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

# **Docker Deployment for Anomaly Detection and Visualization Application**

## **Step-by-Step Guide for Containerizing the Python Application**

### **Prerequisites**

* Docker Engine installed on the deployment machine
* Docker Hub account (for pushing images)
* Python application code (provided above)

### **Step 1: Create a Dockerfile**

* **File Location:** Root directory of the Python application code
* **File Contents:**

<font size="14">  
# Use an official Python runtime as a parent image  
FROM python:3.9-slim  
  
# Set the working directory in the container  
WORKDIR /app  
  
# Copy the current directory contents into the container at /app  
COPY . /app  
  
# Install any needed packages specified in requirements.txt  
RUN pip install --trusted-host pypi.org -r requirements.txt  
  
# Make port 80 available to the world outside this container  
EXPOSE 80  
  
# Define environment variable  
ENV API\_KEY YOUR\_API\_KEY\_HERE  
ENV API\_URL https://example-api.com/data-integrity  
  
# Run app.py when the container launches  
CMD ["python", "main.py"]  
</font>

* **Note:**
* Replace YOUR\_API\_KEY\_HERE with your actual API key.
* Ensure requirements.txt is present in the root directory with required dependencies (e.g., pandas, matplotlib, scikit-learn, requests).

### **Step 2: Create a requirements.txt File**

* **File Location:** Root directory of the Python application code
* **File Contents:**

<font size="14">  
pandas  
matplotlib  
scikit-learn  
requests  
</font>

* **Note:** Add any additional dependencies required by your application.

### **Step 3: Build the Docker Image**

* **Command:**

<font size="14">  
docker build -t anomaly-detection-app .  
</font>

* **Note:** Run this command from the root directory of your application code.

### **Step 4: Run the Docker Container**

* **Command:**

<font size="14">  
docker run -p 80:80 anomaly-detection-app  
</font>

* **Note:**
* The -p flag maps port 80 on the host to port 80 in the container.
* Access your application at http://localhost:80 (if exposed in the Dockerfile).

### **Step 5: Push the Image to Docker Hub (Optional)**

* **Commands:**

<font size="14">  
docker login  
docker tag anomaly-detection-app:latest YOUR\_DOCKER\_HUB\_USERNAME/anomaly-detection-app:latest  
docker push YOUR\_DOCKER\_HUB\_USERNAME/anomaly-detection-app:latest  
</font>

* **Note:**
* Replace YOUR\_DOCKER\_HUB\_USERNAME with your actual Docker Hub username.
* This step is necessary only if you want to share your image or deploy from Docker Hub.

#\*End of AI Generated Content\*