Cloud Trends

„Microsoft IT Professional“ Edition

itnetx

itnetx.ch
itnetx.be

Marcel Zehner

@marcelzehner
marcelzehner.ch

Microsoft
Regional Director

MVP Microsoft® Most Valuable Professional

Everything goes CLOUD

Are ITPros still needed?

Git, GitHub, TFS

Automation kills jobs

C#, Ruby, Python, .NET Core

It's all about DEVELOPERS

LINUX & Open Source

Docker

Machine Learning

Shadow IT

Source Control & Versioning

The business is bypassing IT

Far too high release cadence

Are you afraid of the future?

Windows Server

History

- New version released approx. every 2-4 years
  - 2003, 2008, 2008 R2, 2012, 2012 R2, 2016
- Is that still sufficient in today's world?
  - We need new features at cloud speed
  - For cloud/modern features

Windows As A Service

- Today you can choose how to service your Windows OS

Long-time Servicing Channel (LTSC)

Semi-Annual Channel

Long-Time Servicing Channel

- Major version of Windows Server
  - **This is what we had with Windows Server 2012, 2016 etc.**
  - **Current version: “Windows Server 2016”**
- 5 years of mainstream support, 5 years of extended support
- Cumulative updates available monthly (no change)

Semi-Annual Channel

- Delivers new functionality to the OS at cloud cadence
  - **Current version "Windows Server, Version 1709"**
- No more waiting for important functionality updates
- New releases available twice a year (spring and fall)
- Support 18 months from initial release
  - **No upgrade possible → Re-Install!**
- Cumulative updates available monthly (no change)

Make Your Decision

- Use LTSC for legacy app servers
  - System Center, legacy LOB apps
    - Server with GUI
    - Server Core
- Use SAC for modern workloads
  - Hyper-V, container hosts or container images
    - Server Core
    - Nano Server (container images)

	SAC (Windows Server, Version x)	LTSC (Windows Server 2016)
Nano Server (container only)	Yes	No
Server Core	Yes	Yes
Server with Desktop Experience (GUI)	No	Yes

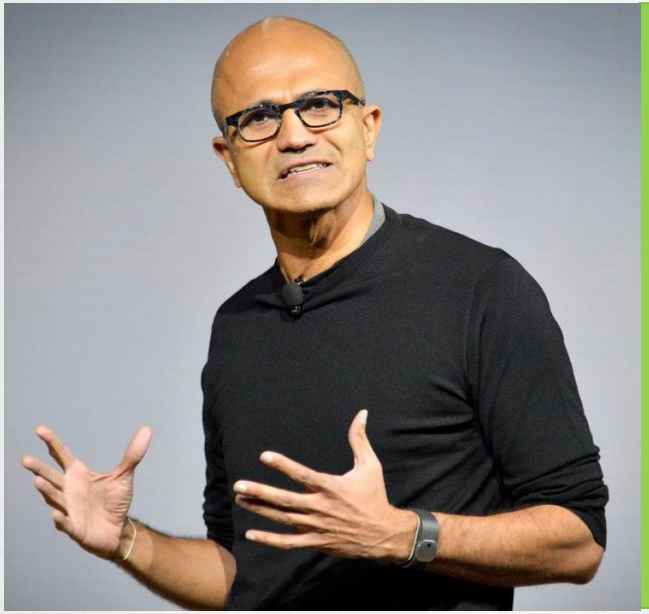
Linux & Open Source

Times Are Changing

"Linux is a cancer that attaches itself in an intellectual property sense to everything it touches"

Steve Ballmer, Microsoft CEO

Times Are Changing



"Microsoft loves Linux"

Satya Nadella, Microsoft CEO

Microsoft & Linux

Linux runs on Windows

Windows Subsystem for Linux, WSL (Bash)

Linux Containers on Windows

Linux Shielded VMs

Etc.

Microsoft Apps run on Linux

.NET Core

PowerShell incl. DSC

SQL Server

Skype client

Visual Studio Code

SCOM agent

OMS agent

Etc.

Microsoft services run on Linux

LinkedIn

Skype

HDInsight

Office 365

Etc.

Azure Features for Linux

Azure Container Service

Azure Virtual Machines

Azure Recovery Services

Azure Management

Azure Log Analytics

Etc.

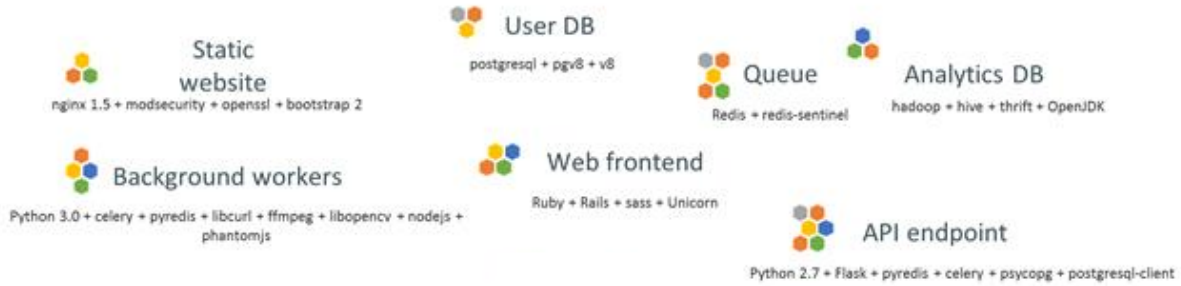
Demo
Microsoft & Linux

Containers

Apps Have Changed

5+ years back	Today+
Long lived	Development is iterative and constant
Monolithic and built on a single stack	Built from loosely coupled components (Microservices)
Deployed to a single server	Deployed to a multitude of servers and cloud platforms

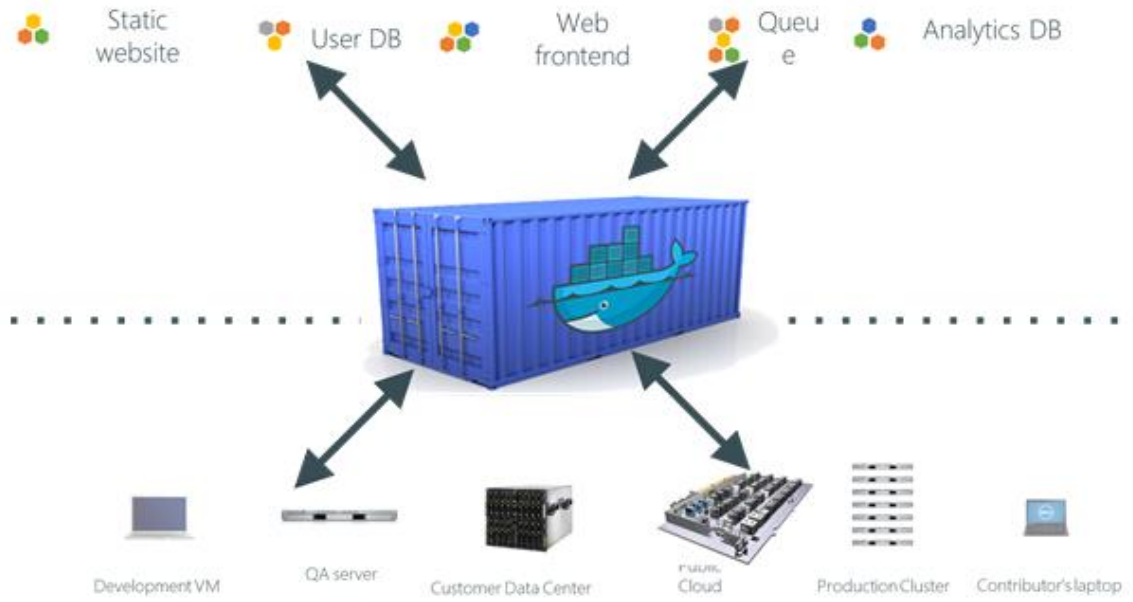
Apps Have Changed



Multiplicity of components and services

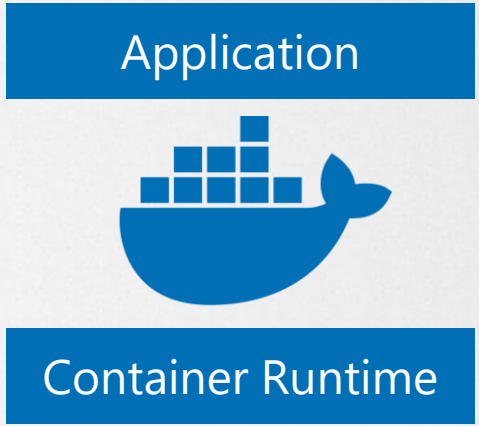
Multiplicity of environments

Containerized Apps



Container to encapsulate any workload and can be executed in virtually any environment

Docker



- Windows containers & Docker
  - Windows containers is a Windows Server 2016 role
  - Docker is a management platform for containers
  - Windows containers are managed with Docker
    - Windows Containers
    - Hyper-V Containers → Isolated

Container Services On Azure

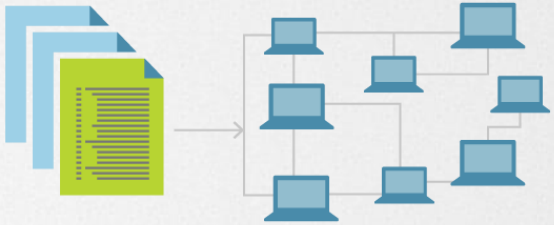
- Azure Container Instance
- Azure Container Services
  - **Docker Swarm, Kubernetes, DC/OS**
- Azure Container Registry

NAME	PUBLISHER
Docker for Azure CE VM	Docker, Inc.
Docker EE for Azure (Basic) - [17.06]	Docker, Inc.
Docker on Ubuntu Server	Canonical + Microsoft
Docker EE for Azure (Standard/Advanced) - [17.06]	Docker, Inc.
Docker EE for Azure (Standard/Advanced) - [17.03]	Docker, Inc.

Demo
Containers

Infrastructure As Code

Infrastructure As Code



- Infrastructure deployment as code
- Fully automate deployment of servers and application components
  - **Declarative approach**
  - **Simple testing, portability, re-deployment etc.**
- Full lifecycle management
  - **Deploy, document and update**

Microsoft Offerings

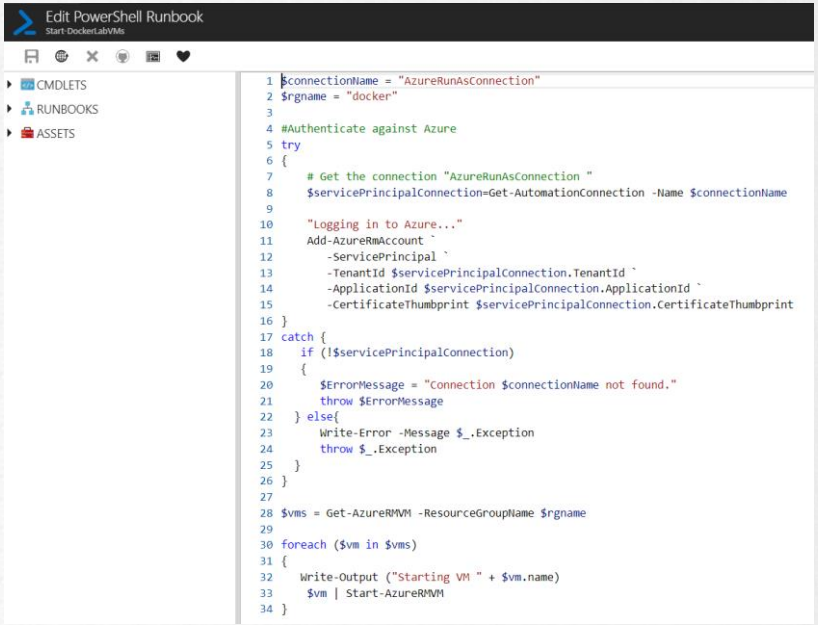
- Azure Resource Manager
  - Describe how Azure resources are deployed
  - Azure and Azure Stack
- Desired State Configuration
  - Describe configuration of servers (Windows, Linux)
- Containers
  - Describe how container images are built
  - Describe how complex container apps are deployed

Demo
Infrastructure As Code

Automation

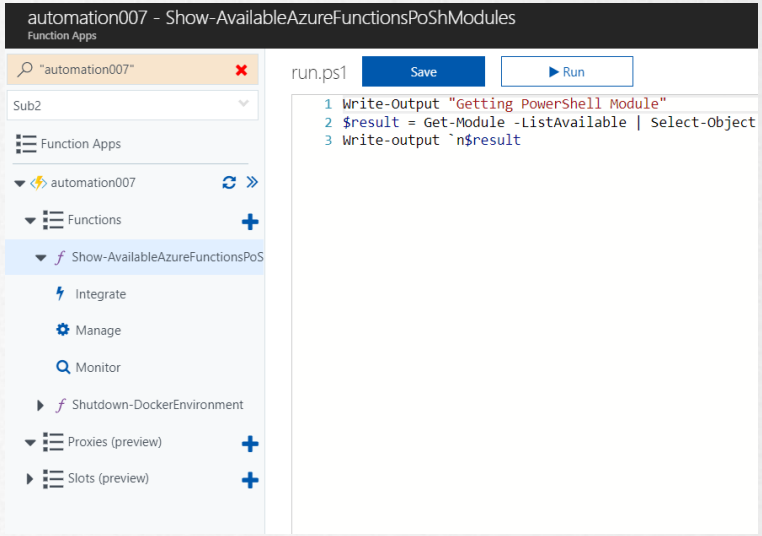
Azure Automation

- Runbooks to orchestrate complex processes
- PowerShell scripts/workflows, graphical or Python
- Central assets store
- Hybrid worker for on-premise execution
  - **Windows & Linux hybrid workers**



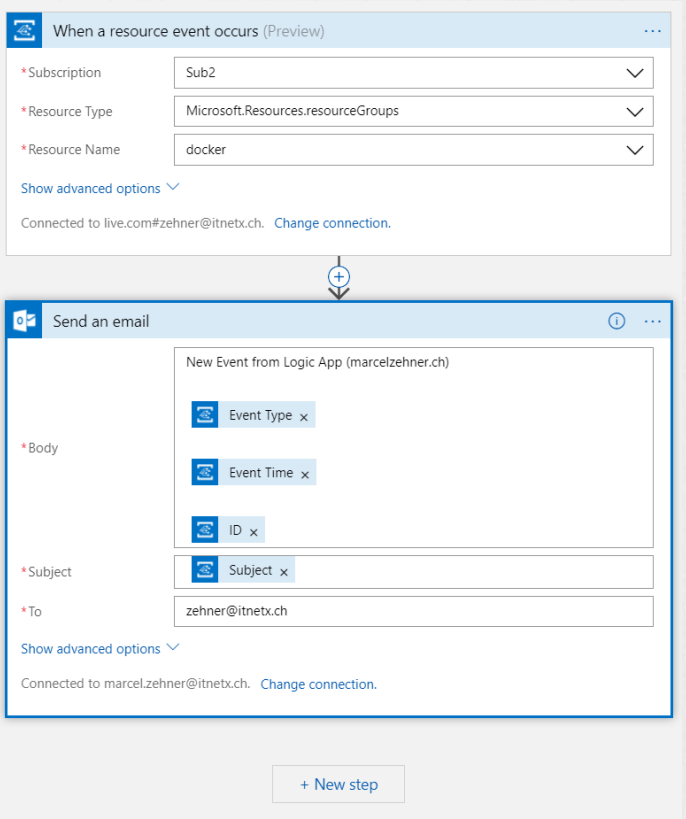
Azure Functions

- Functions as a Service
- Flexible choice of language (PowerShell, C#, Ruby etc.)
- To run small pieces of code in the cloud
- Function runtime is open-source

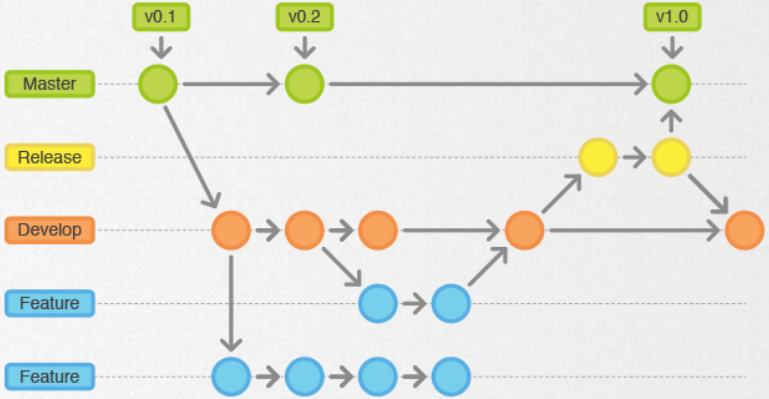


Azure Logic Apps

- Easy workflow generation with triggers and actions
- Connects apps, services and data
  - **Lots of connectors**
- Graphical workflow designer



SCM: Git in 1 Minute



- Open Source SCM
- Files live in repositories
  - Different states
    - Untracked
    - Tracked → Modified, Unmodified, Committed
  - Supports branching
    - Master, Dev etc.
    - Branch merging
      - Always use Dev, merge into Master
- Can run on your desktop (standalone, integrated in tools)

Release Pipeline

- Use a staged deployment process
  - Deploy artifacts automatically to different environments
  - Test deployed components automatically
  - Follows a pre-defined process
- Continuous Integration/Deployment (CI/CD)
  - Microsoft Visual Studio Team Services (VSTS)
  - Others available (non-Microsoft)

Demo
Release Pipelines

Conclusion

Conclusion

- Adapt new technologies and platforms
  - Cloud solutions
    - Azure & Azure Stack
    - Prefer SaaS, Paas and Serverless over IaaS
  - Microsoft and non-Microsoft solutions
- Become agile
  - More releases of your IT Pro artifacts
    - Use SCM, Release Pipelines etc.
  - Use Infrastructure as Code
  - Automate where possible

Conclusion

- No need to fear anything, but prepare for a change!
  - **Adopt developer's patterns**
  - **Be open for new concepts and approaches**
  - **Give up old habits**
- Focus on where you can bring real value to your business and customers – get the rest from the somewhere else!

Cloud Trends

„Microsoft IT Professional“ Edition