Microsoft Azure

Microsoft Azure Training Day: Migrating Applications to the Cloud

Modernizing your application with containers and Serverless

Abdul Rasheed Feroz Khan
Microsoft MVP - Azure
CTO – CodeSizzler
www.codesizzler.in
feroz@codesizzler.info

Goals for this session.

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

Mission control

Your mission, should you choose to accept it…

THE DAILY NEWS

theuselessweb.com THE WORLD’S MOST MEH NEWSPAPER - Since 5 seconds ago

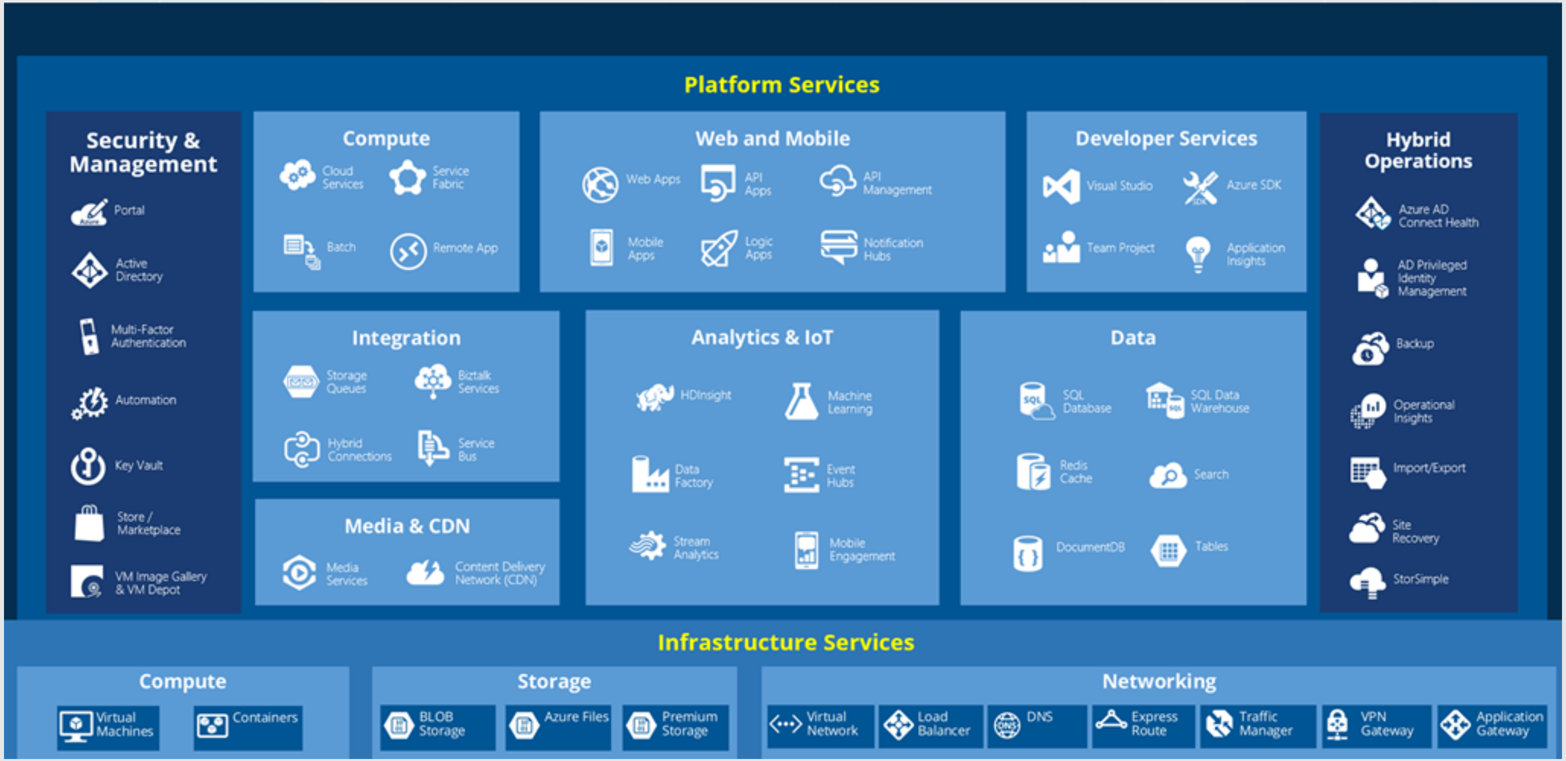
Tailwind Traders Acquires Northwind

REDMOND, WA - **Tailwind** Traders, Inc announced that it has agreed to acquire Northwind **Traders**, the venerable "old guard" international foods company, in an effort to bolster its virtual hardware offerings with some international culinary flare. Tailwind agreed to acquire Northwind for an undisclosed sum and **is** assuming control of all of Northwind's virtual assets, **not** excluding the company's flagship product: Aniseed Syrup/

"This is a profoundly

for all involved, and will help to bring our food products to **a** whole new sector of the market" stated Nancy Daviolo, Executive Vice President of Operations. "We **real**ly couldn't be happier with the arrangement". The recently-promoted Daviolo will join the Tailwind executive team along with two of her close associates, Margaret Peacock and Michael Suyama. Daviolo took over Northwind’s **company** sales operations in 2012 after Andrew Fuller was

But where do you start?



Tailwind architecture

Migrating to Azure

The Tailwind Traders App

- Ecommerce web app
- Three microservices:
  - Frontend (React)
  - Inventory service (C#)
  - Product service (Node.js)

Technologies

- Docker: open source container technology
- Azure Container Registry (ACR): highly available, secure container registry
- Azure App Service for Linux: container orchestrator

Dockerizing Microservices

We'll demo:

- One `Dockerfile` per folder
- Two ways to build images
- Push to ACR

Azure App Service

Azure App Service

- Fully Managed Application Platform
- Deploy your stack natively or with containers (Node, PHP, .NET, Python)
- Support custom domains, SSL certificates, single sign-on
- Plug into Azure's wide variety of services
  - Load Balancing
  - CI/CD
  - Managed Databases

Azure App Service

- Continuous Delivery based on code changes or container pushes
- High Availability with multi-region deployments and autoscaling
- Azure Monitor provides detailed views of resource usage, while Application Insights provides deeper insights into your app’s throughput, response times, memory/CPU utilization, and error trends.

Containers

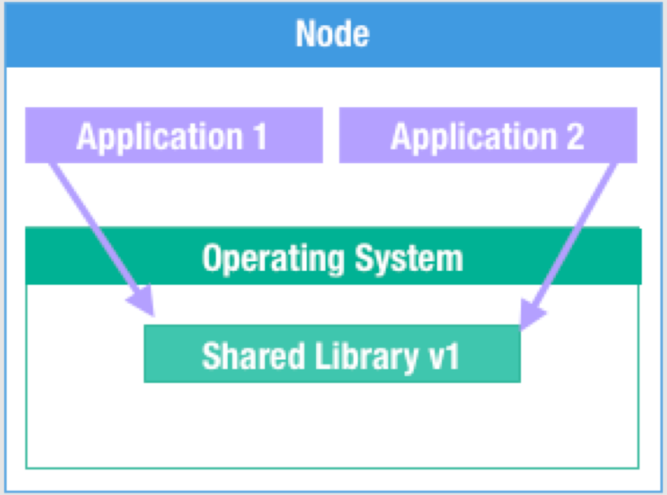
What they are and why they matter

What problem are we trying to solve?

What’s wrong with VMs?

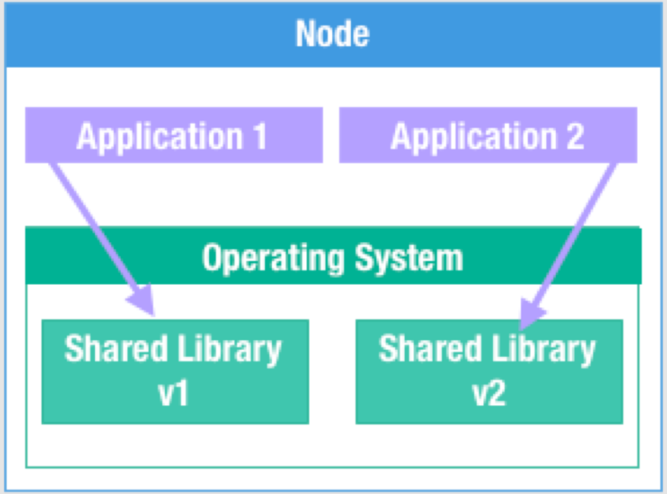
What are Containers?

Traditional Deployment



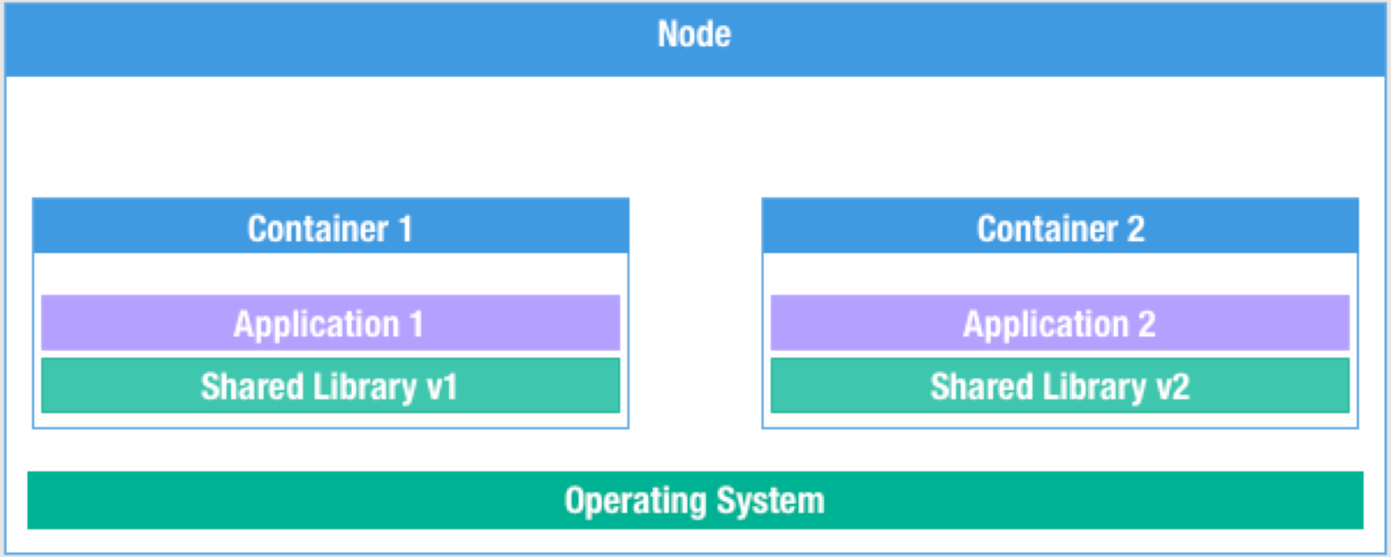
What are Containers?

Traditional Deployment



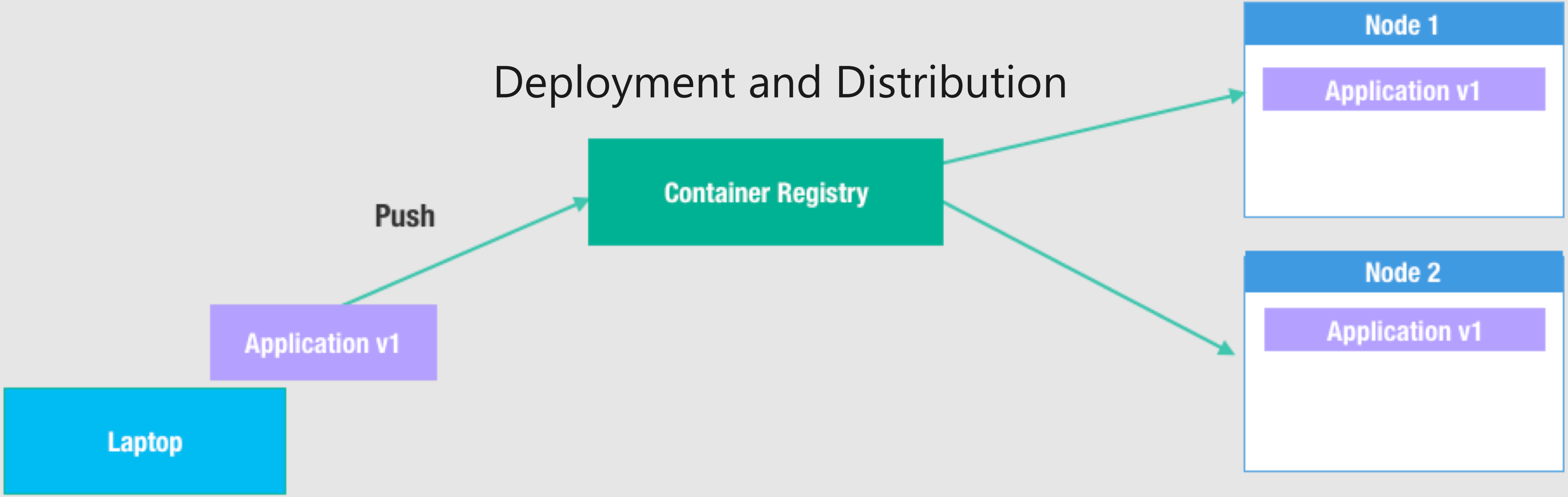
What are Containers?

Packaging and Deployment



What are Containers?

Deployment and Distribution



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.

What else are they good for?

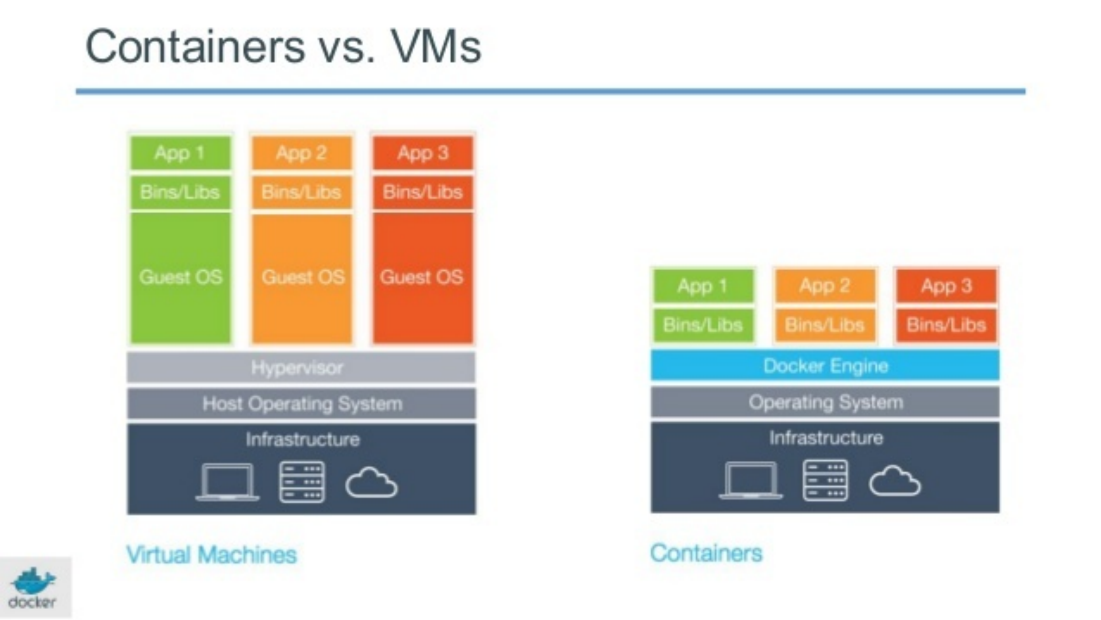
Multi-Stage Builds

```
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" ]
```

How do they work?

A Little Bit Like a VM, But…

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



Containers vs. VMs

Containers are not Virtual Machines

- 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.

How do containers work?

- Containers are basically highly configured processes.
- They leverage kernel features to isolate processes.
- 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)
- The image format is used to move the root directory the process sees around with the container image.

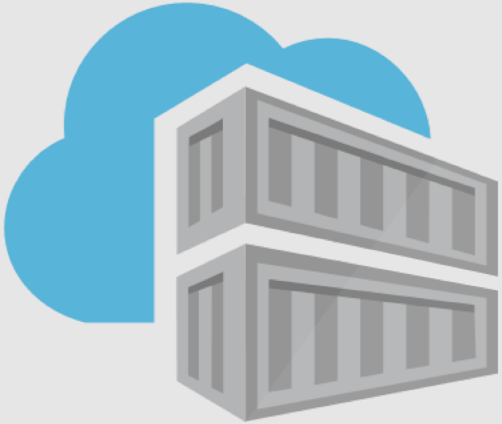
Open Source Ecosystem

- Standards bodies for relevant technologies
- Open source implementations
- Fundamental building block

Azure Container Registry

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
- Azure Container Registry Tasks / Builders



Demo

Building and Pushing Images to Azure

But We’re Developers and Operations

- Why do manually what you can automate?

Demo

Azure DevOps Pipelines

Three Different Ways To Build Containers

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

*manual

Deploying Microservices

- Our images are in the registry
- Azure App Service for Linux runs our containers

Deploying to Azure App Service

```
az webapp create \
-g $RES_GROUP \
-n $WEB_APP_3 \
--plan $APP_SERVICE \
--deployment-container-image-name 'nginx'
```

Demo

Deploying to App Service for Linux

SIDEBAR: Makefile

"A **makefile** is a file (by default named `Makefile`) containing a set of directives used by a **make build automation tool** to generate a target/goal."

Wikipedia

SIDEBAR: Makefile

```
target: dependency
    task
    task
```

SIDEBAR: Makefile

```
deploy: login
    @scripts/up/deploy.sh
login:
    @scripts/login.sh
```

scripts/up/deploy.sh

```
deploy: login
    @scripts/up/deploy.sh
```

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
```

Read Secrets from App Code

```
const KeyVault = require("azure-keyvault");
…
const secrets = await client.getSecrets(uri);
```

Demo

Storing and Accessing Secrets in KeyVault

Azure Functions

Serverless to the next level

Reports?

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

Azure Functions Overview

- React to external **triggers**
- Azure defines supported languages
- "glue" services together
- Small applications, simple deployments

Timers

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

Demo

Calculating a Report Every X Minutes

Putting it All Together

- Code -> Image
- Image -> Azure Container Registry (ACR)
- ACR -> Azure App Service for Linux
- “Cron job” Trigger -> Azure Functions

Microsoft

Thank you!