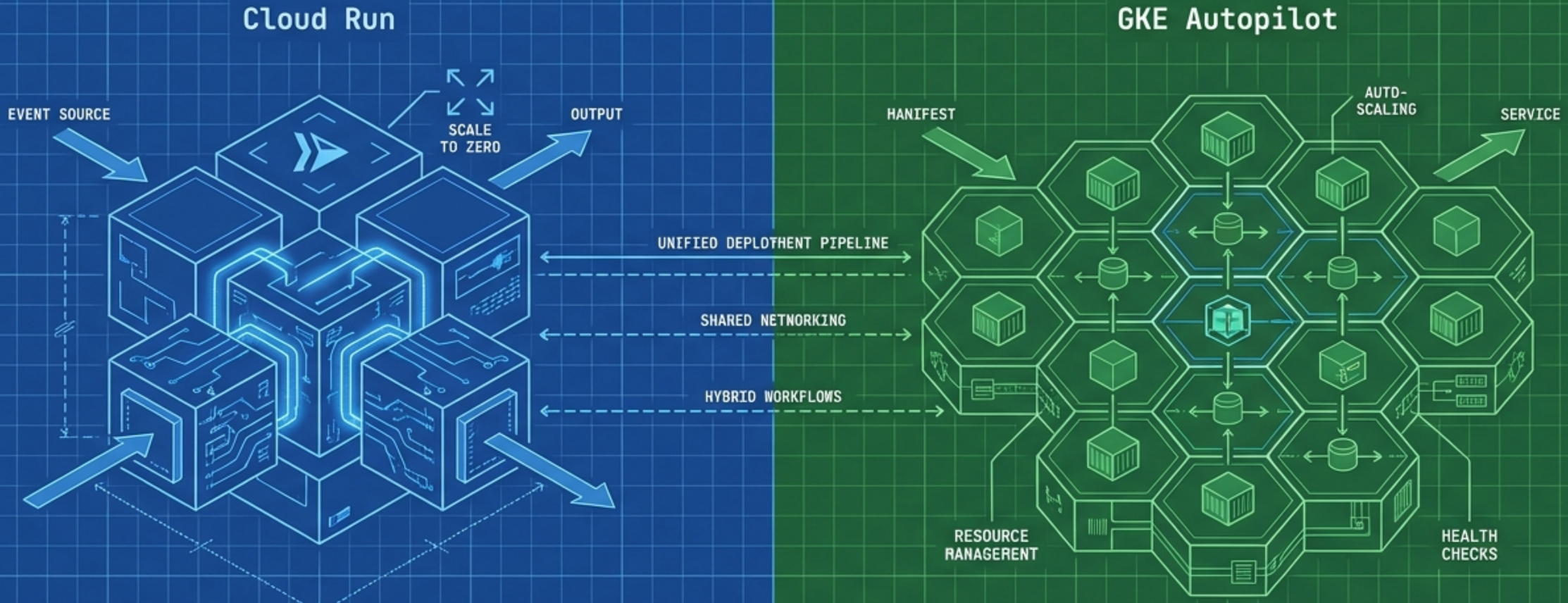
Mastering PCD Section 3: Deploying Applications

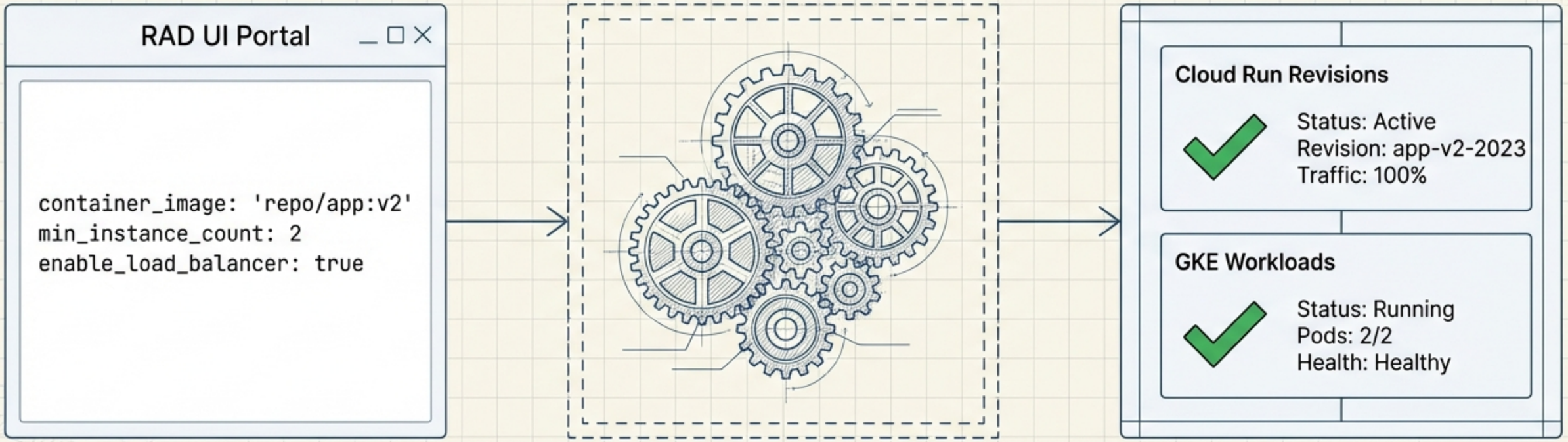
A visual guide to Cloud Run, GKE Autopilot, and the Tech Equity RAD platform.



Architecture, Scaling, and Diagnostic Frameworks for the Exam.

Declarative Inputs to Observable Infrastructure

The Tech Equity RAD platform translates simplified deployment variables into full-scale GCP infrastructure. Understanding this mapping is critical for anticipating exam scenarios.

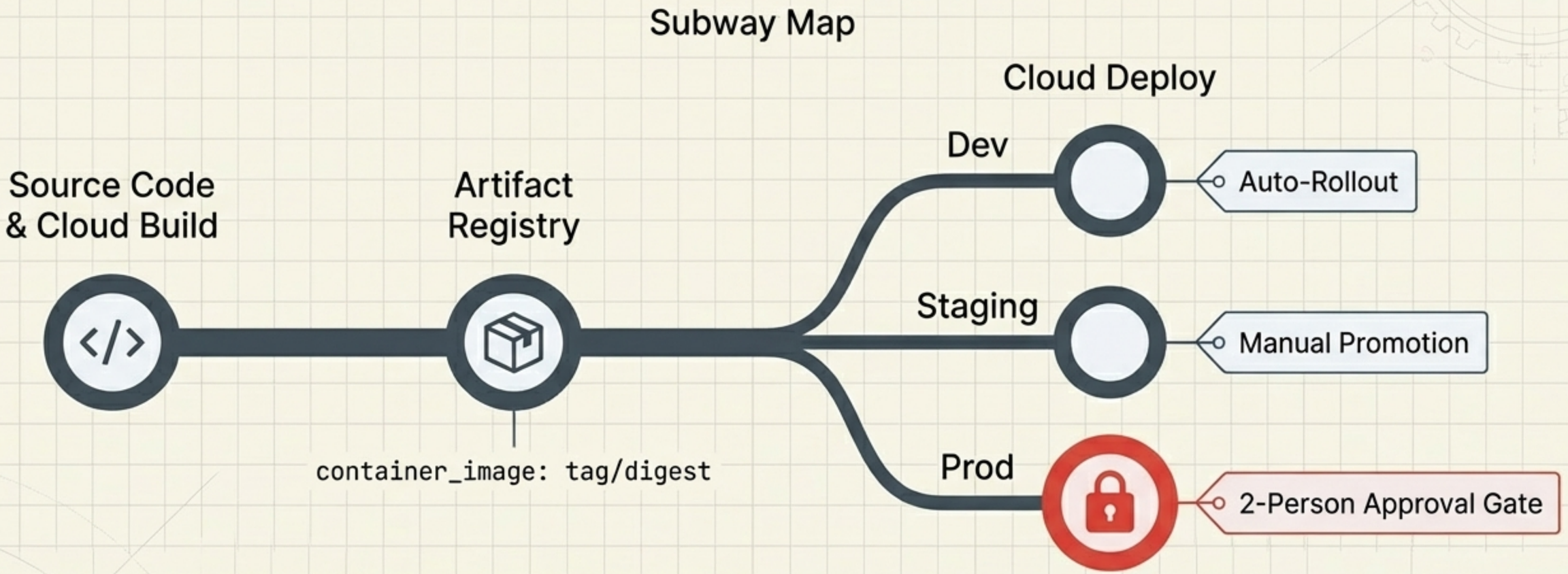


Input

Deployment Orchestration

Output

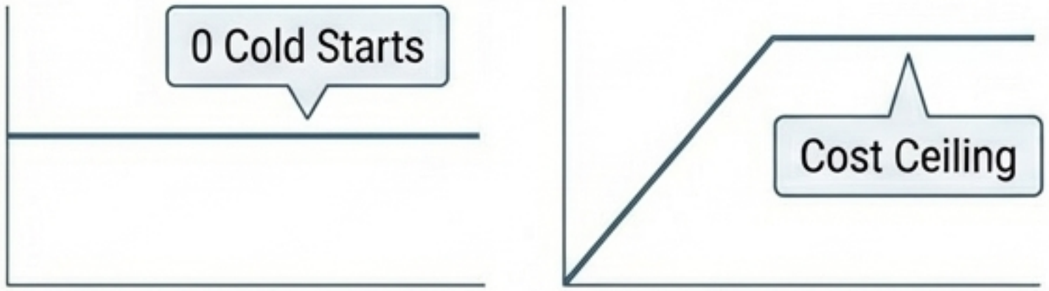
The Cloud Run Deployment Pipeline



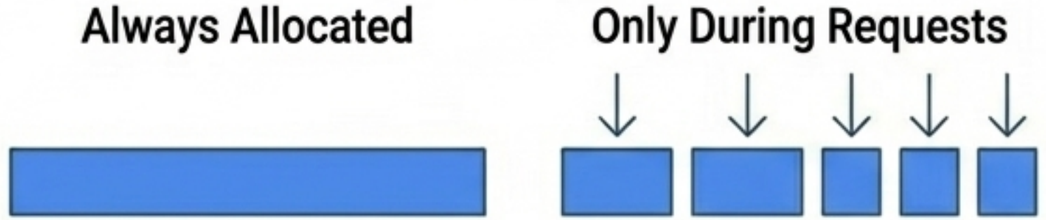
⚠ Cloud Run traffic splitting routes traffic to multiple immutable revisions simultaneously—enabling canary and blue/green patterns independent of Cloud Deploy.

Tuning Cloud Run: Concurrency, Cost, and Cold Starts

Scaling Dimensions



CPU Allocation



Billed per second.
Background tasks allowed.

Billed per request.
Scales to zero.

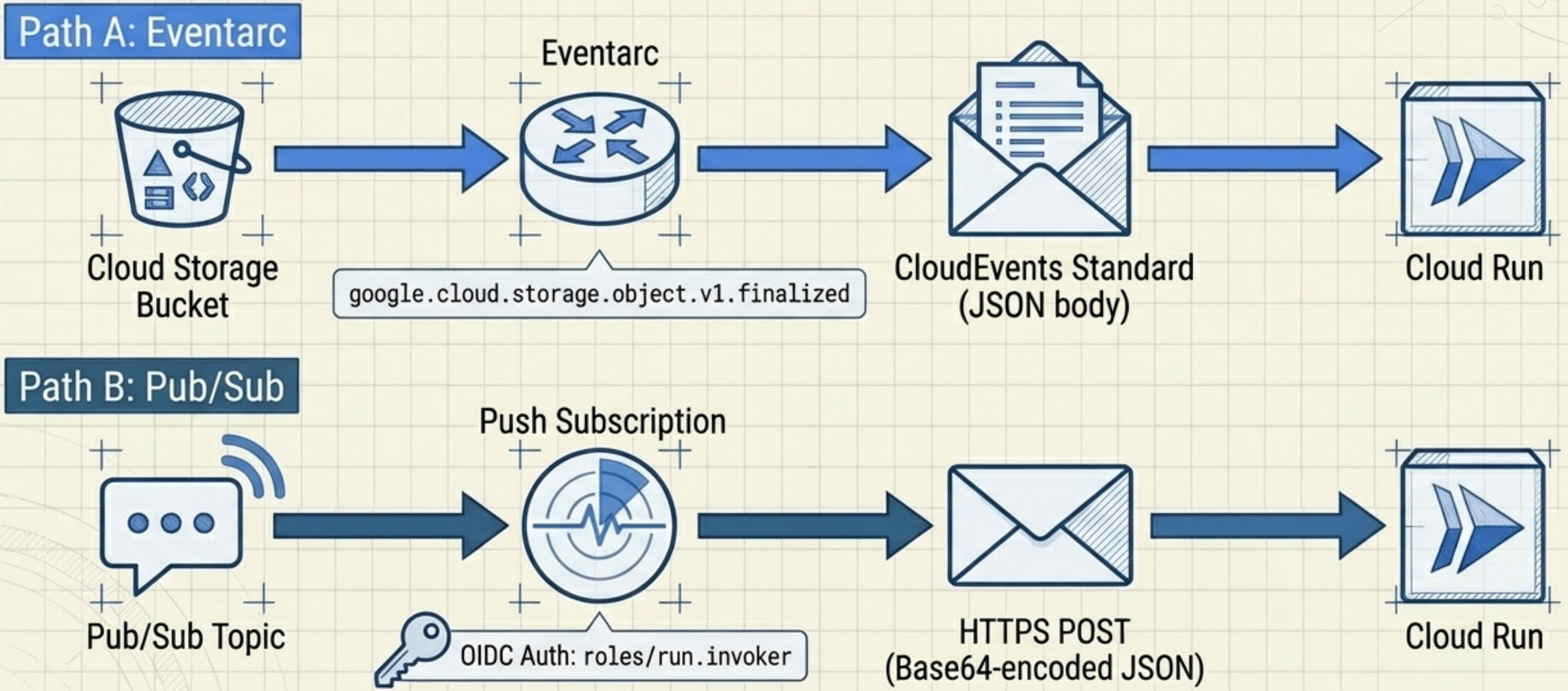
Ingress Control

Workload Profile Match

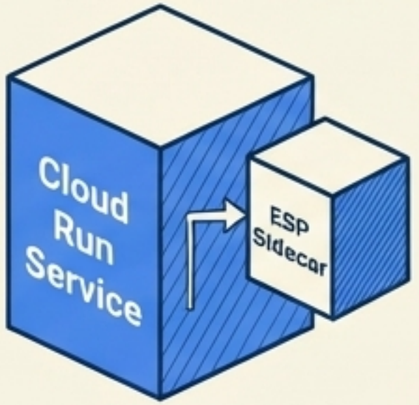
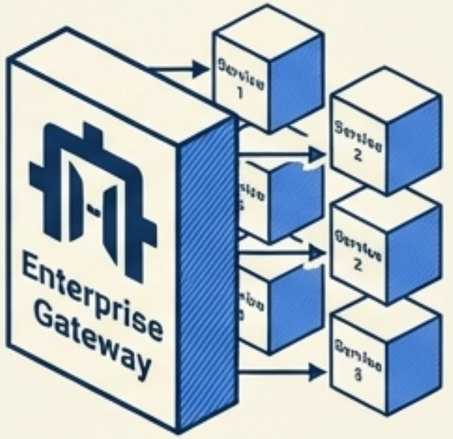
- **Always Allocated**: Ideal for latency-sensitive APIs or background processing.
- **Request-Only**: Optimal for unpredictable, bursty workloads.

Event-Driven Invocations: Eventarc vs. Pub/Sub

```
Python Parsing:
  Eventarc: from_http(request) via cloudevents lib
  Pub/Sub: Decode Base64 message body
```



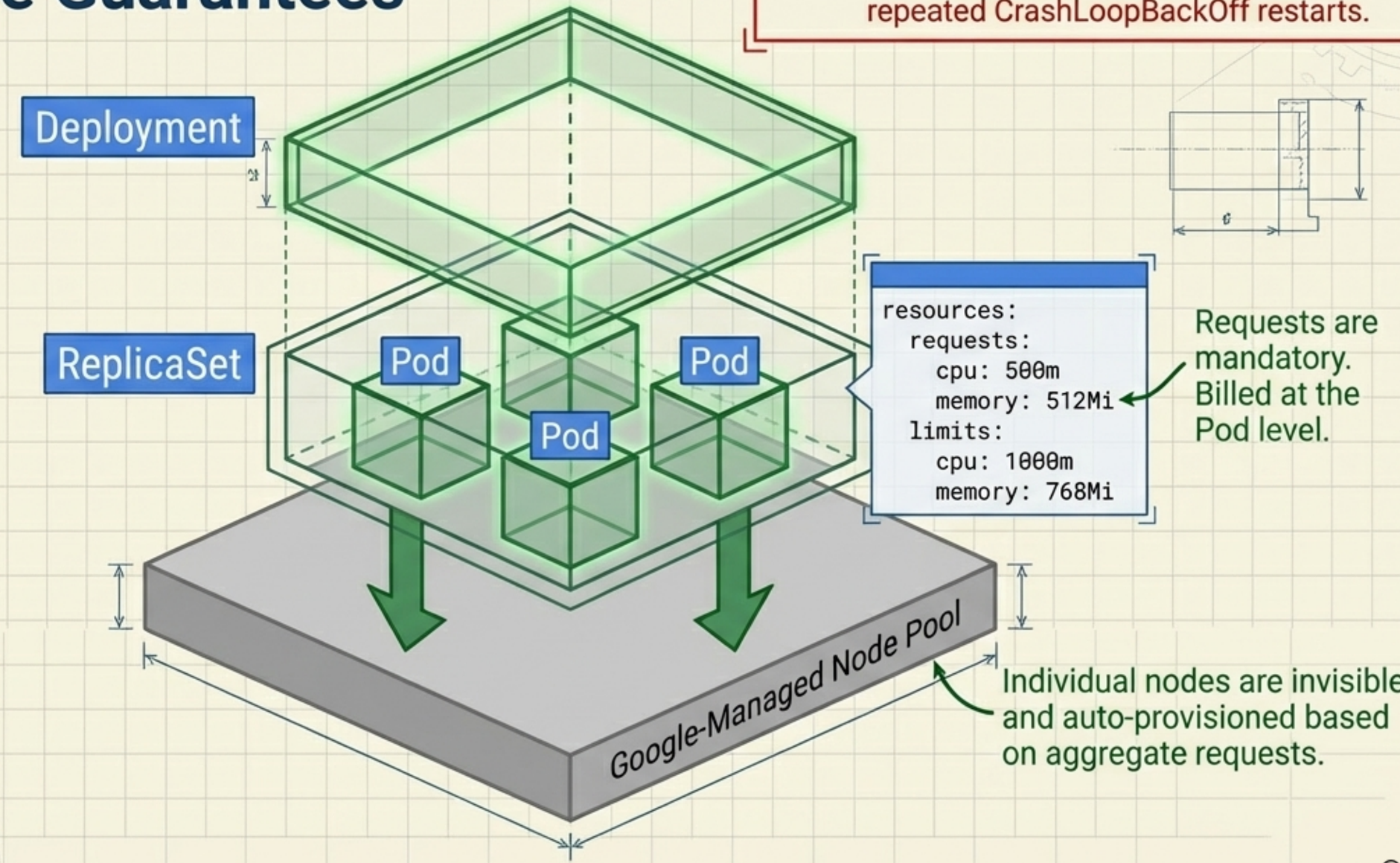
API Management: Endpoints vs. Apigee

Cloud Endpoints	Apigee
	
✔ OpenAPI 2.0 specs ✔ x-google-quota extensions ✔ Simple rate limiting ✔ Service-to-service auth	✔ Developer portals ✔ API monetization ✔ Advanced caching & OAuth 2.0 ✔ Threat protection policies
Ideal for lightweight microservices	Ideal for APIs as a Product

⚠ The API Evolution Principle: Never remove or rename existing fields—only add new optional fields. For breaking changes, deploy a versioned path (/v2/) and deprecate the old.

GKE Autopilot Deployments & Resource Guarantees

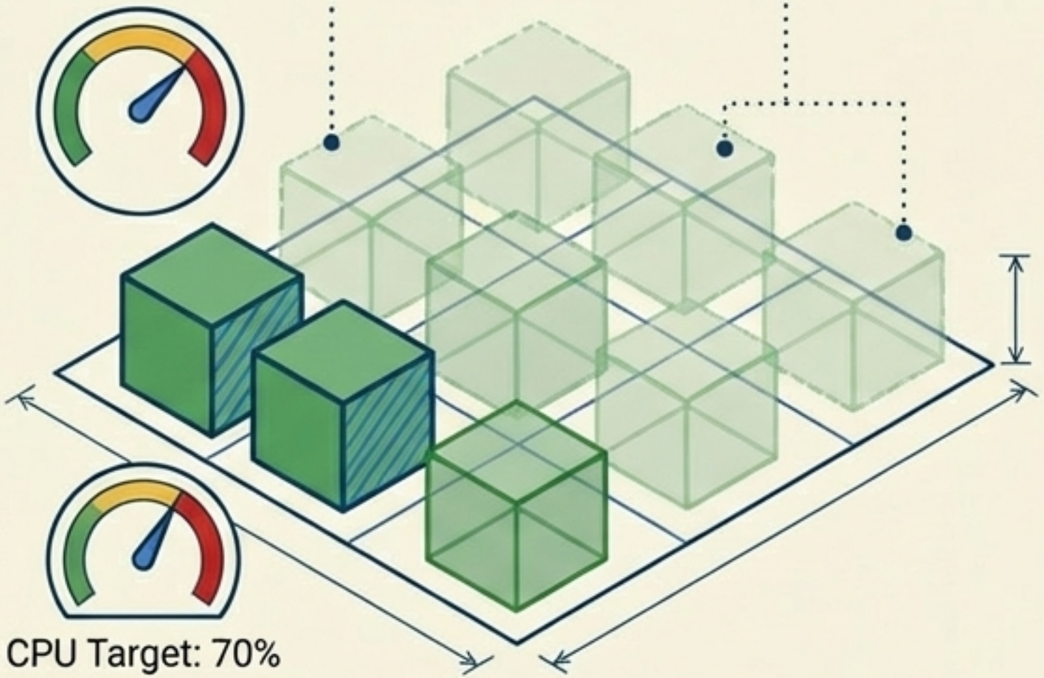
Diagnostic Warning: Under-provisioning memory (e.g., 256Mi for a heavy JVM) triggers the Linux OOM Killer, resulting in repeated CrashLoopBackOff restarts.



The Dual Dimensions of GKE Autoscaling

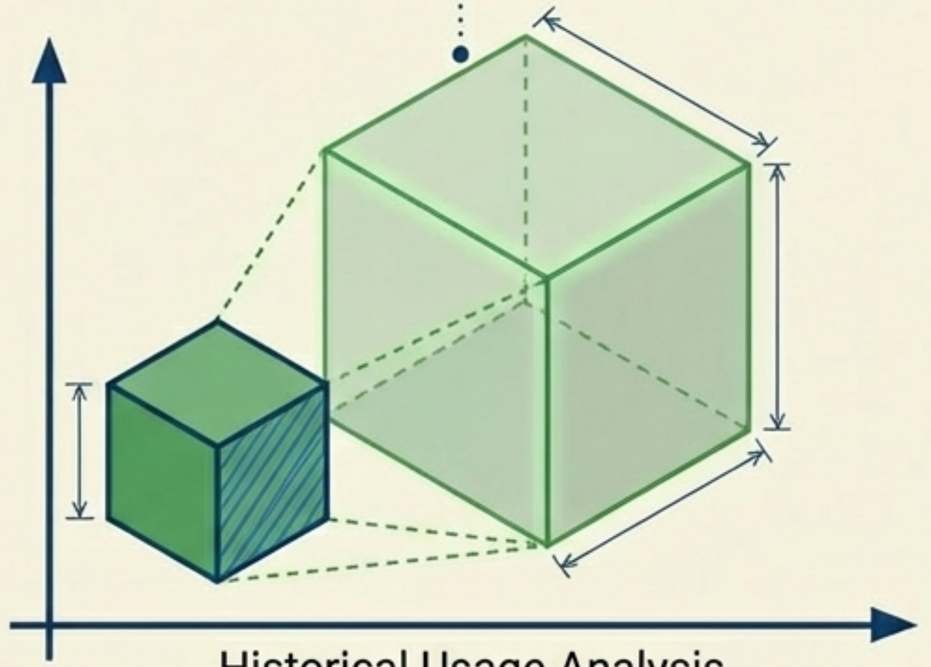
Horizontal Pod Autoscaler (HPA)

`min_instance_count` `max_instance_count`



Vertical Pod Autoscaler (VPA)

`enable_vertical_pod_autoscaling`



⚠ VPA right-sizes what each pod gets based on history.
HPA scales out how many pods run based on real-time load.

Pod Diagnostics: The 3 Health Probes

Pod Initialization

Traffic Routing

Steady State

Startup Probe

Delays liveness and readiness checks for slow-starting apps (e.g., heavy JVMs). Prevents Prevents premature termination during warm-up.

Readiness Probe

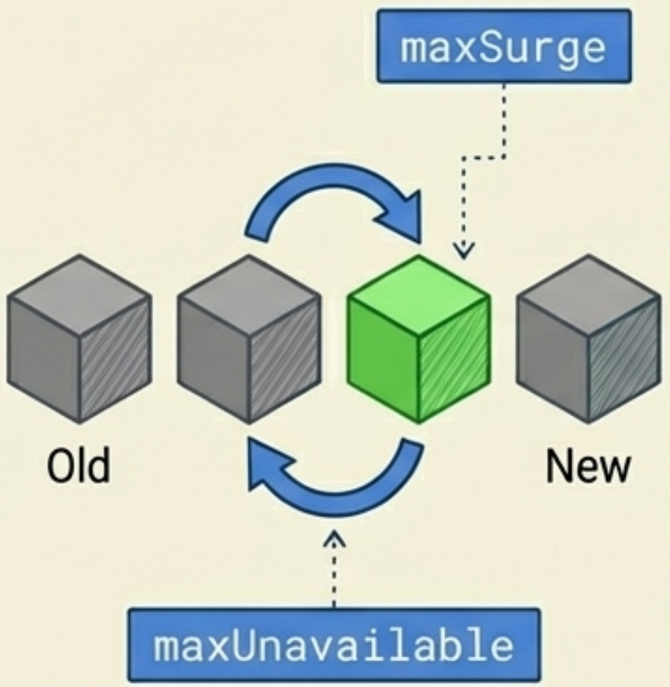
If failed: Pod is removed from the Service Load Balancer endpoint list. No restarts occur, but traffic ceases until ready.

Liveness Probe

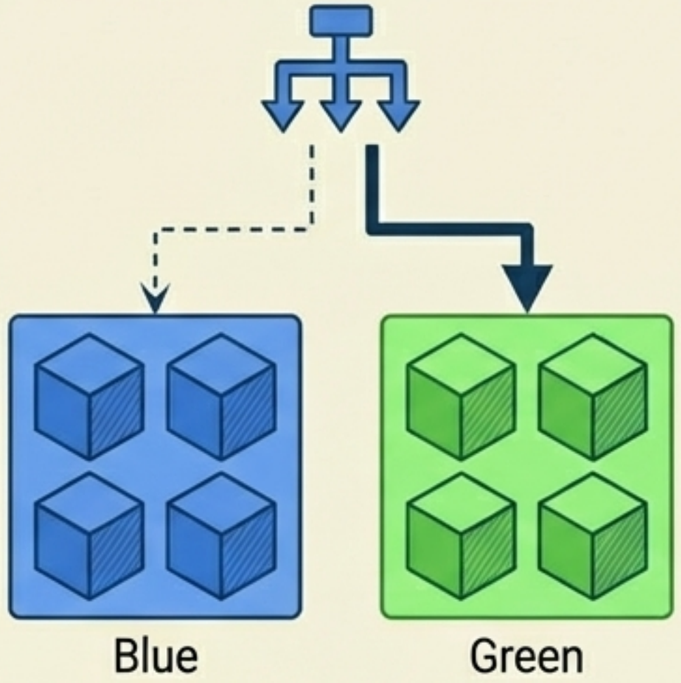
If failed: Kubernetes actively kills and restarts the container to resolve deadlocks or unrecoverable unrecoverable states.

Zero-Downtime Rollout Strategies

Rolling Update (Default)

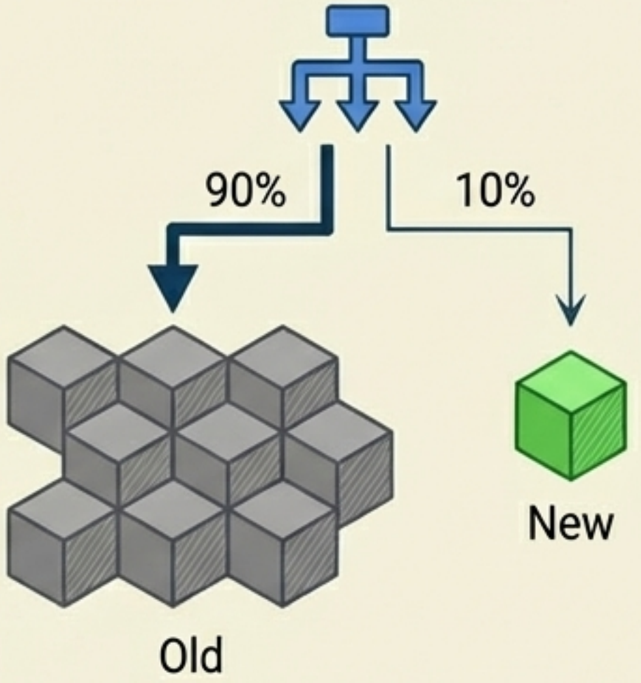


Blue/Green



Instant switch via Service Selector.
Instant Rollback.

Canary



Gradual traffic shift
by replica ratio.

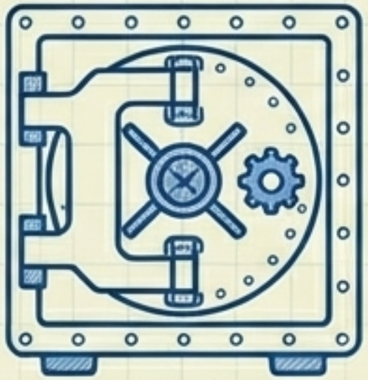
Exam Tip: Native Canary relies on replica counts (1 new out of 10 = 10%). For exact percentage-based traffic splitting independent of replicas, use Cloud Service Mesh.

Configuration, Secrets, and Autopilot Constraints

Constraint Rule: Workloads requiring DaemonSets with host-level access, GPU node pools, or custom kernels must use GKE Standard.

Configuration & Secrets

ConfigMaps
(Non-sensitive)



Secret Manager Integration

Native Kubernetes Secrets are Base64 only! Requires Secrets Store CSI Driver or enable_auto_password_rotation.

Autopilot Admission Constraints

- ✕ **hostPath volumes**
- ✕ **hostNetwork**
- ✕ **Privileged Security Contexts**

Autopilot automatically rejects non-compliant pod specifications at the admission controller level.

The Blueprint Synthesis: Cloud Run vs. GKE Autopilot

	Cloud Run	GKE Autopilot
Compute Model	Serverless abstraction.	Orchestrated containers on managed nodes.
Scaling Mechanism	Concurrency limits & instance counts.	ReplicaSets driven by VPA & HPA.
Pricing Model	Billed per second or per request.	Billed by aggregate Pod resource requests.
Health Checking	GCP Load Balancer active health checks.	Native Kubernetes Liveness & Readiness Probes.
Ideal Use Case	Event-driven, stateless web APIs & webhook handlers.	Complex microservices requiring background processing, strict resource guarantees, & orchestration.