

Docker for Beginners

"Any sufficiently advanced technology is indistinguishable from magic."

Arthur C. Clark

By James Reynolds

The History of Docker

- 2002, Linux Namespaces
- 2006, Linux cgroups
- Mar 2013, Solomon Hykes releases "Docker"
- Sept 2013, Red Hat partners
- Oct 2014+, Microsoft, EC2 (Amazon), IBM join
- By May 2016, Cisco, Google, Huawei have joined



What is Docker?

<pre>com.docker.cli run --rm -it amazonlinux bash-4.2# cat /etc/os-release NAME="Amazon Linux" VERSION="2" ID="amzn" ID_LIKE="centos rhel fedora" VERSION_ID="2" PRETTY_NAME="Amazon Linux 2" ANSI_COLOR="0;33" CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2" HOME_URL="https://amazonlinux.com/" bash-4.2#</pre>	<pre>root@c5ea31c05e10: / root@c5ea31c05e10:/# cat /etc/os-release NAME="Ubuntu" VERSION="20.04.1 LTS (Focal Fossa)" ID=ubuntu ID_LIKE=debian PRETTY_NAME="Ubuntu 20.04.1 LTS" VERSION_ID="20.04" HOME_URL="https://www.ubuntu.com/" SUPPORT_URL="https://help.ubuntu.com/" BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/" PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy" VERSION_CODENAME=focal UBUNTU_CODENAME=focal root@c5ea31c05e10:/#</pre>
<pre>@3bf7d9025de7:/ [root@3bf7d9025de7 /]# cat /etc/os-release NAME="CentOS Linux" VERSION="8" ID="centos" ID_LIKE="rhel fedora" VERSION_ID="8" PLATFORM_ID="platform:el8" PRETTY_NAME="CentOS Linux 8" ANSI_COLOR="0;31" CPE_NAME="cpe:/o:centos:centos:8" HOME_URL="https://centos.org/" BUG_REPORT_URL="https://bugs.centos.org/" CENTOS_MANTISBT_PROJECT="CentOS-8" CENTOS_MANTISBT_PROJECT_VERSION="8" [root@3bf7d9025de7 /]#</pre>	<pre>com.docker.cli run --rm -it alpine / # cat /etc/os-release NAME="Alpine Linux" ID=alpine VERSION_ID=3.13.0 PRETTY_NAME="Alpine Linux v3.13" HOME_URL="https://alpinelinux.org/" BUG_REPORT_URL="https://bugs.alpinelinux.org/" / #</pre>
<pre>root@c1b95c2dd132: / root@c1b95c2dd132:/# cat /etc/os-release NAME="Ubuntu" VERSION="18.04.5 LTS (Bionic Beaver)" ID=ubuntu ID_LIKE=debian PRETTY_NAME="Ubuntu 18.04.5 LTS" VERSION_ID="18.04" HOME_URL="https://www.ubuntu.com/" SUPPORT_URL="https://help.ubuntu.com/" BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/" PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy" VERSION_CODENAME=bionic UBUNTU_CODENAME=bionic root@c1b95c2dd132:/#</pre>	<pre>@923155766e33:/ LOGO=fedora-logo-icon CPE_NAME="cpe:/o:fedoraproject:fedora:33" HOME_URL="https://fedoraproject.org/" DOCUMENTATION_URL="https://docs.fedoraproject.org/en-US/fedora/f33/system-administrators-guide/" SUPPORT_URL="https://fedoraproject.org/wiki/Communicating_and_getting_help" BUG_REPORT_URL="https://bugzilla.redhat.com/" REDHAT_BUGZILLA_PRODUCT="Fedora" REDHAT_BUGZILLA_PRODUCT_VERSION=33 REDHAT_SUPPORT_PRODUCT="Fedora" REDHAT_SUPPORT_PRODUCT_VERSION=33 PRIVACY_POLICY_URL="https://fedoraproject.org/wiki/Legal:PrivacyPolicy" VARIANT="Container Image" VARIANT_ID=container [root@923155766e33 /]#</pre>

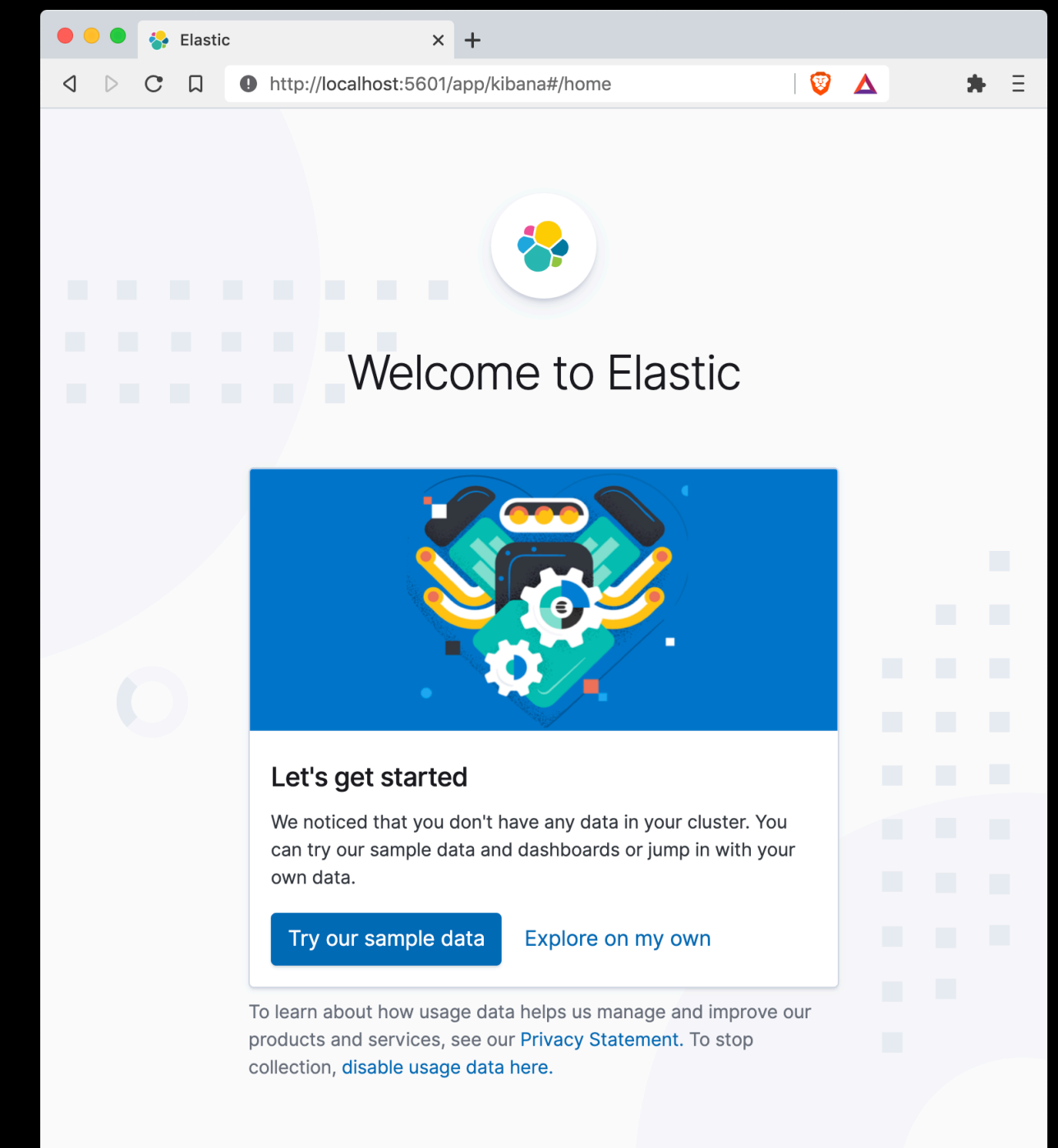
Why is Docker Magic?

- ELK on macOS
 - It was a lot of work
- Now 3 easy commands

```
git clone https://github.com/docker/awesome-compose.git
```

```
cd awesome-compose/elasticsearch-logstash-kibana
```

```
docker-compose up -d
```



Queue Suspense Music

"You're just moving the work from one place to another."

Ben Martin

"Distributing things solves scaling concerns, but introduces a whole new world of problems, many of which were previously solved by monoliths."

<https://stackoverflow.blog/2020/11/30/the-overflow-49-container-orchestration-explained/>

Try it Yourself

- Install Docker - <https://docs.docker.com/engine/install/>
- Launch Docker and install everything
- Run:

```
docker run --rm -it ubuntu ; cat /etc/os-release  
  
git clone https://github.com/docker/awesome-compose.git  
cd awesome-compose/elasticsearch-logstash-kibana  
  
docker-compose up -d
```

Try it Yourself

- Easily cleaned up
- All contained
- Sandboxed
 - The Docker sandbox has been broken
- Please try it out ``docker run --rm -it <image>``
- hub.docker.com

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


What's in the Command?

```
docker run --rm -it <image>
```

- `docker run`
- `--rm` (removes the container when you exit it--but not the image)
- `-it` (interactive terminal)
- `<image>` (will download if you don't have it)

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


Run All These - <https://hub.docker.com/>

`docker run --rm -it <image>`

alpine	php	httpd	puppet/puppetserver
ubuntu	ruby	nginx	jasonrivers/nagios
debian	perl	drupal	osticket/osticket*
photon	pypy	plone	librenms/librenms*
fedora	swift	wordpress	alexandreoda/snort
centos	maridb*	joomla*	atanasi/matlab:v97
python	mysql*	smcline06/cacti	gibby/openconnect

*requires more setup, please audit 3rd party images for security

`docker run --rm -it ubuntu ; cat /etc/os-release`

Clean Up Everything

- Docker Desktop GUI

- Bug icon



- View disk usage

- `docker system df -v`

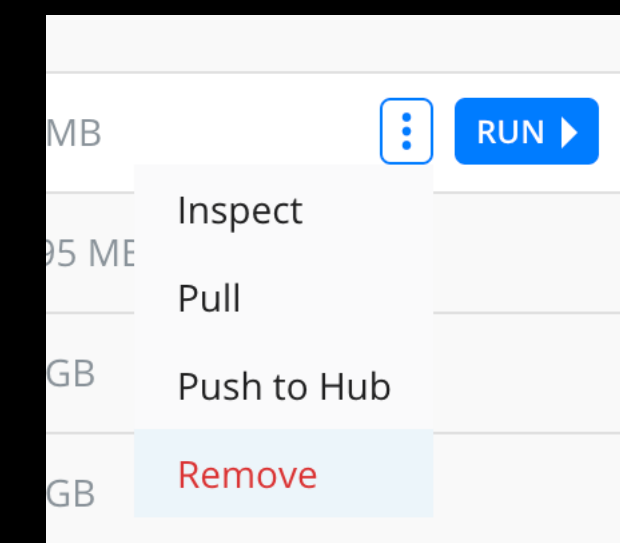
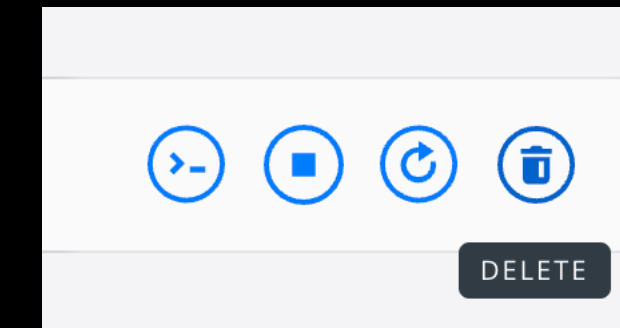
- Remove redundant stuff

- `docker system prune`

- Reclaim space in images

- `docker run --privileged --pid=host docker/desktop-reclaim-space`

`docker run --rm -it ubuntu ; cat /etc/os-release`



My Question To You

- Have you tried it yet?

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

```
git clone https://github.com/docker/awesome-compose.git
```

```
cd awesome-compose/elasticsearch-logstash-kibana
```

```
docker-compose up -d
```


About Docker Desktop for Mac

- <https://docs.docker.com/docker-for-mac/>

- macOS 10.14, 10.15, 11
- 2010+ Intel Mac, 4 GB RAM
- M1 Mac beta
- App is 1.5 GB (it's Electron.js)
- 3 million Docker Desktop users
 - Half are on Mac



```
docker run --rm -it ubuntu ; cat /etc/os-release
```

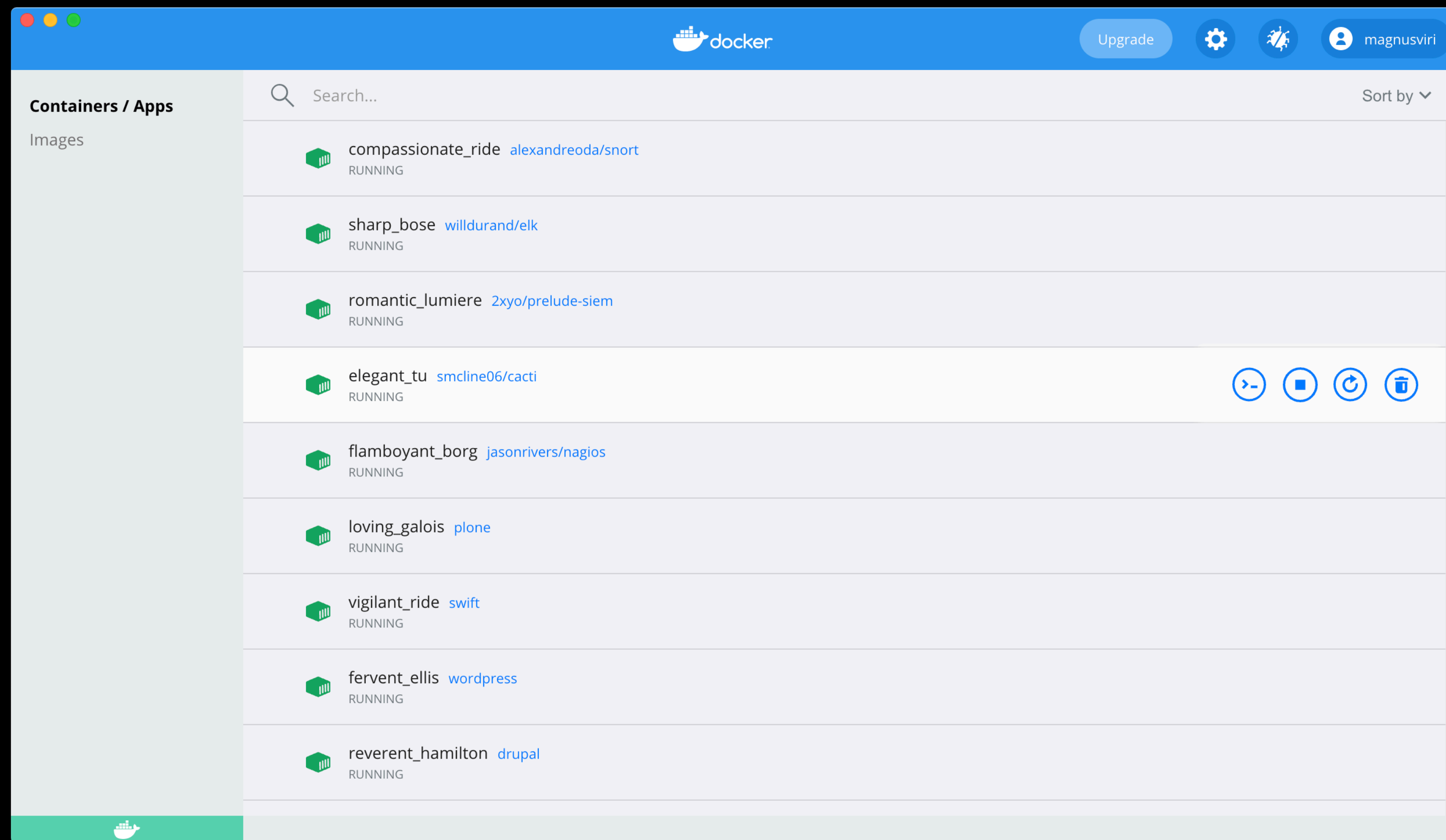
About Docker Desktop for Mac

- <https://docs.docker.com/docker-for-mac/>
- `~/Library/Containers/com.docker.docker/Data/vms/0/Docker.raw`
 - Default max size is 64 GB
- Small Linux VM
 - No end user access
 - Runs in user space
 - No kernel extension
 - Doesn't require VirtualBox or VMware (c.f. Vagrant)

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Docker Desktop GUI

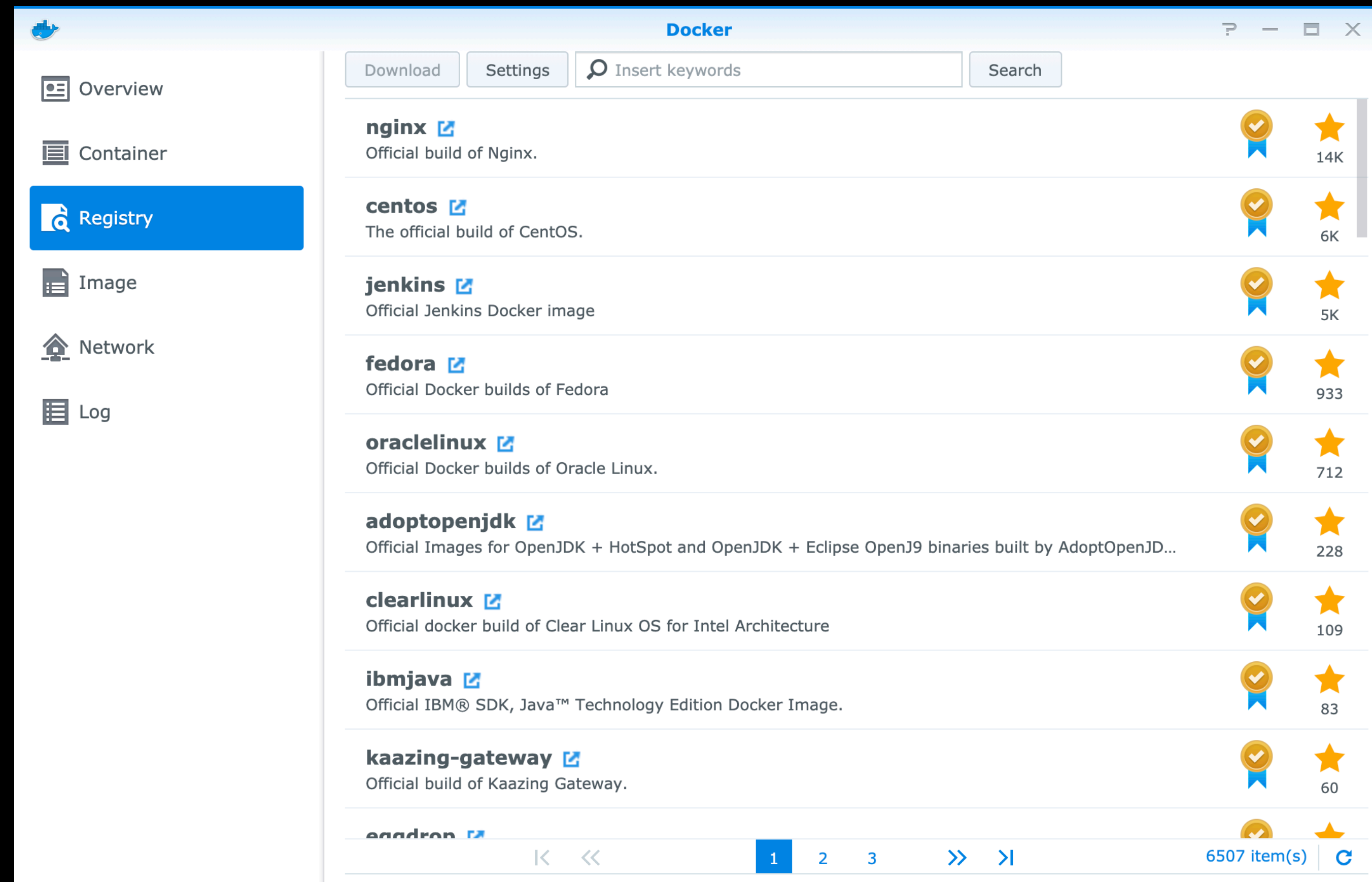
Very Simple



```
docker run --rm -it ubuntu ; cat /etc/os-release
```


Synology's Docker Manager

A Little More Complex

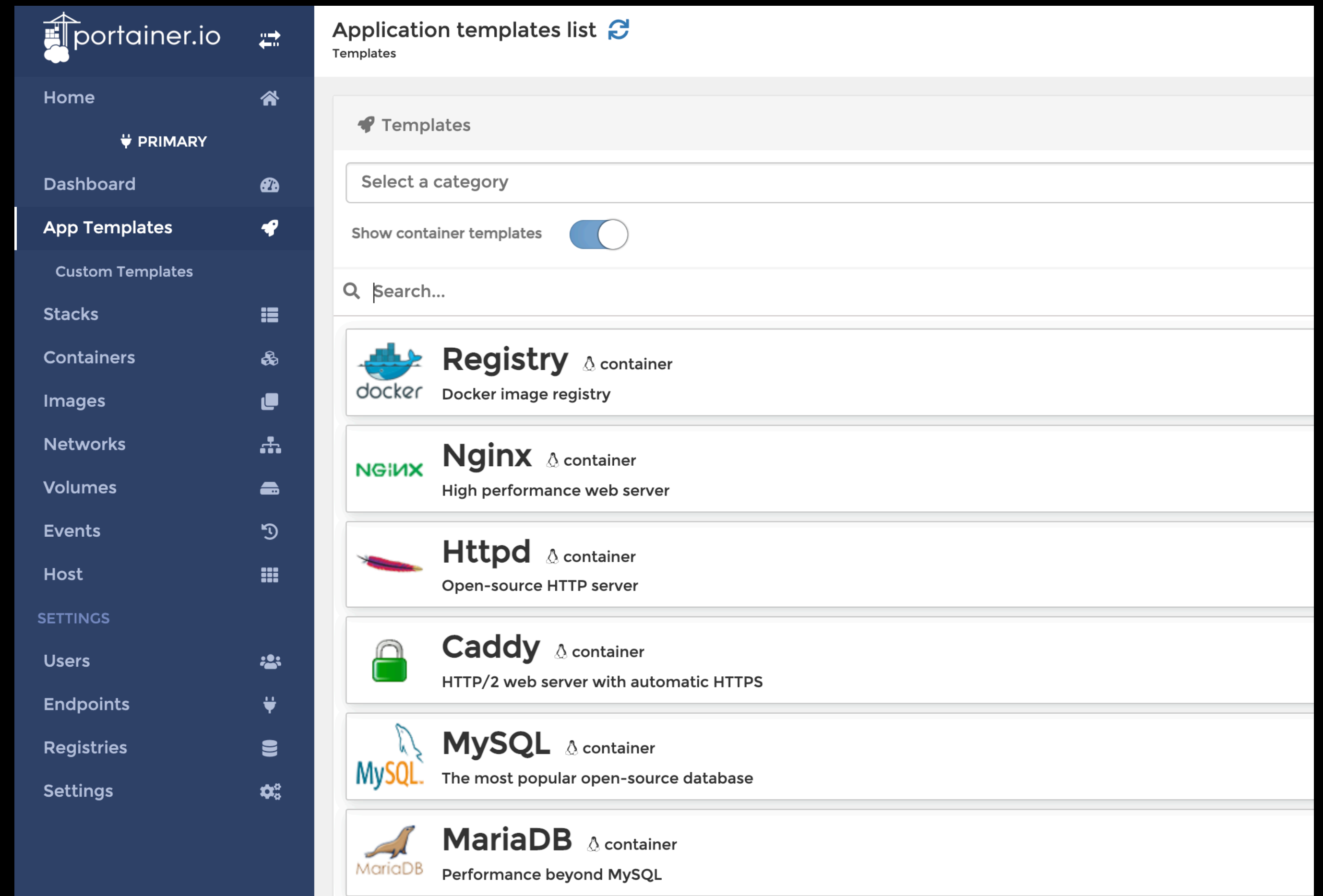
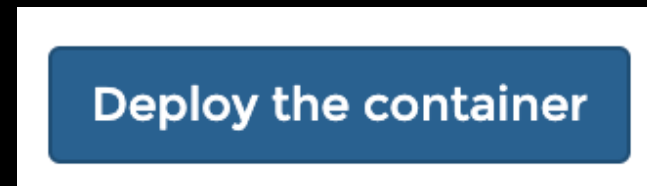


```
docker run --rm -it ubuntu ; cat /etc/os-release
```

portainer.io A Better Docker Manager

Even More Control and Complexity

- One click install
- Portainer also works with Kubernetes

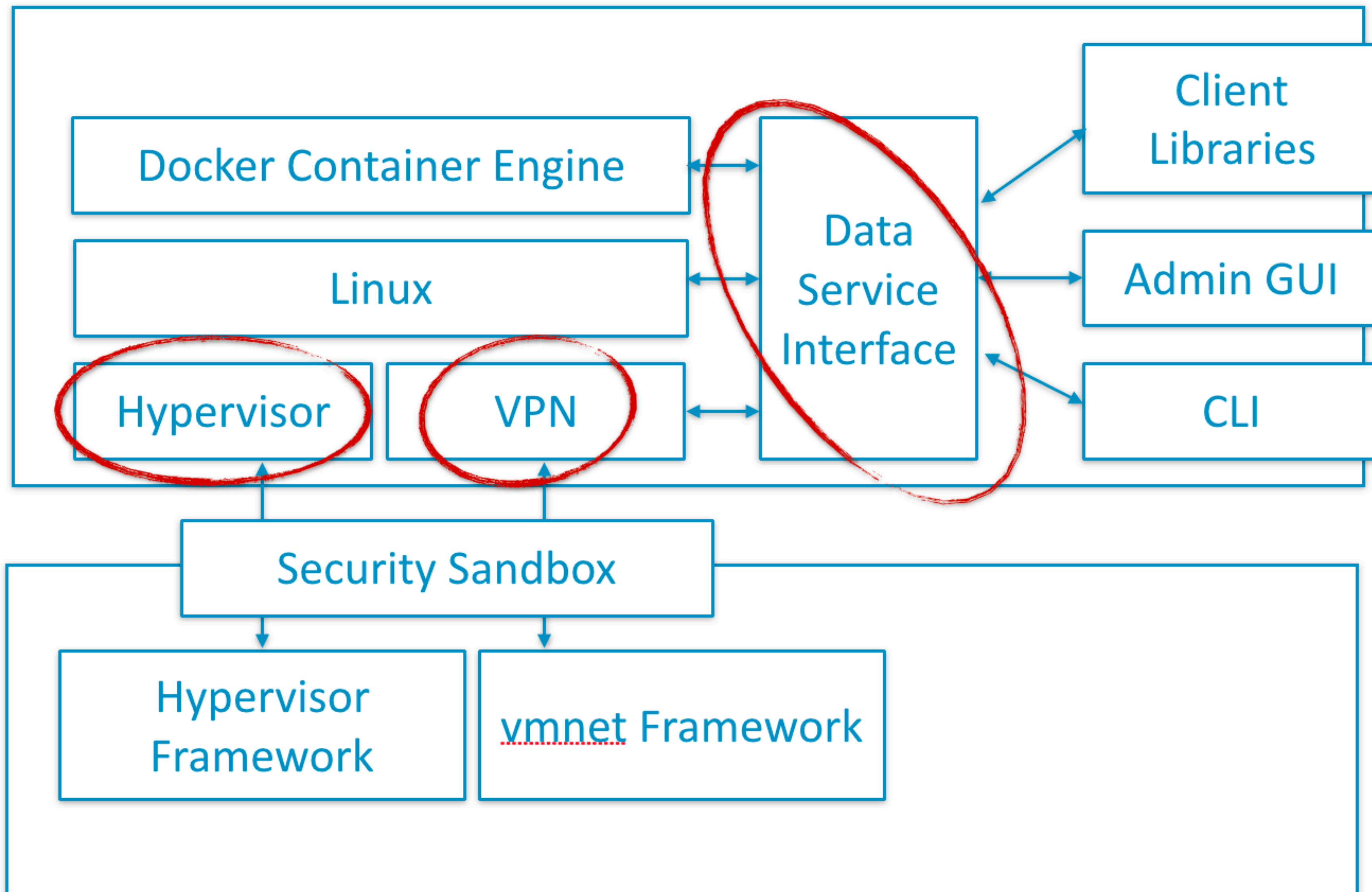


```
docker run --rm -it ubuntu ; cat /etc/os-release
```

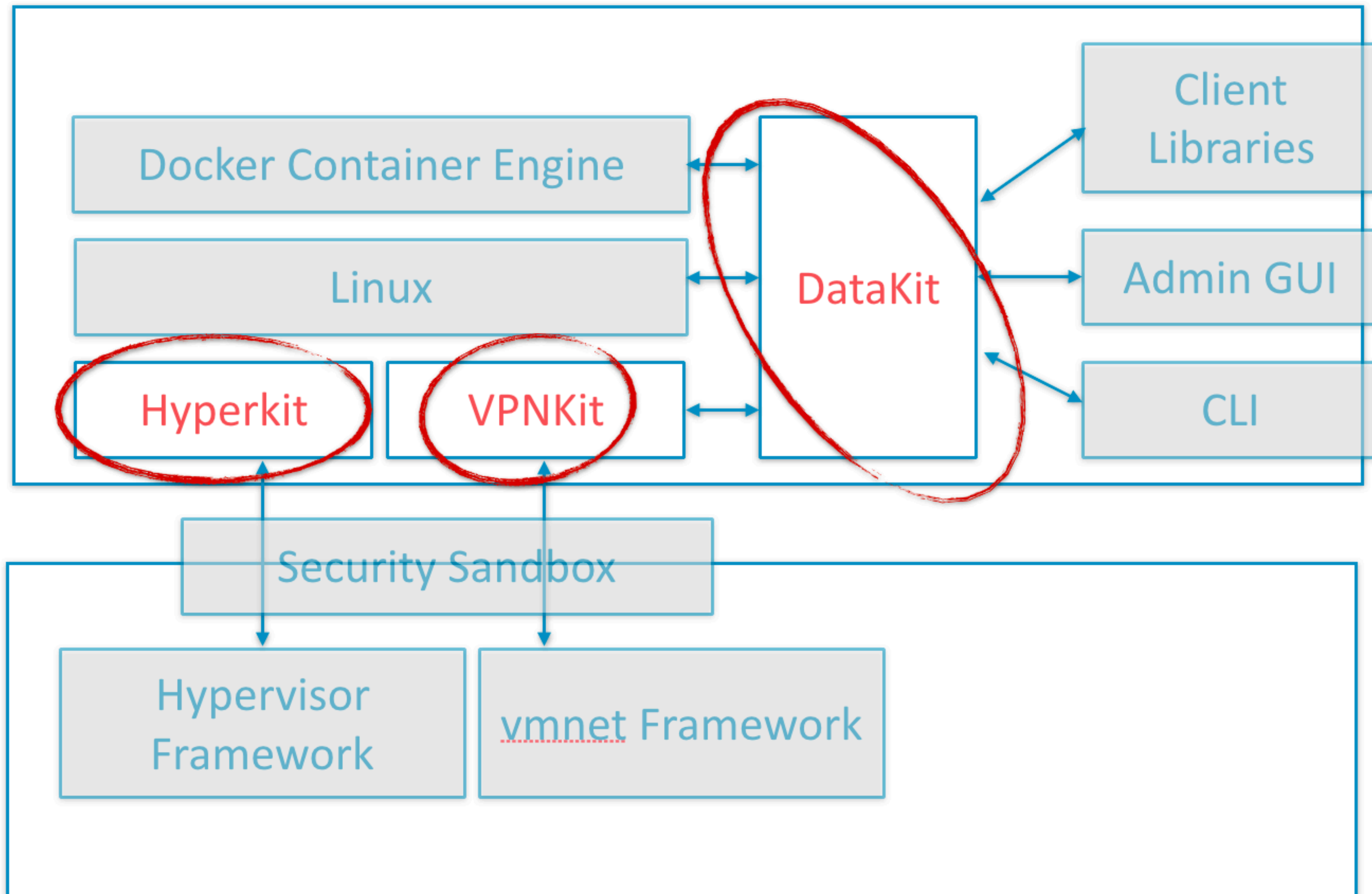
The macOS Docker Hypervisor

- HyperKit hypervisor on x86
 - <https://github.com/docker/hyperkit>
- Apple's Hypervisor framework
 - <https://developer.apple.com/documentation/hypervisor>
- M1 preview release

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

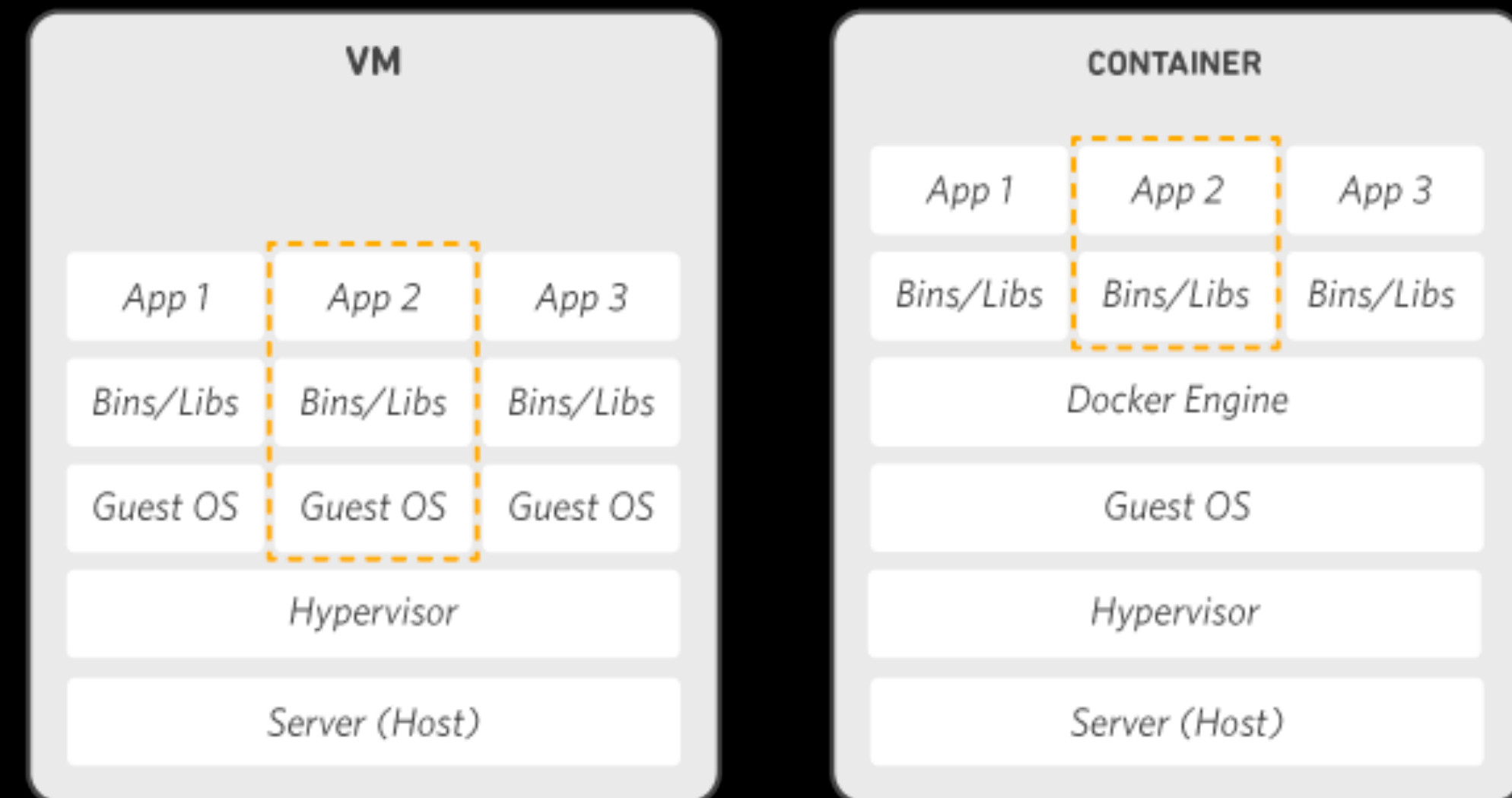


```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Containers vs Virtual Machines

Virtual Machine vs Virtual OS or Virtual Environment (VE)

- VM hypervisor = hardware
 - Hard drive, network card, etc
 - Requires real hardware
- The Docker Engine = OS
 - FS calls, network stack, etc
 - Requires a real host kernel



```
docker run --rm -it ubuntu ; cat /etc/os-release
```


Containers vs Virtual Machines

Persistence

- VM's
 - Ethernet, hard drive, display
- Containers
 - Immutable and ephemeral
 - Persistence must be configured
 - Sandboxed
 - Networking must be configured

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


Containers vs Virtual Machines

Security

- VM's
 - Hypervisor
- Containers
 - Linux kernel
 - Images

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Containers vs Virtual Machines

Booting and Speed

- VM's
 - Like a real computer
 - Several seconds
- Containers
 - Host is often a VM
 - The container OS is virtual
 - It doesn't boot

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Containers vs Virtual Machines

Dependencies

- VM's
 - "DLL Hell"
- Containers
 - A minimal host OS
 - Images contain dependences
 - Image base contains barely anything

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


About Those Runtime Dependencies

The Olden Days

- Hardly anyone coded
- Coding was slow
- Coding was hard
- Hardly any resources for coding
- Hardly any code
- Shared libraries!



```
docker run --rm -it ubuntu ; cat /etc/os-release
```

About Those Runtime Dependencies

Now-a-days

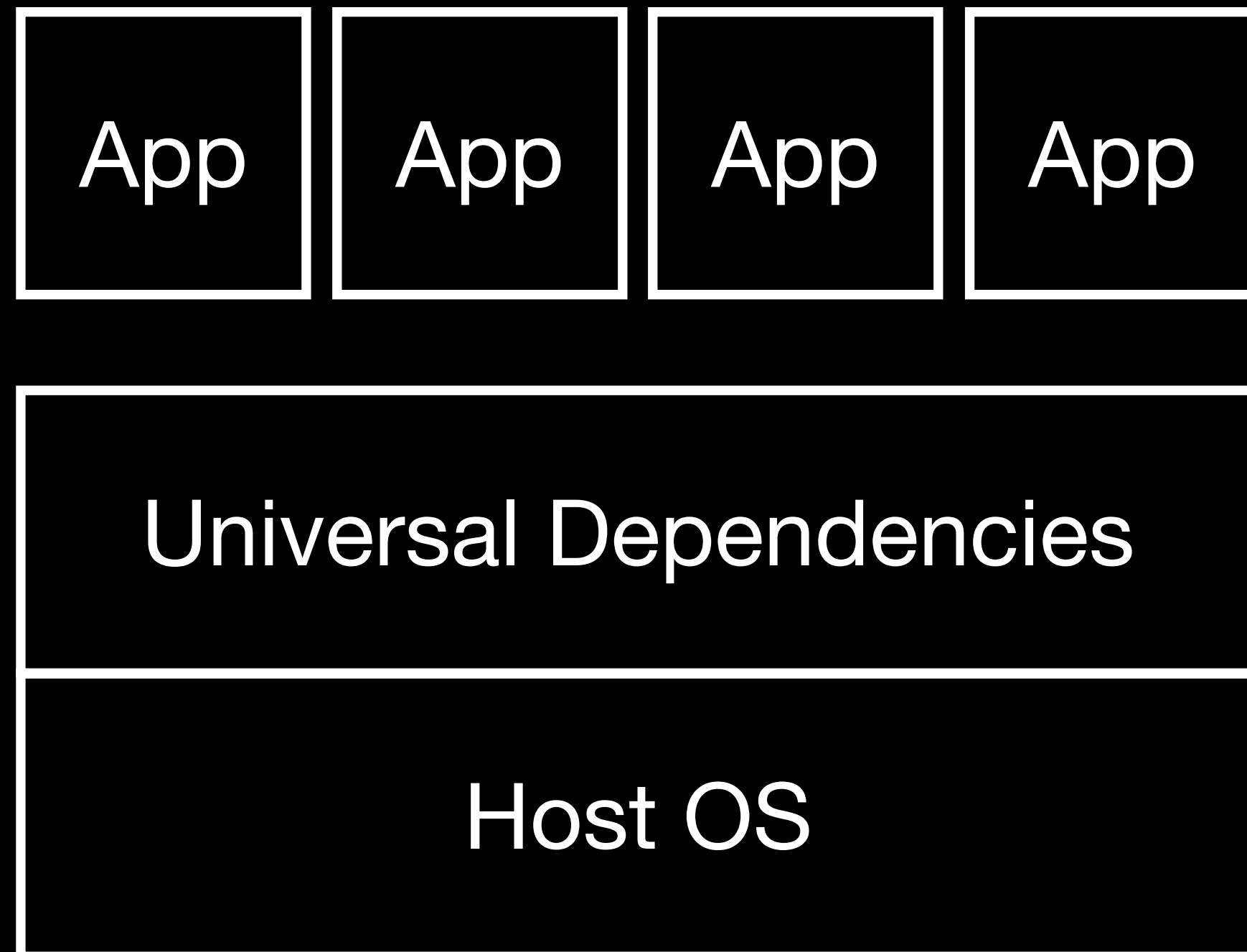
- Coding is fast and easy!
- Everyone codes
- Github
- RAM and hard drive space!
- Code is constantly changing...
- Software libraries are constantly changing



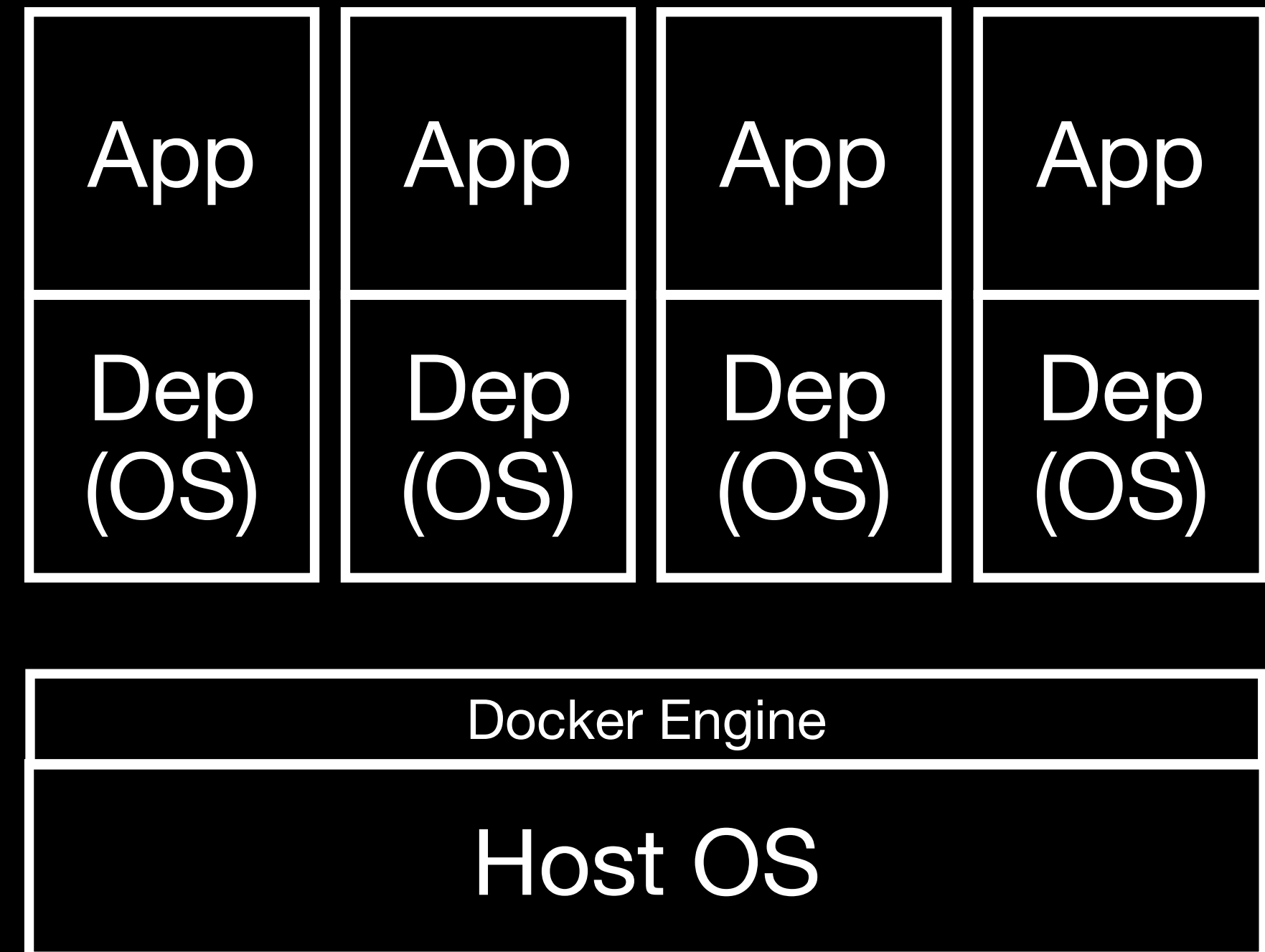
```
docker run --rm -it ubuntu ; cat /etc/os-release
```


About Those Runtime Dependencies

Olden days



Now-a-days



```
docker run --rm -it ubuntu ; cat /etc/os-release
```

About Those Docker Base OS Images

The Dependencies Include the Base OS Image...

Distribution	Version	Size
Fedora	v.25	231 MB
CentOS	7	183 MB
Ubuntu	16.04	118 MB
Alpine	3.6	3.98 MB

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


About Alpine Linux (the Docker version)

- Core components
 - musl libc
 - busybox
 - apk-tools
- It's a kernel with a basic c library
- A few tools to install stuff
- Designed for security



```
docker run --rm -it ubuntu ; cat /etc/os-release
```

How Is This Different From Electron.js?

- Docker can run GUI apps too
- Electron.js
 - Contains dependencies
 - WORA
 - Is not small
 - Is not sandboxed

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

What is Microsoft Doing?

Riding the Wave



- Windows can run Linux or Windows images
- Windows Docker Desktop Linux
 - Hyper V
 - Windows Subsystem for Linux (WSL 2)
- Windows Container OSes
 - Windows Server Core (cf. Ubuntu)
 - Windows Nano Server (cf. Alpine)

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


How to Run Windows Containers on macOS

See: <https://github.com/StefanScherer/docker-windows-box/>

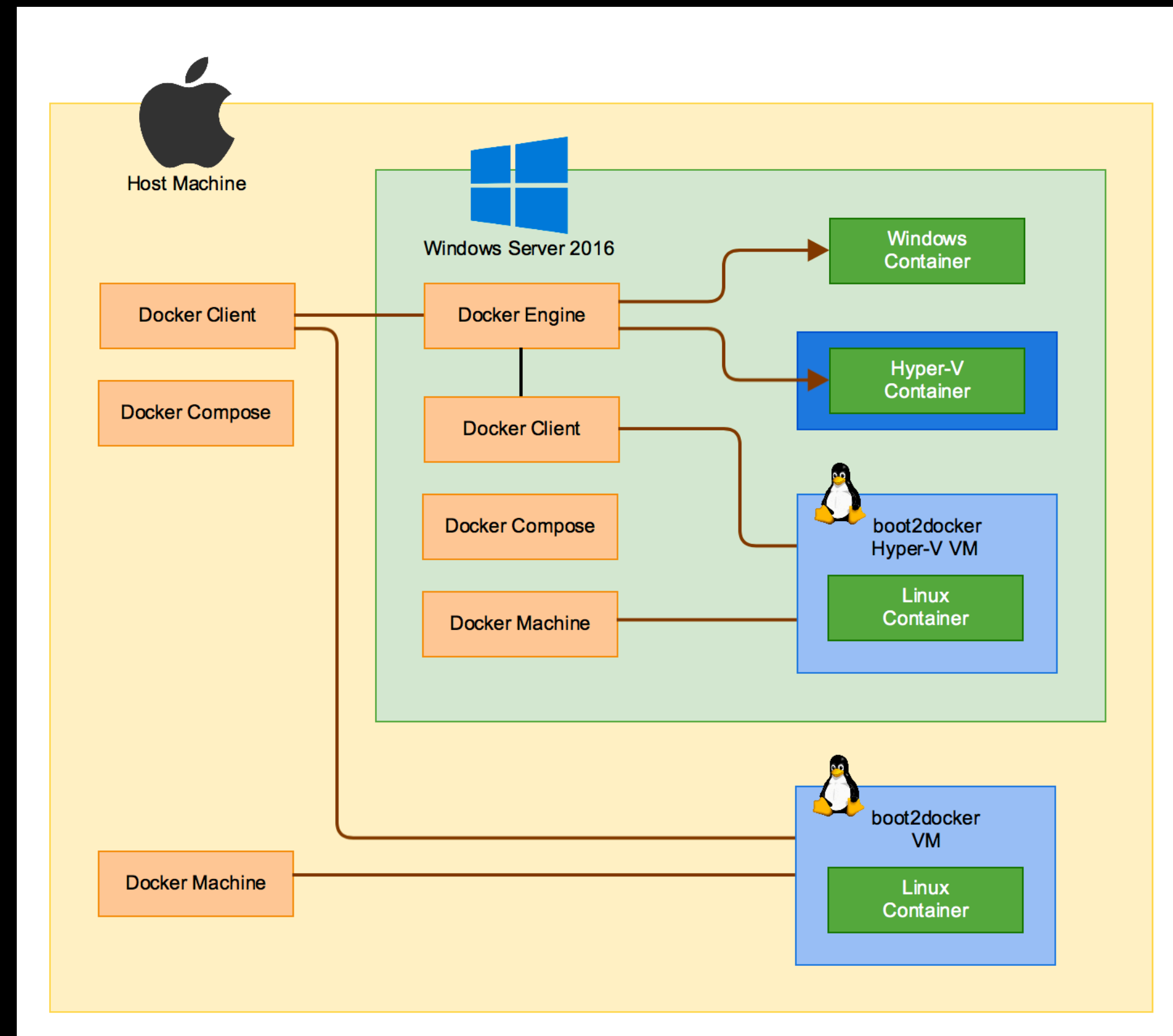
```
git clone https://github.com/  
StefanScherer/docker-windows-box.git
```

```
cd docker-windows-box  
vagrant plugin install vagrant-reload  
vagrant up
```

In Windows VM:

```
git clone https://github.com/  
StefanScherer/dockerfiles-windows
```

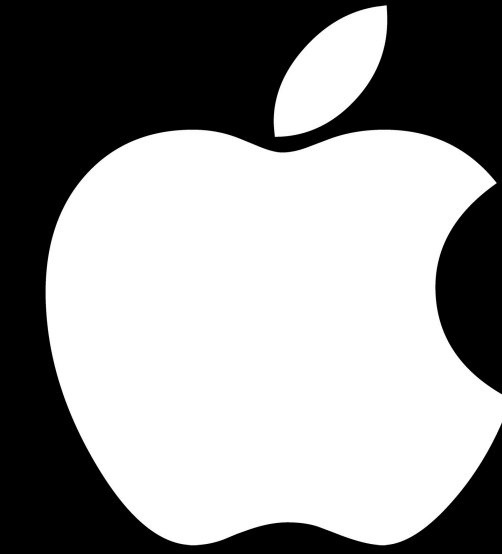
```
cd dockerfiles-windows\node  
PowerShell .\build.ps1
```



```
docker run --rm -it ubuntu ; cat /etc/os-release
```

macOS 10.14+ in Docker?

You Can Orchestrate VM's



- True macOS containerization depends on Apple
 - I'm not holding my breath
- ~~Computer~~ Mac must be booted to Linux
 - **Warning:** hackintosh territory
 - <https://github.com/sickcodes/Docker-OSX>
 - Requires KVM
 - <https://github.com/kholia/OSX-KVM>

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Dockerfile

It All Starts Here

- Each line of a Dockerfile is an image layer
 - Layers are like Radmind overloads
 - Each Dockerfile line is a layer
 - Build (or rebuild) an image: `docker build .`
 - If a line changes everything below is rebuilt
- <https://docs.docker.com/engine/reference/builder/>

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


Dockerfile

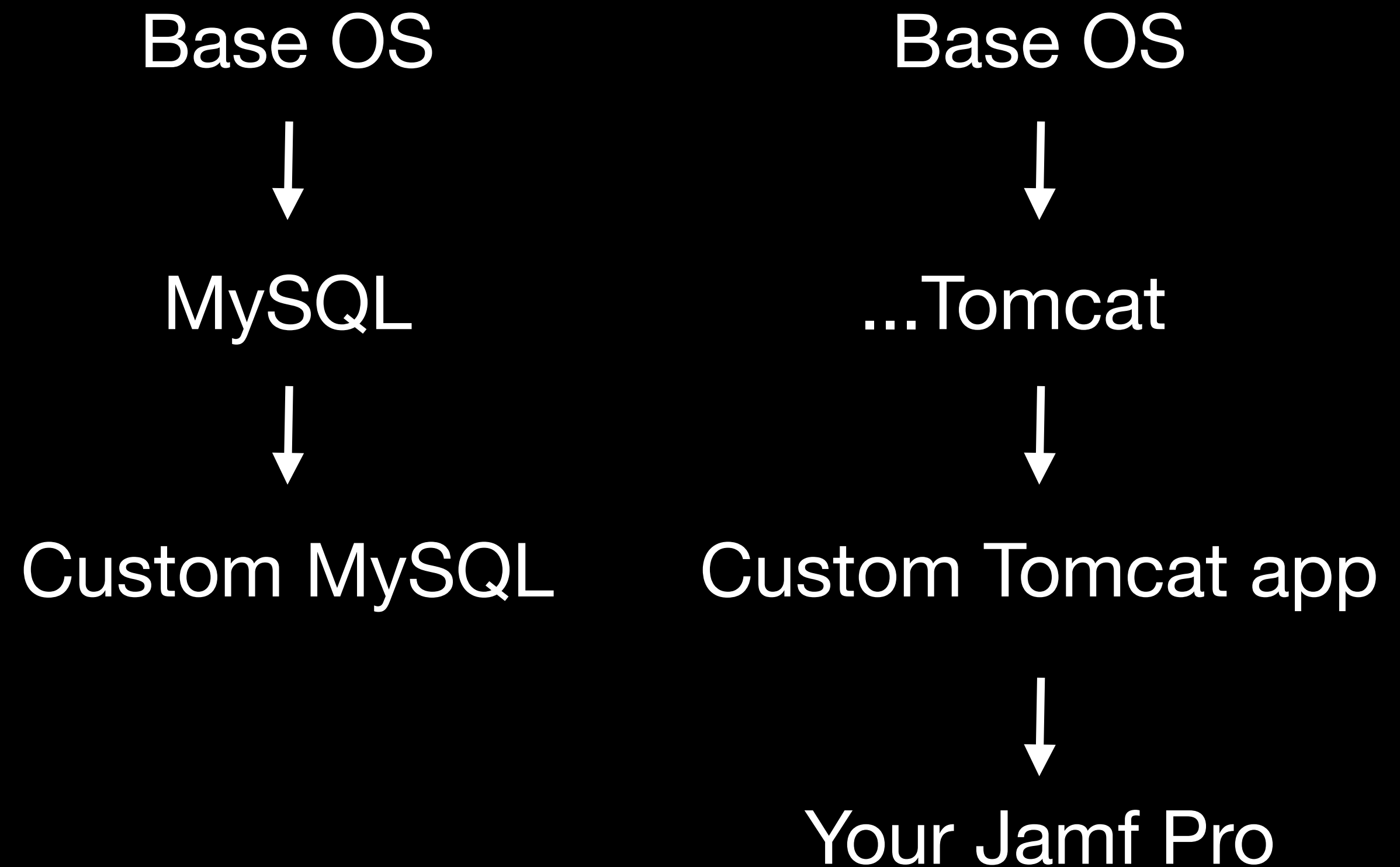
Base OS Image Layer

- The first line is a base OS or contains a base OS
 - Base OS contains no cruft
 - Base OS is small
 - Alpine is an ideal
 - Ubuntu, Fedora, CentOS (RIP), etc
- Add software and configure
 - Bash commands, Puppet, Ansible, Chef, etc

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Dockerfiles for Jamf

- There's 2 images
 - MySQL
 - Tomcat
- These 2 Base OS'es don't have to be the same distro
- I don't know how it handles kernel versioning



```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Dockerfiles for Jamf

<https://github.com/magnusviri/jamfpro-docker-compose>

- Simplified version
- It's this easy because of jamfdevops' Dockerfile

```
FROM jamfdevops/jamfpro:0.0.12
```

```
ADD ROOT.war /data/
```

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


jamfdevops/jamfpro:0.0.12

<https://github.com/jamf/jamfpro>

```
FROM tomcat:8.5.57-jdk11-openjdk-slim-buster
```

```
LABEL Maintaner JamfDevops <devops@jamf.com>
```

```
RUN apt-get update -qq && \
```

```
    DEBIAN_FRONTEND=noninteractive apt-get install --ignore-missing --no-install-recommends -y jq curl unzip && \
```

```
    apt-get clean && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/* && \
```

```
    adduser --disabled-password --gecos '' tomcat && \
```

```
    rm -rf /usr/local/tomcat/webapps && \
```

```
    mkdir -p /usr/local/tomcat/webapps
```

```
COPY startup.sh /startup.sh
```

← This is a 26 line shell script

```
COPY log4j.stdout.replace /log4j.stdout.replace
```

```
COPY configuration.sh /configuration.sh
```

← This is a 180 line shell script

```
CMD ["/startup.sh"]
```

```
VOLUME /usr/local/tomcat/logs
```

```
EXPOSE 8080
```

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

mysql:5.7

<https://github.com/docker-library/mysql/blob/3fde29717aea785d4e37a97c9990ff0fda37e544/5.7/Dockerfile.debian>

```
# NOTE: THIS DOCKERFILE IS GENERATED VIA "apply-templates.sh"
```

```
#
```

```
# PLEASE DO NOT EDIT IT DIRECTLY.
```

```
#
```

```
FROM debian:buster-slim
```

```
# add our user and group first to make sure their IDs get assigned consistently, regardless of whatever dependencies get added
```

```
RUN groupadd -r mysql && useradd -r -g mysql mysql
```

```
RUN apt-get update && apt-get install -y --no-install-recommends gnupg dirmngr && rm -rf /var/lib/apt/lists/*
```

```
..... 100 lines total
```

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Putting it All Together

- The `docker` command line
 - Runs containers
 - Connects containers
 - Opens ports
 - Mounts host folders to container folders
 - But it can't do it all in one line

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


Putting things together: docker-compose

- `docker-compose``

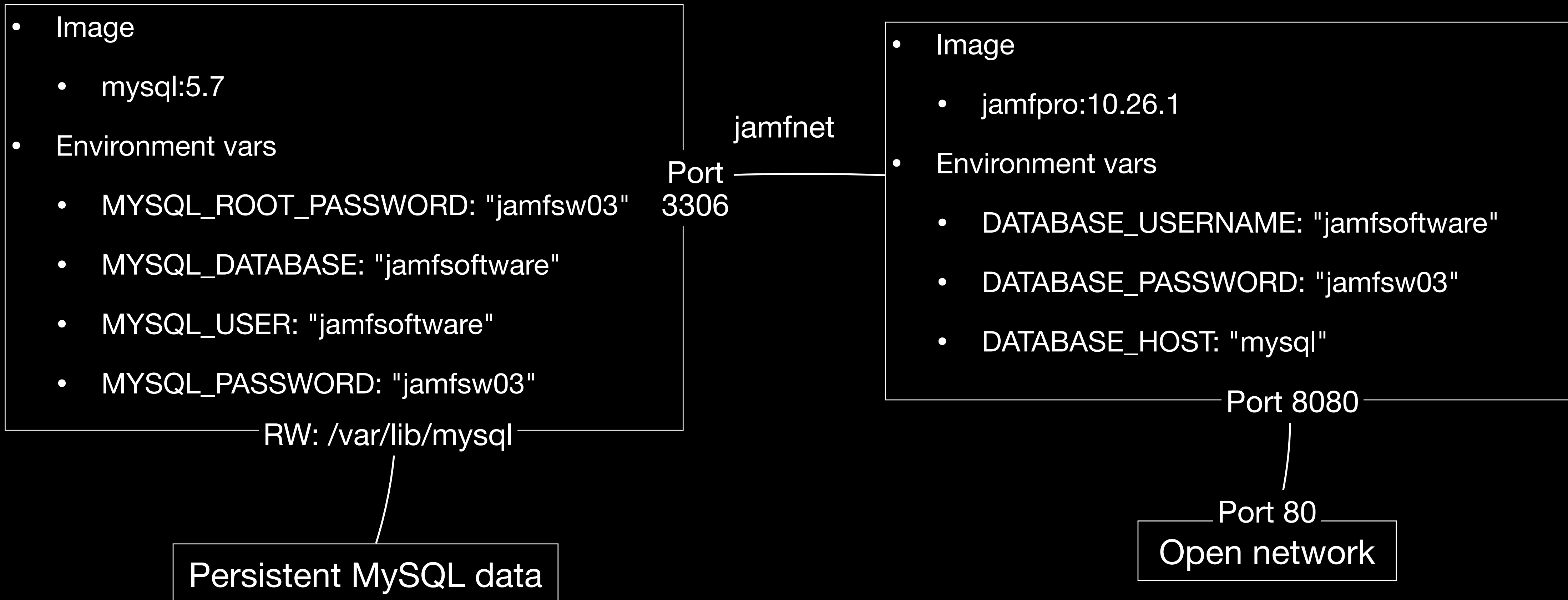
- Start it with

```
docker-compose up -d
```

- `jamf/docker-compose.yml` would create a container named `jamf`

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


Docker Compose Visualized



```
docker run --rm -it ubuntu ; cat /etc/os-release
```

docker-compose.yml for Jamf Pro

<https://github.com/magnusviri/jamfpro-docker-compose>

```
version: "3"

networks:
  jamfnet:

services:
  mysql:
    image: "mysql:5.7"
    networks:
      - jamfnet
    ports:
      - "3306:3306"
    environment:
      - MYSQL_DATABASE
      - MYSQL_PASSWORD
      - MYSQL_ROOT_PASSWORD
      - MYSQL_USER
    volumes:
      - ./mysql-data:/var/lib/mysql

  app:
    image: "jamfpro:${JAMF_PRO_VERSION}"
    build:
      context: ./
      args:
        - JAMF_IMAGE_VERSION
        - JAMF_PRO_VERSION
    networks:
      - jamfnet
    ports:
      - "80:8080"
    environment:
      DATABASE_USERNAME: "${MYSQL_USER}"
      DATABASE_PASSWORD: "${MYSQL_ROOT_PASSWORD}"
      DATABASE_HOST: "mysql"
    depends_on:
      - mysql
```

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Environment Variables

<https://github.com/magnusviri/jamfpro-docker-compose>

JAMF_IMAGE_VERSION=0.0.12

JAMF_PRO_VERSION=10.26.1

MYSQL_ROOT_PASSWORD=jamfsw03

MYSQL_DATABASE=jamfsoftware

MYSQL_USER=jamfsoftware

MYSQL_PASSWORD=jamfsw03

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


Demo

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Development vs Production

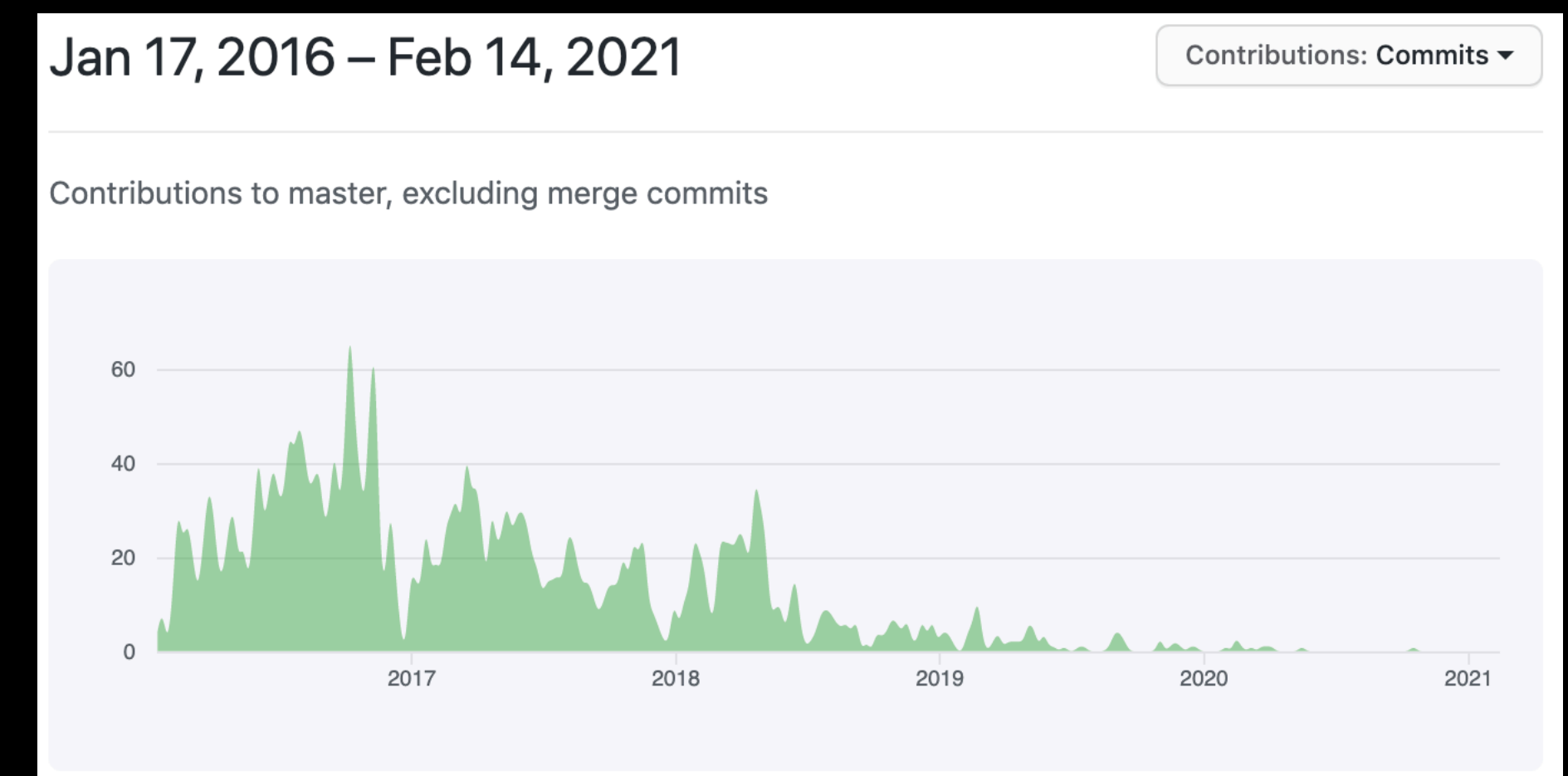
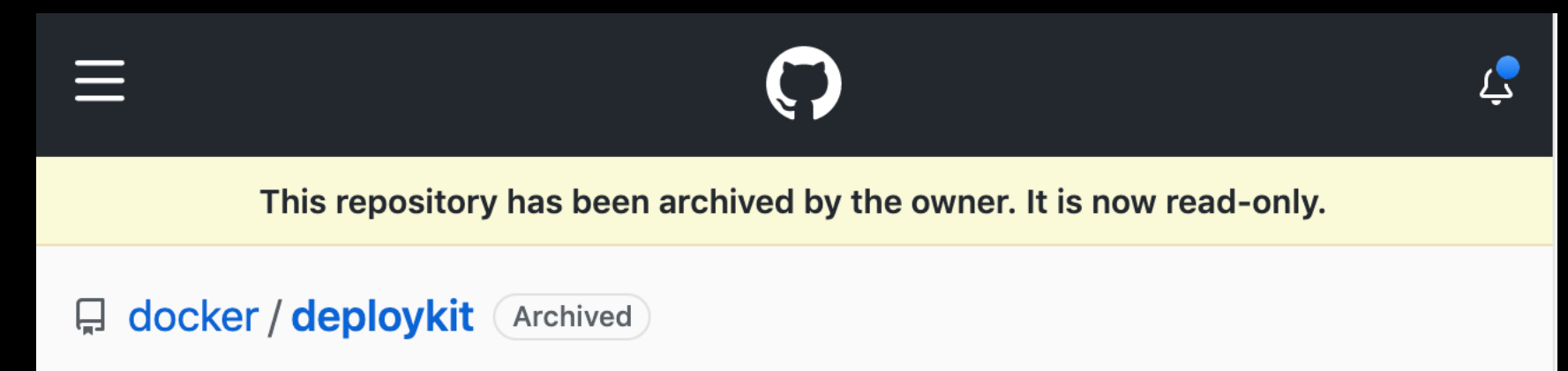
- Developer or learning servers?
 - Use Docker
- Production?
 - Cloud (AWS, Azure, GCP, Docker, IBM, etc)
 - **Warning!** It's like any other server
 - With lots of extra "features"!

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Moving to Production

The Graveyard

- Archived, superseded, or abandoned projects
 - Without notice
- Check the commit graph
 - It's hard to tell "done" from "dead"



```
docker run --rm -it ubuntu ; cat /etc/os-release
```


Setting up a Production Docker Server

For a Single Node Deployment (who does that?...)

- Use Ubuntu
- iptables
- docker-compose
 - Specify "restart: always"
- Run this: <https://github.com/docker/docker-bench-security>

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


Docker Automation

- Ansible, Chef, Puppet, Saltstack, Hashicorp Terraform
- Kubespray
- LinuxKit

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Orchestration

Working with "Services" not "Containers"



- It's just distributed computing
- Docker -> Compose -> Docker Swarm -> ? -> Kubernetes
- Other Orchestrators
 - Nomad, OpenShift, Rancher, Mesos & Marathon, Kontena
 - Microk8s, Minikube, k3d/k3s, k0s, more?

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Kubernetes

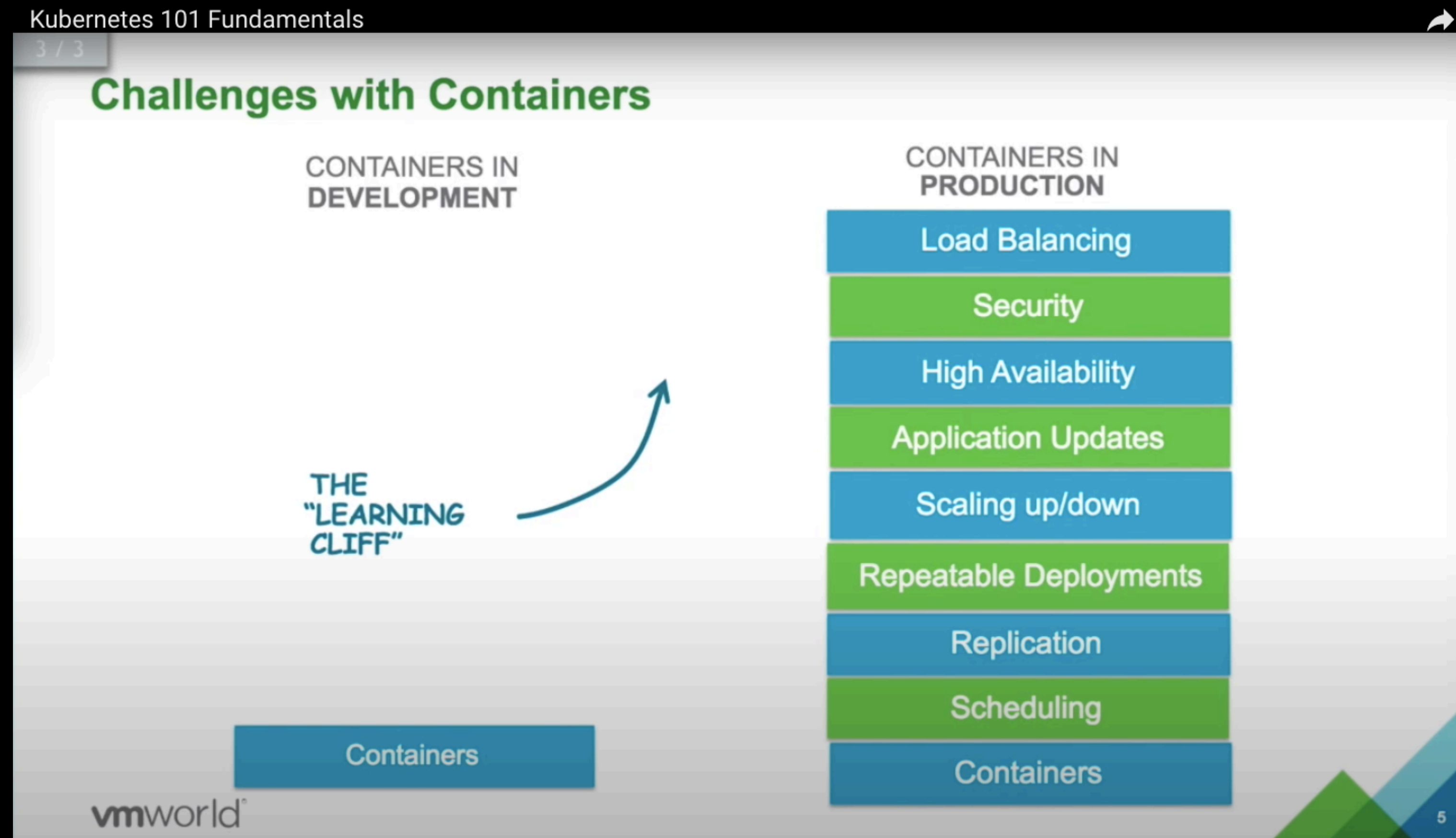
The grand master of orchestration and scheduling



- Very complex and difficult
- "Does everything" with 99.999% uptime
- Google, IBM, Microsoft, VMware
 - No site license for VMware Tanzu
- Relationship with Docker
- CHPC and UIT

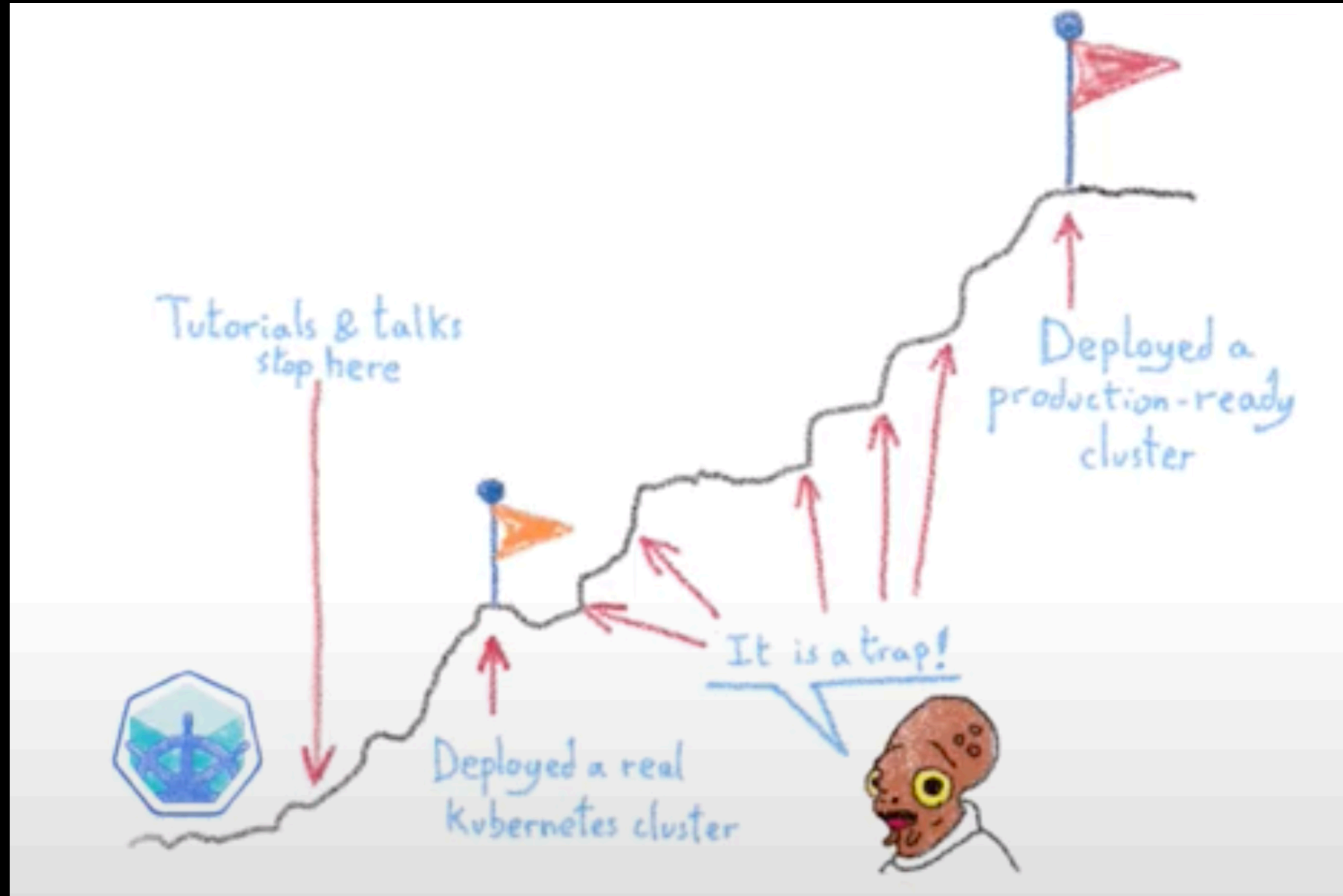
```
docker run --rm -it ubuntu ; cat /etc/os-release
```


The "Learning Cliff"



```
docker run --rm -it ubuntu ; cat /etc/os-release
```

"It's a Trap!"



```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Orchestrating Xcode: AWS, Orka

- AWS and MacStadium's Orka
 - <https://aws.amazon.com/ec2/instance-types/mac/>
 - <https://www.macstadium.com/orka>
- These are VM's not containers
 - Works with Kubernetes
- For Xcode
- Other apps?

```
docker run --rm -it ubuntu ; cat /etc/os-release
```


"I Don't Develop, I Can't Deploy it, How is This Relevant?"

If You Still Aren't Convinced

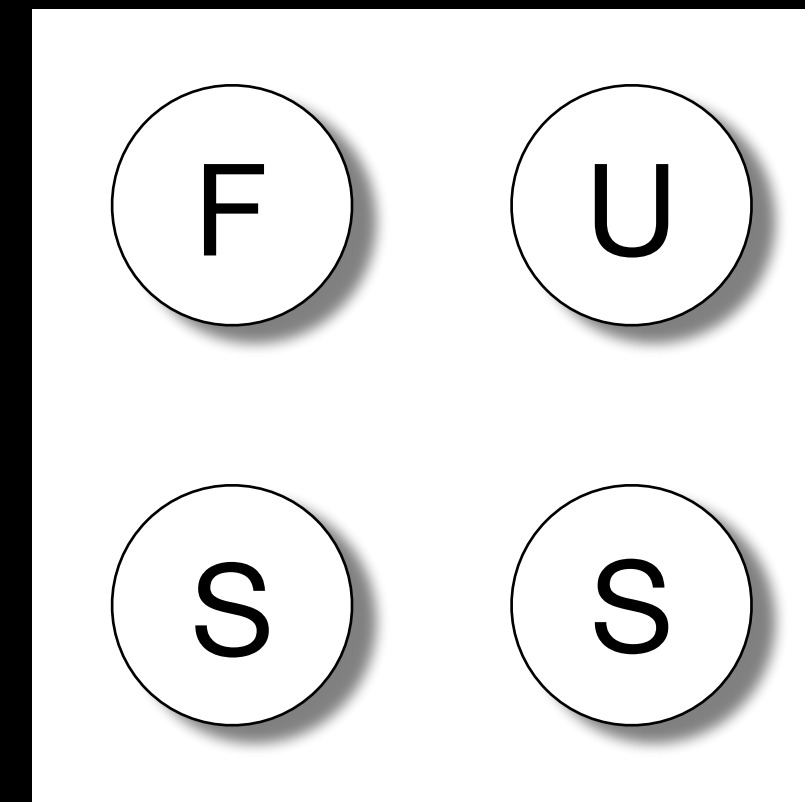
- This is the future of servers
- 3rd way to run VM's
- Easily run
 - Linux, Puppet, ELK, Nagios, OSTicket, Snort, etc
- Whalebrew
 - ansible, ffmpeg, gnupg, imagemagick, lynx, nmap, packer, wget
 - hub.docker.com/r/hartator/wayback-machine-downloader

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

Why Docker is Revolutionary

Solves the 4 Corners of FUSS

- Flexible
 - Lego-like WORA
- Upgradable
 - VIE Images
- Scalable
- Stable
- Bonus feature: nobody controls it



```
docker run --rm -it ubuntu ; cat /etc/os-release
```


This is a Major Transition Technology

My Prediction

- Host OS = kernel + Docker
- Apps will ship only as images
- Server and desktop
 - GUI apps w/ X11 forwarding
 - Fedora Silverblue
 - Catalina+, iOS, Android, Chrome OS



```
docker run --rm -it ubuntu ; cat /etc/os-release
```

More Information

- docker101tutorial
- <https://www.docker.com/get-started> & YouTube
- VMware's "Lightboards - Intro to Containers" on YouTube
- TechWorld with Nana on YouTube
- There's a yearly Dockercon, May 27th, 2021
- 1000's of resources
 - But check the date!
 - YouTube live streams and other community events

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

My Question To You Again

- Have you tried it yet?

```
docker run --rm -it ubuntu ; cat /etc/os-release
```

```
git clone https://github.com/docker/awesome-compose.git
```

```
cd awesome-compose/elasticsearch-logstash-kibana
```

```
docker-compose up -d
```