Google Cloud

Next ’24

The past, present, and future of Google Kubernetes Engine

Proprietary

Gari Singh

Product Manager,

Google Cloud

Drew Bradstock

Sr Director

Product Management,

Google Cloud

Agenda

01 The Past

02 The Present

03 & The Future of Kubernetes

04 Where is GKE heading next?

A short history of Kubernetes

In the beginning...

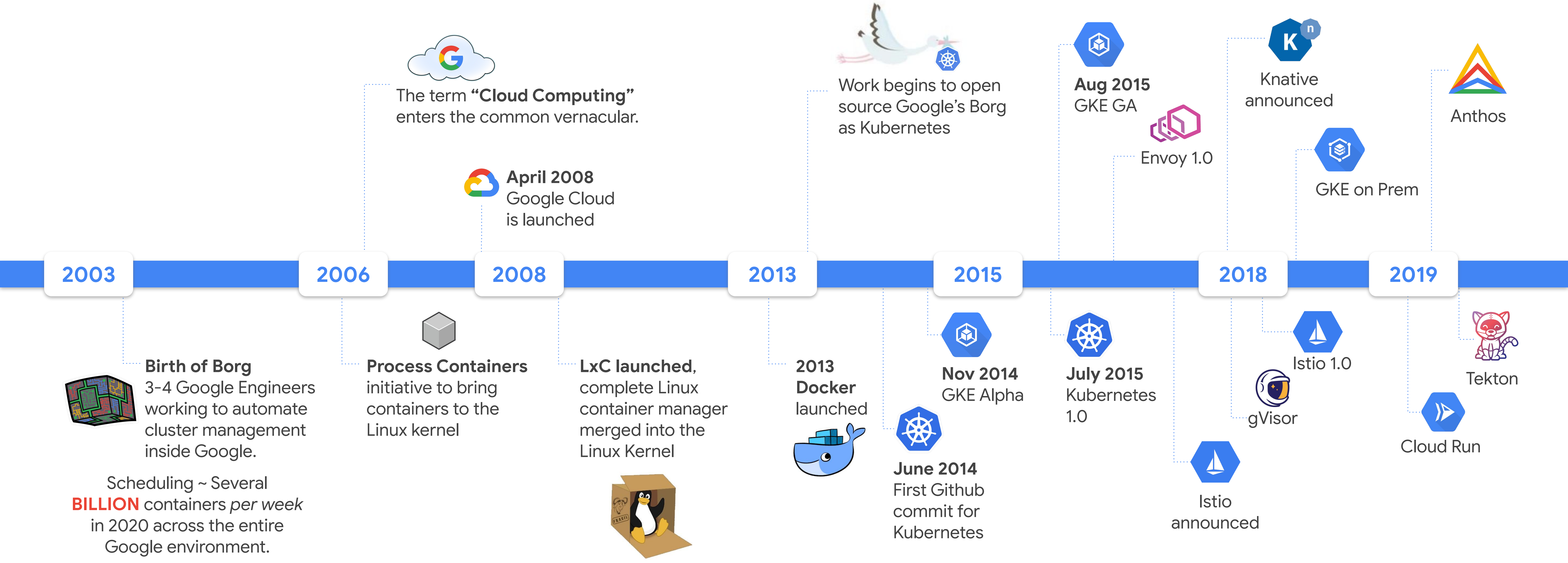
There was the monolith.

Which became many, many, many microservices

Images courtesy of Gemini

History of Kubernetes

Google Innovation Timeline



2003

Birth of Borg
3-4 Google Engineers working to automate cluster management inside Google.

Scheduling ~ Several **BILLION** containers *per week* in 2020 across the entire Google environment.

2006

The term **"Cloud Computing"** enters the common vernacular.

Process Containers initiative to bring containers to the Linux kernel

2008

April 2008
Google Cloud is launched

LxC launched, complete Linux container manager merged into the Linux Kernel

2013

2013 Docker launched

Work begins to open source Google's Borg as Kubernetes

June 2014
First Github commit for Kubernetes

Nov 2014
GKE Alpha

2015

July 2015
Kubernetes 1.0

Aug 2015
GKE GA

Envoy 1.0

Istio announced

2018

Knative announced

GKE on Prem

gVisor

Istio 1.0

2019

Anthos

Tekton

Cloud Run

OSS community is the heart of Kubernetes

314K

commits

74K+

contributors

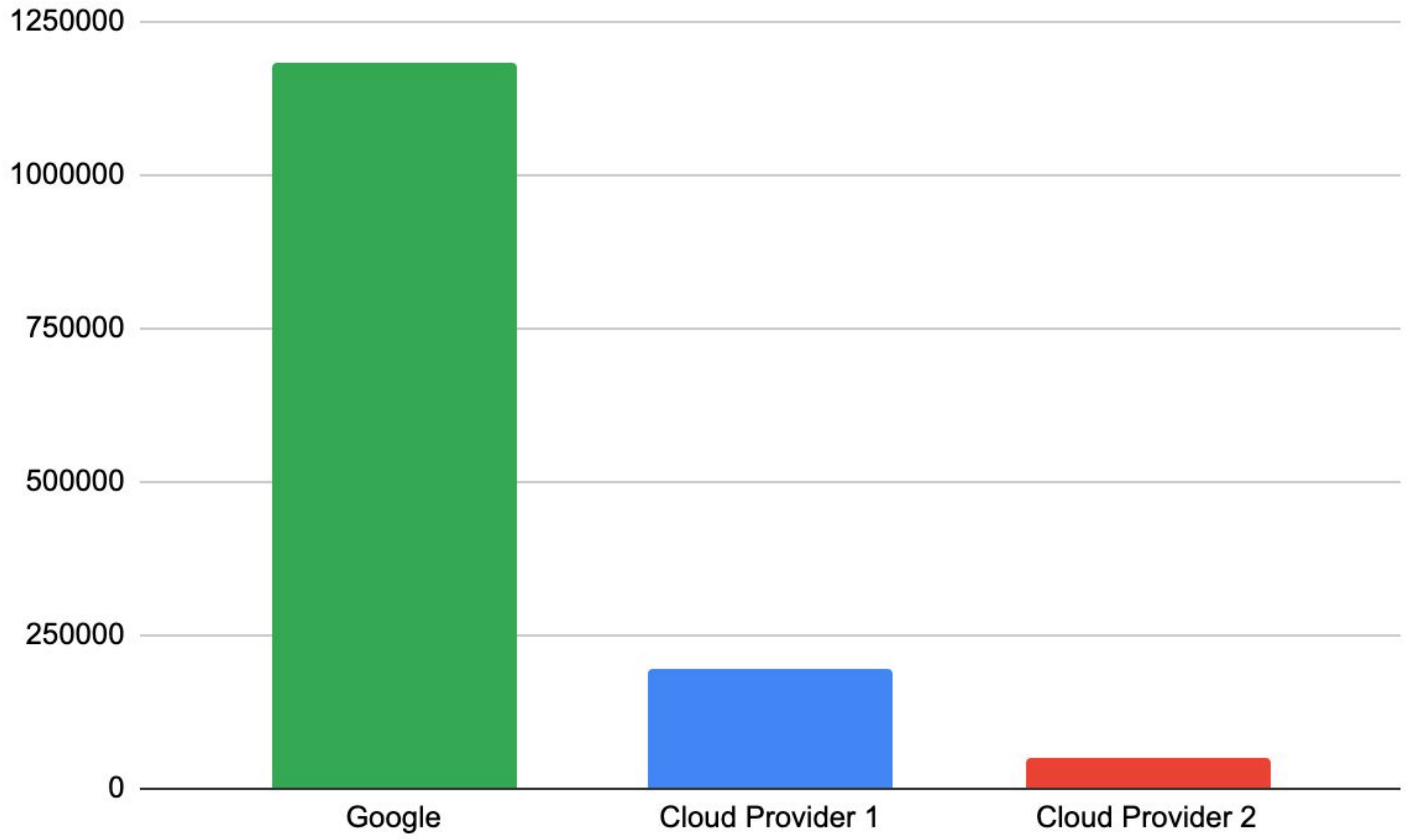
7.8K+

companies

Google leads in contributions to K8s

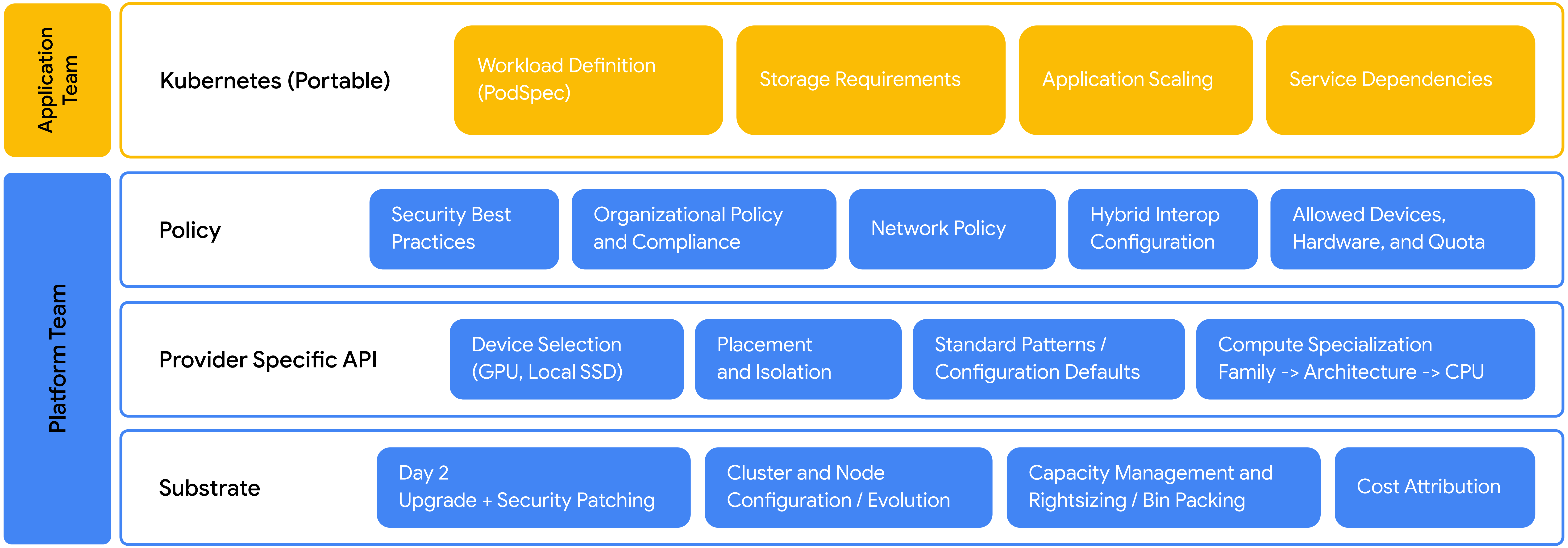
Built, tested and powered by the largest contributor to Kubernetes*

- Entire OSS Kubernetes project is built, tested and distributed on Google Cloud Platform itself
- Run on the same infrastructure which serves billions of requests per day
- Who better to run Kubernetes than the largest engineering contributor to Kubernetes?



Kubernetes contributions in by major cloud vendors

Layers of Kubernetes



Application Team

Kubernetes (Portable)

- Workload Definition (PodSpec)
- Storage Requirements
- Application Scaling
- Service Dependencies

Platform Team

Policy

- Security Best Practices
- Organizational Policy and Compliance
- Network Policy
- Hybrid Interop Configuration
- Allowed Devices, Hardware, and Quota

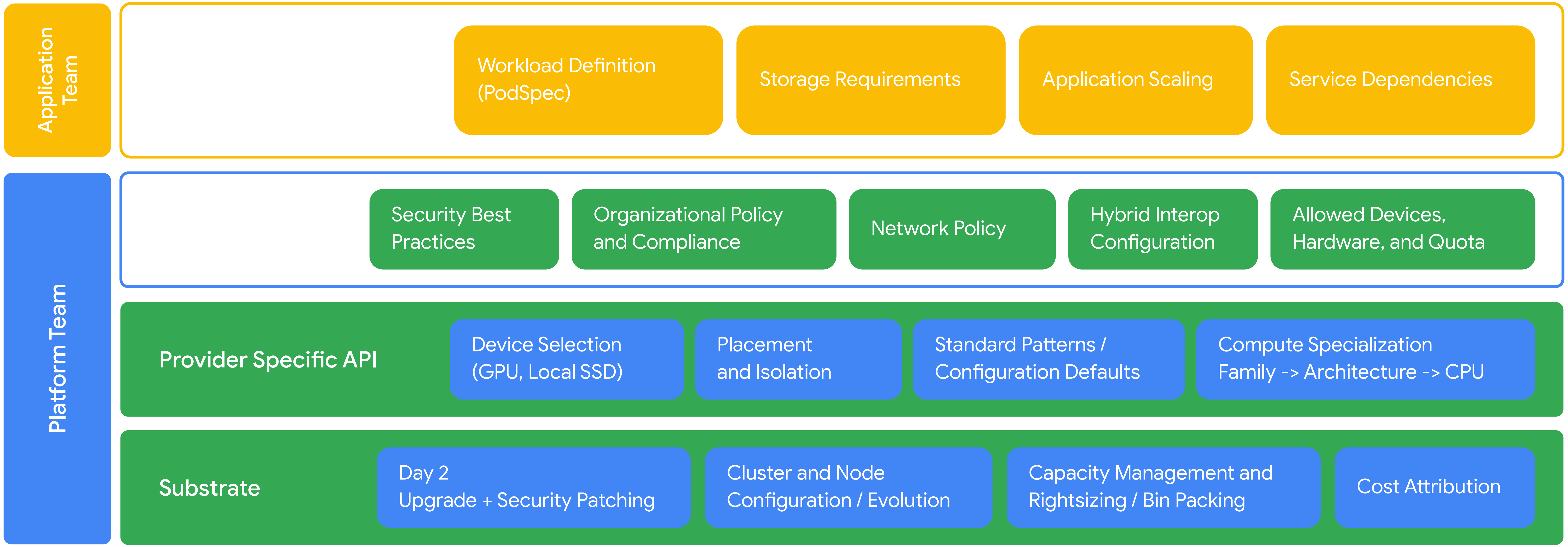
Provider Specific API

- Device Selection (GPU, Local SSD)
- Placement and Isolation
- Standard Patterns / Configuration Defaults
- Compute Specialization Family -> Architecture -> CPU

Substrate

- Day 2 Upgrade + Security Patching
- Cluster and Node Configuration / Evolution
- Capacity Management and Rightsizing / Bin Packing
- Cost Attribution

Why do I want to manage all this?



Application Team

Workload Definition (PodSpec)

Storage Requirements

Application Scaling

Service Dependencies

Platform Team

Security Best Practices

Organizational Policy and Compliance

Network Policy

Hybrid Interop Configuration

Allowed Devices, Hardware, and Quota

Provider Specific API

Device Selection (GPU, Local SSD)

Placement and Isolation

Standard Patterns / Configuration Defaults

Compute Specialization Family -> Architecture -> CPU

Substrate

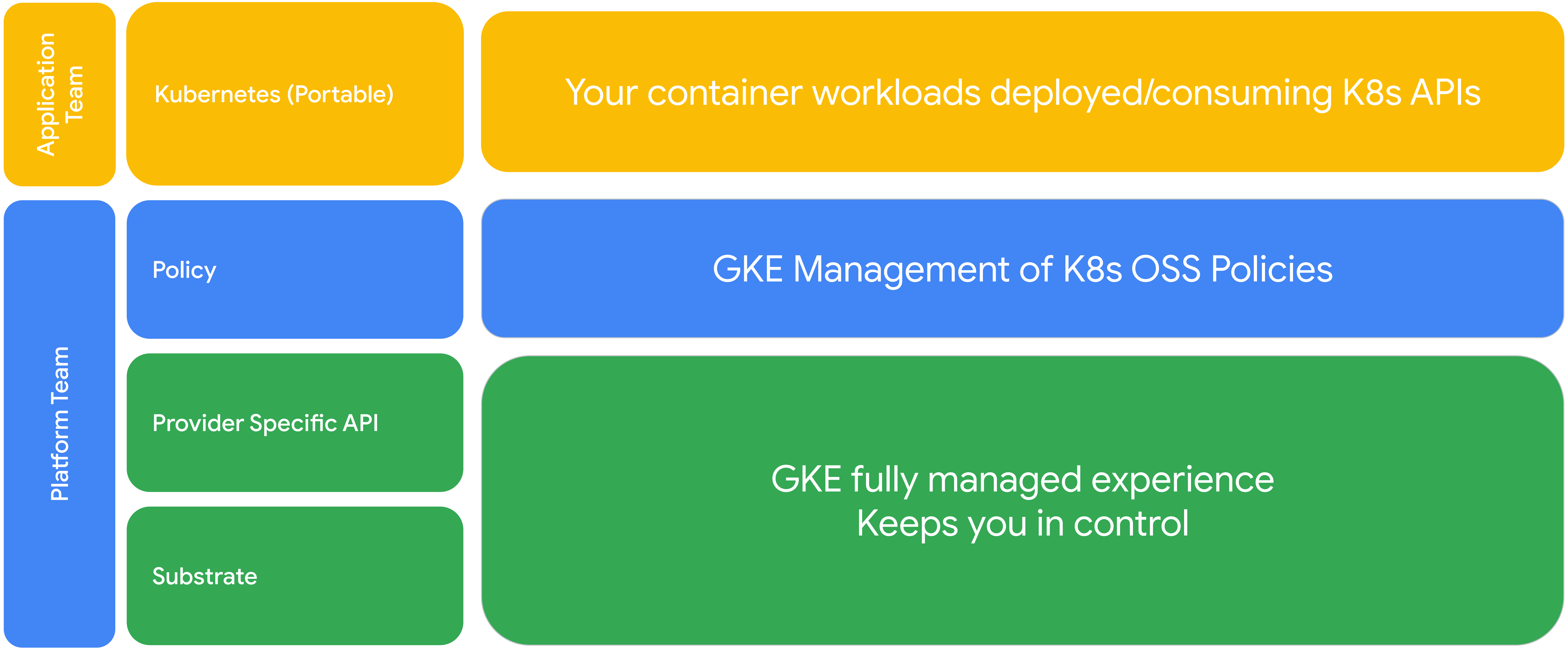
Day 2 Upgrade + Security Patching

Cluster and Node Configuration / Evolution

Capacity Management and Rightsizing / Bin Packing

Cost Attribution

GKE to the rescue



Application Team

Kubernetes (Portable)

Your container workloads deployed/consuming K8s APIs

Platform Team

Policy

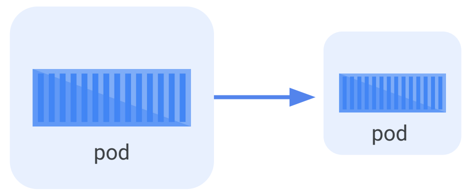
GKE Management of K8s OSS Policies

Provider Specific API

Substrate

GKE fully managed experience
Keeps you in control

Leverage GKE to do more with less



Workload rightsizing

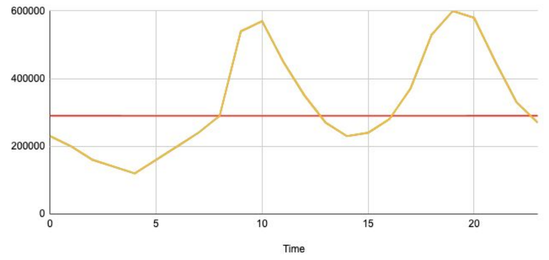
Requested resources
vs Actual utilization

Workload Reliability Engineering

Ensuring your workloads are available and tuned

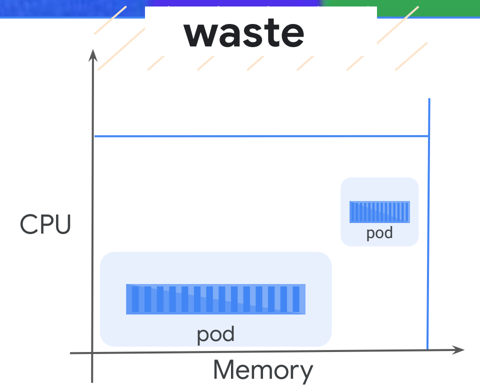
Site Reliability Engineering

Ensuring your infrastructure is available, reliable, and scalable



Demand based downscaling

Horizontal and vertical auto scaling to respond to changes in demand and optimize cloud costs



Cluster bin packing

Optimizing provisioned infrastructure, getting the biggest bang for your buck

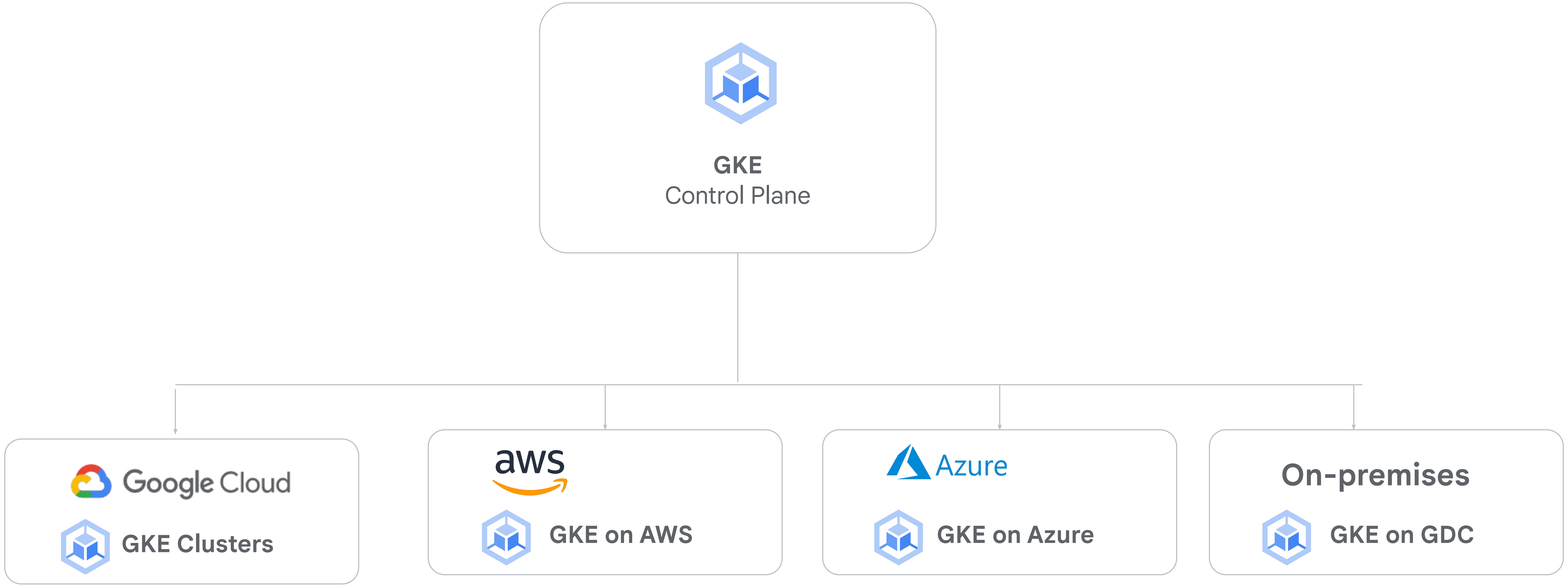
Application Developer

GKE
Pod level SLA

Platform Admin

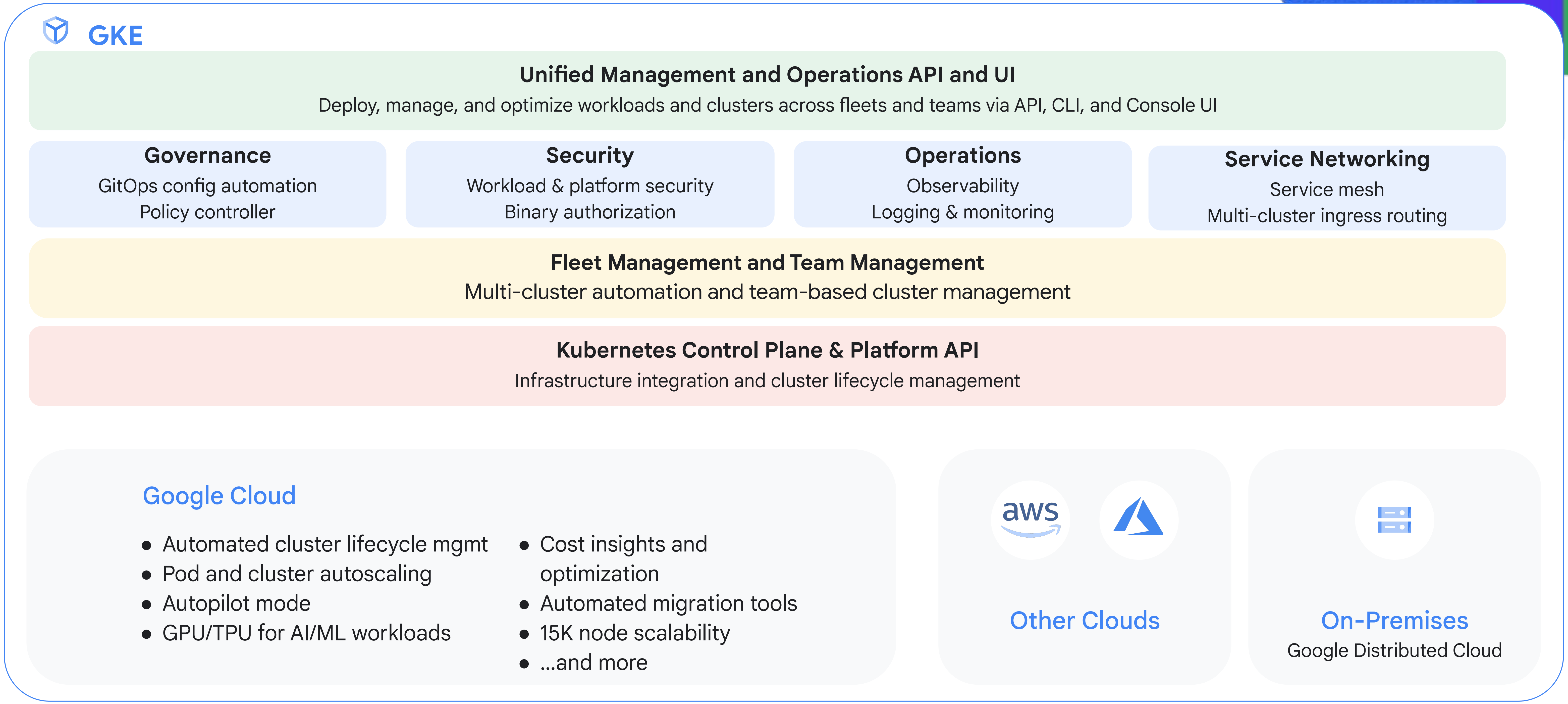
Multi-cloud for Kubernetes is born

Centrally manage the lifecycle of clusters running anywhere with a unified control plane



GKE Now

Building a platform should be easy



GKE

Unified Management and Operations API and UI

Deploy, manage, and optimize workloads and clusters across fleets and teams via API, CLI, and Console UI

Governance

GitOps config automation

Policy controller

Security

Workload & platform security

Binary authorization

Operations

Observability

Logging & monitoring

Service Networking

Service mesh

Multi-cluster ingress routing

Fleet Management and Team Management

Multi-cluster automation and team-based cluster management

Kubernetes Control Plane & Platform API

Infrastructure integration and cluster lifecycle management

Google Cloud

- Automated cluster lifecycle mgmt
- Pod and cluster autoscaling
- Autopilot mode
- GPU/TPU for AI/ML workloads
- Cost insights and optimization
- Automated migration tools
- 15K node scalability
- ...and more

Other Clouds

On-Premises

Google Distributed Cloud

Upgrade safely

Mitigate deprecations

Auto-upgrades are paused for exposed clusters

Insights notify with actionable details for mitigation

Qualify by rolling out in sequence

Fleet-based and **team-based rollout sequences** allow for soak time in staging and testing environments before auto-upgrading production

Upgrade when ready and safe

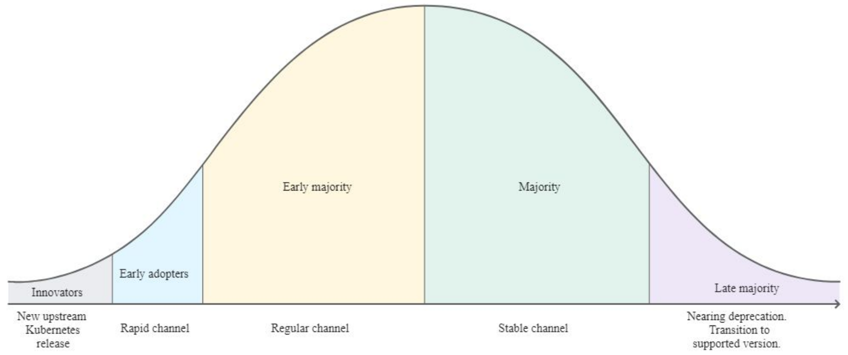
Maintenance exclusions postpone auto-upgrades until ready

Maintenance windows define safe time for upgrades

Release channels

Chrome-like, automated updates. Choose a release cadence and feature set to match risk preference.

Always on reliability

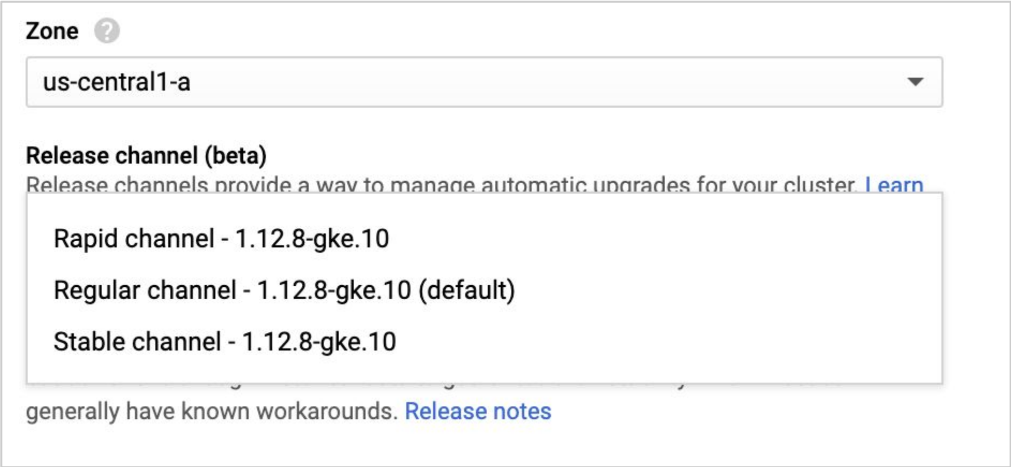


Rapid

Regular

Stable

```
gcloud container clusters create-auto [CLUSTER_NAME] --release-channel=regular
```



Zone

us-central1-a

Release channel (beta)

Release channels provide a way to manage automatic upgrades for your cluster. Learn

- Rapid channel - 1.12.8-gke.10
- Regular channel - 1.12.8-gke.10 (default)
- Stable channel - 1.12.8-gke.10

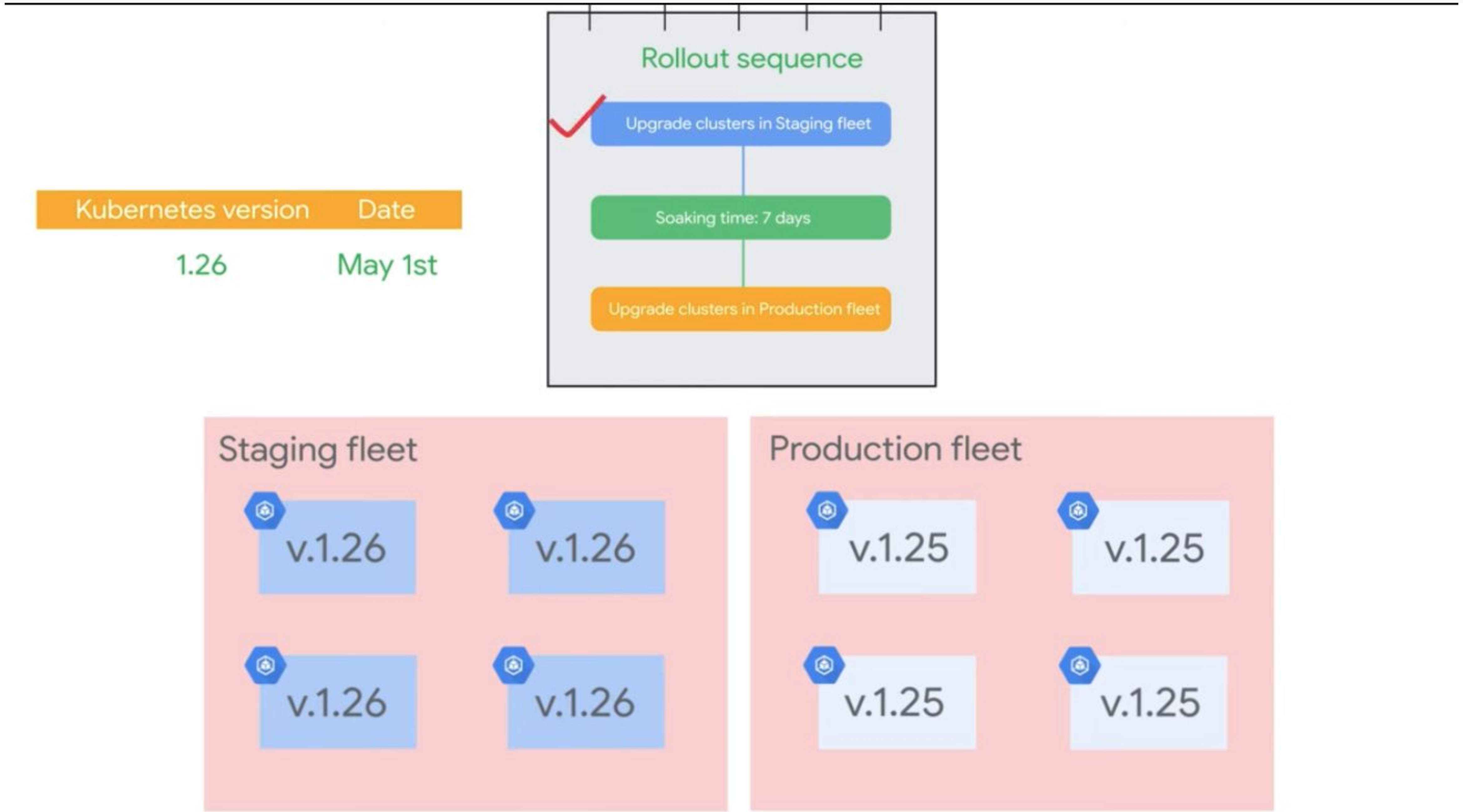
generally have known workarounds. Release notes

And you still have control: manual upgrades, maintenance windows, exclusions, and pod disruption budgets (PDBs) are still respected.

Rollout sequencing

Better predictability: manage the automated rollout sequence of new minor releases and patch versions among clusters

Always on reliability

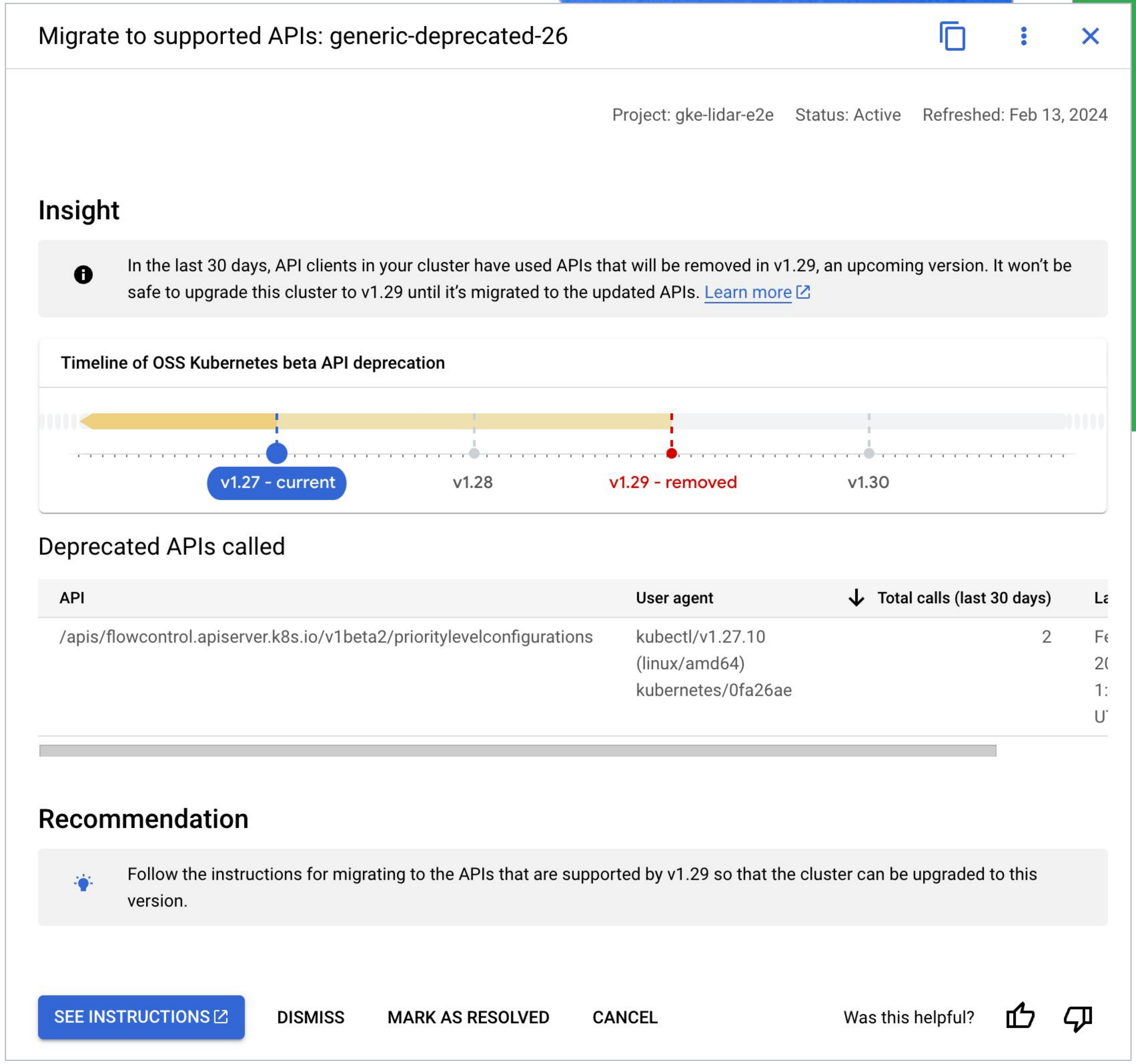


Mitigate deprecations

1. **Get insights about deprecated Kubernetes features and API usage by clusters and at org scale**

2. **Follow migration guides to migrate impacted clusters and unblock upgrades**

3. **Keep Beta APIs disabled by default to avoid future deprecations**



Migrate to supported APIs: generic-deprecated-26

Project: gke-lidar-e2e Status: Active Refreshed: Feb 13, 2024

Insight

In the last 30 days, API clients in your cluster have used APIs that will be removed in v1.29, an upcoming version. It won't be safe to upgrade this cluster to v1.29 until it's migrated to the updated APIs. Learn more

Timeline of OSS Kubernetes beta API deprecation

Deprecated APIs called

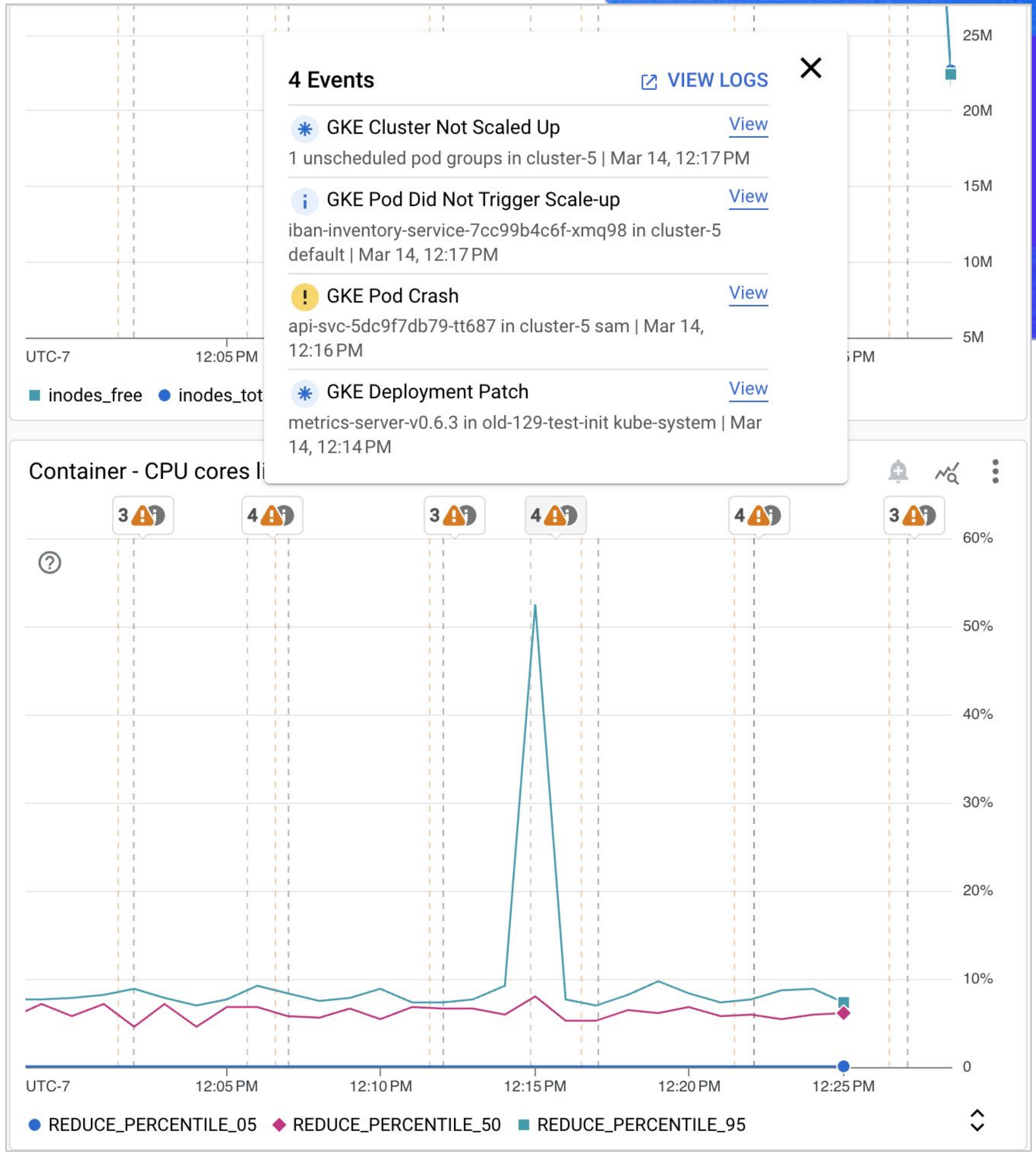
API	User agent	↓ Total calls (last 30 days)	La
/apis/flowcontrol.apiserver.k8s.io/v1beta2/prioritylevelconfigurations	kubectl/v1.27.10 (linux/amd64) kubernetes/0fa26ae	2	Fe 20 1: U

Recommendation

Follow the instructions for migrating to the APIs that are supported by v1.29 so that the cluster can be upgraded to this version.

Troubleshoot easily

- **Discover and resolve issues** using insights and Gemini assistance
- **Correlate metrics with events** using embedded event annotation
- **Follow interactive playbooks** to troubleshoot common issues such as Unschedulable Pods
- **Understand error logs**, possible causes and ways to troubleshoot by asking Gemini to explain log entry

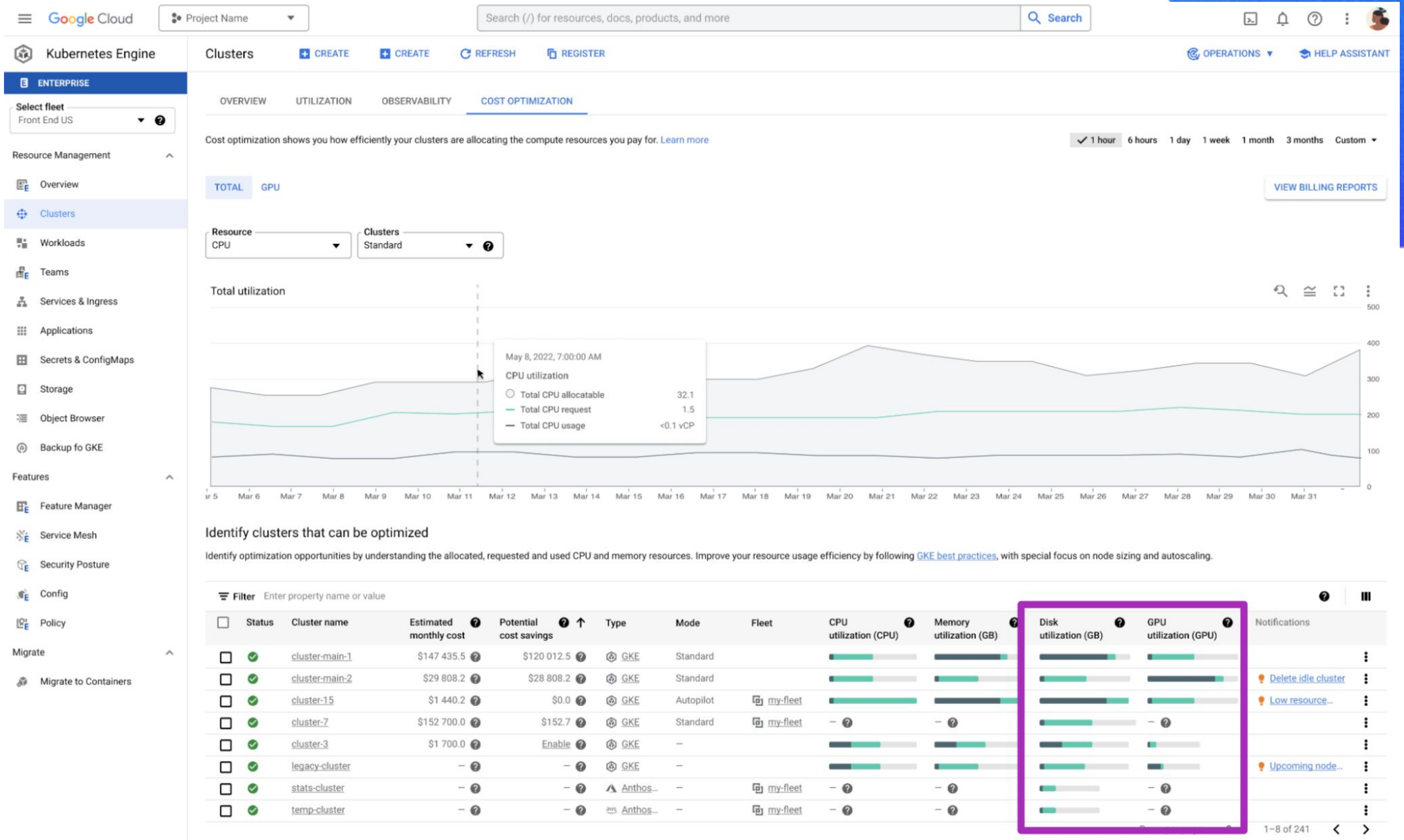


4 Events

VIEW LOGS

- GKE Cluster Not Scaled Up — View
  1 unscheduled pod groups in cluster-5 | Mar 14, 12:17 PM
- GKE Pod Did Not Trigger Scale-up — View
  iban-inventory-service-7cc99b4c6f-xmq98 in cluster-5 default | Mar 14, 12:17 PM
- GKE Pod Crash — View
  api-svc-5dc9f7db79-tt687 in cluster-5 sam | Mar 14, 12:16 PM
- GKE Deployment Patch — View
  metrics-server-v0.6.3 in old-129-test-init kube-system | Mar 14, 12:14 PM

Cluster cost optimization



Google Cloud | Project Name ▾ | Search (/) for resources, docs, products, and more | Search

Kubernetes Engine

ENTERPRISE

Select fleet: Front End US

Resource Management
- Overview
- Clusters
- Workloads
- Teams
- Services & Ingress
- Applications
- Secrets & ConfigMaps
- Storage
- Object Browser
- Backup fo GKE

Features
- Feature Manager
- Service Mesh
- Security Posture
- Config
- Policy

Migrate
- Migrate to Containers

Clusters | CREATE | CREATE | REFRESH | REGISTER | OPERATIONS ▾ | HELP ASSISTANT

OVERVIEW | UTILIZATION | OBSERVABILITY | COST OPTIMIZATION

Cost optimization shows you how efficiently your clusters are allocating the compute resources you pay for. Learn more

✓ 1 hour | 6 hours | 1 day | 1 week | 1 month | 3 months | Custom ▾

TOTAL | GPU | VIEW BILLING REPORTS

Resource: CPU

Clusters: Standard

Total utilization

May 8, 2022, 7:00:00 AM

CPU utilization

○ Total CPU allocatable	32.1
— Total CPU request	1.5
— Total CPU usage	<0.1 vCP

Identify clusters that can be optimized

Identify optimization opportunities by understanding the allocated, requested and used CPU and memory resources. Improve your resource usage efficiency by following GKE best practices, with special focus on node sizing and autoscaling.

Filter: Enter property name or value

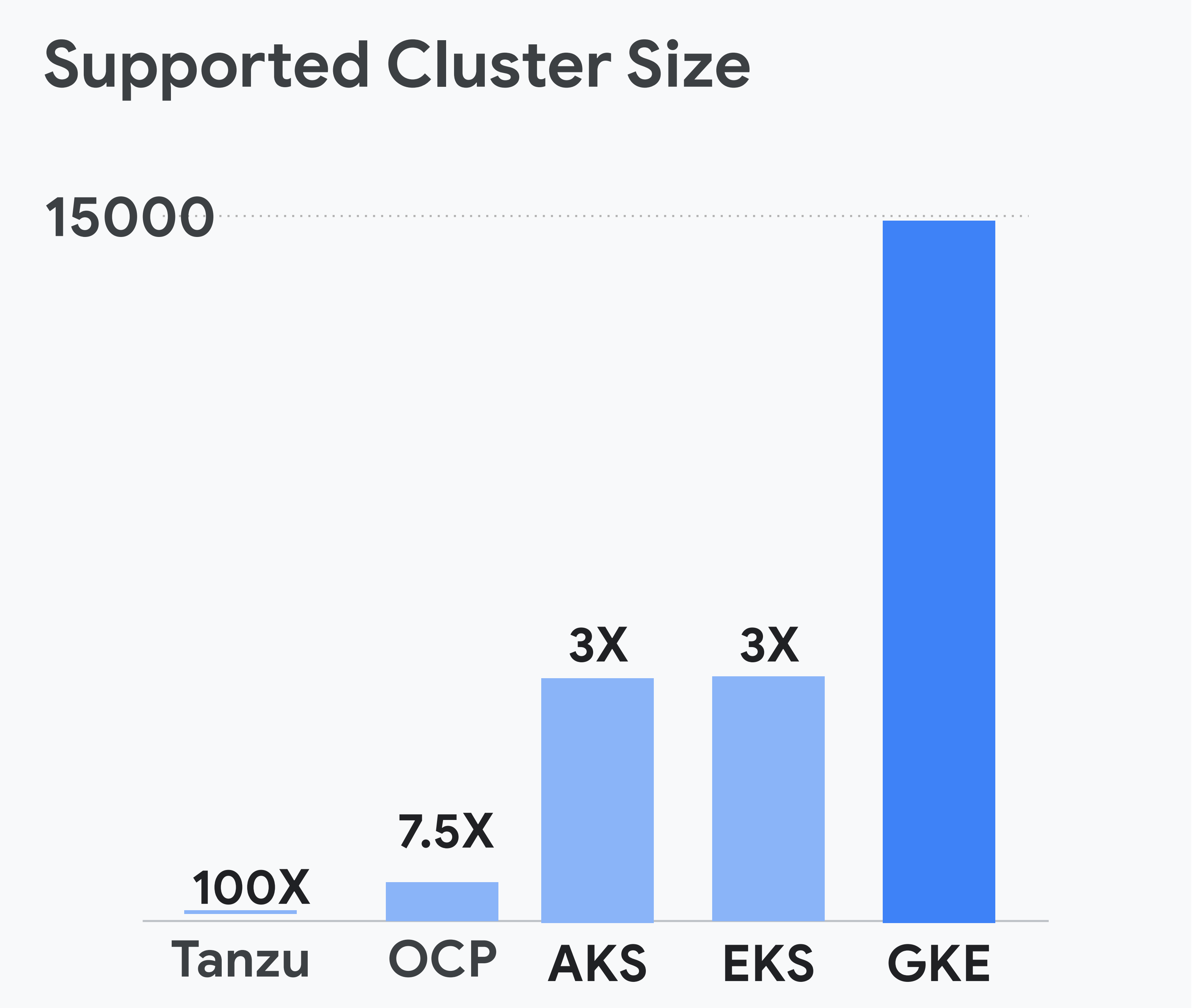
□	Status	Cluster name	Estimated monthly cost	Potential cost savings ↑	Type	Mode	Fleet	CPU utilization (CPU)	Memory utilization (GB)	Disk utilization (GB)	GPU utilization (GPU)	Notifications
□	✓	cluster-main-1	$147 435.5	$120 012.5	GKE	Standard		▬	▬	▬	▬	
□	✓	cluster-main-2	$29 808.2	$28 808.2	GKE	Standard		▬	▬	▬	▬	Delete idle cluster
□	✓	cluster-15	$1 440.2	$0.0	GKE	Autopilot	my-fleet	▬	▬	▬	▬	Low resource...
□	✓	cluster-7	$152 700.0	$152.7	GKE	Standard	my-fleet	–	–	▬	–	
□	✓	cluster-3	$1 700.0	Enable	GKE	–		▬	▬	▬	▬	
□	✓	legacy-cluster	–	–	GKE	–		▬	▬	▬	▬	Upcoming node...
□	✓	stats-cluster	–	–	Anthos...	–	my-fleet	–	–	▬	–	
□	✓	temp-cluster	–	–	Anthos...	–	my-fleet	–	–	▬	–	

1–8 of 241

Proprietary

GKE is the leader in scaling

GKE supports the largest and most scalable clusters in the industry



Scaling with fleet-based multi-team and multi-cluster management

Platform Administrator

- **Provision application teams** as tenants of a multi-cluster fleet.
- **Set per-tenant policies** for access, security and operational controls.
- **View per-team statistics** and recommendations.

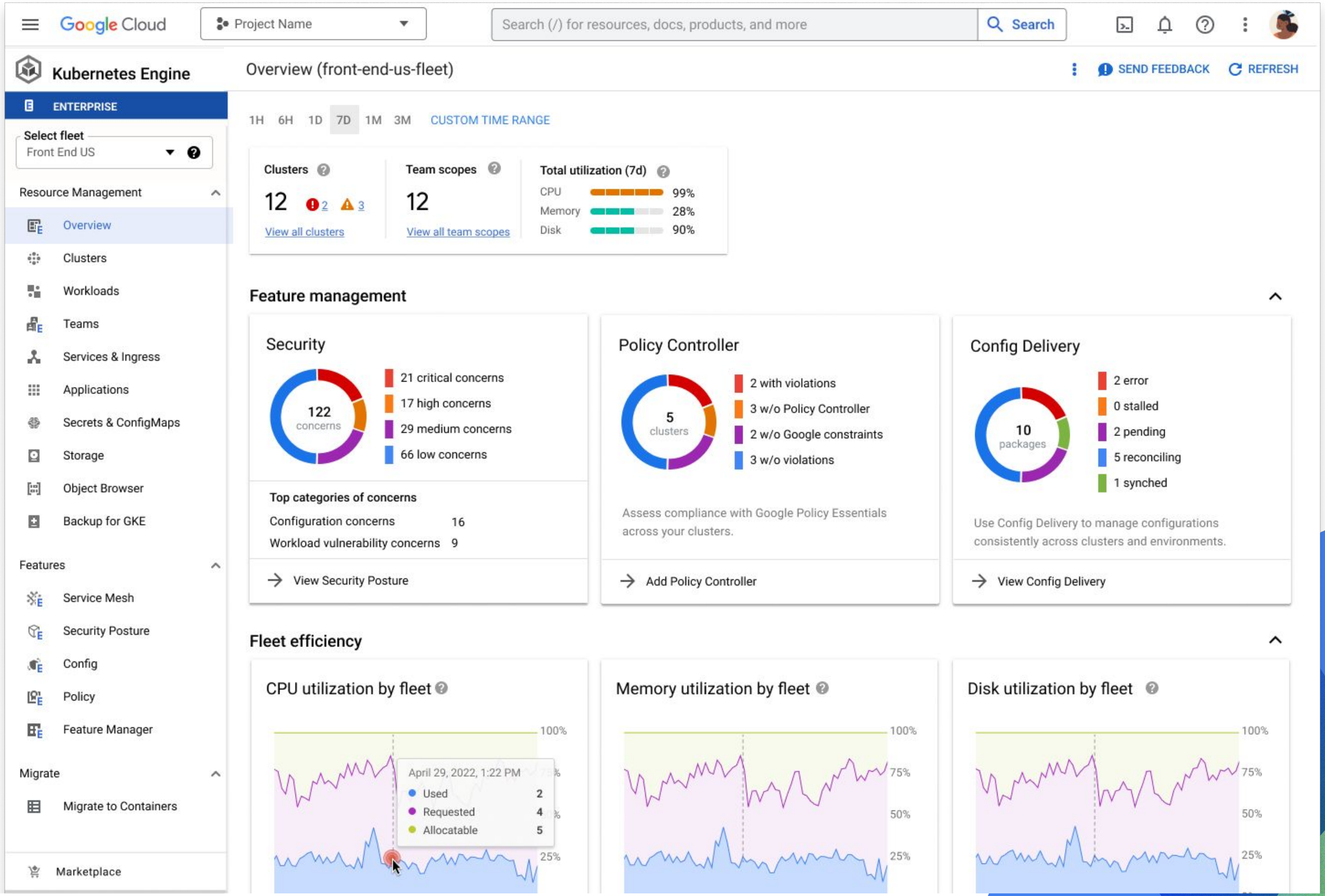
Application Operator

- **Self-service onboarding** and management of apps.
- **View workload** status, logs and metrics.
- **Manage cost, security and operational concerns** for individual applications.

Benefits

- Simplify multi cluster management
- Apply consistent config and policies at scale
- Self-service for application team agility

Multi cluster operations with fleet and Gitops based config



Google Cloud | Project Name | Search (/) for resources, docs, products, and more | Search

Kubernetes Engine

ENTERPRISE

Select fleet: Front End US

Resource Management
- Overview
- Clusters
- Workloads
- Teams
- Services & Ingress
- Applications
- Secrets & ConfigMaps
- Storage
- Object Browser
- Backup for GKE

Features
- Service Mesh
- Security Posture
- Config
- Policy
- Feature Manager

Migrate
- Migrate to Containers

Marketplace

Overview (front-end-us-fleet) | SEND FEEDBACK | REFRESH

1H 6H 1D 7D 1M 3M CUSTOM TIME RANGE

Clusters: 12 — 2 — 3 — View all clusters

Team scopes: 12 — View all team scopes

Total utilization (7d)

CPU	99%
Memory	28%
Disk	90%

Feature management

Security

122 concerns
- 21 critical concerns
- 17 high concerns
- 29 medium concerns
- 66 low concerns

Top categories of concerns

Configuration concerns	16
Workload vulnerability concerns	9

→ View Security Posture

Policy Controller

5 clusters
- 2 with violations
- 3 w/o Policy Controller
- 2 w/o Google constraints
- 3 w/o violations

Assess compliance with Google Policy Essentials across your clusters.

→ Add Policy Controller

Config Delivery

10 packages
- 2 error
- 0 stalled
- 2 pending
- 5 reconciling
- 1 synched

Use Config Delivery to manage configurations consistently across clusters and environments.

→ View Config Delivery

Fleet efficiency

CPU utilization by fleet

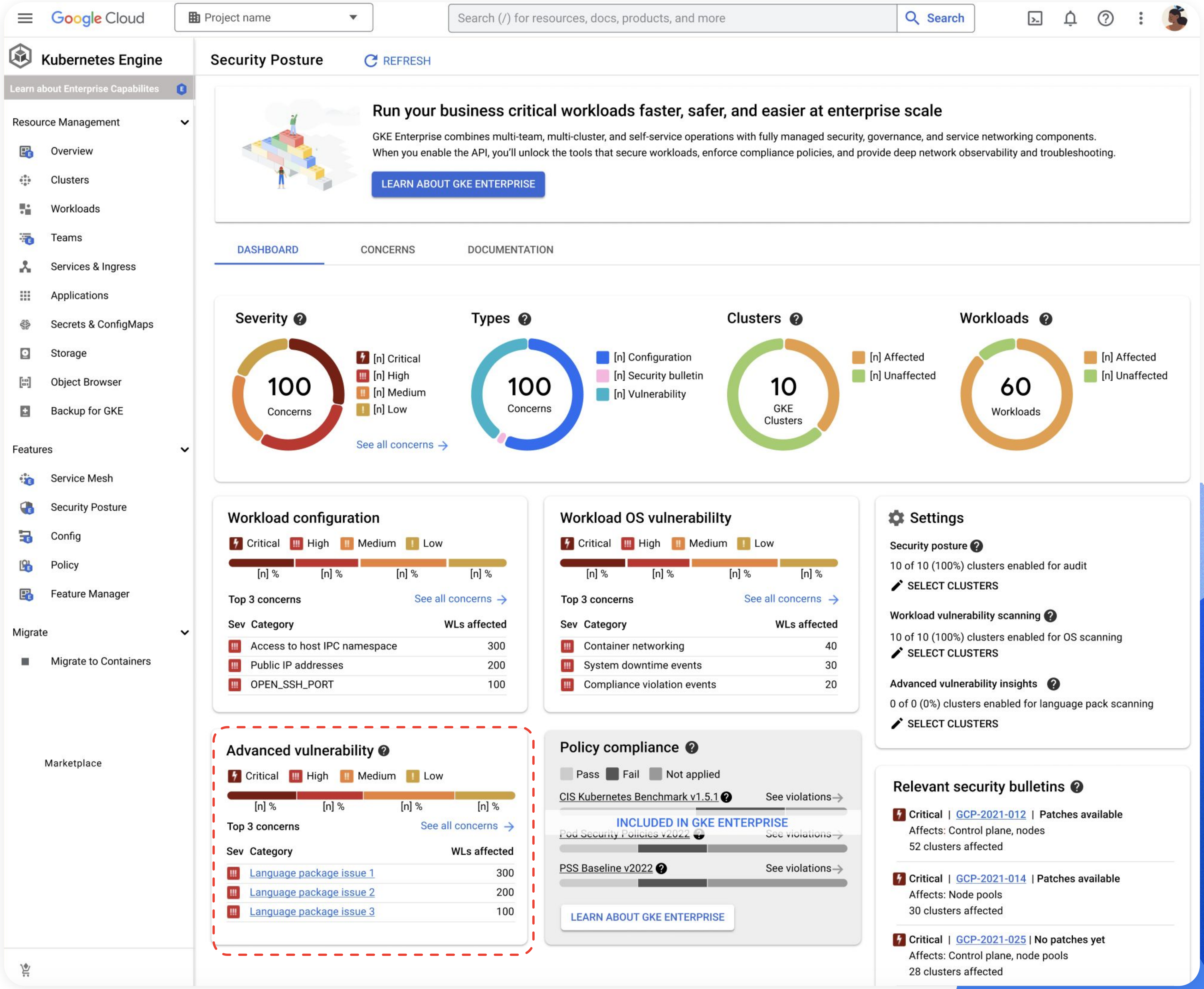
April 29, 2022, 1:22 PM

Used	2
Requested	4
Allocatable	5

Memory utilization by fleet

Disk utilization by fleet

Kubernetes Security Posture dashboard



Google Cloud | Project name | Search (/) for resources, docs, products, and more | Search

Kubernetes Engine

Learn about Enterprise Capabilites

Resource Management
- Overview
- Clusters
- Workloads
- Teams
- Services & Ingress
- Applications
- Secrets & ConfigMaps
- Storage
- Object Browser
- Backup for GKE

Features
- Service Mesh
- Security Posture
- Config
- Policy
- Feature Manager

Migrate
- Migrate to Containers

Marketplace

Security Posture REFRESH

Run your business critical workloads faster, safer, and easier at enterprise scale

GKE Enterprise combines multi-team, multi-cluster, and self-service operations with fully managed security, governance, and service networking components. When you enable the API, you'll unlock the tools that secure workloads, enforce compliance policies, and provide deep network observability and troubleshooting.

LEARN ABOUT GKE ENTERPRISE

DASHBOARD | CONCERNS | DOCUMENTATION

Severity
100 Concerns
- [n] Critical
- [n] High
- [n] Medium
- [n] Low

See all concerns →

Types
100 Concerns
- [n] Configuration
- [n] Security bulletin
- [n] Vulnerability

Clusters
10 GKE Clusters
- [n] Affected
- [n] Unaffected

Workloads
60 Workloads
- [n] Affected
- [n] Unaffected

Workload configuration

Critical | High | Medium | Low
[n] % | [n] % | [n] % | [n] %

Top 3 concerns — See all concerns →

Sev	Category	WLs affected
‼	Access to host IPC namespace	300
‼	Public IP addresses	200
‼	OPEN_SSH_PORT	100

Workload OS vulnerability

Critical | High | Medium | Low
[n] % | [n] % | [n] % | [n] %

Top 3 concerns — See all concerns →

Sev	Category	WLs affected
‼	Container networking	40
‼	System downtime events	30
‼	Compliance violation events	20

Settings

Security posture
10 of 10 (100%) clusters enabled for audit
SELECT CLUSTERS

Workload vulnerability scanning
10 of 10 (100%) clusters enabled for OS scanning
SELECT CLUSTERS

Advanced vulnerability insights
0 of 0 (0%) clusters enabled for language pack scanning
SELECT CLUSTERS

Advanced vulnerability

Critical | High | Medium | Low
[n] % | [n] % | [n] % | [n] %

Top 3 concerns — See all concerns →

Sev	Category	WLs affected
‼	Language package issue 1	300
‼	Language package issue 2	200
‼	Language package issue 3	100

Policy compliance

Pass | Fail | Not applied

CIS Kubernetes Benchmark v1.5.1 — See violations →

INCLUDED IN GKE ENTERPRISE

Pod Security Policies v2022 — See violations →

PSS Baseline v2022 — See violations →

LEARN ABOUT GKE ENTERPRISE

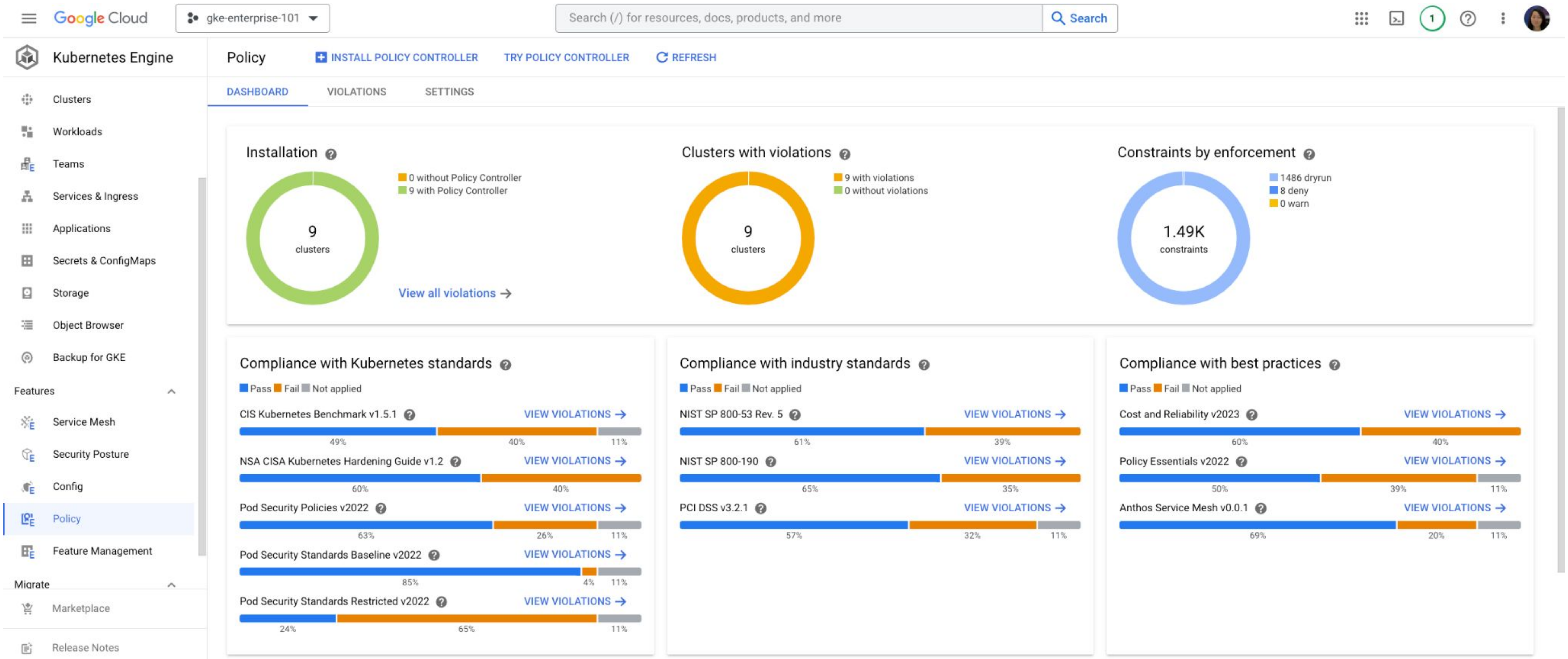
Relevant security bulletins

Critical | GCP-2021-012 | Patches available
Affects: Control plane, nodes
52 clusters affected

Critical | GCP-2021-014 | Patches available
Affects: Node pools
30 clusters affected

Critical | GCP-2021-025 | No patches yet
Affects: Control plane, node pools
28 clusters affected

Policy Controller integrated in GKE



Policy

Installation

9 clusters

🟧 0 without Policy Controller
🟩 9 with Policy Controller

View all violations →

Clusters with violations

9 clusters

🟧 9 with violations
🟩 0 without violations

Constraints by enforcement

1.49K constraints

🟦 1486 dryrun
🟦 8 deny
🟨 0 warn

Compliance with Kubernetes standards

🟦 Pass 🟧 Fail ⬜ Not applied

Standard	Pass	Fail	Not applied	
CIS Kubernetes Benchmark v1.5.1	49%	40%	11%	VIEW VIOLATIONS →
NSA CISA Kubernetes Hardening Guide v1.2	60%	40%		VIEW VIOLATIONS →
Pod Security Policies v2022	63%	26%	11%	VIEW VIOLATIONS →
Pod Security Standards Baseline v2022	85%	4%	11%	VIEW VIOLATIONS →
Pod Security Standards Restricted v2022	24%	65%	11%	VIEW VIOLATIONS →

Compliance with industry standards

🟦 Pass 🟧 Fail ⬜ Not applied

Standard	Pass	Fail	Not applied	
NIST SP 800-53 Rev. 5	61%	39%		VIEW VIOLATIONS →
NIST SP 800-190	65%	35%		VIEW VIOLATIONS →
PCI DSS v3.2.1	57%	32%	11%	VIEW VIOLATIONS →

Compliance with best practices

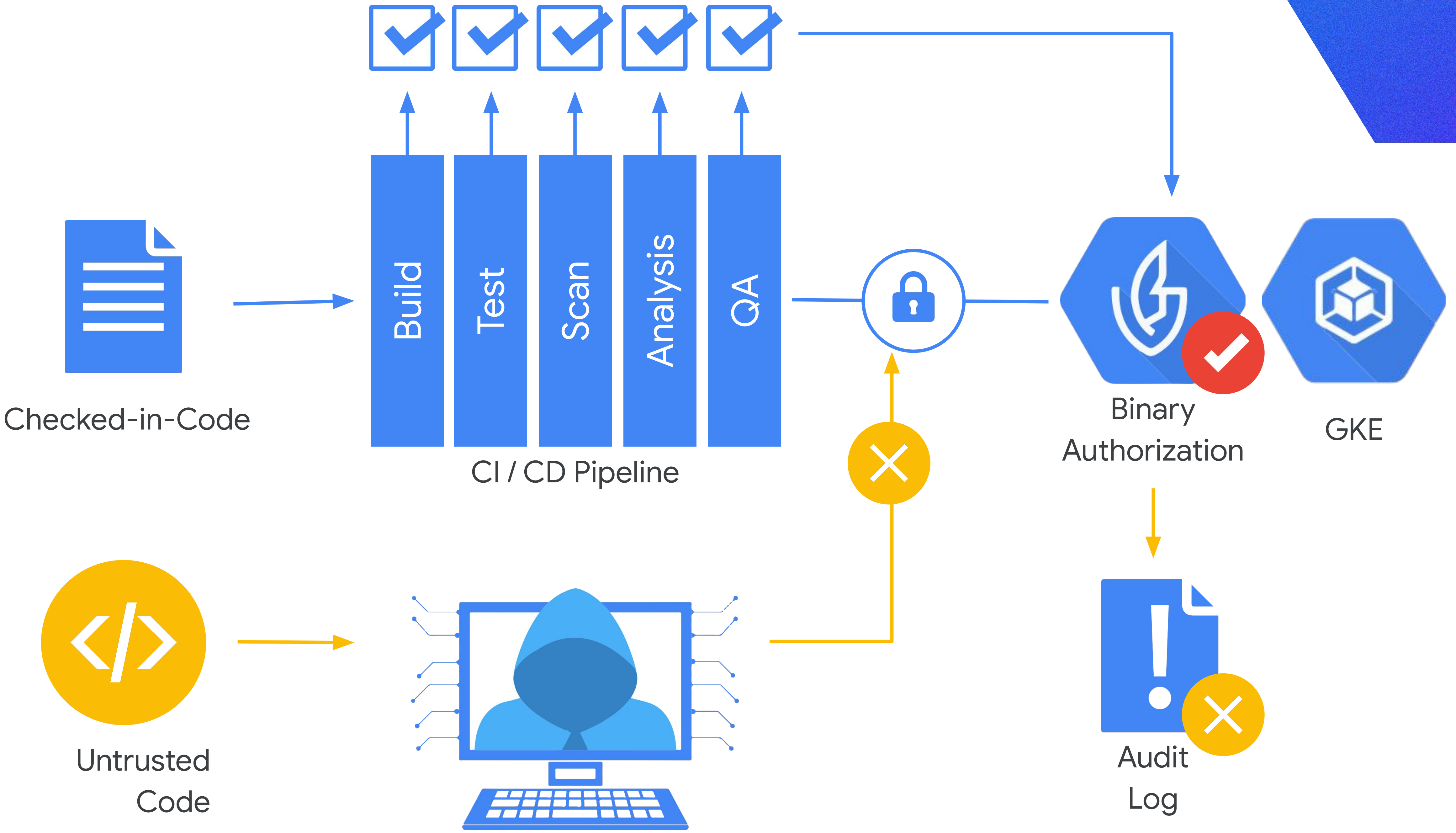
🟦 Pass 🟧 Fail ⬜ Not applied

Standard	Pass	Fail	Not applied	
Cost and Reliability v2023	60%	40%		VIEW VIOLATIONS →
Policy Essentials v2022	50%	39%	11%	VIEW VIOLATIONS →
Anthos Service Mesh v0.0.1	69%	20%	11%	VIEW VIOLATIONS →

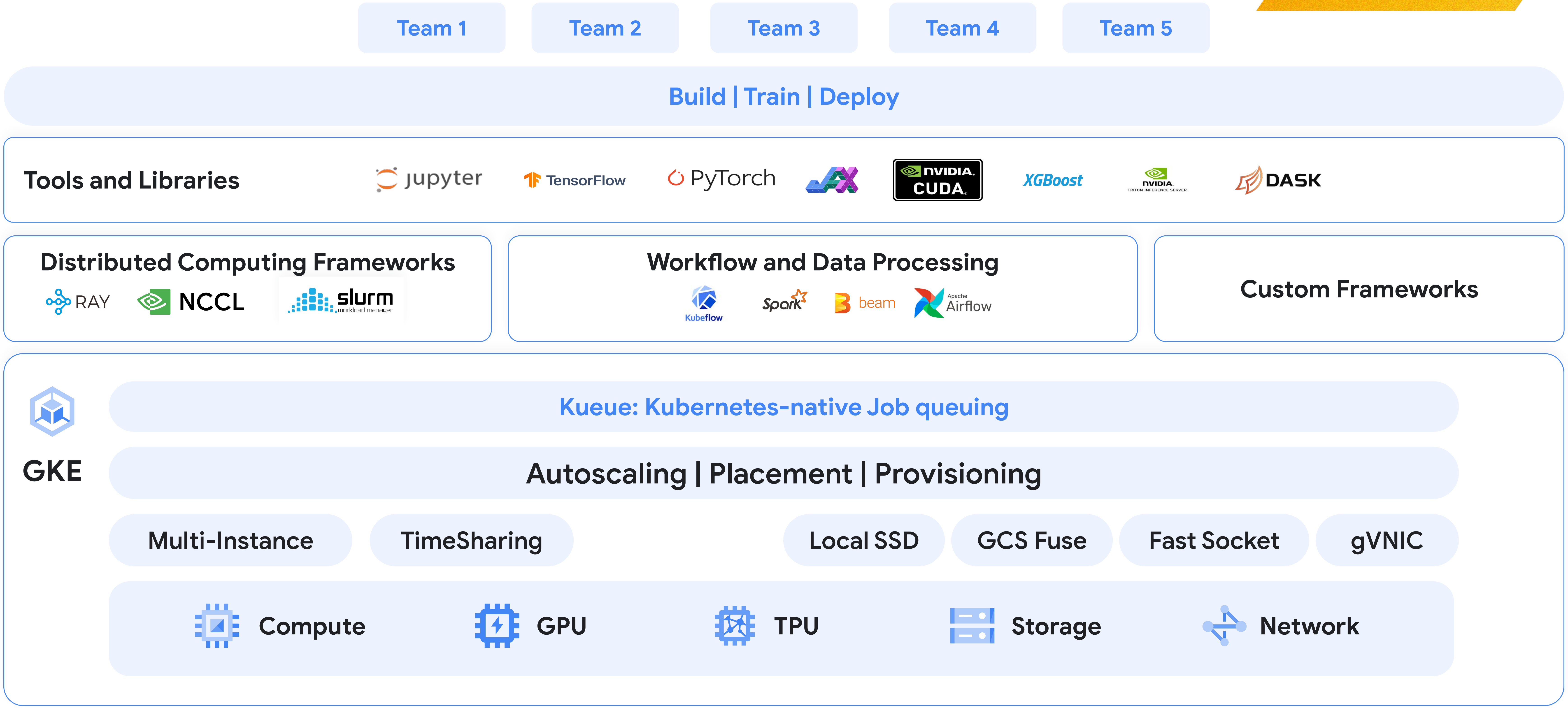
Binary Authorization

Deploy only what you trust

- **Pluggable**, **open** sourced attestation framework
- Integrated with CloudBuild and GCR Vulnerability scan
- Set allow list for 3rd party images.
- In case of emergency, **Break glass**.
- Flexible policy granularity: per project, cluster, identity/namespace (preview)
- **Native integration** with GKE



Unified AI/ML platform for GKE



Team 1 | Team 2 | Team 3 | Team 4 | Team 5

Build | Train | Deploy

Tools and Libraries

jupyter · TensorFlow · PyTorch · JAX · NVIDIA CUDA · XGBoost · NVIDIA TRITON INFERENCE SERVER · DASK

Distributed Computing Frameworks

RAY · NCCL · slurm workload manager

Workflow and Data Processing

Kubeflow · Spark · beam · Apache Airflow

Custom Frameworks

GKE

Kueue: Kubernetes-native Job queuing

Autoscaling | Placement | Provisioning

Multi-Instance · TimeSharing · Local SSD · GCS Fuse · Fast Socket · gVNIC

Compute · GPU · TPU · Storage · Network

AI fast startup

- **Pain point**: AI/ML container images can be very large (20GB+), making them very slow to load.
- **Solution**: Cache the container image on a secondary boot disk.
- **Also works** to cache data such as ML models, weights, etc.
- **Near-constant** latency even at massive scale.
- **In GKE**, enable with a single flag:
  `--secondary-boot-disk`

up to

29x

faster time to mount a 16GB container into *Running* status (from 271 to 9 seconds)

Serve using multiple NVIDIA L4 GPUs

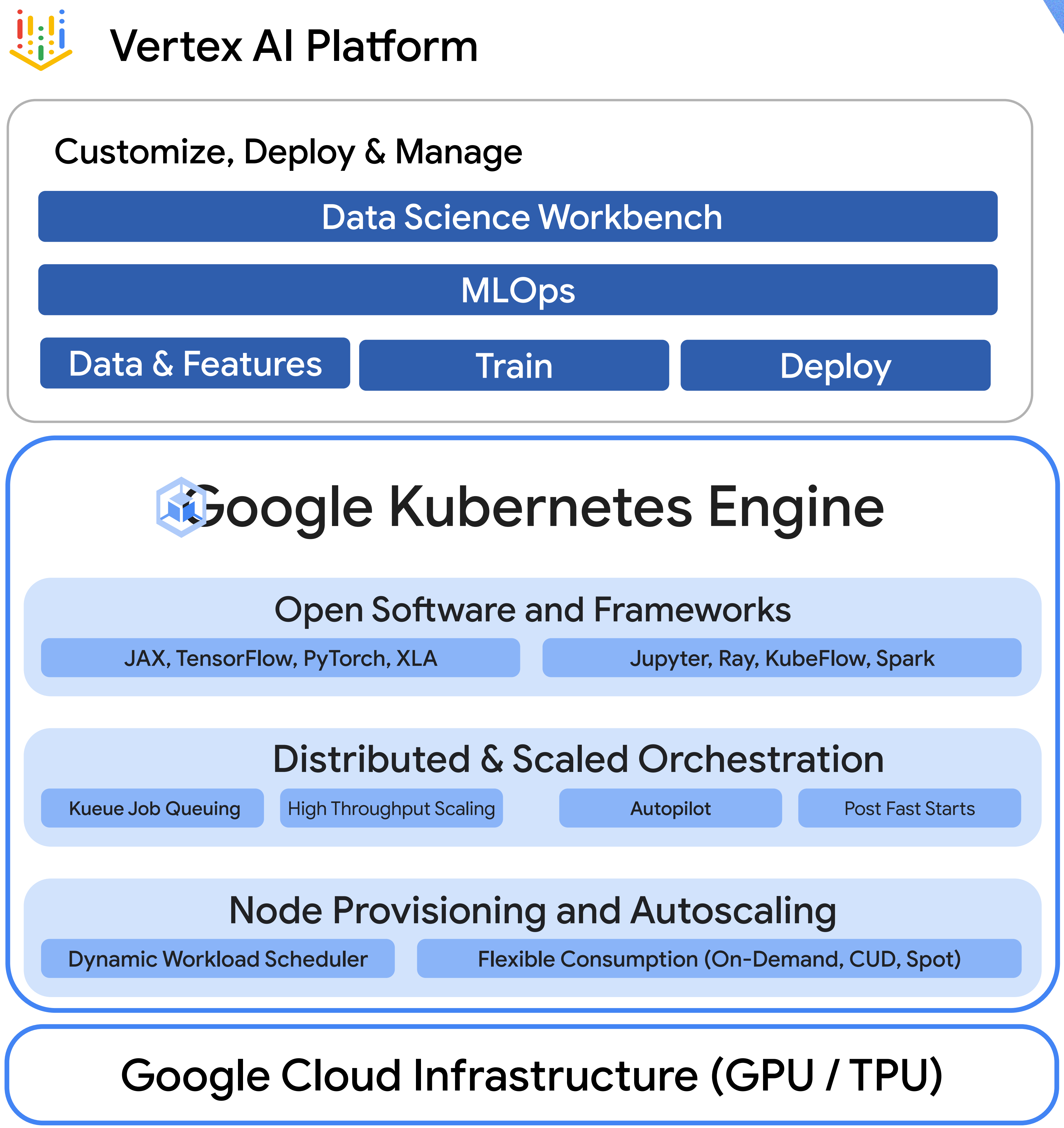
- **Pain point**: NVIDIA A100 or H100 GPUs are very expensive and hard to obtain.
- **Solution**: Shard your ML model and serve it using two or three L4 GPUs (each contains 24GB memory).
- **Save money**: a single L4 offers 30% of the memory of H100 at 1% of the price.
- **Mild latency increase** using L4 GPUs, depending on the ML model.

up to

50%

cost savings switching from an A100 GPU to multiple L4 GPUs

Vertex’s innovative managed AI platform leverages GKE



Vertex AI Platform

Customize, Deploy & Manage

- Data Science Workbench
- MLOps
- Data & Features
- Train
- Deploy

Google Kubernetes Engine

Open Software and Frameworks

- JAX, TensorFlow, PyTorch, XLA
- Jupyter, Ray, KubeFlow, Spark

Distributed & Scaled Orchestration

- Kueue Job Queuing
- High Throughput Scaling
- Autopilot
- Post Fast Starts

Node Provisioning and Autoscaling

- Dynamic Workload Scheduler
- Flexible Consumption (On-Demand, CUD, Spot)

Google Cloud Infrastructure (GPU / TPU)

Spectrum of Stateful Apps on GKE

Do it yourself (DIY)

Eg. Redis, MariaDB, postgresql

Apps deployed as container images and managed by customers

Self Managed

Kubernetes Operator

Eg. Elastic operator

Apps deployed as container images with management shared with operator contracts.

Partially Managed

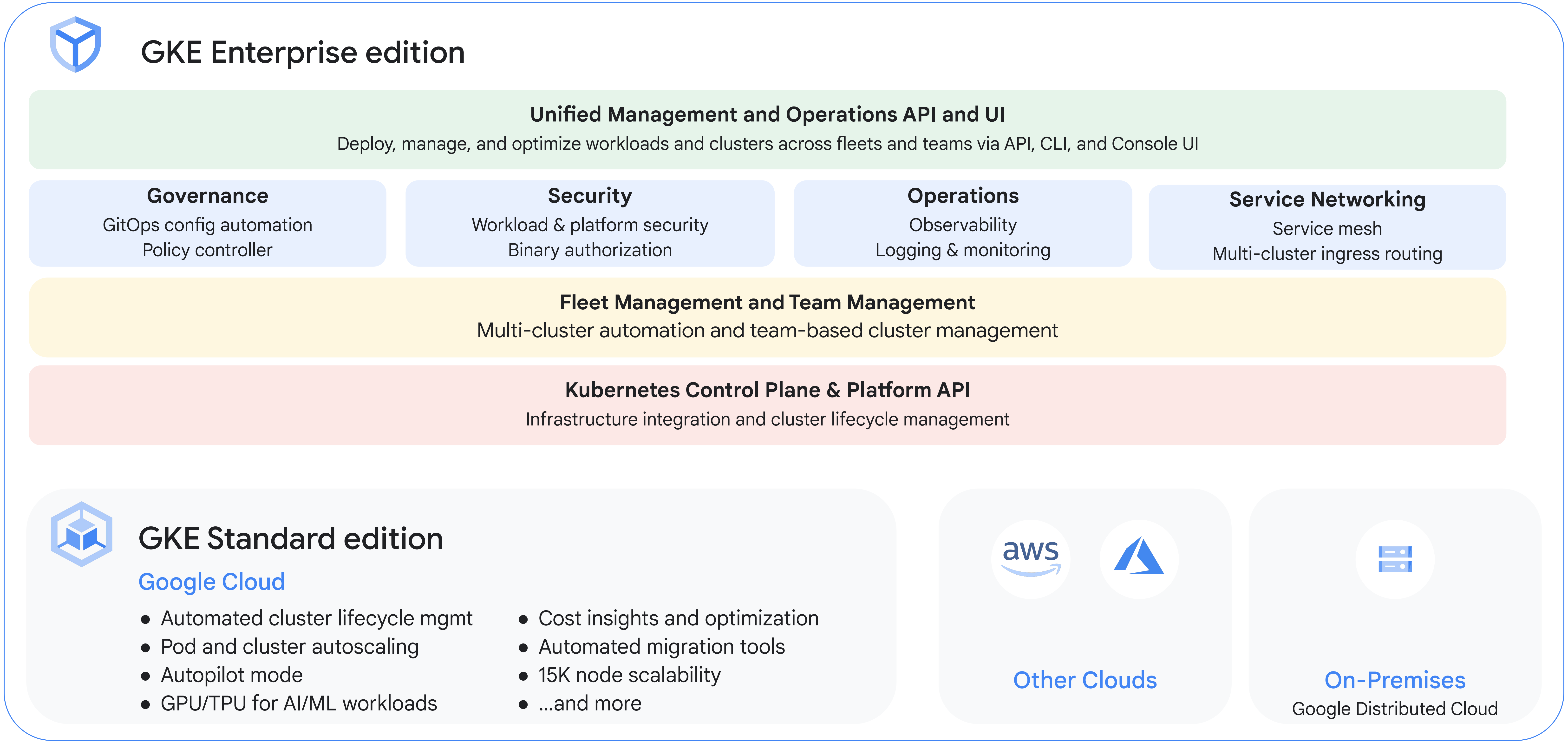
Data SaaS

Eg. MariaDB SkySQL

Apps that are fully managed Saas solutions for end users

Fully Managed

One GKE experience



GKE Enterprise edition

Unified Management and Operations API and UI

Deploy, manage, and optimize workloads and clusters across fleets and teams via API, CLI, and Console UI

Governance

GitOps config automation

Policy controller

Security

Workload & platform security

Binary authorization

Operations

Observability

Logging & monitoring

Service Networking

Service mesh

Multi-cluster ingress routing

Fleet Management and Team Management

Multi-cluster automation and team-based cluster management

Kubernetes Control Plane & Platform API

Infrastructure integration and cluster lifecycle management

GKE Standard edition

Google Cloud

- Automated cluster lifecycle mgmt
- Pod and cluster autoscaling
- Autopilot mode
- GPU/TPU for AI/ML workloads
- Cost insights and optimization
- Automated migration tools
- 15K node scalability
- ...and more

Other Clouds

On-Premises

Google Distributed Cloud

The future of GKE

Compute Classes

Advanced node config options, including fall-back priorities with active reconciliation abstracted to a single node selector in the workload

Node selection prioritization

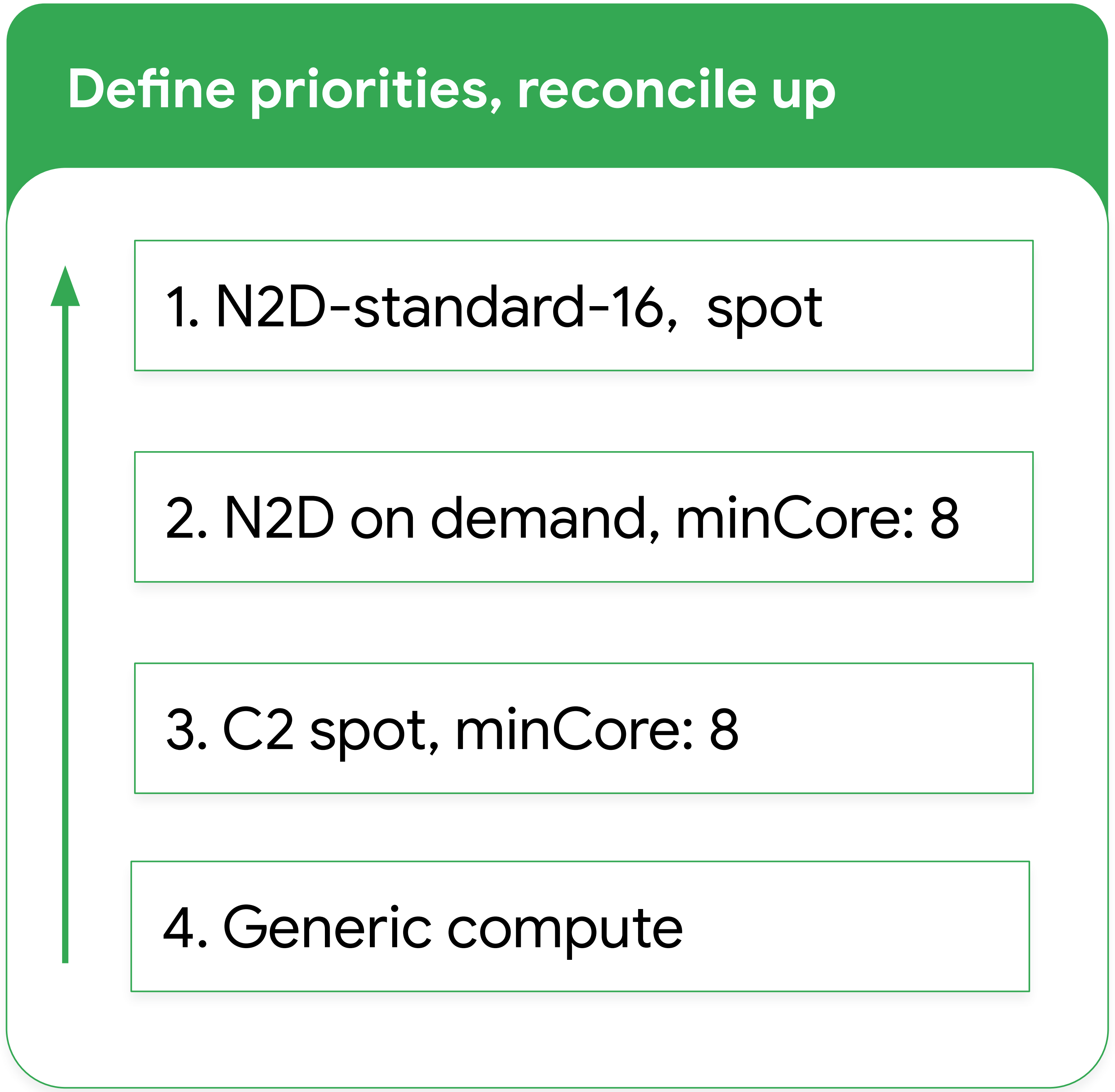
- **Fall-back** priorities for nodes
- **Spot** priorities with **fall-backs**
- Define by **instance characteristics** (machine/ family/ size)
- GPU/TPU support
- **Scaling** profiles
- Named GCE **reservations**

Active reconciliation to top priorities

- Reconcile workloads to top priorities
- Subject to TTL, PDB, etc

Default classes

- Override Autopilot default class per namespace
- Even without nodeSelectors, workloads get desired node config



Define priorities, reconcile up

1. N2D-standard-16, spot
2. N2D on demand, minCore: 8
3. C2 spot, minCore: 8
4. Generic compute

Scaling with compute classes

```
apiVersion: autoscaling.gke.io/v1alpha1
kind: ComputeClass
metadata:
 name: custom-config
spec:
activeMigration:
    optimizeRulePriority : true
nodePoolAutoCreation:
    enabled              : true

priorities:
-   machineType    : n2d-standard-16
    spot           : true

-   family         : c2
    spot           : true
    minCores       : 8

-   family         : n2d
    spot           : false
    minCores       : 8
```

Private Preview
(code will change)

```
apiVersion: v1
kind: Pod
metadata:
 name: nginx
 labels:
   pod: nginx-pod
spec:
 nodeSelector:
   cloud.google.com/compute-class: custom-config
 containers:
   - image: nginx
     name: nginx-container
```

Dynamic workload scheduler

New obtainability capabilities for accelerators

Works across GCP

- Managed Instance Groups on GCE
- Batch on GCE
- GKE
- Vertex AI

Calendar Mode:

Job start times assurance with Future Reservations

Use Cases:

(re)training, recurring fine-tuning

GPUs

Flex Start Mode:

Optimized economics and higher obtainability for on-demand resources

Use Cases:

time flexible experiments, fine tuning, batch inference

GPUs & TPUs

> *"The new DWS scheduling capabilities have been a game-changer in procuring sufficient GPU capacity for our training runs. We didn't have to worry about wasting money on idle GPUs while refreshing the page hoping for sufficient compute resources to become available."*
>
> **- Sahil Chopra, Co-Founder & CEO, Linum AI**

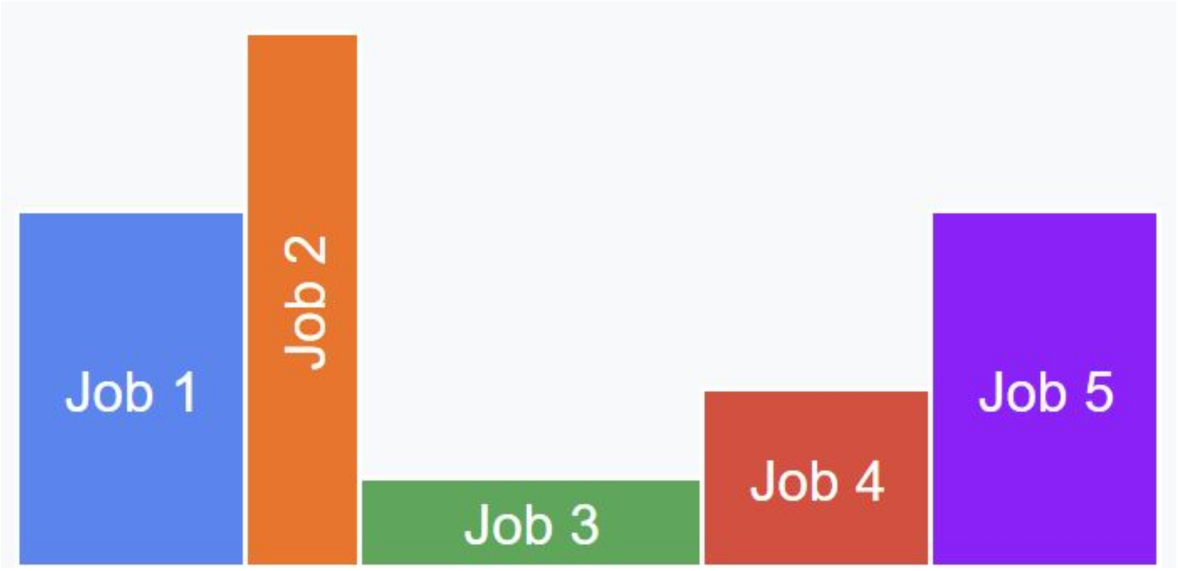
Flex Start mode: AI/ML workloads get served in order of arrival

User scenario:
"I want to run my multi-node training job using 2 A2 VMs with 8 GPUs for 15h in us-central1"

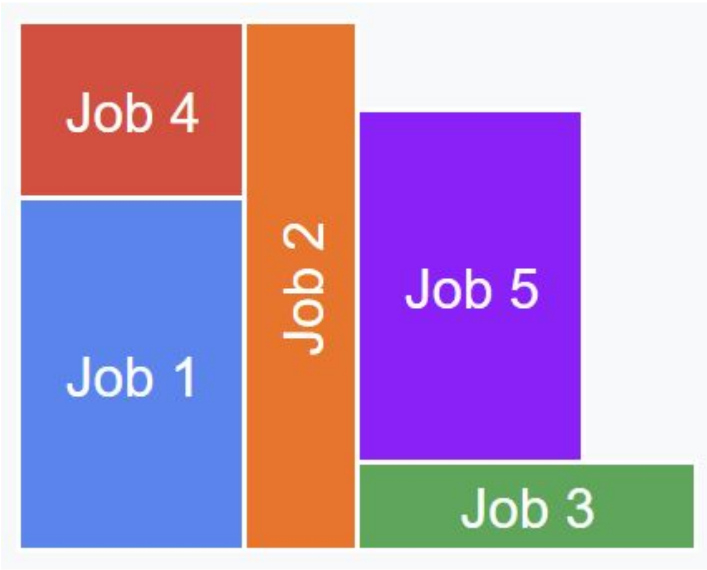
Job parameters:

- Resource quantity (VM count)
- Location (Region or Zone)
- Run duration (default max 7 days)

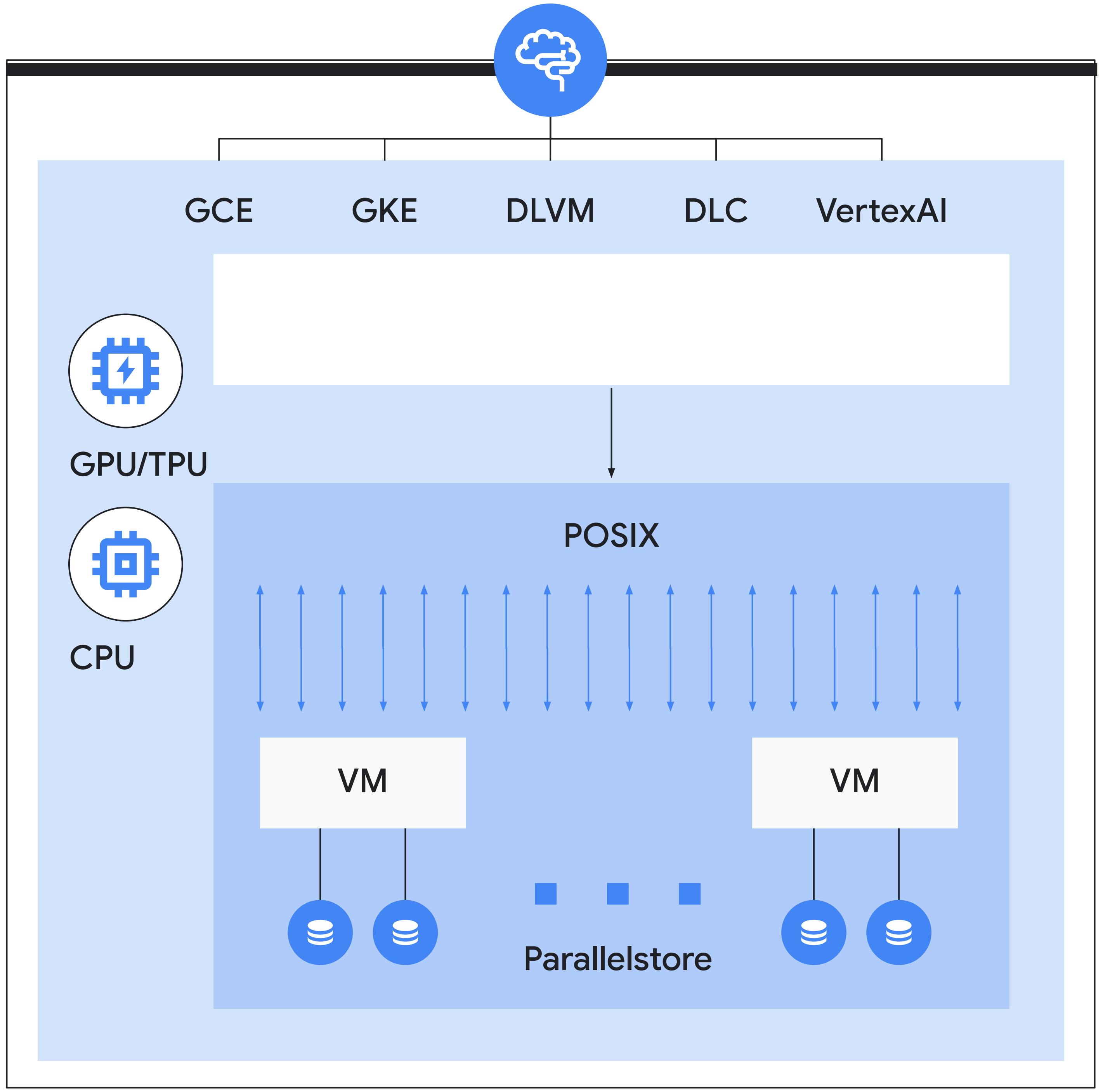
Job queue - requests by arrival time



Capacity usage



Stateful and Training workloads with Parallel Store



Next Generation Parallel File System

- Accelerate high-performance applications that require both high scale and low latency data access
- Maximize GPU/TPU utilization as data is always available

High Performance

- Up to 6.3x read throughput performance compared to competitive Lustre Scratch offerings ~200MB/sec per TB (read)
- Ultra low latency (~0.3ms) and ultra high performing (millions of IOPS and metadata operations)

AI Optimized Architecture

- Distributed metadata management, extreme IOPS, and Key Value architecture are necessary for demanding AI/ML workloads

Powerful Operations

- Integrated data protection across servers to improve availability
- Data transfer from Cloud Storage at 10 GB/s++

kubernetes turns 10!

#k8sturns10

We are interested in your feedback!

Connect with a GKE/Serverless PM or UX researcher.

Thank you