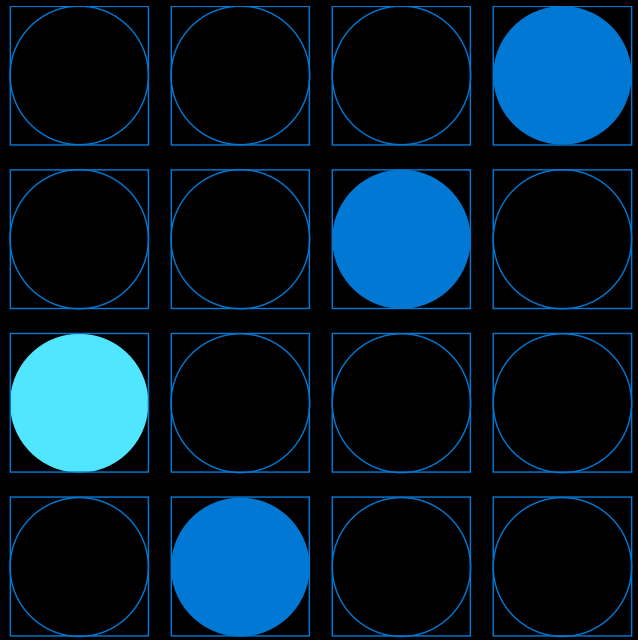
Microsoft Azure

Microsoft Azure Training Day: Modernizing your application with containers and Serverless



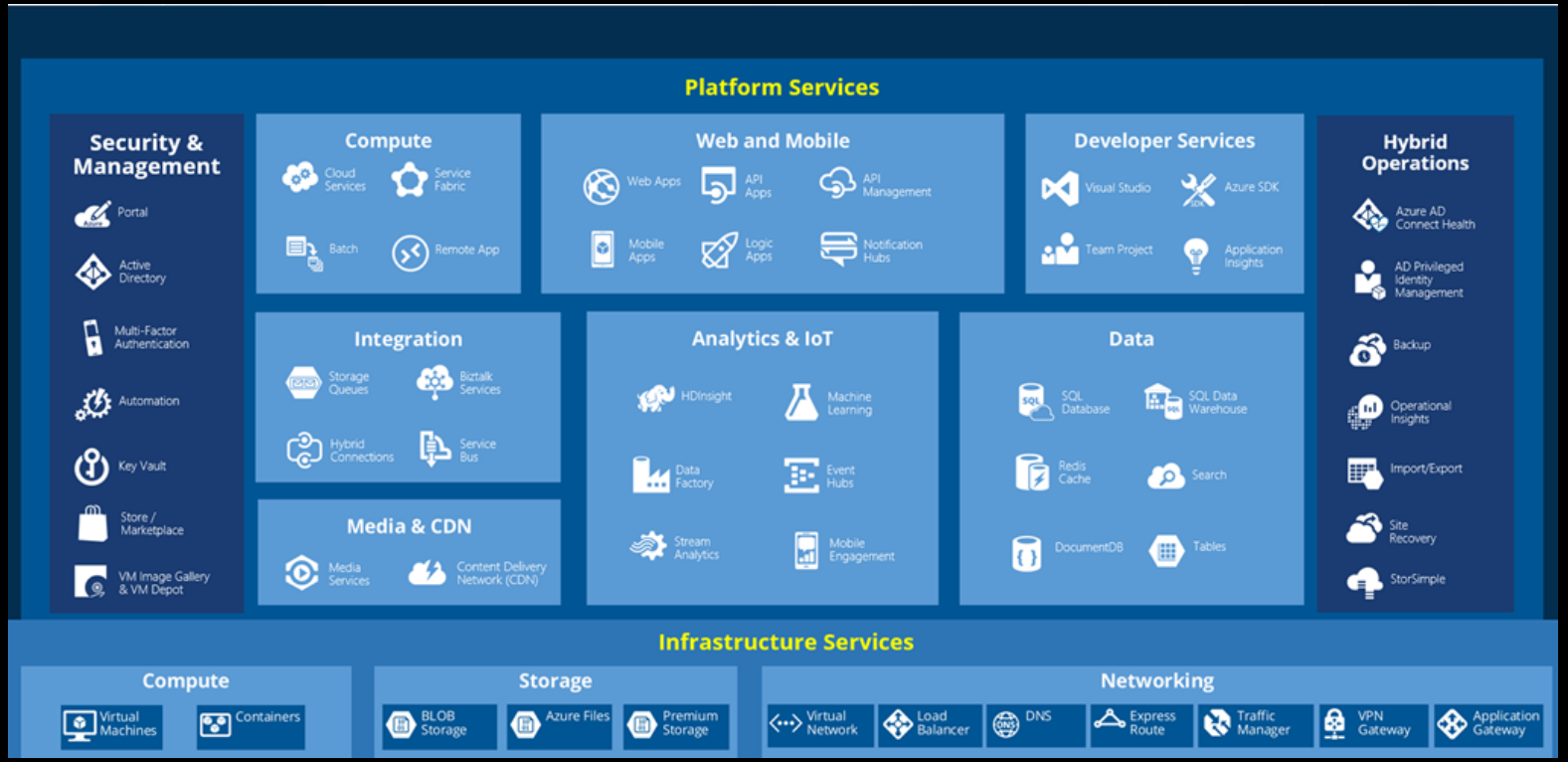
Irina Kostina, Cloud Solution Architect ISV

https://linkedin.com/in/irina-kostina
Irina.Kostina@Microsoft.com

Goals for this session.

- Review Container Service Options + Demo
- Secure Secret Storage + Demo
- Functions + Demo

But where do you start?



Containers

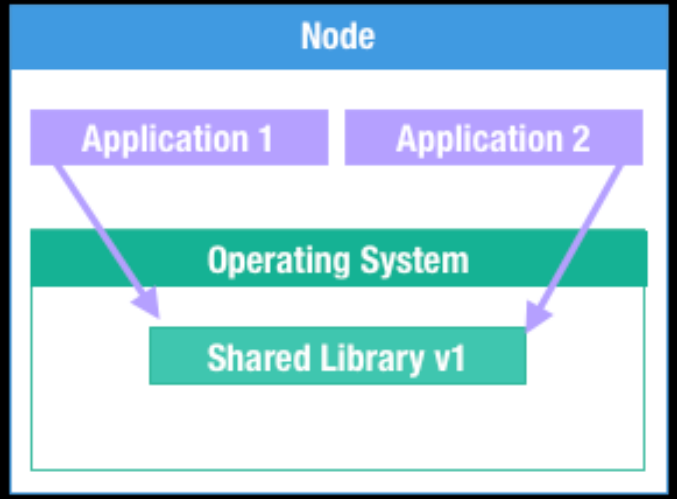
What they are and why they matter

What problem are we trying to solve?

What’s wrong with VMs?

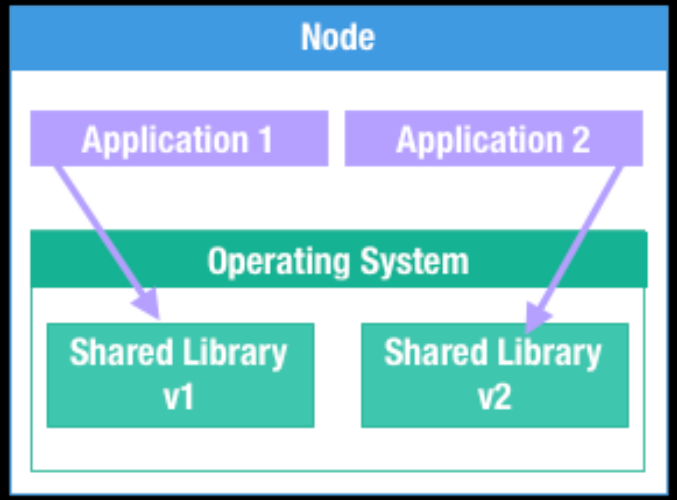
Why Containers?

Traditional Deployment



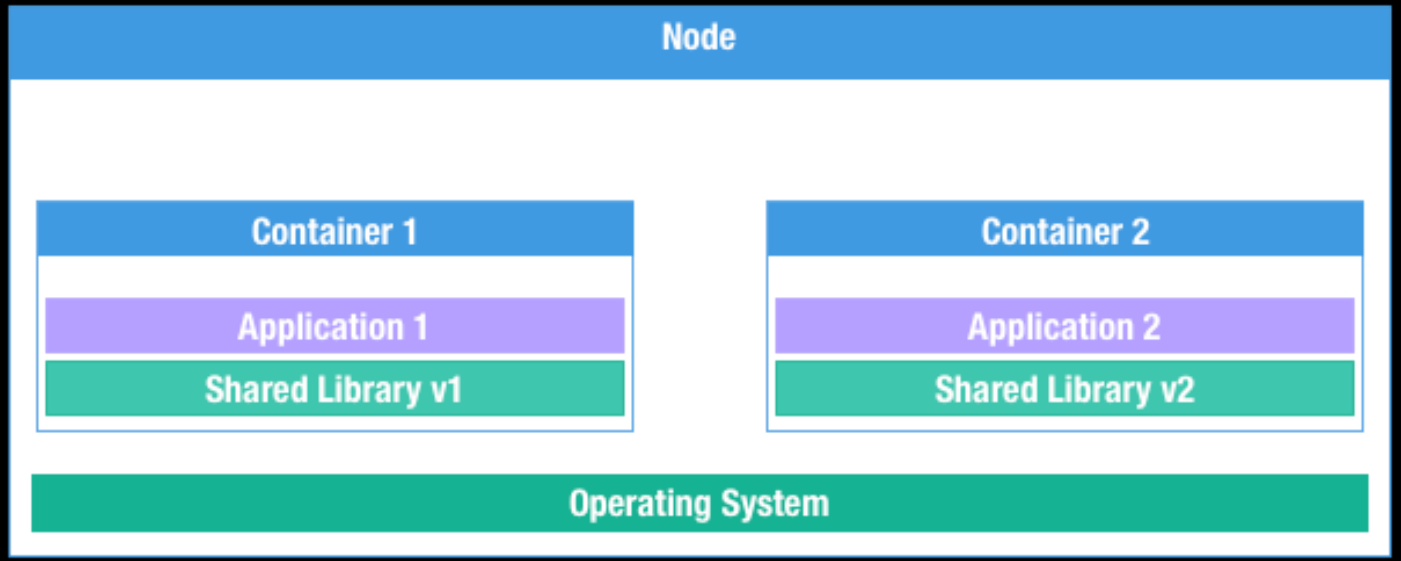
”Shared dependency” problem

Traditional Deployment



”Shared dependency” problem

Packaging and Deployment



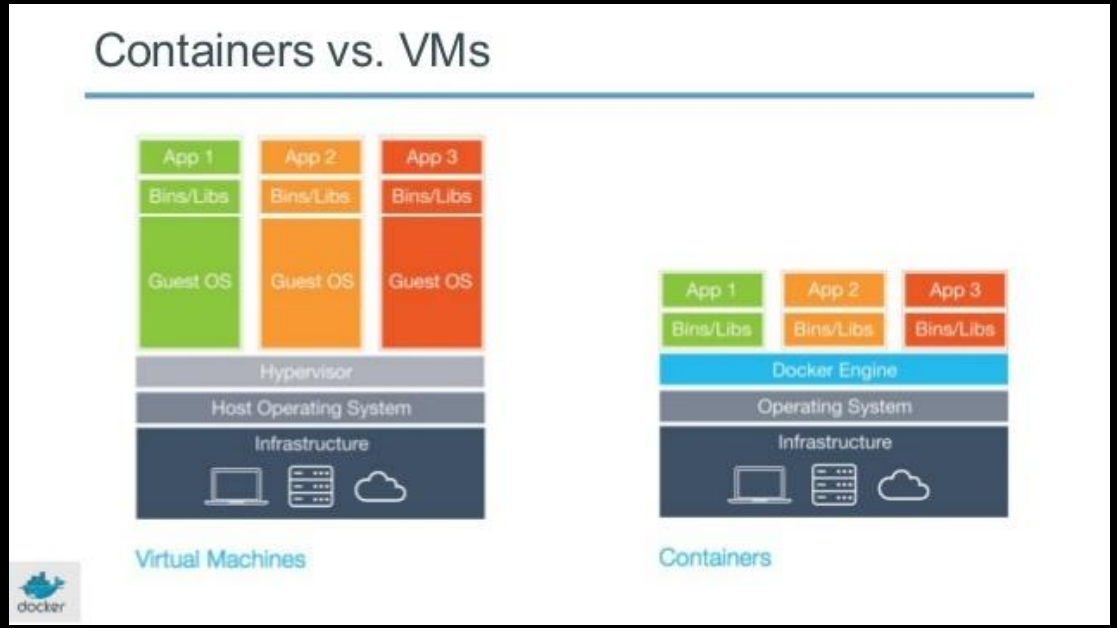
Distribution problem

Deployment and Distribution

How do they work?

A Little Bit Like a VM, But...

- Open format (OCI) & API
- Shared OS Kernel
- *Different security model*
- *Smaller, Faster*



How do containers work?

- Highly configured processes.
- They leverage kernel features:
  - **Cgroups** control the resources it can consume (cpu, memory, blkio, devices, net_prio, etc)
  - **Namespaces** control what the process can see (net, mnt, pid, user, ipc, etc)
- Orchestrator pulls image and executes it

Security considerations

- Containers share a kernel with the host OS
- While containers offer a degree of isolation, they are not a hard security boundary
- Kernel exploits and misconfigurations can allow an attacker to break out of the container
- **SELinux, AppArmor, Seccomp** profiles can help to minimize the attack surface. (Linux)
- Windows Hyper-V containers are highly tuned VMs and *do* provide VM level isolation

Windows containers

- **Windows** - contains the full set of Windows APIs and system services (minus server roles).
- **Windows Server Core** - a smaller image that contains a subset of the Windows Server APIs–namely the full .NET framework. It also includes most server roles (~9GB)
- **Nano Server** - the smallest Windows Server image, with support for the .NET Core APIs and some server roles (~600MB)
- **Windows 10 IoT Core** – for hardware manufacturers for IoT devices that run ARM or x86/x64 processors

Wrap up - What are Containers?

- Clear boundaries for your applications, allowing you to know which assets belong to which application and who owns it.
- No more leftover cruft from previous installations or versions.
- Provides resource isolation, without the overhead of more VMs.
- Better resource utilization.

Demo

- Building docker image

What else are they good for?

Multi-Stage Builds

```
FROM golang:alpine as builder

RUN     apk add --no-cache \
        ca-certificates

RUN go build


FROM scratch

COPY --from=builder /usr/bin/hello-golang /usr/bin/hello-golang
COPY --from=builder /etc/ssl/certs/ /etc/ssl/certs

ENTRYPOINT [ "hello-golang" ]
```

Azure Container Options

I built a container… What do I need?

- Build
  - -Handle and update dependencies including the OS
- Deploy
  - -Run containers across multiple machines
  - -Configuration as code
- Operate
  - -Monitor
  - -Scale up/down

Containers in Azure

App Service

Deploy web apps or APIs using containers in a PaaS environment

Service Fabric

Modernize .NET applications to microservices using Windows Server containers

Kubernetes Service

Scale and orchestrate Linux containers using Kubernetes

Container Instance

Elastically burst from your Azure Kubernetes Service (AKS) cluster

Ecosystem

Bring your Partner solutions that run great on Azure

Azure Container Registry

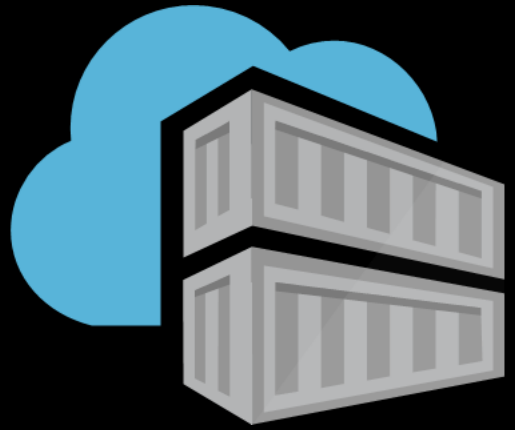
Docker Hub

Choice of developer tools and clients

Azure Container Registry
&
Azure Container Instances

Azure Container Registry

- Use same docker commands as Docker Hub
- Private
- Eliminates egress/ingress traffic and latency
- Geo-replication - a single registry replicated across multiple regions
- Authenticate with Azure Active Directory
- Automatic Vulnerability Scanning (CVE)
- Azure Container Registry Tasks / Builders



Azure Container Instances

- Start in seconds
- No VM management
- Custom CPU/memory
- Billed per second
- Public IP
- Hypervisor-level isolation
- Linux and Windows containers

$48/month for 512MB and 1 vCPU

- Batch processing
- Lift and shift into containers
- Simple web services
- Compute intensive workloads
- CI agents

ACI Permanent Limitations

- No support for:
  - Zero-downtime upgrades
  - Maintain > 1 running replicas
  - Batch workflows
  - Service discovery
  - Integrated load balancing
  - Auto-scaling

If you need these features, you probably need an orchestrator

Azure Container Instances demo

But We’re Developers and Operations

Why do manually what you can automate?

Three Different Ways To Build Containers

- docker build; docker push*
- az acr build*
- Azure Devops Pipeline

*manual

Secrets

Everybody has them

The Problem

- We need database credentials in app
- Insecure: hard-code secrets into `Dockerfile`
- Insecure: hard-code secrets into `scripts`

- How do we securely **store** and **inject** secrets?

Azure KeyVault

- Secure storage for secret data
- Everything stored in Hardware Security Modules (HSMs)
- Integrated with many Azure Services
- Store Keys, Secrets, Certificates
- Strict control and auditing of Key Vault

Create a Keyvault

```
az keyvault create \
--resource-group $RES_GROUP \
--name $AKV_NAME
```

Store a Secret

```
az keyvault secret set \
--vault-name $AKV_NAME \
--name 'web3-mongo-connection’ \
--value $DB_CONNECTION_STRING
```

Read a Secret on the Command Line

```
az keyvault secret show \
--vault-name $AKV_NAME \
--name web3-mongo-connection \
--query value \
-o tsv
```

Azure Functions

Serverless to next level

Azure Functions Overview

- React to external **triggers**
- Azure defines supported languages
- "glue" services together
- Small applications, simple deployments
- Available in: C#, F#, Java, JS, Python(3), PowerShell

Timers

- Built-in Azure functions trigger
- Calls your code on an interval

Reports?

- Replace existing ‘cron’ jobs
- Nightly reports emailed to staff

Demo Azure Functions

HttpTrigger function accessing Secret in KeyVault

Putting it All Together

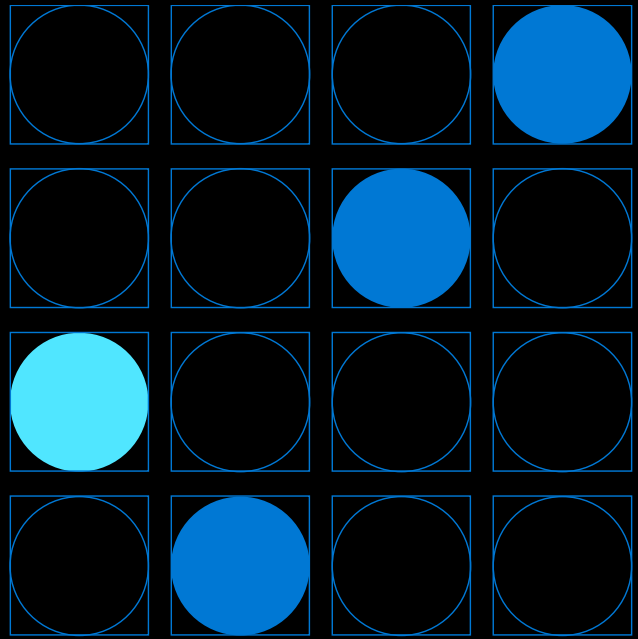
- Website Code ->Docker Image -> Azure Container Registry + ACI
- “Cron job” -> **Azure Functions**

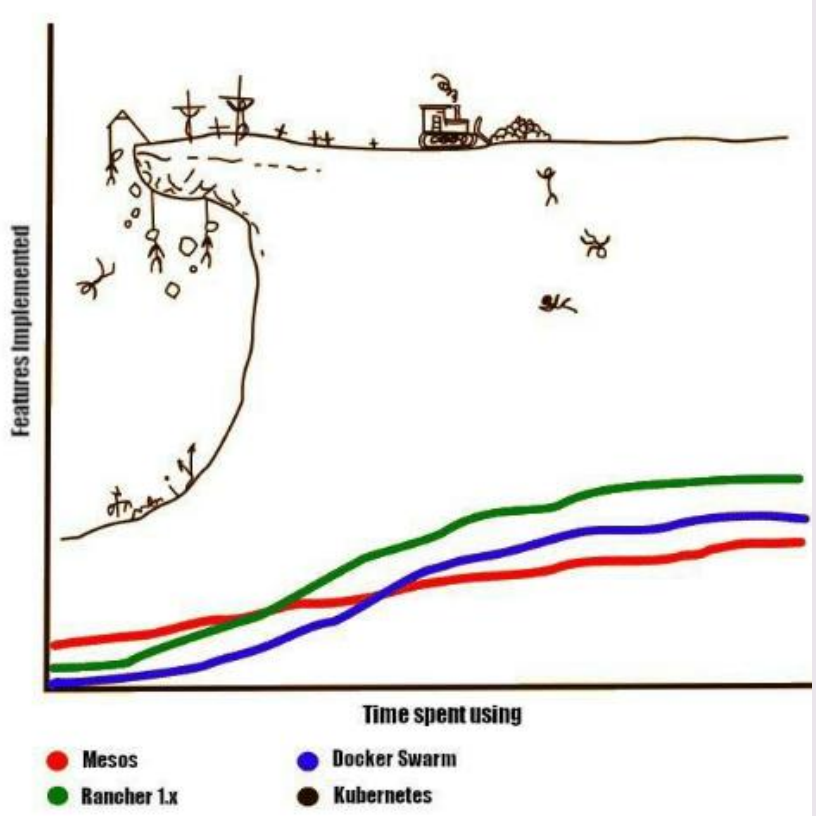
Microsoft Azure

Thank you!

Irina Kostina, Cloud Solution Architect ISV

https://linkedin.com/in/irina-kostina
Irina.Kostina@Microsoft.com





Learning curves for some container orchestration engines