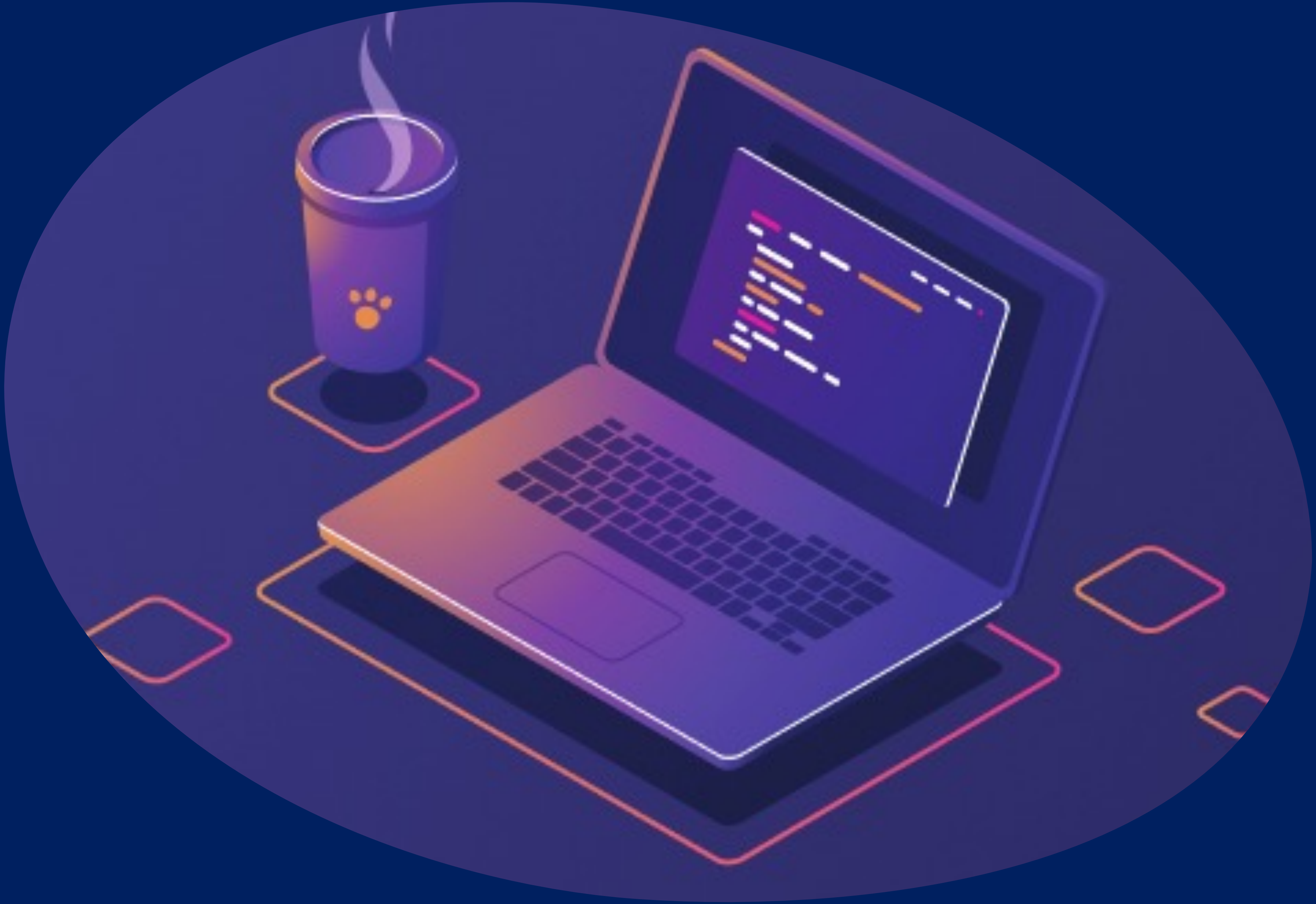
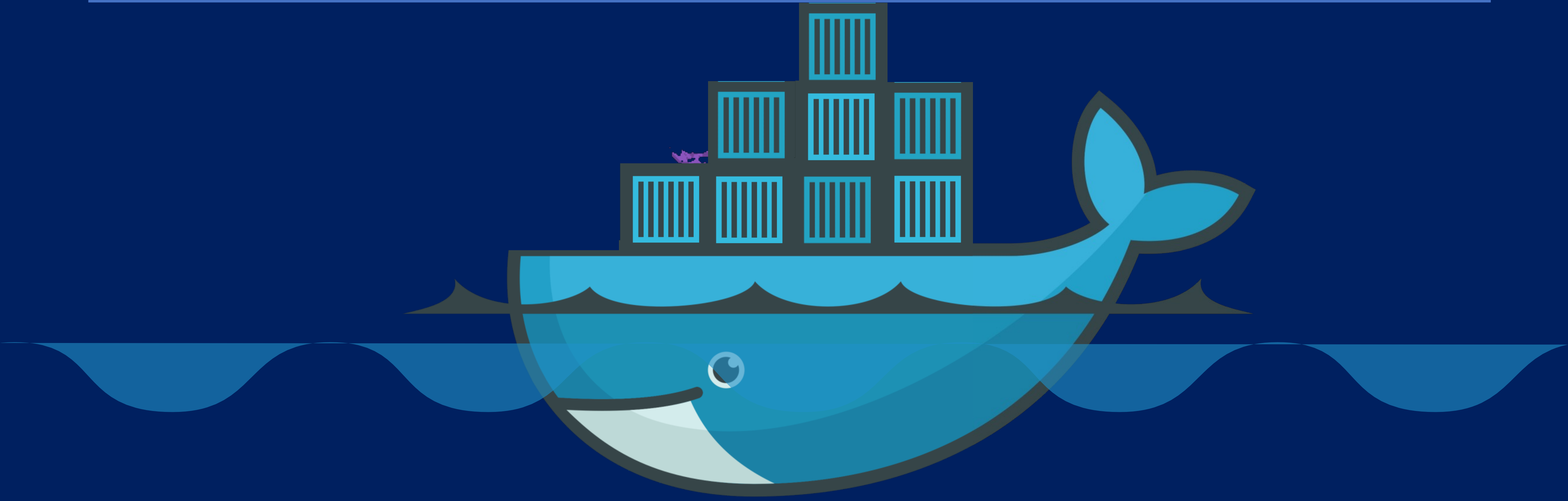
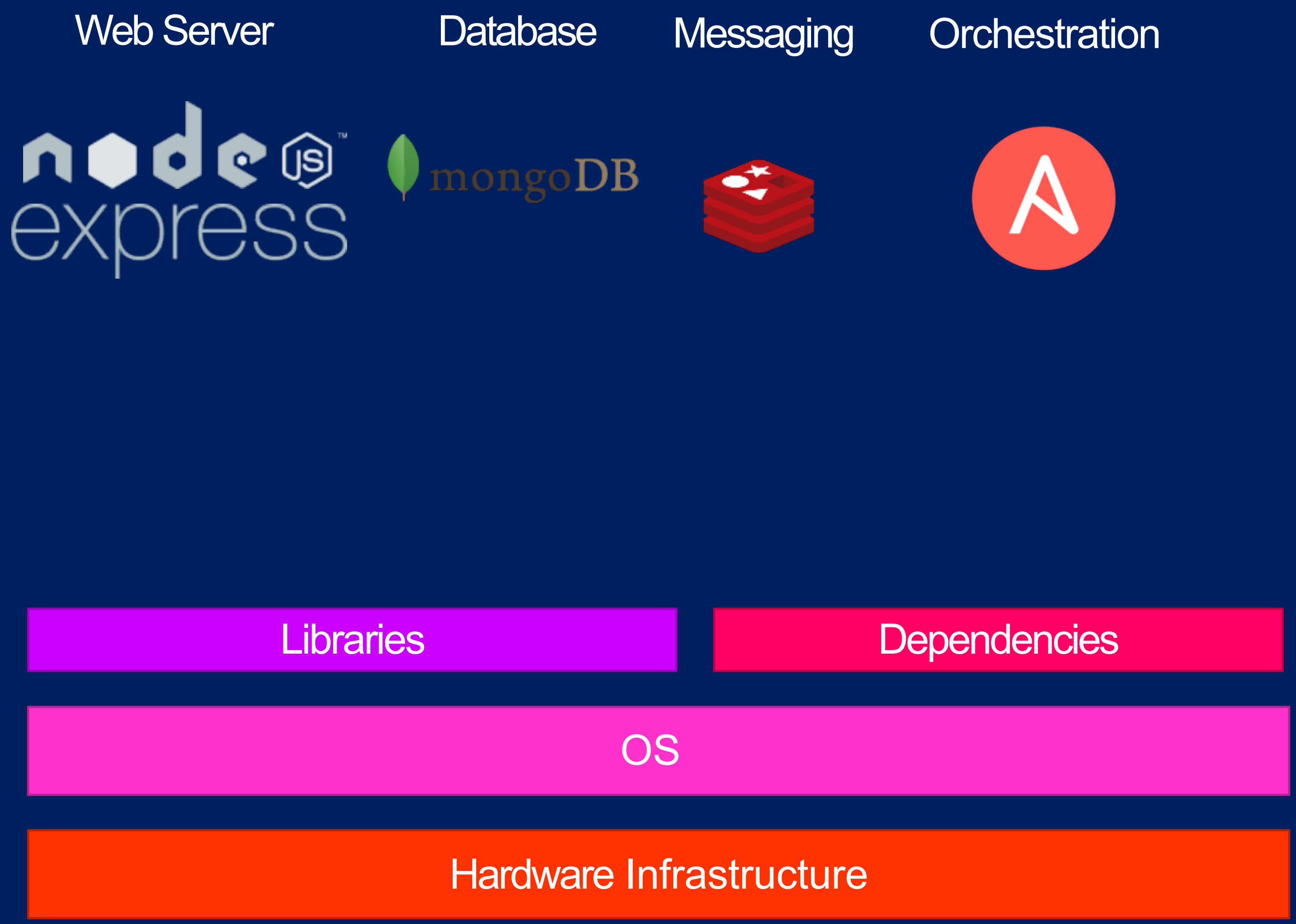


Week 8 : SOFTWARE DEVELOPMENT TOOLS AND ENVIRONMENTS

Docker Overview



Why do you need docker?



Web Server

Database

Messaging

Orchestration

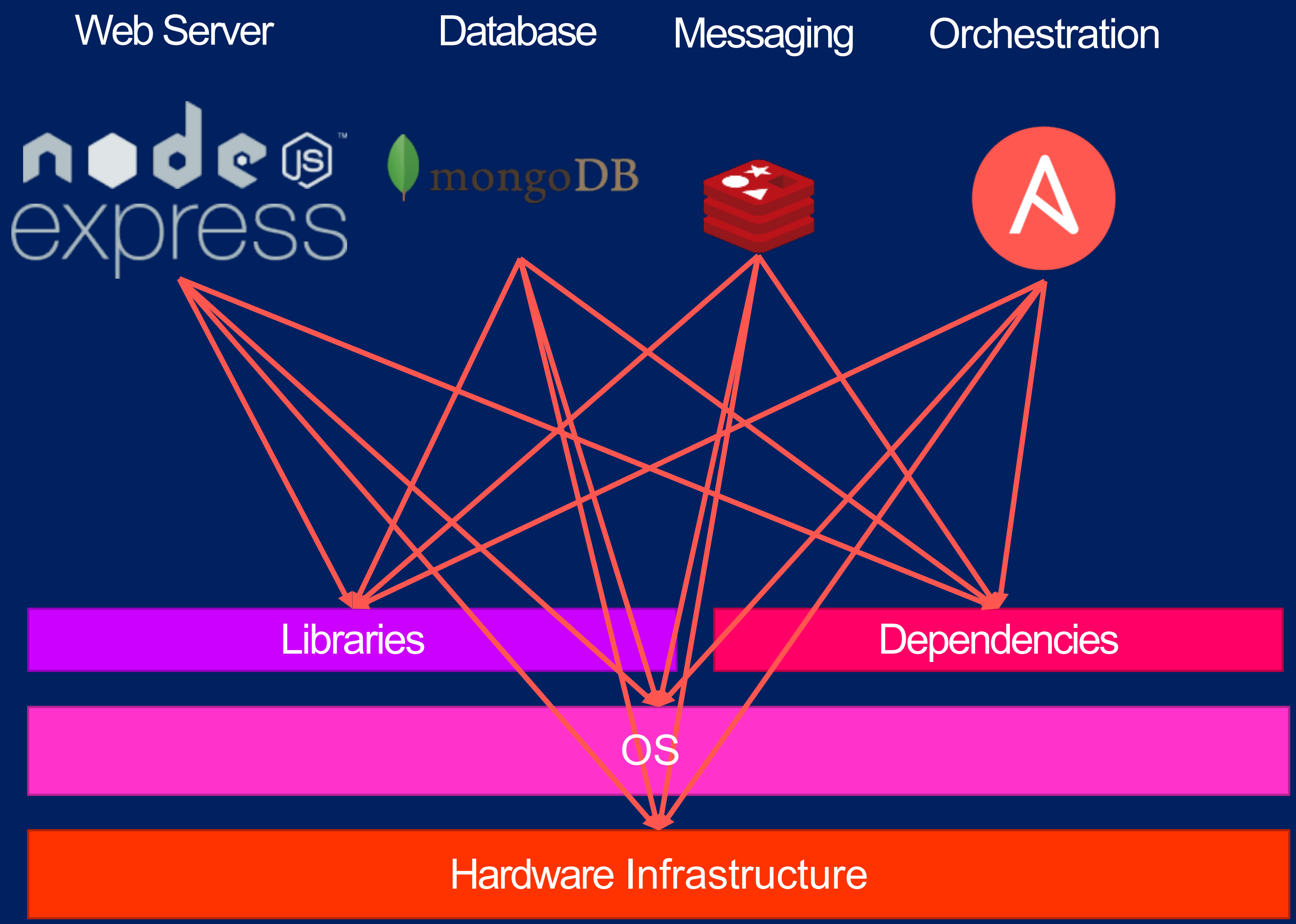
Libraries

Dependencies

OS

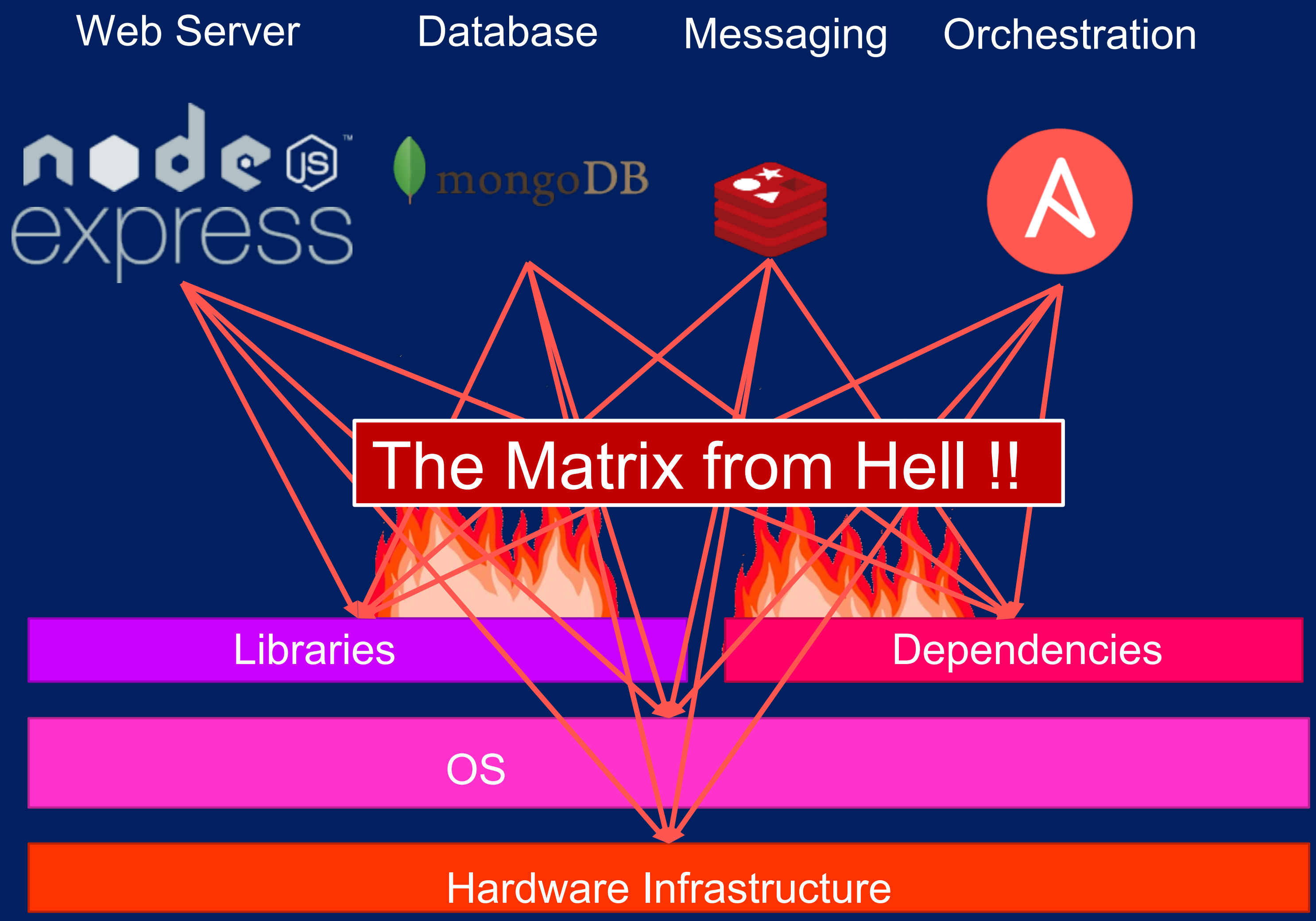
Hardware Infrastructure

Why do you need docker?



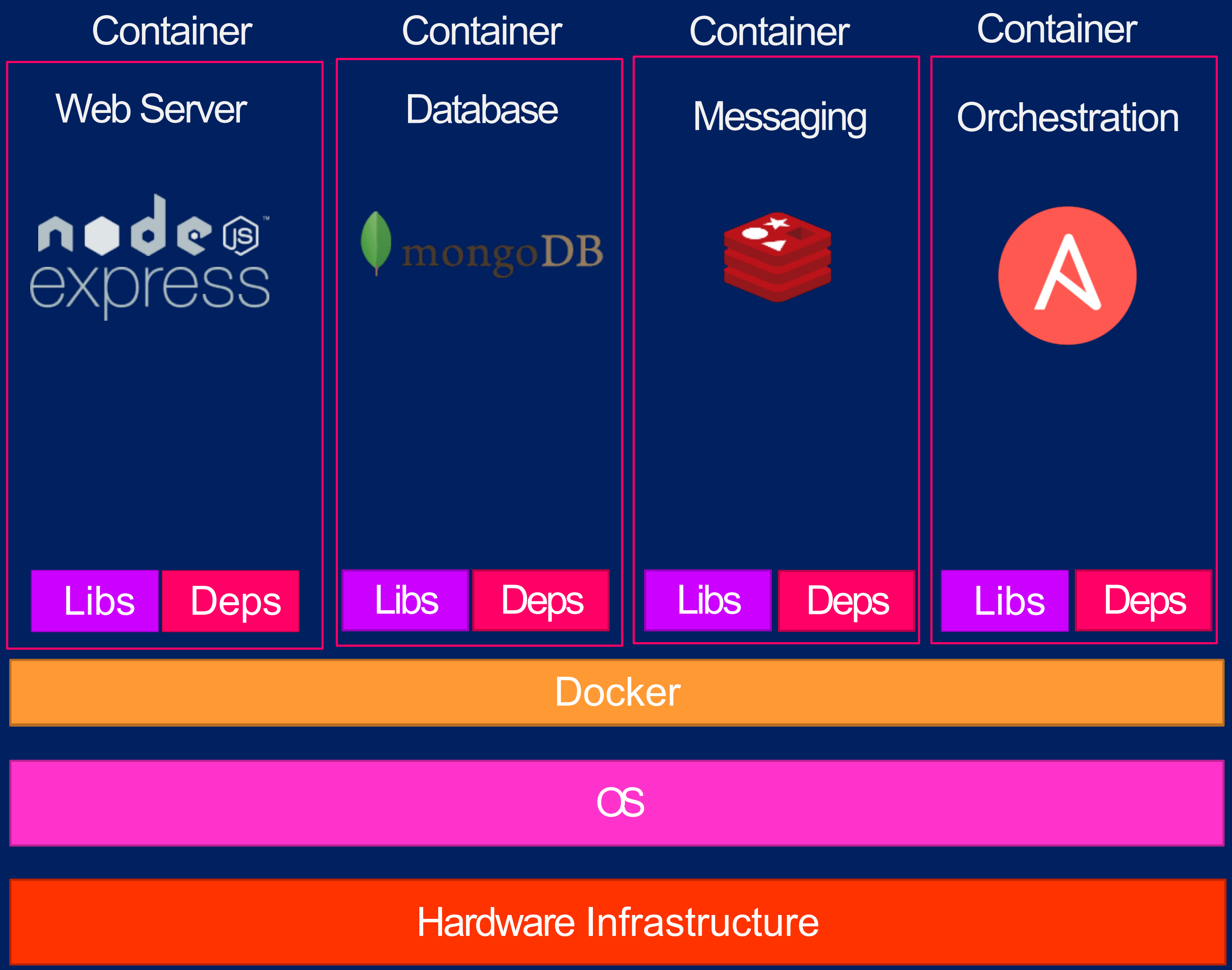
Why do you need docker?

- 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



Container | Container | Container | Container

Web Server | Database | Messaging | Orchestration

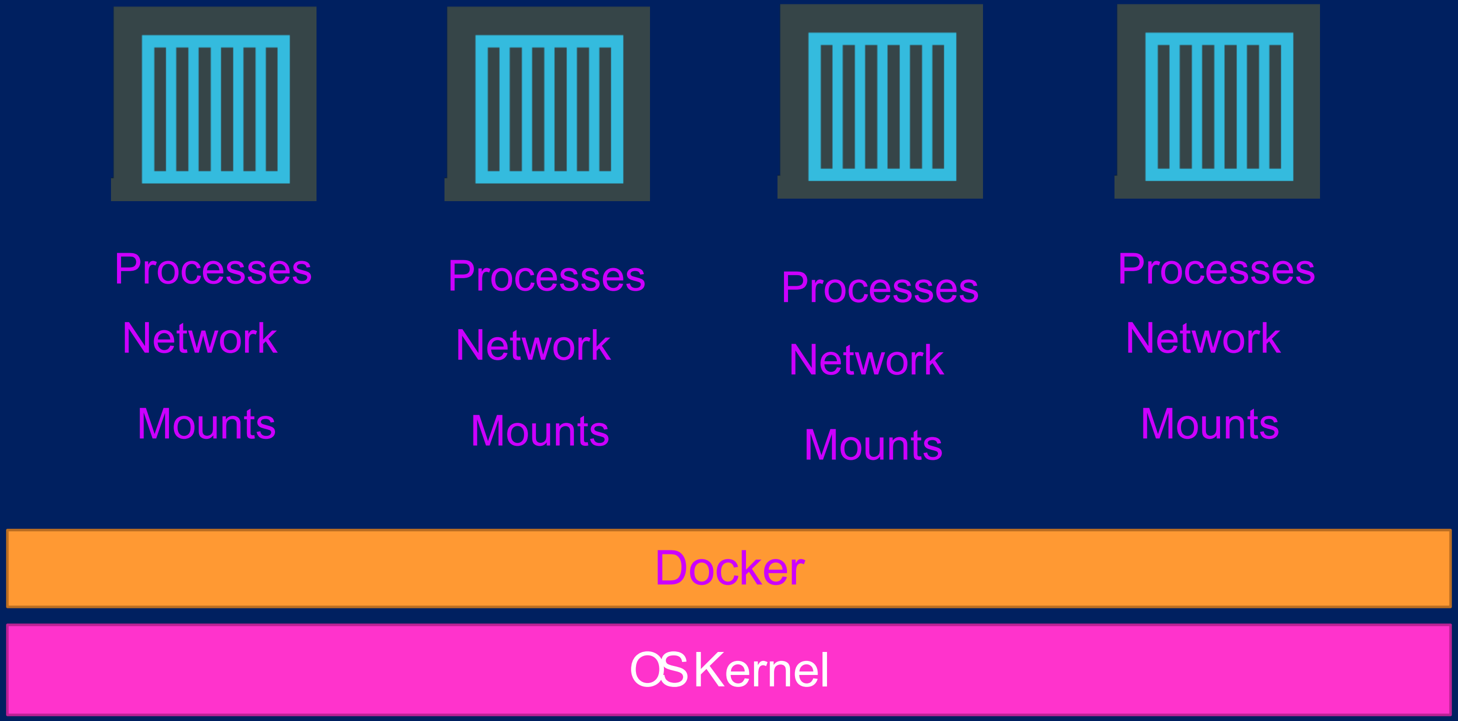
Libs | Deps | Libs | Deps | Libs | Deps | Libs | Deps

Docker

OS

Hardware Infrastructure

What are containers?



Processes

Network

Mounts

Processes

Network

Mounts

Processes

Network

Mounts

Processes

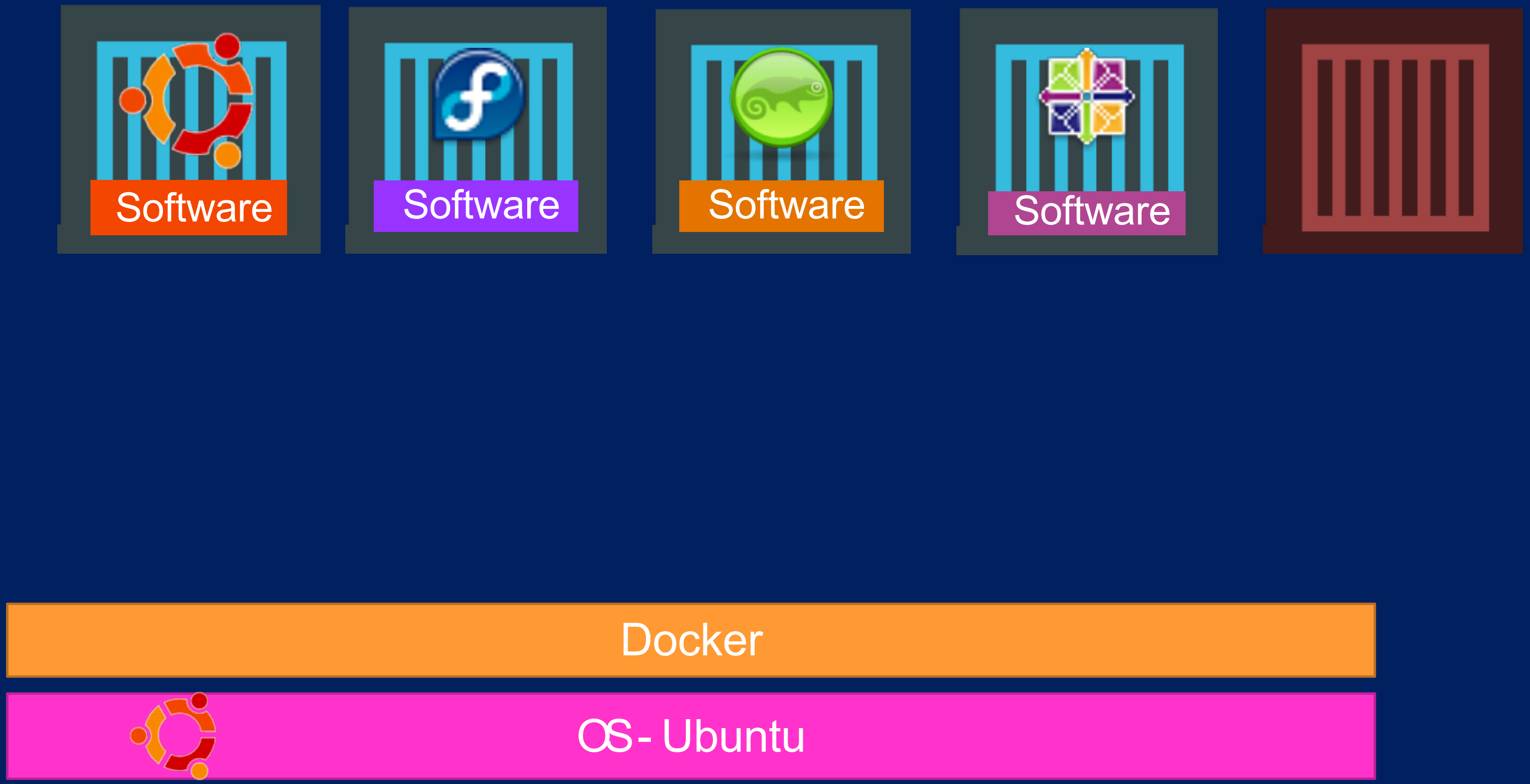
Network

Mounts

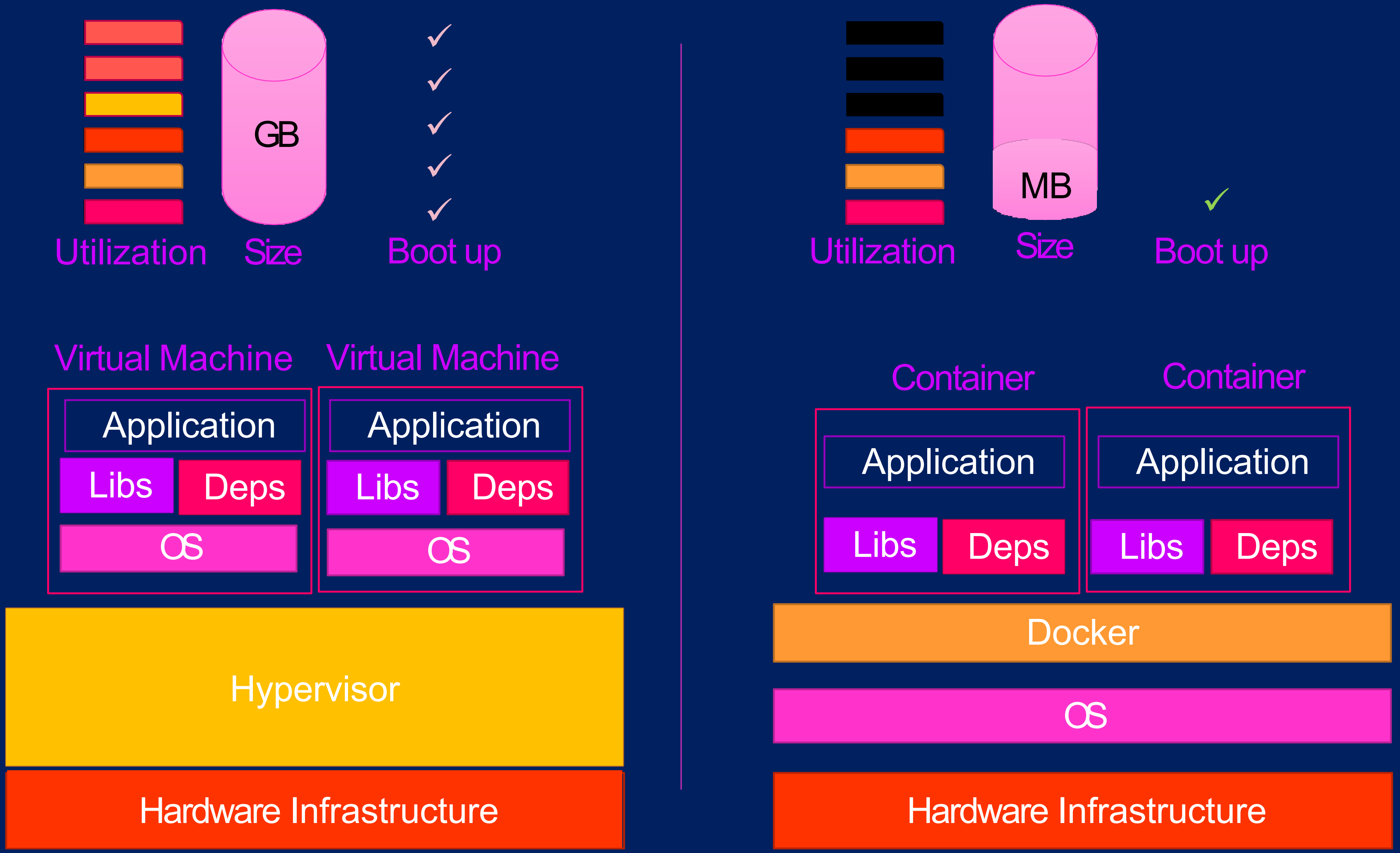
Docker

OS Kernel

Sharing the kernel for Operating System



Containers vs Virtual Machines



Utilization Size Boot up

GB

✓ ✓ ✓ ✓ ✓

Virtual Machine

Application

Libs Deps

OS

Virtual Machine

Application

Libs Deps

OS

Hypervisor

Hardware Infrastructure

Utilization Size Boot up

MB

✓

Container

Application

Libs Deps

Container

Application

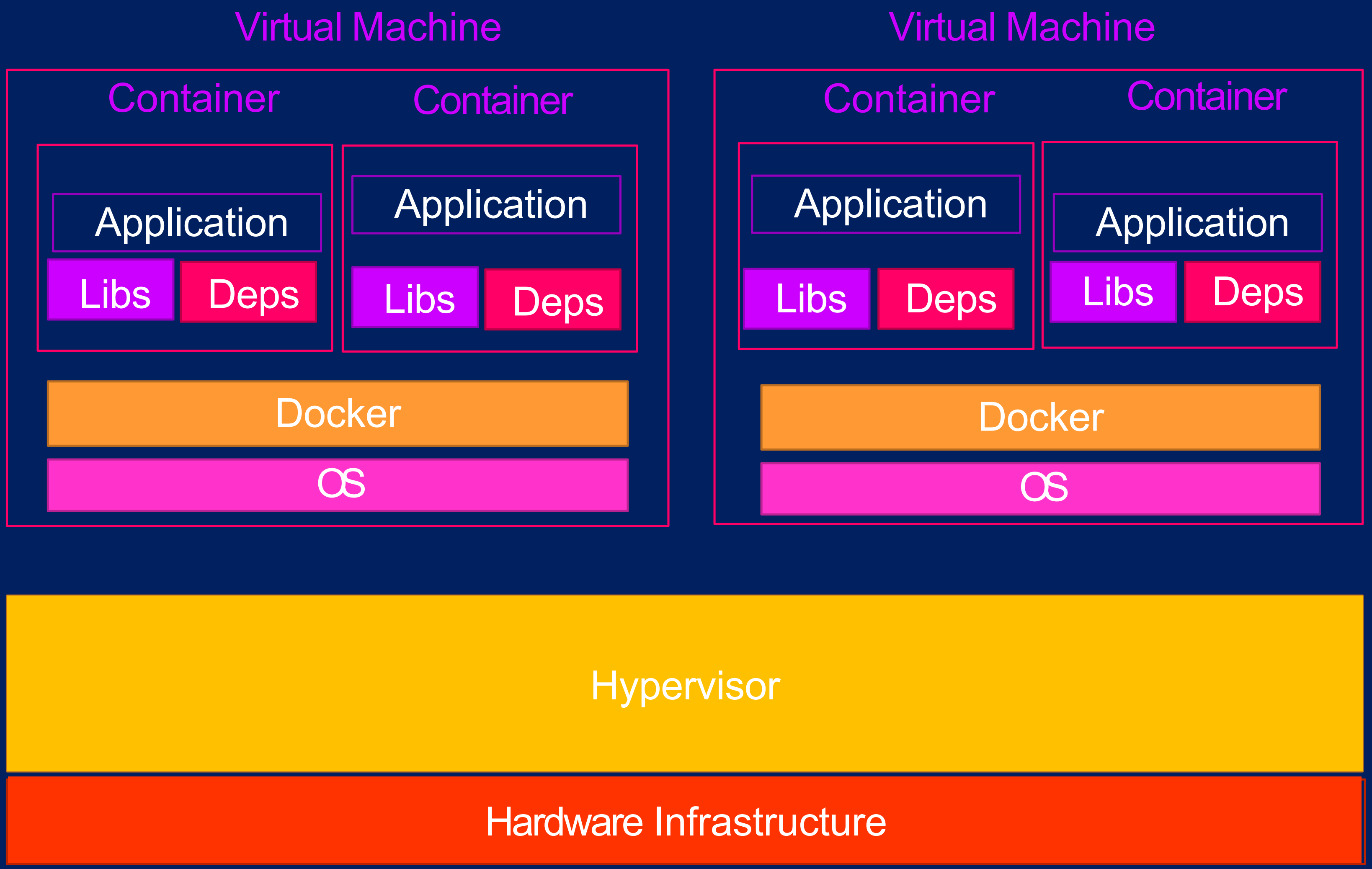
Libs Deps

Docker

OS

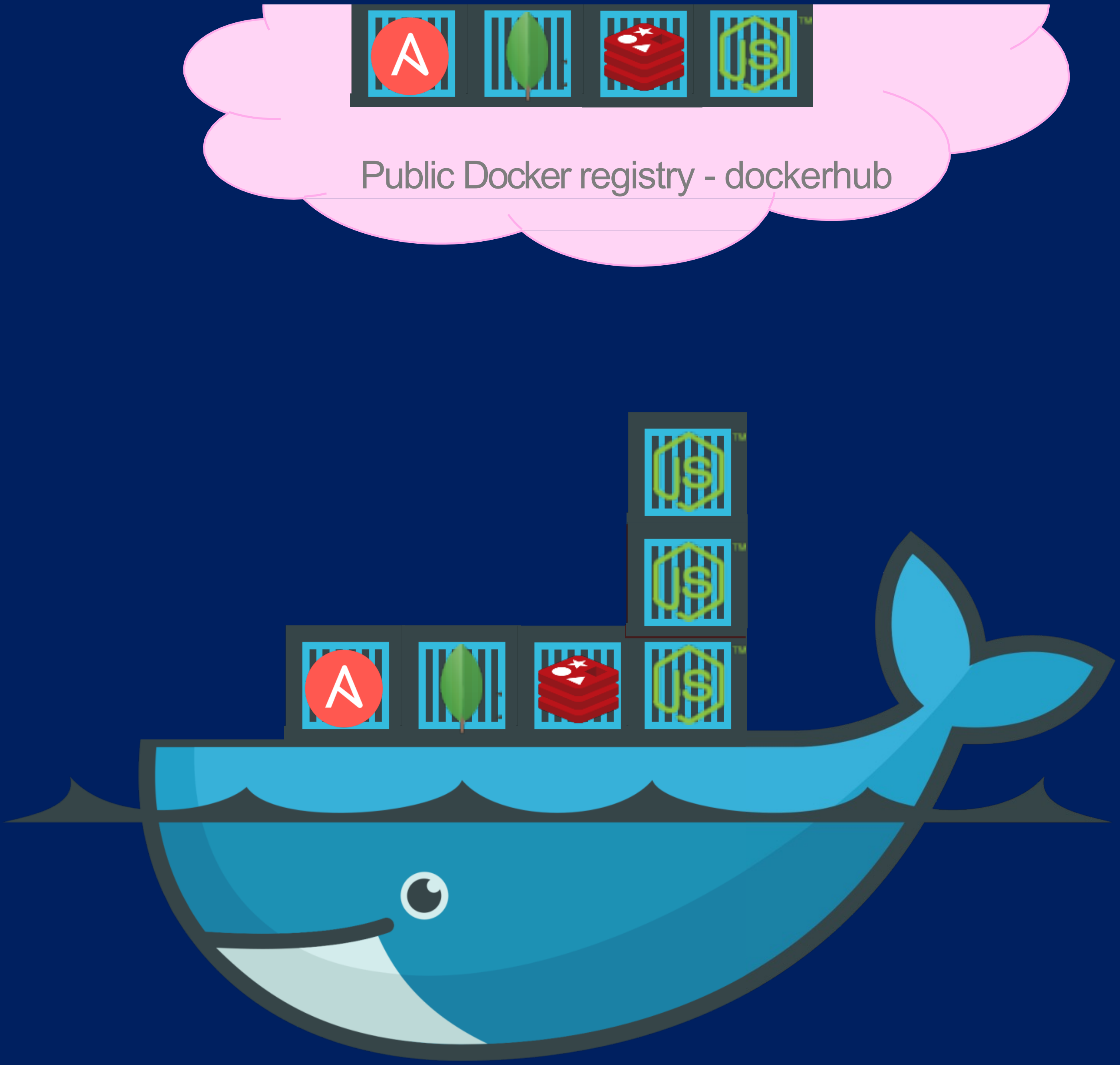
Hardware Infrastructure

Containers & Virtual Machines



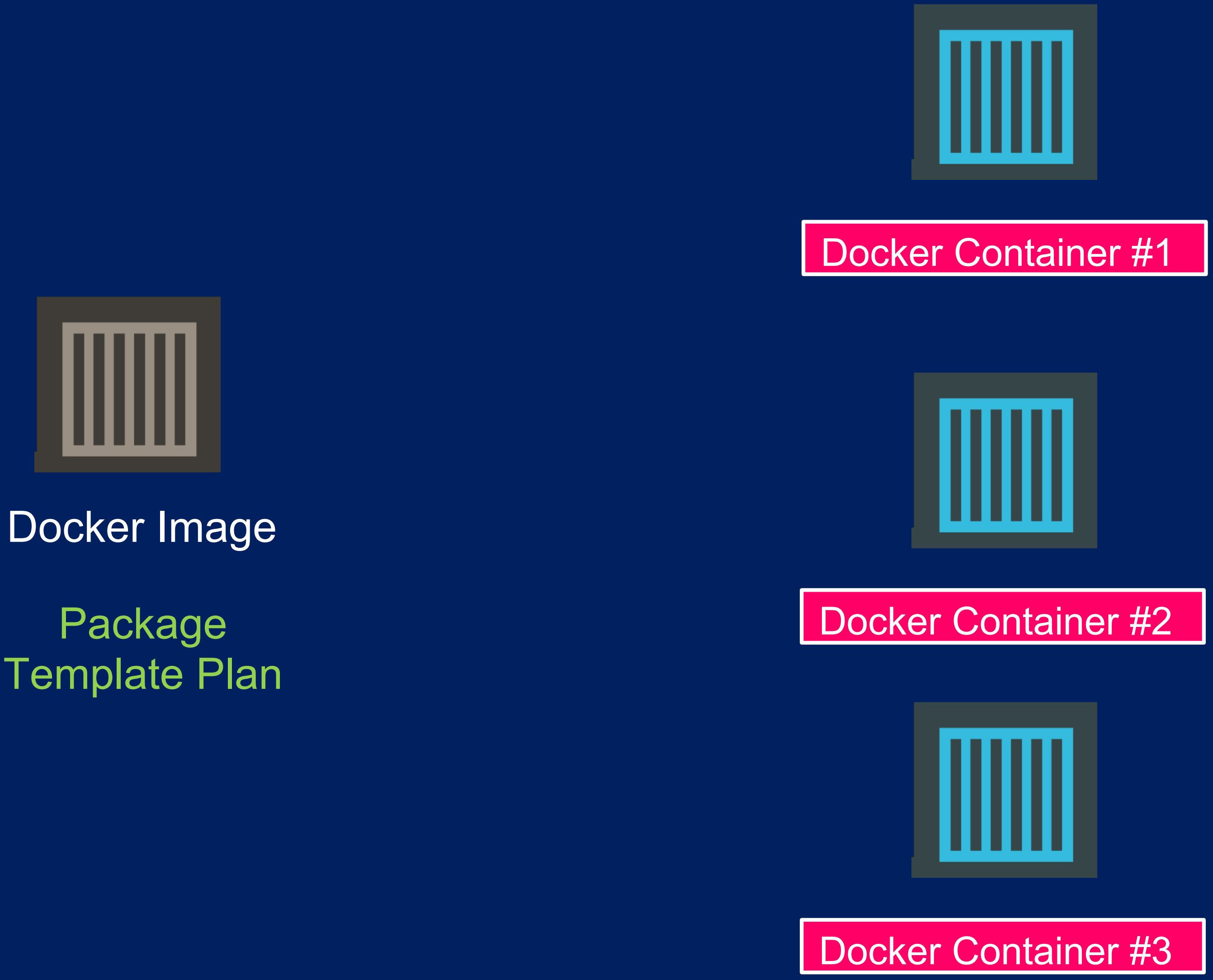
How is it done?

```
docker run ansible

docker run mongodb

docker run redis

docker run nodejs

docker run nodejs

docker run nodejs
```



Public Docker registry - dockerhub

Container vs image



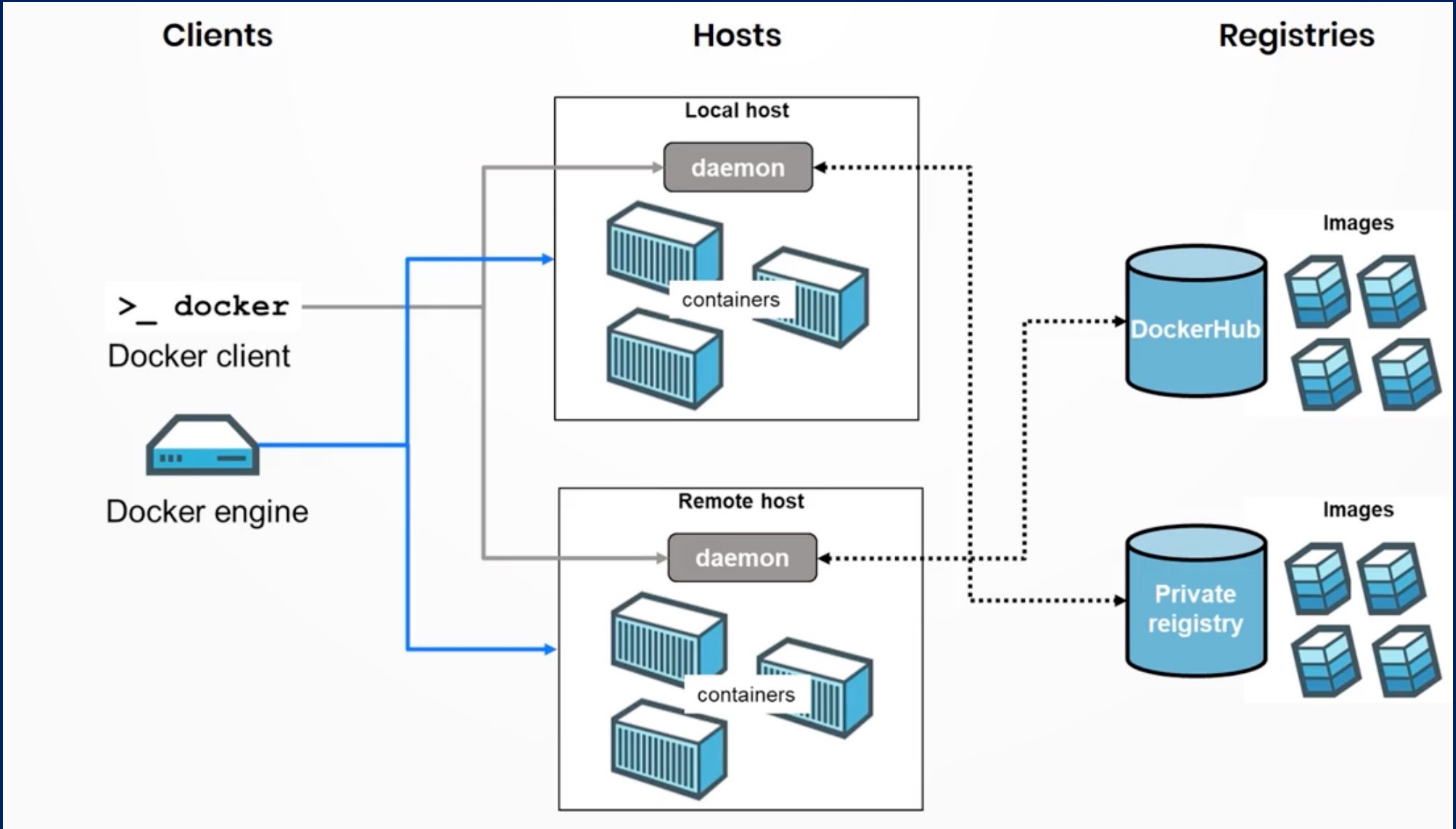
Docker Image

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

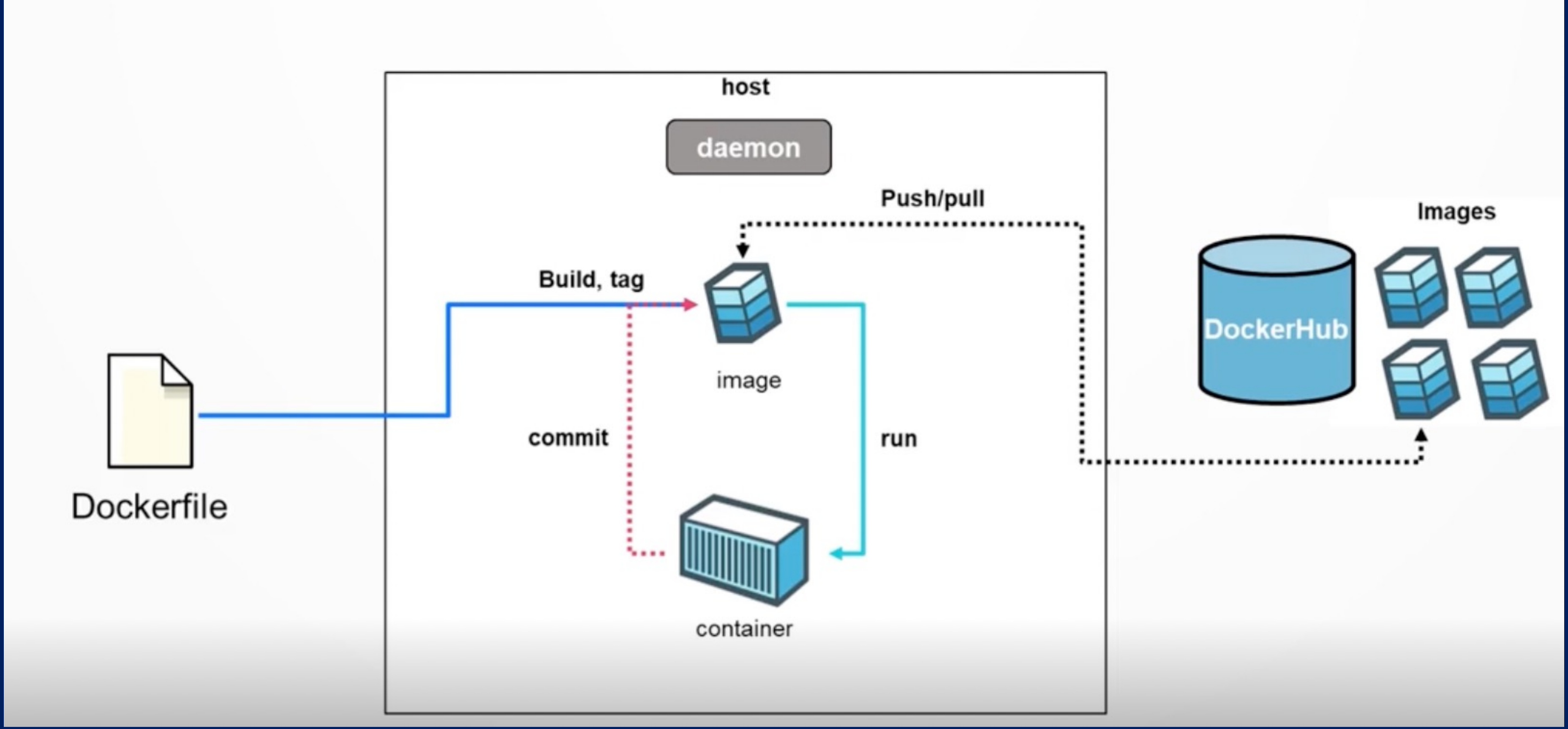
containers

DockerHub

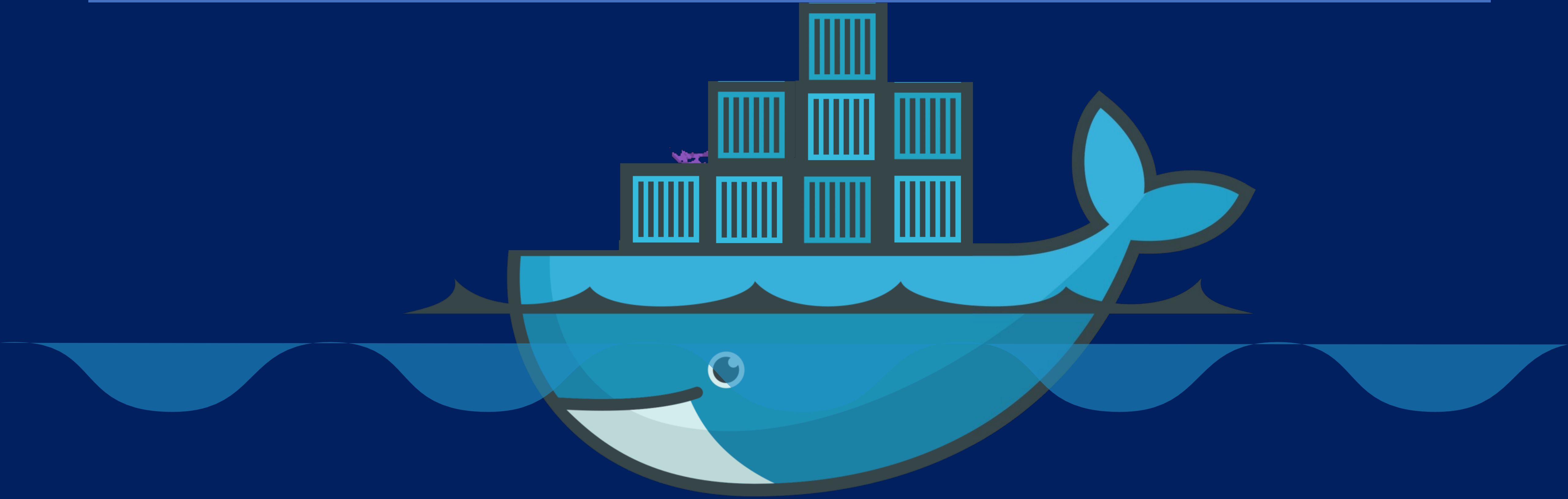
Images

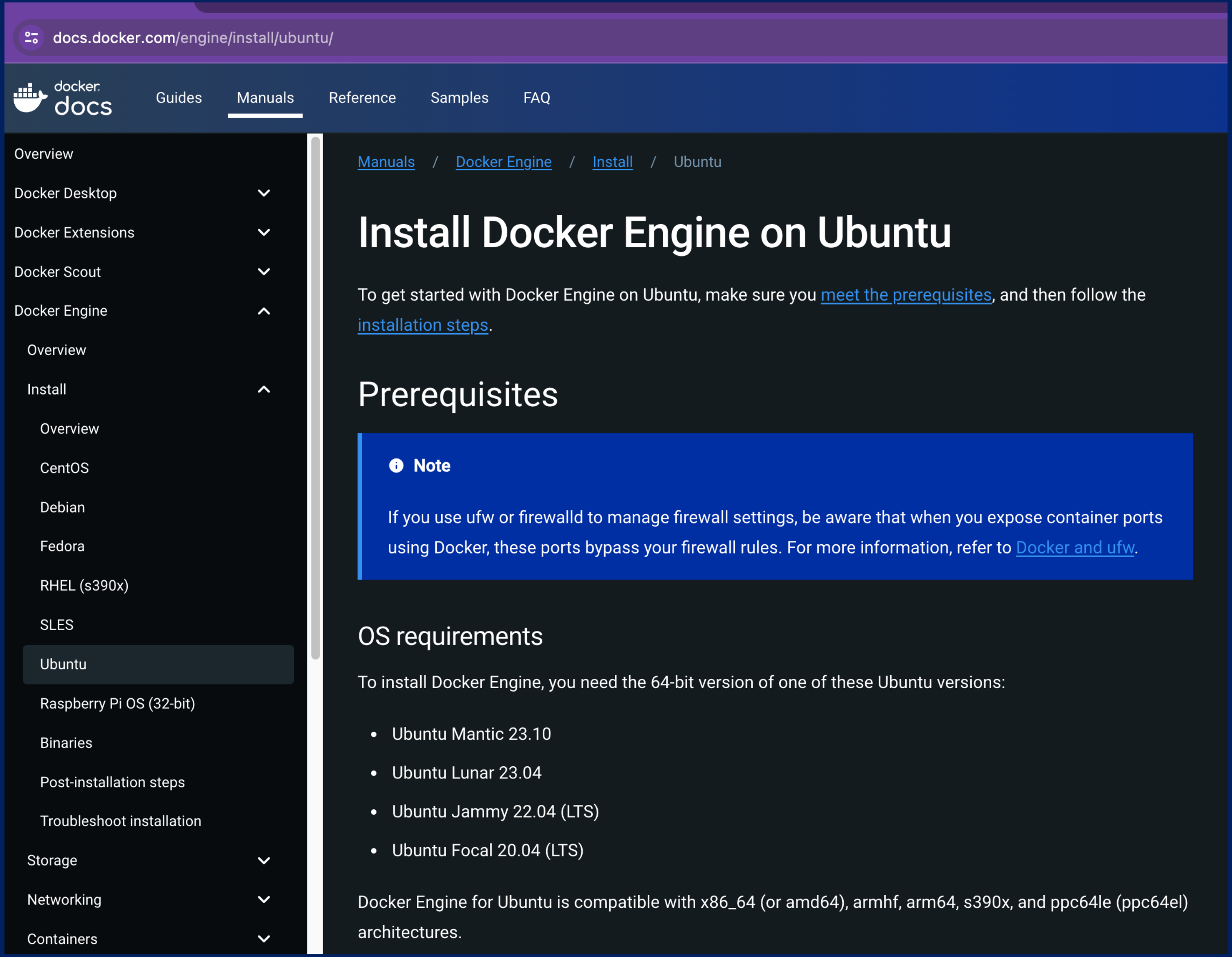
Private reigistry

Images


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

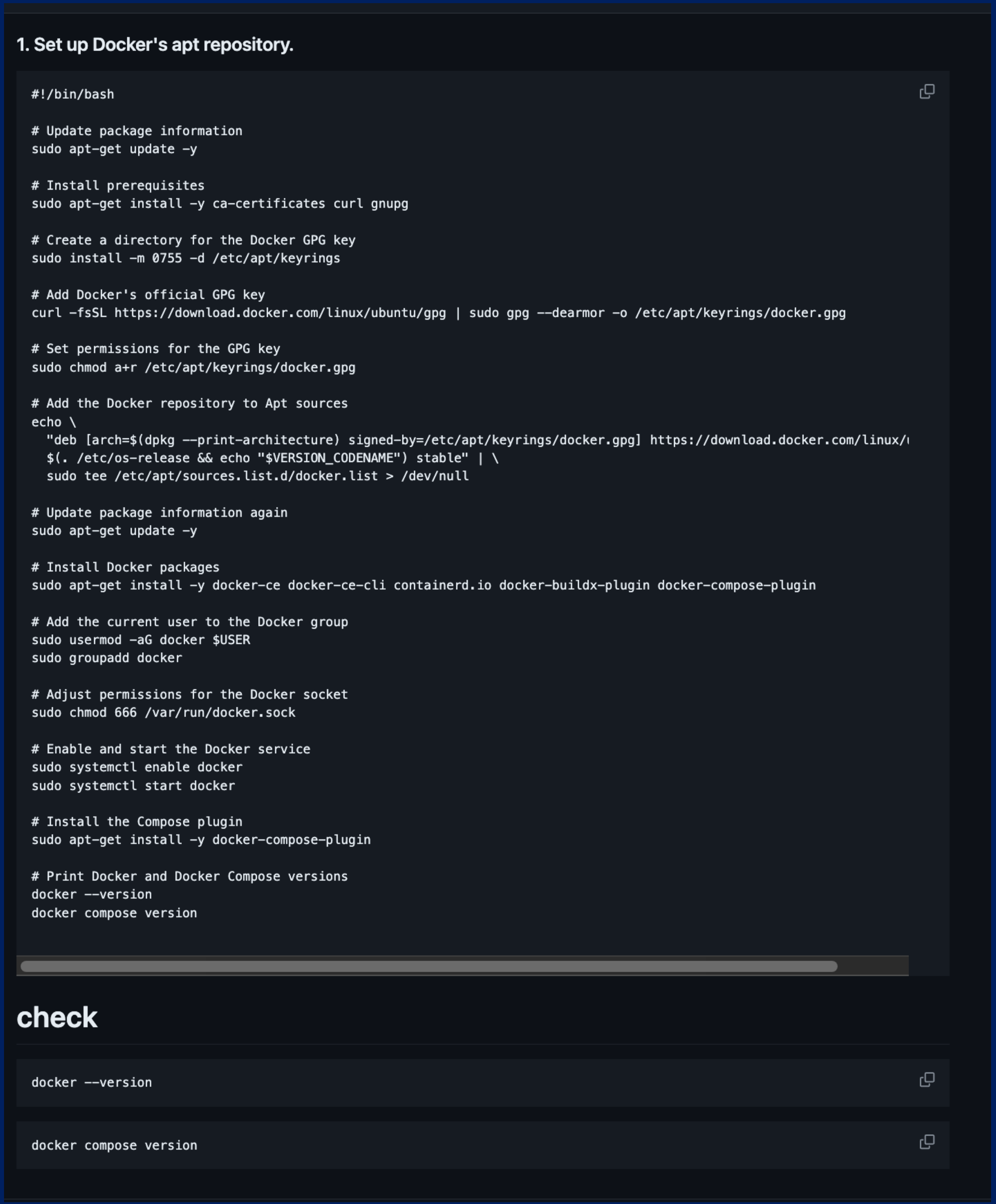
OS requirements

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
```

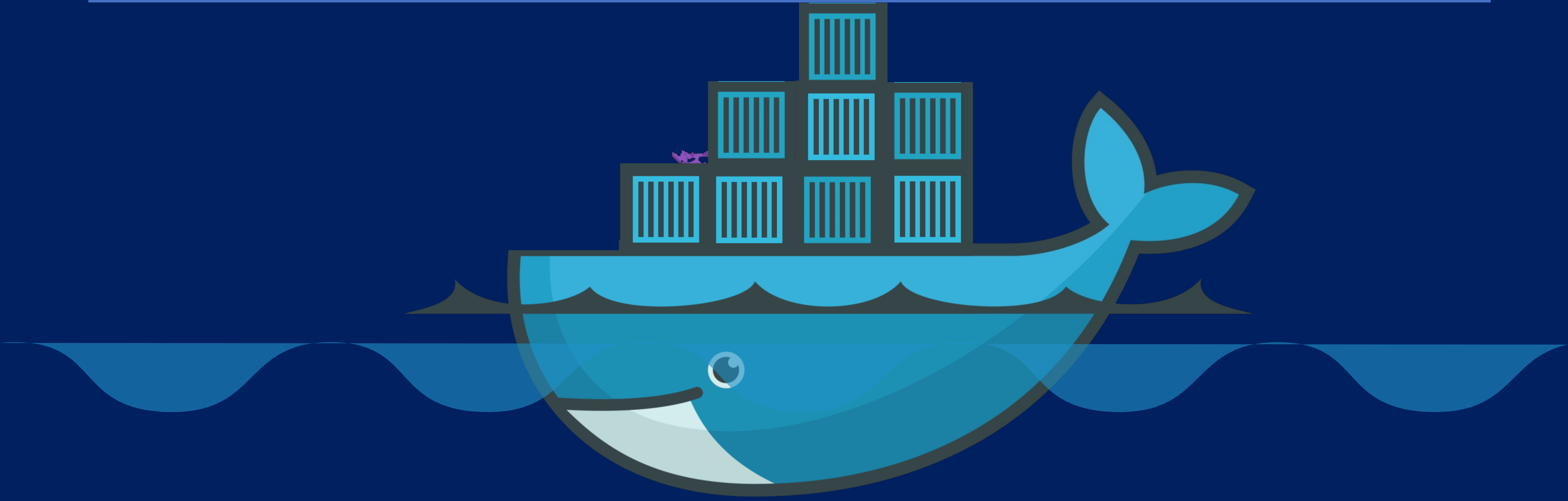
check

```
docker --version
```

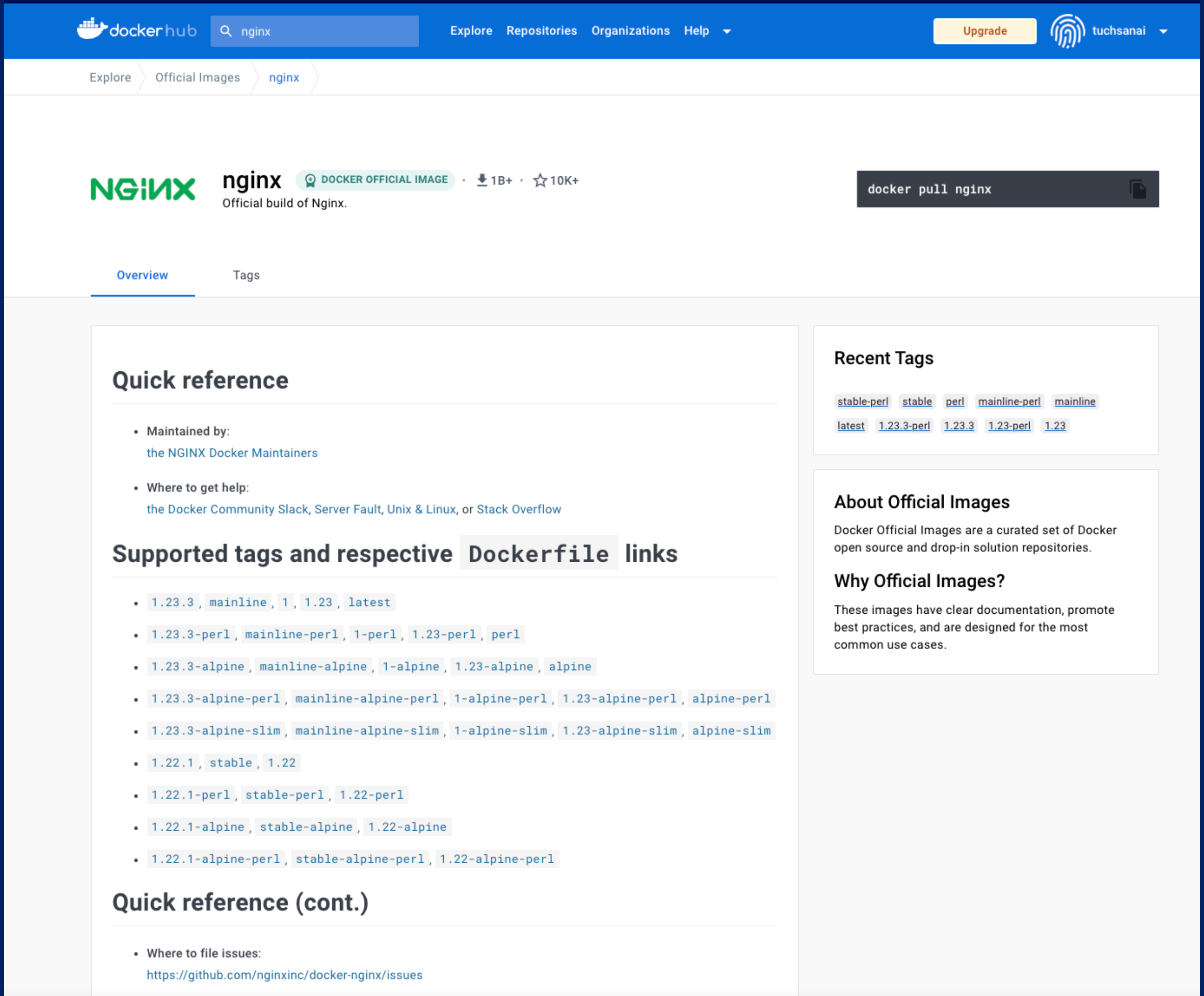
```
docker compose version
```

https://github.com/Tuchsanai/DevTools/tree/main/02_Docker/Week08/LAB1_Install%20Docker

LAB 2 : Docker Run



Docker Registry



docker hub nginx Explore Repositories Organizations Help Upgrade tuchsanai

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

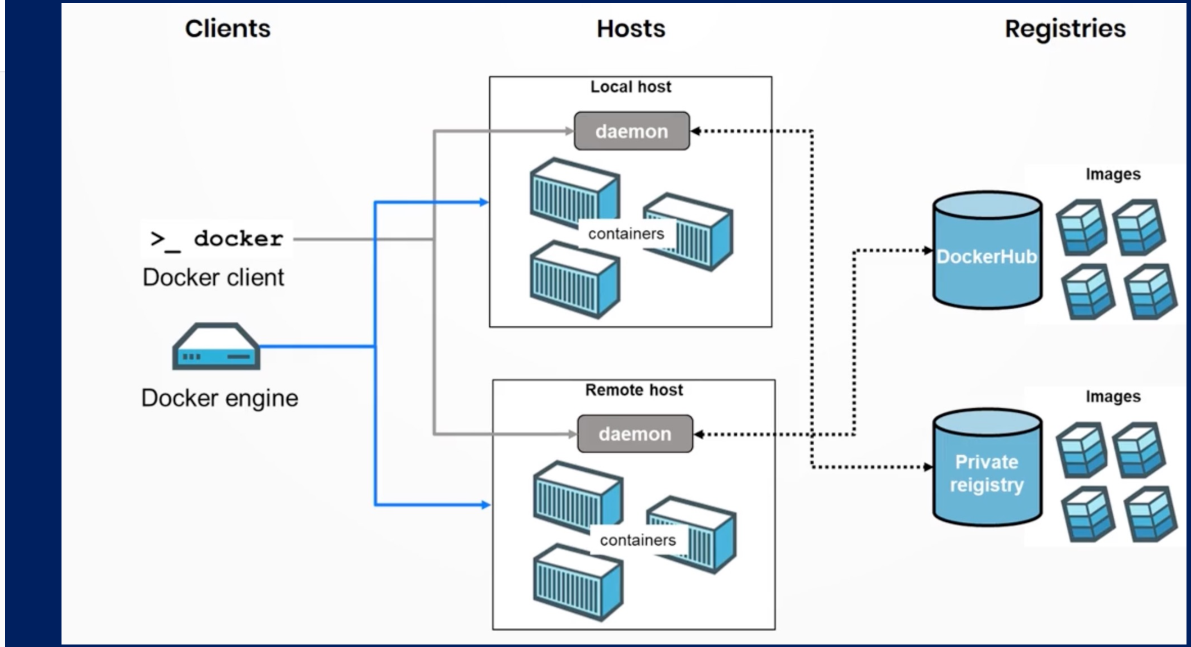
stable-perl stable perl mainline-perl mainline latest 1.23.3-perl 1.23.3 1.23-perl 1.23

About Official Images

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

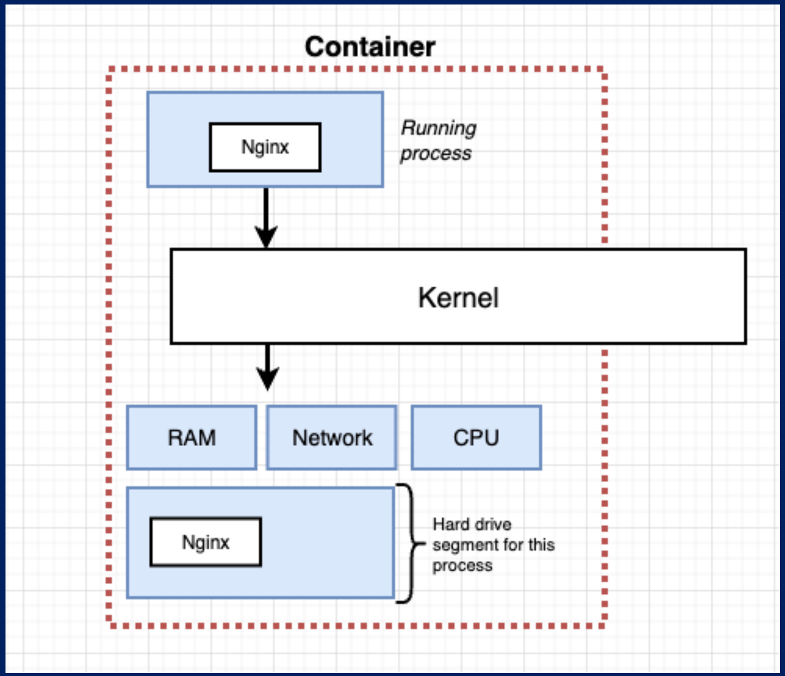
Why Official Images?

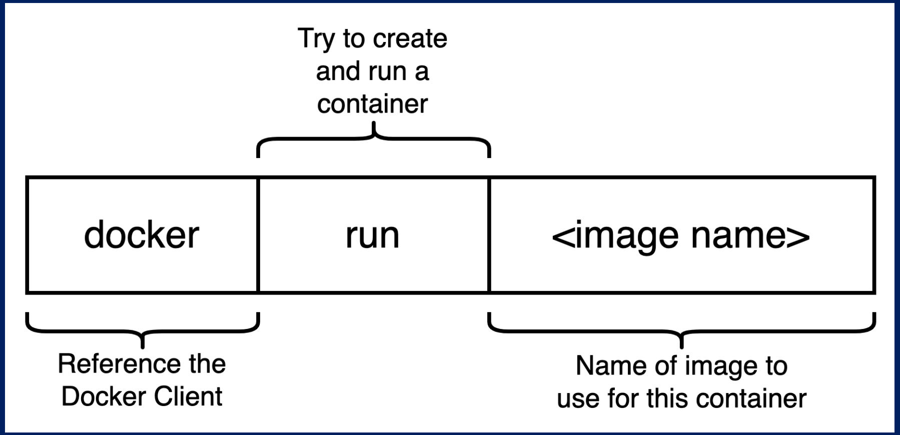
These images have clear documentation, promote best practices, and are designed for the most common use cases.

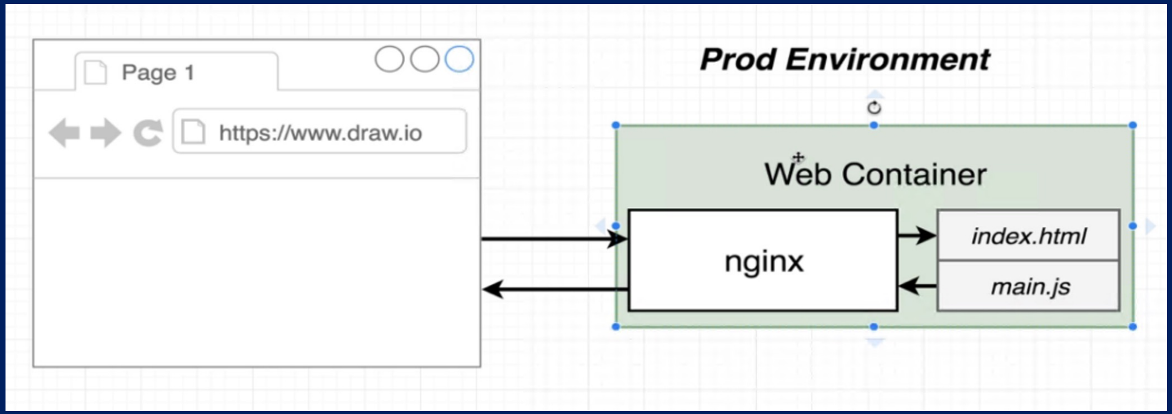


Run – start a container

```
▶ 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
```



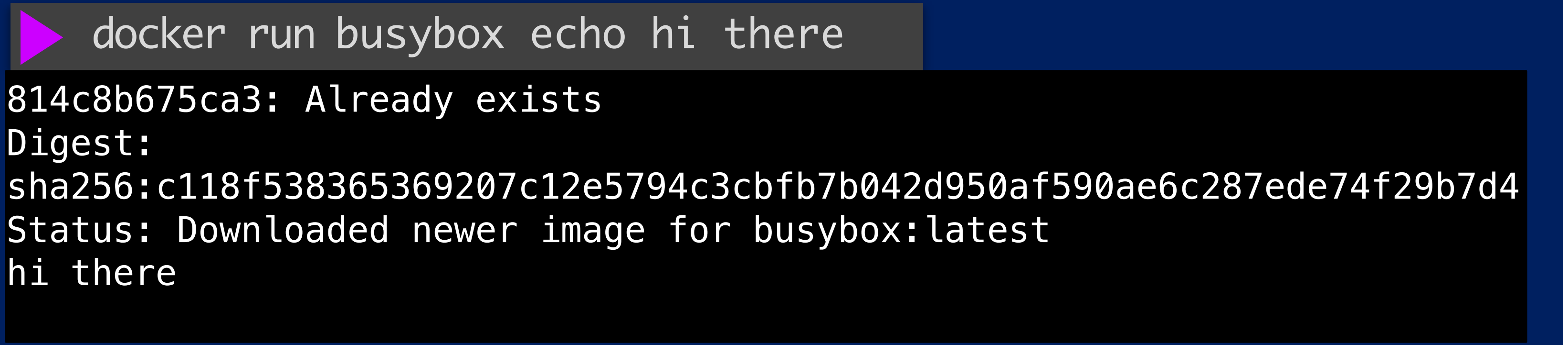

Try to create and run a container
docker
run
<image name>
Reference the Docker Client
Name of image to use for this container



```
▶ docker run ubuntu
▶ docker ps
CONTAINER ID      IMAGE            COMMAND          CREATED          STATUS                           PORTS
```

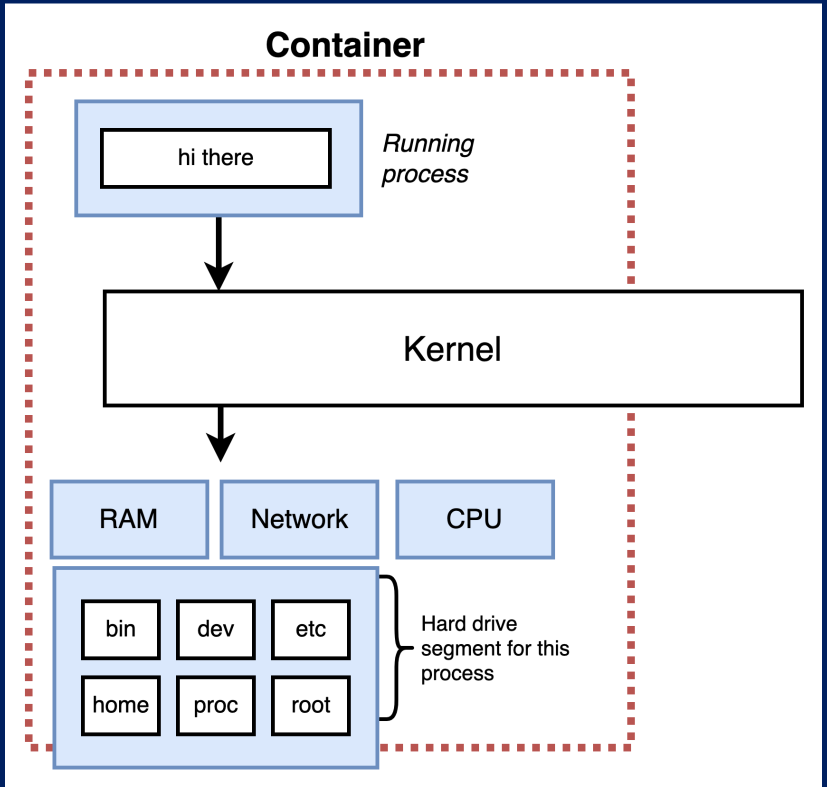
```
▶ docker ps -a
CONTAINER ID      IMAGE            COMMAND          CREATED              STATUS                        PORTS
45aacca36850      ubuntu           "/bin/bash"      43 seconds ago       Exited (0) 41 seconds ago
```

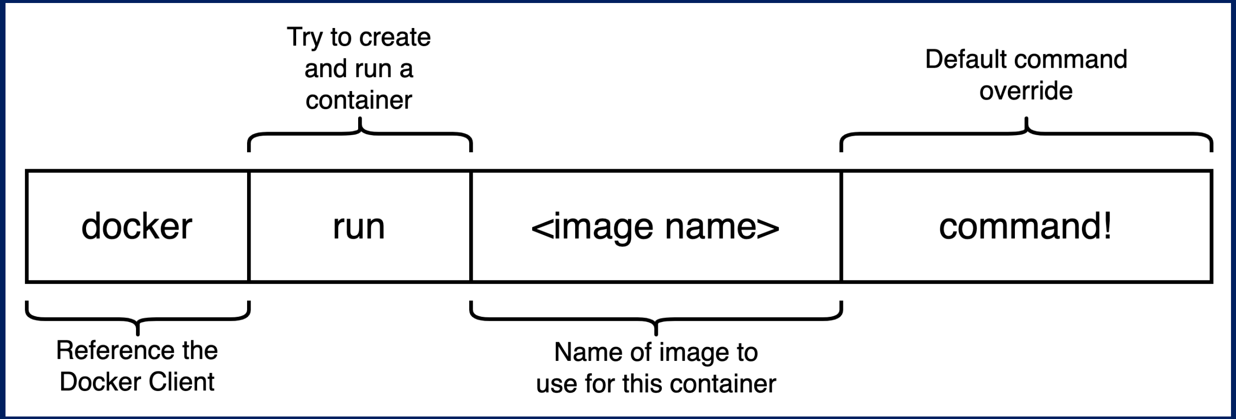
Run – with command

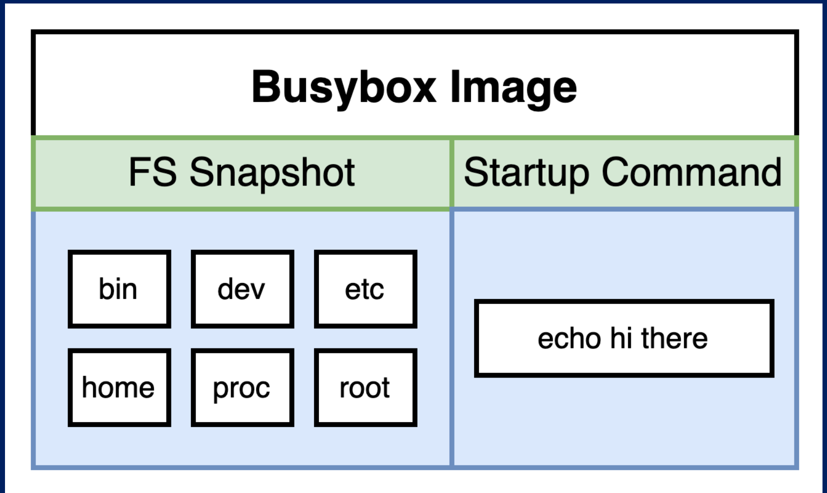


```
docker run busybox echo hi there
```

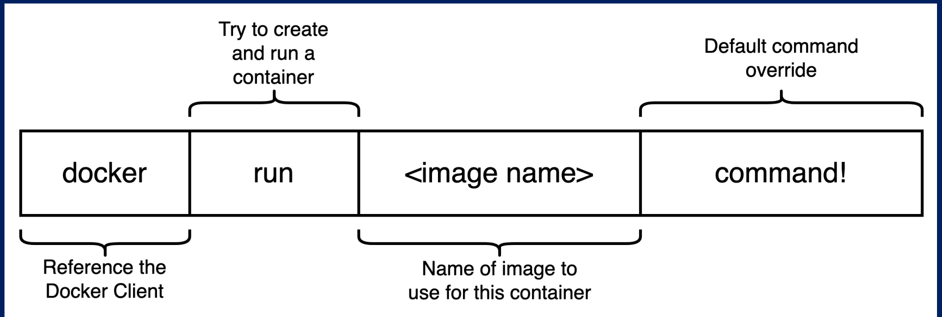
```
814c8b675ca3: Already exists
Digest:
sha256:c118f538365369207c12e5794c3cbfb7b042d950af590ae6c287ede74f29b7d4
Status: Downloaded newer image for busybox:latest
hi there
```




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!

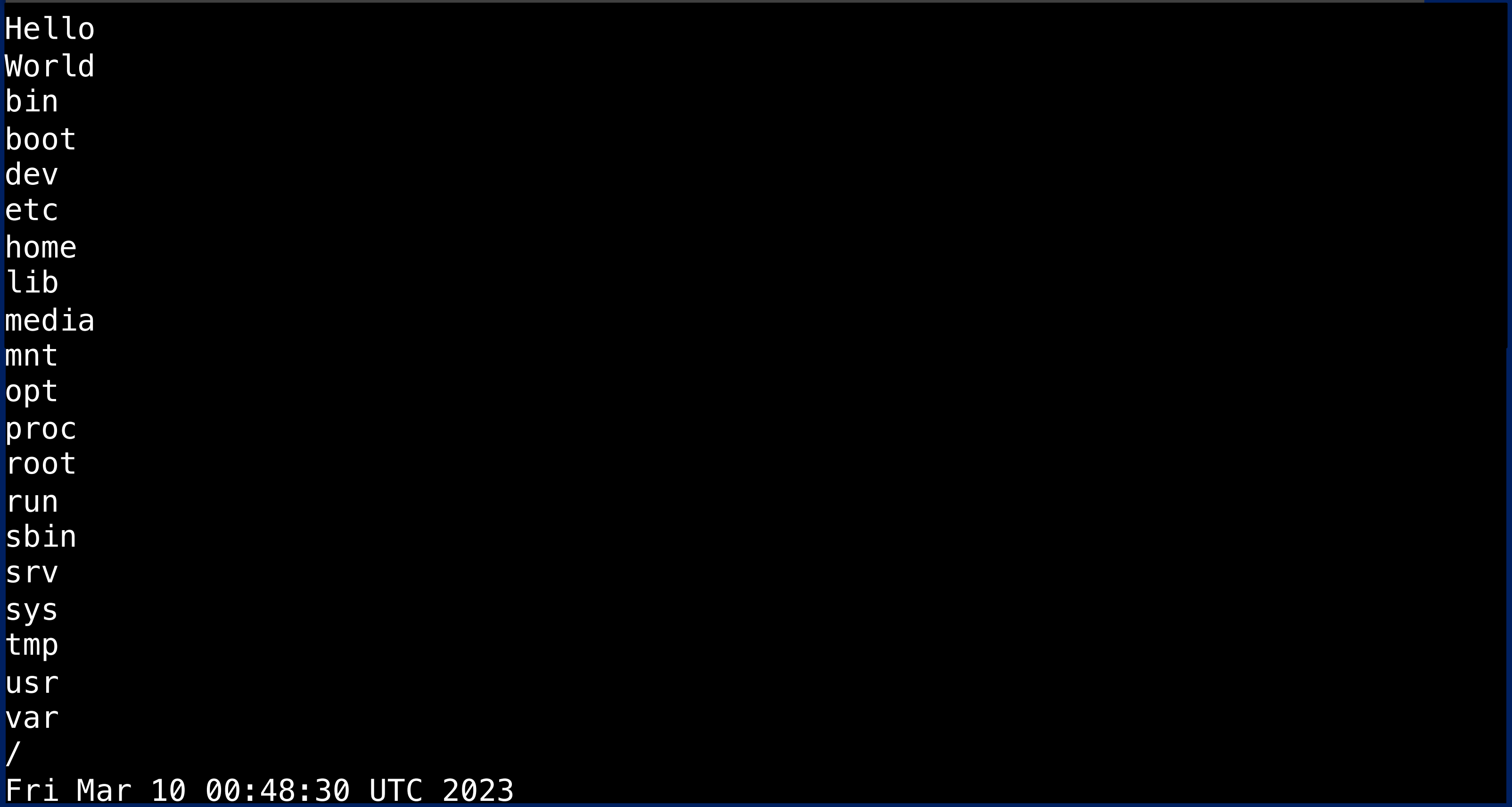
Reference the Docker Client

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

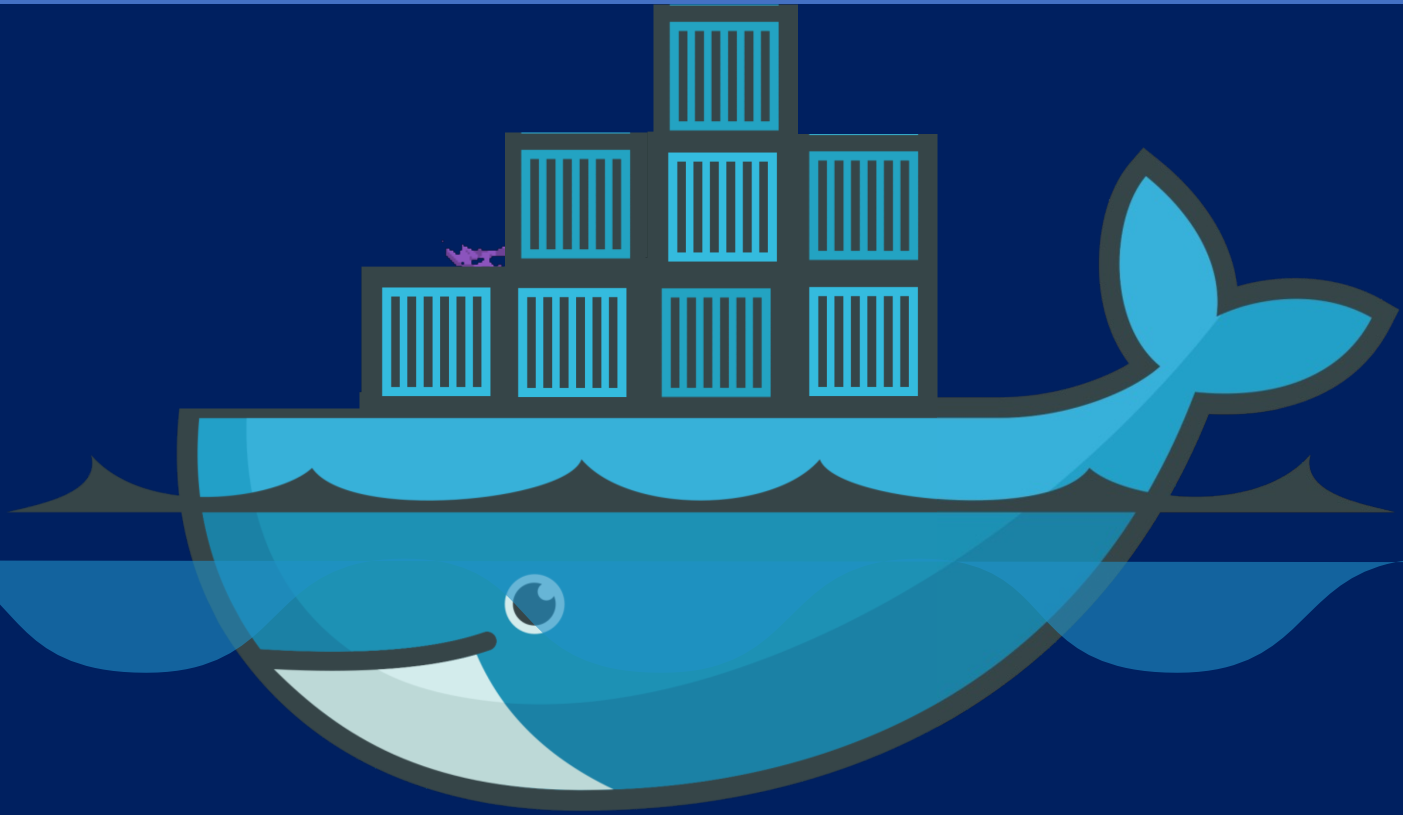
```
docker run nginx
```

```
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
```

LAB 3: Docker Port Mapping



Run – PORT mapping

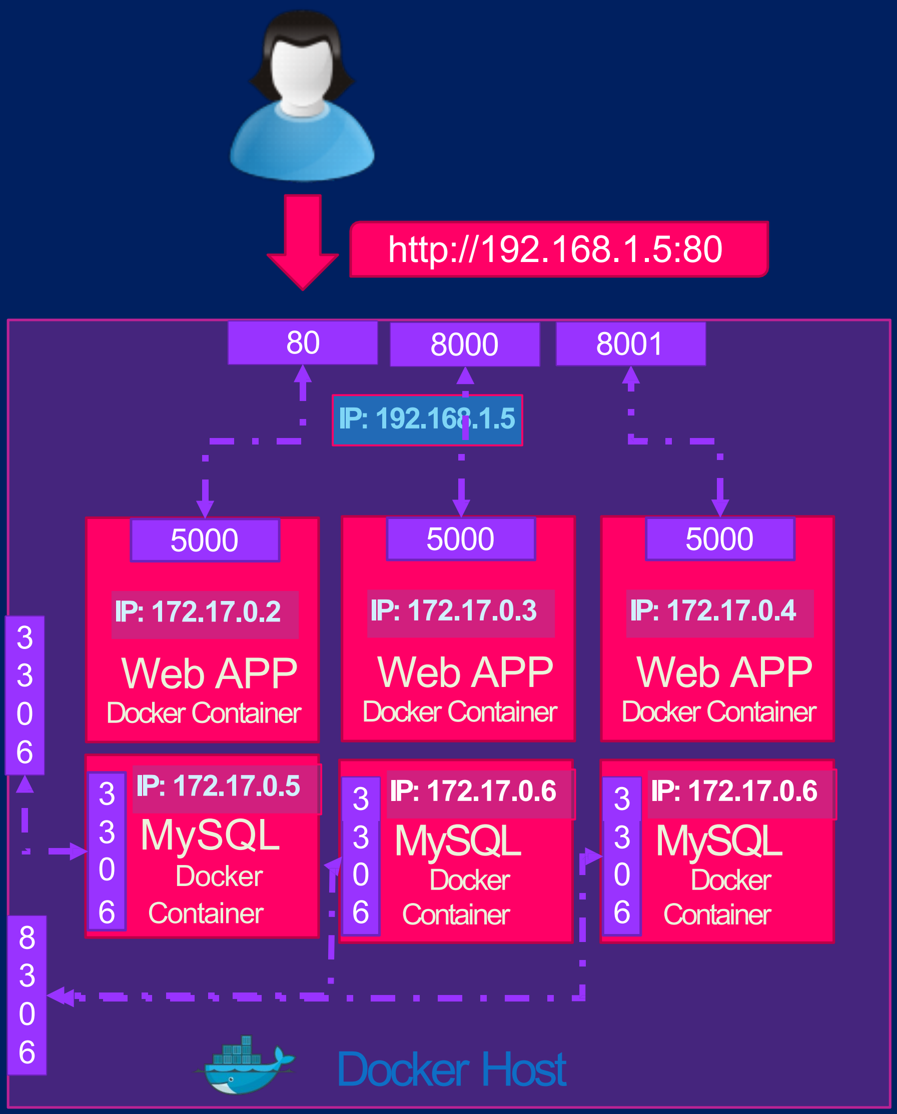
```
docker run myname/webapp
```

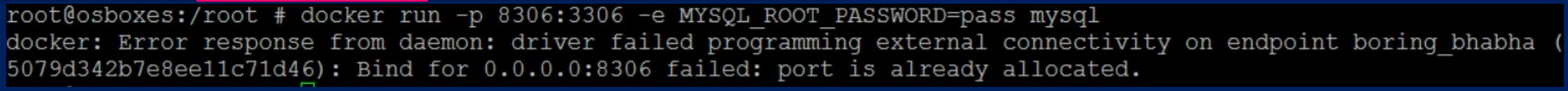
```
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

http://172.17.0.2:5000

Internal IP

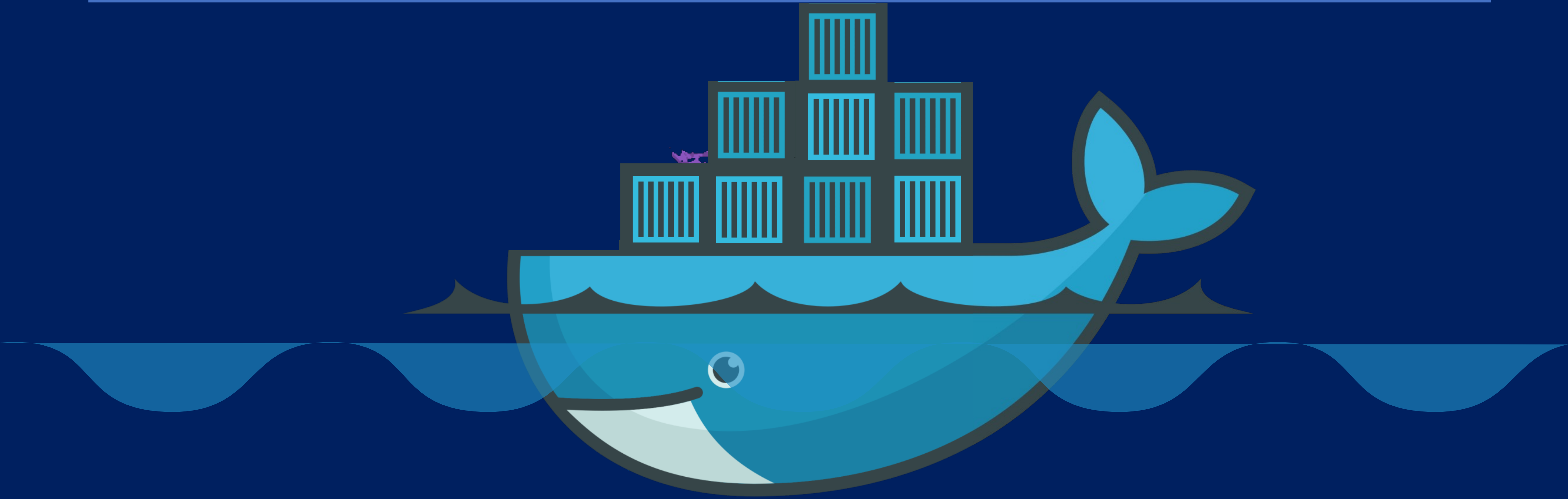
```
docker run -p 80:5000 myname/simple-webapp

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
```



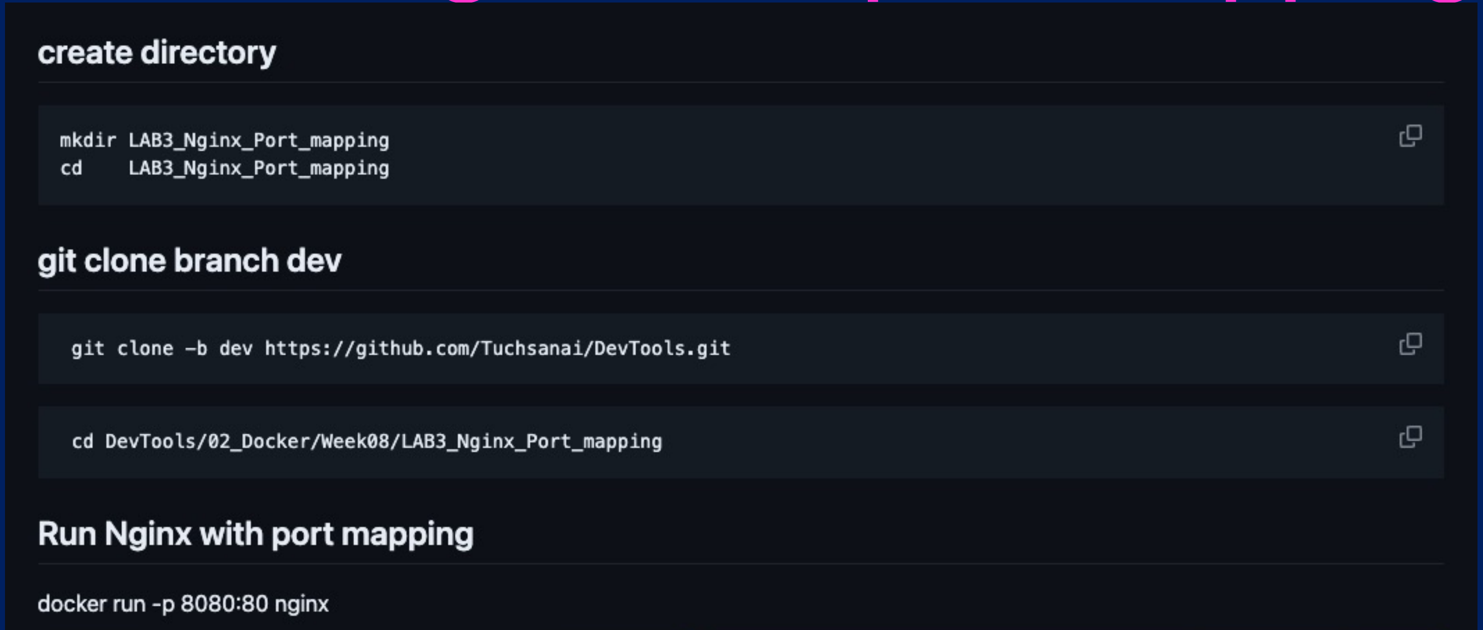


```
root@osboxes:/root # docker run -p 8306:3306 -e MYSQL_ROOT_PASSWORD=pass mysql
docker: Error response from daemon: driver failed programming external connectivity on endpoint boring_bhabha (
5079d342b7e8ee11c71d46): Bind for 0.0.0.0:8306 failed: port is already allocated.
```

Docker run from Repository



LAB 3 : Run Nginx with port mapping



create directory

```
mkdir LAB3_Nginx_Port_mapping
cd    LAB3_Nginx_Port_mapping
```

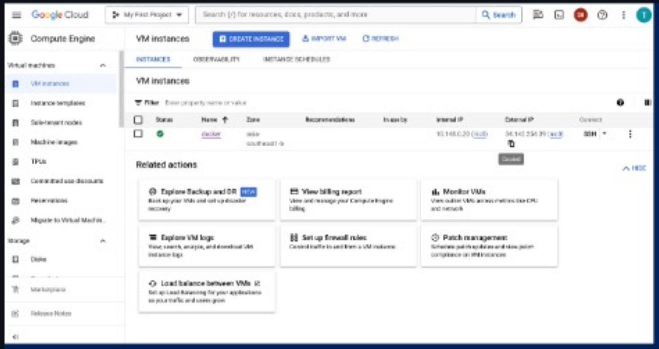
git clone branch dev

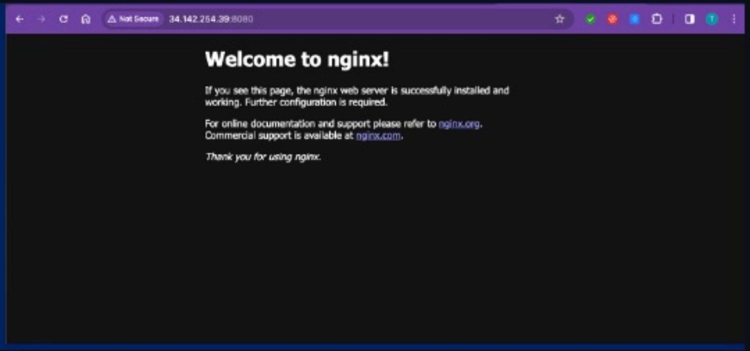
```
git clone -b dev https://github.com/Tuchsanai/DevTools.git
```

```
cd DevTools/02_Docker/Week08/LAB3_Nginx_Port_mapping
```

Run Nginx with port mapping

docker run -p 8080:80 nginx





Welcome to nginx!

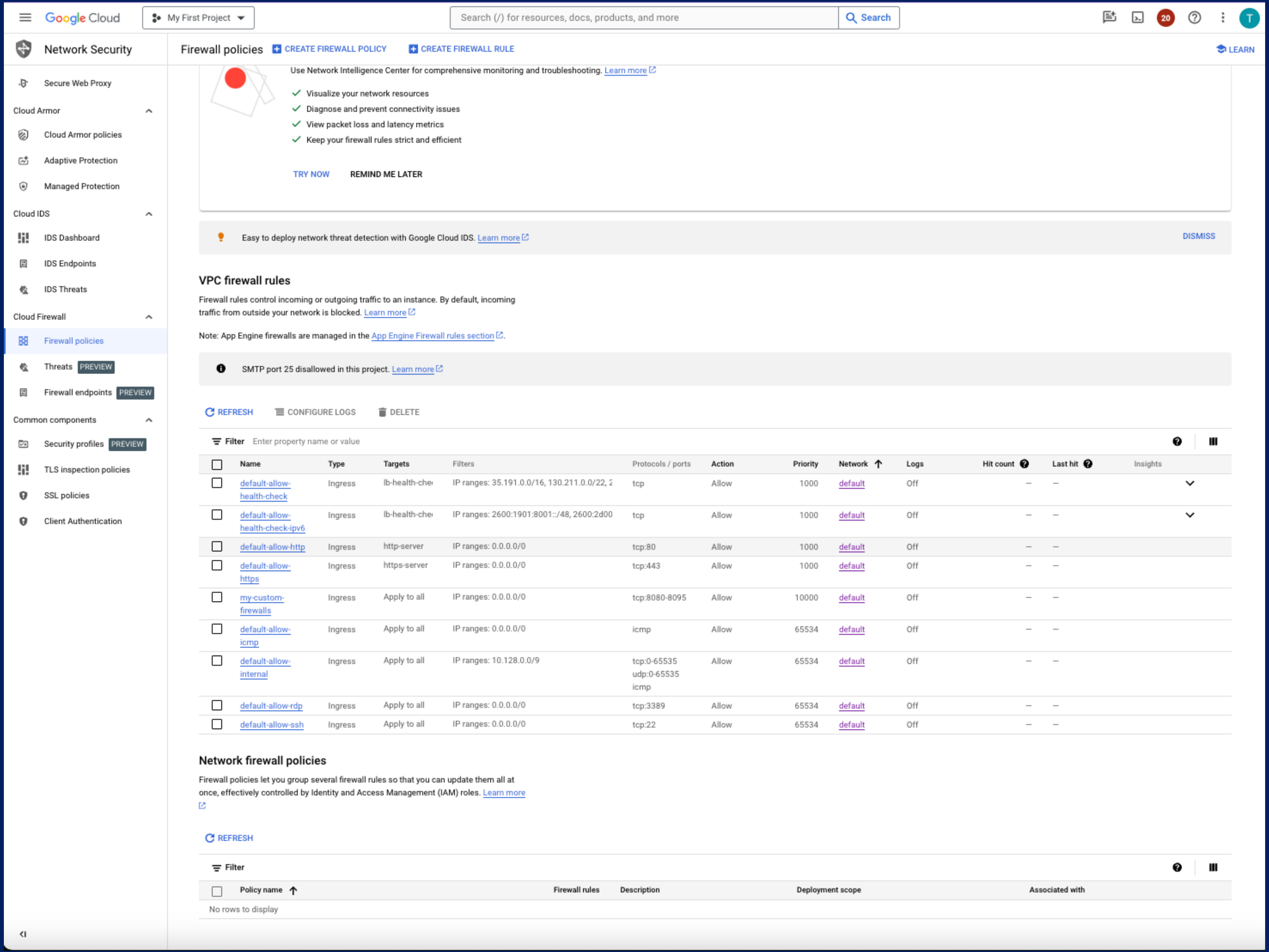
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

https://github.com/Tuchsanai/DevTools/tree/main/02_Docker/Week08/LAB3_Nginx_Port_mapping

Firewall policies



Use Network Intelligence Center for comprehensive monitoring and troubleshooting. Learn more

- ✓ Visualize your network resources
- ✓ Diagnose and prevent connectivity issues
- ✓ View packet loss and latency metrics
- ✓ Keep your firewall rules strict and efficient

Easy to deploy network threat detection with Google Cloud IDS. Learn more

VPC firewall rules

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. Learn more

Note: App Engine firewalls are managed in the App Engine Firewall rules section.

SMTP port 25 disallowed in this project. Learn more

Filter Enter property name or value

□	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network ↑	Logs	Hit count	Last hit	Insights
□	default-allow-health-check	Ingress	lb-health-che	IP ranges: 35.191.0.0/16, 130.211.0.0/22, 2	tcp	Allow	1000	default	Off	–	–	⌄
□	default-allow-health-check-ipv6	Ingress	lb-health-che	IP ranges: 2600:1901:8001::/48, 2600:2d00	tcp	Allow	1000	default	Off	–	–	⌄
□	default-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000	default	Off	–	–	
□	default-allow-https	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000	default	Off	–	–	
□	my-custom-firewalls	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:8080-8095	Allow	10000	default	Off	–	–	
□	default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default	Off	–	–	
□	default-allow-internal	Ingress	Apply to all	IP ranges: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	Allow	65534	default	Off	–	–	
□	default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	default	Off	–	–	
□	default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	default	Off	–	–	

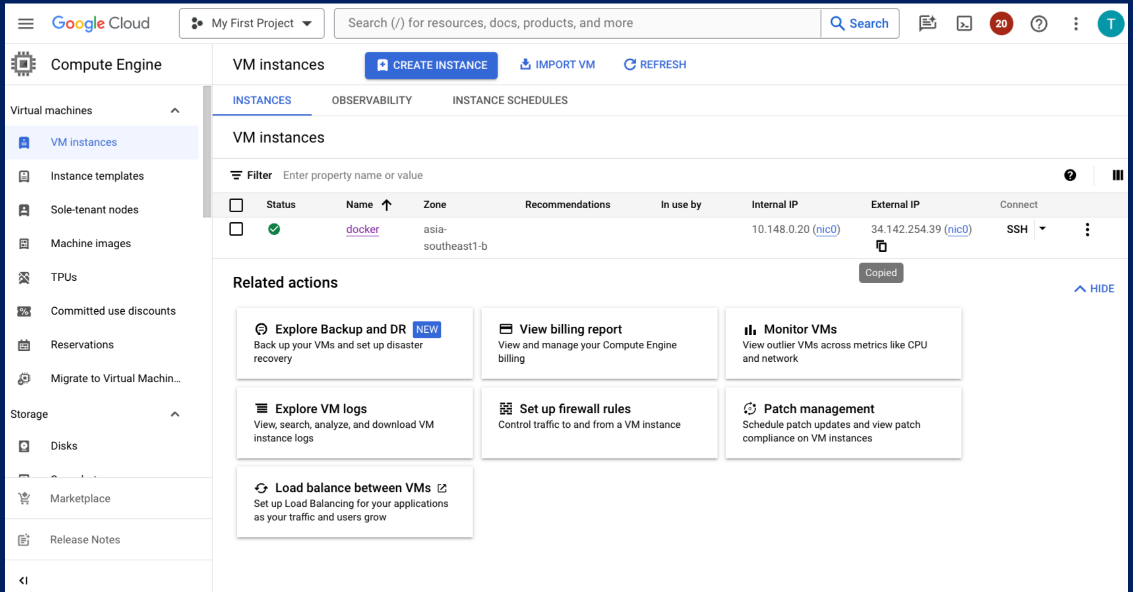
Network firewall policies

Firewall policies let you group several firewall rules so that you can update them all at once, effectively controlled by Identity and Access Management (IAM) roles. Learn more

Filter

□	Policy name ↑	Firewall rules	Description	Deployment scope	Associated with

No rows to display



VM instances

□	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	docker	asia-southeast1-b			10.148.0.20 (nic0)	34.142.254.39 (nic0)	SSH

Related actions

Explore Backup and DR NEW
Back up your VMs and set up disaster recovery

View billing report
View and manage your Compute Engine billing

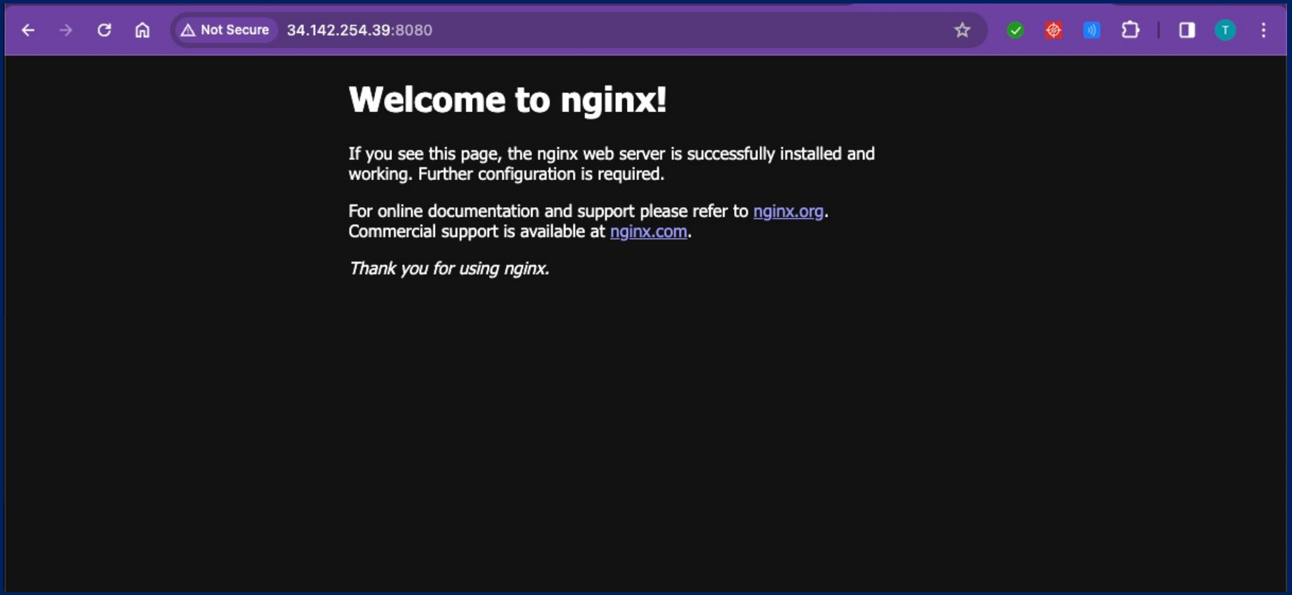
Monitor VMs
View outlier VMs across metrics like CPU and network

Explore VM logs
View, search, analyze, and download VM instance logs

Set up firewall rules
Control traffic to and from a VM instance

Patch management
Schedule patch updates and view patch compliance on VM instances

Load balance between VMs
Set up Load Balancing for your applications as your traffic and users grow



Welcome to nginx!

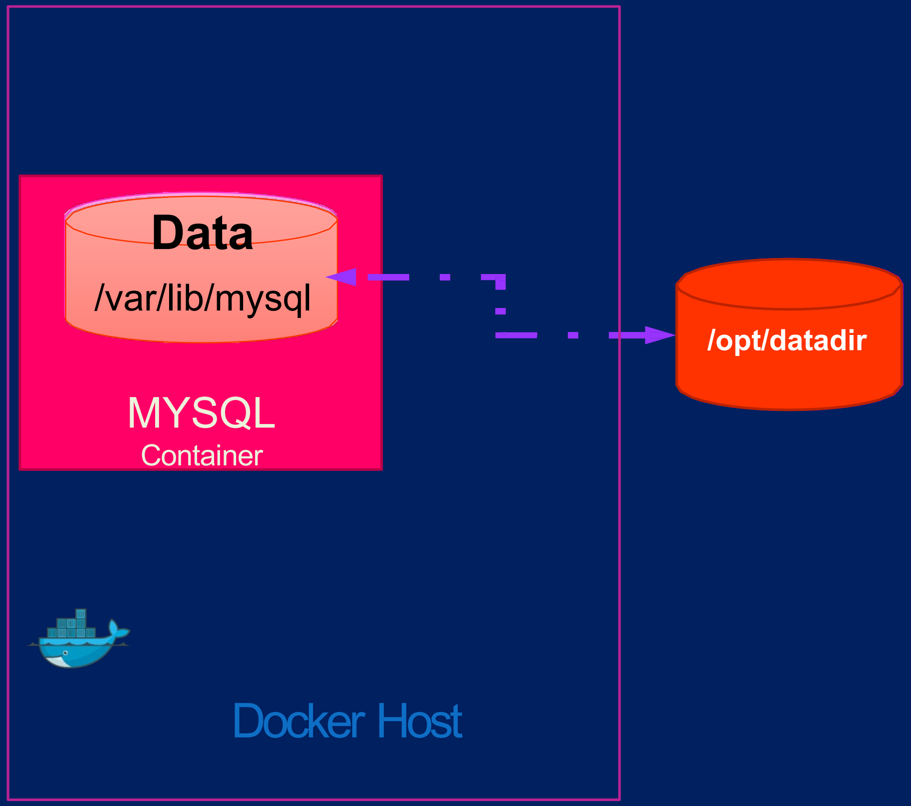
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

LAB4 : RUN – Volume mapping

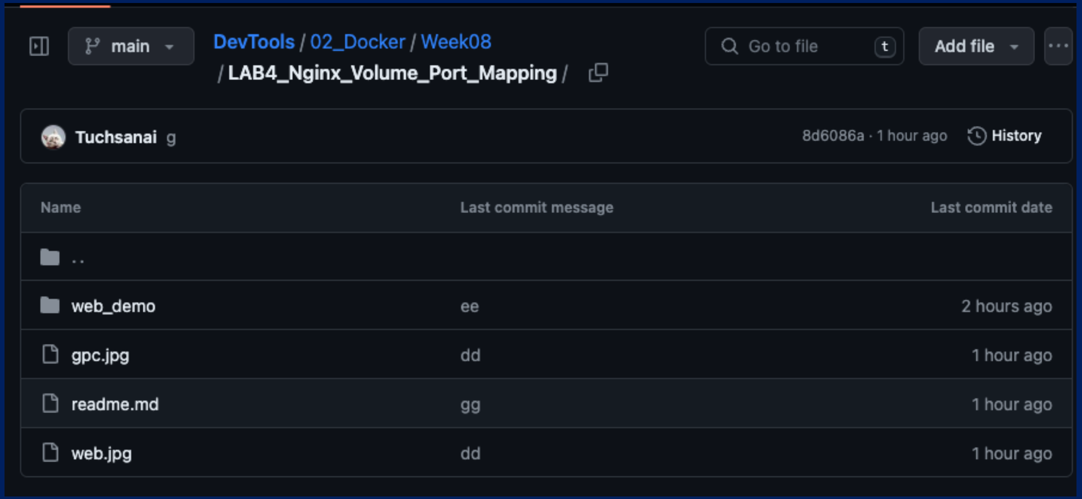
```
docker run -v /opt/datadir:/var/lib/mysql mysql
```



https://github.com/Tuchsanai/DevTools/tree/main/02_Docker/Week08/LAB4_Nginx_Volume_Port_Mapping

LAB4 :Run Nginx with Volume and Port Mapping

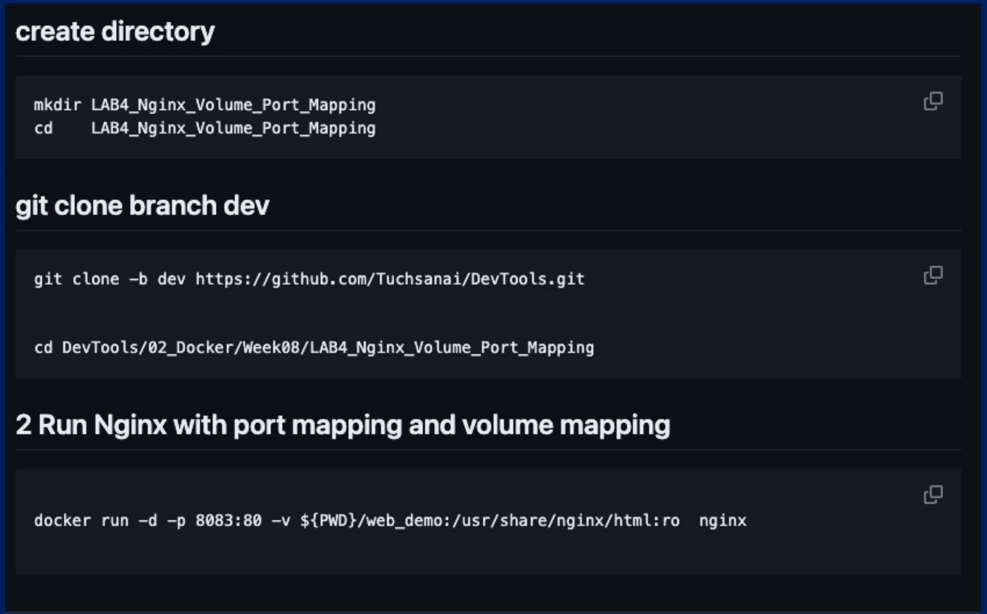
```
docker run -d -p 8080:80 -v ${PWD}/web_demo:/usr/share/nginx/html  nginx
```



DevTools / 02_Docker / Week08 / LAB4_Nginx_Volume_Port_Mapping /

Tuchsanai g   8d6086a · 1 hour ago

Name	Last commit message	Last commit date
..		
web_demo	ee	2 hours ago
gpc.jpg	dd	1 hour ago
readme.md	gg	1 hour ago
web.jpg	dd	1 hour ago



create directory

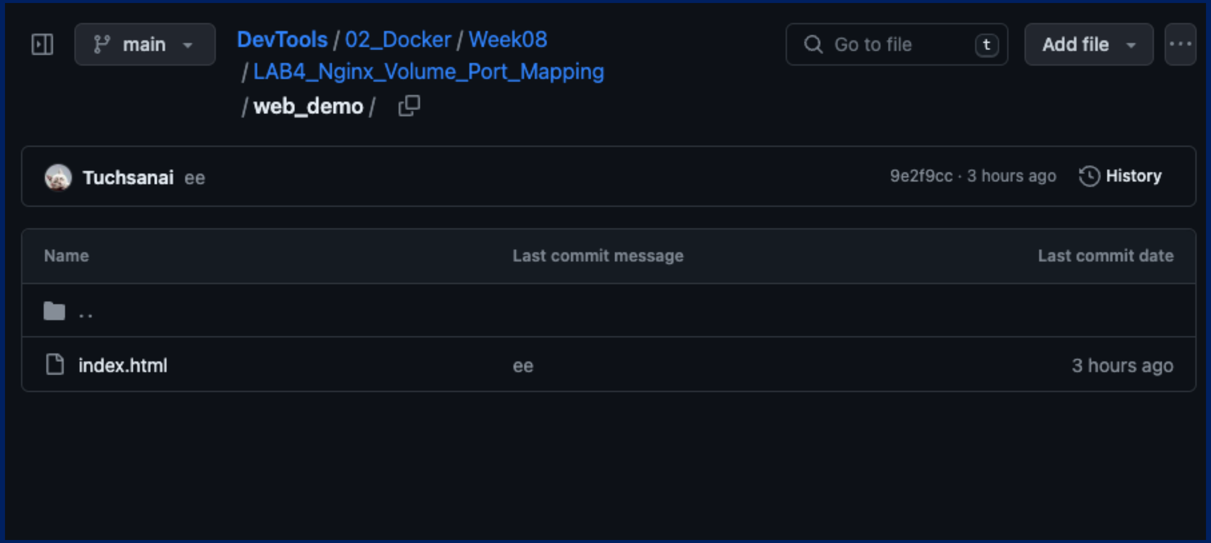
```
mkdir LAB4_Nginx_Volume_Port_Mapping
cd    LAB4_Nginx_Volume_Port_Mapping
```

git clone branch dev

```
git clone -b dev https://github.com/Tuchsanai/DevTools.git

cd DevTools/02_Docker/Week08/LAB4_Nginx_Volume_Port_Mapping
```

2 Run Nginx with port mapping and volume mapping

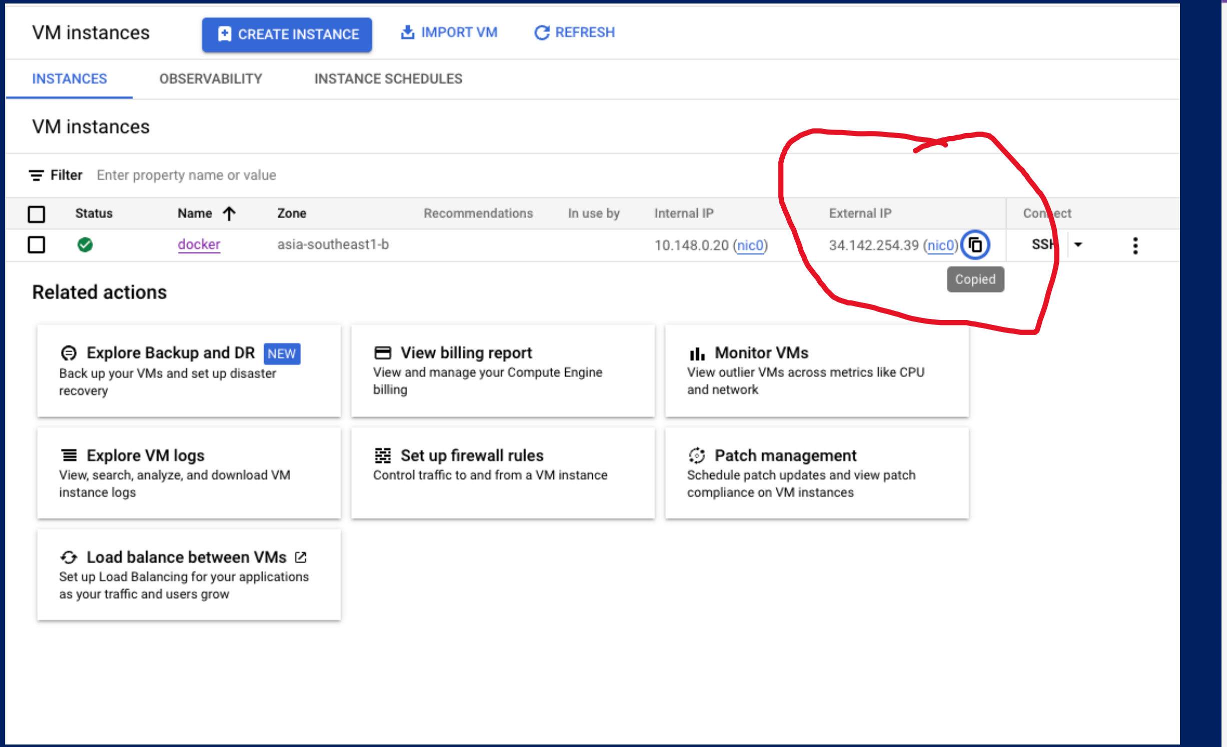
```
docker run -d -p 8083:80 -v ${PWD}/web_demo:/usr/share/nginx/html:ro  nginx
```



DevTools / 02_Docker / Week08 / LAB4_Nginx_Volume_Port_Mapping / web_demo /

Tuchsanai ee   9e2f9cc · 3 hours ago

Name	Last commit message	Last commit date
..		
index.html	ee	3 hours ago



VM instances

CREATE INSTANCE · IMPORT VM · REFRESH

INSTANCES · OBSERVABILITY · INSTANCE SCHEDULES

VM instances

Filter Enter property name or value

	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	docker	asia-southeast1-b			10.148.0.20 (nic0)	34.142.254.39 (nic0)	SSH

Copied

Related actions

Explore Backup and DR NEW
Back up your VMs and set up disaster recovery

View billing report
View and manage your Compute Engine billing

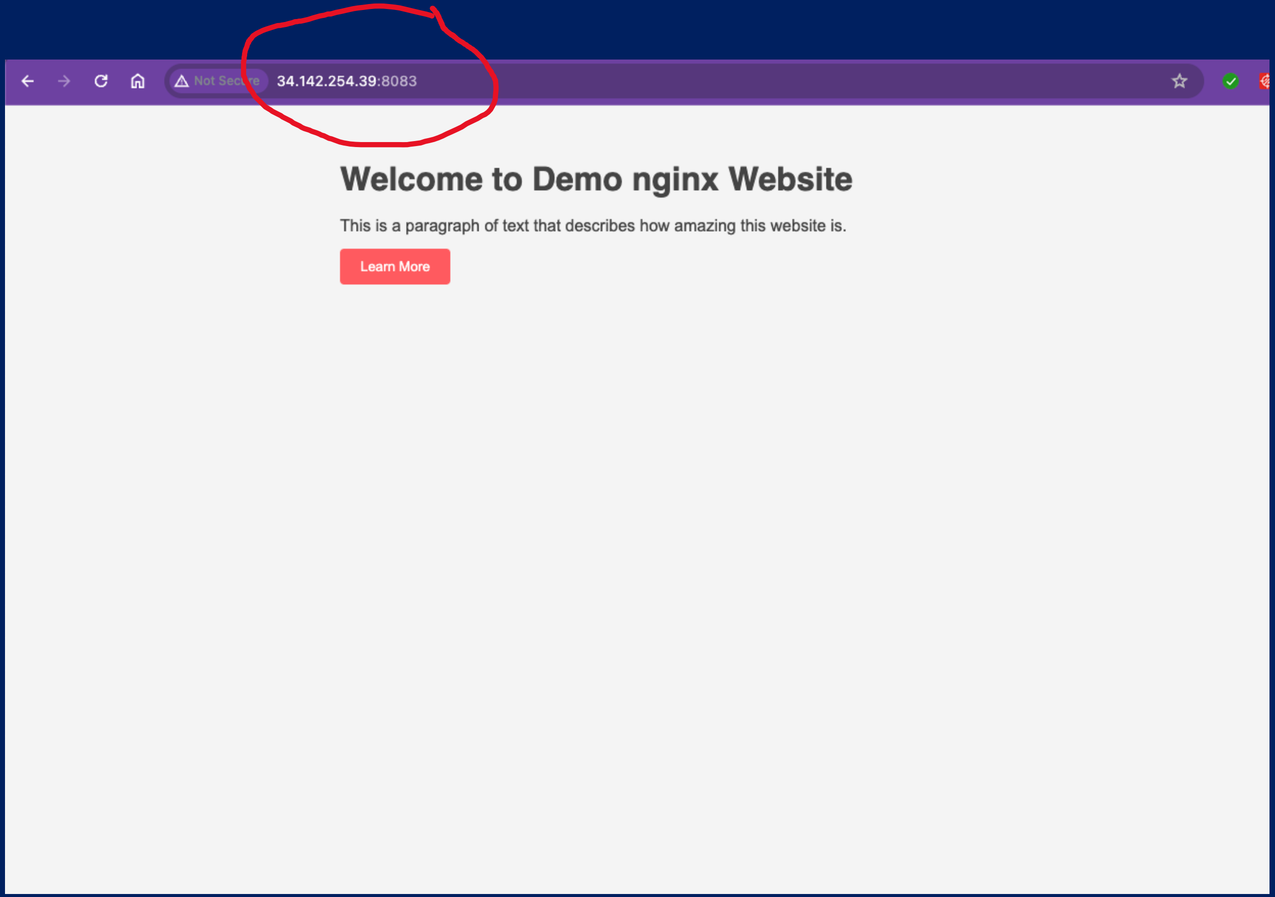
Monitor VMs
View outlier VMs across metrics like CPU and network

Explore VM logs
View, search, analyze, and download VM instance logs

Set up firewall rules
Control traffic to and from a VM instance

Patch management
Schedule patch updates and view patch compliance on VM instances

Load balance between VMs
Set up Load Balancing for your applications as your traffic and users grow



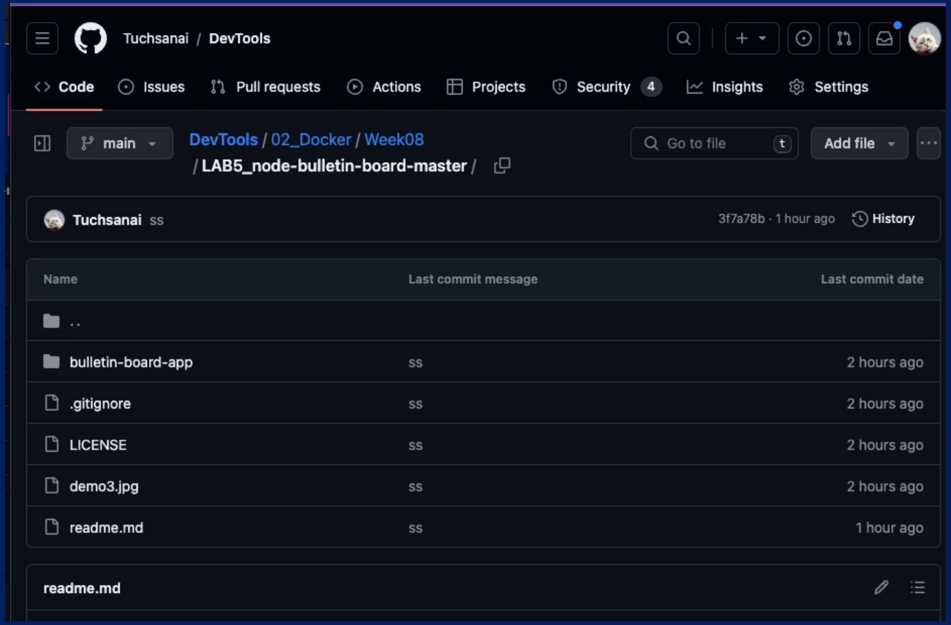
Not Secure 34.142.254.39:8083

Welcome to Demo nginx Website

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

Learn More

LAB5 : Build and Run Docker Image

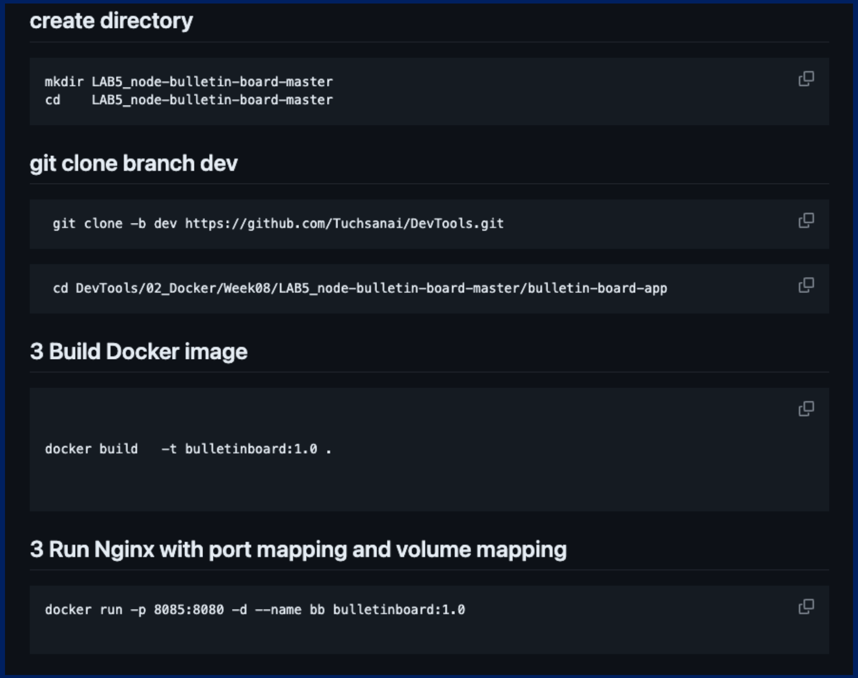


DevTools / 02_Docker / Week08 / LAB5_node-bulletin-board-master /

Tuchsanai ss 3f7a78b · 1 hour ago

Name	Last commit message	Last commit date
..		
bulletin-board-app	ss	2 hours ago
.gitignore	ss	2 hours ago
LICENSE	ss	2 hours ago
demo3.jpg	ss	2 hours ago
readme.md	ss	1 hour ago

readme.md



create directory

```
mkdir LAB5_node-bulletin-board-master
cd    LAB5_node-bulletin-board-master
```

git clone branch dev

```
git clone -b dev https://github.com/Tuchsanai/DevTools.git
```

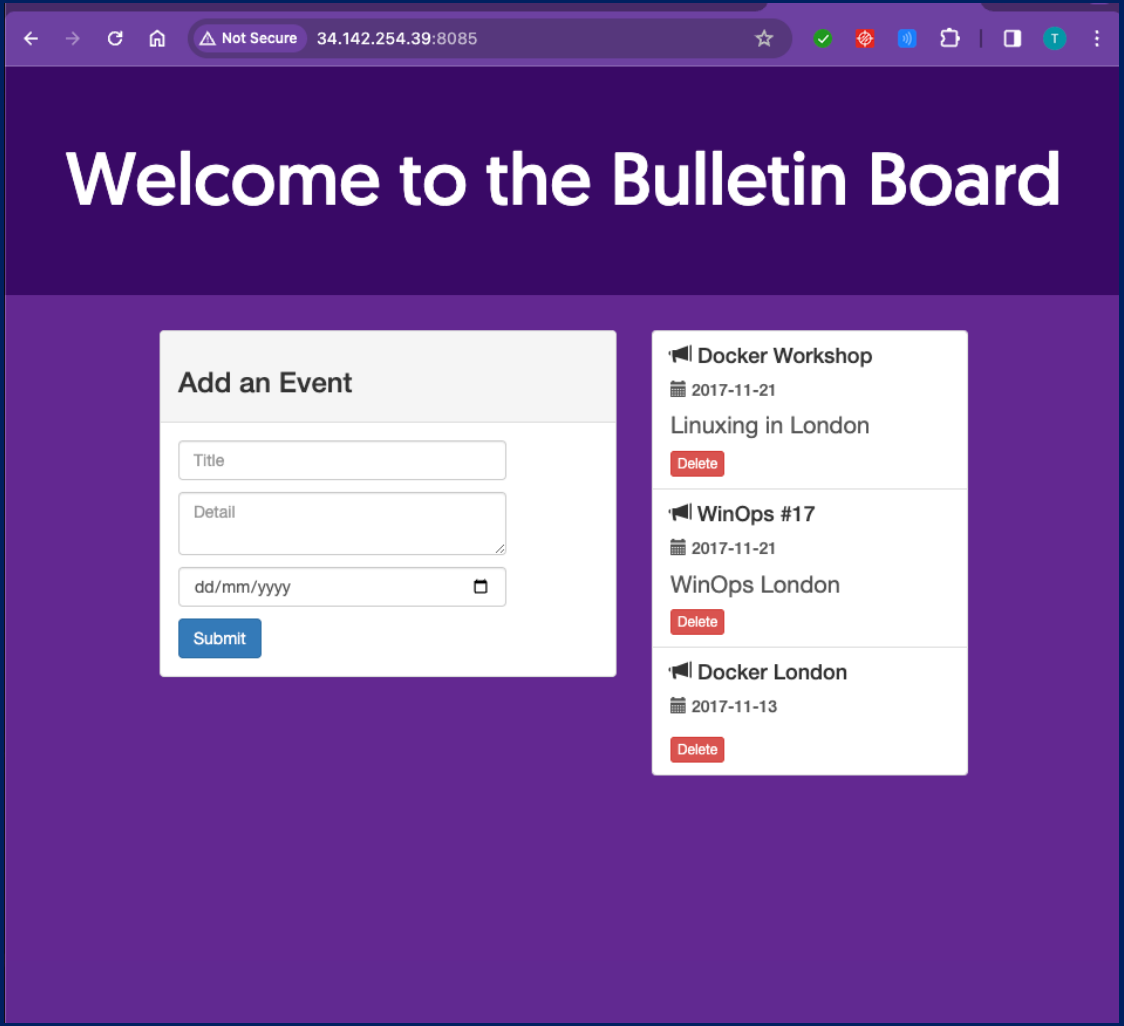
```
cd DevTools/02_Docker/Week08/LAB5_node-bulletin-board-master/bulletin-board-app
```

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

Docker Workshop
2017-11-21
Linuxing in London

WinOps #17
2017-11-21
WinOps London

Docker London
2017-11-13

https://github.com/Tuchsanai/DevTools/tree/main/02_Docker/Week08/LAB5_node-bulletin-board-master