Microsoft

#techsummitCH

Cloud Trends for IT Pros – X-Platform Edition

Marcel Zehner
Founder & Corporate Ambassador, itnetX

Nicolas Christener
CEO/CTO, Adfinis SyGroup

itnetX

Marcel
Zehner

@marcelzehner
marcelzehner.ch

Microsoft
Regional Director

MVP Microsoft® Most Valuable Professional

Adfinis SyGroup

Nicolas
Christener

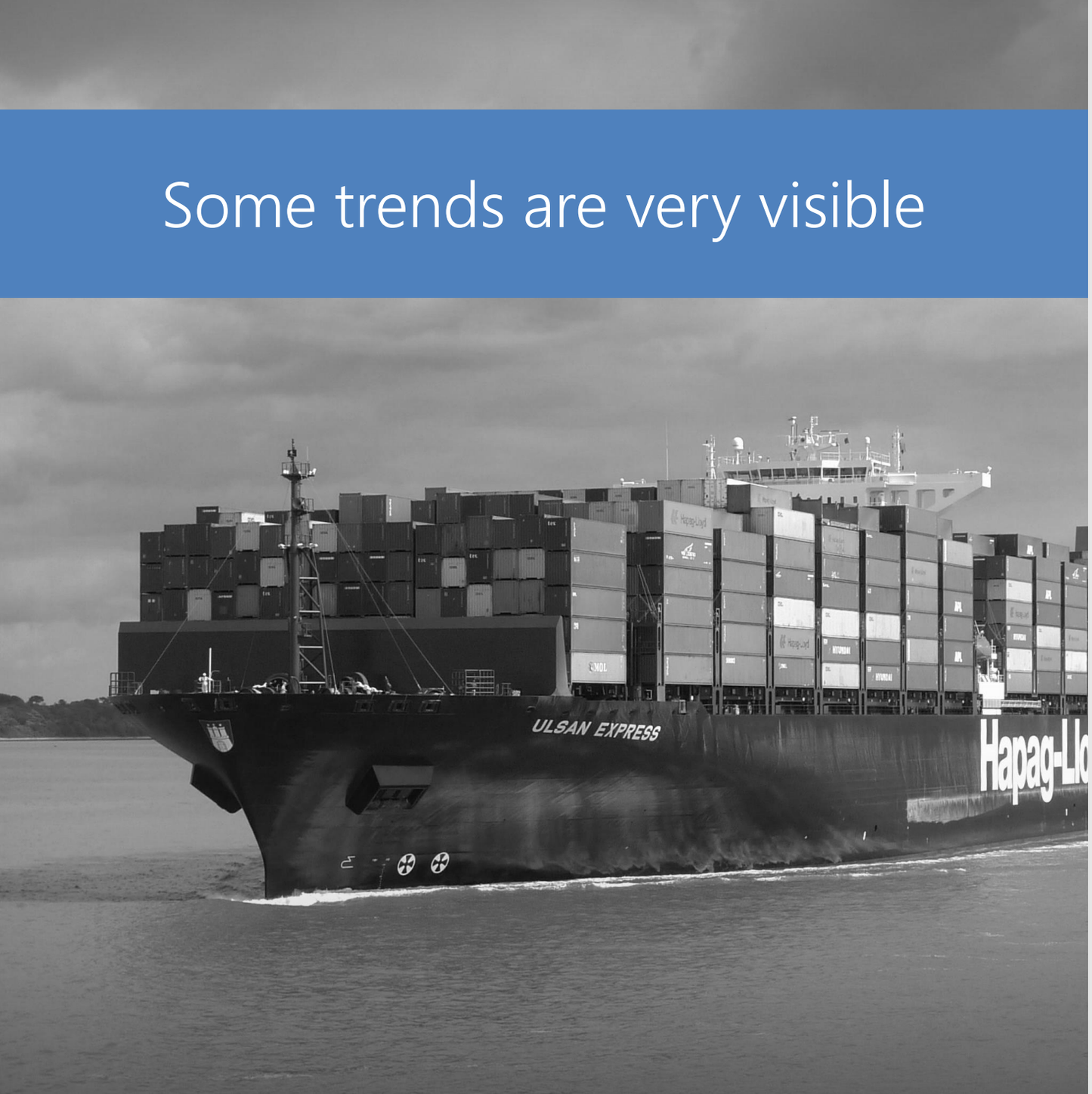
@nikslor
adfinis-sygroup.ch

SUSE Enterprise Linux Certified Engineer

redhat. CERTIFIED ENGINEER

IT is a steady process of innovation

Some trends are very visible

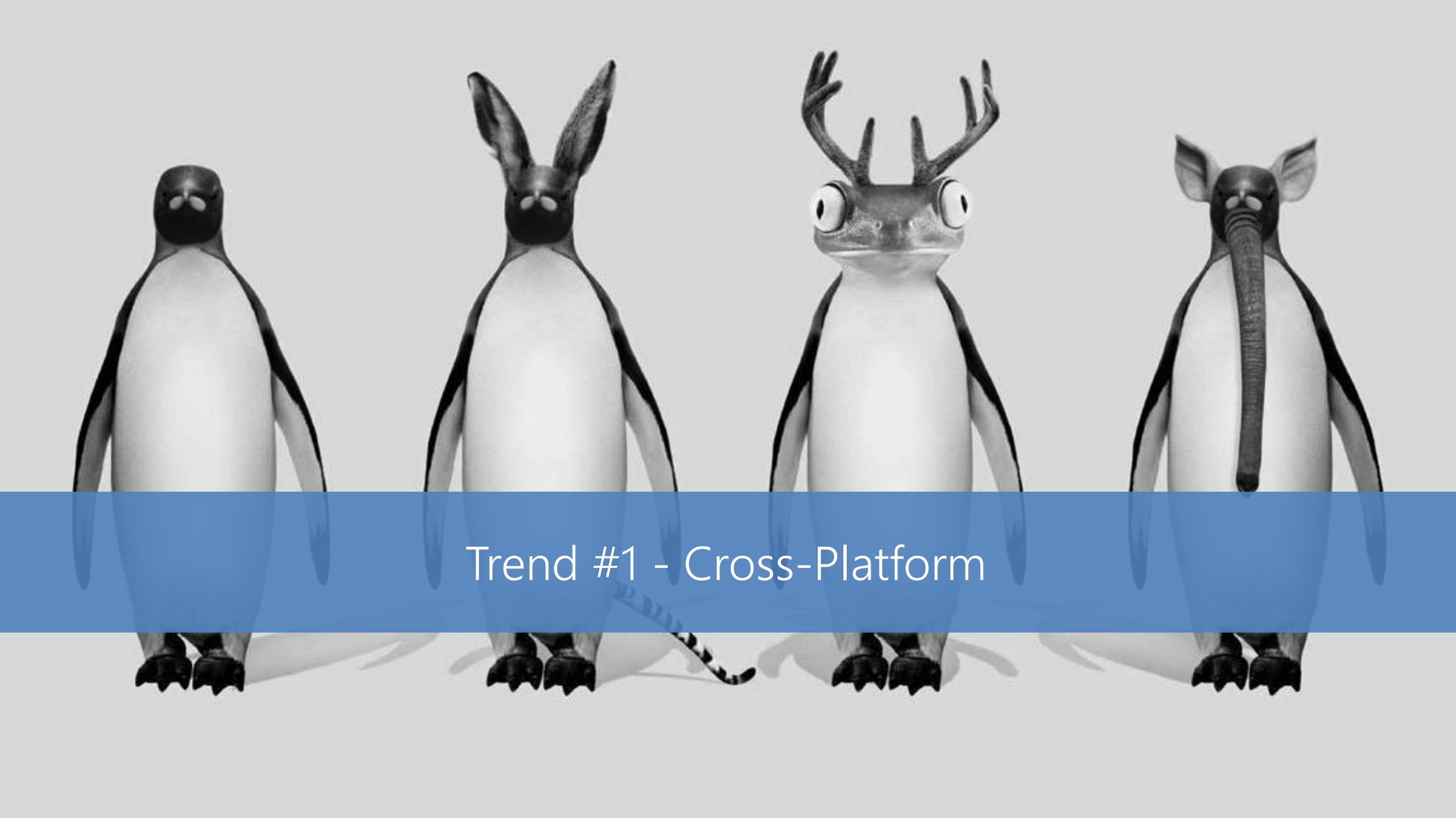


- Shift to container
- Interoperability and APIs are king
- IT services became a commodity
- Automation rules modern IT

Disruption

- Software defined everything - job killer?
- OSS everywhere - do I need to learn Linux?
- Move to the cloud - what about my data?

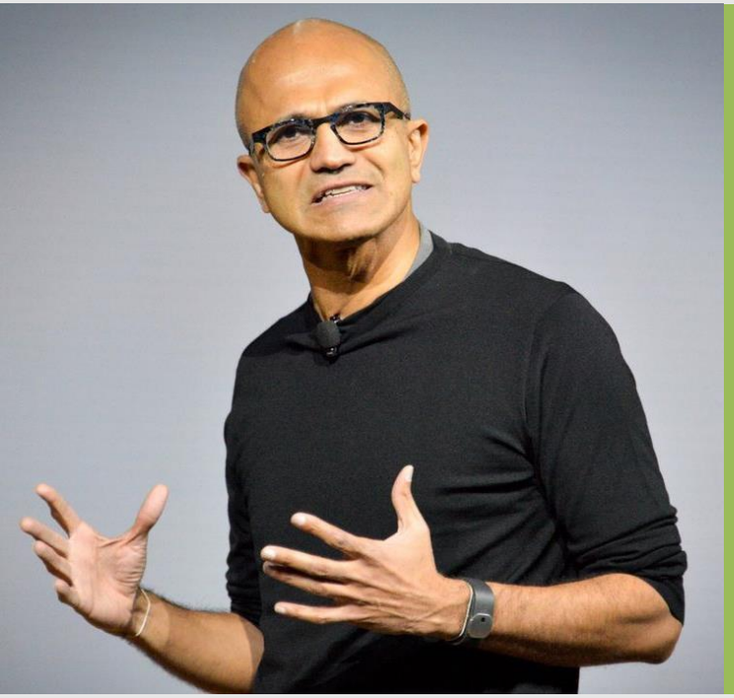
Let's look a bit closer



Trend #1 - Cross-Platform

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

Steve Ballmer, Microsoft CEO (2001)



“Microsoft loves Linux”

Satya Nadella, Microsoft CEO (2014)

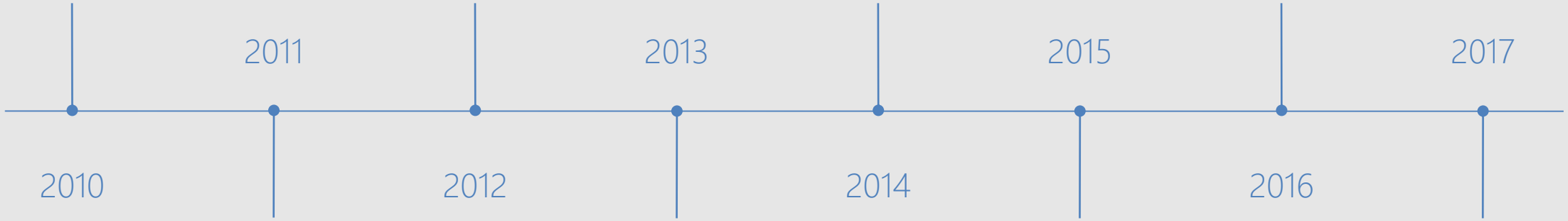
A little bit of history

- PST tools released using Apache License
- OpenStreetMap layer in Bing
- „We love open source“

- Top 20 list of Linux kernel contributors
- ASP.NET MVC, Web API, Razor open sourced
- Open Technologies Inc.

- .NET Core open sourced
- Contributions to OpenJDK
- Microsoft ♥ Linux

- MSSQL on Linux
- Linux Foundation Platinum Member
- PowerShell & Chakra open sourced



- GPL contributions to Samba
- Full Support for PhoneGap on Windows Phone
- Node.JS on Windows

- VM Depot (Linux & FreeBSD VMs for Azure)
- BrowserSwarm open sourced
- Git Support in VS & TFS

- Debian & RHEL on Azure
- OpenSSH on Windows
- Visual Studio Code
- Docker on Azure

- 1380 Repos (no forks) on GitHub
- Deis acquisition
- Windows Subsystem for Linux (WSL)

Microsoft & Linux today

Linux runs on Windows

- Windows Subsystem for Linux (Bash on Windows)
- Linux Containers on Windows (Docker for Windows)
- Linux Shielded VMs

Microsoft apps & tools run on Linux

- PowerShell
- .NET Core
- SQL Server
- Microsoft Monitoring agent (SCOM, OMS)
- Azure CLI

Microsoft services run on Linux

- LinkedIn
- Skype
- HDInsight
- Office 365

Azure Features for Linux & OSS

- Azure Container Service
- Azure Virtual Machines
- Azure Cloud Shell (Bash)
- Marketplace offerings
- Kubernetes as a Service

X-Platform Demo

Marcel Zehner

More about those topics

- SQL Server 2017 on Windows, Linux and Docker containers with deep dive on Linux
  (Level 300), 13:30-14:15, Kino 2
- The open source journey of Microsoft and SUSE: From 2006 till now and beyond
  (Level 200), 14:35-15:20, Kino 2
- Linux meets Azure IoT Hub
  (Level 100), 14:35-15:20, Kino 5

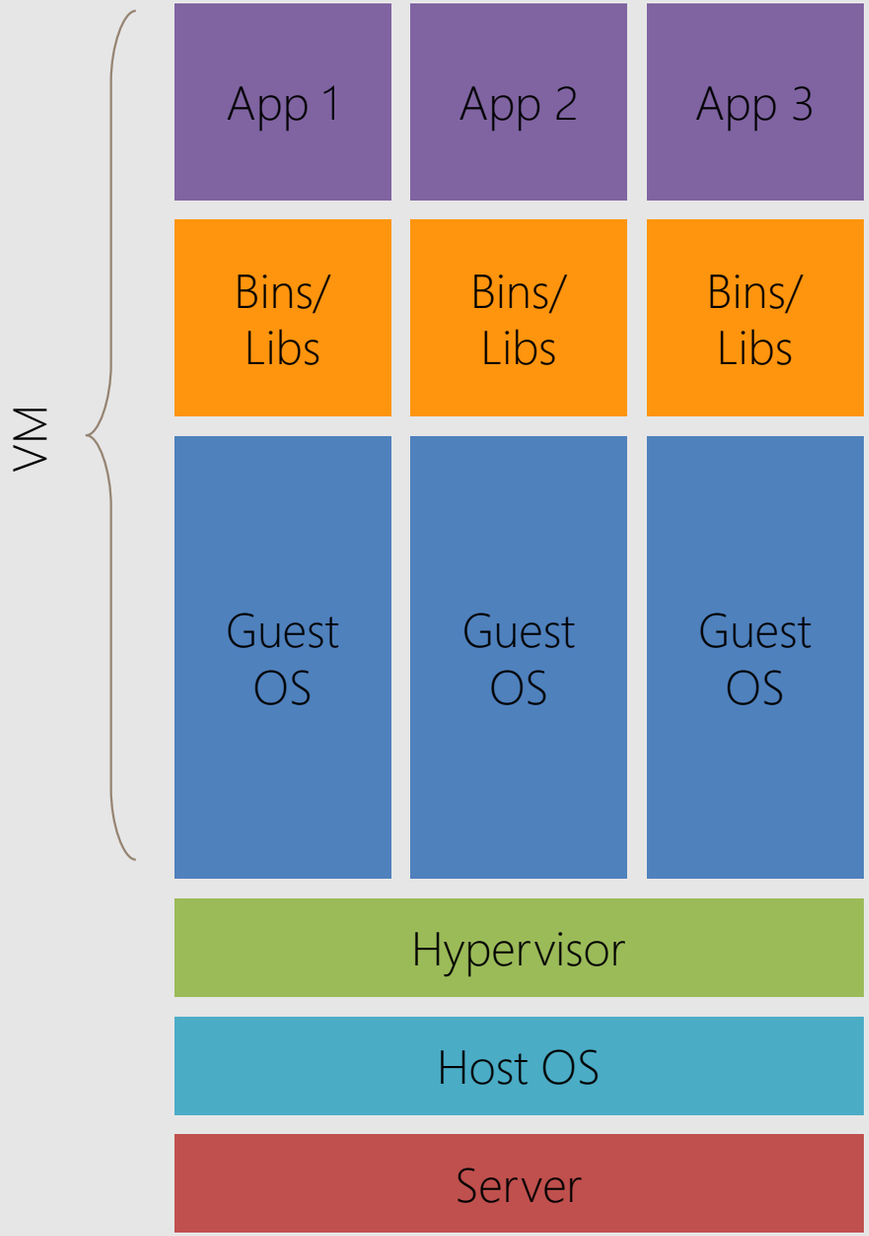
Trend #2 - Container

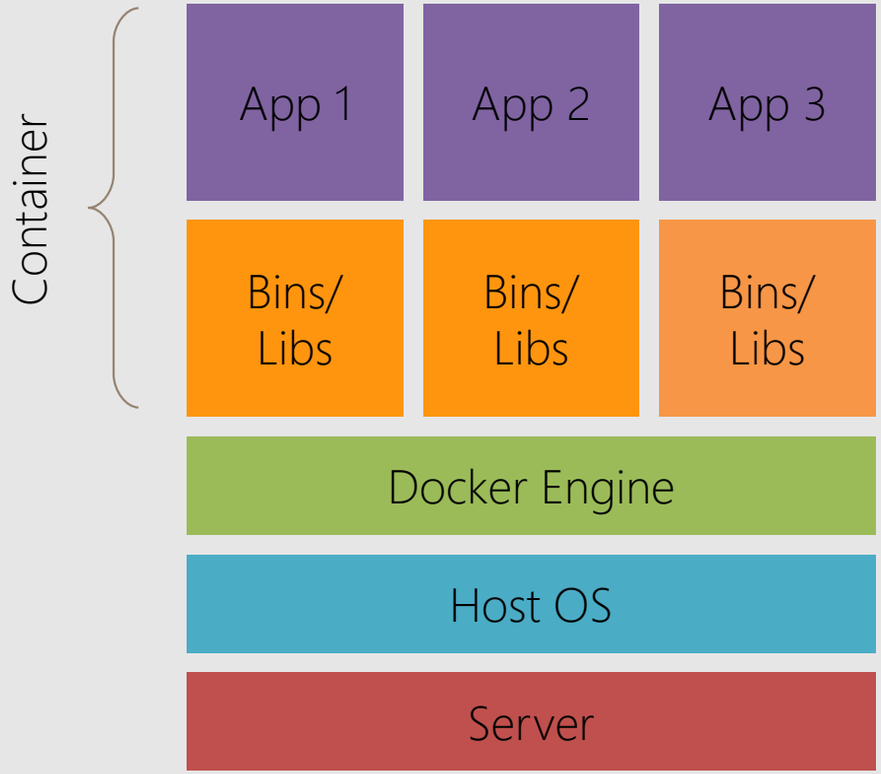
What about the container fuss?

- Container are here to stay
- Application deployment becomes much easier
- The magic lies in the encapsulation and bundling of the ingredients of an app (libs, dependencies, runtimes, etc.)
  - No “dependency hell“ and no version conflicts
  - Developer can ship what's needed
  - Update = replace container with new version
  - Rollback = replace faulty container with working version

Container introduction

- Only process encapsulation, no virtualization
- Old concept, Docker made it popular by making it easy to use
- Container apps are not stateful!
- Not all problems can (should!) be solved with containers

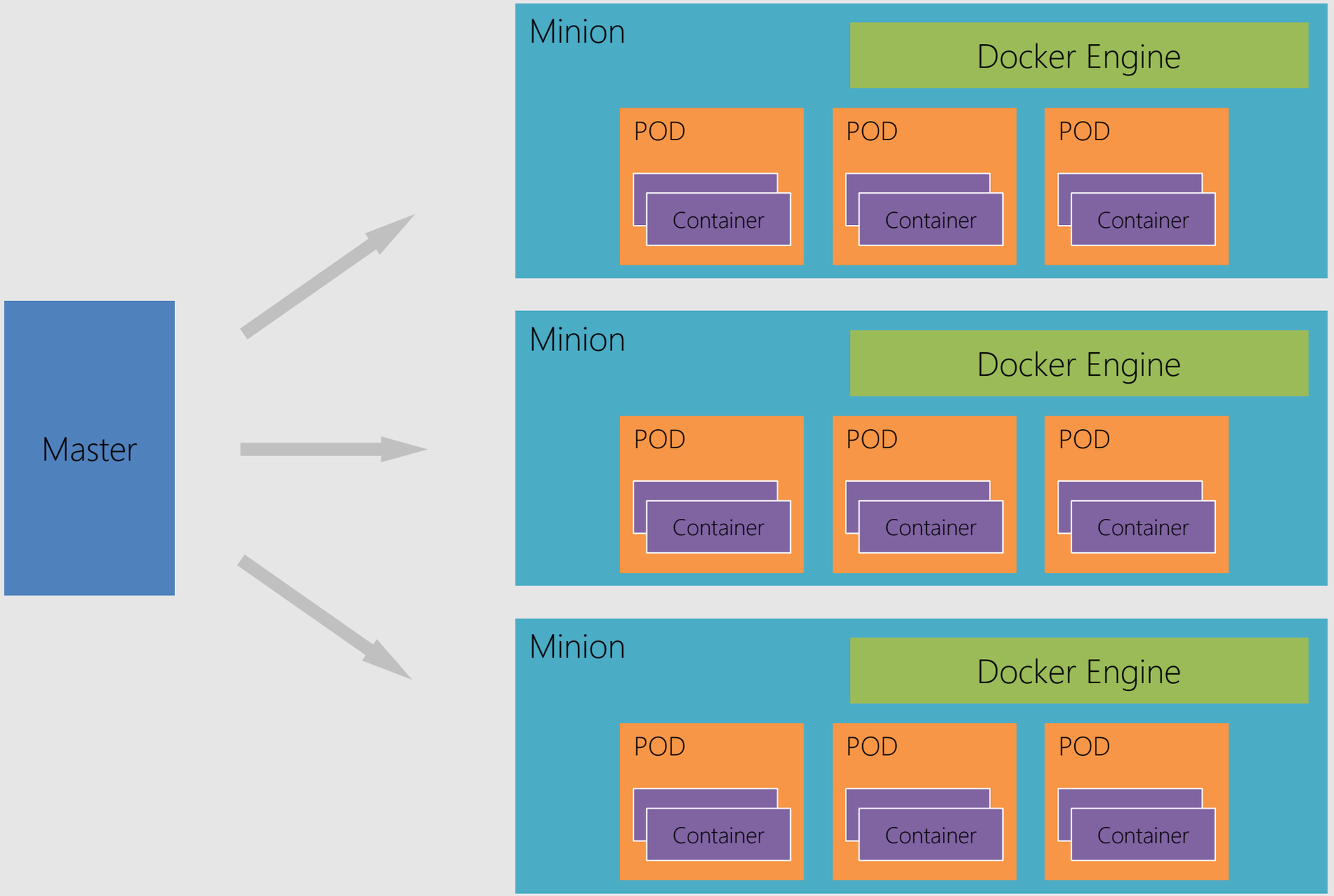

VM
App 1
App 2
App 3
Bins/
Libs
Bins/
Libs
Bins/
Libs
Guest
OS
Guest
OS
Guest
OS
Hypervisor
Host OS
Server


Container
App 1
App 2
App 3
Bins/
Libs
Bins/
Libs
Bins/
Libs
Docker Engine
Host OS
Server

Kubernetes introduction

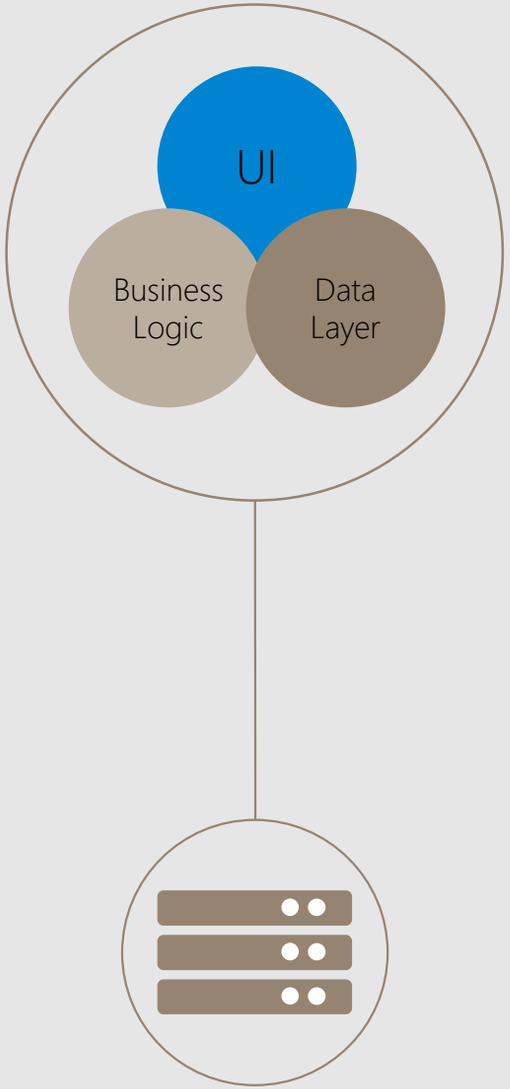
- Also called k8s (k-*ubernete*-s , “ubernetes” = 8 characters)
- Managing containers made easy
- Developed by Google, now a CNCF project
- Kubernetes is the cool kid and has won (against Swarm, Mesos, etc.)

kubernetes

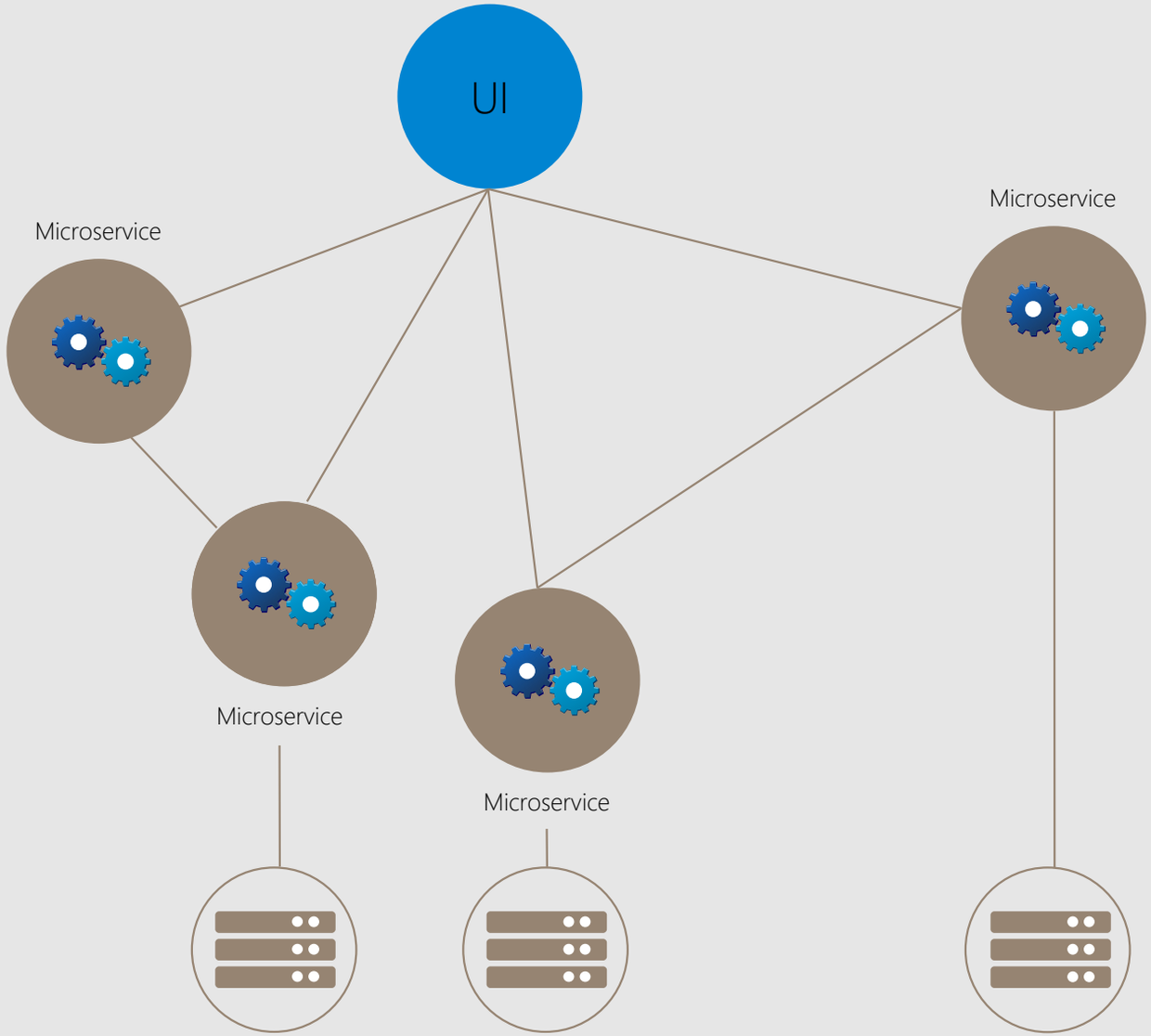

Master
Minion
Docker Engine
POD
Container
POD
Container
POD
Container
Minion
Docker Engine
POD
Container
POD
Container
POD
Container
Minion
Docker Engine
POD
Container
POD
Container
POD
Container

Microservice introduction

- Divide a large application into small parts
- Each part is easy to replace, debug, extend
- Every piece does its part and runs in its container
- The sum of all units reflects a complex solution
- Loosely coupled, communication through message queue, REST API, etc.



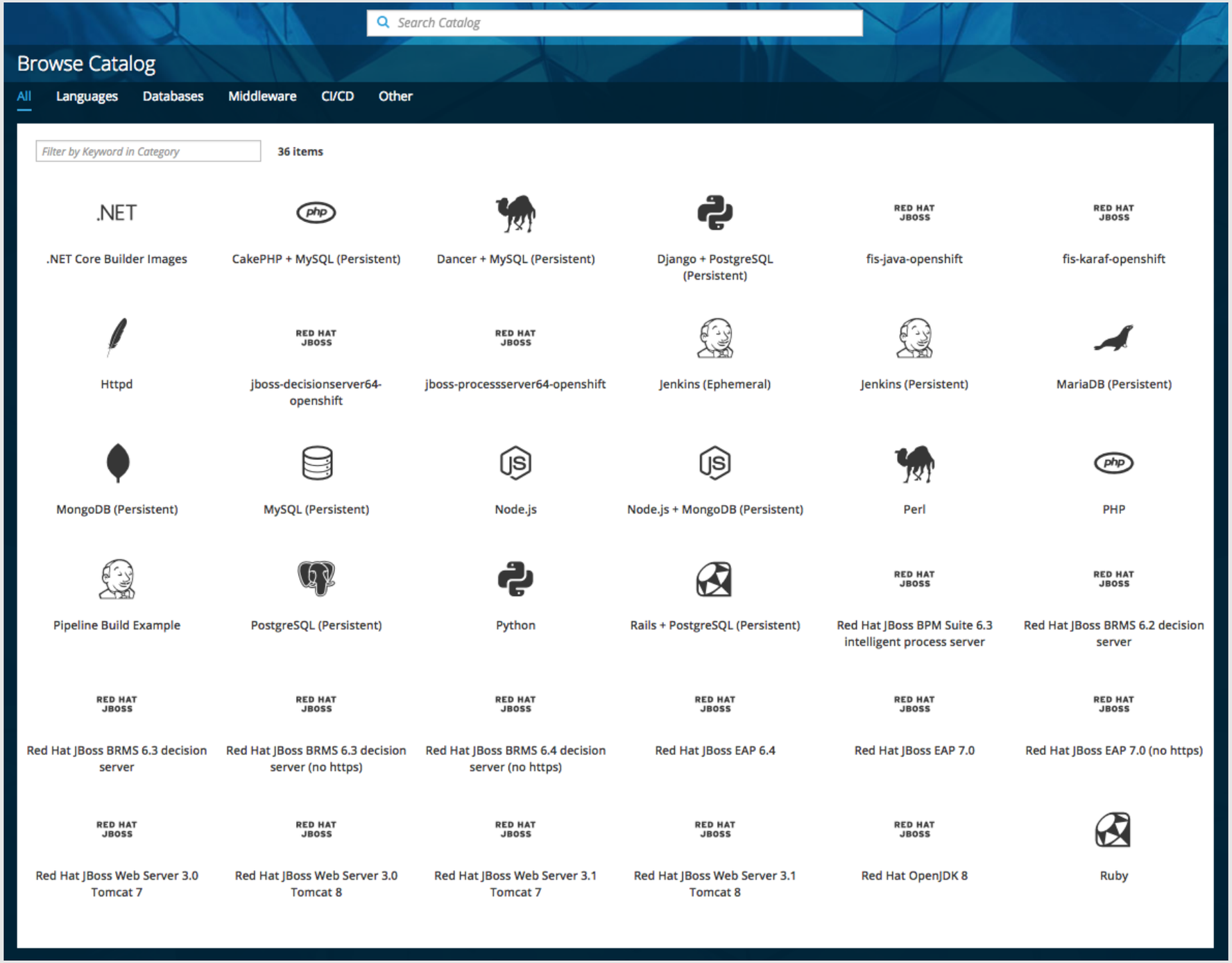
Monolithic Architecture



Microservice Architecture

PaaS introduction

- Everything IT provides can be seen as service
- PaaS abstracts services (network, storage, DB, load balancer, etc.)
- PaaS offers a catalog of services to the user
- Software layer often implemented using containers
- Lower layers through Open Service Broker API


Search Catalog
Browse Catalog
All
Languages
Databases
Middleware
CI/CD
Other
Filter by Keyword in Category
36 items
.NET Core Builder Images
CakePHP + MySQL (Persistent)
Dancer + MySQL (Persistent)
Django + PostgreSQL (Persistent)
fis-java-openshift
fis-karaf-openshift
Httpd
jboss-decisionserver64-openshift
jboss-processserver64-openshift
Jenkins (Ephemeral)
Jenkins (Persistent)
MariaDB (Persistent)
MongoDB (Persistent)
MySQL (Persistent)
Node.js
Node.js + MongoDB (Persistent)
Perl
PHP
Pipeline Build Example
PostgreSQL (Persistent)
Python
Rails + PostgreSQL (Persistent)
Red Hat JBoss BPM Suite 6.3 intelligent process server
Red Hat JBoss BRMS 6.2 decision server
Red Hat JBoss BRMS 6.3 decision server
Red Hat JBoss BRMS 6.3 decision server (no https)
Red Hat JBoss BRMS 6.4 decision server (no https)
Red Hat JBoss EAP 6.4
Red Hat JBoss EAP 7.0
Red Hat JBoss EAP 7.0 (no https)
Red Hat JBoss Web Server 3.0 Tomcat 7
Red Hat JBoss Web Server 3.0 Tomcat 8
Red Hat JBoss Web Server 3.1 Tomcat 7
Red Hat JBoss Web Server 3.1 Tomcat 8
Red Hat OpenJDK 8
Ruby

OpenShift Demo

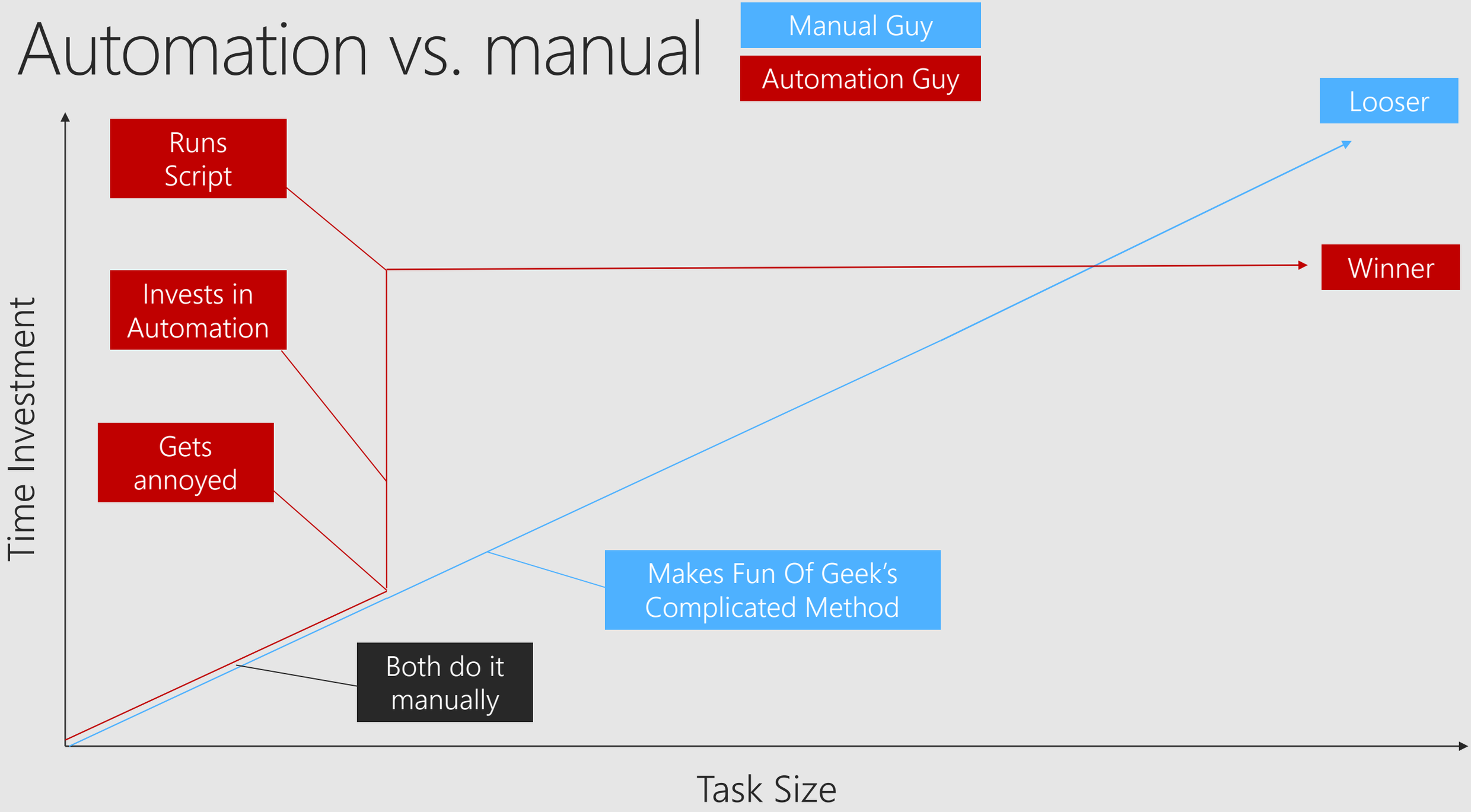
Nicolas Christener

More about those topics

- Drive innovation with containers
  (Level 100), 10:20-11:05, Kino 5
- Power your move to the Cloud with Docker
  (Level 200), 11:25-12:10, Kino 5

Trend #3 - Automation

Automation vs. manual



Automate infrastructure provisioning

- It's getting more complex …
- Infrastructure as Code (IaC)
  - Declarative approach
  - Full lifecycle management
- Azure infrastructure > ARM templates
- Servers > Desired state configuration (DSC)
- Containers & container orchestrators
- Use SCM & CI/CD release pipelines

Automate IT processes

- Microsoft offers …
  - PowerShell
  - Azure Functions
  - Azure Automation
  - Azure LogicApps
- For Windows & Linux

More about those topics

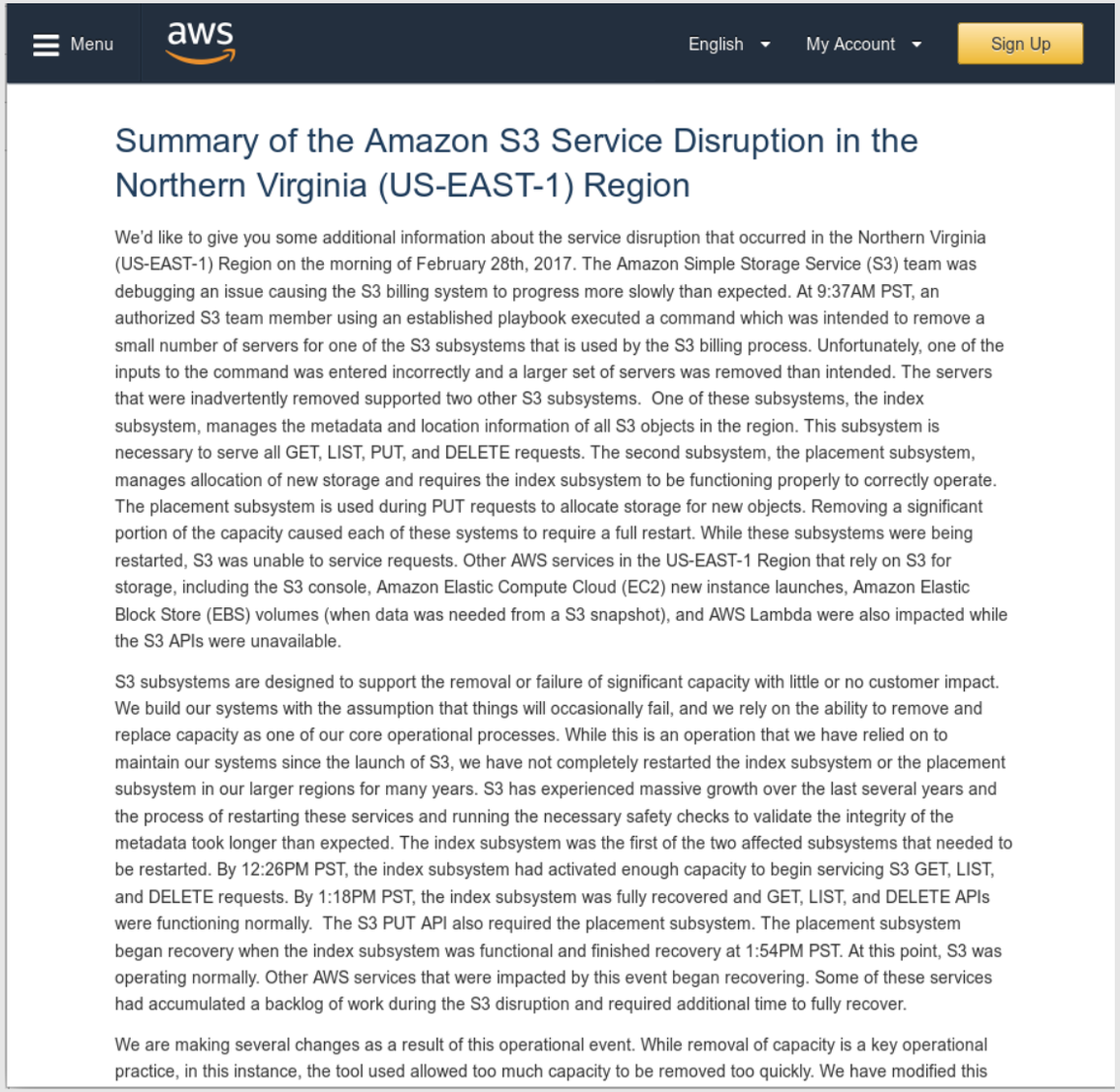
- Mastering Azure Resources
  (Level 300), 15:40-16:25, Kino 1
- Accelerating your journey to the Hybrid Cloud with Ansible Tower
  (Level 200), 15:40-16:25, Kino 5
- DevOps for any Language
  (Level 200), 14:35-15:20, Halle 37

Trend #4 - Hybrid and multi cloud strategies

Cloud is fun!

- Managing your own infrastructure is tedious
  - Network is complicated, storage is hard, patching is cumbersome
- But…
  - What about sensitive data?
  - What about redundancy?

Multi-cloud?



Menu aws English My Account Sign Up

Summary of the Amazon S3 Service Disruption in the Northern Virginia (US-EAST-1) Region

We'd like to give you some additional information about the service disruption that occurred in the Northern Virginia (US-EAST-1) Region on the morning of February 28th, 2017. The Amazon Simple Storage Service (S3) team was debugging an issue causing the S3 billing system to progress more slowly than expected. At 9:37AM PST, an authorized S3 team member using an established playbook executed a command which was intended to remove a small number of servers for one of the S3 subsystems that is used by the S3 billing process. Unfortunately, one of the inputs to the command was entered incorrectly and a larger set of servers was removed than intended. The servers that were inadvertently removed supported two other S3 subsystems. One of these subsystems, the index subsystem, manages the metadata and location information of all S3 objects in the region. This subsystem is necessary to serve all GET, LIST, PUT, and DELETE requests. The second subsystem, the placement subsystem, manages allocation of new storage and requires the index subsystem to be functioning properly to correctly operate. The placement subsystem is used during PUT requests to allocate storage for new objects. Removing a significant portion of the capacity caused each of these systems to require a full restart. While these subsystems were being restarted, S3 was unable to service requests. Other AWS services in the US-EAST-1 Region that rely on S3 for storage, including the S3 console, Amazon Elastic Compute Cloud (EC2) new instance launches, Amazon Elastic Block Store (EBS) volumes (when data was needed from a S3 snapshot), and AWS Lambda were also impacted while the S3 APIs were unavailable.

S3 subsystems are designed to support the removal or failure of significant capacity with little or no customer impact. We build our systems with the assumption that things will occasionally fail, and we rely on the ability to remove and replace capacity as one of our core operational processes. While this is an operation that we have relied on to maintain our systems since the launch of S3, we have not completely restarted the index subsystem or the placement subsystem in our larger regions for many years. S3 has experienced massive growth over the last several years and the process of restarting these services and running the necessary safety checks to validate the integrity of the metadata took longer than expected. The index subsystem was the first of the two affected subsystems that needed to be restarted. By 12:26PM PST, the index subsystem had activated enough capacity to begin servicing S3 GET, LIST, and DELETE requests. By 1:18PM PST, the index subsystem was fully recovered and GET, LIST, and DELETE APIs were functioning normally. The S3 PUT API also required the placement subsystem. The placement subsystem began recovery when the index subsystem was functional and finished recovery at 1:54PM PST. At this point, S3 was operating normally. Other AWS services that were impacted by this event began recovering. Some of these services had accumulated a backlog of work during the S3 disruption and required additional time to fully recover.

We are making several changes as a result of this operational event. While removal of capacity is a key operational practice, in this instance, the tool used allowed too much capacity to be removed too quickly. We have modified this

- What if something breaks?
- Services such as RDS tend to create a vendor lock-in
- In some regions, other providers might be cheaper (e.g. AliCloud in China)

Multi-cloud!

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

team@Azure:~$ cd infrastructure/
team@Azure:~/infrastructure$ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


------------------------------------------------------------------------

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  + azurerm_resource_group.production-america
      id:       <computed>
      location: "westus2"
      name:     "production-america"
      tags.%:   <computed>

  + azurerm_resource_group.production-europe
      id:       <computed>
      location: "westeurope"
      name:     "production-europe"
      tags.%:   <computed>


Plan: 2 to add, 0 to change, 0 to destroy.

------------------------------------------------------------------------

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.
```

- Thanks to Infrastructure as Code and automation a multi-cloud setup is not rocket science
- Solutions such as Terraform abstract the different cloud APIs

Hybrid-cloud?

- Some situations require to store sensitive data on-premise
- Some workloads need a dynamic approach in terms of resource while others are purely static
- Some workloads can’t be run in a public cloud (Oracle DB)

Hybrid-cloud!

- Solutions such as Azure Stack or OpenStack offer the same grade of automation through APIs, ARM, etc.
- Combine such solutions to benefit from the cloud advantages (automation, flexibility, scalability, etc.)
- Examples:
  - Build pipeline in the cloud, resource hungry builds in the public cloud
  - Video encoding in the cloud
  - Webserver in a single datacenter, CDN to reduce latency

More about those topics

- Azure Stack – Your Cloud, Your Datacenter (Level 200-300), Kino 1

#Outro

What's next?

- Adapt new technologies and platforms
  - Cloud solutions
  - Invest in containers and serverless
  - Think X-Platform
- Become agile
  - Manage Infrastructure as Code
  - Invest in automation
  - Use SCM and release pipelines (CI/CD)

Conclusion

- No need to fear anything, but prepare for a change!
  - Be open for new concepts and approaches
- Give up old patterns
  - Procedures and knowledge from 5+ years ago are history - let them go!
- Focus on where you can bring real/unique value to your business and customers

Microsoft

#techsummitCH

Please Complete your Session Evaluations

Get your cool IoT Dev Kit!

Fill out your feedback form and turn it in before you leave.

Microsoft