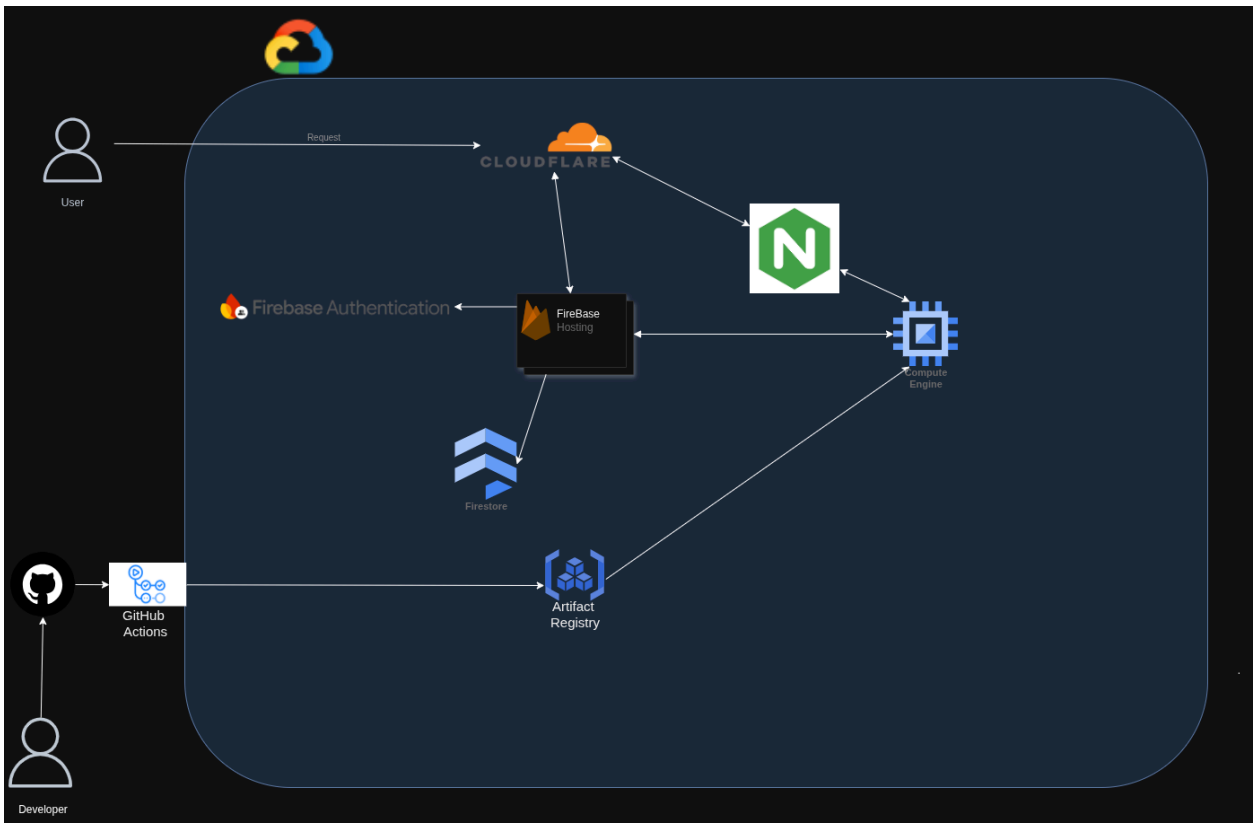
Sandbox Environment: Infrastructure & Deployment Overview

Project: ImagineHumans Platform

1. Current Architecture Overview

High-level Architecture

The platform uses a container-based deployment on Google Cloud with a CI/CD pipeline and a secure edge layer.



Main components involved

- GitHub Actions for CI/CD automation
- Google Artifact Registry for Docker images
- Google Compute Engine VM for hosting
- Nginx for reverse proxy and SSL termination
- Firebase services for authentication and data
- Cloudflare as a DNS, security, and CDN layer

Request Flow (User Side)

1. User requests the application via browser
2. User request first reaches **Cloudflare**
    - DNS resolution
    - DDoS protection
    - Basic security filtering
3. Cloudflare forwards traffic to the VM's public IP
4. Nginx on the VM:
    - Handles HTTPS (SSL)
    - Redirects HTTP to HTTPS
    - Routes traffic to the application container
5. Application communicates with:
    - Firebase Authentication
    - Firestore database

CI/CD Flow (Developer Side)

1. The developer pushes code to the `develop` branch
2. The GitHub Actions pipeline is triggered
3. Pipeline actions:
    - Builds a Docker image using Dockerfile
    - Pushes image to Google Artifact Registry
4. VM pulls the latest image from Artifact Registry
5. Docker Compose restarts the running containers

2. Cloudflare & Access Credentials

Cloudflare Usage

Cloudflare is used for:

- DNS management
- DDoS protection
- Hiding the VM public IP
- Improving performance via CDN
- Optional SSL edge handling

Credentials & Access

Cloudflare account credentials are **client-owned**.

What the client should have access to:

- Cloudflare dashboard login
- Domain DNS settings
- SSL/TLS mode configuration

Note:
I would provide support to set up a Cloudflare account if needed.

3. Current Hosting & Infrastructure Setup

Google Cloud Platform

Project

- Google Cloud Project in `australia-southeast1` region

Compute Engine VM

- Machine type: `e2-medium`
- Resources:
  - 2 vCPU
  - 4 GB RAM
- Network:
  - Default VPC
  - Firewall rules open:
    - HTTP (80)
    - HTTPS (443)
    - SSH (22)
- Static public IP attached

Service Account

The VM is attached to a dedicated service account:

Service account name

- `github-deployer`

Permissions

- Artifact Registry Reader
- Artifact Registry Writer
- Compute Engine related permissions (as required)

This allows the VM to securely pull images from Artifact Registry **without storing any credentials on disk**.

Artifact Registry

- Docker repository hosted in Google Artifact Registry
- Used to store application images built by CI/CD

Cleanup Policy

- Automatically keeps only the **latest 5 images**

- Older images are deleted automatically
- Prevents storage growth and cost issues

Nginx & SSL

- Nginx is installed directly on the VM
- Acts as:
    - Reverse proxy
    - SSL termination layer
- SSL certificates are issued via **Let's Encrypt**
- HTTP traffic is automatically redirected to HTTPS

Benefits

- Secure HTTPS traffic
- Central routing layer
- Easy future domain or routing changes

4. Application Deployment Model

Containerization

- Application is packaged as a Docker image
- Built using a Dockerfile
- No environment secrets are committed to GitHub

Environment Variables

- All sensitive keys (Firebase, Stripe, etc.) are injected via environment variables
- No service account JSON files are stored in the repository
- Secure and CI/CD-friendly setup

5. Security Considerations

- No secrets committed to GitHub
- VM authentication handled via service account
- Artifact Registry access controlled via IAM
- Cloudflare protects the public edge
- SSL enforced end-to-end

6. What the Client Needs to Know

- Deployment is fully automated via GitHub Actions
- No manual server access required for deployments
- Infrastructure is simple, cost-effective, and scalable
- Security best practices are followed
- Cleanup and maintenance policies are already in place

7. Summary

- Modern CI/CD pipeline implemented
- Secure Docker-based deployment on GCP
- Clean image lifecycle management
- SSL and traffic routing via Nginx
- Edge security and DNS via Cloudflare