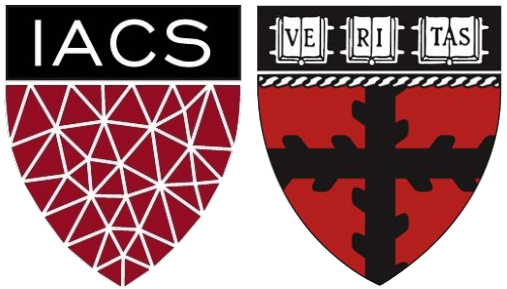
Lecture 20: Deployment

Advanced Practical Data Science, MLOps

AC215

Pavlos Protopapas
Institute for Applied Computational Science, Harvard



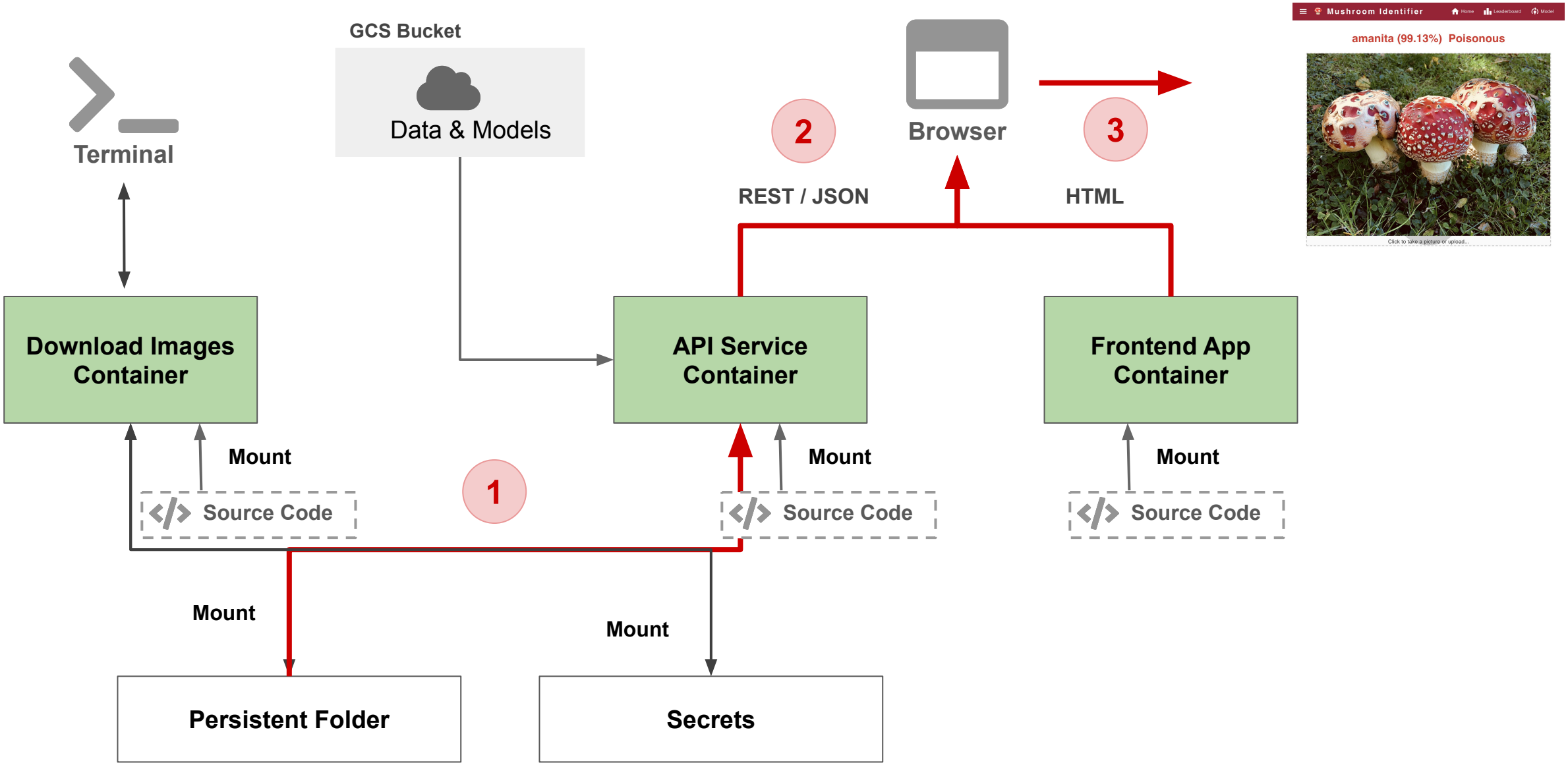
Outline

1. Recap
2. Deployment

Outline

1. **Recap**
2. Deployment

Recap: APIs & Frontend App



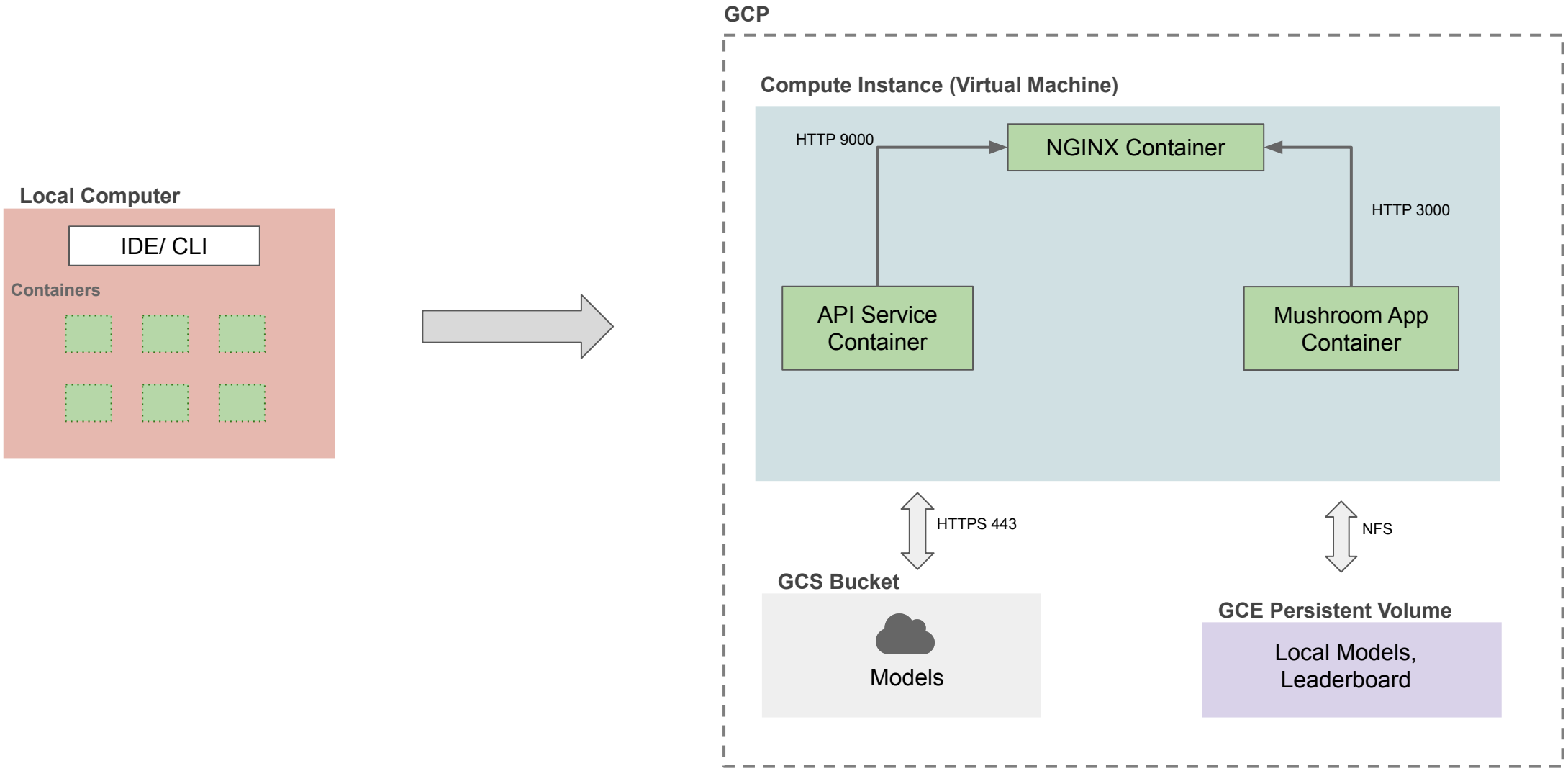
Recap: APIs & Frontend App

- Everything we built is on our **local computer**
- We need to **deploy** this to a server so out users can access

Outline

1. Recap
2. **Deployment**

Deployment: Goal



Deployment Steps (Manual)

1. Build and push docker images to Docker Hub
2. Create Compute Instance (VM) in GCP
3. Provision the server (Installed required softwares)
4. Setup Docker containers in VM Instance
5. Setup a web server to expose our app to the outside world

Tutorial: Deployment to GCP (Manual)

Mushroom App - Deployment to GCP (Manual)

Deployment Automation

In our manual deployment there were various steps to keep track of. We want to automate this?

Ansible

Ansible

- Is a tool for infrastructure automation
- Think of infrastructure as code
- Ansible scripts (playbooks) consist of instructions for tasks like
  - Server & Cluster creation/deletion
  - Software installation & setup
  - Networking setup
- Everything is code, so you can check it into GitHub and share

Deployment Steps (Ansible / Automation)

1. Setup local container to connect to GCP
2. Build and push docker images to GCR
3. Create Compute Instance (VM) in GCP
4. Provision the server (Installed required softwares)
5. Setup Docker containers in VM Instance
6. Setup a web server to expose our app to the outside world

Setup local container /GCP

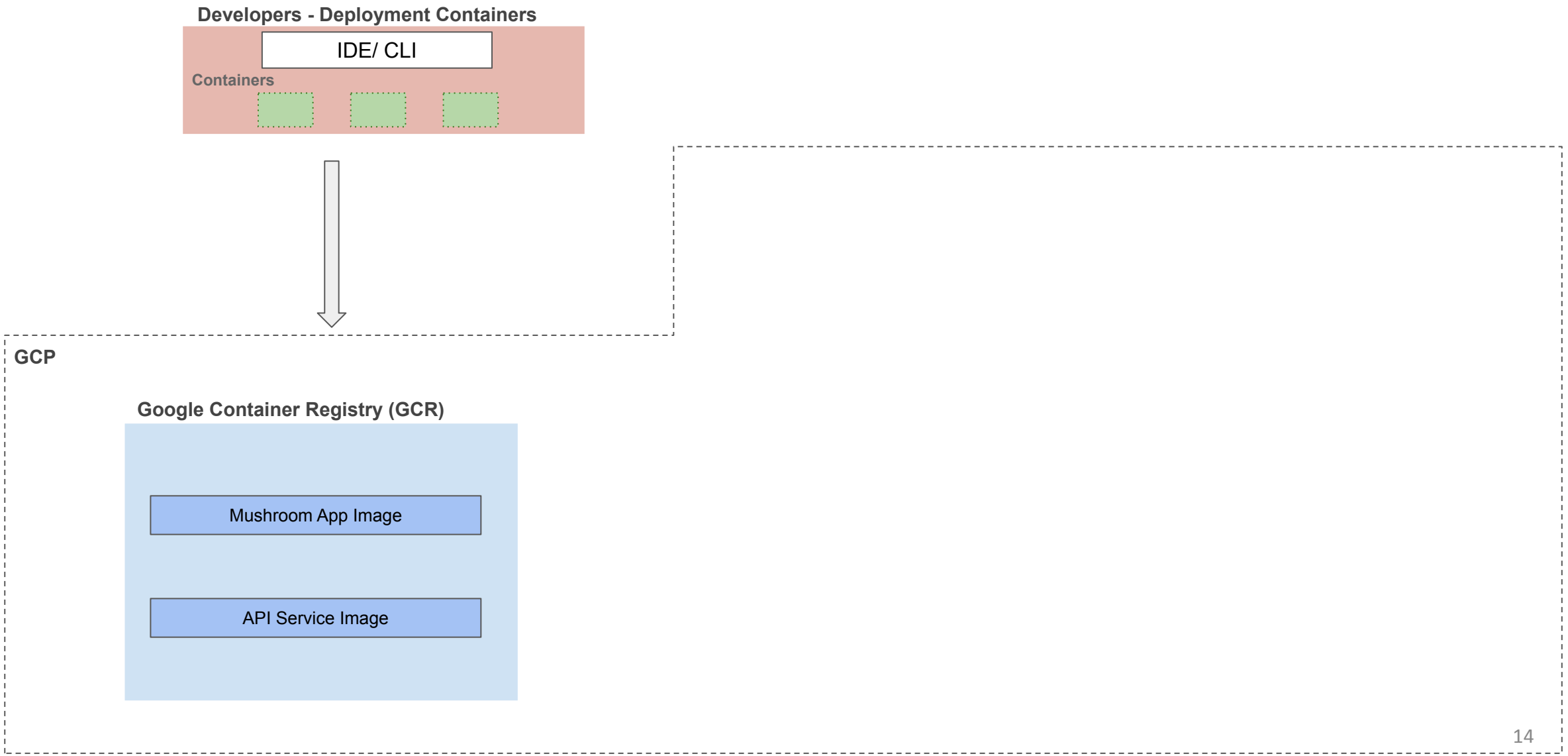
Setup required GCP

- Enable APIs
- Create service accounts
  - **deployment** (To deploy everything to GCP)
  - **gcp-service** (To read containers from GCR in VM)

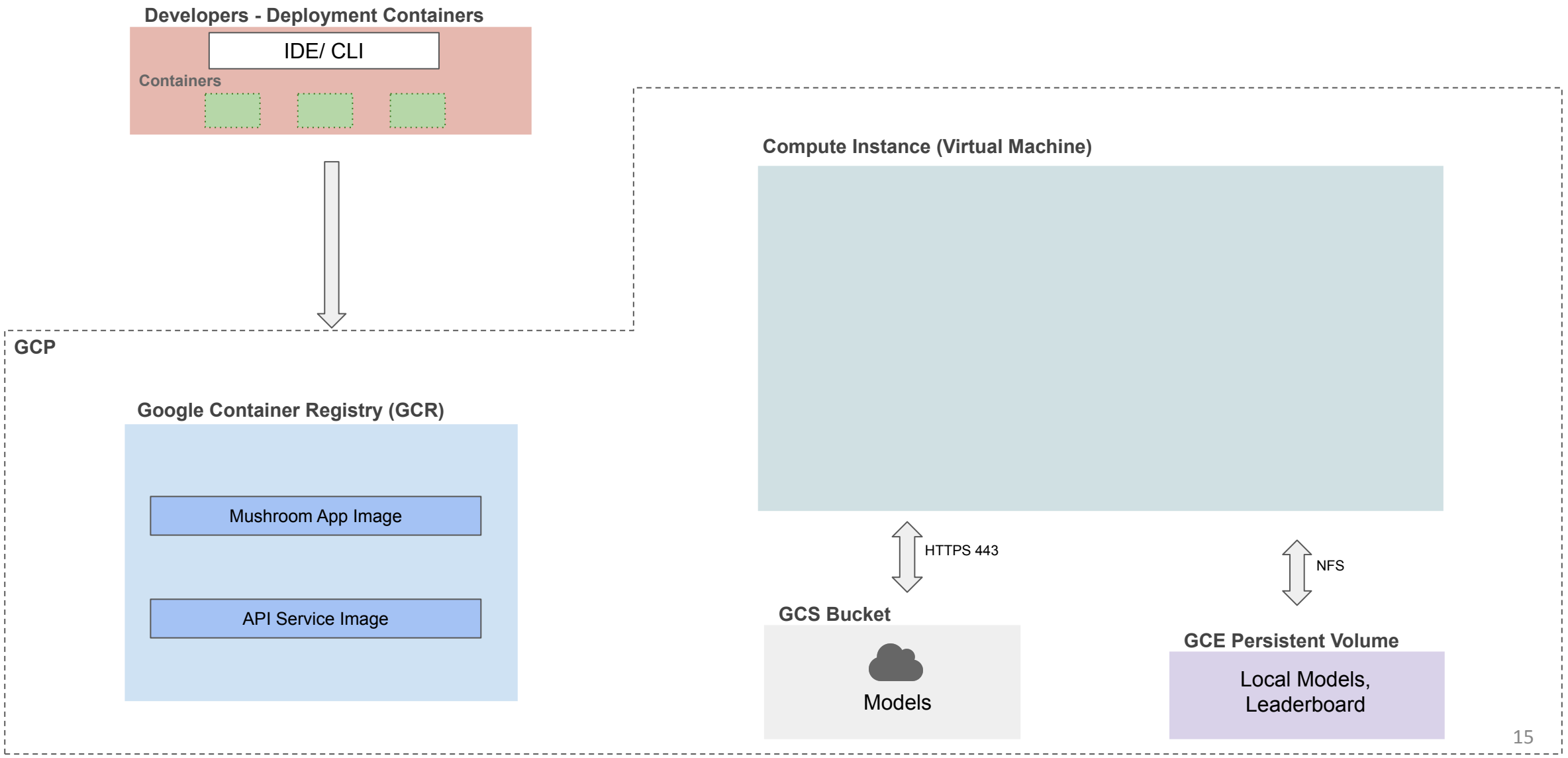
Setup local deployment container

- Add secret keys
- Set GCP project we want to connect to

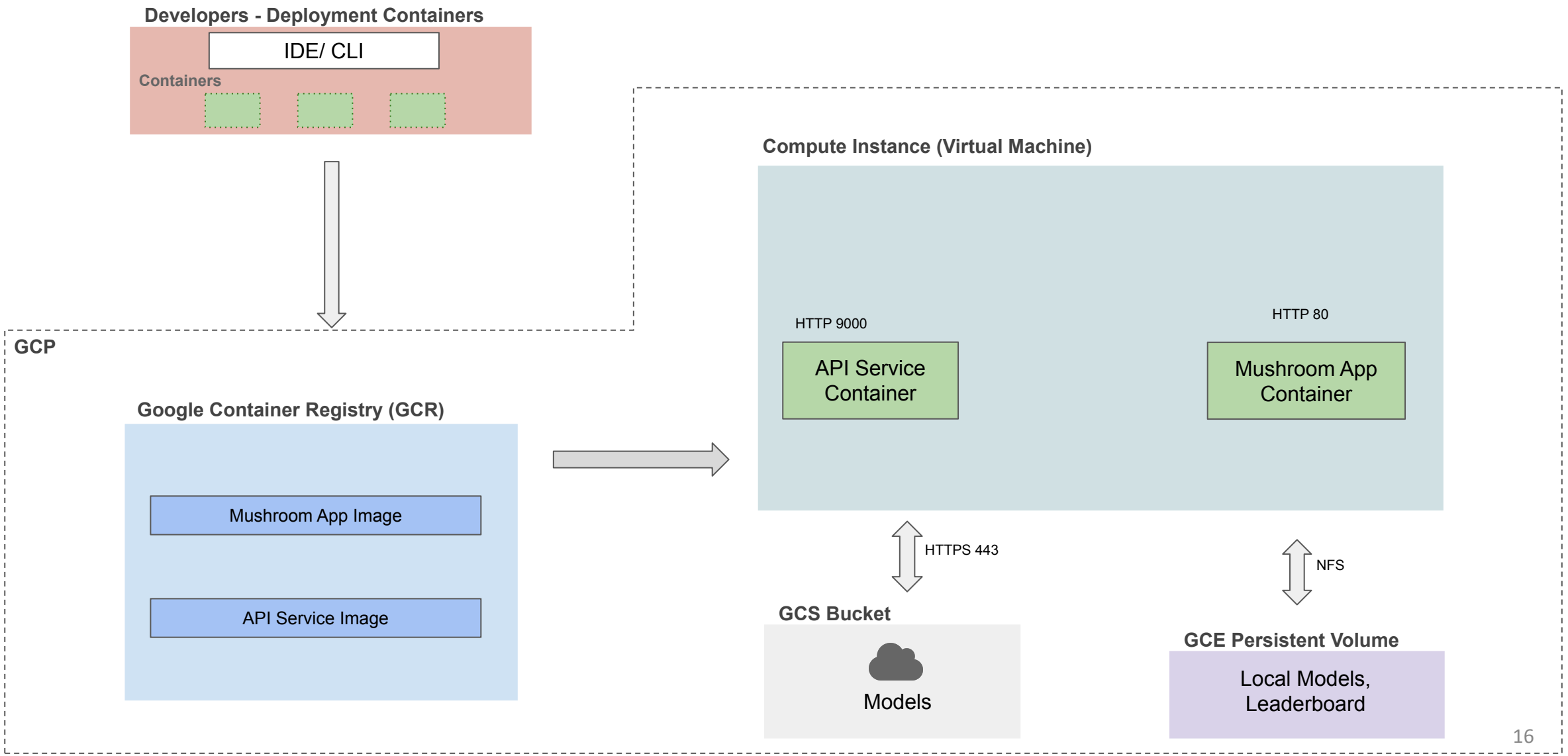
Build & Push Docker Images to GCR



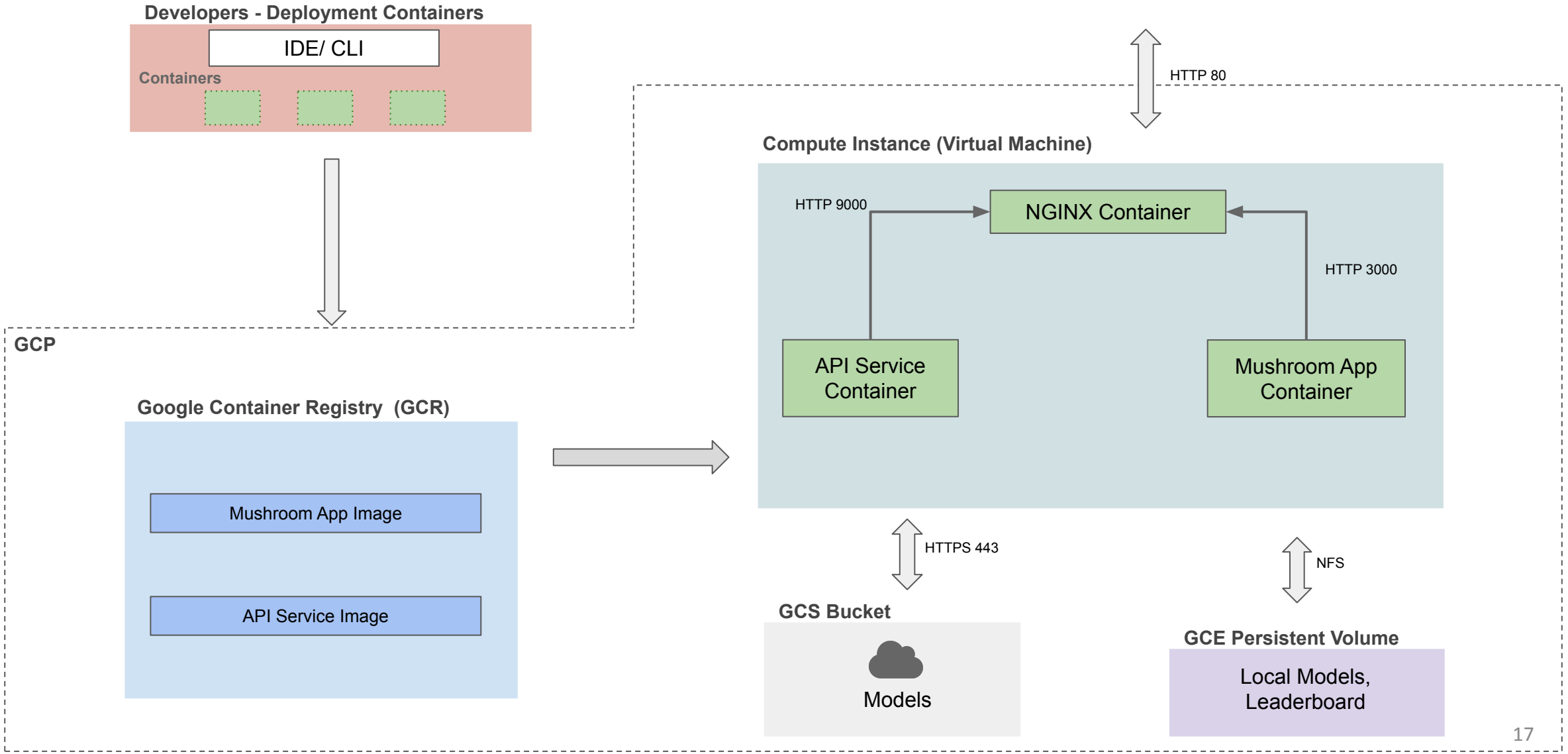
Create Compute Instance (VM)



Setup Docker Containers in VM



Setup Web Server to expose App

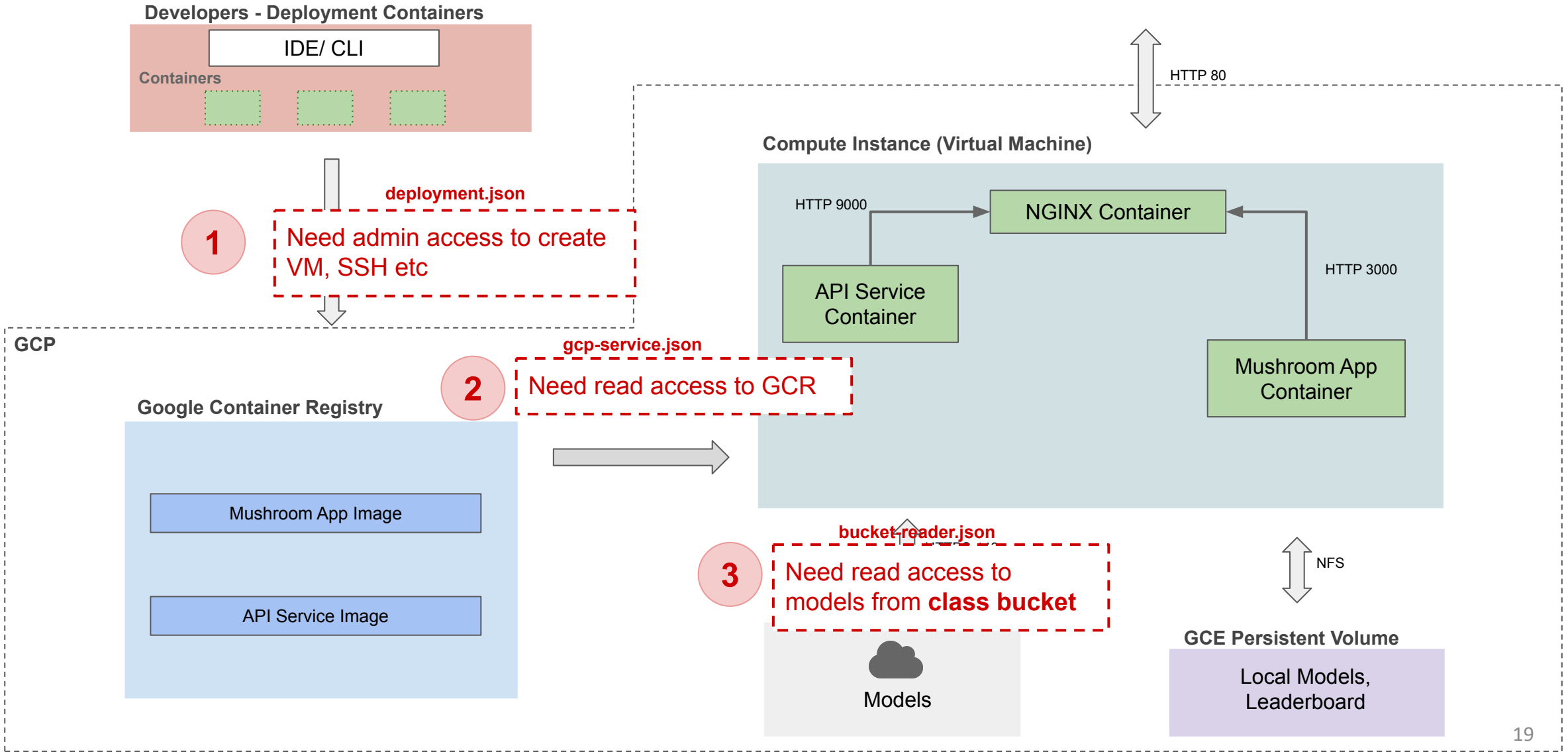


Why did we need 2+1 service accounts?

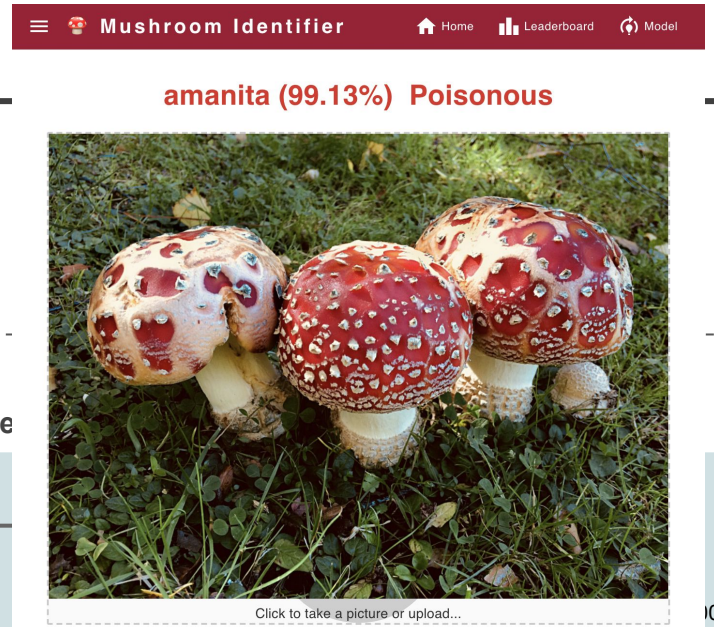
Why do we need the following service accounts?

- **bucket-reader**
  - Has access only to the **class** GCP project where models are stores
- **deployment**
  - Has admin access to **your** group GCP project
- **gcp-service**
  - Has read access to **your** group GCP projects GCR

Why did we need 2+1 service accounts?



Test the App



Developers - Deployment Containers

IDE/ CLI

Containers

GCP

Google Container Registry

Mushroom App Image

API Service Image

Compute Instance

HTTP 9000

API Service Container

Mushroom App Container

HTTPS 443

NFS

GCS Bucket

Models

GCE Persistent Volume

Local Models, Leaderboard

Tutorial: Deployment to GCP (Ansible)

Mushroom App - Deployment to GCP (Ansible)

THANK YOU