Kubernetes 101

A quick intro to Kubernetes and its concepts

Before we begin

heptio

- Get involved! Audience participation is encouraged and requested.
- This presentation will be made available online after the event, so don't worry about taking pictures of every slide. (Although you can if you *really* want to.)
- If you want to participate in the hands-on portion, please ensure you've downloaded and installed Minikube and have run **`minikube start`** successfully.

Who am I?

heptio

- Husband, father, Jeeper, all-around geek
- Blogger (12+ years at https://blog.scottlowe.org)
- Author (8 books so far)
- Speaker (Interop ITX, VMworld, DevOps Networking Forum, OpenStack Summit, local meetups)
- Podcaster (The Full Stack Journey podcast)
- Staff Field Engineer at Heptio

Quick terminology definition

heptio

K8s == Kubernetes

heptio

K8s background

K8s

heptio

- Originated out of container orchestration systems running at Google
- 3rd-generation container orchestration system (Borg, Omega, K8s)
- Intended to be open source from the very beginning
- Cloud Native Computing Foundation (CNCF) now has governance of K8s

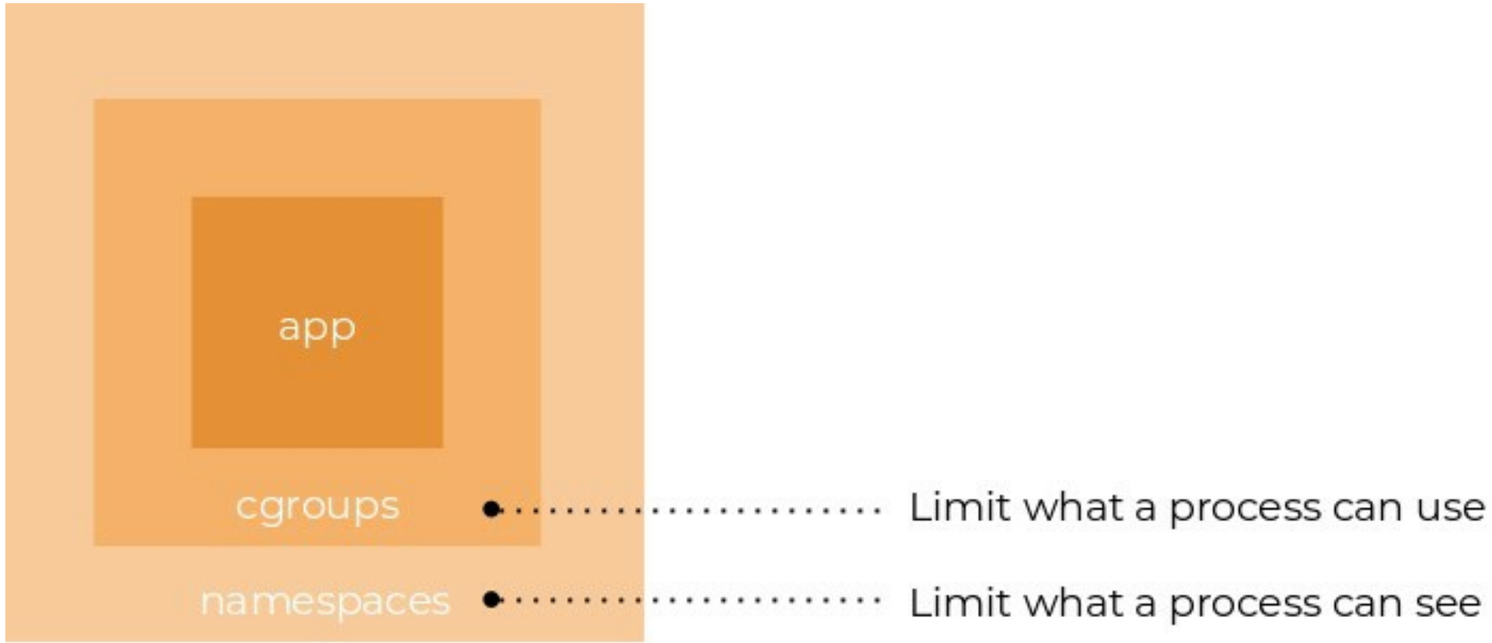
heptio

Kubernetes concepts

K8s concepts: Container

heptio

Containers are a lightweight method for process isolation.
Containers are not VMs!



K8s concepts: Container (cont.)

heptio

There are a few different container runtimes you can use with Kubernetes:

- Docker (most commonly used)
- ContainerD
- CRI-O
- Rkt

K8s concepts: Pod

heptio

- A pod is a group of one or more containers that have shared resources and a shared fate
- Independently update-able and configurable
- Pods, not containers, are the unit of work in K8s

K8s concepts: Replica Set

heptio

- A replica set is a defined number of similar pods generated from a template
- K8s ensures that the defined number of pods are always running, even in the event of machine failure
- Brings a measure of self-healing to K8s

K8s concepts: Deployment

heptio

- A deployment manages a replica set and brings application lifecycle functionality to K8s
- Enables rolling updates, rollback, and deployment history
- Can be used for zero-downtime upgrades

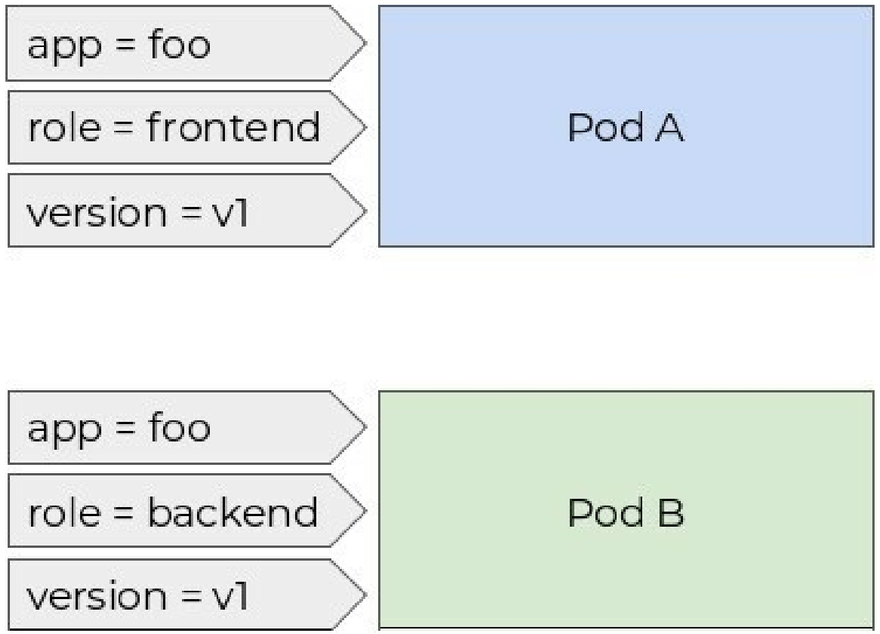
K8s concepts: Label

heptio

- A label is a simple key-value pair attached to K8s resources, like a pod
- No enforced hierarchy
- Labels make selector-driven operations possible

K8s concepts: Label (cont.)

heptio



K8s concepts: Service

heptio

- Reliable addressing of a collection of pods
- Addressing remains constant even as pods start/stop/die
- Provides discoverability (API, DNS)
- Membership is selector-driven (determined by labels)
- Several types (ClusterIP, NodePort, LoadBalancer)

K8s concepts: Volume

heptio

- Volumes (or Persistent Volumes) provide durable storage for ephemeral pods
- Shared file space for containers in a pod
- Variety of options (cloud-based block storage, iSCSI, host paths)
- Lots of vendors providing integration here

K8s concepts: ConfigMaps and Secrets

heptio

- Provide a mechanism for separating code and configuration to enable widely re-usable containers
- Accessible to containers in pods as volumes or as environment variables

heptio

Live demo

heptio

Questions & answers

heptio

Hands-on workshop

Visit https://github.com/scottslowe/minikube-workshop for details and instructions

Thank you!

Please be sure to provide feedback to the organizers about this session