# Dockerfile Example: Creating an Image to Install MongoDB

In this final section for Dockerfiles, we will create a Dockerfile document and populate it step-by-step with the end result of having a Dockerfile, which can be used to create a docker image to run MongoDB containers.

**Note:** After starting to edit the Dockerfile, all the content and arguments from the sections below are to be written (appended) inside of it successively, following our example and explanations from the **Docker Syntax** section. You can see what the end result will look like at the latest section of this walkthrough.

### Creating the Empty Dockerfile

Using the nano text editor, let's start editing our Dockerfile.

nano Dockerfile

### Defining Our File and Its Purpose

Albeit optional, it is always a good practice to let yourself and everybody figure out (when necessary) what this file is and what it is intended to do. For this, we will begin our Dockerfile with fancy comments (#) to describe it.

############################################################

# Dockerfile to build MongoDB container images

# Based on Ubuntu

############################################################

### Setting The Base Image to Use

# Set the base image to Ubuntu

FROM ubuntu

### Defining The Maintainer (Author)

# File Author / Maintainer

MAINTAINER Example McAuthor

### Setting Arguments and Commands for Downloading MongoDB

################## BEGIN INSTALLATION ######################

# Install MongoDB Following the Instructions at MongoDB Docs

# Ref: http://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/

# Add the package verification key

RUN apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10

# Add MongoDB to the repository sources list

RUN echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | tee /etc/apt/sources.list.d/mongodb.list

# Update the repository sources list

RUN apt-get update

# Install MongoDB package (.deb)

RUN apt-get install -y mongodb-10gen

# Create the default data directory

RUN mkdir -p /data/db

##################### INSTALLATION END #####################

### Setting The Default Port For MongoDB

# Expose the default port

EXPOSE 27017

# Default port to execute the entrypoint (MongoDB)

CMD ["--port 27017"]

# Set default container command

ENTRYPOINT usr/bin/mongod

### Saving The Dockerfile

After you have appended everything to the file, it is time to save and exit. Press CTRL+X and then Y to confirm and save the Dockerfile.

This is what the final file should look like:

############################################################

# Dockerfile to build MongoDB container images

# Based on Ubuntu

############################################################

# Set the base image to Ubuntu

FROM ubuntu

# File Author / Maintainer

MAINTAINER Example McAuthor

################## BEGIN INSTALLATION ######################

# Install MongoDB Following the Instructions at MongoDB Docs

# Ref: http://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/

# Add the package verification key

RUN apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10

# Add MongoDB to the repository sources list

RUN echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | tee /etc/apt/sources.list.d/mongodb.list

# Update the repository sources list

RUN apt-get update

# Install MongoDB package (.deb)

RUN apt-get install -y mongodb-10gen

# Create the default data directory

RUN mkdir -p /data/db

##################### INSTALLATION END #####################

# Expose the default port

EXPOSE 27017

# Default port to execute the entrypoint (MongoDB)

CMD ["--port 27017"]

# Set default container command

ENTRYPOINT usr/bin/mongod

### Building Our First Image

Using the explanations from before, we are ready to create our first MongoDB image with docker!

docker build -t my\_mongodb .

**Note:** The **-t [name]** flag here is used to tag the image. To learn more about what else you can do during build, run docker build --help.

### Running A MongoDB Instance

Using the image we have build, we can now proceed to the final step: creating a container running a MongoDB instance inside, using a name of our choice (if desired with **-name [name]**).

docker run -name my\_first\_mdb\_instance -i -t my\_mongodb

**Note:** If a name is not set, we will need to deal with complex, alphanumeric IDs which can be obtained by listing all the containers using docker ps -l.

**Note:** To detach yourself from the container, use the escape sequence CTRL+P followed by CTRL+Q.

Enjoy!

Writing a Dockerfile

There are already existing Dockerfiles in [Fedora-Dockerfiles](https://github.com/fedora-cloud/Fedora-Dockerfiles) repository. You can use them as examples for creating your own Dockerfile. Each directory contains a Dockerfile and a README with instructions how to build the image and run a container from it.

A Dockerfile content can be as simple as:

FROM fedora:latest

CMD env

For description of instructions used in Dockerfile see [Dockerfile reference](https://docs.docker.com/engine/reference/builder/) and [Best practices for writing Dockerfiles](https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices/)

Building an image from Dockerfile

In a directory with a Dockerfile run

$ sudo docker build -t "my-image" .

If the build is successful you can see the my-image image in docker images output.

See also [Build your own image](https://docs.docker.com/engine/getstarted/step_four/) and [Building an image from a Dockerfile](https://docs.docker.com/engine/getstarted/step_four/#step-2-build-an-image-from-your-dockerfile) for more thorough description.

Authors: [Adam Samalik](mailto:%61%73%61%6D%61%6C%69%6B@%72%65%64%68%61%74.%63%6F%6D), [Budh Ram Gurung](mailto:%62%75%64%68%72%61%6D.%67%75%72%75%6E%67%30%31@%67%6D%61%69%6C.%63%6F%6D), [Jiri Popelka](mailto:%6A%70%6F%70%65%6C%6B%61@%72%65%64%68%61%74.%63%6F%6D), [Josef Stribny](mailto:%6A%73%74%72%69%62%6E%79@%72%65%64%68%61%74.%63%6F%6D)