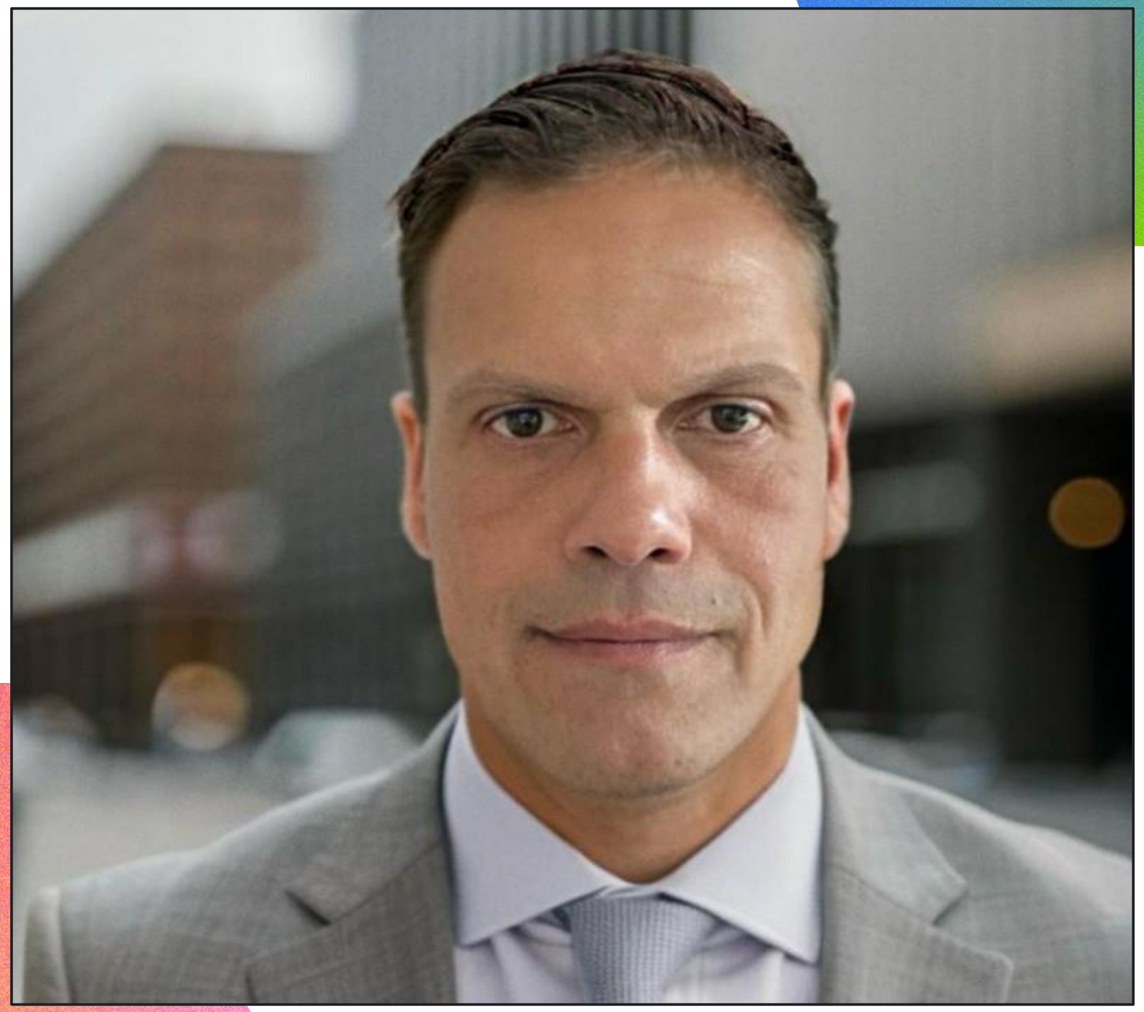
Google Cloud

Next '24

Run workloads not infrastructure with Google Kubernetes Engine

Gari Singh

Product Manager, Google Cloud



Google Kubernetes Engine is the fully managed, most automated and scalable Kubernetes platform run by the largest contributor to Kubernetes.

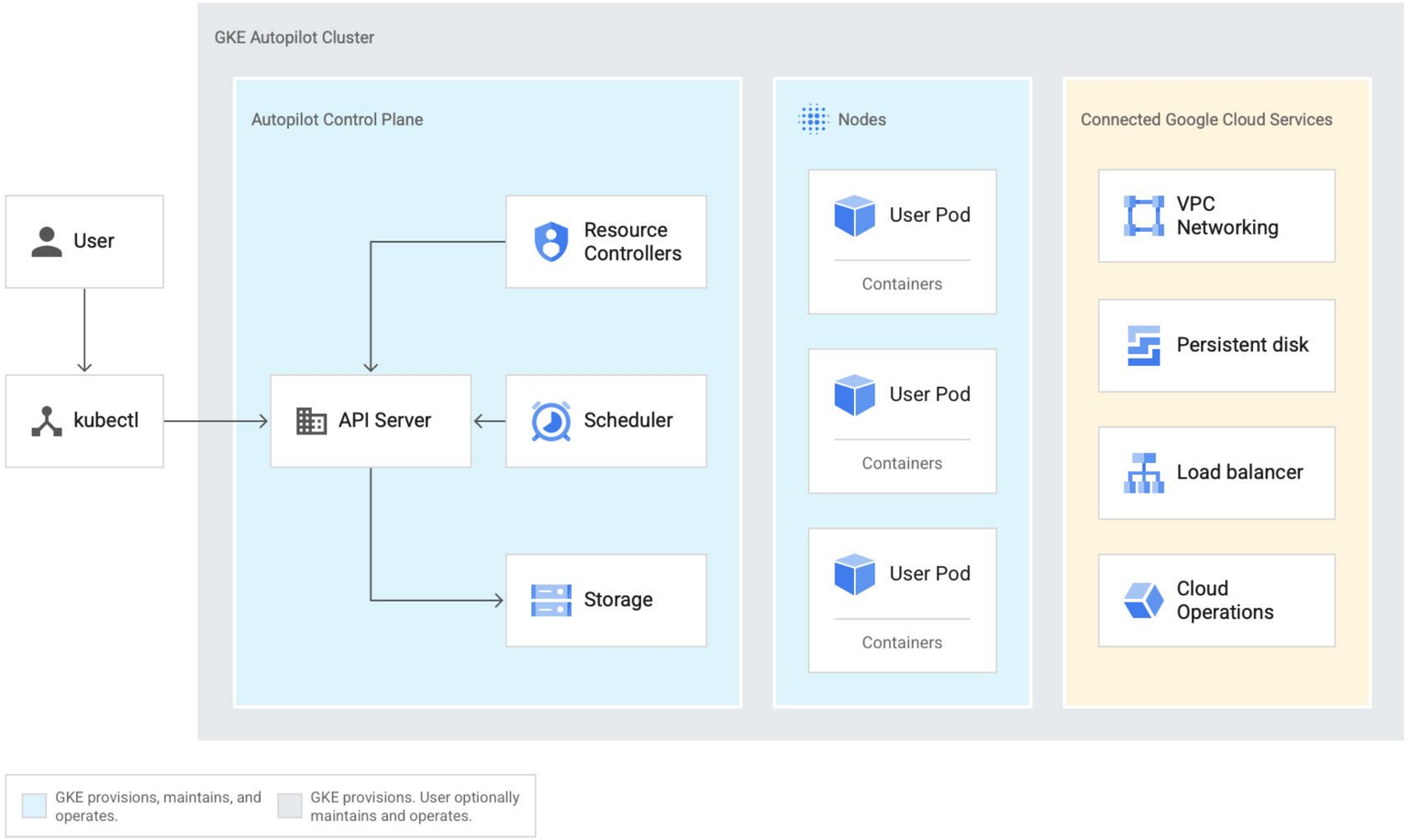
Fully Managed

GKE Architecture

Highly available with regional clusters

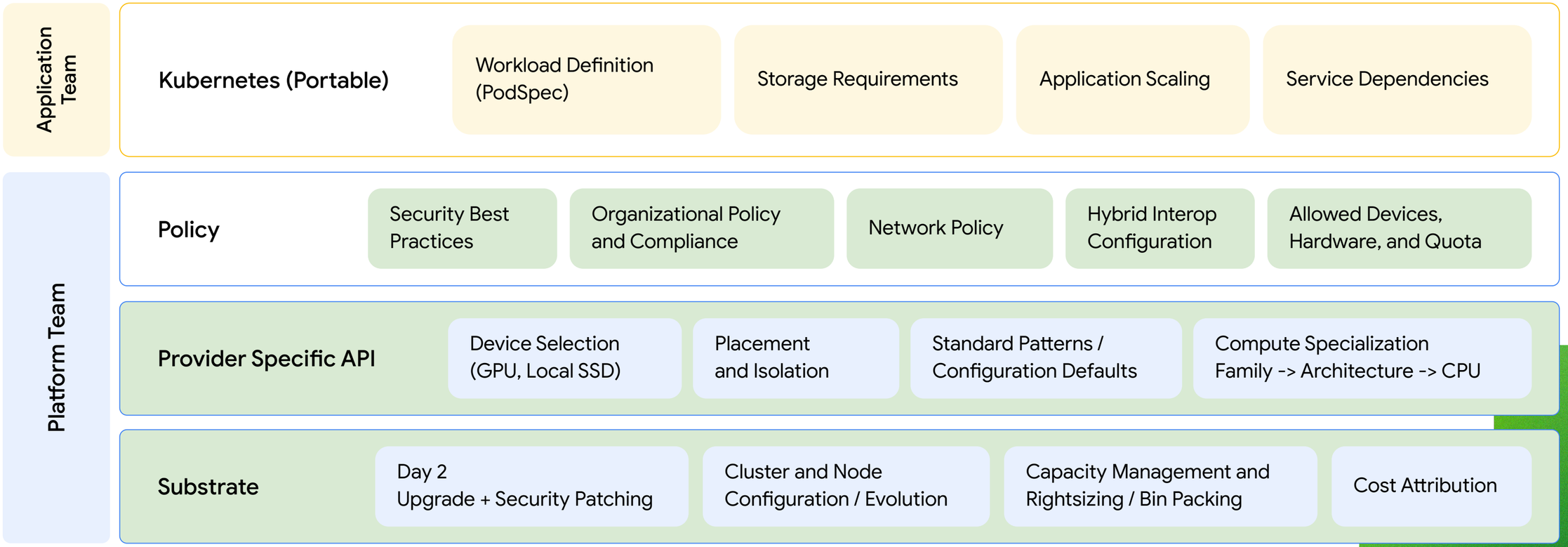
Key benefits

- Rich, powerful UI
- SRE monitoring
- Automated repair of apps
- Resource optimized app deployments
- Load balancing & auto-scaling of resources
- Global Virtual Private Cloud
- SLA for entire cluster

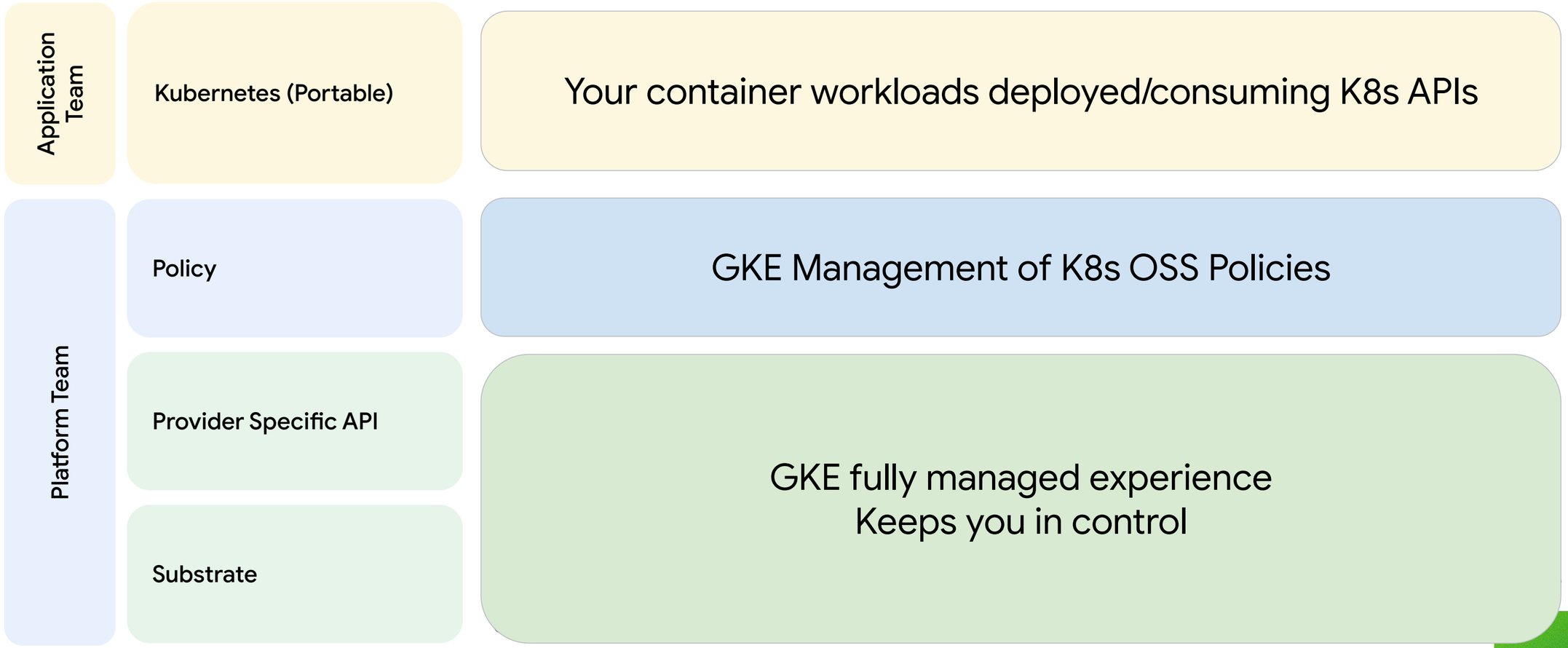


Layers of a Kubernetes Platform

To accommodate all but the simplest workloads, platform teams must also provide a layer of translation to expose provider specific capabilities necessary to fit advanced workload requirements.

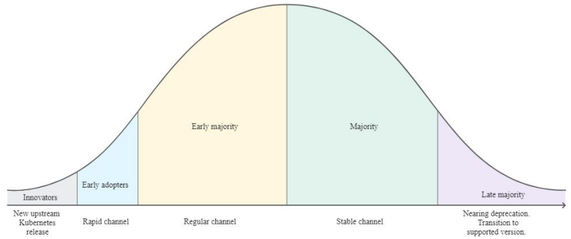


GKE | Accelerator for Platform Teams



Release Channels

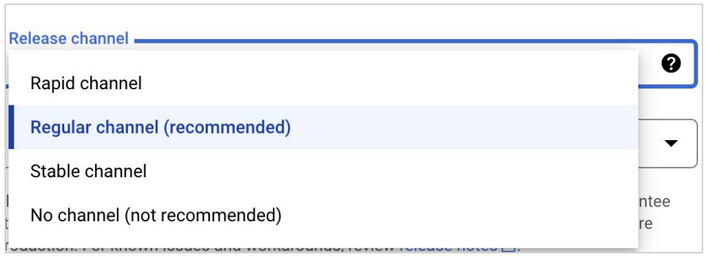
Automated updates. Choose a release cadence and feature set to match risk preference.



Always on **reliability**

Rapid → Regular → Stable

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

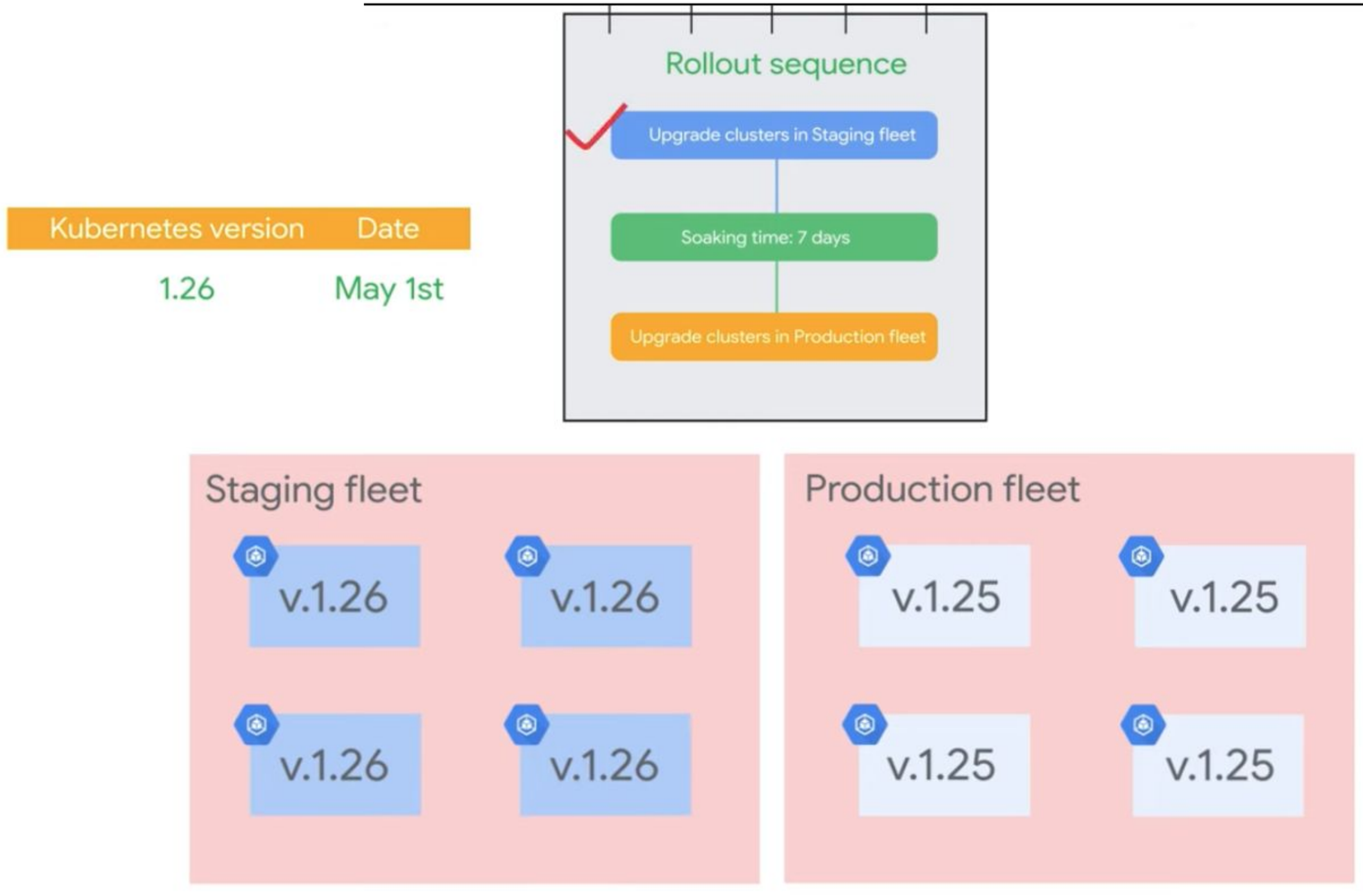


And you still have control: manual upgrades, maintenance windows, exclusions, and pod disruption budgets (PDBs) are still respected.

Rollout Sequencing

Manage the automated rollout sequence of new minor releases and patch versions among clusters

Always on **reliability**



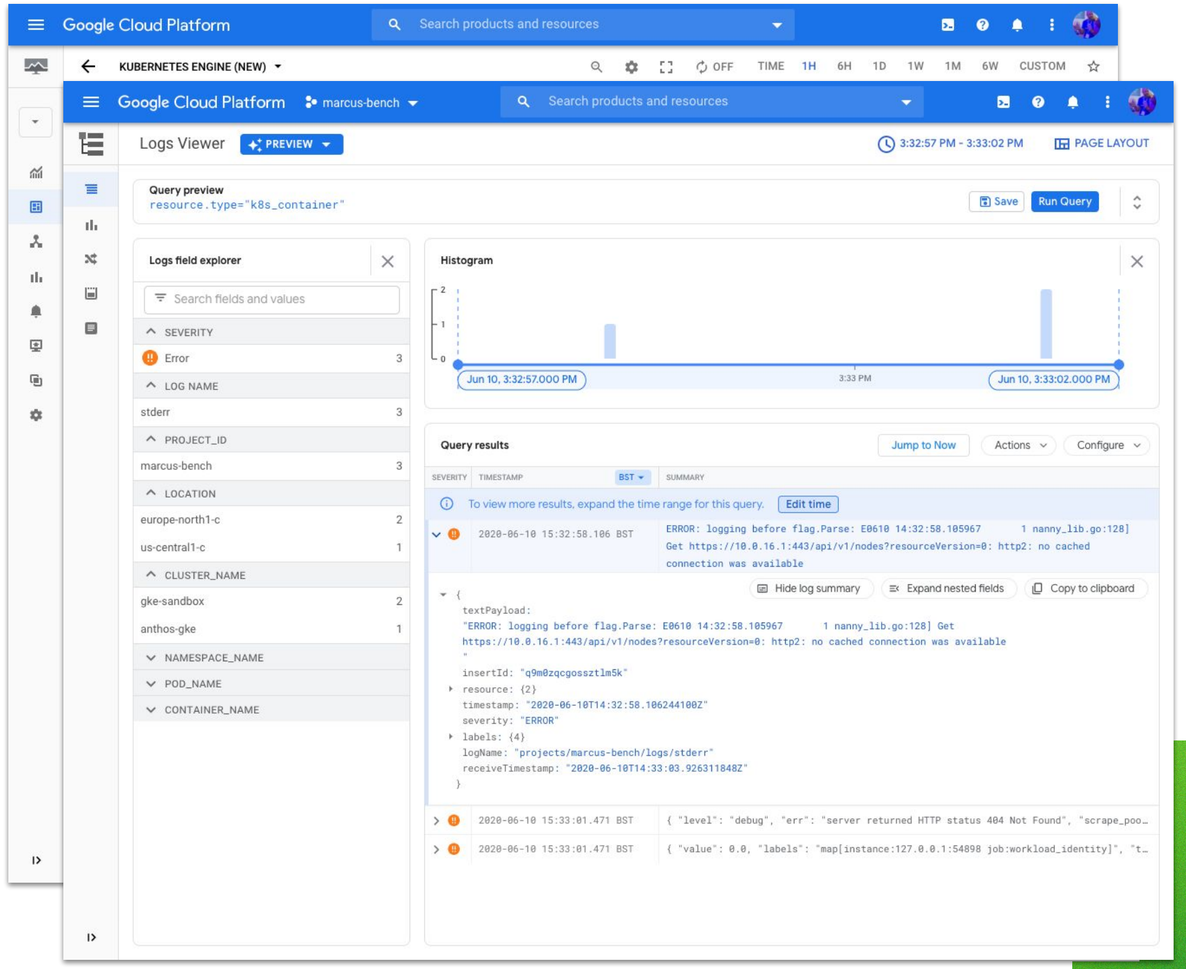
Integrated Logging and Monitoring

Cloud Logging and **Monitoring** are both enabled by default for system workloads

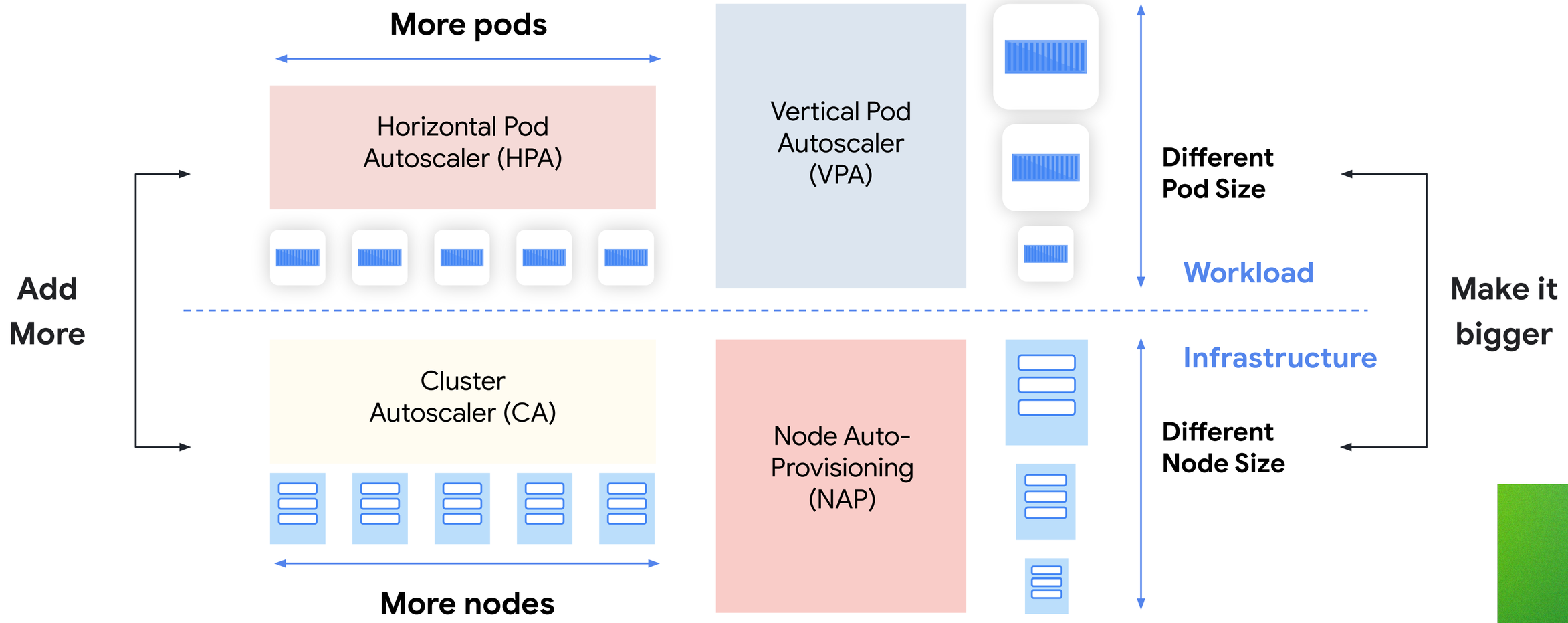
- Control Plane Logging and Monitoring.
- **Workload logs** are enabled, but optional.
- Prometheus-compatible **Workload metrics** can be scraped directly (legacy method) or via the **Managed Service for Prometheus.**

Managed Service for Prometheus can act as a drop-in replacement for self-managed Prometheus or just for feeding Cloud Monitoring with metrics

- Configured with PodMonitoring CRD.
- Billed per sample (high-usage tier available).



Autoscaling



DEMO

It just works.

Workload Optimized

Compute Classes

General-Purpose

Best price/ performance for x86

Great default choice for most compute

- Web serving / API
- Microservices
- Dev environments

Series: **E family** (Default)

Balanced

Consistent performance

Wide range of VM shapes (high Mem/ CPU)

Very flexible and stable

- Web serving / APIs
- Microservices
- Stateful Apps (DB / Cache
- Media/Streaming
- Back office Apps

Series: **N2/ N2D**

Scaled-out

Best price/performance for high throughput workloads

x86 / ARM

- Scaled-out
- Web serving / API
- Microservices
- Data log processing
- Media transcoding
- Large-scale Java applications

Series: **Tau**

Compute Classes

Accelerators

Accelerators

GPU/ TPU

GPU Sharing

- AI workloads
- Inference at large scale
- Small to medium Machine Learning
- Batch

Series: **T4 / A100 / H100 / L4**

More to come...

Performance

High performance machine families

Use all instance resources without restriction

- Performance critical applications
- HPC
- Databases

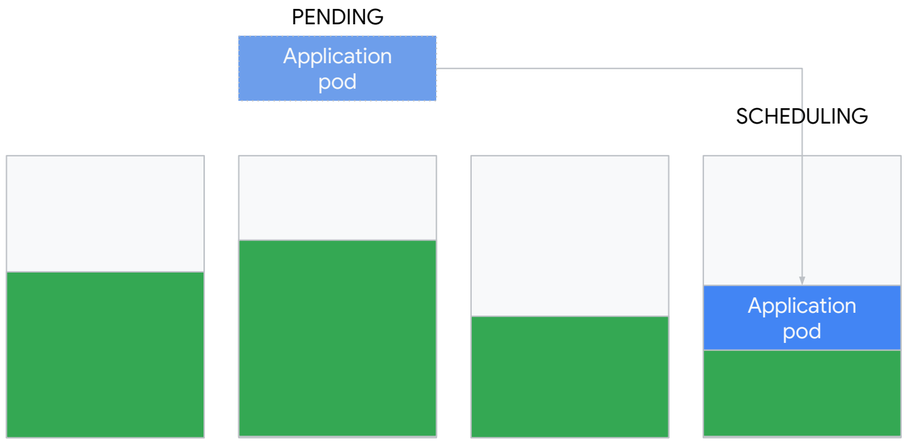
Series: **C3 / H3 /**

More to come

CA Profiles - Different scheduling

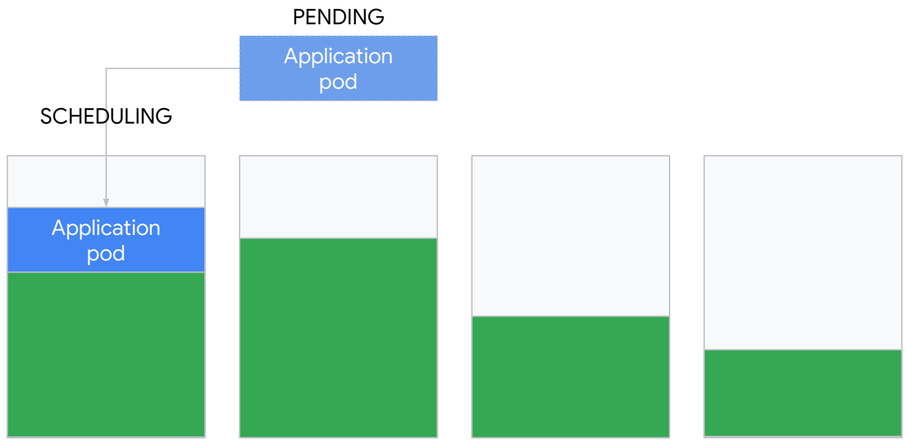
Balanced Profile

TL;DR: Prefer less used nodes
To keep all nodes, well... *balanced*



Optimize Utilization Profile

TL;DR: Prefer most used nodes
To leave nodes with less requested resources that can be scaled down quickly



For more information, see Autoscaling profiles.
Disclaimer: Google may fine tune profiles to improve either reliability or the desire profile proposal

Secure By Design

Delivering Value Through Security

GKE Security Posture Management

Compliance & Governance

Configuration validation against common settings, benchmarks and standards.

Vulnerability Management

Workload vulnerability analysis and assessment of application and OS layers.

Threat Detection

Behavioral evaluation and analysis of workload and cluster activities.

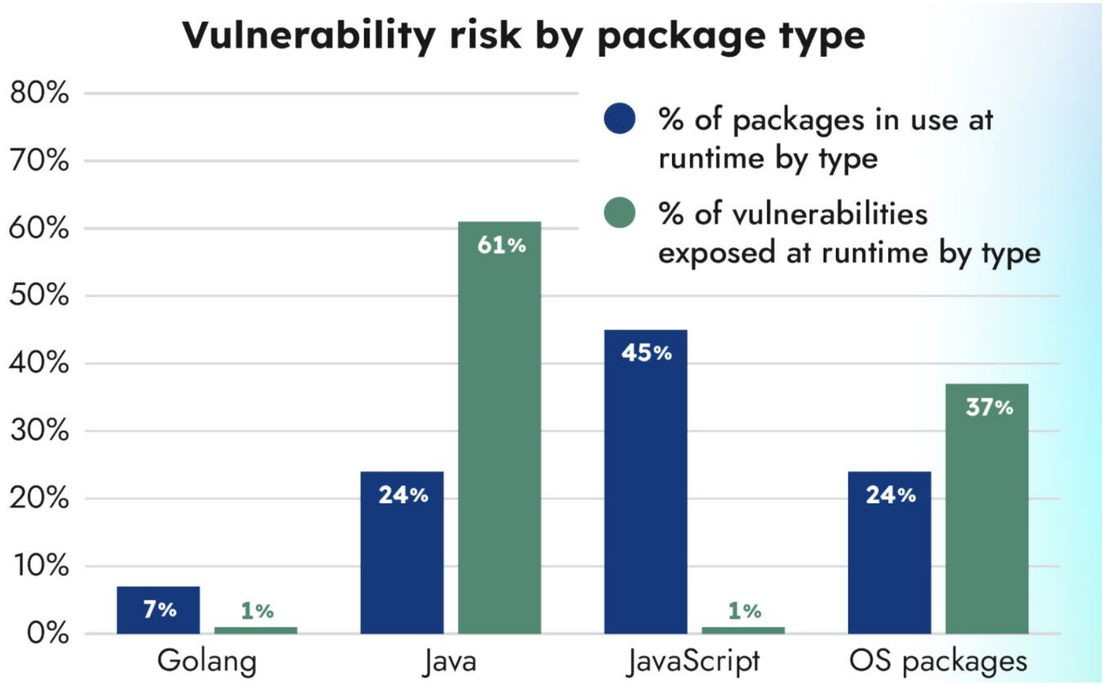
GKE Platform Security

Identity & Access

Hardening & Compliance

Patching & Supply Chain

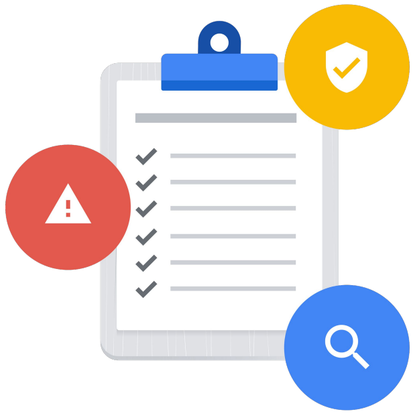
GKE Vulnerability Management



- ~15s container extraction and validation, at scale.
- Support for OS vulnerabilities.
- Support for Golang, Java, Javascript and Python languages.
- Using Common Vulnerability Scoring System (CVSS 3.0).
- Consolidated CVE feeds from eight different sources (NIST, NVD 1.1, Alpine, Debian, Ubuntu, RHEL, COS, Github, and OSV).
- Only actionable results are surfaced to reduce noise and eliminate no-op cycles.

GKE Threat Detection

Managed service to detect threats by analyzing log data. Detections powered by Google's proprietary threat intelligence.



Always monitoring

Continuously monitors your organization or projects and identifies threats within your systems in near-real time to monitor your Kubernetes clusters for threats.

Threat Intelligence

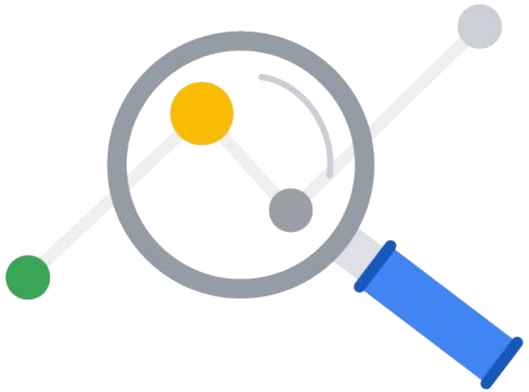
Applies detection logic to identify threats in near-real time. Threat detection by Kubernetes audit log analysis. No additional systems to manage.

Integrated

Powered by Security Command Center Event Threat Detection with a unified user experience.

GKE Threat Detection

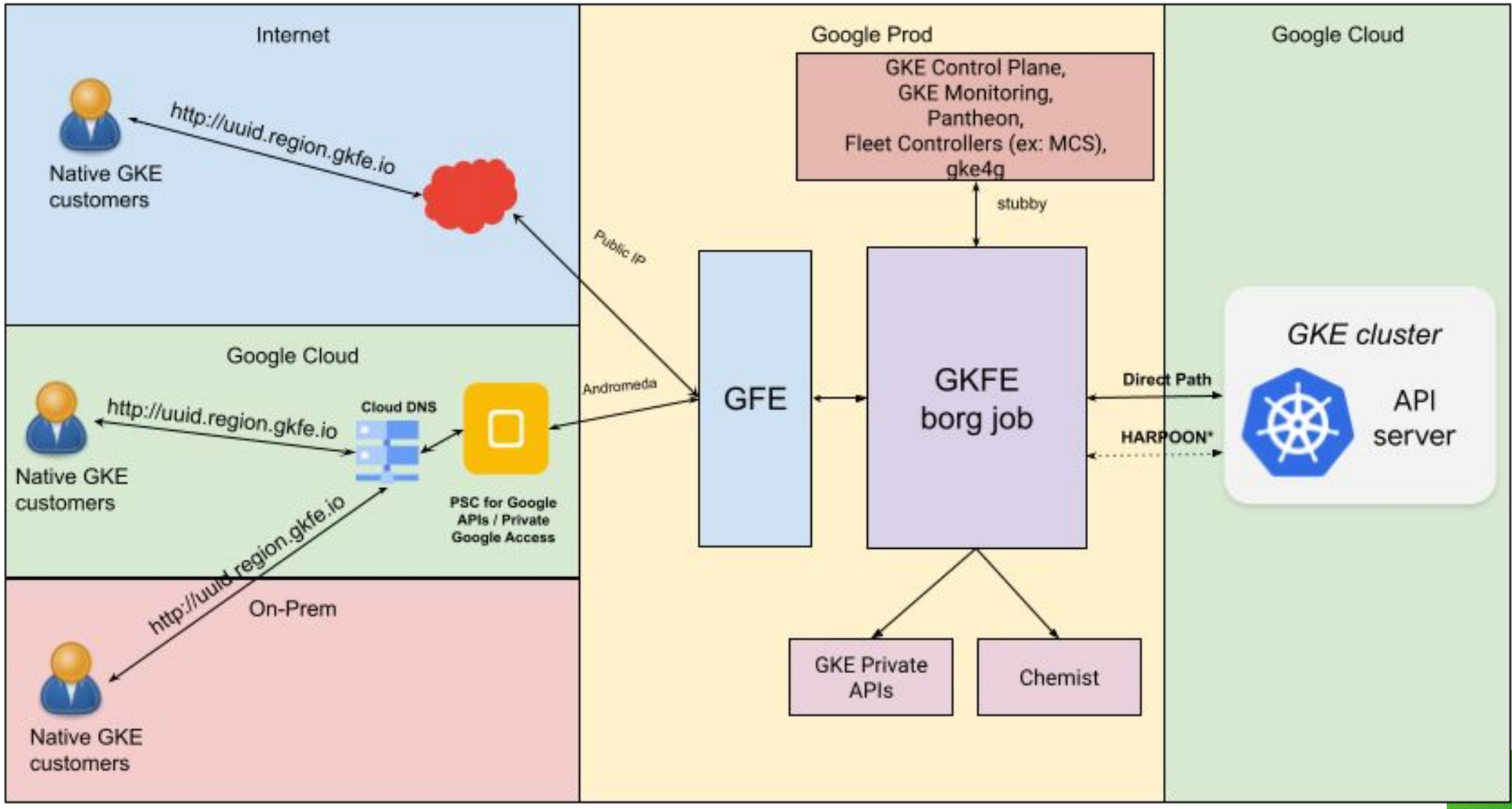
Detections where a malicious actor **attempts to query for or escalate privileges** in GKE.



- Discovery: Can get sensitive Kubernetes object check
- Privilege Escalation: Changes to sensitive Kubernetes RBAC objects
- Privilege Escalation: Create Kubernetes CSR for master cert
- Privilege Escalation: Creation of sensitive Kubernetes bindings
- Privilege Escalation: Get Kubernetes CSR with compromised bootstrap credentials
- Privilege Escalation: Launch of privileged Kubernetes container
- Defense Evasion: Breakglass Workload Deployment Created
- Defense Evasion: Breakglass Workload Deployment Updated
- Credential Access: Secrets Accessed in Kubernetes Namespace
- Initial Access: Anonymous GKE Resource Created from the Internet
- Initial Access: GKE Resource Modified Anonymously from the Internet

Sneek Peak

Control Plane Access



Custom 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
```

We are interested in your feedback!

Connect with a GKE/Serverless PM or UX researcher.

kubernetes turns 10!

#k8sturns10

Thank you