# Disclaimer: This output contains AI-generated content; user is advised to review it before consumption.  
  
#\*Start of AI Generated Content\*

# **Docker Deployment for Synthetic Data Generator API**

## **Step-by-Step Guide for Containerizing and Deploying the Flask API**

### **Prerequisites**

* Docker Engine installed on the deployment machine (version 20.10.0+)
* Docker Compose installed on the deployment machine (version 2.2.3+)
* Python 3.9+ and pip3 for dependency management (for local testing, if required)
* Synthetic Data Generator API codebase (provided above)

### **Step 1: Create a Dockerfile for the API**

* **File:** Dockerfile
* **Location:** Root directory of the Synthetic Data Generator API codebase
* **Contents:**

<font size="2">  
FROM python:3.9-slim  
  
# Set working directory to /app  
WORKDIR /app  
  
# Copy requirements file  
COPY requirements.txt .  
  
# Install dependencies  
RUN pip3 install --no-cache-dir -r requirements.txt  
  
# Copy application code  
COPY . .  
  
# Expose port for the API  
EXPOSE 5000  
  
# Command to run the API  
CMD ["flask", "run", "--host=0.0.0.0", "--port=5000"]  
</font>

### **Step 2: Define Dependencies in requirements.txt**

* **File:** requirements.txt
* **Location:** Root directory of the Synthetic Data Generator API codebase
* **Contents:**

<font size="2">  
Flask==2.0.3  
pandas==1.4.2  
scikit-learn==1.0.2  
numpy==1.22.3  
hashlib==0.0.1  
hmac==0.0.1  
base64==1.0.1  
</font>

### **Step 3: Create a Docker Compose File for Orchestration**

* **File:** docker-compose.yml
* **Location:** Root directory of the Synthetic Data Generator API codebase
* **Contents:**

<font size="2">  
version: '3.8'  
services:  
 api:  
 build: .  
 ports:  
 - "5000:5000"  
 restart: always  
 environment:  
 - FLASK\_ENV=production  
 - APP\_NAME=Synthetic Data Generator  
 - APP\_VERSION=1.0.0  
</font>

### **Step 4: Build and Start the Docker Container**

* **Command:** docker-compose up -d --build
* **Execution Directory:** Root directory of the Synthetic Data Generator API codebase
* **Expected Outcome:** Docker container builds and starts in detached mode, exposing the API on port 5000.

### **Step 5: Verify API Deployment**

* **Method:** Use a tool like curl or a REST client (e.g., Postman) to send a test request to http://localhost:5000/train\_ml\_model with a sample JSON payload.
* **Expected Outcome:** A successful response from the API, indicating the ML model training metadata.

### **Step 6: Secure and Monitor the Deployment (Optional)**

* **Recommendations:**
* Implement HTTPS using SSL/TLS certificates.
* Configure logging and monitoring tools (e.g., Prometheus, Grafana, ELK Stack).
* Ensure regular security updates and dependency patches.
* Consider deploying behind a reverse proxy (e.g., NGINX, Apache) for added security and load balancing.

#\*End of AI Generated Content\*