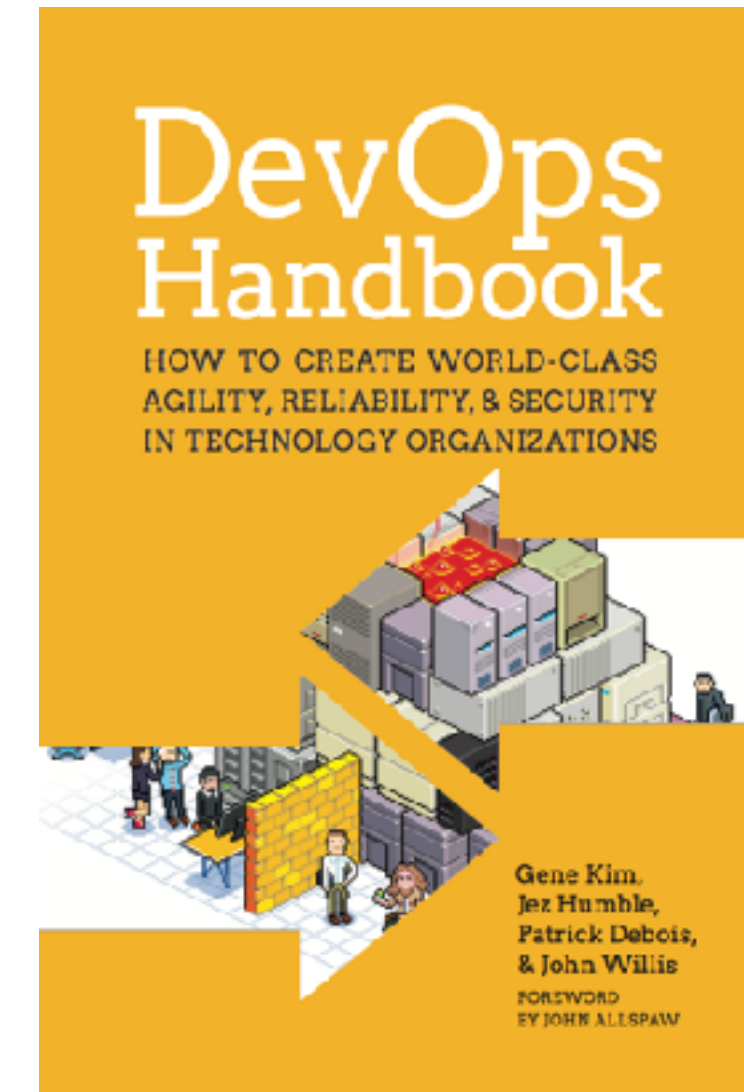




# **Next Generation Infrastructure for Managers**



**John Willis**  
**@botchagalupe**

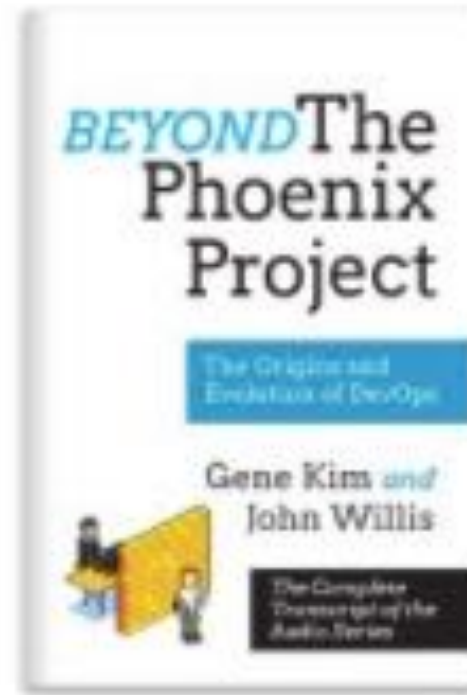


**DEVOPSDAYS**



<https://github.com/botchagalupe/my-presentations>

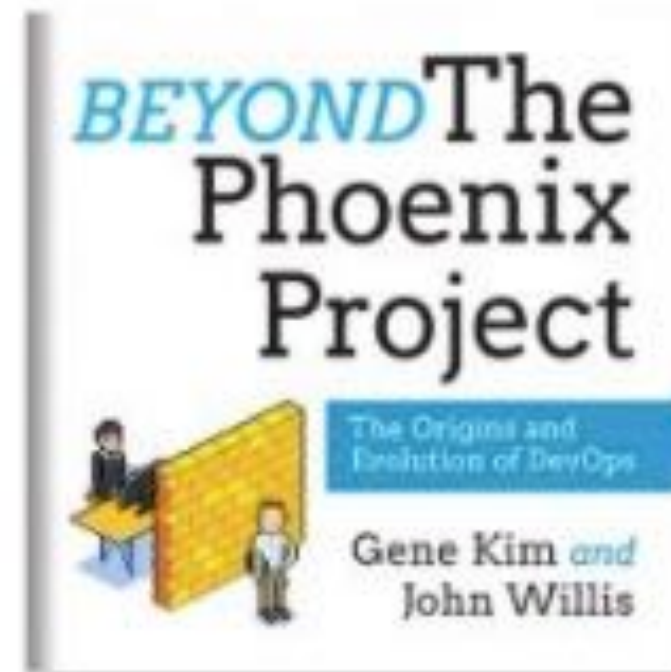




## BEYOND THE PHOENIX PROJECT

**New!**

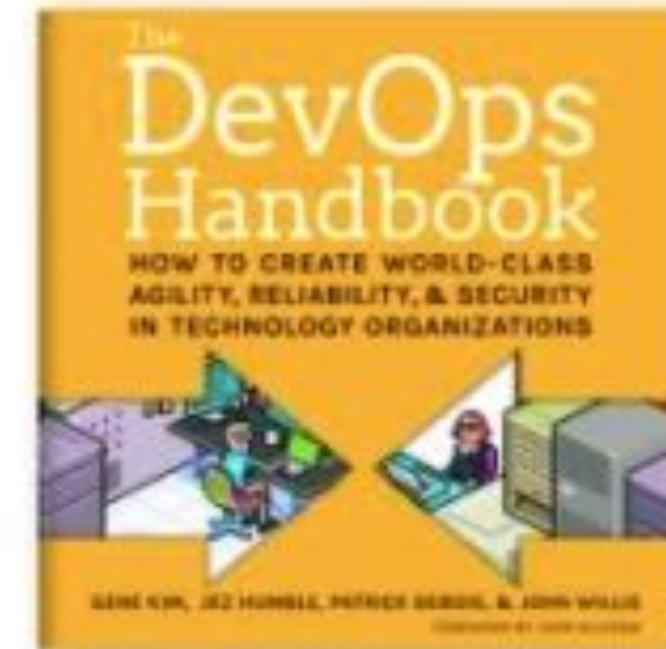
In this transcript of the audio series, Gene Kim and John Willis present a nine-part discussion that includes an oral history of the DevOps movement, as well as discussions around pivotal figures and philosophies that DevOps draws upon, from Goldratt to Deming; from Lean to safety culture to learning organizations. The book is a great way for listeners to take an even



## BEYOND THE PHOENIX PROJECT (AUDIOBOOK)

**New!**

Gene Kim and John Willis present this nine-part series that includes an oral history of the DevOps movement, as well as discussion around pivotal figures and philosophies that DevOps draws upon, from Goldratt to Deming; from Lean to Safety Culture to Learning



## DEVOPS HANDBOOK (AUDIOBOOK)

**New!**

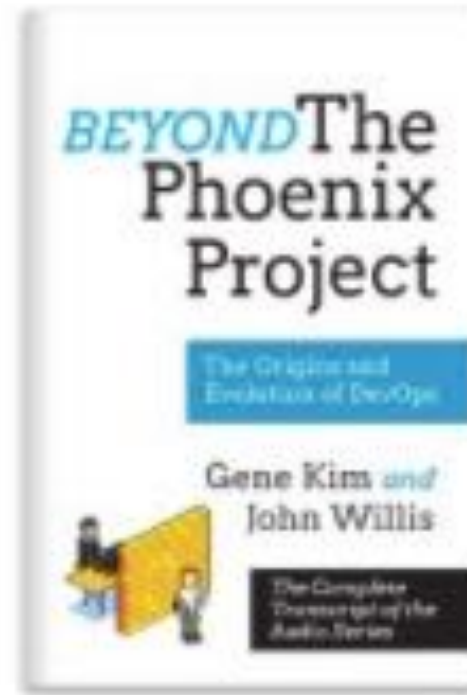
Increase profitability, elevate work culture, and exceed productivity goals through DevOps practices. More than ever, the effective management of technology is critical for business competitiveness. This non-fiction follow-up to *The Phoenix Project* shows leaders



## MYTHBUSTING DEVOPS IN THE ENTERPRISE

Anyone leading a company through a DevOps transformation will encounter minor to significant internal skepticism or lack concrete experience. This document lists the most common leadership and cultural traps and provides high-level reassurance and evidence that DevOps practices are generally applicable and plausibly successful in enterprise environments.

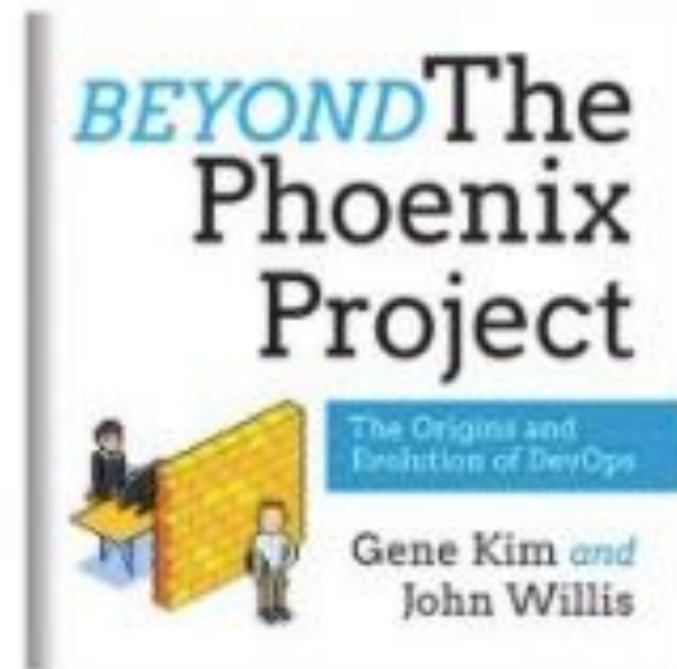




## BEYOND THE PHOENIX PROJECT

**New!**

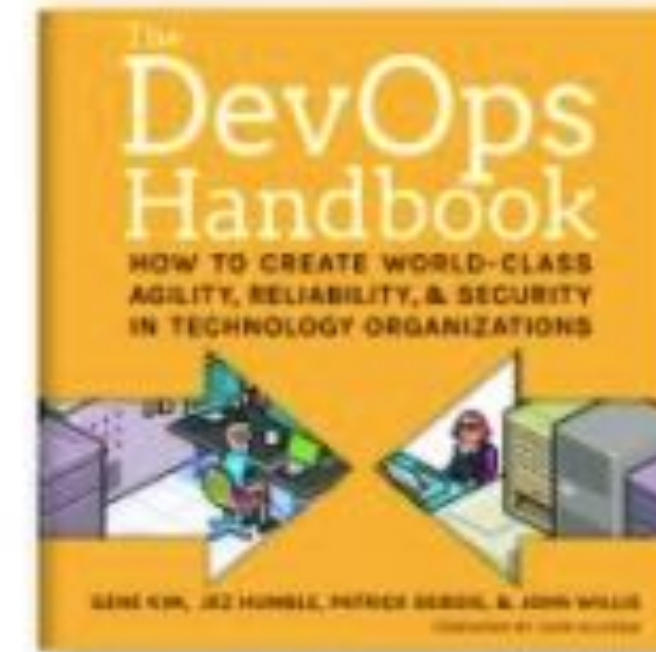
In this transcript of the audio series, Gene Kim and John Willis present a nine-part discussion that includes an oral history of the DevOps movement, as well as discussions around pivotal figures and philosophies that DevOps draws upon, from Goldratt to Deming; from Lean to safety culture to learning organizations. The book is a great way for listeners to take an even



## BEYOND THE PHOENIX PROJECT (AUDIOBOOK)

**New!**

Gene Kim and John Willis present this nine-part series that includes an oral history of the DevOps movement, as well as discussion around pivotal figures and philosophies that DevOps draws upon, from Goldratt to Deming; from Lean to Safety Culture to Learning



## DEVOPS HANDBOOK (AUDIOBOOK)

**New!**

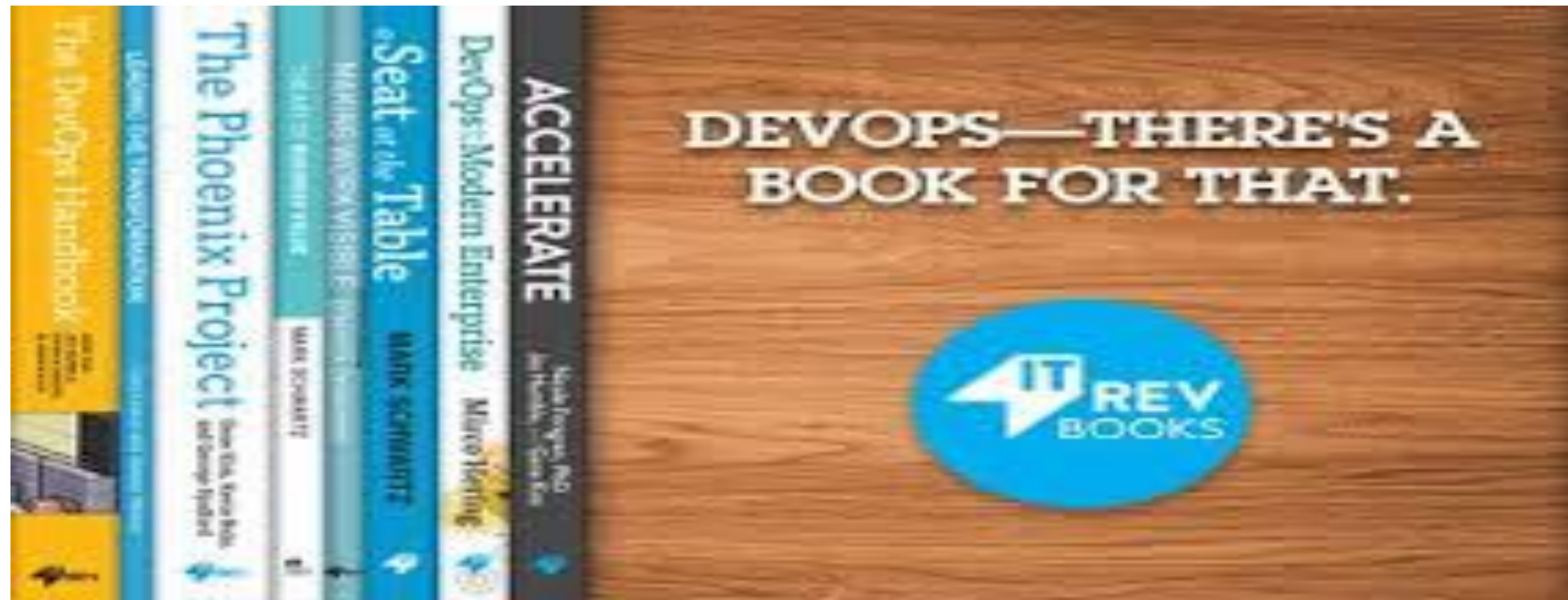
Increase profitability, elevate work culture, and exceed productivity goals through DevOps practices. More than ever, the effective management of technology is critical for business competitiveness. This non-fiction follow-up to *The Phoenix Project* shows leaders



## MYTHBUSTING DEVOPS IN THE ENTERPRISE

Anyone leading a company through a DevOps transformation will encounter minor to significant internal skepticism or lack concrete experience. This document lists the most common leadership and cultural traps and provides high-level reassurance and evidence that DevOps practices are generally applicable and plausibly successful in enterprise environments.





# DevSecOps Handbook (2019)

Shannon Lietz  
James Wicket  
Ernest Mueller  
John Willis









**Containers**

**Kubernetes**

**Whereof what's past is prologue**



**Containers**

**Kubernetes**

**Whereof what's past is prologue**

**Kubernetes is a container management system**





**Containers**

**Kubernetes**

**Whereof what's past is prologue**

**Kubernetes is a container management ~~system~~**





**Containers**

**Kubernetes**

**Whereof what's past is prologue**

**Kubernetes is a container management ~~system~~**

**Kubernetes is a container management platform**





**Containers**

**Kubernetes**

**Whereof what's past is prologue**

**Kubernetes is a container management ~~system~~**

**Kubernetes is a ~~container~~ management platform**





**Containers**

**Kubernetes**

**Whereof what's past is prologue**

**Kubernetes is a container management ~~system~~**

**Kubernetes is a ~~container~~ management platform**

**Kubernetes is a services management platform**







# In Scope

- **Foundations**
- **Container Ecosystem**
- **Service Mesh**
- **Kubernetes API Extensibility**

# Out of Scope

- **Introduction to Containers/**
- **Introduction to Kubernetes**
- **Container/K8 Storage**
- **Container/K8 Networking**
- **Container/K8 Ecosystem Tools**
- **Serverless (knative)**



# Foundations




# Open Container Initiative


Creating open standards around container technology

<https://www.opencontainers.org> [dev@opencontainers.org](mailto:dev@opencontainers.org)

 **Repositories** 14

 **People** 12


 **Projects** 0



## Grow your team on GitHub

GitHub is home to over 28 million developers working together. Join them to grow your own development teams, manage permissions, and collaborate on projects.

[Sign up](#)

 Dismiss

## Pinned repositories

### runc

CLI tool for spawning and running containers according to the OCI specification

 Go  5.1k  923

### runtime-spec

OCI Runtime Specification

 Go  1.2k  255

### image-spec

OCI Image Format

 Go  479  110

### distribution-spec

OCI Distribution Specification

 Makefile  52  20

### go-digest

Common digest package used across the container ecosystem

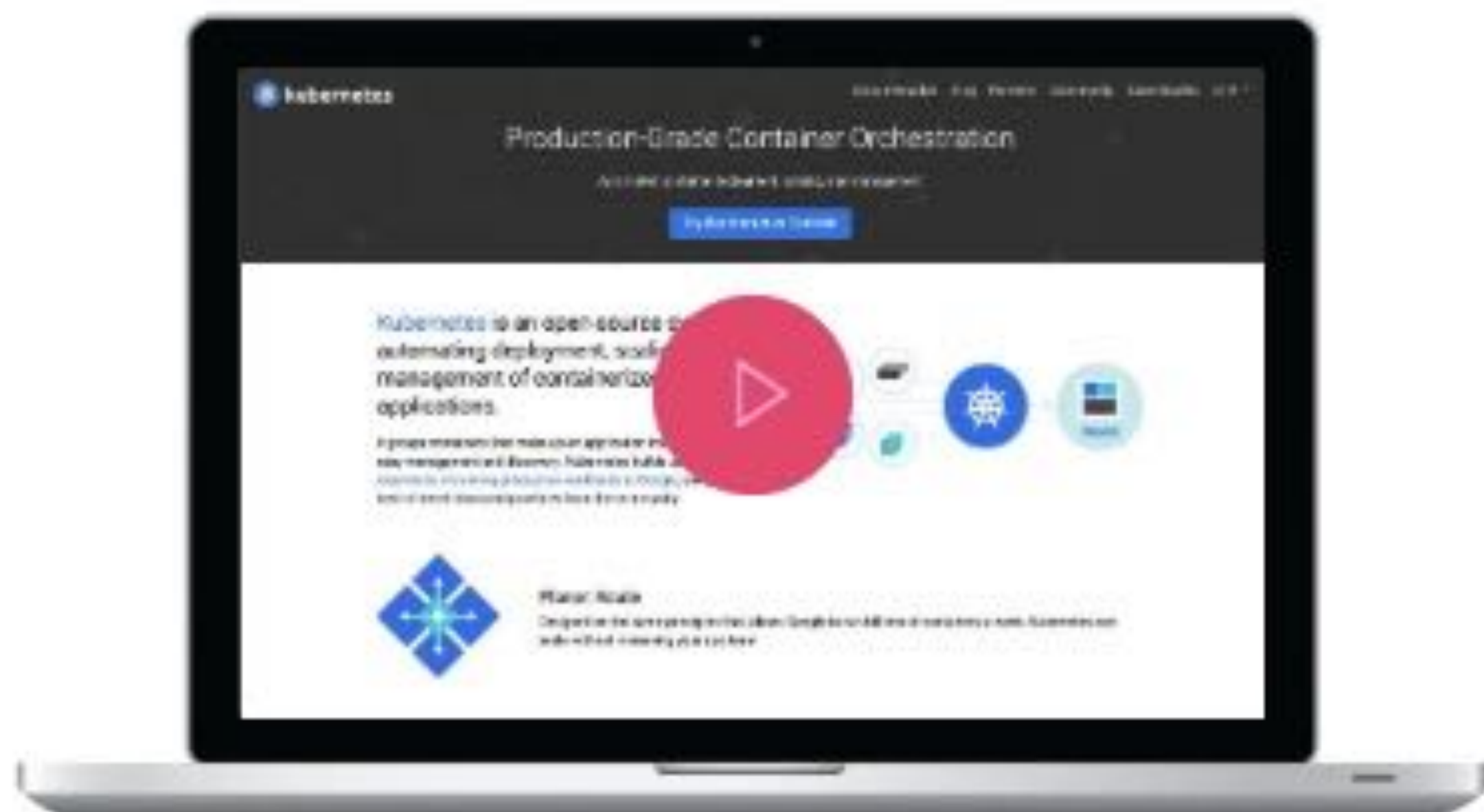
 Go  37  18

### tob

Technical Oversight Board (TOB)

 32  21





# kubernetes

































Kubernetes is the world's most popular container-orchestration platform and the first CNCF project. Kubernetes helps users build, scale and manage modern applications and their dynamic lifecycles. First developed at Google, Kubernetes now counts more than 2,300 contributors and is used by some of the world's most-innovative companies, across a wide range of industries. The cluster scheduler capability lets developers build cloud native applications, while focusing on code rather than ops. Kubernetes future-proofs application development and infrastructure management on-premises or in the cloud, without vendor or cloud-provider lock-in.

[VISIT PROJECT WEBSITE](#)























DevStats projects:











Graduated		Incubating							
<a href="#">Kubernetes</a>	<a href="#">Prometheus</a>	<a href="#">OpenTracing</a>	<a href="#">Fluentd</a>	<a href="#">gRPC</a>	<a href="#">containerd</a>	<a href="#">rkt</a>	<a href="#">CNI</a>	<a href="#">Envoy</a>	<a href="#">Jaeger</a>
									
Incubating									
<a href="#">Notary</a>	<a href="#">TUF</a>	<a href="#">Vitess</a>	<a href="#">CoreDNS</a>	<a href="#">NATS</a>	<a href="#">Linkerd</a>	<a href="#">Helm</a>	<a href="#">Rook</a>	<a href="#">OPA</a>	<a href="#">SPIFFE</a>
									
Sandbox									CNCF
<a href="#">SPIRE</a>	<a href="#">CloudEvents</a>	<a href="#">Telepresence</a>	<a href="#">Harbor</a>	<a href="#">OpenMetrics</a>	<a href="#">TiKV</a>	<a href="#">Cortex</a>	<a href="#">Buildpacks</a>	<a href="#">Falco</a>	<a href="#">All CNCF</a>
									



DevStats projects:











<b>Graduated</b>		<b>Incubating</b>							
<a href="#">Kubernetes</a>	<a href="#">Prometheus</a>	<a href="#">OpenTracing</a>	<a href="#">Fluentd</a>	<a href="#">gRPC</a>	<a href="#">containerd</a>	<a href="#">rkt</a>	<a href="#">CNI</a>	<a href="#">Envoy</a>	<a href="#">Jaeger</a>
									











<b>Incubating</b>								<b>Sandbox</b>	
<a href="#">Notary</a>	<a href="#">TUF</a>	<a href="#">Vitess</a>	<a href="#">CoreDNS</a>	<a href="#">NATS</a>	<a href="#">Linkerd</a>	<a href="#">Helm</a>	<a href="#">Rook</a>	<a href="#">OPA</a>	<a href="#">SPIFFE</a>
									











<b>Sandbox</b>									<b>CNCF</b>
<a href="#">SPIRE</a>	<a href="#">CloudEvents</a>	<a href="#">Telepresence</a>	<a href="#">Harbor</a>	<a href="#">OpenMetrics</a>	<a href="#">TiKV</a>	<a href="#">Cortex</a>	<a href="#">Buildpacks</a>	<a href="#">Falco</a>	<a href="#">All CNCF</a>
									



DevStats projects:











<b>Graduated</b>		<b>Incubating</b>							
<a href="#">Kubernetes</a>	<a href="#">Prometheus</a>	<a href="#">OpenTracing</a>	<a href="#">Fluentd</a>	<a href="#">gRPC</a>	<a href="#">containerd</a>	<a href="#">rkt</a>	<a href="#">CNI</a>	<a href="#">Envoy</a>	<a href="#">Jaeger</a>
									











<b>Incubating</b>								<b>Sandbox</b>	
<a href="#">Notary</a>	<a href="#">TUF</a>	<a href="#">Vitess</a>	<a href="#">CoreDNS</a>	<a href="#">NATS</a>	<a href="#">Linkerd</a>	<a href="#">Helm</a>	<a href="#">Rook</a>	<a href="#">OPA</a>	<a href="#">SPIFFE</a>
									











<b>Sandbox</b>									<b>CNCF</b>
<a href="#">SPIRE</a>	<a href="#">CloudEvents</a>	<a href="#">Telepresence</a>	<a href="#">Harbor</a>	<a href="#">OpenMetrics</a>	<a href="#">TiKV</a>	<a href="#">Cortex</a>	<a href="#">Buildpacks</a>	<a href="#">Falco</a>	<a href="#">All CNCF</a>
									



DevStats projects:











<b>Graduated</b>		<b>Incubating</b>							
<a href="#">Kubernetes</a>	<a href="#">Prometheus</a>	<a href="#">OpenTracing</a>	<a href="#">Fluentd</a>	<a href="#">gRPC</a>	<a href="#">containerd</a>	<a href="#">rkt</a>	<a href="#">CNI</a>	<a href="#">Envoy</a>	<a href="#">Jaeger</a>
									











<b>Incubating</b>								<b>Sandbox</b>	
<a href="#">Notary</a>	<a href="#">TUF</a>	<a href="#">Vitess</a>	<a href="#">CoreDNS</a>	<a href="#">NATS</a>	<a href="#">Linkerd</a>	<a href="#">Helm</a>	<a href="#">Rook</a>	<a href="#">OPA</a>	<a href="#">SPIFFE</a>
									











<b>Sandbox</b>									<b>CNCF</b>
<a href="#">SPIRE</a>	<a href="#">CloudEvents</a>	<a href="#">Telepresence</a>	<a href="#">Harbor</a>	<a href="#">OpenMetrics</a>	<a href="#">TiKV</a>	<a href="#">Cortex</a>	<a href="#">Buildpacks</a>	<a href="#">Falco</a>	<a href="#">All CNCF</a>
									



DevStats projects:

<b>Graduated</b>		<b>Incubating</b>							
<a href="#">Kubernetes</a>	<a href="#">Prometheus</a>	<a href="#">OpenTracing</a>	<a href="#">Fluentd</a>	<a href="#">gRPC</a>	<a href="#">containerd</a>	<a href="#">rkt</a>	<a href="#">CNI</a>	<a href="#">Envoy</a>	<a href="#">Jaeger</a>
									

<b>Incubating</b>								<b>Sandbox</b>	
<a href="#">Notary</a>	<a href="#">TUF</a>	<a href="#">Vitess</a>	<a href="#">CoreDNS</a>	<a href="#">NATS</a>	<a href="#">Linkerd</a>	<a href="#">Helm</a>	<a href="#">Rook</a>	<a href="#">OPA</a>	<a href="#">SPIFFE</a>
									

<b>Sandbox</b>									<b>CNCF</b>
<a href="#">SPIRE</a>	<a href="#">CloudEvents</a>	<a href="#">Telepresence</a>	<a href="#">Harbor</a>	<a href="#">OpenMetrics</a>	<a href="#">TiKV</a>	<a href="#">Cortex</a>	<a href="#">Buildpacks</a>	<a href="#">Falco</a>	<a href="#">All CNCF</a>
									



# Container Ecosystem

# Decoupling the Container Ecosystem Terminology

- **Container Runtimes**
- **Container Engines**
- **Container Orchestration**





# Container Runtime

- **Runc**
- Runlxc
- Rkt
- Runv
- Railcar
- Kata Containers





# Container Engines

- **Container Runtime Interface (CRI)**
- **ContainerD**
  - Docker
  - GKE
- **CRI-O**
  - Openshift





# Docker (engine)

- **Moby**
- **Docker Engine Enterprise**
- **Docker Engine Community**



# As a Service Cloud Engine's

- ECS (Amazon)
- ACS (Azure)
- GKE (Google \*)



**NOTE: ACS and GKE are now offered on-prem**



# Container Orchestration

- ◉ **Kubernetes**
- ◉ Swarm
- ◉ Mesos



# Container Orchestration

- **Kubernetes**
- Swarm
- Mesos
- Nomad (Hashicorp)





# Kubernetes The Hard Way

---

This tutorial walks you through setting up Kubernetes the hard way. This guide is not for people looking for a fully automated command to bring up a Kubernetes cluster. If that's you then check out [Google Kubernetes Engine](#), or the [Getting Started Guides](#).

Kubernetes The Hard Way is optimized for learning, which means taking the long route to ensure you understand each task required to bootstrap a Kubernetes cluster.

The results of this tutorial should not be viewed as production ready, and may receive limited support from the community, but don't let that stop you from learning!

## Target Audience

---

The target audience for this tutorial is someone planning to support a production Kubernetes cluster and wants to understand how everything fits together.

## Cluster Details

---

Kubernetes The Hard Way guides you through bootstrapping a highly available Kubernetes cluster with end-to-end encryption between components and RBAC authentication.

- [Kubernetes](#) 1.12.0
- [containerd Container Runtime](#) 1.2.0-rc.0
- [gVisor](#) 50c283b9f56bb7200938d9e207355f05f79f0d17
- [CNI Container Networking](#) 0.6.0
- [etcd](#) v3.3.9
- [CoreDNS](#) v1.2.2

**<https://github.com/kelseyhightower/kubernetes-the-hard-way>**



# CNCF Certified Kubernetes

- **Alibaba Cloud, Alibaba Cloud Container Service**
- **Apprenda, Kismatic Enterprise Toolkit (KET)**
- **Appscore, Pharmer**
- **Caicloud, Compass**
- **Canonical, Canonical Distribution of Kubernetes**
- **Cisco Systems, Cisco Container Platform**
- **Cloud Foundry Foundation, Cloud Foundry Container Runtime**
- **CoreOS, Tectonic**
- **DaoCloud, DaoCloud Enterprise**
- **Docker, Docker Enterprise Edition**
- **Google, Google Kubernetes Engine**
- **Heptio, AWS-Quickstart**
- **Huawei, Huawei Cloud Container Engine**
- **IBM, IBM Cloud Container Service and IBM Cloud Private**
- **Loodse, Kubermatic Container Engine**
- **Mesosphere, Kubernetes on DC/OS**
- **Microsoft, Azure ACS-Engine**
- **Mirantis, Mirantis Cloud Platform**
- **Netease, Netease Container Service Dedicated**
- **Oracle, Oracle Container Engine, Oracle Linux**
- **Pivotal Software, Pivotal Container Service (PKS)**
- **Poseidon, Typhoon**
- **Rancher, Inc., Rancher Kubernetes**
- **Red Hat, OpenShift**
- **SAP, Cloud Platform – Gardener**
- **SUSE, SUSE CaaS (Container as a Service) Platform**
- **Samsung SDS, Kraken**
- **StackPointCloud, Stackpoint.io**
- **Tencent Cloud, Tencent Cloud Container Service**
- **VMware, Pivotal Container Service (PKS)**
- **Weaveworks, kubernetes**
- **Wise2C Technology, WiseCloud**



# Cloud (orchestration)

- **EKS (Amazon)**
- **AKS (Azure)**
- **GKE (Google)**





# Service Mesh



# Service Mesh Capabilities

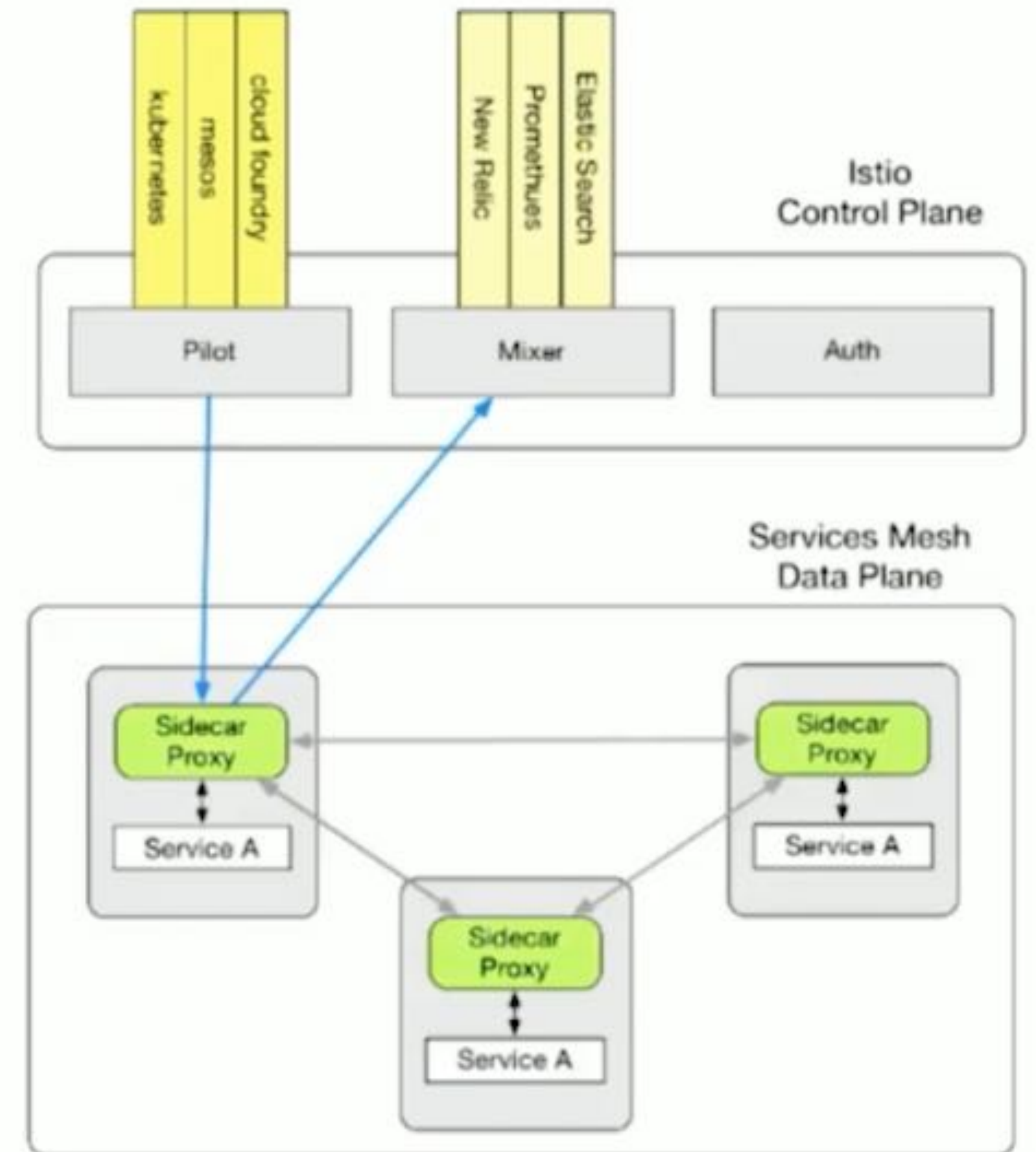
- **Observability**
- **Traffic Control**
- **Service Discovery**
- **Resilience**
- **Deployment Strategies**
- **Security**





# ISTIO Architecture

- **Control Plane**
  - Manages and configures proxies to route traffic, enforce policies, and collect telemetry.
  - Made up of three services: Pilot, Mixer, and Auth.
- **Data Plane**
  - Intercepts all inbound (ingress) and outbound (egress) network traffic.
  - Service mesh sidecar model (proxy) is an implementation of the data plane that runs as a container in a Kubernetes Pod.







# Istio Components



Data Plane

Control Plane

## Envoy

Envoy is deployed as a Sidecar in the same K8S Pod.

- Dynamic Service Discovery
- Load Balancing
- TLS Termination
- HTTP/2 and gRPC Proxies
- Circuit Breakers
- Health Checks
- Staged Rollouts with % based traffic split
- Fault Injection
- Rich Metrics

## Mixer

- Enforces access control and usage policies across service mesh and
- Collects telemetry data from Envoy and other services.
- Also includes a flexible plugin model.

## Pilot

Provides

- Service Discovery
- Traffic Management
- Routing
- Resiliency (Timeouts, Circuit Breakers, etc.)

## Galley

Provides

- Configuration Injection
- Processing and
- Distribution Component of Istio

## Citadel

Provides

- Strong Service to Service and end user Authentication with built-in Identity and credential management.
- Can enforce policies based on Service identity rather than network controls.



# **Kubernetes API Extensibility**

# Kubernetes API

- **Kubenernetes APIs allow users to specify the desired state of a Kubernetes cluster with declarative yaml or json config for a Resource.**
- **Facilitating self-healing APIs that continuously watch the state of the system.**
- **Leveraging tools that work with any Kubernetes config such as kubectl and kustomize.**
- **Integrating with Kubernetes Authz and Authn.**
- **Developers can build and publish their own Kubernetes APIs which may be installed into running clusters by cluster admins.**



# Kubernetes API Extensibility

- The two paths that support extending the API with custom resources are:
  - CustomResourceDefinition for basic CRUD needs.
  - Aggregator for a full set of Kubernetes API semantics to implement their own apiserver.

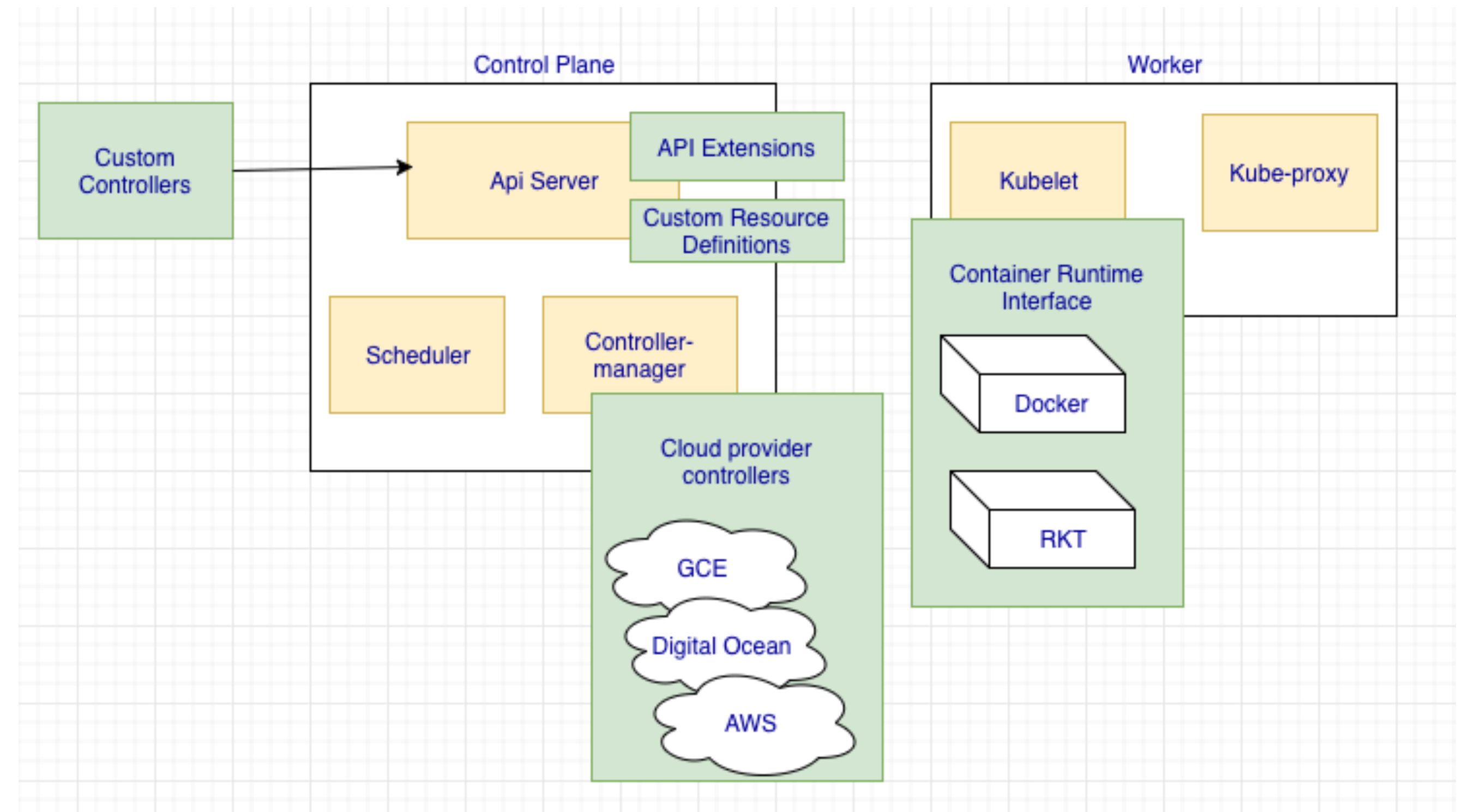
# Custom Resource and Controllers

- Custom Controllers have access to Kubernetes API via a control loop that can see the current state and make decisions in milliseconds.
- Custom Controllers can use custom rules that can watch for events to monitor a cluster, change pods, and scale endpoints of an applications.
- Useful for abstracting complexities of managing state-full applications to desired state in Kubernetes clusters.



# Custom Resource Implementation Examples

- **Tensorflow**
- **Istio - Service Mesh**
- **Calico - Overlay network**
- **Etcd Operator - etcd clusters**
- **Rook Operator - Distributed storage systems**
- **Prometheus Operator - Monitoring**



 k8s-ci-robot Merge pull request #2809 from jboyd01/update-sig-catalog-chairs

31ce047 4 days ago

100 contributors  and others

74 lines (62 sloc) | 23 KB

Raw

Blame

History



## SIGs and Working Groups

Most community activity is organized into Special Interest Groups (SIGs), time bounded Working Groups, and the [community meeting](#).

SIGs follow these [guidelines](#) although each of these groups may operate a little differently depending on their needs and workflow.

Each group's material is in its subdirectory in this project.

When the need arises, a [new SIG can be created](#)



API Machinery
Apps
Architecture
Auth
Autoscaling
AWS
Azure
Big Data
CLI
Cloud Provider
Cluster Lifecycle
Cluster Ops
Contributor Experience
Docs
GCP
IBMCloud
Instrumentation
Multicloud
Network
Node
OpenStack
PM
Release
Scalability
Scheduling
Service Catalog
Storage
Testing
UI
VMware
Windows



# Aggregated API Servers

---

## Abstract

---

We want to divide the single monolithic API server into multiple aggregated servers. Anyone should be able to write their own aggregated API server to expose APIs they want. Cluster admins should be able to expose new APIs at runtime by bringing up new aggregated servers.

## Motivation

---

- Extensibility: We want to allow community members to write their own API servers to expose APIs they want. Cluster admins should be able to use these servers without having to require any change in the core kubernetes repository.
- Unblock new APIs from core kubernetes team review: A lot of new API proposals are currently blocked on review from the core kubernetes team. By allowing developers to expose their APIs as a separate server and enabling the cluster admin to use it without any change to the core kubernetes repository, we unblock these APIs.
- Place for staging experimental APIs: New APIs can be developed in separate aggregated servers, and installed only by those willing to take the risk of installing an experimental API. Once they are stable, it is then easy to package them up for installation in other clusters.
- Ensure that new APIs follow kubernetes conventions: Without the mechanism proposed here, community members might be forced to roll their own thing which may or may not follow kubernetes conventions.

## Goal

---

- Developers should be able to write their own API server and cluster admins should be able to add them to their cluster, exposing new APIs at runtime. All of this should not require any change to the core kubernetes API server.
- These new APIs should be seamless extensions of the core kubernetes APIs (ex: they should be operated upon via kubectl).