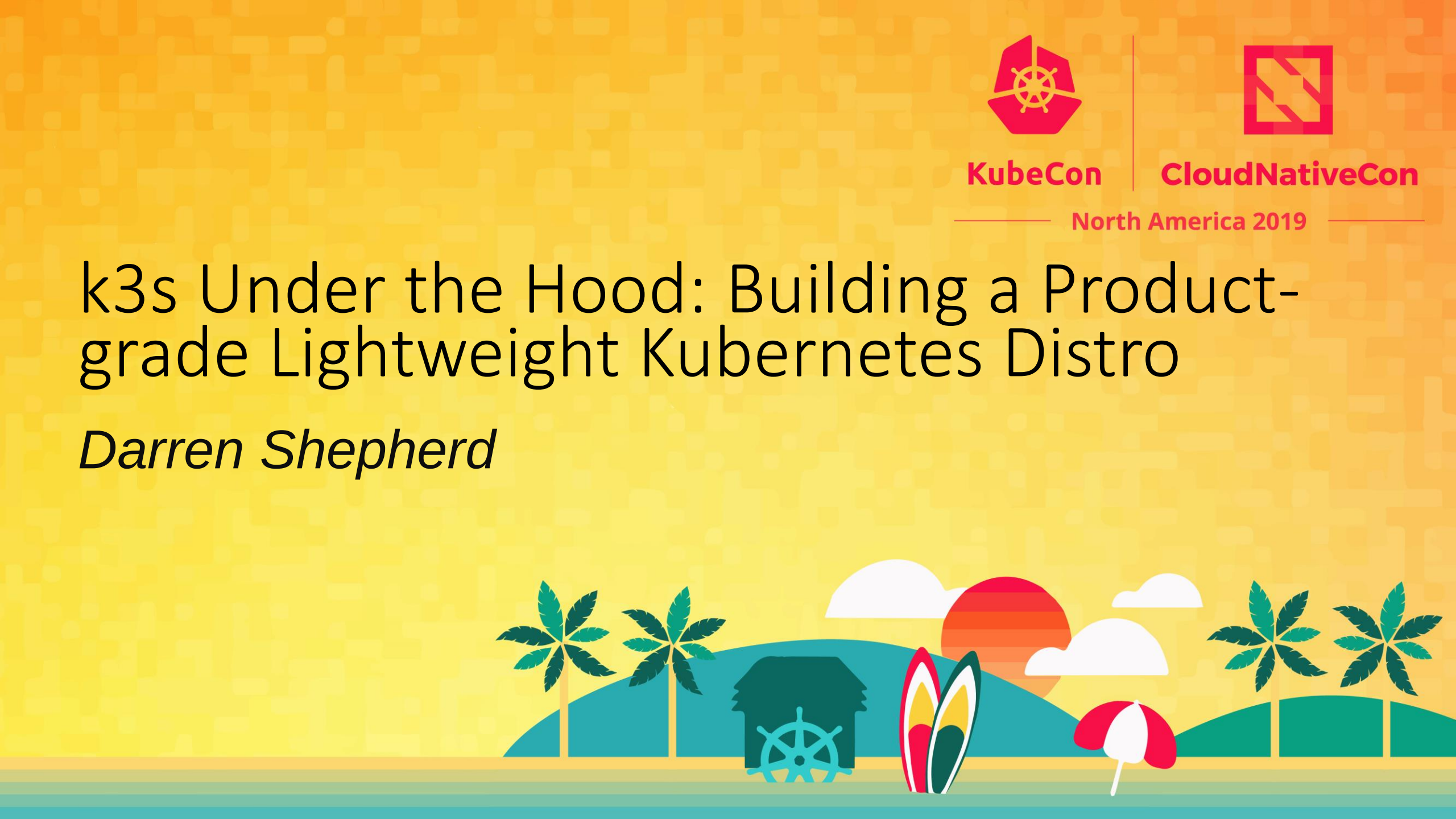


k3s Under the Hood: Building a Product-grade Lightweight Kubernetes Distro

Darren Shepherd

Darren Shepherd - Speaker

- Chief Architect and Co-Founder at Rancher Labs
- @ibuildthecloud on
- Creator of k3s (and many other Rancher projects)
- Been doing cloud orchestration for 15+ years.

What is k3s

Lightweight Kubernetes

- Binary Size (~50mb)
- Memory Size (~300mb)
- Cognative Load (🤯)
- Perfect for the Edge
- Used just about everywhere

Designed for production

- Fully CNCF Certified
- Secure by default
- Best practice defaults

Kubernetes Distribution

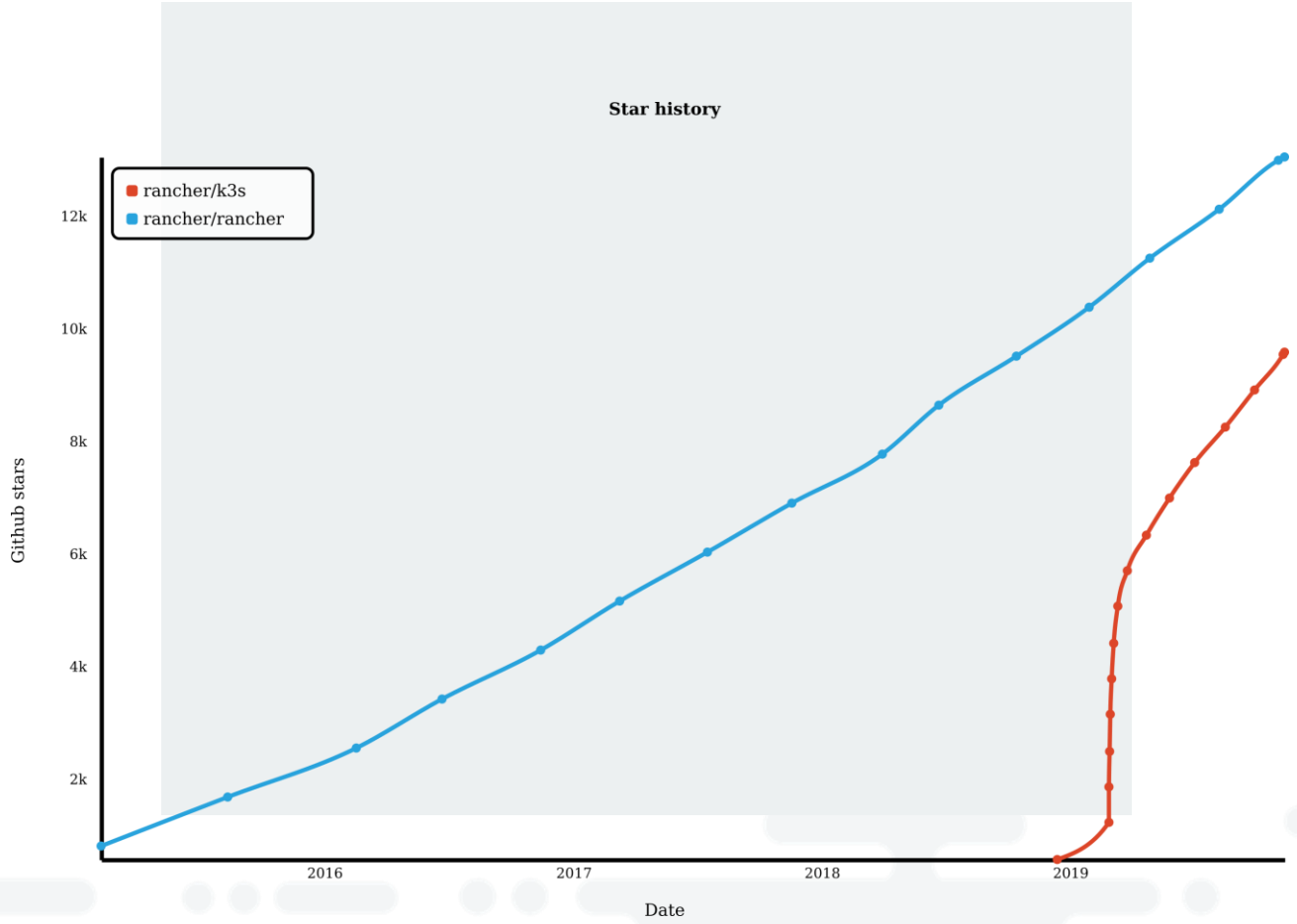
- Opinionated
- Complete
- Strong focus on simplicity

Short History of k3s

1. Total fluke
2. Came from developing Rio (hey check out https://rio.io)
3. Wanted to make running k8s simple so embedded k8s in Rio
4. Figured out Rio was the easiest way to run k8s
5. The internet somehow saw what I was doing and liked it
6. Spun out into separate project
7. Turns out edge is a perfect use case and massive market

Vanity Metrics

- Initial Release: v0.1.0 Feb 26 (9 months ago)
- 9k+ GitHub Stars
- 300k+ Binary Downloads
- 700k+ Image Pulls
- 800k+ Nodes launched
  - *based on pull count of rancher/klipper-lb



Deployment Architecture

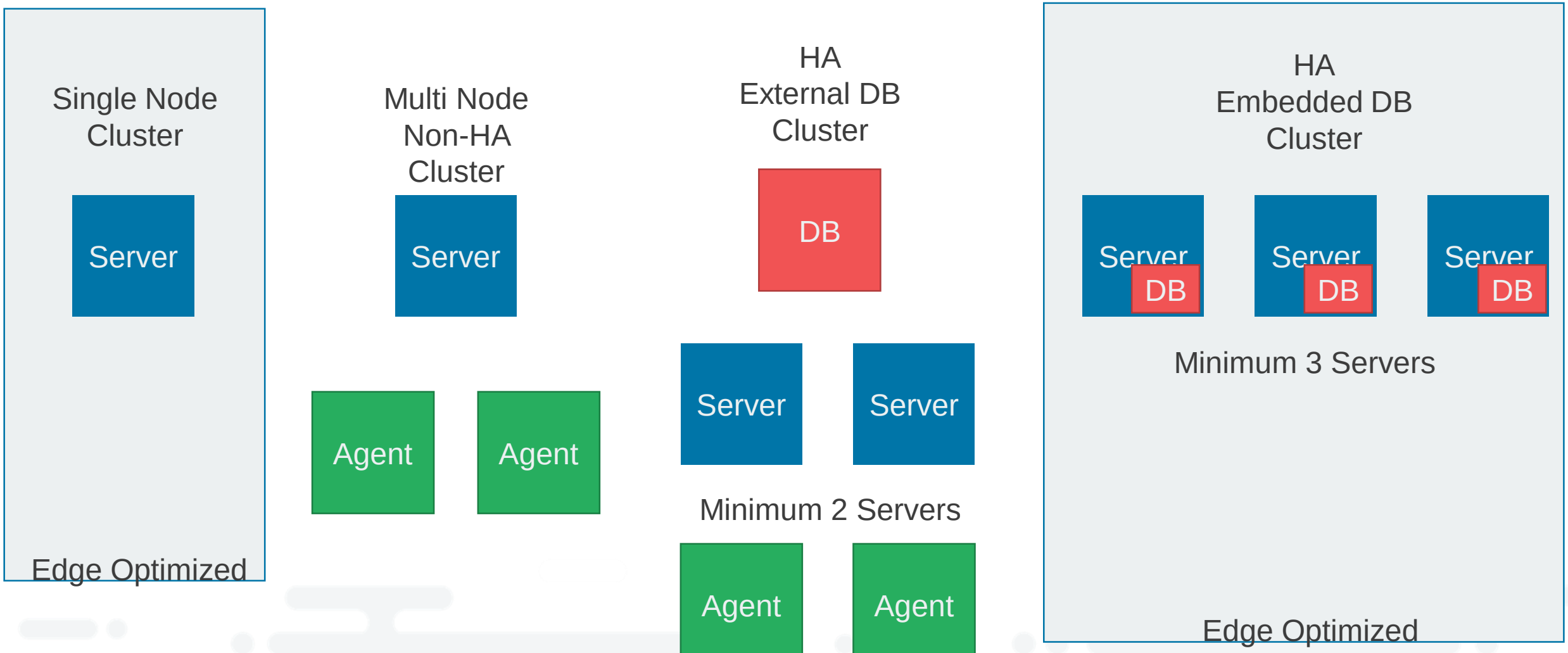
Server

Runs
- Control plane (api server, controller-manager, scheduler)
- "master worker" (kubelet)

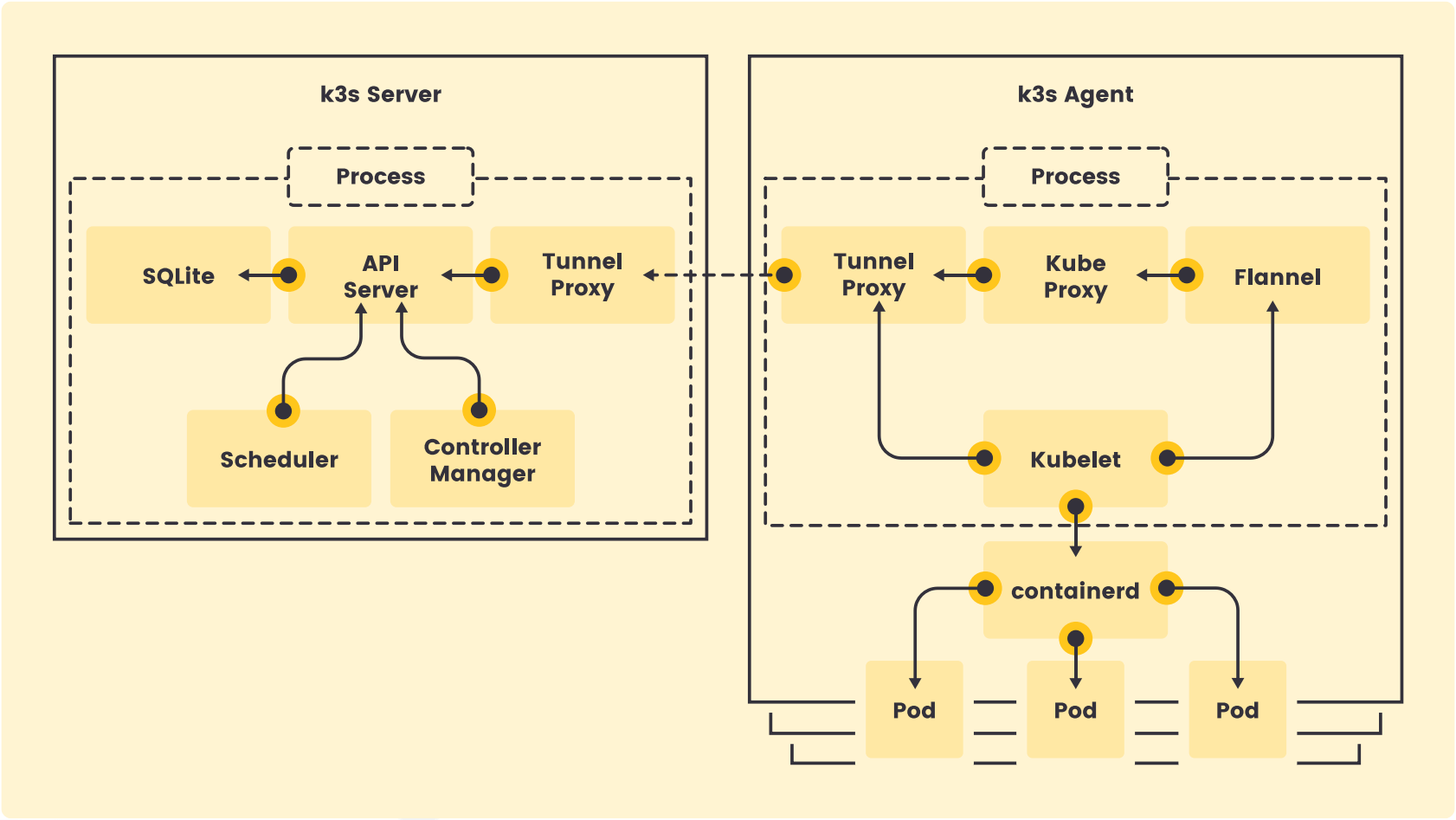
Agent

Runs
- "worker" (kubelet)

Deployment Architectures



Process Architecture



Server and Agent process are combined into one when ran on the same machine

Breakdown

1. How Kubernetes is modified
2. What we develop and added
3. What is packaged

Kubernetes Modifications

Not a fork and much less changes than you'd think.

A lot of effort has been put into ensuring longterm upstream support.

k3s - v0.1.0 over 3 million lines changed

k3s - v0.11.0 less than 1000

Kubernetes Modifications

- Add rootless support (all credit goes to Akihiro Suda from NTT)
- Drop all third-party storage drivers (CSI is supported and preferred)
- Drop cloud provider SDKs (adds a lot of bloat to binary)
- Backported Bug fixes
- Changed to allow embedding k8s

Things we've developed

1. Reverse Tunnel Proxy for kubelet – kubelet only makes outbound connections, makes network firewalling easier
2. Kine – etcd shim that translates etcd api to sqlite, Postgres, Mysql, dqlite
3. dqlite integration – secure transport, discovery, handle failure scenarios
4. Busybox userspace – iptables, du, find, socat, ipset, etc.
5. k3s binary archive – k3s is really a self extracting archive
6. Certificate generation and rotation
7. Server bootstrap

Things we've developed continued

1. Manifest auto deploy
2. Image auto deploy
3. Integrated helm chart management (helmchart CRD and controller)
4. Server bootstrap
5. Kubelet client side load balancer
6. k3s cloud provider – manage external IP
7. Embedded host port based Service Load Balancer
8. Local storage provider

What's included

1. Everything embedded can be disabled and replaced
2. Flannel is compiled into agent (vxlan, ipsec, wireguard support)
3. Network Policy Controller from KubeRouter
4. containerd
5. runc
6. cni binaries
7. Strongswan

What's included continued

1. Traefik for Ingress
2. CoreDNS
3. Metrics Server
4. Busybox user space (k3s can run with an empty root and only proc, sys, dev)
5. And all this works out of the box on amd64, arm64, armhf

Kine (Kine is not etcd)

1. Can be ran standalone so any k8s (not just k3s) can use Kine
2. Implements a subset of etcd API (not usable at all for general purpose etcd)
3. Translates etcd TX calls into the desired API (Create, Update, Delete)
4. Backend drivers for dqlite, sqlite, Postgres, MySQL
5. https://github.com/rancher/kine
6. Performance seems sufficient if not better than etcd – tested 1000's of nodes with no issue.
7. In theory can outperform etcd for k8s in every way.

Where is k3s headed

1. Users have asked for k3s to run everywhere
2. Cloud Provider support
3. WSL2 First Class
4. Windows (POC k3s.exe works)
5. Continue to build ecosystem of easy to install packages on k3s

Interesting k3s projects

1. k3d – k3s in docker, following the style of kind
2. k3sup – provision k3s clusters remoting ("from Zero to KUBECONFIG in < 1 min")