

ASP.NET Core mit VS 2017 und Docker

Marc Müller
Principal Consultant



marc.mueller@4tecture.ch
[@muellermarc](https://twitter.com/muellermarc)
www.4tecture.ch

DIGICOMP





Marc Müller
Principal Consultant
@muellermarc



4tecture[®]
empower your software solutions

Agenda

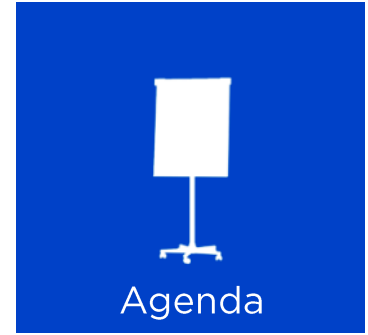
Intro

Container für Web Applikationen

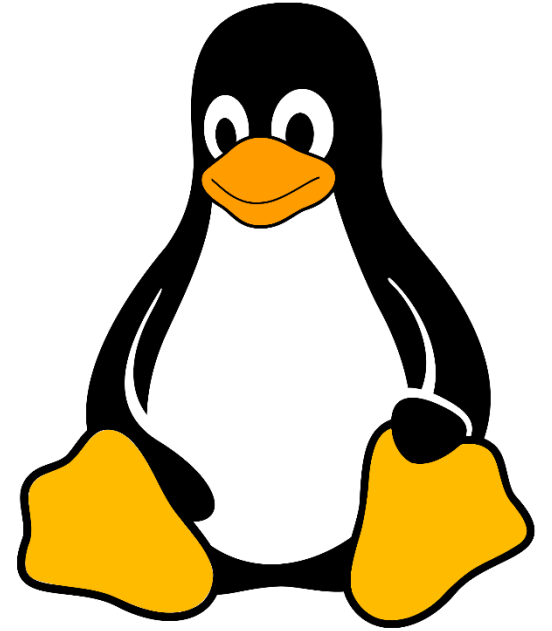
Docker Support in Visual Studio

Docker CI Build

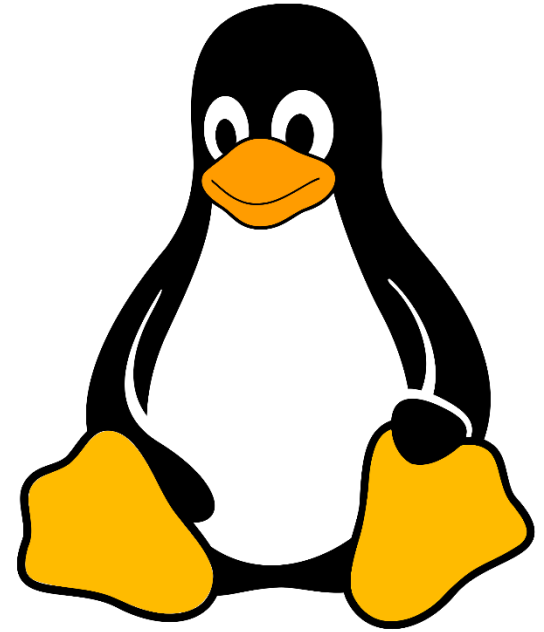
Docker Hosting



Microsoft

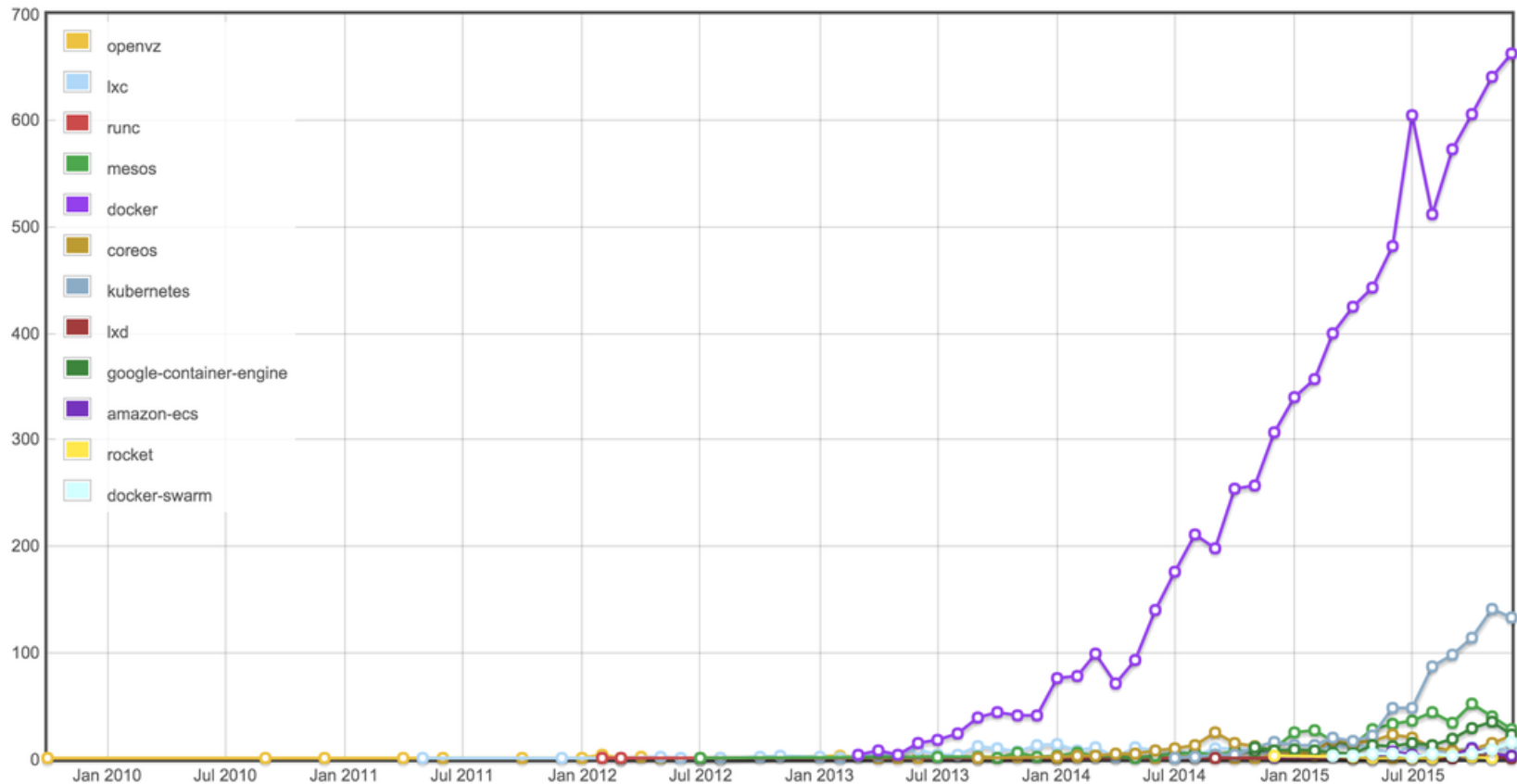


I like



...but I'm a Microsoft Fan-Boy

Docker?



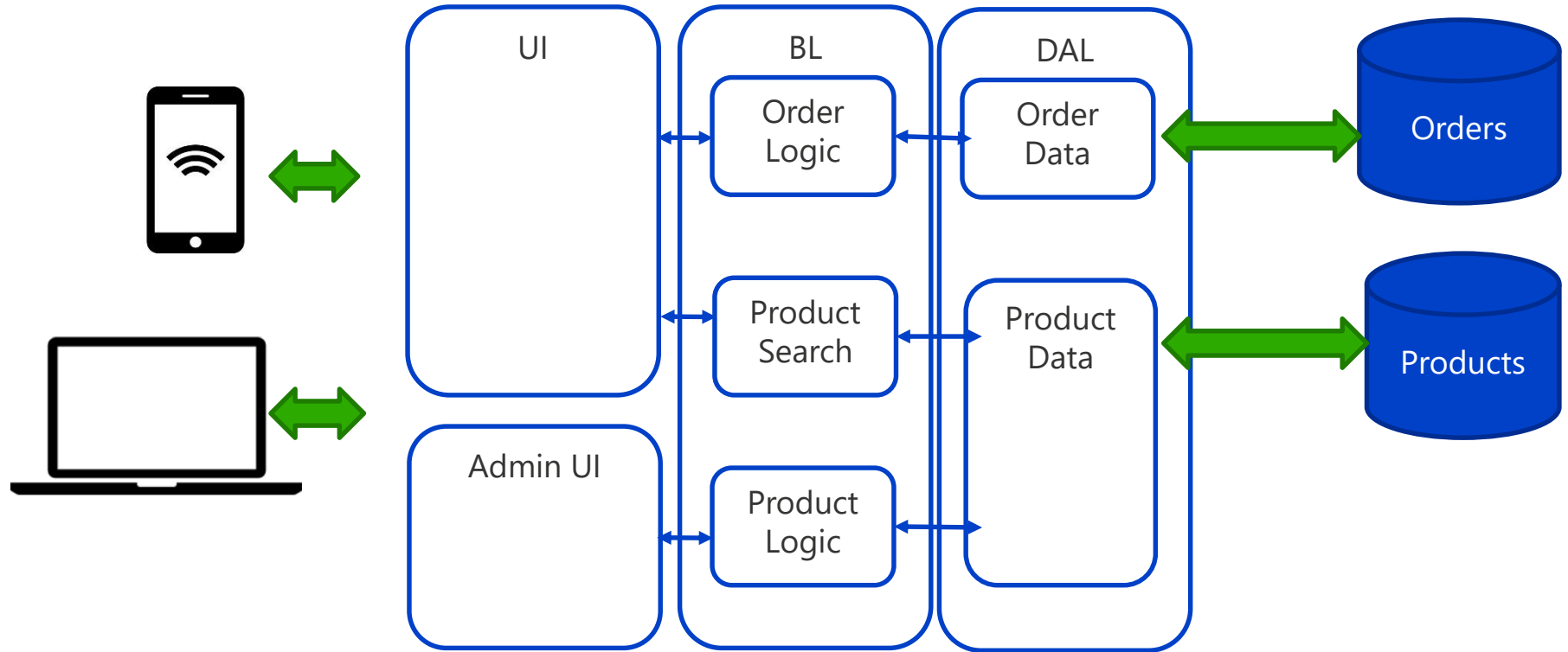
<https://twitter.com/dberkholz/status/689211852157407233>



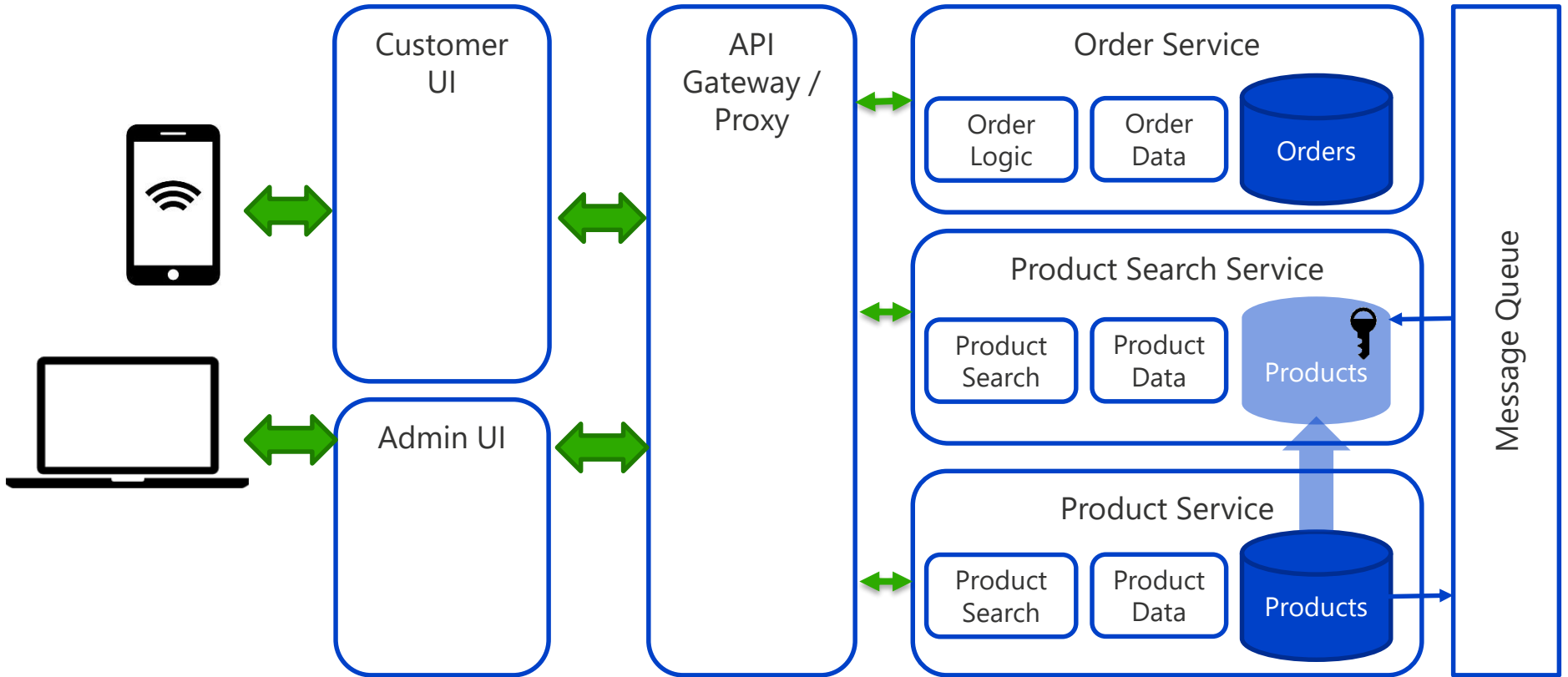
100 Deployments
pro Tag!



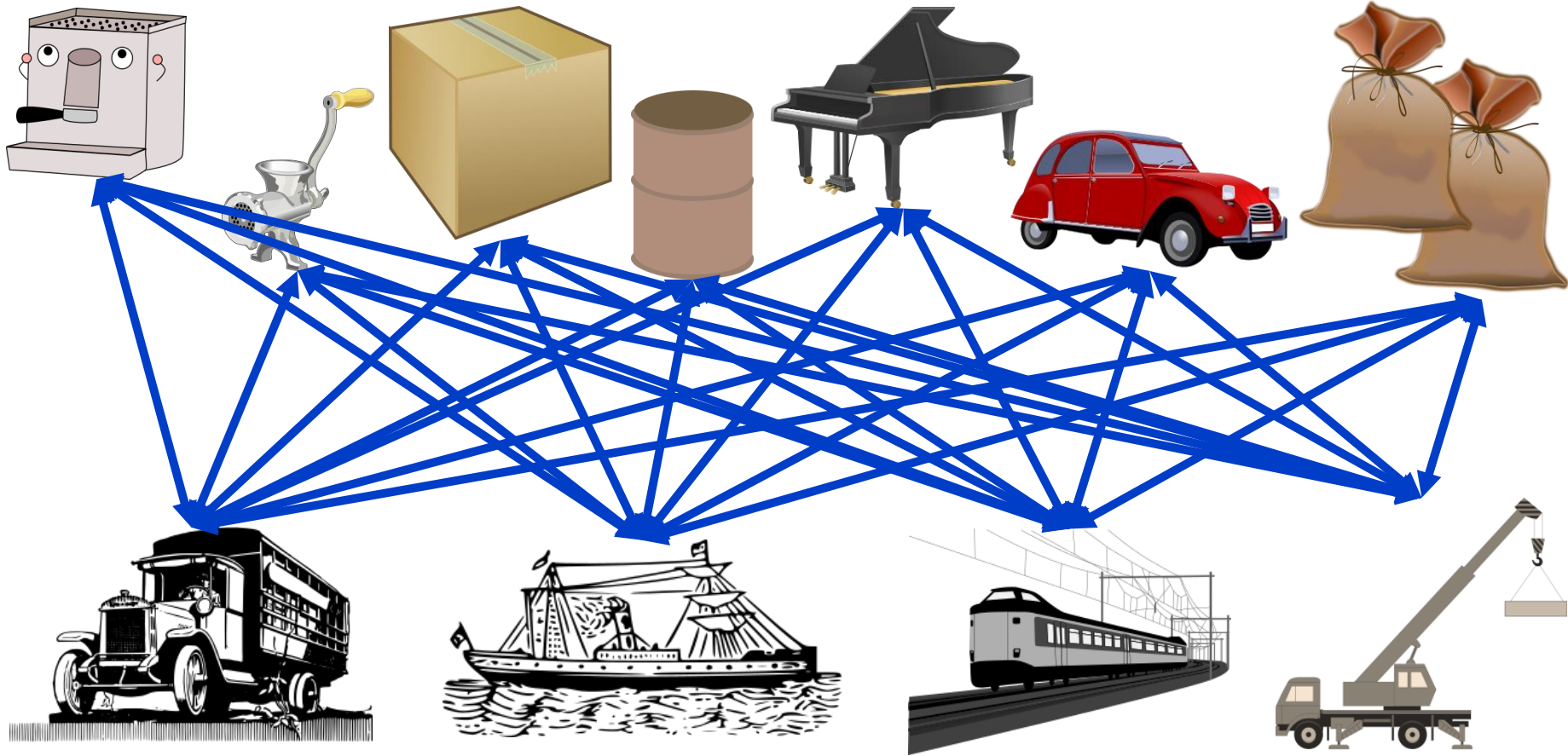
Monolithische Architektur



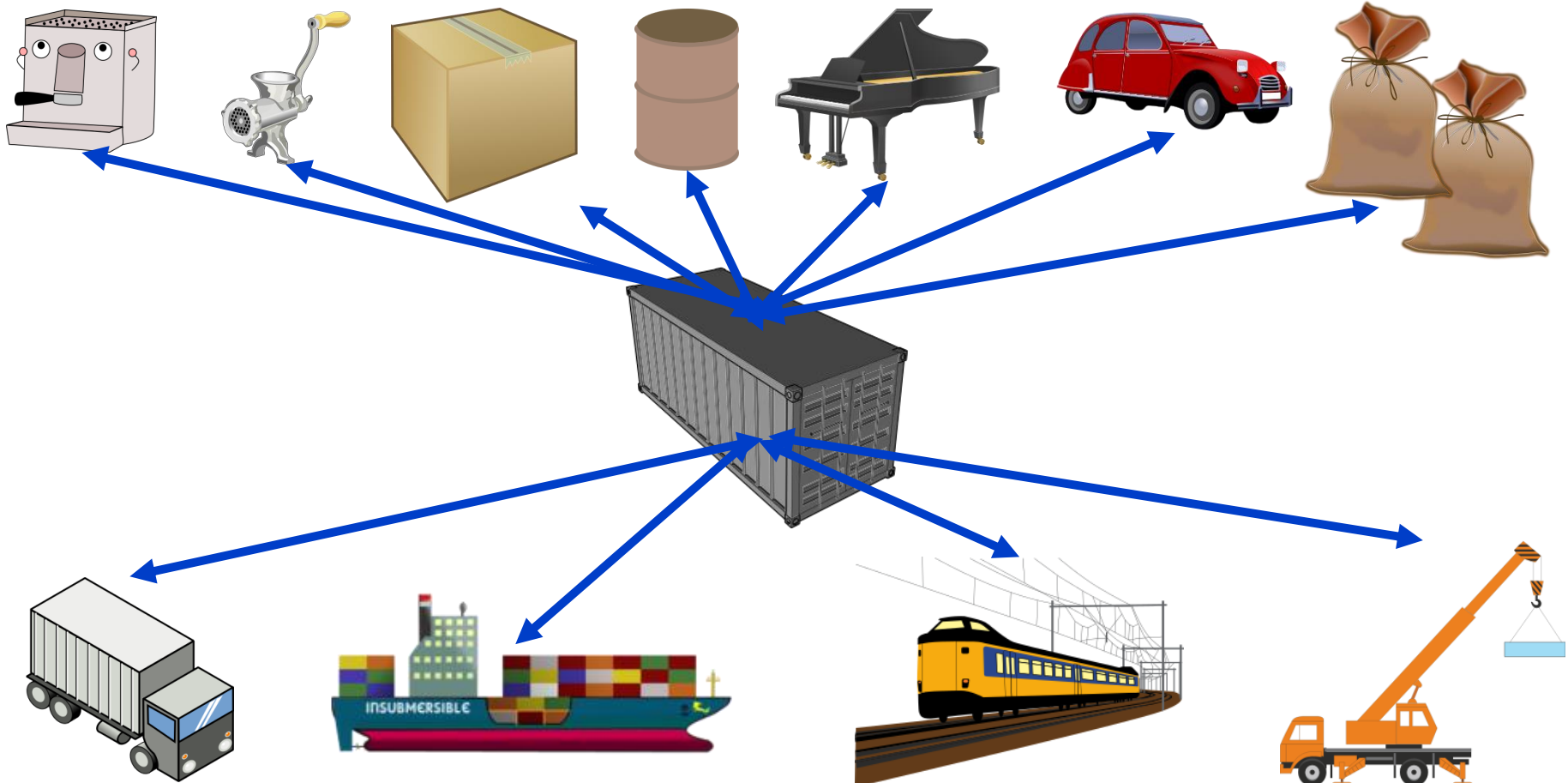
Micro-Service Architektur



Güterverkehr vor 1960



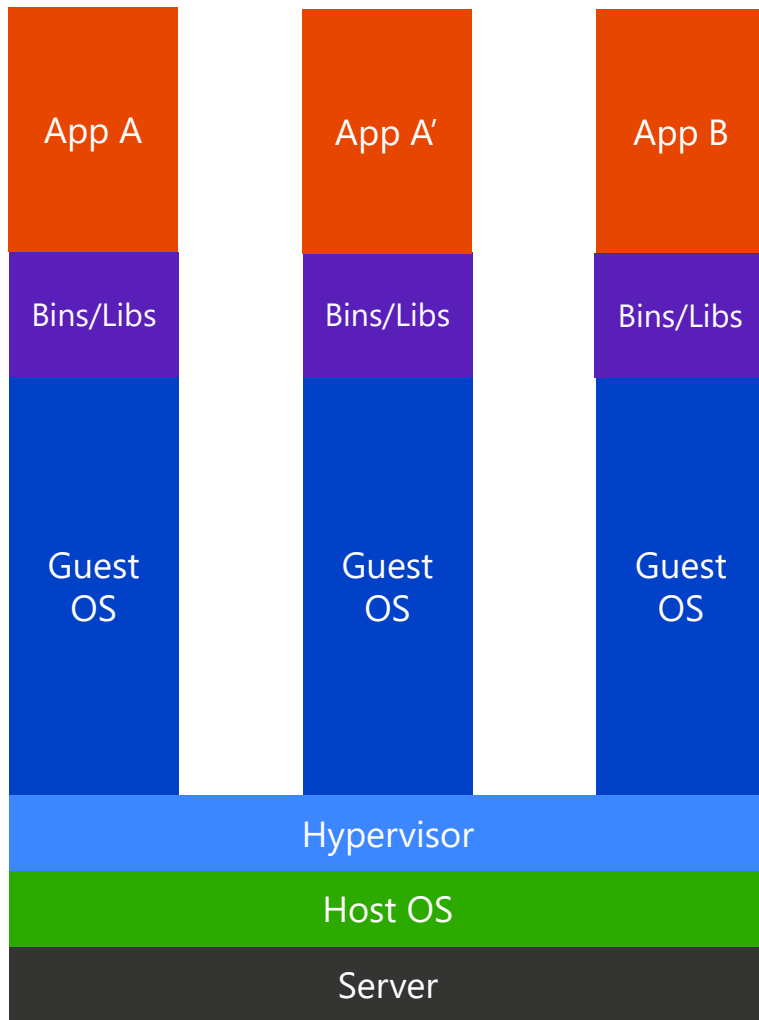
Lösung: Frachtcontainer



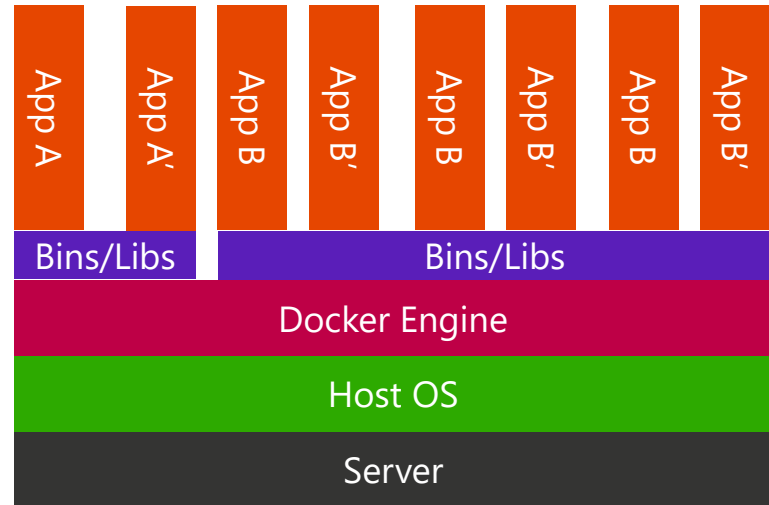
A background image showing a rowing team in a boat. The rowers are wearing blue and red uniforms and are captured in a synchronized rowing motion. The focus is on the oars and the rowers' hands, with a semi-transparent white box overlaid in the center containing the title text.

Container für Web Applikationen

4tecture
empower your software solutions



Containers are isolated, but share OS and, where appropriate, bins/libraries



Docker Layers

My ASP.NET Core Application

microsoft/aspnet

microsoft/dotnet

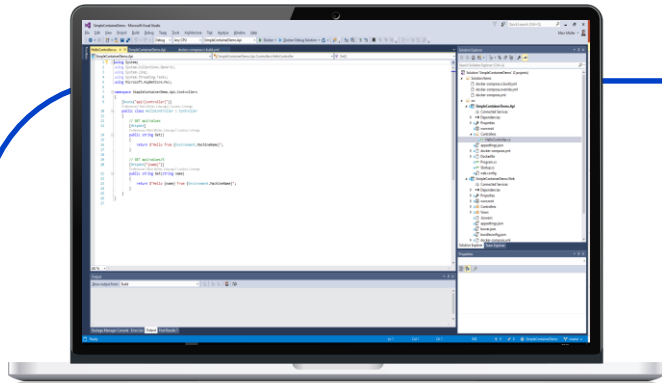
buildpack-deps:jessie-scm

buildpack-deps:jessie-curl

debian:jessie



Environments - Development



Developer PC / Laptop

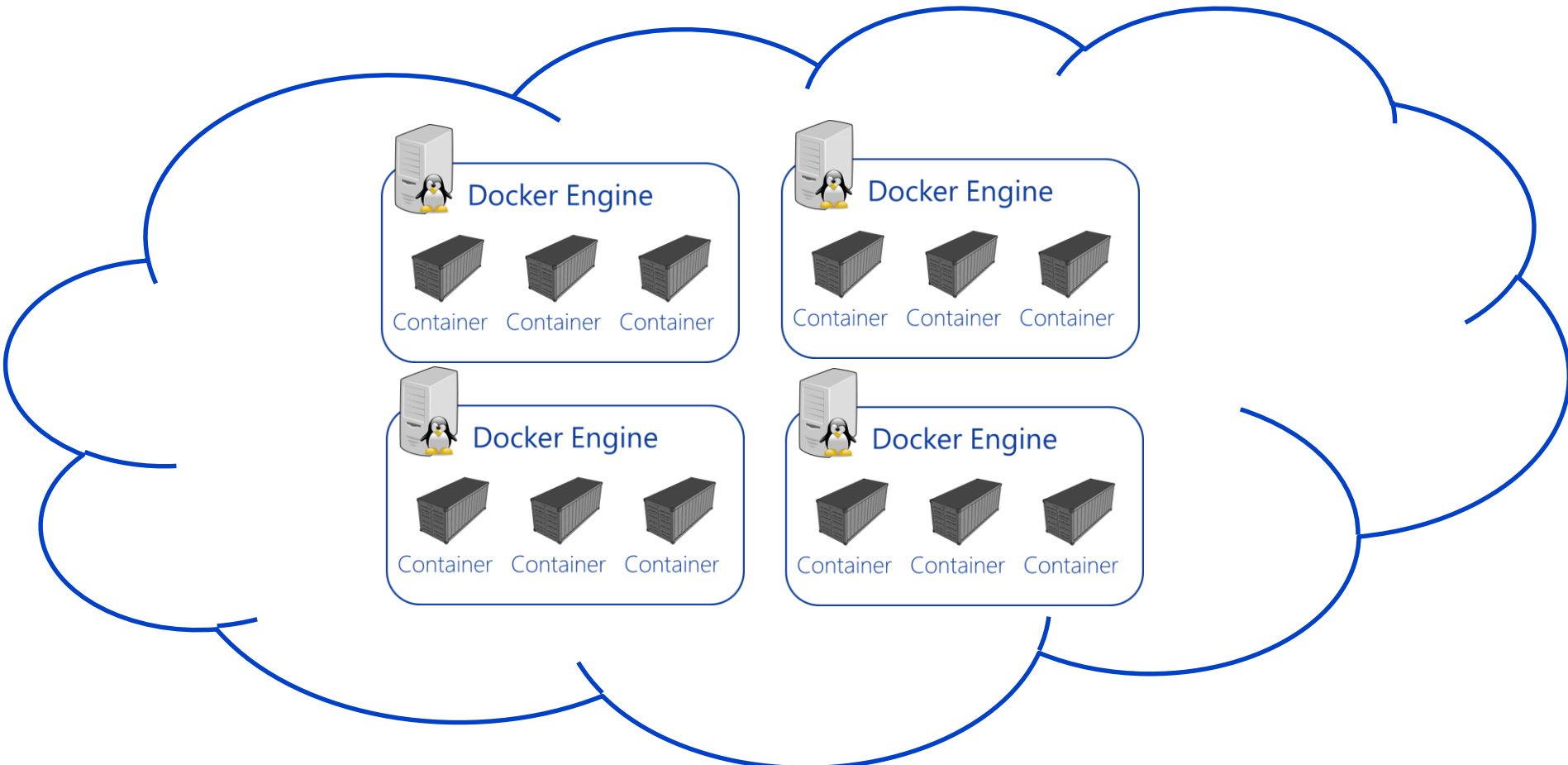


Docker Engine

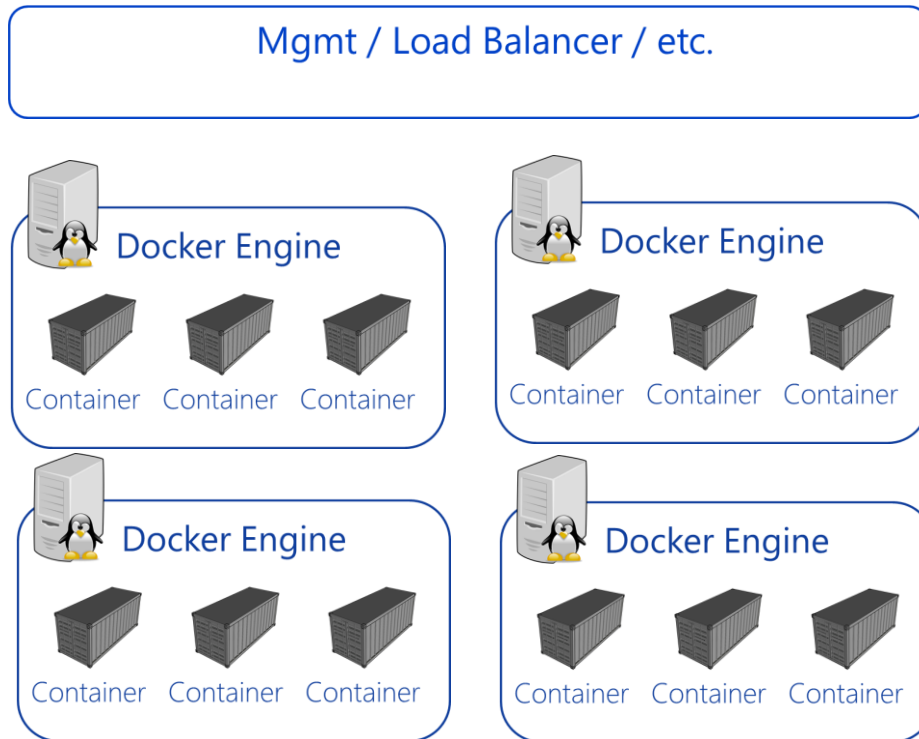
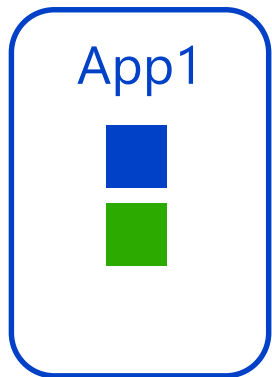


Container Container Container

Environments – Hosting / Cloud



Cluster Management



Vorteile von Containern

- Schnelles Deployment
- Schneller Start
- Einfaches Scaling und Partitioning
- Isolierter, versionierter, wiederverwendbarer Code (Artefakt / Konfigurationselement)

Klare Abgrenzung zwischen Entwicklung und Betrieb

- Entwicklung kümmert sich um den Inhalt des Containers
- Betrieb kümmert sich darum, wie der Container verwendet wird

A background image showing a rowing team in a boat. The rowers are wearing blue long-sleeved shirts and red and white shorts. They are holding oars with yellow handles. The boat is on water, and the background is a bright, overcast sky.

Docker Support in Visual Studio

4tecture
empower your software solutions

Docker CLI

Windows PowerShell

```
C:\Users\mmueller> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3229159f7ff4	devfun.api:dev	"tail -f /dev/null"	5 days ago	Up 42 seconds	0.0.0.0:32769->80/tcp	dockercompose1745314007_devfun.api_1
1c7f868bfe86	devfun.web:dev	"tail -f /dev/null"	5 days ago	Up 43 seconds	0.0.0.0:32768->80/tcp	dockercompose1745314007_devfun.web_1

```
C:\Users\mmueller>
```

Docker for Windows

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

Welcome to the Docs

Get Docker

Install Docker Engine

Docker for Mac

Docker for Windows

Getting Started

Logs and Troubleshooting

FAQs

Example Applications

Open Source Licensing

Release Notes

Docker for Linux

Docker for AWS

Docker for Azure

Docker Toolbox (Legacy)

Get Started

Docker ID

Docker Engine

 **docker**

Join us April 17-20 for **dockercon 17** [Register](#)

DOCS EVENTS COMMUNITY SUPPORT TRAINING PARTNERS BLOG LOG-IN SIGN-UP

What is Docker? Solutions Get Docker Pricing Open Source Company

Get started with Docker for Windows

Estimated reading time: 26 minutes

Welcome to Docker for Windows!

Docker is a full development platform for creating containerized apps, and Docker for Windows is the best way to get started with Docker on Windows systems.

Already have Docker for Windows? If you already have Docker for Windows installed, and are ready to get started, skip down to [Step 3. Check versions of Docker Engine, Compose, and Machine](#) to work through the rest of the Docker for Windows tour, or jump over to getting started tutorials at [Learn Docker](#).

Download Docker for Windows

If you have not already done so, please install Docker for Windows. You can download installers from the stable or beta channel. For more about stable and beta channels, see the [FAQs](#).

Stable channel	Beta channel
This installer is fully baked and tested, and comes with the latest GA version of Docker Engine. This is the best channel to use if you want a reliable platform to work with. These releases follow a version schedule with a longer lead time than the betas, synched with Docker Engine releases and hotfixes. On the stable channel, you can select whether to send usage statistics and other data. Get Docker for Windows [stable] Download checksum: InstallDocker.msi SHA256	This installer offers cutting edge features and comes with the experimental version of Docker Engine, which is described in the Docker Experimental Features README on GitHub. This is the best channel to use if you want to experiment with features under development, and can weather some instability and bugs. This channel is a continuation of the beta program, where you can provide feedback as the apps evolve. Releases are typically more frequent than for stable, often one or more per month. We collect all usage data on betas across the board. Get Docker for Windows [beta] Download checksum: InstallDocker.msi SHA256

Search the docs

Edit This Page

Get started with Docker for Windows

Download Docker for Windows

What to know before you install

About Windows containers and Windows Server 2016

Step 1. Install Docker for Windows

Step 2. Start Docker for Windows

Step 3. Check versions of Docker Engine, Compose, and Machine

Step 4. Explore the application and run examples

Set up tab completion in PowerShell

Docker Settings

General

Shared Drives

Firewall rules for shared drives

Advanced

Network

Proxies

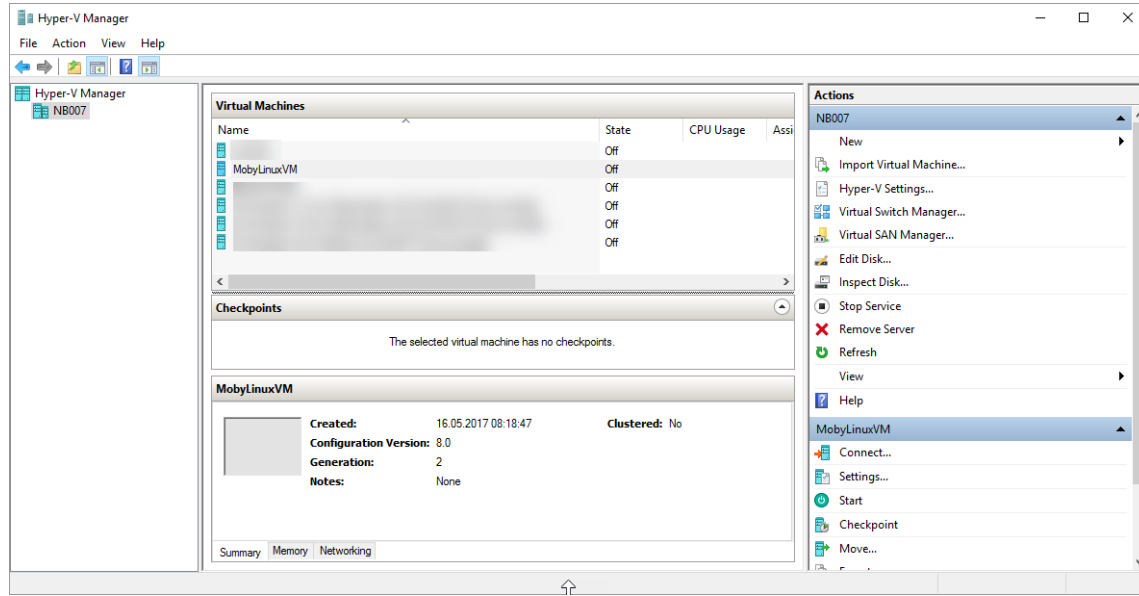
Docker daemon

Experimental mode

Custom registries

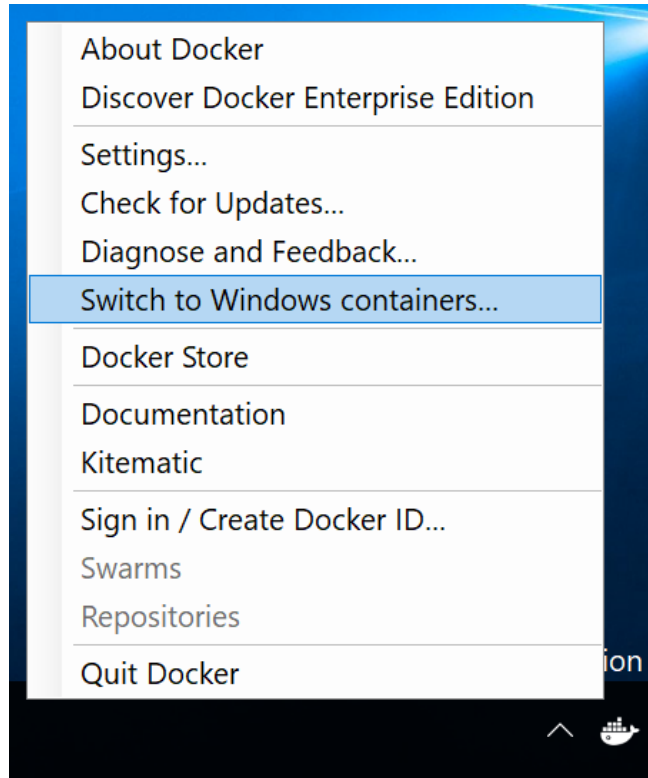
Edit the daemon configuration file

Linux Container (Default)



com.docker.proxy.exe
//./pipe/docker_engine

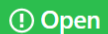
Windows Container



`./.pipe/docker_engine_windows`

Multi-Arch based Tags

.NET Core Docker images will move to multi-arch based tags #239



kendrahavens opened this issue 6 days ago · 2 comments



kendrahavens commented 6 days ago

Contributor



Summary

Docker has a [multi-arch feature](#) that [microsoft/dotnet-nightly](#) recently started utilizing. The plan is to port this to the [official microsoft/dotnet repo](#) shortly. **The multi-arch feature allows a single tag to be used across multiple machine configurations.** Without this feature each architecture/OS/platform requires a unique tag. For example, the microsoft/dotnet:1.0-runtime tag is based on Debian and microsoft/dotnet:1.0-runtime-nanoserver if based on Nano Server. **With multi-arch there will be one common microsoft/dotnet:1.0-runtime tag.** If you pull that tag from a Linux container environment you will get the Debian based image whereas if you pull that tag from a Windows container environment you will get the Nano Server based image. This helps provide tag uniformity across Docker environments thus eliminating confusion.

Current microsoft/dotnet tags

<https://github.com/dotnet/dotnet-docker/issues/239>

Demo

- Docker for Windows
- Visual Studio Docker Support
- Proxy Container

Building	
PK	<u>buildingNr</u>
U1	streetAddress
FK1,1,1,U1	isHeadquarters
U1	countryname
U1	postCode
U1	stateCode
U1	cityName

Sales	
PK,FK2,1,2	<u>productNr</u>
PK,FK1,1,1	<u>countryName</u>
PK	yearNr
PK	quarterNr
	nrUnitsSold

Employee	
PK	<u>employeeId</u>
	type
	dingNr
	omNr
	department
	division
	jobTitle
	manager
	adsDept
	he-1sDivisi
	mobile...
	birthDate

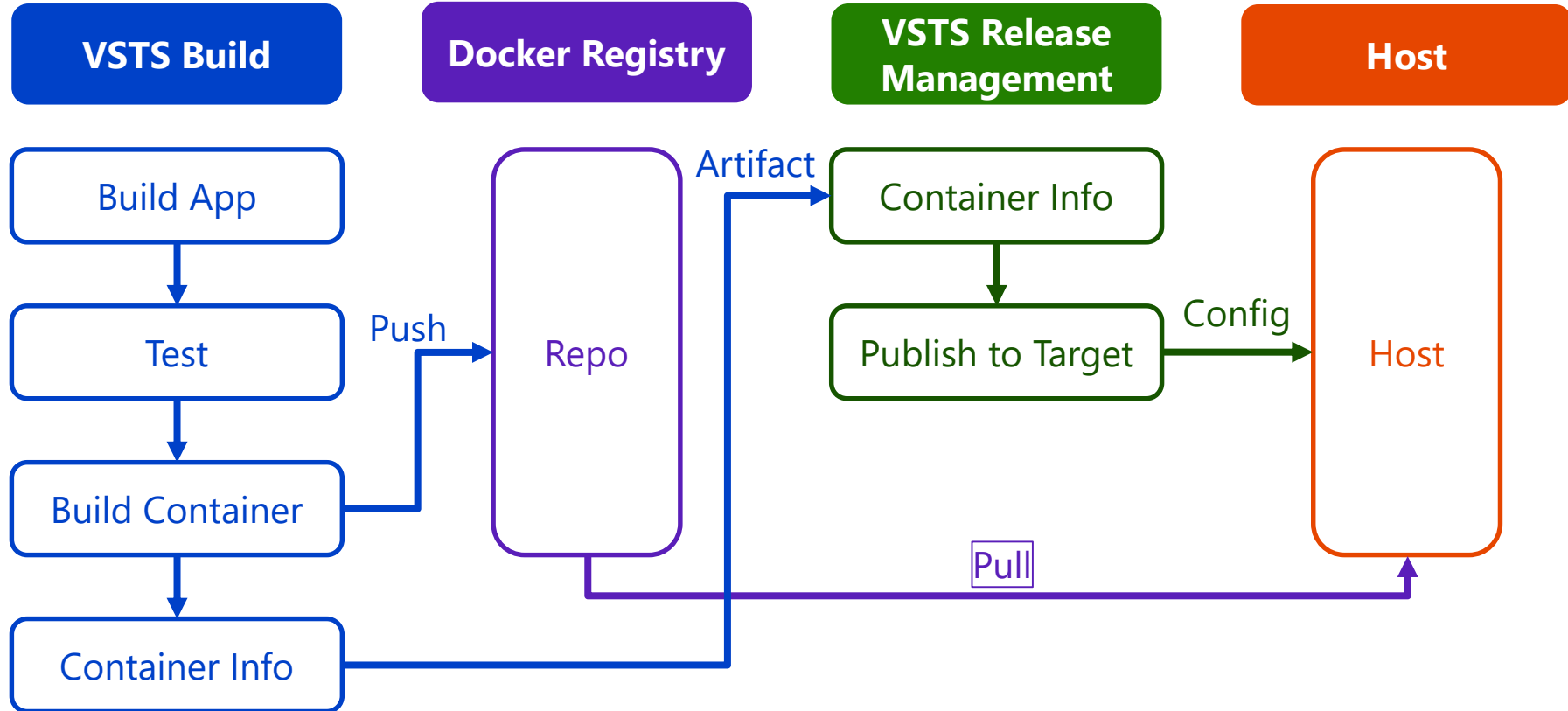
Development	
PK,FK2,1,2	<u>productNr</u>
PK,FK1,1,1	<u>personNr</u>

A background image showing a rowing team in a boat. The rowers are wearing blue and red uniforms and are pulling oars with yellow handles. The boat is on water, and the background is slightly blurred.

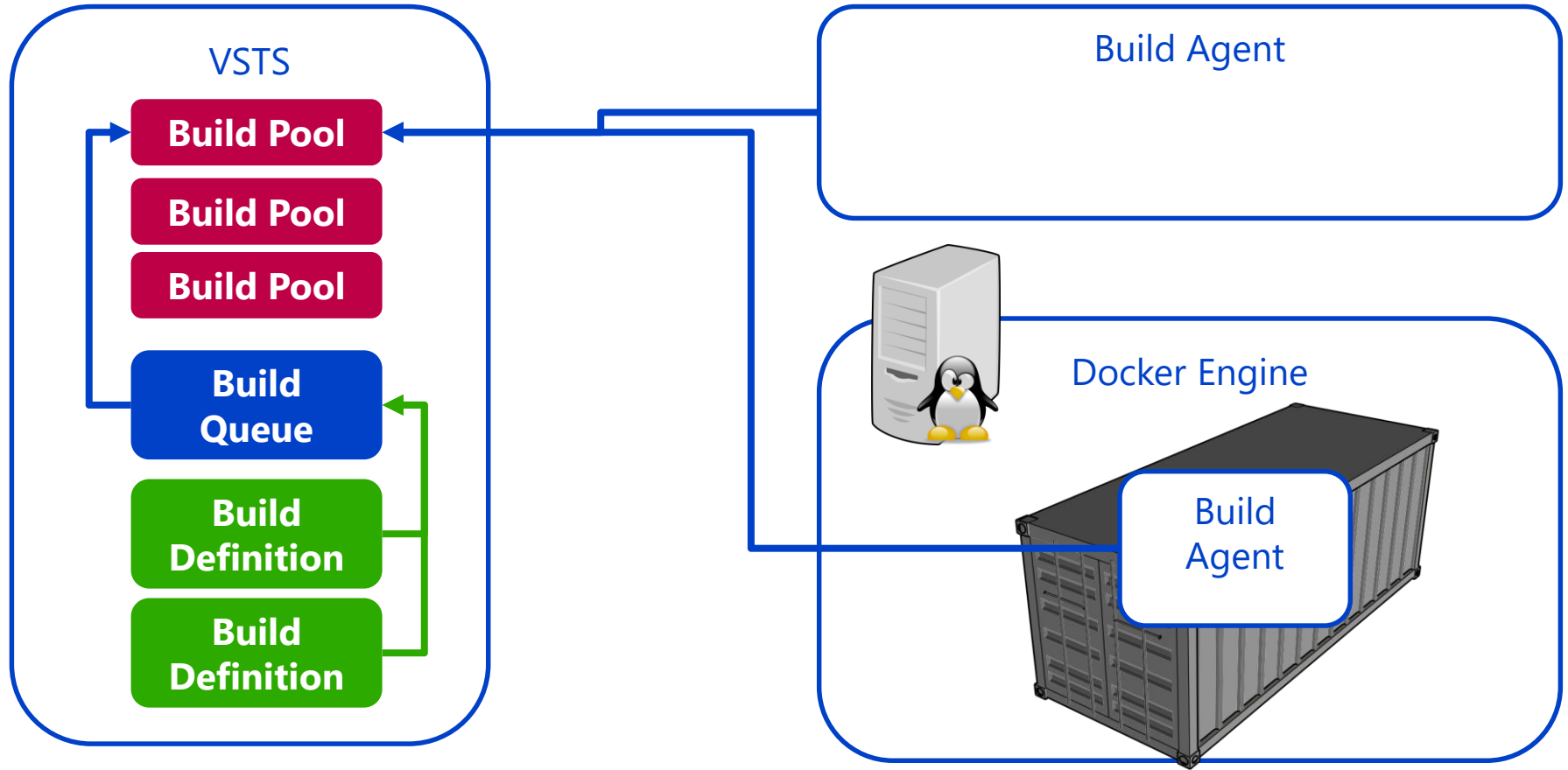
Docker CI Build

4tecture
empower your software solutions

Docker CI / CD



VSTS / TFS Build Agent



Custom Docker Build Agent

The screenshot displays the Visual Studio Team Explorer interface with the 'Agent pools' tab selected. A new pool named 'Docker' has been created. The 'Agents' table shows one agent, '24e6d643b590-ag...', which is online and idle. A 'Download agent' button is visible. On the right, the 'Capabilities' section shows user-defined and system capabilities.

Agents for pool Docker

Enabled	Name	State	Current Status
☒	24e6d643b590-ag...	Online	Idle

USER CAPABILITIES
Shows information about user-defined capabilities supported by this host

SYSTEM CAPABILITIES
Shows information about the capabilities provided by this host

Capability	Value
Agent.ComputerName	24e6d643b590
Agent.Name	24e6d643b590-agent
Agent.OS	linux
Agent.Version	2.111.1
ant	/usr/bin/ant
ANT_HOME	/usr/share/ant
bower	/usr/local/bin/bower
curl	/usr/bin/curl
DOCKER_BUCKET	get.docker.com
DOCKER_COMPOSE_VERSION	1.8.0
DOCKER_SHA256	05ceec7fd937e1416e5dce12b0b6e1c655907d349d52574319e
DOCKER_VERSION	1.12.1
dotnet	/usr/bin/dotnet
git	/usr/bin/git
GRADLE_HOME	/usr/share/gradle
grunt	/usr/local/bin/grunt
gulp	/usr/local/bin/gulp
HOME	/root

Windows PowerShell

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

Loading personal and system profiles took 665ms.
C:\Users\mmueller> docker run -e VSTS_ACCOUNT=4tecture-training -e VSTS_TOKEN=456hjmz3kg45sxics65zbz3xctbsol4vrtaow7bpxizk
f36z1va -e VSTS_AGENT=$(hostname) -agent' -e VSTS_POOL=Docker -e VSTS_WORK=/var/vsts/$VSTS_AGENT' -v /var/run/docker.sock
:/var/run/docker.sock -it microsoft/vsts-agent
Determining matching VSTS agent...
Downloading and installing VSTS agent...

>> End User License Agreements:

Building sources from a TFVC repository requires accepting the Team Explorer Everywhere End User License Agreement. This s
ep is not required for building sources from Git repositories.

A copy of the Team Explorer Everywhere license agreement can be found at:
/var/agent/externals/tee/license.html

>> Connect:

Connecting to server ...

>> Register Agent:

Scanning for tool capabilities.
Connecting to the server.
Successfully added the agent
Testing agent connection.
2017-01-27 08:04:35Z: Settings Saved.
Scanning for tool capabilities.
Connecting to the server.
2017-01-27 08:04:40Z: Listening for Jobs
```

Demo

- Compile in Docker Container
- Setup Docker Build Agent

Building	
PK	<u>buildingNr</u>
U1	streetAddress
FK1,1,1,U1	isHeadquarters
U1	countryname
U1	postCode
U1	stateCode
U1	cityName

Sales	
PK,FK2,1,2	<u>productNr</u>
PK,FK1,1,1	<u>countryName</u>
PK	<u>yearNr</u>
PK	<u>quarterNr</u>
	nrUnitsSold

Employee	
PK	<u>employeeNr</u>
	type
	endingNr
	omNr
	department
	division
	jobTitle
	manager
	tradsDept
	he-1sDivisi
	mobile...
	birthDate

Development	
PK,FK2,1,2	<u>productNr</u>
PK,FK1,1,1	<u>personNr</u>

A background image showing a rowing team in a boat. The rowers are wearing blue and red uniforms and are captured in a synchronized rowing motion. The focus is on the oars and the rowers' hands, with the water visible in the foreground.

Docker Hosting

4tecture
empower your software solutions

Docker Registry im Azure Portal

The screenshot displays the Microsoft Azure Portal interface for managing a Container Registry. The top navigation bar shows the path: **Microsoft Azure** > **Container registries** > **PlaygroundRegistry - Access key**. A search bar is located in the top right corner.

The left sidebar contains various navigation icons, including a hamburger menu, a plus sign, and icons for subscriptions, resources, and settings.

The main content area is divided into two sections:

- Container registries** (left):
 - Subscriptions: 1 of 2 selected – Don't see a subscription? [Switch directories](#)
 - Filter by name...
 - MVP VS Ultimate (dropdown)
 - 1 items
 - NAME (dropdown)
 - PlaygroundRegistry (highlighted)
- PlaygroundRegistry - Access key** (right):
 - Search (Ctrl+ /)
 - Overview
 - Activity log
 - Access control (IAM)
 - Tags
 - SETTINGS
 - Access key (highlighted)
 - Locks
 - SUPPORT + TROUBLESHOOTING
 - New support request

The right-hand form contains the following fields and controls:

- Registry name**: PlaygroundRegistry
- Login Server**: playgroundregistry-on.azurecr.io
- Admin user**:
 - Enable (selected) / Disable
- Username**: PlaygroundRegistry
- Password**: (masked with dots) and a refresh icon

Docker Registry API

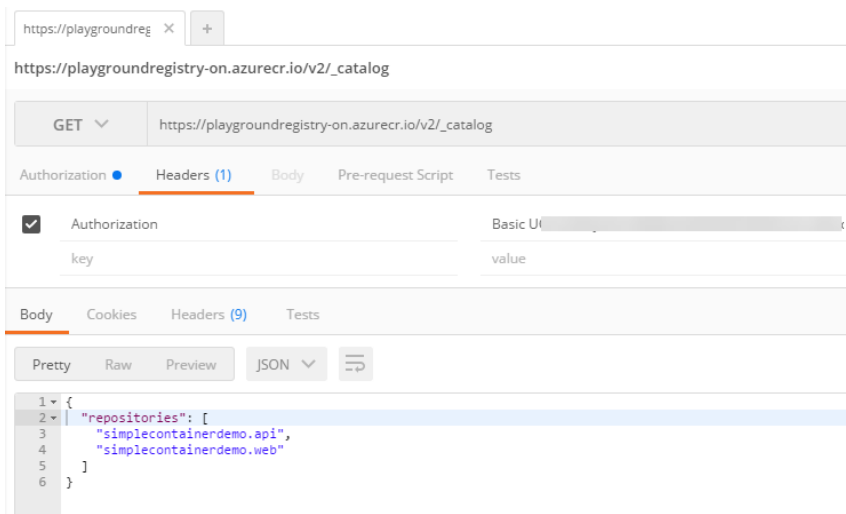
Method	Path	Entity	Description
GET	/v2/	Base	Check that the endpoint implements Docker Registry API V2.
GET	/v2/<name>/tags/list	Tags	Fetch the tags under the repository identified by name.
GET	/v2/<name>/manifests/<reference>	Manifest	Fetch the manifest identified by name and reference where reference can be a tag or digest. A HEAD request can also be issued to this endpoint to obtain resource information without receiving all data.
PUT	/v2/<name>/manifests/<reference>	Manifest	Put the manifest identified by name and reference where reference can be a tag or digest.
DELETE	/v2/<name>/manifests/<reference>	Manifest	Delete the manifest identified by name and reference. Note that a manifest can <i>only</i> be deleted by digest.
GET	/v2/<name>/blobs/<digest>	Blob	Retrieve the blob from the registry identified by digest. A HEAD request can also be issued to this endpoint to obtain resource information without receiving all data.
DELETE	/v2/<name>/blobs/<digest>	Blob	Delete the blob identified by name and digest
POST	/v2/<name>/blobs/uploads/	Initiate Blob Upload	Initiate a resumable blob upload. If successful, an upload location will be provided to complete the upload. Optionally, if the digest parameter is present, the request body will be used to complete the upload in a single request.
GET	/v2/<name>/blobs/uploads/<uuid>	Blob Upload	Retrieve status of upload identified by uuid. The primary purpose of this endpoint is to resolve the current status of a resumable upload.
PATCH	/v2/<name>/blobs/uploads/<uuid>	Blob Upload	Upload a chunk of data for the specified upload.
PUT	/v2/<name>/blobs/uploads/<uuid>	Blob Upload	Complete the upload specified by uuid, optionally appending the body as the final chunk.
DELETE	/v2/<name>/blobs/uploads/<uuid>	Blob Upload	Cancel outstanding upload processes, releasing associated resources. If this is not called, the unfinished uploads will eventually timeout.
GET	/v2/_catalog	Catalog	Retrieve a sorted, json list of repositories available in the registry.

Image Source: <https://docs.docker.com/registry/spec/api/>

Docker Registry API - Samples

List Repositories: `/v2/_catalog`

List Tags: `/v2/<repository>/tags/list`



https://playgroundreg x +

https://playgroundregistry-on.azurecr.io/v2/_catalog

GET v https://playgroundregistry-on.azurecr.io/v2/_catalog

Authorization Headers (1) Body Pre-request Script Tests

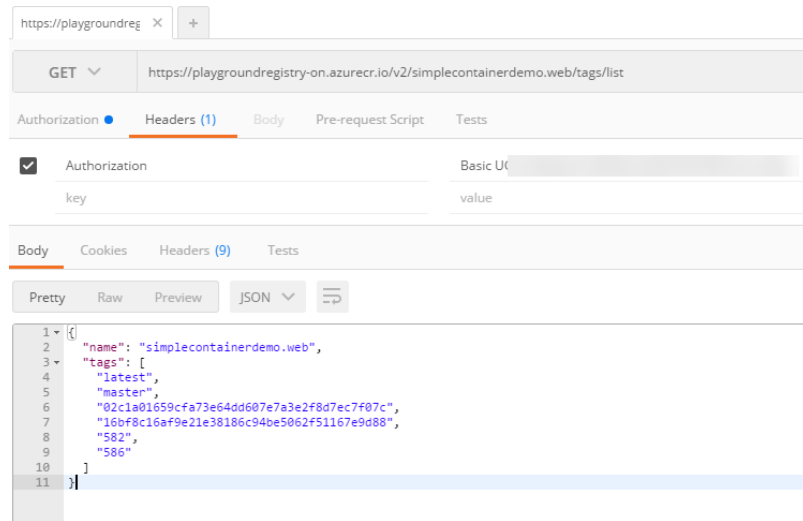
☒ Authorization Basic UI

key value

Body Cookies Headers (9) Tests

Pretty Raw Preview JSON v

```
1 {
2   "repositories": [
3     "simplecontainerdemo.api",
4     "simplecontainerdemo.web"
5   ]
6 }
```



https://playgroundreg x +

GET v https://playgroundregistry-on.azurecr.io/v2/simplecontainerdemo.web/tags/list

Authorization Headers (1) Body Pre-request Script Tests

☒ Authorization Basic UI

key value

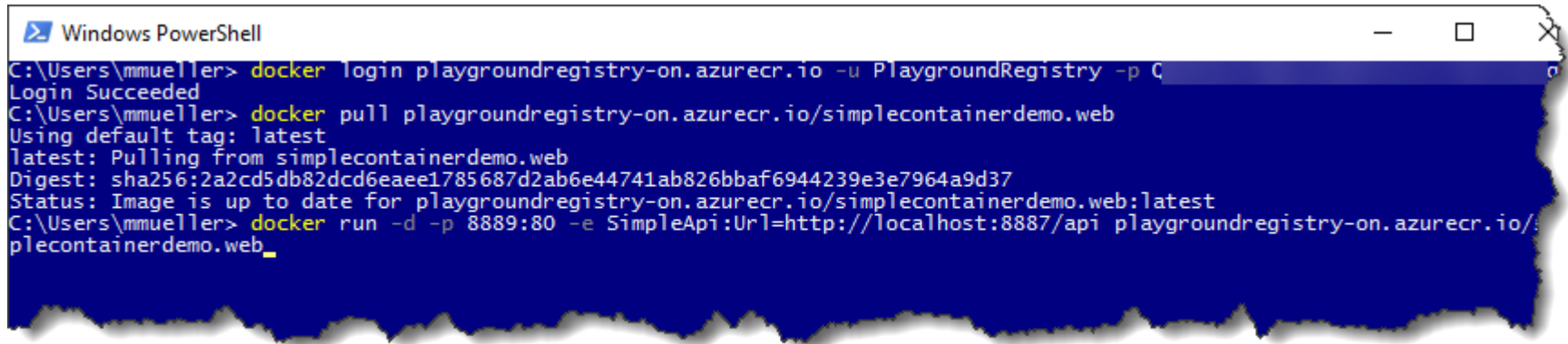
Body Cookies Headers (9) Tests

Pretty Raw Preview JSON v

```
1 {
2   "name": "simplecontainerdemo.web",
3   "tags": [
4     "latest",
5     "master",
6     "02c1a01659cfa73e64dd607e7a3e2f8d7ec7f07c",
7     "16bf8c16af9e21e38186c94be5062f51167e9d88",
8     "582",
9     "586"
10  ]
11 }
```

Azure Container Registry und Docker

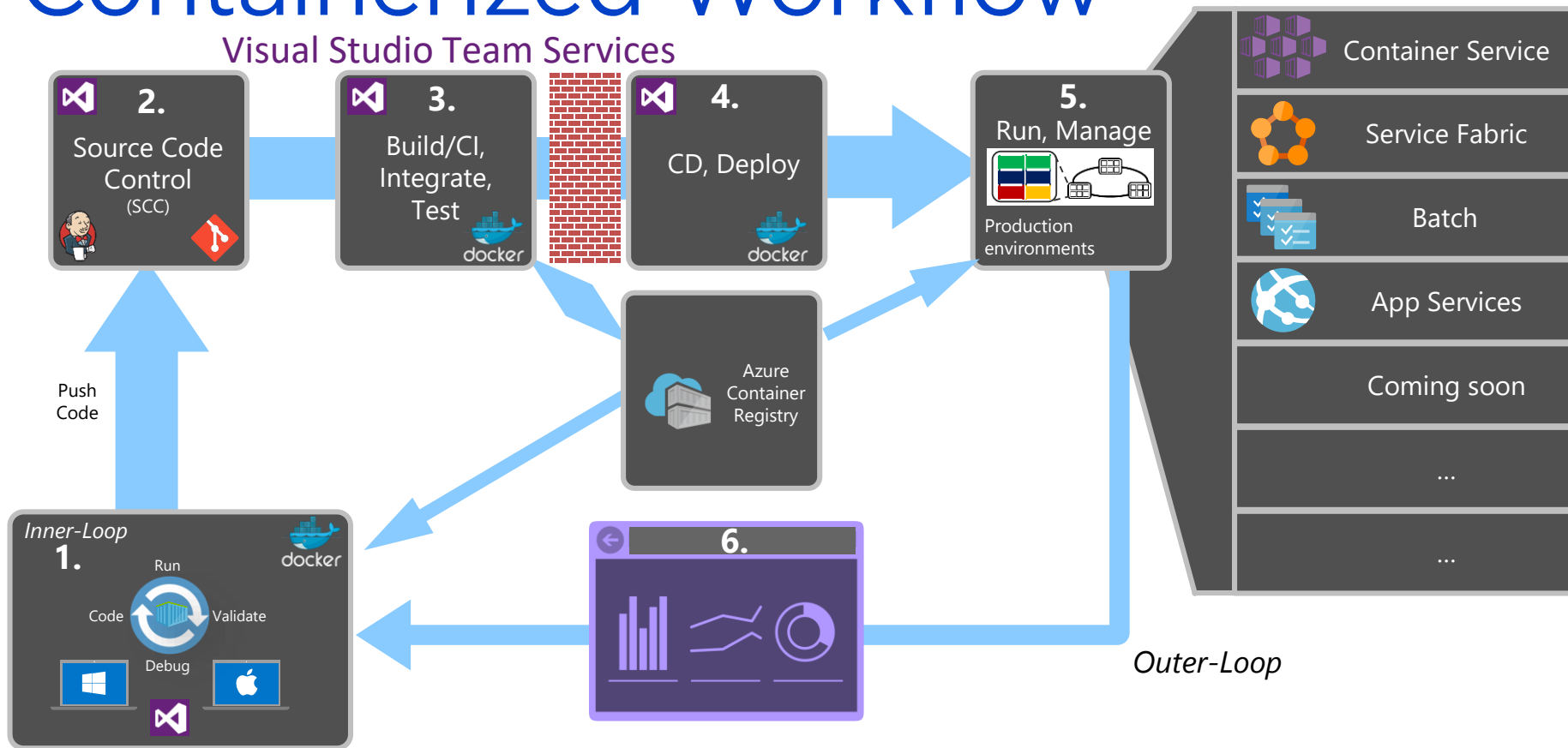
- Login to your private registry `docker login <registry> -u username -p password`
- Pull the desired image `docker pull <registry>/<repository>:<tag>`
- Run the container `docker run...`

A screenshot of a Windows PowerShell terminal window. The title bar reads "Windows PowerShell". The terminal content shows a sequence of Docker commands and their outputs. The first command is 'docker login playgroundregistry-on.azurecr.io -u PlaygroundRegistry -p C...', which results in 'Login Succeeded'. The second command is 'docker pull playgroundregistry-on.azurecr.io/simplecontainerdemo.web', which results in 'Using default tag: latest', 'latest: Pulling from simplecontainerdemo.web', and a long SHA256 digest. The third command is 'docker run -d -p 8889:80 -e SimpleApi:Url=http://localhost:8887/api playgroundregistry-on.azurecr.io/simplecontainerdemo.web', which results in a prompt character '_'.

```
Windows PowerShell
C:\Users\mmueller> docker login playgroundregistry-on.azurecr.io -u PlaygroundRegistry -p C...
Login Succeeded
C:\Users\mmueller> docker pull playgroundregistry-on.azurecr.io/simplecontainerdemo.web
Using default tag: latest
latest: Pulling from simplecontainerdemo.web
Digest: sha256:2a2cd5db82dcd6eae1785687d2ab6e44741ab826bbaf6944239e3e7964a9d37
Status: Image is up to date for playgroundregistry-on.azurecr.io/simplecontainerdemo.web:latest
C:\Users\mmueller> docker run -d -p 8889:80 -e SimpleApi:Url=http://localhost:8887/api playgroundregistry-on.azurecr.io/!
plecontainerdemo.web_
```

Containerized Workflow

Visual Studio Team Services

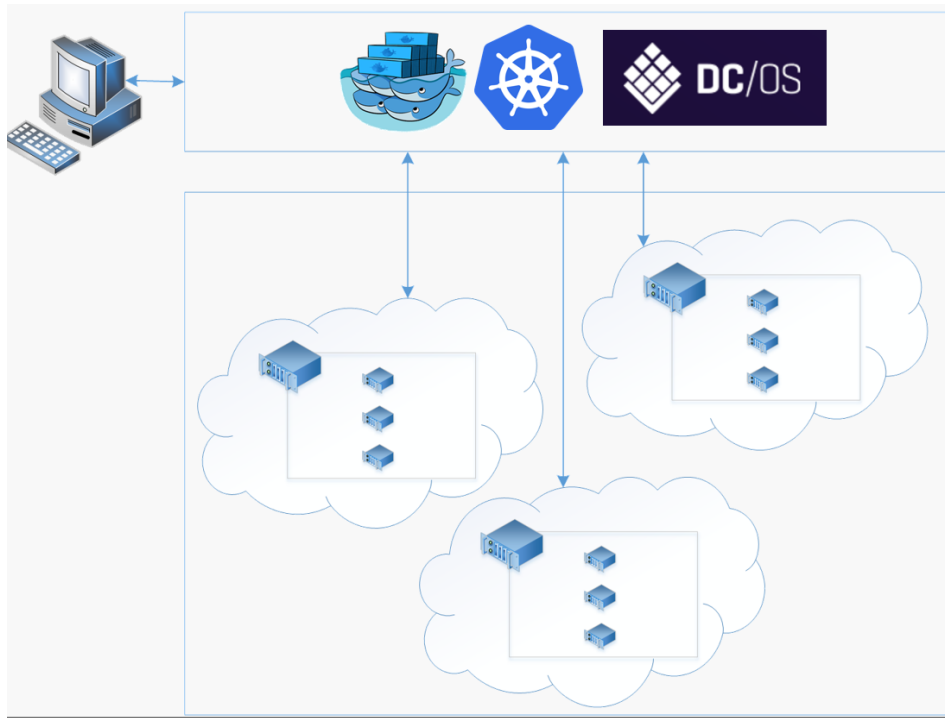


Azure Container Service

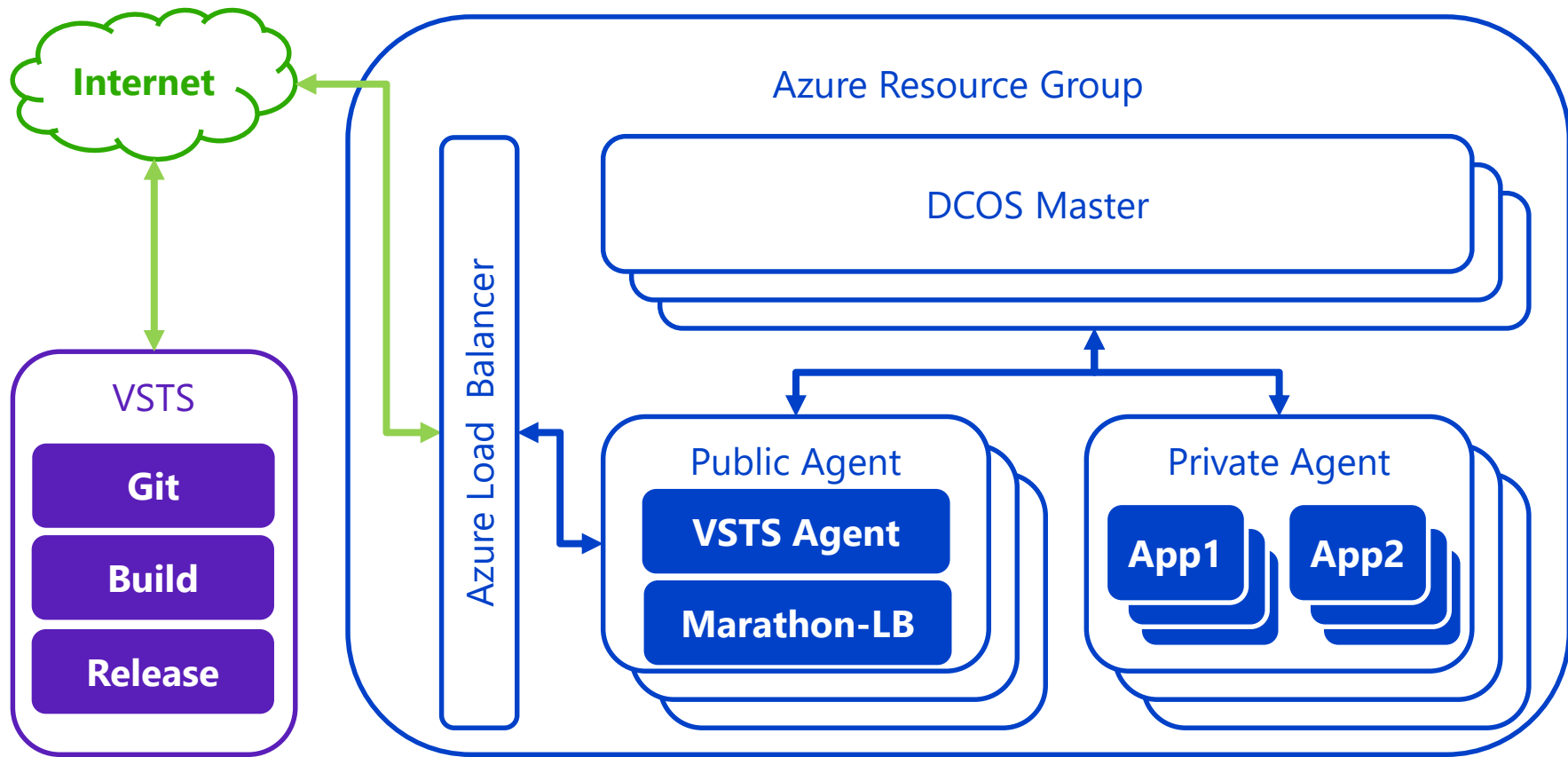
ACS = ARM Templates

3 Varianten

- DC/OS with Marathon
- Docker Swarm
- Kubernetes



Simplified ACS DCOS Setup



Load Balancer Settings for Services

ACSPlayground

SimpleContainerDemo

Files History Branches Pull Requests

SimpleContainerDemo

- src
- .gitattributes
- .gitignore
- docker-compose.ci.build.yml
- docker-compose.env.dev.yml**
- docker-compose.env.production.yml
- docker-compose.env.test.yml
- docker-compose.override.yml
- docker-compose.vs.debug.yml
- docker-compose.vs.release.yml
- docker-compose.yml
- SimpleContainerDemo.sln

BuildACS SimpleContainerDemo / **docker-compose.env.dev.yml**

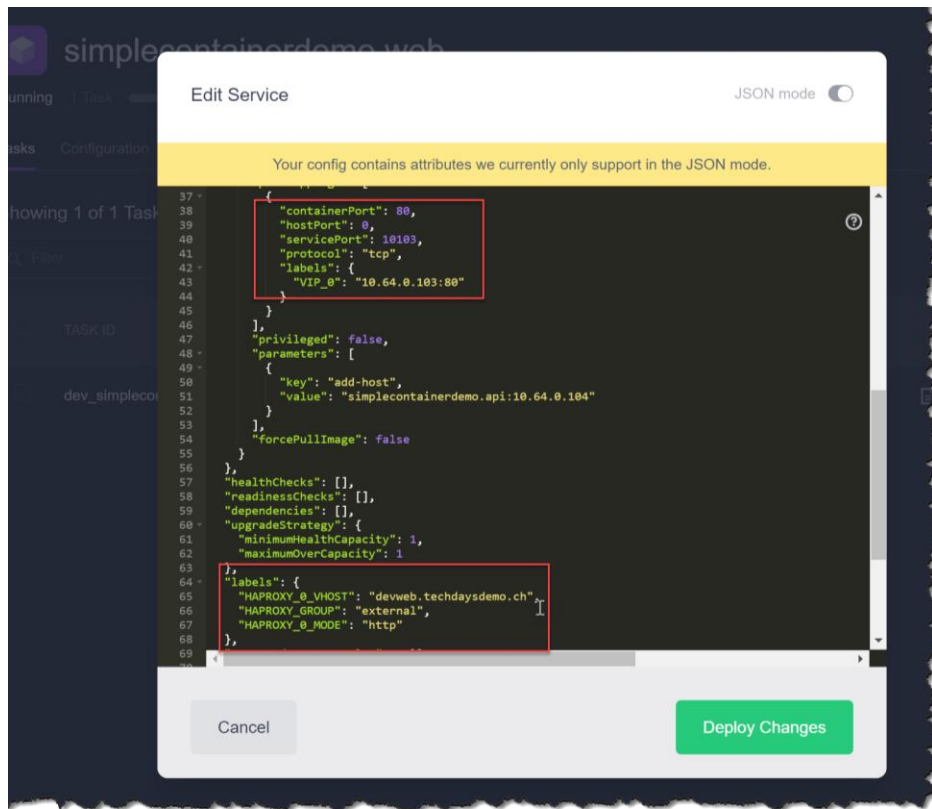
You updated Proxy 41 minutes ago — [Create a pull request](#)

Contents History Compare

Edit Download

```
1 version: "2"
2 services:
3   simplecontainerdemo.api:
4     expose:
5       - "80"
6     labels:
7       HAProxy_GROUP: "external"
8       HAProxy_0_VHOST: "devapi.techdaysdemo.ch"
9       HAProxy_0_MODE: "http"
10  simplecontainerdemo.web:
11    expose:
12      - "80"
13    labels:
14      HAProxy_GROUP: "external"
15      HAProxy_0_VHOST: "devweb.techdaysdemo.ch"
16      HAProxy_0_MODE: "http"
17  environment:
18    - SimpleApi:Url=http://devapi.techdaysdemo.ch/api
19    - DeploymentEnvironment=Development
20
```


Load Balancer Settings for Services



Use Environment Variables

The screenshot shows the ACSPlayground interface with a file explorer on the left and a code editor on the right. The file explorer lists files under 'SimpleContainerDemo', with 'docker-compose.env.dev.yml' selected. The code editor displays the 'Startup' and 'ConfigureServices' methods of a web application. Red boxes highlight the use of environment variables in the code and the docker-compose file.

```
15 {
16     0 references | Marc Müller, 6 days ago | 1 author, 1 change
17     public Startup(IHostingEnvironment env)
18     {
19         var builder = new ConfigurationBuilder()
20             .SetBasePath(env.ContentRootPath)
21             .AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)
22             .AddJsonFile($"appsettings-{env.EnvironmentName}.json", optional: true)
23             .AddEnvironmentVariables();
24         Configuration = builder.Build();
25     }
26
27     4 references | Marc Müller, 6 days ago | 1 author, 1 change
28     public IConfigurationRoot Configuration { get; }
29
30     // This method gets called by the runtime. Use this method to add services to the container.
31     0 references | Marc Müller, 6 hours ago | 1 author, 3 changes
32     public void ConfigureServices(IServiceCollection services)
33     {
34         // Add framework services.
35         services.AddMvc();
36
37         services.AddSingleton<SimpleApiOptions>(new SimpleApiOptions() { ApiUrl = Configuration["SimpleApi:Url"], DeploymentEnvironment = Configuration["DeploymentEnvironment"] });
38     }
39
40     expose:
41     - "80"
42     labels:
43     - HAPROXY_GROUP: "external"
44     - HAPROXY_0_VHOST: "devweb.techdaysdemo.ch"
45     - HAPROXY_0_MODE: "http"
46     environment:
47     - SimpleApi:Url=http://devapi.techdaysdemo.ch/api
48     - DeploymentEnvironment=Development
```

A close-up, low-angle shot of several rowers in a boat, wearing blue and red uniforms, pulling their oars. The oars are black with yellow handles and are connected to a complex mechanical system. The background is a bright, slightly blurred body of water.

Zusammenfassung

4tecture
empower your software solutions

Zusammenfassung

- Micro-Service Architektur
Reduktion der Komplexität, einfacher und schnellere Deployments
- Automatisierung der Build- und Release-Prozesse
- Isolation der Services als Container
- Isolation der Build Umgebung (Agent)
- Load Balancing mittels Service Discovery

Besten Dank für Ihre Aufmerksamkeit!

Für allfällige Fragen stehen wir Ihnen jederzeit gerne zur Verfügung:

4tecture GmbH
Aathalstrasse 84
CH-8610 Uster

Marc Müller
Principal Consultant

+41 44 508 37 00
info@4tecture.ch
www.4tecture.ch

marc.mueller@4tecture.ch
@muellermarc

4^{te}cture[©]

empower your software solutions

DIGICOMP

