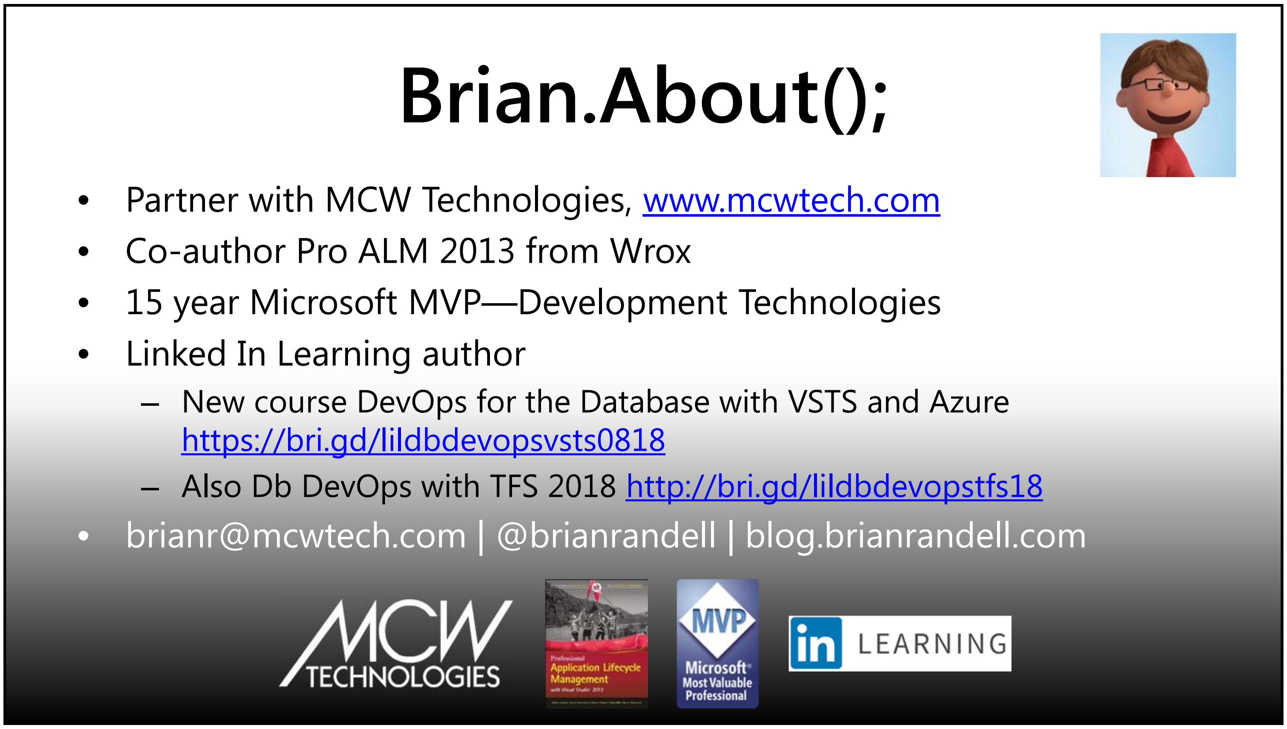
Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS | San Diego

Azure DevOps with VSTS, Docker, and K8s

Brian A. Randell
Partner, MCW Technologies

Level: Beginner

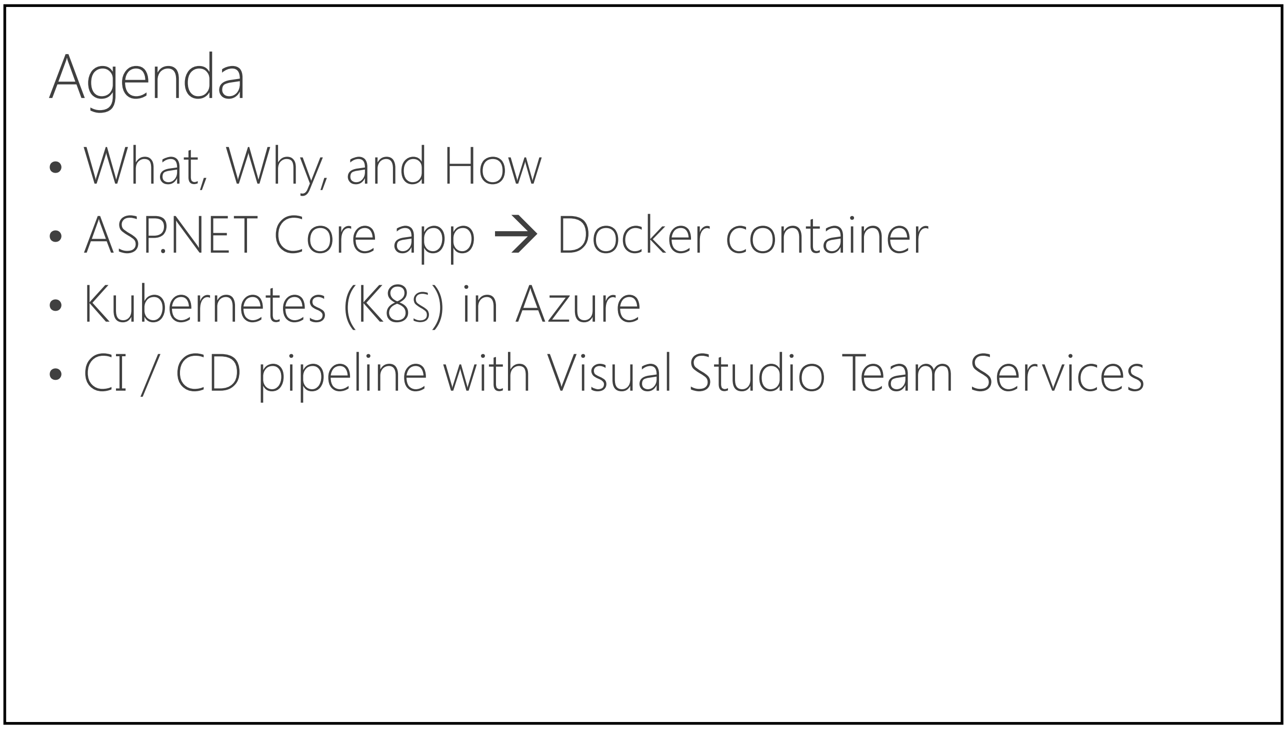
Code Again for the First Time!
Visual Studio 25 YEARS OF CODING INNOVATION



Brian.About();

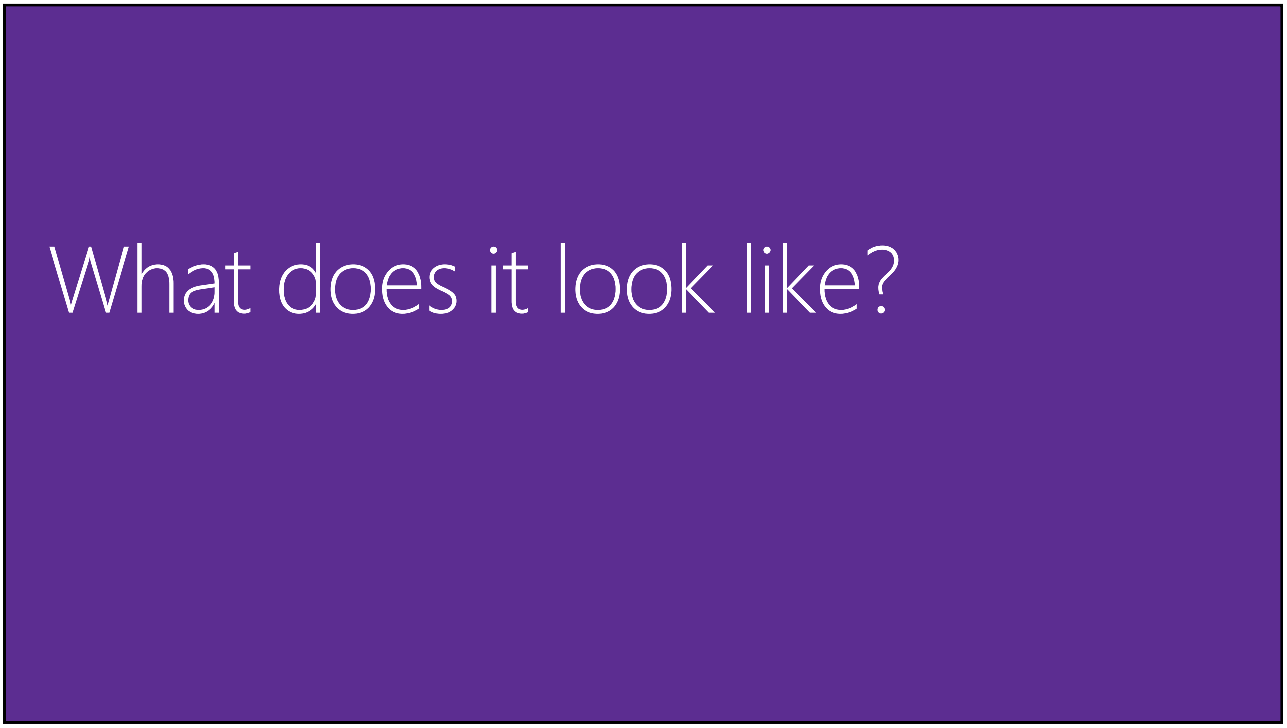
- Partner with MCW Technologies, www.mcwtech.com
- Co-author Pro ALM 2013 from Wrox
- 15 year Microsoft MVP—Development Technologies
- Linked In Learning author
  - New course DevOps for the Database with VSTS and Azure https://bri.gd/lildbdevopsvsts0818
  - Also Db DevOps with TFS 2018 http://bri.gd/lildbdevopstfs18
- brianr@mcwtech.com | @brianrandell | blog.brianrandell.com

MCW TECHNOLOGIES
Professional Application Lifecycle Management
MVP Microsoft Most Valuable Professional
in LEARNING



Agenda

- What, Why, and How
- ASP.NET Core app → Docker container
- Kubernetes (K8s) in Azure
- CI / CD pipeline with Visual Studio Team Services

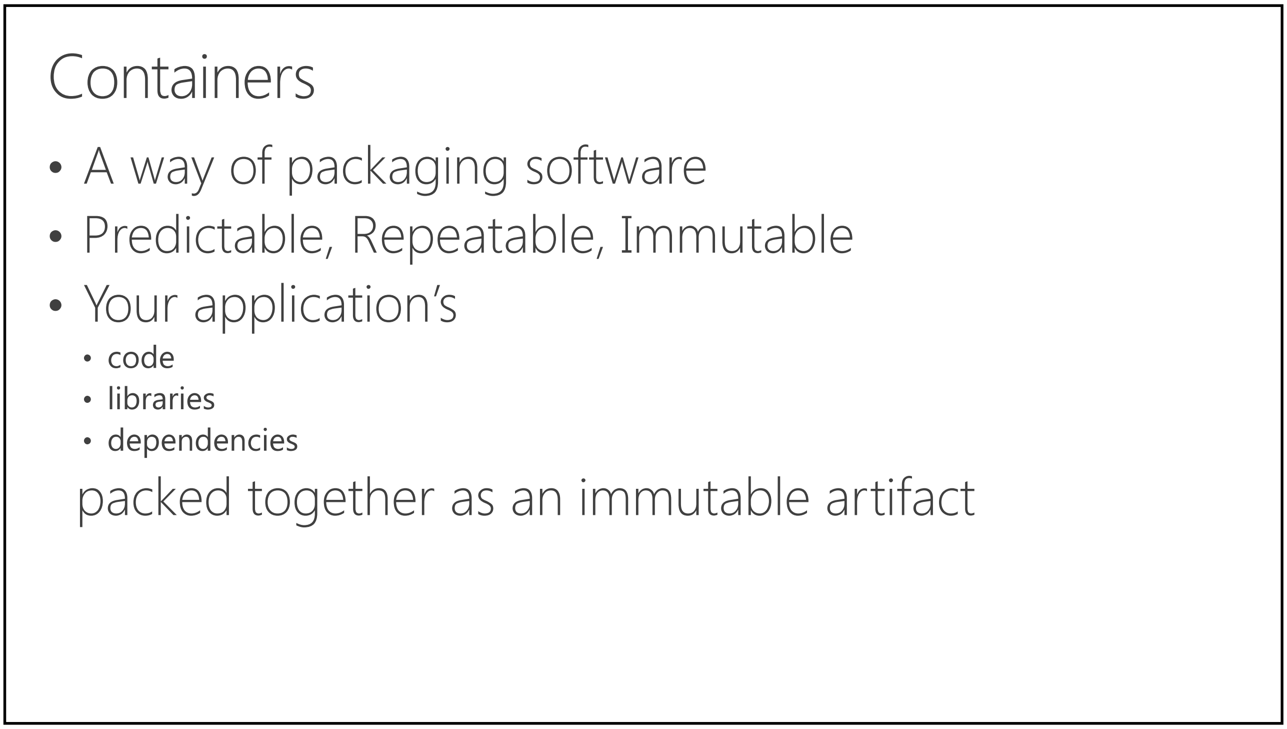


What does it look like?

What we use and why

.NET Core and ASP.NET Core

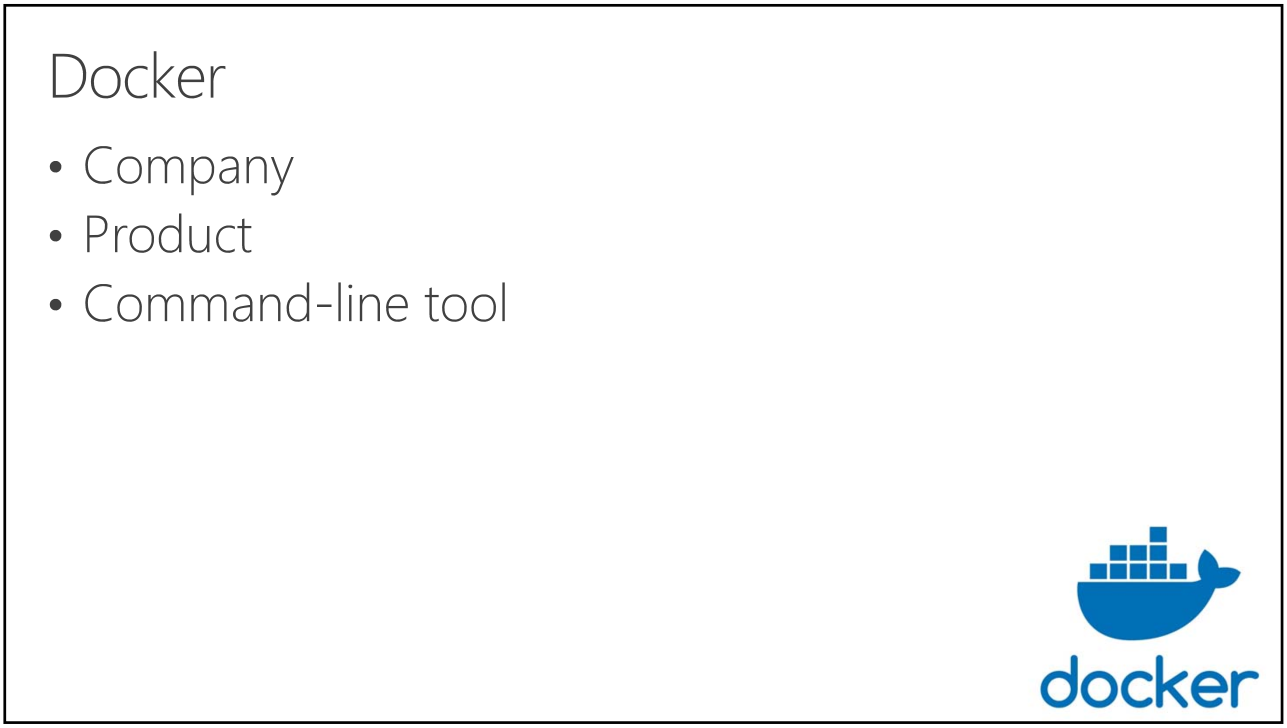
- Next generation managed runtime
- Cross-platform
- Container and micro-service optimized
- High performance



Containers

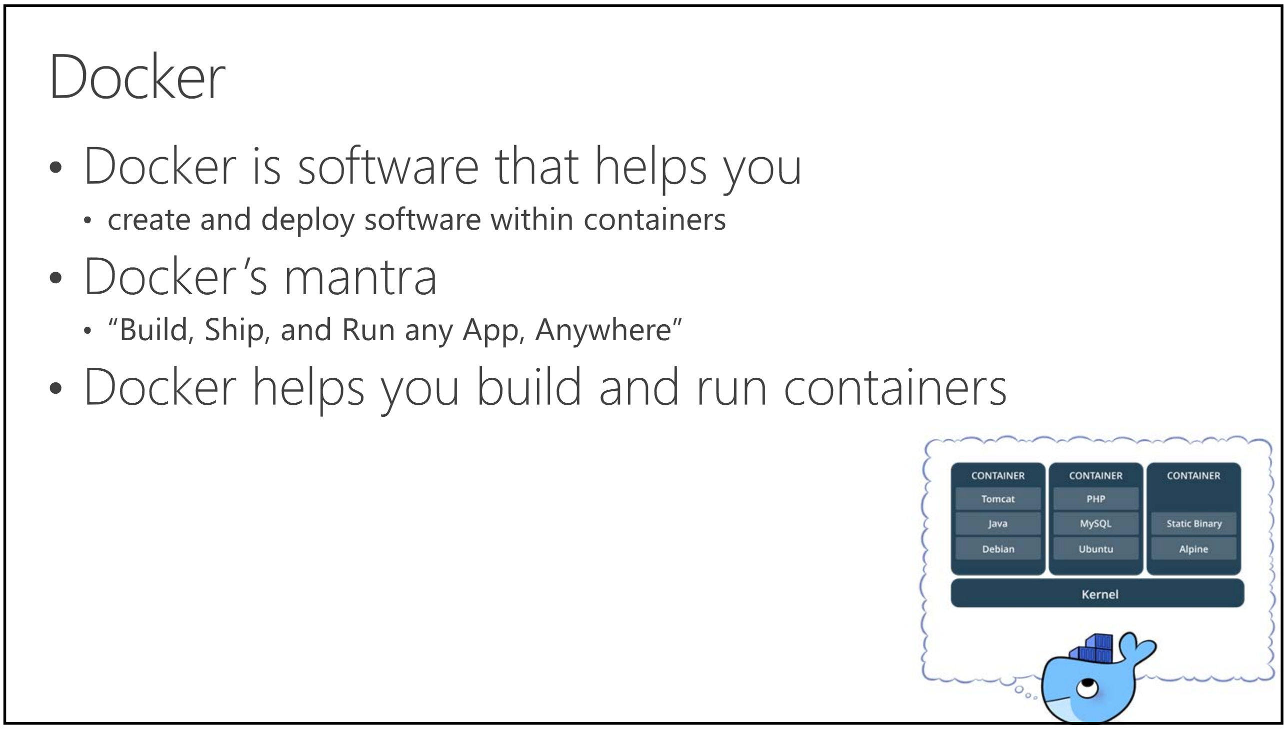
- A way of packaging software
- Predictable, Repeatable, Immutable
- Your application's
  - code
  - libraries
  - dependencies

packed together as an immutable artifact



Docker

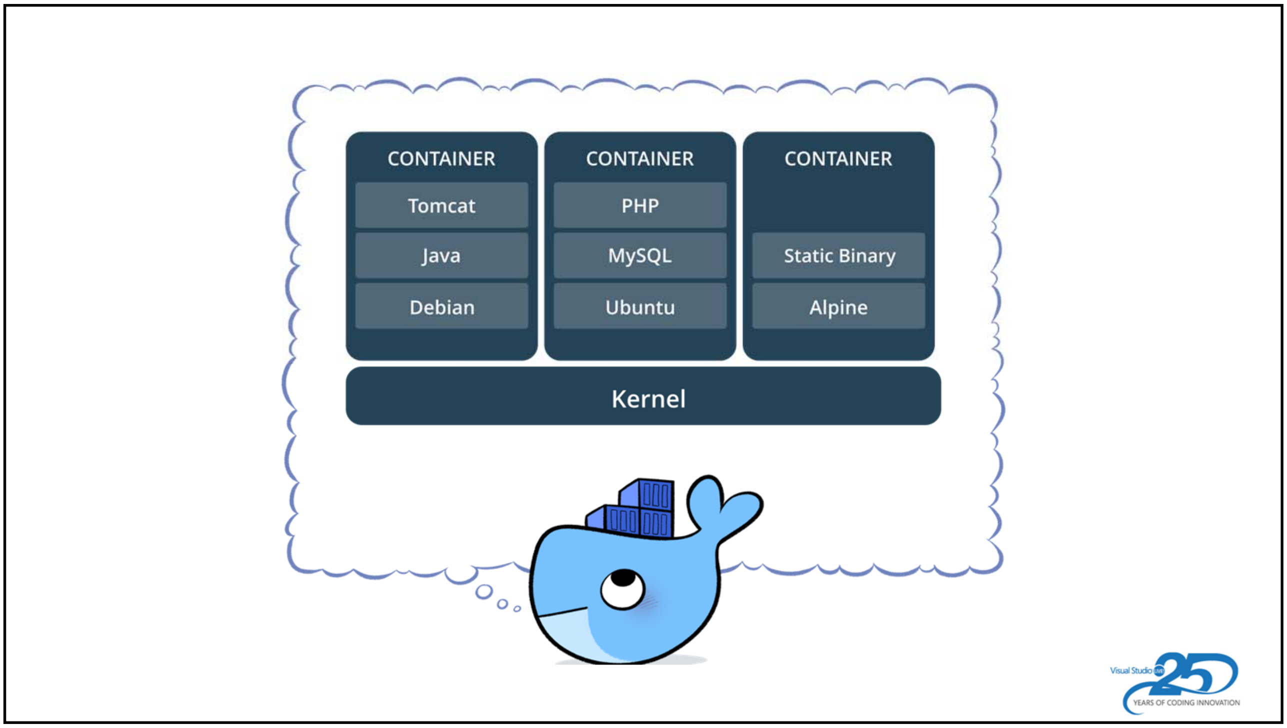
- Company
- Product
- Command-line tool



Docker

- Docker is software that helps you
  - create and deploy software within containers
- Docker’s mantra
  - “Build, Ship, and Run any App, Anywhere”
- Docker helps you build and run containers

CONTAINER
Tomcat
Java
Debian
CONTAINER
PHP
MySQL
Ubuntu
CONTAINER
Static Binary
Alpine
Kernel

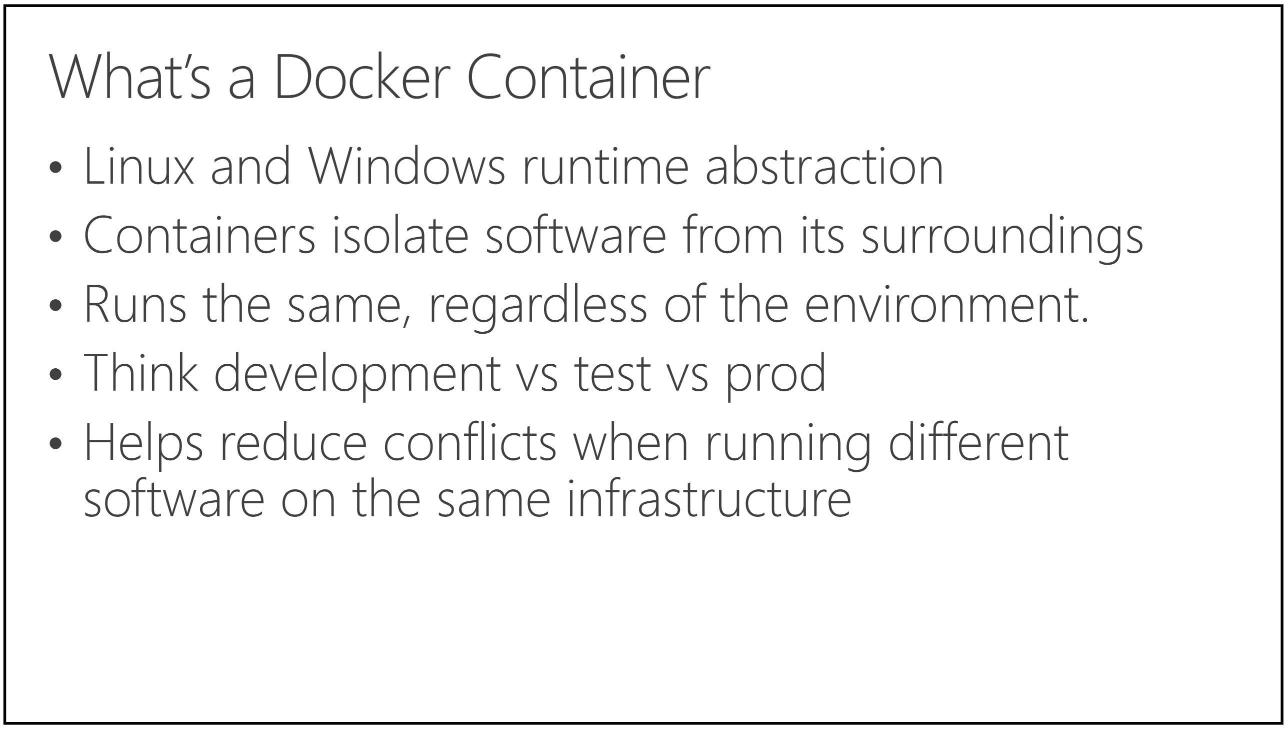


CONTAINER | Tomcat | Java | Debian

CONTAINER | PHP | MySQL | Ubuntu

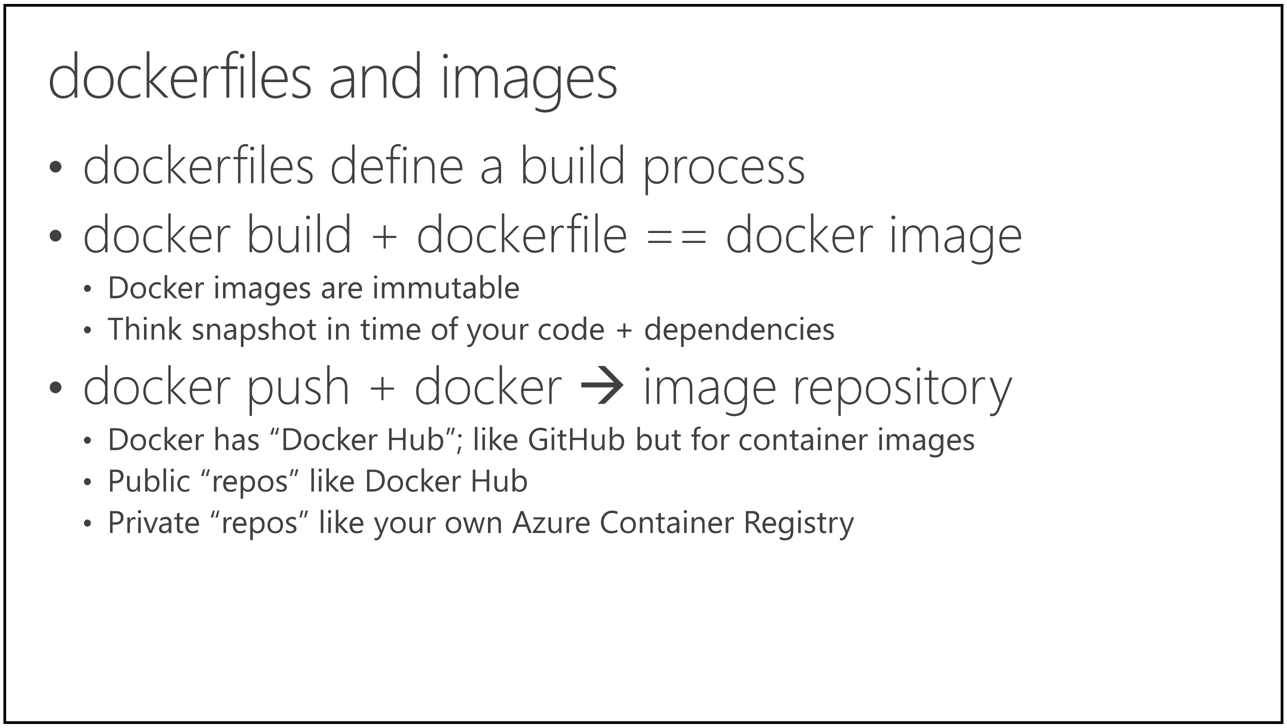
CONTAINER | Static Binary | Alpine

Kernel



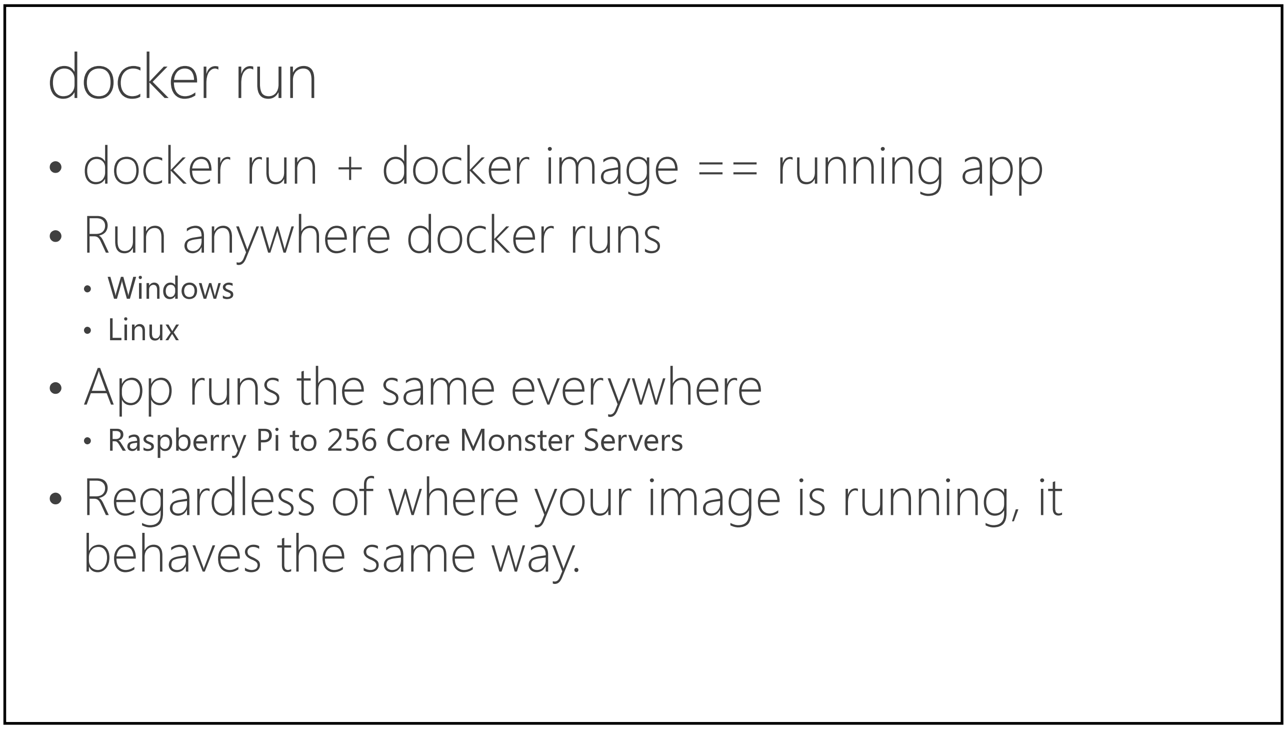
What’s a Docker Container

- Linux and Windows runtime abstraction
- Containers isolate software from its surroundings
- Runs the same, regardless of the environment.
- Think development vs test vs prod
- Helps reduce conflicts when running different software on the same infrastructure



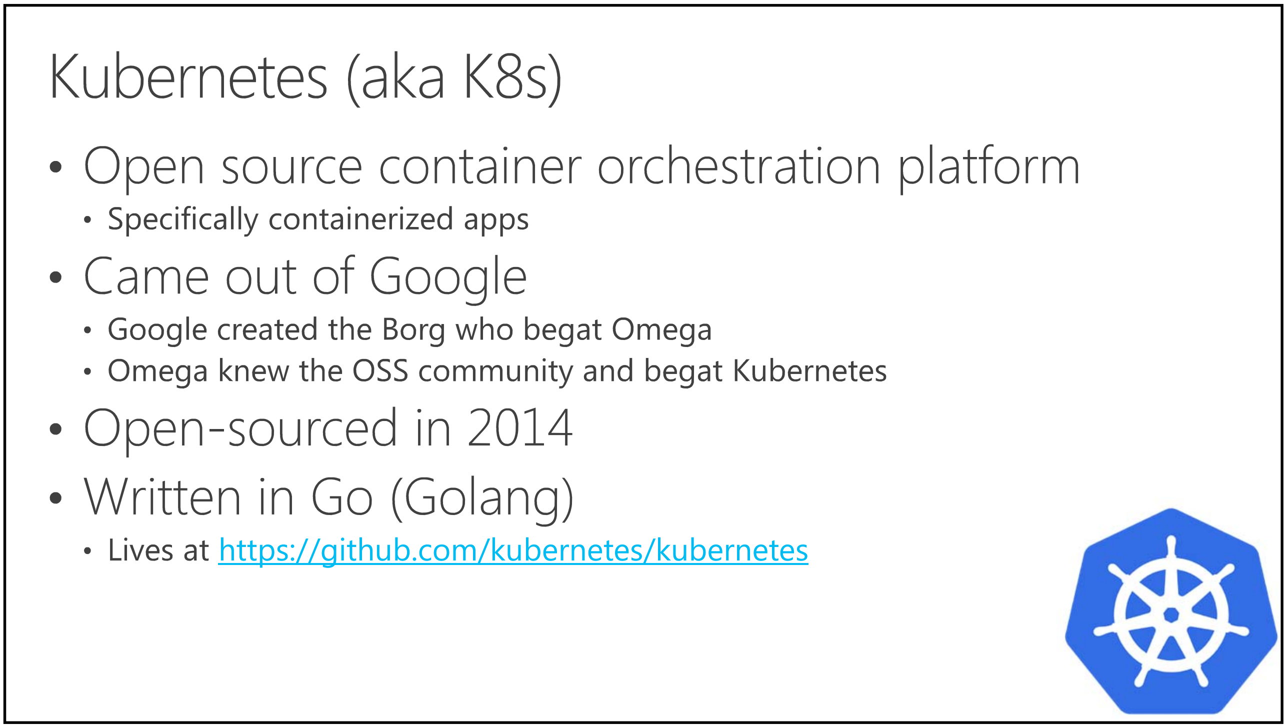
dockerfiles and images

- dockerfiles define a build process
- docker build + dockerfile == docker image
  - Docker images are immutable
  - Think snapshot in time of your code + dependencies
- docker push + docker → image repository
  - Docker has “Docker Hub”; like GitHub but for container images
  - Public “repos” like Docker Hub
  - Private “repos” like your own Azure Container Registry



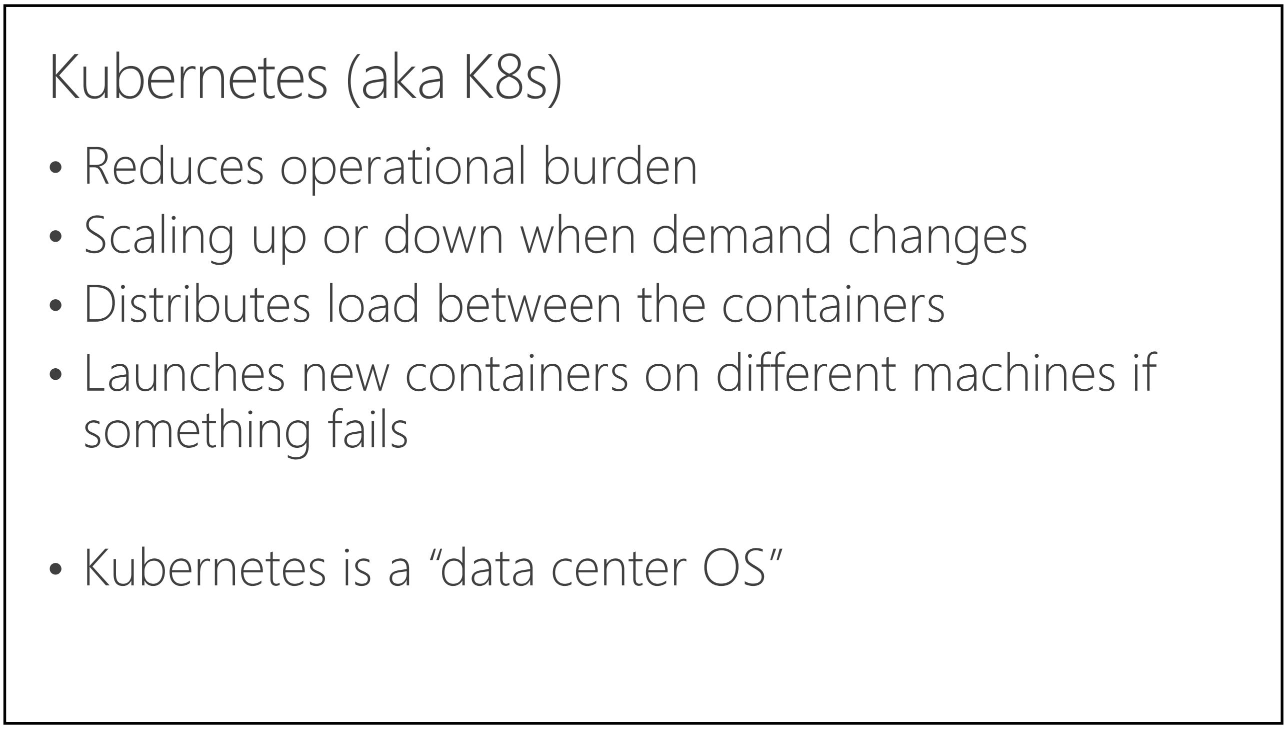
docker run

- docker run + docker image == running app
- Run anywhere docker runs
  - Windows
  - Linux
- App runs the same everywhere
  - Raspberry Pi to 256 Core Monster Servers
- Regardless of where your image is running, it behaves the same way.



Kubernetes (aka K8s)

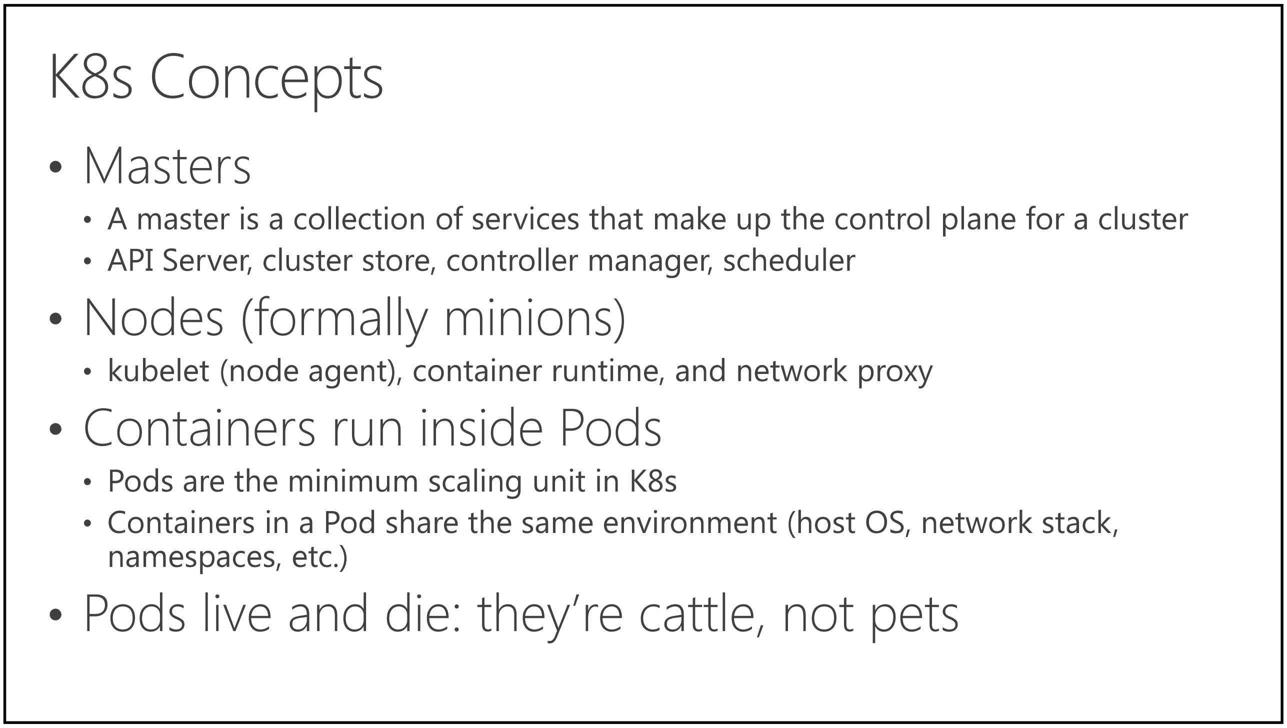
- Open source container orchestration platform
  - Specifically containerized apps
- Came out of Google
  - Google created the Borg who begat Omega
  - Omega knew the OSS community and begat Kubernetes
- Open-sourced in 2014
- Written in Go (Golang)
  - Lives at https://github.com/kubernetes/kubernetes



Kubernetes (aka K8s)

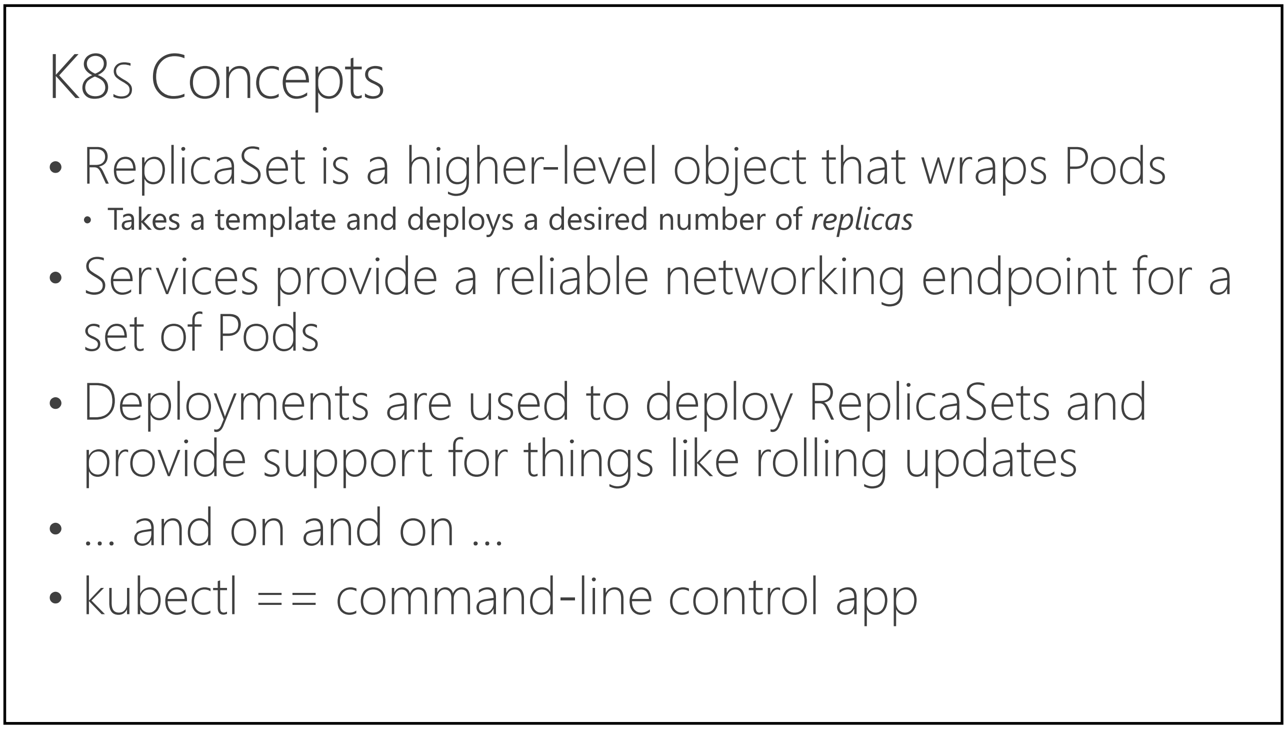
- Reduces operational burden
- Scaling up or down when demand changes
- Distributes load between the containers
- Launches new containers on different machines if something fails

- Kubernetes is a “data center OS”



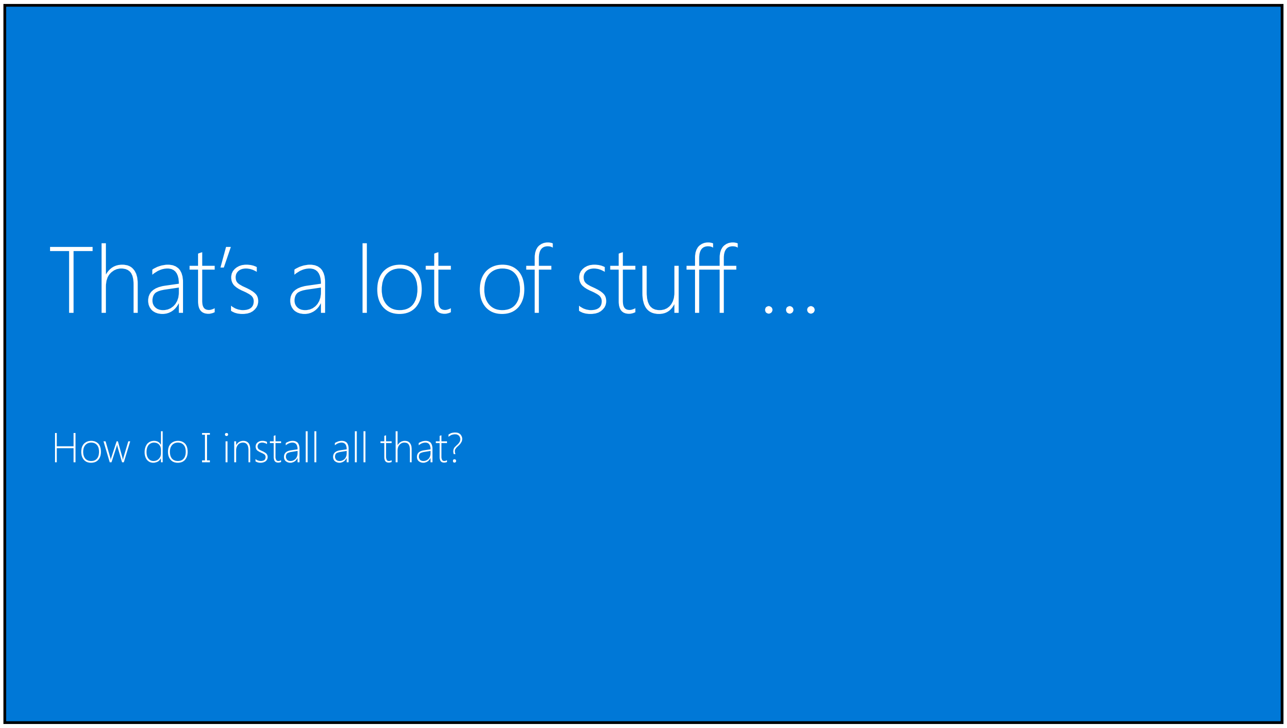
K8s Concepts

- Masters
  - A master is a collection of services that make up the control plane for a cluster
  - API Server, cluster store, controller manager, scheduler
- Nodes (formally minions)
  - kubelet (node agent), container runtime, and network proxy
- Containers run inside Pods
  - Pods are the minimum scaling unit in K8s
  - Containers in a Pod share the same environment (host OS, network stack, namespaces, etc.)
- Pods live and die: they’re cattle, not pets



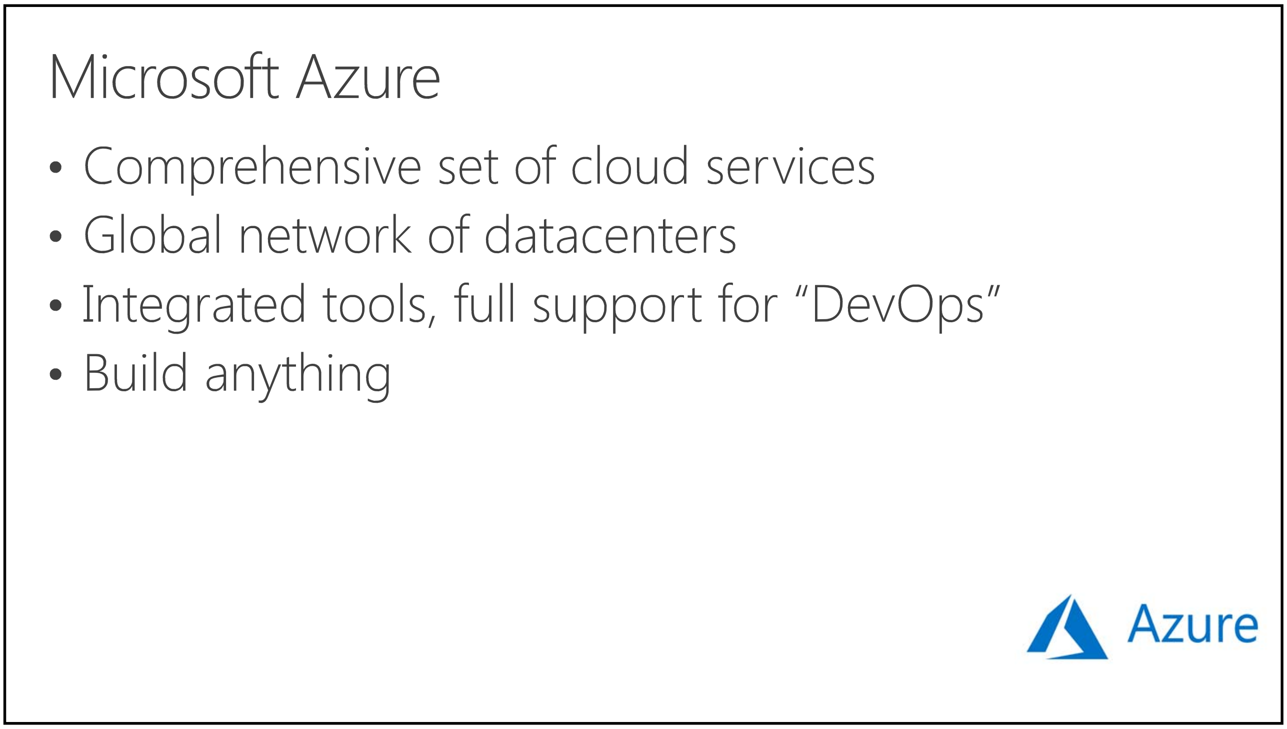
K8s Concepts

- ReplicaSet is a higher-level object that wraps Pods
  - Takes a template and deploys a desired number of *replicas*
- Services provide a reliable networking endpoint for a set of Pods
- Deployments are used to deploy ReplicaSets and provide support for things like rolling updates
- … and on and on …
- kubectl == command-line control app



That's a lot of stuff …

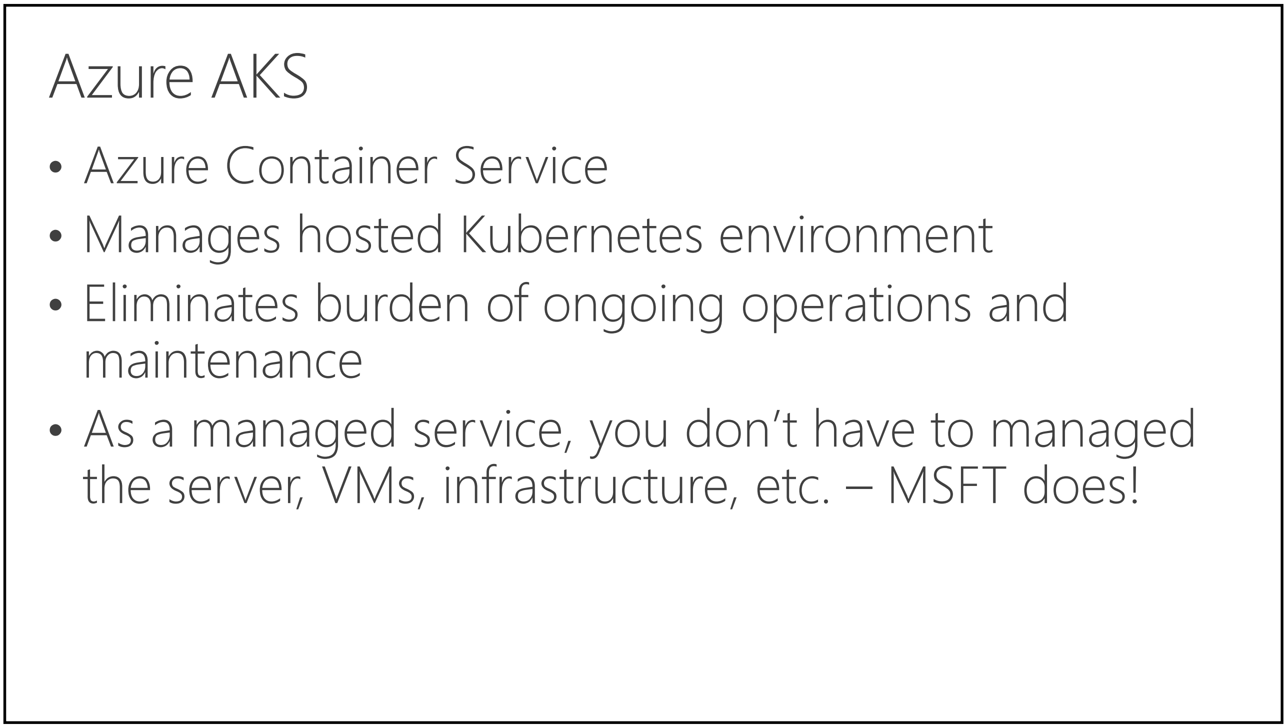
How do I install all that?



Microsoft Azure

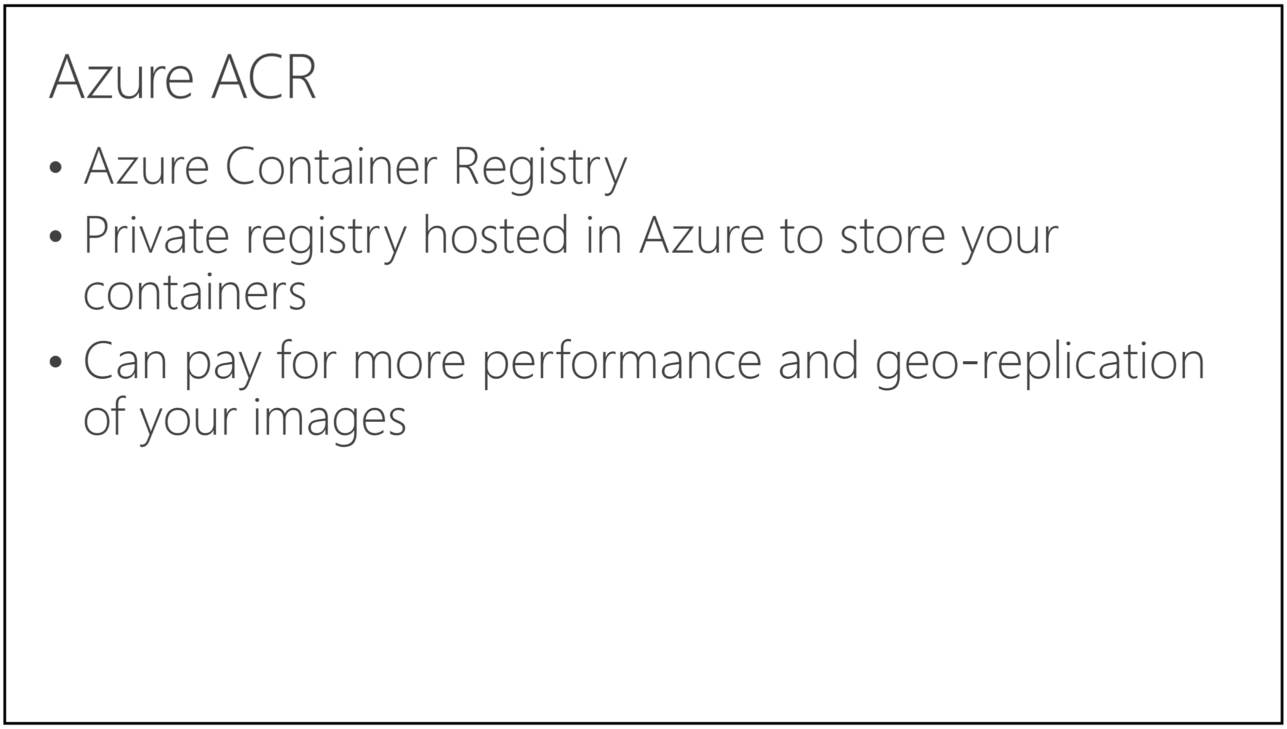
- Comprehensive set of cloud services
- Global network of datacenters
- Integrated tools, full support for “DevOps”
- Build anything

Azure



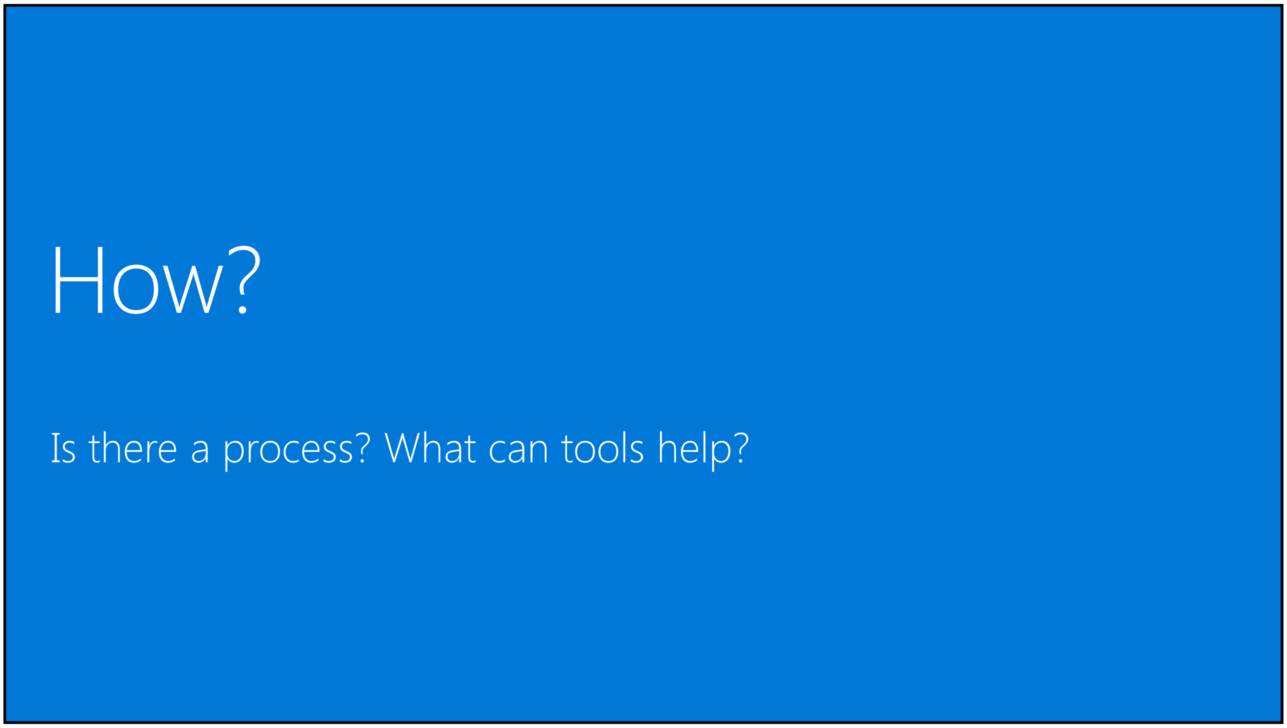
Azure AKS

- Azure Container Service
- Manages hosted Kubernetes environment
- Eliminates burden of ongoing operations and maintenance
- As a managed service, you don’t have to managed the server, VMs, infrastructure, etc. – MSFT does!



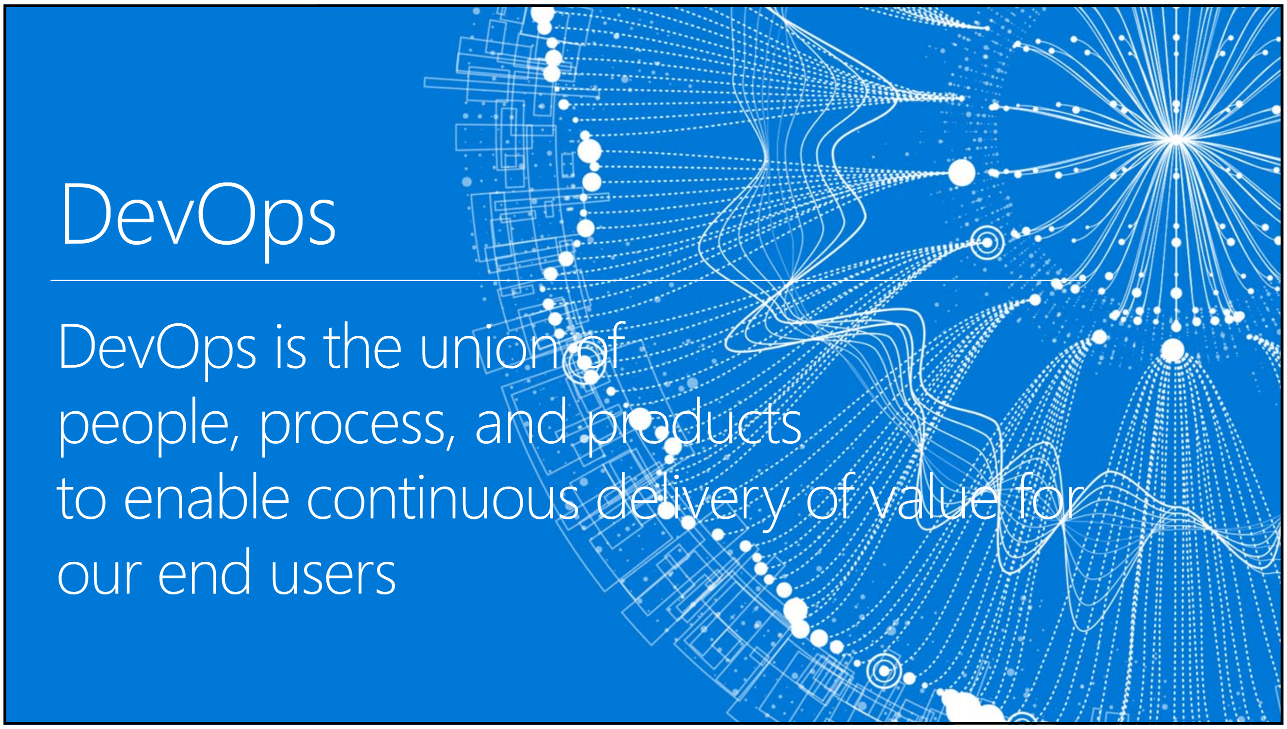
Azure ACR

- Azure Container Registry
- Private registry hosted in Azure to store your containers
- Can pay for more performance and geo-replication of your images



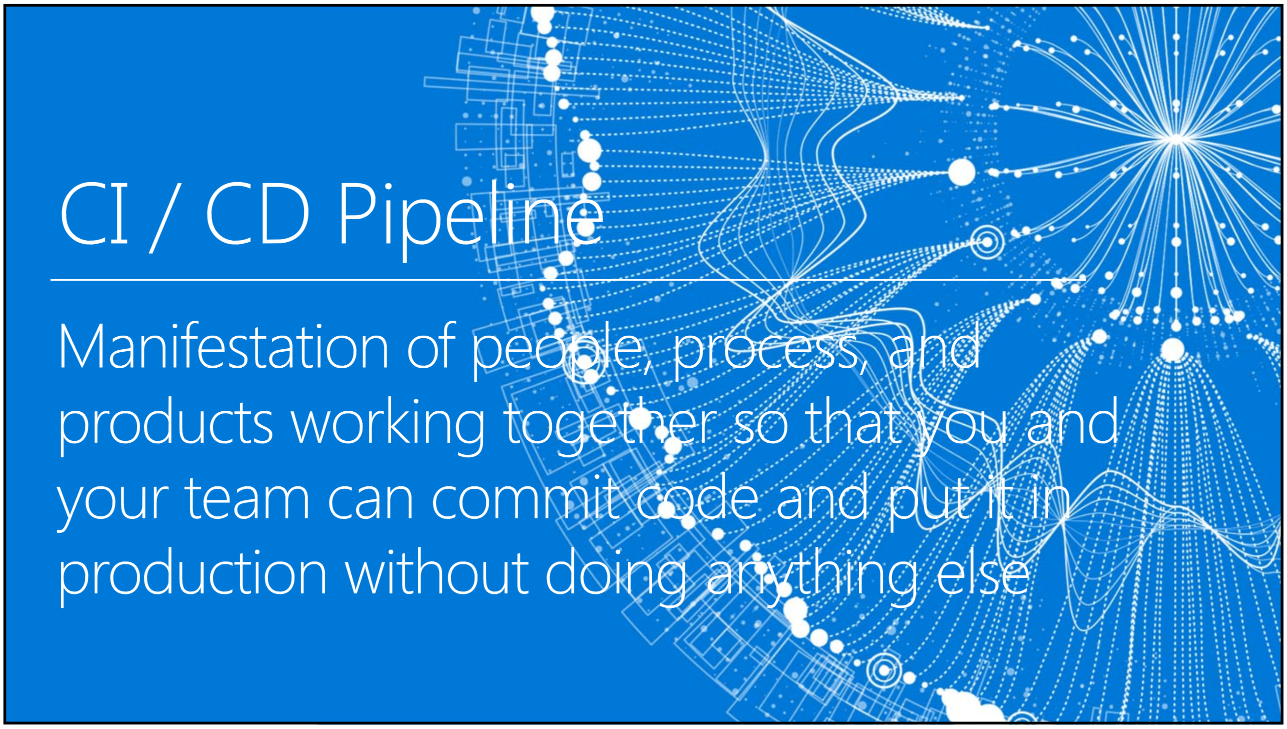
How?

Is there a process? What can tools help?



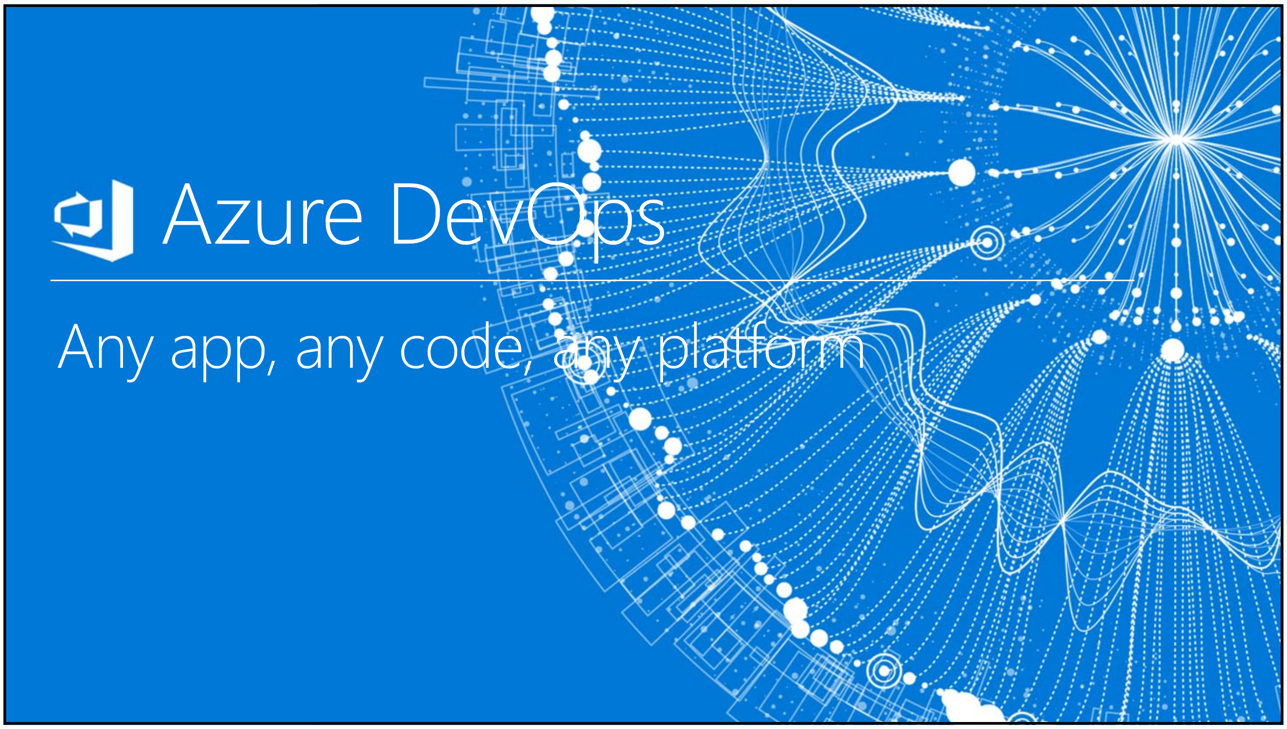
DevOps

DevOps is the union of
people, process, and products
to enable continuous delivery of value for
our end users



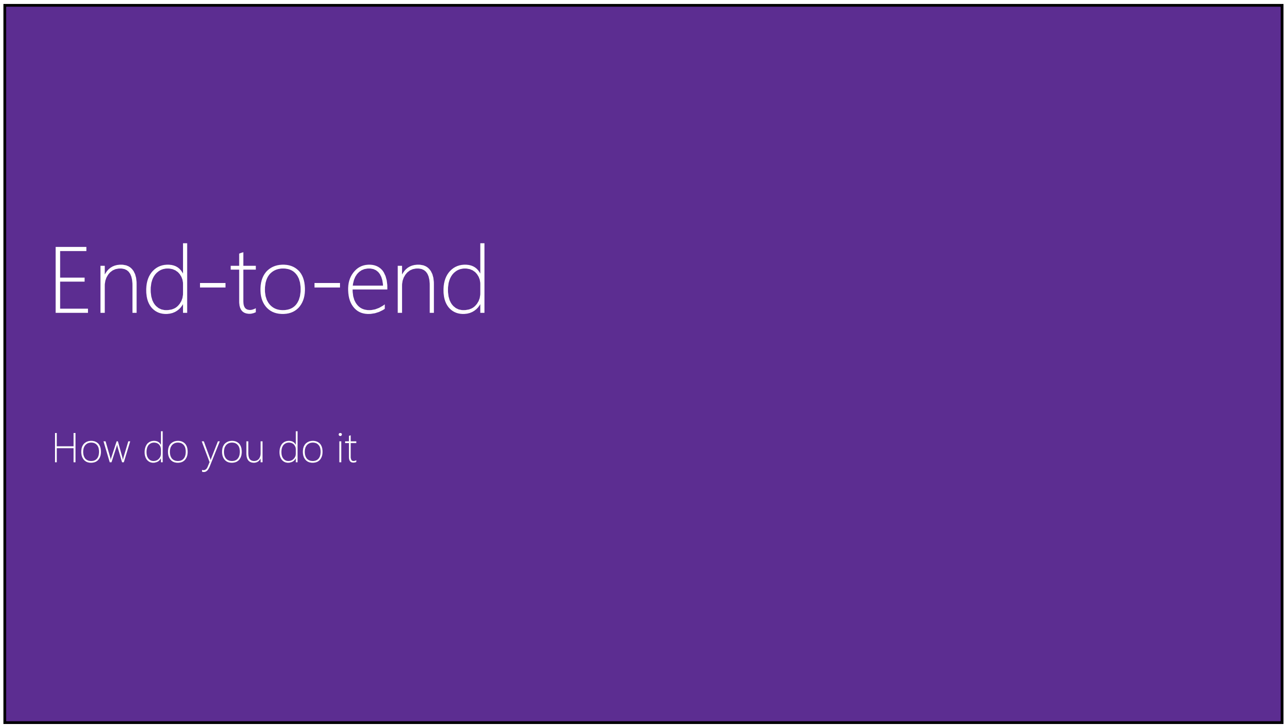
CI / CD Pipeline

Manifestation of people, process, and products working together so that you and your team can commit code and put it in production without doing anything else



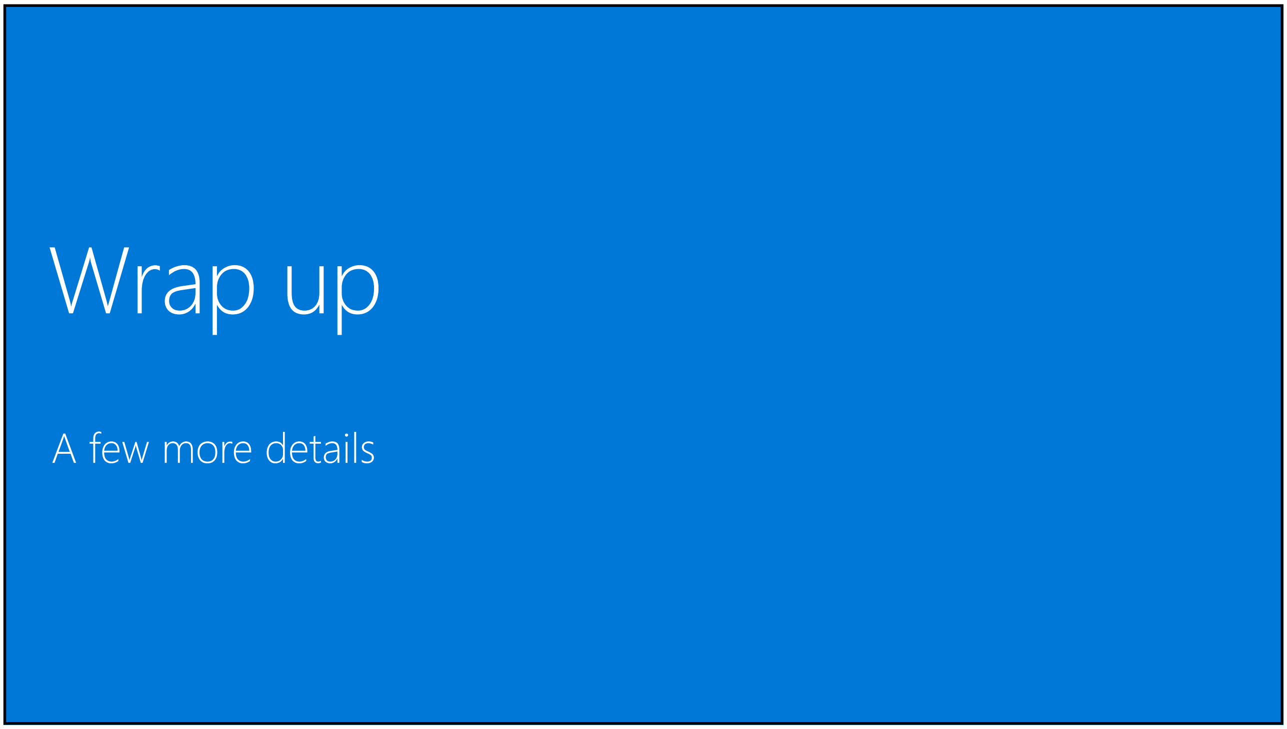
Azure DevOps

Any app, any code, any platform



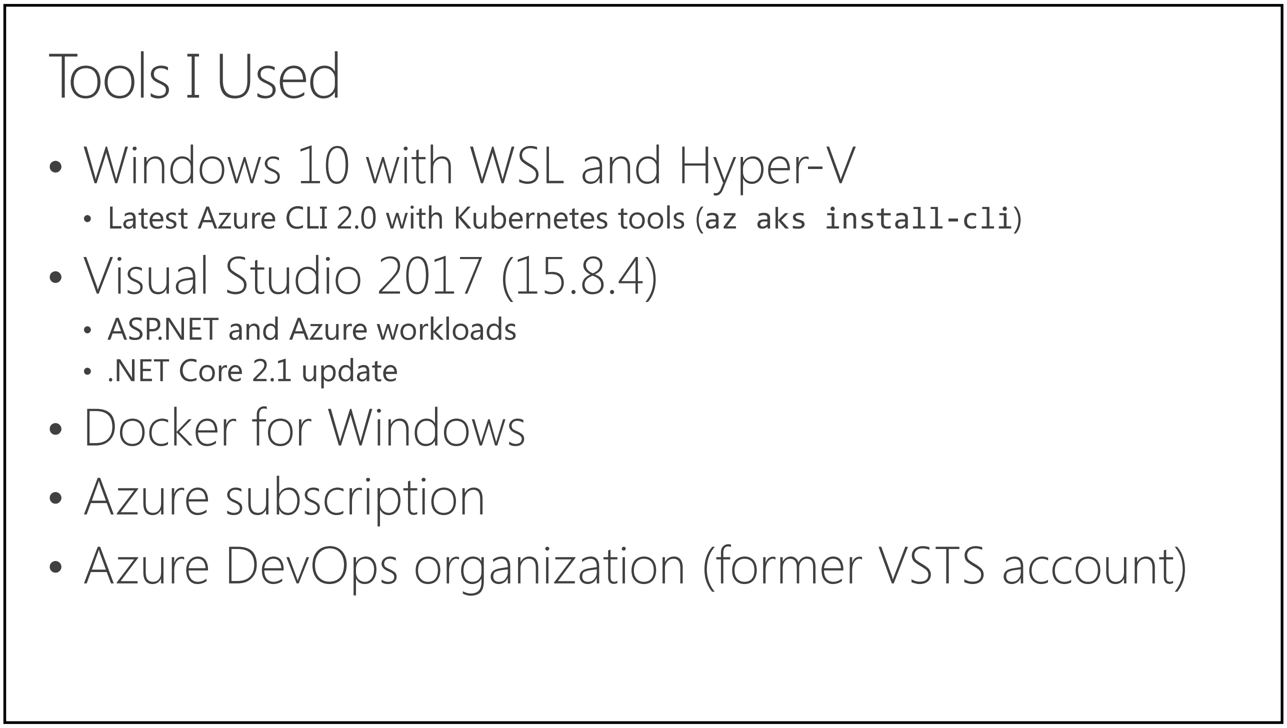
End-to-end

How do you do it



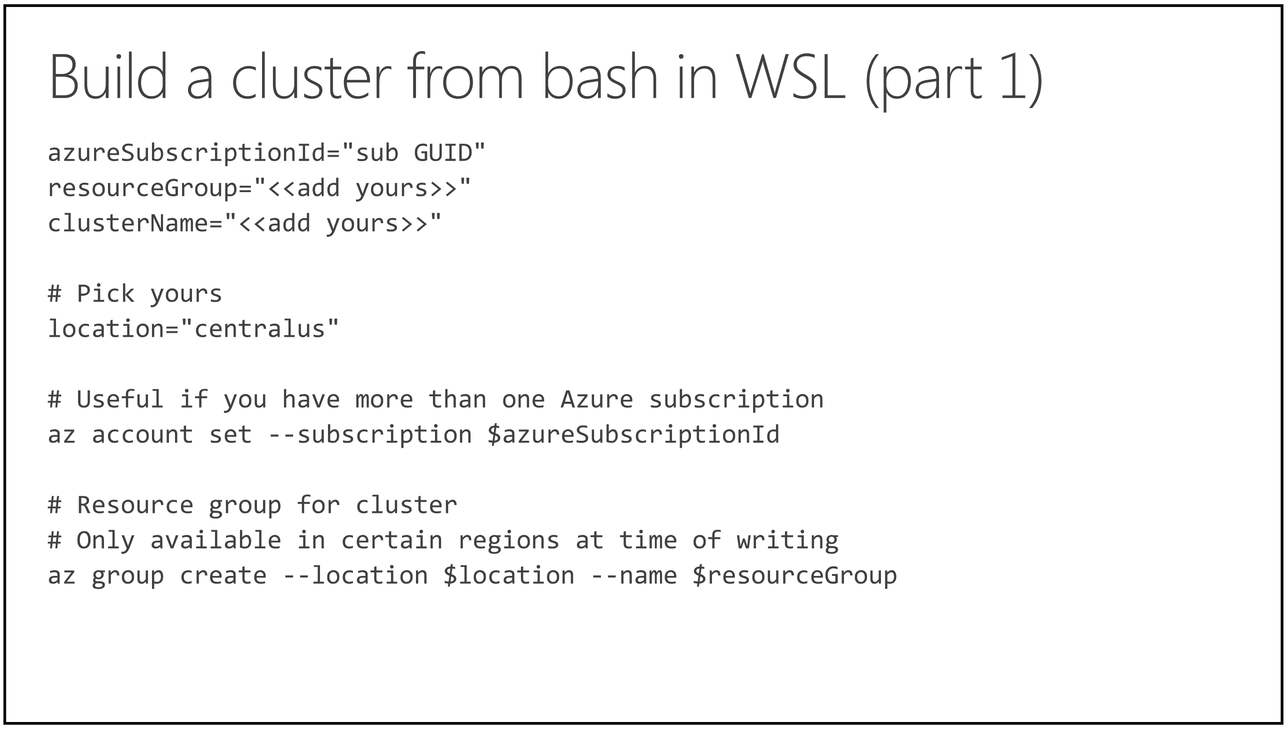
Wrap up

A few more details



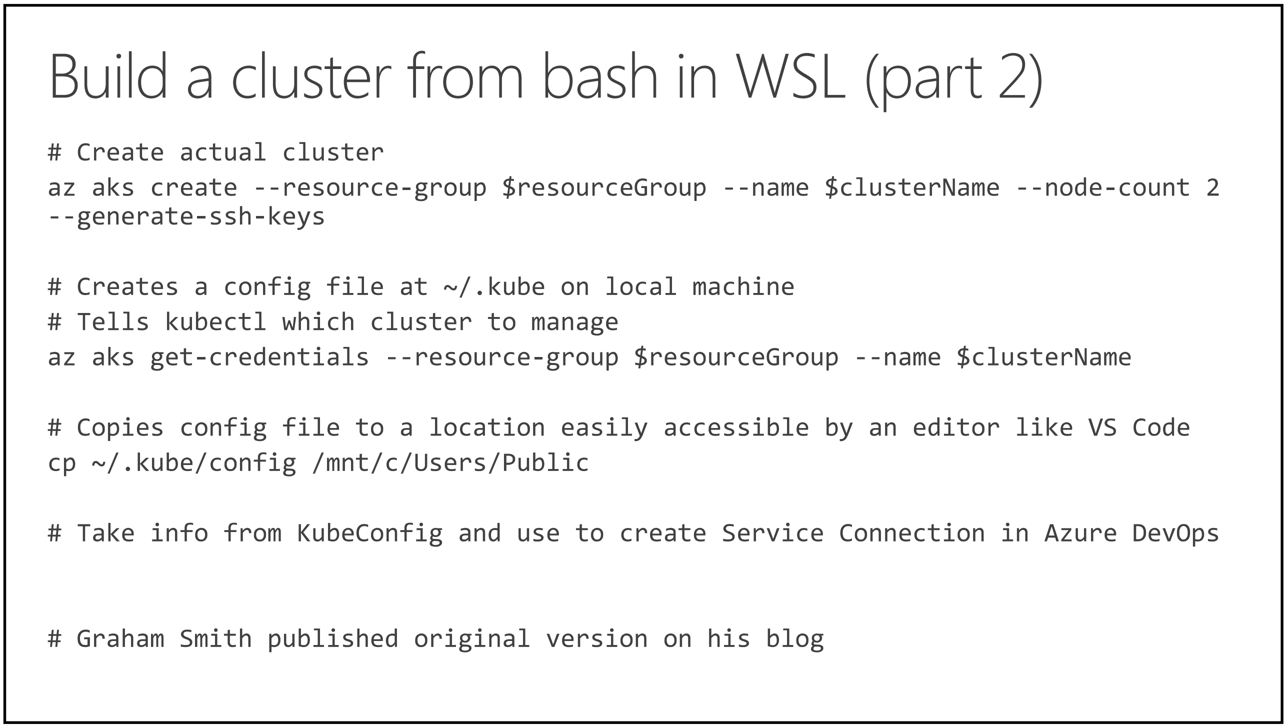
Tools I Used

- Windows 10 with WSL and Hyper-V
  - Latest Azure CLI 2.0 with Kubernetes tools (`az aks install-cli`)
- Visual Studio 2017 (15.8.4)
  - ASP.NET and Azure workloads
  - .NET Core 2.1 update
- Docker for Windows
- Azure subscription
- Azure DevOps organization (former VSTS account)



Build a cluster from bash in WSL (part 1)

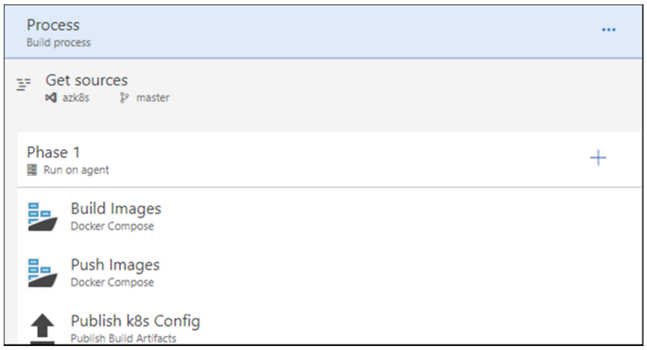
```
azureSubscriptionId="sub GUID"
resourceGroup="<<add yours>>"
clusterName="<<add yours>>"

# Pick yours
location="centralus"

# Useful if you have more than one Azure subscription
az account set --subscription $azureSubscriptionId

# Resource group for cluster
# Only available in certain regions at time of writing
az group create --location $location --name $resourceGroup
```



Build a cluster from bash in WSL (part 2)

```
# Create actual cluster
az aks create --resource-group $resourceGroup --name $clusterName --node-count 2
--generate-ssh-keys

# Creates a config file at ~/.kube on local machine
# Tells kubectl which cluster to manage
az aks get-credentials --resource-group $resourceGroup --name $clusterName

# Copies config file to a location easily accessible by an editor like VS Code
cp ~/.kube/config /mnt/c/Users/Public

# Take info from KubeConfig and use to create Service Connection in Azure DevOps


# Graham Smith published original version on his blog
```

Team Build



Process
Build process

Get sources
azk8s master

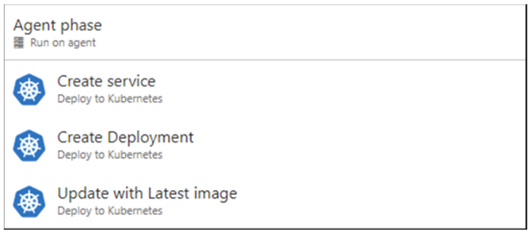
Phase 1
Run on agent

Build Images
Docker Compose

Push Images
Docker Compose

Publish k8s Config
Publish Build Artifacts

Release Management

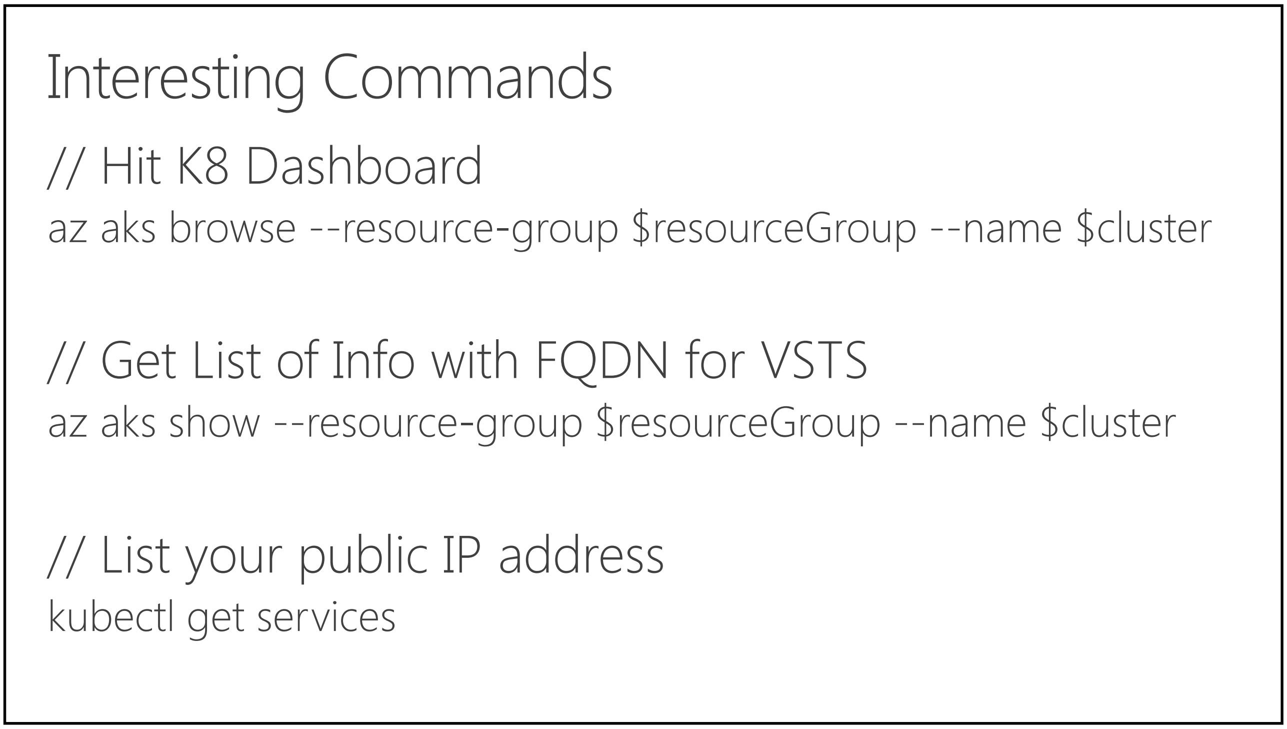


Agent phase
Run on agent

Create service
Deploy to Kubernetes

Create Deployment
Deploy to Kubernetes

Update with Latest image
Deploy to Kubernetes



Interesting Commands

// Hit K8 Dashboard

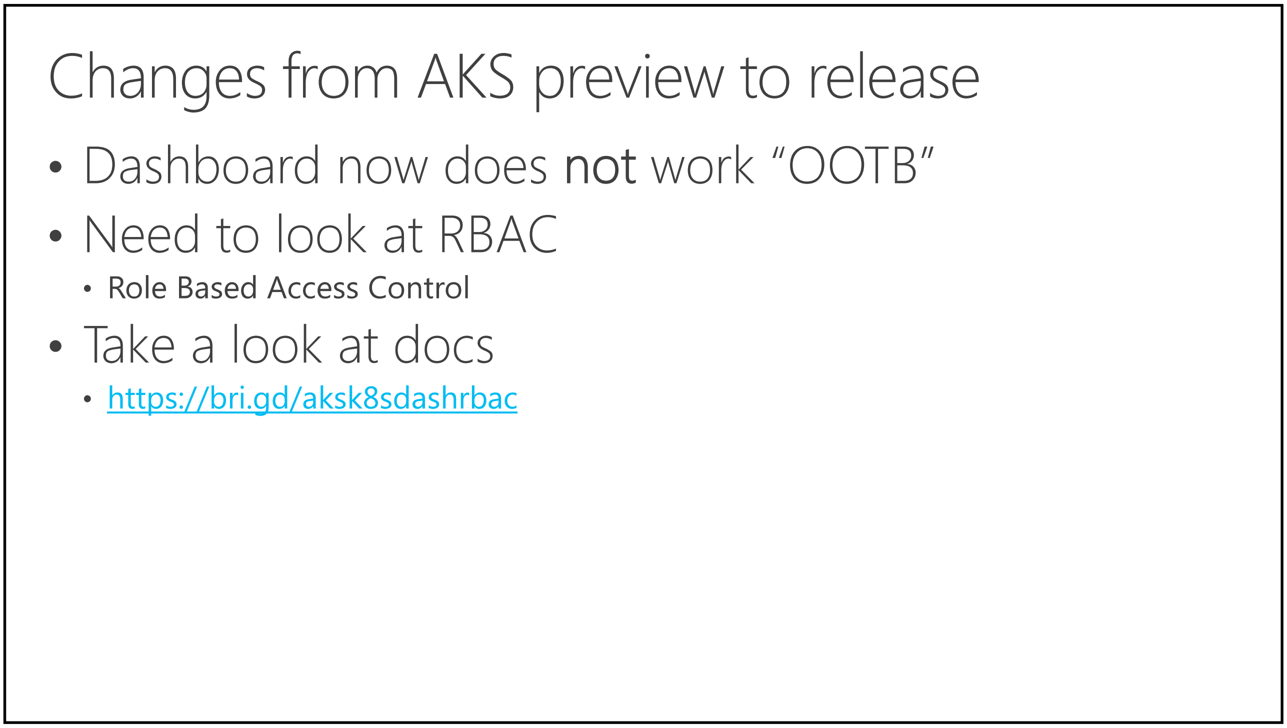
```
az aks browse --resource-group $resourceGroup --name $cluster
```

// Get List of Info with FQDN for VSTS

```
az aks show --resource-group $resourceGroup --name $cluster
```

// List your public IP address

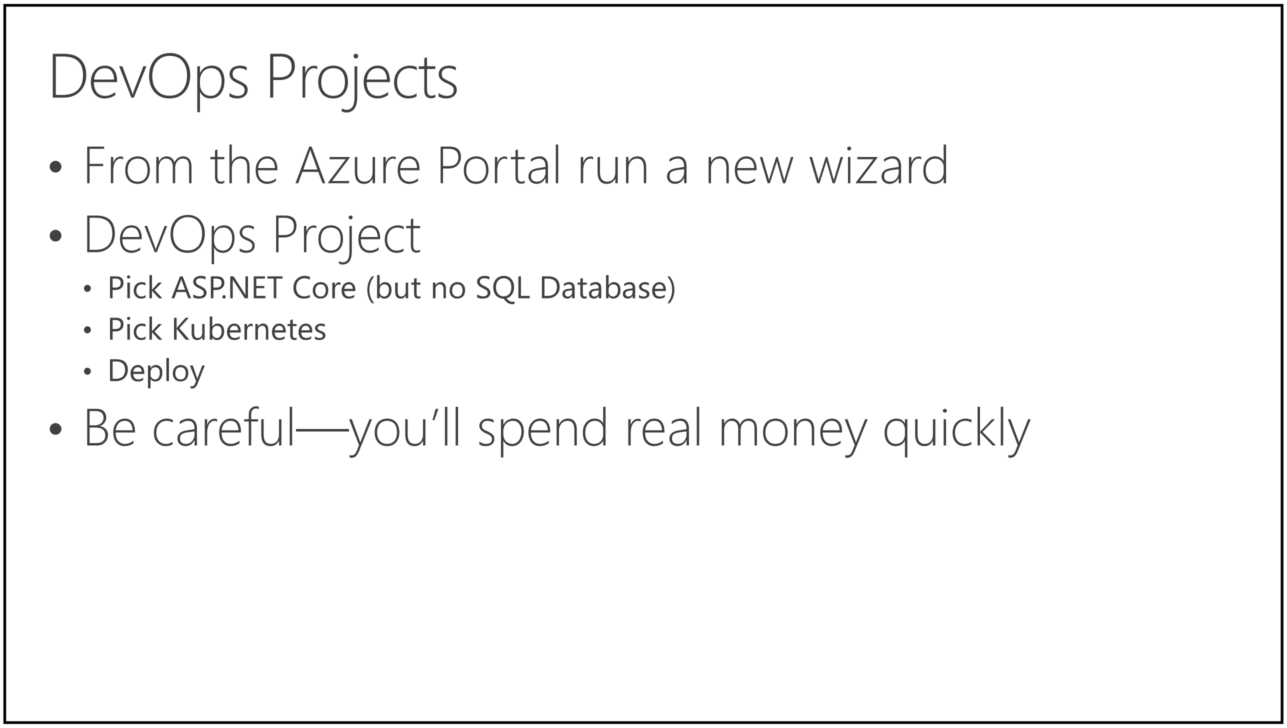
```
kubectl get services
```



Changes from AKS preview to release

- Dashboard now does **not** work “OOTB”
- Need to look at RBAC
  - Role Based Access Control
- Take a look at docs
  - https://bri.gd/aksk8sdashrbac

One more thing …

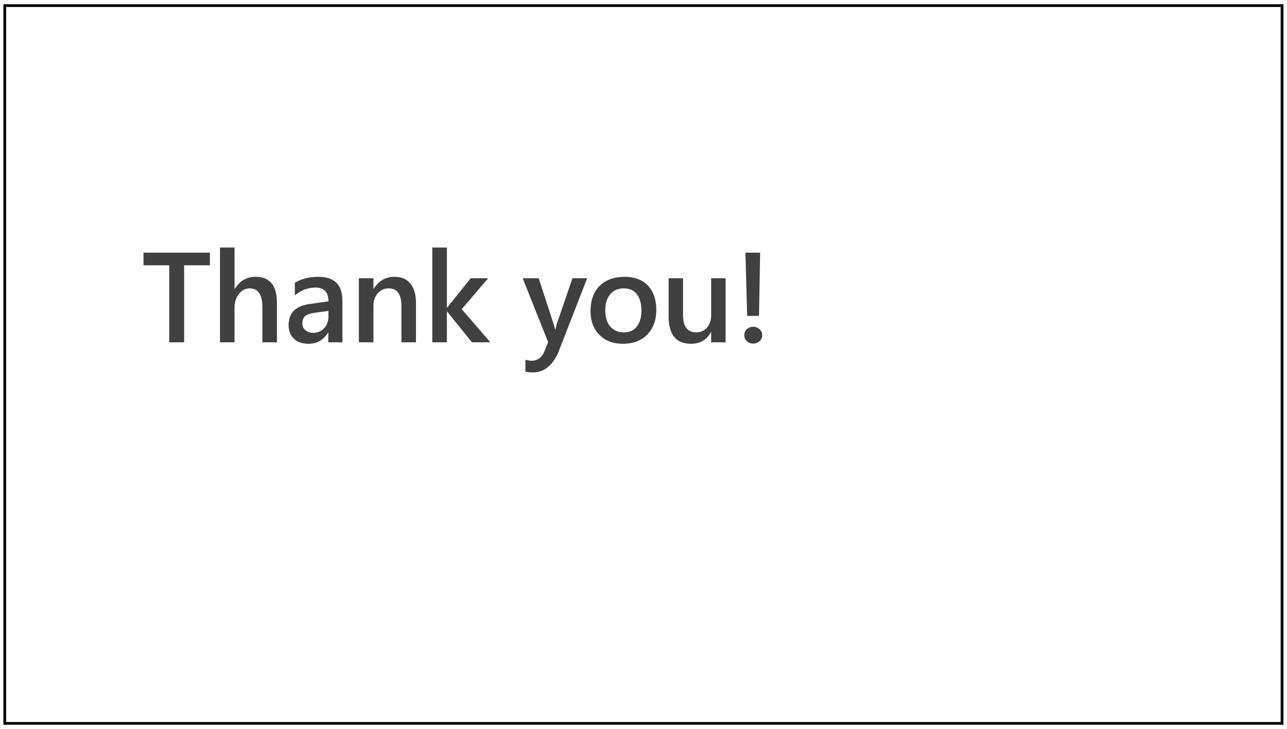


DevOps Projects

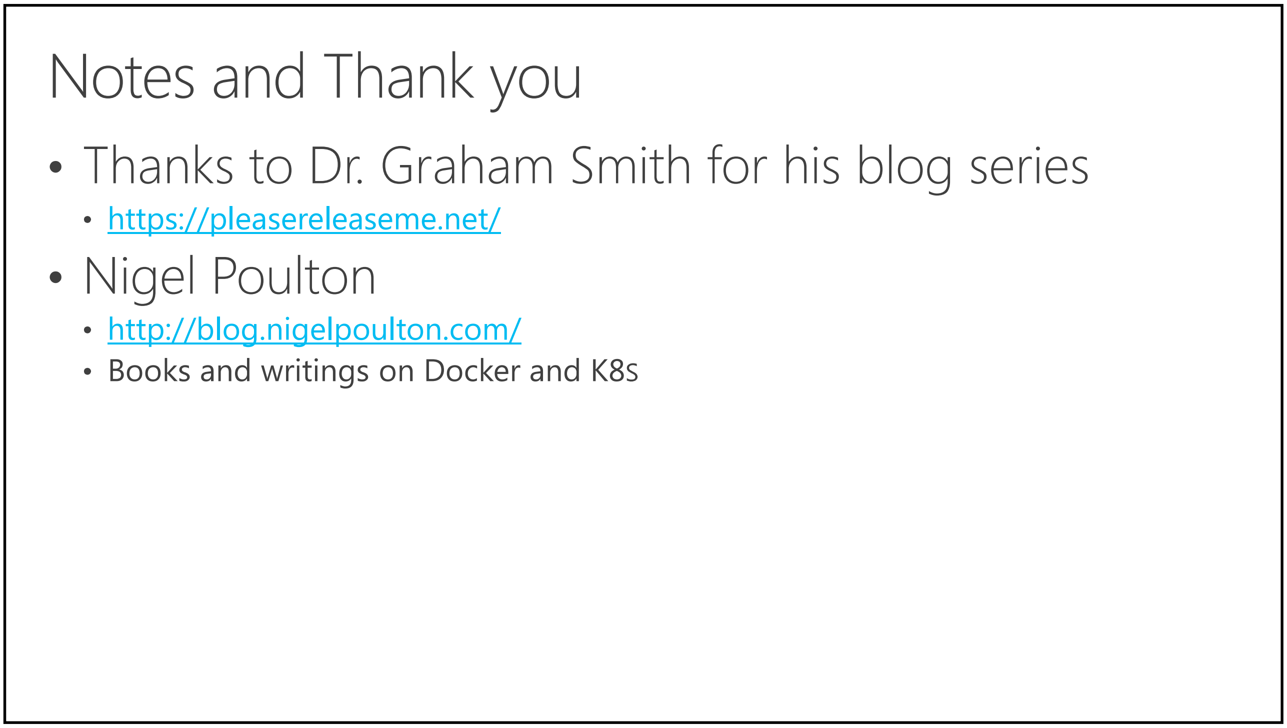
- From the Azure Portal run a new wizard
- DevOps Project
  - Pick ASP.NET Core (but no SQL Database)
  - Pick Kubernetes
  - Deploy
- Be careful—you'll spend real money quickly

A final thought …

Don’t let your **experience**
be an **impediment** to your ability
to **learn** and **grow**.

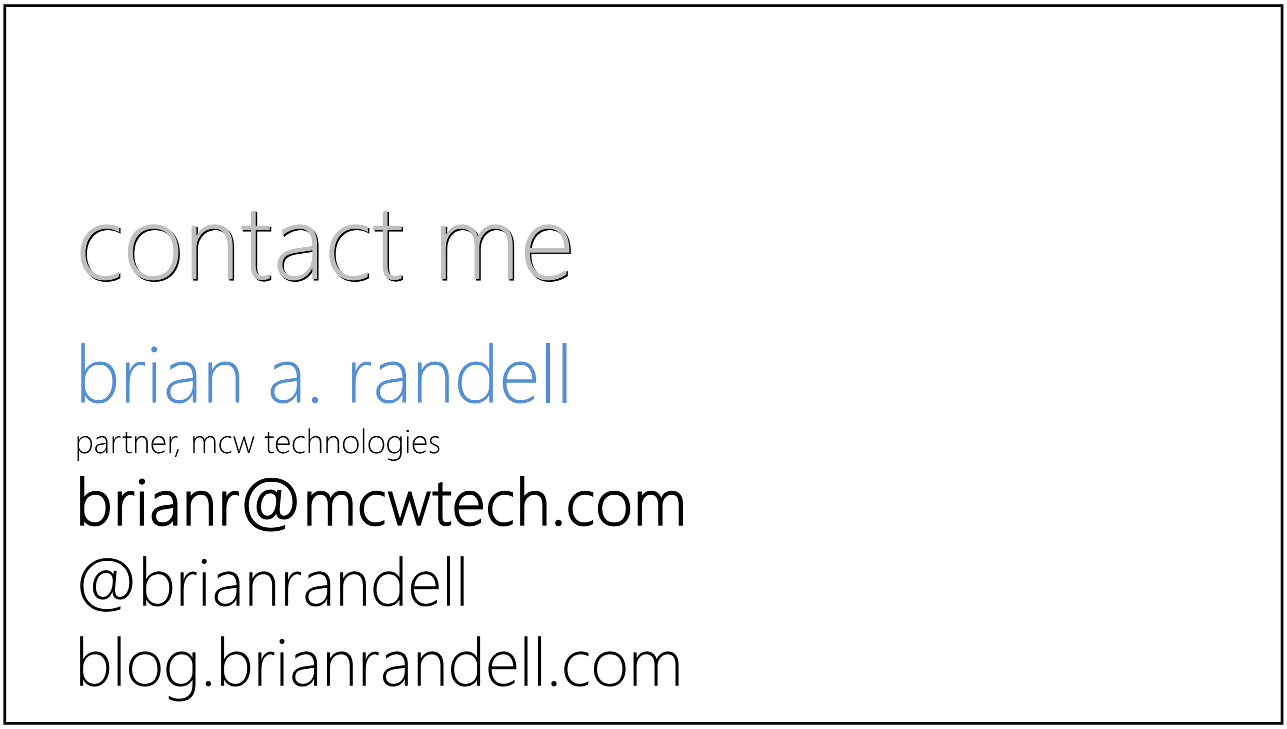


Thank you!



Notes and Thank you

- Thanks to Dr. Graham Smith for his blog series
  - https://pleasereleaseme.net/
- Nigel Poulton
  - http://blog.nigelpoulton.com/
  - Books and writings on Docker and K8s



contact me

brian a. randell

partner, mcw technologies

brianr@mcwtech.com

@brianrandell

blog.brianrandell.com