Intro to Kubernetes

Tim Fogarty
Developer Advocate, mLab

Who is this for?

- No prior knowledge of Kubernetes needed
- If you need to brush up on containers, Naomi’s workshop on containers is on the meetup page.

Workshop Goals

- Learn what Kubernetes is
- Understand the components of a Kubernetes cluster
- Deploy our first applications using Kubernetes
- Explore the `kubectl` command line tool

Intro to Kubernetes

- **Learn what Kubernetes is**
- Understand the components of a Kubernetes cluster
- Deploy our first applications using Kubernetes
- Explore the `kubectl` command line tool

kubernetes

An open-source system for

deploying, scaling, and managing applications

K8s

An open-source system for

deploying, scaling, and managing applications

Intro to Kubernetes

An open-source system for

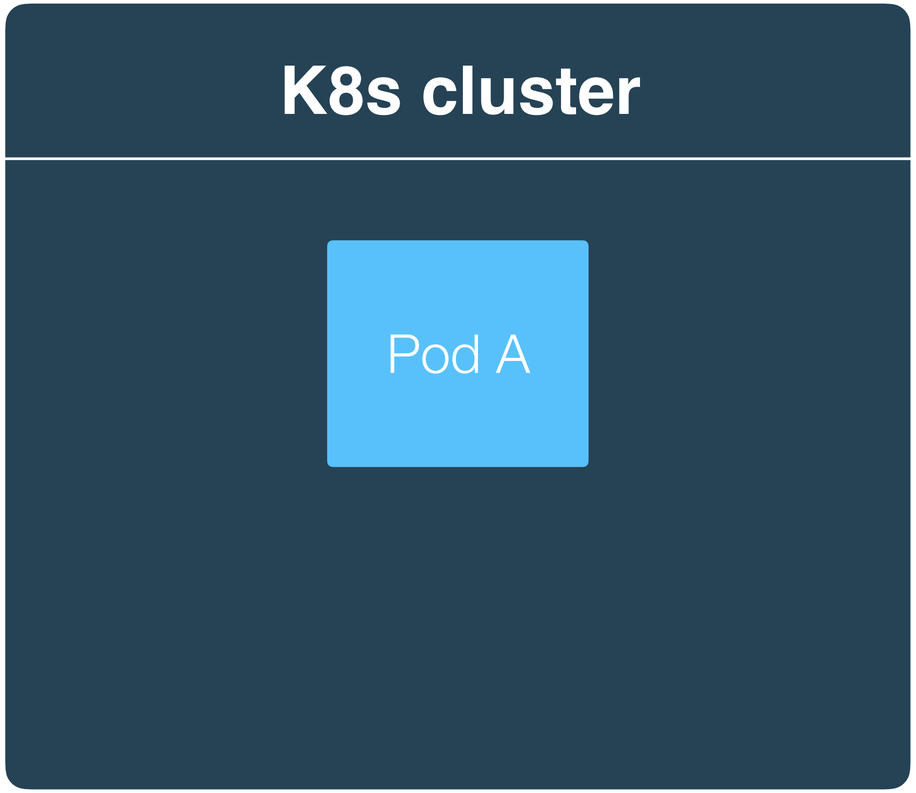
deploying, scaling, and managing applications

- **2003** - Google starts development on Borg
- **2014** - Google open sources Kubernetes
- **2015** - Google donates Kubernetes to the CNCF
- **2016** - Kubernetes powers Pokémon GO
- **2018** - 40% of enterprise companies are using K8s

What does Kubernetes do?

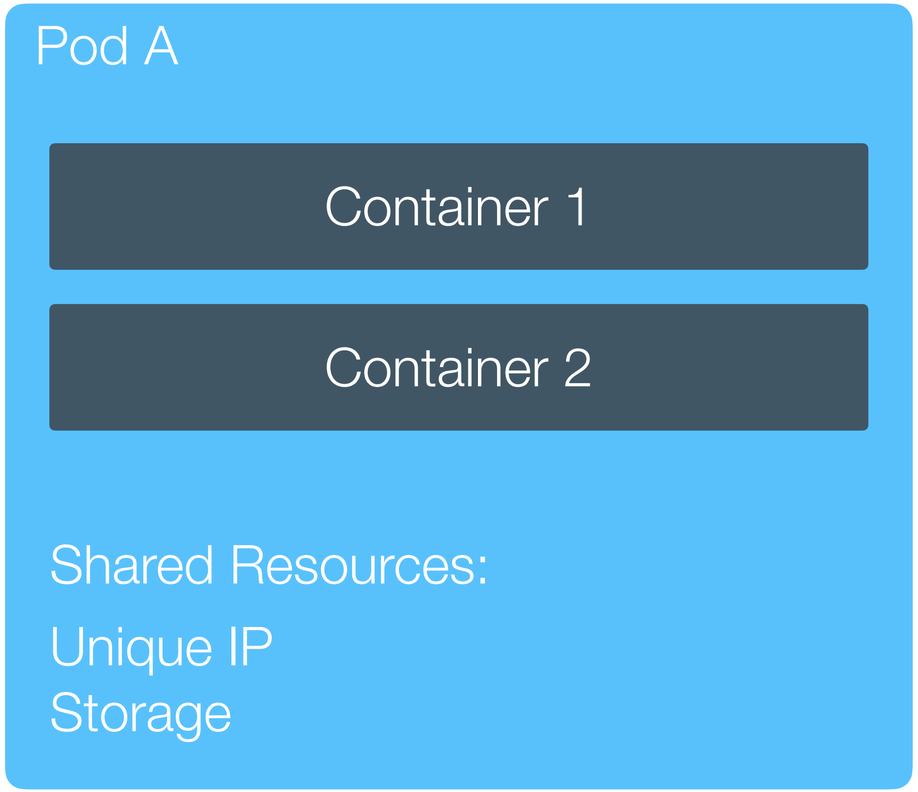
App A definition

Container: A
Port: 80
Memory: 500MB



Pods

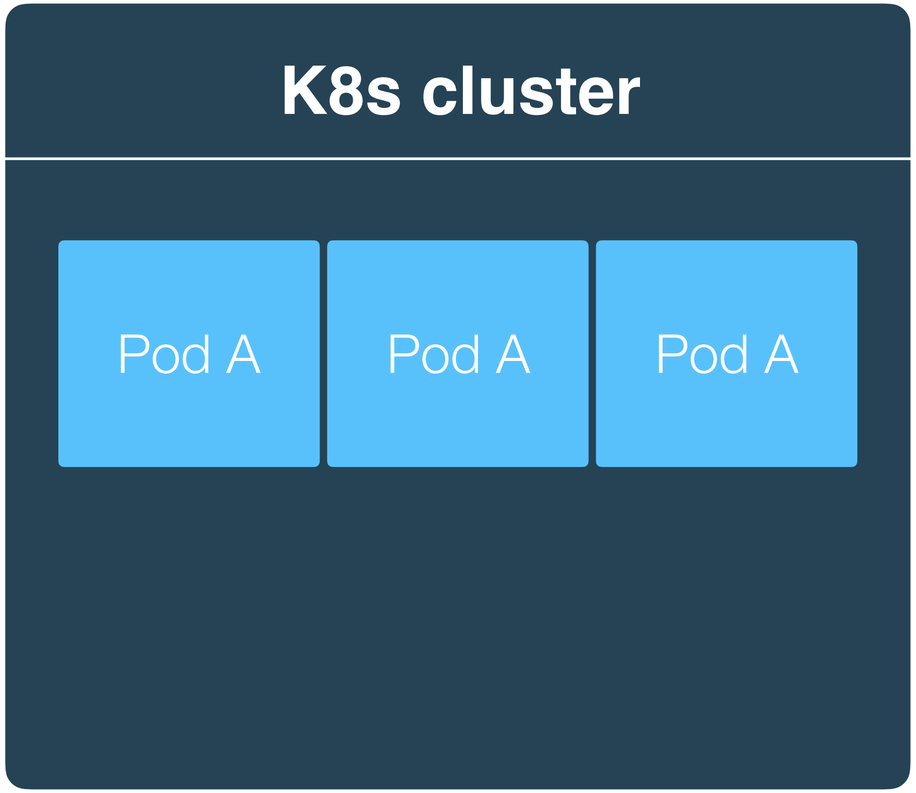
- Pods are the basic unit of deployment in Kubernetes
- Wrapper around one or more containers
- Containers in a pod share resources such as storage and a unique network IP



Easy to scale

App A definition

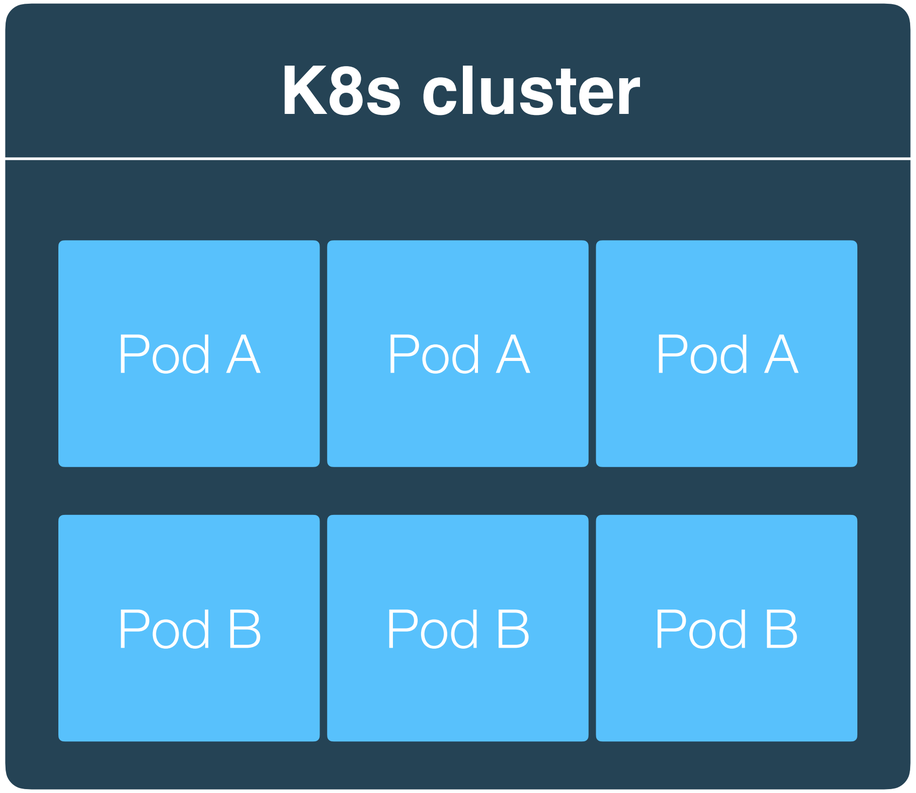
Container: A
Port: 80
Memory: 500MB
Replicas: 3



Maximize Capacity

App B definition

Container: B
Port: 80
Memory: 500MB
Replicas: 3



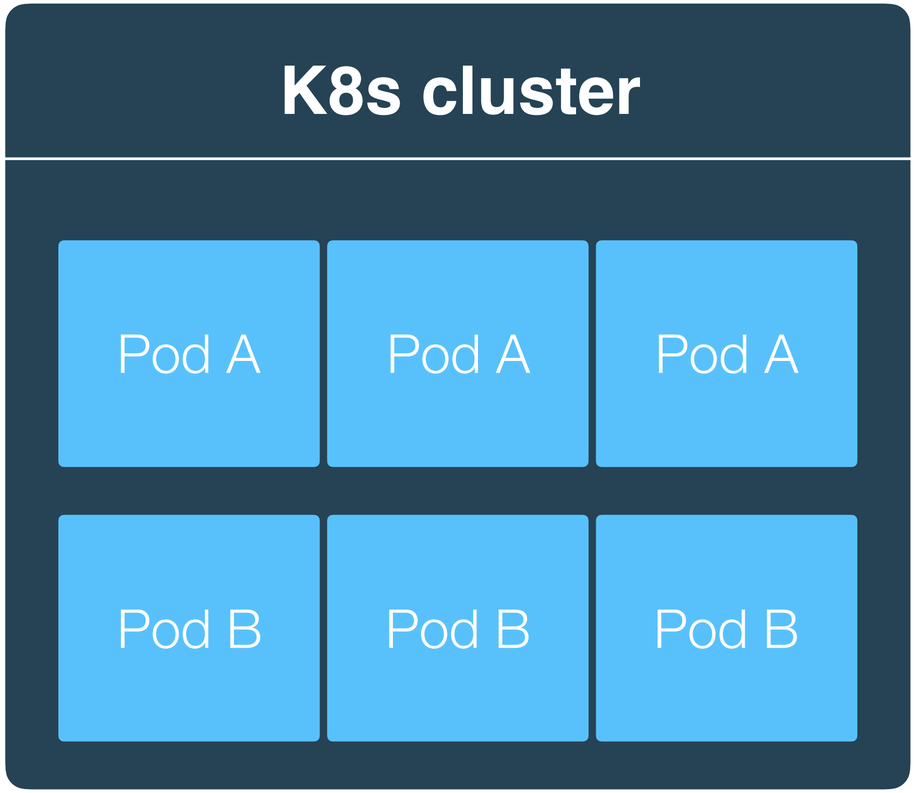
Reconciliation Loop

Self healing

Monitoring…

App A: Healthy

App B: Healthy



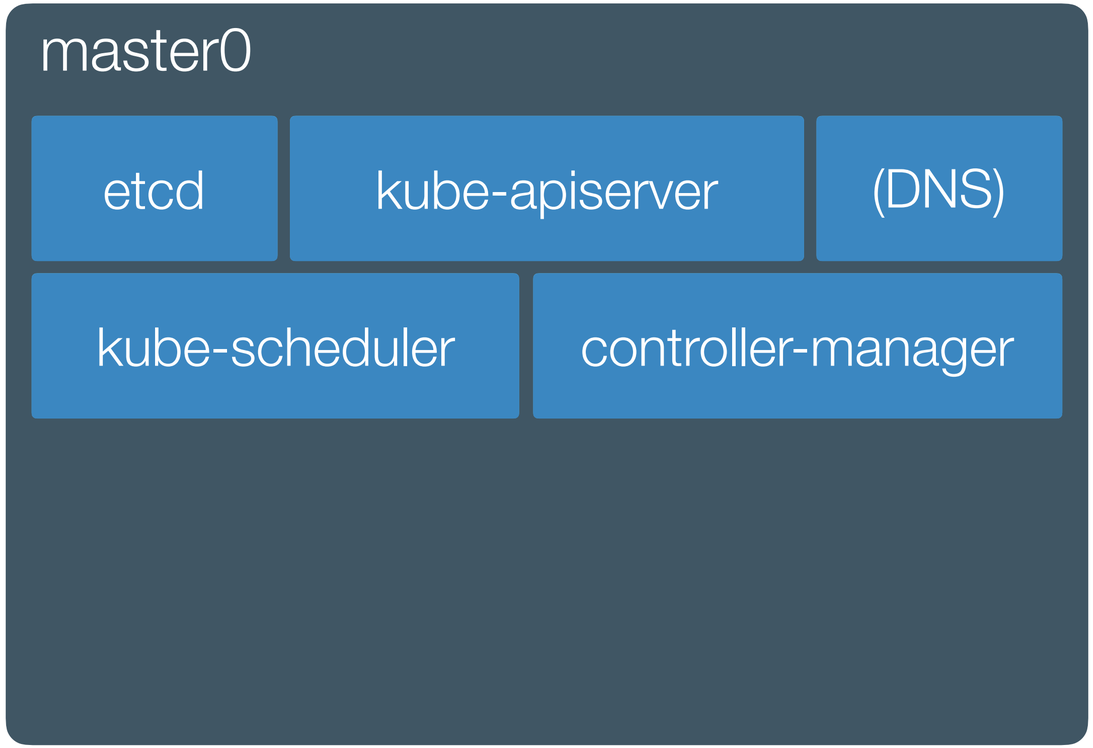
What makes Kubernetes great?

- Declarative configuration
- Easy to scale
- Maximize capacity
- Self healing

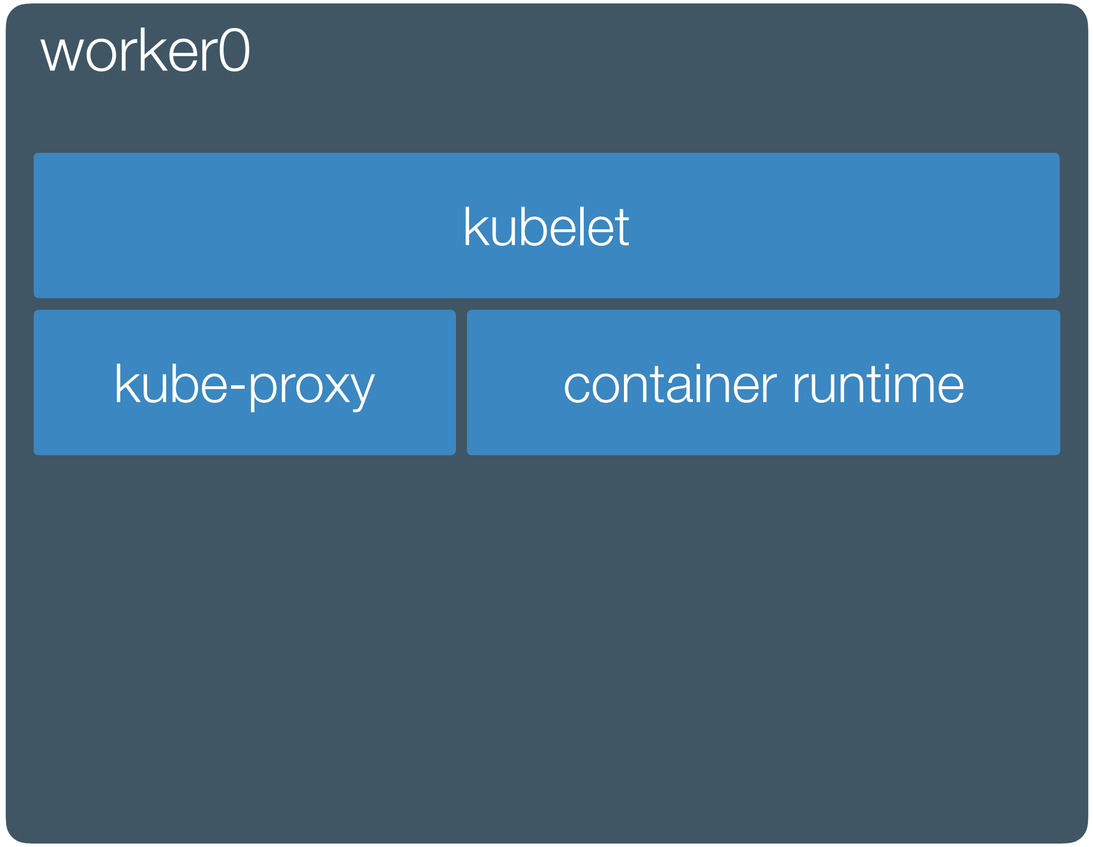
Intro to Kubernetes

- Learn what Kubernetes is
- **Understand the components of a Kubernetes cluster**
- Deploy our first applications using Kubernetes
- Explore the `kubectl` command line tool
- See the basics of storage in Kubernetes

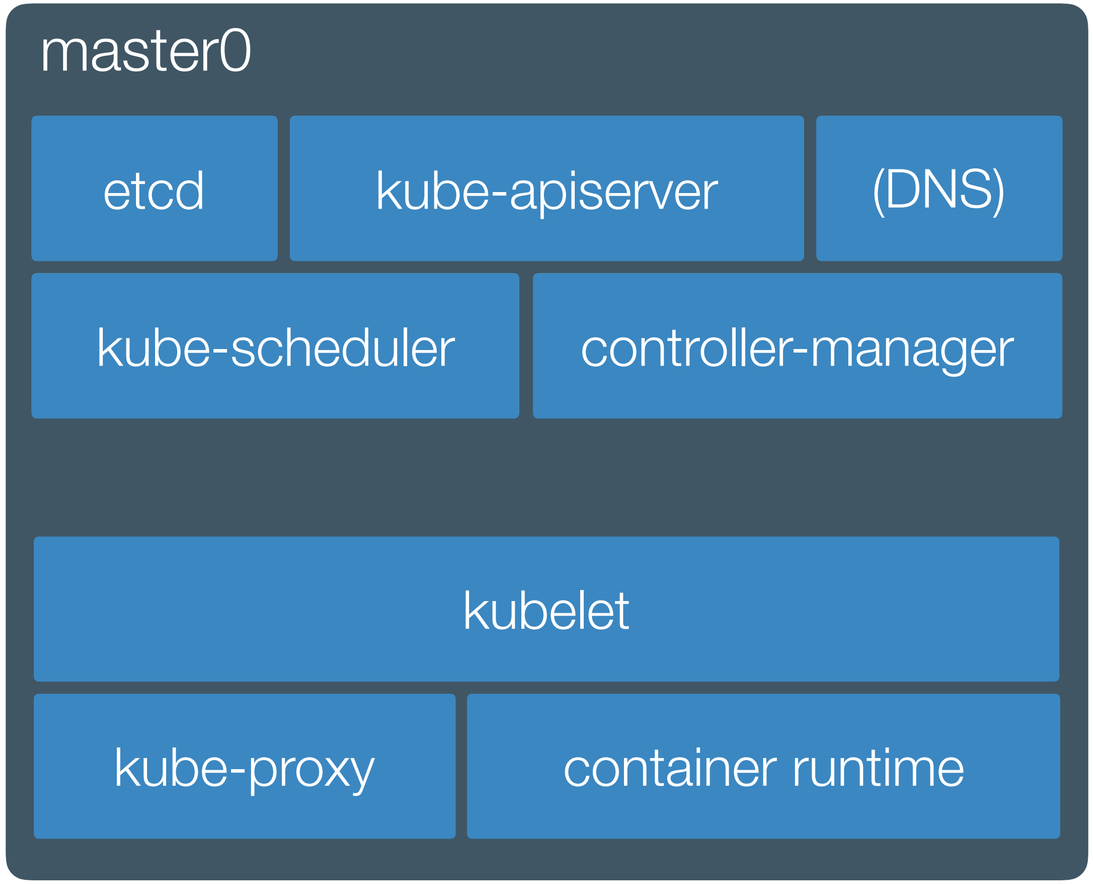

K8s cluster
Master Nodes
master0
master1
master2
Worker Nodes
worker0
worker1
worker2
worker3
worker4
worker5


master0
etcd
kube-apiserver
(DNS)
kube-scheduler
controller-manager


K8s cluster
Master Nodes
master0
master1
master2
Worker Nodes
worker0
worker1
worker2
worker3
worker4
worker5


worker0
kubelet
kube-proxy
container runtime


K8s cluster
Master Nodes
master0
master1
master2
Worker Nodes
worker0
worker1
worker2
worker3
worker4
worker5


master0
etcd
kube-apiserver
(DNS)
kube-scheduler
controller-manager
kubelet
kube-proxy
container runtime

What makes Kubernetes great?

- Declarative configuration
- Easy to scale
- Maximize capacity
- Self healing
- Configurable and extensible

Setting up your own cluster

- **Hard way:** set up the components manually
- **Easier way:** tools like kops, kubeadm, kubespray
- **Easiest way:** managed clusters on Google, AWS, or Azure
- **Local testing:** minikube

Go to Katacoda and sign up: bit.ly/smooth-k8s

What we've learned

- Learn what Kubernetes is
- Understand the components of a Kubernetes cluster
- Deploy our first applications using Kubernetes
- Explore the `kubectl` command line tool
    - Used Pods, Deployments, and Services

There's more!

- How do you deal with stateful services?
    - StatefulSets, Volumes, Persistent Volumes, Persistent Volume Claims, Storage Classes
- How do you use secrets?
    - ConfigMaps, Secrets
- What about jobs?
    - Jobs, CronJobs

Learn more

- kubernetes.io
- *Kubernetes Up and Running*
- *The Kubernetes Book*
- katacoda.com

Thank You!

Tim Fogarty
Developer Advocate, mLab

Smooth DevOps