Kubernetes at Cruise

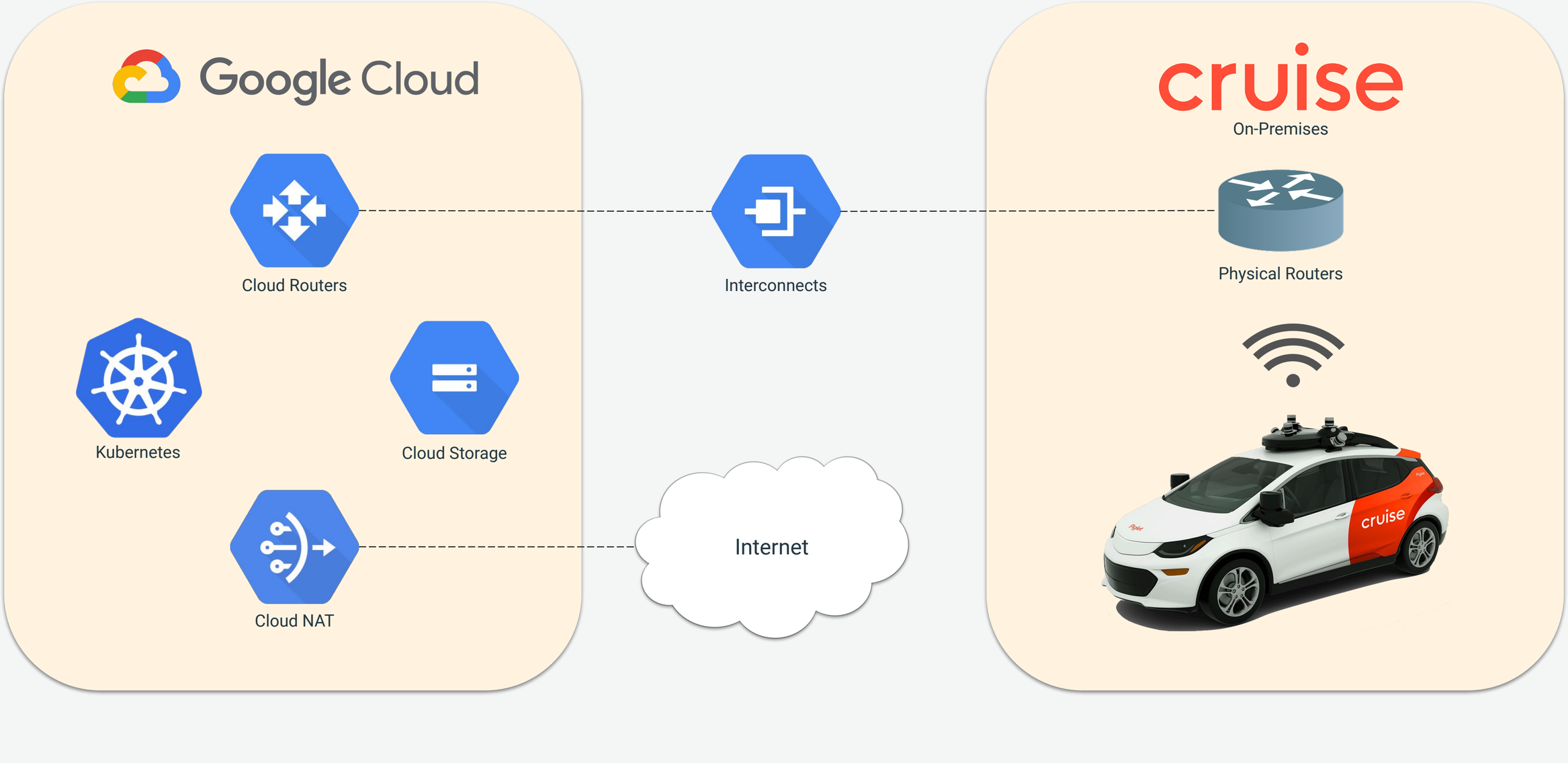
Two Years of Multitenancy

Karl Isenberg, Cruise

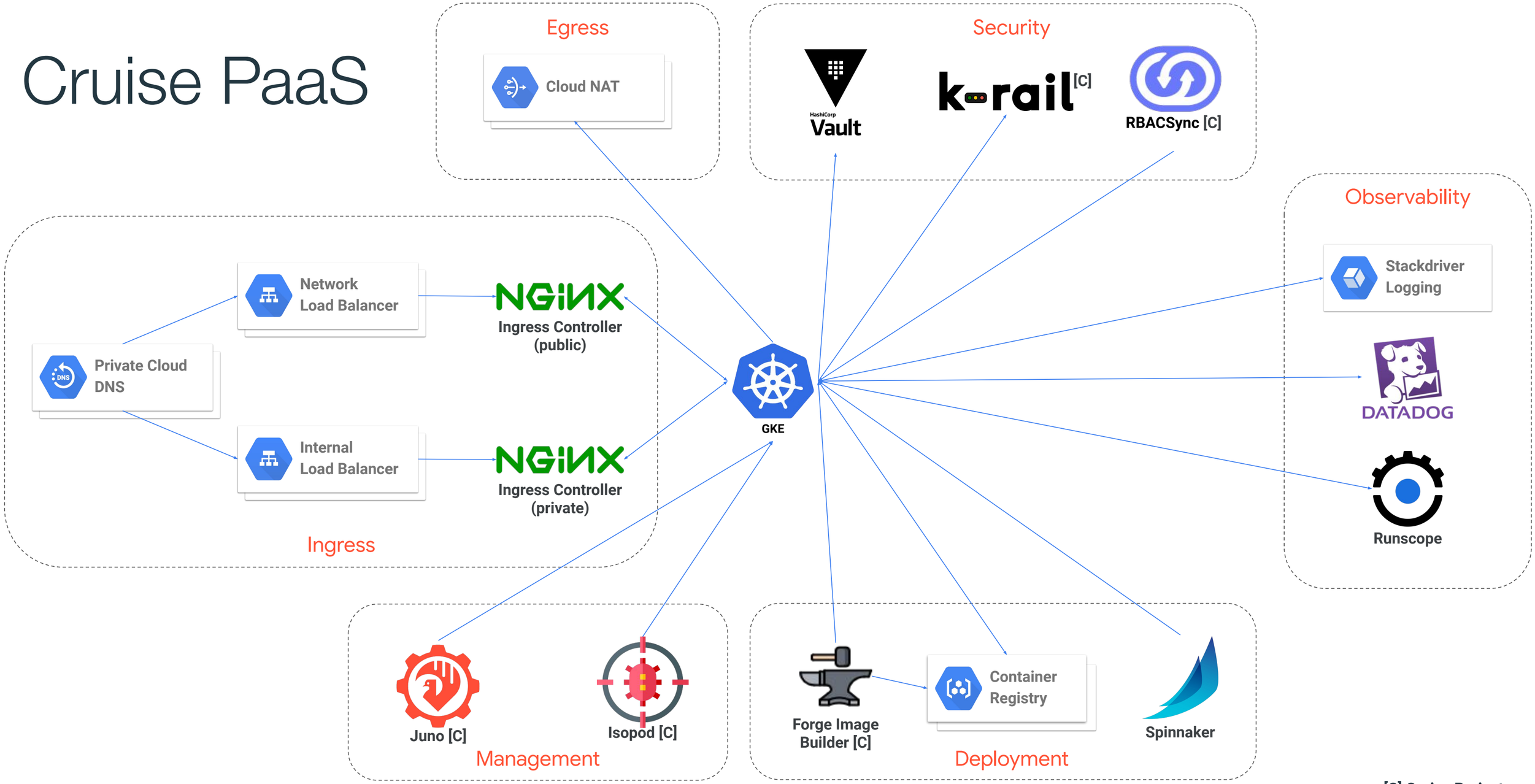
@karlkfi

Building the world's most advanced autonomous vehicles...

...and running the backend on Kubernetes.


Google Cloud
Cloud Routers
Interconnects
Kubernetes
Cloud Storage
Cloud NAT
Internet
cruise
On-Premises
Physical Routers
cruise

Cruise PaaS

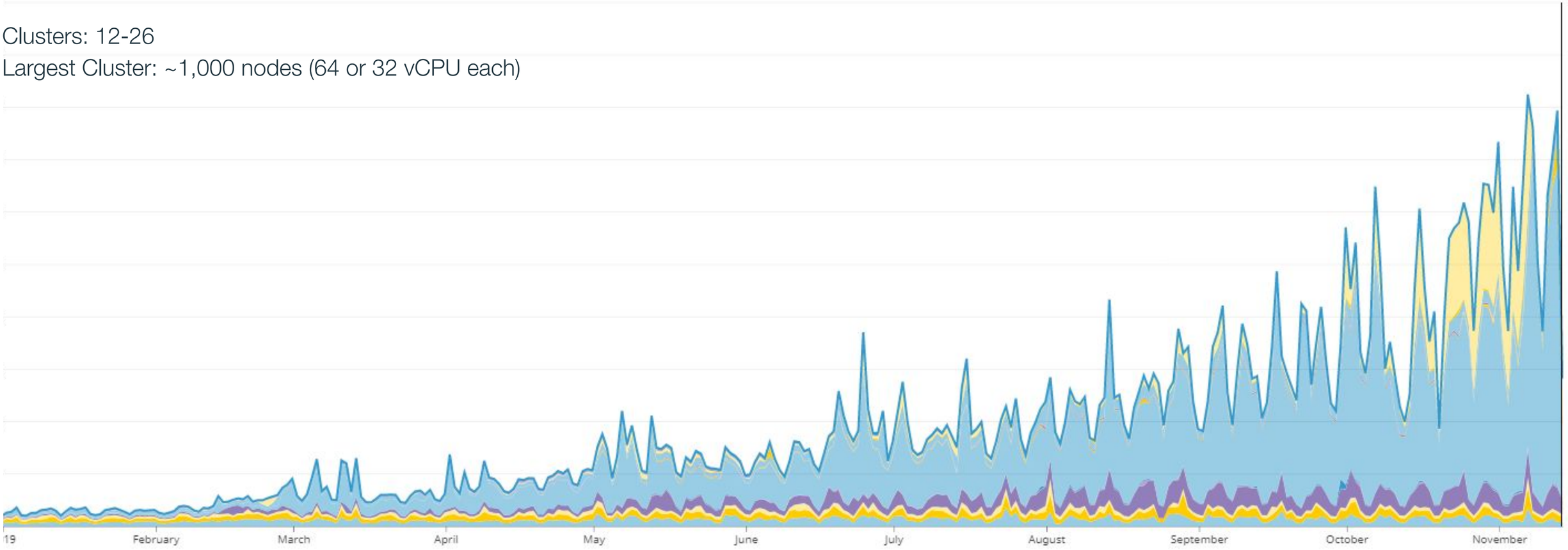


[C] Cruise Projects
Other logos unaffiliated with Cruise.

Multitenancy at Scale

Clusters: 12-26

Largest Cluster: ~1,000 nodes (64 or 32 vCPU each)



Multitenancy is when multiple applications operate in a shared environment.

Tenants are **logically isolated, but physically integrated**.

The more physical integration, the harder it is to preserve logical isolation.

Why Multitenancy?

Lower Cloud Costs

Higher collocation allows for higher utilization of cloud resources (compute, network, storage).

Lower Operational Costs

Fewer clusters can be managed by fewer platform engineers.

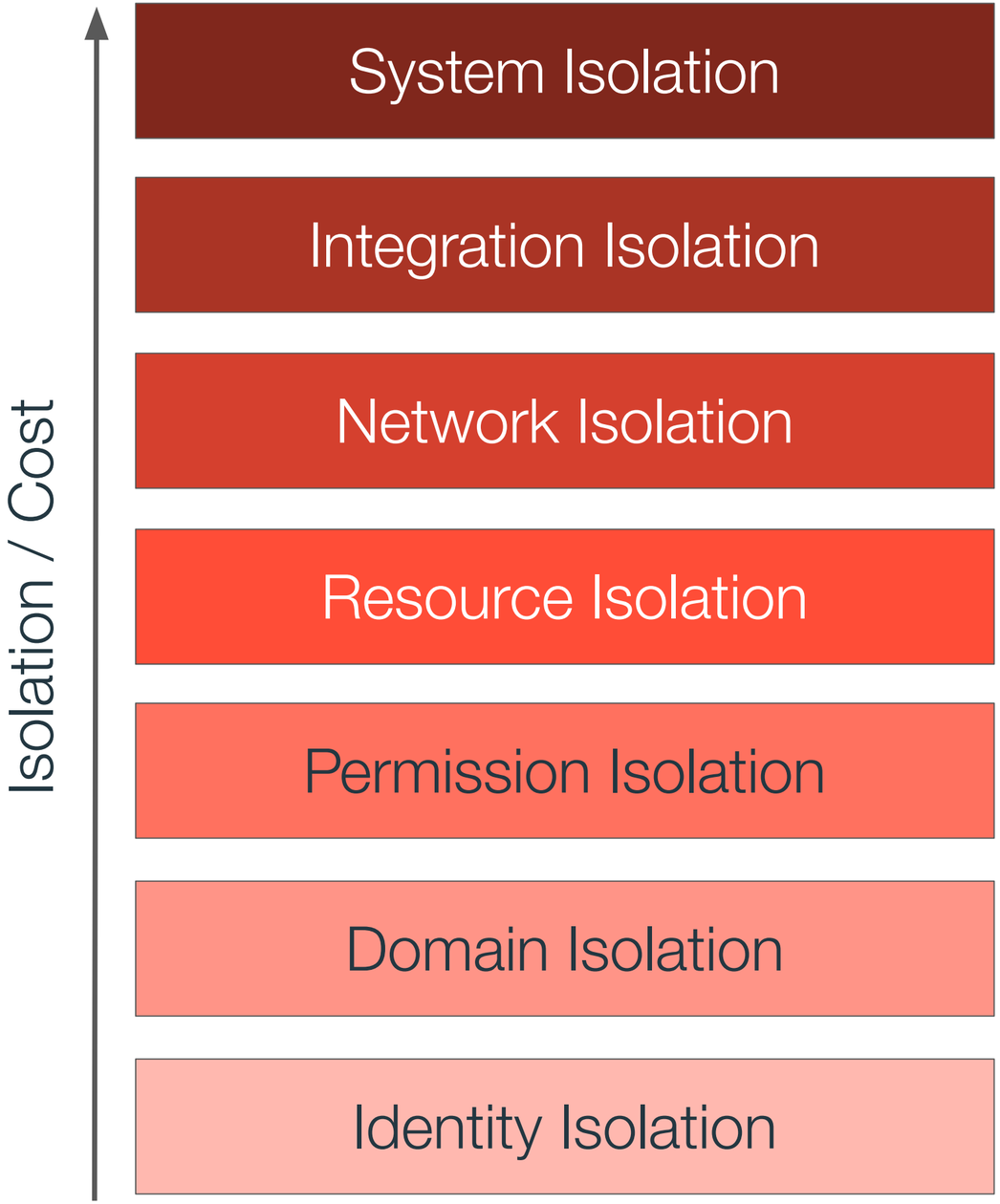
Higher Scale Validation

Validate real workloads at real scale by postponing clusters proliferation.

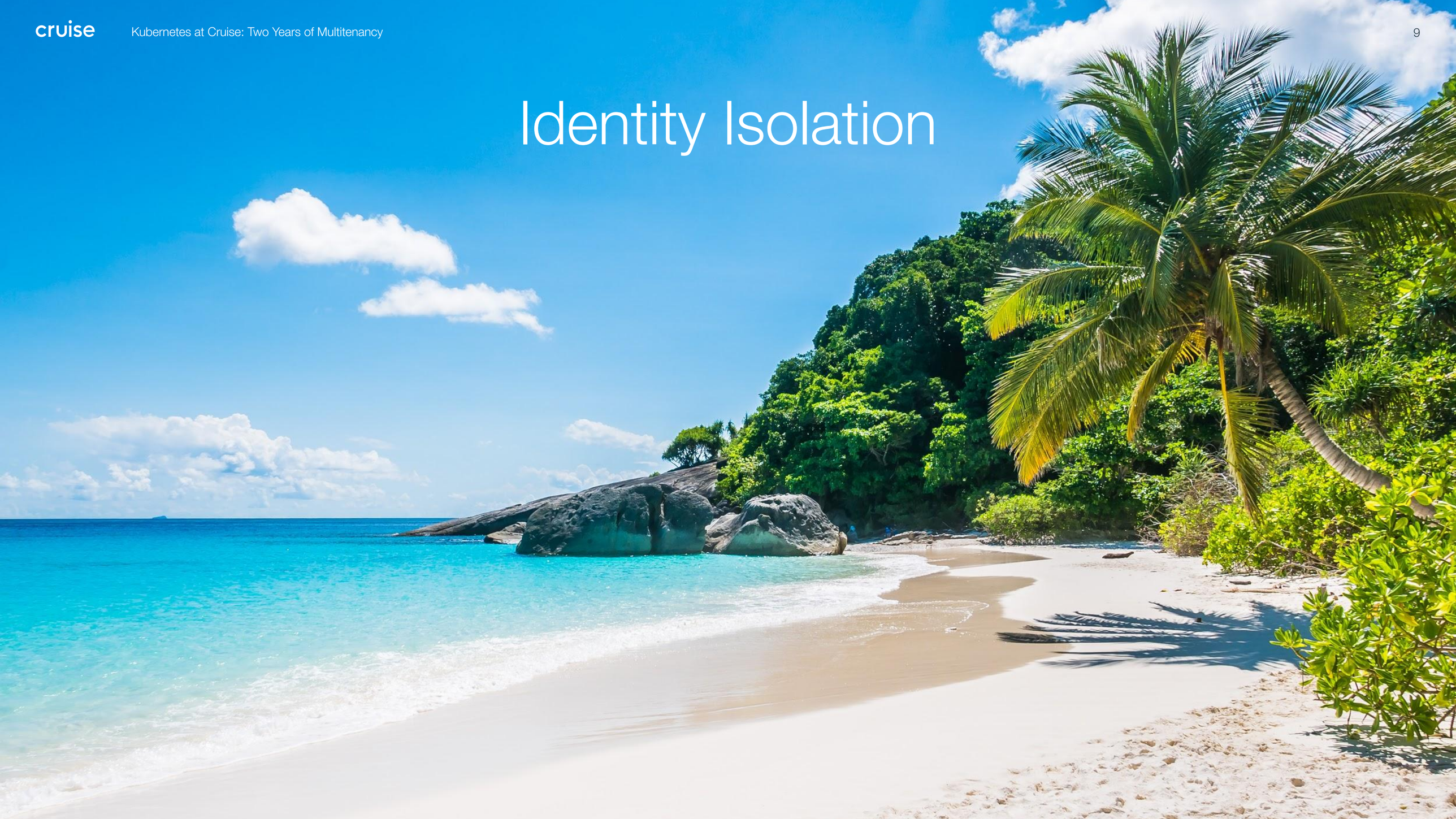
Higher Consistency

Focus on production readiness and tenant-facing improvements before scaling cluster operations.

Multitenancy: Layers of Isolation



Identity Isolation



Identity & Authentication

User Identity

- G Suite User Accounts
- Okta Single Sign-On (SSO)
- Duo Security (2FA)

Service Identity

- GCP Service Accounts
- K8s Service Accounts
- Signed Certificates
- JSON Web Tokens (JWT)

DAYTONA

Vault client for servers & containers.

Vault Login

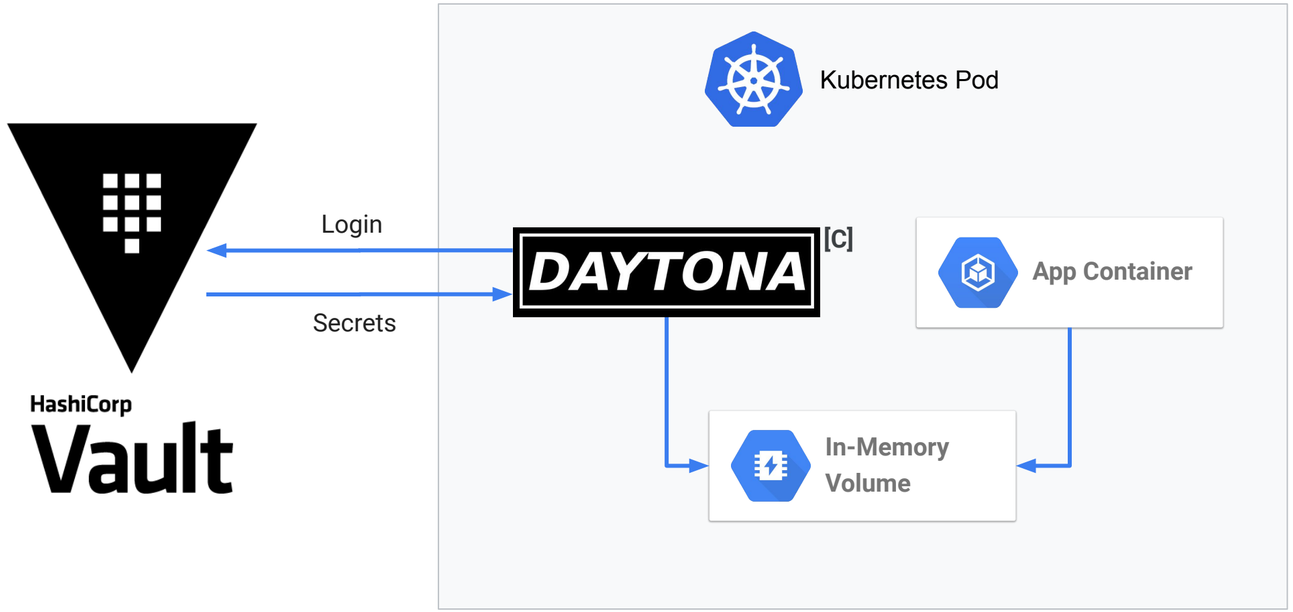
Kubernetes service accounts used for Vault authentication.

Secrets Injection

DAYTONA Init container side-loads secrets

Identity Translation

Vault generates temporary credentials on-demand



See also: GKE Workload Identity (beta)

[C] Cruise Projects
Other logos unaffiliated with Cruise.

(OSS) https://github.com/cruise-automation/daytona

k-rail

Security & operational policy enforcement tool.

Audit

Validating webhook logs policy violations

Enforce

Validating webhook optionally enforces policies

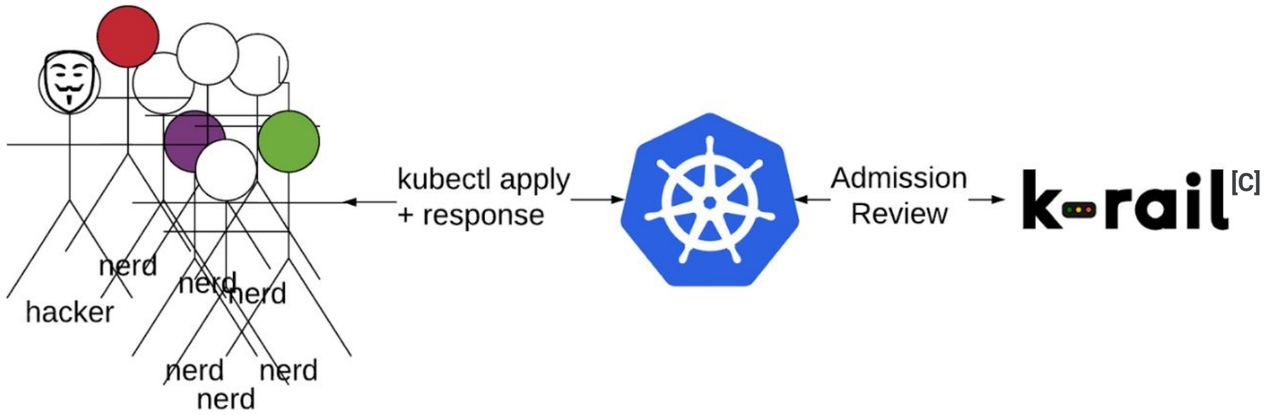
Apply Defaults

Mutating webhook applies policy defaults

See also: GKE Metadata Concealment (beta)

Prevent Privilege Escalation & Lateral Movement

- No Bind Mounts
- No Host Network
- No Host PID
- No New Capabilities
- No Privileged Container
- No Helm Tiller
- Default Docker Seccomp profile



[C] Cruise Projects
Other logos unaffiliated with Cruise.

(OSS) https://github.com/cruise-automation/k-rail

Domain Isolation

Environmental

Dev, Test, Stage, Prod

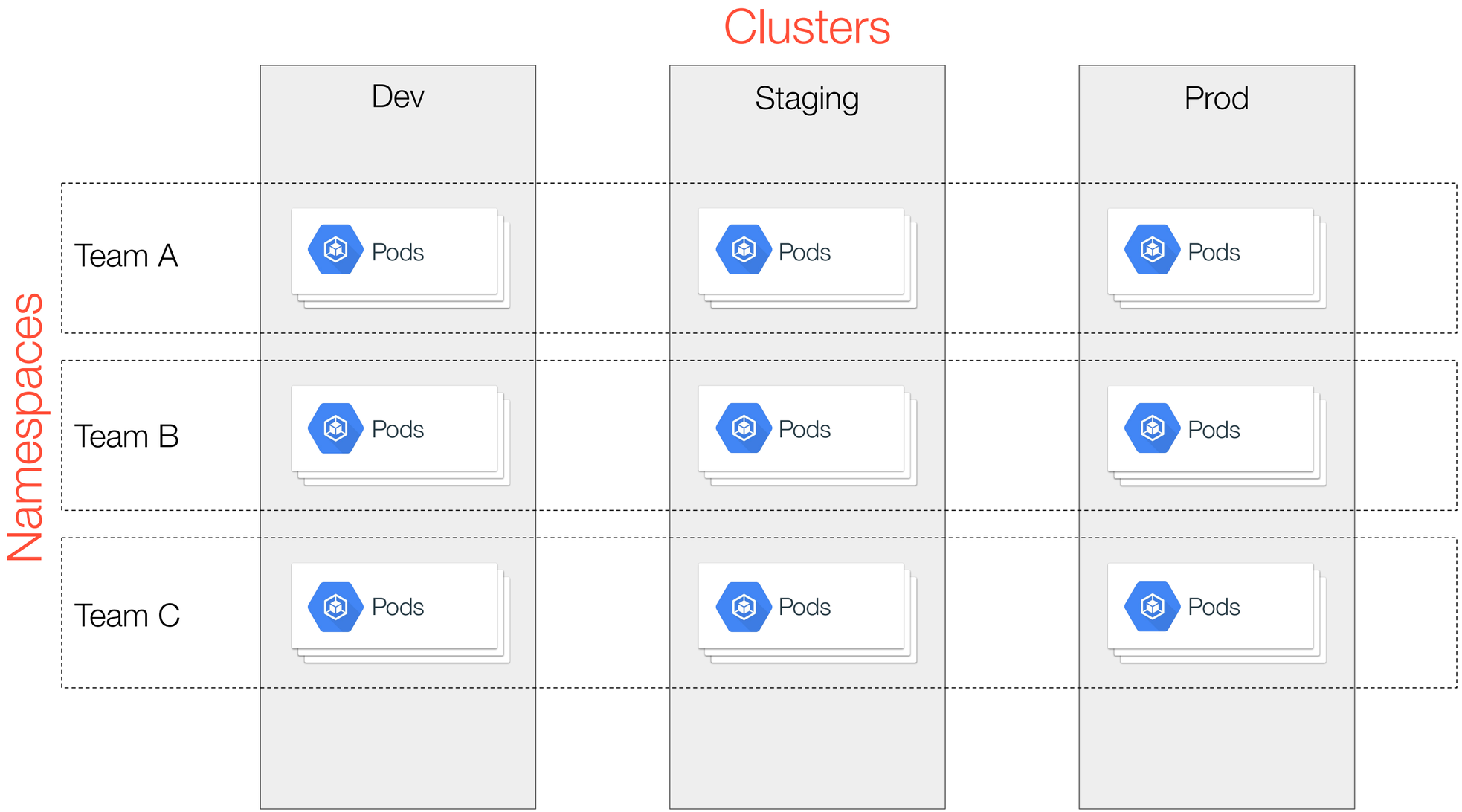
Organizational

Org, Dept, Team, Personal

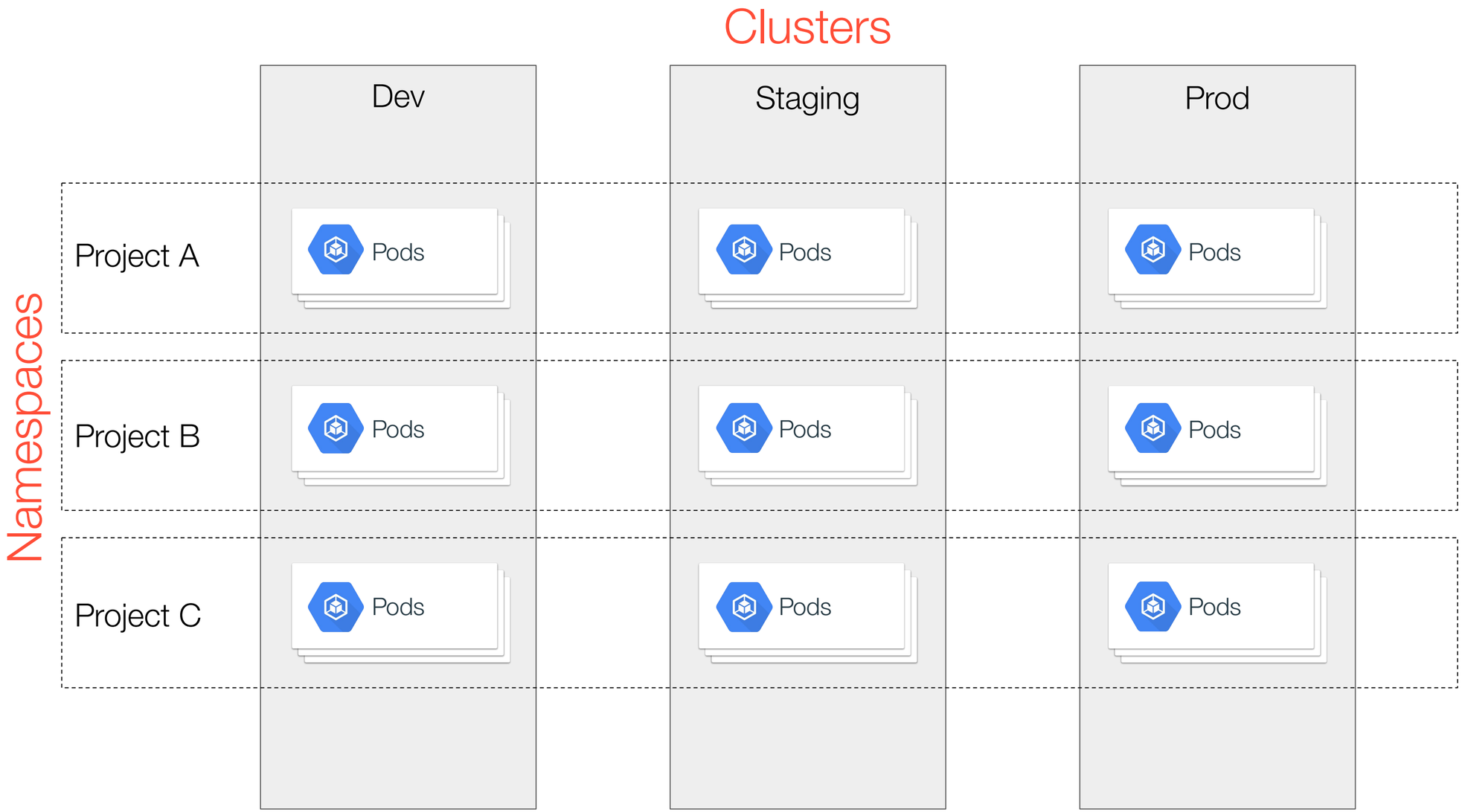
Architectural

Project, System, Component

Environmental vs Organizational Domains

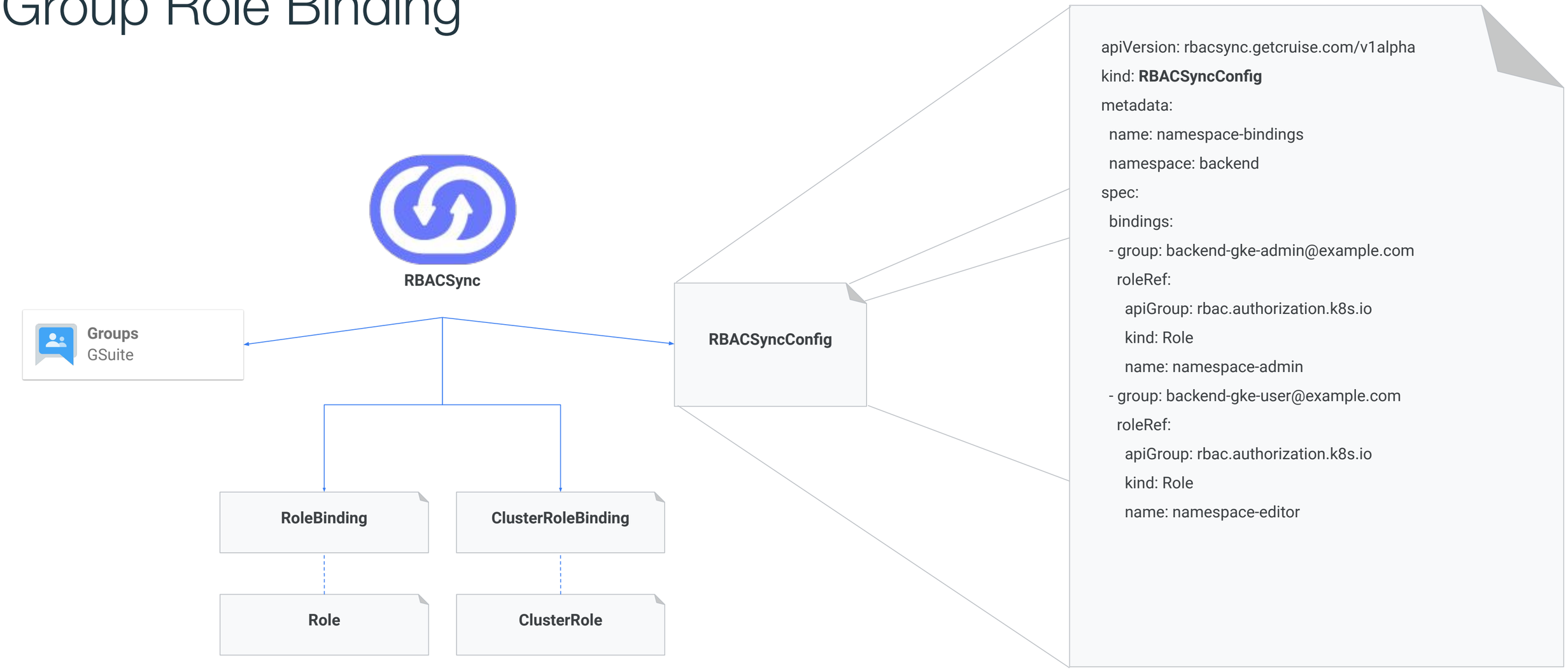


Environmental vs Architectural Domains



Permission Isolation

Group Role Binding



See also: Google Groups for GKE (beta)

(OSS) https://github.com/cruise-automation/rbacsync

Vault Workspaces

Standard hierarchy for storing and authorizing application secrets.

Group	Permissions	Path
Tenant Admin	admin	secret/<prefix>/<namespace>/*
Tenant Contractor	list	secret/<prefix>/<namespace>/*
App Service Account	list, get	secret/<prefix>/<namespace>/<env>/<app>/*

Isopod

DSL for Kubernetes configuration without YAML.

Domain Specific Language

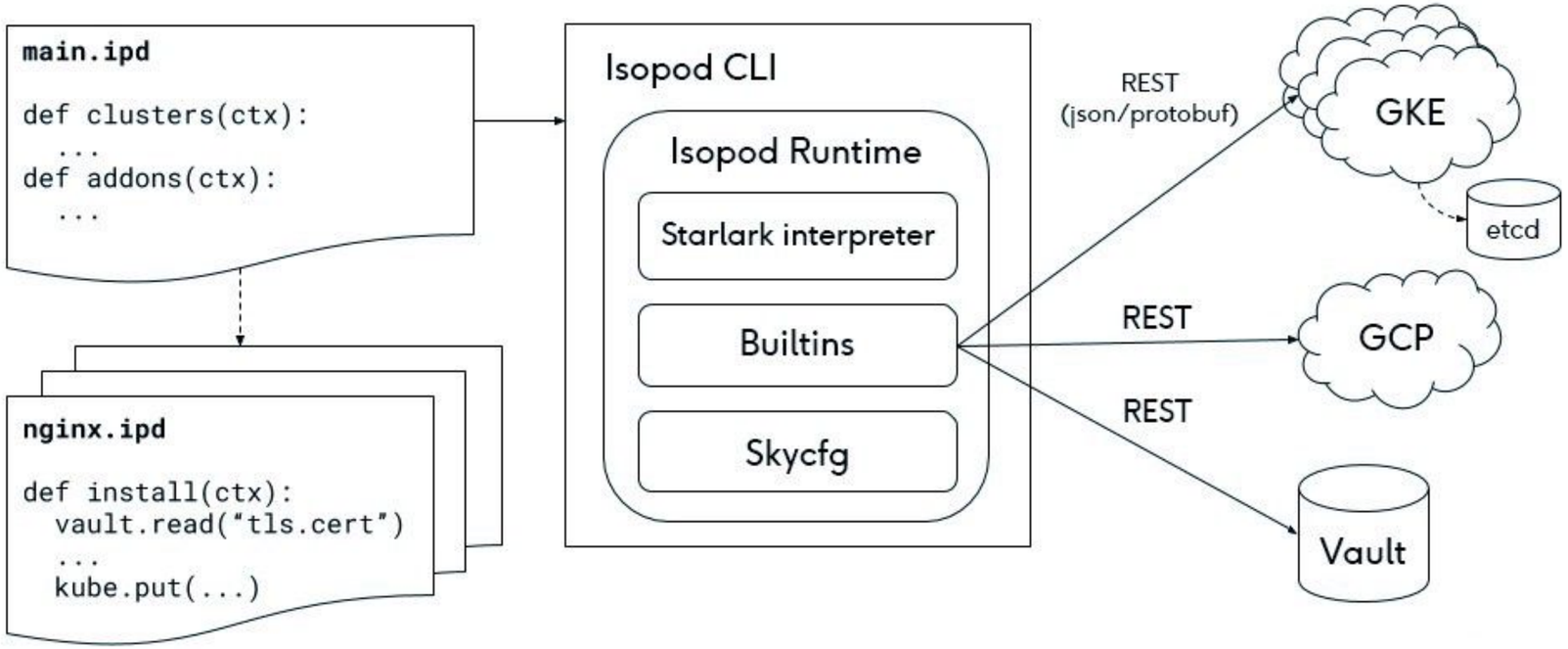
Loosely typed with local runtime type validation

Less YAML

Skylark backed by Kubernetes Go Client

Flexible Reuse

Alternative to Helm & Terraform for addon management



(OSS) https://github.com/cruise-automation/isopod

Juno

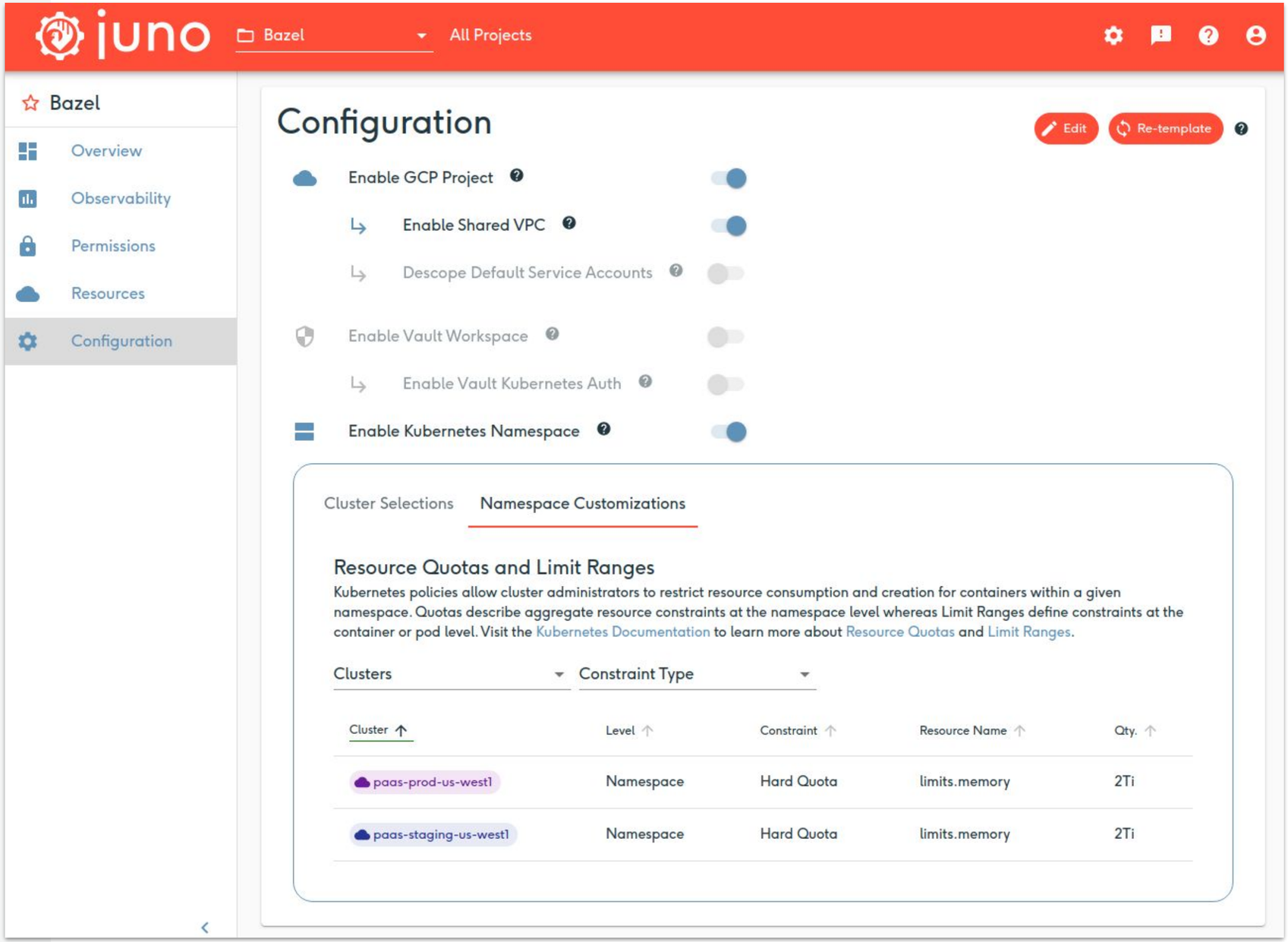
Cruise infra self-service resource provisioner.

Resource Management

- GCP Project
- Vault Workspace
- K8s Namespace

Related OSS Projects

- namespace-configuration-operator
- rbac-permissions-operator



(internal project)

Resource Isolation

Built-In Types

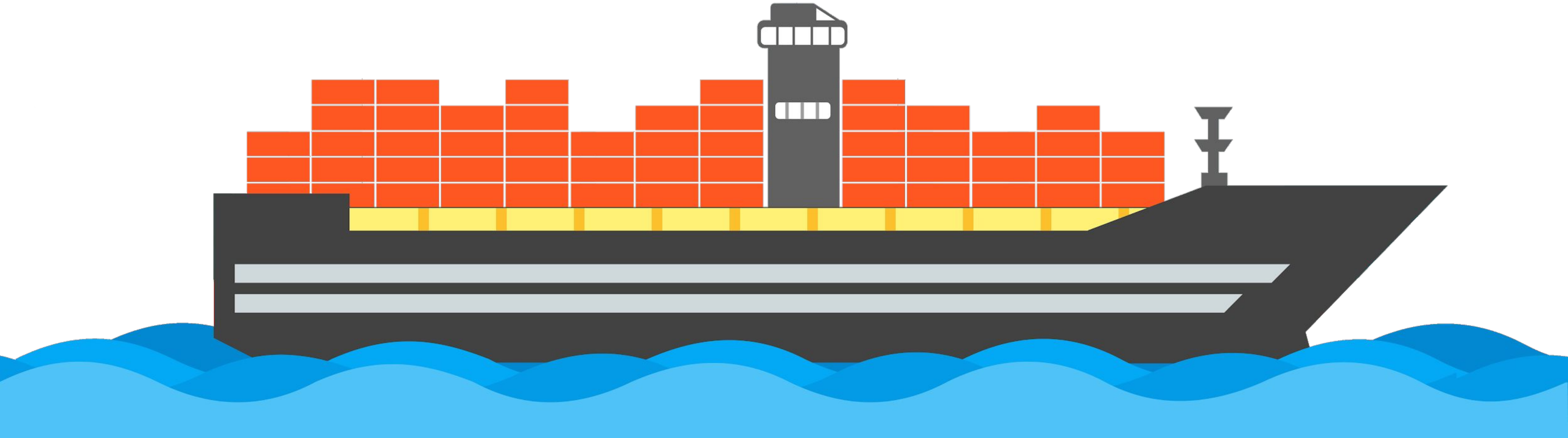
- CPU, GPU
- Memory
- Persistent Storage
  (for each Storage Class)
- Ephemeral Storage

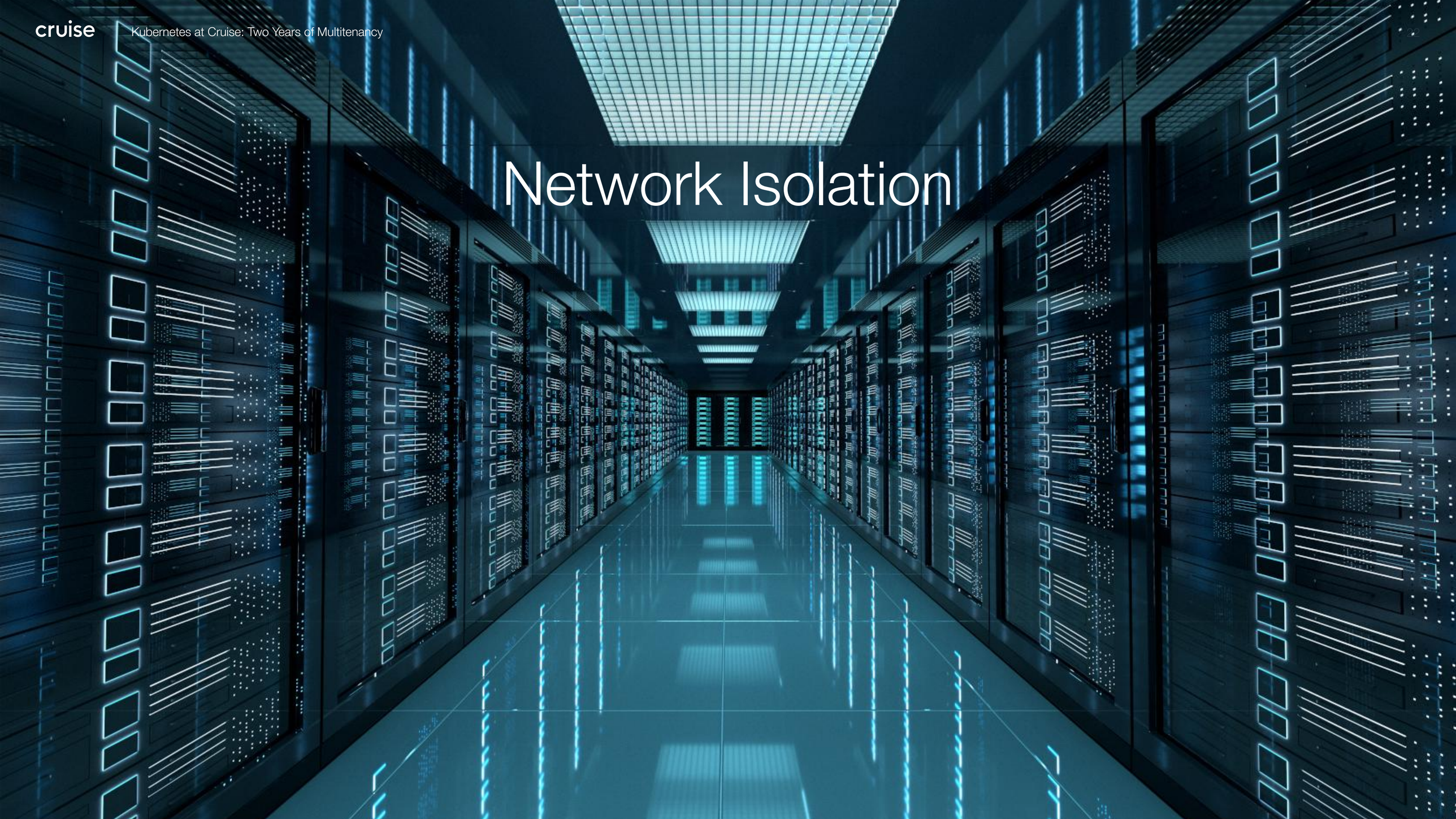
Storage Volumes

- OS Root
- Container Images
- Container Root
- Ephemeral Storage Volumes
- Persistent Local Storage Volumes

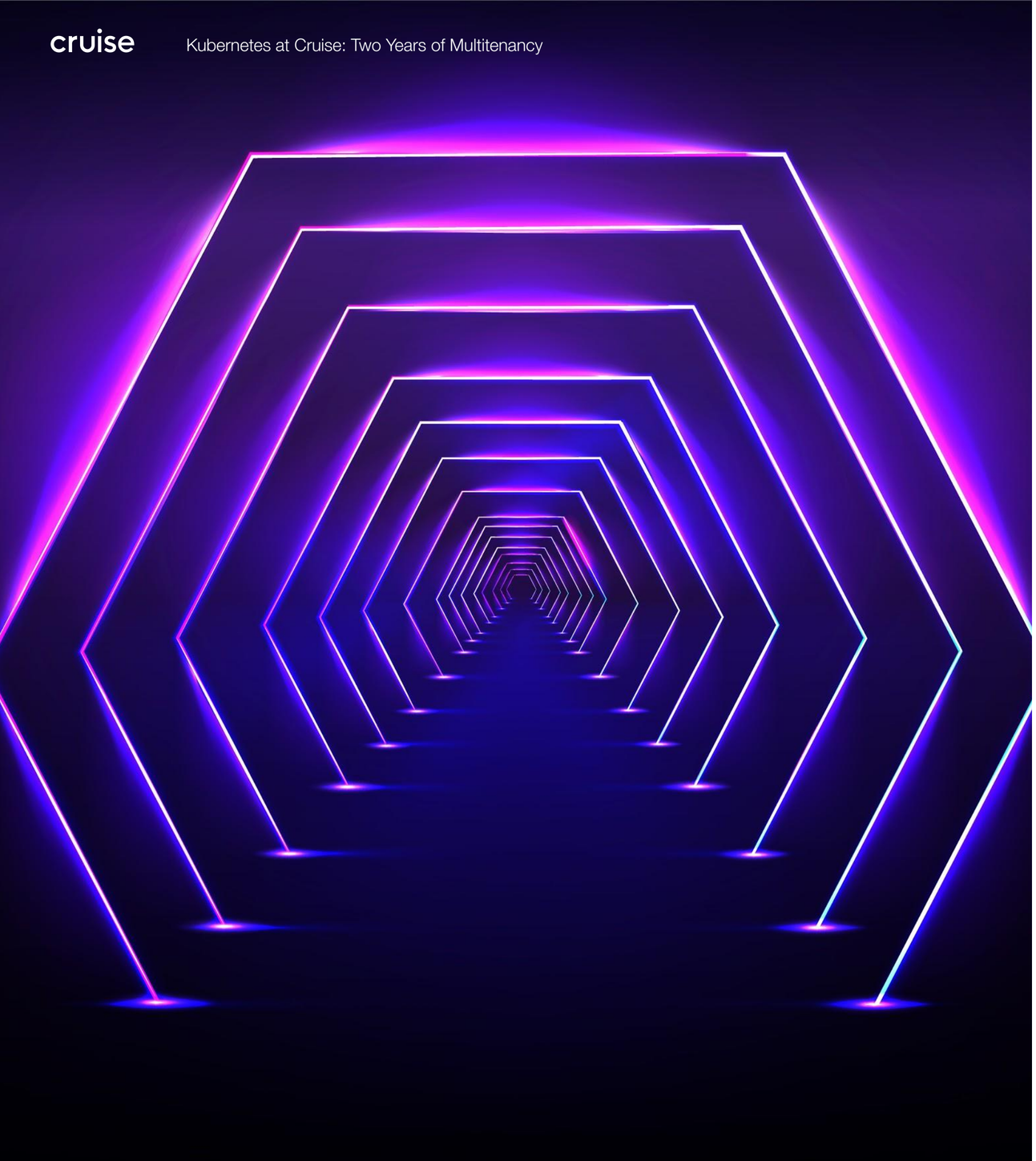
Quotas & Limits

- Resource Quota:
  Namespace Limits & Usage
- Limit Range:
  Pod Default Requests & Limits
- Defaults & Overrides (Juno)





Network Isolation



Shared Tunnels

NAT Gateways

- NAT Gateway Terraform Module (network label routing)
- Cloud NAT
- Whitelists

Ingress / Egress QPS

- No Built-In Isolation
- Network Stack shared with Network Storage (NAS/SAN/Cloud)

Bandwidth

- No Built-In Isolation
- CNI Bandwidth Plugin (Calico)
- Istio Rate Limits (Quota Rules)

Virtual Firewalls

Network Policy

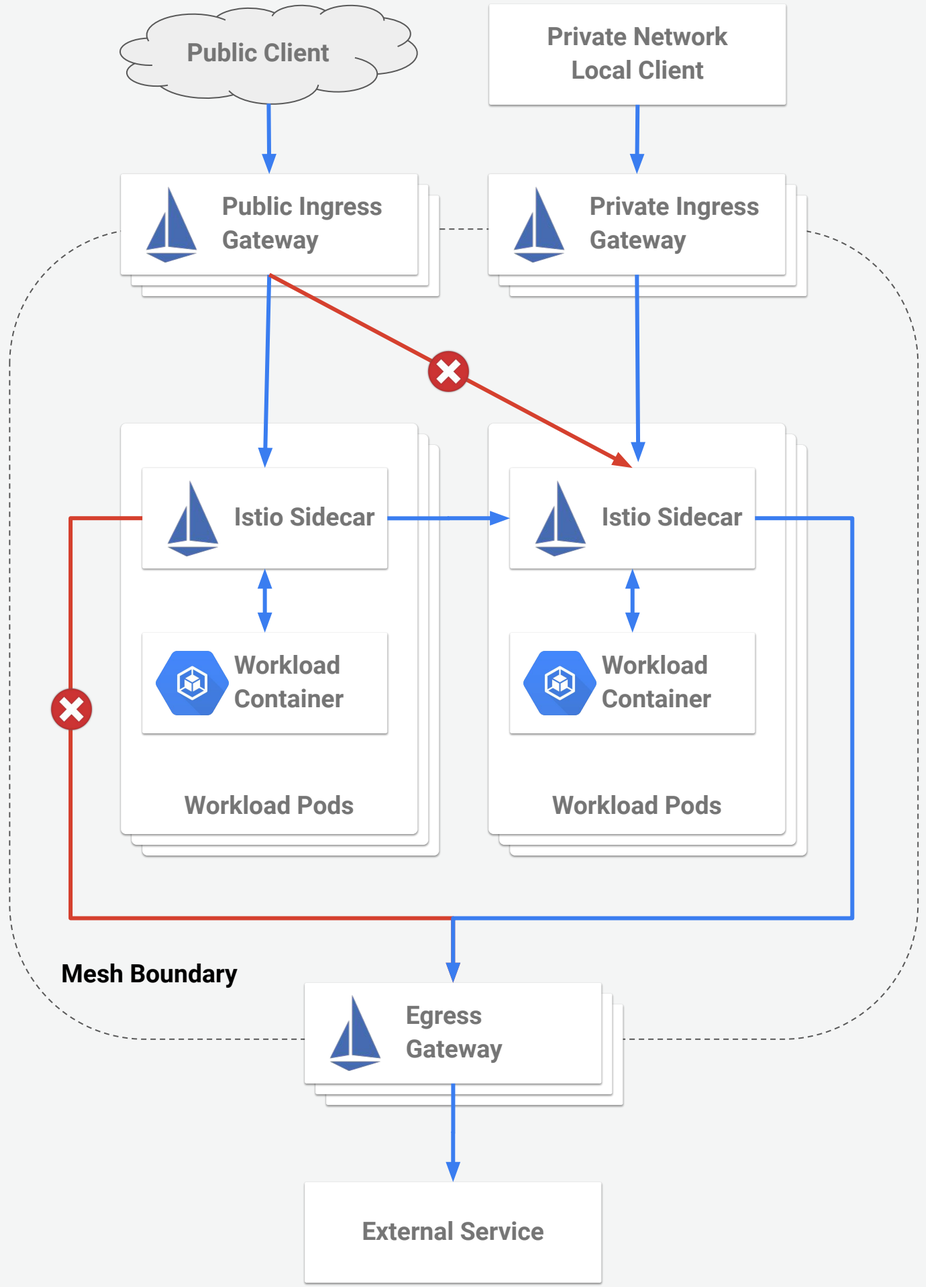
- IP Block
- Namespace Selector
- Pod Selector

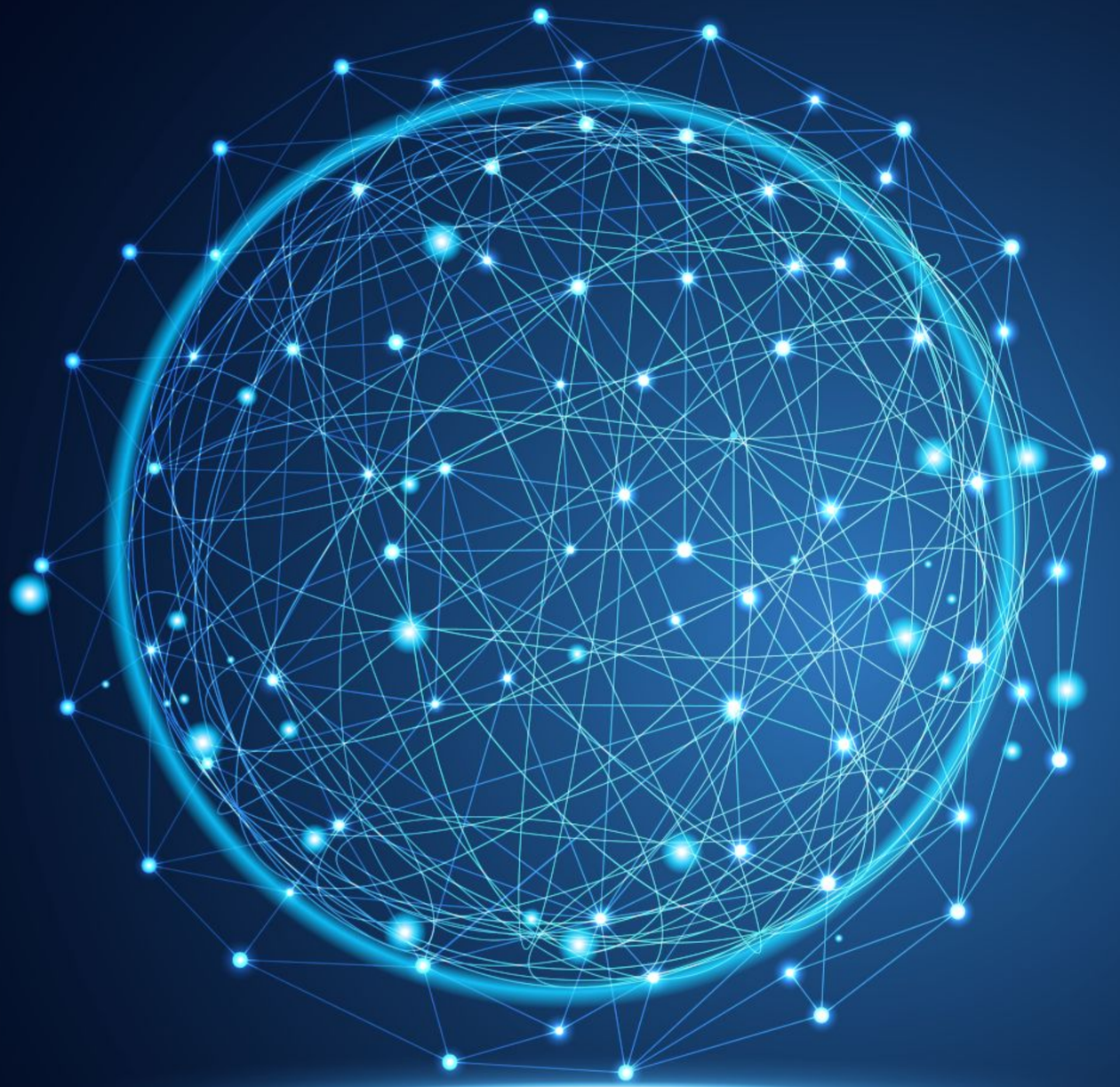
Service Authorization

- Istio mTLS
- Istio Authorization Policy

Rule Based Access

- Istio Denier Rule
- Istio List Checker Adapter



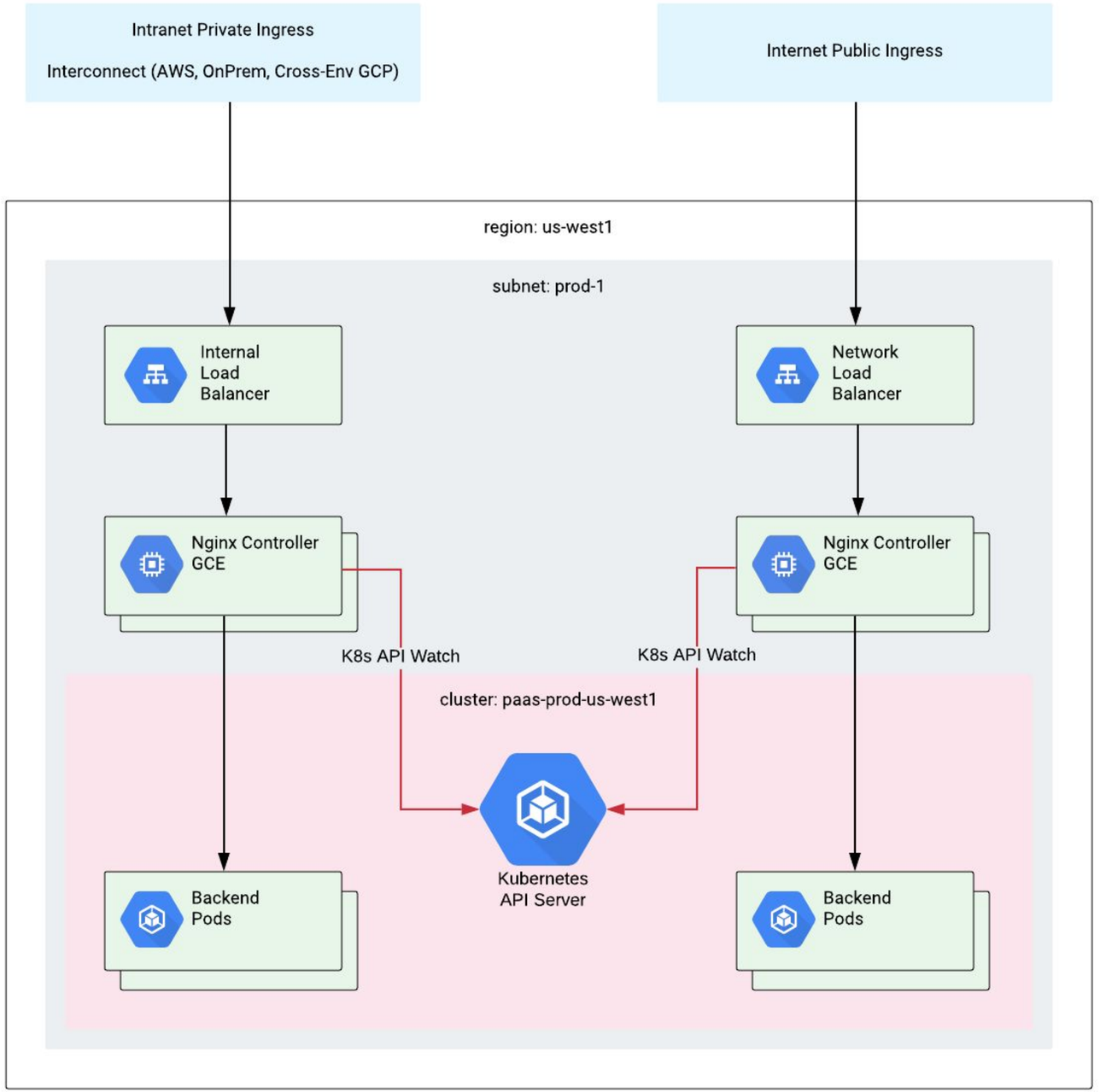


Integration Isolation

Shared Ingress

Isolation Options

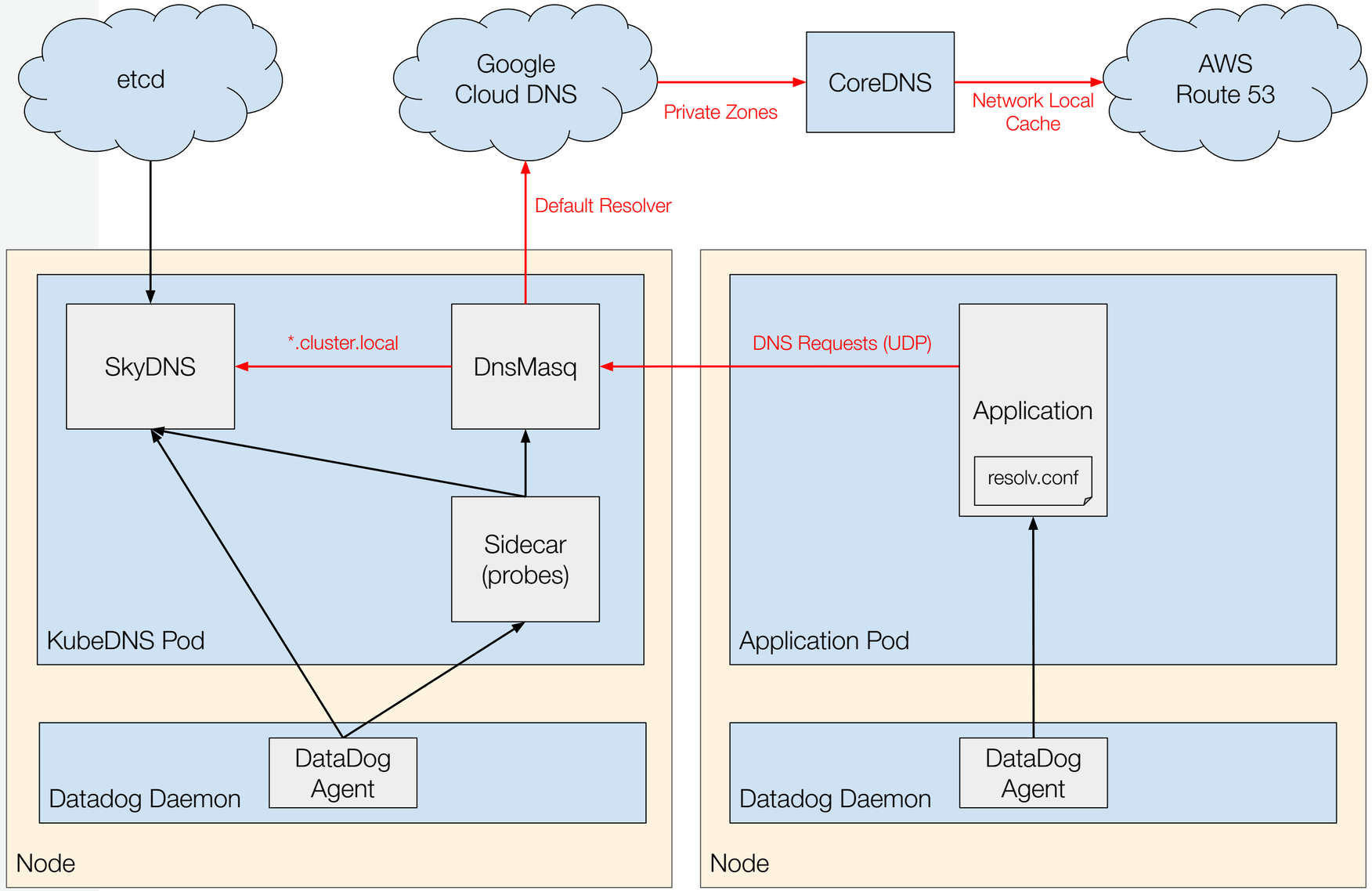
- Separate Private & Public (shown)
- Dedicated Ingress Node Pool
- Dedicated Ingress VMs (shown)
- Dedicated Ingress Per Tenant



Shared DNS

Isolation Options

- Node Local Cache
- Dedicated DNS Per Node Pool
- Dedicated DNS Per Cluster



Shared Observability

Logs

- Log Visibility (Container, Platform, Audit)
- Log-Based Metrics (Edit Perms)
- Fluentd DaemonSet Vertical Autoscaling

Metrics

- Kube State Metrics not HA
- DaemonSet Agent HA & Slow or Local & Fast
- Sidecar Agent Duplicate Metrics
- DogStatsD vs Prometheus Style

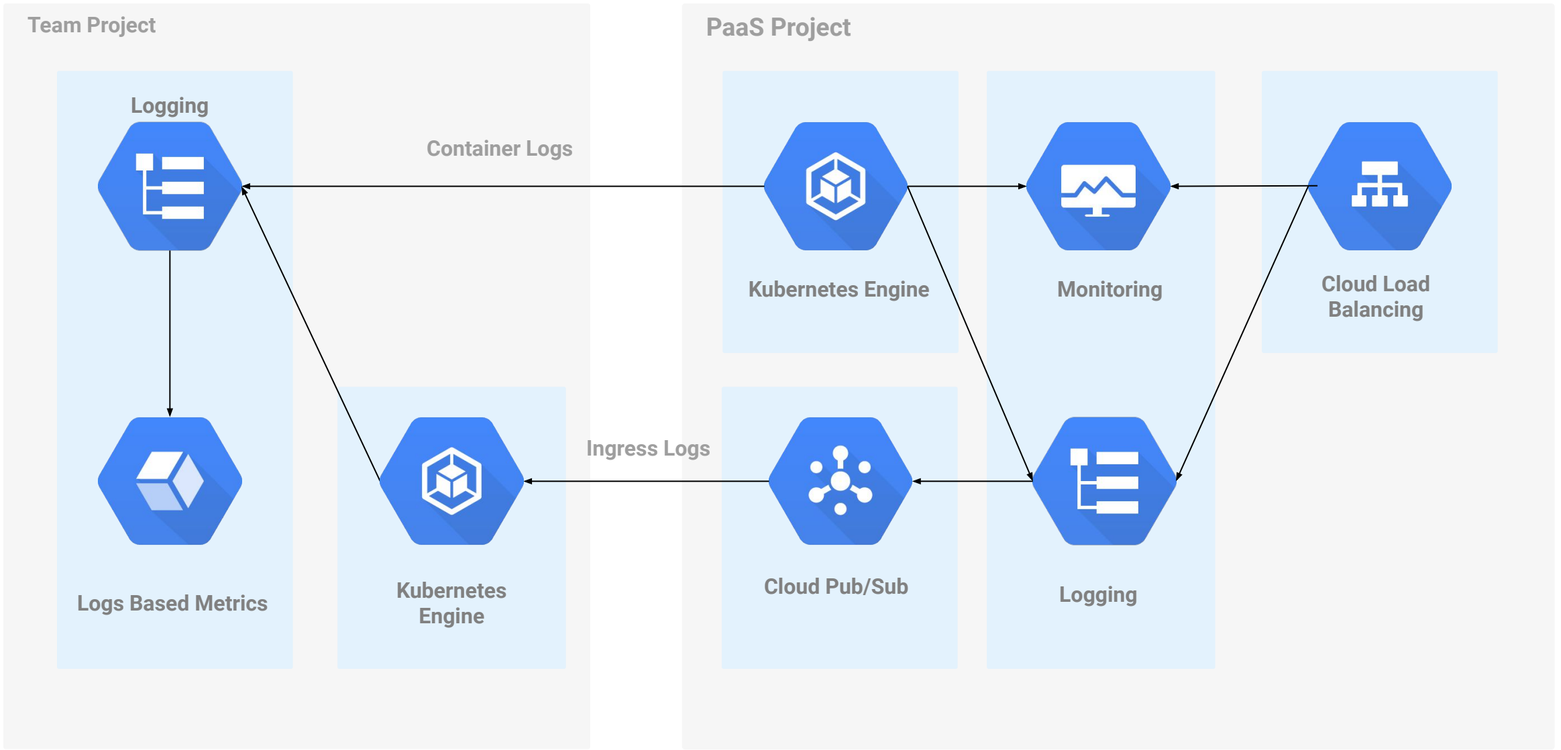
Distributed Tracing

- OpenTelemetry vs OpenCensus vs OpenTracing
- Stackdriver vs DataDog vs Jaeger vs Zipkin

Dashboard Management

- Platform Dashboards
- Workload Dashboards
- Dashboard Templates

Container & Ingress Log Export



No Isolation
(Shared Cluster Admin)

Logical Isolation
(Soft Multitenancy)

Physical Isolation
(Hard Multitenancy)

System Isolation
(Single Tenancy)

System Isolation

Machines

- Dedicated Node Pool
- Dedicated Cluster

Cluster Components

- API Server
- Scheduler
- Cluster Autoscaler
- Kube Proxy (iptables)

Networks

- Dedicated IP Ranges
- Dedicated Subnet
- Dedicated Network
- Dedicated Interconnects

Was it worth it?

Costs

- Shared Downtime
- Incompatible Tooling

Challenges

- Single Tenant Integrations
- Managed CRD Installation
- Managed Internal Platform Model
- Kubernetes Itself

Benefits

- Lower Cloud Costs
- Lower Operational Costs
- Higher Scale Validation
- Higher Consistency
- Prioritized Security Investments
- Expertise Building

Thank you

Karl Isenberg, Cruise

@karlkfi