Dockerfile

### Example

In our example, we will enter a simple **Hello World** echo in our Docker File and create an image and launch a container from it.

**Step 1** − Build the Docker File with the following commands −

FROM ubuntu

MAINTAINER demousr@gmail.com

CMD [“echo” , “hello world”]

FROM ubuntu

MAINTAINER demousr@gmail.com

ENV var1=Tutorial var2=point

$sudo docker build –t=”envno”

Dockerfile

Create an empty directory. Change directories (cd) into the new directory, create a file called Dockerfile, copy-and-paste the following content into that file, and save it. Take note of the comments that explain each statement in your new Dockerfile.

# Use an official Python runtime as a parent image

FROM python:2.7-slim

# Set the working directory to /app

WORKDIR /app

# Copy the current directory contents into the container at /app

ADD . /app

# Install any needed packages specified in requirements.txt

RUN pip install --trusted-host pypi.python.org -r requirements.txt

# Make port 80 available to the world outside this container

EXPOSE 80

# Define environment variable

ENV NAME World

# Run app.py when the container launches

CMD ["python", "app.py"]

**Are you behind a proxy server?**

Proxy servers can block connections to your web app once it’s up and running. If you are behind a proxy server, add the following lines to your Dockerfile, using the ENV command to specify the host and port for your proxy servers:

# Set proxy server, replace host:port with values for your servers

ENV http\_proxy host:port

ENV https\_proxy host:port

Add these lines before the call to pip so that the installation succeeds.

This Dockerfile refers to a couple of files we haven’t created yet, namely app.py and requirements.txt. Let’s create those next.