Building and Deploying Container Based Applications

Anthony E. Nocentino
aen@centinosystems.com

Course Overview

- **Module 0** - Introduction
- **Module 1** - Container Fundamentals
- **Module 2** - Kubernetes Architecture and API Objects
- **Lunch @ 12:00-12:45**
- **Module 3** - Interacting With Your Cluster
- **Module 4** - Deploying Applications in Kubernetes
- **Module 5** - Building and Deploying Container-based Applications in Kubernetes

Agenda

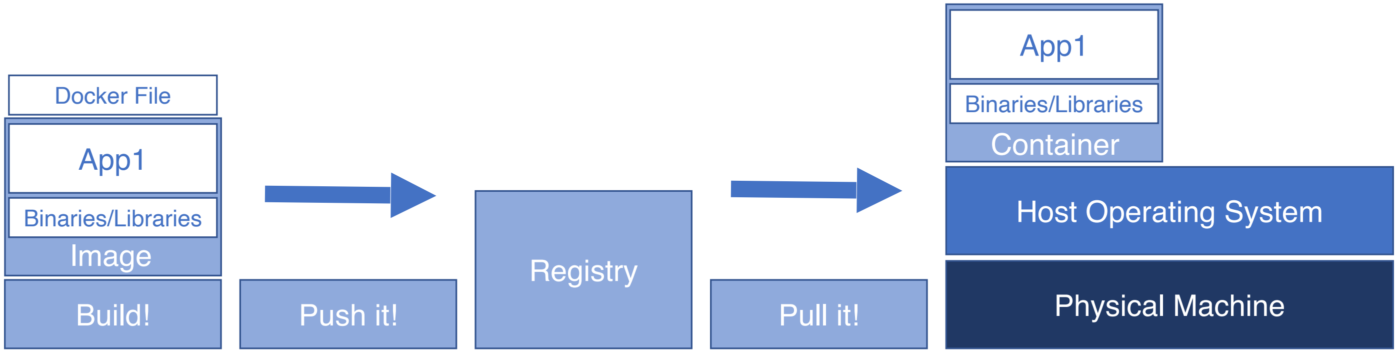
- **Creating a Container Image**
- **Working with a Container Registry**
- **Deploying and Scaling our Application in Azure Kubernetes Service**
- **Updating our Container Image and Deployment**

Containerizing Apps and Data Centers

- Reducing development time
- Deployment automation – speed and consistency
- Enables DevOps and CI/CD scenarios
- Rethink how you deploy - it’s the application service, not the server

Getting/Creating Containers

- **Images** – code, runtimes, libraries, environment variables
- **Registries** – where images live. Docker Hub, Azure Container Registry, internal
- **Docker Files** – defines the container image



Docker Files

```
FROM mcr.microsoft.com/dotnet/aspnet:7.0
RUN mkdir /app
WORKDIR /app
COPY ./webappv1/bin/Release/netcoreapp7.0/publish ./
EXPOSE 80
ENTRYPOINT ["dotnet", "webapp.dll"]
```

```
docker build -t webappimage:v1 .
```

https://docs.docker.com/engine/reference/builder/

Multi-Stage Builds

```
FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build
WORKDIR /source

# copy csproj and restore as distinct layers
COPY v1/webapp/*.csproj ./webappv1/
WORKDIR /source/webappv1
RUN dotnet restore

# copy everything else and build app
COPY v1/webapp/. ./webappv1/
WORKDIR /source/webappv1
RUN dotnet publish -c release -o /app --no-restore

# final stage/image
FROM mcr.microsoft.com/dotnet/aspnet:7.0
WORKDIR /app
COPY --from=build /app ./
EXPOSE 8080
ENTRYPOINT ["dotnet", "webapp.dll"]
```

Pull dotnet sdk

Copy project file in and restore dependencies

Copy source code in and compile the application

Build the final image

Why do this?

Container Registries

- Store container images
- Public or private
- Secured
  - Transport - HTTPS
  - Image digests - hash of image
- Key component of building a CI/CD pipeline
- Images are organized by tags
- Docker Hub
- Azure Container Registry
  - mcr.microsoft.com

What’s Next?

- Building a Data Tier
  - Database Service
  - Database Connections
- Production Ready App Tier
  - Connection Strings in Azure Key Vault
  - SSL Termination (AppGW, Ingress…etc)
- DevOps
  - Automatically build container image
  - Automatically deploy to Kubernetes using a **`Deployment`**
  - Azure DevOps

Hands On Lab!

- Creating a container based application
- Push our container to Docker Hub
  - **`Deployments`**
  - **`Services`**

Scaling our application from 1 to 10 Replicas

Updating a container image and deployment

More Resources

- **Docker for Windows/Mac**
- **Managed Service Providers**
  - Azure Kubernetes Service **(AKS)**
    - https://docs.microsoft.com/en-us/azure/aks/kubernetes-walkthrough
- **Pluralsight!**
  - https://app.pluralsight.com/profile/author/anthony-nocentino

Review

- **Creating a Container Image**
- **Working with a Container Registry**
- **Deploying and Scaling our Application in Azure Kubernetes Service**
- **Updating our Container Image and Deployment**

Need more data or help?

http://www.centinosystems.com/blog/talks/

Links to resources
Demos
Presentation
Pluralsight

aen@centinosystems.com
@nocentino
www.centinosystems.com

Solving tough business challenges with technical innovation

Thank you!

- Questions