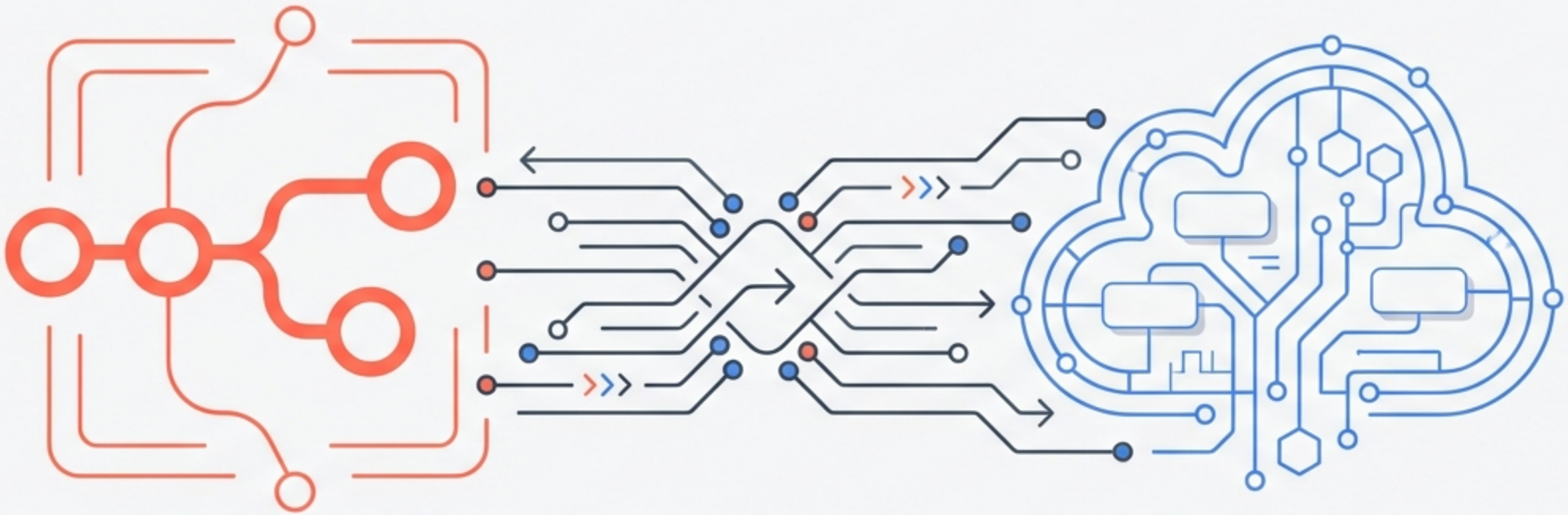
N8N on Google Cloud Platform

Architecture, Security, and Configuration of the RAD Platform modules/N8N Implementation.



RAD Platform Documentation | modules/N8N

The Wrapper Module Approach

The Concept:

The `modules/N8N` module functions as a wrapper leveraging core logic from `modules/CloudRunApp`.

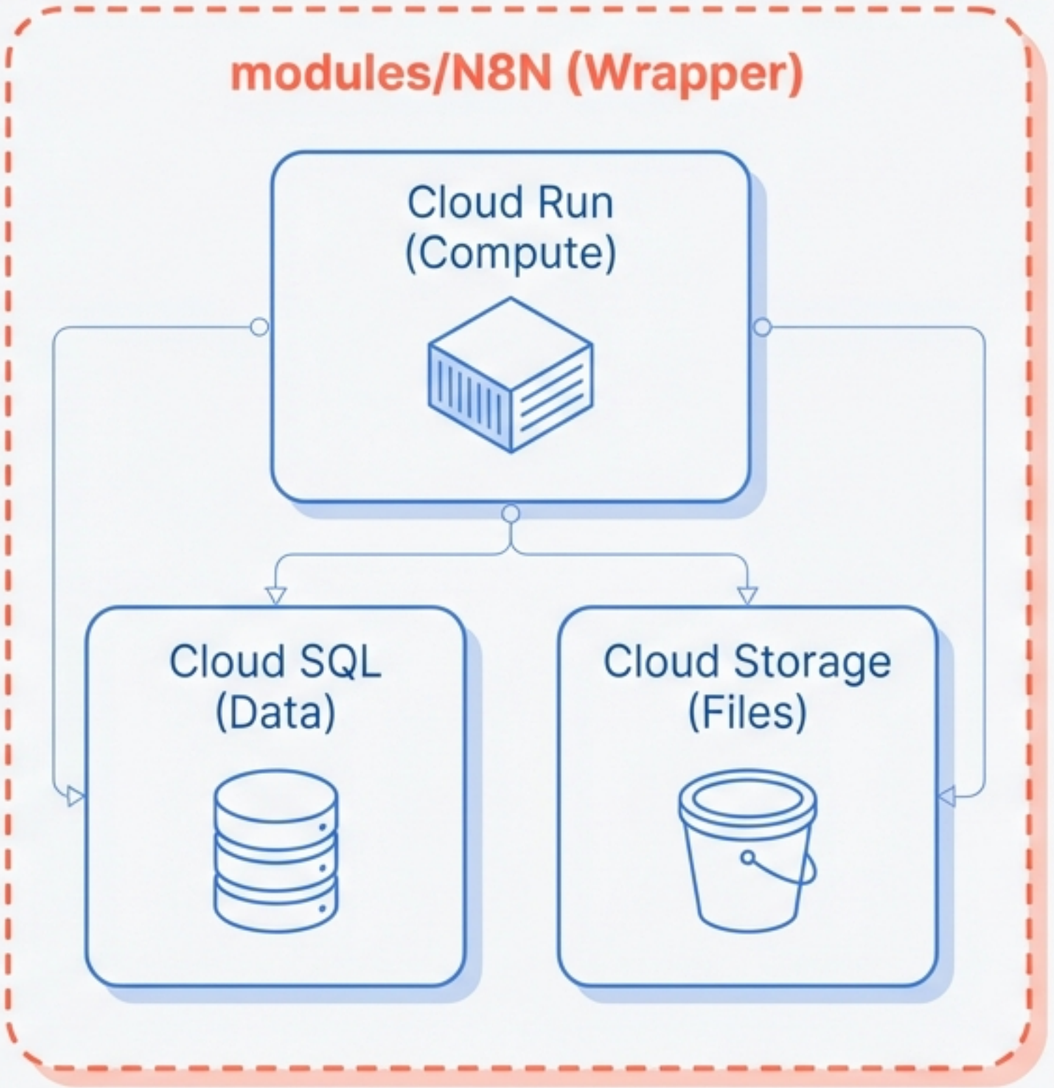
It streamlines the deployment of the n8n workflow automation platform by pre-packaging infrastructure.

The Challenge:

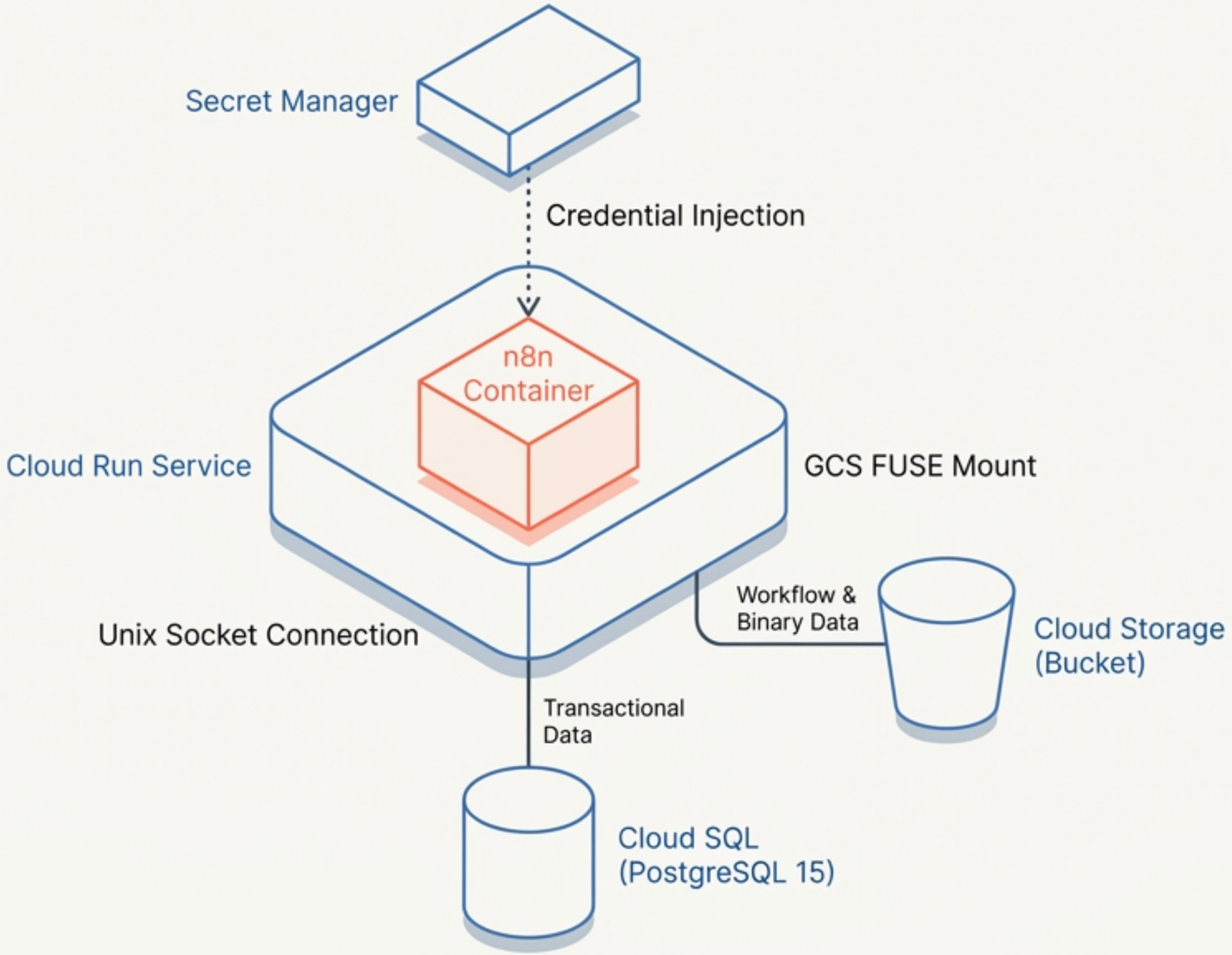
Running stateful automation applications on stateless serverless containers.

The Solution:

A containerized service on Google Cloud Run, backed by Cloud SQL (PostgreSQL) for transactional data and Cloud Storage for file persistence.



System Architecture and Interconnectivity



The Compute Layer: Serverless Execution

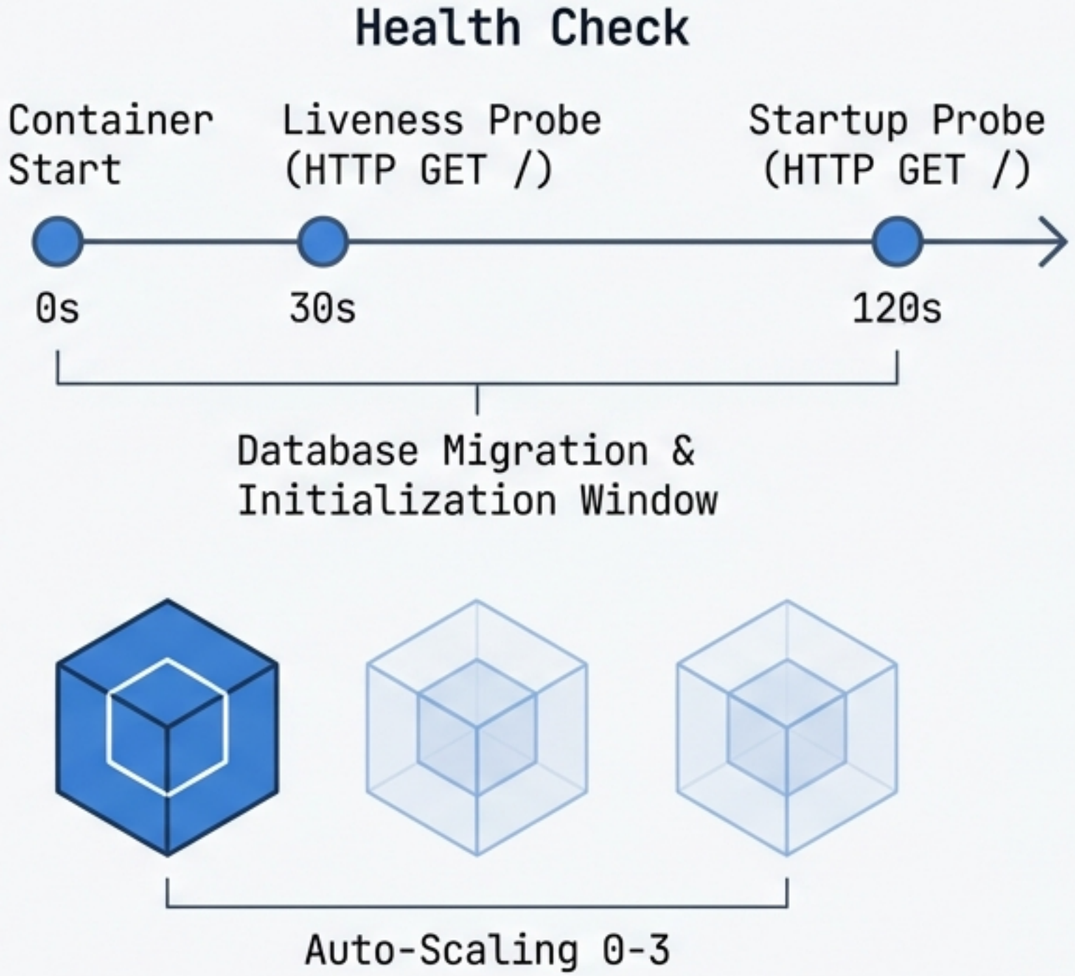
Service: Google Cloud Run

Image: n8nio/n8n (Default)
or custom via scripts/n8n/Dockerfile

Resources: 2 vCPU / 4Gi Memory
(Configurable)

Scaling:
Min 0 / Max 3 Instances

Session Affinity:
Enabled (session_affinity = true)



The Data Layer: Robust Relational Storage

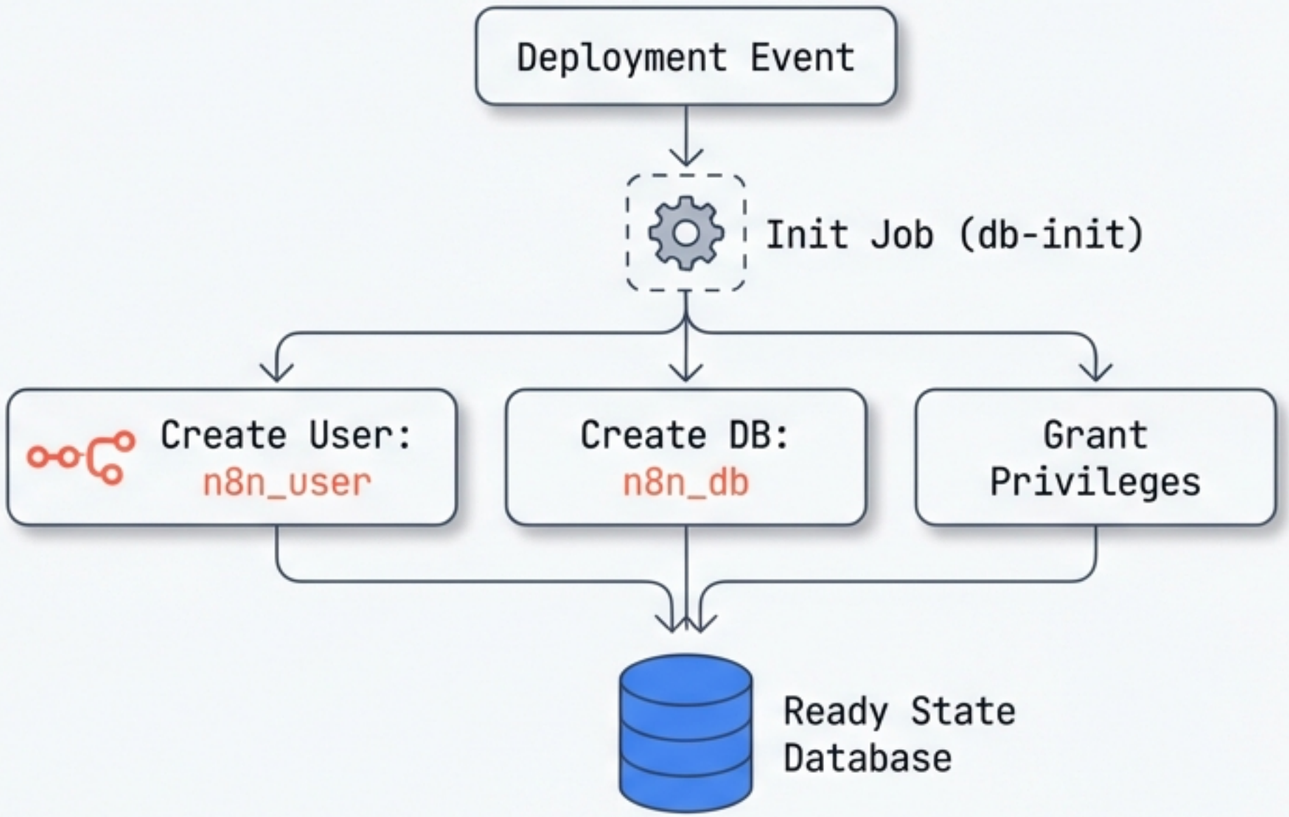
Specs

Service: Cloud SQL

Engine: PostgreSQL 15

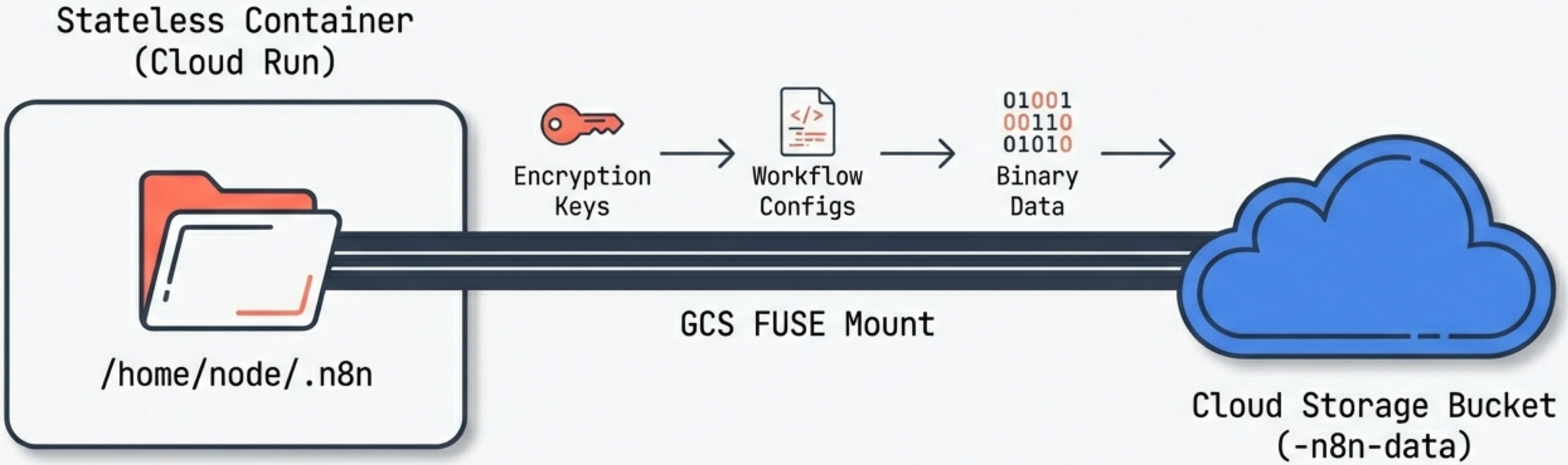
Connection: Secure Unix Socket (/cloudsql/...)

The Init Job



Value Add: Eliminates manual database setup toil and ensures the application lands in a ready-to-use state.

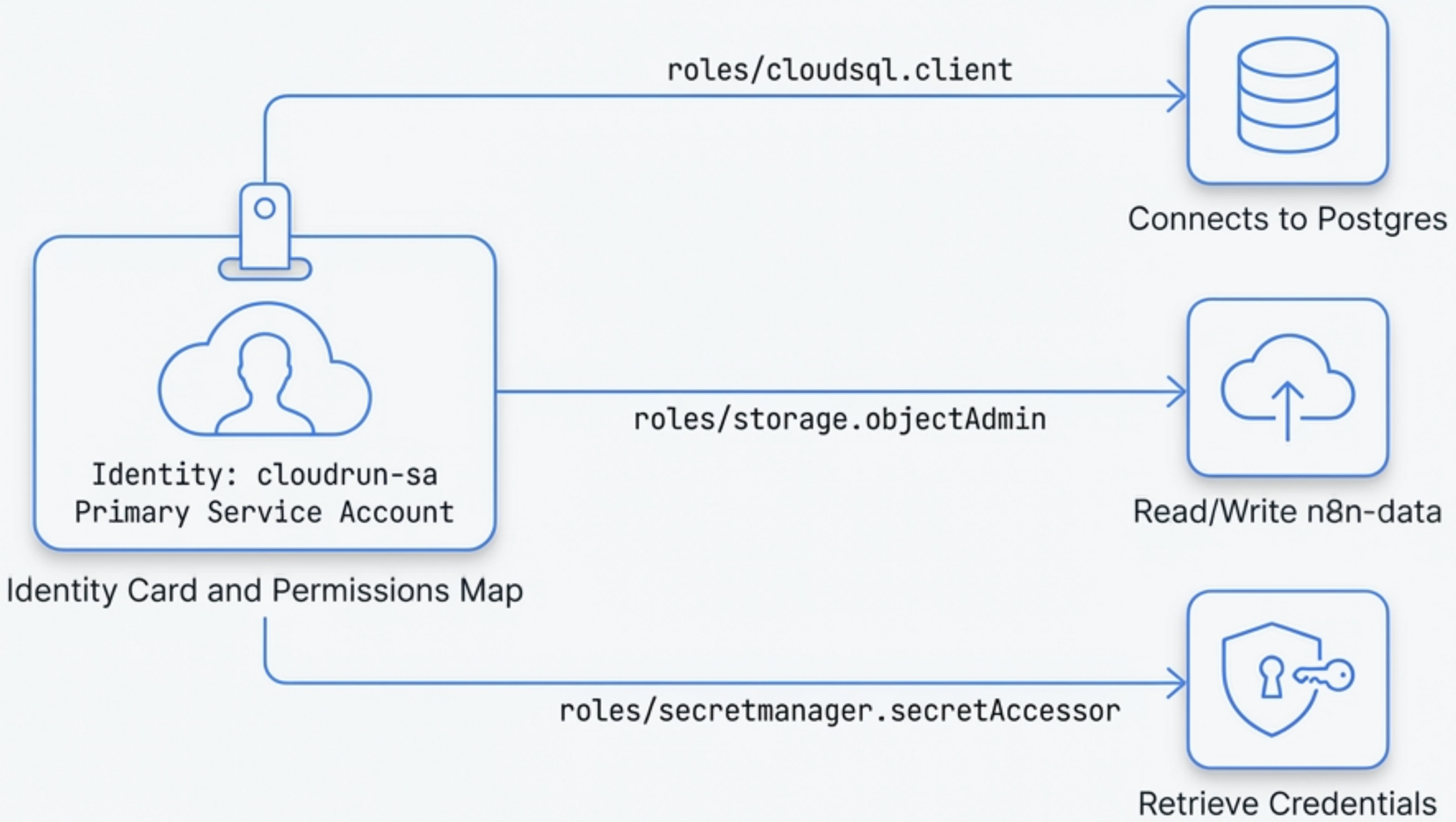
The Persistence Layer: Solving for Statelessness



Mechanism: Filesystem in User Space (FUSE)

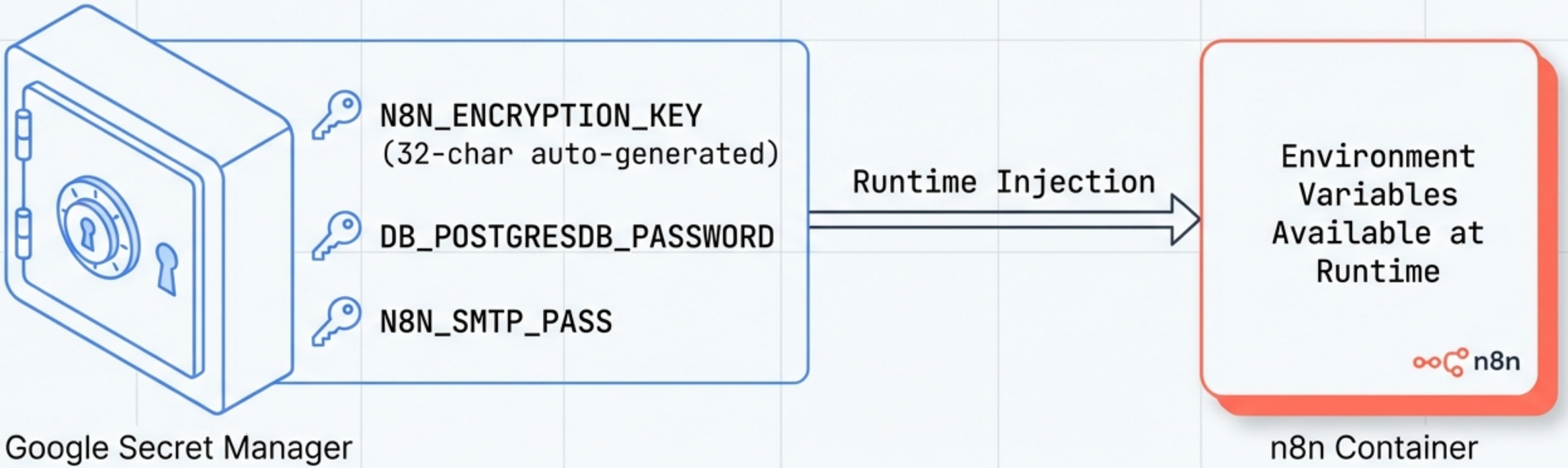
Key Benefit: Decouples storage from compute, preventing data loss when Cloud Run scales to zero or recycles containers.

Identity and Access Management Strategy



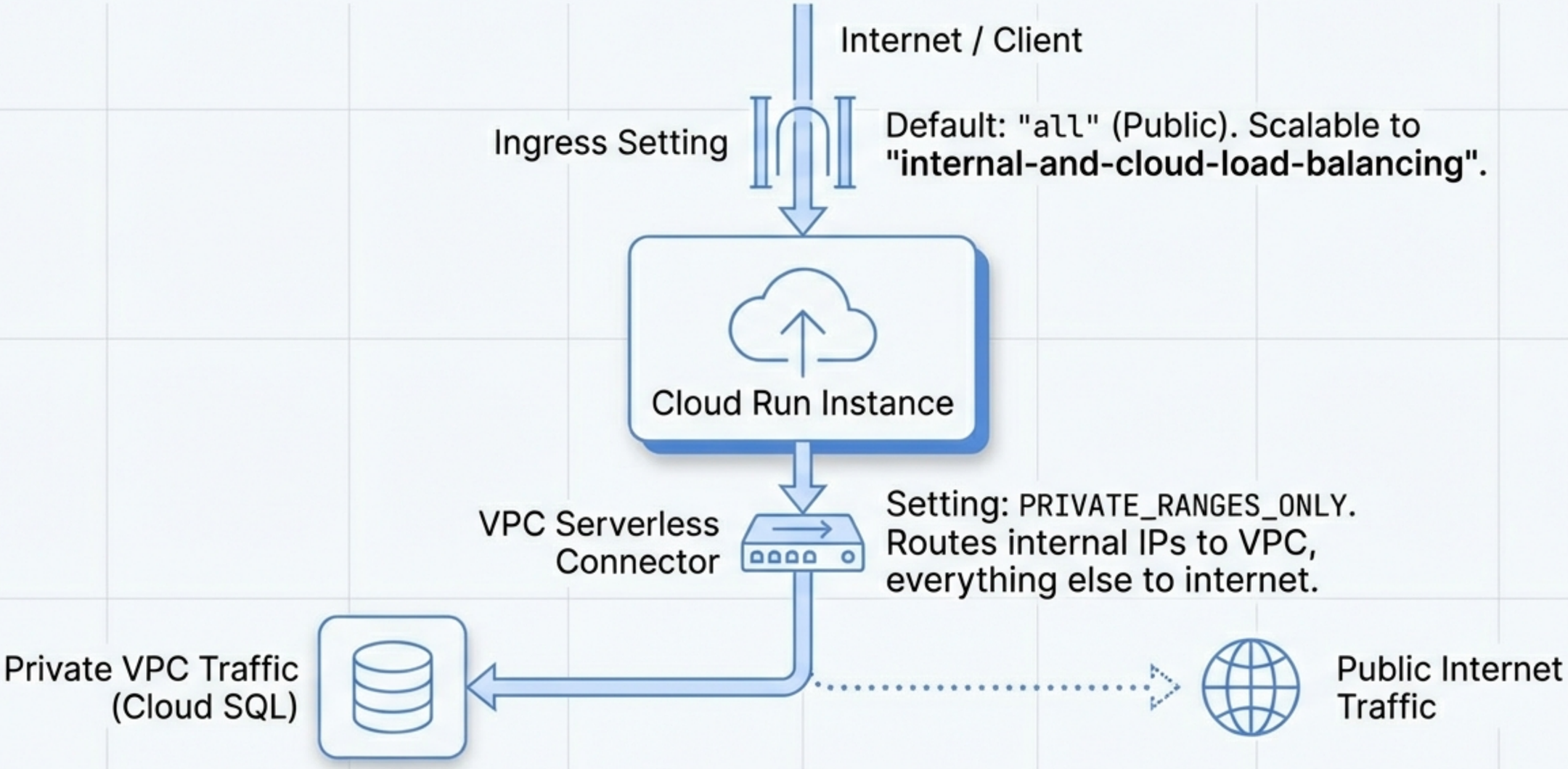
Secondary Identity: cloudbuild-sa (CI/CD Operations)

Secret Management and Credential Injection



Policy: No hardcoded credentials. Ever. Secrets are auto-generated during infrastructure provisioning.

Network Traffic and Connectivity Controls



Runtime Configuration and Environment Variables

System Configuration

Core Variables

N8N_PORT	: 5678
DB_TYPE	: postgresdb
N8N_ENCRYPTION_KEY	: [Secret Reference]

Execution Logging

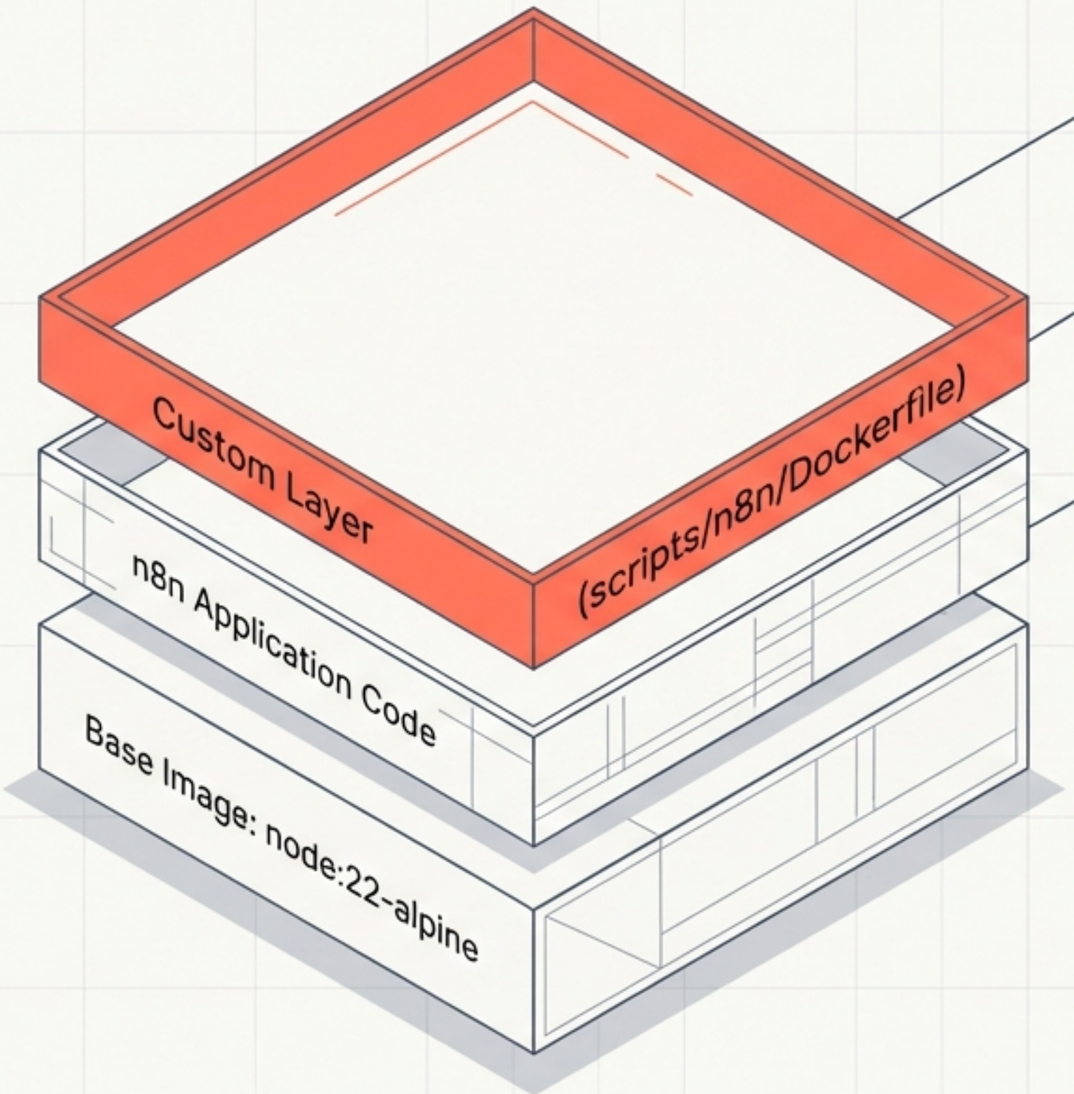
EXECUTIONS_DATA_SAVE_ON_ERROR	: all
EXECUTIONS_DATA_SAVE_ON_SUCCESS	: all

Binary Handling

N8N_DEFAULT_BINARY_DATA_MODE	: filesystem

Note: Optimization for Cloud Run memory limits by offloading to GCS mount.

Build Strategy and Container Customization



OS Level Deps:
`python3, jq, system packages`

App Level Deps:
`npm install custom-nodes`

Storage Config:
`Optional NFS mapping to /mnt`

Default Image: `n8nio/n8n`.
Fully extensible build pipeline.

n8n

Future Scaling: Moving to Queue Mode

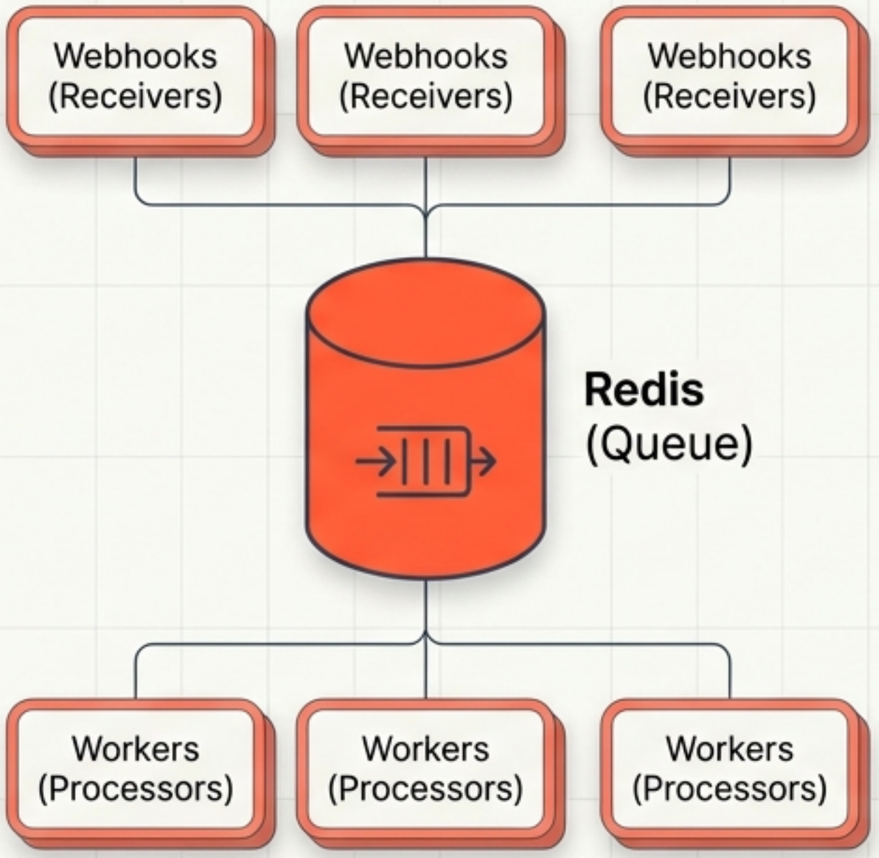
Current: Monolith

n8n (Monolith)

Webhook Receiver
+
Worker Processor

Relies on Sticky Sessions.
Harder to scale.

Future: Microservices



- Architecture Change: Split receivers and processors.
- Requirement: Introduce Redis via `modules/GCP_Services/redis.tf`
- Config: `QUEUE_BULL_REDIS_HOST`

n8n

Performance Optimization and Maintenance

Database Hygiene

Problem: Log Bloat via `EXECUTIONS_DATA_SAVE_ON_SUCCESS=all`

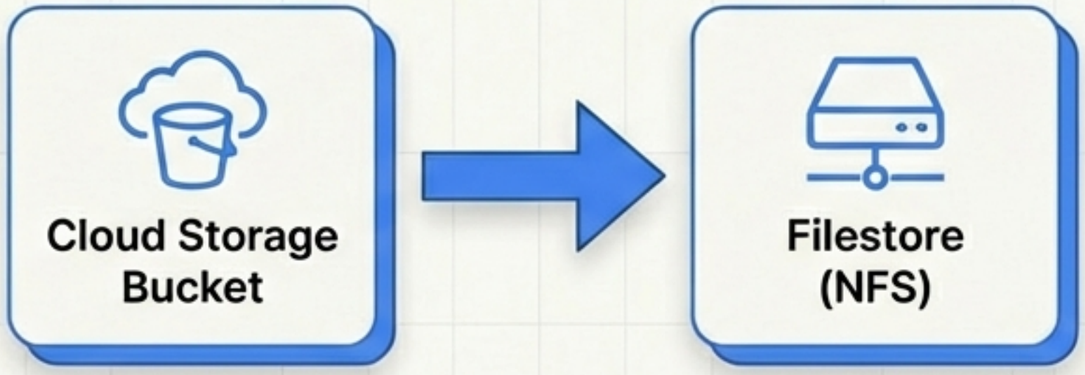
Action: Enable Auto-Pruning

```
EXECUTIONS_DATA_PRUNE=true
EXECUTIONS_DATA_MAX_AGE=168 (1 week)
```

High I/O Storage

Problem: GCS FUSE Latency on heavy operations

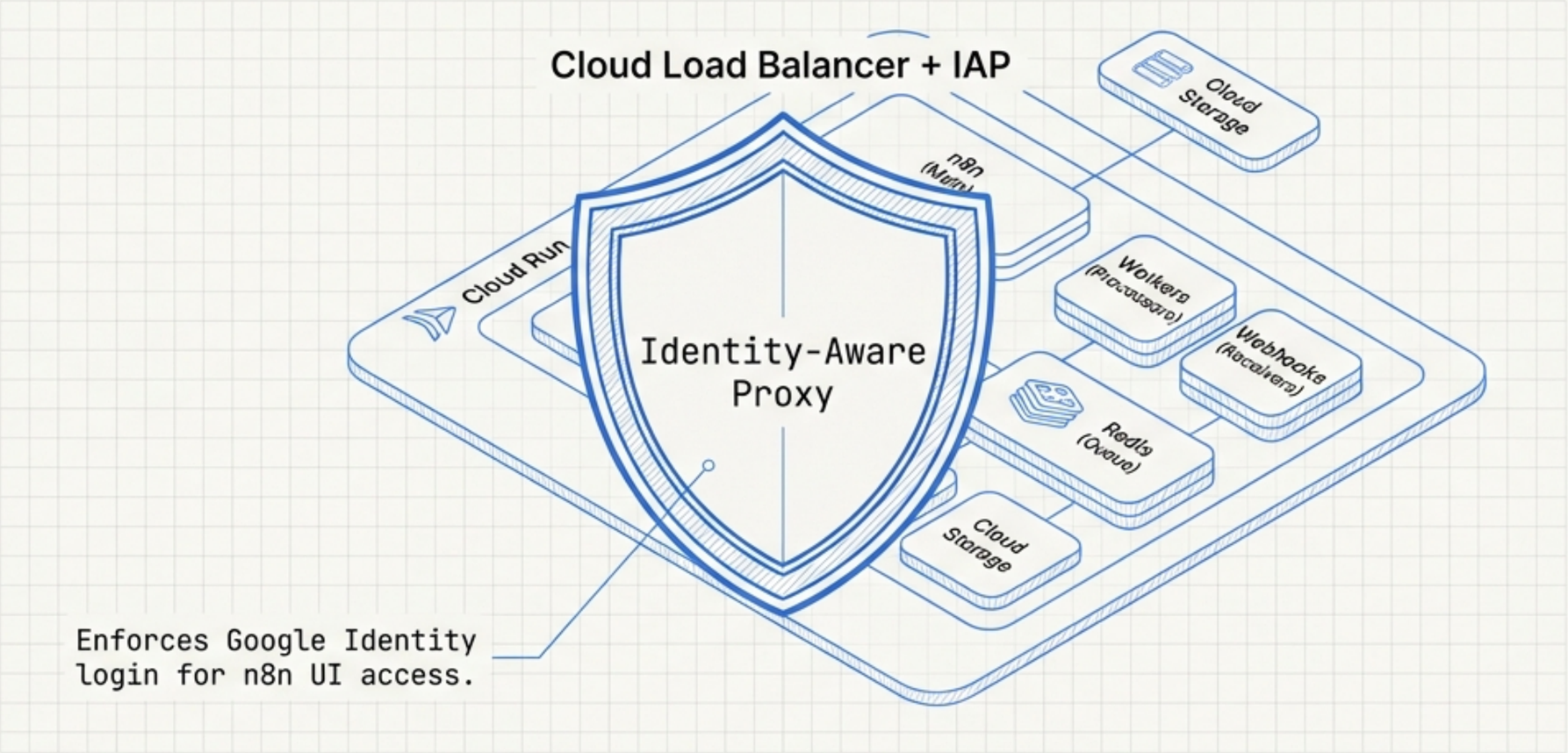
Action: Migrate Binary Storage



Use Filestore for lower latency on high-throughput binary processing.

n8n

Advanced Security Hardening



IAM Scoping

Audit Action: Verify `cloudrun-sa` adheres to Least Privilege.

Constraint: Ensure Service Account does NOT possess 'Project Editor' roles.

Deployment Roadmap and Feature Summary

Current Capabilities (Ready Now)

- Automated Database Setup (Init Jobs)
- Stateless Compute with Stateful Persistence (GCS FUSE)
- Secure Defaults (Secret Manager integration)
- Custom Build Support (Alpine/Python/Node)

Recommended Evolution (Roadmap)

- Redis Implementation for Queue Mode
- IAP Integration for Zero-Trust Access
- OpenTelemetry Monitoring Sidecar (N8N_METRICS)

A robust, scalable foundation for workflow automation on Google Cloud.