Grundlagen der Containerisierung

Tobias Fenster

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 BC
- Docker Captain, Portainer und Traefik ambassador

Social und Blog

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

CONTAINER-BASICS

(mit Docker)

Nicht: Container-Orchestrierung!

CONTAINER-BETRIEB

ENTWICKLUNG MIT CONTAINERN

(Build)

CONTAINER-BASICS

(mit Docker)

Nicht: Container-Orchestrierung!

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?

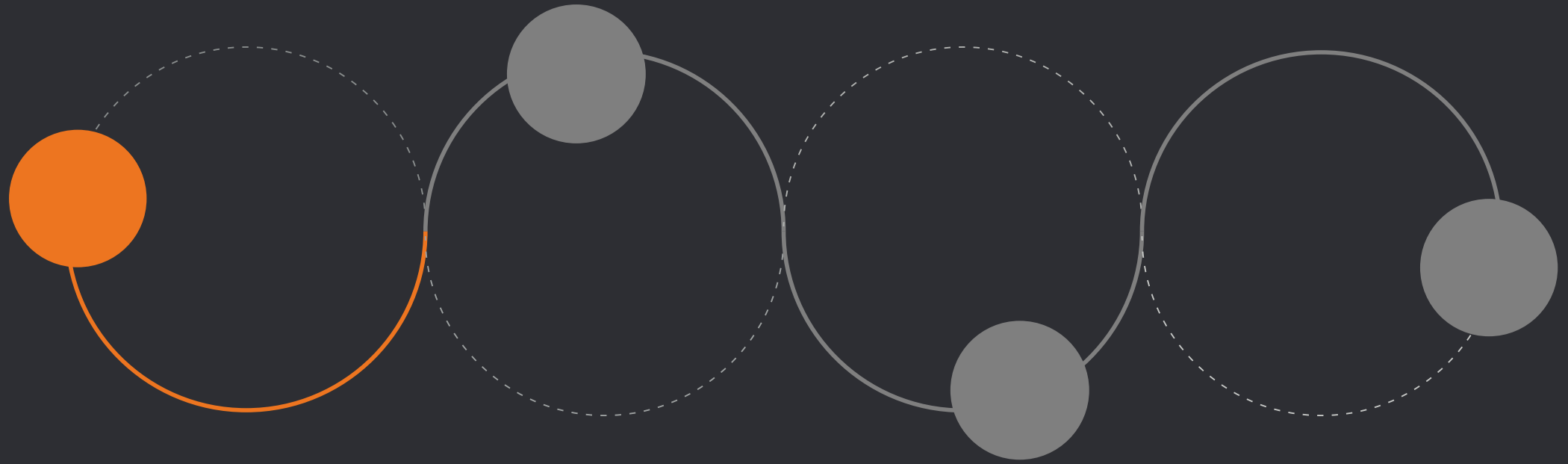


Image identifizieren

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

CONTAINER-BASICS

HELLO WORLD: WIE KOMME ICH ZU MEINEM ERSTEN CONTAINER?

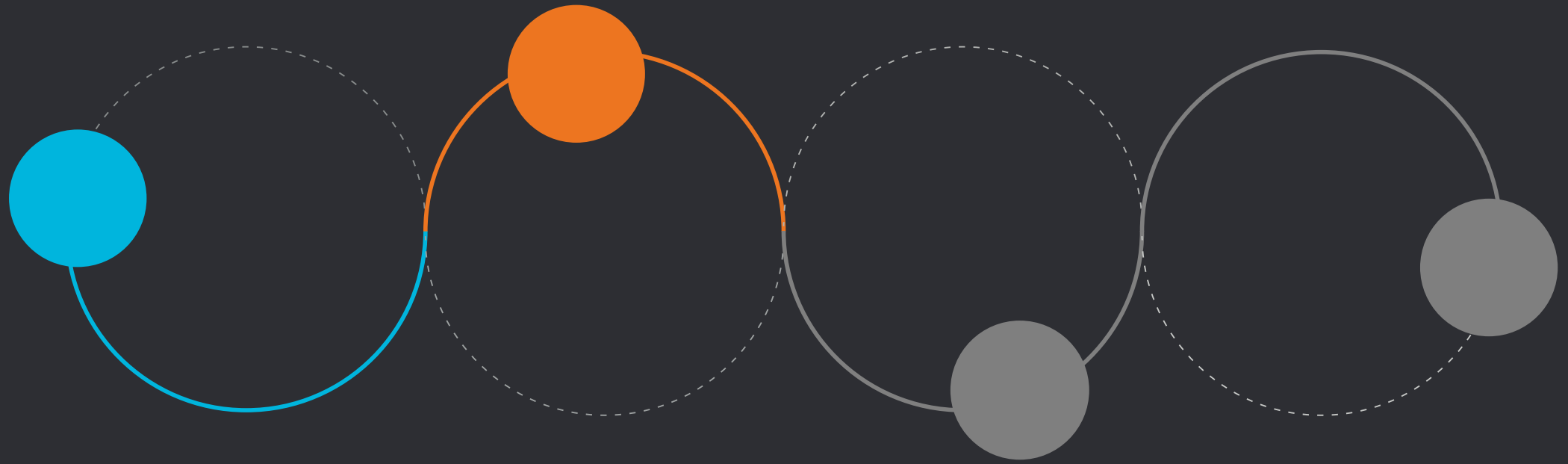


Image identifizieren → Image laden → → →

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

CONTAINER-BASICS

HELLO WORLD: WIE KOMME ICH ZU MEINEM ERSTEN CONTAINER?

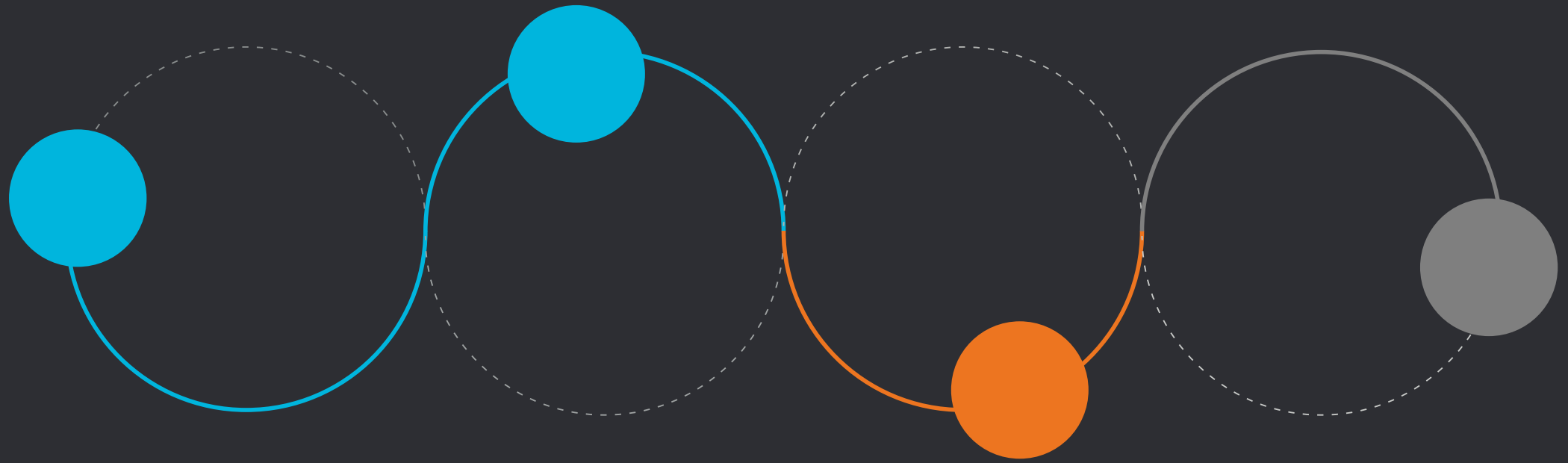


Image identifizieren → Image laden → Container erstellen →

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

CONTAINER-BASICS

HELLO WORLD: WIE KOMME ICH ZU MEINEM ERSTEN CONTAINER?

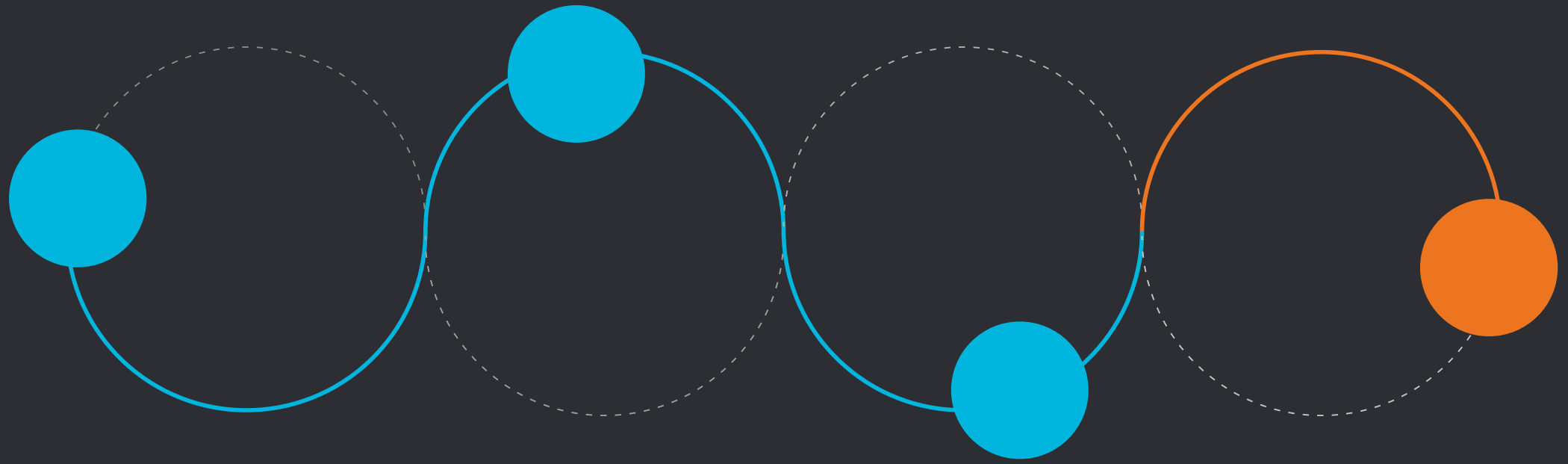


Image identifizieren → Image laden → Container erstellen → Container starten

```
docker start <id>
```

CONTAINER-BASICS

HELLO WORLD: WIE KOMME ICH ZU MEINEM ERSTEN CONTAINER?

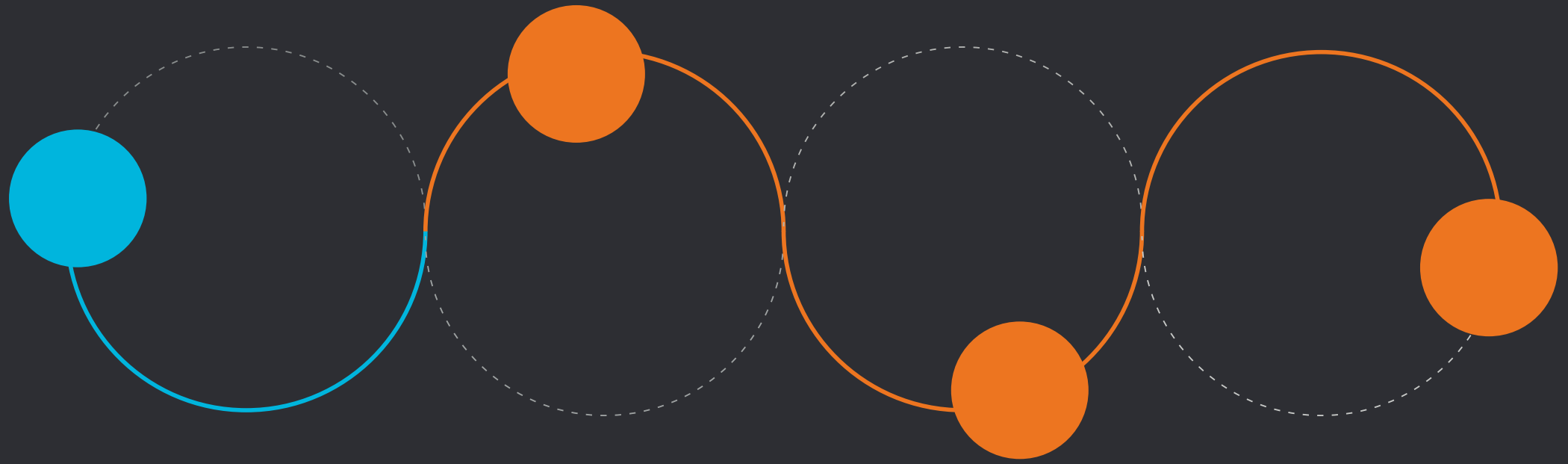
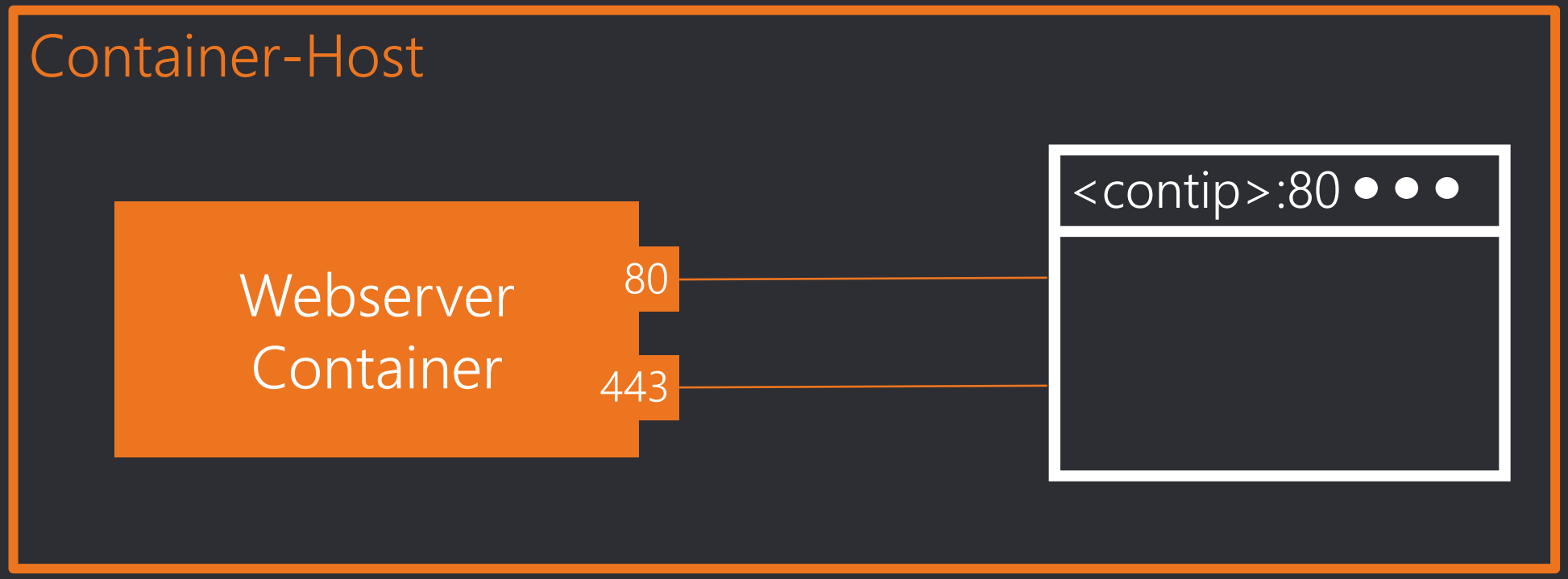


Image identifizieren → Image laden → Container erstellen → Container starten

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

CONTAINER-BASICS

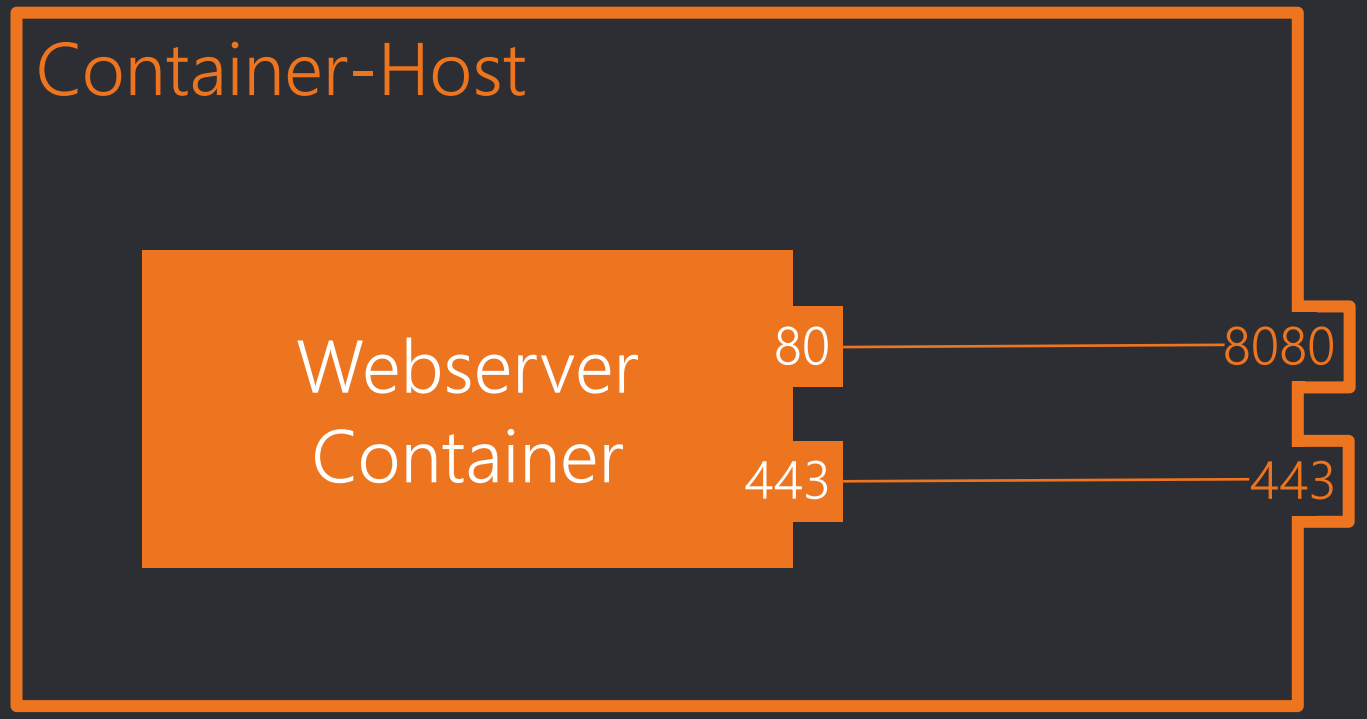
HELLO WORLD: WIE ERREICHE ICH MEINEN ERSTEN CONTAINER?



docker run mcr.microsoft.com/...

CONTAINER-BASICS

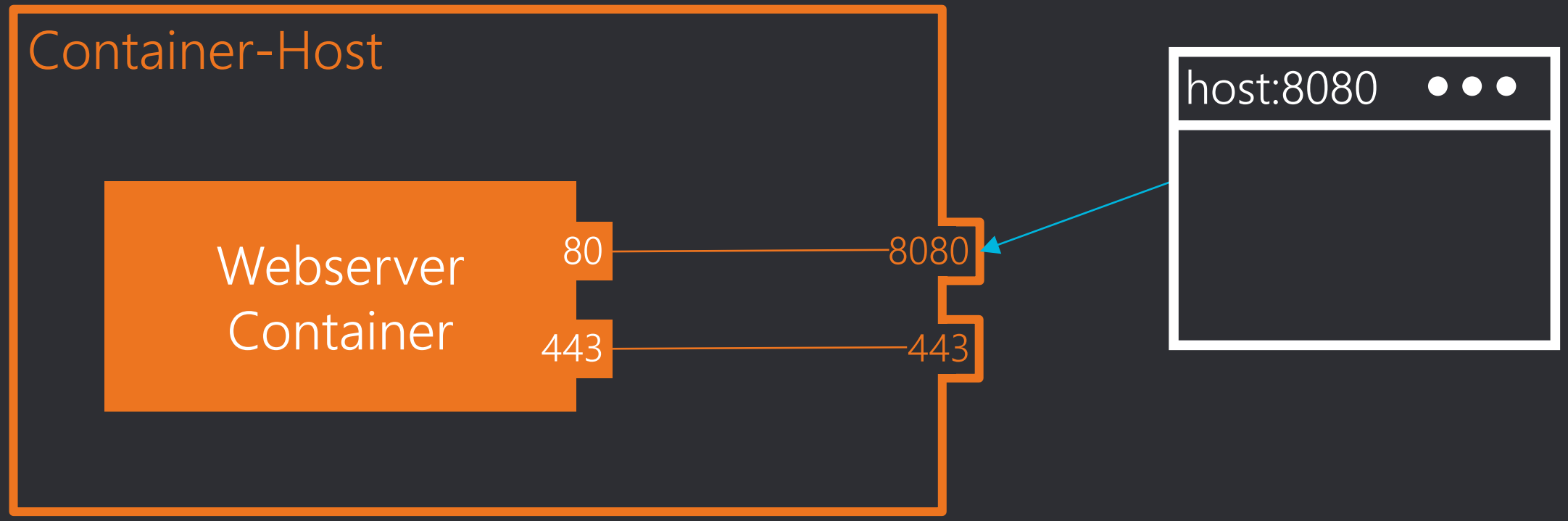
HELLO WORLD: WIE ERREICHE ICH MEINEN ERSTEN CONTAINER?



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

CONTAINER-BASICS

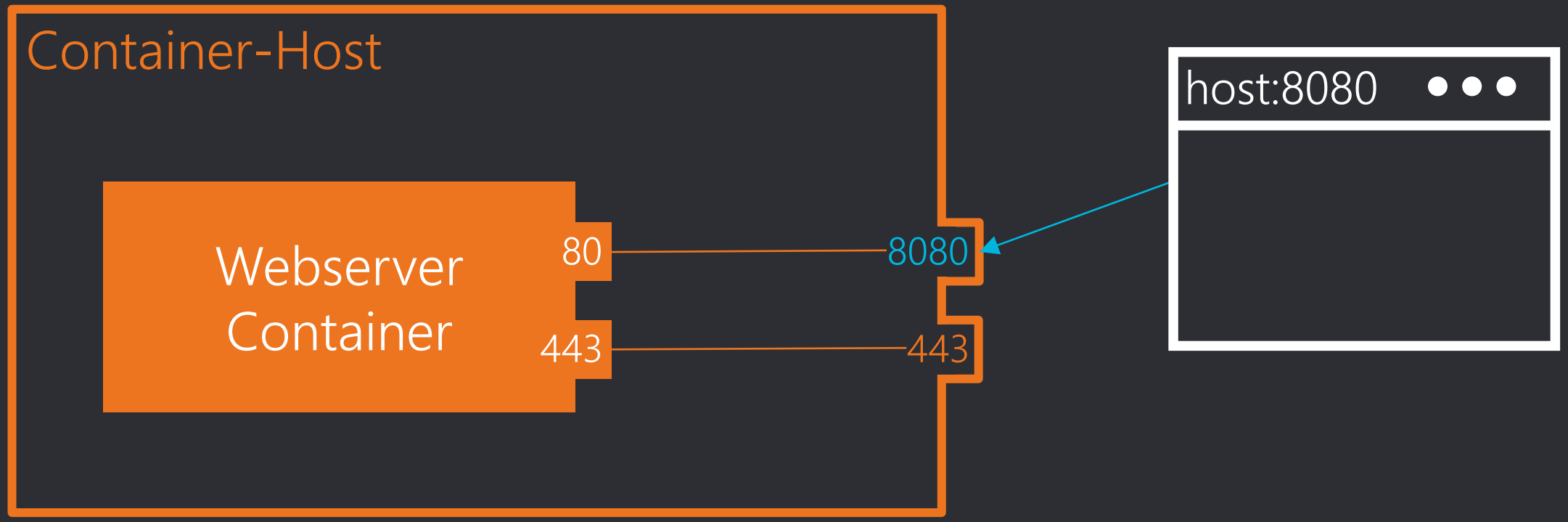
HELLO WORLD: WIE ERREICHE ICH MEINEN ERSTEN CONTAINER?



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

CONTAINER-BASICS

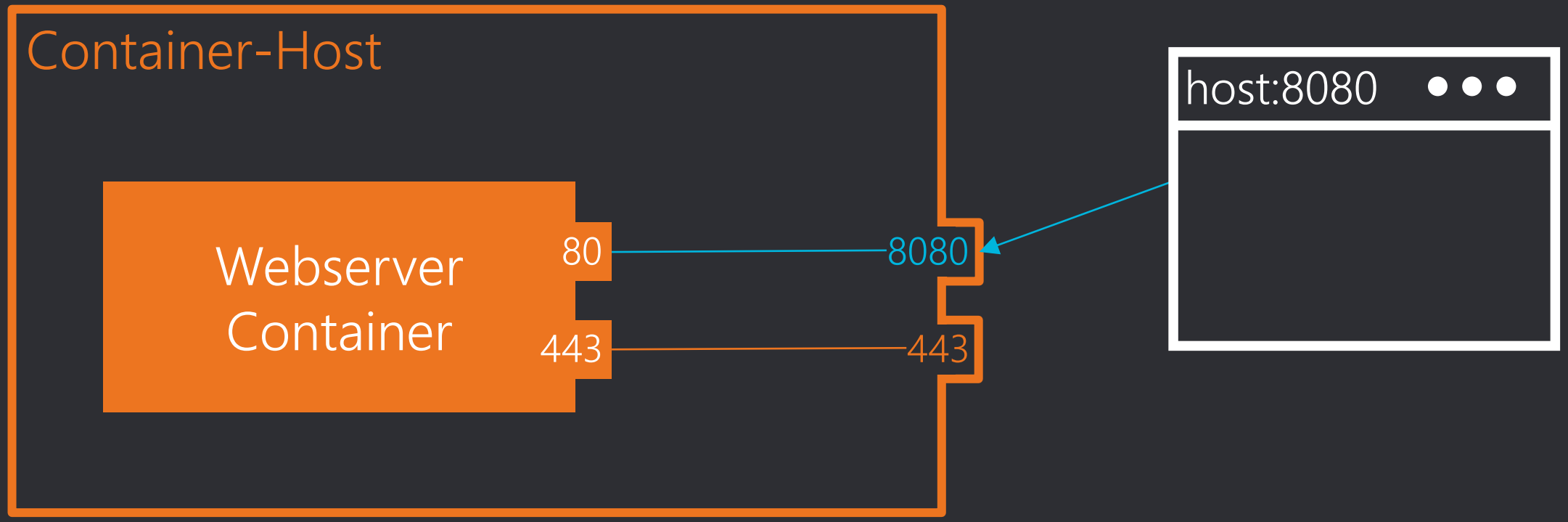
HELLO WORLD: WIE ERREICHE ICH MEINEN ERSTEN CONTAINER?



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

CONTAINER-BASICS

HELLO WORLD: 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?

- Drei Möglichkeiten für Netzwerkverbindungen zum Container:
  - Standardeinrichtung: NAT-Netzwerk erlaubt nur Verbindungen vom Host zum Container
  - Port Mapping von 1-n Ports im Container auf 1-n ggf. abweichende Ports im Host
    - Host-Firewall beachten
  - Geteilter Netzwerkadapter über transparente Einrichtung bringt eine dedizierte IP (statisch oder dynamisch) für jeden Container und macht ihn im Netzwerk erreichbar
    - Abhängig von allgemeiner Netzwerkeinrichtung
    - Benötigt MAC-Address-Spoofing

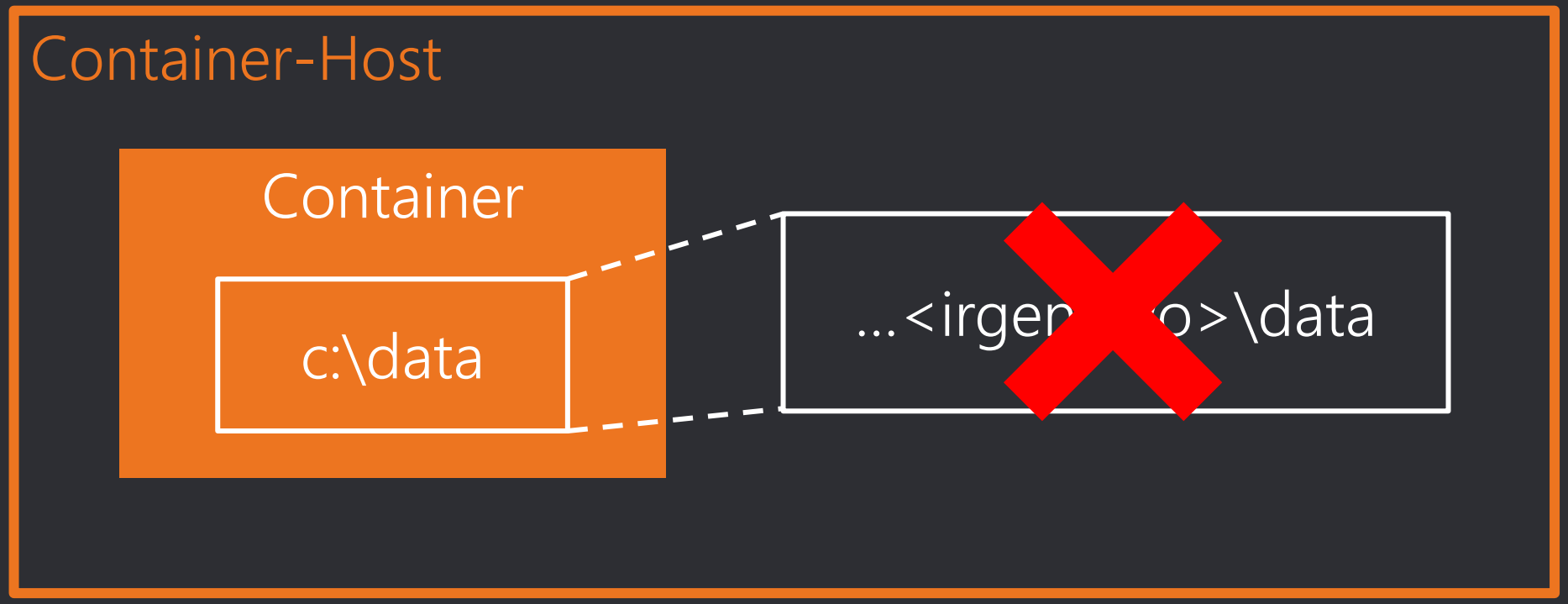
CONTAINER-BASICS

HELLO WORLD: WO SIND DIE DATEN DES CONTAINERS?

Container gelöscht → Daten gelöscht!

CONTAINER-BASICS

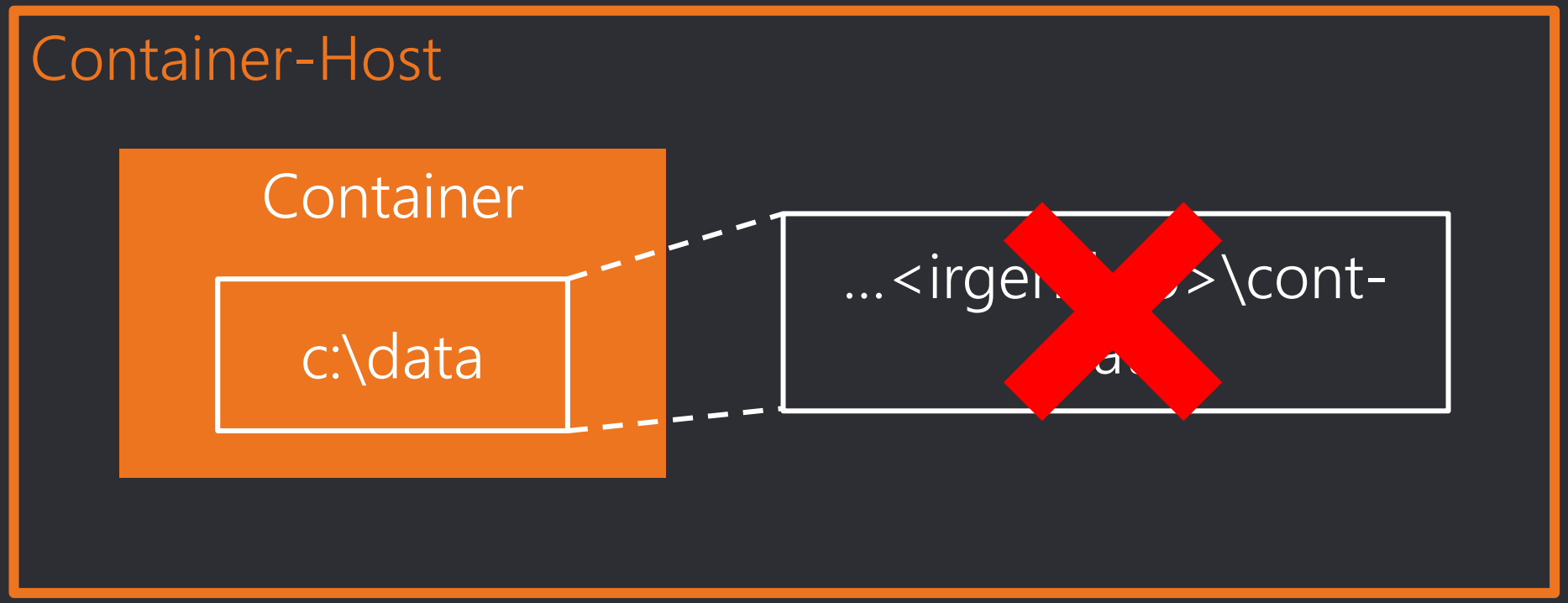
HELLO WORLD: WO SIND DIE DATEN DES CONTAINERS?



Ohne Parameter beim Startup

CONTAINER-BASICS

HELLO WORLD: WO SIND DIE DATEN DES CONTAINERS?

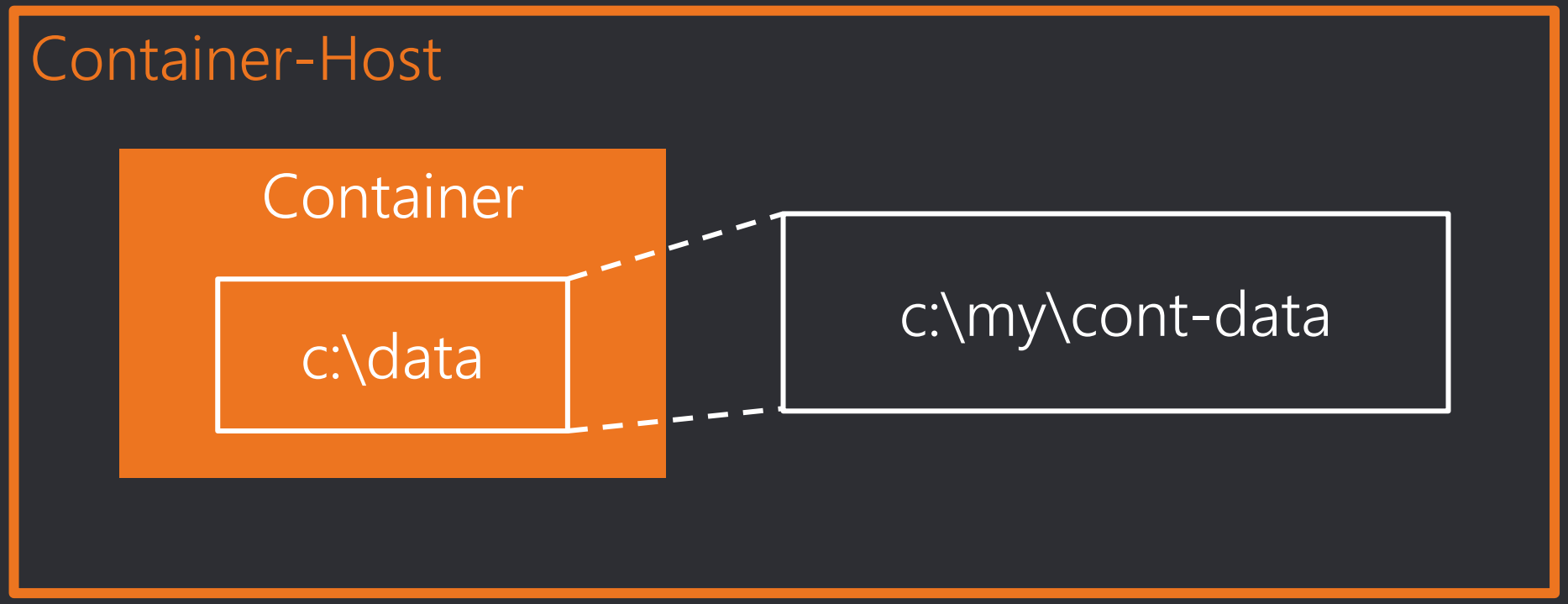


docker run -v cont-data:c:\data mcr.microsoft.com/...

Container gelöscht → Daten nicht gelöscht!

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 cp c:\whatever\file.txt abc123:c:\data

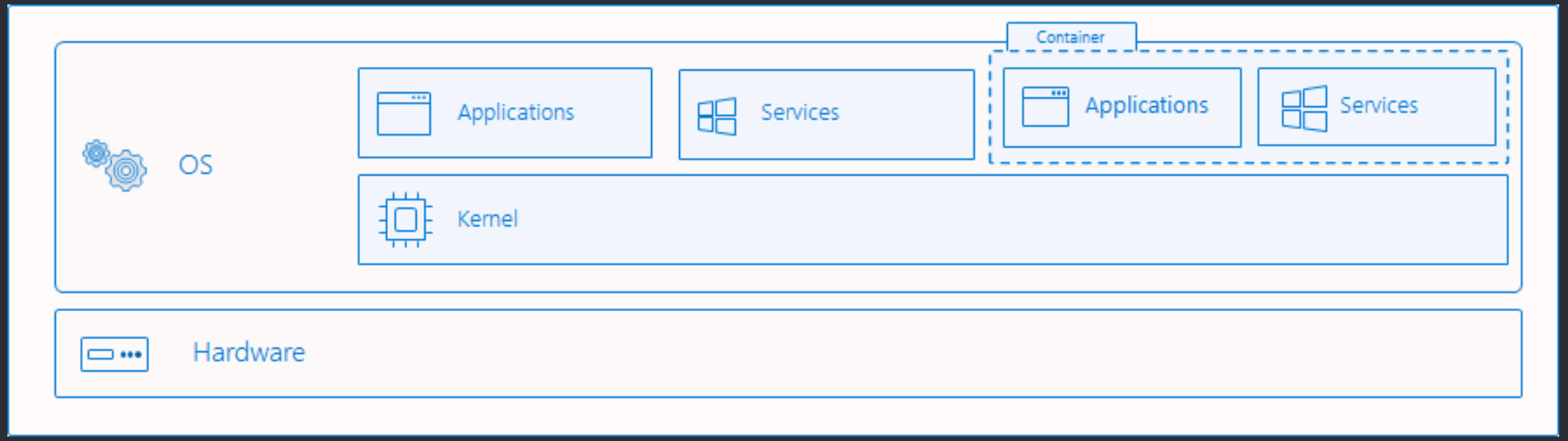
HELLO WORLD: WIE ERREICHE ICH MEINEN ERSTEN CONTAINER?

- Zwei Möglichkeiten für Dateiaustausch zwischen Host und Container:
  - Kopieren von Dateien mit docker cp führt zu zwei identischen, aber nicht verknüpften Dateien
    - funktioniert immer (in process Isolation)
  - Parameter -v
    - Erzeugt ein Volume, das geteilte Ordner und Dateien zwischen Host und Container erlaubt (Volume Mount)
    - Oder bindet einen Ordner des Hosts an einen Ordner im Container (Bind Mount)
    - kann nur bei Containerstart eingerichtet werden
    - Plugins für die direkte Anbindung von Storage-Lösungen

CONTAINER-BASICS

HELLO WORLD: WIE SIND CONTAINER ISOLIERT?

Process isolation

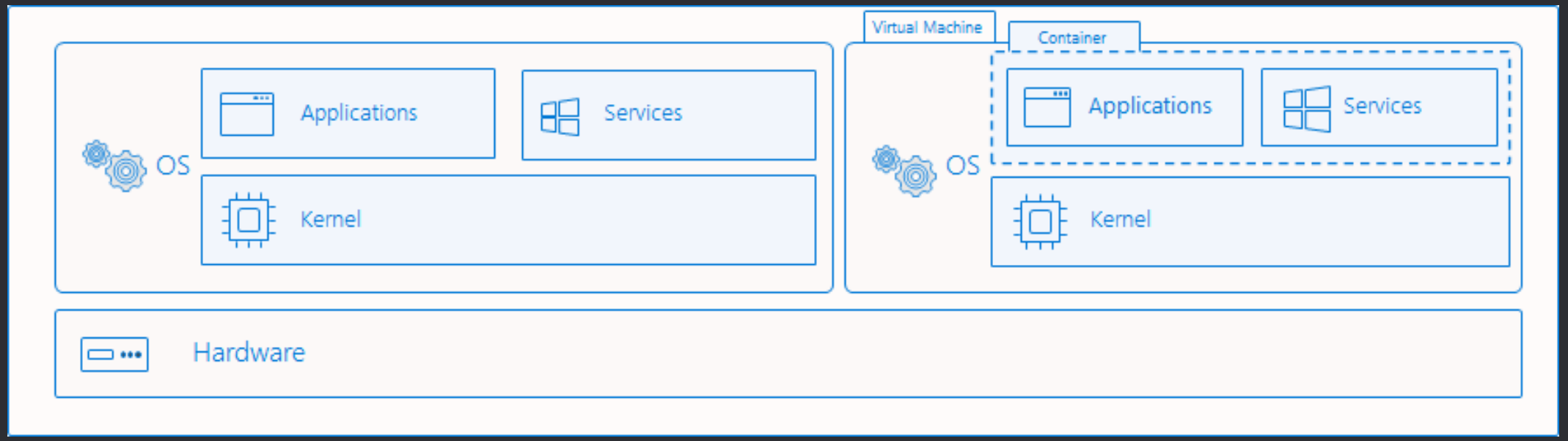


https://learn.microsoft.com/en-us/virtualization/windowscontainers/manage-containers/hyperv-container

CONTAINER-BASICS

HELLO WORLD: WIE SIND CONTAINER ISOLIERT?

Hyperv isolation

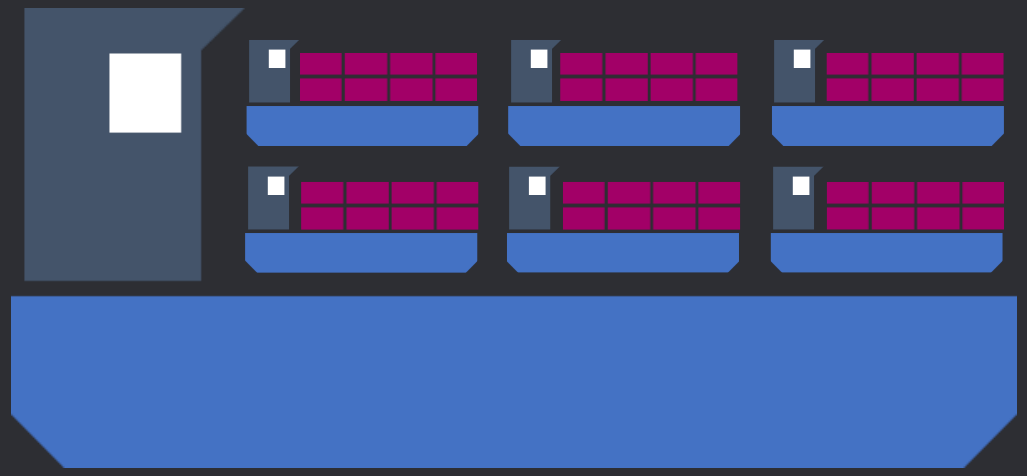


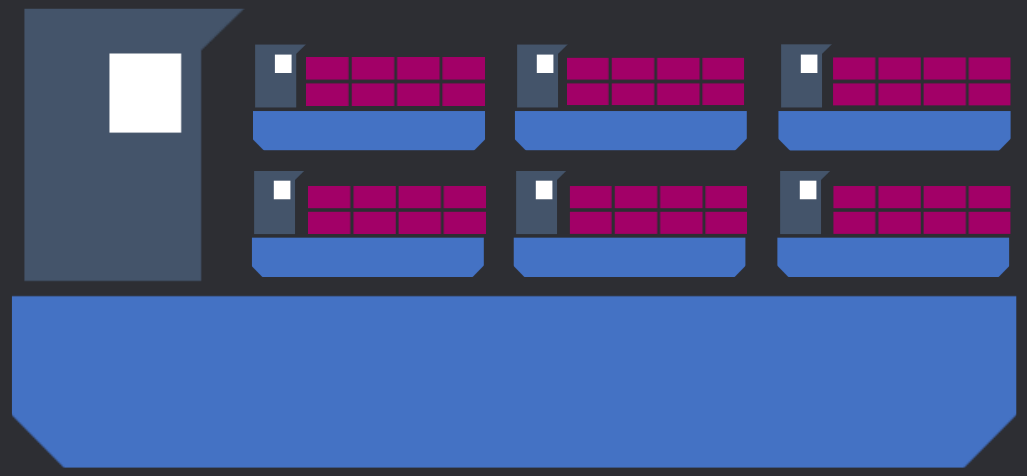
https://learn.microsoft.com/en-us/virtualization/windowscontainers/manage-containers/hyperv-container

CONTAINER-BETRIEB

CONTAINER-BETRIEB

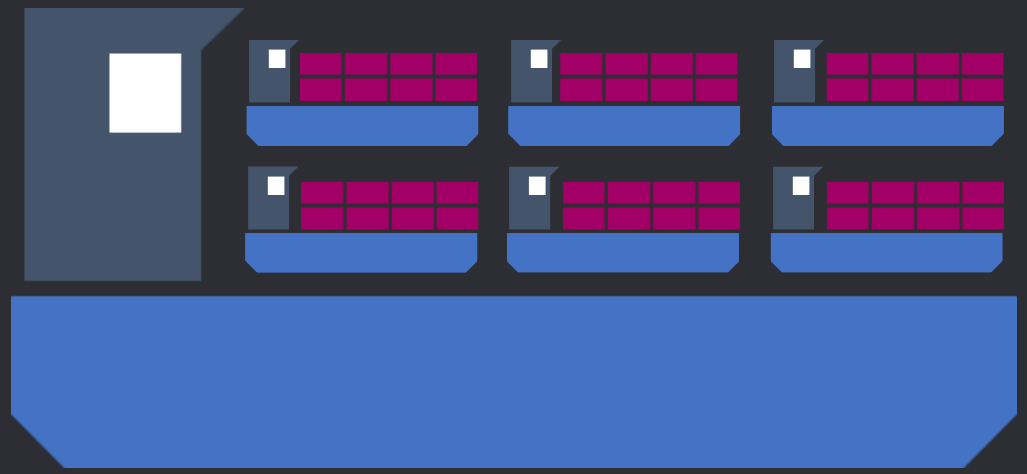
VIRTUELLE MASCHINEN VS. CONTAINER





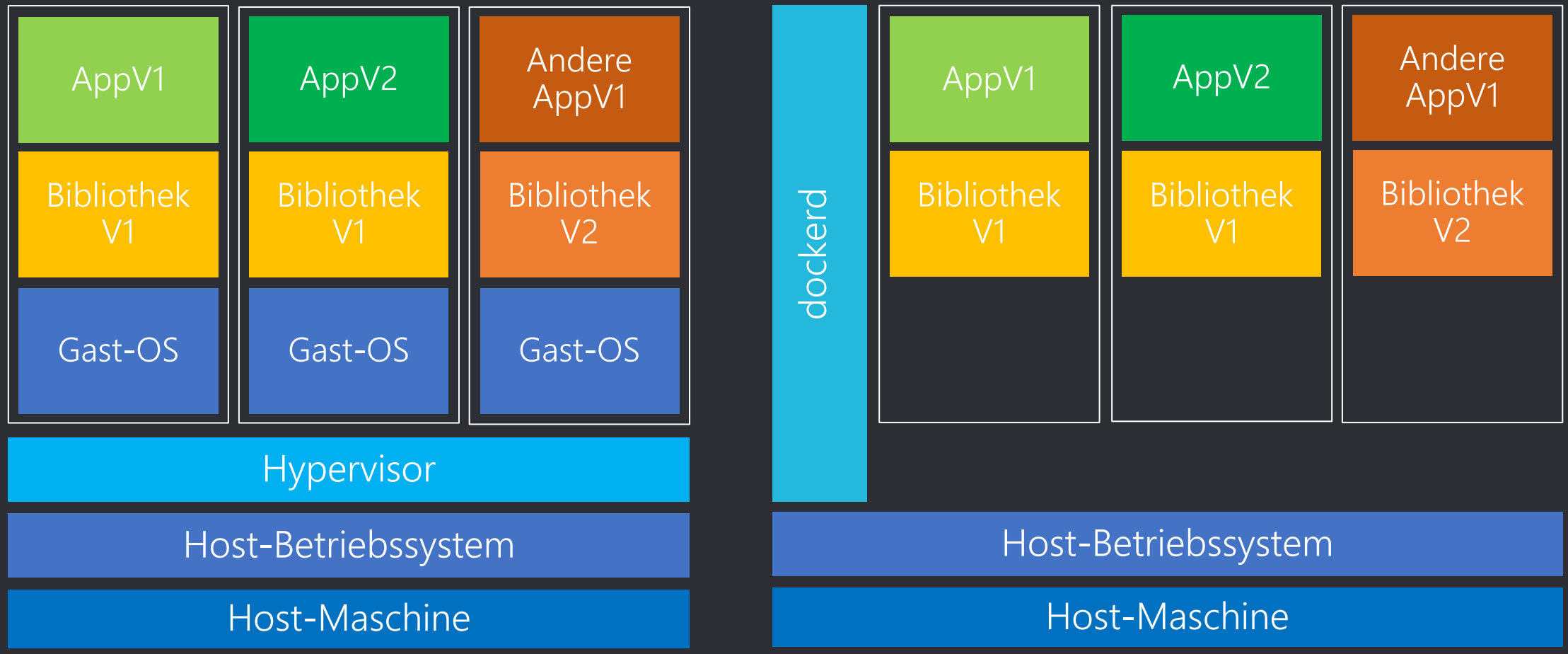
CONTAINER-BETRIEB

VIRTUELLE MASCHINEN VS. CONTAINER



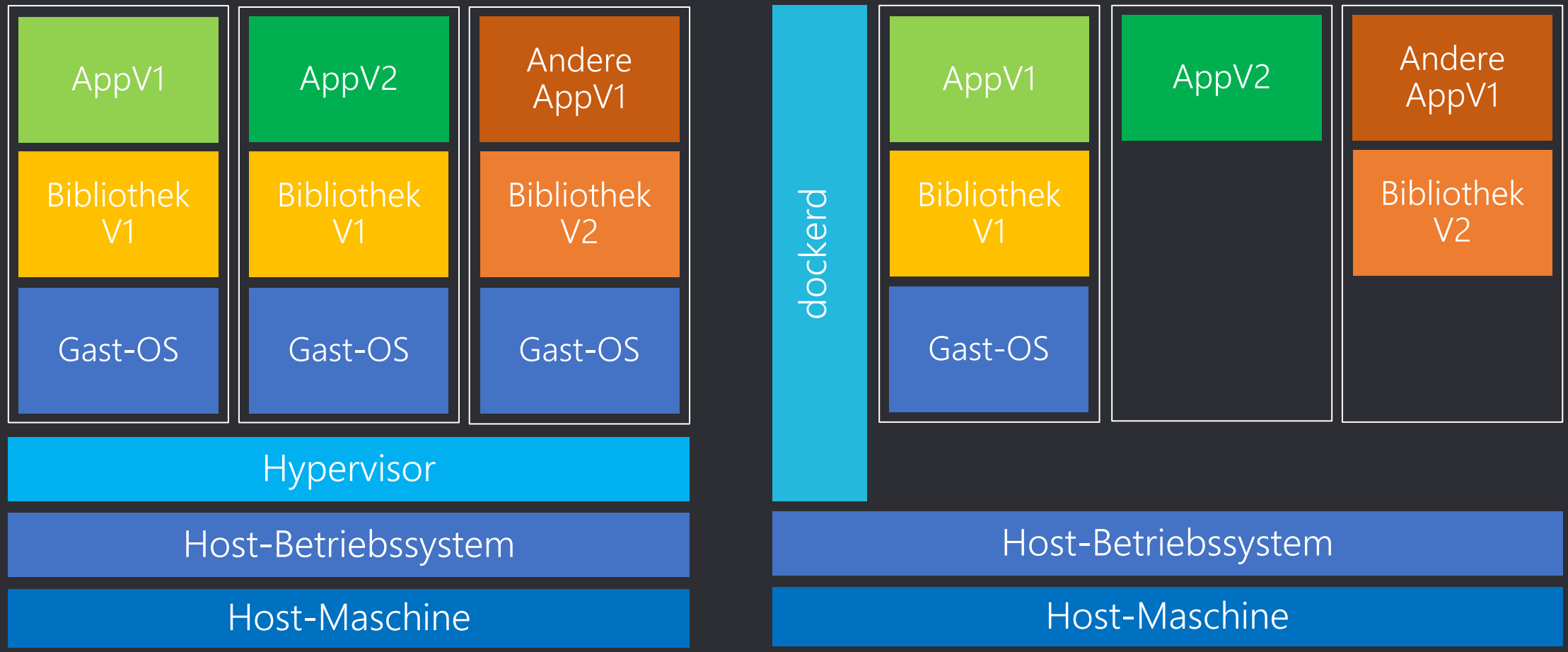
CONTAINER-BETRIEB

VIRTUELLE MASCHINEN VS. CONTAINER – ZUR LAUFZEIT

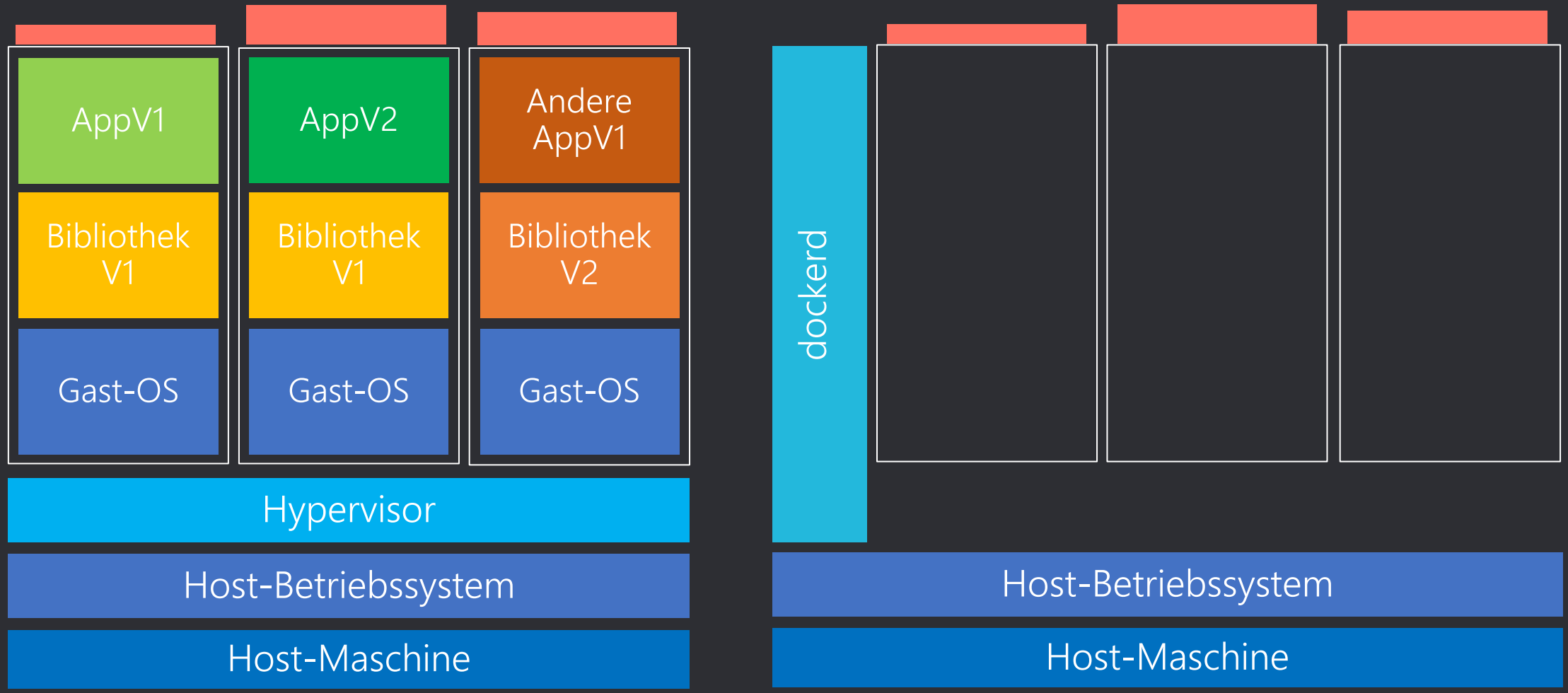


CONTAINER-BETRIEB

VIRTUELLE MASCHINEN VS. CONTAINER – SPEICHER IMAGES



VIRTUELLE MASCHINEN VS. CONTAINER – SPEICHER CONTAINER



CONTAINER-BETRIEB

HELLO WORLD: WIE STEUERE ICH MEHRERE CONTAINER?

- Mit Docker Compose über sehr einfache Definitionen möglich
- Skalierung in beschränktem Rahmen
- Für komplexere Orchestrierung Werkzeuge wie Docker Swarm oder Kubernetes

- Demo: Web API mit EF Core / MS SQL: https://jasonwatmore.com/net-7-dapper-ms-sql-server-crud-api-tutorial-in-aspnet-core#tools-required

CONTAINER-BETRIEB

VORTEILE DURCH CONTAINER

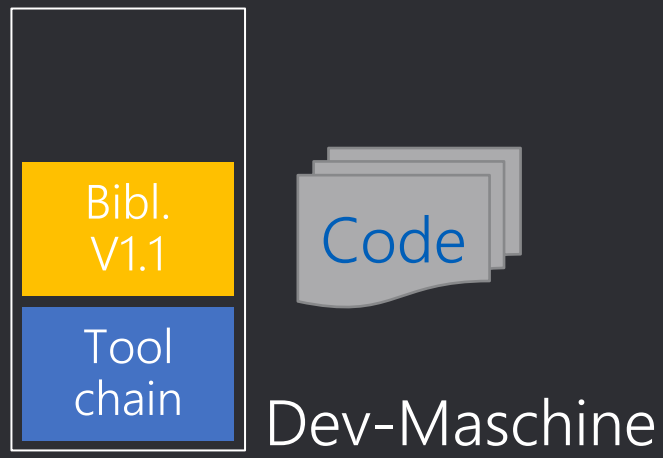
- Deutlich weniger Overhead als in VMs, da kein Gast-Betriebssystem notwendig ist
- Optimierte Ressourcennutzung durch Layer-Konzept bei der Ablage von Images und im laufenden Betrieb
- Alle für den Betrieb notwendigen Komponenten im Dockerfile bzw. Docker Image enthalten
- Stabil reproduzierbar und einfach aktualisierbar auf allen Zielsystemen
- Möglichkeit für Ressourcenbeschränkung, auch wenn die Anwendung das nicht nativ unterstützt

ENTWICKLUNG MIT CONTAINERN

(Build)

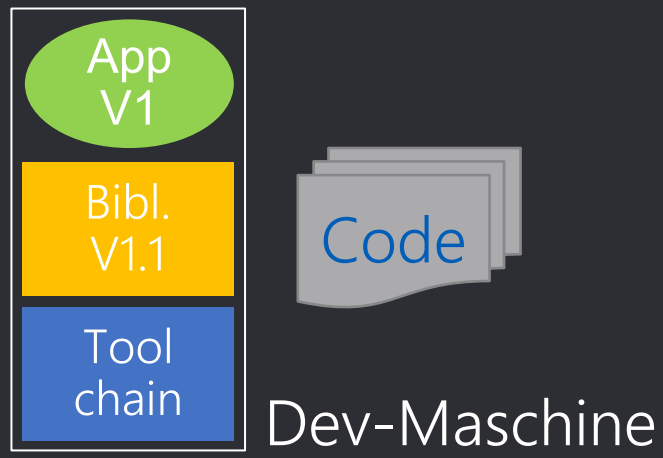
ANWENDUNGSENTWICKLUNG OHNE CONTAINER

- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung



ANWENDUNGSENTWICKLUNG OHNE CONTAINER

- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung



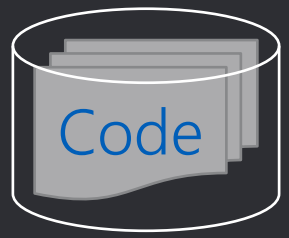
ANWENDUNGSENTWICKLUNG OHNE CONTAINER

- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung

ANWENDUNGSENTWICKLUNG OHNE CONTAINER

- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung

Dev-Maschine



Code repo

ANWENDUNGSENTWICKLUNG OHNE CONTAINER

- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung

ANWENDUNGSENTWICKLUNG OHNE CONTAINER

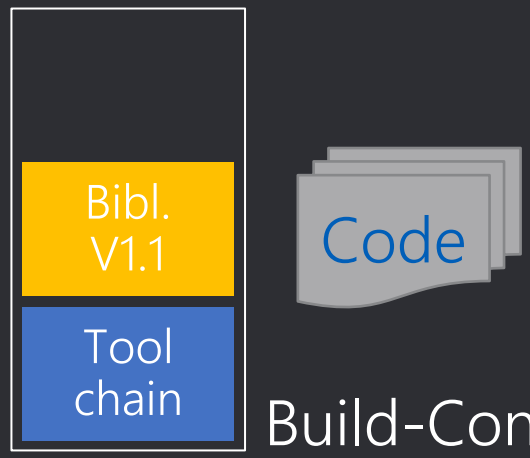
- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung

ANWENDUNGSENTWICKLUNG OHNE CONTAINER

- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung

ANWENDUNGSENTWICKLUNG MIT CONTAINER

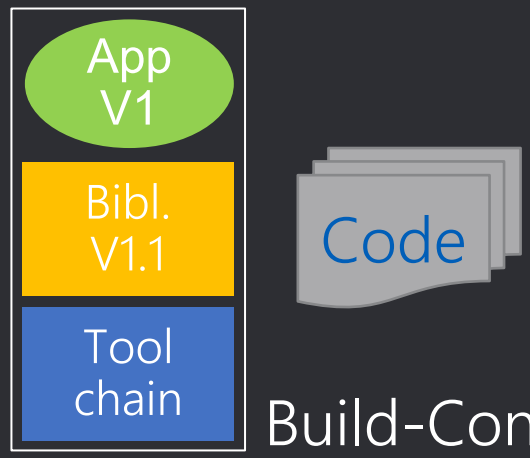
- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung



Build-Container
auf Dev-Maschine

ANWENDUNGSENTWICKLUNG MIT CONTAINER

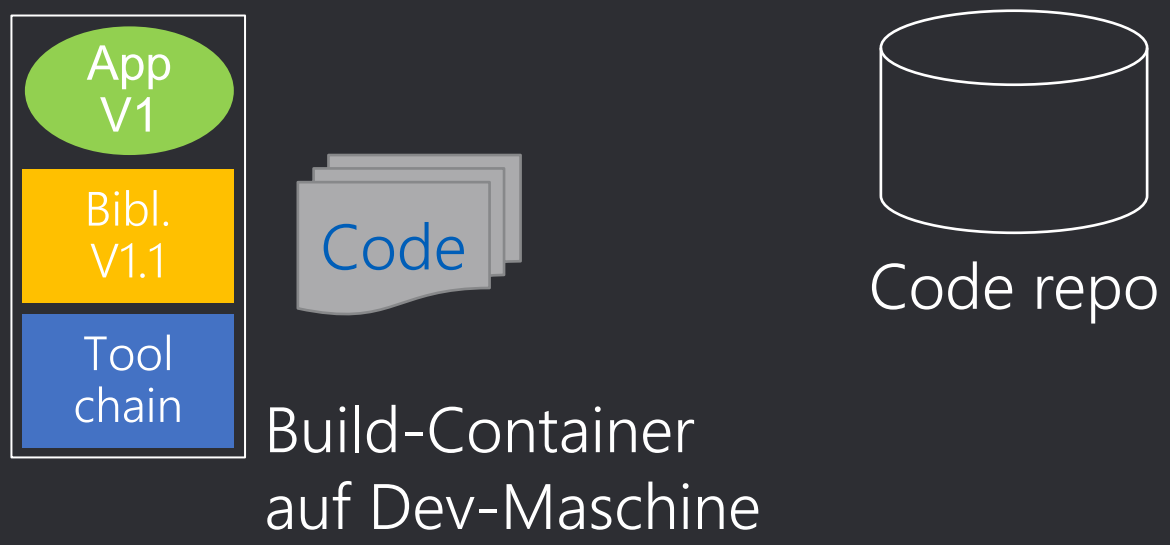
- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung



Build-Container
auf Dev-Maschine

ANWENDUNGSENTWICKLUNG MIT CONTAINER

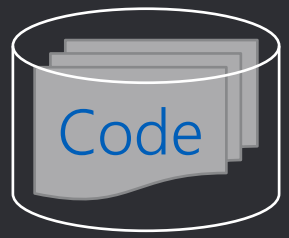
- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung



ANWENDUNGSENTWICKLUNG MIT CONTAINER

- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung

Build-Container
auf Dev-Maschine

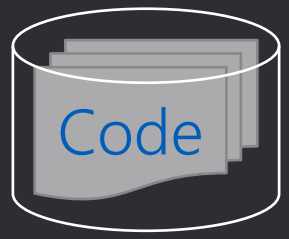


Code repo

ANWENDUNGSENTWICKLUNG MIT CONTAINER

- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung

Build-Container
auf Dev-Maschine



Code repo

Build-Container
auf Build-Maschine

ANWENDUNGSENTWICKLUNG MIT CONTAINER

- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung

ANWENDUNGSENTWICKLUNG MIT CONTAINER

- Entwicklung und Probe-Build in lokaler Umgebung, Codeänderungen werden in ein Repository übertragen
- Kontinuierlicher Build nimmt sie auf und erzeugt die Anwendung

CONTAINER-BETRIEB

VORTEILE DURCH CONTAINER

- Alle für das Erzeugen einer Anwendung notwendigen Informationen und Tools im Dockerfile bzw. Docker Image enthalten
  - Stabil reproduzierbar und einfach aktualisierbar auf allen Entwicklungssystemen
  - Kein “bei mir funktioniert es”
- Schneller, sauberer Wechsel zwischen verschiedenen Toolsets, Versionen etc. (und zurück)
- Übergabe an Betriebs-Team(-Kollegen) deutlich vereinfacht
- Nächste Stufe: Entwicklung im Container und best practices für Dockerfiles
  - S. meine Session „Containerisierung für .NET-Entwickler“ vom Dienstag

VIELEN DANK!

WELCHE FRAGEN DARF ICH BEANTWORTEN?

Wer 20 Minuten Zeit hat: Why we built Docker (https://youtube.com/watch?v=3N3n9FzebAA), erste öffentliche Session zu Docker bei der PyCon 2013, von Solomon Hykes, einem der Gründer des OpenSource-Projekts Docker und der daraus entstandenen Firma Docker Inc.