instaclustr

Cassandra and Kubernetes

Ben Bromhead, CTO Instaclustr

January 2018

/usr/bin/whoami

- Ben Bromhead, CTO of Instaclustr
- We provide managed Cassandra, Spark and Kafka in the cloud (AWS, GCP, Azure & Softlayer).
- We provide support and services as well for those in private data centers.
- Manage and support 2k+ nodes.

Agenda

- Containers and Kubernetes
- Kubernetes and state
- Running Cassandra on Kubernetes

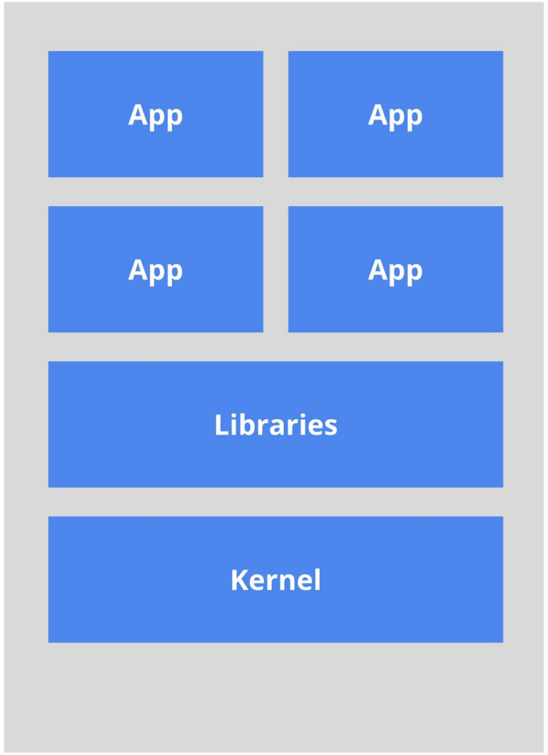
Containers - For managers

Essentially a way to bundle all the dependencies of a given process and keep it isolated…

Containers - For managers

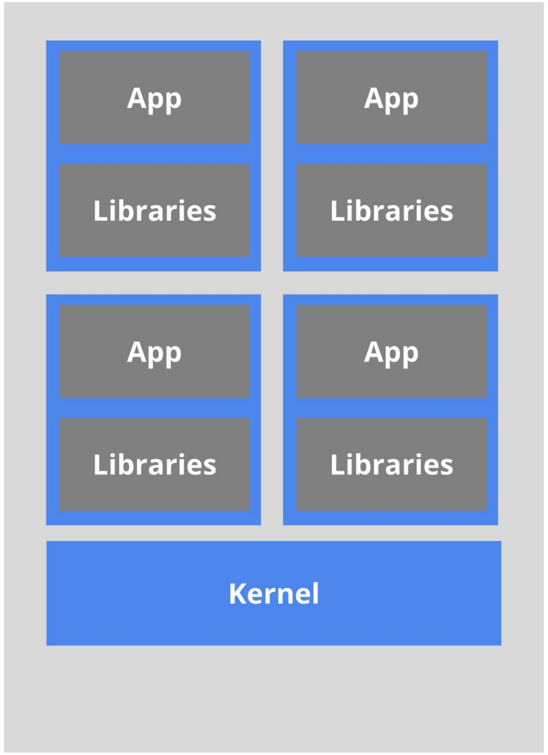
instaclustr

The old way: Applications on host



Heavyweight, non-portable
Relies on OS package manager

The new way: Deploy containers



Small and fast, portable
Uses OS-level virtualization

Containers - For managers

What does this actually get you

- A separation of concerns. Developers can build an application / service and deliver it as a container that has defined interfaces. Operators don’t (generally) care what’s inside the container.
- Reproducible artefacts that are the same across all environments. That image you built on your laptop can be validated, tested and put into production with no changes.
- Lightweight VMs
- Simple package management
- A building block for microservices architecture

Containers - For engineers

A container is made up of a few things:

- Process and resource isolation. Shares the host kernel but can’t “see” other processes etc.
- Some sort of chroot environment. Bring your own userland. Need specific/unique libraries / services / programs /distro for your app? Done.
- Some sort of image, that contains everything that will be run in the isolated environment.

Containers

Containers	Zones	Jails	VMs
Cgroups Namespaces chroot env AUFS etc...	First class concept	First class concept	First class concept

With apologies to Jessie Frazelle - https://twitter.com/jessfraz

Containers

This mix of components is not a bug, but a feature!

- VMs, Jails, Zones etc do everything for you, with minimal choice.
- Containers, everything is optional or pluggable
    - Want to allow two container to share the same network namespace? Sure go for it!
    - Don’t want AUFS, fine use BTRFS.
    - Want a good filesystem, mount a host directory into the container (yay XFS)
- Docker, rkt, containerd, kubernetes etc all try to give you sane defaults so that containers work (somewhat) like VM/Jail/Zones.

Containers

Awesome so a container is an isolated process that gets its own userspace, which has the side effect of making operations easier!

Kubernetes - For managers

A service that runs your containers for you on lots of computers and tries to be smart about it.

Kubernetes- For managers

Officially: Kubernetes is an open-source platform designed to automate deploying, scaling, and operating application containers.

Kubernetes- For managers

It won the war:

AWS ECS, Mesosphere, Docker Swarm

All support Kubernetes as a first class citizen

Kubernetes- For managers

And it’s taking over the world:

AWS EKS

Google Cloud Kubernetes Engine

Azure Kubernetes Service

Red Hat OpenShift

Pivotal Kontainer Service

CoreOS

Mesosphere

Docker Swarm

Kubernetes - For engineers

Kubernetes is made up of a few things:

- A database that manages state.
- Services that manage your system and move it from its current state to its intended state
- Tools, methods and formats for telling kubernetes what state you want it to be in.

Kubernetes

What do you get with Kubernetes? A lot!

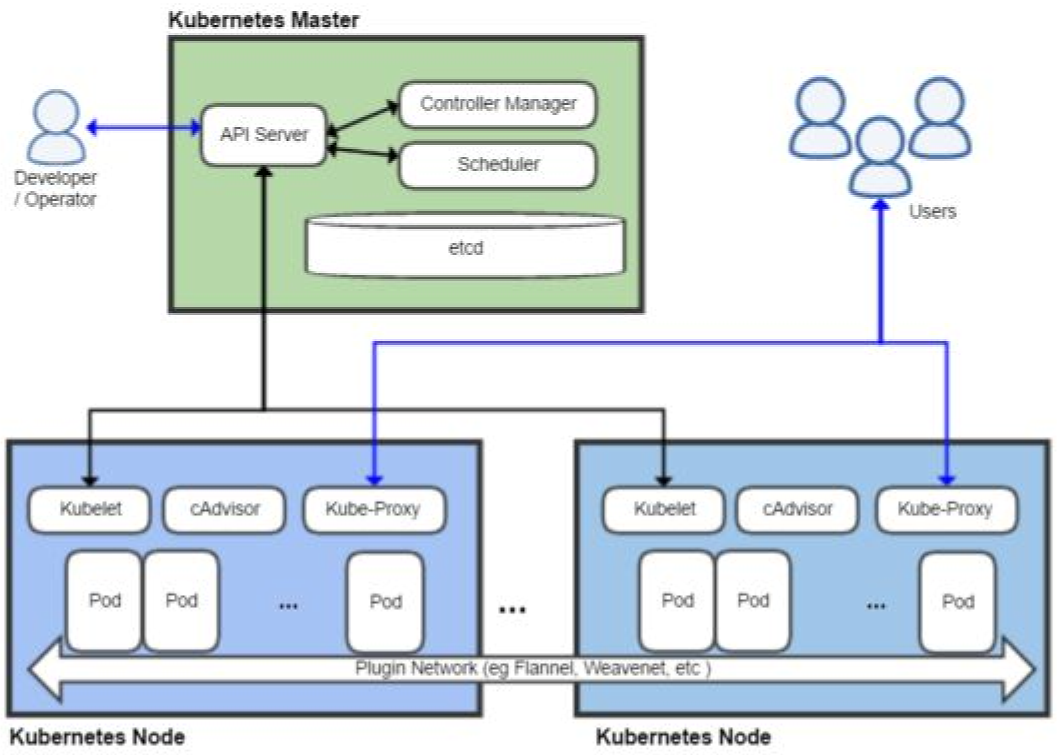
- Managing dependent/related containers
- Managing storage
- Distributing secrets
- Managing application health
- Replication
- Scaling
- Load balancing
- Updates
- RBAC!
- more

Fundamentals of Kubernetes

Before we get any deeper, an introduction to some Kubernetes specific terms

- K8s (industry approved abbreviation for Kubernetes)
- Pod - Represents a running process on your cluster.
- Controller - A control loop that resolves intended state to actual, the fundamental automation process in Kubernetes. E.g.
    - ReplicaSet - A controller that ensures there are N pods for a ReplicaSet
    - Deployment controller - declarative updates for Pods and ReplicaSets.

Fundamentals of Kubernetes



Fundamentals of Kubernetes

Controllers are the primary method of mutating infrastructure in Kubernetes. All controllers use the following basic control loop:

- Observe - Gather the current state of the system
- Analyze - Determine the differences between the current state and intended state
- Act - Implement a single action to drive current state closer to intended state.

Kubernetes

But also it sucks… at dealing with state

Kubernetes

I would say blame Docker, but state is hard in a distributed system

Kubernetes

instaclustr

Thought (troll) leadership of the day:

If you don’t deal with state, is it really a distributed system?

Kubernetes - Baby steps

Kubernetes has evolved on managing state as it has matured:

- PetSets in Kubernetes 1.3
- StatefulSets in Kubernetes 1.5 (beta)
- StatefulSets in Kubernetes 1.9 (GA)

Kubernetes - StatefulSets

instaclustr

- The workload API object used to manage stateful applications
- StatefulSet maintains a sticky identity for each of their Pods.
- StatefulSets are managed by a controller like any other Kubernetes component.
- You use StatefulSets when you need any of the following:
  - Stable, unique network identifiers.
  - Stable, persistent storage.
  - Ordered, graceful deployment and scaling.
  - Ordered, graceful deletion and termination.
  - Ordered, automated rolling updates.

Kubernetes

So we now have the building blocks for managing state in Kubernetes

Kubernetes

instaclustr

Let’s take a step back

Putting it all together

- Containers - Build, run and deploy things easier
- Kubernetes - Run, manage, operate things easier
- Kinda hard to run stateful things, but the fundamentals are there.

So... what about Cassandra?

instaclustr

- As Kubernetes becomes a defacto orchestration API, people will (and do) want to run Cassandra on Kubernetes
- It’s easy to get started, harder to run.
- Running in Docker?
- For Instaclustr, Kubernetes does a lot of what we had to do in the past
  - It abstracts the environment we run in
  - Let’s us focus on doing cool Cassandra things
  - Less focus on doing boring cloud things

Introducing Cassandra-operator

instaclustr

- Let’s build something that runs and **operates** Cassandra in Kubernetes
- Cassandra-as-a-Service on top of Kubernetes
- Instaclustr in a box

Introducing Cassandra-operator

- Let’s build something that runs and **operates** Cassandra in Kubernetes
- Cassandra-as-a-Service on top of Kubernetes
- Instaclustr in a box
    - **Open Source!**

Introducing Cassandra-operator

https://github.com/benbromhead/cassandra-operator

Introducing Cassandra-operator

And of course supporting Docker images:

https://github.com/instaclustr/cassandra-docker

Awesome!...what does it get me?

- Operations “free” Cassandra
- Consistent, reproducible environments
- Best practices are built in
- Let’s your team focus on what matters

What is an operator?

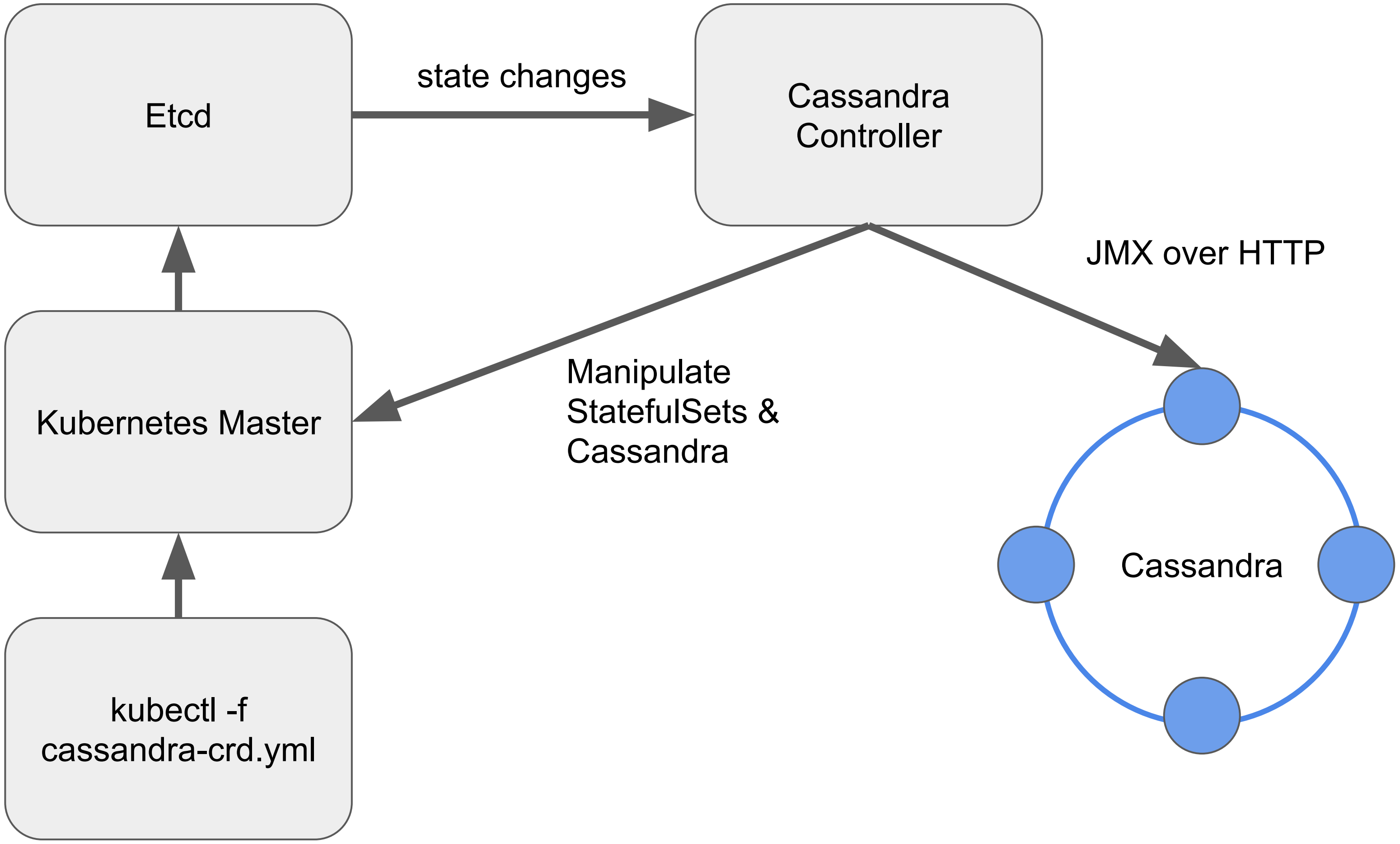
A Kubernetes operator consists of two components:

- A controller
- A Custom Resource Definition

How does it work?

- A custom resource definition (CRD) allows end users to create “Cassandra” objects in Kubernetes.
  - Contains configuration options for Cassandra (e.g. node count, jvm tuning options).
- The Cassandra controller listens to state changes on the Cassandra custom resource definition.
- Modifies StatefulSets to match the requirements specified in the Cassandra CRD.

How does it work?



Where to get it

instaclustr

- Get it on github
- Pull requests accepted
- See https://github.com/benbromhead/cassandra-operator/ROADMAP.md for current and future features

instaclustr

Questions?

Ben Bromhead
CTO
ben@instaclustr.com