# **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# **\*\*\*\* Assignment submission by Arjun Shrivatsan**

# **\*\*\*\* EAI 6010 - Assignment No: Module 5 - Face Mask Detection Microservice**

# **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## **About**

This microservice detects whether a person is wearing a face mask using a trained deep learning model based on **InceptionV3** architecture. It was developed as part of **EAI 6010 - Applications of AI** course.

The API accepts an uploaded image and returns whether the subject is **Wearing a Mask**, **Not Wearing a Mask**, or **Wearing a Mask Incorrectly** along with confidence scores.

## **What does this assignment do?**

* Trains a deep learning model using annotated XML image data
* Builds a REST API using FastAPI
* Allows real-time prediction of face mask usage from uploaded images
* Supports deployment via Docker and Render

## **How to Run the App**

### **Run via Swagger (Render Deployment)**

👉 URL: https://facial-mask-detection.onrender.com/docs

1. Go to the Swagger UI URL above
2. Click /predict → "Try it out"
3. Upload an image (JPG/PNG) → Execute
4. View Prediction and Confidence

### **Run Locally via Docker**

git clone https://github.com/arcsphere/facial-mask-detection.git

cd facial-mask-detection

# Build Docker image

docker build -t face-mask-api .

# Run the container

docker run -p 8000:8000 face-mask-api

Access API at: http://localhost:8000/docs

### **Run via cURL**

curl -X 'POST' \

'http://localhost:8000/predict' \

-F 'file=@your-image.jpg' \

-H 'accept: application/json' \

-H 'Content-Type: multipart/form-data'

## **⚙️ Installation Instructions (Local)**

# Clone repo

git clone https://github.com/arcsphere/facial-mask-detection.git

cd facial-mask-detection

# Create a virtual environment

python -m venv maskenv

source maskenv/bin/activate

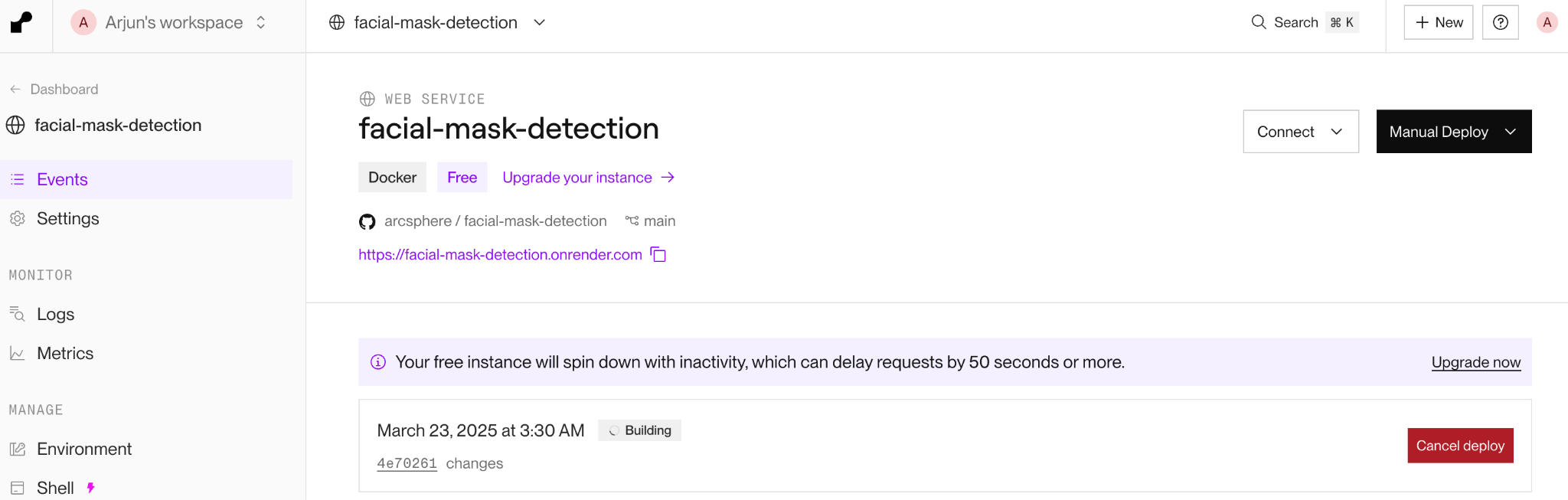
# Install dependencies

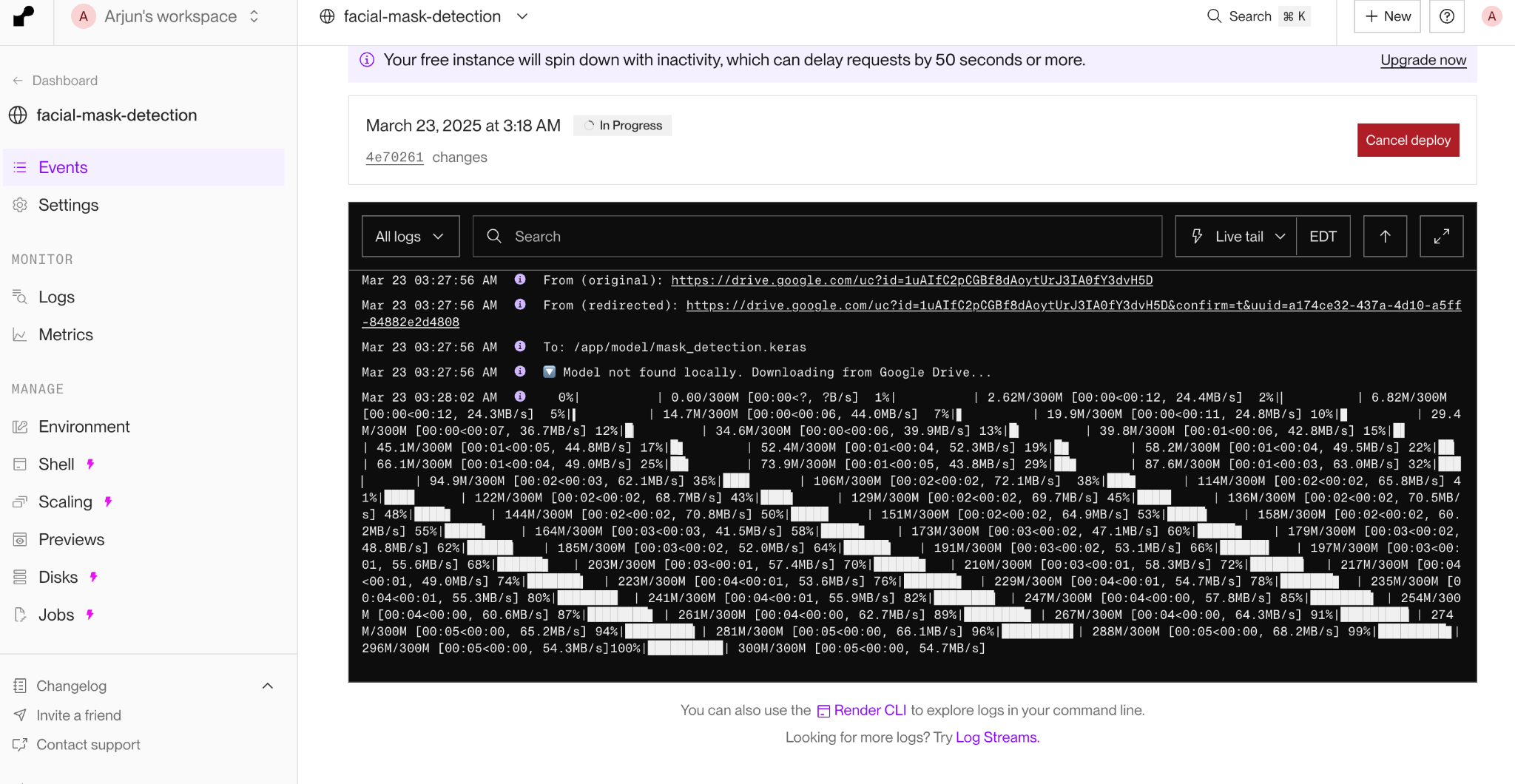
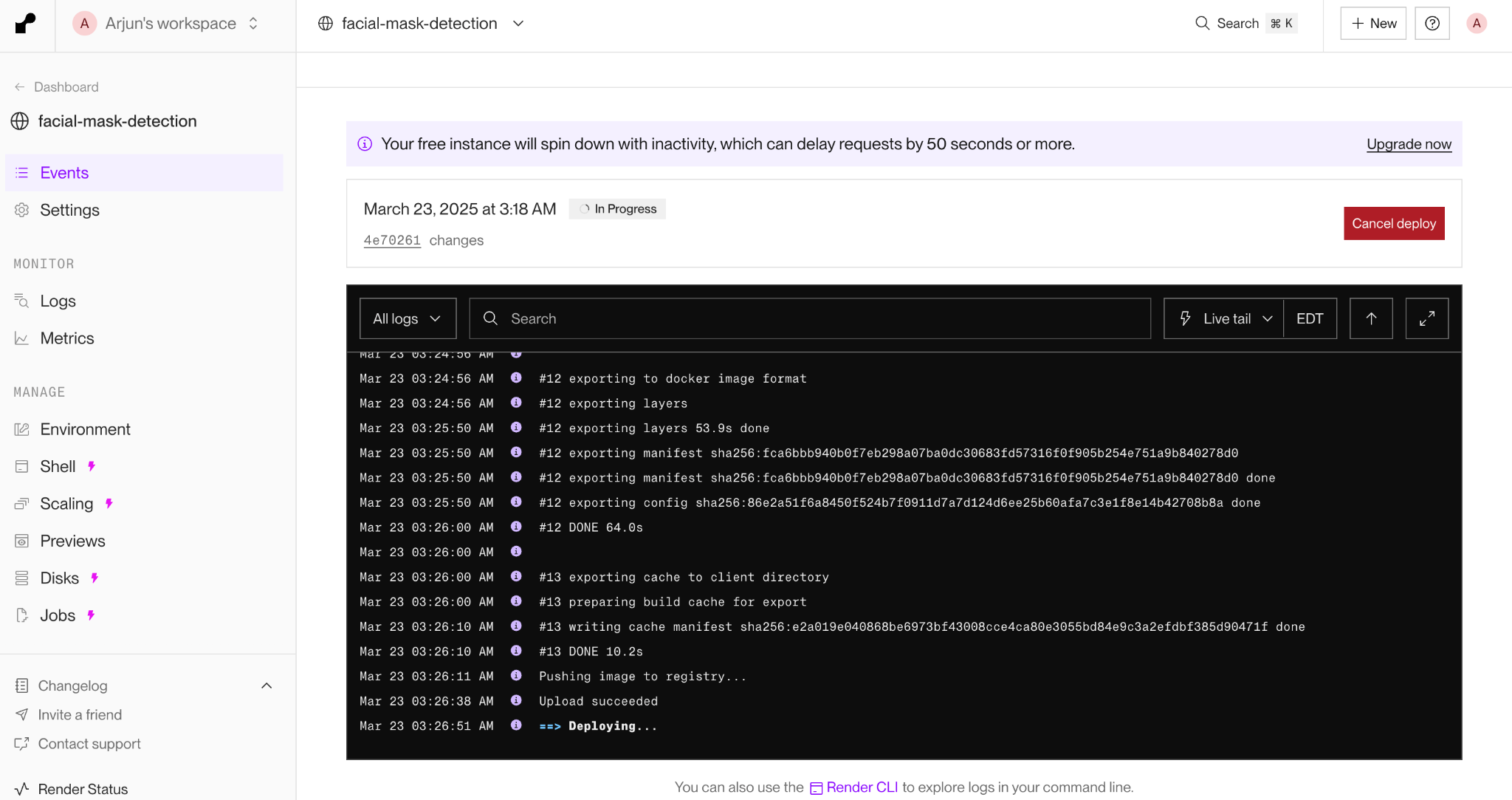
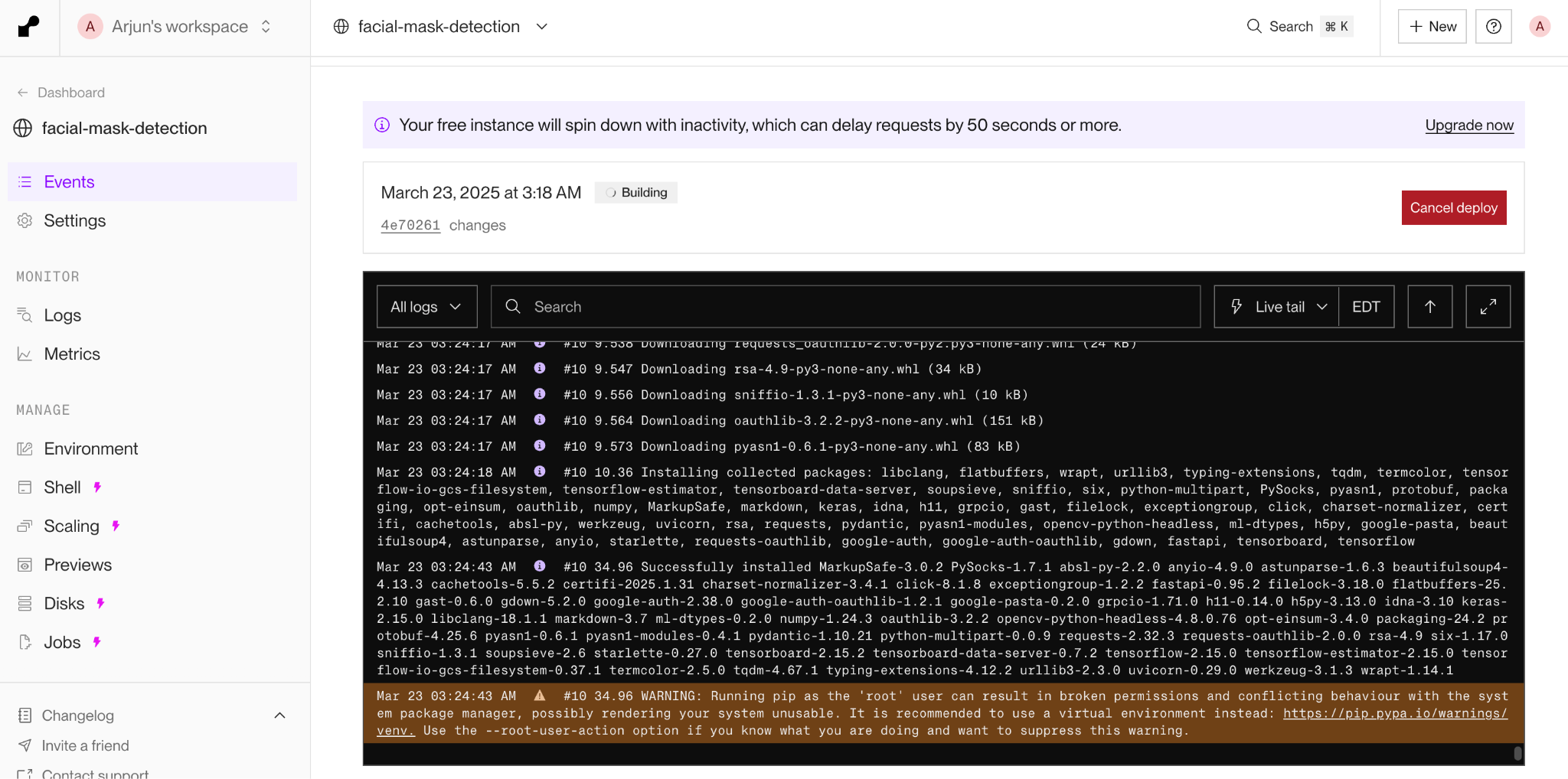
pip install -r requirements.txt

# Run FastAPI locally

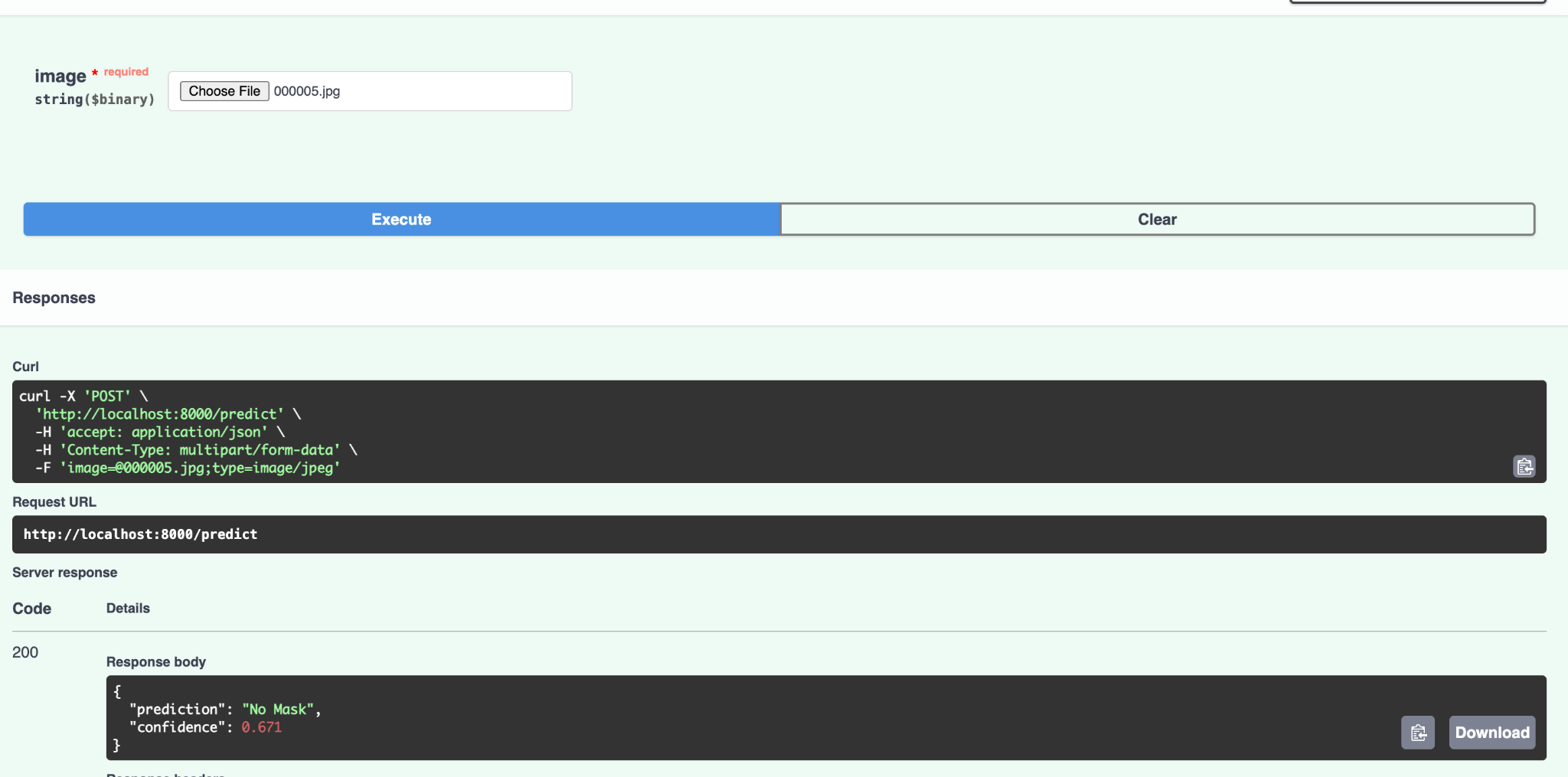
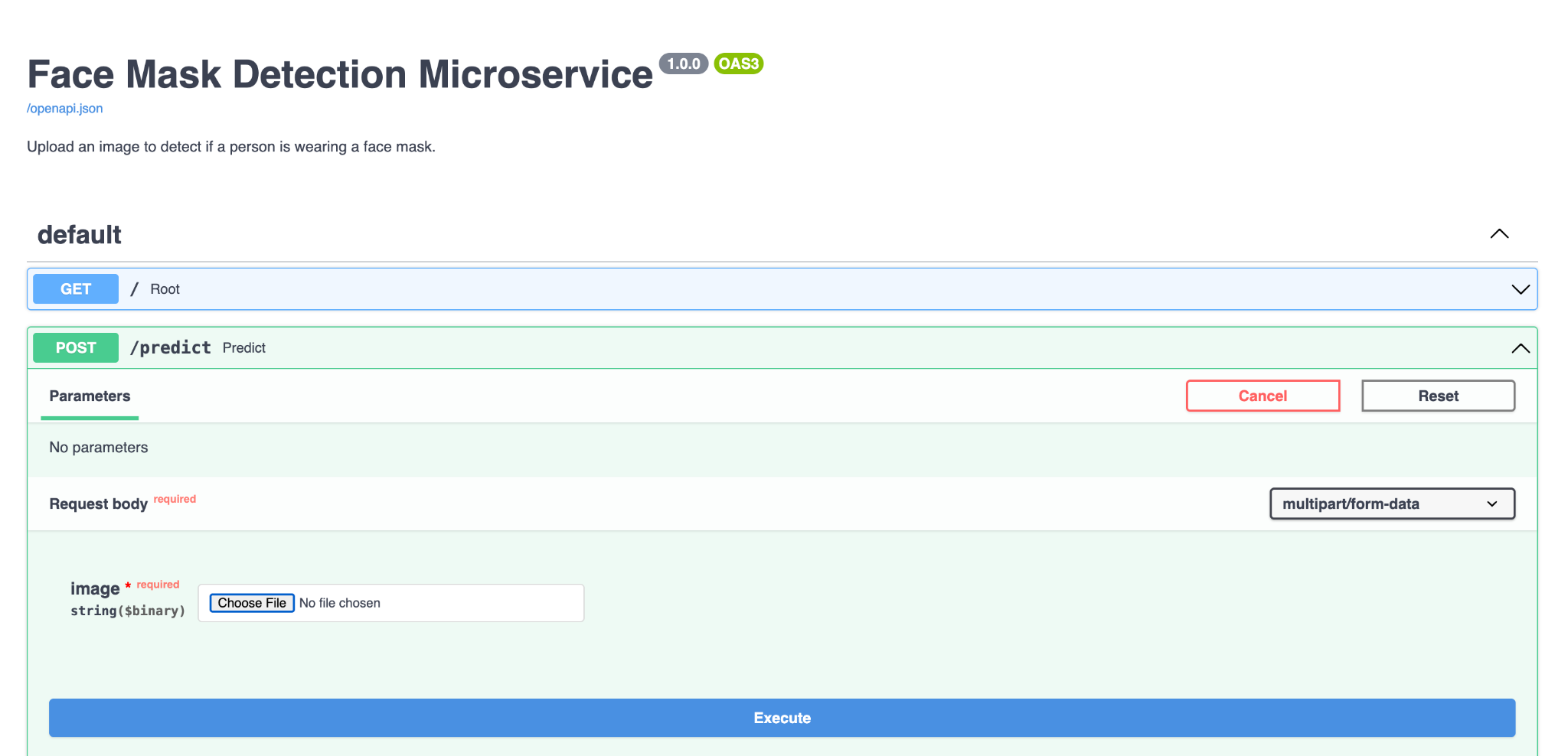
uvicorn app.app:app --reload

## **Deployment at onrender.com**





## **Usage**



## 

## **Tech Stack**

* **TensorFlow 2.15 / Keras** – Model Training
* **InceptionV3** – Base CNN architecture
* **OpenCV** – Image processing
* **FastAPI** – API development
* **Docker** – Containerization
* **Render** – Cloud deployment
* **Swagger / OpenAPI** – API docs

## **Directory Structure**

.

├── app/

│ └── app.py

├── model/

│ └── mask\_detection.keras (downloaded via gdown in utils)

├── utils.py

├── requirements.txt

├── Dockerfile

└── README.md

## **Resources**

* FastAPI Docs
* Keras Applications
* TensorFlow InceptionV3
* [Google Drive File Download using gdown](https://pypi.org/project/gdown/)

Model Download Link is included in utils.py to automatically fetch the model from Google Drive if it’s not found locally.

## **Final Notes**

* All local runs, Docker runs, and Render deployment are functional.
* Model file is **excluded from GitHub repo** due to size.
* Render fetches model at runtime from Drive using gdown.
* Swagger provides easy-to-use UI for testing and demonstrating the microservice.

**Submitted by:** Arjun Shrivatsan (002028814)  
**Course:** EAI 6010 – Applications of AI  
**Module:** 5 – Face Mask Detection