When to rm -rf Kubernetes

Paul Steele



About Me

- Software Engineer at SEP
- Purdue University
- <u>https://blog.paul-steele.com/</u>



Kubernetes at home





Developing for clusters



You can run Docker containers without Kubernetes





Who has used...

• Containers?



Who has used...

- Containers?
- Container Orchestrators?



Who has used...

- Containers?
- Container Orchestrators?
- Kubernetes?



kubernetes

Agenda

- Containers & Container Orchestration
- Kubernetes
 - What does it offer?
 - \circ When it isn't ideal
 - \circ When it is ideal
- Do I need Kubernetes?
 - \circ alternatives

What are Containers?

Containers

- Packaging for applications
 - \circ Standardized
 - \circ Modular
 - Fast to spin up
 - \circ Portable



Like Virtual Machines but not



Running Containers



Running More Containers





...and more containers





...too many containers







What is Container Orchestration?

- Abstraction of underlying hardware
 - Machines
 - Storage
 - Networks



Orchestrator

• Workload scheduling



• Horizontal scaling





Applications

- Self healing
 - Machine failures
 - App hiccups

Orchestrator Node Node Node app app app version A version A version B app version B Scales up new Tears down old version version alongside old

• Rolling Updates

What Orchestrator to Pick?

- Docker Swarm?
- Mesos DC/OS?
- Kubernetes?







Kubernetes

- Open source version of Google's Borg
 - \circ $\,$ 1.0 release in July 2015 $\,$
- Written in Go
- Greek for "Helmsman"
- "Production-Grade Container Orchestration"
- K8S for short
- Becoming De-facto orchestrator



What does Kubernetes offer?

Nodes



Nodes



Kuberneters Cluster

Pods



Pods



https://kubernetesbootcamp.github.io/kubernetes-bootcamp/

Pods



Services



Services



Distinctive Features

- Role Based Authentication
 - For all actions


Distinctive Features

- Namespaces
 - Isolate resources
 - Share hardware between Test & production



When it isn't ideal

#IndyCloudConf

- Provisioning Tool
 - Custom built
 - Used for allocating customer resources
 - Not containerized at all

- Minikube for local development
- Dependencies included:



- Minikube for local development
- Dependencies included:
 - o Mysql



- Minikube for local development
- Dependencies included:
 - o Mysql
 - Mongodb



- Minikube for local development
- Dependencies included:
 - o Mysql
 - Mongodb
 - Redis







- Minikube for local development
- Dependencies included:
 - o Mysql
 - Mongodb
 - Redis
 - Jenkins







- Minikube for local development
- Dependencies included:
 - o Mysql
 - Mongodb
 - Redis
 - Jenkins
 - Jenkins Connector









- Minikube for local development
- Dependencies included:
 - o Mysql
 - Mongodb
 - Redis
 - Jenkins
 - Jenkins Connector
 - Jaeger







mongoDB

- Minikube for local development
- Dependencies included:
 - o Mysql
 - Mongodb
 - Redis
 - Jenkins
 - Jenkins Connector
 - Jaeger
 - Elasticsearch



JAEGER

mongoDB



- Minikube for local development
- Dependencies included:
 - o Mysql
 - Mongodb
 - Redis
 - Jenkins
 - Jenkins Connector
 - Jaeger
 - Elasticsearch
 - LocalStack





• What didn't this setup include?



- What didn't this setup include?
 - $\circ~$ A container for the application itself
- Application is currently 6 months behind
 - \circ No end in sight

Kubernetes is not

- A fix for bad architecture
 - Just because you can, doesn't mean you should
- Your architecture
 - It is a place for your architecture to run

Kubernetes is not

- A development only environment
 - Unused configuration
 - Wasted time
 - Introduces bugs



Kubernetes is

- A development + production environment
 - Consolidated configurations



Kubernetes is not

- For singular applications
 - overkill
 - Has a high initial cost of setting up
 - Not just for running docker





Where Kubernetes Isn't Ideal

- Self Hosted
 - Managing K8S is a full time job
 - API server for your infrastructure
 - High recurring cost of maintaining
 - Can get into weird scenarios

Where Kubernetes Isn't Ideal

Not taking advantage of the services it offers

- Service discovery 0
- Load balancing 0
- **Role Based Authentication** \bigcirc
- Configuration management 0
- Horizontal scaling 0



Batch execution

solutions.

In addition to services, Kubernetes can manage your batch and CI workloads, replacing containers that fail, if desired,

requirements and other constraints, while not sacrificing availability. Mix critical and best-effort workloads in order to

containers when nodes die, kills containers that don't respond to your user-defined health check, and doesn't advertise them to

without rebuilding your image and without exposing secrets in

Horizontal scaling

Scale your application up and down with a simple command, with a UI, or automatically based on CPU usage.

When it is ideal

#IndyCloudConf

Where Kubernetes Is Ideal

- Running multiple applications that interact with each other
- When you can take advantage of the services it offers
 - Service discovery
 - Load balancing
 - Role Based Authentication
 - Configuration management
 - Horizontal scaling

Where Kubernetes Is Ideal

- Microservices
 - \circ The 12 Factor App
 - Guidelines for Software as a Service

Codebase

- All code for an application lives in one repo
- Kubernetes configurations live In source control



Configuration

- Configuration for application doesn't live in code
- Key Value Pairs
 - configMaps
 - Secrets
 - \circ Env vars
 - Mounted files



- Pods
- Services

Processes

- Application itself is stateless
- Data that persists is in backing resource

Concurrency

- Application should be able to scale horizontally
- Horizontal Pod Autoscalers
 - Cpu
 - \circ Memory
 - custom



Disposability

• Application can be stopped and started at any notice

Dev / Prod Parity

- Use the same services in development & production
- Change what orchestrator you're using not environment

Logs

- App doesn't concern itself with storage of logs
- K8S handles it

Kubernetes is

- A bridge between Dev & Ops
 - Ops provides K8S as a service
 - Devs are in charge of their deployments



#IndyCloudConf

So should I use Kubernetes?

#IndyCloudConf

Should I use Kubernetes?

- More than one application?
 - If not, focus on automating its deployment

Should I use Kubernetes?

- Can follow deployment through to production?
 - If not, focus on using docker-compose

Should I use Kubernetes?

- Have the team to support it
 - Or ability to pay someone to (Amazon / Azure)
 - Focus on reducing complexity of application
Should I use Kubernetes?

- Applications follow the 12 factor app?
 - Statelessness
 - Focus on automation / re-architecting application

Should I use Kubernetes?

- Can you take advantage of?
 - Service discovery
 - Load balancing
 - Role Based Authentication
 - Configuration management
 - Horizontal scaling

Summary

- Kubernetes is a tool
 - Pros
 - Cons
- Can solve many problems
 - \circ Can cause them too
- For organizations, not projects

Questions?

Thanks!

• Slides can be found: https://info.sep.com/2019indycloudconf