Juno / Design Agency

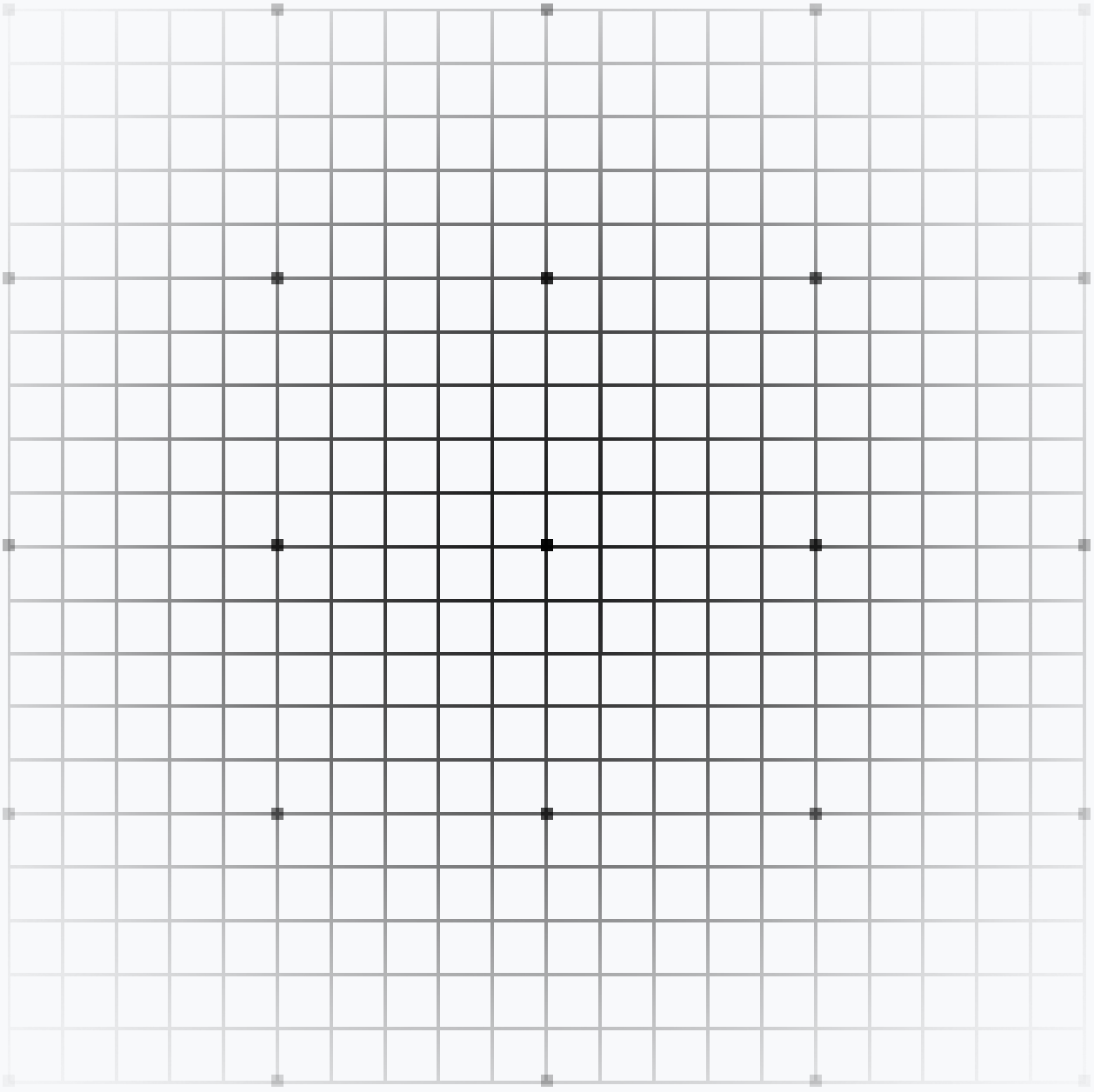
May 24, 2024

Prepared for
Ingenius Team

Prepared by
Vidushi Meel
Asritha Bodepudi

Cloud Architecture Proposal

Migration to GCP & Implementation of Best Practices



Contents

Introduction

Overview | Migrate the Ingenius codebase from the existing Vercel x Heroku monolithic deployment to microservices hosted on Google Cloud Platform. This proposal includes price breakdowns including a detailed cost analysis which finds room for Ingenius to cut costs on cloud fees as well as a proposal for the overall migration.

Infrastructure and Security Setup for GCP Migration

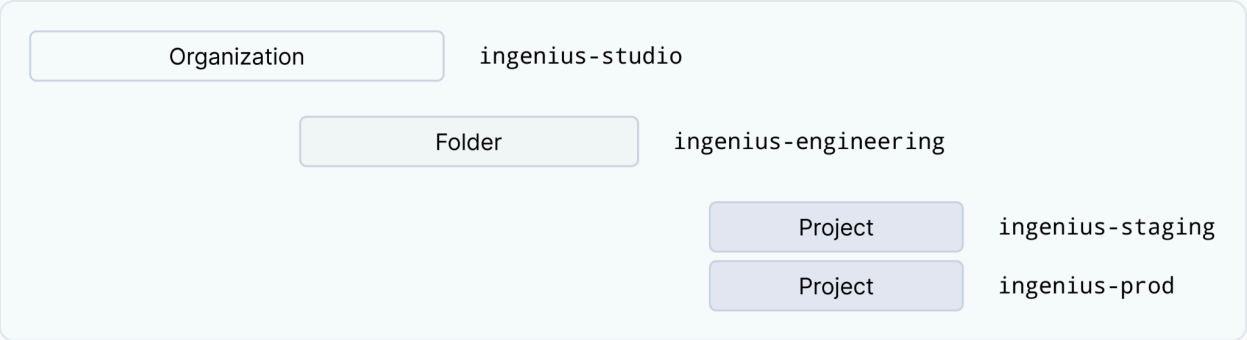
Goals | Deploy a scalable microservices architecture on Google Cloud Platform (GCP) tailored to the needs of Ingenius Studio. In this section,we provide a comprehensive overview of the proposed infrastructure and security measures for the GCP migration.

1. API Quotas

Based on predicted traffic patterns, adjustments to request higher API quotas will be necessary to accommodate the anticipated load. These adjustments will be made through the Cloud Quotas API console.

2. Resource Hierarchy

A resource hierarchy structure helps in managing permissions, organizing billing, and applying policies at different levels. For instance, a staging environment allows for changes to take place internally for the purpose of demonstrations or trials, whereas the production environment is public-facing, viewable by all public users.



Proposed Resource Hierarchy

3. Infrastructure as Code (IaaC)

Terraform with GCP

Terraform is recommended because it enables infrastructure as code, allowing for automated, consistent, and reproducible setups across different environments. Terraform programmatically documents "what happened" in the infrastructure setup, ensuring that each time you run `terraform apply`, you achieve consistent and identical infrastructure deployments. **Using Terraform now will conserve Ingenius' runway in switching to Terraform later.**

The entire state of the Ingenius GCP will be managed using Terraform Cloud, with state files stored in GCP Cloud Storage Buckets to ensure best DevOps practices and maintain a single source of truth. Corresponding lab and prod environments will be set up using Terraform to ensure consistency and reproducibility across deployments.

4. Networking

Overview | Each Project resource will use the default VPC (Virtual Private Cloud). We propose using CloudDNS to manage DNS records within our Project.

- CloudDNS for DNS record management.
- Cloud Load Balancing for traffic distribution across various backend services.
- NAT Gateway for internet access from private subnets.
- Network Endpoint Groups (NEG) for directing traffic to Cloud Run or App Engine services.
- Firewall Rules to secure the network.

Environments | There will be varied pricing for both environments, and the staging environment will be priced lower than the production.

- Production (prod): Full-scale resources.
- Lab: Scaled-down resources for testing and development.

Service	Environment	Monthly	Details
Cloud Load Balancing	Production	$20	Up to 100 GiB inbound/outbound, 5 forwarding rules
NAT Gateway	Production	$8-$15	Per environment, up to 100 GiB data processed
Network Endpoint Group (NEG)	Production	Included in Load Balancing	Connects to Cloud Run/App Engine with Load Balancer
CloudDNS	Production	Included	Manage DNS Records
Firewall Rules	Production	Included	Security rules for networking

Networking Pricing

5. Identity & Access Management (IAM)

Overview | This structure supports the Principle of Least Privilege, ensuring minimal access levels necessary for operational effectiveness.

IAM Principles

Juno-Developers (Google Group)

Ingenius-Developers (Google Group)

Admin (Google Group)

Finance (Google Group)

IAM Policies (Project scope)

Juno-Developers: roles/editor

Ingenius-Developers: roles/editor

Admin: roles/viewer

IAM Policies (Billing Account scope)

Juno-Developers: roles/billing.admin, roles/billing.user

Ingenius-Developers: roles/billing.admin, roles/billing.user

Admin: roles/billing.admin

Finance: roles/billing.creator

6. Container Registry

An Artifact Registry will be created and managed within each Project to host Docker images securely and efficiently.

Deployment Options

Introduction | Here, we provide two deployment options with associated pricing and advantage/disadvantage breakdowns: Google Kubernetes Engine (GKE) and Google Cloud Run.

Overview | GKE is a container orchestration platform, whereas Cloud Run is a serverless platform for stateless containerized microservices. GKE enables Kubernetes features like namespaces while Cloud Run is fully managed and does not have such fine-grained Kubernetes-based functionality.

Pricing | Note that the prices provided are for the production environment. The staging environment will be built with much less compute. **We recommend Cloud Run over GKE based on the cost analysis.**

1. GKE | Google Kubernetes Engine

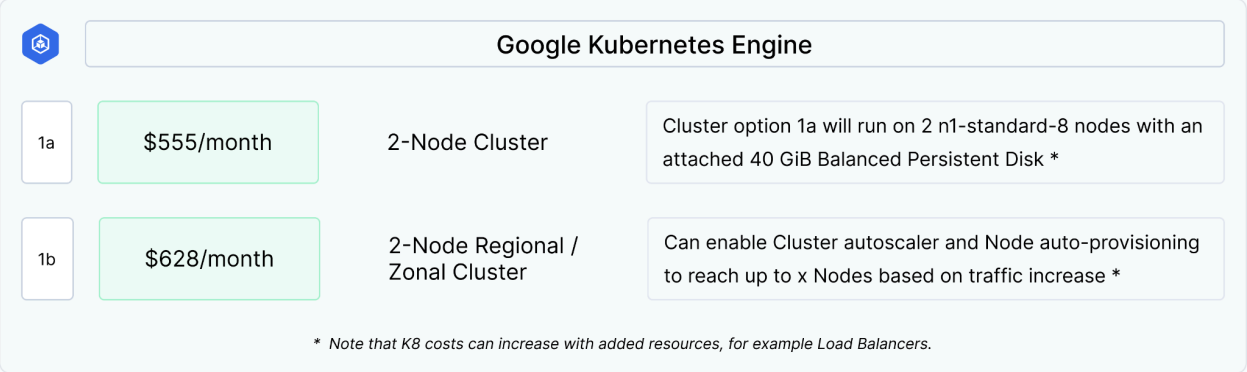
Advantages

- Integrated management of certificates and load balancing.
- Scalable to handle increasing applications and traffic.
- Supports autoscaling to efficiently manage resource allocation.

Disadvantages Higher cost relative to other solutions.

Options | Note: Costs can increase with additional resources such as Load Balancers. View pricing breakdown chart below for visual depiction of cost analysis.

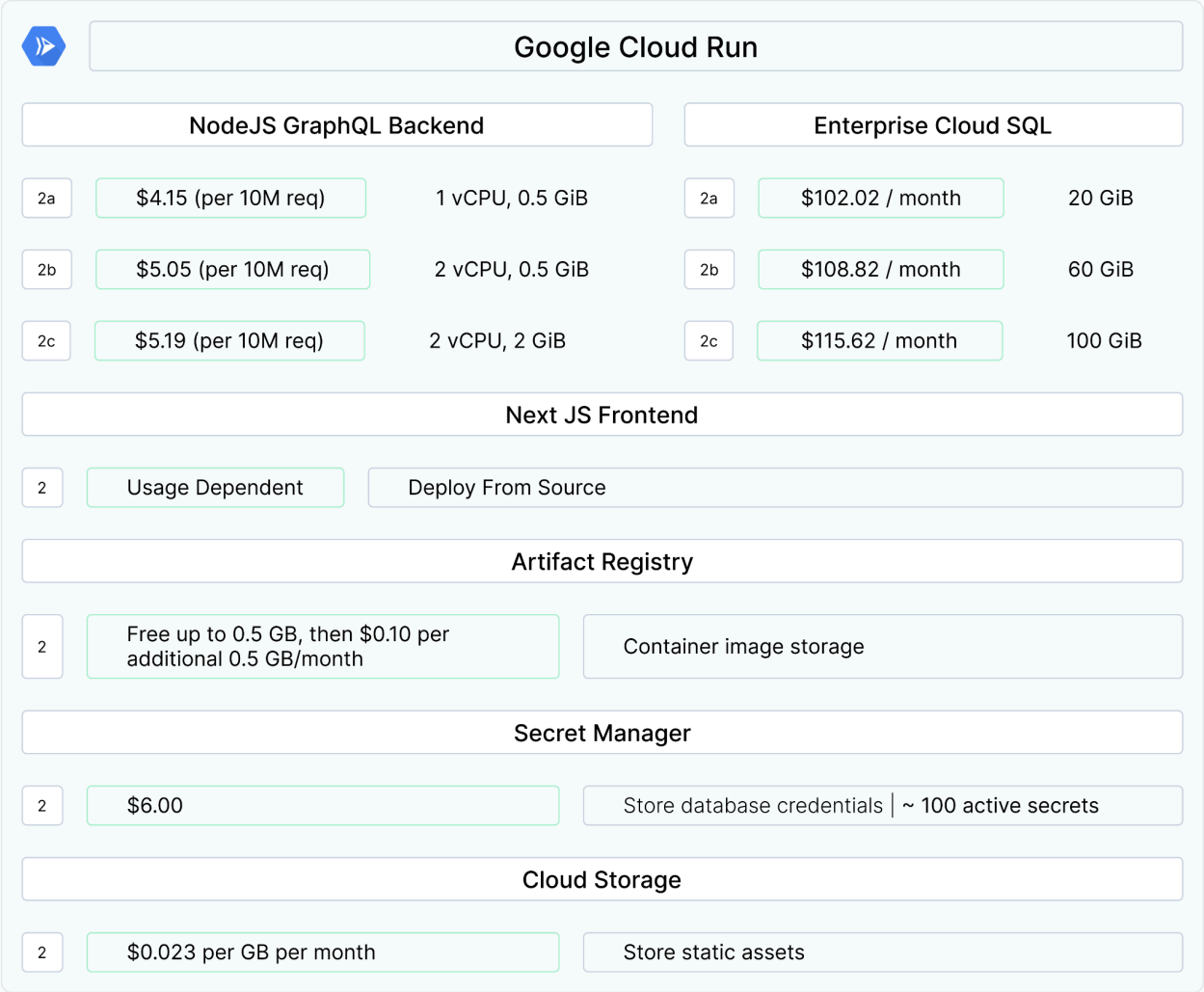
1. 2 Node Cluster (Cost-effective): Runs on 2 n1-standard-8 nodes with a 40 GiB Balanced Persistent Disk at $555 per month.
2. 2 Node Regional/Zonal Cluster: Priced at $628 per month, with capabilities to scale up dynamically with Cluster autoscaler and Node auto-provisioning. Cost calculations follow the formula: base price + (275(x)*y), where x is the number of additional nodes, and y is the fraction of the month they are operational.



GKE Pricing

2. Google Cloud Run

Cloud Run can be operated for under $200/month across various tiers. For the calculations, the `Enterprise Cloud SQL MySQL with one db-highmem-4` instance was excluded due to its alignment with more advanced enterprise operations rather than the current stage of Ingenius, which is focused on codebase development and user base size. All NodeJS GraphQL Backend run services are based on `Enterprise CloudSQL MYSQL w/ 1 db-standard-2` instance (cost friendly).



Cloud Run Pricing

Cloud Run Breakdown

NodeJS GraphQL Backend:

1. 1 vCPU, 0.5 GiBs per instance: $4.15 per 10 million requests.
2. 2 vCPU, 0.5 GiBs per instance: $5.05 per 10 million requests.
3. 2 vCPU, 2 GiBs per instance: $5.19 per 10 million requests.

Next JS Frontend: Deployed dynamically from source, scaling based on traffic.

Steps to Deploy:

1. Deploy backend and frontend on Cloud Run from source, which will automatically build container images.
2. Migrate SQLite database data to CloudSQL.
3. Connect CloudSQL instance to the Cloud Run backend.

3. Google App Engine

Must be used with either GKE or Cloud run. **We recommend F1 Instance**.

Standard Environment

- F1 Instance: 1 instance with 40 GiB: $1.15/month; 2 instances: $31.57/month.
- F2 Instance: 1 instance with 40 GiB: $31.57/month; 2 instances: $104.57/month.

Flexible Environment

Scales dynamically based on load: 1 vCPU, 2 GiB, 10-40 GiB persistent disk, 40 GiB data transfer: $48.76 per month; 2 vCPU, 2 GiB, 10-40 GiB persistent disk, 40 GiB data transfer: $87.56 per month.

Database

A schema diagram generated using DBVisualizer is included and can be found here.

Cache

Recommendations | We recommend Memcache for caching. We compared Memcache with Redis. Memcache is best for simple, high-speed caching needs where persistence and complex data types are not required. Redis offers more versatility with support for complex data structures, persistence, and a broader range of use cases, making it suitable for more advanced applications.

MemoryStore

- With 2 nodes: $249.49/month
- With 1 node: $124.98/month

Security Compliance

Image Security Options

Automatic Scanning | For enhanced security, automatic image scanning can be enabled in Artifact Registry at a higher cost. This feature automatically checks for vulnerabilities within stored images.

Alternative Scanning Solutions:

- CI/CD Integration: We recommend tools like Aqua Security's Trivy can be integrated directly into the CI/CD pipeline, allowing for continuous scanning of images during the development process.
- Docker Hub Vulnerability Scanner: Utilize Docker Hub's built-in scanner to assess images for vulnerabilities before deployment.

Access Control

Policy Configuration: We recommend Google Cloud's Policy Analyzer to ensure that both users and service accounts are configured with the principle of least privilege, minimizing the access rights of cloud resources to only what is necessary.

Logging and Monitoring

- Cloud Logging:
    - Cost Options:
        - 50 GiB per month: $1/month
        - 70 GiB per month: $11.40/month
        - 100 GiB per month: $27/month

Integration: Configure Cloud Logging to collect logs which can be analyzed using Cloud Monitoring dashboards, BigQuery for extensive querying, and Cloud Trace for performance insights.

Specific Logs

- Platform Logs: Logs from Google Kubernetes Engine (GKE) workloads.
- Cloud Audit Logs: Track and record all administrative activities across Google Cloud services, providing a comprehensive audit trail.

Security Testing

Pentesting Tools

Open Source Options | We recommend tools like Prowler and AquaSecurity CloudSploit for auditing and ensuring the security of your Google Cloud Platform (GCP) environments.

CI/CD

Options | Github Actions vs. Cloud Build

Recommendation | Cloud Build for integrating directly with Google Cloud services, providing efficient build and deployment pipelines with Google Cloud's infrastructure.

Cloud Build

Cost: $0.003 per build-minute

GitHub Actions

Free Tier: 2,000 build minutes per month and 500 MB of storage free for organizations.
Cost Beyond Free Tier: $0.008 per GB of storage, per minute of usage.

Note: Well-suited for projects already utilizing GitHub for version control, offering extensive integration with third-party tools and services.

Workflow Processes

Development to Lab Environment

Procedure | When a pull request (PR) is merged into the main repository, the changes are automatically deployed to the lab environment. This allows for testing and validation in a controlled setting before production deployment.

Lab to Production Promotion

Manual Trigger | A separate workflow that must be manually triggered is used to promote changes from the lab environment to production. This step ensures an additional layer of oversight and validation, enhancing security and stability in the production deployment.