



DevOps with Kubernetes and Helm

Jessica Deen
Cloud Developer Advocate

HELLO!

I am Jessica Deen

I am here because I love technology and community.

I focus heavily on Linux, OSS, DevOps and Containers.

I love Disney and CrossFit/Fitness.

You can find me at @jldeen on GitHub, Twitter, and Instagram.



Disclaimer

The next 60 minutes will NOT make you an expert, but it will:

- Get you thinking
- Show you what's possible
- Give you some sample code for you to get started on your own time

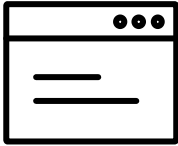
GOING DIGITAL

1 million/hour
new devices
coming online
by 2020

12 years
average age of S&P
500 corporations
by 2020

60% computing
in the public cloud
by 2025

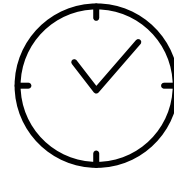
What we hear from **developers**



I need to create applications
at a competitive rate without
worrying about IT



New applications run smoothly
on my machine but malfunction
on traditional IT servers



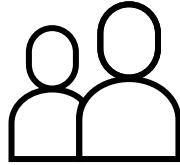
My productivity and application
innovation become suspended
when I have to wait on IT



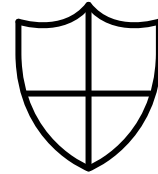
What we hear from IT



I need to manage servers and maintain compliance with little disruption



I'm unsure of how to integrate unfamiliar applications, and I require help from developers



I'm unable to focus on both server protection and application compliance



IT stress points

Security
threats



Datacenter
efficiency



Supporting
innovation



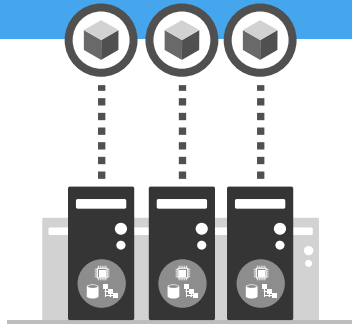
Cloud is a new way to think about a datacenter

Traditional model

Dedicated infrastructure for each application
Purpose-built hardware
Distinct infrastructure and operations teams
Customized processes and configurations

Cloud model

Loosely coupled apps and micro-services
Industry-standard hardware
Service-focused DevOps teams
Standardized processes and configurations



Servers



Services

DevOps: The Three Stage Conversation

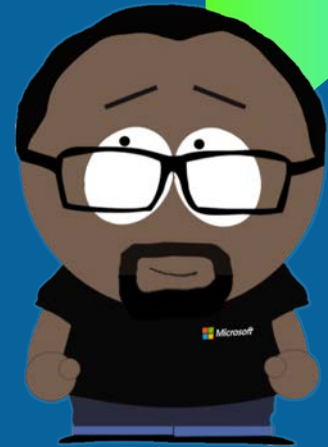
People

Process

Products

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

-Donovan Brown



Key DevOps Practices

Infrastructure as Code	Continuous Integration	Continuous Deployment
Automated Testing	Release Management	Performance Monitoring
Availability Monitoring	Load Testing & Auto Scale	Automated Recovery (Rollback & Roll Forward)

DevOps Benefits

IT Performance Metrics

	2015	2016	2017
Deployment Frequency	30x more frequent	200x more frequent	46x more frequent
Lead Time for Changes	200x faster	2,555x faster	440x faster
Mean Time to Recover (MTTR)	168x faster	24x faster	96x faster
Change Failure Rate		3x lower (1/3 as likely)	5x lower (1/5 as likely)

Why Containers?



Developers

- Enable 'write-once, run-anywhere' apps
- Enables microservice architectures
- Great for dev/test of apps and services
- Production realism
- Growing Developer Community



Operations

- Portability, Portability, Portability
- Standardized development, QA, and prod environments
- Abstract differences in OS distributions and underlying infrastructure
- Higher compute density
- Easily scale-up and scale-down in response to changing business needs

DevOps



What is a Container?

Not a real thing. An application delivery mechanism with **process isolation** based on several **Linux kernel** features.

Namespaces (what a process can see)

- ❖ PID
- ❖ Mount
- ❖ Network
- ❖ UTS
- ❖ IPC
- ❖ User
- ❖ Cgroup

Cgroups (what a process can use)

- ❖ Memory
- ❖ CPU
- ❖ Blkio
- ❖ Cpuacct
- ❖ Cpuset
- ❖ Devices
- ❖ Net_prio

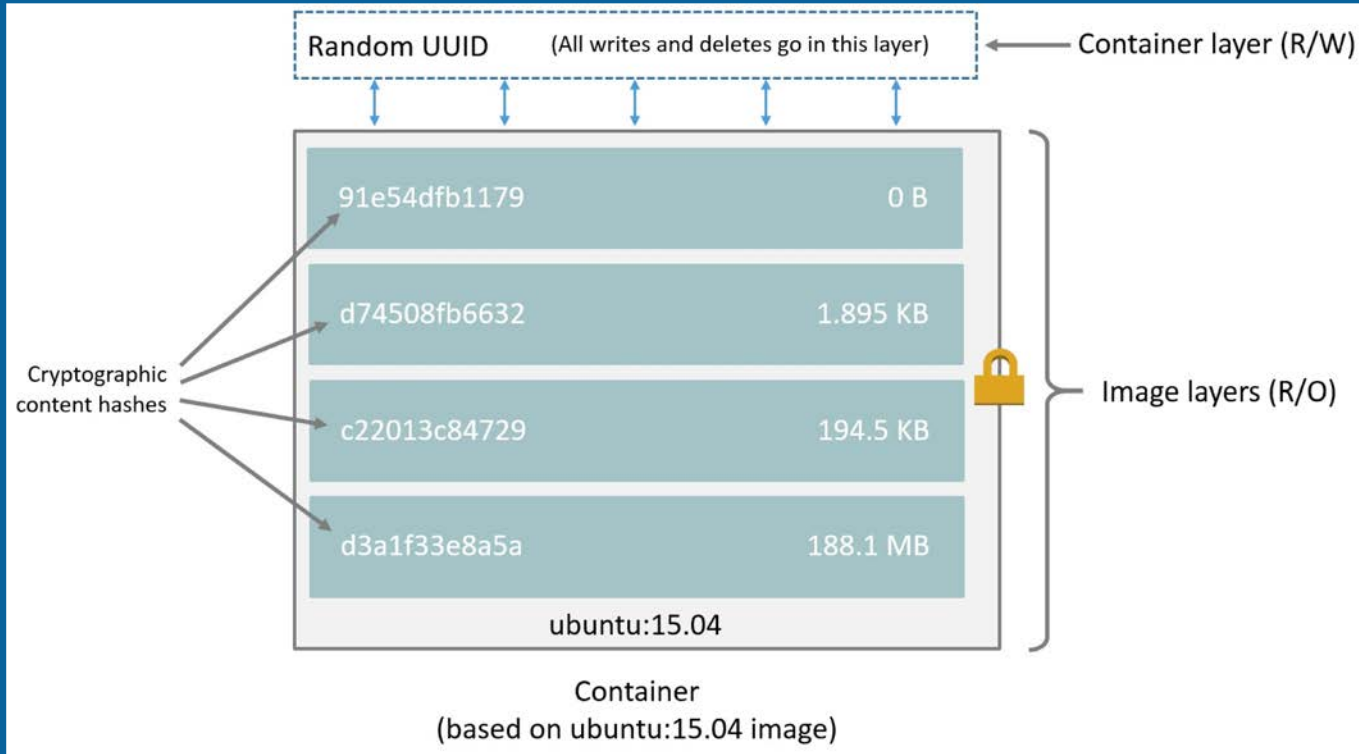
What is



docker

- ❖ **Open Source Container Runtime**
- ❖ **Mac, Linux, Windows Support**
- ❖ **Command Line Tool**
- ❖ **“Dockerfile” format**
- ❖ **The Docker image format with layered filesystem**

Docker Layered Filesystem

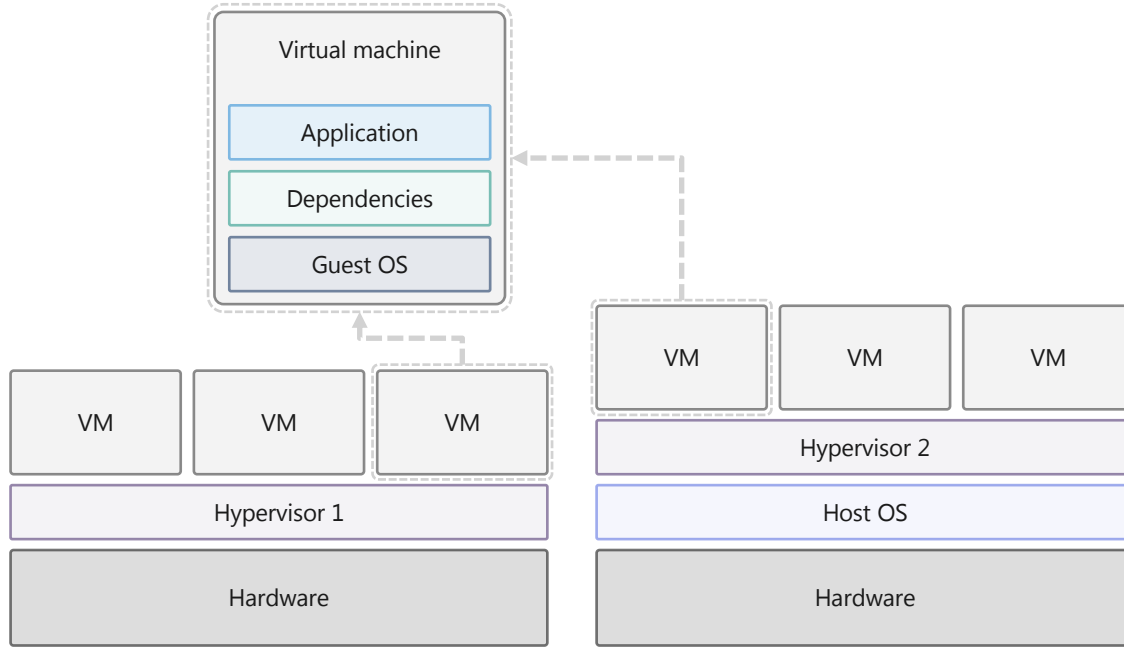


Docker Layered Filesystem

	LABEL	<code>io.codefresh.repo.branch=master io.codefresh.repo.hash=81be5f6 ..</code> SHOW MORE
	EXPOSE	<code>8080</code>
	CMD	<code>["croc-hunter"]</code>
5.77 MB	RUN	<code> 1 VCS_REF=531102d cd \$GOPATH/src/github.com/lachie83/croc-hun ..</code> SHOW MORE
	ENV	<code>GOPATH=/go</code>
	ENV	<code>GIT_SHA=531102d</code>
427.83 KB	COPY	<code>dir:f45c86e50dda1db46e1756352f9125f8fcb7c55a86750fb7b356eddd5a ..</code> SHOW MORE
1.30 MB	COPY	<code>dir:faa4a35eele82989750f1delc393abb0964bc839e6683ce46fddb317e5 ..</code> SHOW MORE
	LABEL	<code>org.label-schema.vcs-ref=531102d org.label-schema.vcs-url=http ..</code> SHOW MORE
	ARG	<code>BUILD_DATE</code>
	ARG	<code>VCS_REF</code>
	MAINTAINER	<code>Lachlan Evenson <lachlan.evenson@gmail.com></code>
2.42 KB	COPY	<code>file:ea7c9f4702f94a0df05f60648914e97f7876c4a7c5163e7870dd98fa8 ..</code> SHOW MORE
	WORKDIR	<code>/go</code>
	RUN	<code>mkdir -p "\$GOPATH/src" "\$GOPATH/bin" && chmod -R 777 "\$GOPATH"</code>
	ENV	<code>PATH=/go/bin:/usr/local/go/bin:/usr/local/sbin:/usr/local/bin: ..</code> SHOW MORE
	ENV	<code>GOPATH=/go</code>

Virtualization versus containerization

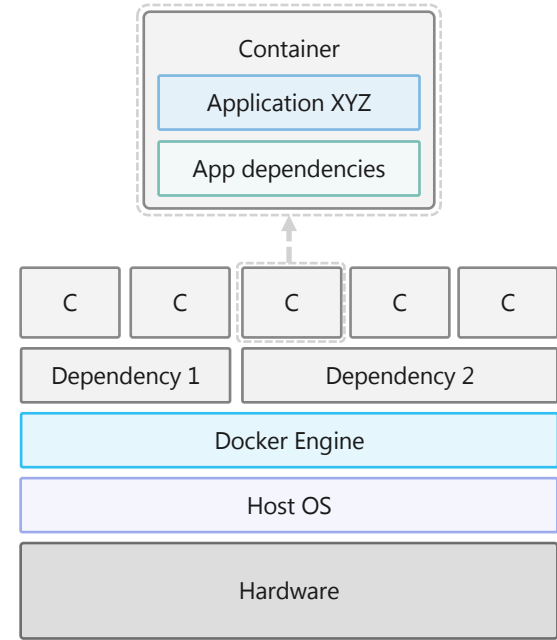
Virtualization



Type 1

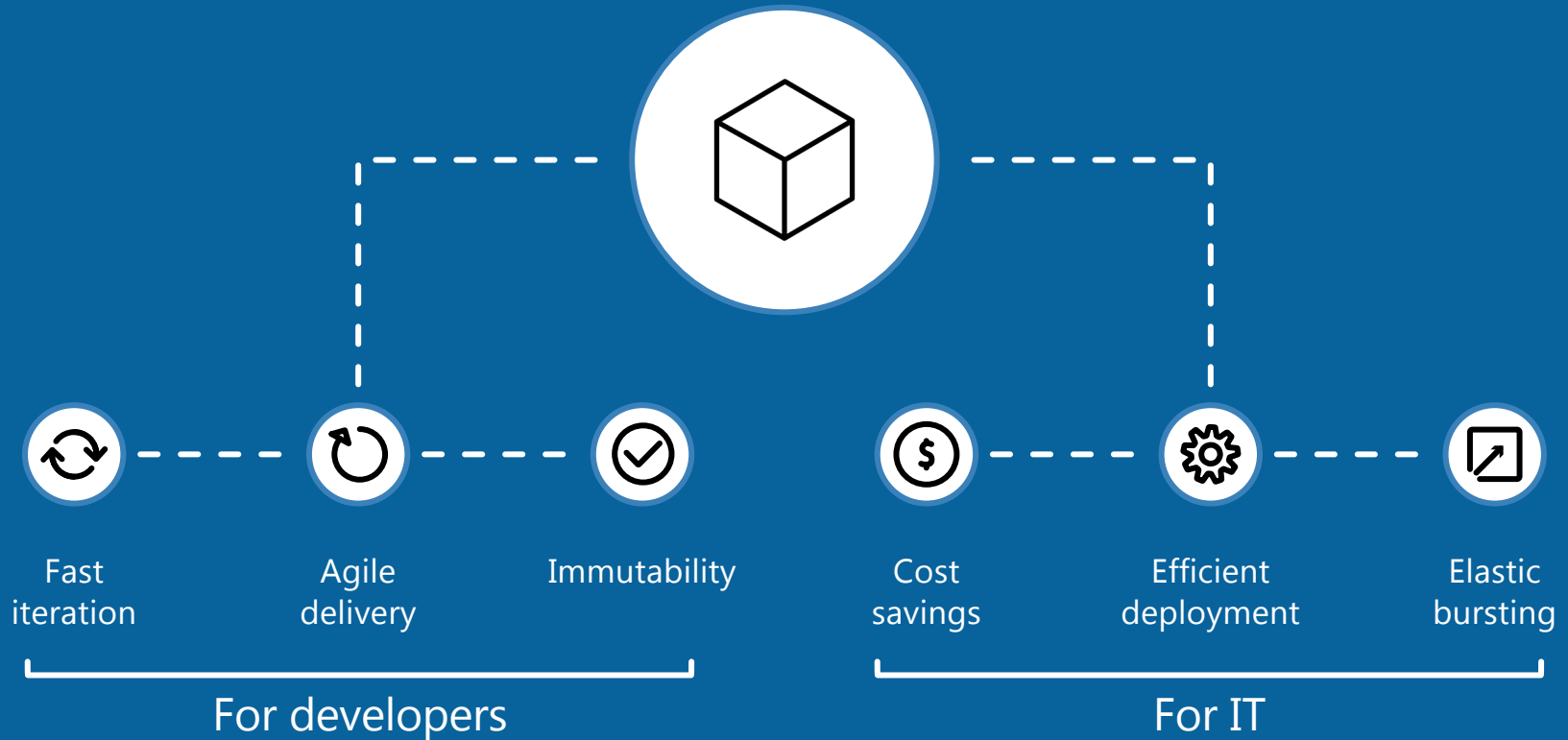
Type 2

Containerization



Hardware

The container advantage



The background is a solid blue gradient. On the left and right sides, there are abstract, overlapping geometric shapes in various colors including green, cyan, blue, orange, pink, and red. These shapes are layered and appear to be floating or overlapping each other, creating a sense of depth and movement. The word "Demo" is centered in the middle of the image in a white, bold, sans-serif font.

Demo

What did we just do?



Kubernetes Pipeline

Build



Kubernetes Repository

Package



JFrog Artifactory

Deploy



JFrog Xray

Test



JFrog Xray



Container Orchestration: Kubernetes

What is Kubernetes?

Open source container orchestrator that automates deployment, scaling, and management of applications.

Features include:

- ❖ Automatic bin packing
- ❖ Self-healing
- ❖ Horizontal scaling
- ❖ Service discovery
- ❖ Load balancing
- ❖ Automated rollouts and rollbacks
- ❖ Secret and configuration management
- ❖ Designed by Google
 - ❖ Based on their system used to run BILLIONS of containers per week
- ❖ Over 2,300 contributors
- ❖ Graduated from CNCF

The background is a solid blue gradient. It features several overlapping, semi-transparent geometric shapes in various colors including green, cyan, blue, orange, red, and pink. These shapes are arranged in a way that creates a sense of depth and movement, with some appearing to recede into the background and others coming forward.

Who is using Kubernetes?

Bloomberg



DENSO



GitHub



Goldman Sachs



intuit



Morgan Stanley



The New York Times



vevo



Walmart 

 CONCUR

VIACOM

 buffer



 Arkena

Goldman
Sachs


monzo

EVE
ONLINE

 Pearson

The
New York
Times

box

OpenAI

ticketmaster

SKY


COMCAST

Bloomberg

POKÉMON
GO



unacast.



Azure Kubernetes Service (AKS)

Your Kubernetes Cluster Managed by Azure

Why AKS?

Easy to use:

- ❖ Fastest path to Kubernetes on Azure
- ❖ Up and running with 3 simple commands
- ❖ I argue there are 2.5 commands

Easy to manage:

- ❖ Automated upgrades and patching
- ❖ Easily scale the cluster up and down
- ❖ Self-healing control plane

Uses open APIs – 100% upstream Kubernetes

Getting Started with AKS

```
$ az aks create -g myResourceGroup -n myCluster --generate-ssh-keys  
\ Running ..
```

```
$ az aks install-cli  
Downloading client to /usr/local/bin/kubectl ..
```

```
$ az aks get-credentials -g myResourceGroup -n myCluster  
Merged "myCluster" as current context ..
```

```
$ kubectl get nodes
```

NAME	STATUS	AGE	VERSION
aks-mycluster-36851231-0	Ready	4m	v1.8.1
aks-mycluster-36851231-1	Ready	4m	v1.8.1
aks-mycluster-36851231-2	Ready	4m	v1.8.1

Managing an AKS Cluster

```
$ az aks list -o table
```

Name	Location	ResourceGroup	KubernetesRelease	ProvisioningState
myCluster	westus2	myResourceGroup	1.7.7	Succeeded

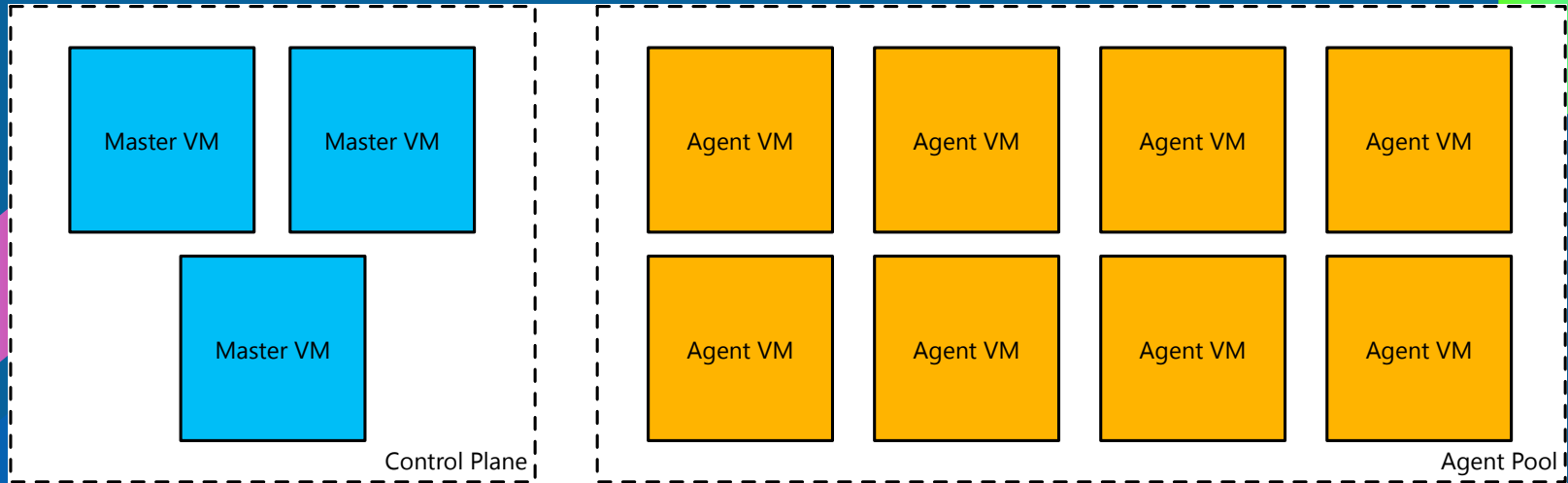
```
$ az aks upgrade -g myResourceGroup -n myCluster --kubernetes-version 1.8.1  
\ Running ..
```

```
$ kubectl get nodes
```

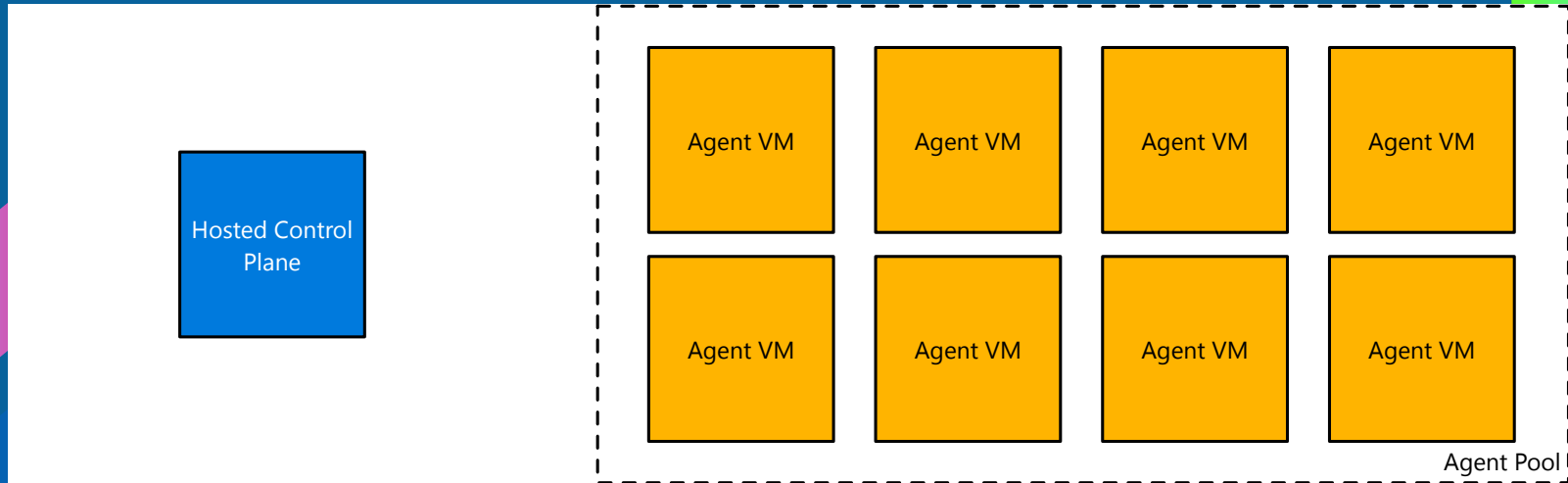
NAME	STATUS	AGE	VERSION
aks-mycluster-36851231-0	Ready	12m	v1.8.1
aks-mycluster-36851231-1	Ready	8m	v1.8.1
aks-mycluster-36851231-2	Ready	3m	v1.8.1

```
$ az aks scale -g myResourceGroup -n myCluster --agent-count 10  
\ Running ..
```

Kubernetes without AKS



Kubernetes with AKS





Azure Container Service (AKS)



Azure Container Instances (ACI)



Azure Container Registry



Open Service Broker API (OSBA)



Release Automation Tools

Release automation tools

Simplifying the Kubernetes experience



Streamlined
Kubernetes
development



The package
manager for
Kubernetes



Event-driven
scripting for
Kubernetes



Visualization
dashboard for
Brigade





Azure Container Service (AKS)



Azure Container Instances (ACI)



Azure Container Registry



Open Service Broker API (OSBA)



Release Automation Tools

Helm

The best way to find, share, and use software built for Kubernetes



Manage complexity

Charts can describe complex apps; provide repeatable app installs, and serve as a single point of authority



Easy updates

Take the pain out of updates with in-place upgrades and custom hooks



Simple sharing

Charts are easy to version, share, and host on public or private servers



Rollbacks

Use `helm rollback` to roll back to an older version of a release with ease





Azure Container Service (AKS)



Azure Container Instances (ACI)



Azure Container Registry



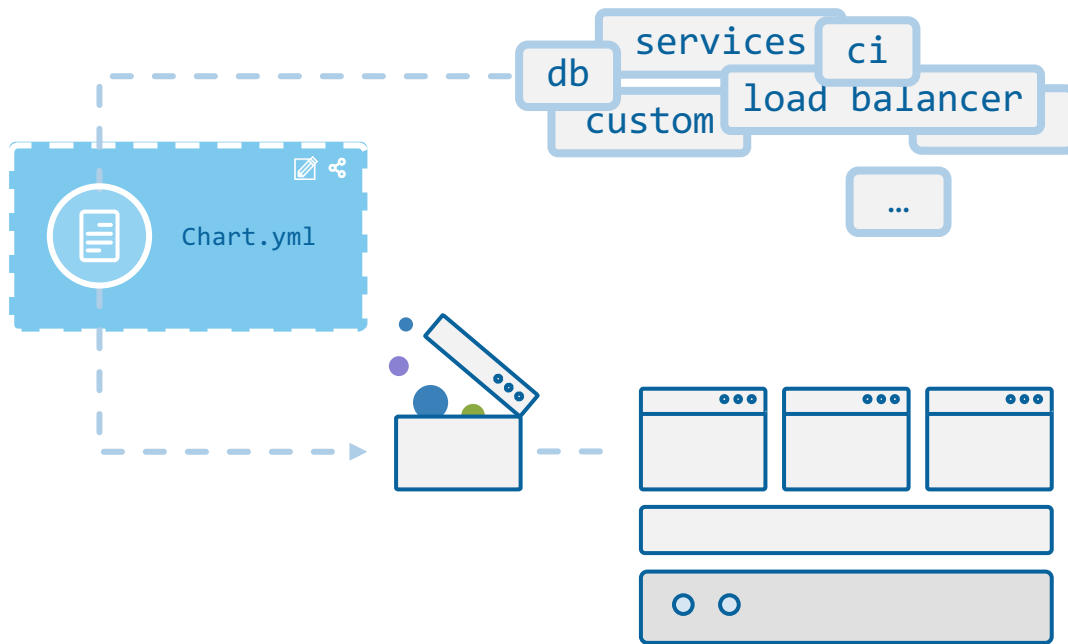
Open Service Broker API (OSBA)



Release Automation Tools

Helm

Helm Charts helps you define, install, and upgrade even the most complex Kubernetes application





Azure Container Service (AKS)



Azure Container Instances (ACI)



Azure Container Registry



Open Service Broker API (OSBA)



Release Automation Tools

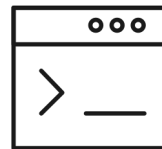
Draft

Simple app development and deployment – into any Kubernetes cluster



Simplified development

Using two simple commands, developers can now begin hacking on container-based applications without requiring Docker or even installing Kubernetes themselves



Language support

Draft detects which language your app is written in, and then uses packs to generate a Dockerfile and Helm Chart with the best practices for that language





Demo

5 Kubernetes Best Practices

- ❖ **Build small containers**
- ❖ **Application architecture**
 - ❖ **Use Namespaces**
 - ❖ **Use helm charts**
 - ❖ **RBAC**
- ❖ **Implement Health checks**
- ❖ **Set requests and limits**
- ❖ **Be mindful of your services**
 - ❖ **Map external services**
 - ❖ **Don't rely on load balancers**

THANKS!

Resources

aka.ms/devops/jaxlondon2018

Any questions?

You can find me at:

@jldeen · jessica.deen@microsoft.com

