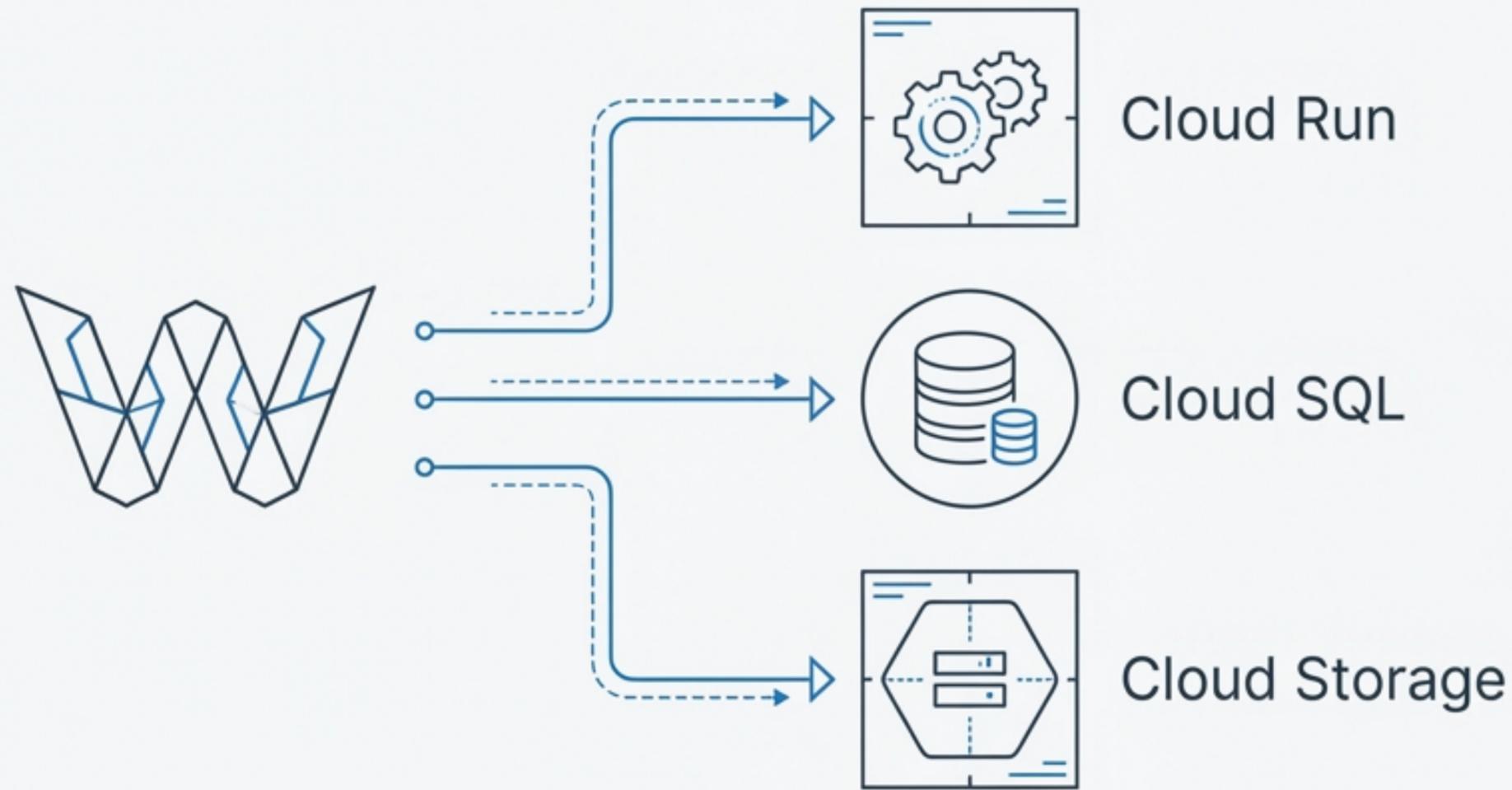


# Wiki.js on Google Cloud Platform

Architecture, Security, and Configuration Guide for the RAD Platform Module



# High-Level Architecture

## Compute

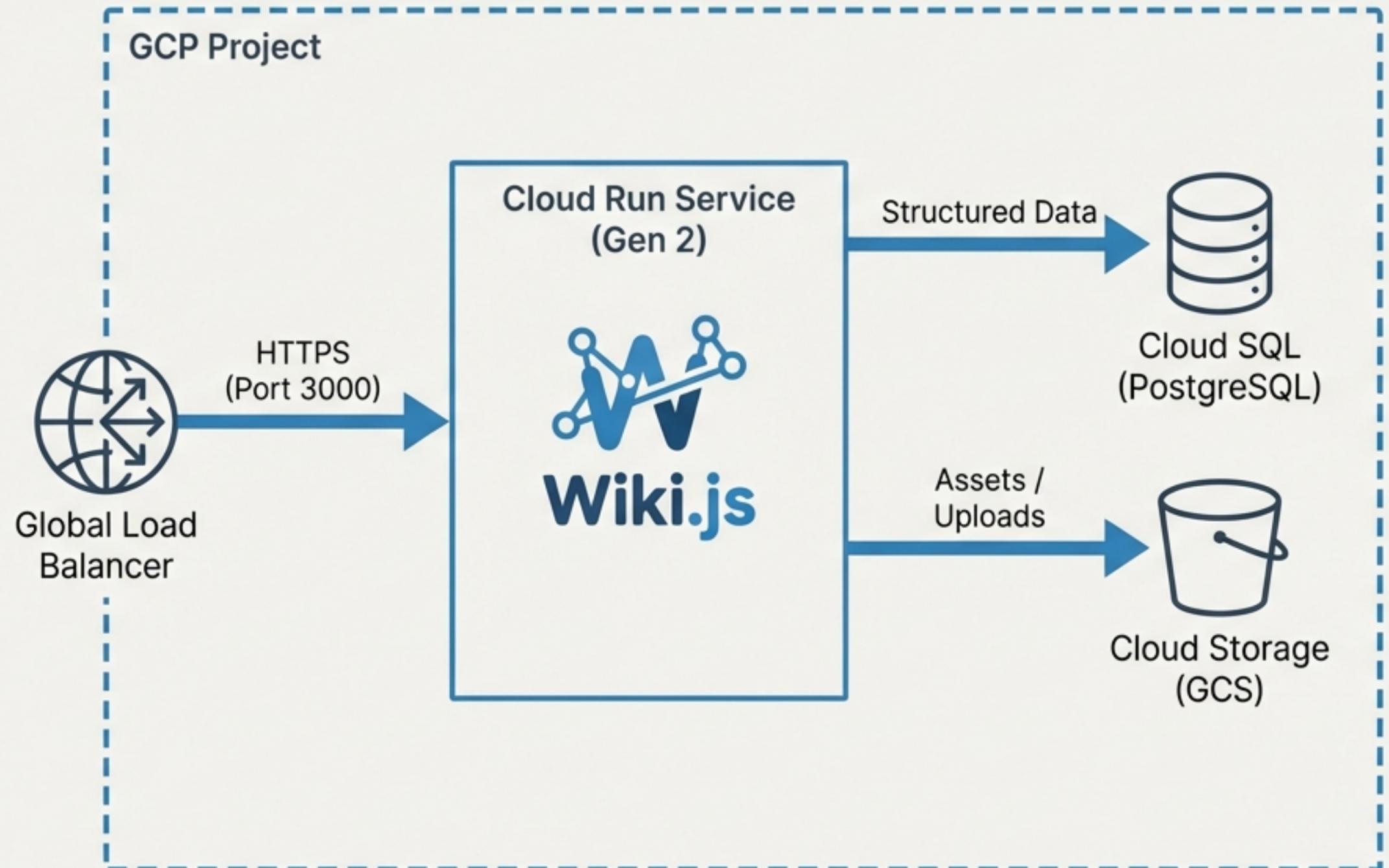
Wiki.js runs as a stateless container on Cloud Run. Scaling is handled automatically.

## State

Persistence is decoupled. Structured text resides in Cloud SQL; binary assets reside in GCS.

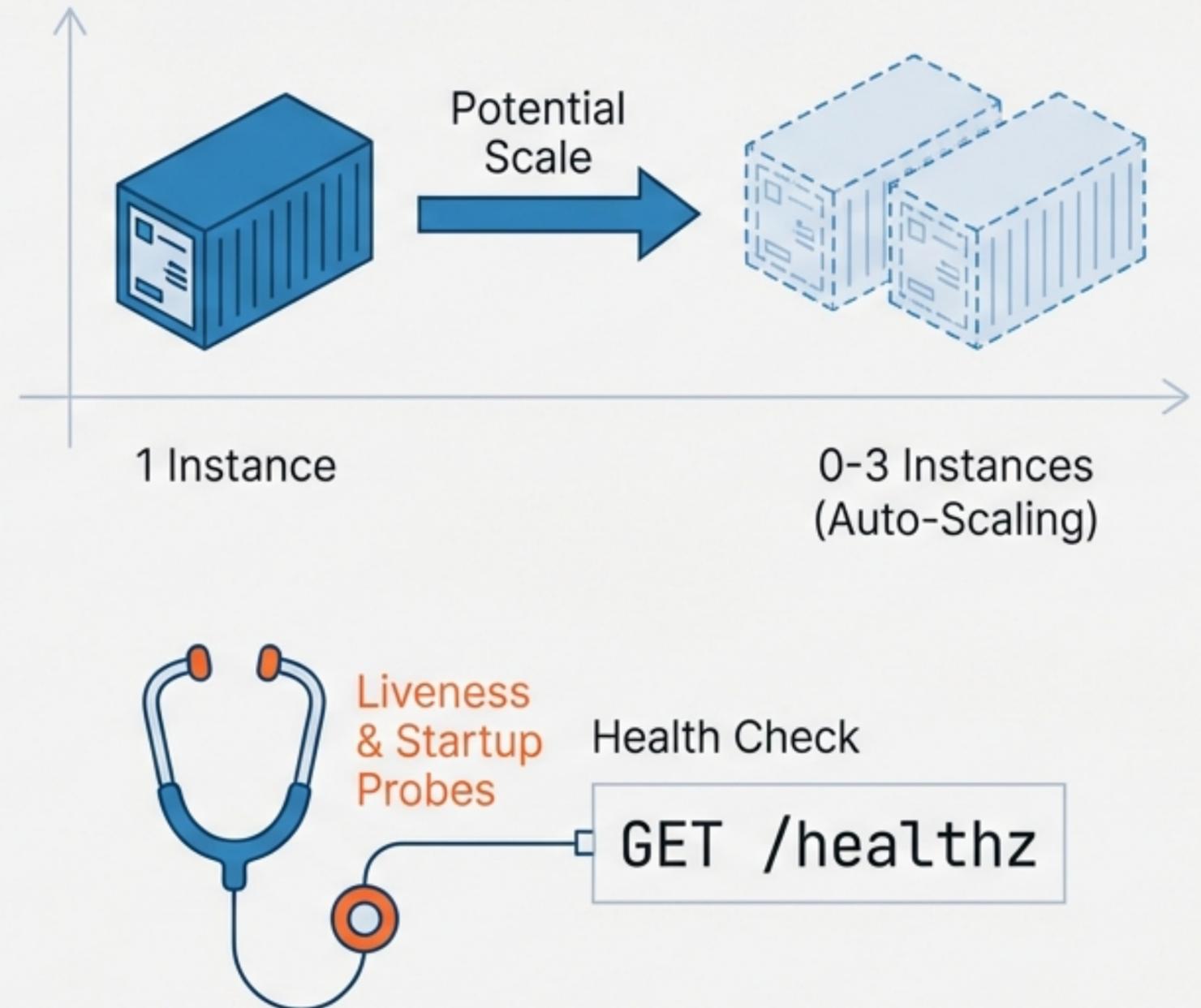
## Networking

Service listens on port 3000. Ingress configurable for internal or public access.

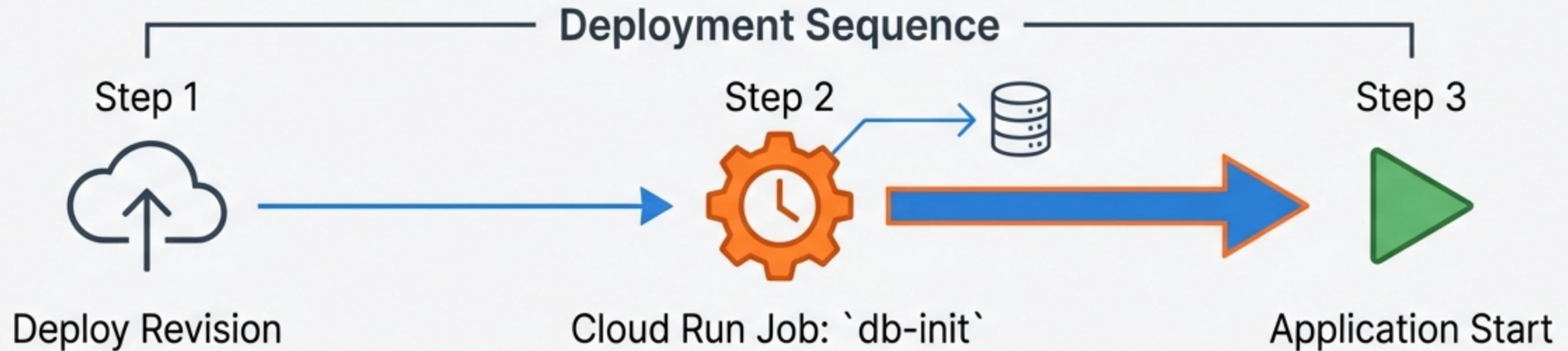


# The Runtime Environment: Cloud Run Configuration

- **Execution Environment:** Gen2 (Stateless Container)
- **Auto-Scaling:** 0 to 3 instances (Default)
- **Networking:** Port 3000
- **Wrapper Pattern:** Uses shared core CloudRunApp with `wikijs.tf` logic.



# Data Persistence & The 'Self-Healing' Database



## Engine Specs

Managed Cloud SQL running PostgreSQL 15. `pg_trgm` extension enabled for native full-text search.

## Connectivity

Connects via Unix Socket at `/cloudsql` or Internal IP. `DB_SSL` set to `false` for Auth Proxy compatibility.

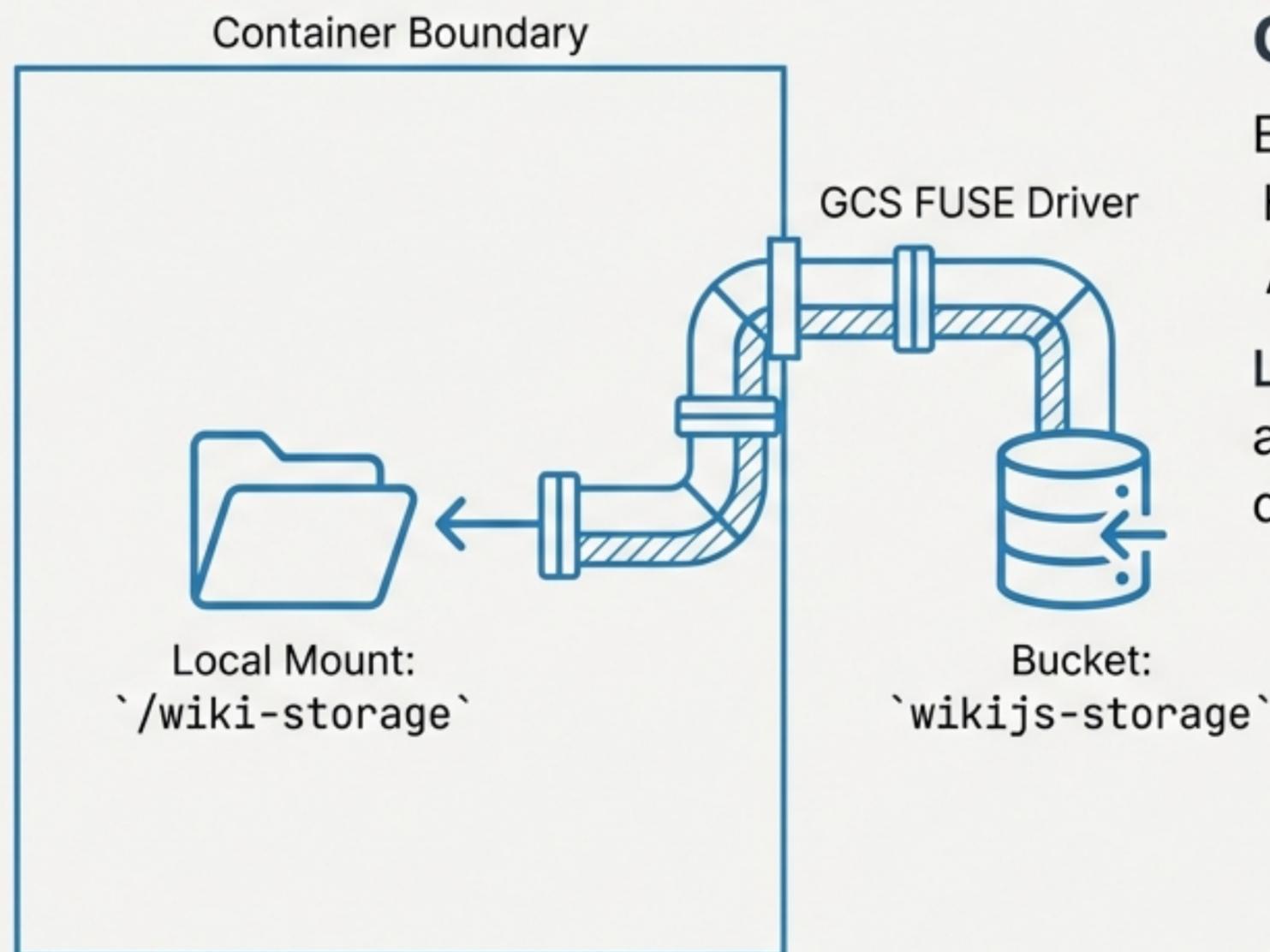
## Automation

Zero-touch provisioning. The `db-init` job ensures user/schema existence automatically.

# Solving Serverless Storage with GCS FUSE

## Problem/Solution

- **Problem:** Containers are ephemeral. Local writes vanish on restart.
- **Solution:** Mount object storage as a file system.

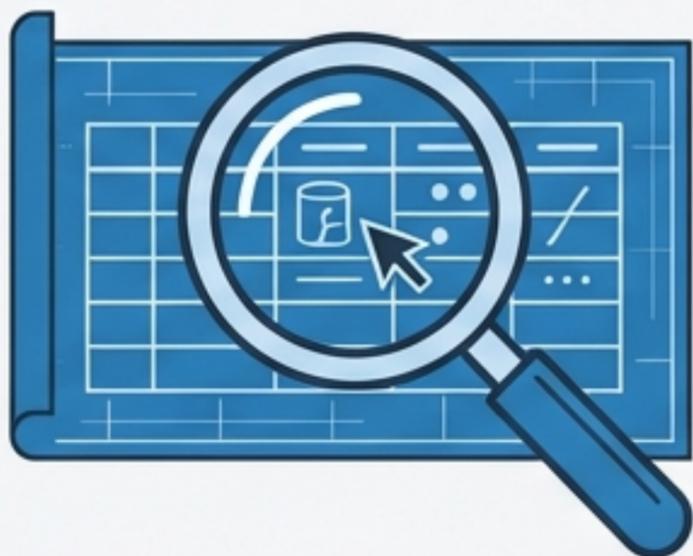


## Config

Environment Variable:  
`HA_STORAGE_PATH = /wiki-storage`

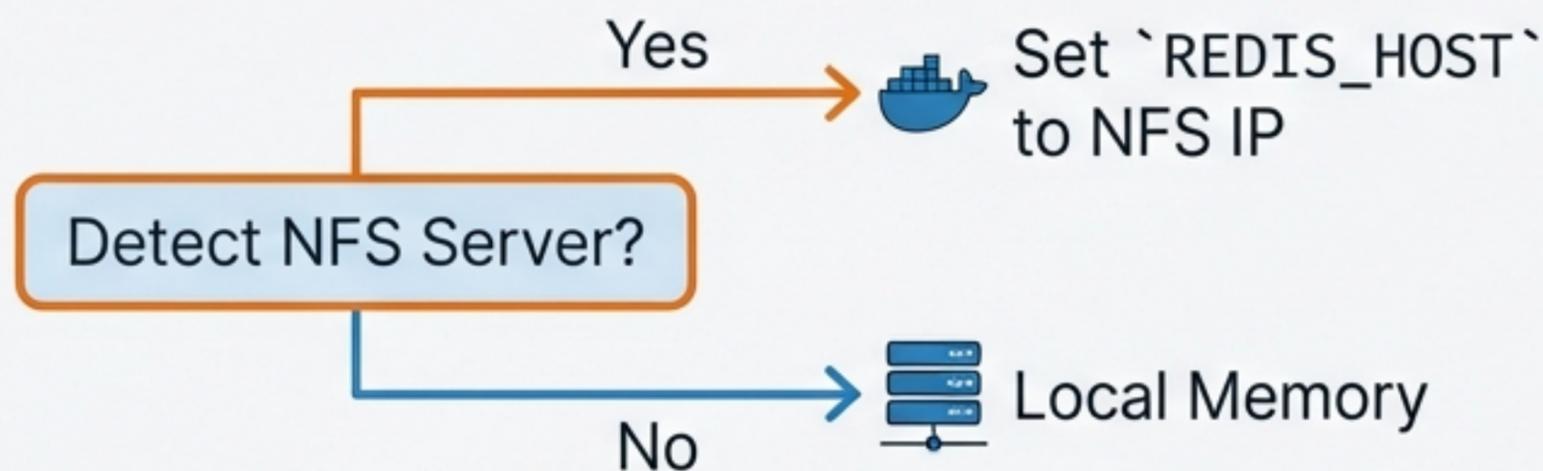
Legacy Option: NFS support at `/mnt` if NFS server detected.

# Search Indexing & Cache Strategy



## Search Architecture

- Native Wiki.js search engine.
- Backed by PostgreSQL `pg\_trgm` extension.
- Enabled **automatically** via module configuration.

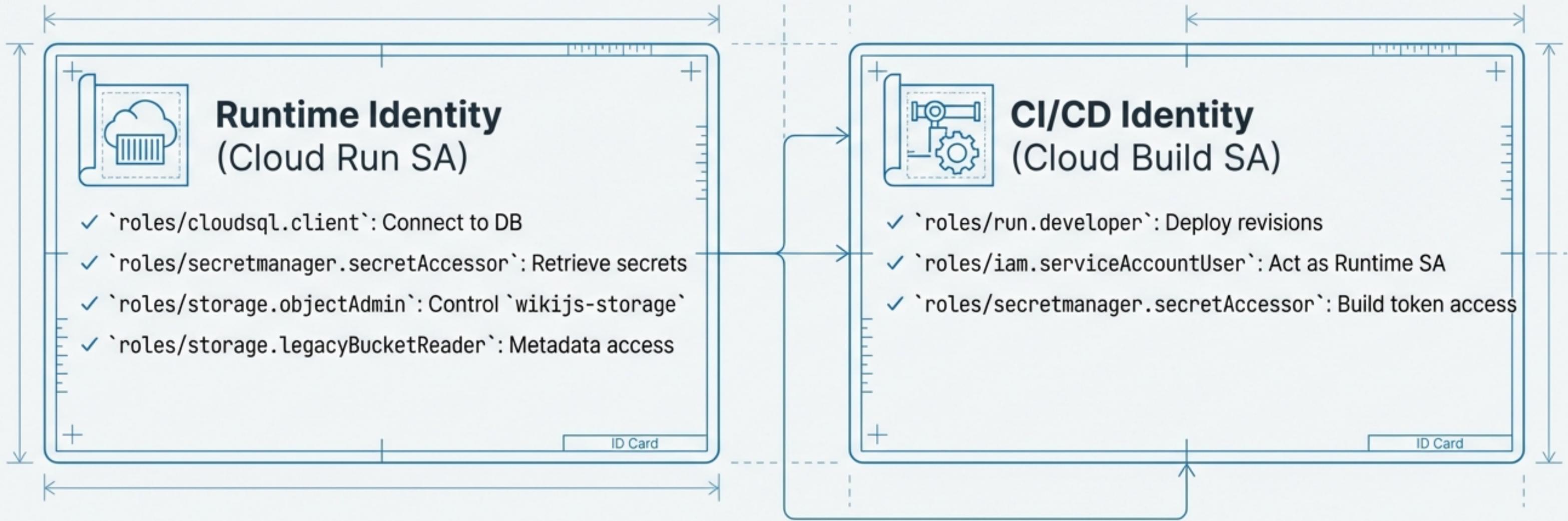


## Caching Strategy

Auto-detection logic utilizes shared infrastructure. If NFS is present, it acts as the Redis host for session caching.

# Identity & Access Management (IAM)

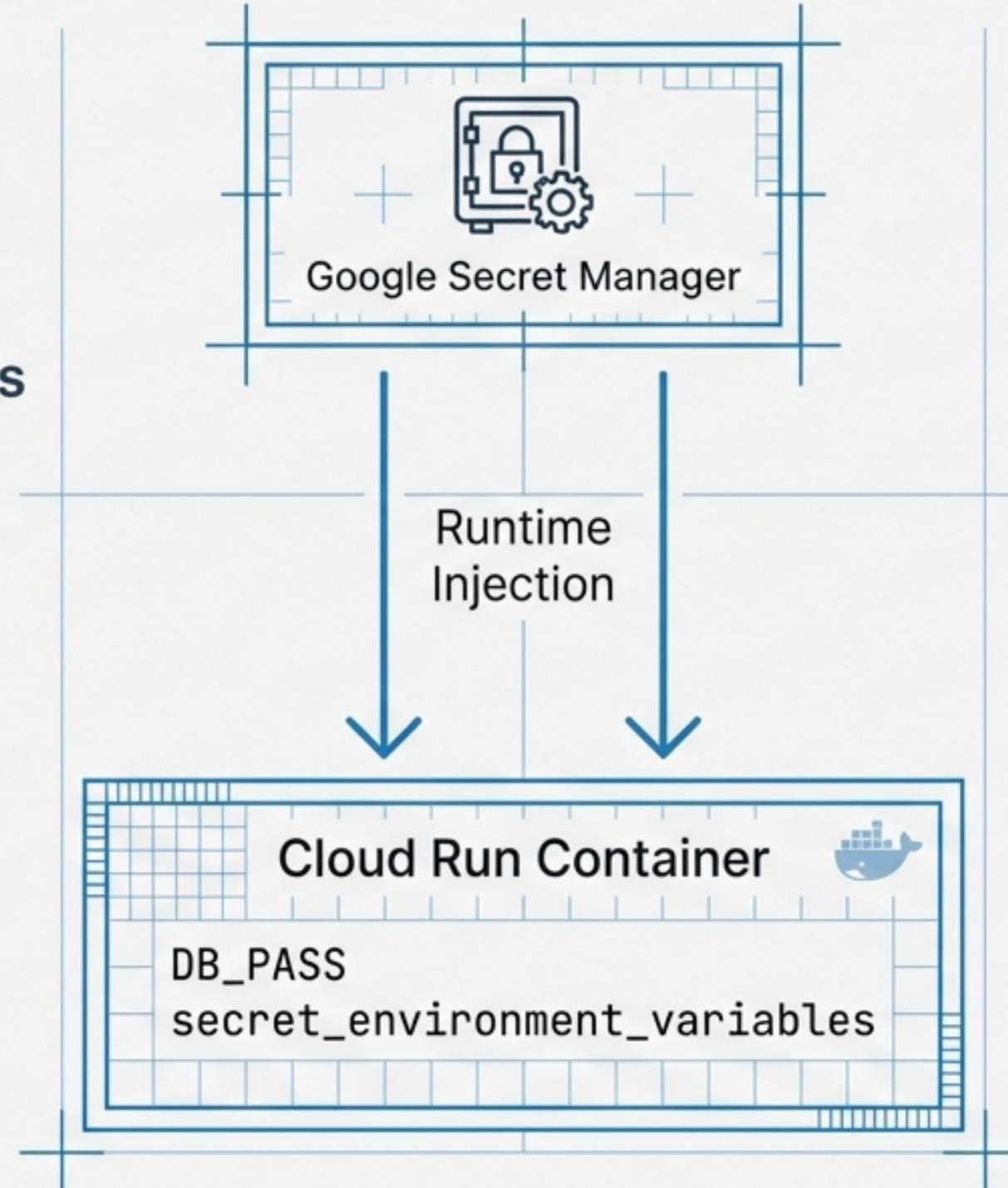
## Principle of Least Privilege Implementation



# Secret Management Strategy

## Zero Hardcoded Credentials

Database passwords are generated in Secret Manager and never exposed in the repo.



## Environment Mapping

Sensitive variables are mapped directly from Secret Manager versions to container environment variables.

# Infrastructure as Code Configuration

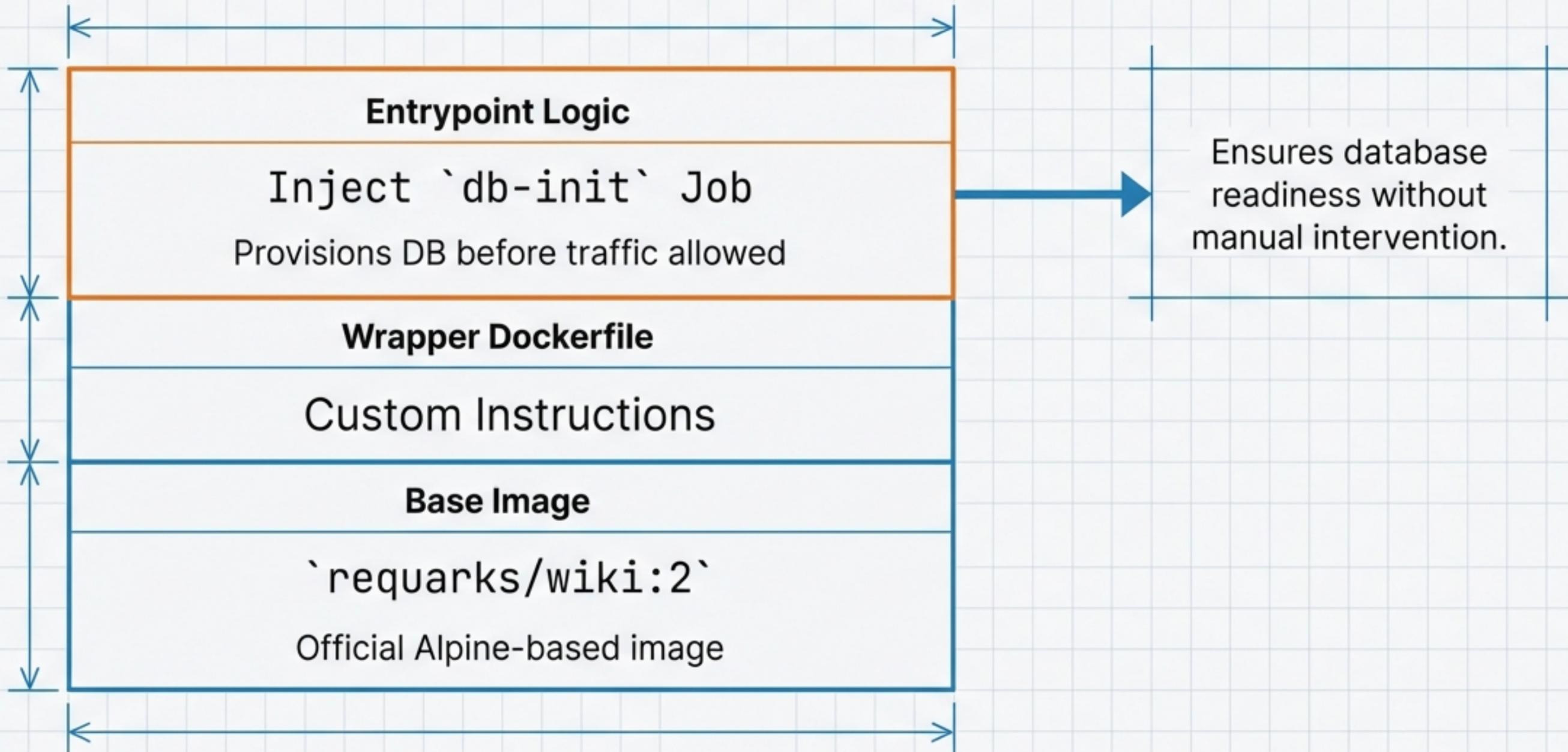
File: wikijs.tf

```
-----  
variable 'DB_TYPE' {  
  default = 'postgres'  
}  
  
variable 'HA_STORAGE_PATH' {  
  default = '/wiki-storage'  
}  
  
variable 'DB_SSL' {  
  default = 'false' // Uses Unix Socket  
}  
  
resource 'allocation' {  
  cpu      = '1000m'  
  memory  = '2Gi'  
}
```

## CI/CD Control:

```
`enable_cicd_trigger`  
JetBrains Mono  
toggle available for  
automated Cloud Build  
setup.
```

# Container Strategy & Customization



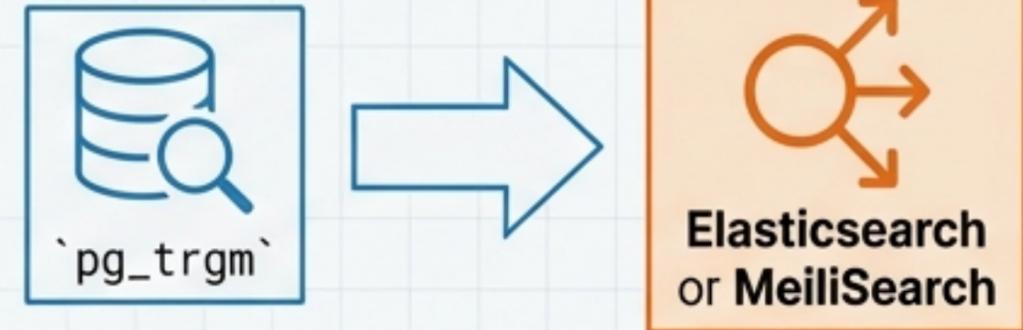
# Roadmap: Performance & Functionality

## Scalability



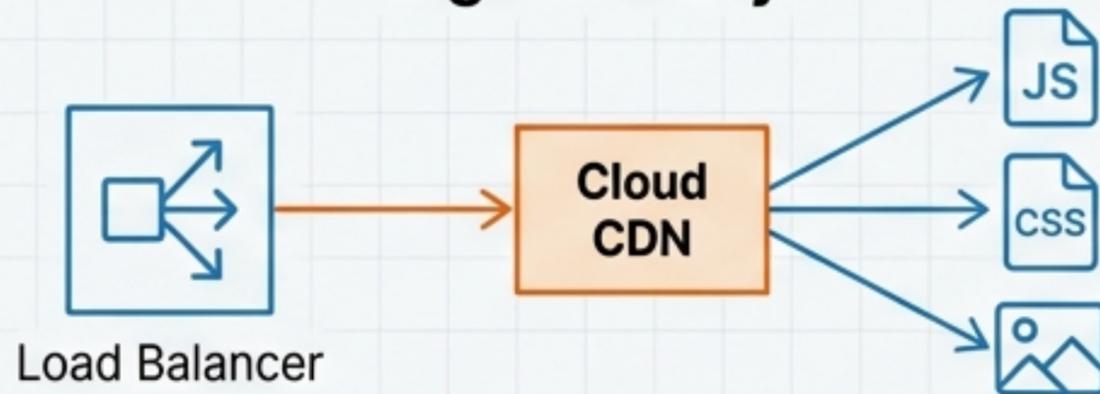
Replace NFS-Redis with **Cloud Memorystore**.  
Provides high-availability caching closer to compute.

## Search Engine



Upgrade ``pg_trgm`` to **Elasticsearch** or **MeiliSearch**.  
Essential for large-scale knowledge bases.

## Edge Delivery



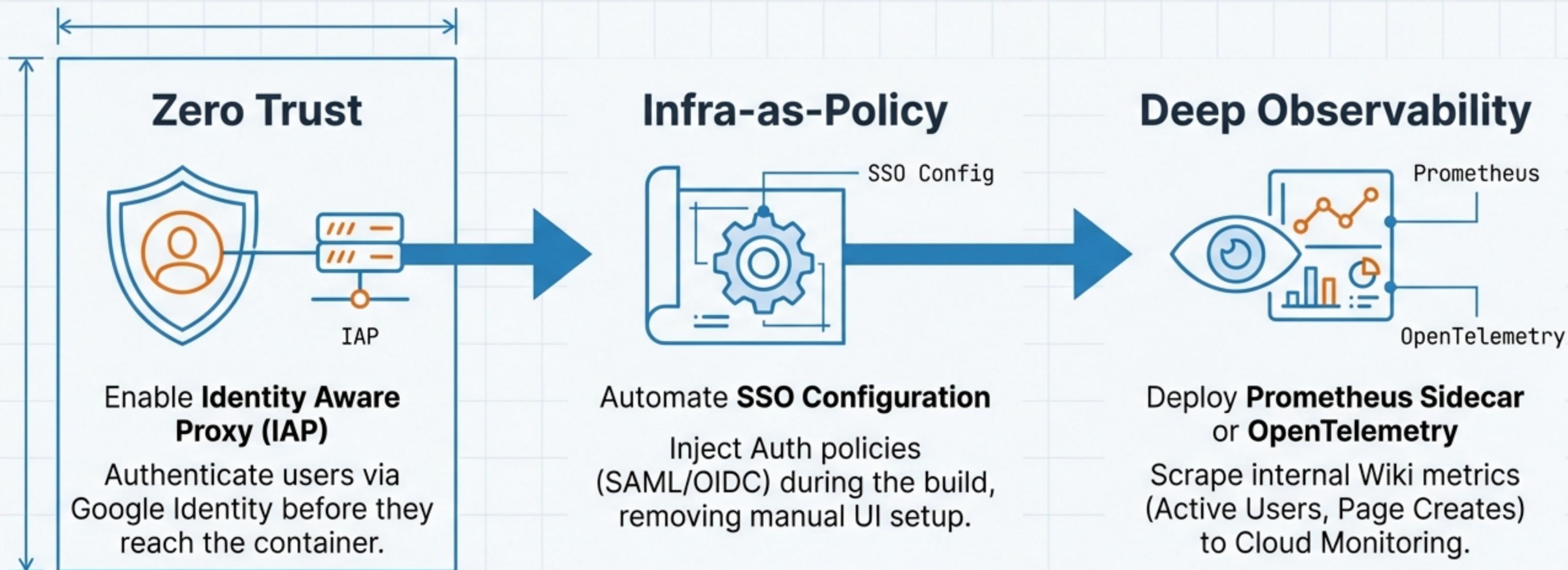
Enable **Cloud CDN** on Load Balancer.  
Cache static assets (JS, CSS, Images) at the edge.

## Export Features



Add Chromium libraries to Dockerfile.  
Enables server-side **Export to PDF** functionality.

# Roadmap: Security & Observability



Disaster Recovery Goal: Cold storage export for assets & DB dumps.