Kubernetes Overview

What is Kubernetes

Kubernetes is a platform for managing containerized distributed applications based on the microservices architecture.

Planet Scale

Never Outgrow

Run Anywhere

Core Ideas of Microservices Architectures

- Decoupled Services
- Lightweight Communication
- Built around business capabilities
- Independently changeable and deployable
- Polyglot
- Resilience
  - Immutable Artifacts
  - Designed To Fail
  - Elastic in Scale

Microservices are Designed To Fail

Availabiliy = MTBF / (MTBF + MTTR)

Focus on minimizing MTTR instead of MTBF

Kubernetes History

Based on 15 years of experience of running production workloads at Google. "Borg, Omega, and Kubernetes" March 2016

Coordination and consensus via Paxos (etcd). "Consensus in the Cloud: Paxos Systems Demystified" February 2016

Kubernetes Architecture

Core Principles

Portable

- public, private, hybrid, multi-cloud

Extensible

- modular, pluggable, hookable, composable

Self-healing

- auto-placement, auto-restart, auto-replication, auto-scaling

Kubernetes Features

- co-locating helper processes, facilitating composite applications and preserving the one-application-per-container model,
- mounting storage systems,
- distributing secrets,
- application health checking,
- replicating application instances,
- horizontal auto-scaling,

Kubernetes Features continued

- naming and discovery,
- load balancing,
- rolling updates,
- resource monitoring,
- log access and ingestion,
- support for introspection and debugging, and
- identity and authorization.

Community

4k+ Commits in 1.4 ~35k total

+50% Unique Contributors

Top 0.01% of all Github Projects

1200+ External Projects Based on K8s

Companies Contributing

REDAPT, IBM, CoreOS, intel, MIRANTIS, HUAWEI, vmware, redhat, MESOSPHERE, HITACHI Inspire the Next, bitnami, FUJITSU, hp

Companies Using

VIACOM, ebay, box, Goldman Sachs, whitepages, SOUNDCLOUD, The New York Times, wepay, apprenda, zulily

Why Kubernetes

- Mature Platform
  - 15 years of experience at Google
  - Best-of-breed ideas and practices from the community
- Vibrant and active community that is inclusive
- Primitives instead of Frameworks
- Low lock-in to single technologies
  - Through abstraction of container engine, networking, storage