Docker

Docker for Developers
Claudius Link

http://bit.ly/1qHJmCb

Wer kennt Docker?

Wer weiss was Docker ist?

Wer nutzt Docker?

Wer entwickelt für Linux?

(bzw. etwa was irgendwann mal auf einem Linux Server läuft)

Wer arbeitet unter

* Linux?
* Mac OS X?
* Windows?

Warning

- Opinions expressed are my own
- Views of a developer
  (not a Cloud Operator)
- Not a long time Docker User

What is Docker?

- Lightweight “VM”
- Tooling
  - Creation, Distribution, Deployment
  - Versioning
- for Linux

Lightweight!?

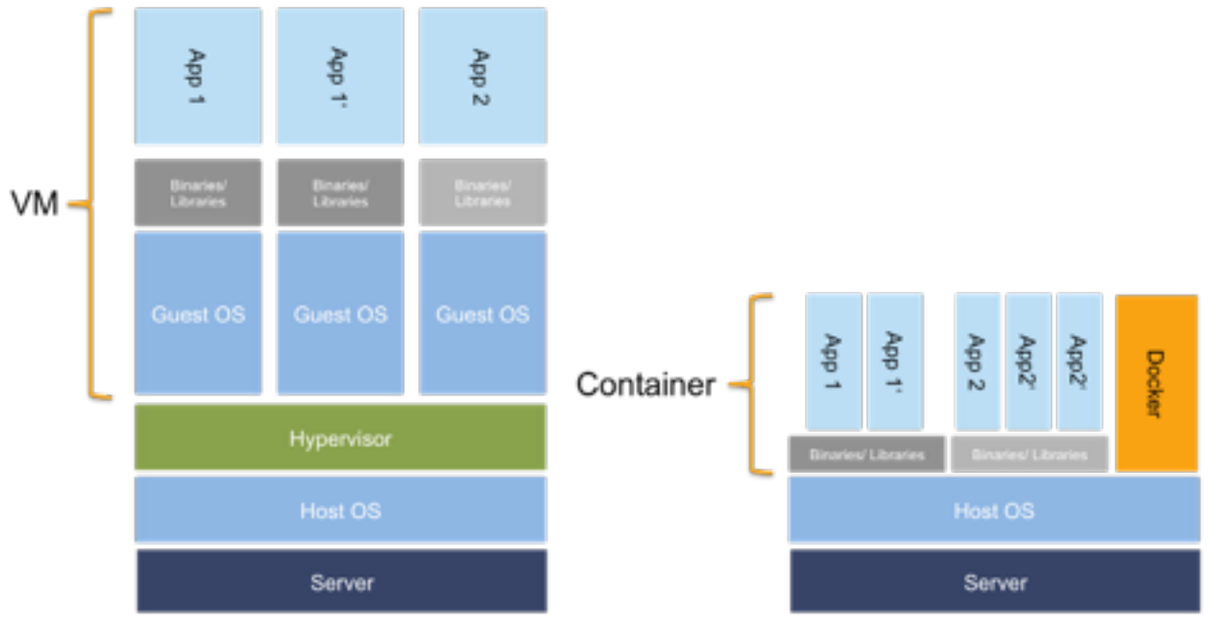
Native

```
$ time cat /etc/issue
Ubuntu 12.04.4 LTS (Precise Pangolin) \n \l
real  0m0.002s
user  0m0.001s
sys   0m0.002s
```

Docker Container

```
$ time docker run -i -t ubuntu:14.04 cat /etc/issue
Ubuntu 14.04.1 LTS \n \l
real  0m0.455s
user  0m0.017s
sys   0m0.005s
```

Virtualisierung: Virtuelle Maschinen vs. Docker-Container



Quelle: Docker, Crisp Research, 2014

Docker vs. VM

- No Init
- No Services
* Shared Kernel
+ Isolation:
    + Process & Memory
    + Network

Ubuntu 12.04
42 init.d start scripts on runlevel 2 (ls /etc/rc2.d/S* | wc -l)
14 running services (service --status-all 2>&1 | grep + | wc -l)
Isolation through cgroups/lx containers
Changeroot ond steroids

pid: process isolation (PID: Process ID).
net: managing network interfaces (NET: Networking).
ipc: managing access to IPC resources (IPC: InterProcess Communication).
mnt: managing mount-points (MNT: Mount).
uts: isolating kernel and version identifiers. (UTS: Unix Timesharing System).

Inside Docker

- Images
- Containers
- Layers

Inside Docker: Images

- **Images**
- Containers
- Layers

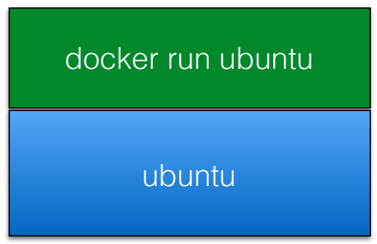

```
$ docker pull ubuntu:latest
$ docker images
REPOSITORY  TAG     IMAGE ID      CREATED     VIRTUAL SIZE
ubuntu      latest  826544226fdc  5 days ago  194.2 MB
…
```

Read-only templates

- Typically you get them from a repository
  (docker search, docker pull)
- or build them yourself and put them into an repository
  (docker import, docker push)
- or from a local cache
- Careful if you don’t specify a Tag ALL matching images are downloaded

Inside Docker: Containers

- Images
- **Containers**
- Layers



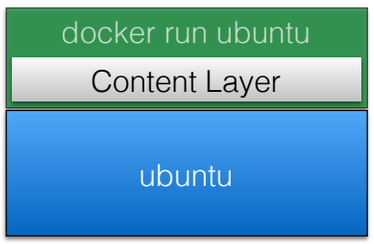

```
$ docker run -i -t ubuntu
$ docker ps
CONTAINER ID   IMAGE         COMMAND     CREATED         STATUS         PORTS  NAMES
589e9d33c919   ubuntu:latest "/bin/bash" 24 seconds ago  Up 24 seconds         thirsty_almeida
```

-i interactive

-t attach an terminal

Inside Docker: Layers

- Images
- Containers
- **Layers**

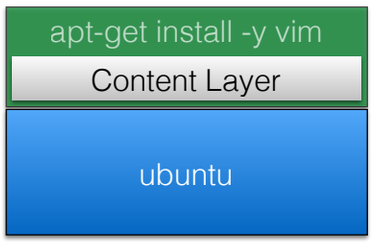



```
$ docker run -i -t ubuntu
$ docker ps
CONTAINER ID   IMAGE         COMMAND     CREATED  STATUS PORTS NAMES
589e9d33c919   ubuntu:latest "/bin/bash" 24s ago  Up 24s      thirsty_almeida
```

docker diff

Inside Docker:

- **Images** =
  Image + Layer




```
# apt-get update
# apt-get install vim

$ docker commit -p 589e9d33c919 ubuntu_vim
$ docker images | grep ubuntu_v
ubuntu_vim   latest    33e231eca143   2 minutes ago 237.7 MB
```

Inside Docker

- **Images** =
  Image + Layer

```
# Dockerfile
FROM ubuntu
RUN apt-get update -y
RUN apt-get install -y vim
```

```
$ docker build  -t ubuntu_vim .

$ docker images | grep ubuntu_v
ubuntu_vim  latest  2d7bea1fd284   4m ago  258 MB
```

To address caching “problems” --no-cache might be necessary when building

~~Turtles~~ Layers all the way down

http://commons.wikimedia.org/wiki/File:River_terrapin.jpg

Containers and the World

Filesystem

- Share files
- Folder
- Data Voluments

```
# Dockerfile
…
COPY listen.sh listen.sh
```

```
$ docker run -i -t -v $PWD/listen.sh:/listen2.sh ubuntu_vim
$ docker run -i -t -v /tmp:/mnt/tmp ubuntu_vim

$ docker run -d -v /data --name data ubuntu_vim echo "Data"
$ docker run -i -t --volumes-from data  ubuntu_vim
```

docker run -i -t --volumes-from data ubuntu_vim

ls /data

touch /data/fobar

docker run -i -t --volumes-from data ubuntu_vim

ls /data

Networking

- Port forwarding
- Linking

```
$ docker run -p 5001:5001 -i -t ubuntu_vim
# ./listen.sh 5001

$ telnet 192.168.59.103  5001
```

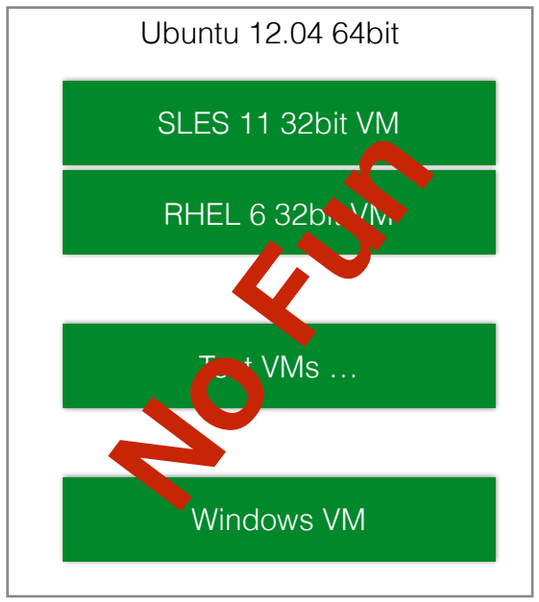
Use Cases

http://i.imgur.com/3CS31.jpg

Ship development machines

Multi-platform Development

- Workstation: Ubuntu 12.04 64bit
- Target 1: SLES 11 32bit
- Target 2 RHEL 6 32bit
- …



> 2 VMs are no fun

Version Management

Replace

- rvm
- npm
- CLASSPATH
- ….

Boot2Docker

(Windows & Mac OS X)

```
# Create VM
$ boot2docker init
# Start VM
$ boot2docker start
# Get the endpoint for the client
$ boot2docker socket
# Export Endpoint
$ export DOCKER_HOST=tcp://<IP>:<PORT>

# Happy Dockering
$ docker run -t -i ubuntu
```

On windows some ssh tools are required

Users

Docker Users

Security

More information

- https://www.docker.com
- Slides https://github.com/linkclau/docker-doc