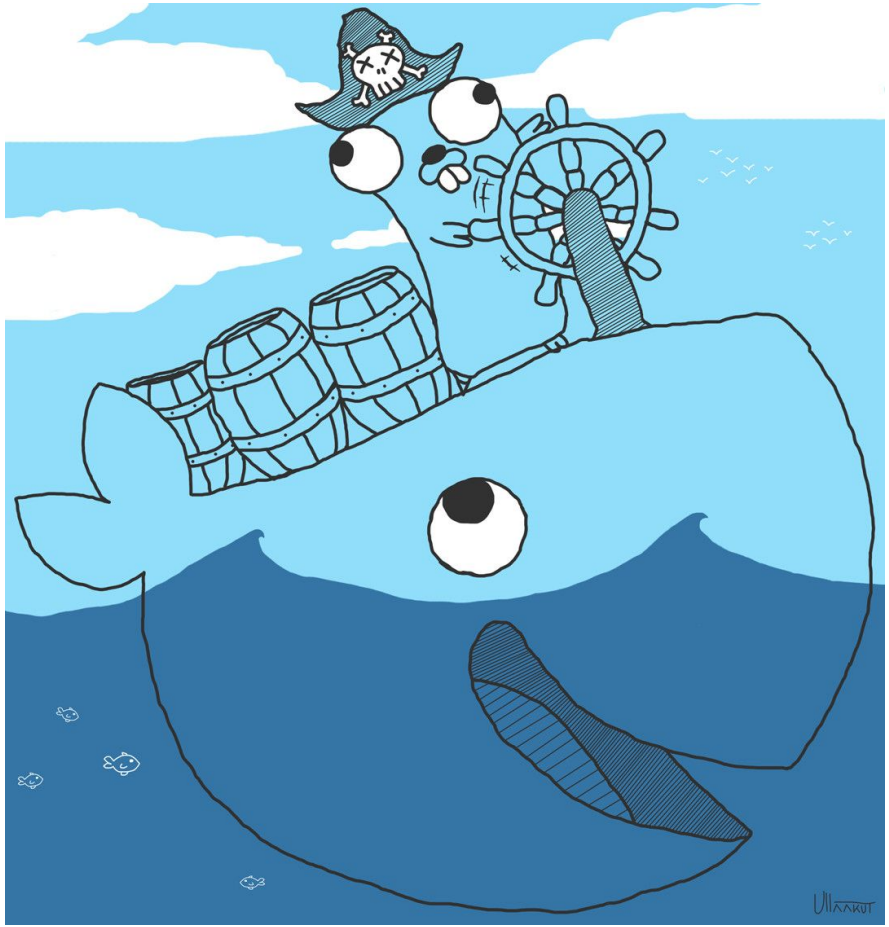
Docker Intro Workshop

Davy Jones

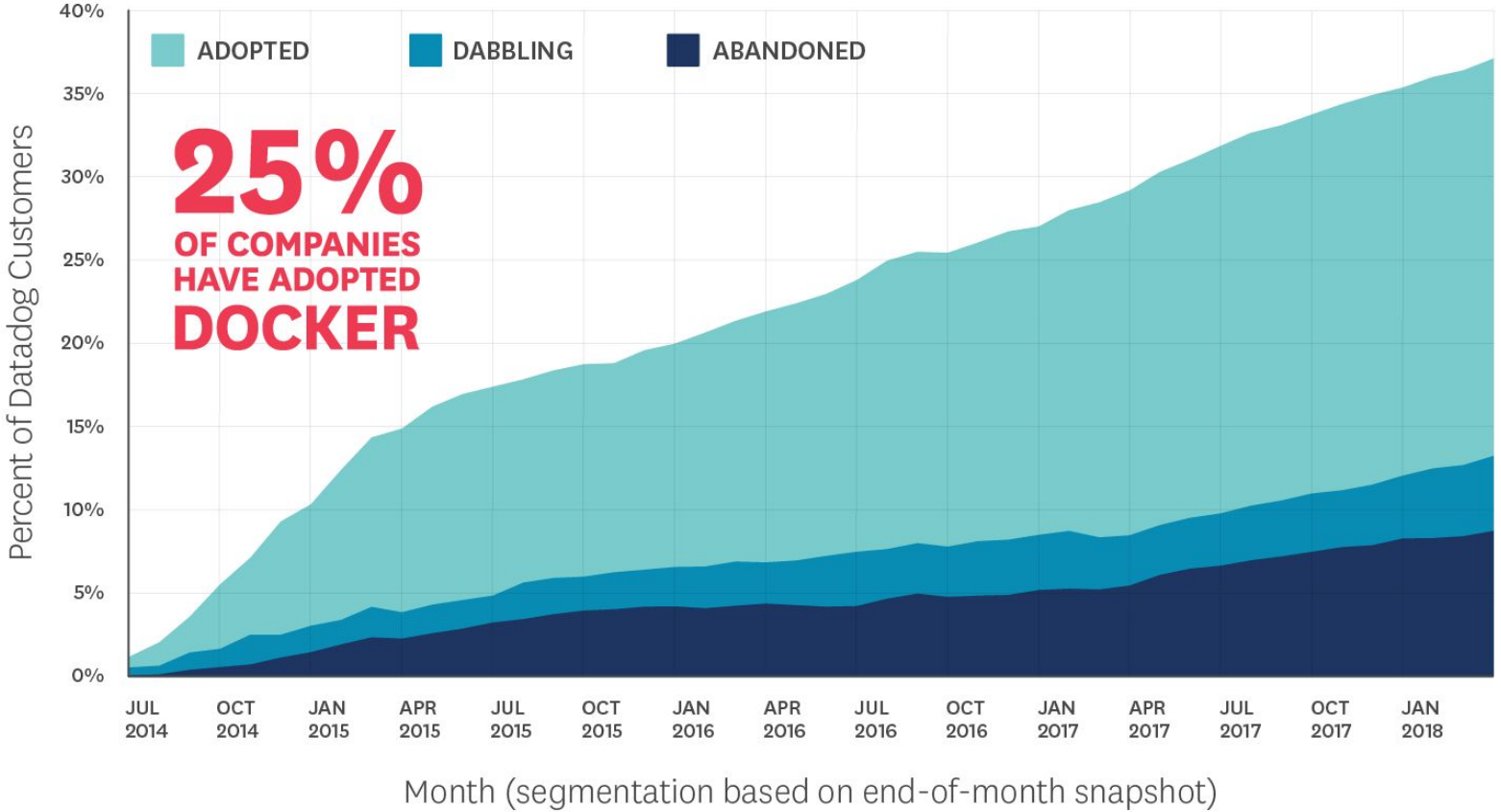


Who has used Docker before?

So, what actually is Docker?

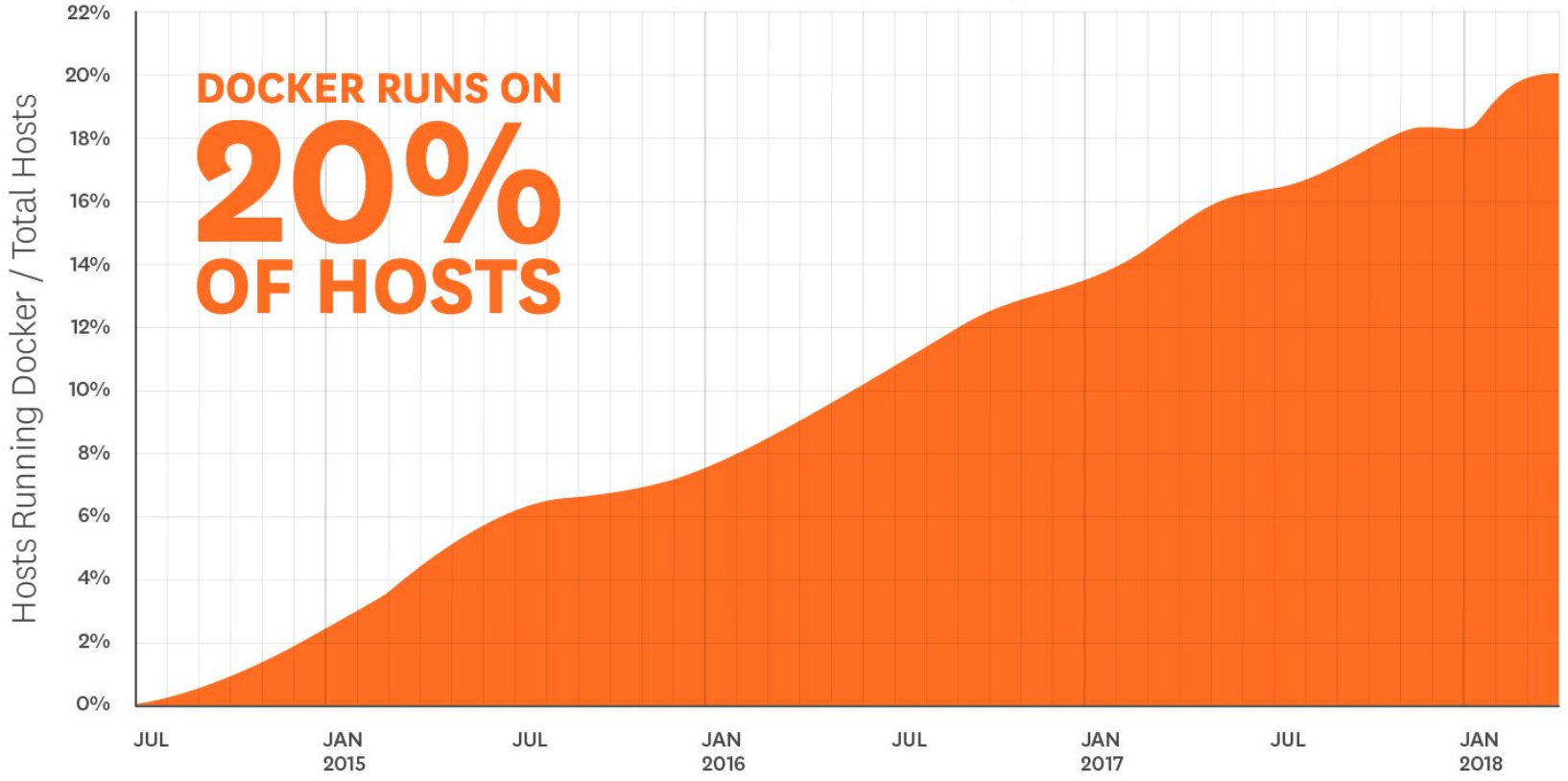
A way of packaging and running software

Docker Adoption Behavior



Source: Datadog

Portion of Hosts Running Docker



Source: Datadog

Docker does to apt-get what apt-get did to tar

- Bryan Cantrill

Wrapper and helpers to use Linux Containers

So then what are containers?

BSD Jails

Solaris Zones

LXC

They’re just a *process

*Albeit a very tightly controlled process through CGroups, Namespaces and Chroot

CGroups (Control Groups)

Means of controlling what **system resources** (think CPU, Memory) that a container **can access**

Namespaces

Ways of controlling what a container **can see** on your system and share.

So containers in the same network namespace can communicate over a network

chroot

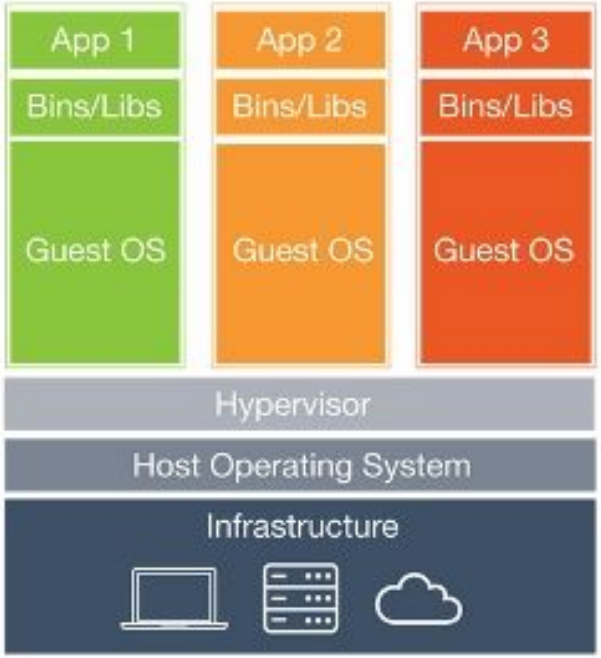
A means of providing a container within its own **root directory structure** with correct permissions

- **CGroups (Control Groups) -** Means of controlling what system resources (think CPU, Memory) that a container can have.
- **Namespaces -** Ways of controlling what a container can see on your system and share. So containers in the same network namespace can communicate over a network
- **chroot** - A means of providing a container within its own file-system directory structure with correct permissions

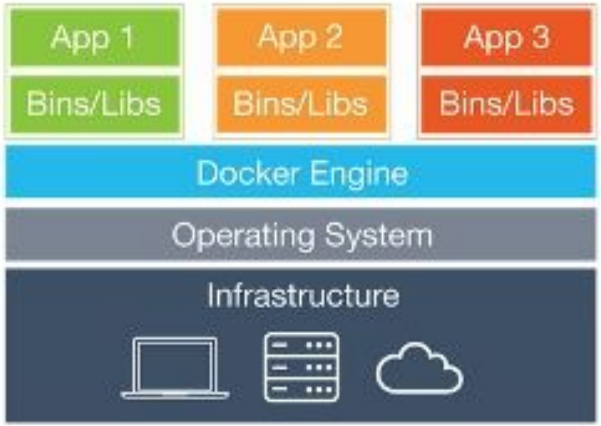
So they’re pretty much VMs then right?


5991


03/OCT
TO
22/NOV



Virtual Machines

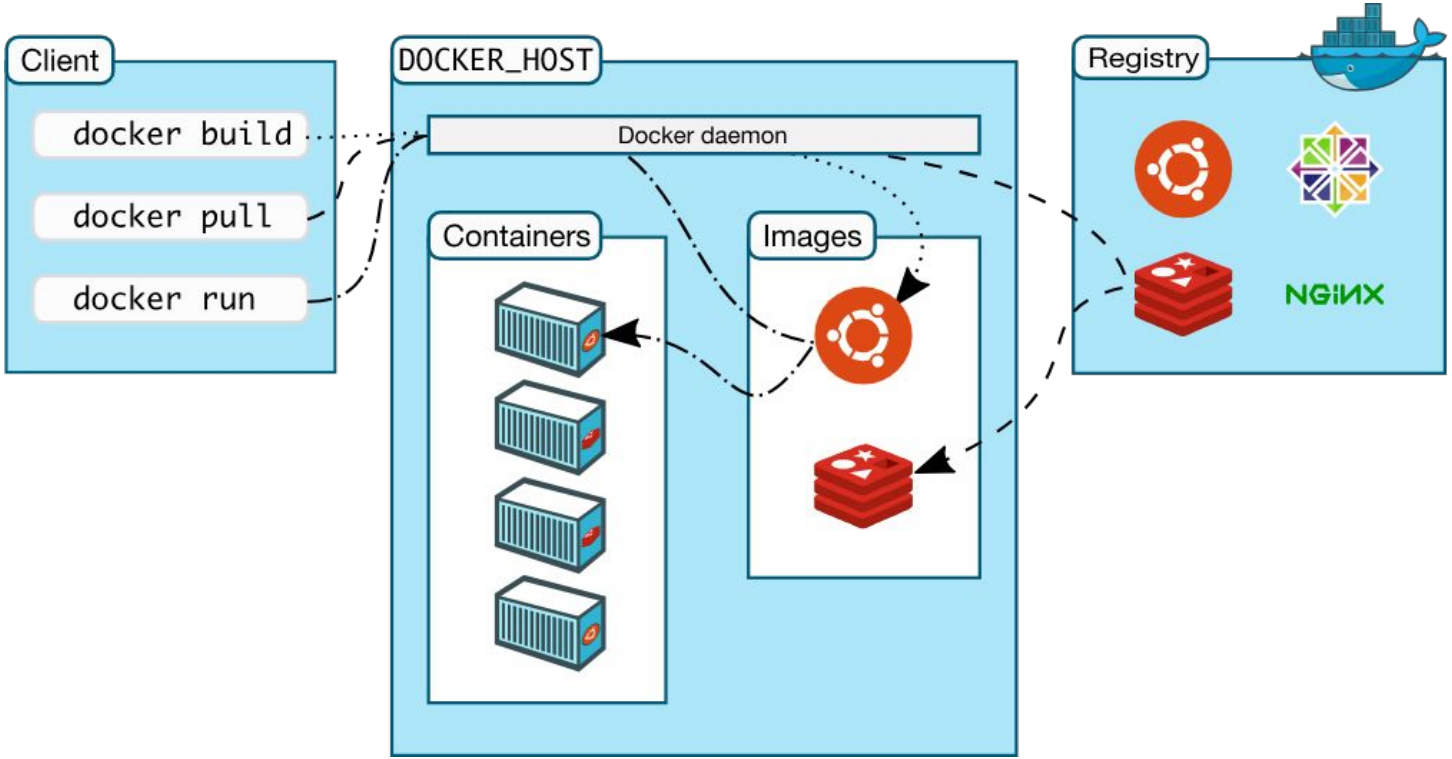


Containers

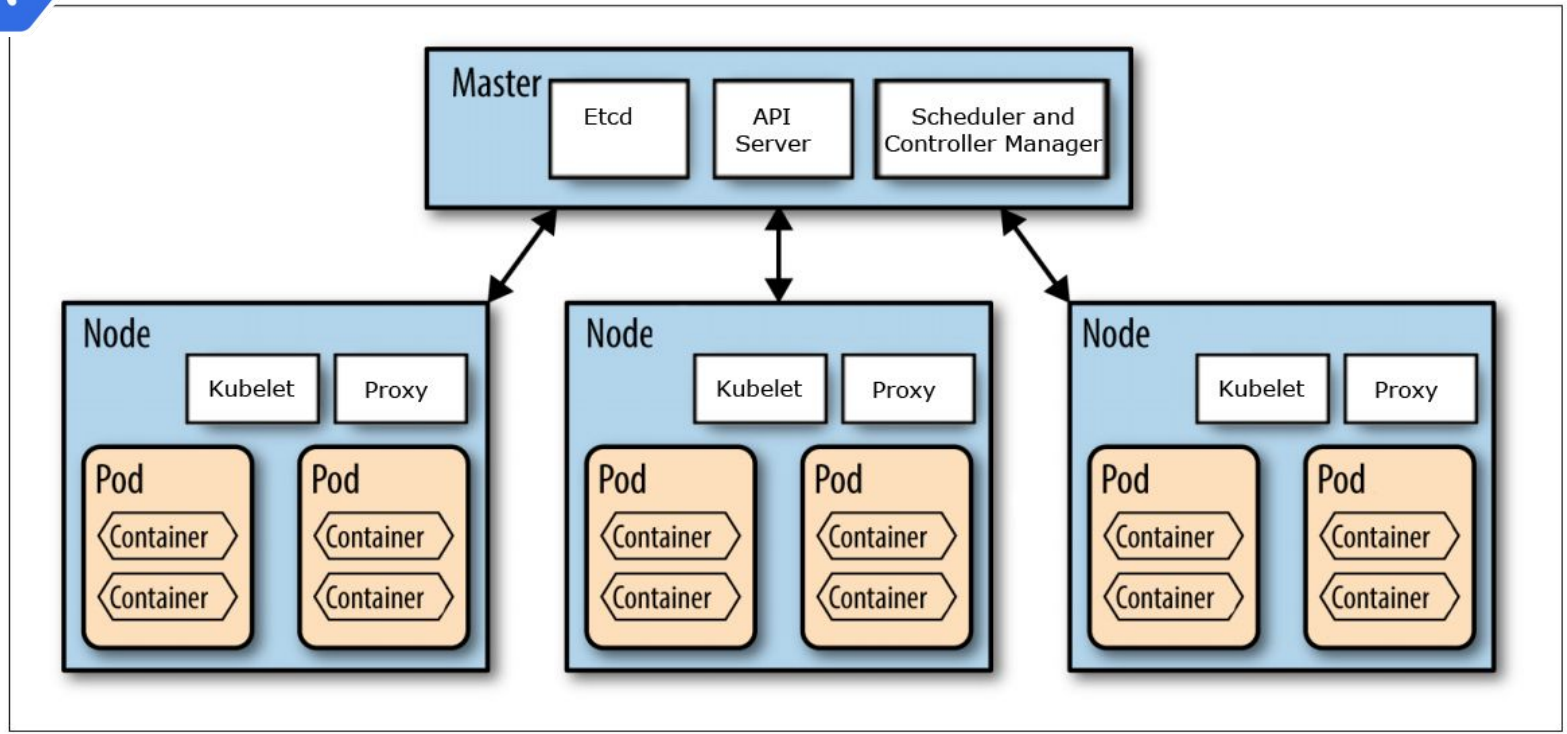
Components

- Docker Engine
  - Dockerd
  - runc
- Docker Client
  - OCI Runtime Spec
- Dockerfile
  - OCI Image Spec
- Docker Registry
- docker-compose

Components


Client
docker build
docker pull
docker run
DOCKER_HOST
Docker daemon
Containers
Images
Registry
NGINX

Kubernetes


Master
Etcd
API Server
Scheduler and Controller Manager
Node
Kubelet
Proxy
Pod
Container
Container
Pod
Container
Container
Node
Kubelet
Proxy
Pod
Container
Container
Pod
Container
Container
Node
Kubelet
Proxy
Pod
Container
Container
Pod
Container
Container

Hello World

```
$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Other OSes

```
$ docker run --rm -it ubuntu /bin/bash
root@77d822124a24:/# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.1 LTS"
```

Port Forwarding

```
$ docker run --rm -d -p 8080:80 nginx
$ curl -I http://localhost:8080
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Thu, 07 Feb 2019 17:23:07 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 25 Dec 2018 09:56:47 GMT
Connection: keep-alive
ETag: "5c21fedf-264"
Accept-Ranges: bytes
```

Swagger UI

```
$ docker run --rm \
-p 8765:8080 \
-e SWAGGER_JSON=/swagger.yaml \
-v $PWD/openapi.compiled.yml:/swagger.yaml
swaggerapi/swagger-ui
```

Workshops

https://www.katacoda.com/courses/docker