



Introduction to Docker

Module: Project&Job

Agenda



- What is Docker?
 - Docker vs. Virtual Machine
 - History, Status, Run Platforms
 - Hello World
- Images and Containers
- Volume Mounting, Port Publishing, Linking
- Around Docker, Docker Use Cases
- Hands-On Workshop



What is Docker?

Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating system–level virtualization on Linux.

[Source: en.wikipedia.org]

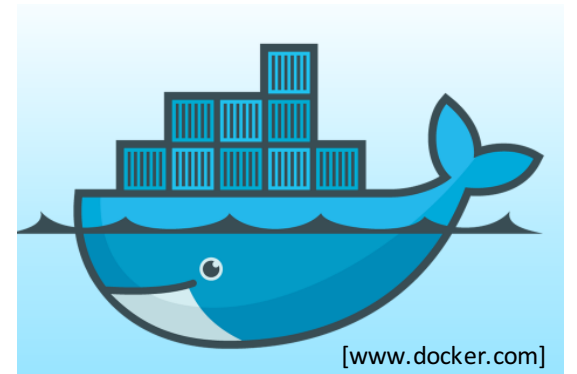
Docker: Name



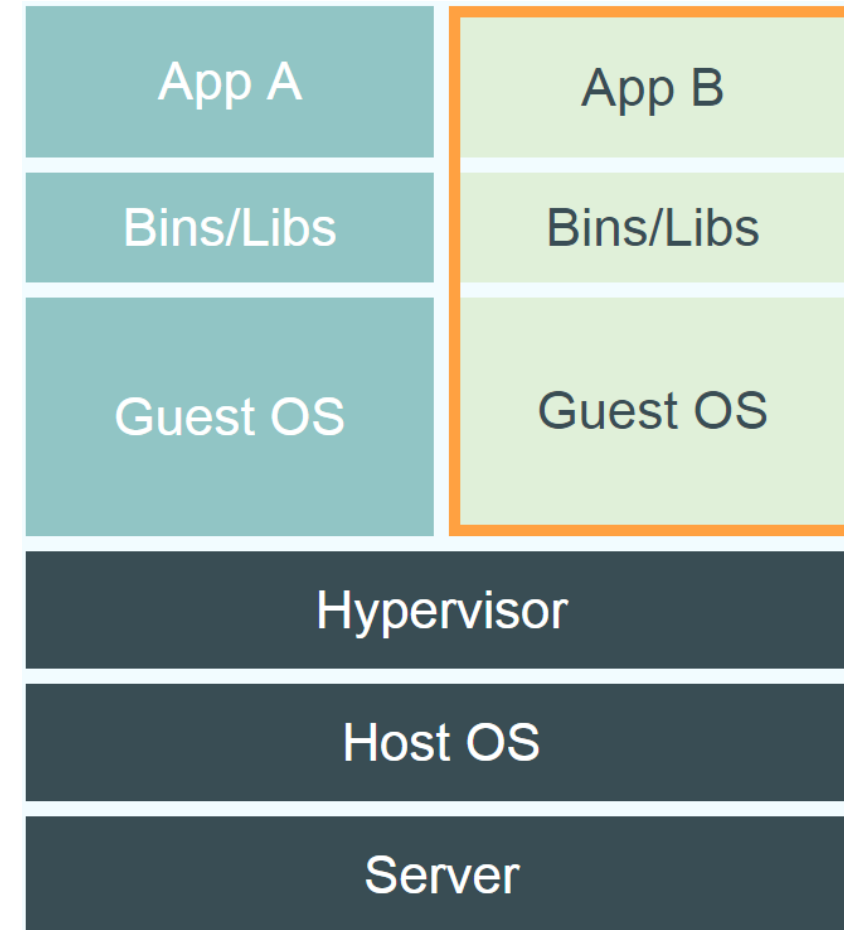
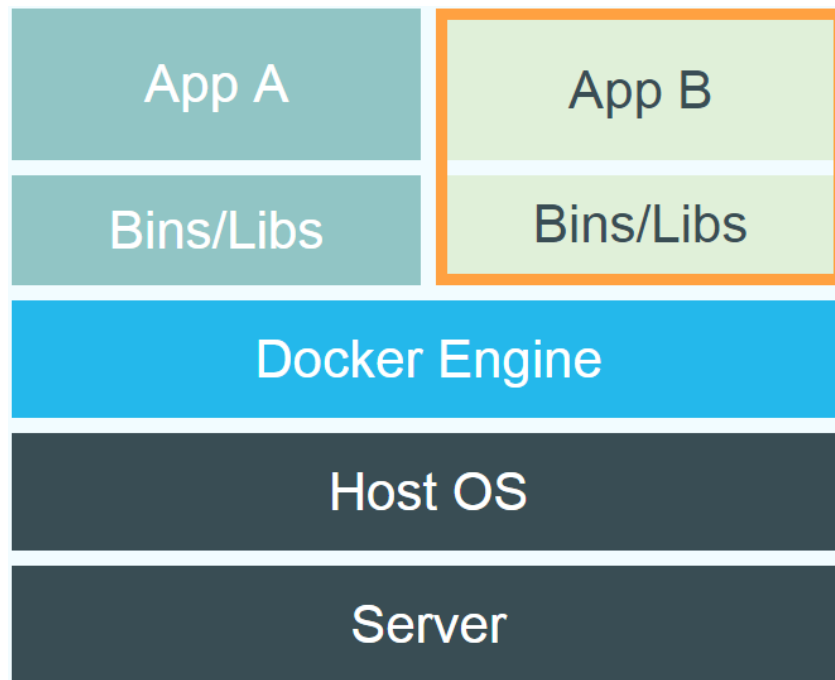
docker [naut.]: der Dockarbeiter, der Hafenarbeiter

Source: leo.org

- Provide a uniformed wrapper around a software package: «*Build, Ship and Run Any App, Anywhere*»
[www.docker.com]
 - Similar to shipping containers: The container is always the same, regardless of the contents and thus fits on all trucks, cranes, ships, ...



Docker vs. Virtual Machine



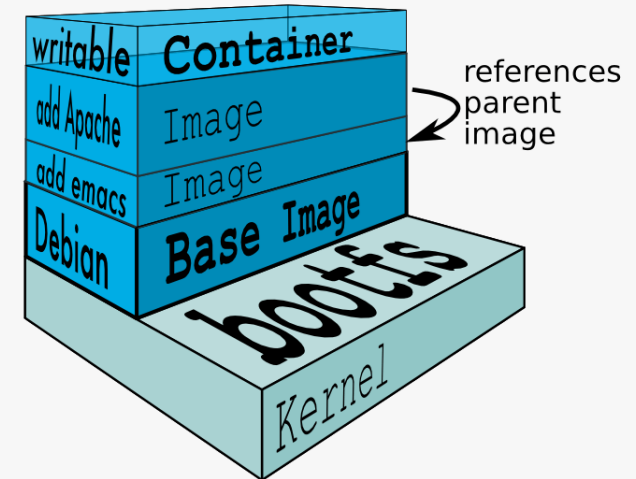
Source: <https://www.docker.com/whatisdocker/>



Docker Technology



- libvirt: Platform Virtualization
- LXC (Linux Containers): Multiple isolated Linux systems (containers) on a single host
- Layered File System



[Source: <https://docs.docker.com/terms/layer/>]



Docker History



- 2013-03: Releases as Open Source
- 2013-09: Red Hat collaboration (Fedora, RHEL, OpenShift)
- 2014-03: 34th most starred GitHub project
- 2014-05: JAX Innovation Award (most innovative open technology)



Technology Radar



- 2014-01: Assess
- 2014-07: Trial
- Source: <http://www.thoughtworks.com/radar/tools/docker>



Run Platforms

- Various Linux distributions (Ubuntu, Fedora, RHEL, Centos, openSUSE, ...)
- Cloud (Amazon EC2, Google Compute Engine, Rackspace)
- 2014-10: Microsoft announces plans to integrate Docker with next release of Windows Server

Hello World



Simple Command - Ad-Hoc Container

- `docker run ubuntu echo Hello World`
 - `docker images [-a]`
 - `docker ps -a`



Terminology - Image

- Persisted snapshot that can be run
 - *images*: List all local images
 - *run*: Create a container from an image and execute a command in it
 - *tag*: Tag an image
 - *pull*: Download image from repository
 - *rmi*: Delete a local image
 - This will also remove intermediate images if no longer used

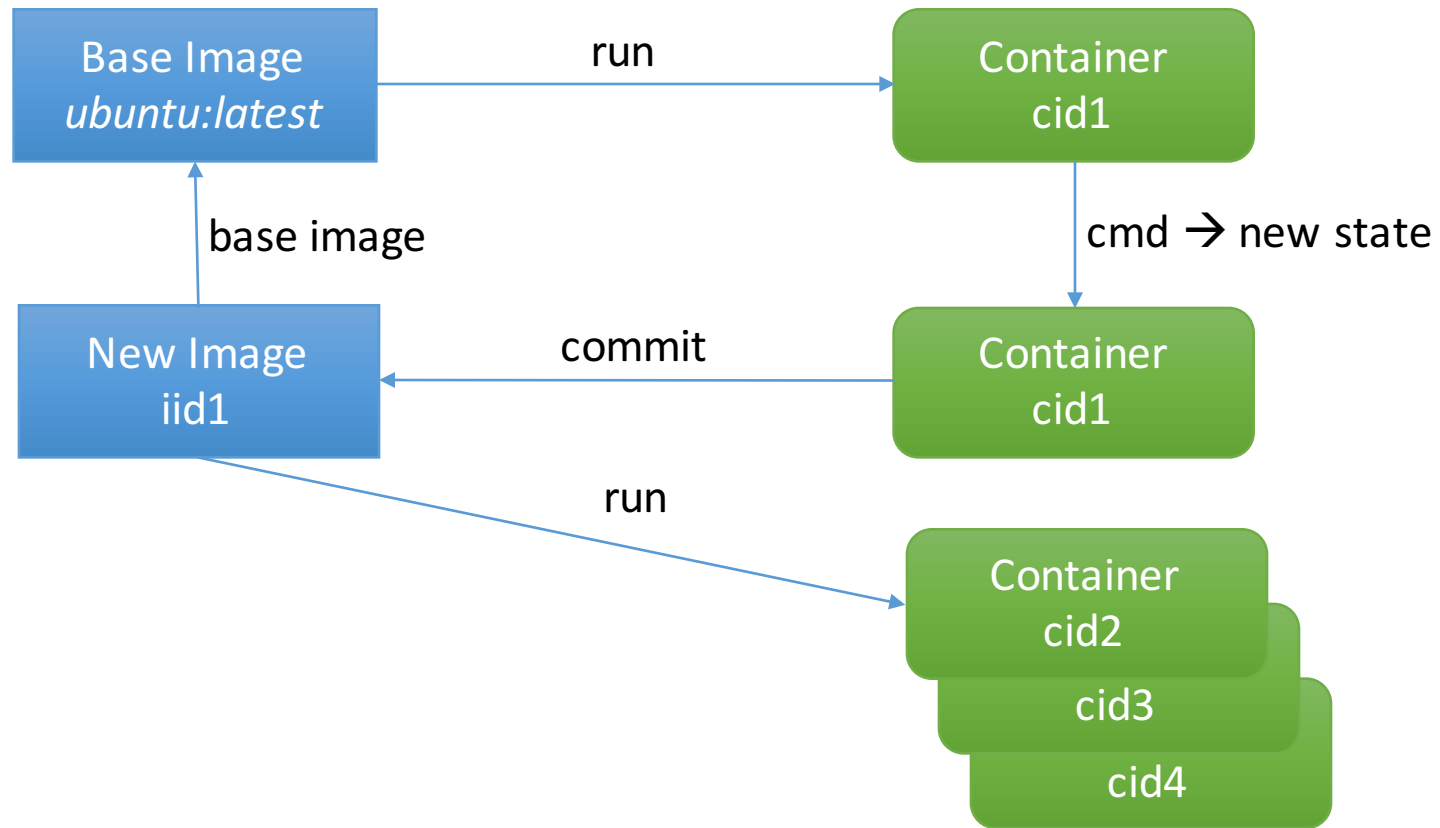
Terminology - Container



- Runnable instance of an image
 - *ps*: List all running containers
 - *ps -a*: List all containers (incl. stopped)
 - *top*: Display processes of a container
 - *start*: Start a stopped container
 - *stop*: Stop a running container
 - *pause*: Pause all processes within a container
 - *rm*: Delete a container
 - *commit*: Create an image from a container



Image vs. Container



Dockerfile



- Create images automatically using a build script: «Dockerfile»
- Can be versioned in a version control system like Git or SVN, along with all dependencies
- Docker Hub can automatically build images based on dockerfiles on Github



Dockerfile Example

- Dockerfile:
 - FROM ubuntu
 - ENV DOCK_MESSAGE Hello My World
 - ADD dir /files
 - CMD ["bash", "someScript"]
- docker build [DockerFileDir]
- docker inspect [imageId]



Mount Volumes

- `docker run -ti -v /hostLog:/log ubuntu`
- Run second container: Volume can be shared
 - `docker run -ti --volumes-from firstContainerName ubuntu`



Publish Port

- `docker run -t -p 8080:80 ubuntu nc -l 80`
 - Map container port 80 to host port 8080
 - Check on host: `nc localhost 8080`
- Link with other docker container
 - `docker run -ti --link containerName:alias ubuntu`
 - See link info with `set`

Around Docker

- Docker Images: Docker Hub
- Vagrant: «Docker for VMs»
- Automated Setup
 - Puppet, Chef, Ansible, ...
- Docker Ecosystem
 - skydock / skydns
 - fig

Docker Hub



- Public repository of Docker images
 - <https://hub.docker.com/>
 - docker search [term]
- Automated: Has been automatically built from Dockerfile
 - Source for build is available on GitHub



Resource Usage

- top / ps / free -m
- Start 100 WebServer containers
 - docker run -d -p \$hostPort:5000 -e "PROVIDER=\$provider" training/webapp
- docker ps [containerId]
- top / ps / free -m

Docker Use Cases

- Development Environment
- Environments for Integration Tests
- Quick evaluation of software
- Microservices
- Multi-Tenancy
- Unified execution environment (dev → test → prod (local, VM, cloud, ...))

Documentation

- Docker homepage: <https://www.docker.com/>
 - Introduction: <https://www.docker.com/whatisdocker/>
 - Online tutorial: <https://www.docker.com/tryit/>
 - Installation and user guide: <https://docs.docker.com/>
- InfTec TecBoard:
<https://inftec.atlassian.net/wiki/display/TEC/Docker>
 - Includes this presentation

Hands On



- <https://bitbucket.org/inftec/vagrant-playground/branch/docker-demo>
- Multi-Container-Setup
 - Logging-Container
 - Echo-Container
 - Client-Container

