Cloud Deployment Guide

Full Deployment Guide: Vercel (Frontend) + Google Cloud VM (FastAPI Backend)

Frontend Deployment on Vercel (Next.js 15 with App Router)

Deploy your frontend using the Vercel web interface or CLI.

Step-by-step using the **Vercel Dashboard (Web UI)**:

1. Visit https://vercel.com

2. Log in with your GitHub, GitLab, or Bitbucket account.

3. Click '+ Add New Project' to import your Next.js repository.

4. During configuration:

- Set the framework preset to: Next.js

- Build command: npm run build

- Output directory: .next

5. Click 'Deploy'. Vercel will build and host your site instantly.

6. Access the live site via the provided Vercel subdomain.

Optional **Command Line (CLI) Method**:

Install and deploy using the following commands:

```bash npm i -g vercel vercel login vercel --prod ```

Backend Deployment on Google Cloud (FastAPI on Compute Engine)

Step-by-step using **Google Cloud Web Console**:

1. Visit: https://console.cloud.google.com/

2. Create or select a Google Cloud project.

3. Go to Compute Engine → VM Instances → Click 'Create Instance'.
```

4. Use the following configuration:

- Name: fastapi-backend

- Region: Closest to your target audience

- Machine type: e2-standard-2 (2 vCPUs, 8 GB RAM)

- Boot Disk: Ubuntu 22.04 LTS, 120 GB SSD

- Firewall: Check 'Allow HTTP' and 'Allow HTTPS'

5. Click 'Create' to launch the instance.

Alternative **Google Cloud SDK (gcloud CLI) Method**:

Install SDK from https://cloud.google.com/sdk/docs/install

Then run:

```bash gcloud auth login gcloud config set project YOUR_PROJECT_ID gcloud compute instances create fastapi-backend \ --zone=us-central1-a \ --machine-type=e2-standard-2 \ --image-family=ubuntu-2204-lts \ --image-project=ubuntu-os-cloud \ --boot-disk-size=120GB \ --tags=http-server,https-server ```

Backend Setup on the Virtual Machine

SSH into your VM:

```bash gcloud compute ssh fastapi-backend ```

Update and install required packages:

```bash sudo apt update && sudo apt upgrade -y sudo apt install python3.11 python3.11-venv python3-pip -y sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.11 1 ```

Clone your FastAPI repo and install dependencies:

```bash git clone https://github.com/your-username/your-fastapi-repo.git cd your-fastapi-repo python3 -m venv venv source venv/bin/activate pip install -r requirements.txt ```

Run the FastAPI server using Uvicorn:

```bash uvicorn main:app --host 0.0.0.0 --port 8000 --reload ```

To run it in the background:
```

```bash nohup uvicorn main:app --host 0.0.0.0 --port 8000 & ```

Open Port 8000 in Firewall

In the Google Cloud Console, go to VPC Network > Firewall Rules and create a rule:

- Name: allow-fastapi

- Protocols/Ports: tcp:8000

- Source IP Ranges: 0.0.0.0/0

Optional Enhancements: NGINX, Domain, SSL

```bash sudo apt install nginx -y ```

Use NGINX as a reverse proxy to Uvicorn. Point your domain to the VM’s external IP.

Install SSL using Certbot:

```bash sudo apt install certbot python3-certbot-nginx -y sudo certbot --nginx -d yourdomain.com ```

Deployment Summary

Frontend: https://yourproject.vercel.app

Backend: http://[VM-IP]:8000

Professional Tips

• Use systemd or PM2 to keep your backend running.

• Always use environment variables (.env) for secrets.

• For databases, consider Cloud SQL or managed MongoDB.
```