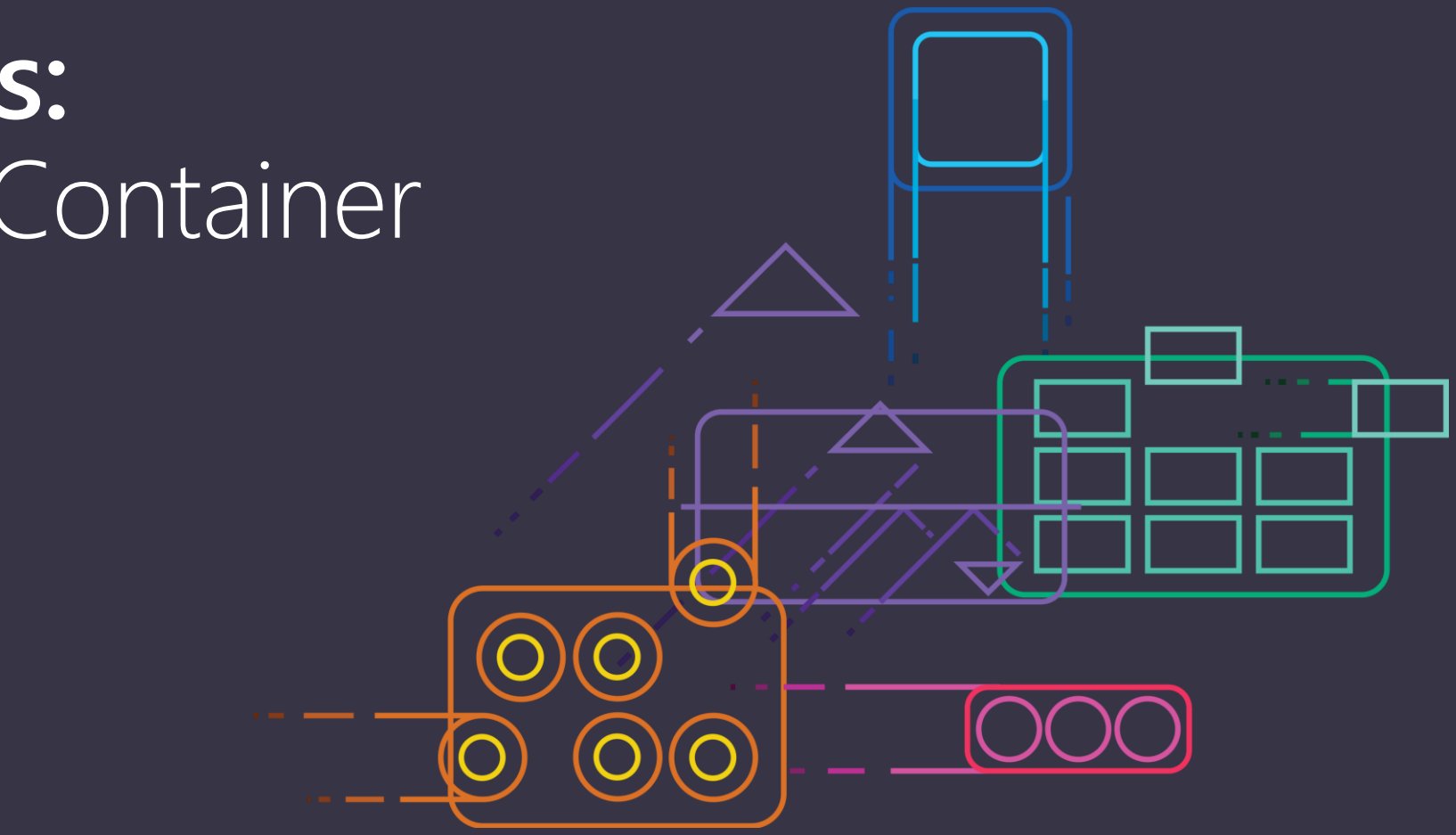
Microsoft

Azure DevOps:
… und Docker-Container



Kostenloses Zusatzmaterial

Nicht verpassen: Zu diesem Vortrag gibt es von uns noch passende Checklisten und Tipps unter:

https://go.nenoloje.com/**msvtw2020**

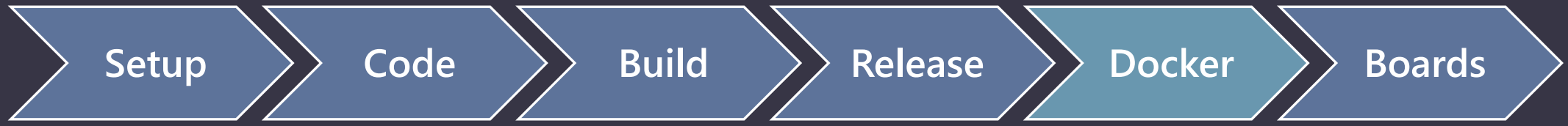
Neno Loje
www.nenoloje.com

Marc Müller
www.4tecture.ch

Willkommen!

Azure DevOps Webinare

- ✔ Folge 1: Schnellstart mit Azure DevOps
- ✔ Folge 2: Moderne Versionsverwaltung mit Git und Pull Requests
- ✔ Folge 3: Build-Automatisierung und Continuous Integration (CI)
- ✔ Folge 4: Deployment-Automatisierung mit Release Pipelines
- 👉 Folge 5: Docker-Container und Azure DevOps
- 📅 Folge 6: Agiles Arbeiten mit Azure Boards



Agenda

1. Kurze Einführung in Docker
2. Erstellen von Docker Images im Build
3. Docker als Build-Environment
4. Erweiterte Szenarien mit Docker und Build-Pipelines

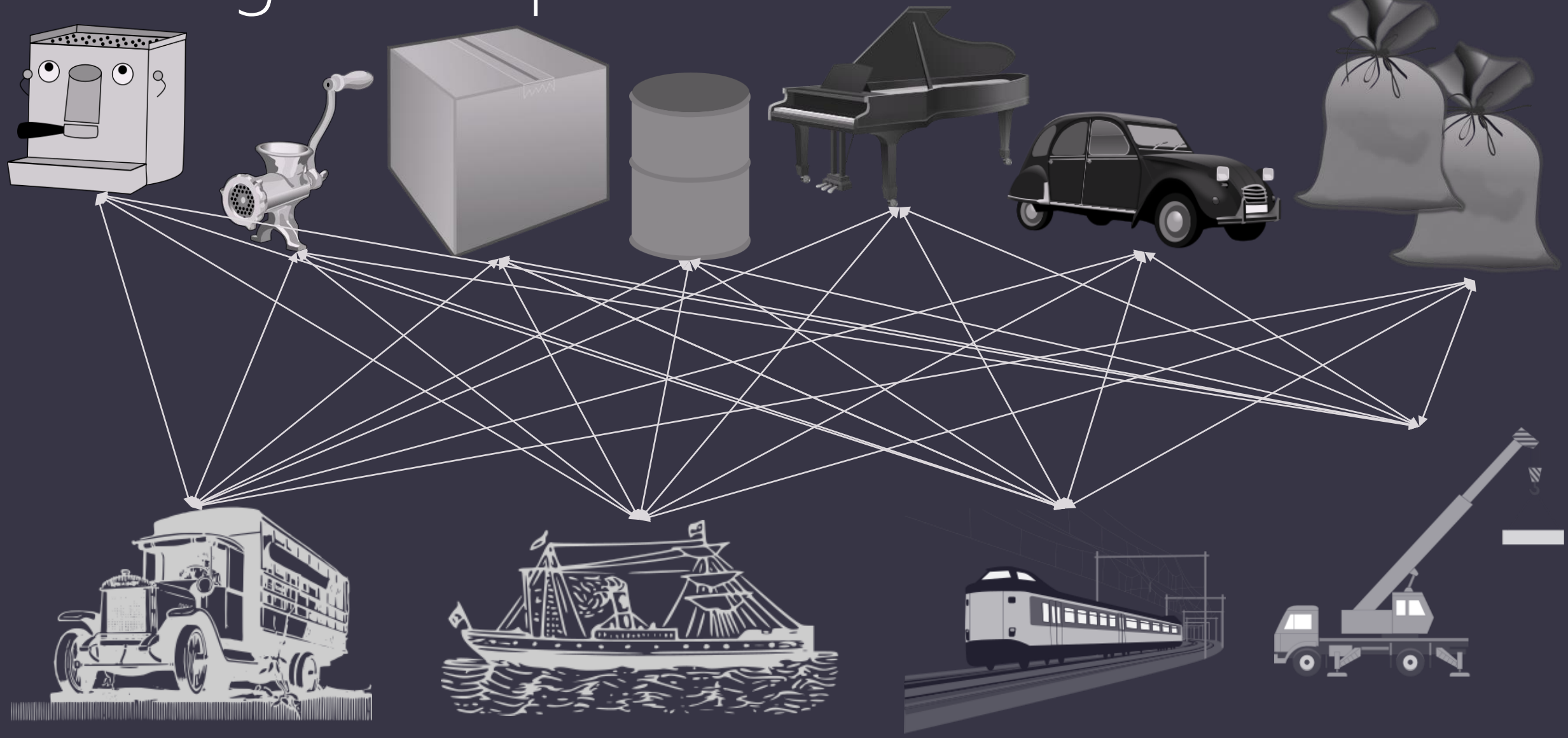
Docker

Im DevOps-Prozess mit Azure DevOps / Azure Pipelines

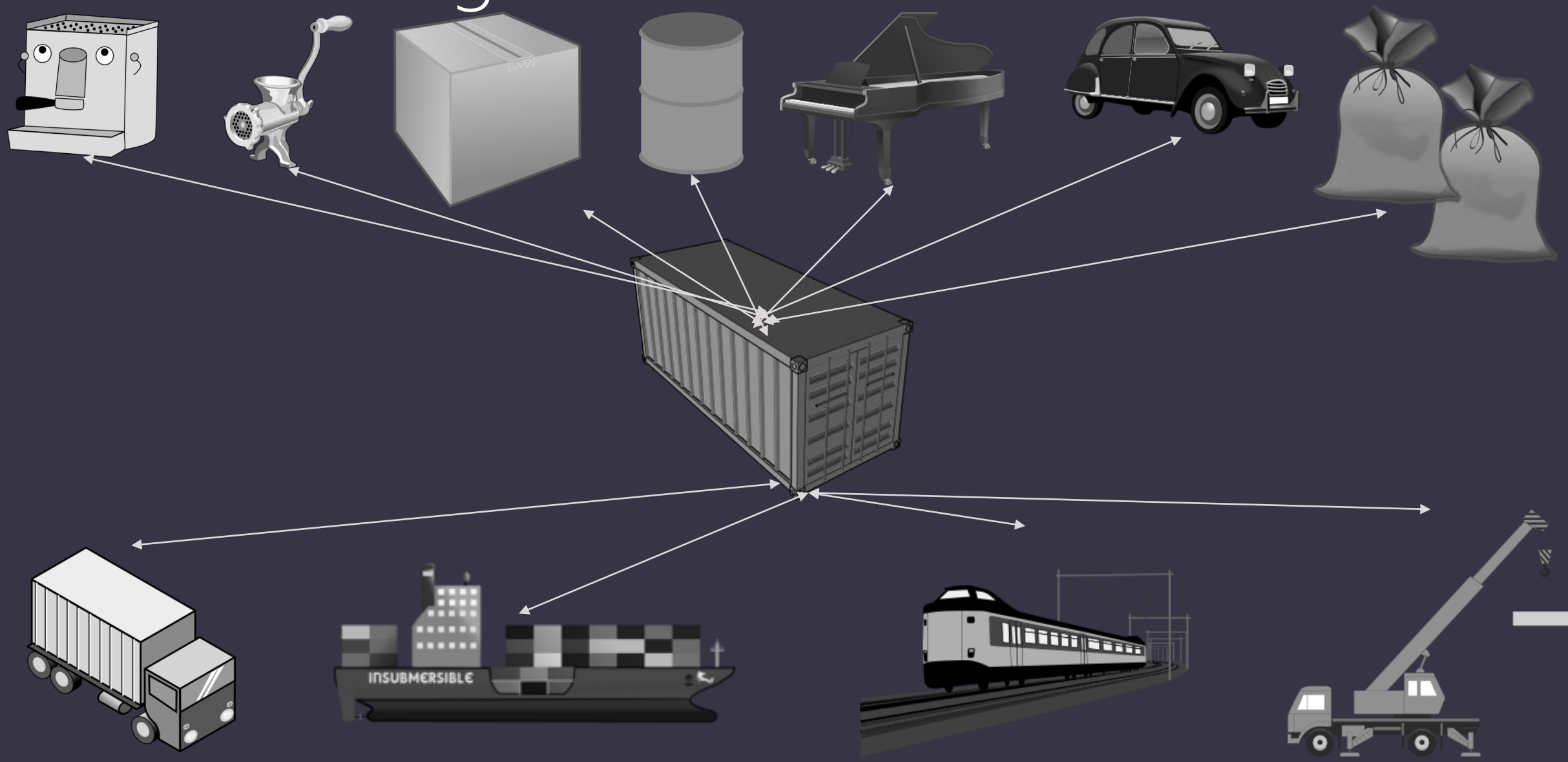
Warum Docker?

Vorteile von Container-basierten Entwicklungsprozessen

Freight transport before 1960



Die Lösung: Container



Vorteile von Containern

Versioniert

Unveränderlich

Keine “Abhängigkeits-Hölle”

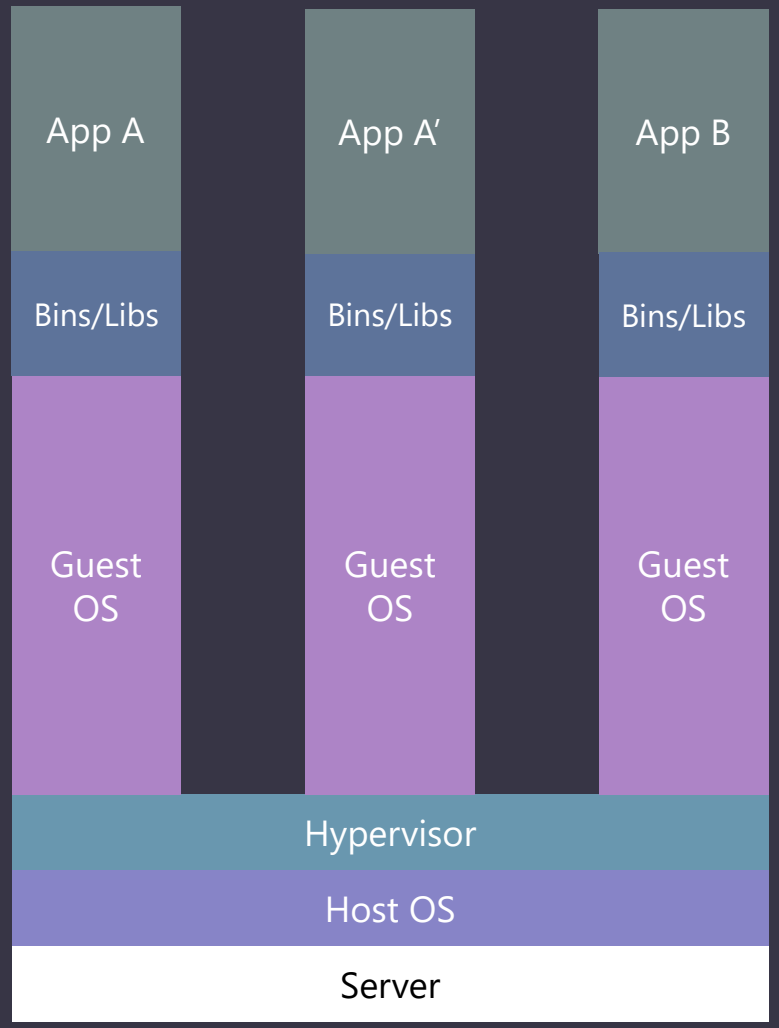
Läuft überall

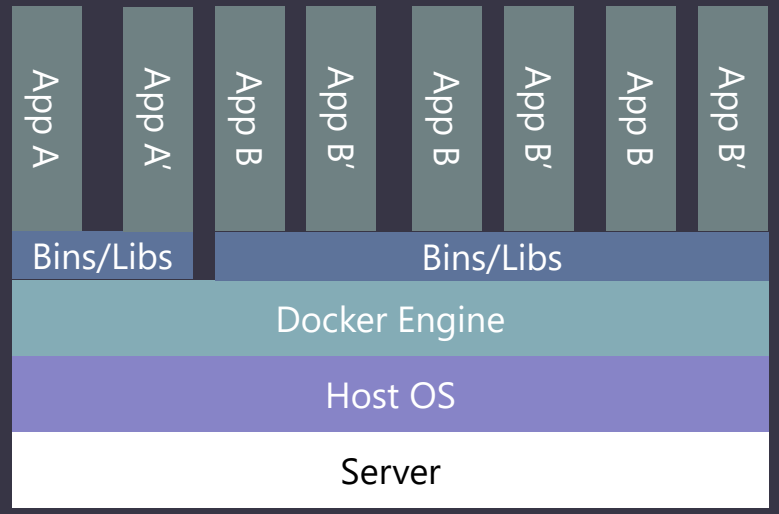
Minimalistische Basis-Installation

Schnell

Ideal zum Testen, keine «no repro»-Bugs mehr

Virtuelle Maschine vs. Container





Docker Layers

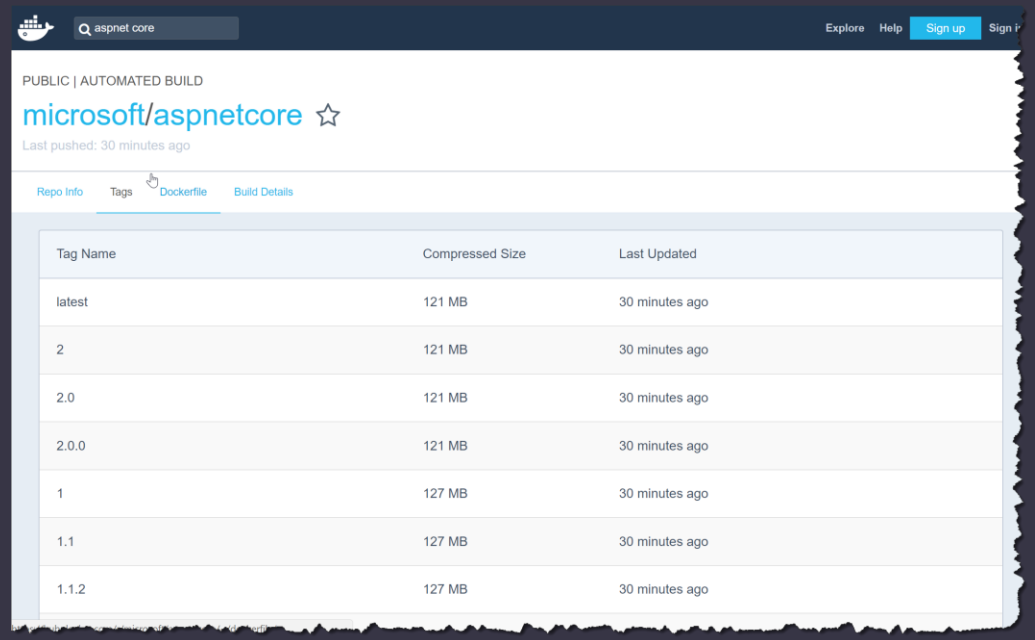
My ASP.NET Core Application

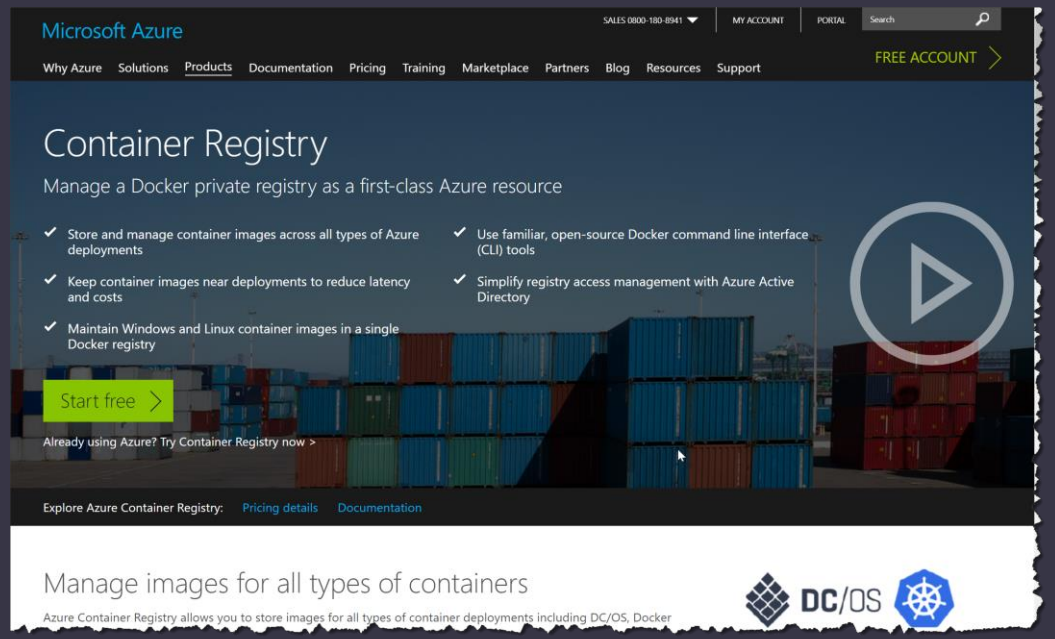
mcr.microsoft.com/dotnet/core/runtime:3.1

mcr.microsoft.com/dotnet/core/runtime-deps:3.1-buster-slim

debian:buster-slim

Wo speichert man Docker-Images?





- Docker Hub
- Azure Container Registry
- …

Container Registry

Pull Images

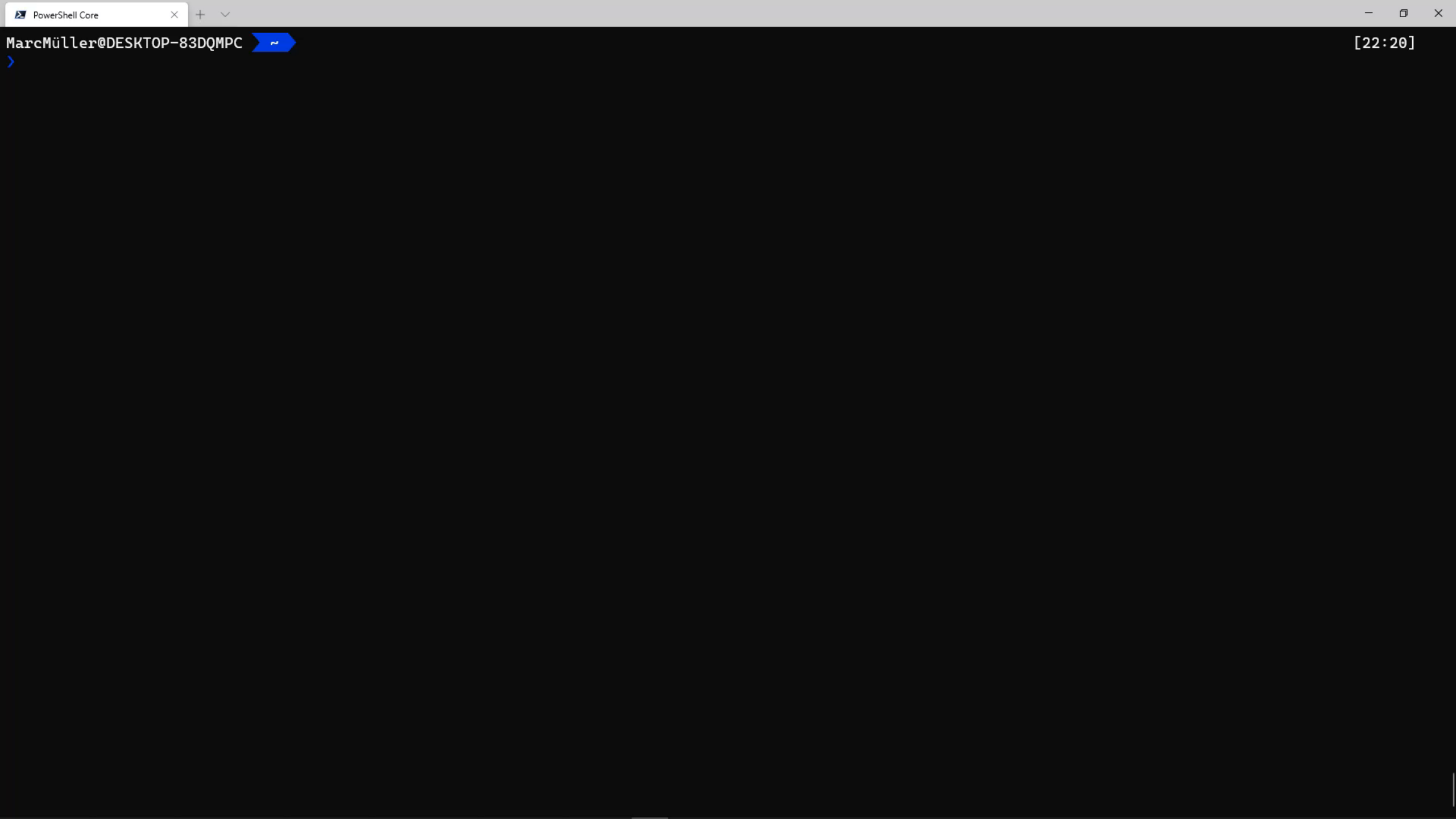
Login to your registry	docker login <registry> -u <user> -p <pwd>
Pull the desired image	docker pull <registry>/<repository>:<tag>
Run the container	docker run…

Push Images

Login to your registry	docker login <registry> -u <user> -p <pwd>
Build the image	docker build –t <reponame>:<tag> <dockerfile>
Push the image	docker push <registry>/<reponame>:<tag>

Docker "Hello World"

Demo


PowerShell Core
MarcMüller@DESKTOP-83DQMPC ~
[22:20]

Docker Demo

- ✔ Docker Layer
- ✔ Schneller Start
- ✔ Image und Container

Docker-Image erstellen

Am Beispiel von ASP.NET Core

Dockerfile – Einfache Variante

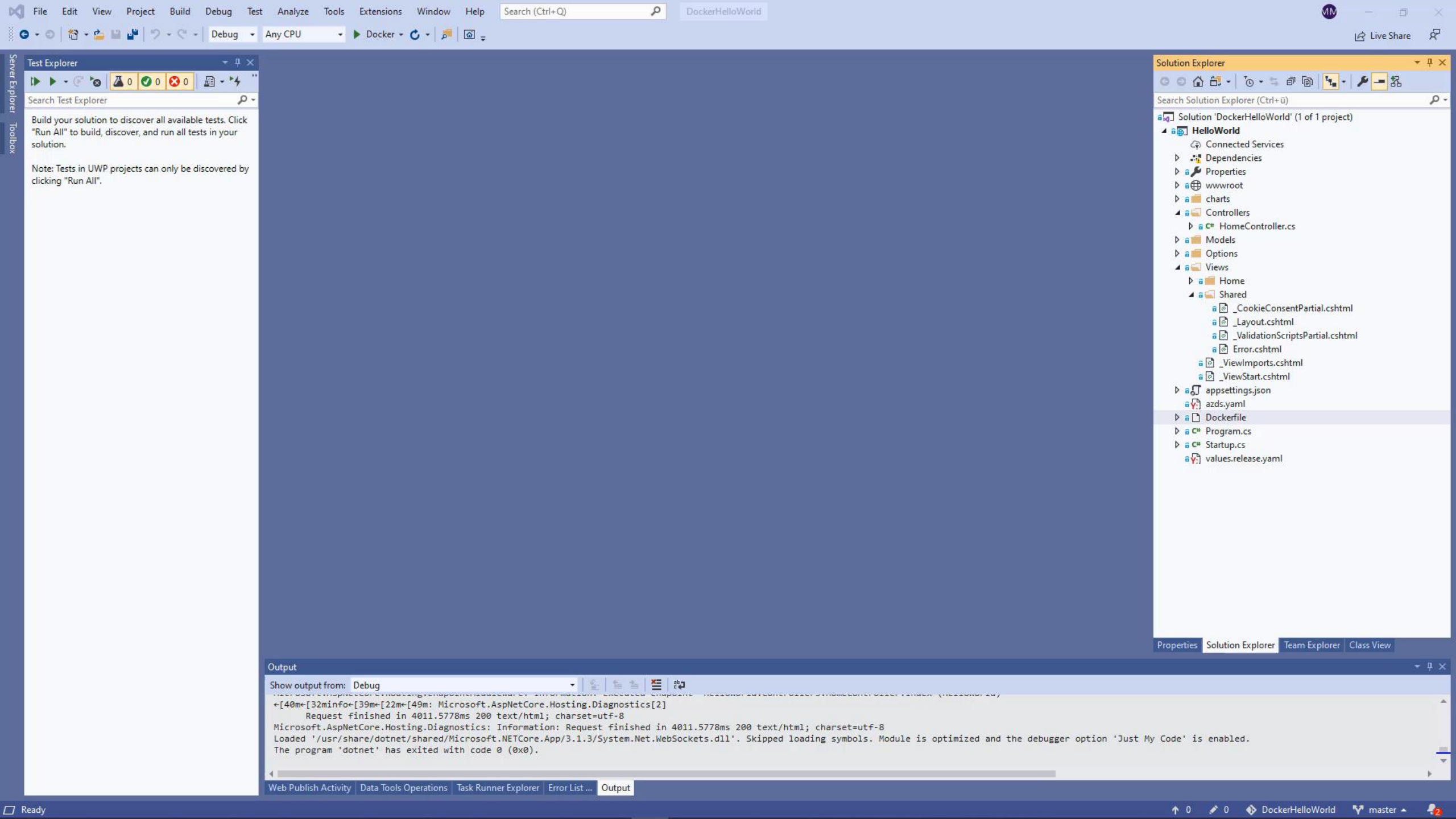
```
FROM mcr.microsoft.com/dotnet/core/aspnet:3.1-buster-slim
WORKDIR /app
COPY output/app .
ENTRYPOINT ["dotnet", "HelloWorld.dll"]
```

Dockerfile – Mit CI-Umgebung

```
FROM mcr.microsoft.com/dotnet/core/aspnet:3.1-buster-slim AS base
WORKDIR /app
EXPOSE 80

FROM  mcr.microsoft.com/dotnet/core/sdk:3.1-buster AS build
WORKDIR /src
COPY ["HelloWorld/HelloWorld.csproj", "HelloWorld/"]
RUN dotnet restore "HelloWorld/HelloWorld.csproj"
COPY . .
WORKDIR "/src/HelloWorld"
RUN dotnet build "HelloWorld.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "HelloWorld.csproj" -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "HelloWorld.dll"]
```

Azure Pipelines mit Docker Build

DEMO


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
DockerHelloWorld
Debug
Any CPU
Docker
Live Share
Server Explorer
Toolbox
Test Explorer
Search Test Explorer
Build your solution to discover all available tests. Click "Run All" to build, discover, and run all tests in your solution.
Note: Tests in UWP projects can only be discovered by clicking "Run All".
Solution Explorer
Search Solution Explorer (Ctrl+ü)
Solution 'DockerHelloWorld' (1 of 1 project)
HelloWorld
Connected Services
Dependencies
Properties
wwwroot
charts
Controllers
HomeController.cs
Models
Options
Views
Home
Shared
_CookieConsentPartial.cshtml
_Layout.cshtml
_ValidationScriptsPartial.cshtml
Error.cshtml
_ViewImports.cshtml
_ViewStart.cshtml
appsettings.json
azds.yaml
Dockerfile
Program.cs
Startup.cs
values.release.yaml
Properties
Solution Explorer
Team Explorer
Class View
Output
Show output from: Debug
←[40m←[32minfo←[39m←[22m←[49m: Microsoft.AspNetCore.Hosting.Diagnostics[2]
Request finished in 4011.5778ms 200 text/html; charset=utf-8
Microsoft.AspNetCore.Hosting.Diagnostics: Information: Request finished in 4011.5778ms 200 text/html; charset=utf-8
Loaded '/usr/share/dotnet/shared/Microsoft.NETCore.App/3.1.3/System.Net.WebSockets.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.
The program 'dotnet' has exited with code 0 (0x0).
Web Publish Activity
Data Tools Operations
Task Runner Explorer
Error List ...
Output
Ready
DockerHelloWorld
master

Azure Pipelines mit Docker Build

- ✔ CI-Prozessdefinition in Dockerfile
- ✔ Azure Pipeline erstellt Docker-Image
- ✔ Docker-Registry zur Speicherung von Docker-Images

Docker als Build-Umgebung

Docker und YAML-Pipelines

Herausforderungen für CI/CD

Configuration as Code

Code Änderungen haben Einfluss auf die Pipeline Definition

Erstellen von einer sehr alten Version → Build Umgebung?

Validierung von Änderungen an der Pipeline / Build Umgebung

Herausforderungen für Build Umgebungen

Speicherung von VM-Images benötigt viel Speicherplatz

Erstellen von virtuellen Maschinen dauert

Snapshots von Build-VMs sind nicht in den Standard-Pipeline-Prozess integriert

Wo speichern wir die VM Images?

Container

Reduzieren die Komplexität für den Unterhalt von Build-Umgebungen

Verwendet eine «Kochbuch-Anleitung» (Dockerfile) um die Umgebungen (Images) zu erstellen

Layering → Wiederverwendbarkeit und ‘Separation of Concerns’

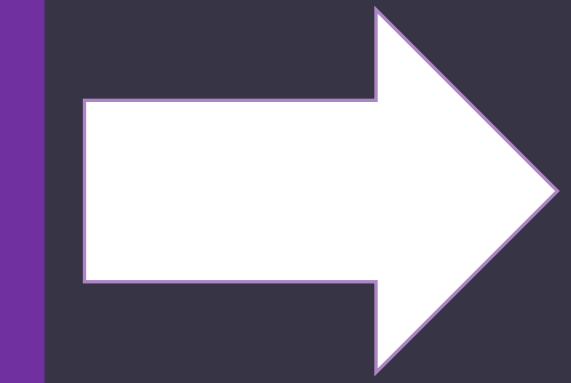
Schneller Start

Geringer Platzbedarf

Container Jobs

Pipeline Agent

Pipeline Job



Pipeline Agent

Versionierung des gesamten Build Environments

Git

Container Registry

V1

CI YAML

reference: <registry>/<repository>:<tag>

i.e.: 4tectureregistry.azurecr.io/custombuildenv:1.0.0

Container Image

V2

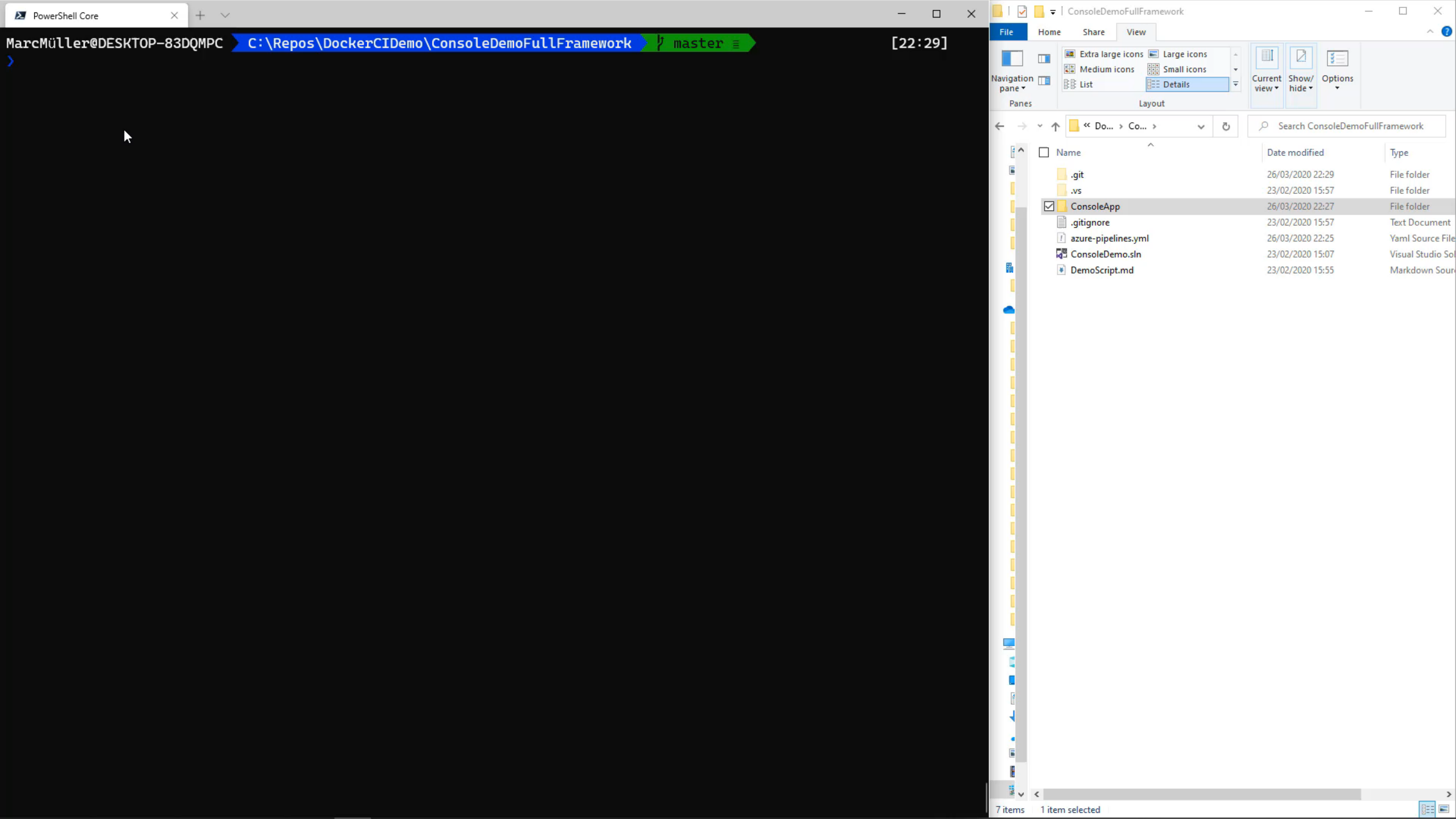
CI YAML

i.e.: 4tectureregistry.azurecr.io/custombuildenv:2.0.0

Container Image

.NET Full Framework-Umgebung in Docker

DEMO


PowerShell Core
MarcMüller@DESKTOP-83DQMPC C:\Repos\DockerCIDemo\ConsoleDemoFullFramework master [22:29]
ConsoleDemoFullFramework
File Home Share View
Navigation pane
Panes
Extra large icons
Large icons
Medium icons
Small icons
List
Details
Layout
Current view
Show/ hide
Options
Do... Co...
Search ConsoleDemoFullFramework
Name
Date modified
Type
.git 26/03/2020 22:29 File folder
.vs 23/02/2020 15:57 File folder
ConsoleApp 26/03/2020 22:27 File folder
.gitignore 23/02/2020 15:57 Text Document
azure-pipelines.yml 26/03/2020 22:25 Yaml Source File
ConsoleDemo.sln 23/02/2020 15:07 Visual Studio Sol
DemoScript.md 23/02/2020 15:55 Markdown Sour
7 items
1 item selected

.NET Full Framework-Umgebung in Docker

- ✔ Interaktiver Start der lokalen Build Umgebung im Container
- ✔ Interaktion mit Filesystem
- ✔ Kompilierung der Visual Studio Solution

Azure Pipelines Container Job

DEMO

ConsoleDemoFullFramework - R | ConsoleDemoFullFramework - P

https://dev.azure.com/4tecture-demo/DockerCIDemo/_git/ConsoleDemoFullFramework

Not syncing

4tecture-demo / DockerCIDemo / Repos / Files / ConsoleDemoFullFramework

Search

Users may experience build delays due to capacity constraints due to increased demand from the global pandemic. See this link for more detail: link

ConsoleDemoFullFramework

- ConsoleApp
- .gitignore
- azure-pipelines.yml
- ConsoleDemo.sln
- DemoScript.md

master / Type to find a file or folder...

Files

succeeded | Clone

Contents | History

Name	Last change	Commits
ConsoleApp	Feb 23	4a40186b initial demo applicationMarc Müller
.gitignore	Feb 23	66692e92 .gitignore (VisualStudio) filesMarc Müller
azure-pipelines.yml	Feb 26	5c0af2a6 Update azure-pipelines.yml for Azure PipelinesMarc Müller
ConsoleDemo.sln	Feb 23	4a40186b initial demo applicationMarc Müller
DemoScript.md	Feb 23	4a40186b initial demo applicationMarc Müller

Azure Pipelines Container Job

- ✔ Definition des Build-Images
- ✔ Alle Tasks laufen in Container Job
- ✔ Keine Veränderung der bestehenden Tasks

Erweiterte Docker-Szenarien

Docker und YAML Pipelines

Dockerfile – Azure Artifacts

```
docker build -f HelloWorld\Dockerfile -t 4tecture/HelloWorld . –build-arg PAT=<token>
```

```
FROM microsoft/dotnet:2.1-aspnetcore-runtime AS base
WORKDIR /app
EXPOSE 80

FROM microsoft/dotnet:2.1-sdk AS build
WORKDIR /src

ARG PAT
RUN wget -qO- https://raw.githubusercontent.com/Microsoft/artifacts-credprovider/master/helpers/installcredprovider.sh | bash
ENV NUGET_CREDENTIALPROVIDER_SESSIONTOKENCACHE_ENABLED true
ENV VSS_NUGET_EXTERNAL_FEED_ENDPOINTS "{\"endpointCredentials\": [{\"endpoint\":\"https://pkgs.dev.azure.com/4tecture-demo/_packagingSamples/nuget/v3/index.json\", \"password\":\"${PAT}\"}]} "

COPY ["HelloWorld/HelloWorld.csproj", "HelloWorld/"]
RUN dotnet restore -s "https://pkgs.dev.azure.com/4tecture-demo/_packagingSamples/nuget/v3/index.json" -s "https://api.nuget.org/v3/index.json" "HelloWorld/HelloWorld.csproj"
COPY . .
WORKDIR "/src/HelloWorld"
RUN dotnet build "HelloWorld.csproj" -c Release -o /app

FROM build AS publish
RUN dotnet publish "HelloWorld.csproj" -c Release -o /app

FROM base AS final
WORKDIR /app
COPY --from=publish /app .
ENTRYPOINT ["dotnet", "HelloWorld.dll"]
```

Dockerfile – UT & Results

```
FROM microsoft/dotnet:2.1-aspnetcore-runtime AS base
WORKDIR /app
EXPOSE 80

FROM microsoft/dotnet:2.1-sdk AS build
WORKDIR /src
COPY ["HelloWorld/HelloWorld.csproj", "HelloWorld/"]
RUN dotnet restore "HelloWorld/HelloWorld.csproj"
COPY . .
WORKDIR "/src/HelloWorld"
RUN dotnet build "HelloWorld.csproj" -c Release -o /app
```

```
FROM build AS test
LABEL test=true
RUN dotnet tool install dotnet-reportgenerator-globaltool --version 4.0.6 --tool-path /tools --configfile ../nuget.official-only.config
RUN dotnet test --results-directory /testresults --logger "trx;LogFileName=test_results.xml" /p:CollectCoverage=true /p:CoverletOutputFormat=cobertura
/p:CoverletOutput=/testresults/coverage/ ../HelloWorld/HelloWorld.Unit.Tests.csproj
RUN /tools/reportgenerator "-reports:/testresults/coverage/coverage.cobertura.xml" "-targetdir:/testresults/coverage/reports" "-
reporttypes:HTMLInline;HTMLChart"
```

```
FROM build AS publish
RUN dotnet publish "HelloWorld.csproj" -c Release -o /app

FROM base AS final
WORKDIR /app
COPY --from=publish /app .
ENTRYPOINT ["dotnet", "HelloWorld.dll"]
```

```
export id=$(docker images --filter "label=test=true" -q | head -1)

docker create --name testcontainer $id

docker cp testcontainer:/testresults ./testresults

docker rm testcontainer
```

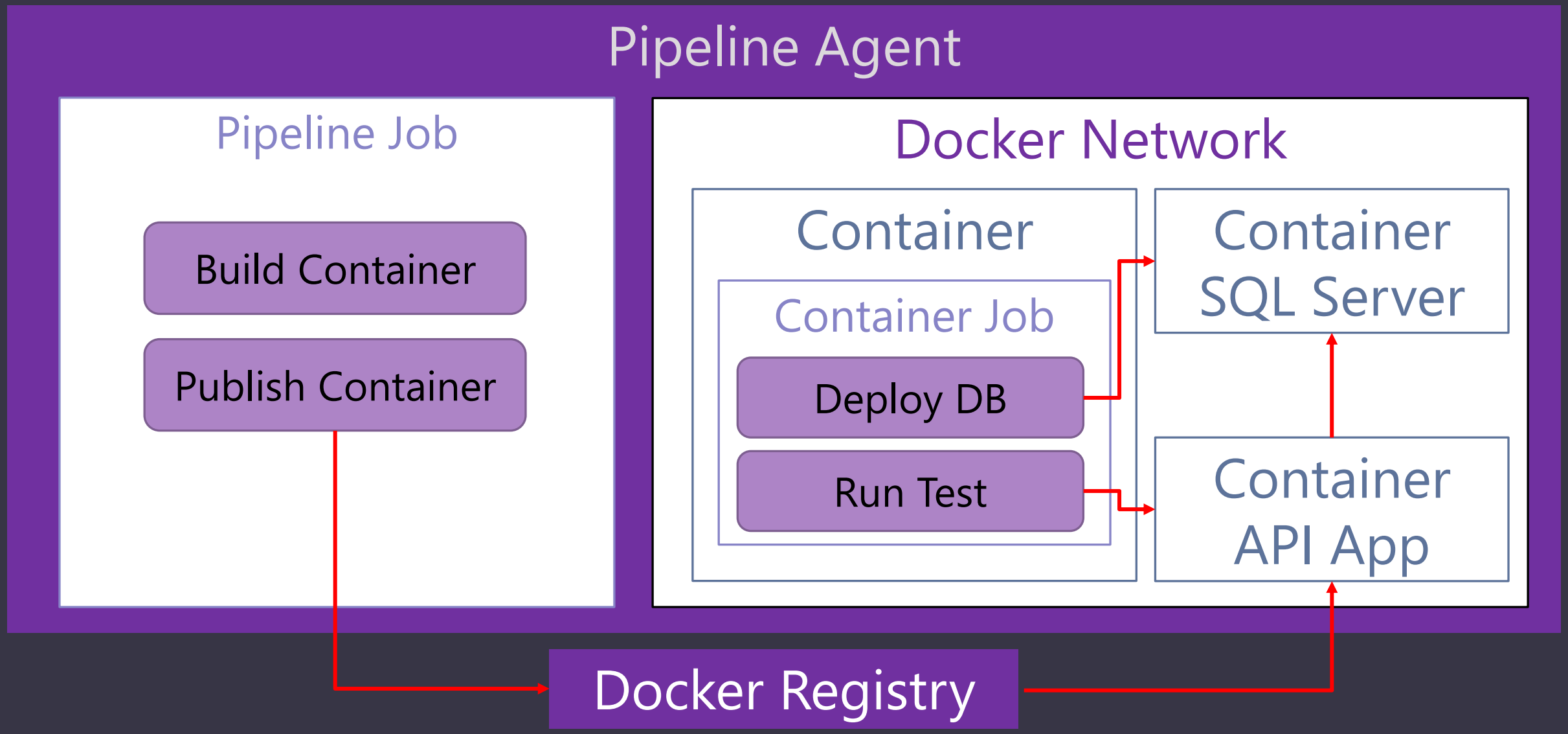
Anforderungen an moderne CI-Prozesse

Komplexes Systemtest-Environment

Shift-Left / Fast Feedback

Testen von mehreren Versionen / Konfigurationen

Bringing all together – Shift left



Erweiterte Docker-Szenarien

DEMO

DevFunApi-CI - Pipelines | buildAndPublishDockerImage.yml | Dockerfile - Repos

https://dev.azure.com/4tecture-demo/DockerCIDemo/_apps/hub/ms.vss-build-web.ci-designer-hub?pipelineId=31&branch=master

Not syncing

4tecture-demo / DockerCIDemo / Pipelines / Builds / DevFunApi-CI / YAML

Search

DevFunApi-CI

Variables | Save

master | DevFun / pipelines/**devfunapi-ci.yml** *

Show assistant

```yaml
stages:
- stage: Build
  jobs:
  - job: CI_Build
    displayName: 'Build the Api'
    pool:
      #vmImage: 'ubuntu-latest'
      name: Default
    variables:
      DockerFile: DevFun.Api/DevFun.Api/Dockerfile
      BuildContext: DevFun.Api
      ChartPath: DevFun.Api/DevFun.Api/charts/devfunapi
      ArtifactSourceFolder: DevFun.Api/DevFun.Api
      ArtifactName: DevFunApi
    steps:
    - template: templates/buildVersioning.yml
    - template: templates/buildAndPublishDockerImage.yml
    - template: templates/createHelmPackage.yml
```

https://dev.azure.com/4tecture-demo/DockerCIDemo/_build?definitionId=31&_a=summary

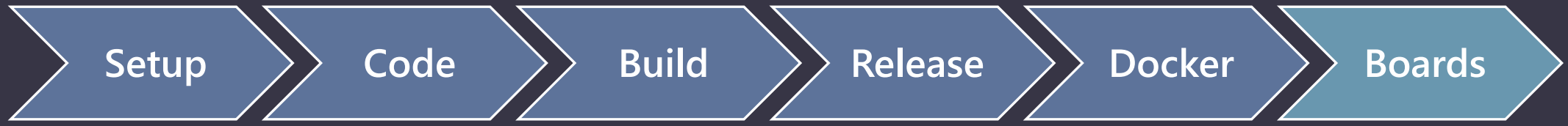
Erweiterte Docker-CI-Szenarien

- ✔ Kombination von Container Builds und Container Jobs
- ✔ Testumgebungen während dem CI-Prozess
- ✔ Container Jobs und Service Containers
- ✔ Netzwerk-Isolation mit Docker Networking

Danke!

Azure DevOps Webinare

- ✔ Folge 1: Schnellstart mit Azure DevOps
- ✔ Folge 2: Moderne Versionsverwaltung mit Git und Pull Requests
- ✔ Folge 3: Build-Automatisierung und Continuous Integration (CI)
- ✔ Folge 4: Deployment-Automatisierung mit Release Pipelines
- ✔ Folge 5: Docker-Container und Azure DevOps
- 👉 Folge 6: Agiles Arbeiten mit Azure Boards



Ihre Referenten

Haben Sie noch Fragen zu Azure DevOps?

Weiter geht es unter:
https://go.nenoloje.com/**msvtw2020**

Neno Loje
www.nenoloje.com

Neno Loje

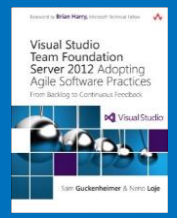
Freier Berater und Erfolgs-Coach für
agile Softwareentwicklungsteams

MVP Microsoft Most Valuable Professional

TFS Camp – Team Foundation Server Camp

E-Mail: neno@nenoloje.de
Webseite: https://www.nenoloje.com
Schulungen: https://go.nenoloje.com/schulungen
Blog: https://go.nenoloje.com/blog
Twitter: @nenoloje

Neno Loje – teamsystempro.de



Marc Müller
www.4tecture.ch

Marc Müller

Principal Consultant
für DevOps, ALM, TFS / VS, .NET

MVP Microsoft Most Valuable Professional

4tecture

E-Mail: marc.mueller@4tecture.ch
Webseite: http:s//www.4tecture.ch
Schulungen: https://www.4tecture.ch/trainings
Blog: https://www.4tecture.ch/blog
Twitter: @muellermarc

4tecture GmbH
Aathalstrasse 84
CH-8610 Uster

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