

- [Building Your Own Mule Docker Image](#)
 - [For Local Development or Community Edition Use](#)
 - [Step-by-Step Example \(Community or Licensed Mule\)](#)
 - [1. Create a Dockerfile](#)
 - [2. Build the Docker image](#)
 - [3. Run the container](#)
 - [Using Docker Compose \(Optional\)](#)
 - [Important Considerations](#)

Building Your Own Mule Docker Image

For Local Development or Community Edition Use

This guide will help you run Mule in Docker locally.

Step-by-Step Example (Community or Licensed Mule)

1. Create a Dockerfile

```
FROM openjdk:8-jdk-alpine

ENV MULE_HOME /opt/mule
ENV PATH $PATH:$MULE_HOME/bin

# Copy Mule distribution and app
COPY mule-ee-distribution-standalone-4.x.x.zip /tmp/
COPY your-app.zip /opt/mule/apps/

# Install Mule
RUN apk add --no-cache unzip && \
    unzip /tmp/mule-ee-distribution-standalone-4.x.x.zip -d /opt/ && \
    mv /opt/mule-enterprise-standalone-* $MULE_HOME && \
    rm /tmp/mule-ee-distribution-standalone-4.x.x.zip

# Expose HTTP listener default port
EXPOSE 8081
```

```
# Start Mule
CMD ["mule"]
```

2. Build the Docker image

```
docker build -t my-mule-app .
```

3. Run the container

```
docker run -p 8081:8081 my-mule-app
```

Using Docker Compose (Optional)

If your application needs a database, Redis, or other services, you can use Docker Compose to orchestrate multiple containers:

```
version: '3'
services:
  mule:
    build: .
    ports:
      - "8081:8081"
    volumes:
      - ./logs:/opt/mule/logs
```

Important Considerations

- **License:** You need a Mule Enterprise Edition license to use the official Mule EE runtime legally.
- **Memory:** Set appropriate `JAVA_OPTS` in Docker for performance (e.g., `-Xmx1G`).
- **Persistence:** Mount volumes if you want logs or configuration persistence.
- **Automation:** Use CI/CD pipelines (Jenkins, GitHub Actions, etc.) to build and deploy Mule apps to Docker/Kubernetes environments.