

# Docker Tutorial for Beginners

<https://www.youtube.com/watch?v=3ciBn73dDE>

The image shows a screenshot of a presentation slide. On the left, there is a list of topics with blue circular bullet points. On the right, there is a large illustration of a blue whale carrying several shipping containers on its back, swimming in water.

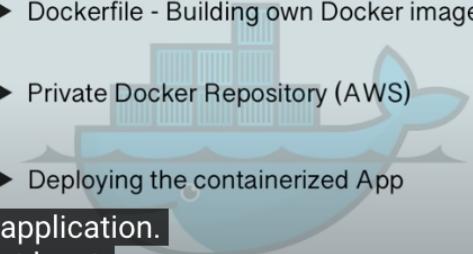
- ▶ What is Docker? What is a Container?
- ▶ Docker vs. Virtual Machine
- ▶ Docker Installation
- ▶ Main Commands
- ▶ Debugging a Container

After that,

**Demo Project:**

- ▶ What is Docker? What is a Container?
- ▶ Docker vs. Virtual Machine
- ▶ Docker Installation
- ▶ Main Commands
- ▶ Debugging a Container
- ▶ Developing with Containers
- ▶ Docker Compose - Running multiple services
- ▶ Dockerfile - Building own Docker image
- ▶ Private Docker Repository (AWS)
- ▶ Deploying the containerized App

containerized application.  
Last but not least,



1:22 / 2:46:14 • Intro and Course Overview > CC Settings Share Close

**Demo Project:**

- ▶ What is Docker? What is a Container?
- ▶ Docker vs. Virtual Machine
- ▶ Docker Installation
- ▶ Main Commands
- ▶ Debugging a Container
- ▶ Volumes - Persisting Data
- ▶ Developing with Containers
- ▶ Docker Compose - Running multiple services
- ▶ Dockerfile - Building own Docker image
- ▶ Private Docker Repository (AWS)
- ▶ Deploying the containerized App
- ▶ Volumes Demo





# What is Docker?

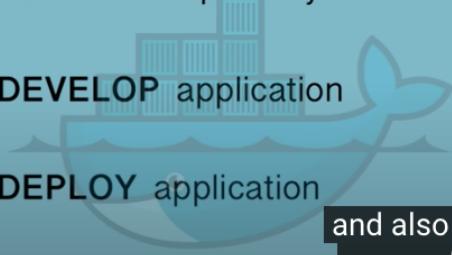


## Overview

### What is Docker?



- ✓ What is a Container and what problems does it solve?
- ✓ Container Repository
- ✓ DEVELOP application
- ✓ DEPLOY application



and also how they solve some of the problems



## What is a Container?

- ▶ A way to **package** application with **all the necessary dependencies** and **configuration**



- ▶ **Portable artifact**, easily shared and moved around

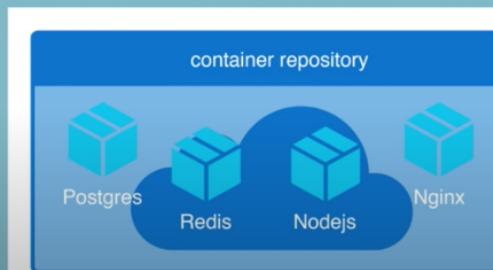


- ▶ Makes development and deployment **more efficient**

## Where do containers live?

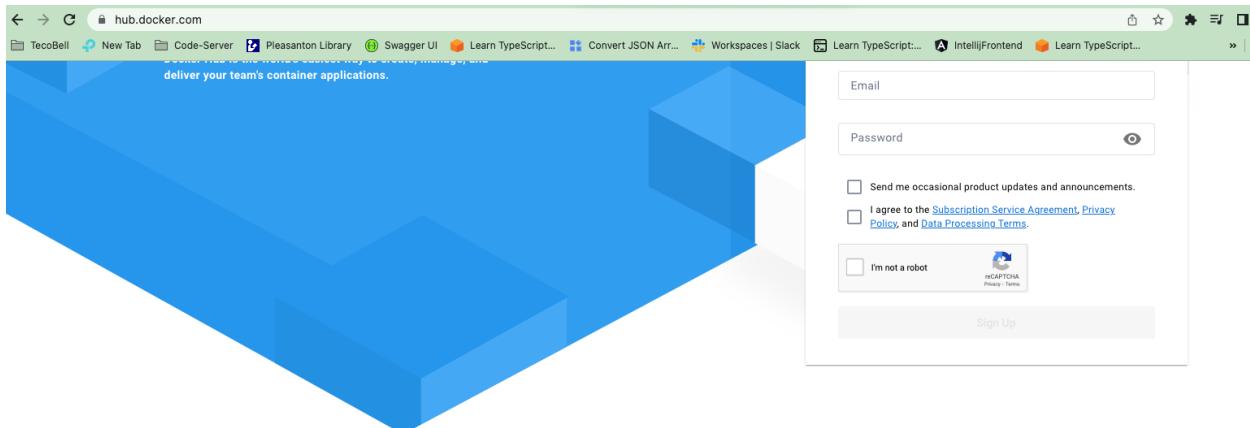
- ▶ Container Repository
- ▶ Private repositories
- ▶ Public repository for Docker

DockerHub



<http://hub.docker.com>

docker images- publicly available



Docker Hub is the world's largest  
library and community for container images

Browse over 100,000 container images from software vendors, open-source projects, and the community.



A screenshot of a video player interface. The main content area has a teal gradient background. In the center, the text "How containers improved.." is displayed above a white rounded rectangle containing the words "Application Development". At the bottom of the screen, there is a dark footer bar with various video control icons (play, pause, volume, etc.) and a timestamp indicating the video is 4:59 / 2:46:14 long. The title of the video is "What is Docker? &gt;"



Install required software on local development environment



PostgreSQL  
v9.3



Developer



Redis  
v5.0

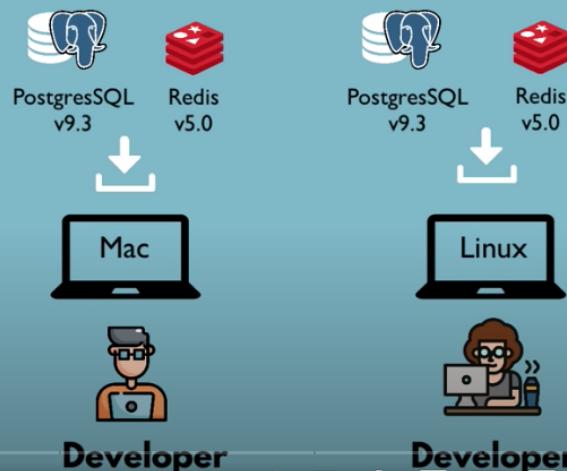


Developer

## Application Development

### Before containers

- ▶ Installation process different on each OS environment
- ▶ Many steps where something could go wrong



6:18 / 2:46:14 • What is Docker? >

## Application Development

### After containers



- ▶ own isolated environment
- ▶ packaged with all needed configuration

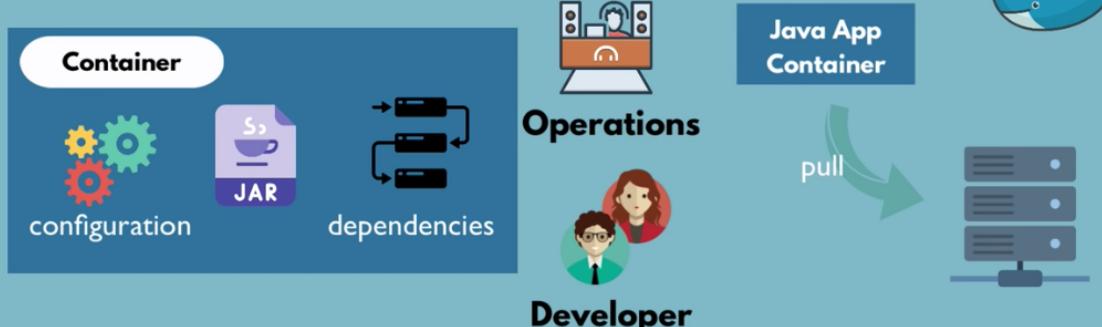


RECOMMENDED

## Application Deployment

## After containers

- ▶ Developers and Operations work together to package the application in a container
- ▶ No environmental configuration needed on server  - except Docker Runtime



## Application Development

## After containers

- ▶ own isolated environment
- ▶ packaged with all needed configuration
- ▶ one command to install the app



**Application Development**

**After containers**



▶ own isolated environment

▶ packaged with all needed configuration

▶ one command to install the app

▶ run same app with 2 different versions

**Container**

- configuration
- PostgreSQL v9.3
- Start script

**PostgreSQL Container**

**PostgreSQL Container**

`> cmd docker run postgres execution...`





**Application Deployment**

**Before containers**



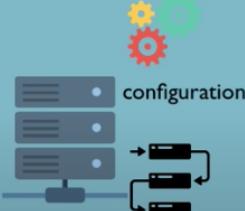
▶ configuration on the server needed

**Developer**

- JAR
- Deployment descriptor
- Database schema
- Configuration file



**Operations**



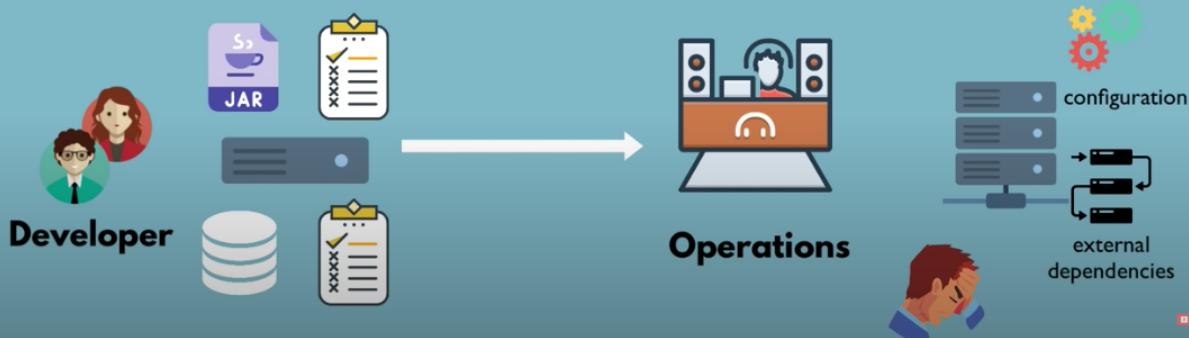
configuration

external dependencies

## Application Deployment

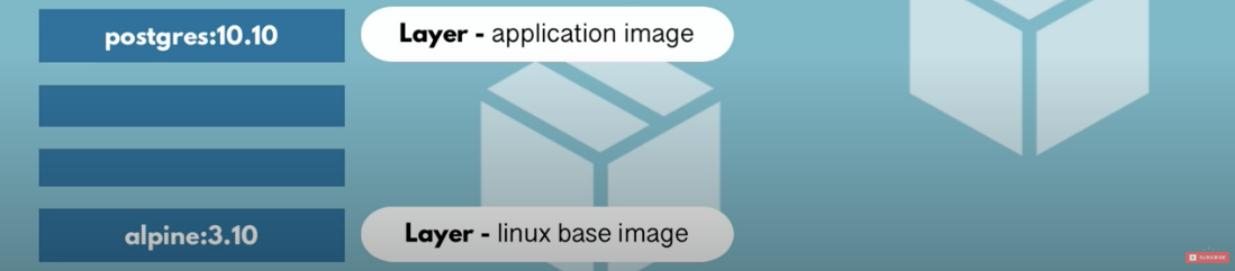
## Before containers

- ▶ configuration on the server needed
- ✖ dependency version conflicts
- ▶ textual guide of deployment
- ✖ misunderstandings



## What is a Container?

- ▶ Layers of images
- ▶ Mostly **Linux Base Image**, because small in size
- ▶ Application image on top



The screenshot shows the Docker Hub page for the 'postgres' repository. The 'DESCRIPTION' tab is selected, displaying a list of supported tags and their corresponding Dockerfile links. To the right, there are three prominent buttons: 'public repo', 'no login necessary', and 'no authentication'. A message box encourages users to log in to write reviews.

**Supported tags and respective Dockerfile links**

- 12-rc1, 12
- 12-rc1-alpine, 12-alpine
- 11.5, 11, latest
- 11.5-alpine, 11-alpine, alpine
- 10.10, 10
- 10.10-alpine, 10-alpine
- 9.6.15, 9.6, 9
- 9.6.15-alpine, 9.6-alpine, 9-alpine
- 9.5.19, 9.5

**public repo**

**no login necessary**

**no authentication**

Please log in to write a review of this product.

The screenshot shows a terminal window displaying the output of the 'docker ps' command. It lists various Docker containers, their status, and the commands they are running. The output includes:

```
Last login: Sat Sep 28 10:55:08 on ttys006
Nanas-MBP:~ nanabiz$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS          NAMES
8f91359f1fff: Pulling from library/postgres
c6115f5efcde: Pulling fs layer
28a9c19d8188: Pulling fs layer
2da4beb7be31: Waiting
fb9ca792da89: Waiting
cedc20991511: Waiting
b866c2f2559e: Waiting
5d4459cf6645c: Waiting
b59ec97820c9: Waiting
01e040230c2f: Waiting
d618b32512c9: Waiting
d694ad4e08d5: Waiting
f1fc54212826: Waiting
2debab4418fd: Waiting
```

```
Last login: Sat Sep 28 10:55:08 on ttys006
Nanas-MBP:~ nanabiz$ docker ps
CONTAINER ID      IMAGE             COMMAND           CREATED          STATUS          PORTS          NAMES
Nanas-MBP:~ nanabiz$ docker run postgres:9.6
Unable to find image 'postgres:9.6' locally
9.6: Pulling from library/postgres
8f91359f1fff: Pull complete
c6115f5efcde: Pull complete
28a9c19d8188: Pull complete
2da4beb7be31: Pull complete
fb9ca792da89: Pull complete
cedc20991511: Pull complete
b866c2f2559e: Pull complete
5d459cf6645c: Pull complete
b59ec97820c9: Downloading [=====] 30.76 MB/49.33 MB
01e040230c2f: Download complete
d618b32512c9: Download complete
d694ad4e08d5: Download complete
f1fc54212826: Download complete
2debab4418fd: Download complete
```

**advantage:** only different layers  
are downloaded

```
Last login: Sat Sep 28 10:55:08 on ttys006
Nanas-MBP:~ nanabiz$ docker ps
CONTAINER ID      IMAGE             COMMAND           CREATED          STATUS          PORTS          NAMES
Nanas-MBP:~ nanabiz$ docker run postgres:9.6
Unable to find image 'postgres:9.6' locally
9.6: Pulling from library/postgres
8f91359f1fff: Downloading [=====] 19.79 MB/22.51 MB
c6115f5efcde: Download complete
28a9c19d8188: Download complete
2da4beb7be31: Download complete
fb9ca792da89: Download complete
cedc20991511: Download complete
b866c2f2559e: Download complete
5d459cf6645c: Download complete
b59ec97820c9: Downloading [=====] 10.09 MB/49.33 MB
01e040230c2f: Download complete
d618b32512c9: Download complete
d694ad4e08d5: Download complete
f1fc54212826: Download complete
2debab4418fd: Download complete
```

**separate images are downloaded**

```
This will allow anyone with access to the
Postgres port to access your database. In
Docker's default configuration, this is
effectively any other container on the same
system.

Use "-e POSTGRES_PASSWORD=password" to set
it in "docker run".
*****
waiting for server to start....LOG:  database system was shut down at 2019-09-28 09:07:37 UTC
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
done
server started

/usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*
waiting for server to shut down....LOG:  received fast shutdown request
LOG:  aborting any active transactions
LOG:  autovacuum launcher shutting down
LOG:  shutting down
LOG:  database system is shut down
done
server stopped

PostgreSQL init process complete; ready for start up.

LOG:  database system was shut down at 2019-09-28 09:07:38 UTC
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
```

i

```
Postgres port to access your database. In
Docker's default configuration, this is
effectively any other container on the same
system.

Use "-e POSTGRES_PASSWORD=password" to set
it in "docker run".
*****
waiting for server to start....LOG:  database system was shut down at 2019-
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
done
server started

/usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*
waiting for server to shut down....LOG:  received fast shutdown request
LOG:  aborting any active transactions
LOG:  autovacuum launcher shutting down
LOG:  shutting down
LOG:  database system is shut down
done
server stopped

PostgreSQL init process complete; ready for start up.

LOG:  database system was shut down at 2019-09-28 09:07:38 UTC
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
▶ ▶ ◀ 15:44 / 2:46:14 • What is a Container? >
```

Docker ps will show all the running containers

```
— docker run postgres:9.6          ~ -- bash
Last login: Sat Sep 28 10:57:20 on ttys001
Nanas-MBP:~ nanabiz$ docker ps
CONTAINER ID        IMAGE       COMMAND      CREATED     STATUS      PORTS
S
fad0f8456ca7      postgres:9.6    "docker-entrypoint..."   45 seconds ago   Up 47 seconds   5432/tcp
eless_habit
Nanas-MBP:~ nanabiz$
```

```
$ docker run --name some-postgres -e POSTGRES_PASSWORD=mysecretpassword -d postgres
```

## Docker Image

- ▶ the actual package



Image



configuration PostgreSQL v9.3 Start script

## Docker Container

- ▶ actually **start the application**
- ▶ container environment is created
- ▶ **artifact**, that can be moved around



not running

```
Last login: Sat Sep 28 10:57:20 on ttys001
Nanas-MBP:~ nanabiz$ docker ps
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS          PORTS
S
fad0f8456ca7      postgres:9.6      "docker-entrypoint..."   45 seconds ago   Up 47 seconds   5432/tcp
eless_haibt
Nanas-MBP:~ nanabiz$
```

```
~ — docker run postgres:9.6 ~ — docker run postgres:10.10
Last login: Sat Sep 28 10:57:20 on ttys001
Nanas-MBP:~ nanabiz$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
S
fad0f8456ca7      postgres:9.6       "docker-entrypoint..."   45 seconds ago    Up 47 seconds     5432/tcp
eless_habit
Nanas-MBP:~ nanabiz$ docker run postgres:10.10
Unable to find image 'postgres:10.10' locally
10.10: Pulling from library/postgres
8f91359f1fff: Already exists
c6115f5efcde: Already exists
28a9c19d8188: Already exists
2da4beb7be31: Already exists
fb9ca792da89: Already exists
cedc20991511: Already exists
b866c2f2559e: Already exists
5d459cf6645c: Already exists
6de9d066d892: Downloading [=====>]
401fcdb8e29c4: Download complete
9b130e26214a: Download complete
1c048e77610c: Download complete
431b5e6c27b3: Download complete
4eca80d7c24a: Download complete
some layers already exist ✓
```

```
~ — docker run postgres:9.6 ~ — docker run postgres:10.10 ~ — -bash
Last login: Sat Sep 28 11:08:17 on ttys003
Nanas-MBP:~ nanabiz$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
NAMES
8633ea431a47      postgres:10.10      "docker-entrypoint..."   18 seconds ago    Up 20 seconds     5432/tcp
ecstatic_stallman
fad0f8456ca7      postgres:9.6       "docker-entrypoint..."   5 minutes ago     Up 5 minutes      5432/tcp
priceless_habit
Nanas-MBP:~ nanabiz$
```

## Docker Course

i

# Docker vs. Virtual Machine



SUBSCRIBE

## Overview



## Docker vs Virtual Machine



- Docker on OS level
- Different levels of abstractions
- Why linux-based docker containers don't run on Windows



REPLY

Kernel communicates with hardware

Applications **2. Layer**



OS Kernel **1. Layer**

Hardware

**Operating Systems  
have 2 Layers**

Applications run on Kernel layer

Applications **2. Layer**

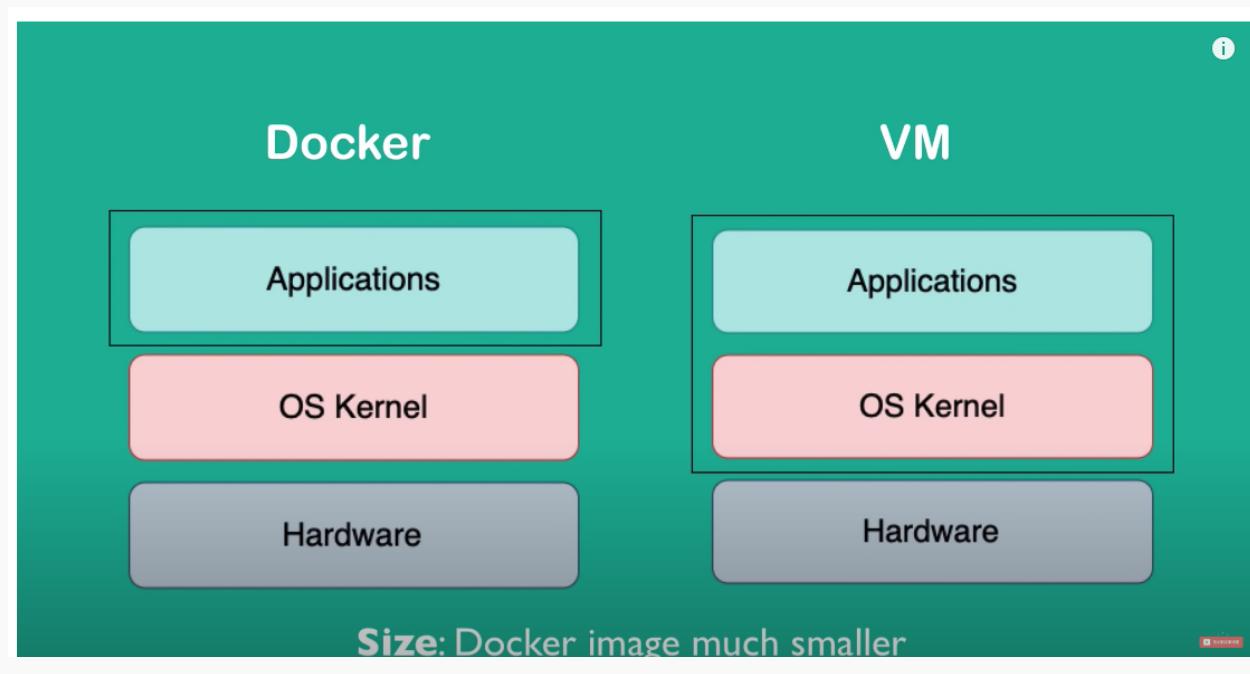
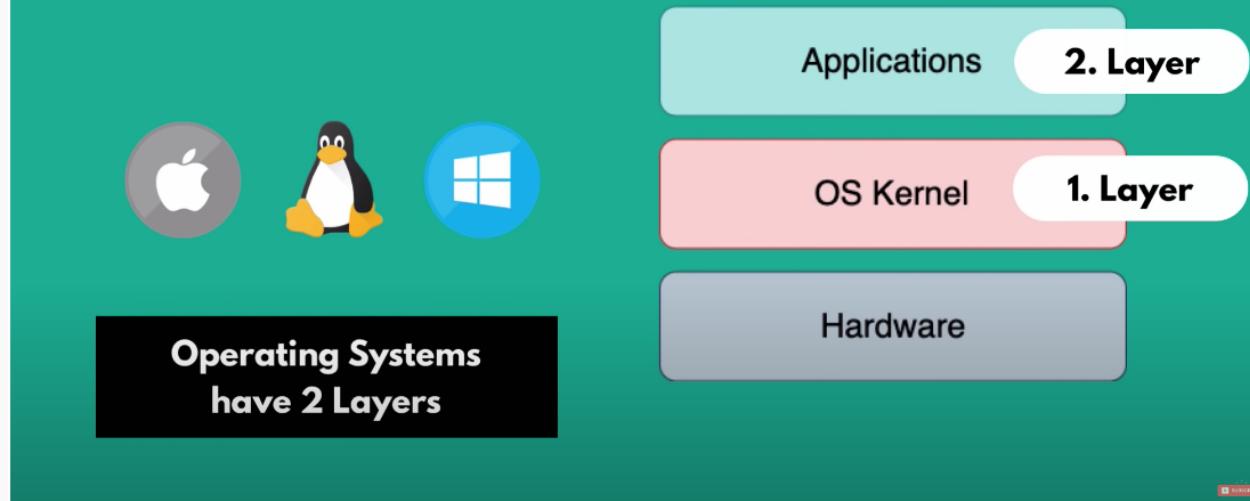


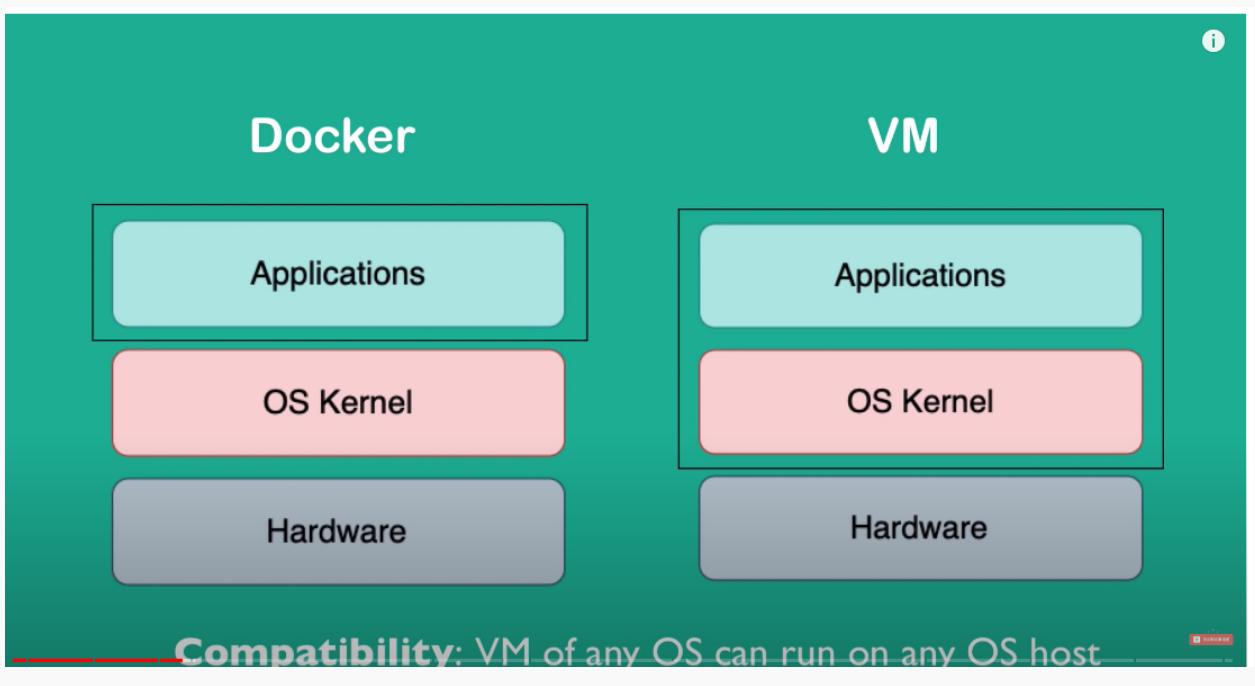
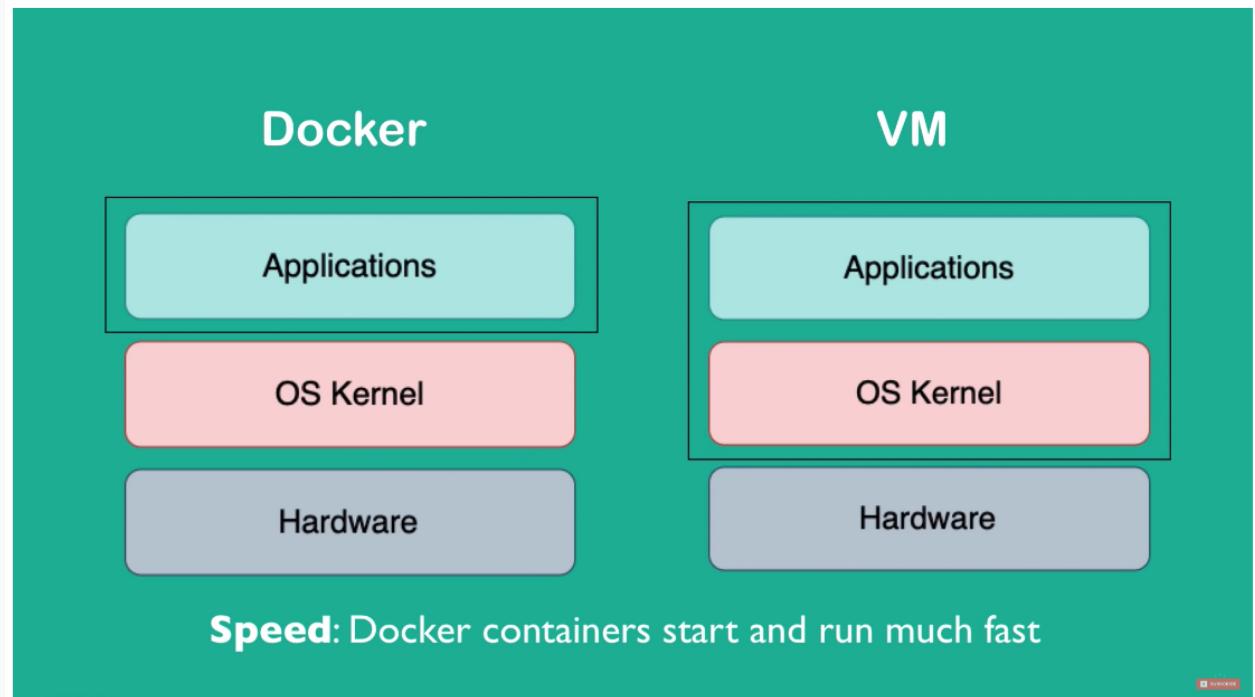
OS Kernel **1. Layer**

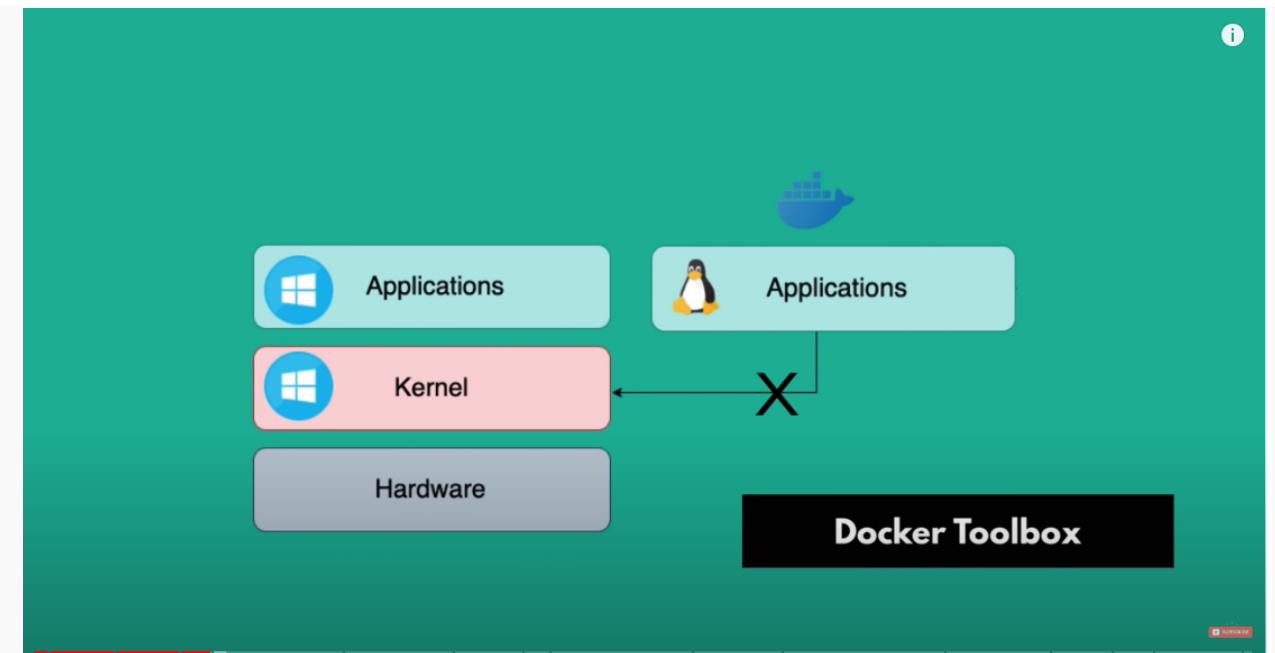
Hardware

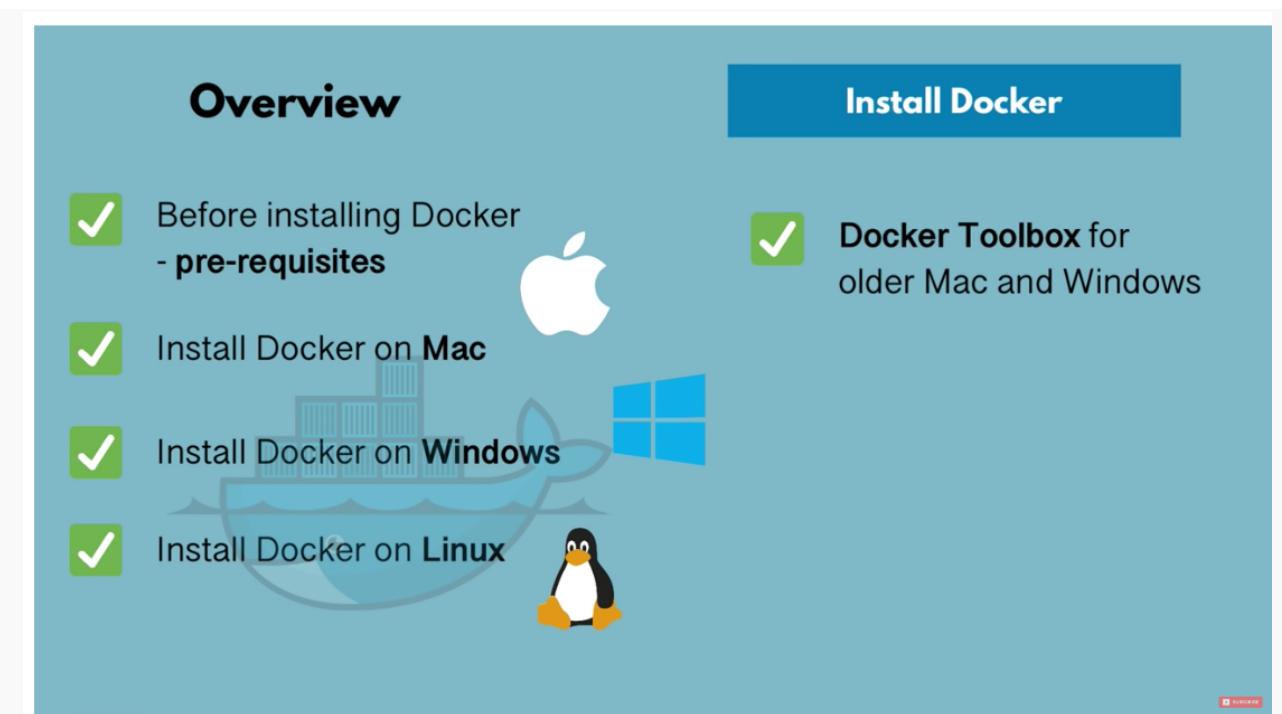
**Operating Systems  
have 2 Layers**

They use same Linux Kernel, but implemented different application on top









The screenshot shows the Docker documentation page for "Install Docker". The top navigation bar includes links for "Guides", "Product manuals", "Glossary", "Reference", and "Samples". The main content area has a sidebar titled "Get Docker" with sections like "Install Docker", "Docker EE", "Docker CE", "Platforms supporting Docker EE and Docker CE", "Optional Linux post-installation steps", "Docker Edge", "Docker for IBM Cloud (Beta)", "Docker for AWS", "Docker for Azure", "Docker Toolbox (legacy)", "Compatibility between Docker versions", "Release notes", "Set started", and "Develop with Docker". The main content area starts with a heading "Install Docker" and a note about estimated reading time (8 minutes). It explains that Docker is available in two editions: Community Edition (CE) and Enterprise Edition (EE). The Community Edition (CE) is ideal for developers and small teams, while the Enterprise Edition (EE) is designed for enterprise development and IT teams. It mentions that Docker CE has two update channels: "stable" and "edge". Below this, there's a table comparing capabilities across different editions:

Capabilities	Community Edition	Enterprise Edition Basic	Enterprise Edition Standard	Enterprise Edition Advanced
Container engine and built in orchestration, networking, security	✓	✓	✓	✓
Certified infrastructure, plugins and ISV containers	✓	✓	✓	✓
Image management	✓	✓	✓	✓



## Get Docker

[Install Docker](#)[Docker EE](#)[Docker CE](#)[Mac](#)[Windows](#)[Ubuntu](#)[Debian](#)[CentOS](#)[Fedora](#)[Binaries](#)[Platforms supporting Docker EE and Docker CE](#)[Optional Linux post-installation steps](#)[Docker Edge](#)[Docker for IBM Cloud \(Beta\)](#)[Docker for AWS](#)

In

Estin

Doc

Doc

exp

•

For

Doc

criti

Ente

Ca

Co

on

Ce

co

Im

**Before Installation - Pre-Requisites**

**Docker Course**

i

# **Basic Commands**

