Containers

Mustafa Bilal Demirkan

İçerik

Nereden geldik
Container kavramı nasıl ortaya çıktı?

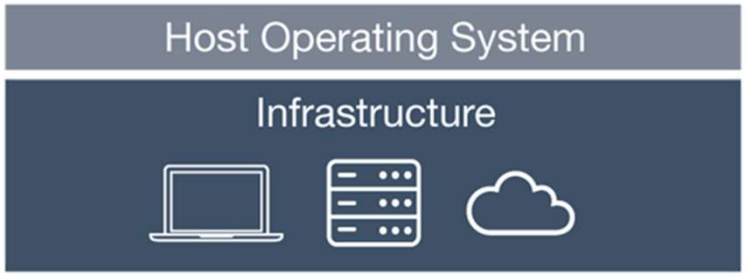
Container bileşenleri
ns, cgroups, SELinux …

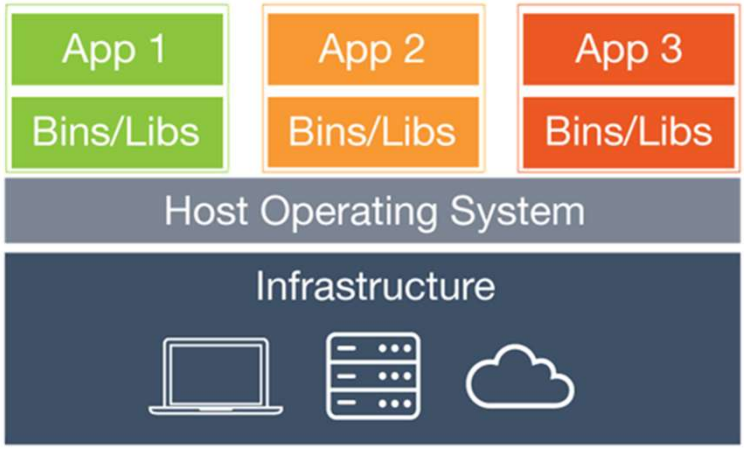
Docker
Docker neden bu kadar popüler?

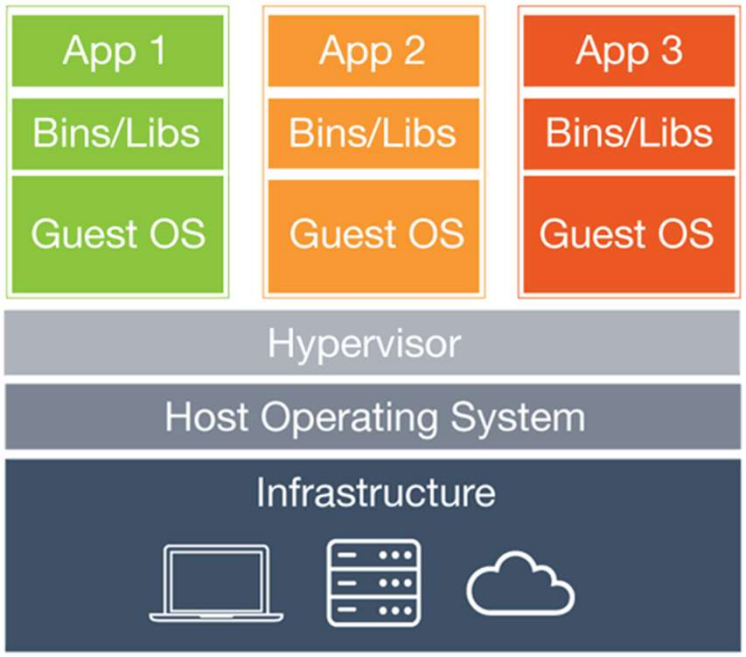
Build
Kendimize ait bir container nasıl yapabiliriz?

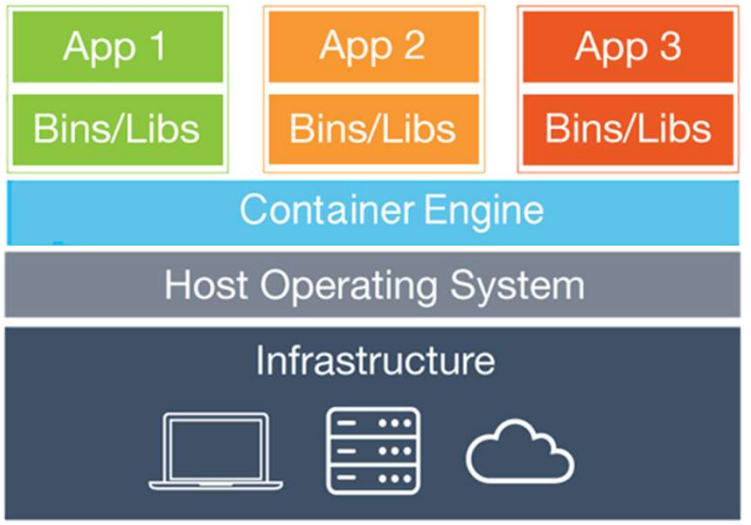
Run
Uygulamalarımızı container içinde çalıştıralım!

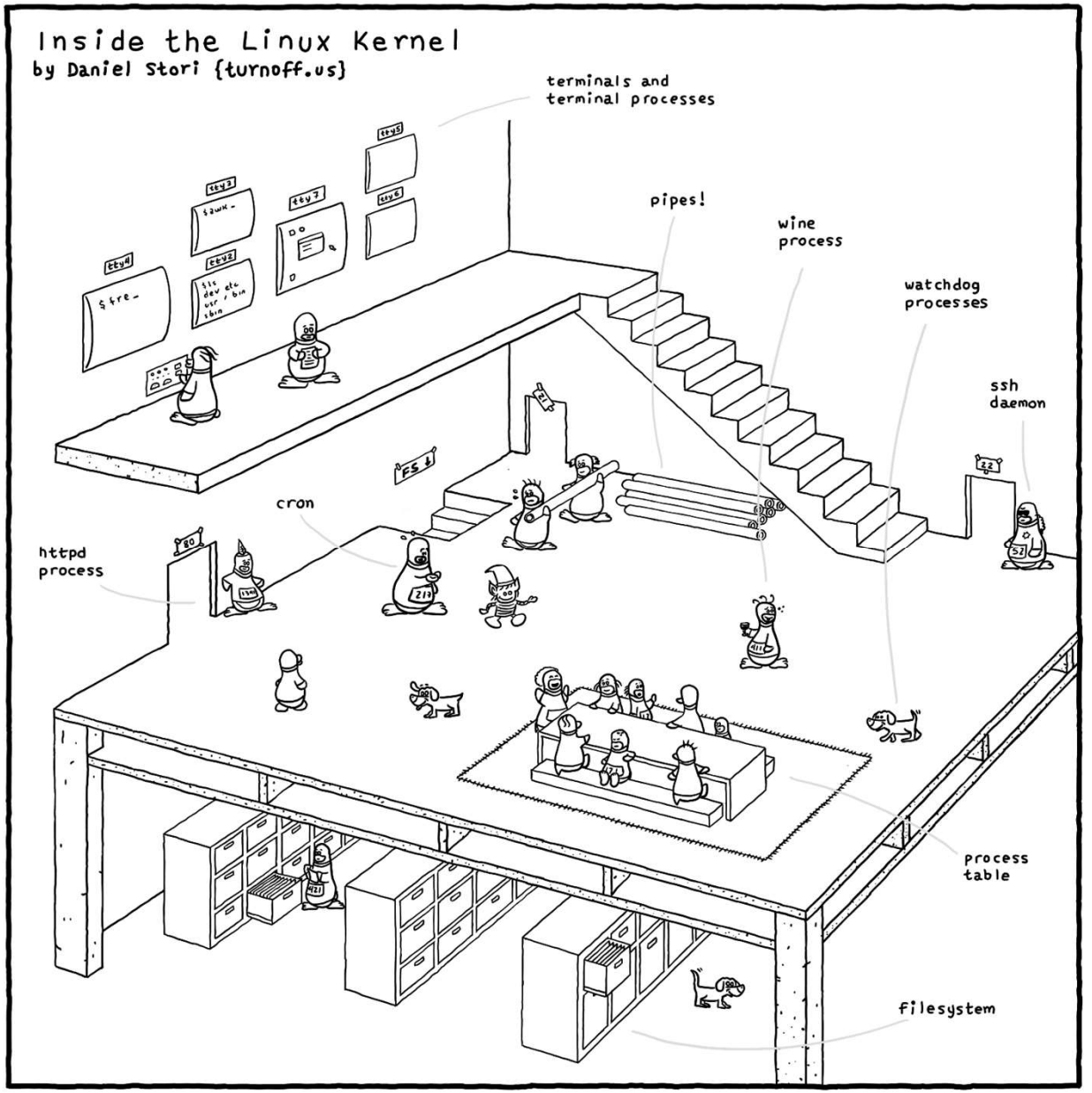
Container


Host Operating System
Infrastructure


App 1
Bins/Libs
App 2
Bins/Libs
App 3
Bins/Libs
Host Operating System
Infrastructure


App 1
Bins/Libs
Guest OS
App 2
Bins/Libs
Guest OS
App 3
Bins/Libs
Guest OS
Hypervisor
Host Operating System
Infrastructure


App 1
Bins/Libs
App 2
Bins/Libs
App 3
Bins/Libs
Container Engine
Host Operating System
Infrastructure


Inside the Linux Kernel
by Daniel Stori {turnoff.us}
terminals and terminal processes
pipes!
wine process
watchdog processes
ssh daemon
cron
httpd process
process table
filesystem

Avantajları

- Düşük donanım ihtiyacı
- Ortam izalasyonu
- Hızlı kurulum
- Çoklu ortam kurulumu
- Tekrar kullanılabilirlik
- Hızlı mikroservis geliştirme

Separation of concerns

the developer inside my container:	**the ops outside the container:**
my code	logging
my libraries	remote access
my package manager	network configuration
my app	monitoring
my data	

Tarihçe

1979

UNIX v7

2000

FreeBSD Jails

2001

Linux VServer

2004

Solaris Zones

2005

Open VZ

Tarihçe

LXC

Google

LMCTFY

2006

2008

2011

2013

2013

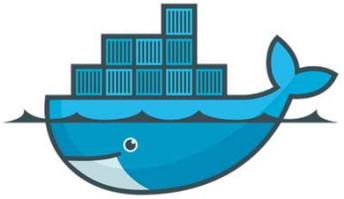
Process Containers

Google

Warden

CLOUD FOUNDRY

Docker



En iyisi Docker mı?

- Hepsi aynı kernel fonksiyonlarını kullanıyor
- Performansları aynı

Ayırt edici olarak bakılacaklar:

- Tasarım
- Ekosistem

LAB - chroot

Namespaces

Namespace	Isolates
PID	Process IDs
IPC	System V IPC, POSIX message queues
Network	Network devices, stacks, ports, etc.
Mount	Mount points
User	User and group IDs
UTS	Hostname and NIS domain name

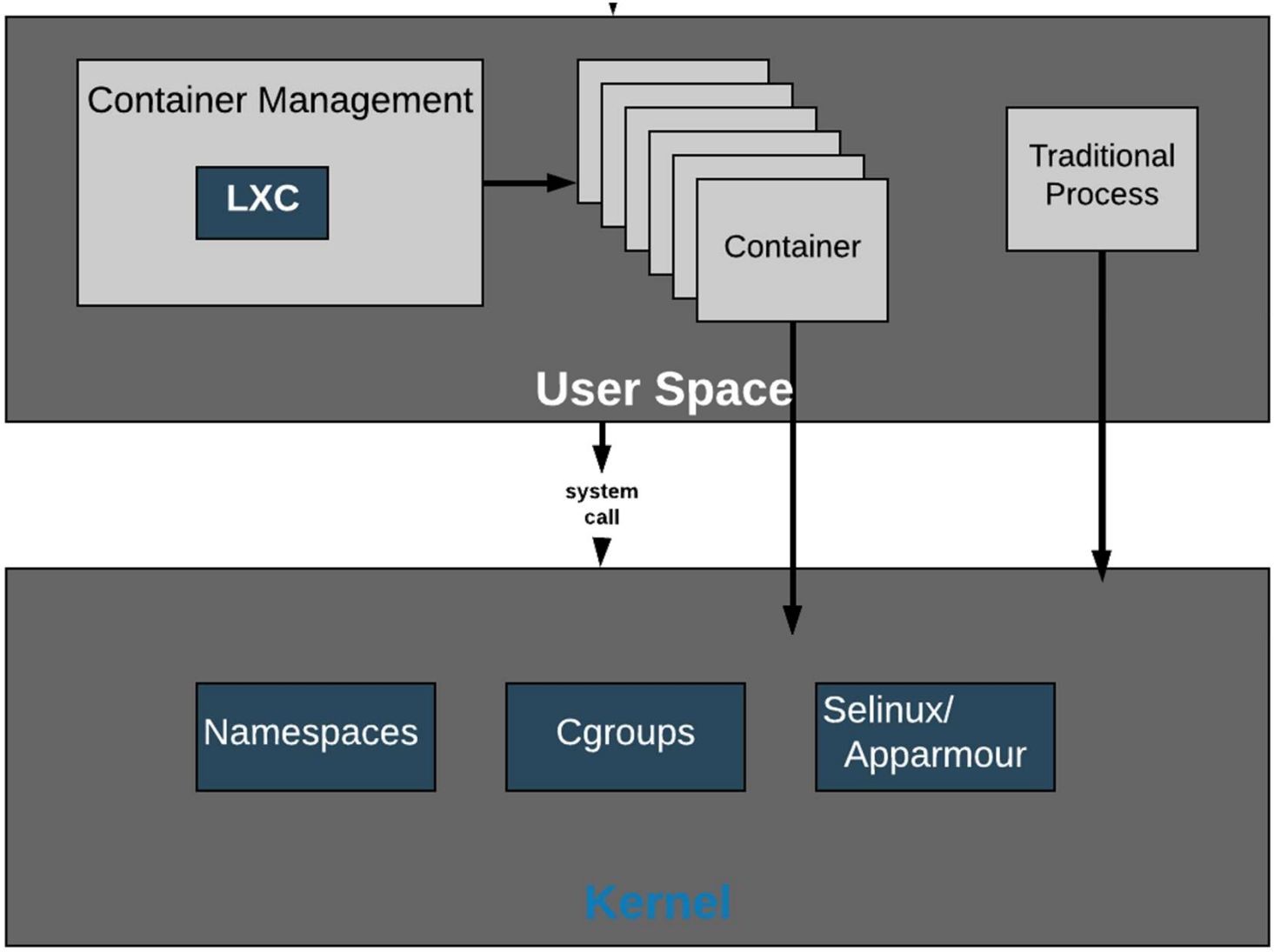
LAB - namespace

cgroups (Control Groups)

Name	Controls
cpuset	assigns individual processor/memory nodes
cpu	access to the processor resources
cpuacct	reports about processor usage
io	sets limit to read/write
memory	sets limit on memory usage
devices	allows access to devices
freezer	allows to suspend/resume
net_cls	allows to mark network packets
net_prio	set the priority of network traffic
perf_event	provides access to perf events
hugetlb	activates support for huge pages
pid	sets limit to number of processes

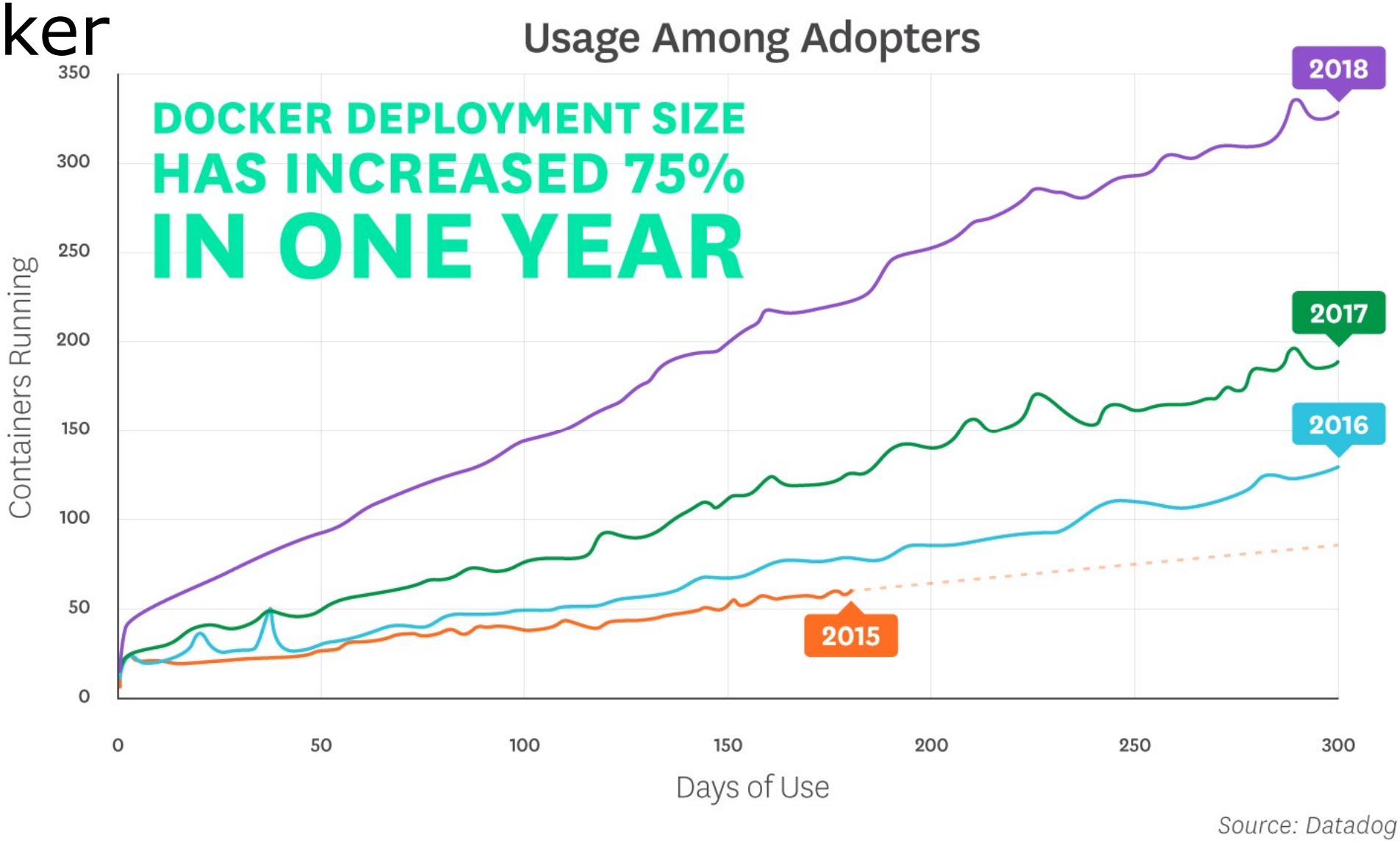
LAB - cgroups

LXC


Container Management
LXC
Container
Traditional Process
User Space
system call
Namespaces
Cgroups
Selinux/ Apparmour
Kernel

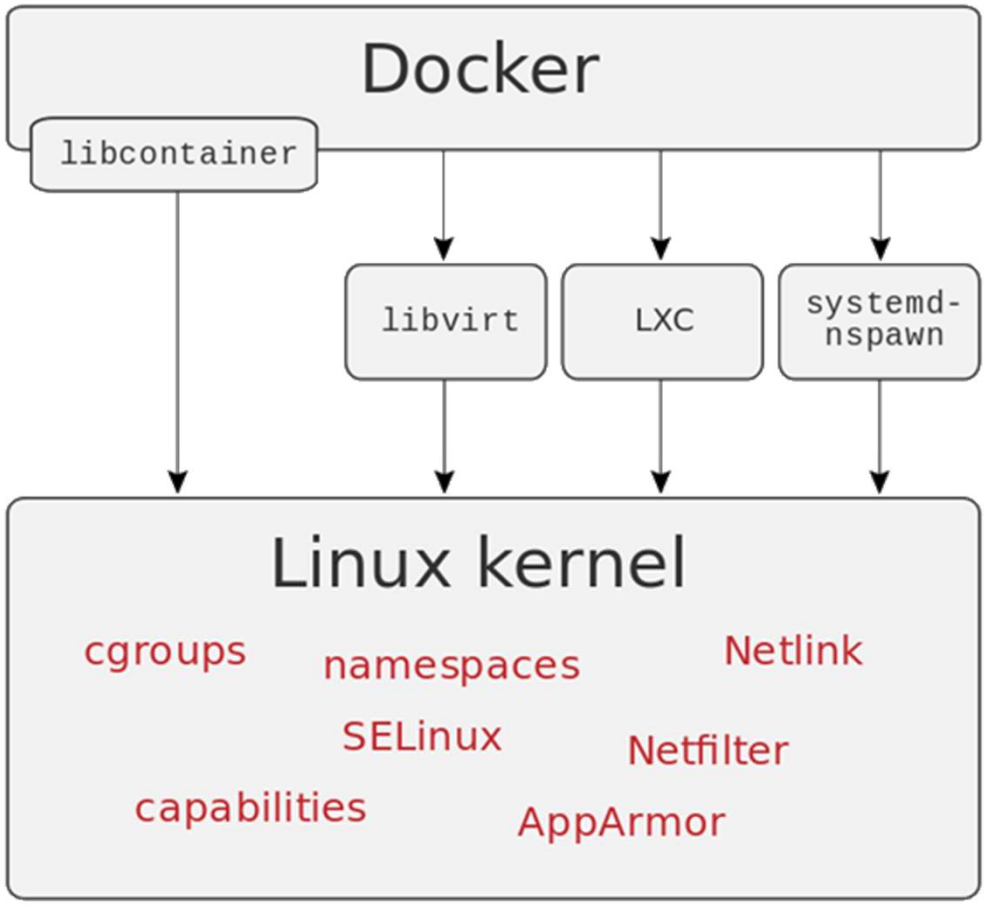
LAB - LXC

Docker

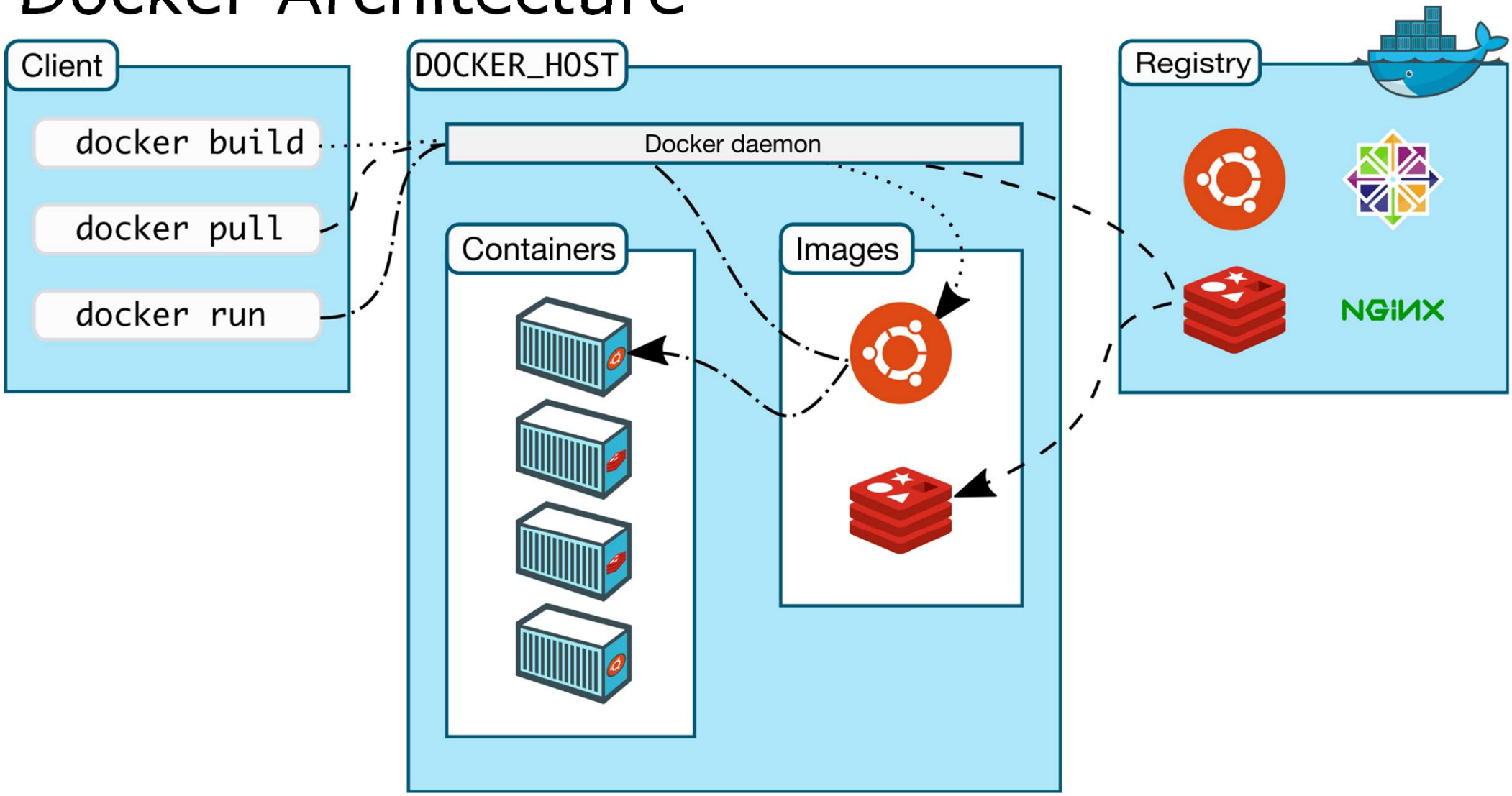


Source: Datadog

Docker


Docker
libcontainer
libvirt
LXC
systemd-
nspawn
Linux kernel
cgroups
namespaces
Netlink
SELinux
Netfilter
capabilities
AppArmor

Docker Architecture



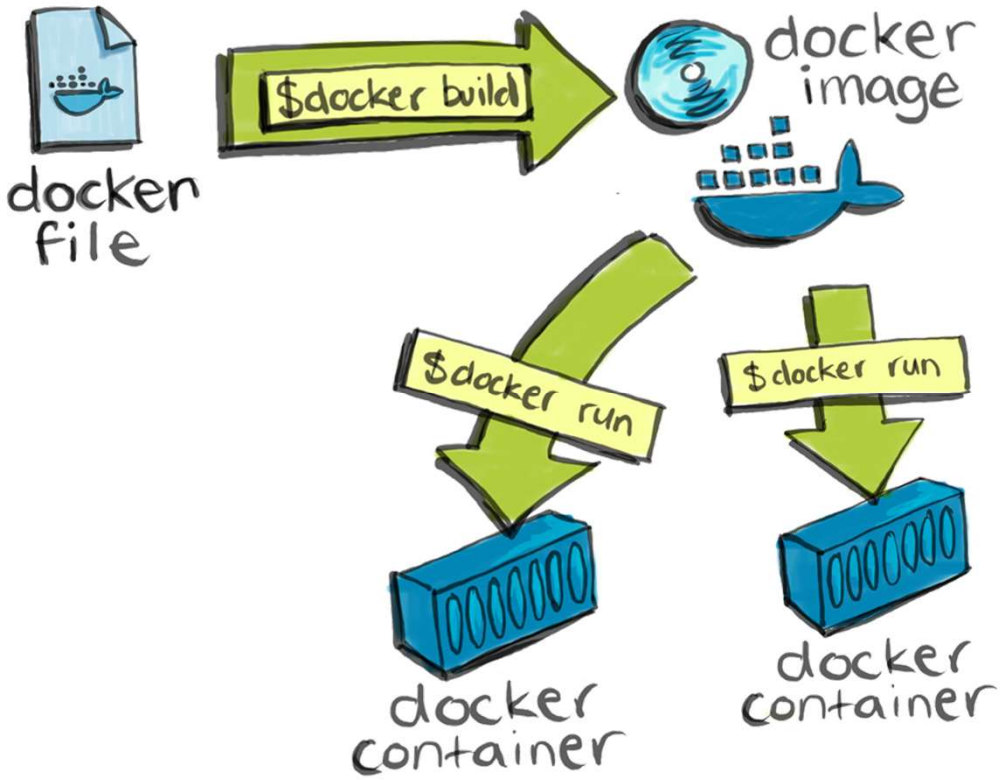
LAB - Docker

Docker

How First Time, Solomon Hykes shows docker to the public : The future of Linux Containers

https://www.youtube.com/watch?v=362sHaO5eGU

Dockerfile



Comment
INSTRUCTION arguments

docker build context [-f docker_file_path]

FROM

FROM scratch

FROM <image>[:<tag>] [AS <name>]

Base image karşılaştırma

WORKDIR

Sonraki komutlar için çalışma dizini belirler

COPY & ADD

COPY [--chown=<user>:<group>] <src>... <dest>

COPY [--chown=<user>:<group>] ["<src>",... "<dest>"]

ADD [--chown=<user>:<group>] <src>... <dest>

ADD [--chown=<user>:<group>] ["<src>",... "<dest>"]

RUN

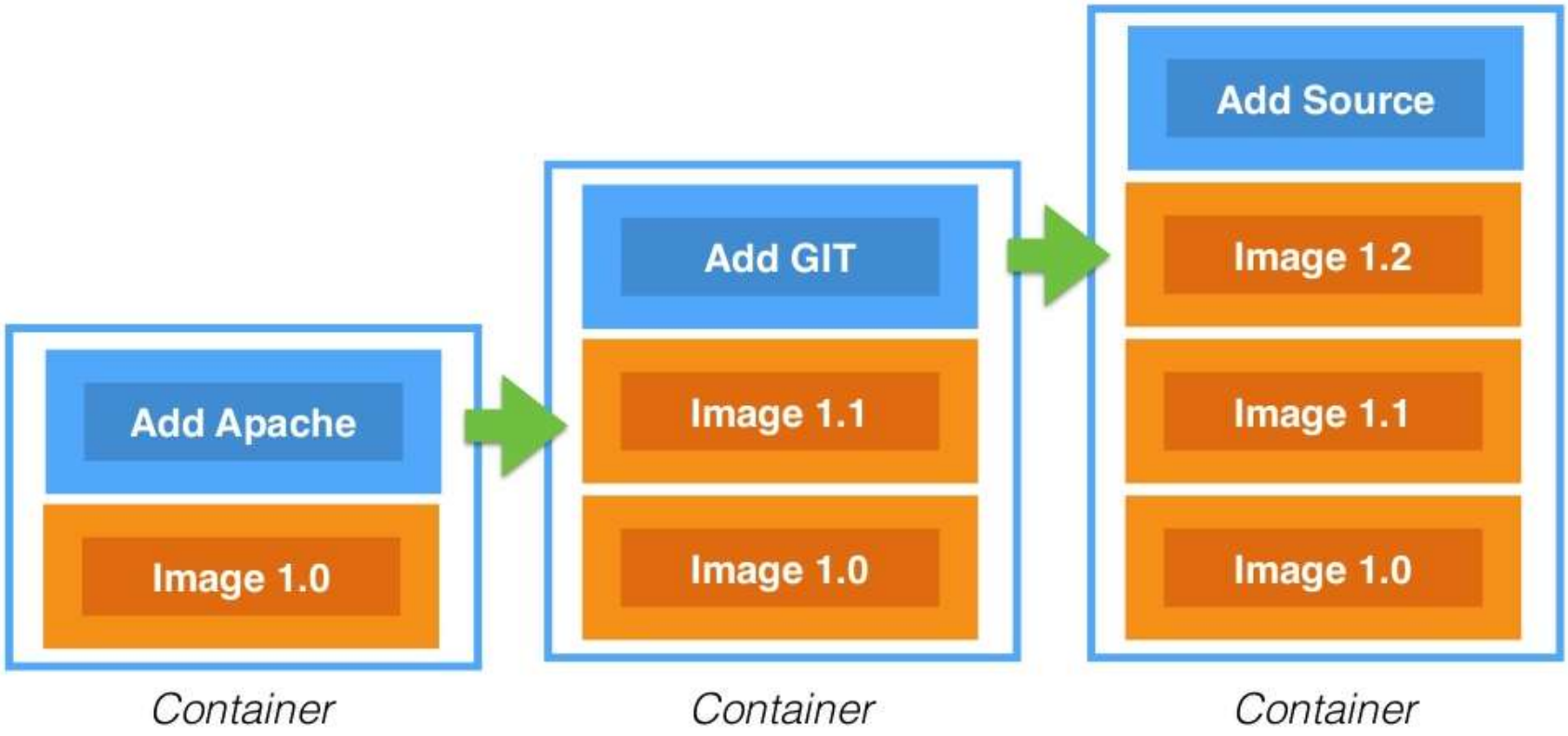
RUN ["executable", "param1", "param2"]

RUN <command>

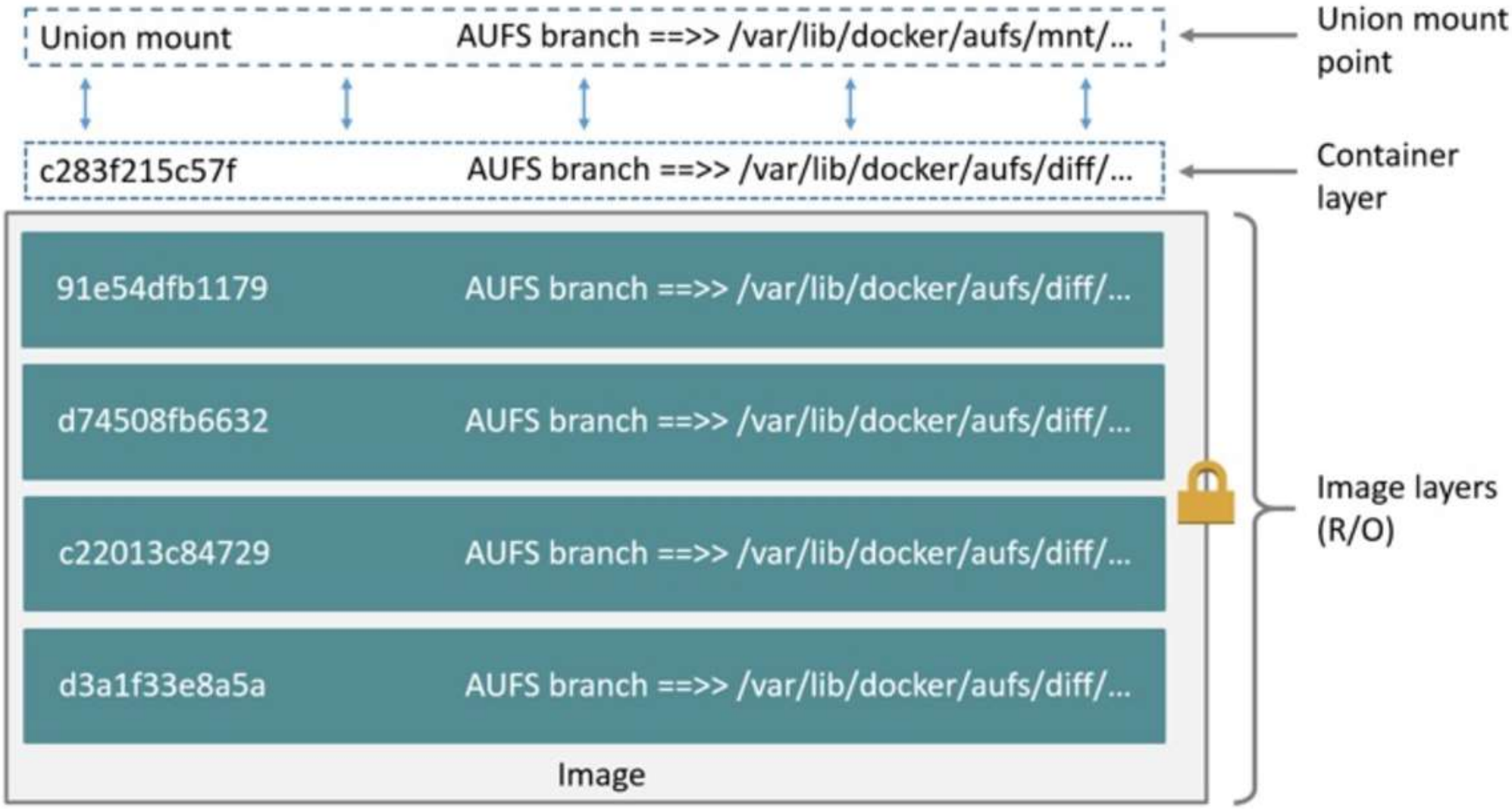
Docker Image Layers - AUFS

- **Union Mount** is a way of combining numerous directories into one directory that looks like it contains the content from all the them.
- **AUFS** stands for Another union filesystem or Advanced multi-layered unification filesystem (as of version 2). AUFS implements a union mount for Linux file systems.
- **AUFS storage driver** implements Docker image layers using the union mount system.
- **AUFS Branch**es (each Docker image layer)

AUFS


Add Apache
Image 1.0
Container
Add GIT
Image 1.1
Image 1.0
Container
Add Source
Image 1.2
Image 1.1
Image 1.0
Container

AUFS


Union mount
AUFS branch ==>> /var/lib/docker/aufs/mnt/...
Union mount point
c283f215c57f
AUFS branch ==>> /var/lib/docker/aufs/diff/...
Container layer
91e54dfb1179
AUFS branch ==>> /var/lib/docker/aufs/diff/...
d74508fb6632
AUFS branch ==>> /var/lib/docker/aufs/diff/...
c22013c84729
AUFS branch ==>> /var/lib/docker/aufs/diff/...
d3a1f33e8a5a
AUFS branch ==>> /var/lib/docker/aufs/diff/...
Image
Image layers (R/O)

CMD

CMD ["executable", "param1", "param2"]

CMD command param1 param2

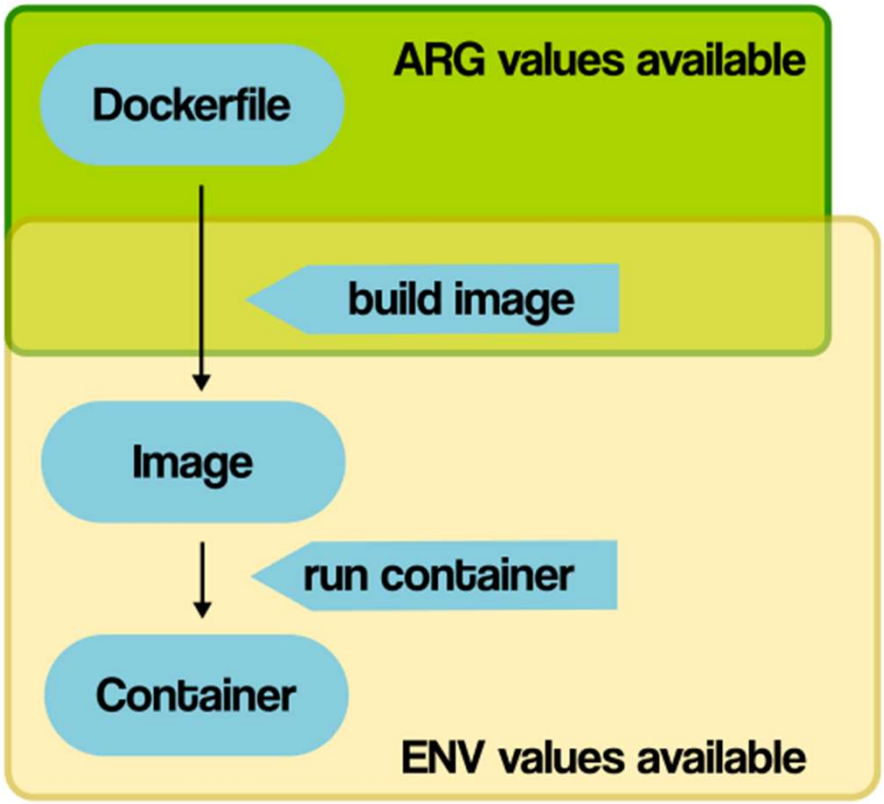
CMD ["param1", "param2"]

ENTRYPOINT

ENTRYPOINT ["executable", "param1", "param2"]

ENTRYPOINT command param1 param2

ARG & ENV



Dockerfile:
ARG required_var
ARG var_name=default_value
ENV foo=foo_value
ENV bar=${var_name}

Override ARG values:
docker build . --build-arg var_name=value

Overrice ENV values:
docker run -e "foo=other_foo_value" [...]
docker run --env-file=env_file_name [...]

EXPOSE

EXPOSE <port> [<port>/<protocol>...]

VOLUME

VOLUME ["/data"]

LAB - Dockerfile

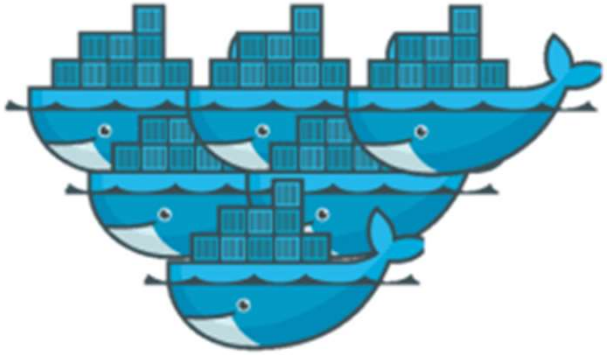
LAB – Multi-Stage Build

```
 ---> Running in b9dd4c16f4a0
Removing intermediate container b9dd4c16f4a0
 ---> 9365dcf93110
Step 3/17 : RUN adduser -D [redacted]
 ---> Running in 971cb3de76a2
Removing intermediate container 971cb3de76a2
 ---> 9bd8e44709ff
Step 4/17 : COPY [redacted]-SNAPSHOT.tar /tmp/
 ---> f1b15dc08104
Step 5/17 : RUN mkdir -p [redacted]-service
 ---> Running in e45c1db9d895
Removing intermediate container e45c1db9d895
 ---> 033c5286d6ab
Step 6/17 : RUN tar -xvf /tmp/[redacted]-SNAPSHOT.tar -C /
 ---> Running in 824fb78987bd
lib/
```

```
37Z  ---> Using cache
61Z  ---> a2f6783eae8a
07Z Step 4/27 : RUN update-ca-certificates
27Z  ---> Using cache
53Z  ---> b79629867426
15Z Step 5/27 : WORKDIR /work_space
2020-03-06T14:11:41.0011353Z  ---> Using cache
2020-03-06T14:11:41.0027139Z  ---> f936d3202603
2020-03-06T14:11:41.0042938Z Step 6/27 : COPY . .
2020-03-06T14:11:41.0057817Z  ---> b316e3a1c3ed
2020-03-06T14:11:41.0071867Z Step 7/27 : RUN chmod +x /work_space/[redacted]
2020-03-06T14:11:41.0092022Z  ---> Running in 13a0ef410530
2020-03-06T14:11:41.5415622Z Removing intermediate container 13a0ef410530
2020-03-06T14:11:41.5431076Z  ---> 5d8586939437
2020-03-06T14:11:41.5463772Z Step 8/27 : RUN dos2unix /work_space/[redacted]
2020-03-06T14:11:41.5653290Z  ---> Running in 2d015f419cfe
2020-03-06T14:11:42.2408575Z Removing intermediate container 2d015f419cfe
2020-03-06T14:11:42.2427617Z  ---> 154be716aab6
2020-03-06T14:11:42.2444000Z Step 9/27 : RUN /work_space/[redacted] clean buil
2020-03-06T14:11:42.2674685Z  ---> Running in 9c42b8aa1313
```

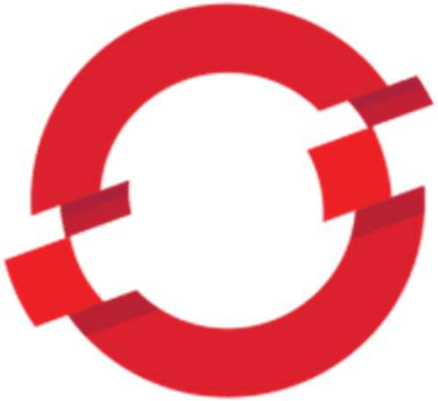
LAB – Docker-Compose

Gelecek Program

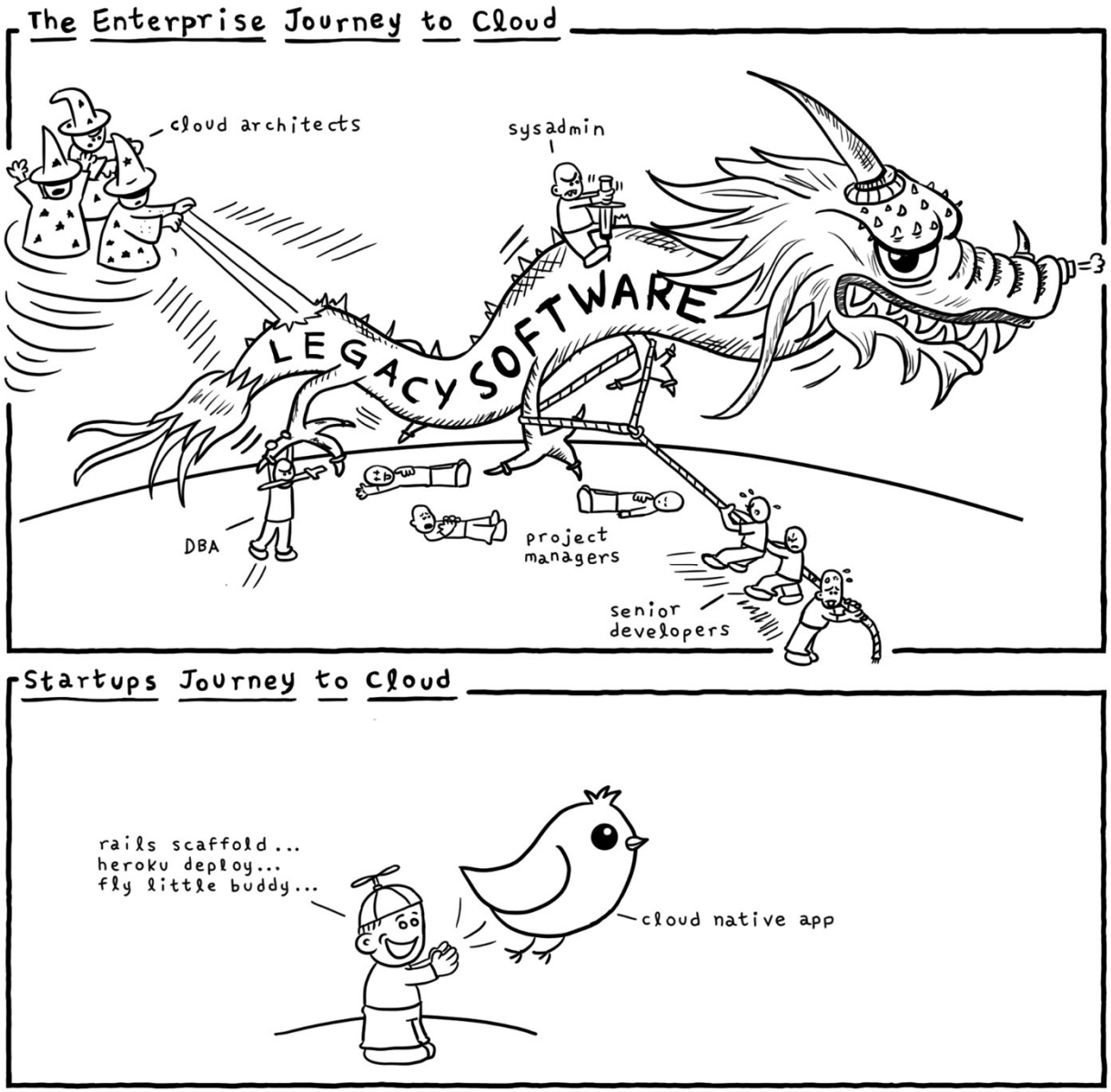


Docker Swarm

Kubernetes



OpenShift



Daniel Stori {turnoff.us}
Thanks to Michael Tharrington


Teşekkürler
https://github.com/mbdemirkan/containerCourse