

Docker Tutorial for Beginners

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

The image shows a screenshot of a YouTube video player. The video has a teal header bar with the title 'Docker Tutorial for Beginners' and a play button icon. Below the header is a light gray sidebar containing a list of topics with blue right-pointing arrows:

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

On the right side of the video frame, there is a large, stylized illustration of a blue whale carrying several shipping containers on its back, swimming in the water. Below the whale, the text 'After that,' is displayed in a white box. In the bottom right corner of the video frame, there is a small red 'SUBSCRIBE' button.

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

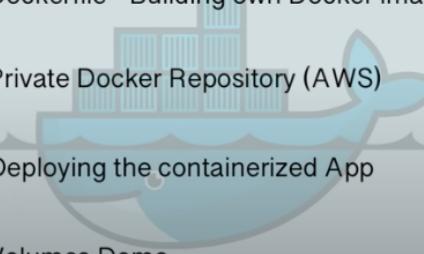
containerized application.
Last but not least,



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



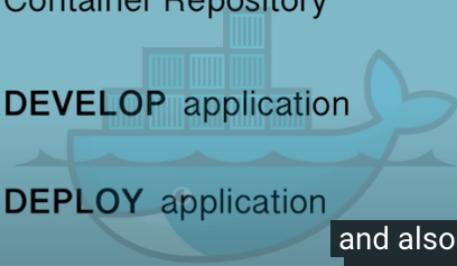
What is Docker?



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

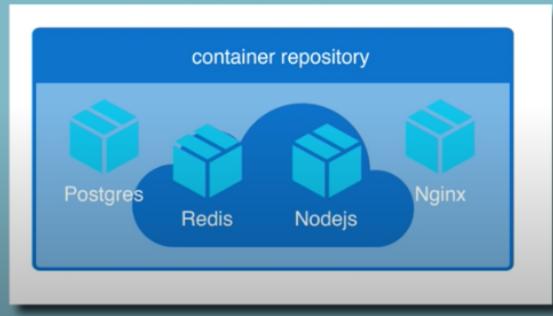


- ▶ Makes development and deployment **more efficient**

Where do containers live?

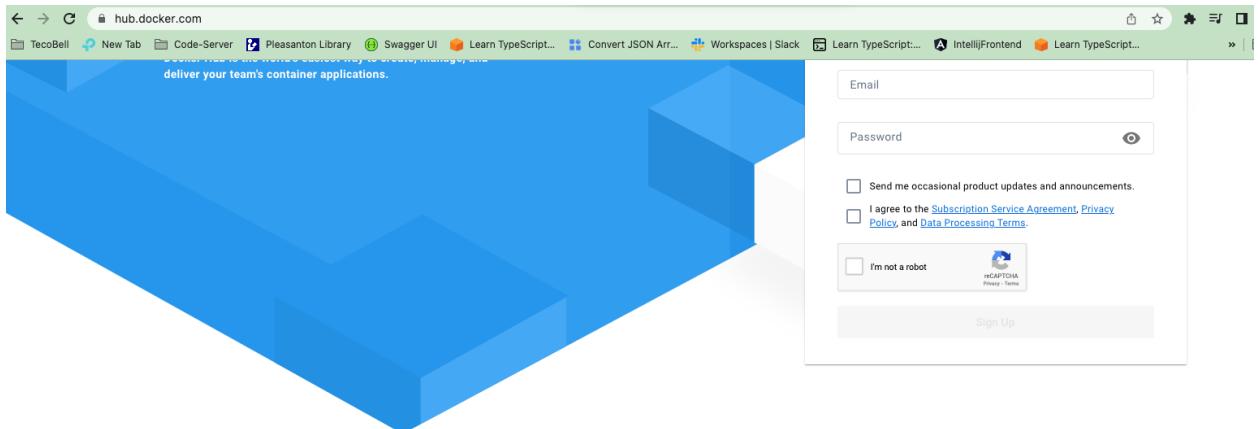
- ▶ Container Repository
- ▶ Private repositories
- ▶ Public repository for Docker

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 presentation slide with a teal background. At the top, it says "How containers improved..". In the center is a white rounded rectangle containing the text "Application Development". At the bottom, there is a video player interface showing a progress bar at 4:59 / 2:46:14, a title "What is Docker? >", and various control icons like play, volume, and settings.



Install required software on local development environment



PostgreSQL
v9.3



Mac



Developer



Redis
v5.0



Linux

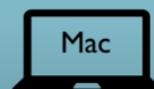
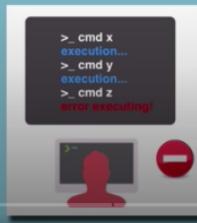


Developer

Application Development

Before containers

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



Mac



Developer



Linux



Developer

6:18 / 2:46:14 • What is Docker? ▶

Application Development

After containers



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

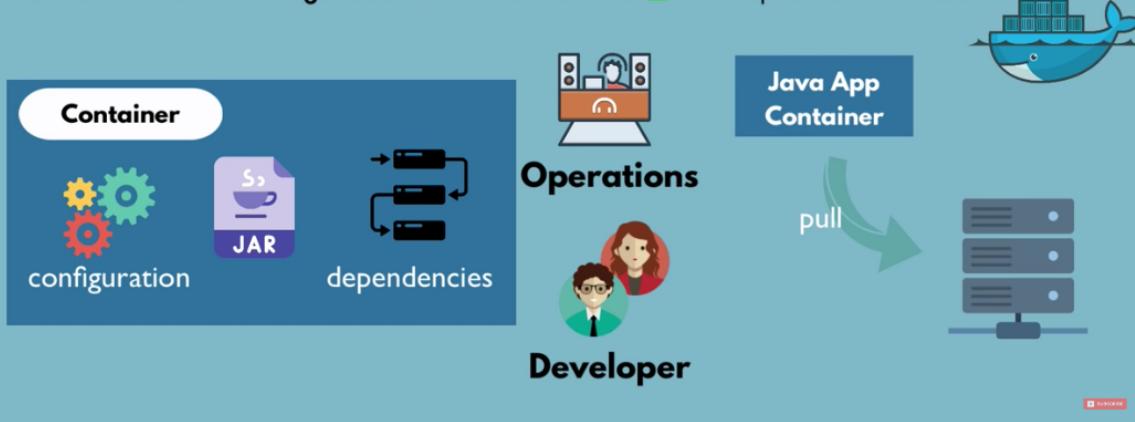


▶

Application Deployment

After containers

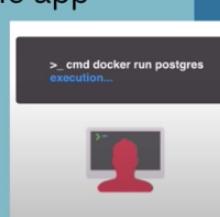
- ▶ Developers and Operations work together to package the application in a container
- ▶ No environmental configuration needed on server  - except Docker Runtime



Application Development

After containers

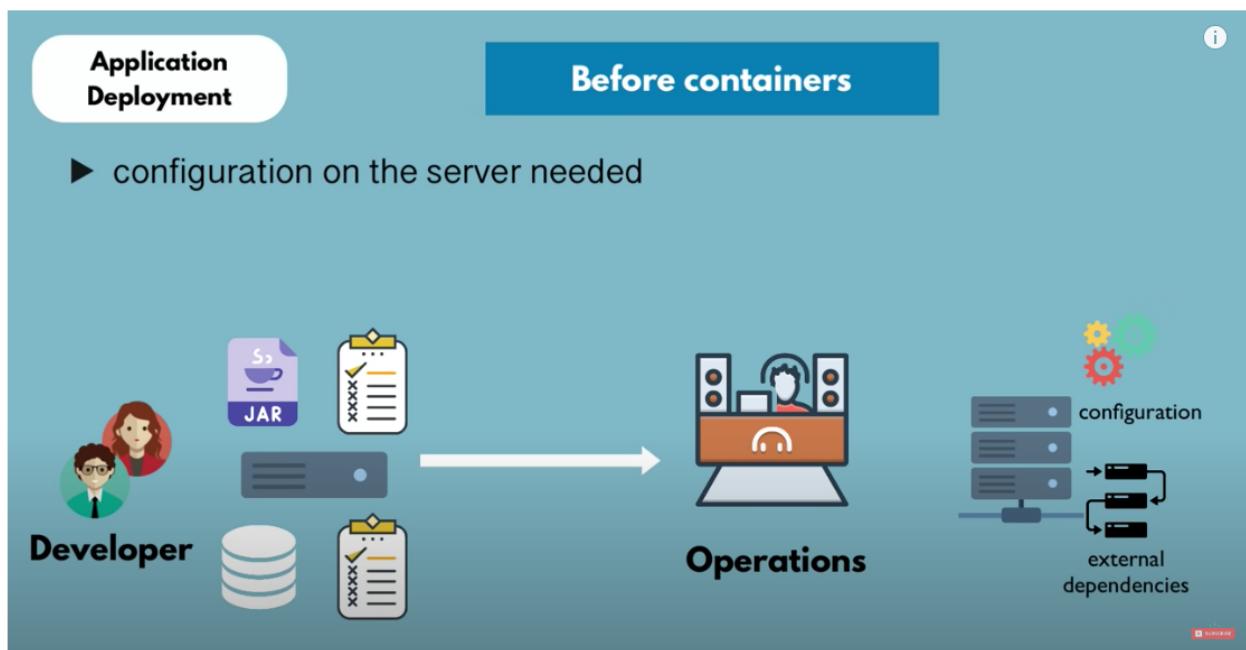
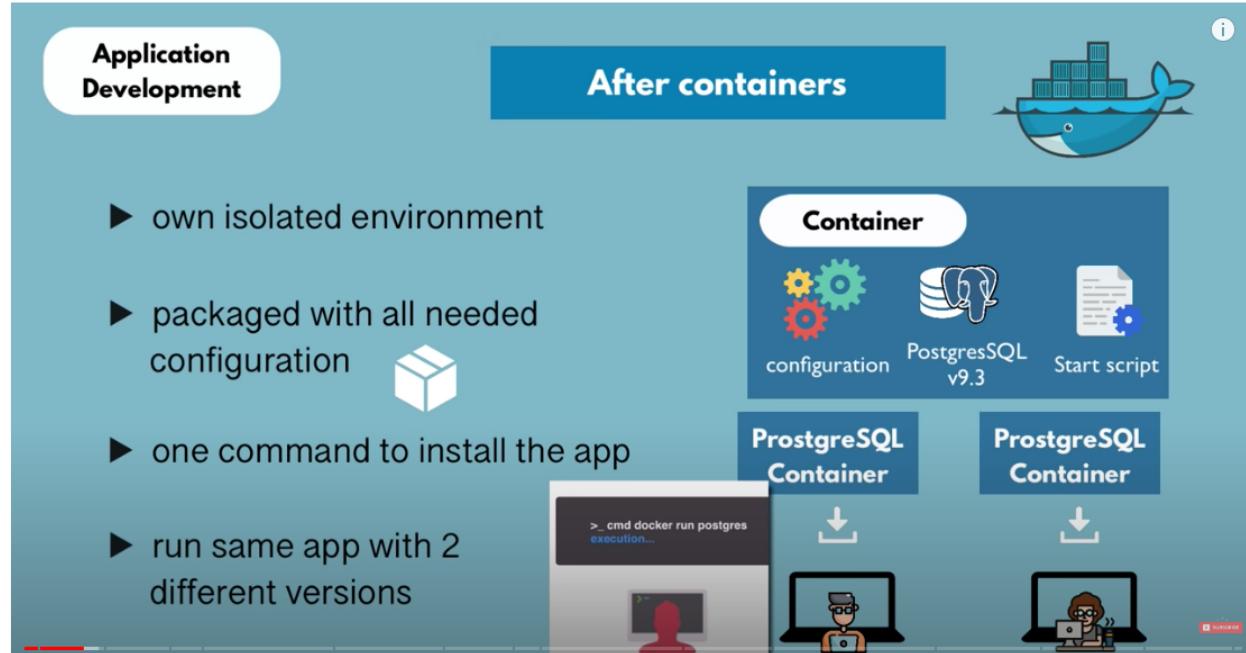
- ▶ own isolated environment
- ▶ packaged with all needed configuration 
- ▶ one command to install the app



PostgreSQL Container

PostgreSQL Container

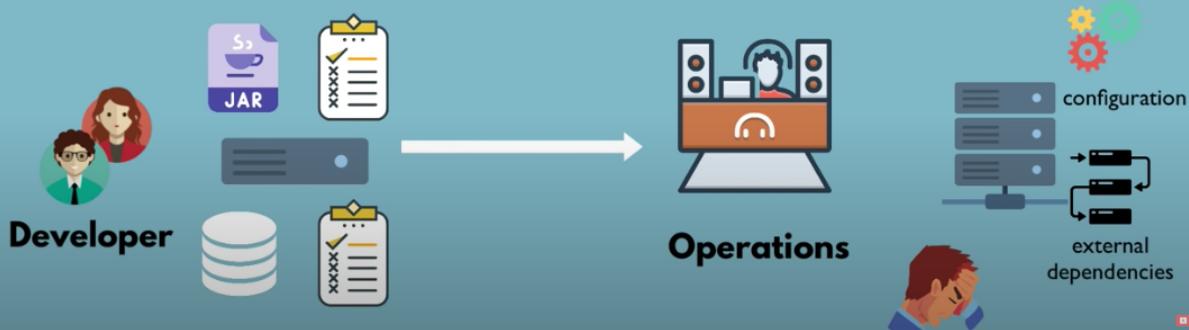
Pause (k)



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



postgres:10.10

Layer - application image

alpine:3.10

Layer - linux base image

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 note at the top right says 'Please log in to write a review of this product.'

Supported tags and respective Dockerfile links

- 12-rcl, 12
- 12-rcl-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

The screenshot shows a terminal window on a Mac OS X system (Nanas-MBP) displaying the output of the 'docker ps' command. The table lists various Docker containers, their IDs, images, commands, creation times, statuses, ports, and names. Most containers are in a 'Waiting' state.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8f91359f1fff	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres
c6115f5efcde	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_1
28a9c19d8188	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_2
2da4beb7be31	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_3
fb9ca792da89	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_4
cedc20991511	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_5
b866c2f2559e	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_6
5d459cf6645c	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_7
b59ec97820c9	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_8
01e040230c2f	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_9
d618b32512c9	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_10
d694ad4e08d5	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_11
f1fc54212826	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_12
2debab4418fd	postgres:9.6	postgres:9.6	2 days ago	Up 2 days	0.0.0.0:5432->5432	postgres_13

```
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
```

```
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

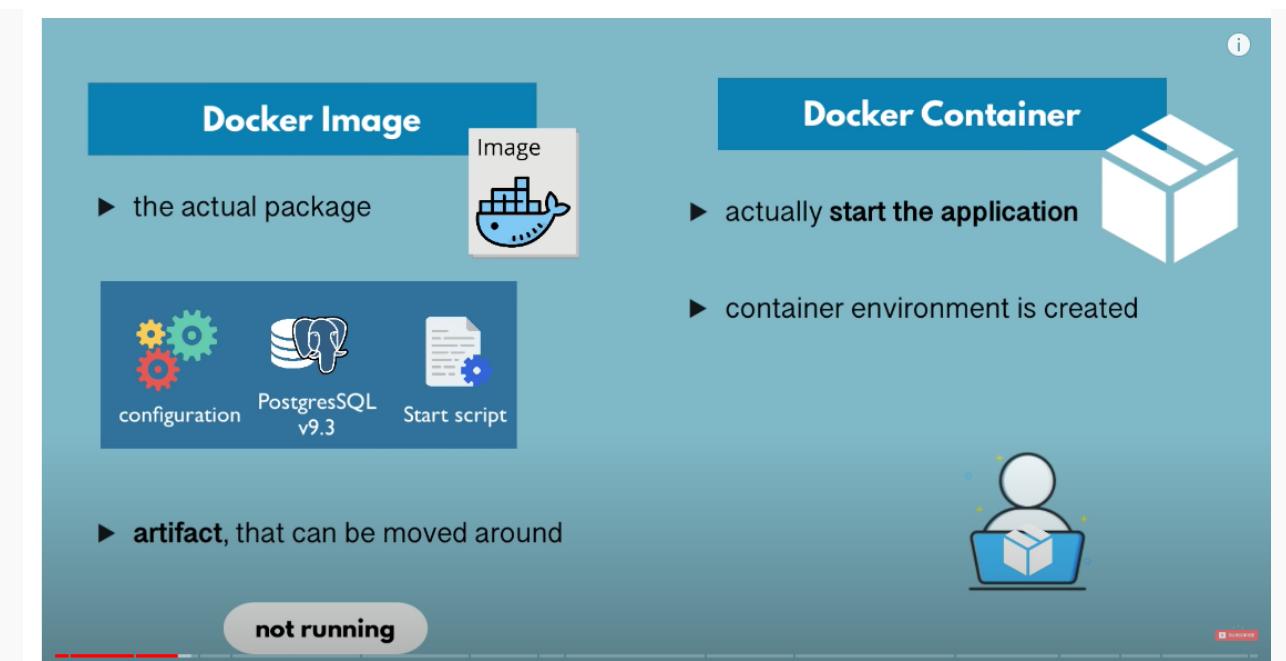
PostgreSQL init process complete; ready for start up.

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
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 run postgres:9.6
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:9.6
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:10.10
Unable to find image 'postgres:10.10' locally
10.10: Pulling from library/postgres
8f91359f1fff: Already exists
c611575efcde: Already exists
28a9c19d8188: Already exists
2da4beb7be31: Already exists
fb9ca792da89: Already exists
cedc20991511: Already exists
b866c2f2559e: Already exists
5d459cf6645c: Already exists
6de9d066d892: Downloading [=====]
401fcfd8e29c4: Download complete
9b130e26214a: Download complete
1c048e77610c: Download complete
431b5e6c27b3: Download complete
4eca80d7c24a: Download complete
some layers already exist ✓
```

```
-- docker run postgres:9.6
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_haibt
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



▶ SUBSCRIBE

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



**Operating Systems
have 2 Layers**

Applications

2. Layer

OS Kernel

1. Layer

Hardware

Docker

Applications

OS Kernel

Hardware

VM

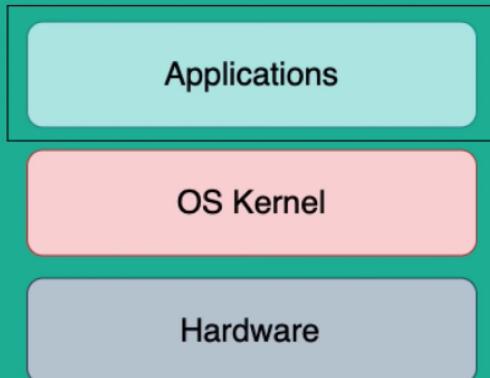
Applications

OS Kernel

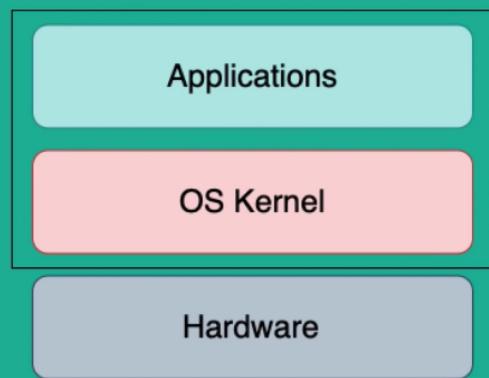
Hardware

Size: Docker image much smaller

Docker

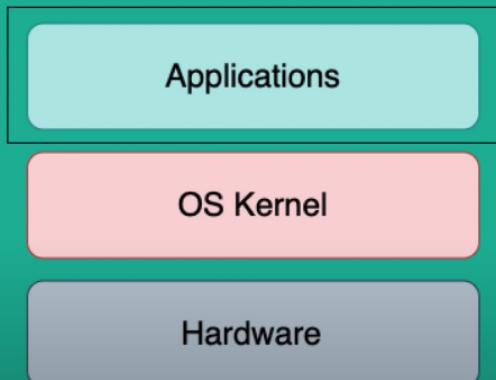


VM

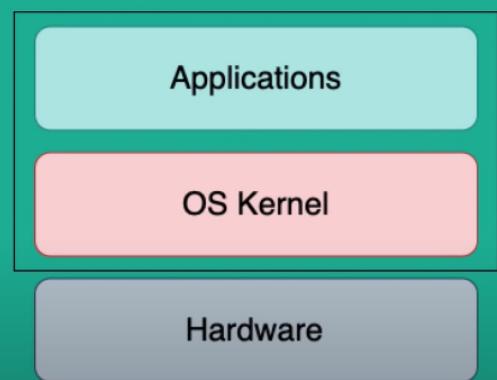


Speed: Docker containers start and run much faster

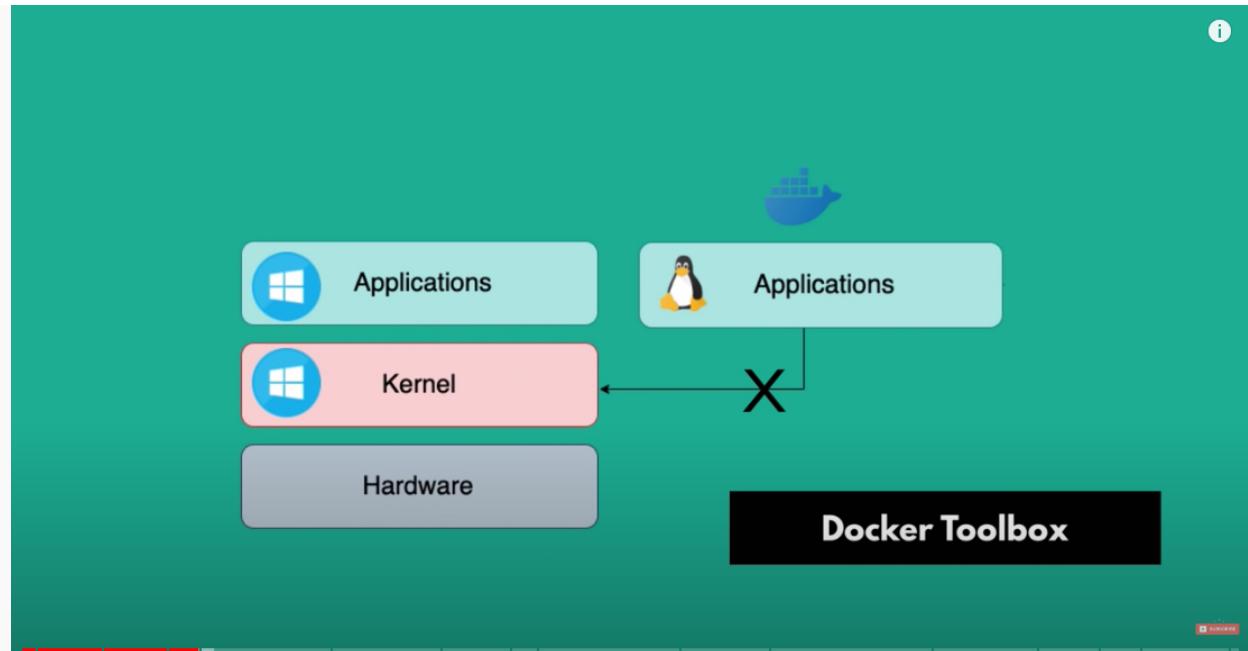
Docker



VM



Compatibility: VM of any OS can run on any OS host



Overview

Install Docker

- ✓ Before installing Docker - pre-requisites
- ✓ Install Docker on Mac
- ✓ Install Docker on Windows
- ✓ Install Docker on Linux

Docker Toolbox for older Mac and Windows

[docker docs](#)

Guides Product manuals Glossary Reference Samples

Get Docker

- 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
- Develop with Docker

Install Docker

Estimated reading time: 8 minutes

Docker is available in two editions: [Community Edition \(CE\)](#) and [Enterprise Edition \(EE\)](#).

Docker Community Edition (CE) is ideal for developers and small teams looking to get started with Docker and experimenting with container-based apps. Docker CE has two update channels, [stable](#) and [edge](#):

- **Stable** gives you reliable updates every quarter
- **Edge** gives you new features every month

For more information about Docker CE, see [Docker Community Edition](#).

Docker Enterprise Edition (EE) is designed for enterprise development and IT teams who build, ship, and run business critical applications in production at scale. For more information about Docker EE, including purchasing options, see [Docker Enterprise Edition](#).

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

<https://docs.docker.com/v17.09/engine/installation/linux/ubuntu/>

Before Installation - Pre-Requisites

Docker Course

i

Basic Commands

