

# ADC

## Containerbasierte Entwicklung für .NET-Entwickler

Tobias Fenster



# Tobias Fenster

- Business
  - Managing Director bei 4PS by Hilti
- Community
  - MS Regional Director und MVP für Azure und Business Applications
  - Docker Captain
- Social, Blog und Podcast
  - tobiasfenster bei X und LinkedIn
  - tobiasfenster.io
  - „Window on Technology“ podcast



# CONTAINERBASIERTE ENTWICKLUNG FÜR .NET-ENTWICKLER

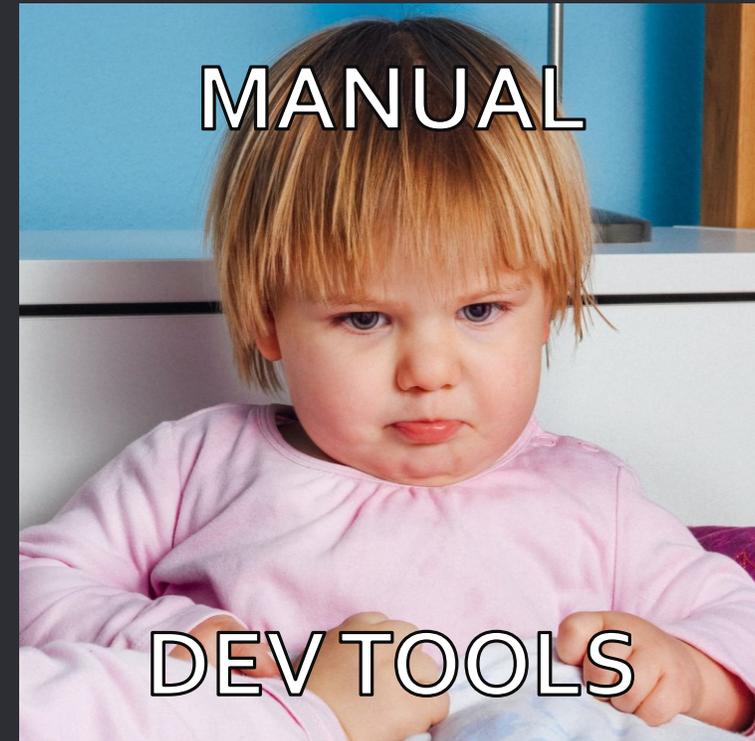
WÄHREND DER  
ENTWICKLUNG

WÄHREND DER  
LAUFZEIT

# ENTWICKLUNG IM CONTAINER

CONTAINERBASIERTE ENTWICKLUNG

WARUM?



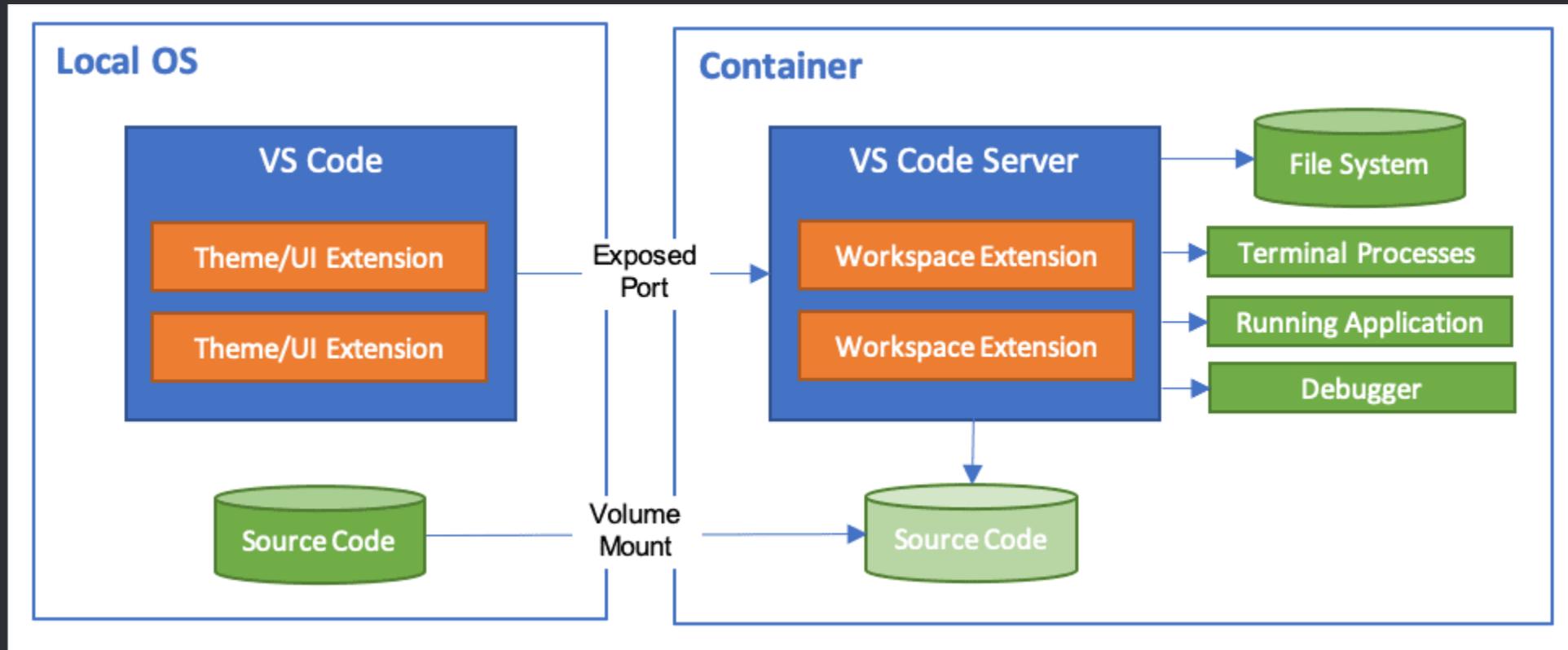
# WARUM?

- **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

# 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



DEMO

# BEST PRACTICES FÜR PRODUKTIV-CONTAINER

BEST PRACTICES

# DEMO SZENARIO

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

DEMO

# WELCHES BASE IMAGE?

- Image zum **Build**: [mcr.microsoft.com/dotnet/sdk](https://mcr.microsoft.com/dotnet/sdk)
  - .NET CLI
  - .NET runtime
  - ASP.NET Core runtime
- **Niemals latest!**
  - Heute 8.0, bald 9.0...
  - Bei Abhängigkeiten auch z.B. 8.0.203
- An die Größe (Download, Attack Surface) denken → **alpine**



# 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



# 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



# WIE GEHT ES DEM CONTAINER?

- Container können „health“ Status anzeigen
  - Definition kann für einfache Container ebenfalls im Dockerfile erfolgen
- Wichtig z.B. bei Orchestratoren wie Kubernetes
  - Erkennen, wann ein Container ersetzt werden muss
  - Dann aber oft mit eigenen Mitteln des Orchestrator



# WIE SOLL DER CONTAINER SICH VERHALTEN?

- Normale **Konfigurationsmechanismen** in .NET 1:1 auch auf Container abbildbar:
  - AspNetcore\_environment
  - Environment variables



# WELCHER USER?

- **Non-root** unter Linux
  - User anlegen, ohne Passwort
  - Zugriff auf Verzeichnis geben
  - Oder in .NET Standard-Images vorbereiteten User verwenden
- System = gMSA unter Windows



# WELCHE SHELL?

- **Chiselled images** seit .NET 8 nicht mehr in Preview
  - Noch weiter optimiert
  - Keine Shell...



# GEHT DAS NOCH EINFACHER UND OHNE CONTAINER-TOOLING?

- Seit .NET 7 **direkter Support** für die Erstellung von Container Images
  - Ganz ohne Dockerfile (und damit auch mit weniger Kontrolle)
  - Lokale Images unter lokaler Nutzung von Docker
  - Bei Nutzung von remote Repos komplette ohne Docker möglich
- Bisher nur Erfahrung durch erstes Ausprobieren und [lippertmarkus.com](https://lippertmarkus.com) / offizielle Announcements
- Mehr wird kommen, Gaps sind bereits dokumentiert (kein Windows, kein Auth gegen remote Repos)



# Fragen?



# Ich freue mich auf Feedback!

Scan mich!

A black curved arrow points from the text 'Scan mich!' towards the QR code.

# Vielen Dank!

