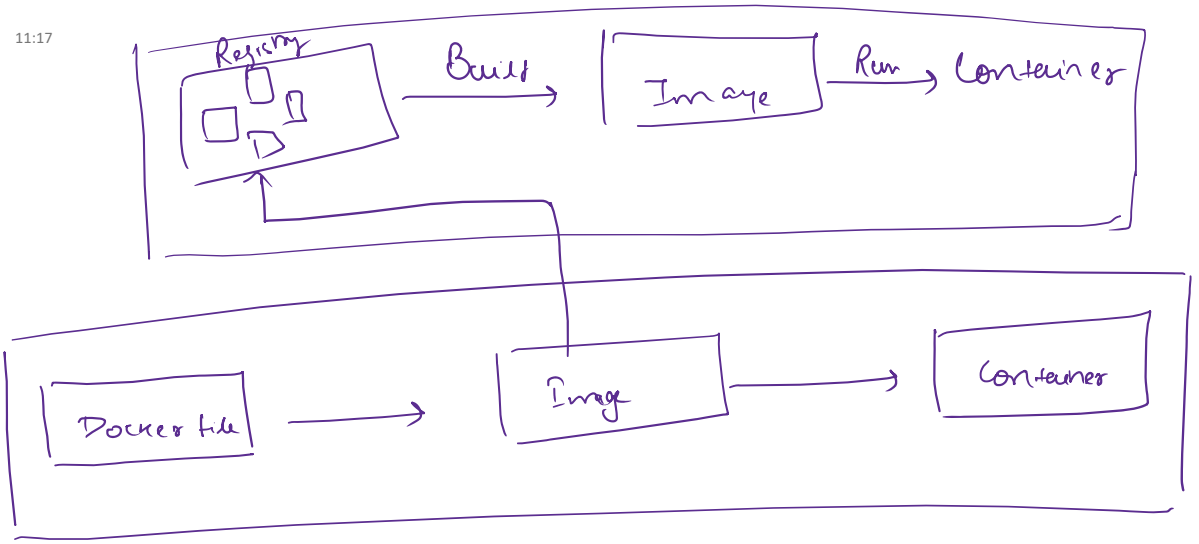
15 February 2026 11:17




How do we run our Image in a 3rd party environment.

eg:- Cloud, Vertex AI - - -

1. Hugging face Spaces
2. GCP → free acount 27000 Rs. credits
3. Compute
4. Cloud run. → Dessert
5. Vertex AI
6. Microservices Architecture.
7. DMR, Docker MCP

I want to share but I want you guy to use

I want to share but I want you guy to use your own .env file

```
docker run -p 8000:8000 --env-file .env my-genai-app
```

Upload to Docker hub.

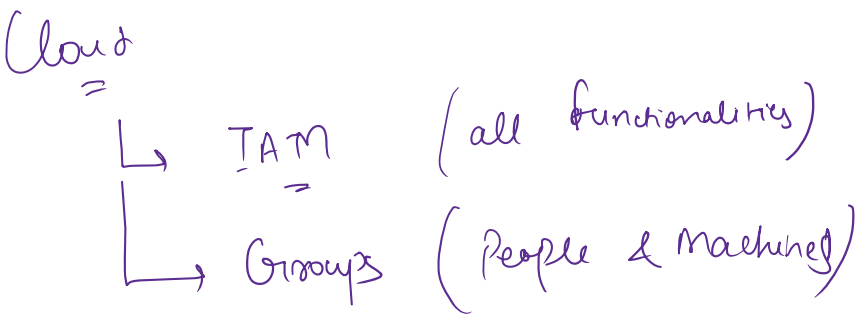
```
docker tag my-genai-app <their-username>/my-genai-app:v1

docker login
docker push <their-username>/my-genai-app:v1
```

Run on your local

```
docker run -p 8000:8000 --env-file .env sameerz567/my-genai-app:v1
```

```
docker run --platform=linux/amd64 -p 8000:8000 sameerz567/my-genai-app:v1
```




Creating a compute

This is **Module 2: The "Manual Labor"** phase.

We are going to rent a raw Linux server, SSH into it, install Docker manually, and run your app.

This teaches students: *"The Cloud is just a computer in a warehouse that you configure yourself."*

**Step 1: Rent the Computer (Create VM)**

1. Go to **Google Cloud Console**.

2. Search for **"Compute Engine"** -> **"VM Instances"**.
3. Click **"Create Instance"**.
4. **Configure it (The Cheapest Option):**
    - **Name:** class-demo-server
    - **Region:** us-central1 (or your local region).
    - **Machine type: e2-micro** (This is often Free Tier eligible).
5. 🔥 **CRITICAL STEP (The Firewall):**
    - Scroll down to the **Firewall** section.
    - **Check the box: [x] Allow HTTP traffic.**
    - *(If you forget this, the website will never load).*
6. Click **Create**.

**Step 2: Walk into the Server (SSH)**

Wait about 30 seconds for the green checkmark ☑.

1. Find your VM in the list.
2. Click the **"SSH"** button.
3. A black window will pop up. **You are now inside the remote computer.**

**Step 3: Install Docker (The "Manual Labor")**

Since this is a raw computer, it doesn't have Docker yet. You have to be the SysAdmin. Run these commands in the SSH window:

Bash

```
# 1. Update the "App Store"
sudo apt-get update

# 2. Install Docker
sudo apt-get install -y docker.io

# 3. Check if it's alive
sudo docker --version
```

**Step 4: The Secret Sauce (Creating the .env file)**

Your image on Docker Hub is "brainless" (secure). It needs your API keys to work. We will create a .env file right here on the server.

1. Type this to open a text editor:

Bash

```
nano .env
```

2. Paste your keys in (Right-click to paste in the browser terminal):

Ini, TOML

```
GOOGLE_API_KEY=AIzaSyYourRealKeyHere...
# Add any other variables you need
```

3. **Save and Exit:**
    - Press Ctrl + O then Enter (to Save).
    - Press Ctrl + X (to Exit).

**Step 5: Run the App**

Now for the magic command. We will map Port 80 (the default web port) to Port 8000.

**Replace YOUR_USERNAME with your actual Docker Hub username.**

Bash

```
sudo docker run -d --name my-api --env-file .env -p 80:8000 sameerz567/my-genai-app:v1
```

- **-d**: Detached mode (runs in background so it doesn't die when you close the window).

**-p 80:8000**: Traffic hits the server on Port 80 -> Docker sends it to Port 8000.

docker.io/sameerz567/my-genai-app:v1

## Why these changes are necessary (Class Notes):

- **The "User" Problem:** Hugging Face runs in a shared environment. For security, they won't let your app run with "Admin" (Root) privileges. The useradd lines create a safe "sandbox" user.
- **The Port Change:** Unlike Cloud Run (which is flexible), Hugging Face Spaces are hard-wired to look for traffic on **7860**. If you use 8000, the build will succeed, but the app will show a "Lobby" or "Connection Error" screen.
- **The Path:** Notice we changed the working directory to /home/user/app. This ensures your app has permission to write temporary files if needed.