# 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 in Time Series Data using Machine Learning**

## **Step-by-Step Guide for Containerizing and Deploying the Application**

### **Step 1: Create a Dockerfile** (font size: 14)

...............................................................................

* **File Name:** Dockerfile
* **Content:**

<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 PYTHONUNBUFFERED 1  
  
# Run app.py when the container launches  
CMD ["python", "app.py"]  
</font>

* **Note:**
* Create a requirements.txt file in the same directory with the following content:

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

+ Rename your Python script (currently unnamed) to `app.py` and ensure it's in the same directory as the `Dockerfile`.

### **Step 2: Build the Docker Image** (font size: 14)

...............................................................................

* **Command:**

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

* **Explanation:**
* docker build: Command to build an image from a Dockerfile.
* -t anomaly-detection-app:latest: Tags the image with the name anomaly-detection-app and version latest.
* .: Specifies the current directory as the build context.

### **Step 3: Run the Docker Container** (font size: 14)

...............................................................................

* **Command (Interactive Mode):**

<font size="14">  
docker run -it -v $(pwd)/data:/app/data -p 8080:80 anomaly-detection-app:latest  
</font>

* **Command (Detached Mode):**

<font size="14">  
docker run -d -v $(pwd)/data:/app/data -p 8080:80 anomaly-detection-app:latest  
</font>

* **Explanation:**
* docker run: Command to start a new container from the image.
* -it (Interactive Mode) or -d (Detached Mode): Run the container in the specified mode.
* -v $(pwd)/data:/app/data: Mounts a volume from the current directory (./data) to /app/data inside the container for data persistence.
* -p 8080:80: Maps port 8080 on the host to port 80 in the container.
* anomaly-detection-app:latest: Specifies the image to use.

### **Step 4: Verify Deployment** (font size: 14)

...............................................................................

* **Access the Application:**
* Open a web browser and navigate to http://localhost:8080 (if you mapped port 8080).
* Since the application is designed to visualize results, ensure that the Docker container has access to a display environment if running in a non-detached mode, or verify the logs for successful execution in detached mode.

### **Step 5: Push the Image to a Docker Registry (Optional)** (font size: 14)

...........................................................................................

* **Command (after logging into your Docker Hub account):**

<font size="14">  
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>

* **Explanation:**
* Tags the local image with your Docker Hub username.
* Pushes the tagged image to Docker Hub.

#\*End of AI Generated Content\*