Practical Container Scenarios in Azure

Anthony E. Nocentino

aen@centinosystems.com

SQLBITS 2020

Anthony E. Nocentino

- **Consultant and Trainer**

- **Founder and President of Centino Systems**

  - Specialize in system architecture and performance

  - Masters Computer Science

  - Microsoft MVP - Data Platform - 2017 - 2020

  - Linux Foundation Certified Engineer

  - Friend of Redgate - 2015-2019

- **email**: aen@centinosystems.com

- **Twitter**: @nocentino

- **Blog**: www.centinosystems.com/blog

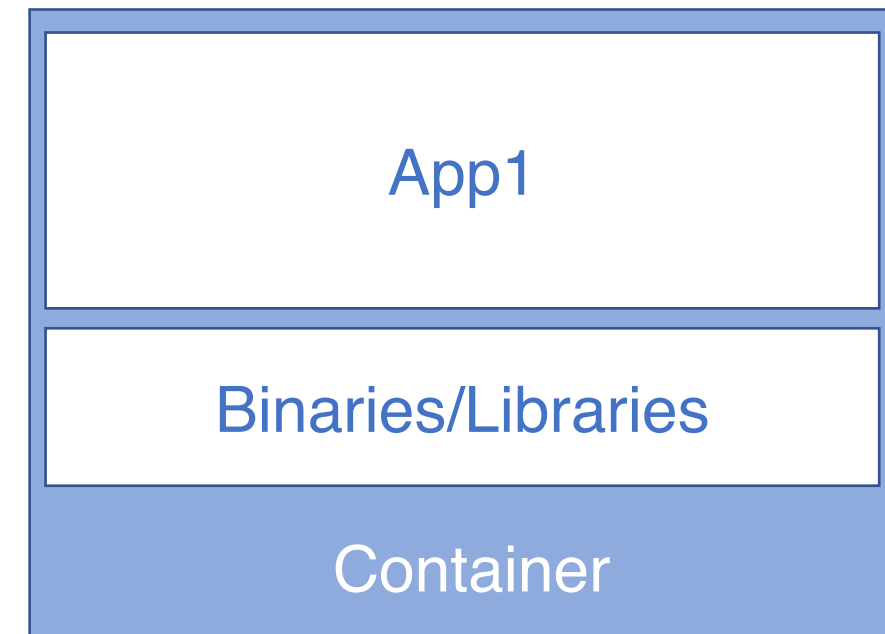- **Pluralsight Author**: www.pluralsight.com

Centino systems

Agenda

- **Container Fundamentals**

- **Creating a Container Image**

- **Working with Azure Container Registry**

- **Deploying our Application in Azure Kubernetes Service**

Containerizing Apps and Data Centers

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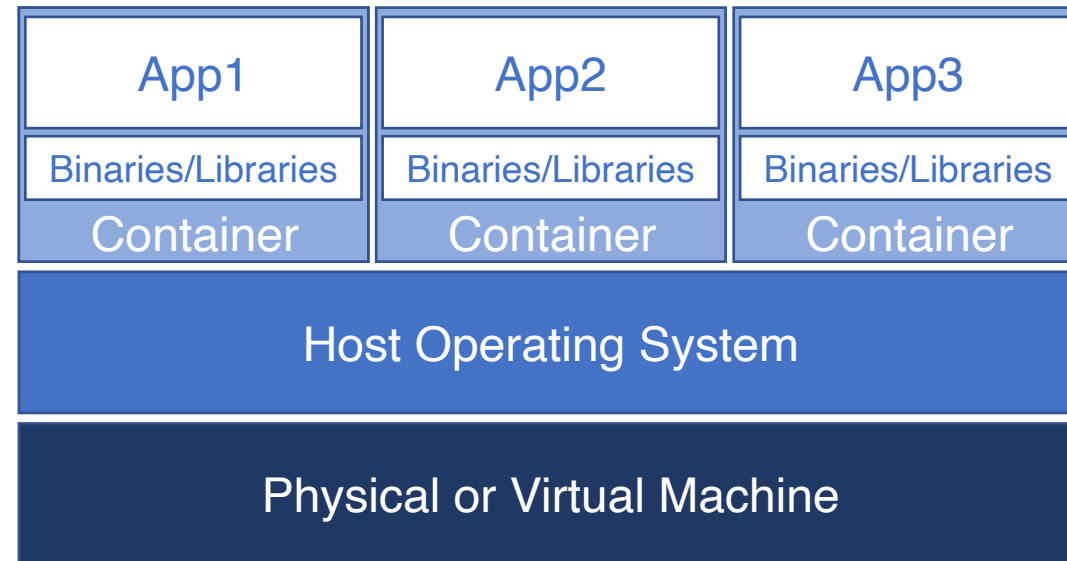- Rethink how you deploy - it's the application service, not the server

Container Fundamentals

- Operating system virtualization

  - Shared kernel and system resources

- Container…contain…

  - Binaries, libraries and file system

- One app inside the container

  - This is the unit of work

- Containers are ephemeral

App1
Binaries/Libraries
Container

Containers

Patching/Deployments/Whatever

App1	App2	App3
Binaries/Libraries	Binaries/Libraries	Binaries/Libraries
Container	Container	Container

Host Operating System

Physical or Virtual Machine

Centino systems

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

App1

Binaries/Libraries
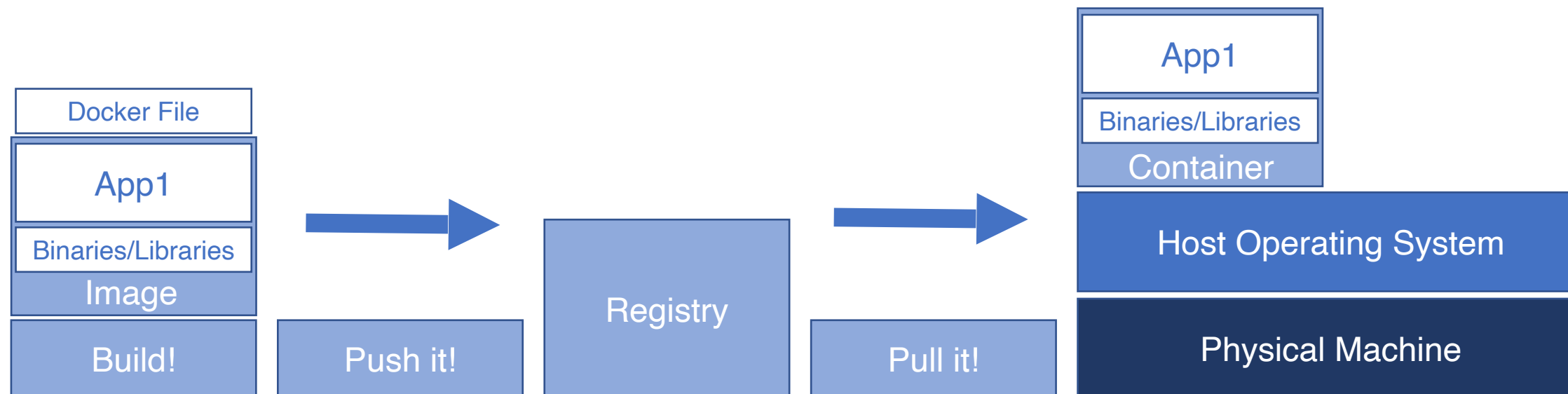
Container

- Describes the commands to build an **image**

```
FROM mcr.microsoft.com/dotnet/core/aspnet:3.1
COPY ./myWebApp/bin/Release/netcoreapp3.1/publish app/
ENTRYPOINT ["dotnet", "app/myWebApp.dll"]
EXPOSE 80
```
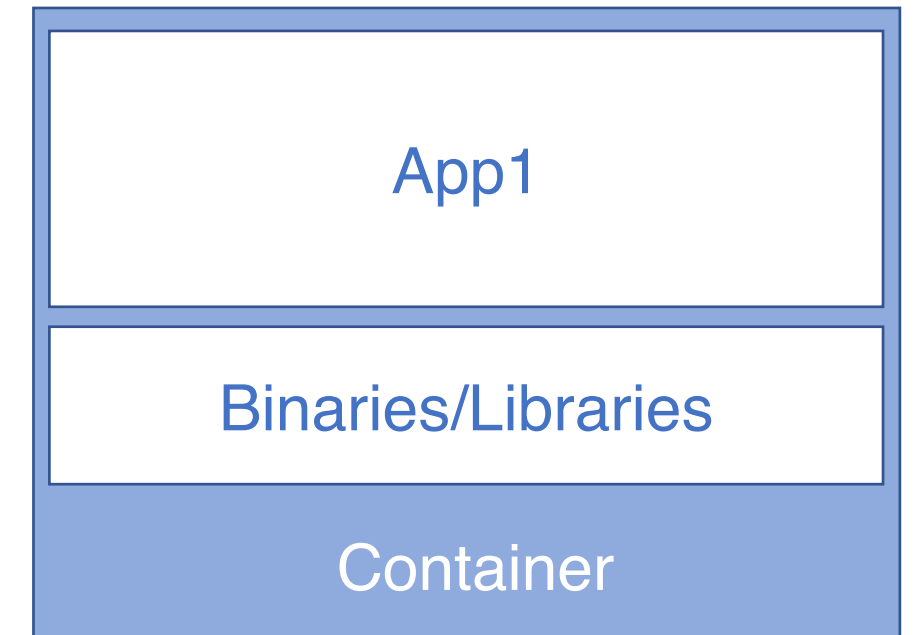
```
docker build –t mywebappimage .
```

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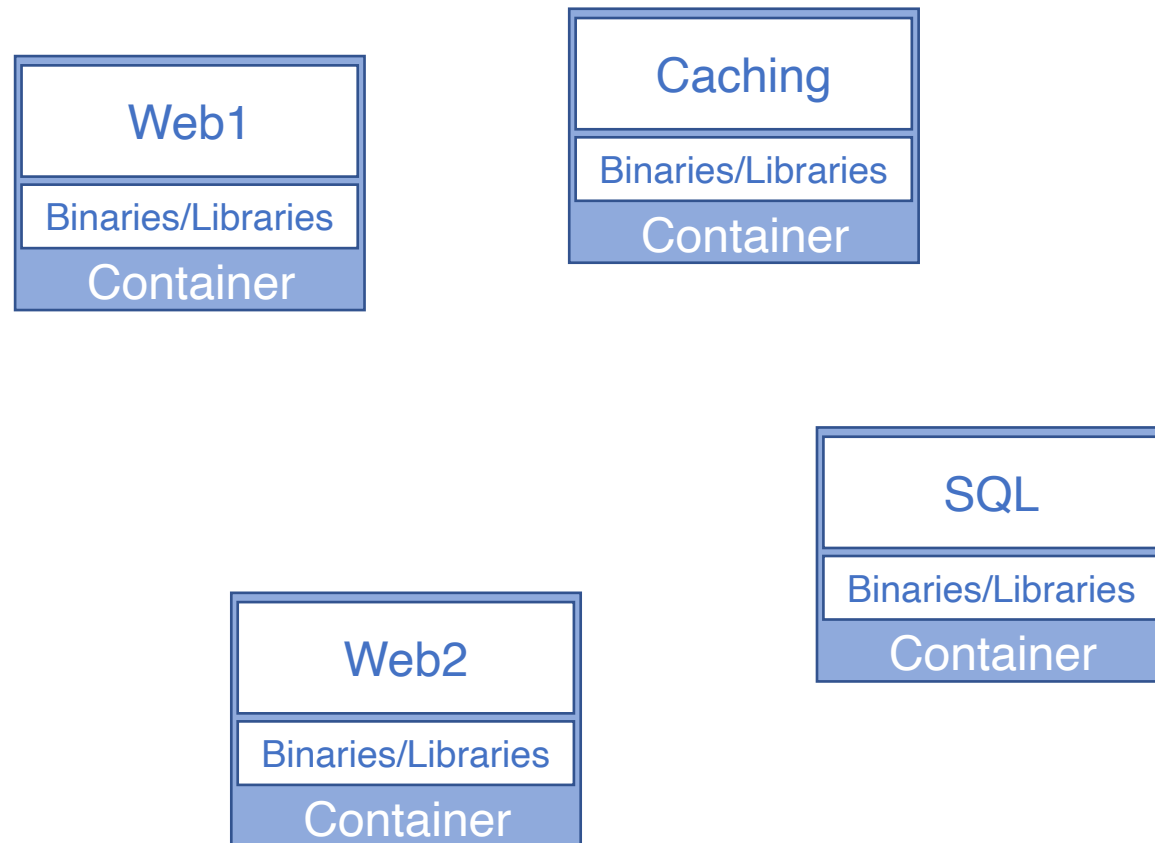
Centino systems

Container Registries

- Store container images

- Public or private

- Secured

  - Transport - HTTPS

  - Content - image digests hash of manifest and image's layers

- Key component of building a CI/CD pipeline

- Images are organized by tags

- Docker Hub

- Azure Container Registry

  - mcr.microsoft.com

Demo!

- **Creating a container based application**

- **Building it in Azure Container Registry**

Modern Application Deployment

Web1
Binaries/Libraries
Container

Caching
Binaries/Libraries
Container

Web2
Binaries/Libraries
Container

SQL
Binaries/Libraries
Container

- **Where do I run the application?**

- **How do I scale the application?**

- **How do I consistently deploy?**

- **How do I access the application?**

Centino
systems

Container Orchestration

- Workload placement

- Managing state, starting things up and keeping things up

- Networking and Services

- Load balancing services

- Persistent storage

- Declarative model

Container Orchestrators

- Docker Swarm/Enterprise

- Kubernetes

- Red Hat OpenShift

- Managed Services

  - Azure Kubernetes Services (AKS)

  - Google Kubernetes Engine (GKE)

  - Amazon Elastic Container Service for Kubernetes (EKS)

Centino
systems

Kubernetes Cluster

Caching

Web 1

SQL

Web 2

Cluster

Centino
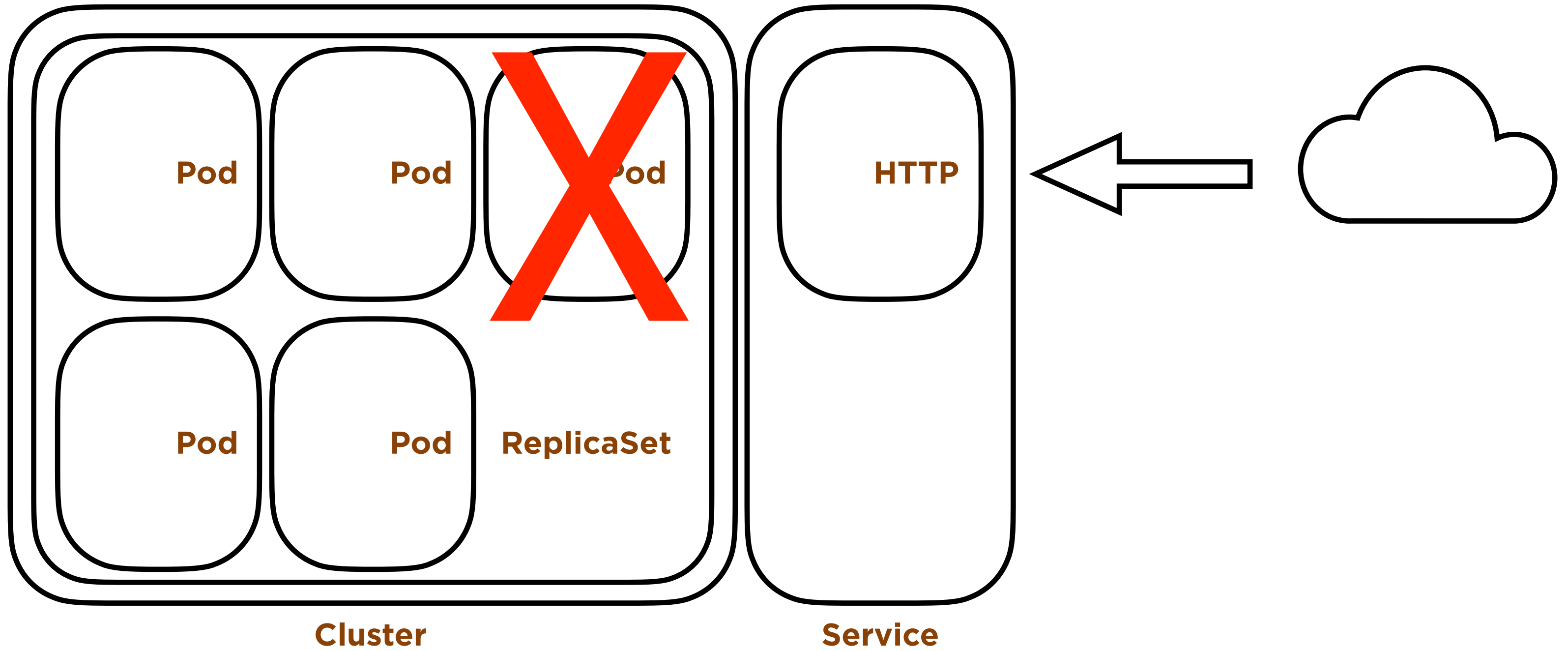systems
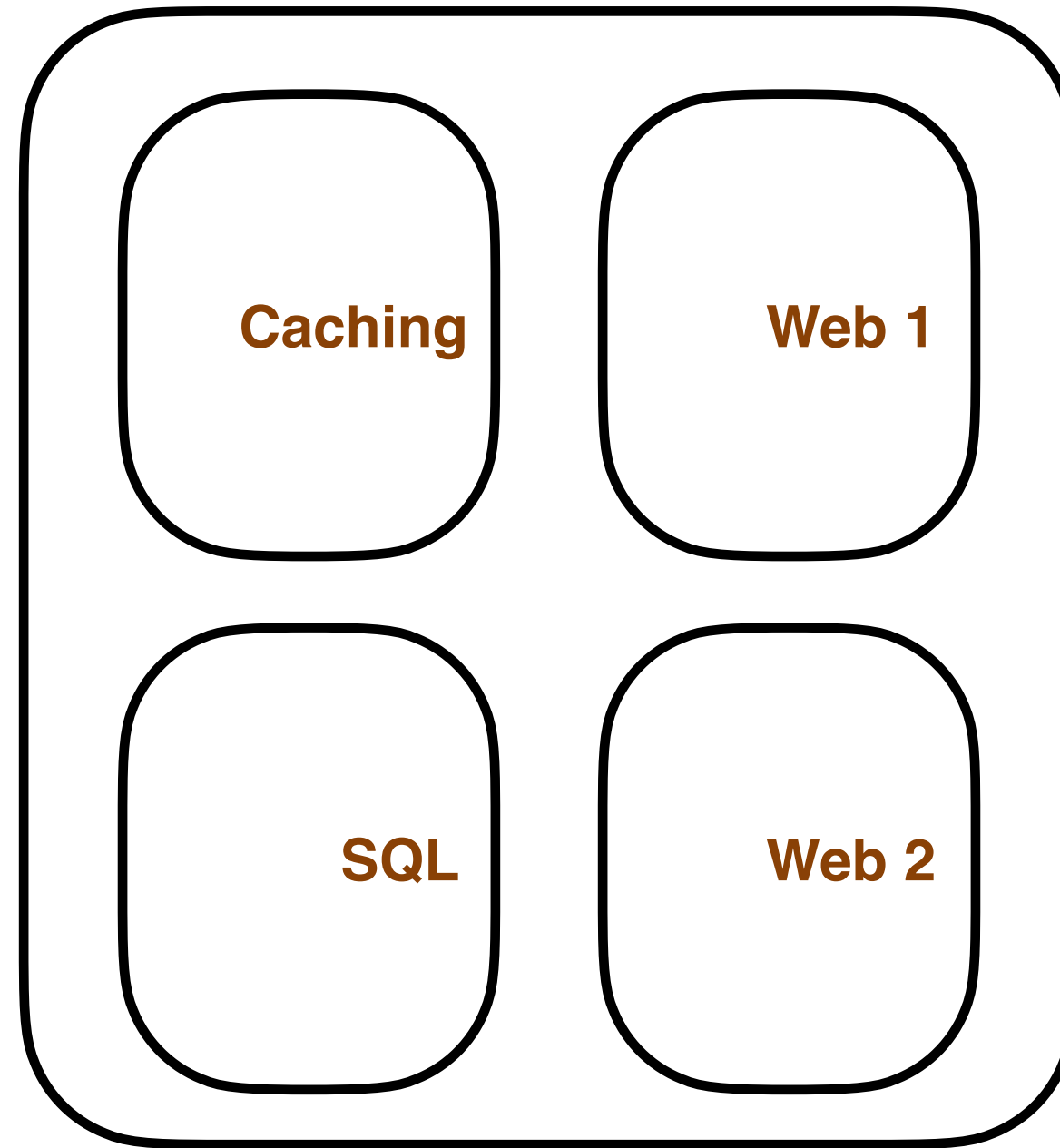
Azure Kubernetes Service

- Managed Cluster

- Upgrades handled in Azure (CLI/Portal)

- Define a number of Nodes (Agents)

- Nodes are in Availability Sets
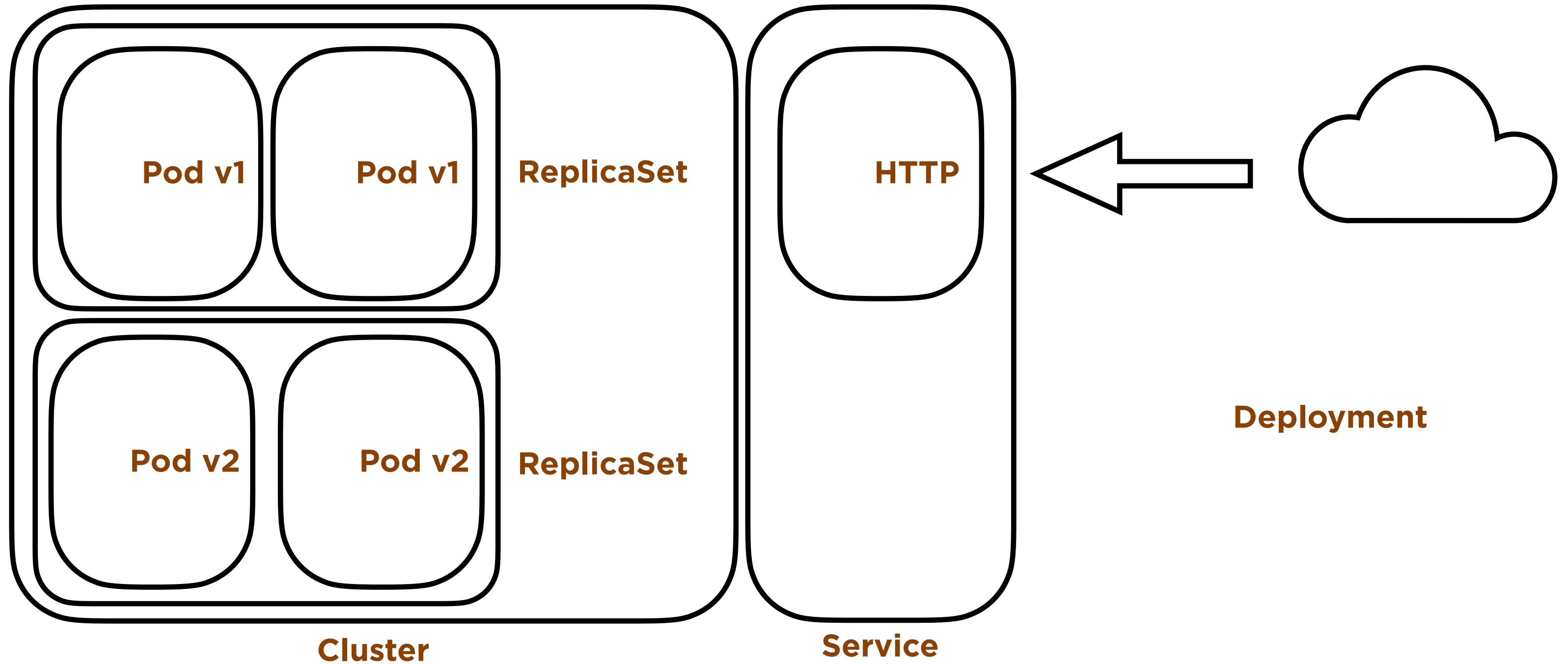
- Nodes are in Node Pools

Kubernetes API

- **API Objects** - Represent resources in your system

  - Really an API to the resources in your cluster…

  - **Pods** - your container based applications

  - **Controllers** - maintain desired state

  - **Services** - persistent access to your apps

  - **Storage** - persistent storage for your data

  - …and more

Centino
systems

Services and ReplicaSets

Pod

Pod

Pod

Pod

Pod

ReplicaSet

Cluster

HTTP

Service

Centino systems

Controller Operations - `Deployment`

Application Deployment in Kubernetes

- **Imperative**

  - `kubectl create deployment mywebapp --image=centinosystems.azurecr.io/mywebappimage`

- **Declarative**

  - Define our desired state in code

  - Manifest

  - YAML or JSON

  - `kubectl apply -f deployment.yaml`

Demos!

Declaratively Deploying Applications in AKS

- **Deployments**

- **Services**

Scaling our application from 1 to 50 Replicas

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

Review

- **Container Fundamentals**

- **Creating a Container Image**

- **Working with Azure Container Registry**

- **Deploying our Application in Azure Kubernetes Service**

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

Need more data or help?

http://www.centinosystems.com/blog/talks/
http://github.com/nocentino/presentations

Links to resources
Demos
Presentation
Pluralsight

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

Solving tough business challenges with technical innovation

Centino
systems

Thank you!

Pluralsight: www.pluralsight.com
email: aen@centinosystems.com
Twitter: @nocentino
GitHub: http://github.com/nocentino/presentations