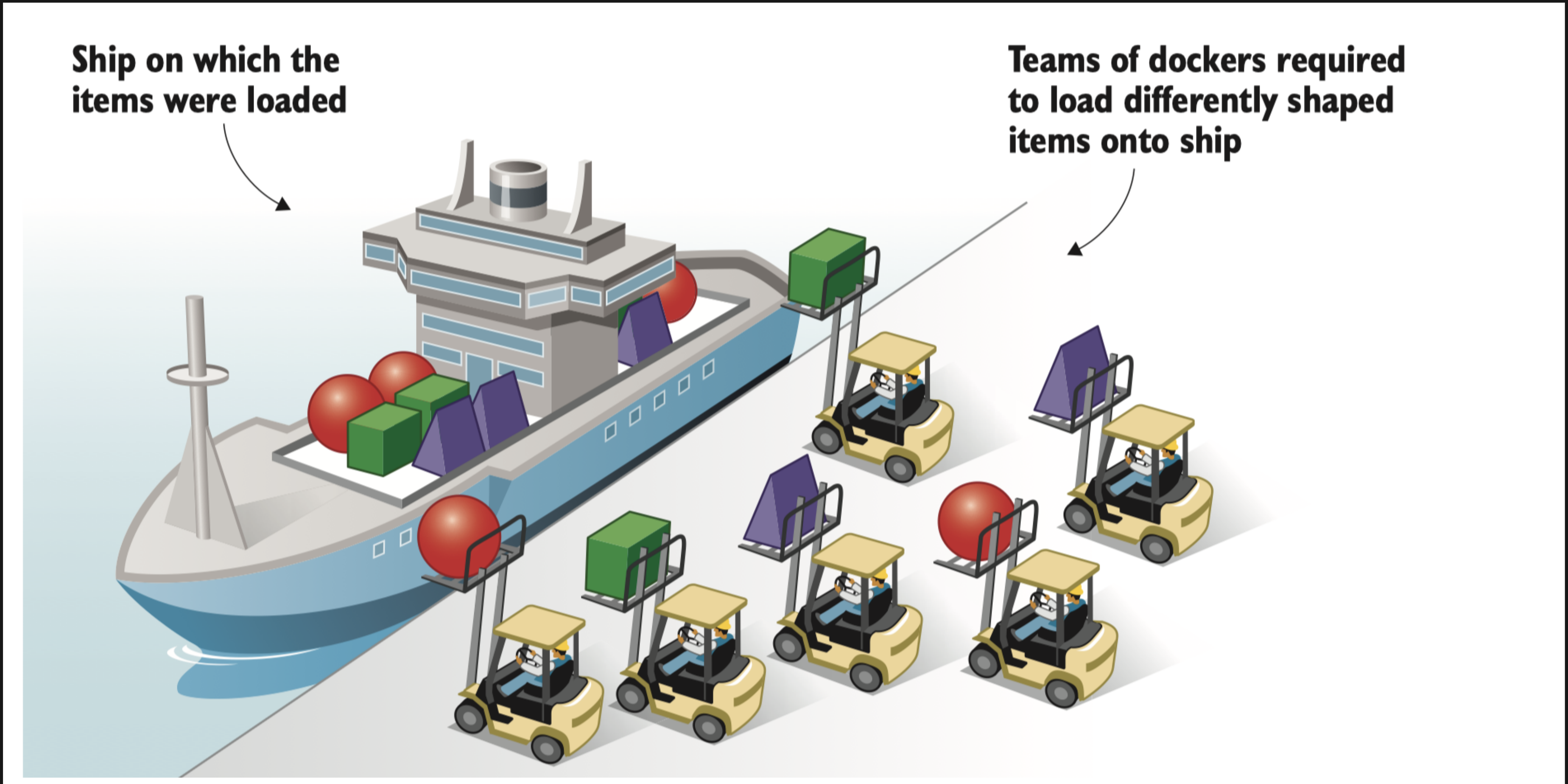
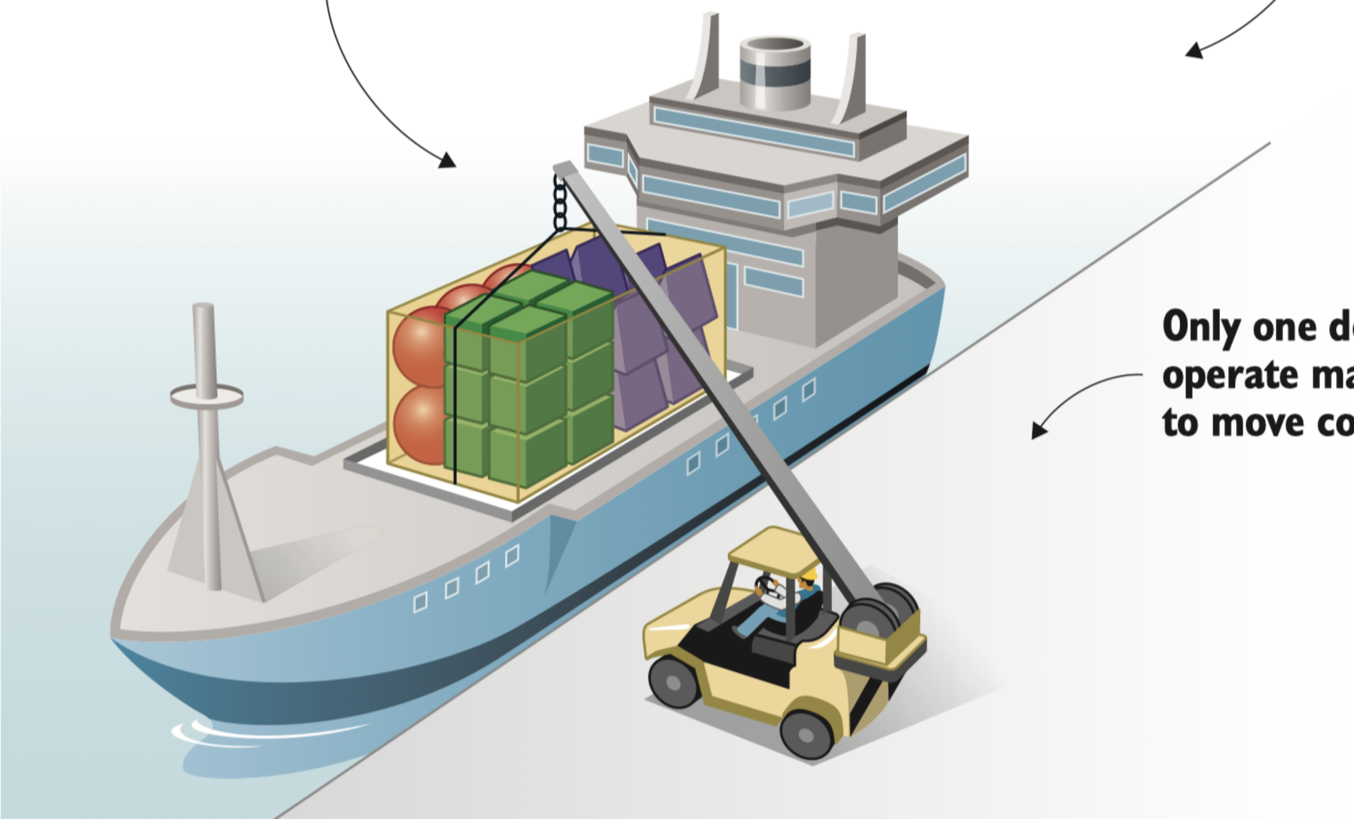
DOCKER

Sep 20, 2019

WHAT IS DOCKER?



Ship on which the items were loaded
Teams of dockers required to load differently shaped items onto ship



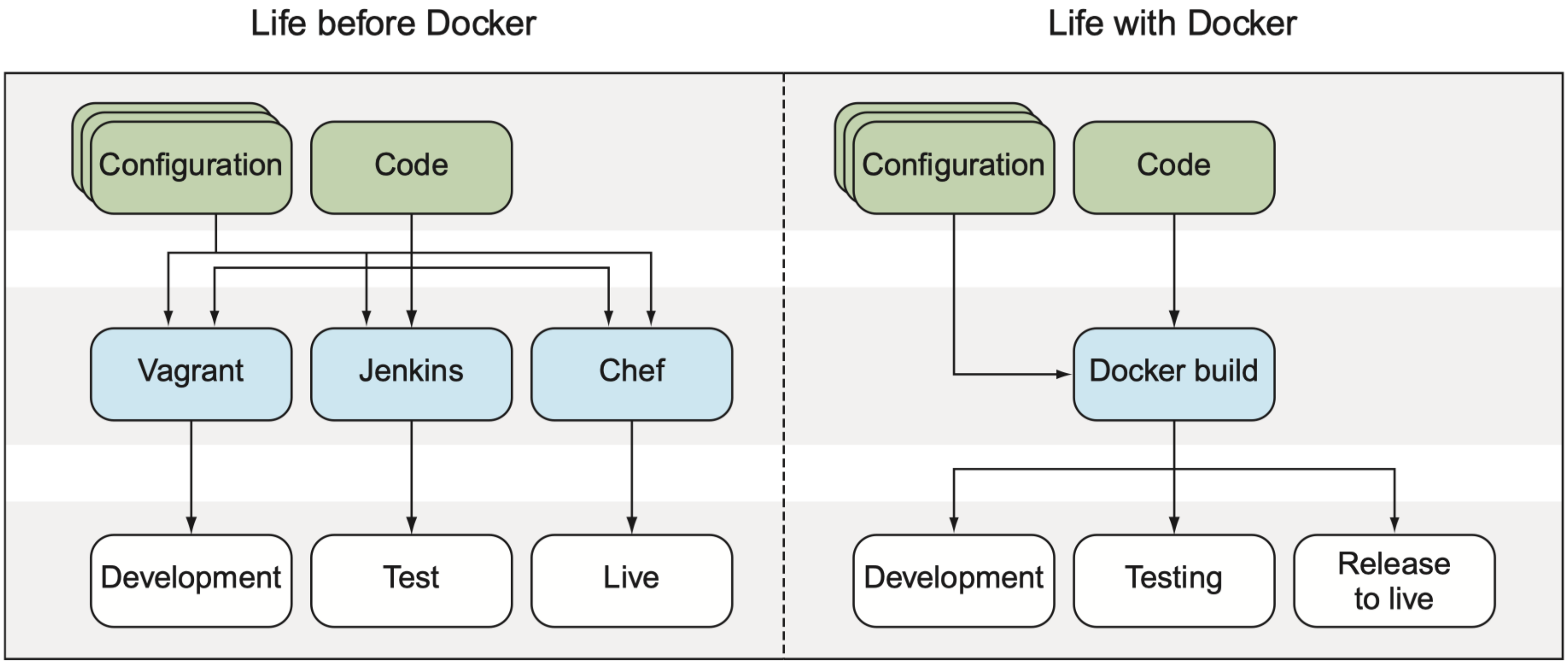
Single container with different items in it. It doesn't matter to the carrier what's inside the container. The carrier can be loaded up elsewhere, reducing the bottleneck of loading at port.

Ship can be designed to carry, load, and unload predictably shaped items more efficiently.

Only one docker needed to operate machines designed to move containers.

WHAT IS DOCKER GOOD FOR?

EASE TOOL MAINTENANCE BURDEN

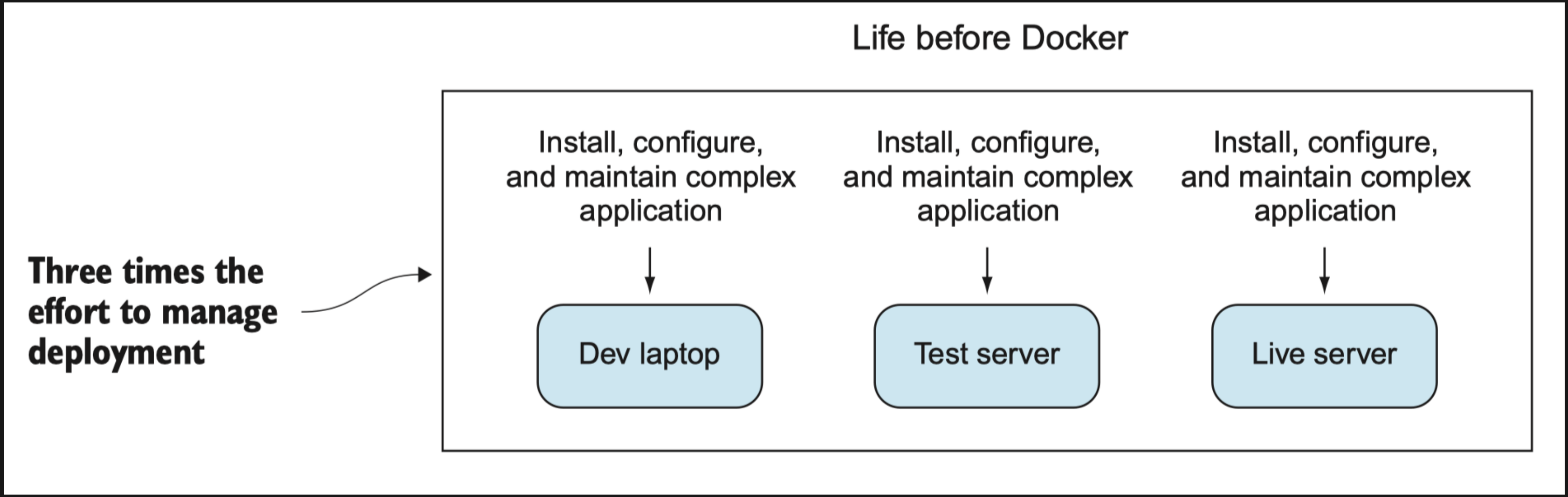


🟩 Inputs to the system requiring manual maintenance—fewer inputs here mean less of a maintenance burden.

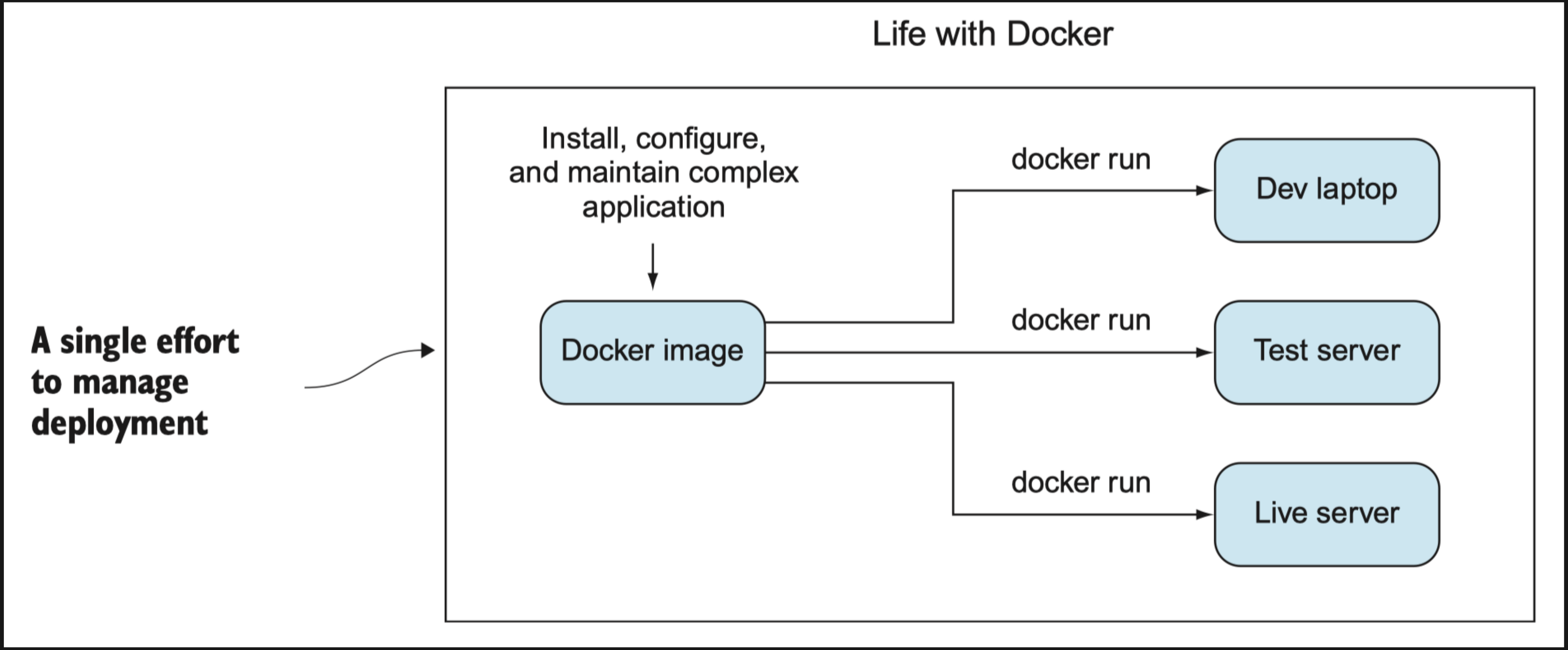
🟦 Tools that use the inputs to create environments for software development

⬜ Stages of software development requiring an environment to run in

EASE SOFTWARE DELIVERY


Life before Docker
Three times the effort to manage deployment
Install, configure, and maintain complex application
Install, configure, and maintain complex application
Install, configure, and maintain complex application
Dev laptop
Test server
Live server

EASE SOFTWARE DELIVERY



- Replacing Virtual Machines (lightweight, faster)

- Replacing Virtual Machines (lightweight, faster)
- Prototyping software (e.g., python dependencies)

- Replacing Virtual Machines (lightweight, faster)
- Prototyping software (e.g., python dependencies)
- Packaging software

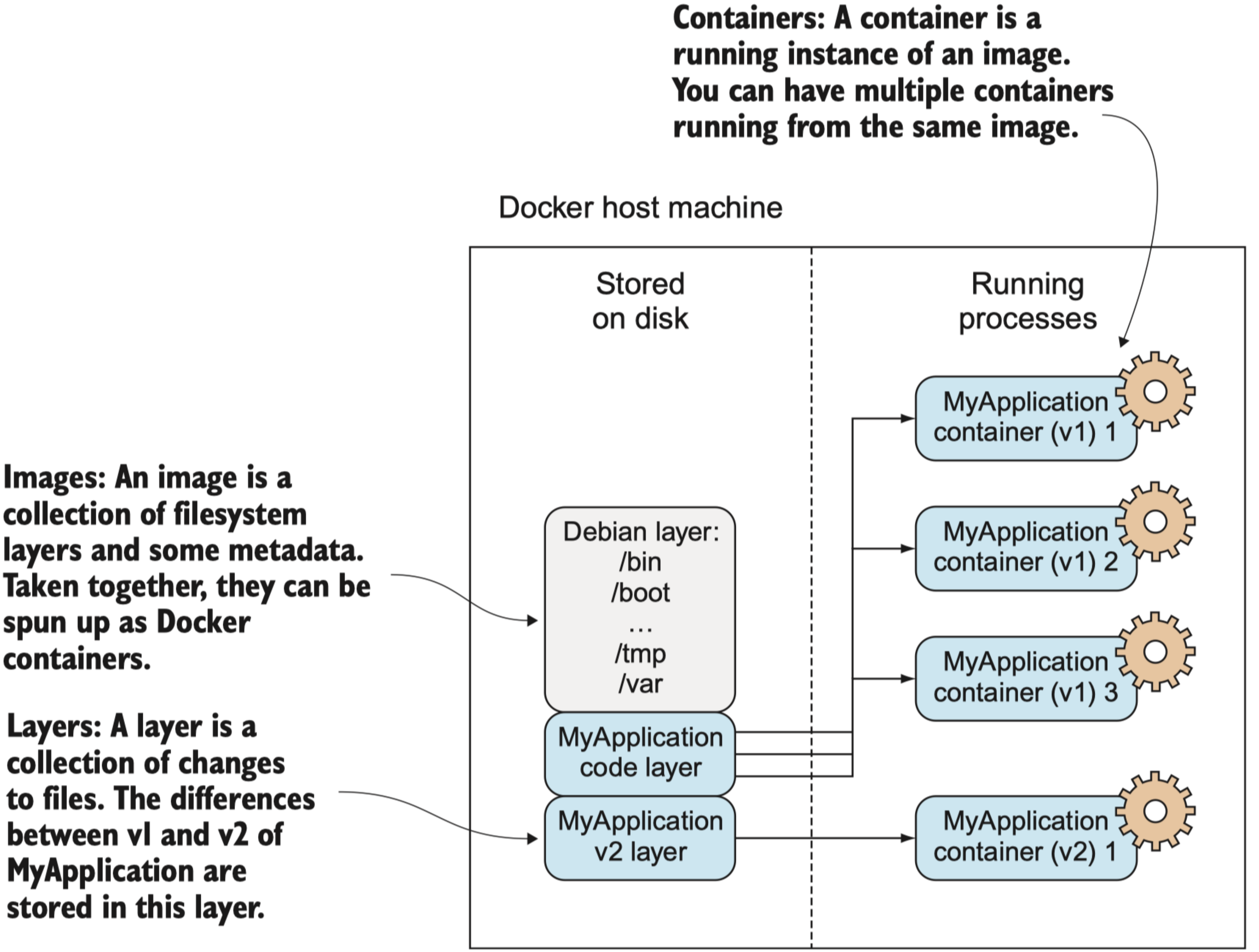
- Replacing Virtual Machines (lightweight, faster)
- Prototyping software (e.g., python dependencies)
- Packaging software
- Microservices architecture

- Replacing Virtual Machines (lightweight, faster)
- Prototyping software (e.g., python dependencies)
- Packaging software
- Microservices architecture
- Modeling network

- Replacing Virtual Machines (lightweight, faster)
- Prototyping software (e.g., python dependencies)
- Packaging software
- Microservices architecture
- Modeling network
- ...

CORE CONCEPTS

Image, Container, Layer



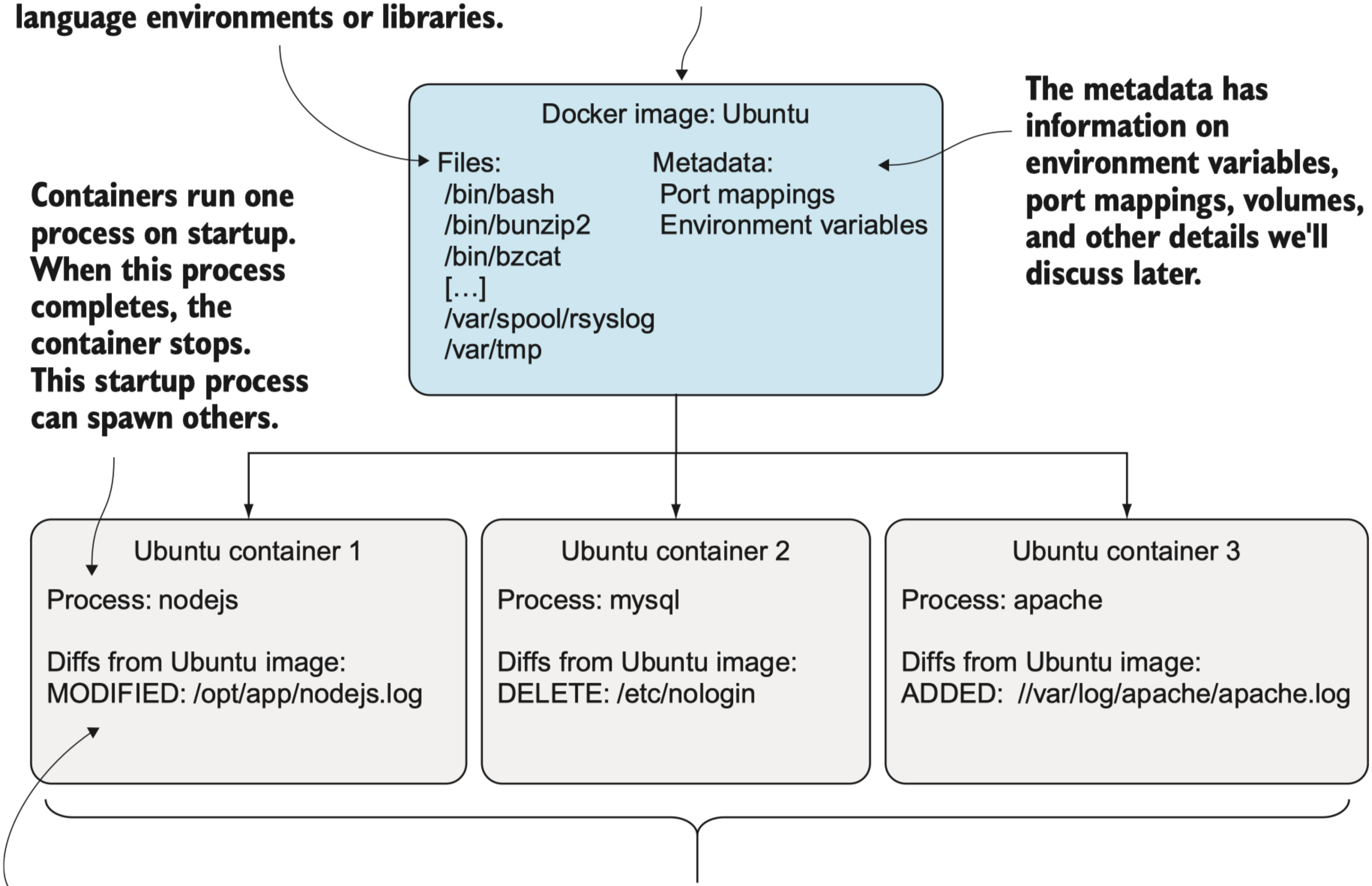
Containers: A container is a running instance of an image. You can have multiple containers running from the same image.

Images: An image is a collection of filesystem layers and some metadata. Taken together, they can be spun up as Docker containers.

Layers: A layer is a collection of changes to files. The differences between vl and v2 of MyApplication are stored in this layer.

Because of the isolation each container provides, they must have their own copy of any required tools, including language environments or libraries.

of files and metadata. This is the base image for the containers below.

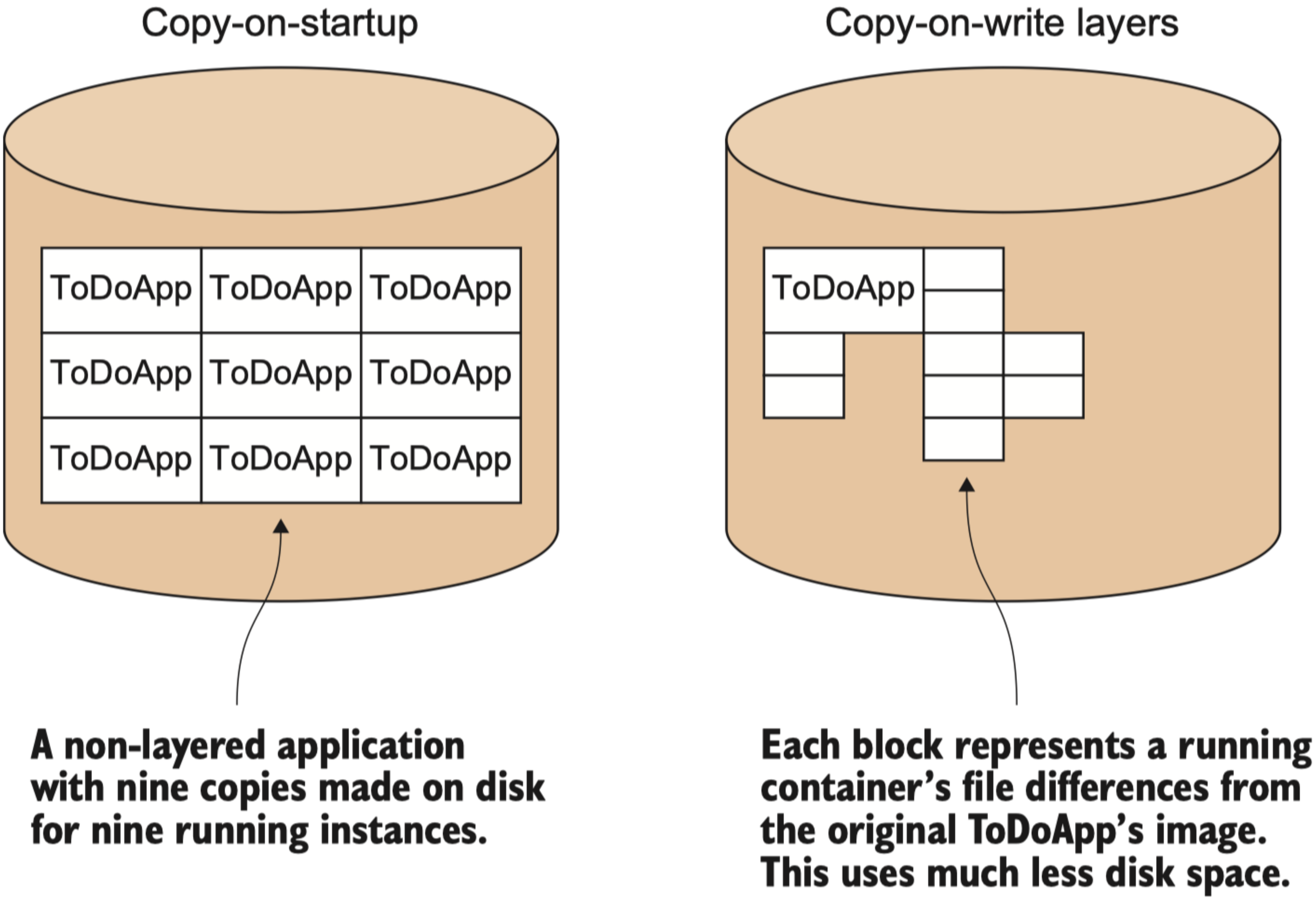


The metadata has information on environment variables, port mappings, volumes, and other details we'll discuss later.

Containers run one process on startup. When this process completes, the container stops. This startup process can spawn others.

Changes to files are stored within the container in a copy-on-write mechanism. The base image cannot be

Containers are created from images, inherit their filesystems, and use their metadata to determine their startup configurations. Containers are separate but can be configured



Copy-on-startup

Copy-on-write layers

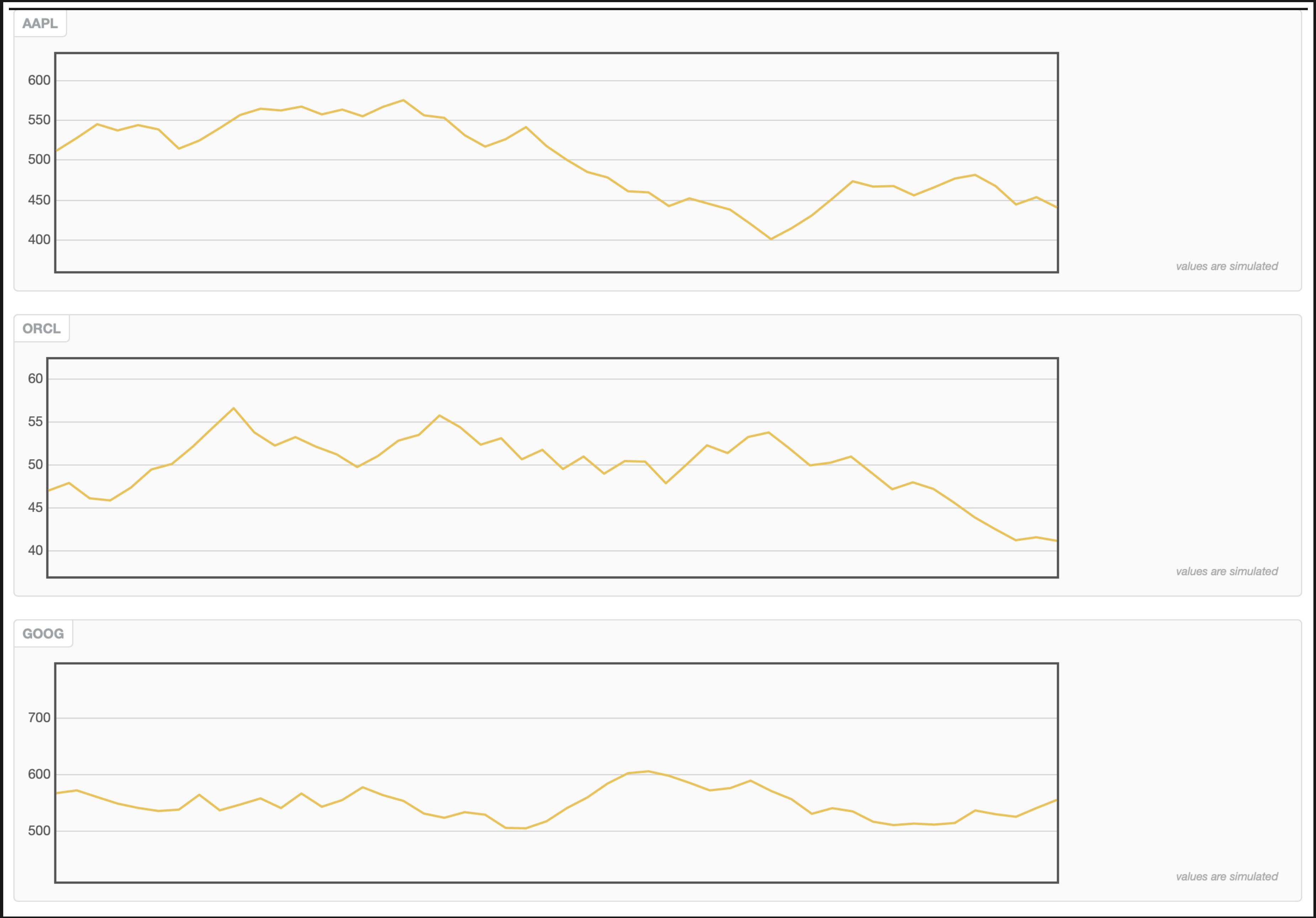
A non-layered application with nine copies made on disk for nine running instances.

Each block represents a running container's file differences from the original ToDoApp's image. This uses much less disk space.

DOCKER BASICS

DOCKER COMMANDS

- `docker pull`: download image from registry
- `docker build`: create a new image from a Dockerfile
- `docker run`: create container from image and run it
- `docker images`: see downloaded or built images
- `docker ps -a`: show running containers


AAPL
600
550
500
450
400
values are simulated
ORCL
60
55
50
45
40
values are simulated
GOOG
700
600
500
values are simulated

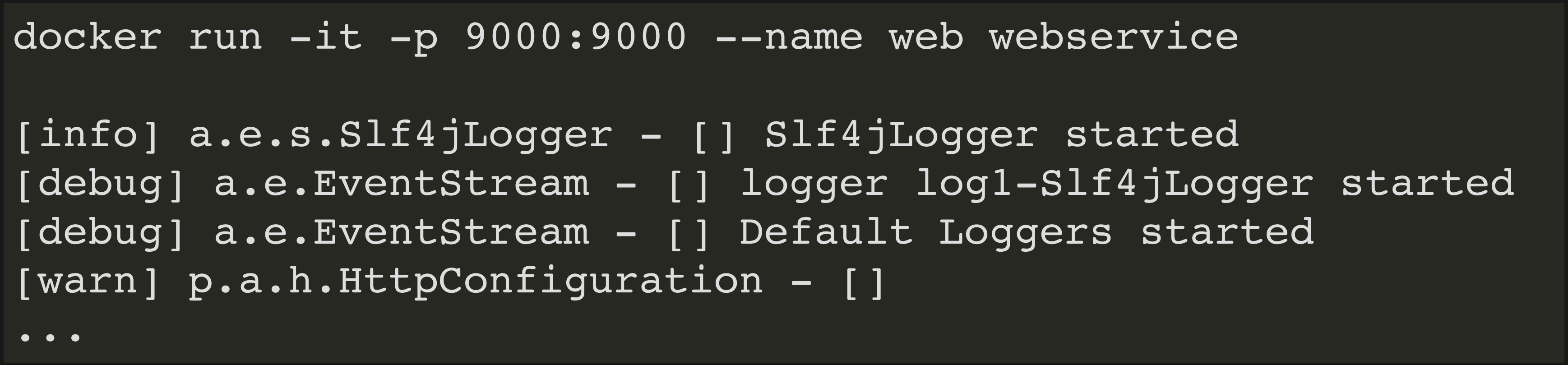
DOCKERFILE

```
FROM openjdk:8
COPY . /usr/src/webservice
WORKDIR /usr/src/webservice/bin
EXPOSE 9000
CMD ./webservice -Dplay.http.secret.key=abcdefghijklmn
```

```
docker build .

Sending build context to Docker daemon  47.65MB
Step 1/5 : FROM openjdk:8
 ---> e8d00769c8a8
Step 2/5 : COPY . /usr/src/webservice
 ---> ae33607c09a1
Step 3/5 : WORKDIR /usr/src/webservice/bin
 ---> Running in 878355c7e617
Removing intermediate container 878355c7e617
 ---> 72204bf76082
Step 4/5 : EXPOSE 9000
 ---> Running in b0f120f05528
Removing intermediate container b0f120f05528
 ---> 283c9ba584d7
```

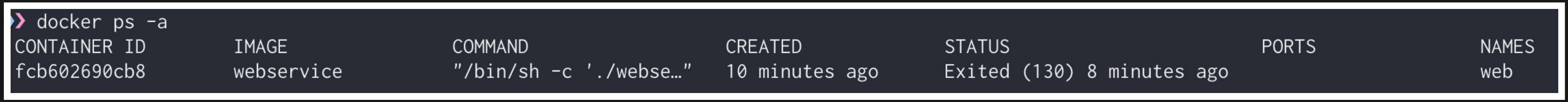
```
docker tag 6eb69f9d75b6 webservice
```



```
docker run -it -p 9000:9000 --name web webservice

[info] a.e.s.Slf4jLogger - [] Slf4jLogger started
[debug] a.e.EventStream - [] logger log1-Slf4jLogger started
[debug] a.e.EventStream - [] Default Loggers started
[warn] p.a.h.HttpConfiguration - []
...
```

```
docker ps -a
```

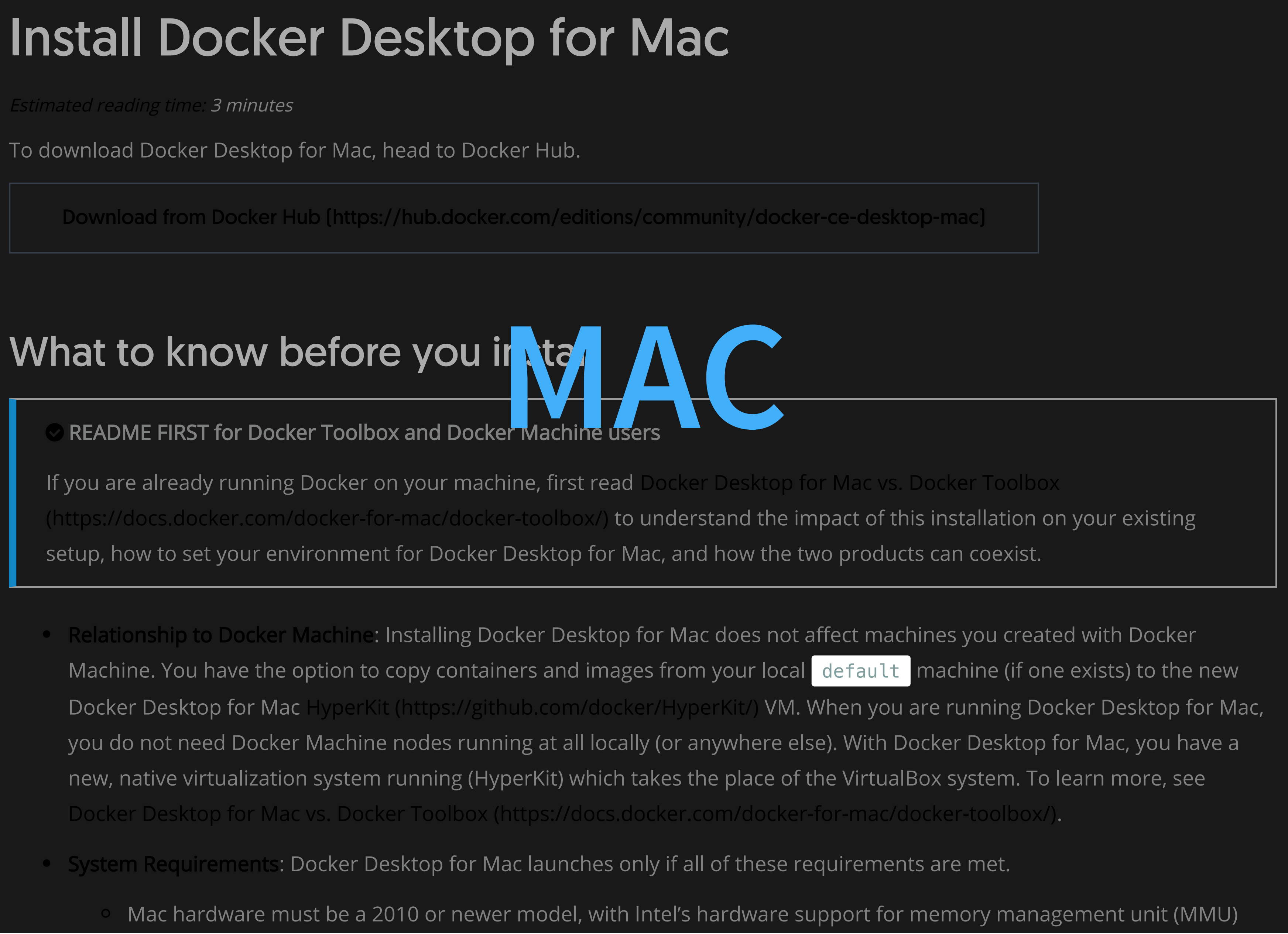


```
❯ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                        PORTS               NAMES
fcb602690cb8        webservice          "/bin/sh -c './webse…"   10 minutes ago      Exited (130) 8 minutes ago                       web
```

```
docker rm -f web
web
```

INSTALL

Docker Community Edition



MAC

Install Docker Desktop for Mac

Estimated reading time: 3 minutes

To download Docker Desktop for Mac, head to Docker Hub.

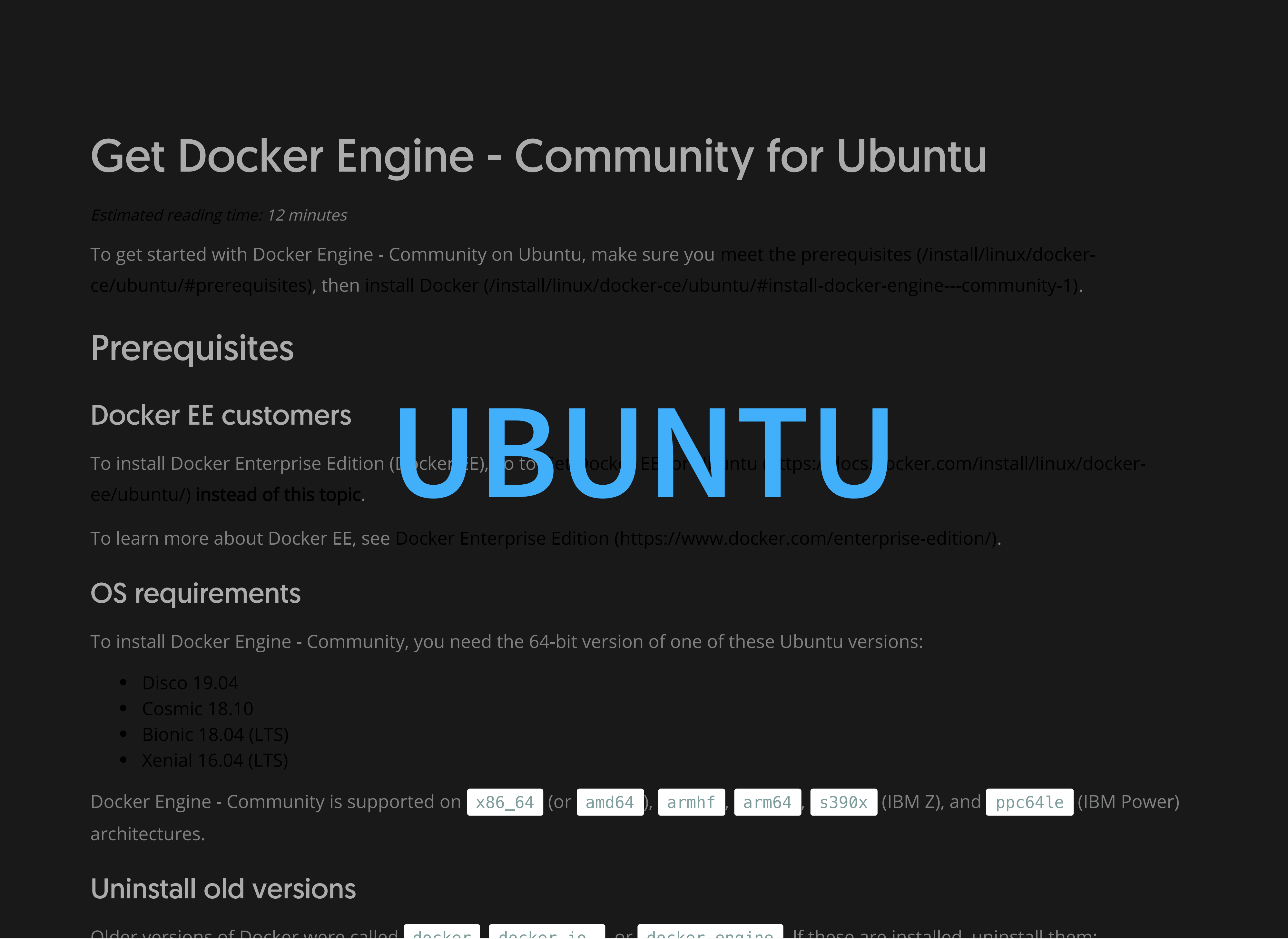
Download from Docker Hub (https://hub.docker.com/editions/community/docker-ce-desktop-mac)

What to know before you install

✓ README FIRST for Docker Toolbox and Docker Machine users

If you are already running Docker on your machine, first read Docker Desktop for Mac vs. Docker Toolbox (https://docs.docker.com/docker-for-mac/docker-toolbox/) to understand the impact of this installation on your existing setup, how to set your environment for Docker Desktop for Mac, and how the two products can coexist.

- **Relationship to Docker Machine**: Installing Docker Desktop for Mac does not affect machines you created with Docker Machine. You have the option to copy containers and images from your local `default` machine (if one exists) to the new Docker Desktop for Mac HyperKit (https://github.com/docker/HyperKit/) VM. When you are running Docker Desktop for Mac, you do not need Docker Machine nodes running at all locally (or anywhere else). With Docker Desktop for Mac, you have a new, native virtualization system running (HyperKit) which takes the place of the VirtualBox system. To learn more, see Docker Desktop for Mac vs. Docker Toolbox (https://docs.docker.com/docker-for-mac/docker-toolbox/).
- **System Requirements**: Docker Desktop for Mac launches only if all of these requirements are met.
    - Mac hardware must be a 2010 or newer model, with Intel's hardware support for memory management unit (MMU)



UBUNTU

Get Docker Engine - Community for Ubuntu

Estimated reading time: 12 minutes

To get started with Docker Engine - Community on Ubuntu, make sure you meet the prerequisites (/install/linux/docker-ce/ubuntu/#prerequisites), then install Docker (/install/linux/docker-ce/ubuntu/#install-docker-engine---community-1).

Prerequisites

Docker EE customers

To install Docker Enterprise Edition (Docker EE), go to Get Docker EE for Ubuntu (https://docs.docker.com/install/linux/docker-ee/ubuntu/) instead of this topic.

To learn more about Docker EE, see Docker Enterprise Edition (https://www.docker.com/enterprise-edition/).

OS requirements

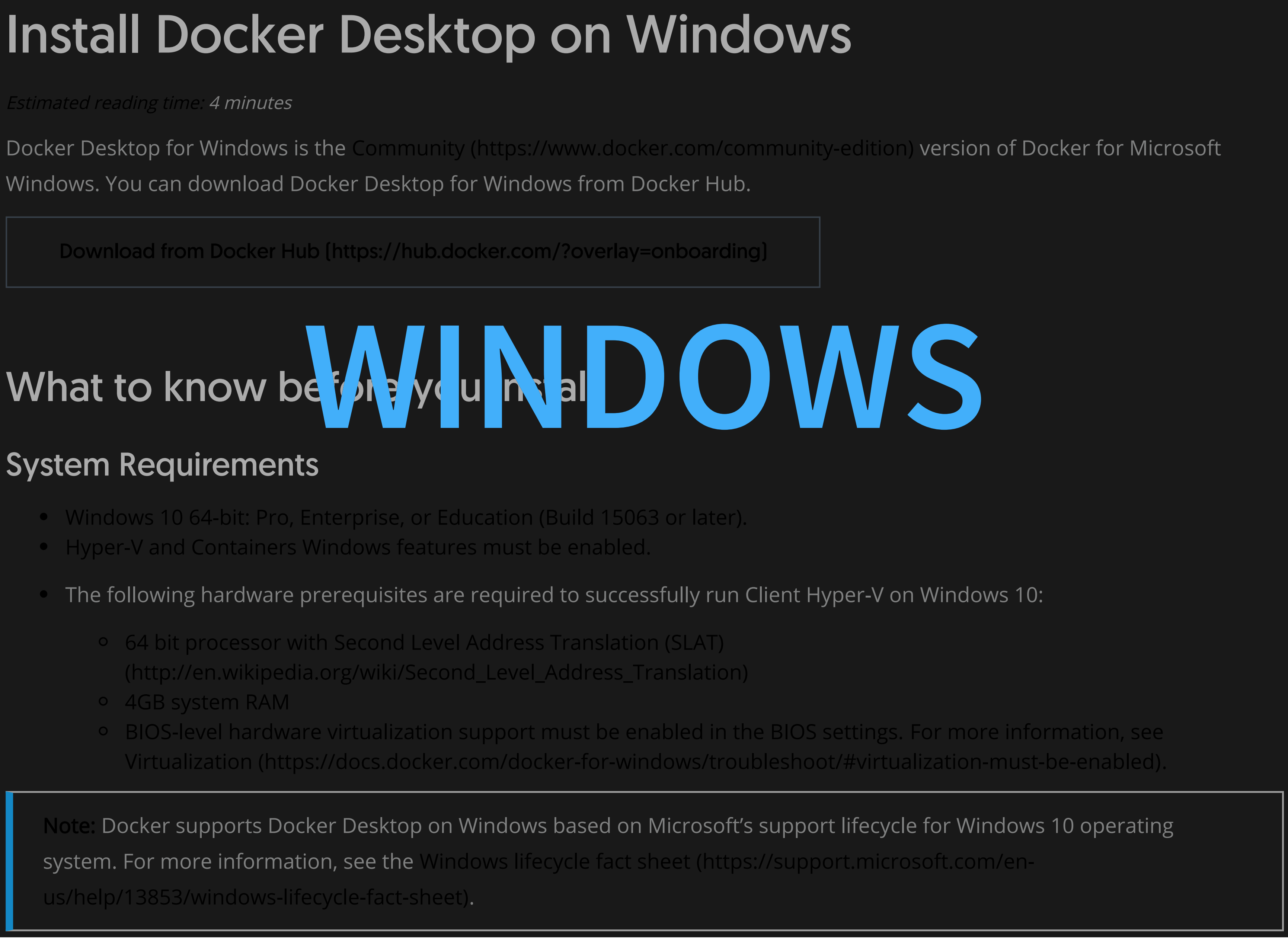
To install Docker Engine - Community, you need the 64-bit version of one of these Ubuntu versions:

- Disco 19.04
- Cosmic 18.10
- Bionic 18.04 (LTS)
- Xenial 16.04 (LTS)

Docker Engine - Community is supported on `x86_64` (or `amd64`), `armhf`, `arm64`, `s390x` (IBM Z), and `ppc64le` (IBM Power) architectures.

Uninstall old versions

Older versions of Docker were called `docker`, `docker.io`, or `docker-engine`. If these are installed, uninstall them:



Install Docker Desktop on Windows

Estimated reading time: 4 minutes

Docker Desktop for Windows is the Community (https://www.docker.com/community-edition) version of Docker for Microsoft Windows. You can download Docker Desktop for Windows from Docker Hub.

Download from Docker Hub (https://hub.docker.com/?overlay=onboarding)

What to know before you install

WINDOWS

System Requirements

- Windows 10 64-bit: Pro, Enterprise, or Education (Build 15063 or later).
- Hyper-V and Containers Windows features must be enabled.
- The following hardware prerequisites are required to successfully run Client Hyper-V on Windows 10:
  - 64 bit processor with Second Level Address Translation (SLAT) (http://en.wikipedia.org/wiki/Second_Level_Address_Translation)
  - 4GB system RAM
  - BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see Virtualization (https://docs.docker.com/docker-for-windows/troubleshoot/#virtualization-must-be-enabled).

> **Note:** Docker supports Docker Desktop on Windows based on Microsoft’s support lifecycle for Windows 10 operating system. For more information, see the Windows lifecycle fact sheet (https://support.microsoft.com/en-us/help/13853/windows-lifecycle-fact-sheet).

EXERCISE

- Download a toy linear regression model here
- Create two Docker containers
  - Dockerize the model as a web service
  - Create a client docker that sends a request to the server (e.g., using `curl`)

```
curl -i -X POST -H 'Content-Type:
application/json' -d '{"exp": 3.45}'
http://localhost:5000/api
```

HINTS

Dependencies:

```
FROM ubuntu:latest
RUN apt update && apt install -y python3 python3-pip && pip3 i
ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8
```

The server can be run with:

```
env FLASK_APP=server.py flask run
```

The server listens on port 5000

Use the `--network host` option to `docker run` for both server and client

SERVER

```
FROM ubuntu:latest
RUN apt update && apt install -y python3 python3-pip && pip3 i
ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8
COPY model.pkl /home/
COPY server.py /home/
WORKDIR /home/
EXPOSE 5000
CMD ["env", "FLASK_APP=server.py", "flask", "run"]
```

CLIENT

```
FROM ubuntu:latest
RUN apt update && apt install -y curl
CMD curl -i -X POST -H 'Content-Type: application/json' -d '{"
```

RUN THE SERVER

```
docker run --network host server

 * Serving Flask app "server.py"
 * Environment: production
   WARNING: This is a development server. Do not use it in a p
   Use a production WSGI server instead.
 * Debug mode: off
/usr/local/lib/python3.6/dist-packages/sklearn/base.py:306: Us
  UserWarning)
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [20/Sep/2019 14:00:13] "POST /api HTTP/1.1" 200
```

RUN THE CLIENT

```
docker run --network host client

  % Total    % Received % Xferd  Average Speed   Time    Time
                                 Dload  Upload   Total   Spent
100    31  100    18  100    13   2000   1444 --:--:-- --:--:-
HTTP/1.0 200 OK
Content-Type: application/json
Content-Length: 18
Server: Werkzeug/0.16.0 Python/3.6.8
Date: Fri, 20 Sep 2019 14:00:13 GMT

59059.69367280337
```