



# Microservices mit ASP.NET Core 1.0 und dem Docker Ökosystem

# Alexander Zeitler

Solution Architect und Entwickler bei PDMLab

Microsoft MVP für ASP.NET

ASP.NET Web API / Microsoft Azure Advisor

<http://alexanderzeitler.com>

[alexander.zeitler@pdmlab.com](mailto:alexander.zeitler@pdmlab.com)

[@alexzeitler\\_](#)

# Agenda

- Microservices
- Herausforderungen
- Wie hilft Docker?
- ASP.NET Core mit Docker deployen

# Microservices

In computing, microservices is a software architecture style, in which complex applications are composed of small, independent processes communicating with each other using language-agnostic APIs.

These services are small, highly decoupled and focus on doing a small task.

<http://en.wikipedia.org/wiki/Microservices>

# Beispiel: E-Commerce

- Web Front-End
- Product Data Management (PDM)
- Lager
- Einkauf
- Versand
- Payment-Dienstleister
- Rechnungs- und Mahnwesen
- Identity

# Vorteile

- Verschiedene Plattformen für verschiedene Services
- Austausch / Refactoring von Services einfacher
- Entkopplung beim Deployment
- "Two-Pizza"-Regel umsetzbar
- (Code) Ownership einfacher

**Herausforderung:  
Komplexität**



# Komplexität

- Significant Operations Overhead
- Substantial DevOps Skills Required
- Implicit Interfaces
- Duplication Of Effort
- Distributed System Complexity
- Asynchronicity Is Difficult!
- Testability Challenges

<http://highscalability.com/blog/2014/4/8/microservices-not-a-free-lunch.html>

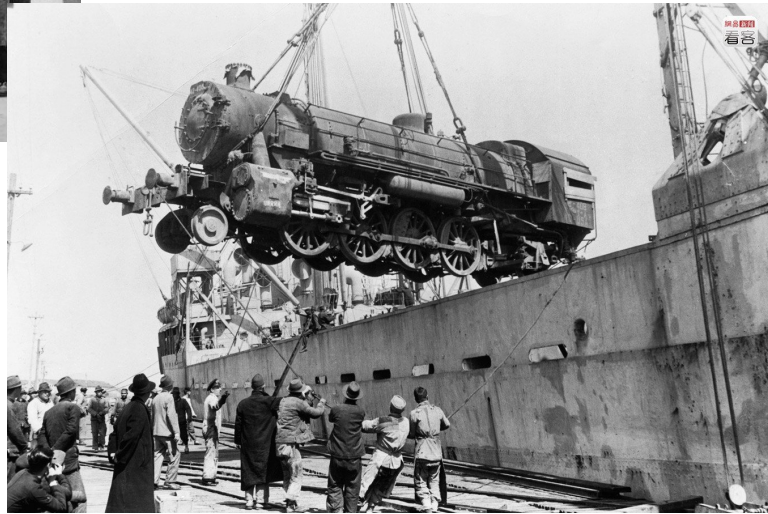
# Anforderungen an Deployment

- Schnell
- Zuverlässig
- Reproduzierbar

**Deployment  
automatisieren!**

**Wie hilft Docker?**

**Build, ship and run any  
application anywhere**



[http://upload.wikimedia.org/wikipedia/commons/c/c5/Britain\\_Delivers\\_the\\_Goods\\_in\\_Wartime-Dock\\_Workers\\_in\\_Bristol,\\_England,\\_194](http://upload.wikimedia.org/wikipedia/commons/c/c5/Britain_Delivers_the_Goods_in_Wartime-Dock_Workers_in_Bristol,_England,_194)

<http://i1.wp.com/www.chinasmack.com/wp-content/uploads/2013/09/historical-shanghai-photos-early-20th-century-14.jpg>

**VS...**



<https://homefaraway.files.wordpress.com/2012/04/april-24-2012-026.jpg>



**Put it in a Container**



docker

# Build

Package application in a container

# Ship

Move container to another machine

# Run

Execute that container

# **Any Application**

**All applications running on Linux**

# Anywhere

VM on-premises, cloud, bare metal

**Build**



# Dockerfile

```
FROM ubuntu:14.04
MAINTAINER Docker Team <education@docker.com>
RUN apt-get update
RUN apt-get install -y nginx
RUN echo 'Hi, I am your father...erm in your container' \
>/usr/share/nginx/html/index.html
CMD [ "nginx", "-g", "daemon off;" ]
EXPOSE 80
```

**Ship**

# Docker Hub

Image name <username>/<reponame>

z.B. microsoft/dotnet

docker push

docker pull

<https://hub.docker.com/>

**Run**

- Leichtgewichtig
- Schnell

**Wie schnell?**

# Benchmark

```
$ time docker run ubuntu echo hello world
```

# Docker Footprint

```
$ docker images
```

```
$ docker history <CID>
```



**Any application**

- Web apps
- API backends
- Datenbanken (SQL, NoSQL)
- Big data
- Message queues
- ...

**Runs on Linux == Runs in Docker**

# Windows Container?

Nano Server

**Getting started**

# Try

<http://www.docker.com/tryit/>

# Install

## On OS X / Windows

<https://www.docker.com/products/docker-toolbox>

<https://beta.docker.com/>

## On Linux

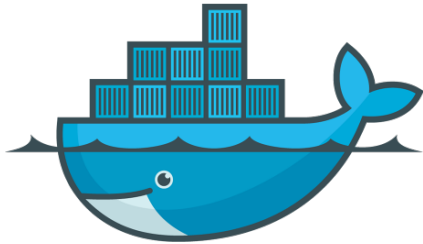
```
curl -fsSL https://get.docker.com/ | sh
```

# ASP.NET / .NET Core in Docker

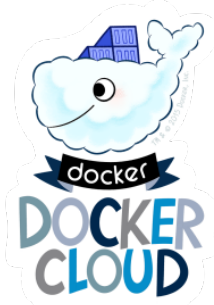
microsoft/aspnet (< RC2)

microsoft/dotnet (>= RC2)

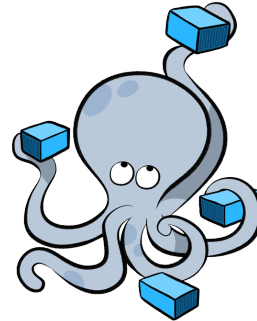
# Docker ecosystem



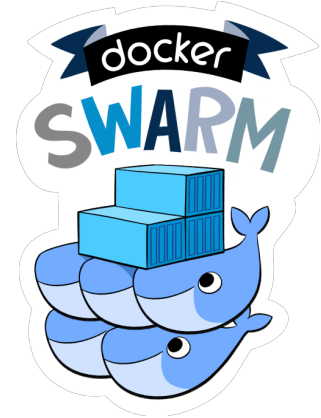
Docker engine



Docker Cloud



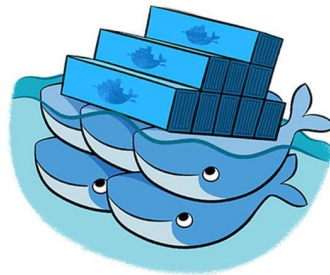
Docker Compose



Docker Swarm



Docker Machine



Docker Hub



Docker Registry



# Next Steps

- Docker Machine
- Docker Compose
- Docker Swarm

# Docker Machine

```
$ docker-machine create --driver virtualbox dev
```

# Docker Compose

```
version: "2"

services:
  web:
    build: .
    links:
      - db
    ports:
      - "8000:8000"
  db:
    image: postgres
```

# **Demo**

## **ASP.NET Core 1.0 RC2**

### **in Docker**

**Fragen?**

**Danke!**