Kubernetes is Boring
And the Future is Exciting

Clayton Coleman
Contributor, Engineer, Architect - Red Hat
What Is Kubernetes?
Meet Kubernetes

- Greek for “Helmsman,” also the root of the word “Governor” (from Latin: gubernator)
- Container orchestrator
- Supports multiple cloud and bare-metal environments
- Inspired by Google’s experience with containers
- Open source, written in Go
- Manage applications, not machines
But What Does it Do???
WHAT WOULD YOU SAY
YOU DO HERE?
Simple model for keeping apps running

- Let apps scale up and down
- Roll out new versions
- Move apps around if a machine fails
- Hand out resources fairly among different apps
- Debug like you would on your own machine
- “I’m tired of reinventing this wheel!”
Is Kubernetes for me?
Are you:

- A team that runs lots of services that you want to have fairly consistent practices and patterns?
- An enterprise infrastructure team that has to manage hundreds or thousands of applications?
- An enthusiast?
Are you:

- A team that runs lots of services that you want to have fairly consistent practices and patterns?
- An enterprise infrastructure team that has to manage hundreds or thousands of applications?
- An enthusiast?

Then...

Maybe

Yes!
But remember
Kubernetes is not magic
Why is it boring?

* Besides the lack of unicorns
Which part of flying should be exciting?
Which part of flying should be exciting?
Which part of flying should be exciting?
Destination, not the Journey
@smarterclayton

- ccoleman@redhat.com
- Working in app focused infra (aka PaaS!) since 2010
- Part of Kubernetes project since June 2014
- Should be blamed for anything you hate in Kube
- Always looking for technologies that enable developers to deliver better software faster and operations to sleep easy at night
- Has been saying Kubernetes should be boring since 2014

Before Kubernetes
@smarterclayton

- ccoleman@redhat.com
- Working in app focused infra (aka PaaS!) since 2010
- Part of Kubernetes project since June 2014
- Should be blamed for anything you hate in Kube
- Always looking for technologies that enable developers to deliver better software faster and operations to sleep easy at night
- Has been saying Kubernetes should be boring since 2014
- Wishes he could still sleep at night

After Kubernetes
After 6+ years, Kubernetes is mostly boring

- Predictable
- Stable
- Extensible
- Thriving community

- But definitely not done, over, or finished
What’s Next?
Kubernetes
Kubernetes

Not the point
YOU

...are the point
Open Source

People Empowering Each Other
portability  
resiliency  
ease-of-use  
low-code  
data privacy  
security  
Knowledge sharing  
live debugging  
alerting  
reliability  
alerting  
cloud services  
code reuse  
data gravity  
Kubernetes?
It Should be Easier...

- ... to go from development on your laptop to production anywhere in the world.
- ... to debug your running applications (if you’ve heard of topics like tracing and observability and log aggregation you should say - I want that!)
- ... to enable self-service within your team / org / company AND have reasonable controls.
- ... to track the security of all the open source software you run and get prompted when your dependencies need to be updated
Kubernetes

useful abstractions
Monitor automatically

Instance 1
https
metrics

Instance 2
https
metrics

Instance 3
https
metrics
Monitor automatically

Service 1

- Instance 1
  - https
  - metrics

- Instance 2
  - https
  - metrics

- Instance 3
  - https
  - metrics
Monitor automatically

Service 1

Instance 1
https
metrics

Instance 2
https
metrics

Instance 3
https
metrics

Prometheus

scrape

use
Monitor automatically

Service 1

Instance 1
https
metrics

Instance 2
https
metrics

Instance 3
https
metrics

Prometheus

use already defined app topology to gather metrics - Prometheus does less, user does less
Service Mesh

Service 1

- Instance 1
- Instance 2
- Instance 3
  - https

Service 2

- Instance 1
- Instance 2
  - https
Inject and replace networking, provide additional capabilities like authentication and authorization.
Service Mesh + Monitor

Instance 1
https
Instance 2
https
Instance 3
https

Service 2
Instance 1
https
Instance 2
https

Prometheus

Auto discover injected service
Create Awesome New Abstractions
Examples from Community

- Schedule applications into clusters, detect failures, and rebalance automatically
- Describe your service dependencies by interface and have them provisioned automatically (dev, test, prod)
- Get rid of secrets by giving every instance its own identity and credentials, and centralize authz
- Provision and deploy your apps by introspecting source code directly, rather than having to define YAML
No Crystal Ball, but...

- This is a golden age of building services
- Kubernetes is a solid foundation, among many
- Everything will get more complex
- You’ll be the ones building the services and tools of tomorrow
The future belongs to you.

Thanks!