Containerbasierte Entwicklung für .NET-Entwickler

Tobias Fenster

Special Day

Cloud-Native Business Applications

Thema	Sprecher	Datum, Uhrzeit
Cloud-Native: Definition, Praktiken & Patterns	Thorsten Hans, Christian Weyer	DO, 13. Oktober 2022, 09.00 bis 10.00
Containerbasierte Entwicklung für .NET-Entwickler	Tobias Fenster	DO, 13. Oktober 2022, 10.30 bis 11.30
Ein Wegweiser durch den Dschungel der Web-API Möglichkeiten	Gregor Biswanger	DO, 13. Oktober 2022, 12.00 bis 13.00
Cloud-native Microservices für alle: On-Premises oder Cloud – mit Dapr	Christian Weyer	DO, 13. Oktober 2022, 15.45 bis 16.45
Serverless-Container mit Azure Container Apps	Thorsten Hans	DO, 13. Oktober 2022, 17.15 bis 18.15

SPRECHER INTRO

TOBIAS FENSTER

Business

- Managing Partner bei 4PS Deutschland, Teil der 4PS Gruppe
  - Hersteller eines Cloud-basierten ERP für die Baubranche

Community

- Microsoft Regional Director und MVP für Azure und Business Central
- Docker Captain, Portainer und Traefik ambassador

Social und Blog

- tobiasfenster bei Twitter und LinkedIn
- tobiasfenster.io
- “Window on Technology” podcast

VORWISSEN

CONTAINER, WAS IST DAS?

- Session macht nur Spaß mit Basis-Wissen zu Containern
- Habt ihr das?

  Lasst es mich über menti.com mit Code 7973 9605 wissen

- Falls ja:

- Falls nein:

CONTAINER-BASICS

CONTAINER-BASICS

CONTAINER, IMAGES UND docker

- Was ist ein Container und ein (Container) Image?
  - Ein Image ist eine Vorlage mit der minimalen Menge an Betriebssystem, Bibliotheken und Anwendungsdateien, die benötigt werden.
  - Ein Container ist eine Instanz eines Image mit einer unveränderlichen Basis und seinen Änderungen darauf
  - Ein Container ist KEINE VM, du hast vor allem keine GUI und nichts, mit dem du dich per RDP verbinden kannst!
  - Images haben Tags für die Versionierung
- Was ist Docker?
  - Ein cross-platform Containerisierungsplattform, die Container im Tech-Mainstream bekannt gemacht und etabliert hat
  - Anbieter von Tools und Werkzeugen (Docker Desktop, docker compose, etc.) zur einfacheren Entwicklung im Kontext von Containern
  - Aktiver und wichtiger Maintainer des Open Source Tech-Stacks rund um Containerisierung

CONTAINER-BASICS

HOST, REGISTRY, REPOSITORY, OS

- Was ist ein (Container) Host?
  - Der (physische oder virtuelle) Rechner, auf dem die Container ausgeführt werden
- Was ist eine (Container) Registry?
  - Ein Ort, an dem du und andere Images hochladen (Push) und herunterladen (Pull) können. Docker bietet Docker Hub als kostenlose Registry an, aber es gibt auch viele andere wie die Azure Container Registry (ACR)
- Was ist ein (Container) Repository?
  - Ein Teil einer Registry (wie ein "Unterordner"), der einer Person oder einem Unternehmen gehört und die Images enthält.
- Welches Betriebssystem nutzen Container?
  - Linux und Windows sind beide möglich mit großen Überschneidungen, aber auch Unterschieden im Detail

CONTAINER-BASICS

HELLO WORLD: WIE KOMME ICH ZU MEINEM ERSTEN CONTAINER?

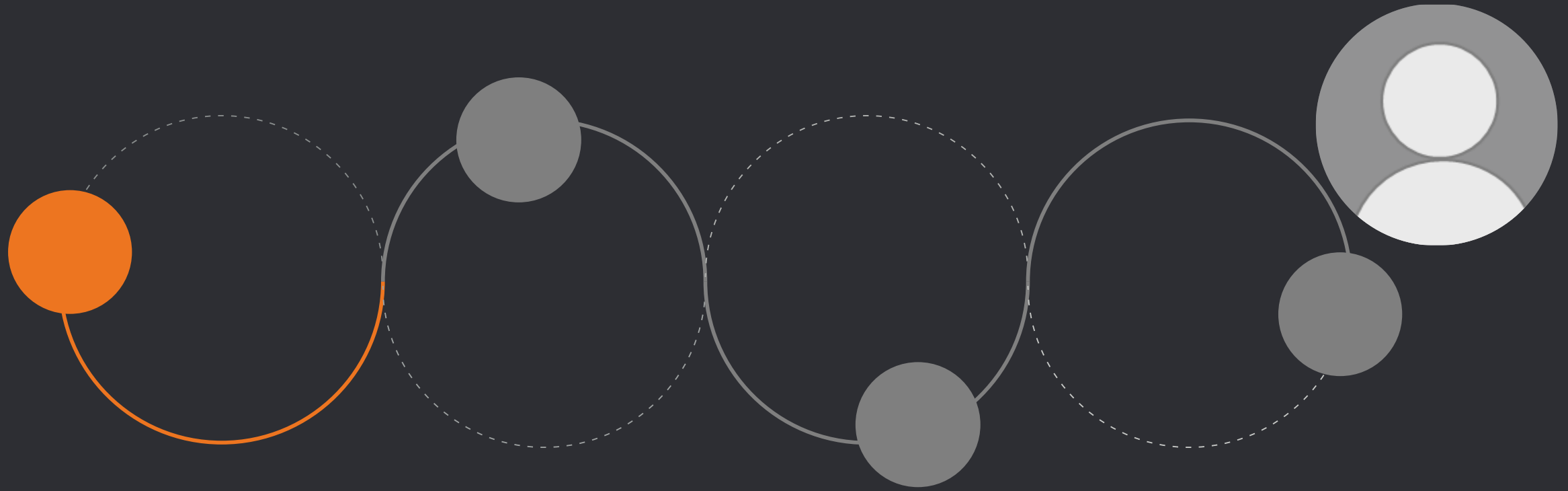


Image identifizieren

mcr.microsoft.com/windows/servercore/iis:windowsservercore-ltsc2022

CONTAINER-BASICS

HELLO WORLD: WIE KOMME ICH ZU MEINEM ERSTEN CONTAINER?

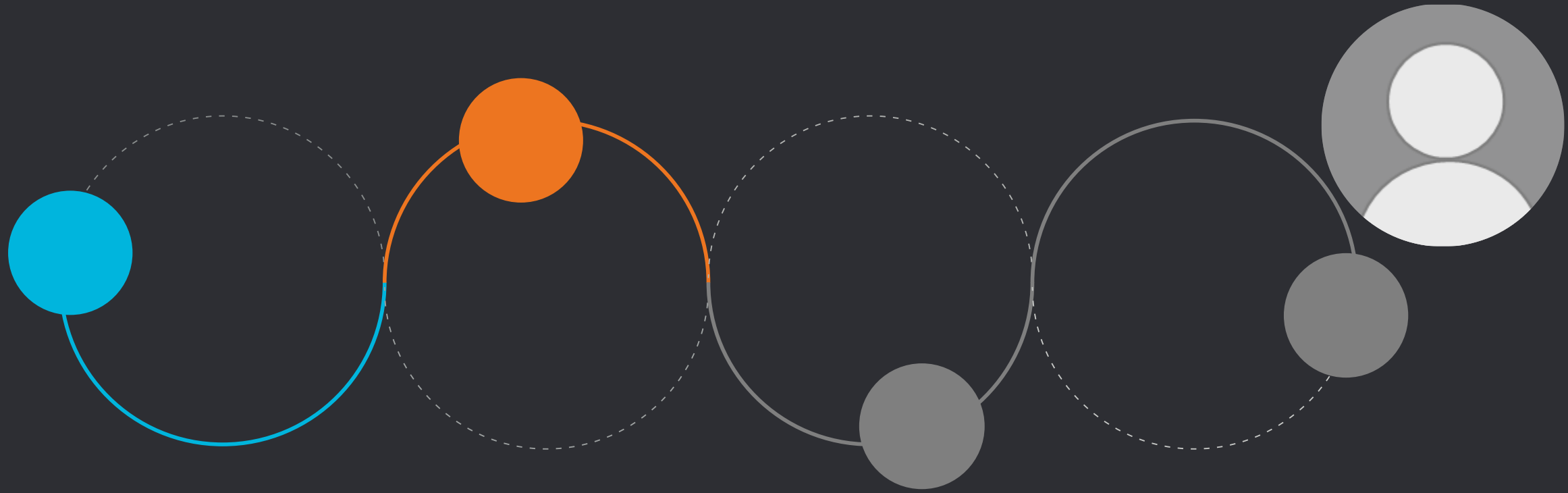


Image identifizieren → Image laden → → →

docker pull mcr.microsoft.com/windows/servercore/iis:windowsservercore-ltsc2022

CONTAINER-BASICS

HELLO WORLD: WIE KOMME ICH ZU MEINEM ERSTEN CONTAINER?

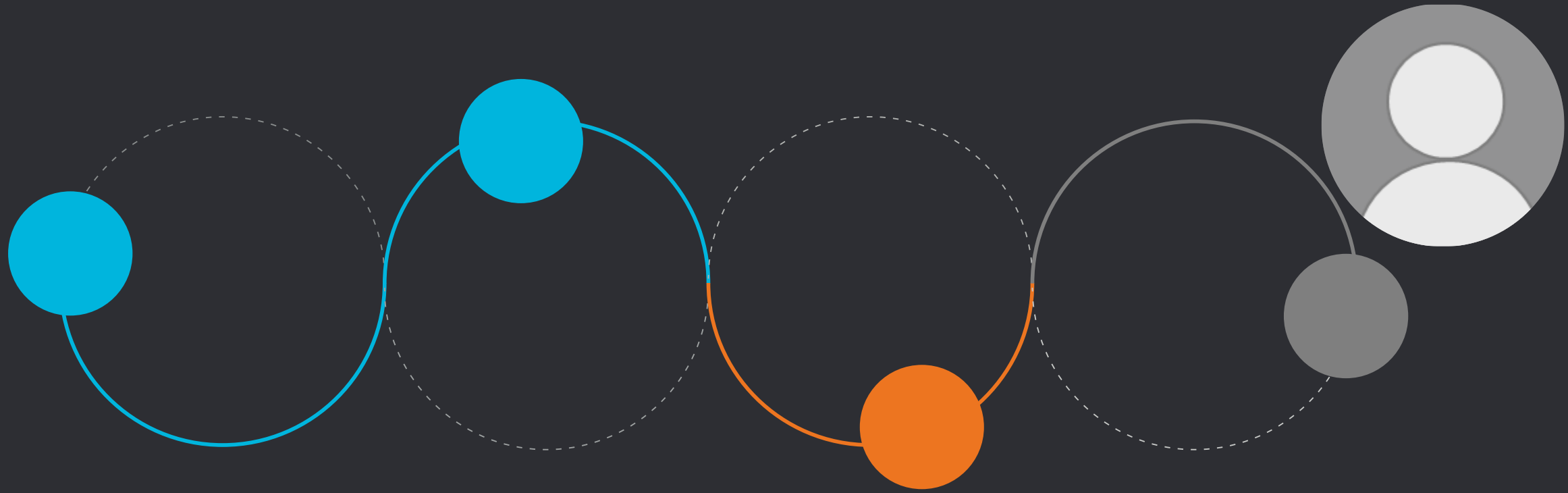


Image identifizieren → Image laden → Container erstellen →

```
docker create mcr.microsoft.com/windows/servercore/iis:windowsservercore-ltsc2022
```

CONTAINER-BASICS

HELLO WORLD: WIE KOMME ICH ZU MEINEM ERSTEN CONTAINER?

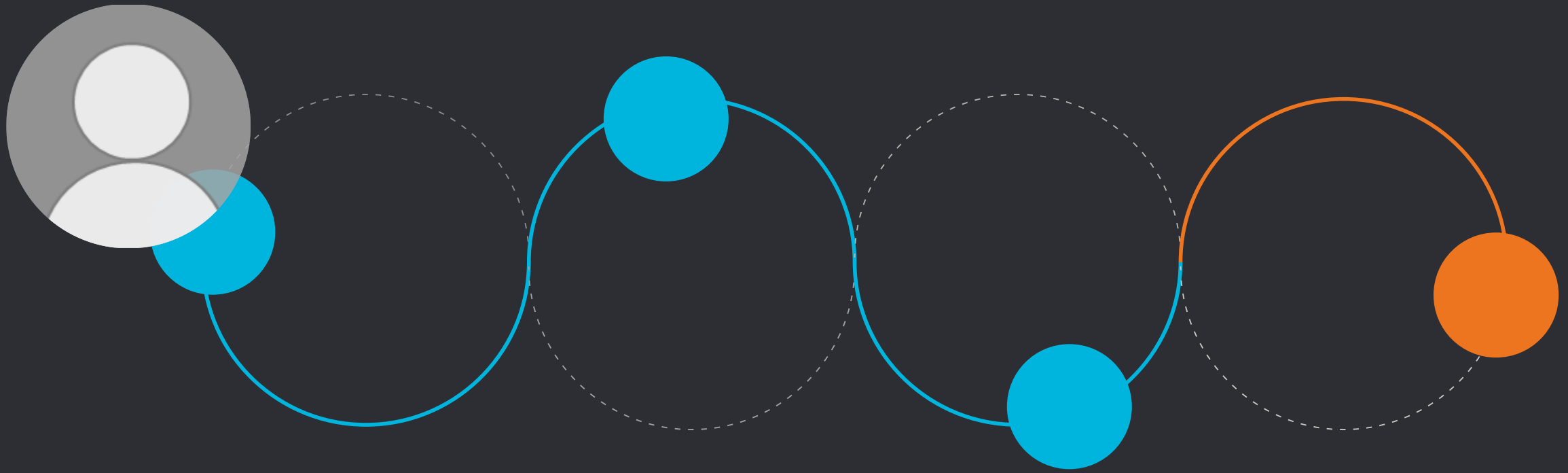


Image identifizieren → Image laden → Container erstellen → Container starten

```
docker start <id>
```

CONTAINER-BASICS

HELLO WORLD: WIE KOMME ICH ZU MEINEM ERSTEN CONTAINER?

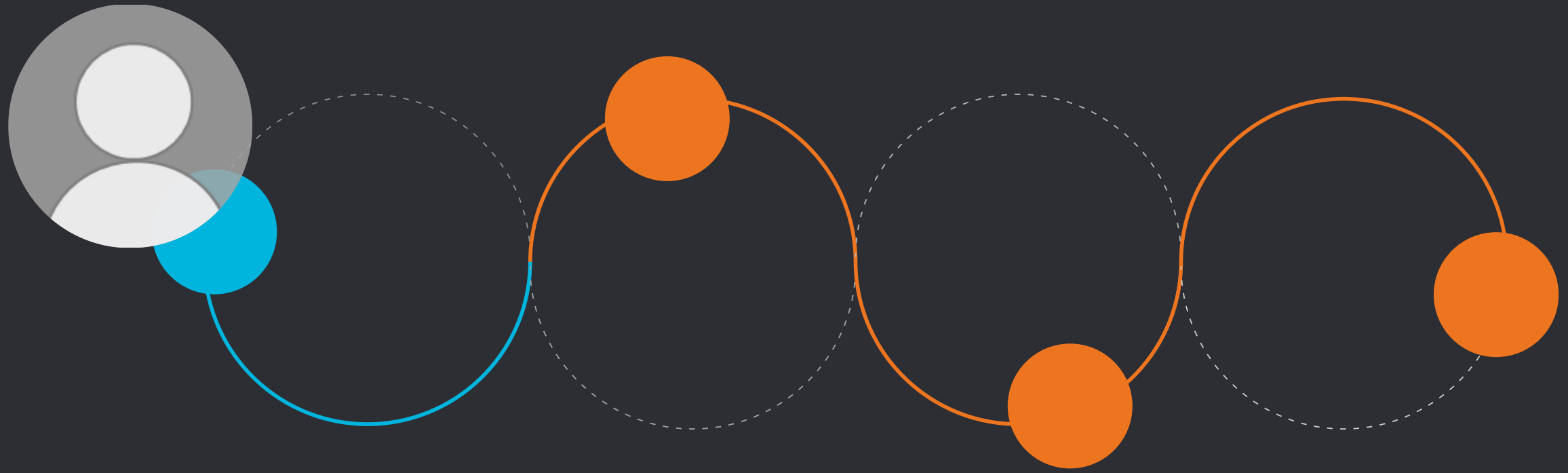
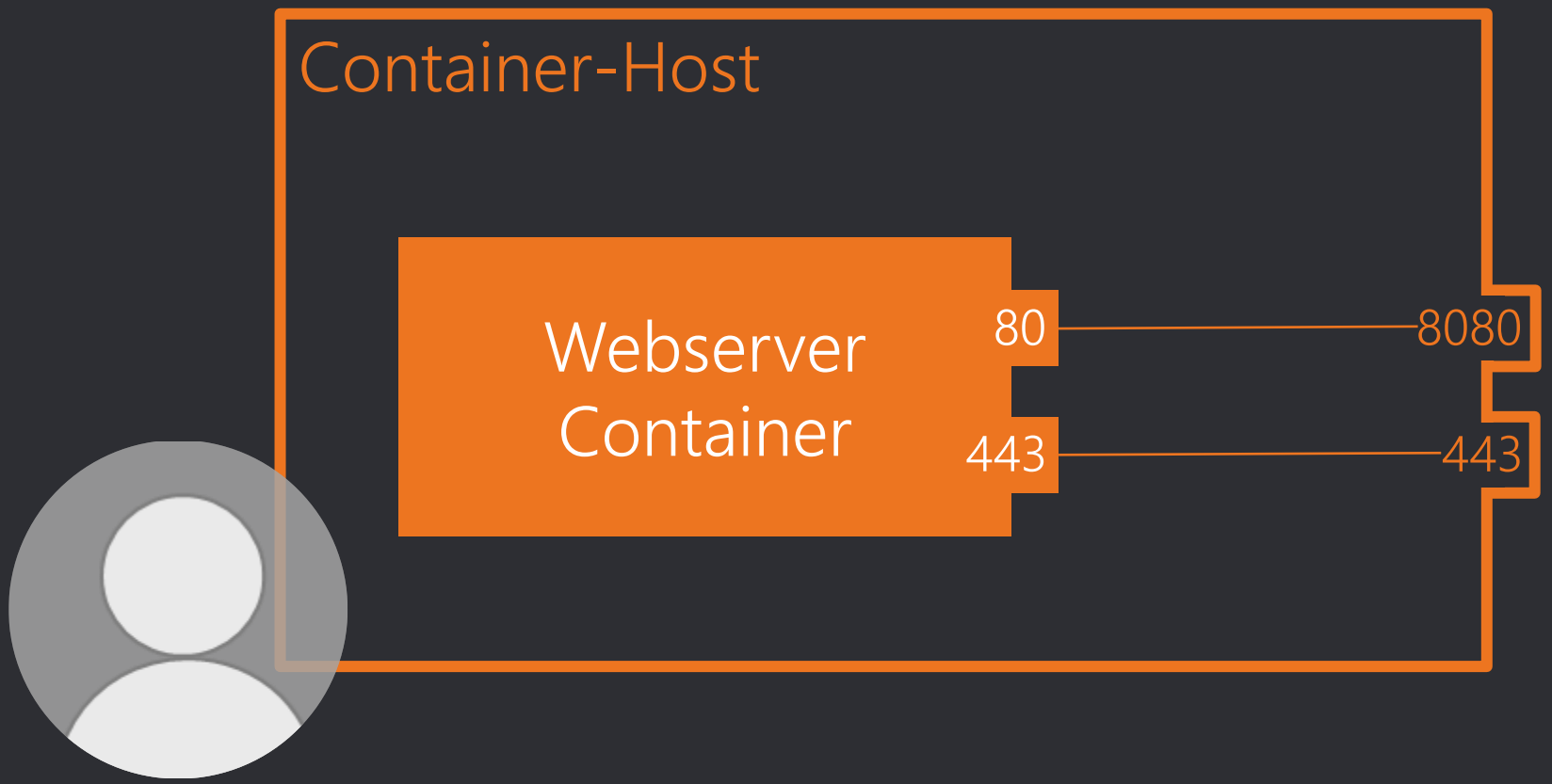


Image identifizieren → Image laden → Container erstellen → Container starten

```
docker run mcr.microsoft.com/windows/servercore/iis:windowsservercore-ltsc2022
```

CONTAINER-BASICS

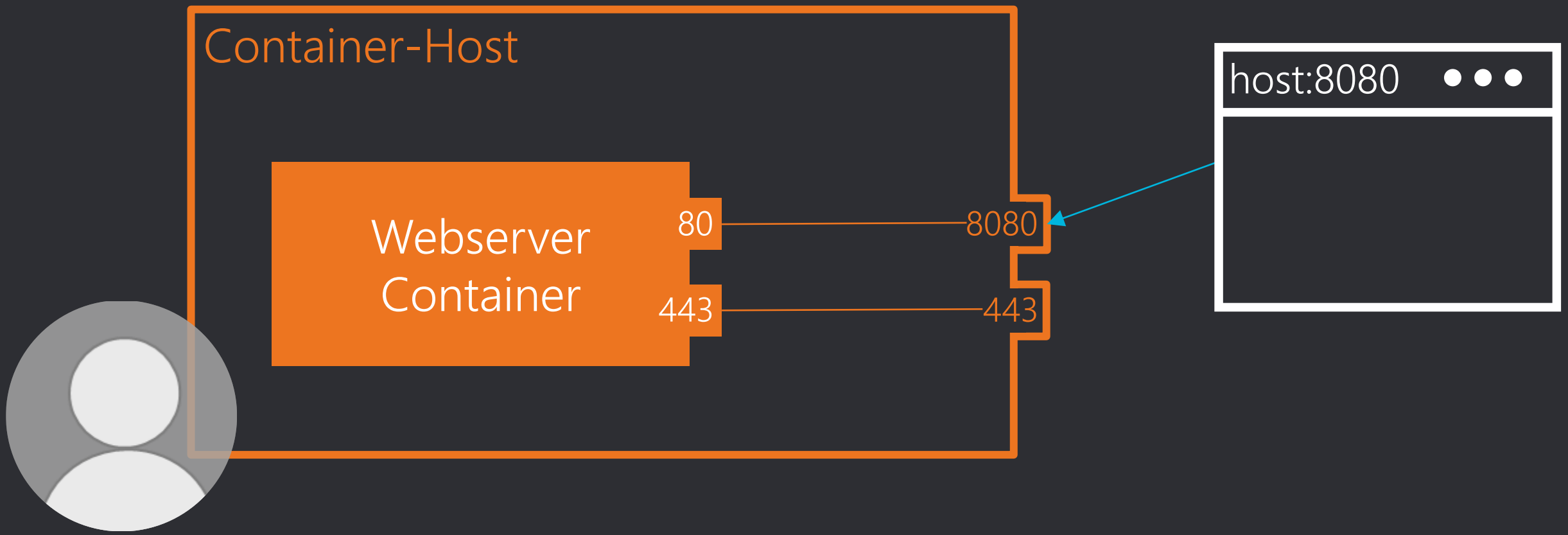
HELLO WORLD: WIE ERREICHE ICH MEINEN ERSTEN CONTAINER?



docker run -p 8080:80 -p 443:443 mcr.microsoft.com/...

CONTAINER-BASICS

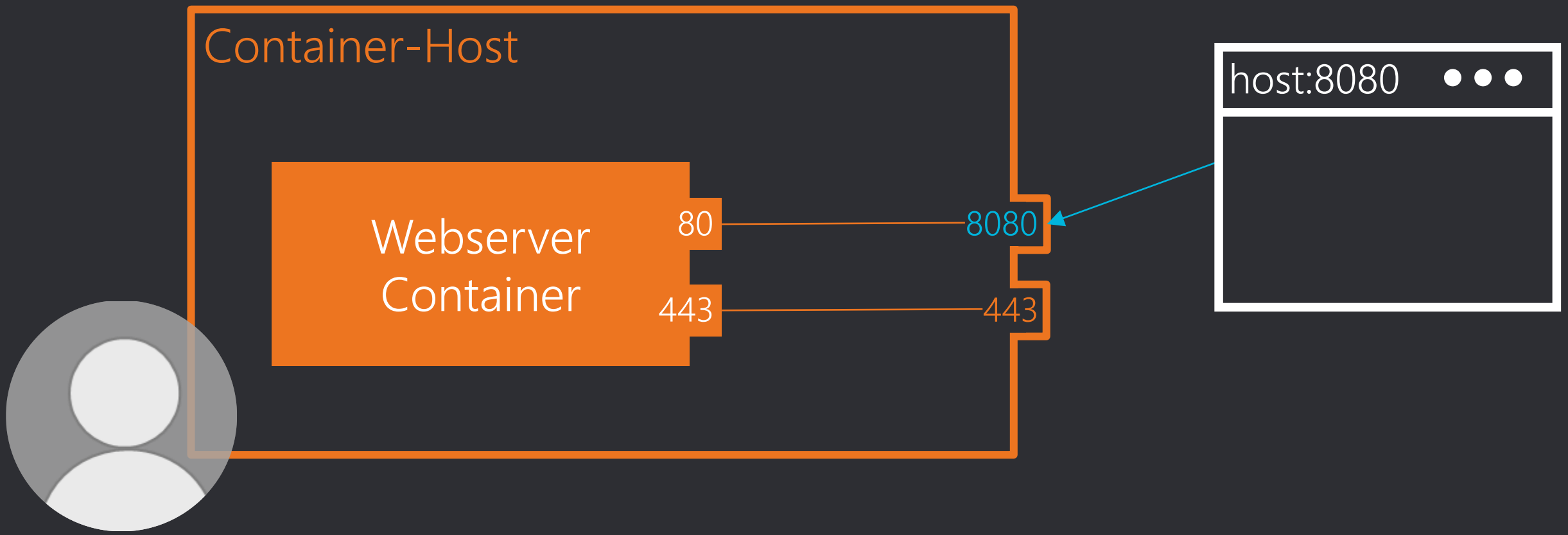
HELLO WORLD: WIE ERREICHE ICH MEINEN ERSTEN CONTAINER?



docker run -p 8080:80 -p 443:443 mcr.microsoft.com/...

CONTAINER-BASICS

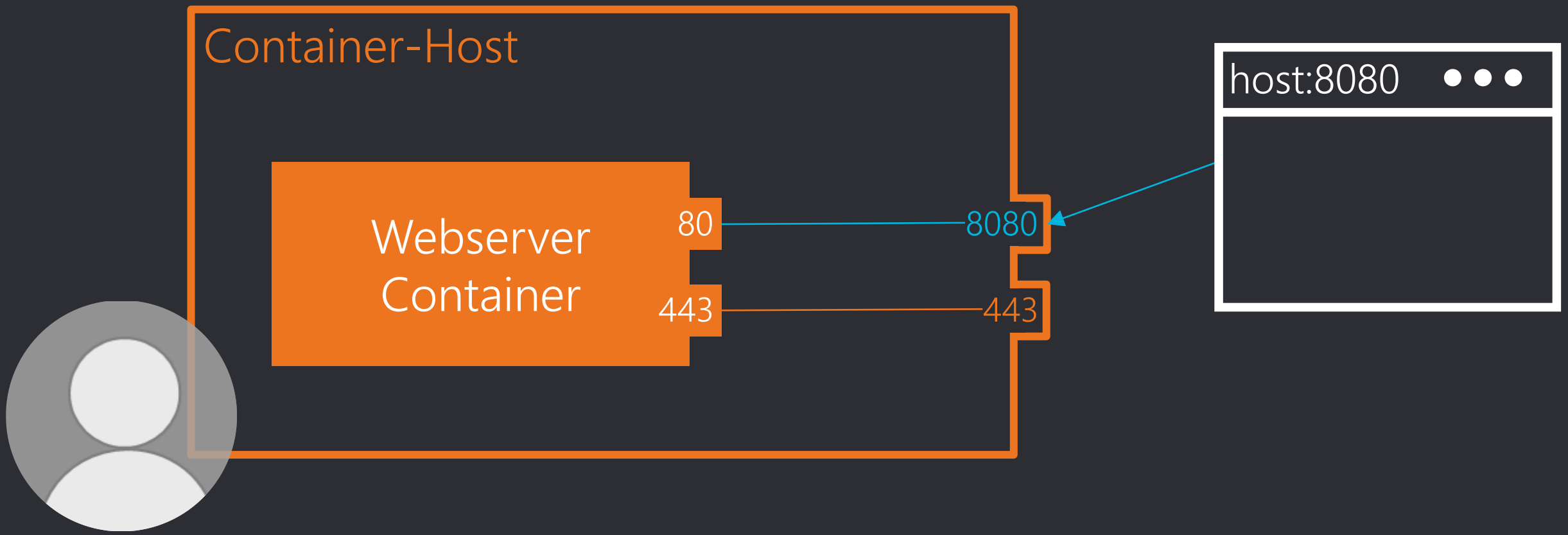
HELLO WORLD: WIE ERREICHE ICH MEINEN ERSTEN CONTAINER?



docker run -p 8080:80 -p 443:443 mcr.microsoft.com/...

CONTAINER-BASICS

HELLO WORLD: WIE ERREICHE ICH MEINEN ERSTEN CONTAINER?



docker run -p 8080:80 -p 443:443 mcr.microsoft.com/...

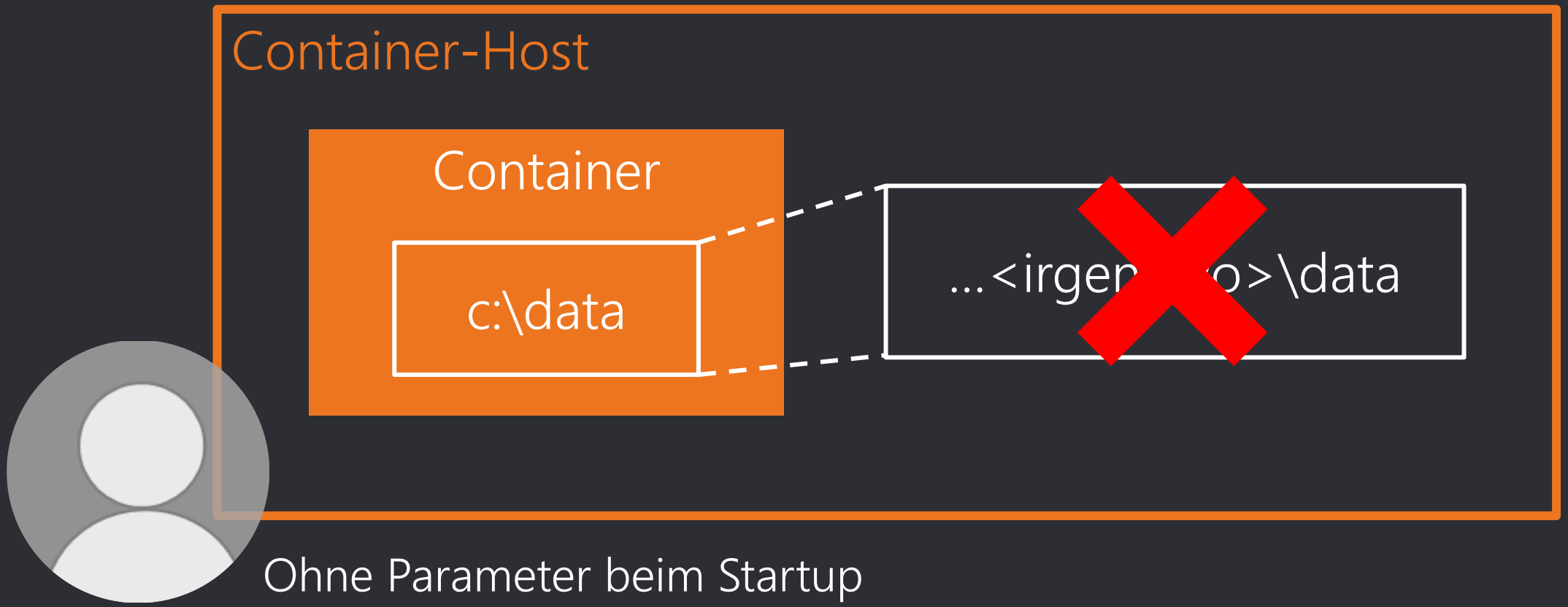
CONTAINER-BASICS

HELLO WORLD: WO SIND DIE DATEN DES CONTAINERS?

Container gelöscht → Daten gelöscht!

CONTAINER-BASICS

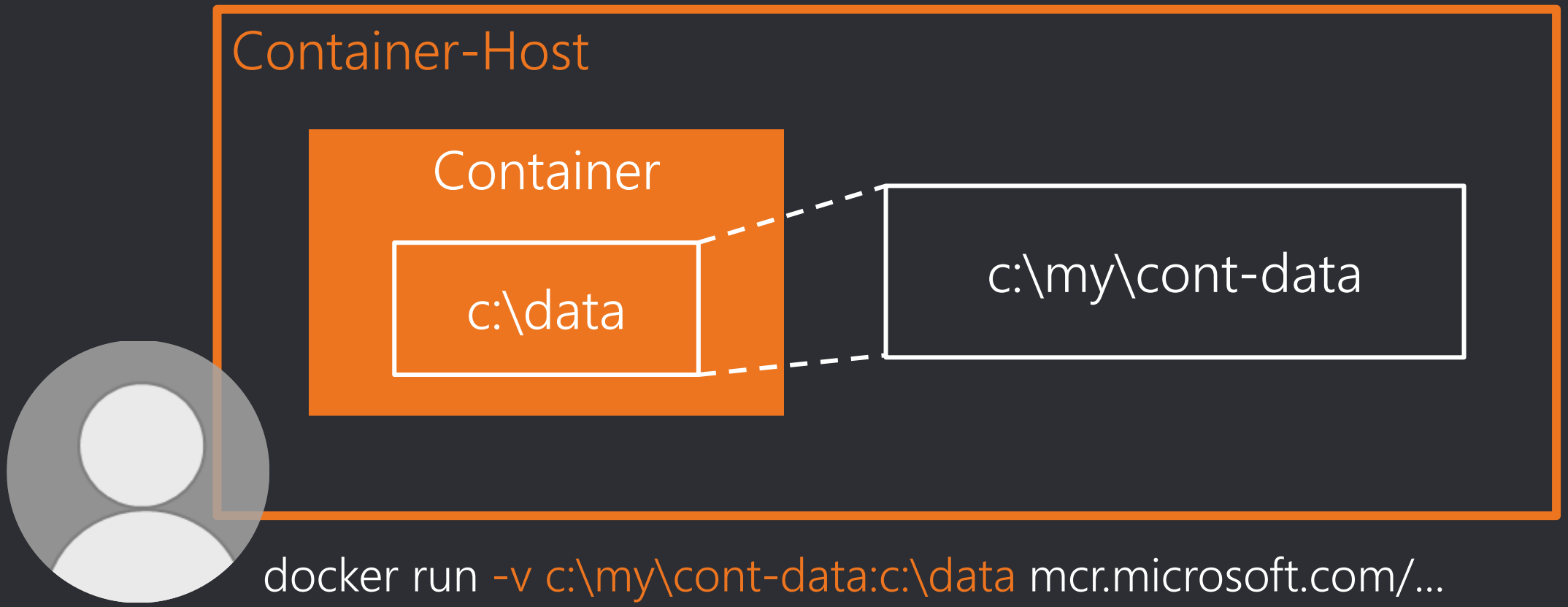
HELLO WORLD: WO SIND DIE DATEN DES CONTAINERS?



Ohne Parameter beim Startup

CONTAINER-BASICS

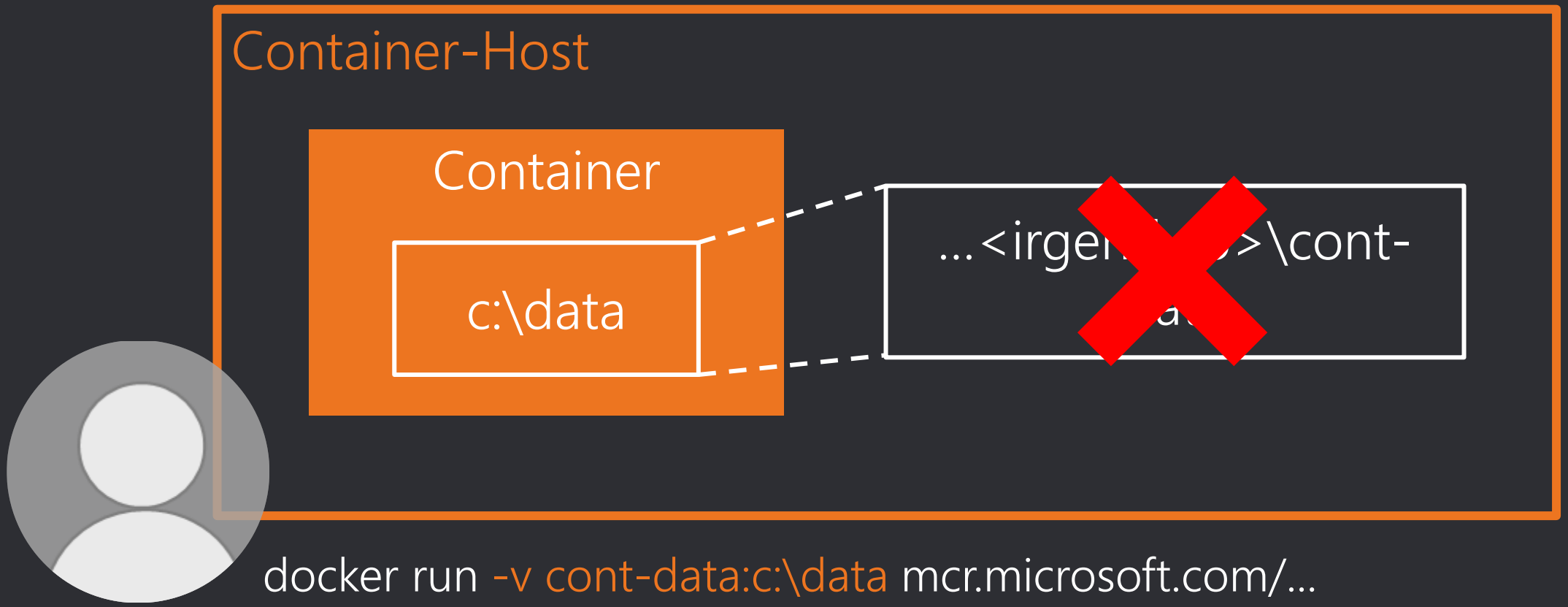
HELLO WORLD: WO SIND DIE DATEN DES CONTAINERS?



docker run -v c:\my\cont-data:c:\data mcr.microsoft.com/...

CONTAINER-BASICS

HELLO WORLD: WO SIND DIE DATEN DES CONTAINERS?



docker run -v cont-data:c:\data mcr.microsoft.com/...

Container gelöscht → Daten nicht gelöscht!

DEMO

CONTAINERBASIERTE ENTWICKLUNG FÜR .NET

CONTAINERBASIERTE ENTWICKLUNG

WARUM?

CONTAINERBASIERTE ENTWICKLUNG

WARUM?

- Sauber getrennte Entwicklungssysteme bei besserer Ressourcennutzung als z.B. mit VMs
  - Keine Versionskonflikte und Seiteneffekte
  - Kein „Vermüllen“ → einfach wegwerfen und neu erstellen
- Alle Abhängigkeiten, Tools, etc. inklusive Versionen in Konfigurationsdateien im Repo beschrieben
  - IaC-Ansatz für lokale Entwicklungsumgebungen
  - Kein Auseinanderdriften unterschiedlicher Entwickler
  - Klarer und einfacher Rollout von Änderungen im Entwicklungs-Stack
- Extrem schnelles Setup von Entwicklungsumgebungen
  - Damit auch extrem schnelles Onboarding neuer Entwicklung
  - Einfaches und sauberes Wechseln zwischen Projekten

CONTAINERBASIERTE ENTWICKLUNG

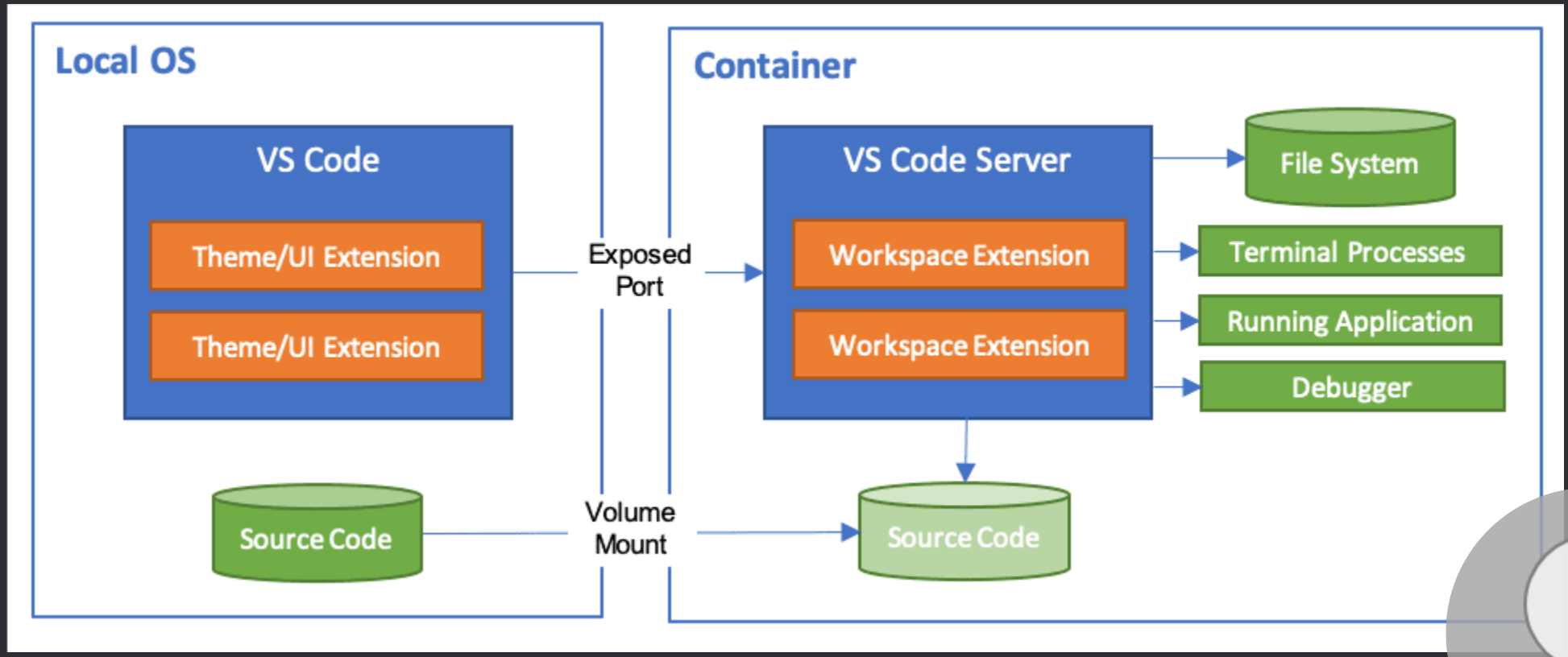
WIE?

- Visual Studio Code Entwicklungscontainer
- Containerisierte, konfigurierbare, lokale Entwicklungsumgebung
- Verbindung über VS Code
- Volle Funktionalität inkl. Extensions und Zugriff auf lokales und Offline-Support

→ Sehr gute Entwicklungsumgebung für alle Szenarien (außer Windows-basierte Entwicklung…)

- Idealer Ausgangspunkt für GitHub Codespaces (gleiche Technologie)

VISUAL STUDIO CODE ENTWICKLUNGSCONTAINER



Quelle: https://code.visualstudio.com/docs/remote/containers

DEMO

BEST PRACTICES FÜR PRODUKTIV-CONTAINER

Gold Microsoft Partner | 4PS

BEST PRACTICES

DEMO SZENARIO

- Projekt „project“ in devcontainer: dotnet new webapi
- Aufgaben:
  - „Containerisieren“
  - Container image optimieren
  - Vulnerabilities finden und fixen

BEST PRACTICES

WELCHES BASE IMAGE?

- Image zum Build: mcr.microsoft.com/dotnet/sdk
  - .NET CLI
  - .NET runtime
  - ASP.NET Core runtime
- Niemals latest!
  - Heute 6.0, bald 7.0…
  - Bei Abhängigkeiten auch z.B. 6.0.8
- An die Größe (Download, Attack Surface) denken → alpine

BEST PRACTICES

WIE DAS IMAGE AUFBAUEN?

- Anfangs nur .csproject kopieren und restore ausführen
  - Bei Änderungen am Code Cache nutzbar (Layering-Mechanismus in Images)
- Schneller, optimierter Publish
  - no-restore
  - PublishTrimmed, PublishReadyToRun, PublishSingleFile
- Multi-stage images
  - Image für Runtime:
    - mcr.microsoft.com/dotnet/runtime: .NET runtime
    - mcr.microsoft.com/dotnet/aspnet: ASP.NET Core runtime
  - Image für Runtime mit self-contained executable:
    - mcr.microsoft.com/dotnet/runtime-deps: nur notwendige dependencies

BEST PRACTICES

WAS KOMMT IN DEN BUILD CONTEXT?

- So wenig lokale Artefakte wie möglich im Build Context
  - .dockerignore analog .gitignore
  - Generieren lassen von VS Code…
  - Aber wir schauen rein

BEST PRACTICES

WIE GEHT ES DEM CONTAINER?

- Container können „health“ Status anzeigen
  - Definition ebenfalls im Dockerfile
- Wichtig z.B. bei Orchestratoren wie Kubernetes
  - Erkennen, wann ein Container ersetzt werden muss

BEST PRACTICES

WIE SOLL DER CONTAINER SICH VERHALTEN?

- Normale Konfigurationsmechanismen in .NET 1:1 auch auf Container abbildbar:
  - Aspnetcore_environment
  - Environment variables

BEST PRACTICES

WELCHER USER?

- Non-root on Linux
  - User anlegen, ohne Passwort
  - Zugriff auf Verzeichnis geben
- System = gMSA unter Windows

BONUS THEMA: MULTI-CONTAINER ENTWICKLUNG

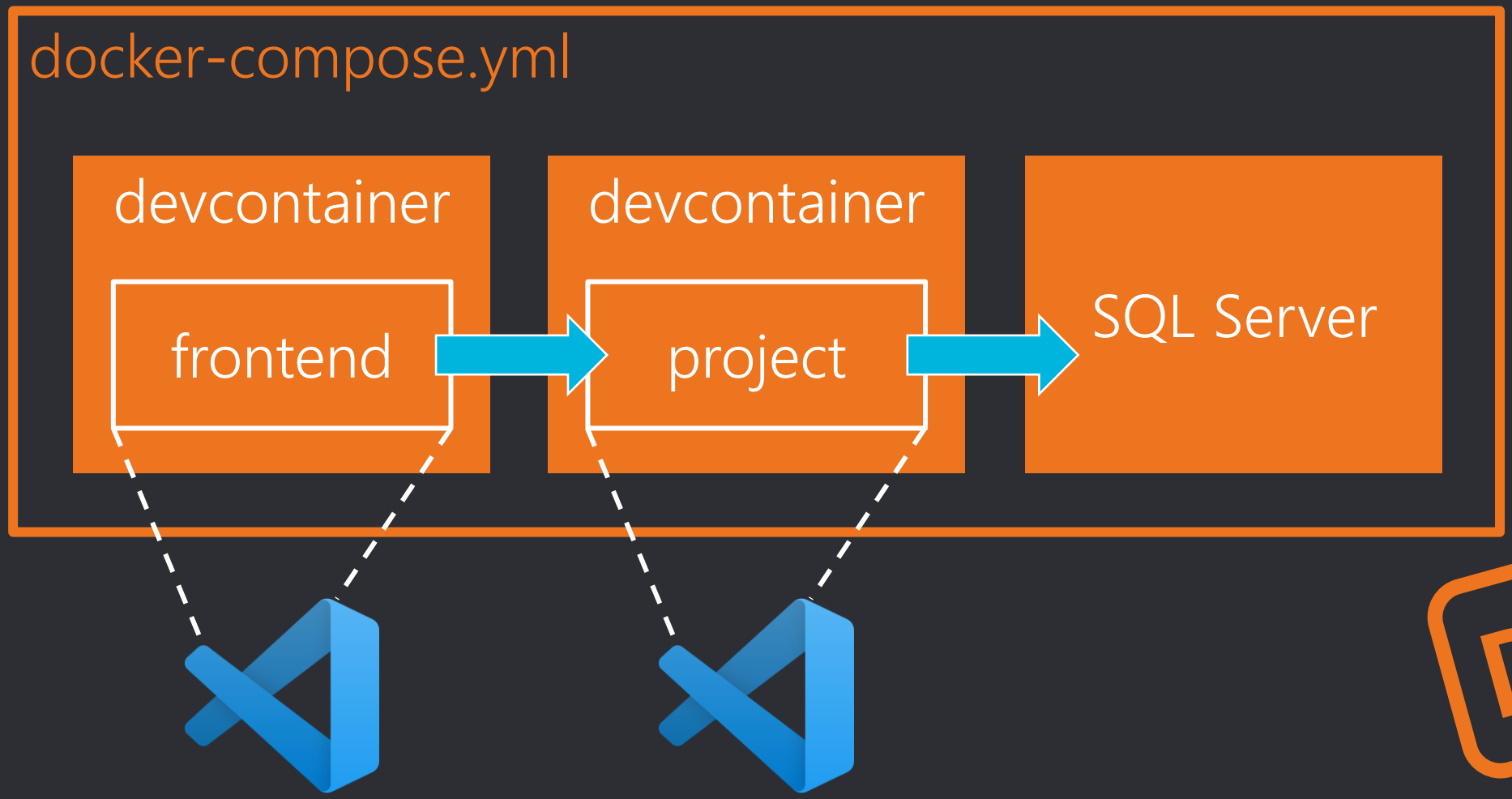
MULTI-CONTAINER ENTWICKLUNG

WAS WENN EIN CONTAINER NICHT REICHT?

- Szenario:
  - Backend wie bisher mit .NET webapi
  - Frontend mit Blazor
  - SQL Server
- Definiert in gemeinsamer docker-compose.yml
- Zwei VS Code Instanzen verbunden mit je einem Container

MULTI-CONTAINER ENTWICKLUNG

WAS WENN EIN CONTAINER NICHT REICHT?



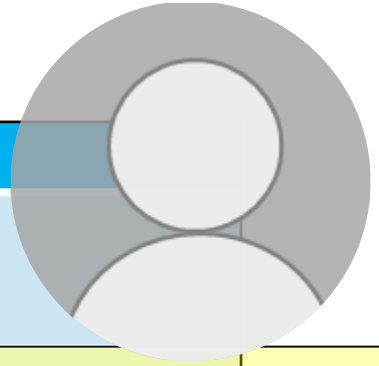
DEMO

VIELEN DANK!

WELCHE FRAGEN DARF ICH BEANTWORTEN?

Special Day

Cloud-Native Business Applications



Thema	Sprecher	Datum, Uhrzeit
Cloud-Native: Definition, Praktiken & Patterns	Thorsten Hans, Christian Weyer	DO, 13. Oktober 2022, 09.00 bis 10.00
Containerbasierte Entwicklung für .NET-Entwickler	Tobias Fenster	DO, 13. Oktober 2022, 10.30 bis 11.30
Ein Wegweiser durch den Dschungel der Web-API Möglichkeiten	Gregor Biswanger	DO, 13. Oktober 2022, 12.00 bis 13.00
Cloud-native Microservices für alle: On-Premises oder Cloud – mit Dapr	Christian Weyer	DO, 13. Oktober 2022, 15.45 bis 16.45
Serverless-Container mit Azure Container Apps	Thorsten Hans	DO, 13. Oktober 2022, 17.15 bis 18.15