

Objective: Design a scalable and secure REST API solution for cloud deployment (GCP or Azure), implement a basic version of the API that runs locally, and design its infrastructure. The focus is on your architectural design, Infrastructure as Code quality, and CI/CD pipeline design, rather than live cloud deployment.

Scenario: Simple Note-Taking API

Your task is to design and partially implement a basic "Note-Taking API" for a new internal service. This API will allow users to create, retrieve, update, and delete simple text notes.

Core Requirements:

1. Application (API):

- Implement a simple REST API using Python.
- It should expose at least the following endpoints:
 - **POST /notes:** Create a new note.
 - **GET /notes:** Retrieve all notes.
 - **GET /notes/{id}:** Retrieve a specific note by ID.
 - **PUT /notes/{id}:** Update an existing note.
 - **DELETE /notes/{id}:** Delete a note.
- **Persistence:** Notes should be stored in a **PostgreSQL database**.
- **Containerization:** The API application must be containerized using Docker.
- **Local Execution:** The API and its database must be runnable locally using **Docker Compose**.

2. Cloud Infra Design

Draw an architectural diagram (for the app above) highlighting the services you can use for example on GCP or Azure, including Networking, IAM, Secrets, etc.

3. DevOps - CI/CD Pipeline Design

How would you ensure that the best practices around deployment of infra and code are followed? How would you build, push and deploy the image? Think of executing unit tests, code linters, terraform deployment, packaging, etc. Show your creativity here.

Deliverables:

Please provide a link to a **public GitHub repository** containing:

1. **Application Code:** The source code for your Note-Taking API.
2. **Docker Compose File:** A **docker-compose.yml** file to run the API and its database locally.
3. **README.md:** It should include:
 - **Architecture Diagram:** A high-level diagram illustrating your *proposed* cloud solution.
 - **Local Setup Instructions:** Clear, step-by-step instructions on how to clone the repository and run your application locally using Docker Compose.
 - **API Usage:** Instructions on how to interact with your *locally running* API (e.g., example **curl** commands).
 - **Design Choices & Justifications (Cloud Focus):**

- Why you chose a specific cloud compute service (Cloud Run vs. GKE vs. others) for cloud deployment.
- Why you chose a specific CI/CD tool.
- Detailed explanation of your security considerations (IAM, secrets management, network segmentation) for cloud deployment.
- Any significant trade-offs or alternative approaches you considered (e.g., cost, complexity, scalability, security) for the cloud architecture.