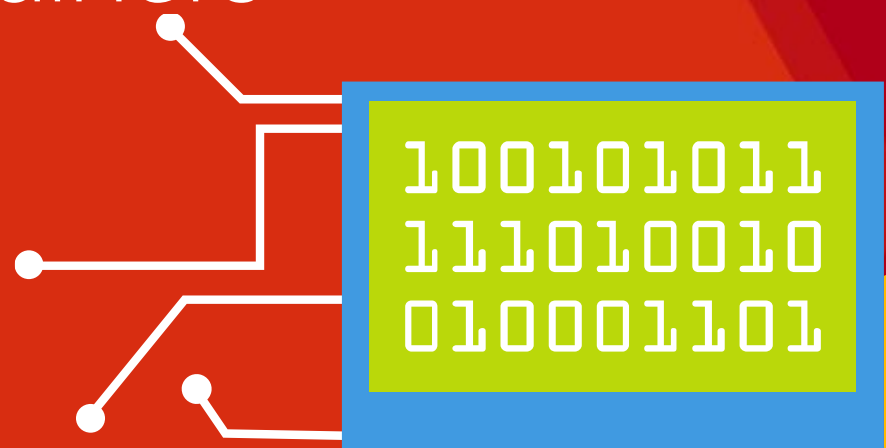




Release Management and Containers in the Cloud, on Windows and on Linux

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

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

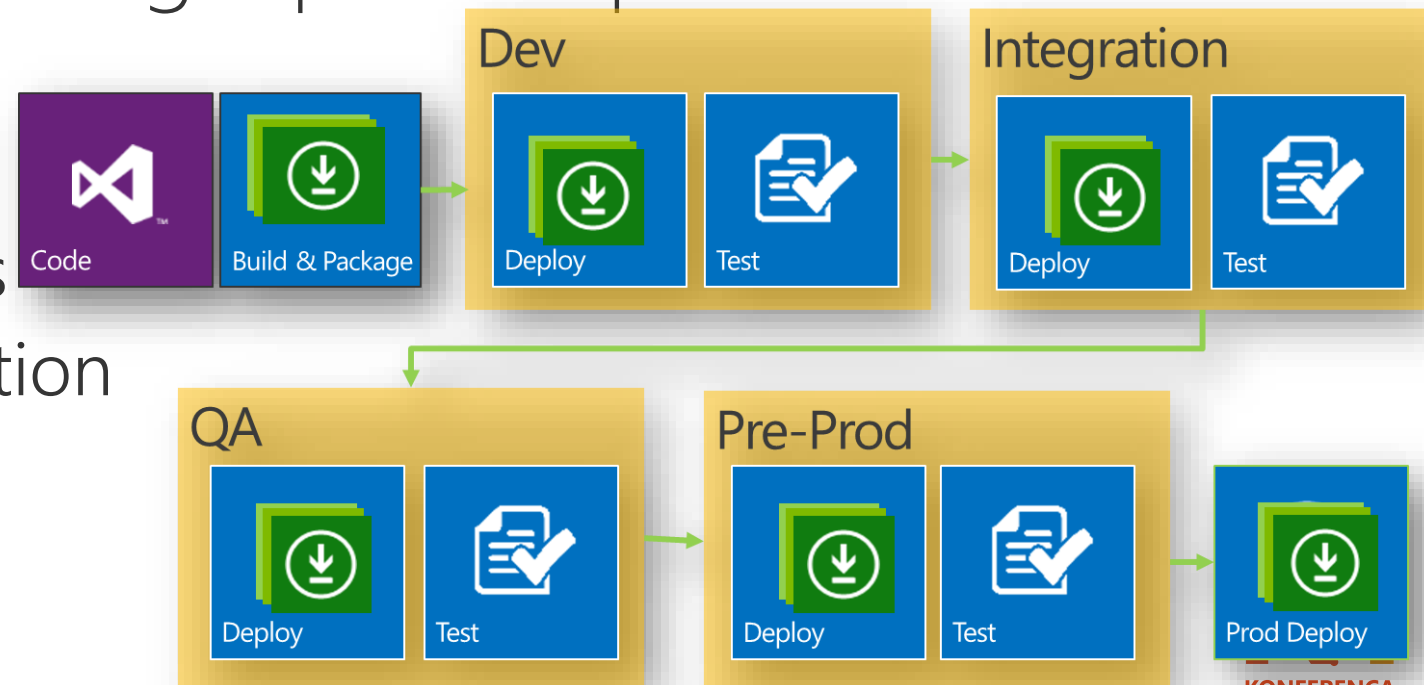
Push-button deployments

Deployment pipeline is the single path to production for all changes

Code

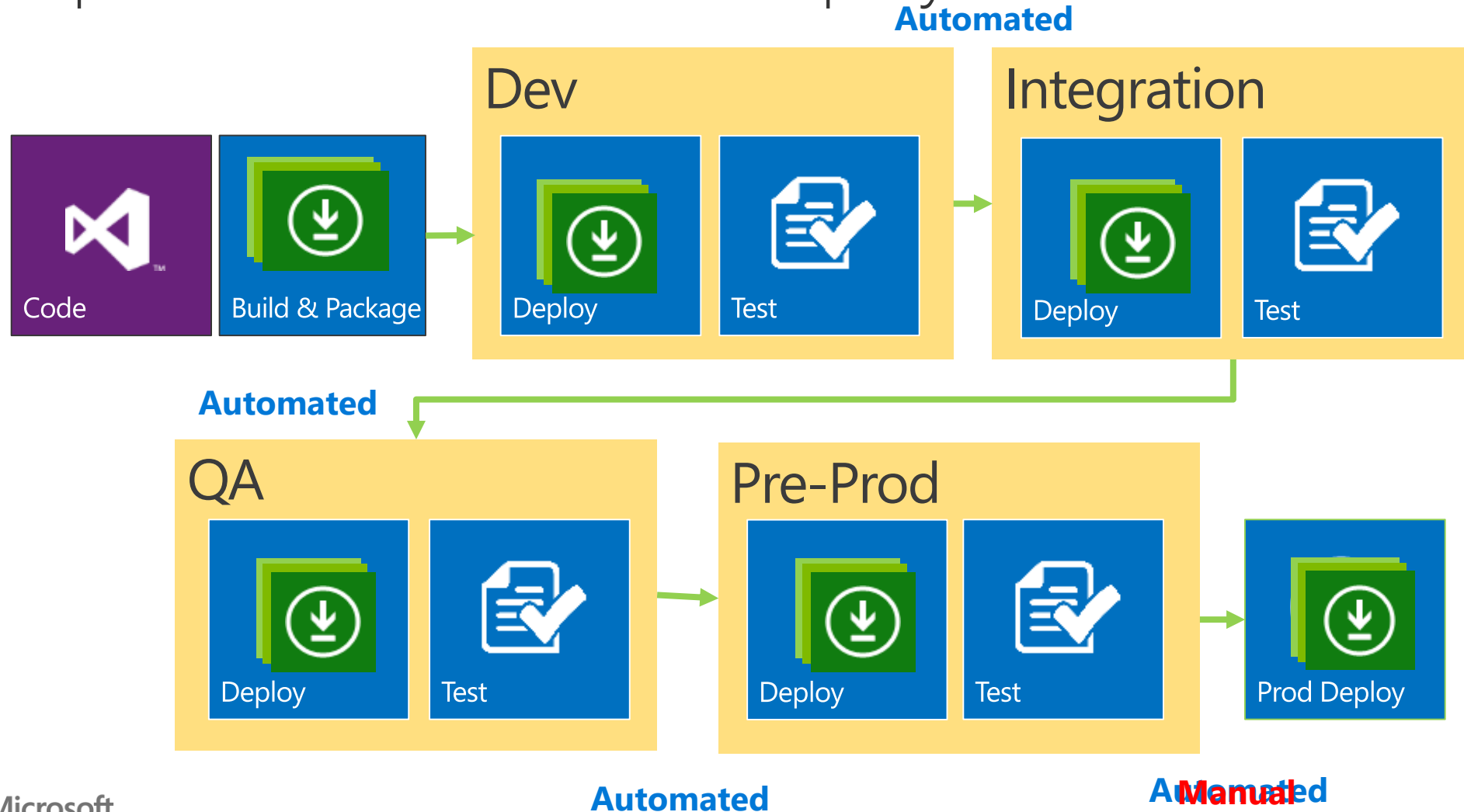
Infrastructure and environments

Database schemas or configuration

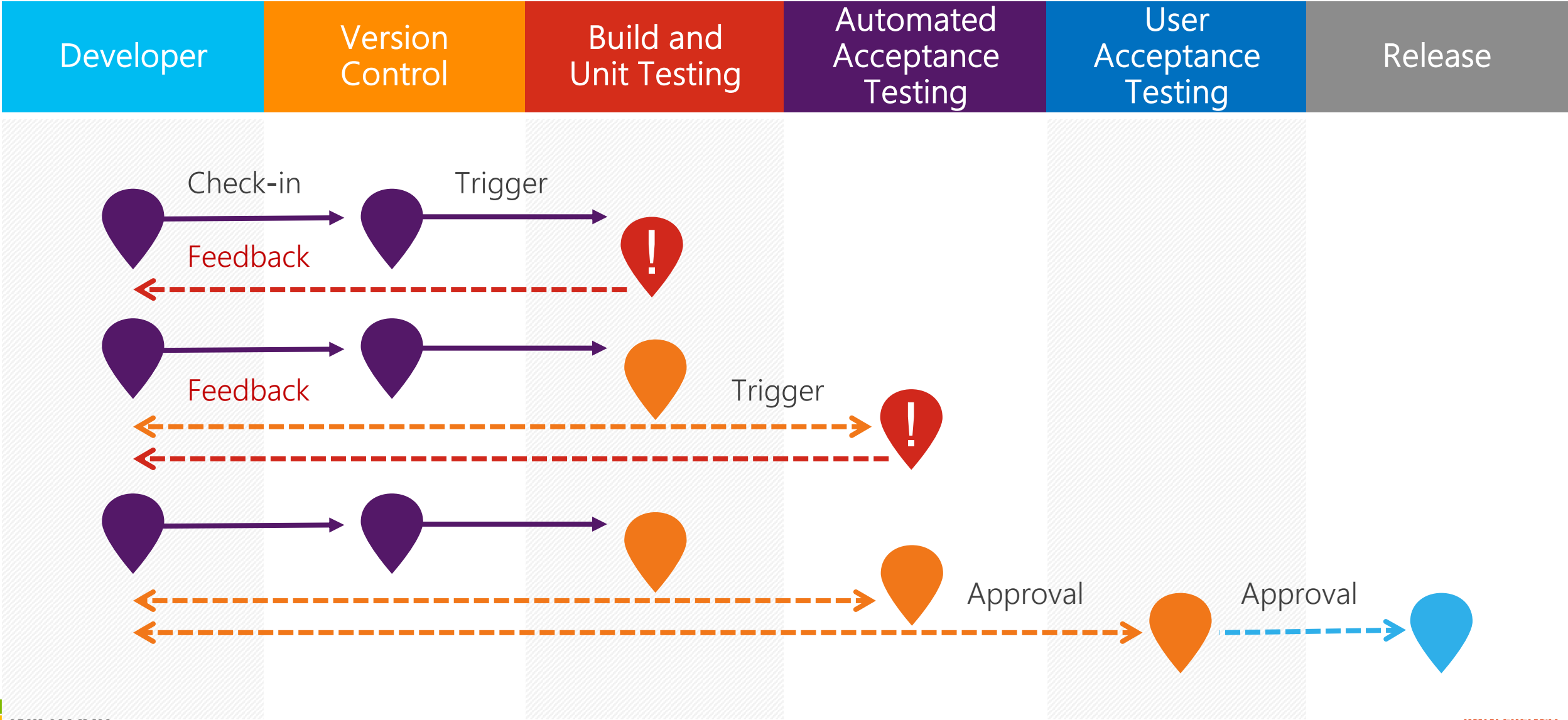


Release Management

Repeatable Automated Deployment Model



Continuous Quality for Continuous Value Delivery



Release Management in TFS/VSTS

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

The screenshot displays the Release Management interface. On the left, a 'Select a template' panel lists various build templates under 'Featured' and 'Others' categories. The 'Featured' section includes Visual Studio, ASP.NET (PREVIEW), Universal Windows Platform, Azure Cloud Services, Azure WebApp, and Xcode. The 'Others' section includes CMake and Docker. On the right, a 'Build Succeeded' notification bar is visible, indicating that the build ran for 17 seconds (VSALMTest) and was completed 18 days ago. Below this, a 'Logs' section shows the build process steps and their execution times.

Select a template
Select a build process template to add a set of tasks and apply typical settings.

Featured

- Visual Studio**
Build and run tests using Visual Studio. This template requires that Visual Studio be installed on the build agent.
- ASP.NET (PREVIEW)**
Build ASP.NET web applications

Others

- Universal Windows Platform**
Build Universal Windows Platform applications using Visual Studio. This template requires that Visual Studio and the Universal templates are installed on the build agent.
- Azure Cloud Services**
Build, package, test and deploy your Azure Cloud Service.
- Azure WebApp**
Build, package, test and deploy your Azure WebApp.
- Xcode**

Add tasks
Don't see what you need? [Check out our Marketplace.](#)

Build Succeeded
Build Ran for 17 seconds (VSALMTest), completed 18 days ago

Logs

```
1 2017-04-24T14:51:44.6121133Z ##[section]Starting: Build
2 2017-04-24T14:51:44.6381293Z Current agent version: '2.105.7'
3 2017-04-24T14:51:44.9436652Z ##[section]Starting: Get Sources
4 2017-04-24T14:51:45.2393241Z Prepending Path environment variable with directory containing 'tf.exe'.
5 2017-04-24T14:51:45.2393241Z Querying workspace information.
6 2017-04-24T14:51:47.2922568Z ##[command]tf vc get /version:88 /recursive /overwrite C:\agent\_work\1\s /
7 2017-04-24T14:51:48.0366517Z C:\agent\_work\1\s\FabrikamFiber.CallCenter\FabrikamFiber.Web\Views\Shared:
8 2017-04-24T14:51:48.0366517Z Replacing _Layout.cshtml
9 2017-04-24T14:51:48.2085274Z ##[section]Finishing: Get Sources
10 2017-04-24T14:51:48.2085274Z ##[section]Starting: NuGet restore **/*.sln
```

Release Management for VSTS/TFS

Continuous Delivery

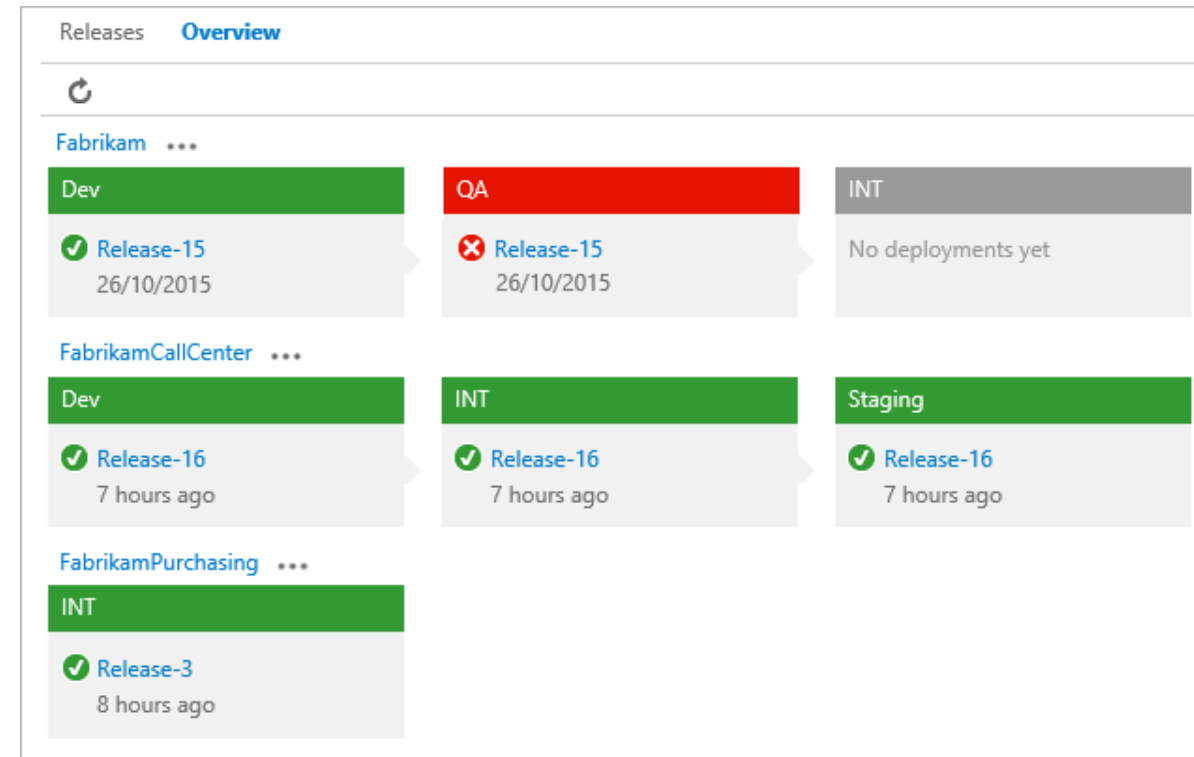
- Define per-environment release pipeline
- Trigger manually or via successful builds

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

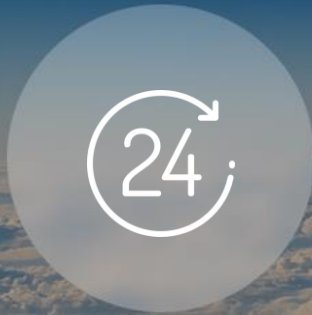
Easy customization and versioning

Provides analytics and reporting

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

Introduction to Containers

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

What is a Container?

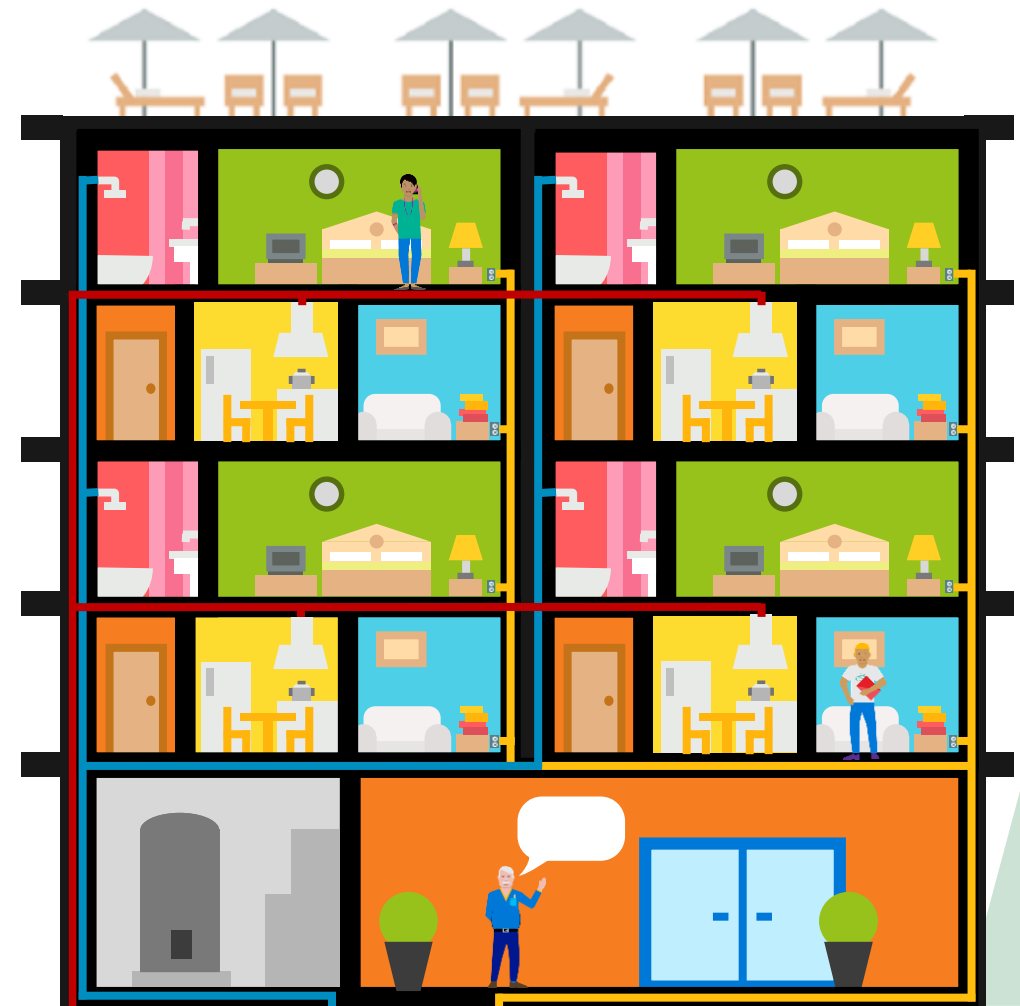
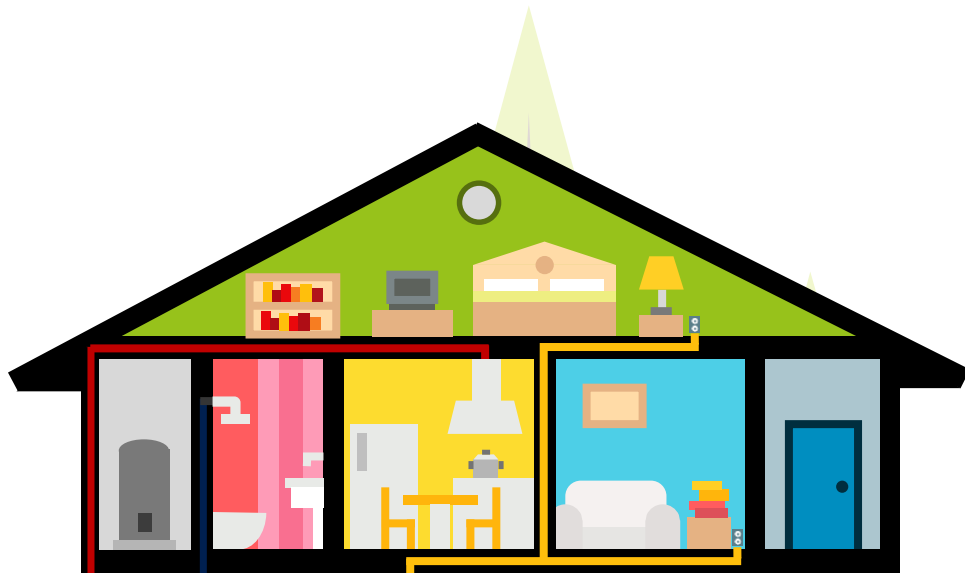
Virtual Machine (VM) vs. Container



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

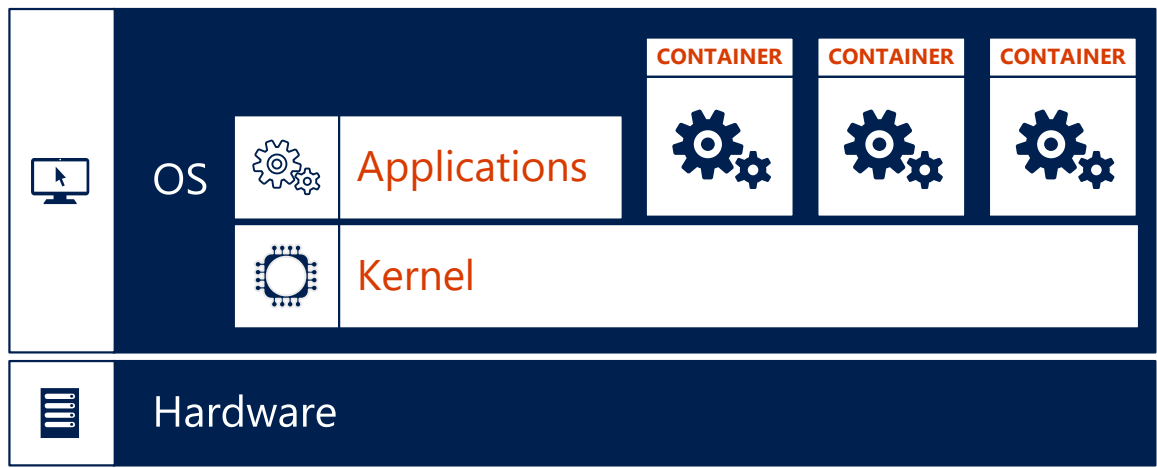
Arsenal

- ✓ 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

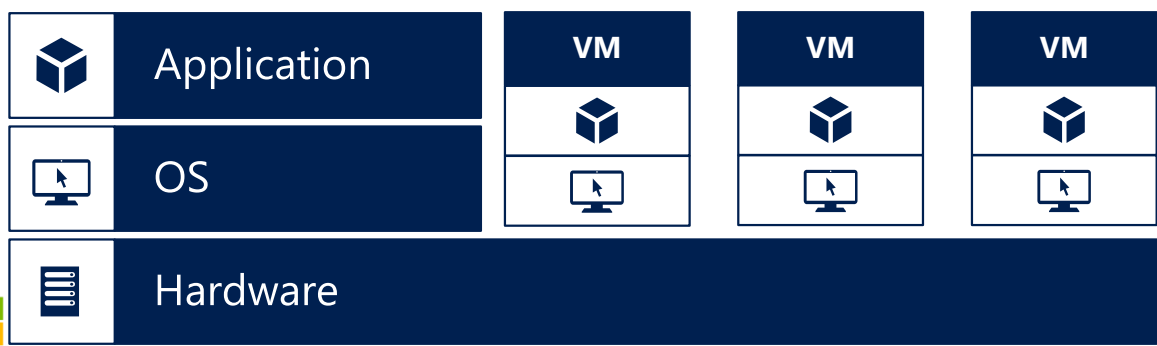
Windows Containers

What is a Container?

Containers = Operating system virtualization



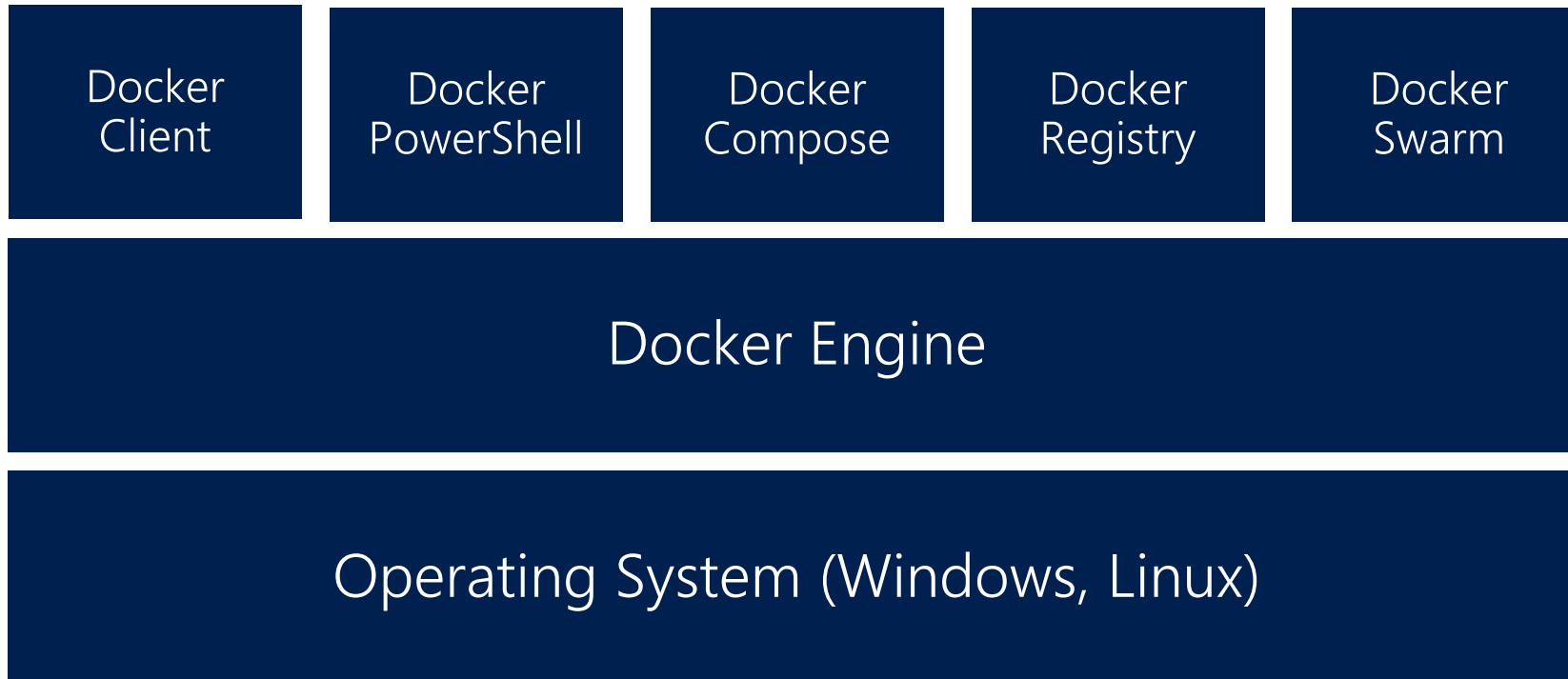
Traditional virtual machines = hardware virtualization



KONFERENCA

Reach us with #ntk17

How Do Containers Work?

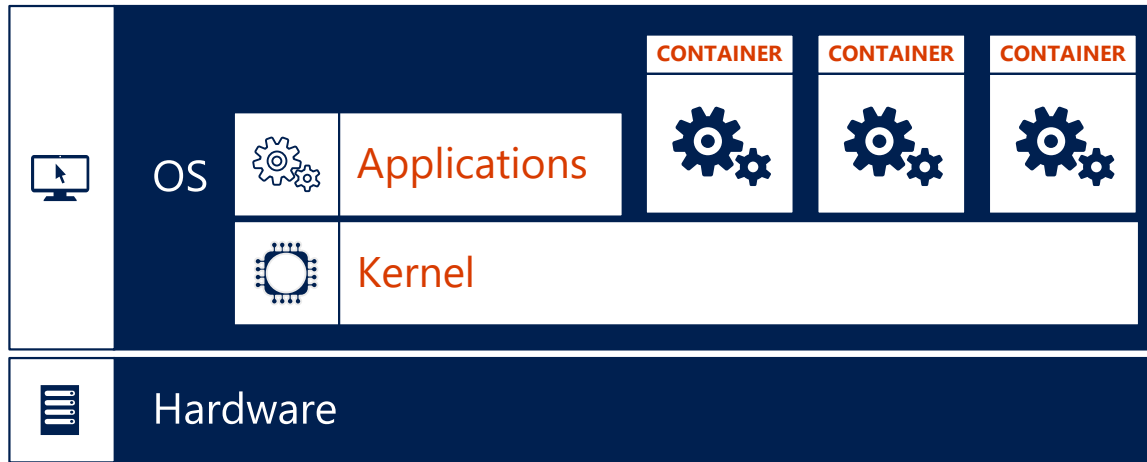


Container
Development and
Management Toolset

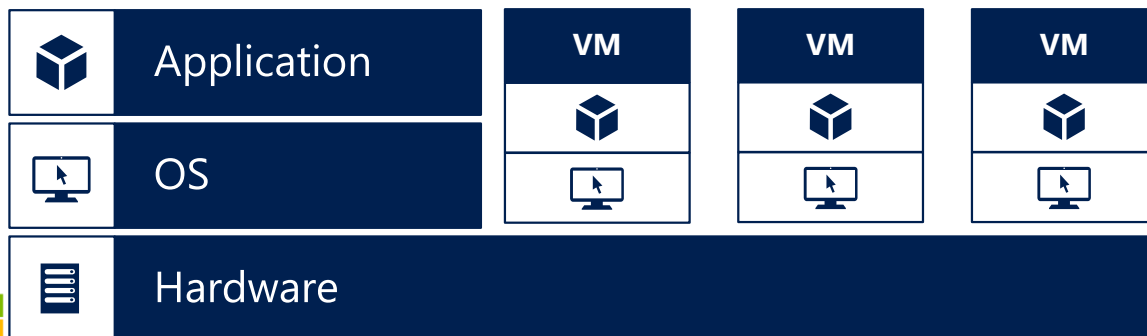
Container Runtime

What is a Container?

Containers = Operating system virtualization

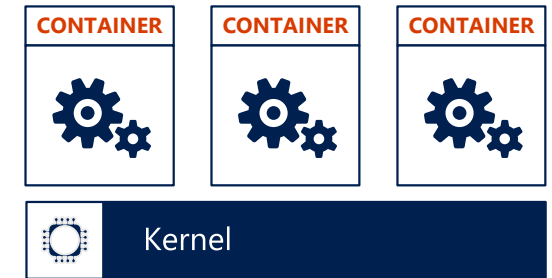


Traditional virtual machines = hardware virtualization



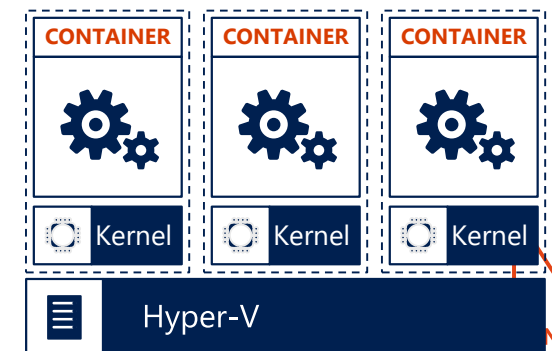
Windows Server Containers

Maximum speed and density



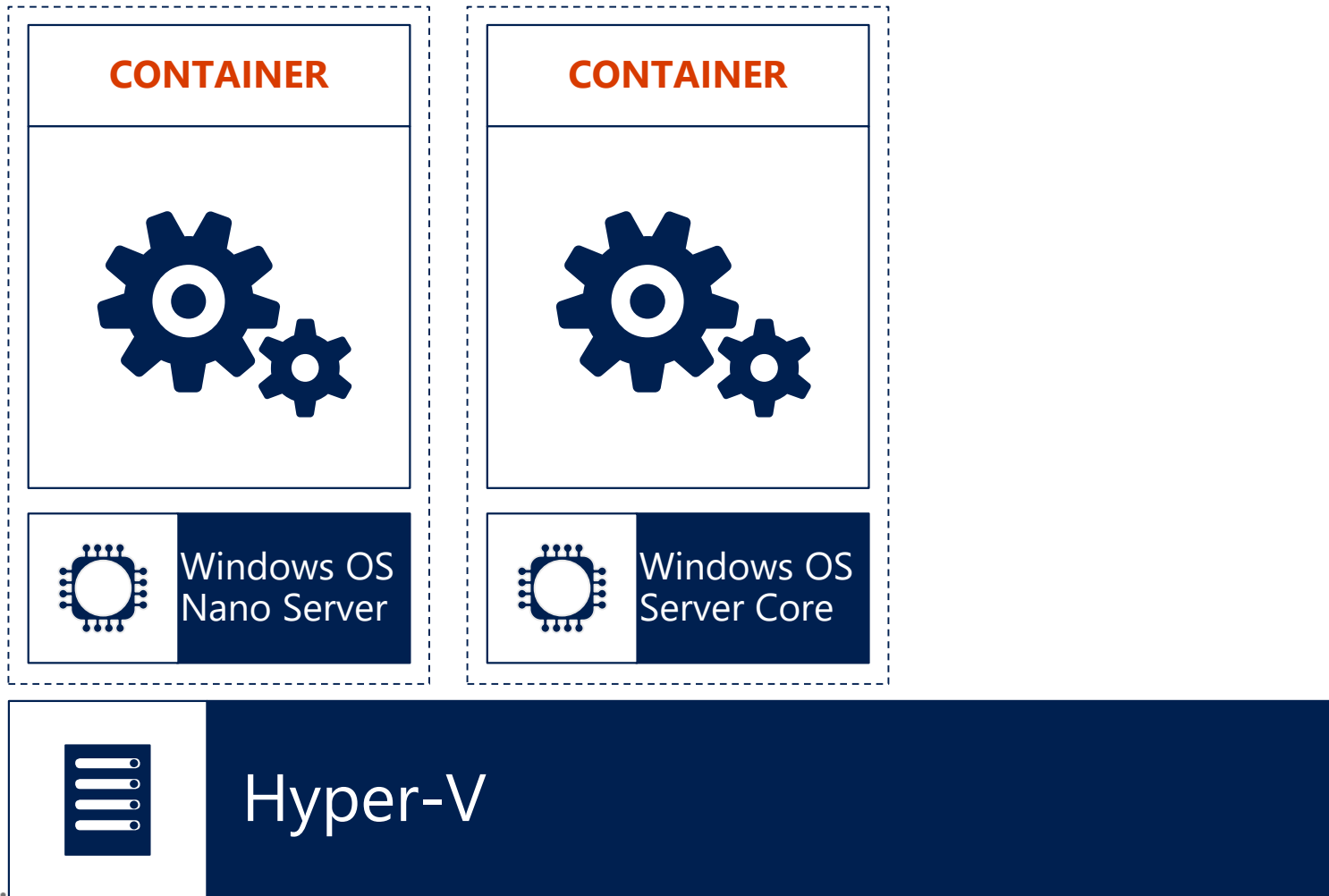
Hyper-V Containers

Isolation plus performance



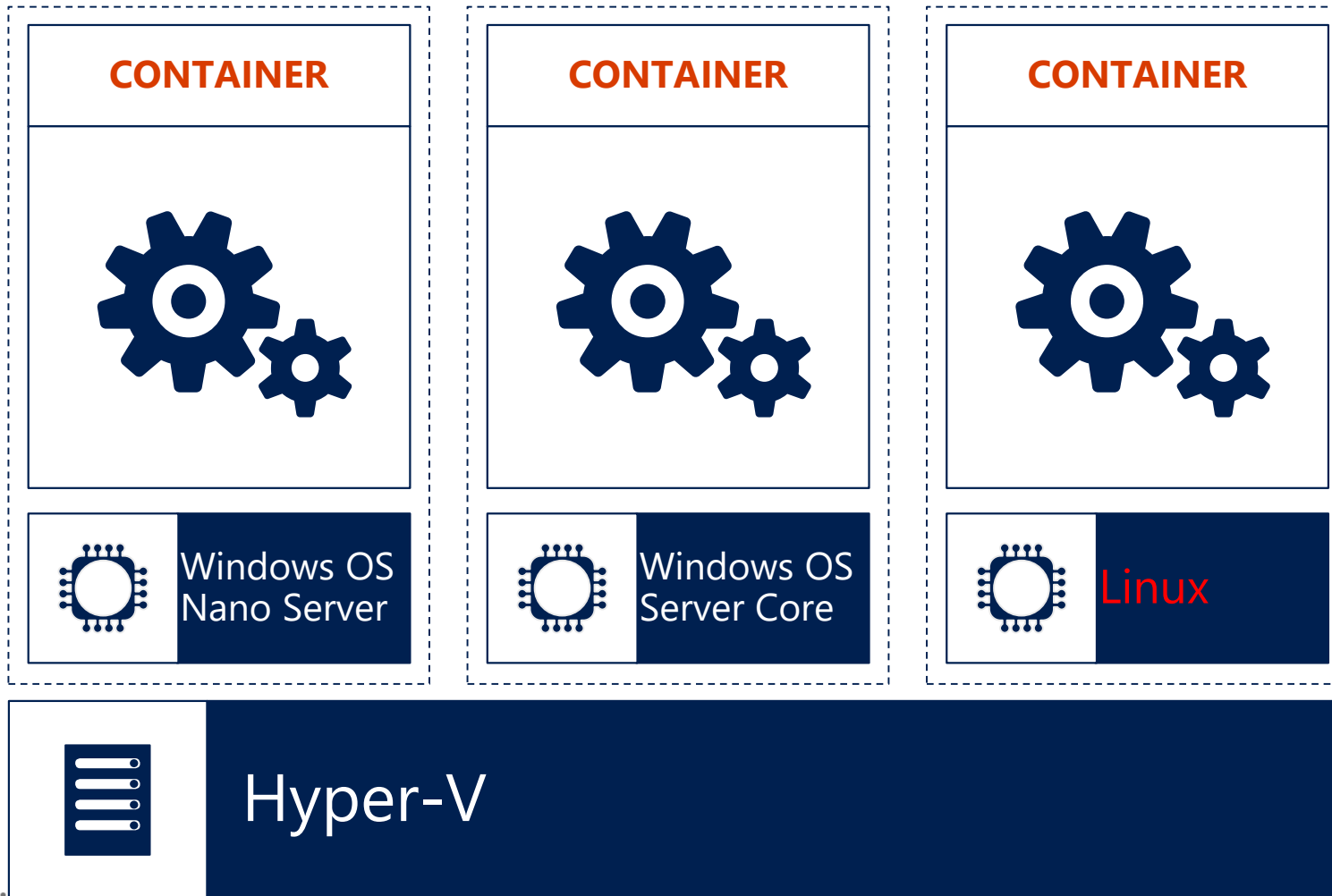
Hyper V Containers

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

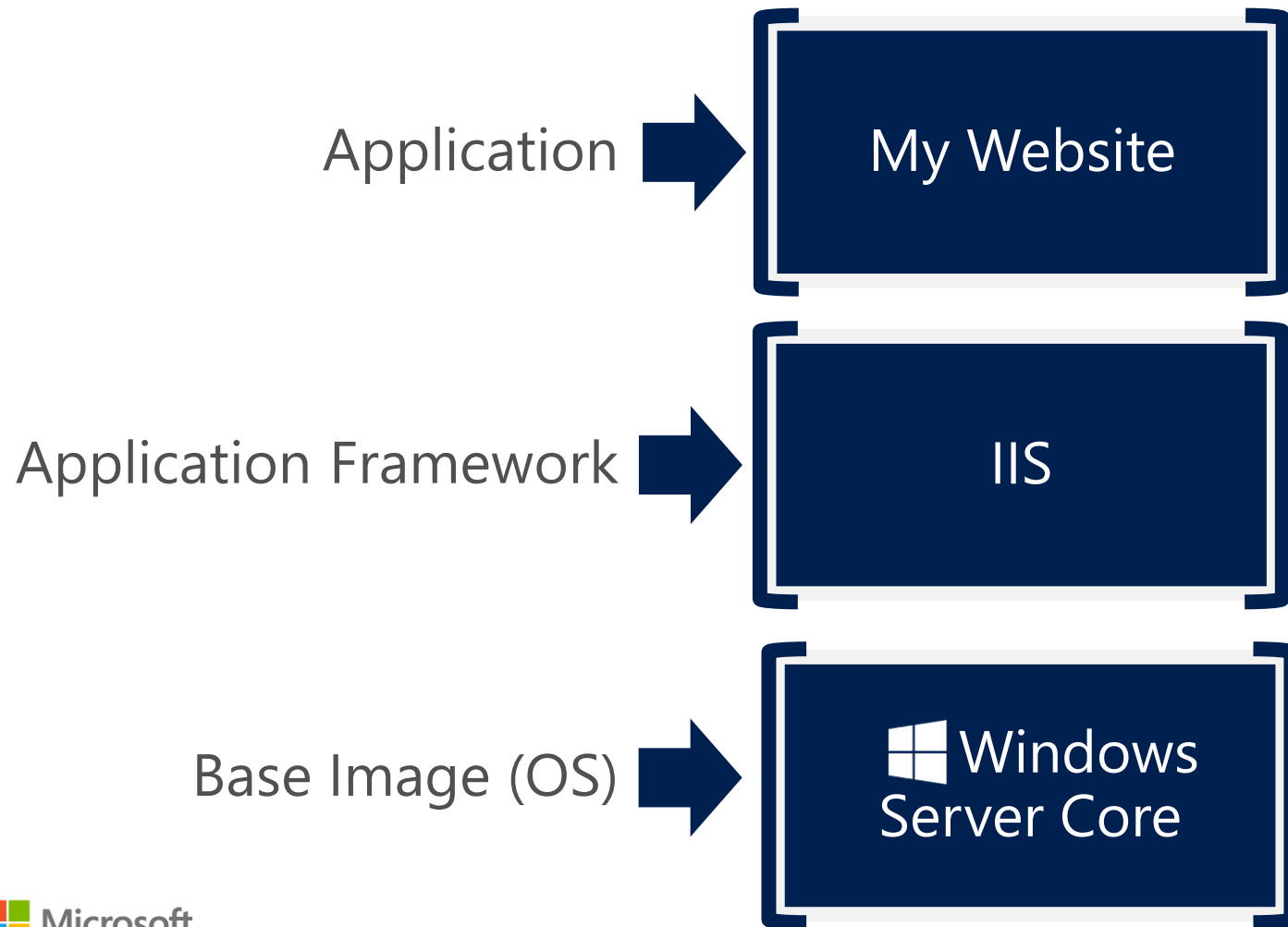


Image Registries

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

Developers and Containers

Build Once, Run Anywhere

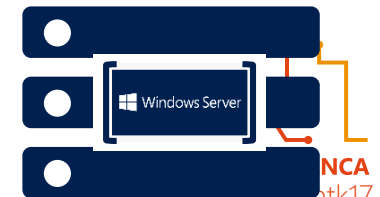
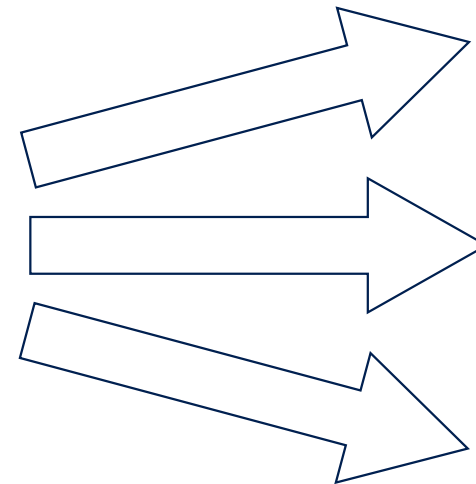
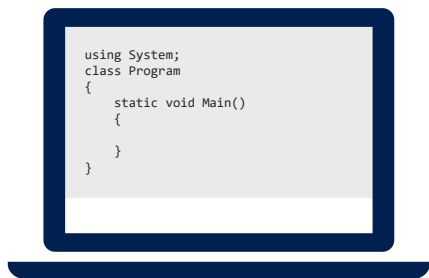
Easy to create, share and distribute

Start immediately, scale easily

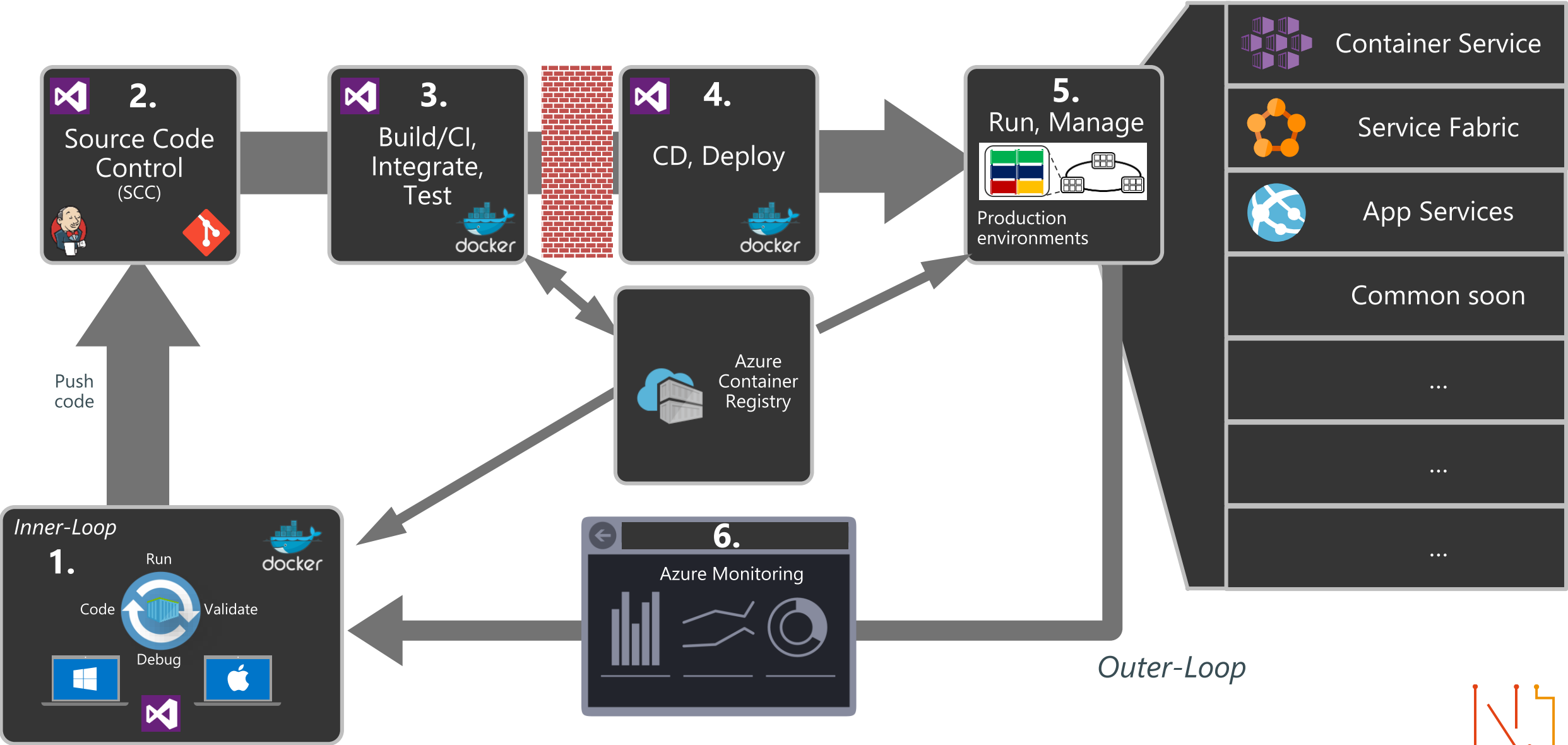
Always consistent - include all prerequisites

Enable microservice architecture

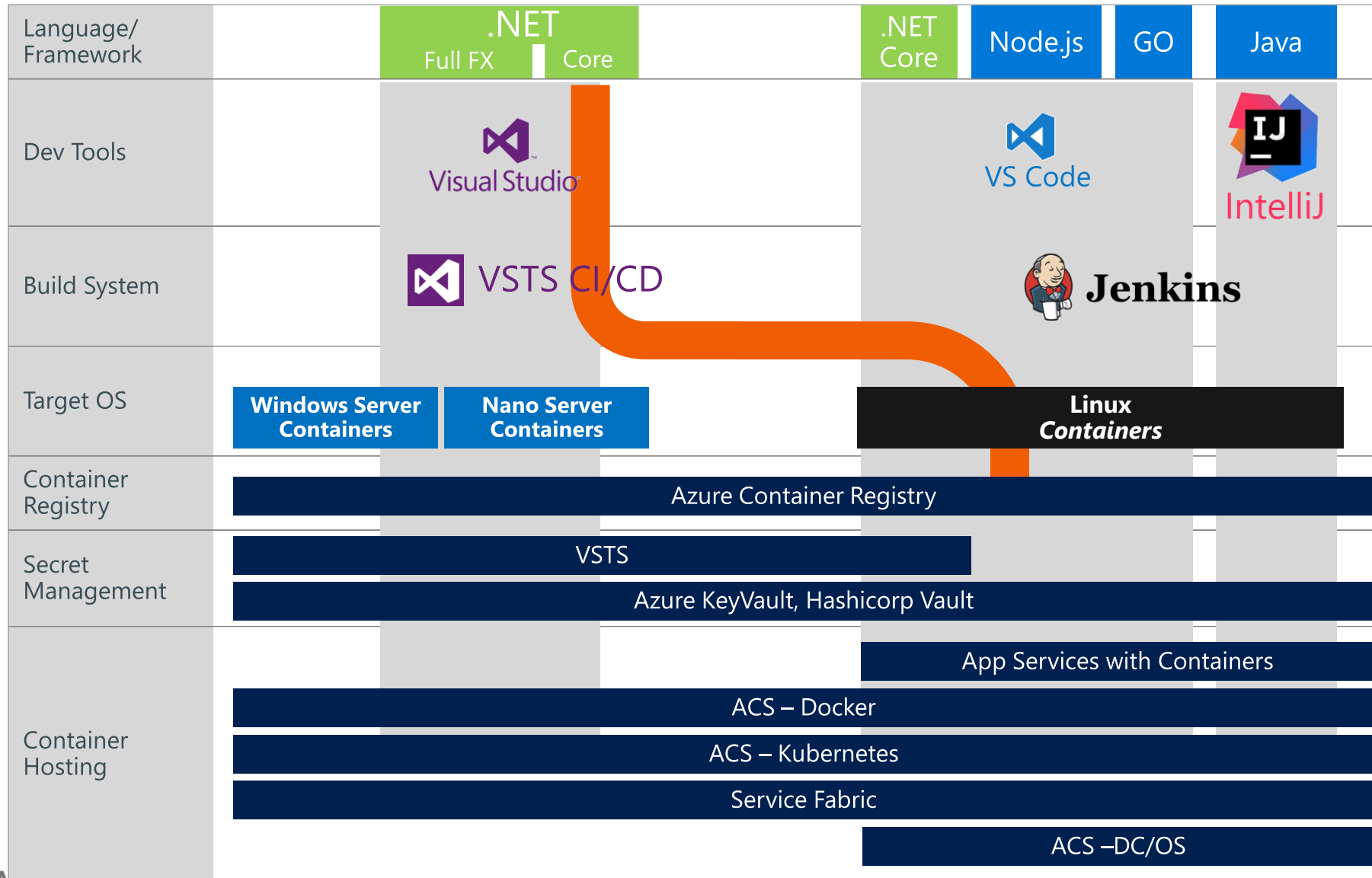
Foster Dev & Ops collaboration



Containerized workflow



Microsoft Container Integrations



KONFERENCA

Reach us with #ntk17

Visual Studio 2017 Container Tools

Run, debug, test web, and console apps in Docker Containers

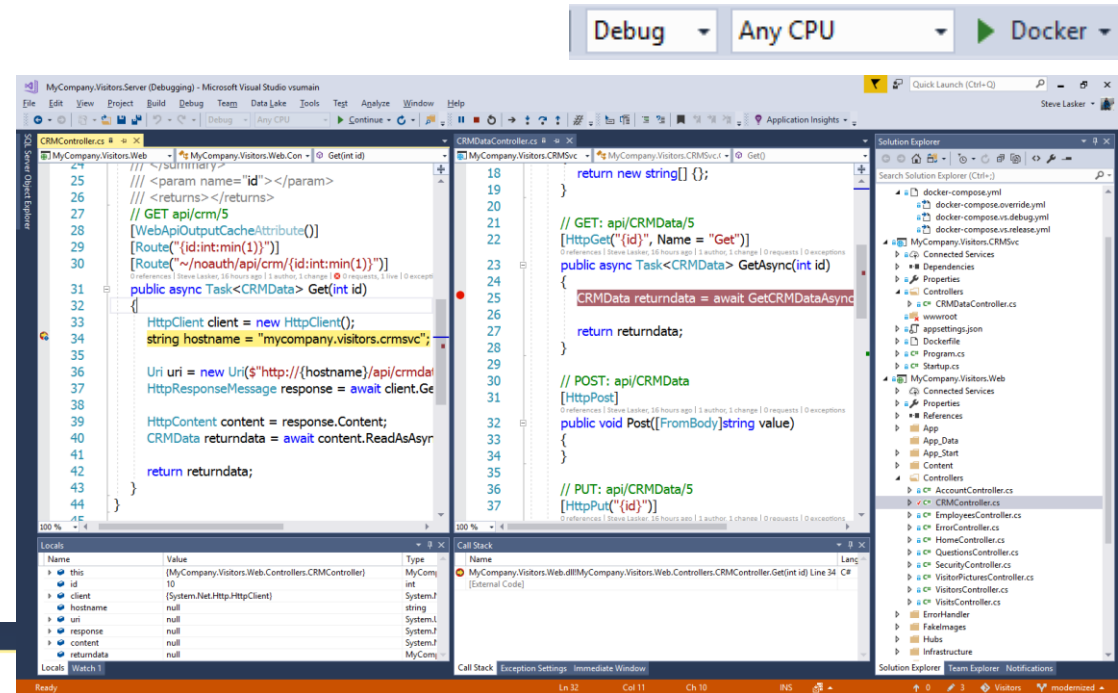
.NET FX w/Windows Server, .NET Core with Nano Server, and Linux

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
```

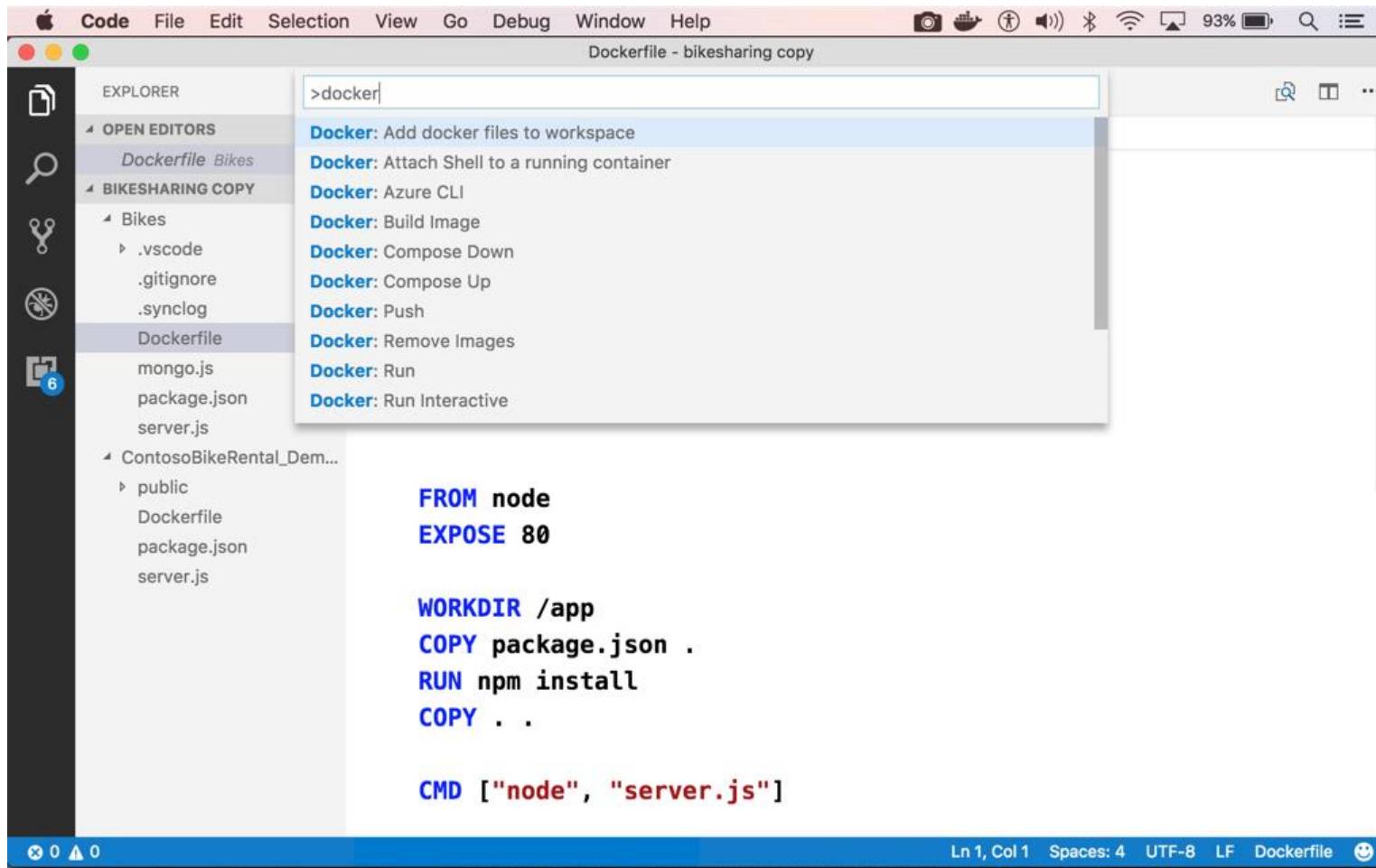
```
Dockerfile*
FROM microsoft/aspnetcore:1.1
ARG source
WORKDIR /app
EXPOSE 80
COPY ${source:-obj/Docker/publish} .
ENTRYPOINT ["dotnet", "Api.dll"]
```

```
docker-compose.yml
version: '3'

services:
  web:
    image: stevelasemos.azurecr.io/web
    build:
      context: ./Web
      dockerfile: Dockerfile
  api:
    image: stevelasemos.azurecr.io/api
    build:
      context: ./Api
      dockerfile: Dockerfile
```

Linux and Windows Containers Containerize Application Demo

Visual Studio code



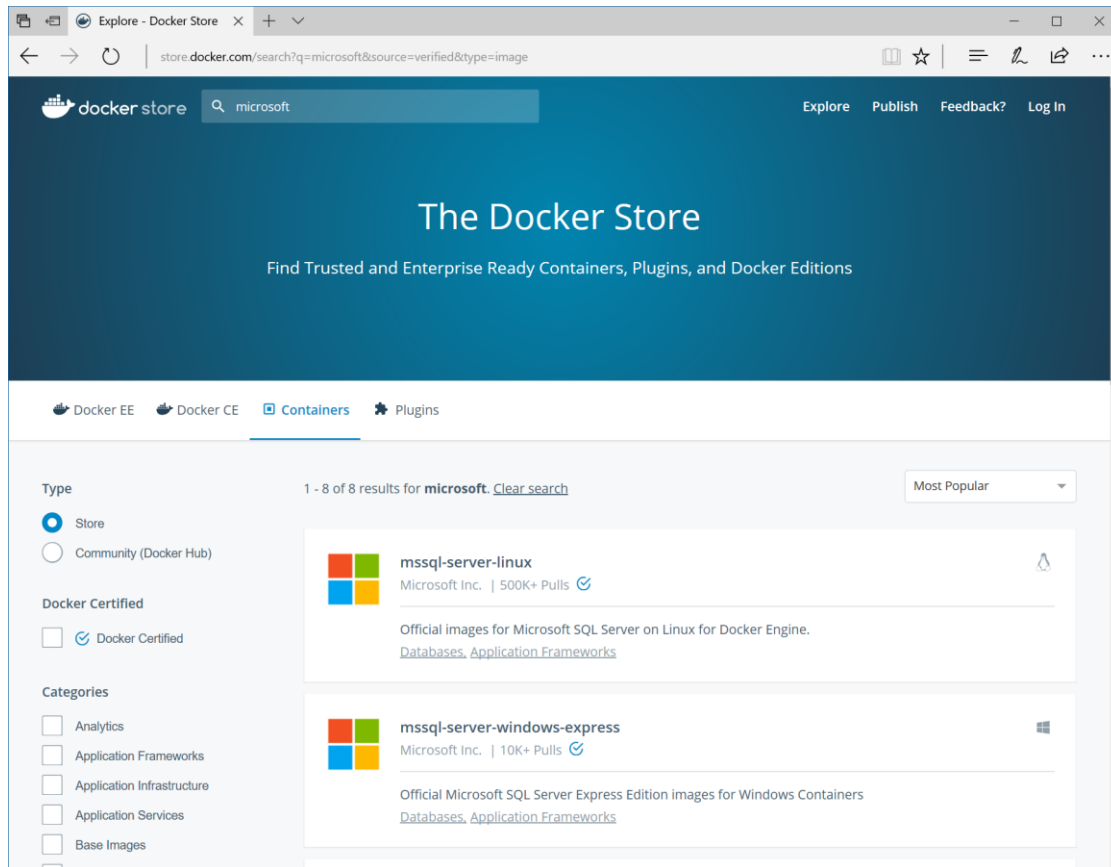
OS X, Linux, Windows
Node, Python, Go,
.NET Core
Git SCC integration
Extensible platform
Docker Asset Editing –
extension

aka.ms/VSCode

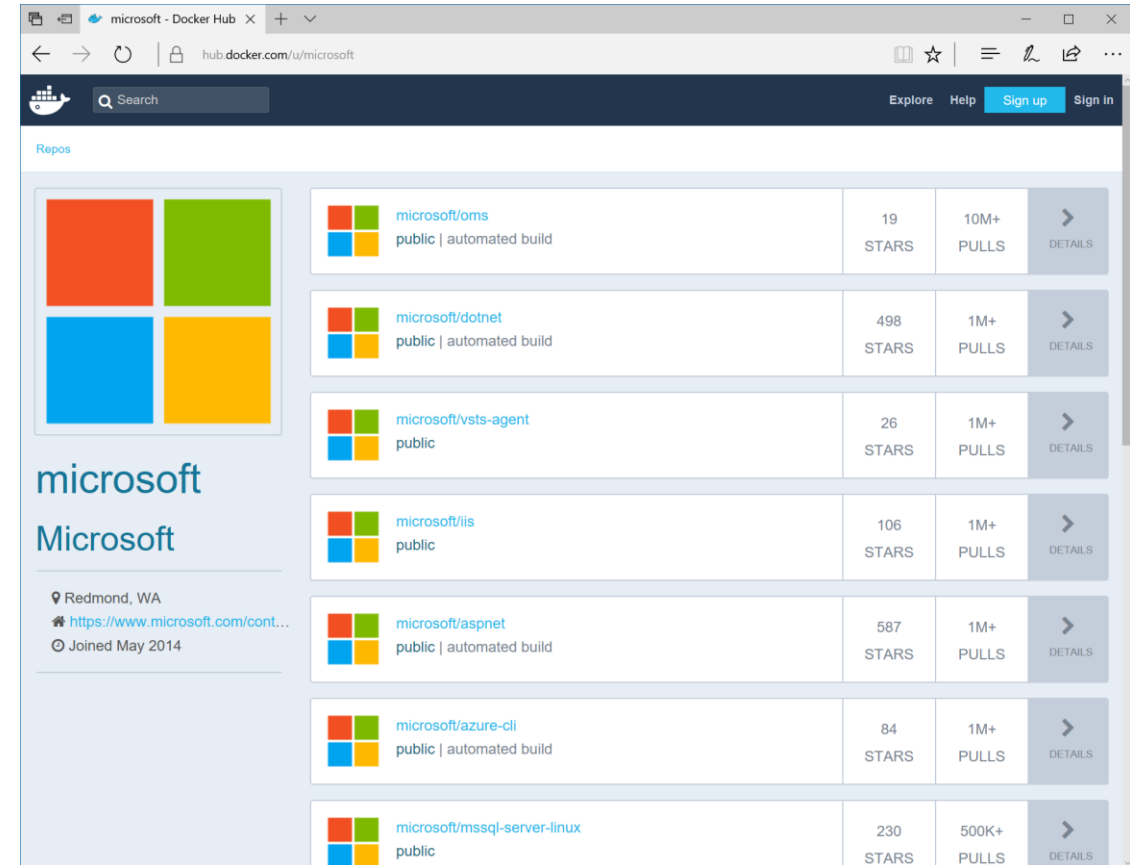
aka.ms/DockerToolsForVSCode

Maintained images by Microsoft

<https://store.docker.com/>



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



Containerize Release Pipeline

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

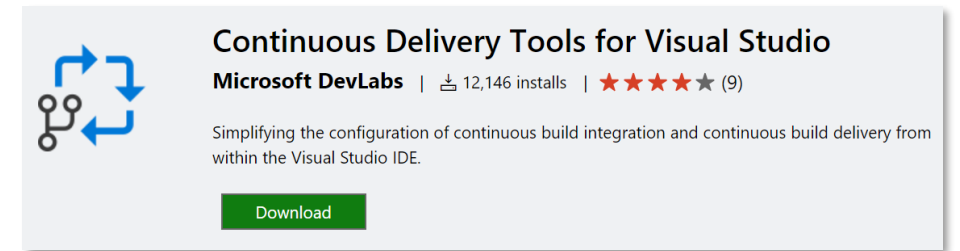
Continuous Delivery Tools for Visual Studio

Create Continuous Delivery pipeline easily

Creates automated build and release management workflows

Delivery in the cloud

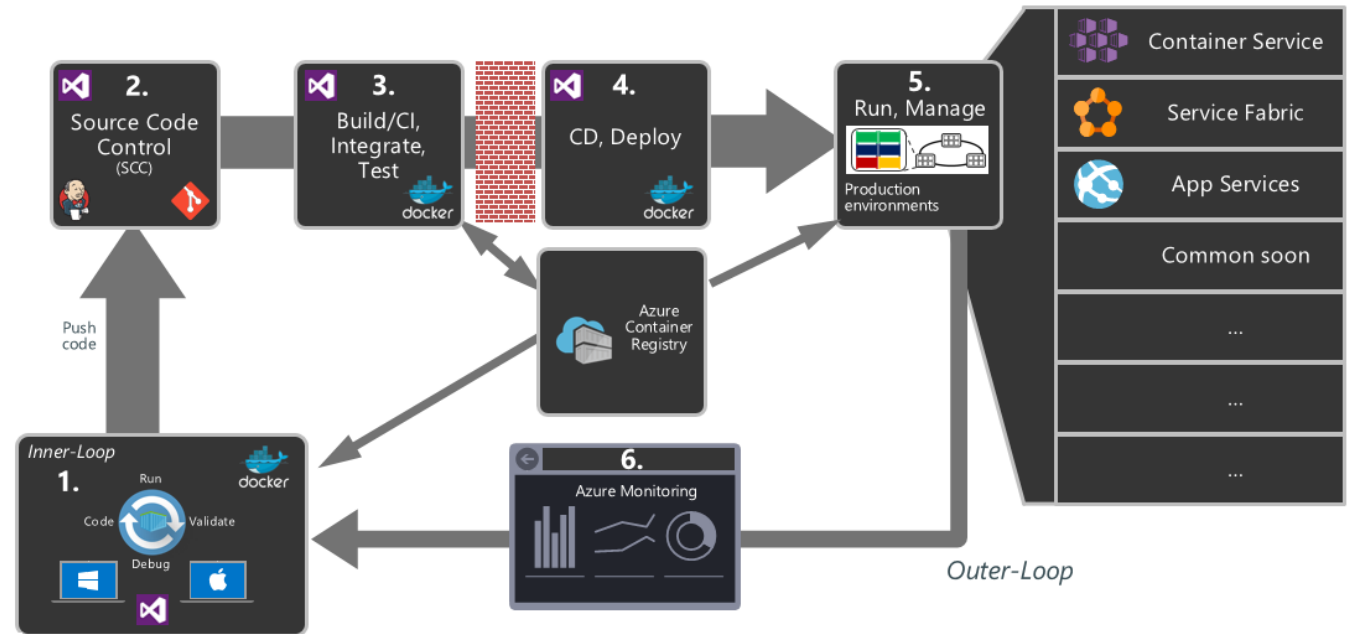
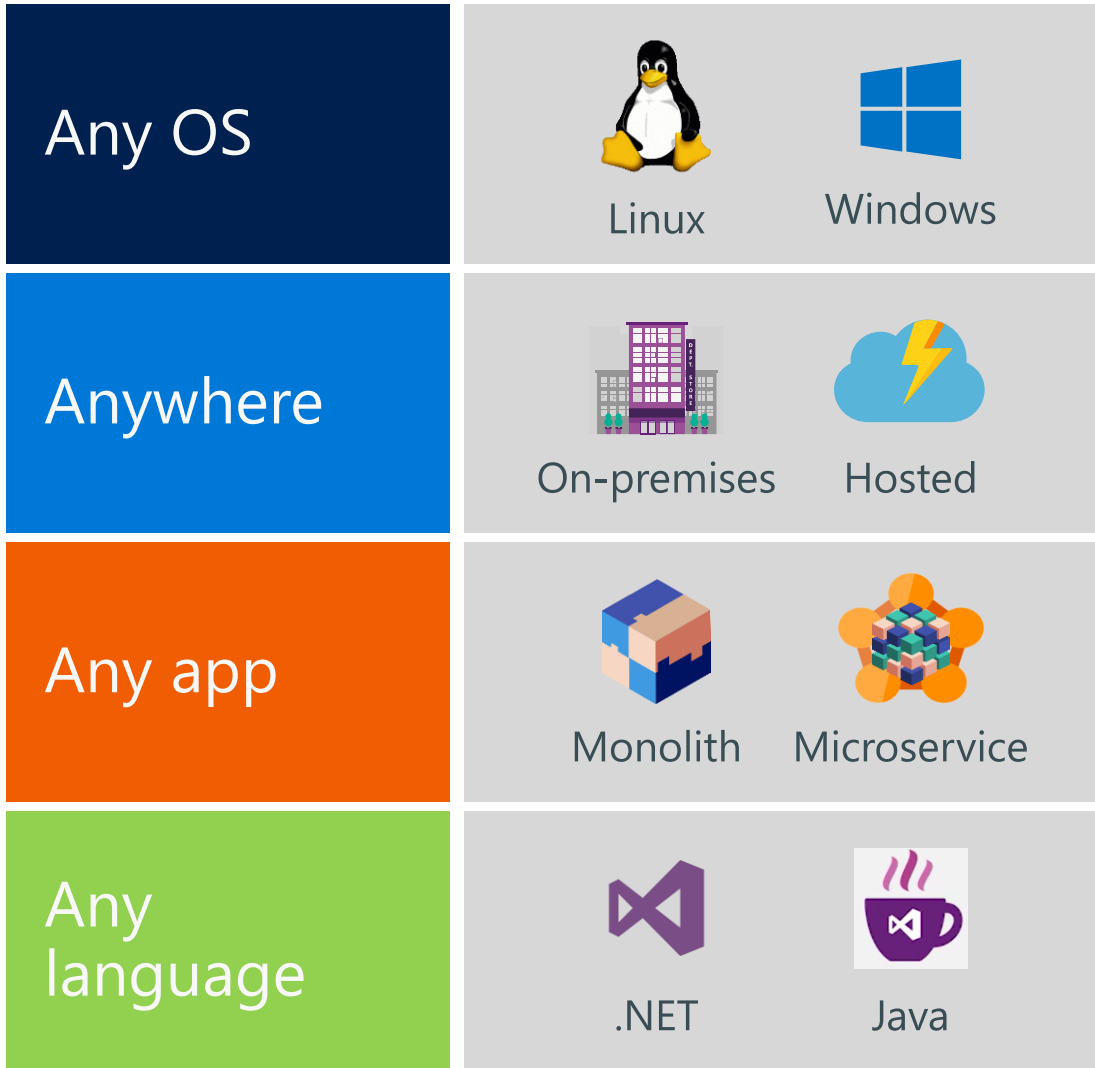
- ASP.Net projects to Azure App Service
- .Net Core to Azure Container Service



<http://aka.ms/CD4VS>

Instant Continuous Delivery To Containers in the Cloud Demo

Release Management and Containers



Thank you!

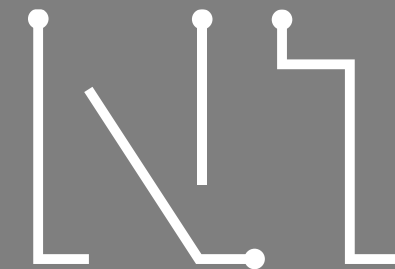
Questions?

Ognjen Bajić

obajic @ ekobit.hr

Ana Roje Ivančić

aroje @ ekobit.hr



KONFERENCA

PORTOROŽ, 15. DO 17. MAJ 2017

© Copyright Microsoft Corporation. All rights reserved.



KONFERENCA

Reach us with #ntk17