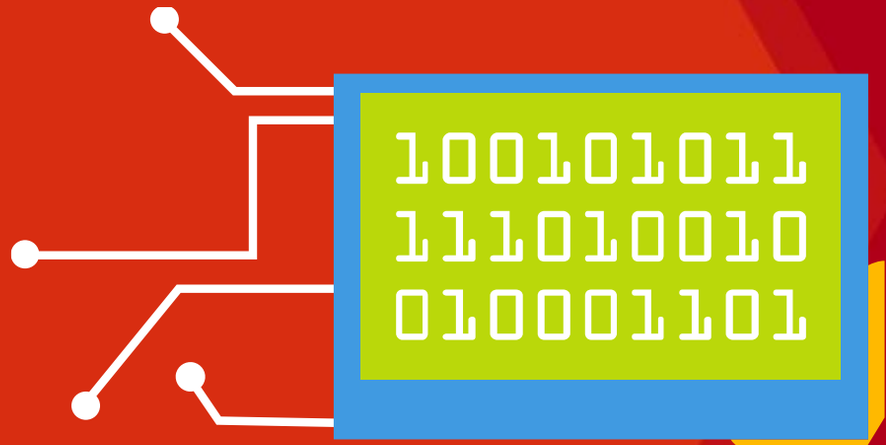
KONFERENCA

Microsoft

Release Management and Containers in the Cloud, on Windows and on Linux



Ana Roje Ivančić, VS ALM MVP, Ekobit
Ognjen Bajić, VS ALM MVP, Ekobit

TEHNOLOGIJA

Speakers

Working with VS ALM tools since 2004.

WinDays 2005 preconf day on VSTS

Worked as Dev, PM, Test, RM, SM, PO…

VS ALM MVPs

Agenda

DevOps and Release Management with TFS

Introduction to Containers. What is a Container?

Windows Containers

Development And Containers

Containerize Application Demo

Containerize Release Pipeline

DevOps and Release Management

Microsoft

KONFERENCA

DevOps

Union of people, process, and tools to enable continuous delivery of value to end users

Cornerstones of DevOps:

- Culture supporting sharing and collaboration
- Continuously optimized lean process
- Automated deployment pipeline

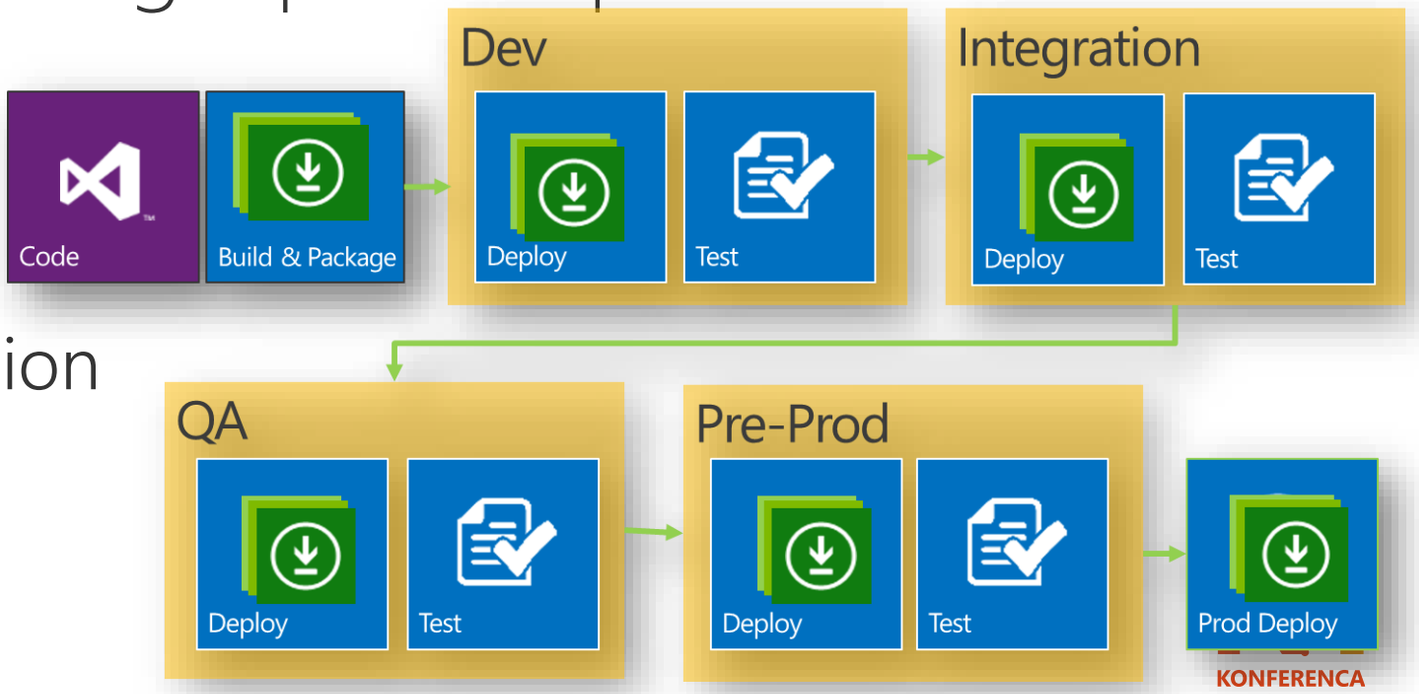
Automated Deployment Pipeline

Automate build, deploy and test to achieve low lead times and rapid feedback

Push-button deployments

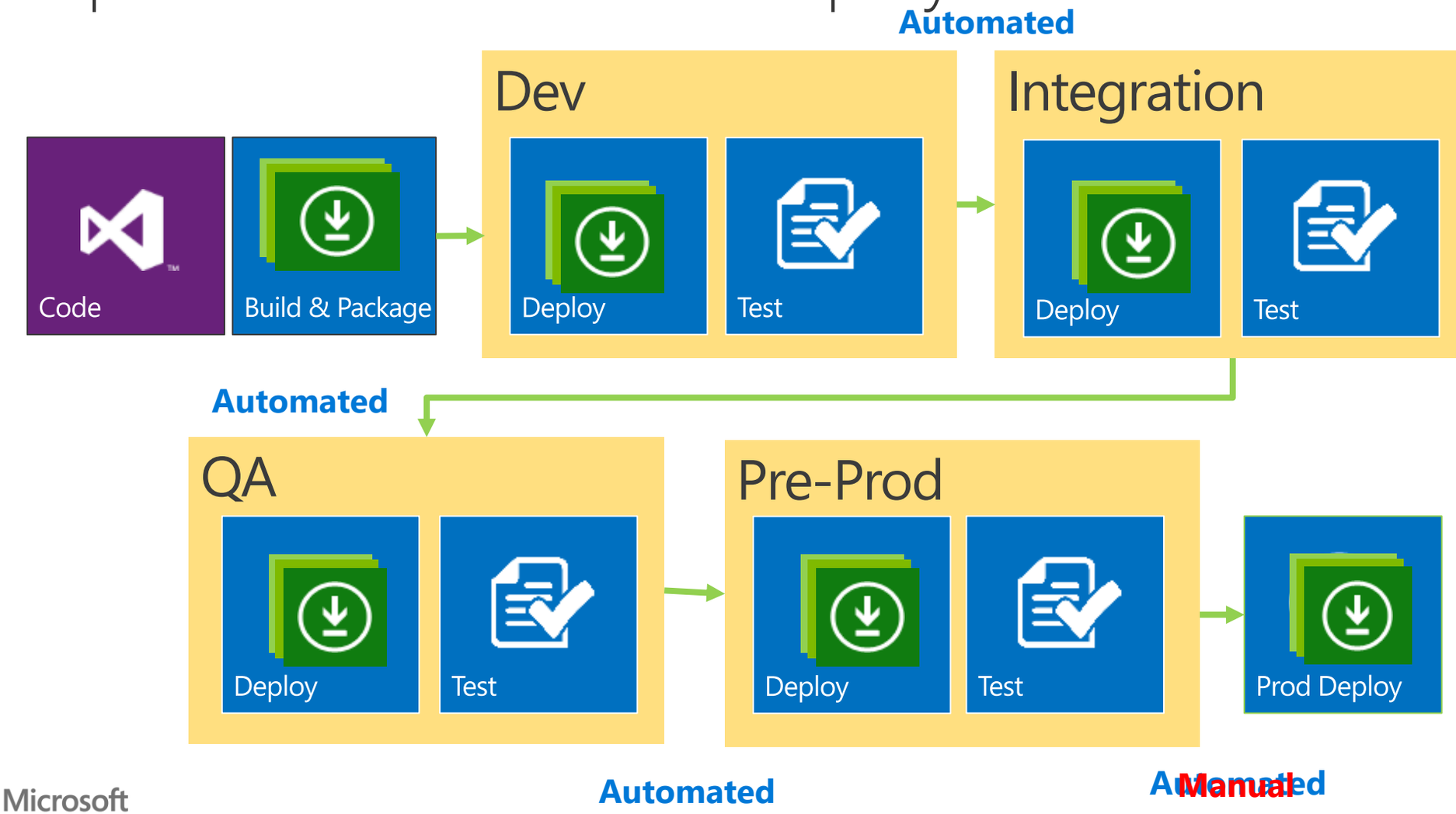
Deployment pipeline is the single path to production for all changes

- Code
- Infrastructure and environments
- Database schemas or configuration



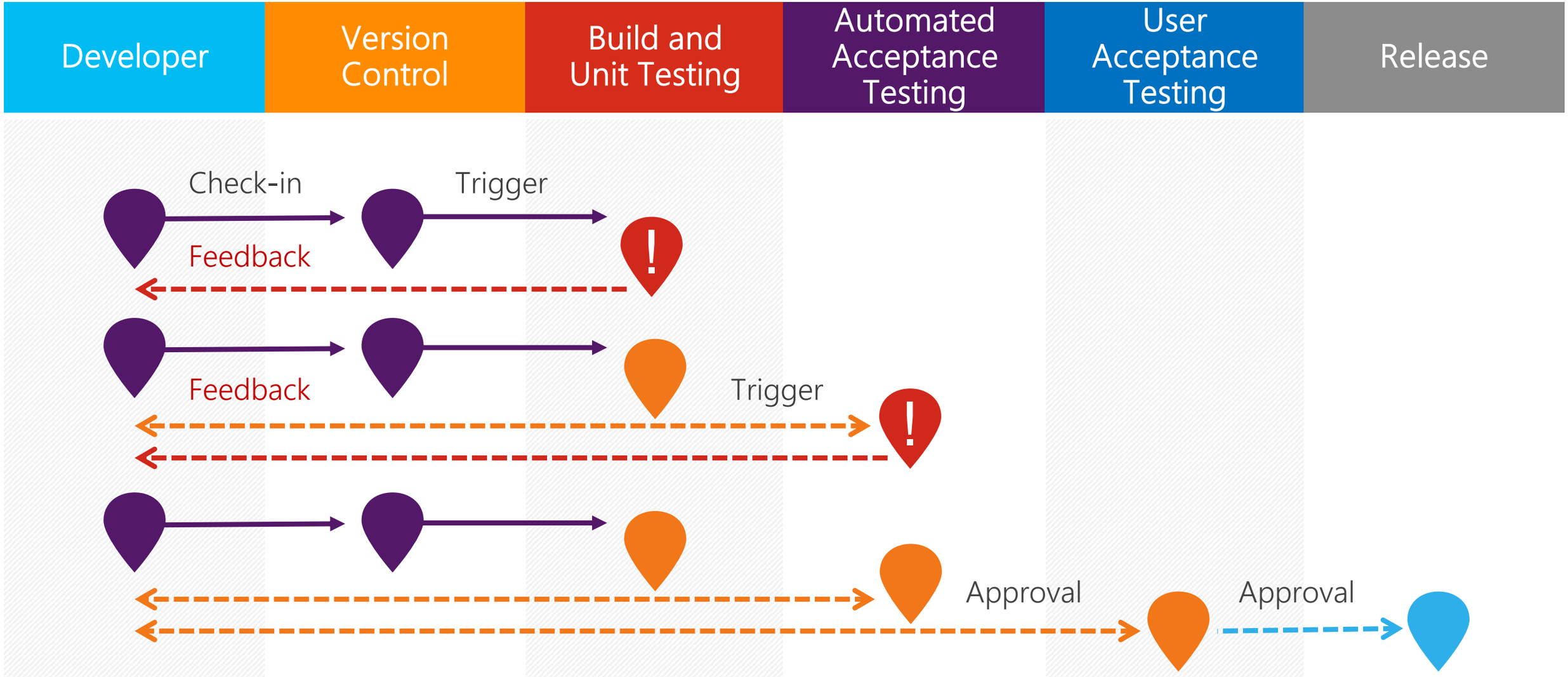
Release Management

Repeatable Automated Deployment Model



Continuos Delivery

Continuous Quality for Continuous Value Delivery



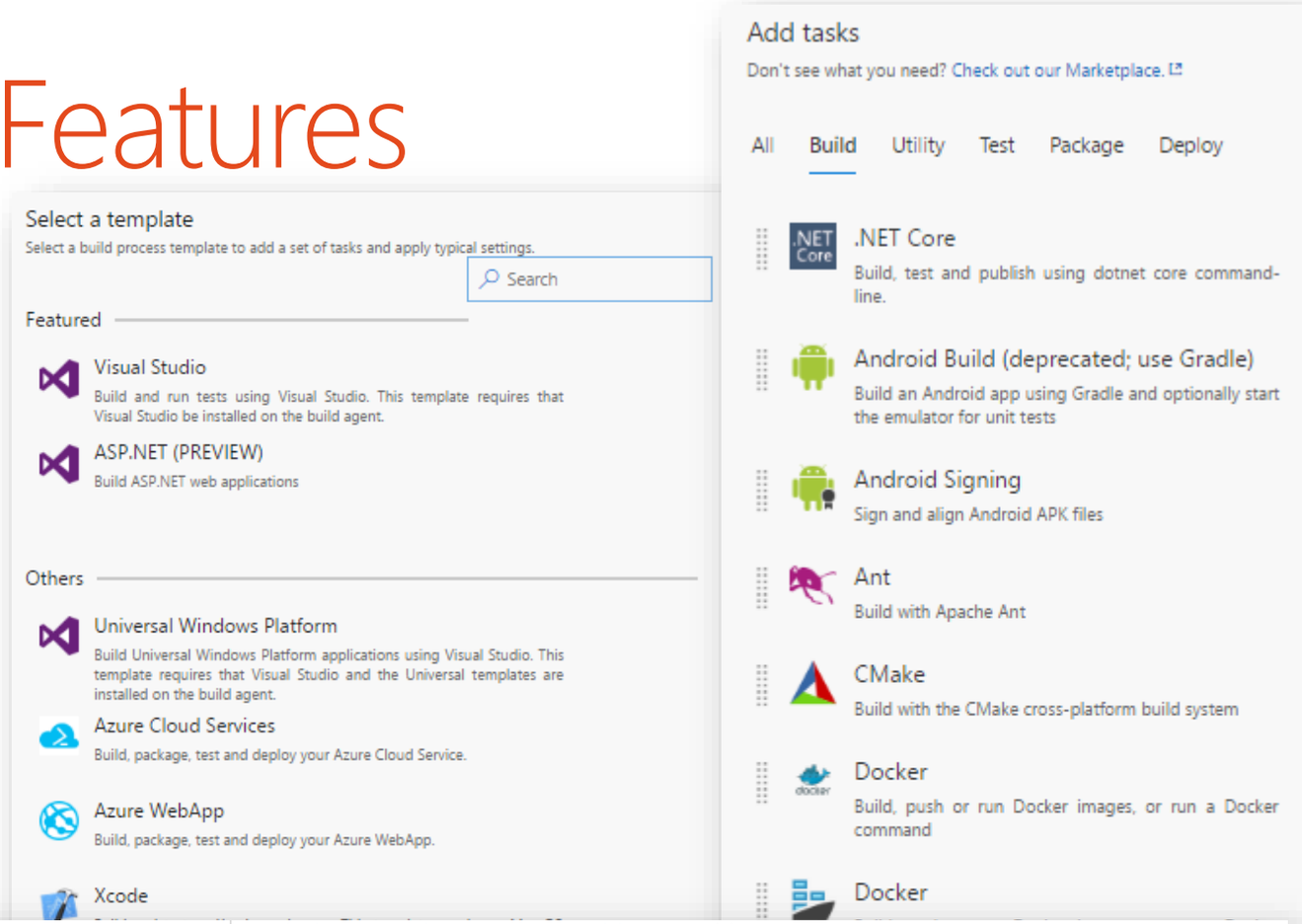
Release Management in TFS/VSTS

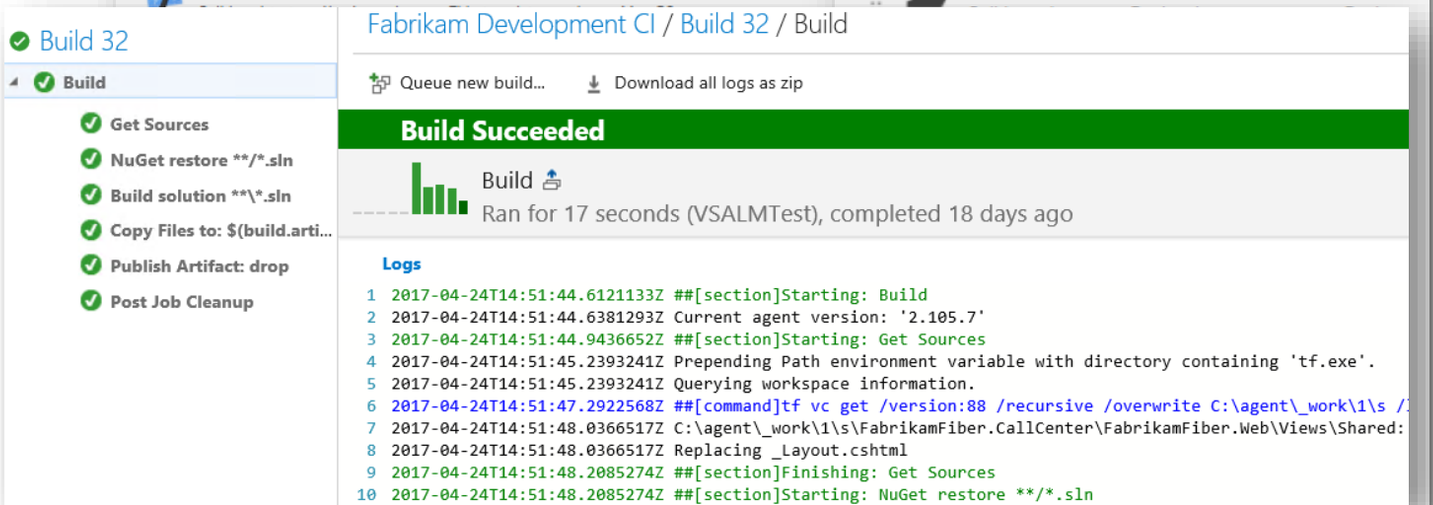
Microsoft

KONFERENCA

Release Management Features

- Custom workflows
  - Steps and templates
  - Simple customization
  - Extensible
- Real-time log output
- Versioning and auditing
- Cross-platform support
  - Windows, Mac, Linux
  - Build .NET, Java, Android, iOS
  - Ant, CMake, Maven, Xcode Build, Android Build, …
- Shared agents
  - Across projects and collections





Microsoft

Release Management for VSTS/TFS

Continuous Delivery

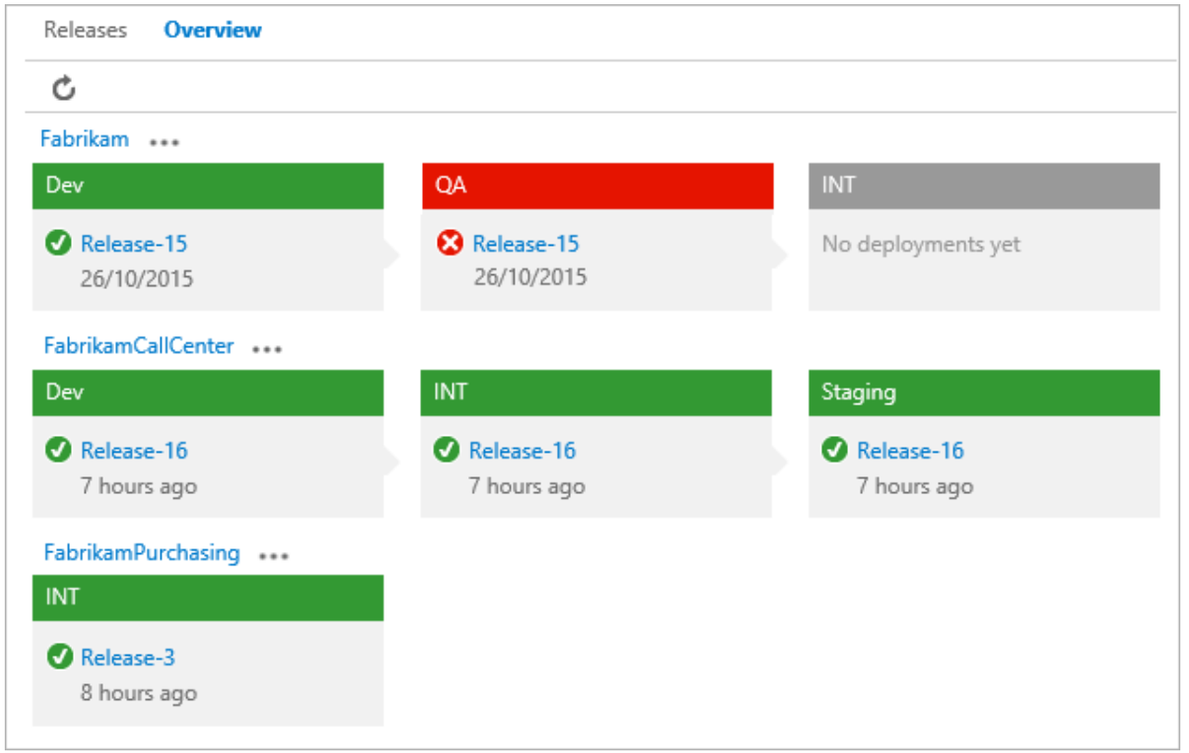
- Define per-environment release pipeline
- Trigger manually or via successful builds

Approval Policies

- Configure per-environment sign-offs
- Manual validation requirements

Release Visibility

- View per-environment release status
- Track release back to commits



Automating the Build and Release Cycle

Continuous build and deployment solution for high-quality DevOps

- Helps automate the deployment and testing in multiple environments
- Integrated with VSTS and TFS
- Improves collaboration throughout the process

Builds any kind of app

- Cross-platform support
- Flexible and open architecture

Automates the deployment process

- More frequent releases, push button releases
- Fully automate the delivery all the way to production
- Set up semi-automated processes with approvals and on-demand deployments

Easy customization and versioning

Provides analytics and reporting

Release Management for VSTS/TFS and Support for Containers/Docker Demo

Microsoft

KONFERENCA

Introduction to Containers

Microsoft

KONFERENCA

The cloud has changed expectations

Availability

100% uptime

Hyper-scale

From startup to enterprise

Agility

Deliver just in time speed

Containers deliver speed, flexibility, and savings

Availability

62%

Report reduction in MTTR

10X

Cost reduction in maintaining existing applications

Hyper-scale

41%

Move workloads across private/public clouds

Eliminate

"Works on my machine" issues

Agility

13X

More software releases

65%

Reduction in developer onboarding time

State of App development Survey: Q1 2016, Cornell University case study

Microsoft

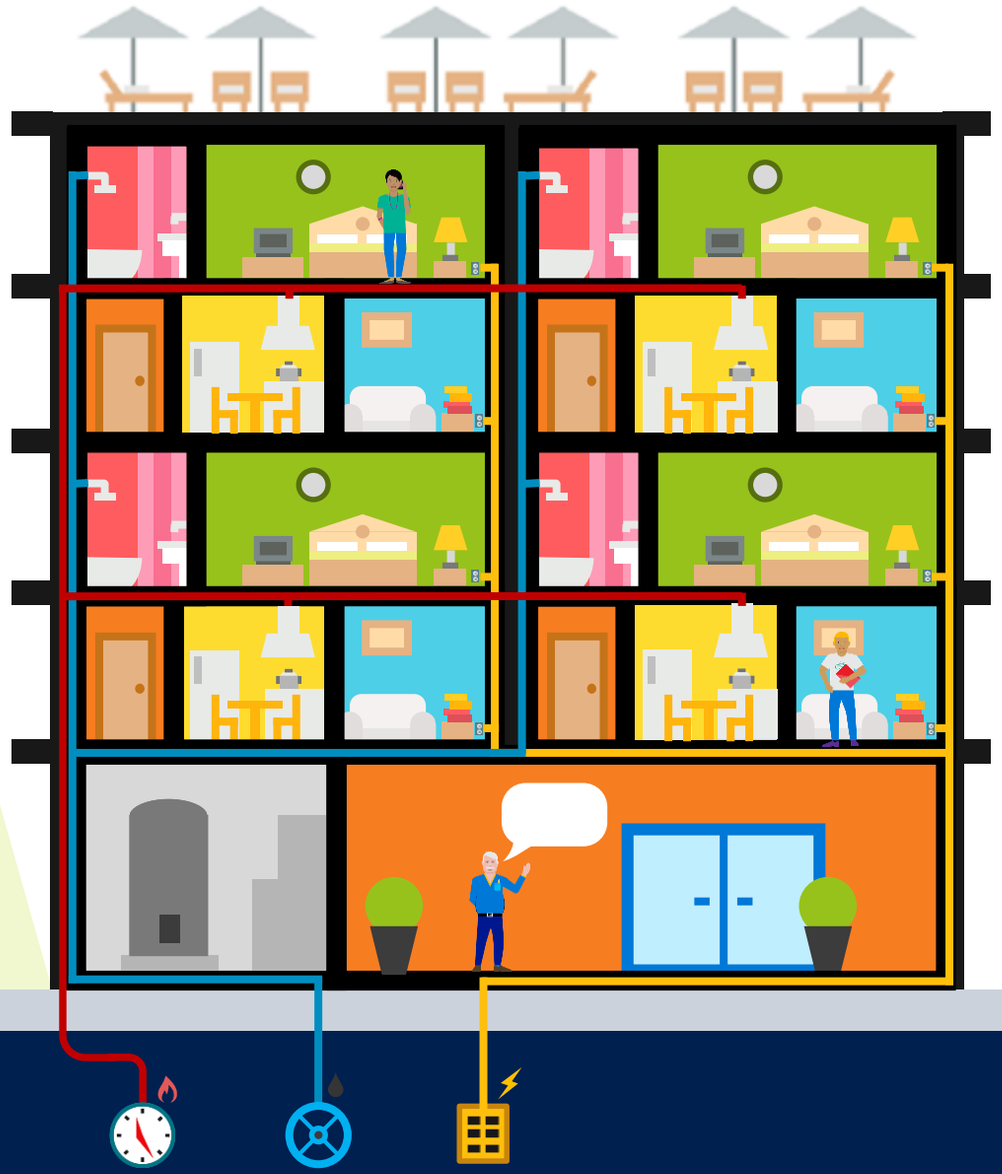
What is a Container?

Microsoft

KONFERENCA

Virtual Machine (VM) vs. Container

Virtual Machine (VM) vs. Container



Virtual Machine (VM) vs. Container

Containers – They cover it all

Any OS	 Linux	 Windows
Anywhere	 On-premises	 Hosted
Any app	 Monolith	 Microservice
Any language	 .NET	 Java

Arsenal

- ✓ Linux Containers, Windows Server Containers, and Hyper-V Isolation
- ✓ Run on-premises, on Azure, on other public clouds, service provider clouds
- ✓ Monoliths, Microservices, and other app types
- ✓ Development frameworks and environments supported: Microsoft and ecosystem

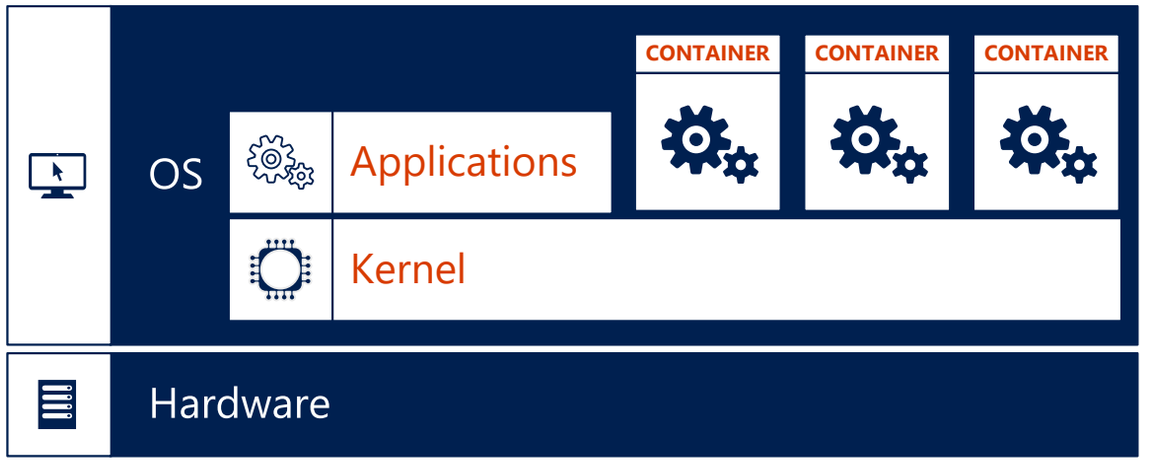
Windows Containers

Microsoft

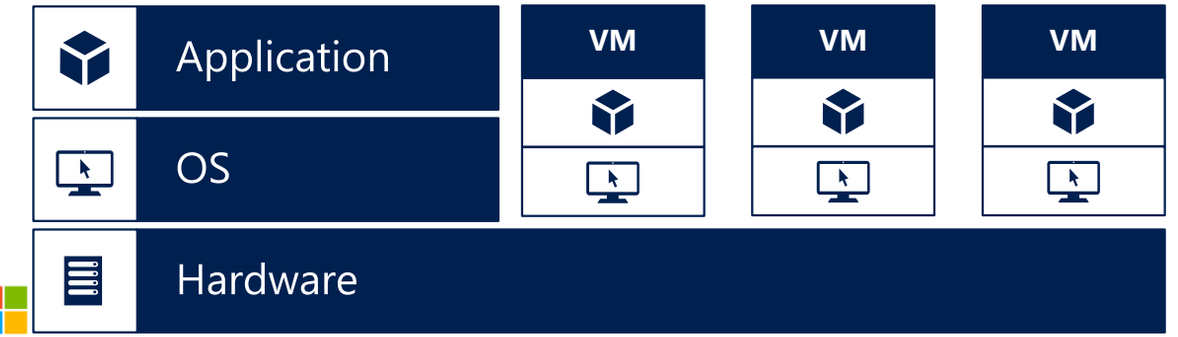
KONFERENCA

What is a Container?

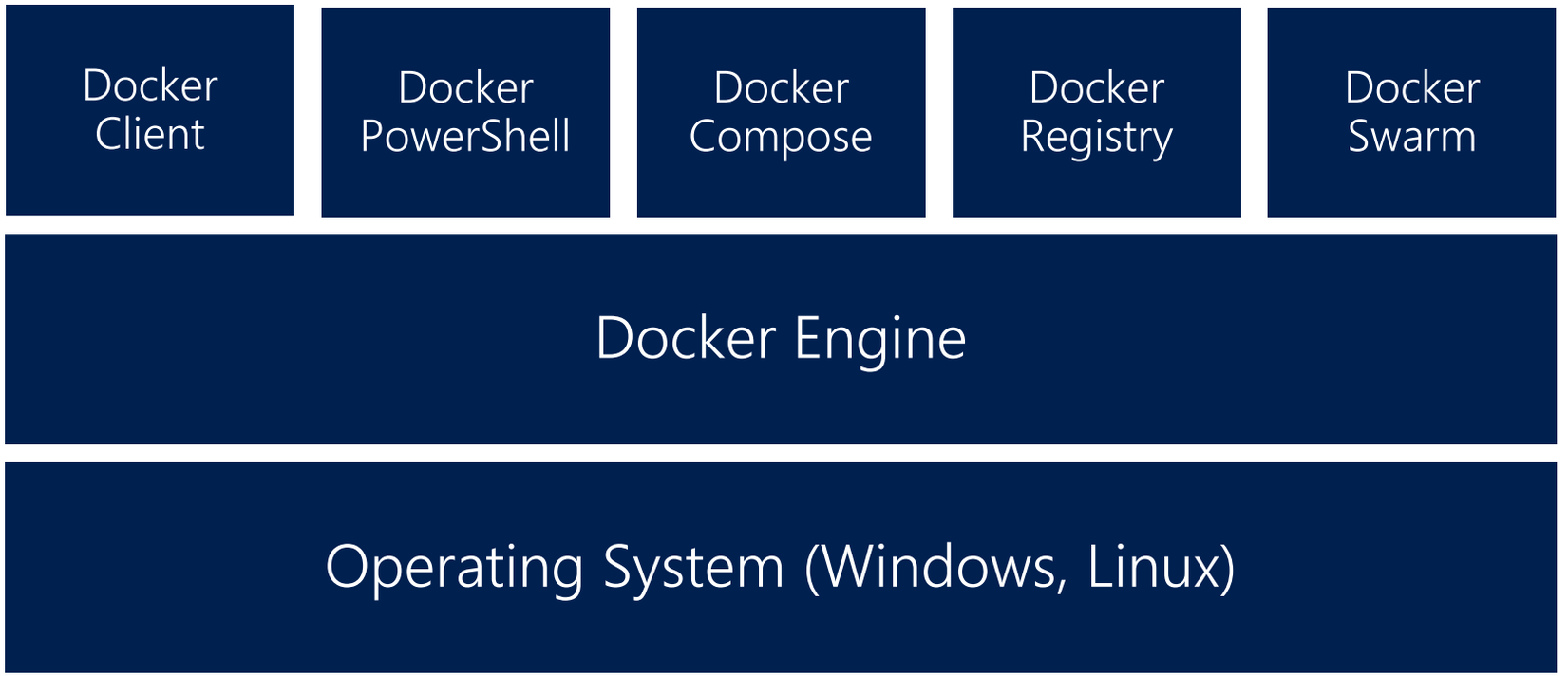
Containers = Operating system virtualization



Traditional virtual machines = hardware virtualization



How Do Containers Work?

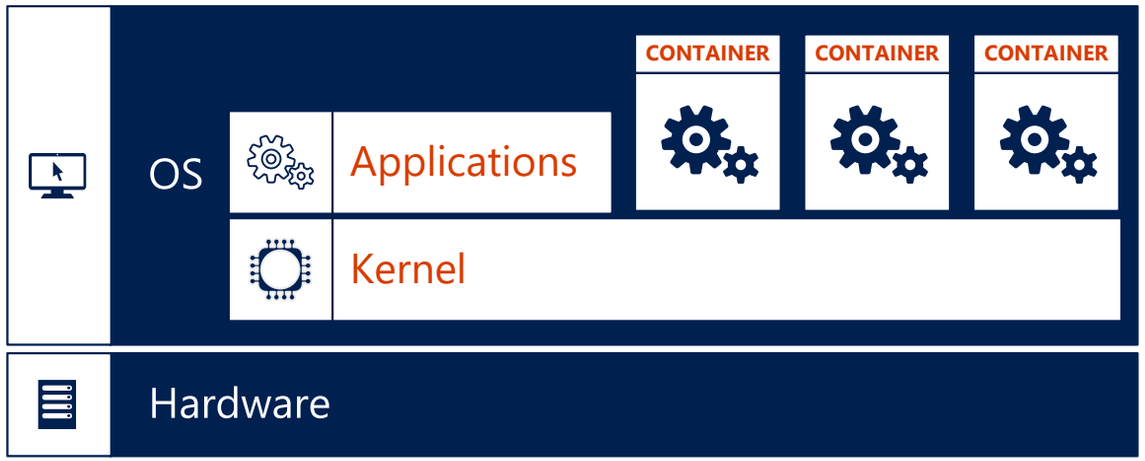


Container Development and Management Toolset

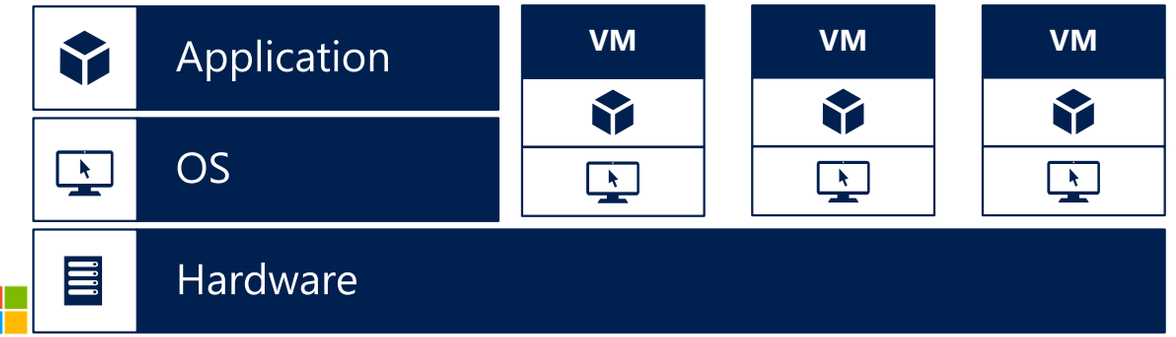
Container Runtime

What is a Container?

Containers = Operating system virtualization

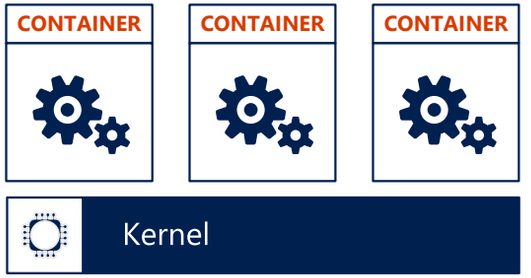


Traditional virtual machines = hardware virtualization



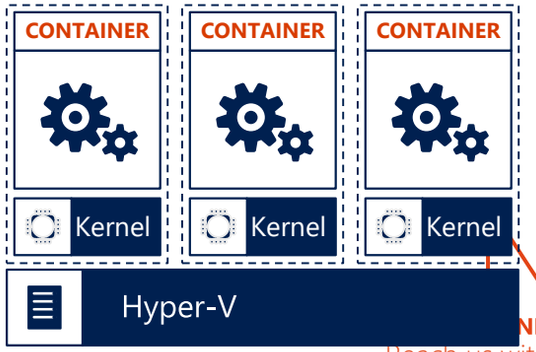
Windows Server Containers

Maximum speed and density



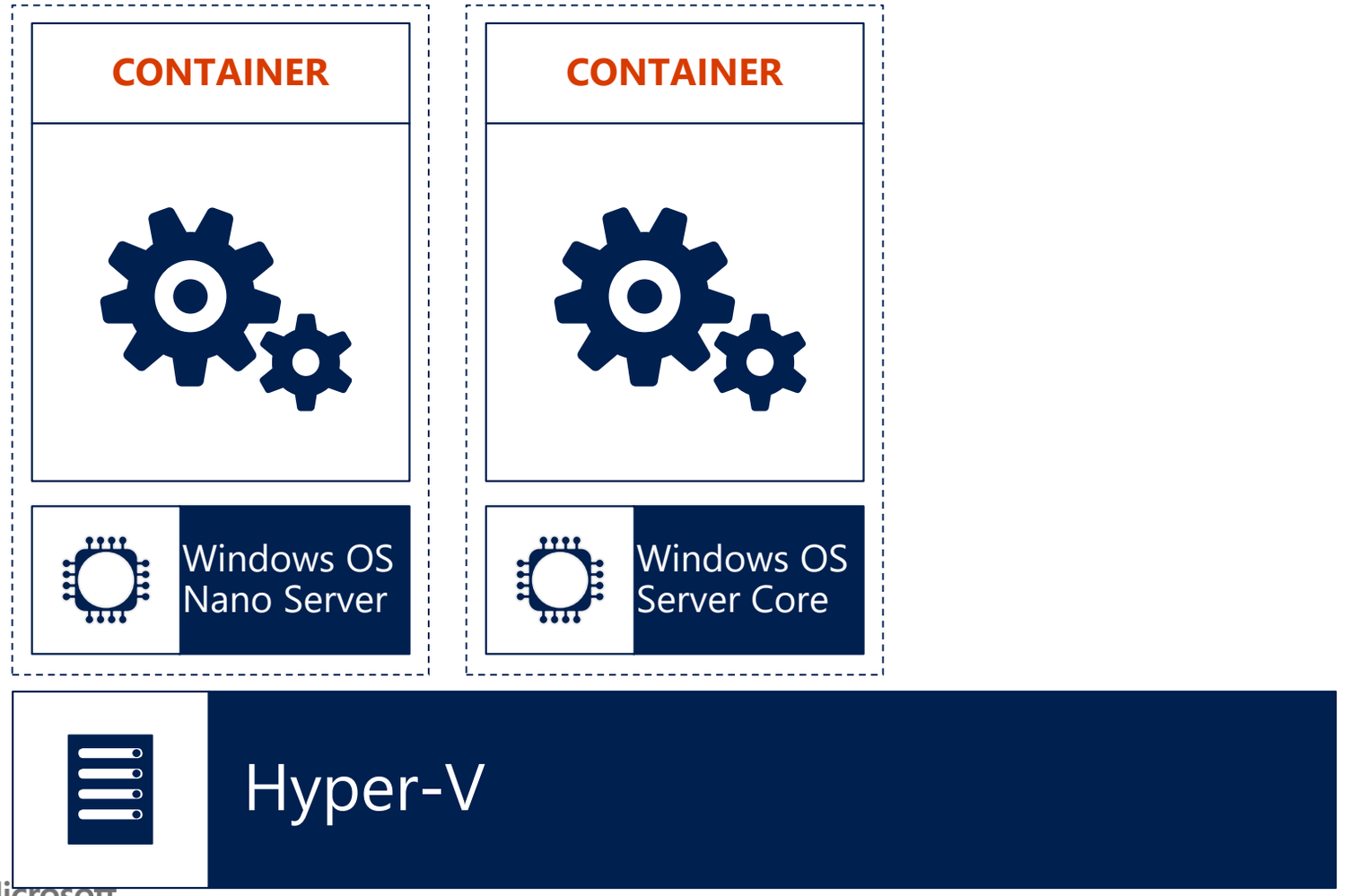
Hyper-V Containers

Isolation plus performance



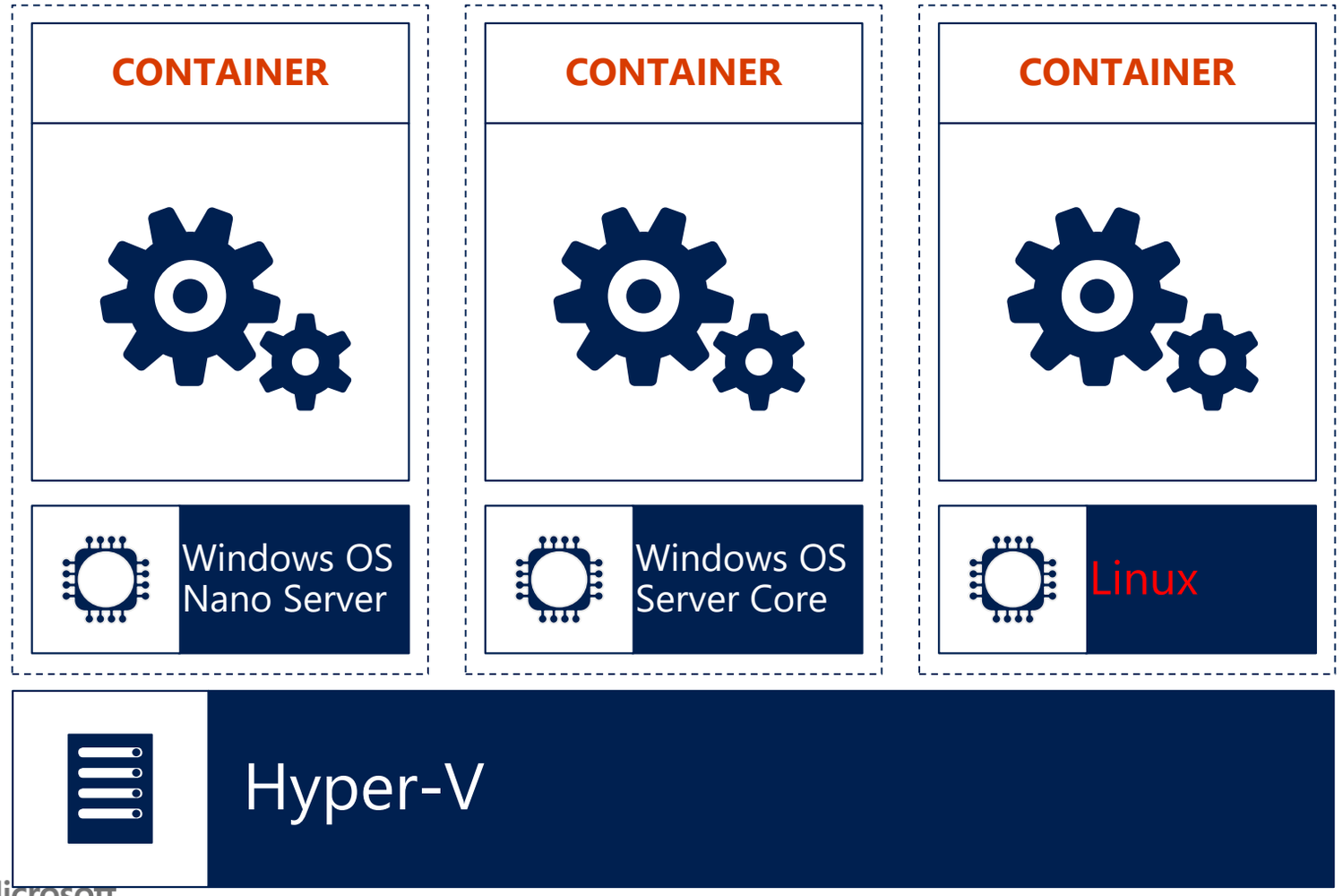
Hyper V Containers

Support different operating systems within containers



Hyper V Containers

Support different operating systems within containers



Even mixing Windows and Linux containers on the same host!

Container Images

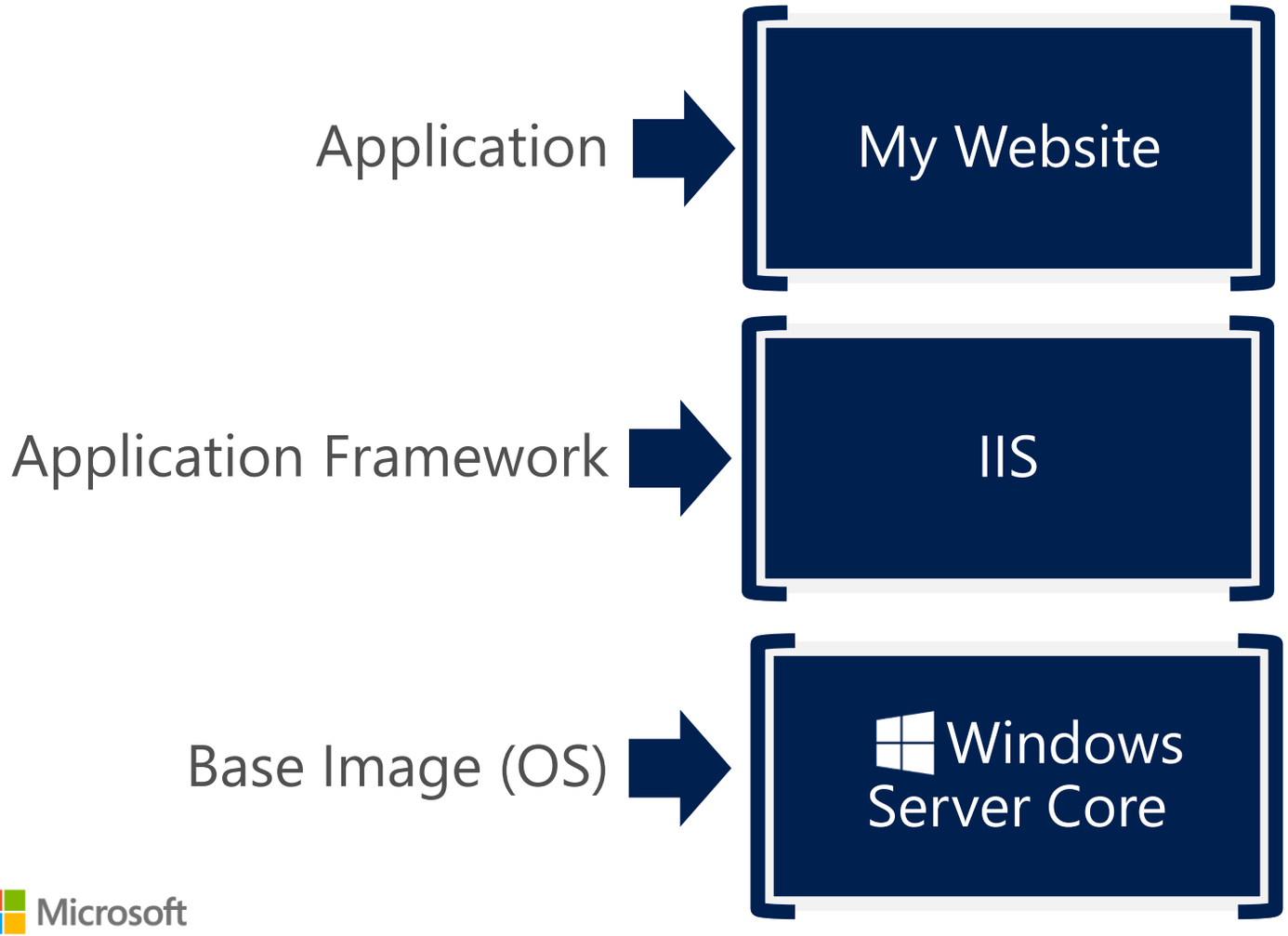


Image Registries

What is a registry?

- Stores container images
  - Images are Pushed into a registry
  - Images are Pulled from a registry
  - Images are Searched for within a registry

Development and Containers

Microsoft

KONFERENCA

Developers and Containers

Build Once, Run Anywhere

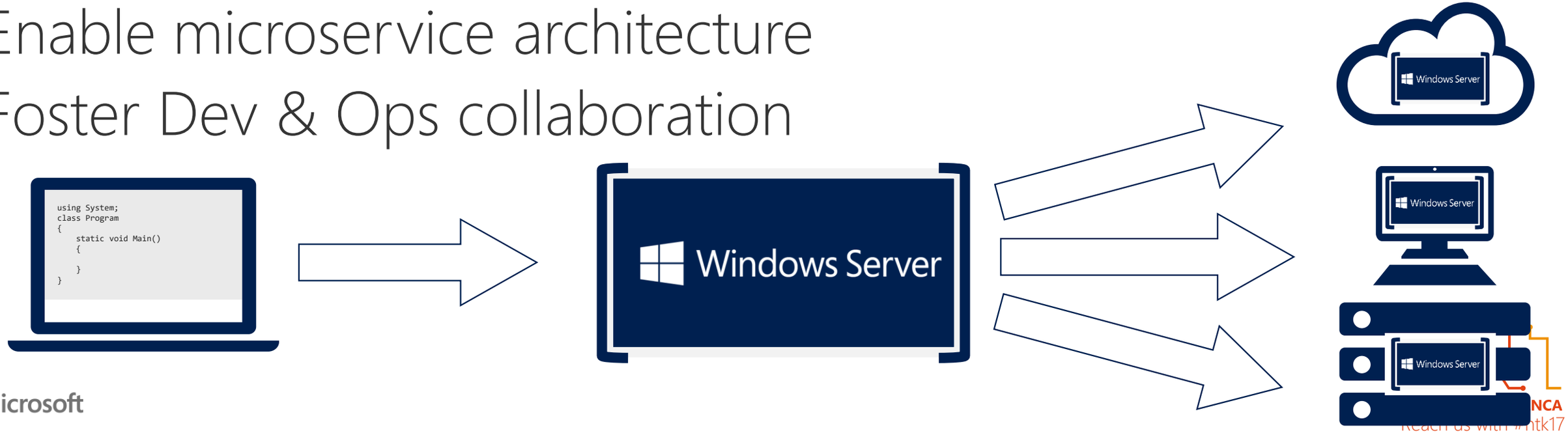
Easy to create, share and distribute

Start immediately, scale easily

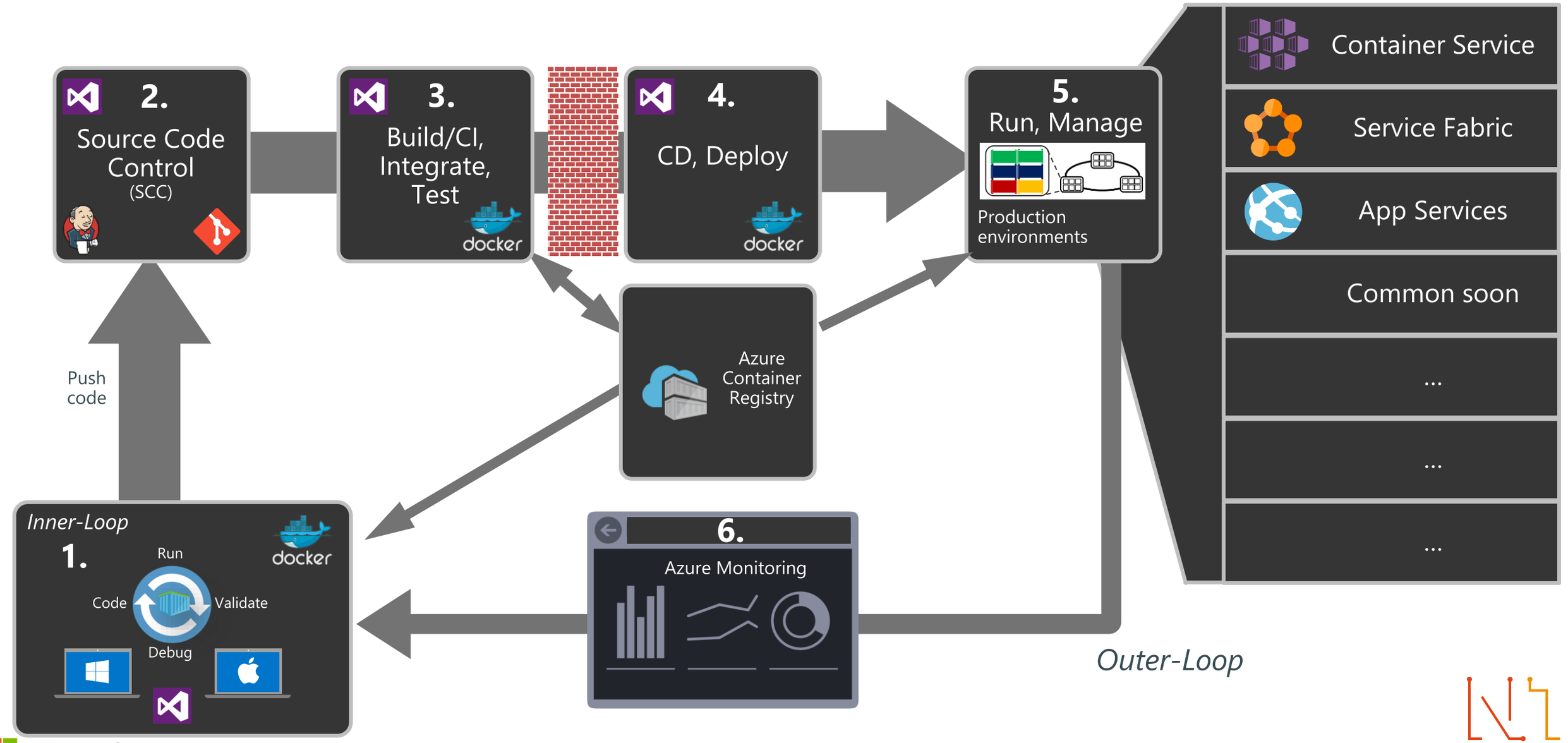
Always consistent - include all prerequisites

Enable microservice architecture

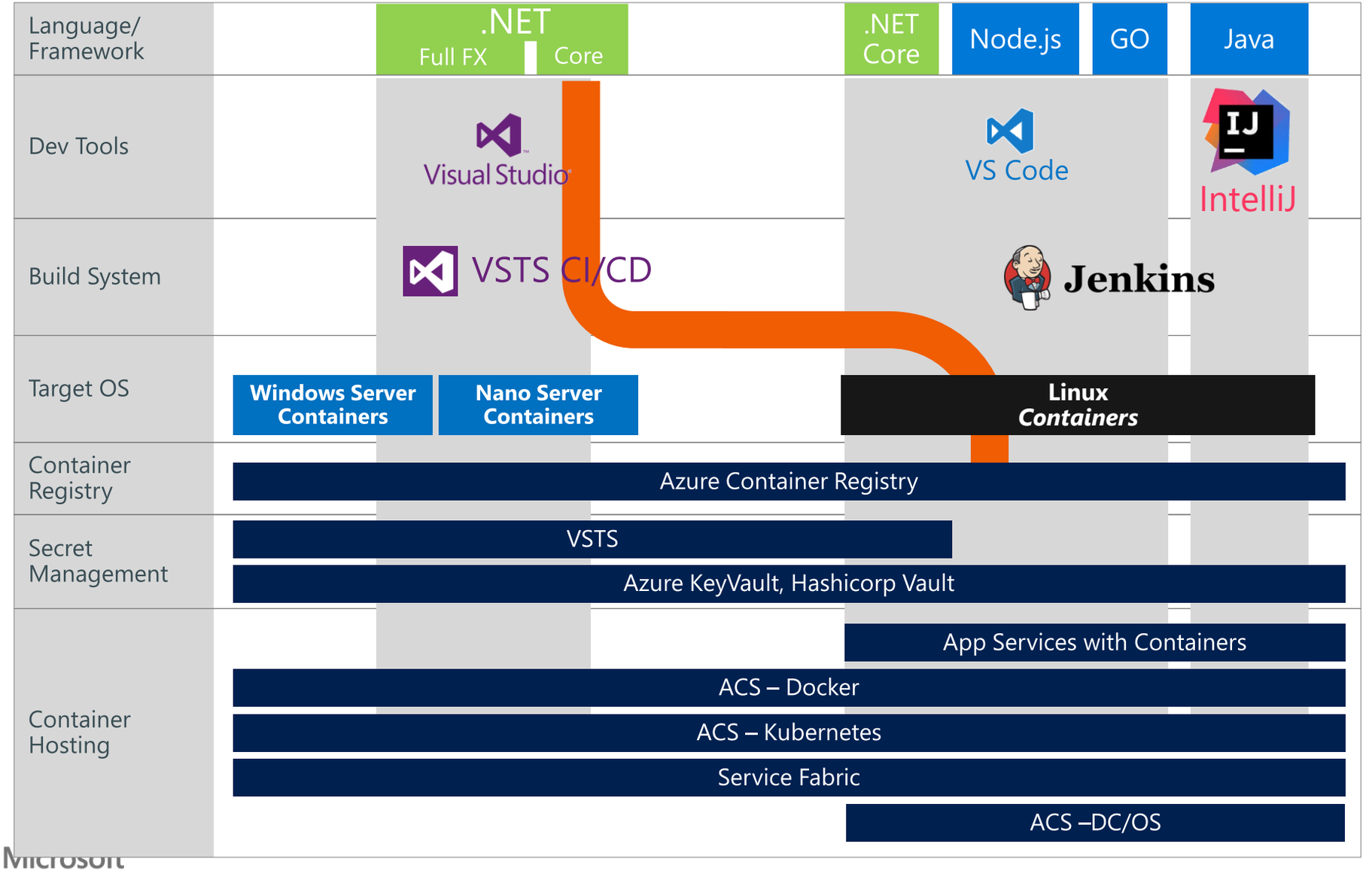
Foster Dev & Ops collaboration



Containerized workflow

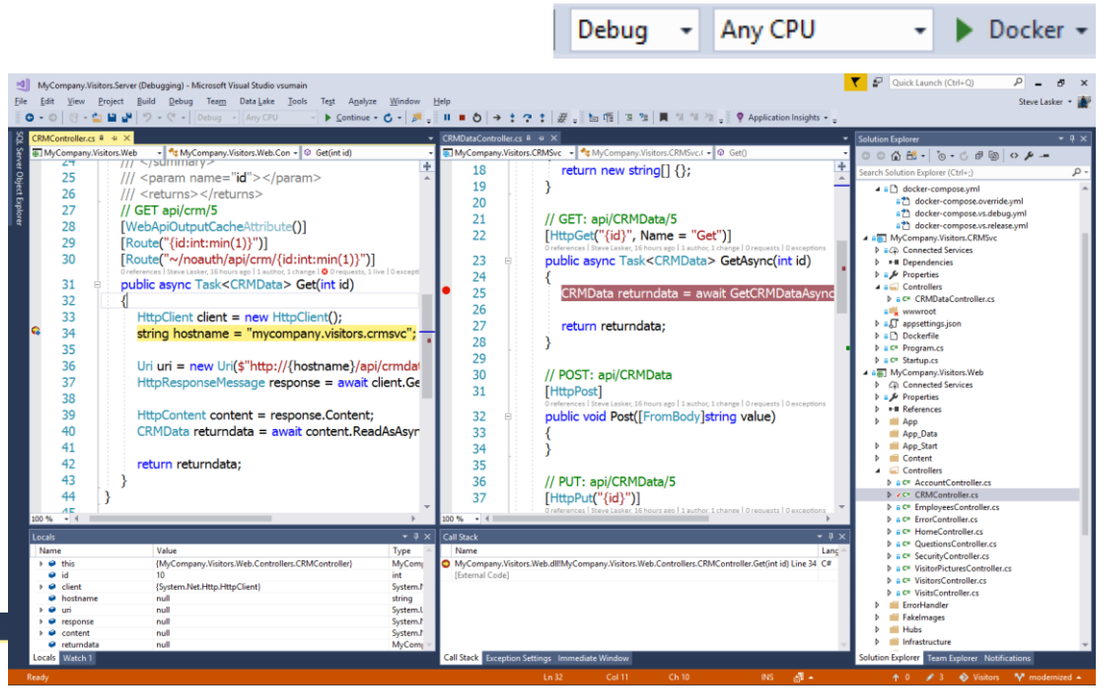


Microsoft Container Integrations



Visual Studio 2017 Container Tools

- Run, debug, test web, and console apps in Docker Containers
  - .NET FX w/Windows Server, .NET Core with Nano Server, and Linux
- Breakpoint debugging
- Edit & refresh of code
- Scaffolds Docker assets
  - Dockerfile, docker-compose.yml



Dockerfile

```
FROM microsoft/aspnet:4.6.2
ARG source
WORKDIR /inetpub/wwwroot
COPY ${source:-obj/Docker/publish} .
EXPOSE 80
```

Dockerfile*

```
FROM microsoft/aspnetcore:1.1
ARG source
WORKDIR /app
EXPOSE 80
COPY ${source:-obj/Docker/publish} .
ENTRYPOINT ["dotnet", "Api.dll"]
```

docker-compose.yml

```
version: '3'

services:
  web:
    image: stevelasdemos.azurecr.io/web
    build:
      context: ./Web
      dockerfile: Dockerfile

  api:
    image: stevelasdemos.azurecr.io/api
    build:
      context: ./Api
      dockerfile: Dockerfile
```

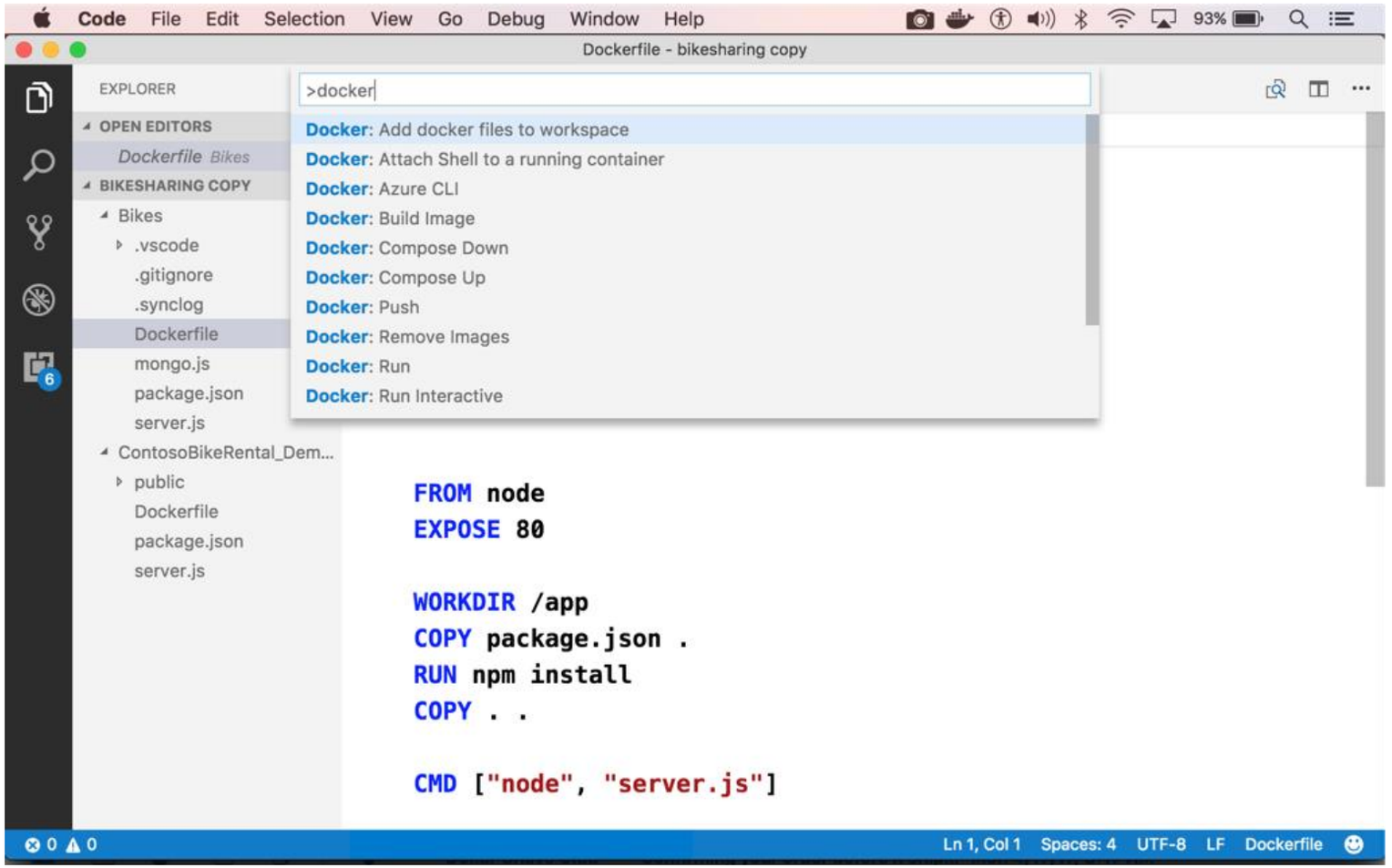
Microsoft

Linux and Windows Containers Containerize Application Demo

Microsoft

KONFERENCA

Visual Studio code



OS X, Linux, Windows

Node, Python, Go, .NET Core

Git SCC integration

Extensible platform

Docker Asset Editing – extension

aka.ms/VSCode

aka.ms/Dock`erToolsForVSCode

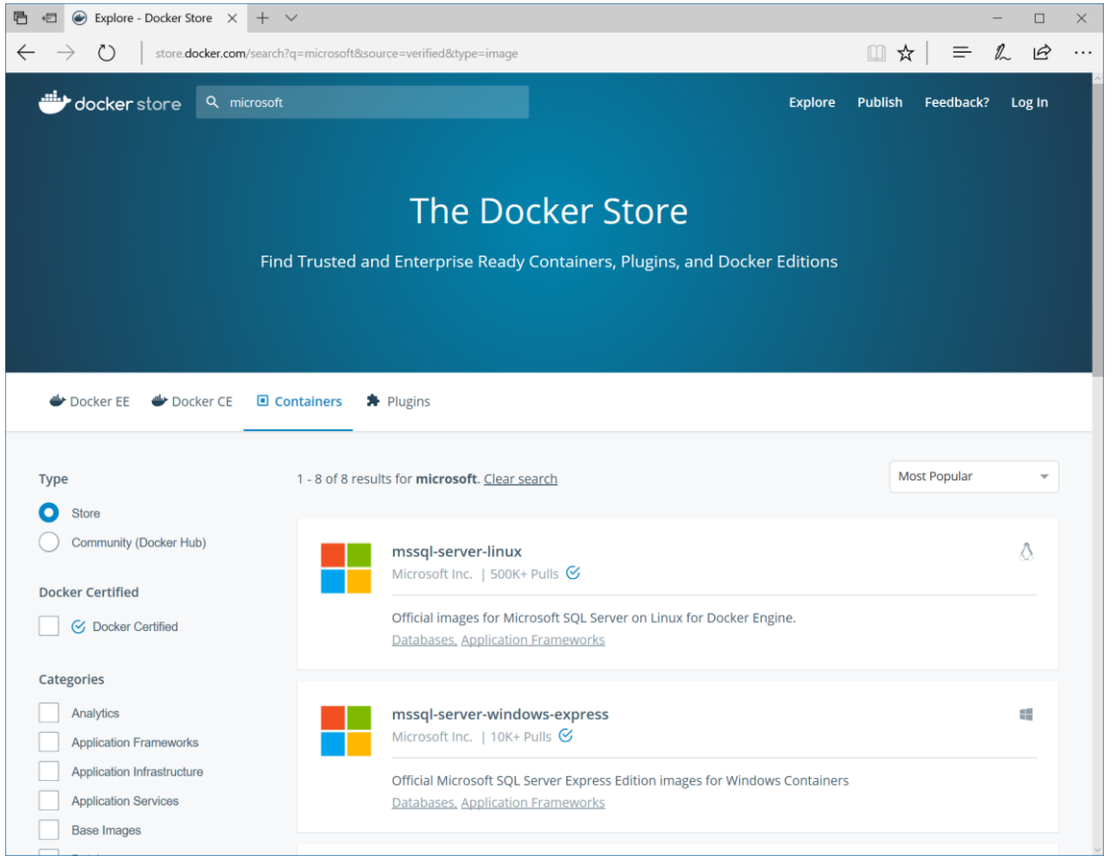
Microsoft

KONFERENCA

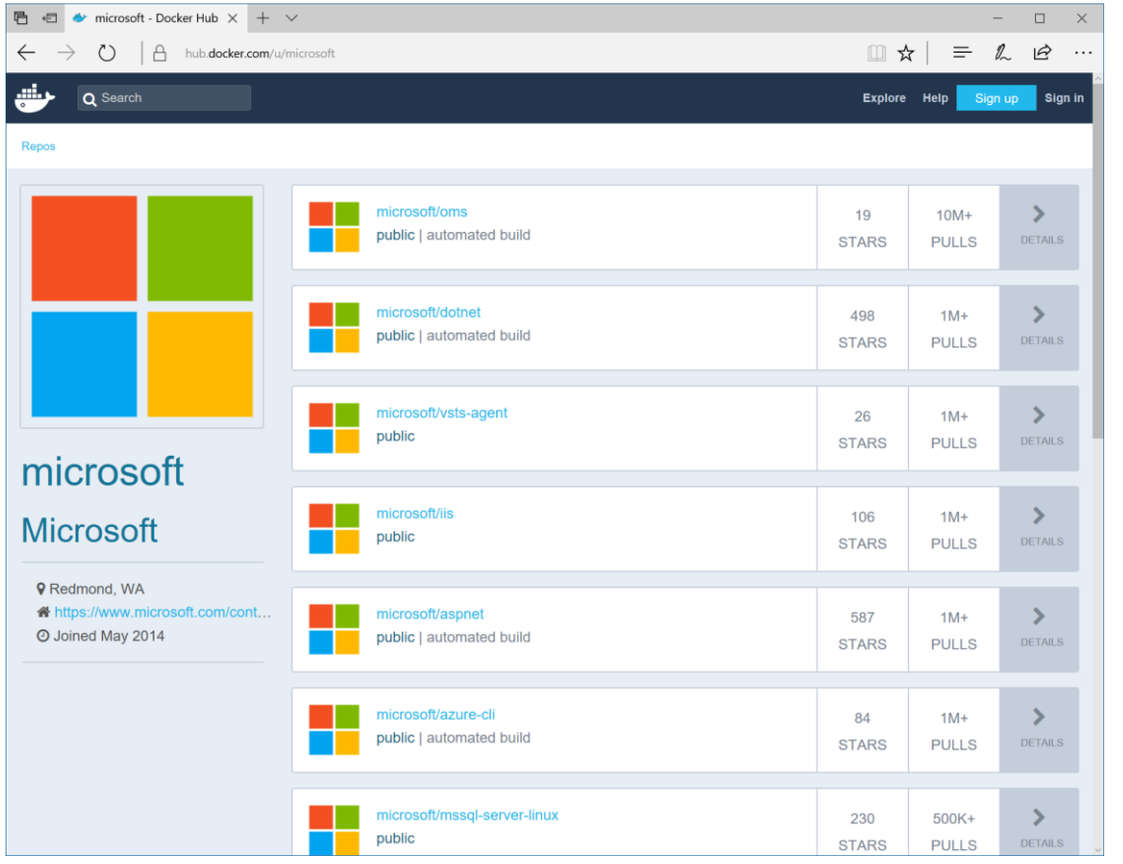
Reach us with #ntk17

Maintained images by Microsoft

https://store.docker.com/



https://hub.docker.com/u/microsoft/



Containerize Release Pipeline

Microsoft

KONFERENCA

How Containers Influence CI/CD?

CI/CD used to be about deploying code

- You pushed your code to each environment
- You hoped your new code didn't conflict with the old code

Container Deployment

- You push new OS Instances, with your code
- Toss out your old instance - Immutable

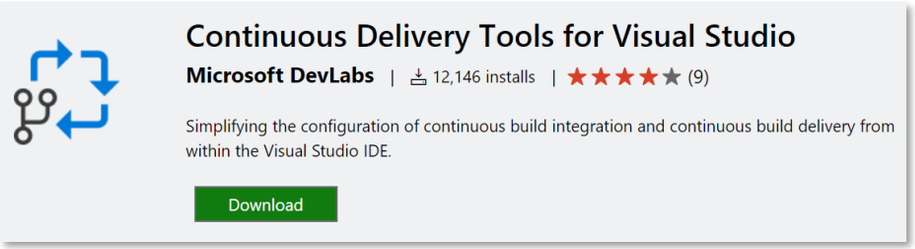
Continuous Delivery Tools for Visual Studio

Create Continuous Delivery pipeline easily

Creates automated build and release management workflows

Delivery in the cloud

- ASP.Net projects to Azure App Service
- .Net Core to Azure Container Service



http://aka.ms/CD4VS

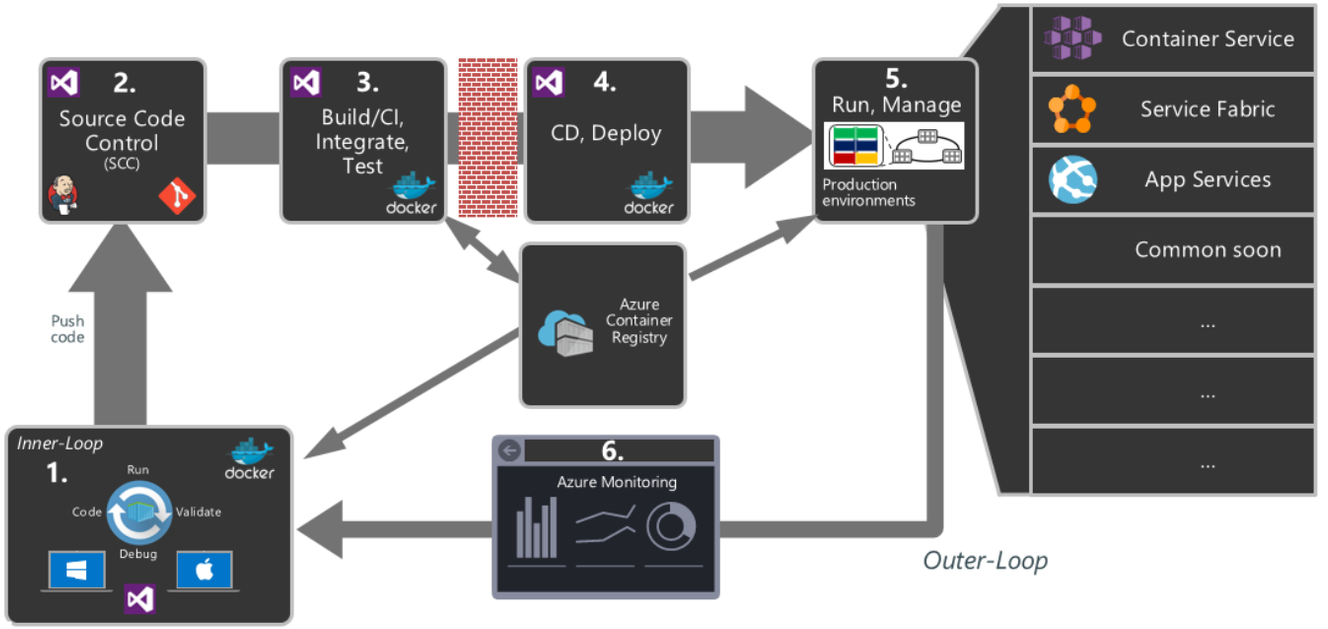
Instant Continuous Delivery To Containers in the Cloud Demo

Microsoft

KONFERENCA

Release Management and Containers

Any OS	Linux	Windows
Anywhere	On-premises	Hosted
Any app	Monolith	Microservice
Any language	.NET	Java



Thank you!

Questions?

Ognjen Bajić obajic @ ekobit.hr
Ana Roje Ivančić aroje @ ekobit.hr

Microsoft

KONFERENCA

Microsoft

KONFERENCA

PORTOROŽ, 15. DO 17. MAJ 2017