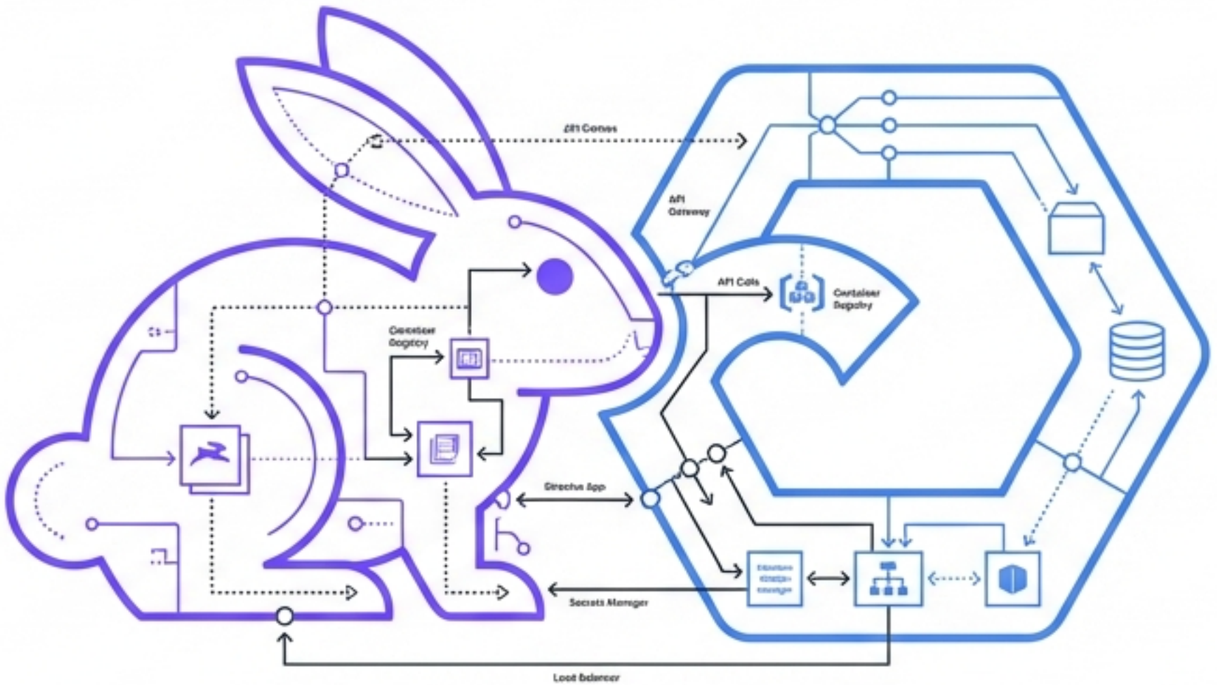
Directus on GCP

The Lifecycle Anatomy of a Cloud-Native Deployment

A technical deep dive into the Wrapper Module pattern, Least Privilege security, and self-healing container logic provided by the RAD Platform.



Architecture: Serverless & Stateless

Security: Zero Plain Text Credentials

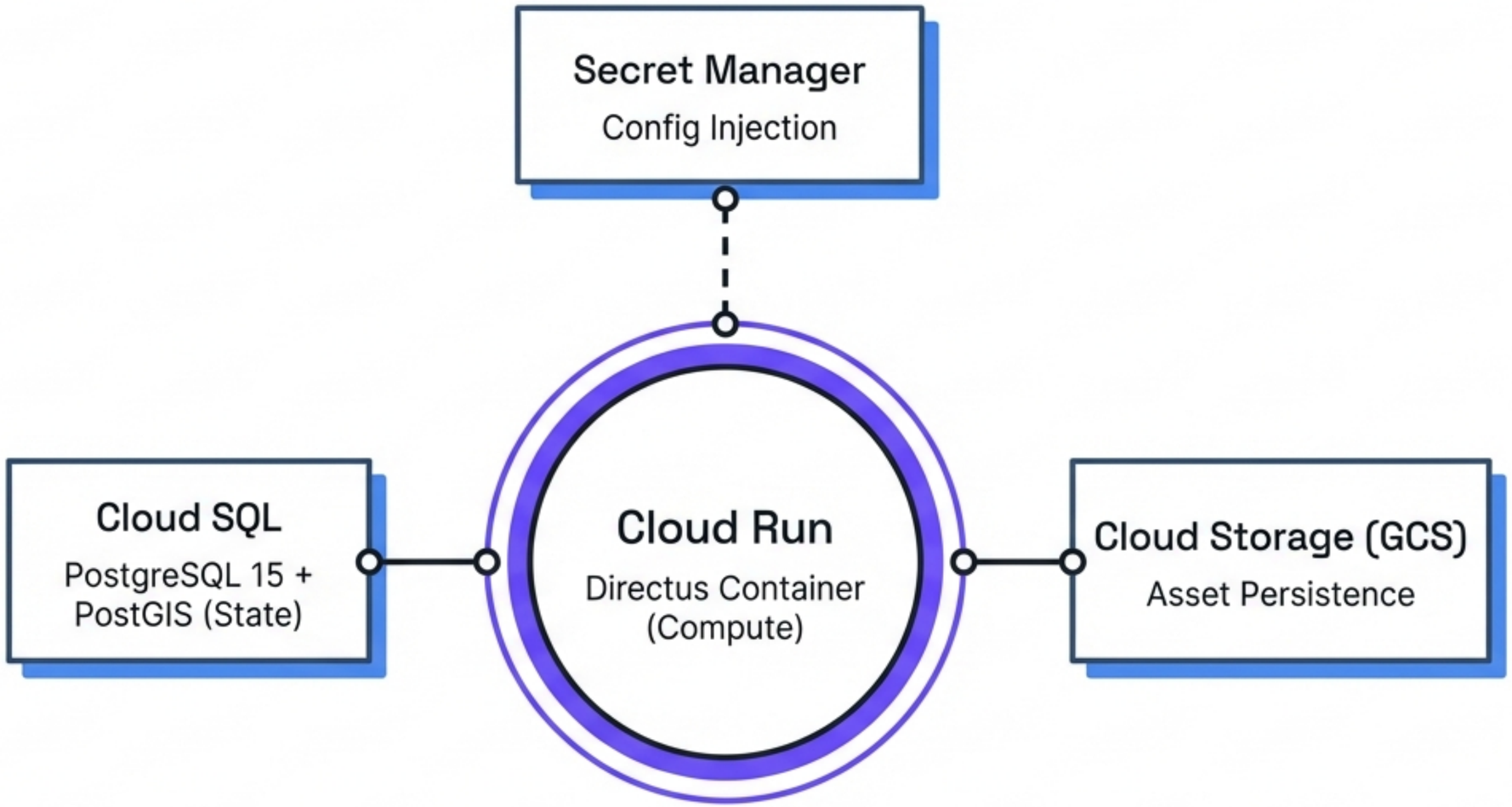
Operations: Self-Healing Startup Sequence

The Wrapper Module Architecture

The Blueprint Strategy

Compute Strategy:
Leverages Serverless Containers for auto-scaling and zero maintenance.

Separation of Concerns:
The core CloudRunApp module provisions resources, while the wrapper injects Directus-specific logic.



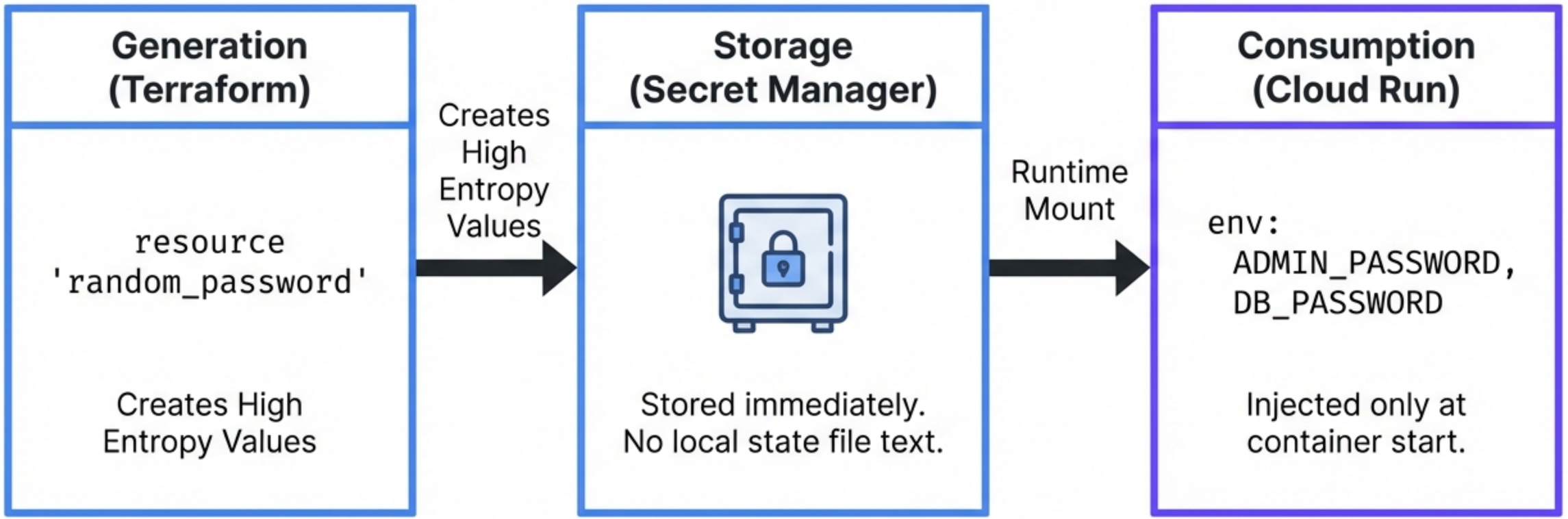
Engineering Editorial

Locking Down the Perimeter (IAM)

Enforcing strict Least Privilege by mapping roles directly to operational needs.

	Infrastructure	Action	Justification
	`roles/cloudsql.client`	Connects via Auth Proxy	Removes need for public IP access.
	`roles/secretmanager.secretAccessor`	Mounts sensitive Env Vars	Runtime injection of KEY, SECRET, DB_PASSWORD.
	`roles/storage.objectAdmin`	Reads/Writes to `/directus/uploads`	Enables GCS Volume mount for user assets.
	`roles/logging.logWriter`	Writes to Cloud Logging	Standard observability and audit trails.

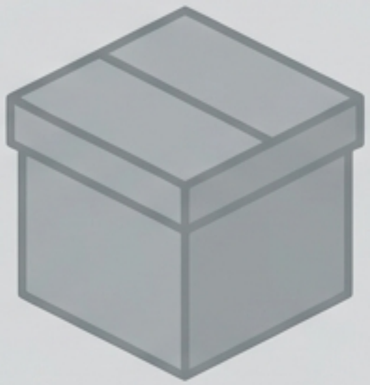
Zero Plain Text: The Secrets Lifecycle



Result: Humans never see or manage the raw credentials.

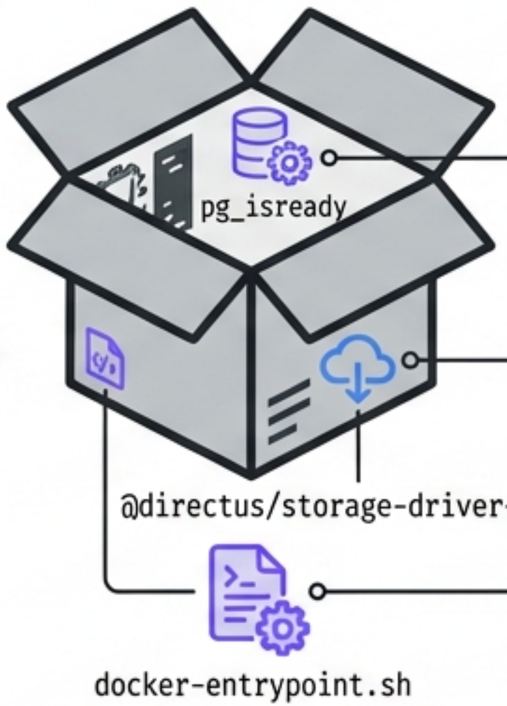
Beyond Vanilla: The Custom Build Strategy

Base Image



`directus/directus:11.1.0`

Custom Implementation



Critical Additions:

1. Resilience Tooling

`Install: postgresql-client`

Why: Required for pg_isready health checks.

2. Storage Drivers

`Install: @directus/storage-driver-gcs`

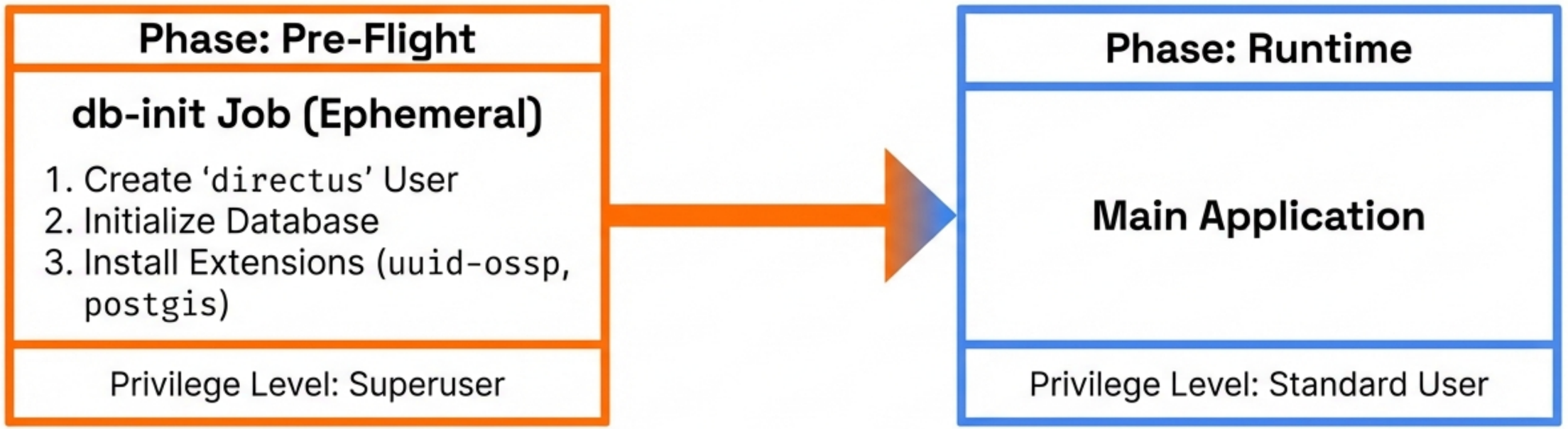
Why: Enables native GCS support options.

3. Logic Injection

`Copy: docker-entrypoint.sh`

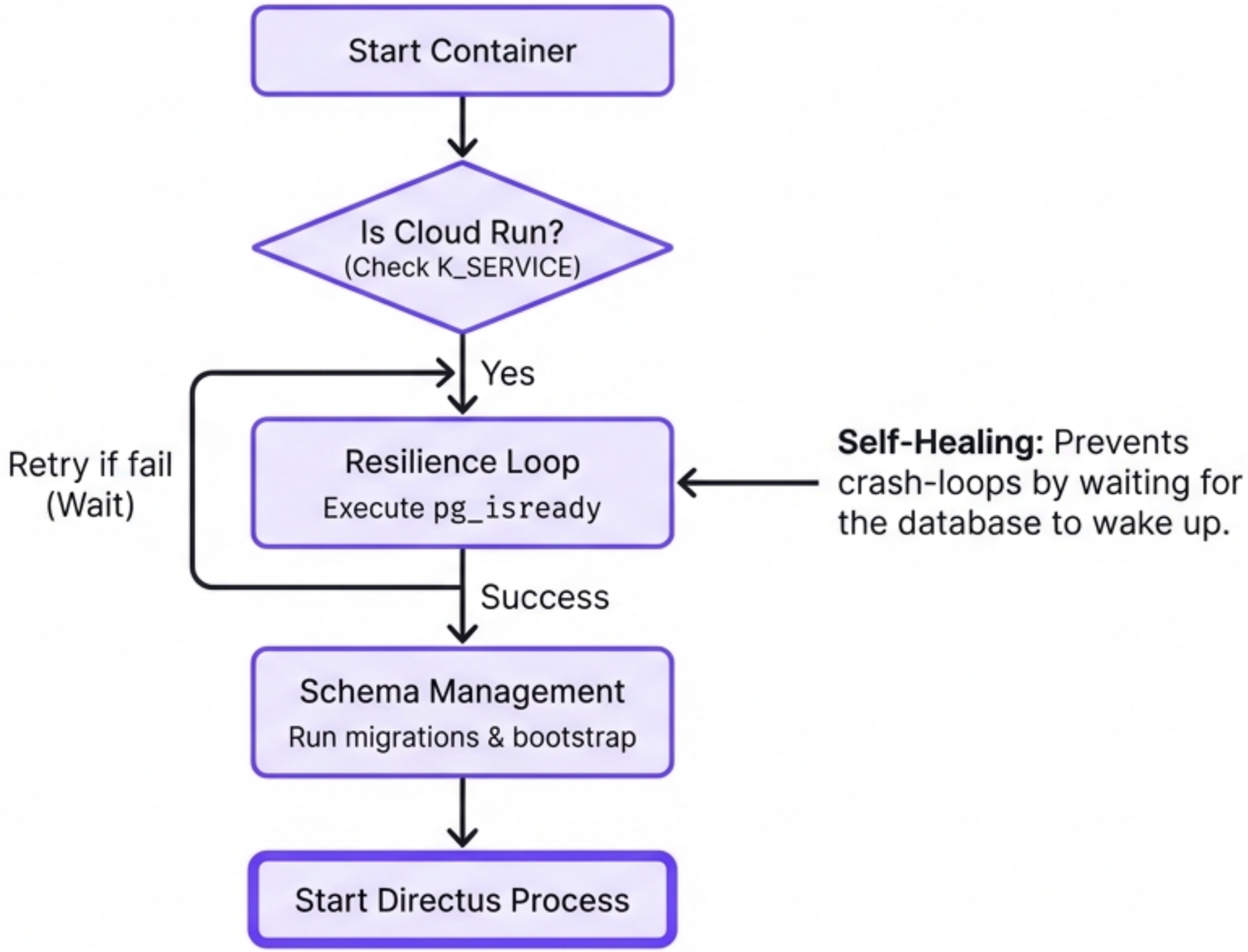
Why: Controls the self-healing startup sequence.

Pre-Flight Initialization: The db-init Job



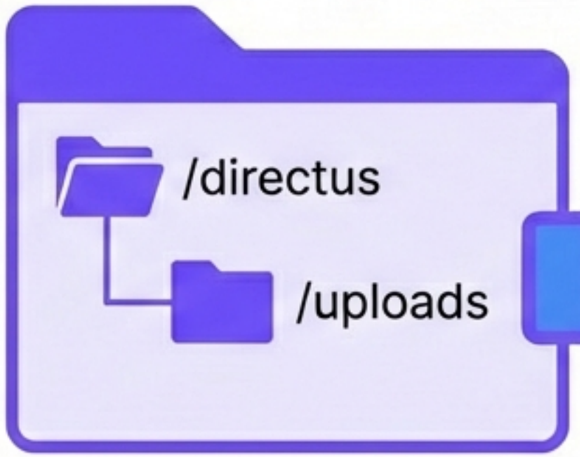
Objective: Provision structure securely without granting the main application permanent superuser rights.

The Startup Sequence: `docker-entrypoint.sh`



Storage Integration: **The GCS Fuse Abstraction**

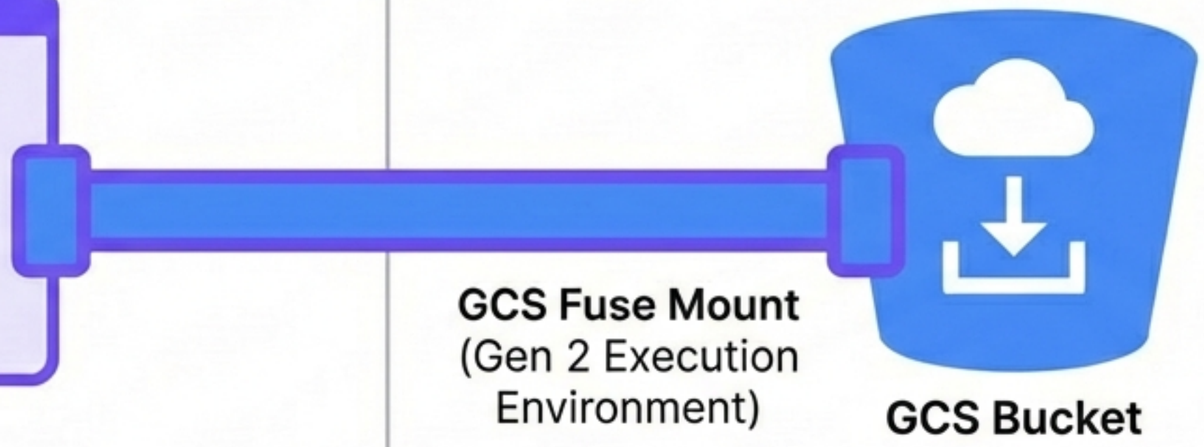
The App View




```
STORAGE_LOCATIONS: local
STORAGE_LOCAL_ROOT: /directus/uploads
```

Directus thinks it is writing to a local disk.

The Infrastructure Reality

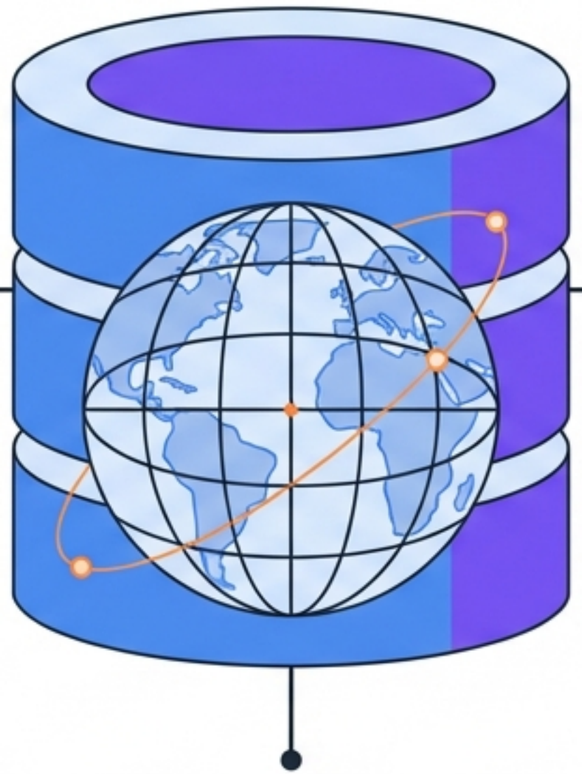


GCS Fuse Mount (Gen 2 Execution Environment)

GCS Bucket

Data is physically persisted in Google Cloud Storage.

Database Capabilities: PostgreSQL 15 & PostGIS



PostGIS Enabled

Extension pre-loaded by `db-init` job.

Connectivity

Uses standard ‘`pg`’ driver defined by `DB_CLIENT`.

Application Impact

Native handling of map coordinates and location data types.

Runtime Configuration Map

Connectivity Group

`PORT: 8055`
(Internal mapping)

`CORS_ENABLED: true`
(Headless usage)

`WEBSOCKETS_ENABLED: true`
(Real-time)

Storage Group

`STORAGE_LOCATIONS: local`

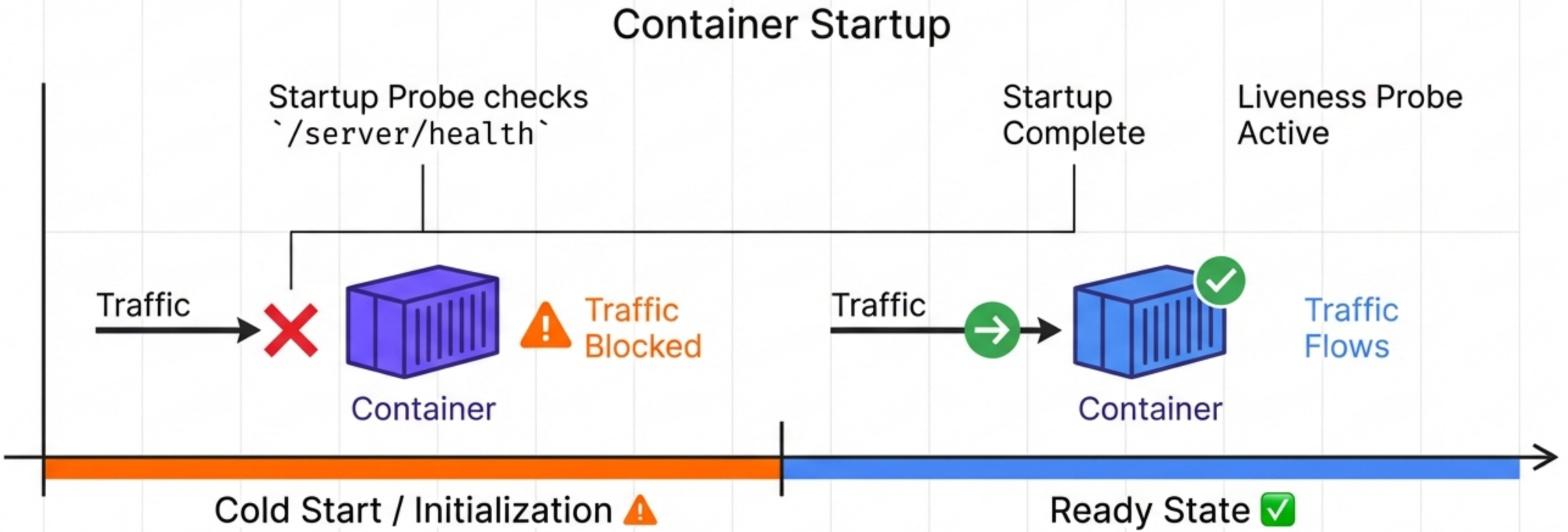
`STORAGE_LOCAL_ROOT: /directus/uploads`

Performance Group

`CACHE_ENABLED: false`
(Default)

`RATE_LIMITER_ENABLED: false` (Default)

Operational Resilience & Zero Downtime in Space Grotesk, Deep Charcoal (#1A1A1A)

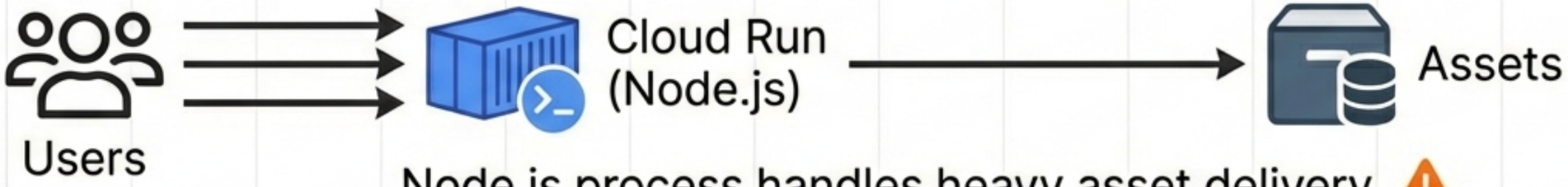


Mechanism: Cloud Run Probes prevent traffic from hitting cold containers, eliminating 502 Bad Gateway errors during deployments.

Evolution: Offloading to Cloud CDN

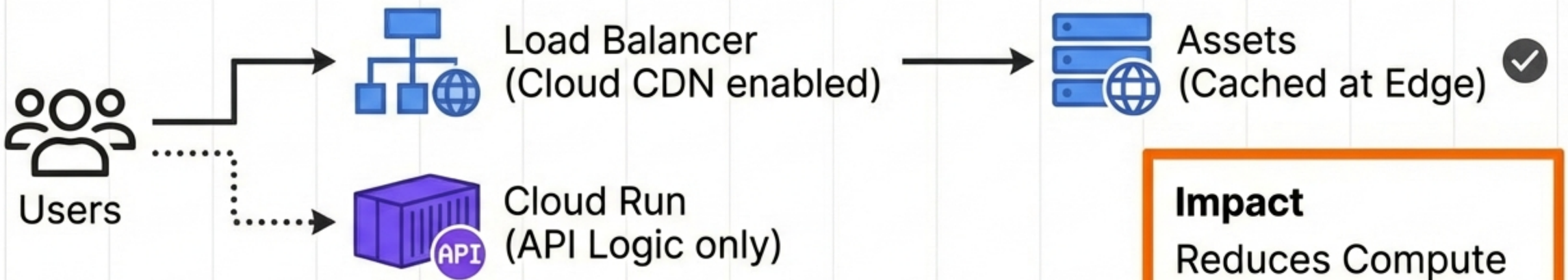
Current vs Future

Current State



Node.js process handles heavy asset delivery.

Future Optimization



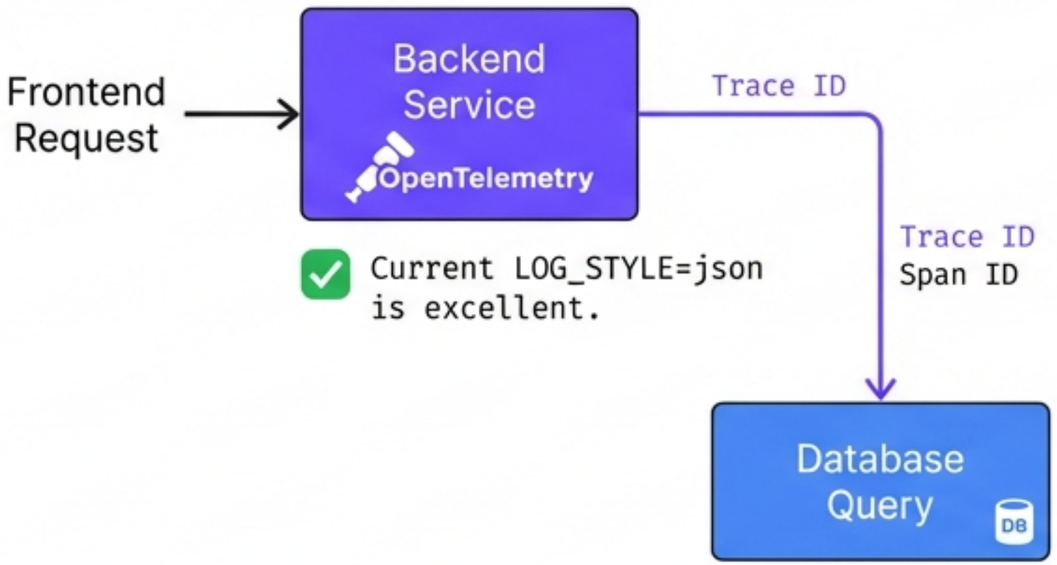
Assets served from edge. Node.js focuses on API.

Impact

Reduces Compute Costs & Latency.

Enhancing Observability & Security

Structured Logging & Tracing



⚠️ **Upgrade:** Implement OpenTelemetry.

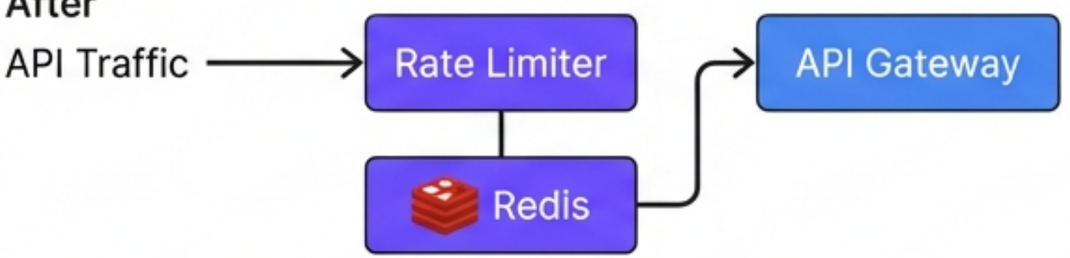
Benefit: Trace frontend requests through to backend DB queries.

Rate Limiting

Before

⚠️ **Status:** Currently Disabled.

After



Upgrade: Enable Rate Limiter backed by Redis.

Benefit: Protect API from brute-force & runaway scripts.

Engineering Editorial

Production Readiness Checklist

- **Security:** Least Privilege IAM & Encrypted Secrets.
- **Resilience:** `wait_for_db` logic prevents crash-loops.
- **Persistence:** GCS Fuse for assets & Cloud SQL for state.
- **Scalability:** Serverless Cloud Run architecture.
- **Observability:** JSON logging enabled by default.

Resources & Documentation

Core References

Module Source: modules/Directus

Infrastructure: directus.tf

Logic: docker-entrypoint.sh

External Documentation

Directus Docs: docs.directus.io

Google Cloud Run: cloud.google.com/run

This architecture provides a secure, self-healing foundation for headless CMS operations on Google Cloud.