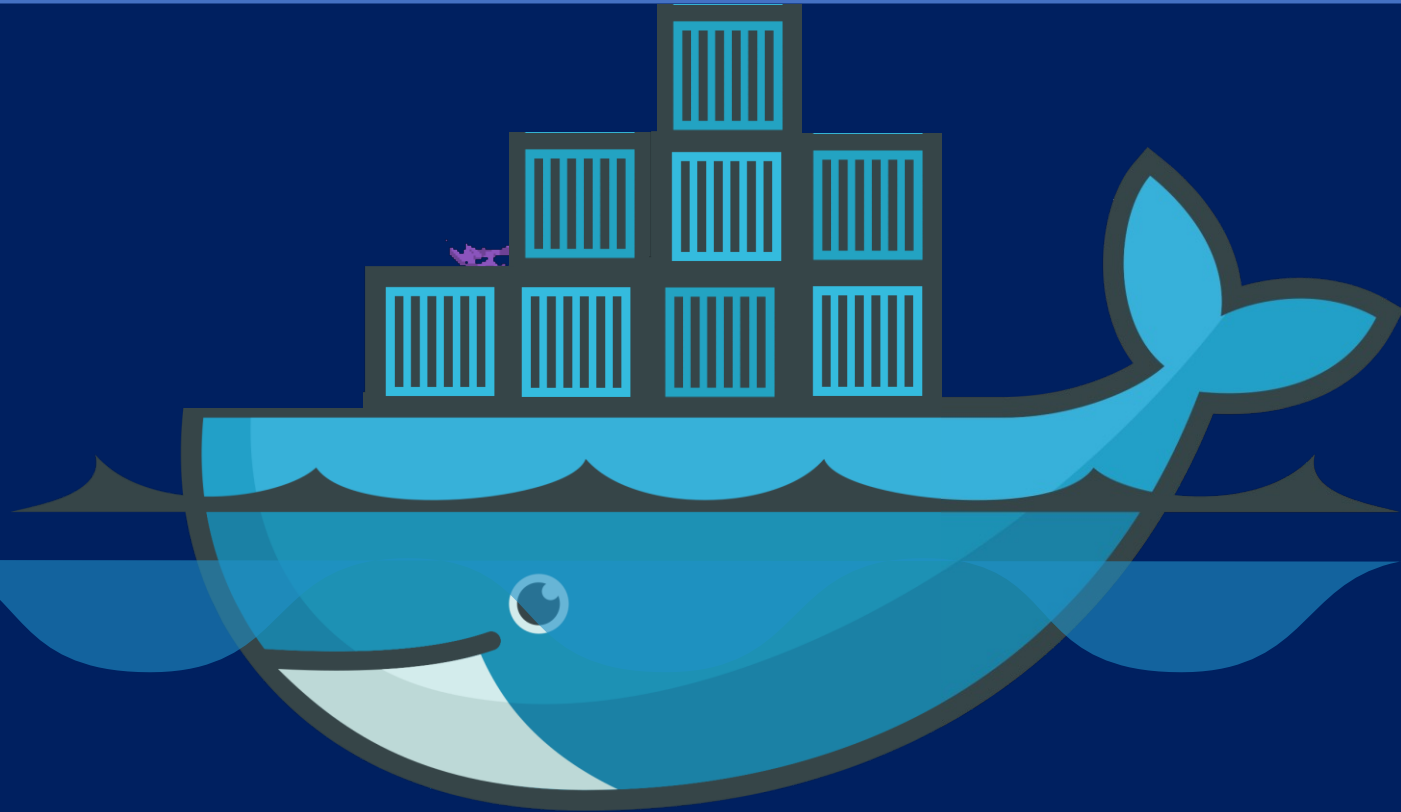
# Welcome to Demo nginx Website

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
```

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

Title

Detail

dd/mm/yyyy

Submit

Docker Workshop

2017-11-21

Linuxing in London

Delete

WinOps #17

2017-11-21

WinOps London

Delete

Docker London

2017-11-13

Delete

▼ 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