Standard Operating Procedure (SOP)

# Serverless Deployment of a Python 3.12 Lambda Function using Docker on Windows (x86\_64)

This SOP outlines the exact steps to deploy the `my-app-test` serverless application using AWS Lambda, Docker containers, the Serverless Framework, and API Gateway on a Windows 11 machine. It includes solutions for issues like image manifest incompatibility.

## 1. Document Purpose

Provide a repeatable guide for deploying the my-app-test serverless app using Dockerized Lambda with Serverless Framework on AWS.

## 2. Prerequisites

Ensure the following tools are installed and configured:

* - AWS Account with CLI credentials configured
* - AWS CLI
* - Node.js and npm (latest LTS)
* - Serverless Framework (`npm install -g serverless`)
* - Docker Desktop with Buildx
* - WSL2 enabled
* - Python 3.12

## 3. Project Setup

Project Directory Structure (D:\AWS\app2):

├── Dockerfile  
├── main.py  
├── requirements.txt  
└── serverless.yml

### main.py

def lambda\_handler(event, context):  
 return {  
 "statusCode": 200,  
 "body": "Hello from Docker Lambda!"  
 }

### requirements.txt

# Add dependencies here (optional)

### Dockerfile

FROM public.ecr.aws/lambda/python:3.12   
WORKDIR ${LAMBDA\_TASK\_ROOT}  
COPY requirements.txt .  
RUN pip install --upgrade pip && \  
 pip install --no-cache-dir --prefer-binary -r requirements.txt  
COPY main.py .  
CMD ["main.lambda\_handler"]

### serverless.yml

service: my-app-test  
  
provider:  
 name: aws  
 runtime: provided.al2  
 region: us-east-1  
 timeout: 29  
 memorySize: 2048  
 ecr:  
 images:  
 app-lambda-test:  
 path: .   
 platform: linux/amd64  
 provenance: false   
  
functions:  
 chatbot:  
 image:  
 name: app-lambda-test  
 architecture: x86\_64  
 events:  
 - http:  
 path: chat  
 method: post  
 environment:  
 GROQ\_API\_KEY: ${env:GROQ\_API\_KEY}

## 4. Docker Setup

Initialize Docker Buildx:

docker buildx create --use  
docker buildx inspect --bootstrap

## 5. Deployment & Cleanup

To deploy:

serverless deploy

To remove all resources:

serverless remove