

# Dockerfile for FastAPI Application using Pyenv and Poetry

M. Pabiszczak

February 21, 2024

## 1 Description of Dockerfile

### 1.1 Base Image

#### Dockerfile Instruction:

```
FROM debian:latest
```

Choose a base image. Debian is a good choice due to Pyenv.

### 1.2 Installing Dependencies

#### Dockerfile Instruction:

```
RUN apt-get update && apt-get install -y \  
    aptitude \  
    tmux vim-nox nano mc git curl termshark procs \  
    sudo iproute2 iputils-ping bridge-utils \  
    ifupdown udev make build-essential libssl-dev zlib1g-dev \  
    libbz2-dev libreadline-dev libsqlite3-dev wget llvm libncurses5-dev \  
    libncursesw5-dev xz-utils libffi-dev liblzma-dev \  
    sqlite3 default-libmysqlclient-dev pkg-config
```

```
RUN echo 'root:pass' | chpasswd
```

```
RUN useradd -m user && echo 'user:pass' | chpasswd && adduser user sudo
```

Update the package list and install necessary dependencies.

### 1.3 Installing Pyenv

#### Dockerfile Instruction:

```
RUN git clone https://github.com/pyenv/pyenv .pyenv
```

```
RUN git clone https://github.com/pyenv/pyenv-virtualenv .pyenv/plugins/pyenv-virtualenv
```

```
ENV HOME /home/user
```

```
ENV PYENV_ROOT $HOME/.pyenv
```

```
ENV PATH $PYENV_ROOT/shims:$PYENV_ROOT/bin:$PATH
```

```
RUN echo 'export PYENV_ROOT="$HOME/.pyenv"' >> .bashrc
```

```
RUN echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> .bashrc
```

```
RUN echo 'eval "$(pyenv init -)"' >> .bashrc
```

```
RUN echo 'eval "$(pyenv virtualenv-init -)"' >> .bashrc
```

Install Pyenv and Pyenv-virtualenv.

## 1.4 Installing Python 3.11

### Dockerfile Instruction:

```
RUN bash -i -c "source ~/.bashrc && env PYTHON_CONFIGURE_OPTS='--enable-shared' pyenv install 3.11"
```

```
RUN bash -i -c "source ~/.bashrc && pyenv virtualenv 3.11 p3.11"
```

```
RUN bash -i -c "source ~/.bashrc && pyenv local 3.11"
```

Install Python 3.11 using Pyenv.

## 1.5 Installing Poetry

### Dockerfile Instruction:

```
RUN curl -sSL https://install.python-poetry.org | python3 -
```

Install Poetry.

## 1.6 Project Configuration

### Dockerfile Instruction:

```
COPY --chown=user:user . /home/user/fapi
```

```
COPY ./_confs/* ./
```

```
WORKDIR /home/user/fapi
```

```
ENV PATH="$HOME/.local/bin:$PATH"
```

```
RUN bash -i -c "source ~/.bashrc && poetry config virtualenvs.create false && poetry install"
```

Copy the project into the image, configure the Poetry path, and install dependencies.

## 1.7 Running the Application

### Dockerfile Instruction:

```
CMD ["/bin/bash", "-c", "python entrypoint.py"]
```

Run the application.

# 2 Running the Container

## 2.1 Running with Default Command

### Docker Command:

```
docker run --rm -dit -p 9999:9999 fapi
```

Run the container with the default command (CMD).

## 2.2 Running with Bash Shell

### Docker Command:

```
docker run --rm -dit -p 9999:9999 --name cfapi fapi bash
```

Run the container with a bash shell.

## 2.3 Accessing Container Shell

### Docker Command:

```
docker exec -it cfapi bash
```

Allows accessing the shell of the container named cfapi.

## A Complete Dockerfile with Database Creation

```
# Choose a base image. Debian is a good choice due to pyenv
FROM debian:latest

# Update the package list and install necessary dependencies
RUN apt-get update && apt-get install -y \
    aptitude \
    tmux vim-nox nano mc git curl termshark procps \
    sudo iproute2 iputils-ping bridge-utils \
    ifupdown udev make build-essential libssl-dev zlib1g-dev \
    libbz2-dev libreadline-dev libsqlite3-dev wget llvm libncurses5-dev \
    libncursesw5-dev xz-utils libffi-dev liblzma-dev \
    sqlite3 default-libmysqlclient-dev pkg-config

# Set password for root user (optional, for configuration)
RUN echo 'root:rootpass' | chpasswd

# Add a new user 'user' with password 'pass' and prepare the environment
RUN useradd -m user && echo 'user:pass' | chpasswd && adduser user sudo

# Switch to user 'user'
USER user
WORKDIR /home/user

# Install pyenv
RUN git clone https://github.com/pyenv/pyenv .pyenv

# Install pyenv-virtualenv
RUN git clone https://github.com/pyenv/pyenv-virtualenv .pyenv/plugins/pyenv-virtualenv

# Set environment variables
ENV HOME /home/user
ENV PYENV_ROOT $HOME/.pyenv
ENV PATH $PYENV_ROOT/shims:$PYENV_ROOT/bin:$PATH

# Add pyenv and pyenv-virtualenv configuration to .bashrc
RUN echo 'export PYENV_ROOT="$HOME/.pyenv"' >> .bashrc
RUN echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> .bashrc
RUN echo 'eval "$(pyenv init -)"' >> .bashrc
RUN echo 'eval "$(pyenv virtualenv-init -)"' >> .bashrc

# Install Python 3.11 with --enable-shared
RUN bash -i -c "source ~/.bashrc && env PYTHON_CONFIGURE_OPTS='--enable-shared' pyenv install 3.11"

# Create a virtual environment p3.11
RUN bash -i -c "source ~/.bashrc && pyenv virtualenv 3.11 p3.11"

# Set p3.11 as the default virtual environment, docker doesn't like ;)
```

```
RUN bash -i -c "source ~/.bashrc && pyenv local 3.11"

# Install Poetry
RUN curl -sSL https://install.python-poetry.org | python3 -

# Copy the project into the image
COPY --chown=user:user . /home/user/fapi
COPY ./_confs/* ./

# Set the working directory for installing dependencies
WORKDIR /home/user/fapi

# Add Poetry path to PATH directly in the Dockerfile
ENV PATH="$HOME/.local/bin:$PATH"

RUN bash -i -c "source ~/.bashrc && poetry config virtualenvs.create false && poetry install"

# Run the application
CMD ["/bin/bash", "-c", "python entrypoint.py"]
```