# #009 Dockerfile

# Introduction

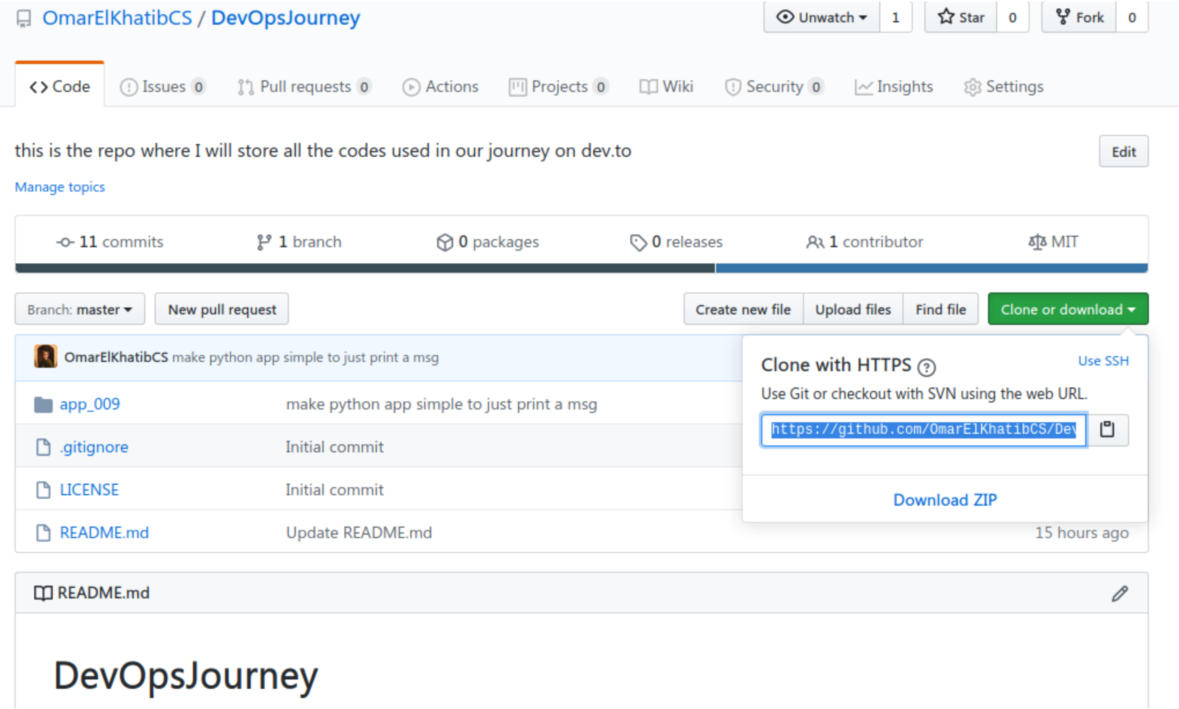
this is part 9 from the journey it's a long journey(360 day) so go please check previous parts , and if you need to walk in the journey with me please make sure to follow because I may post more than once in 1 Day but surely I will post daily at least one 😍.

And I will cover lot of tools as we move on.

---

# setup

first go to DevOpsJourney folder on github .



OmarElKhatibCS / DevOpsJourney

this is the repo where I will store all the codes used in our journey on dev.to

11 commits · 1 branch · 0 packages · 0 releases · 1 contributor · MIT

OmarElKhatibCS make python app simple to just print a msg

| | |
|---|---|
| app_009 | make python app simple to just print a msg |
| .gitignore | Initial commit |
| LICENSE | Initial commit |
| README.md | Update README.md |

15 hours ago

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

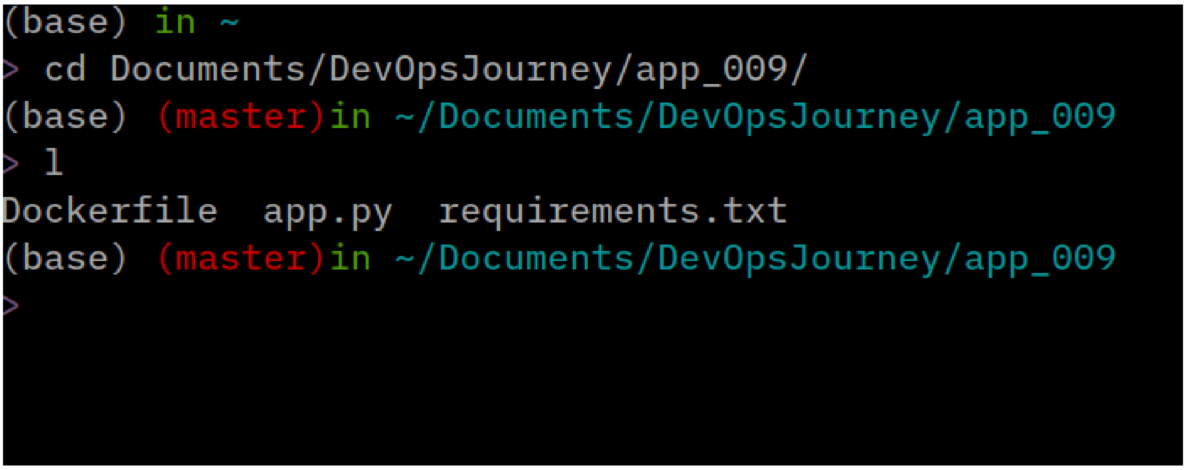
https://github.com/OmarElKhatibCS/Dev

DevOpsJourney

and open terminal then type

```
git clone https://github.com/OmarElKhatibCS/DevOpsJourney.git
cd DevOpsJourney/app_009
```

a look inside the folder



```
(base) in ~
> cd Documents/DevOpsJourney/app_009/
(base) (master)in ~/Documents/DevOpsJourney/app_009
> l
Dockerfile  app.py  requirements.txt
(base) (master)in ~/Documents/DevOpsJourney/app_009
>
```

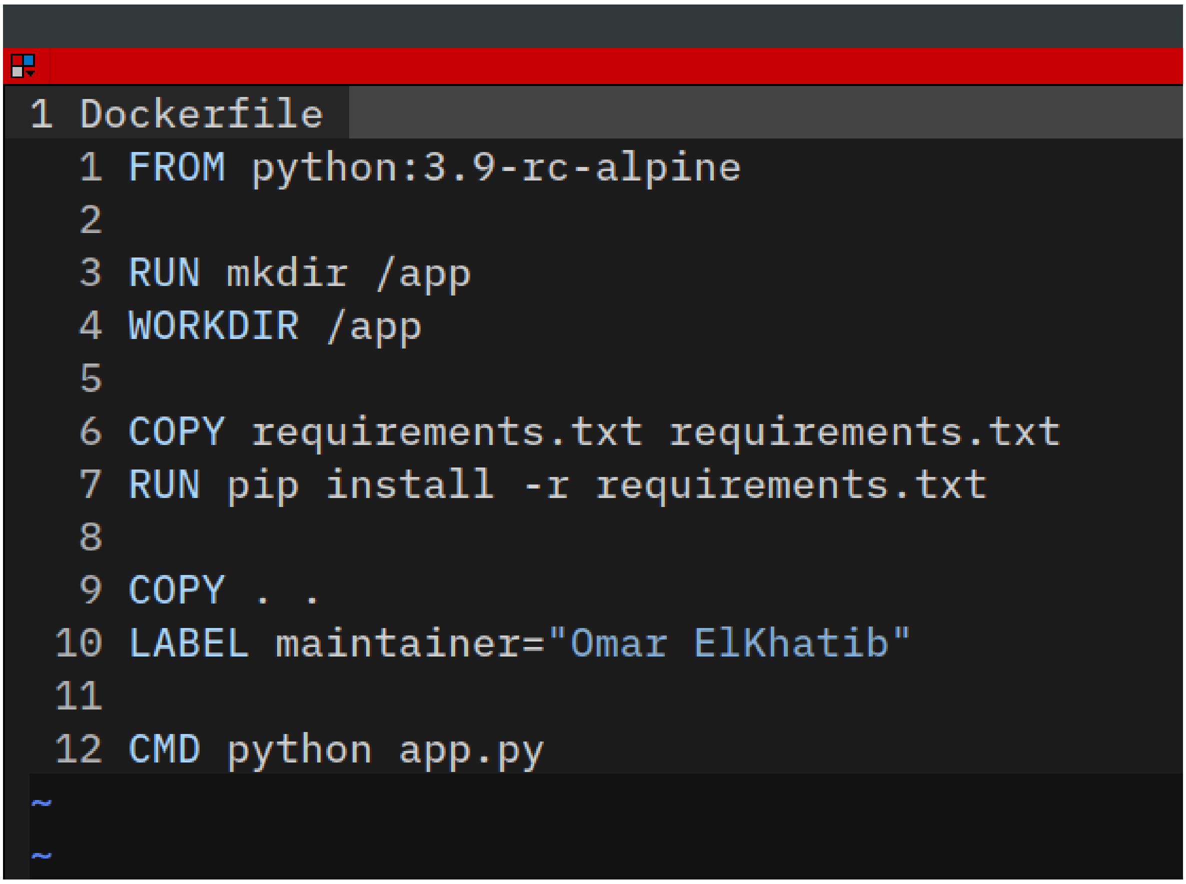
now we have the files for this lesson ready to work on.

---

# Dockerfile

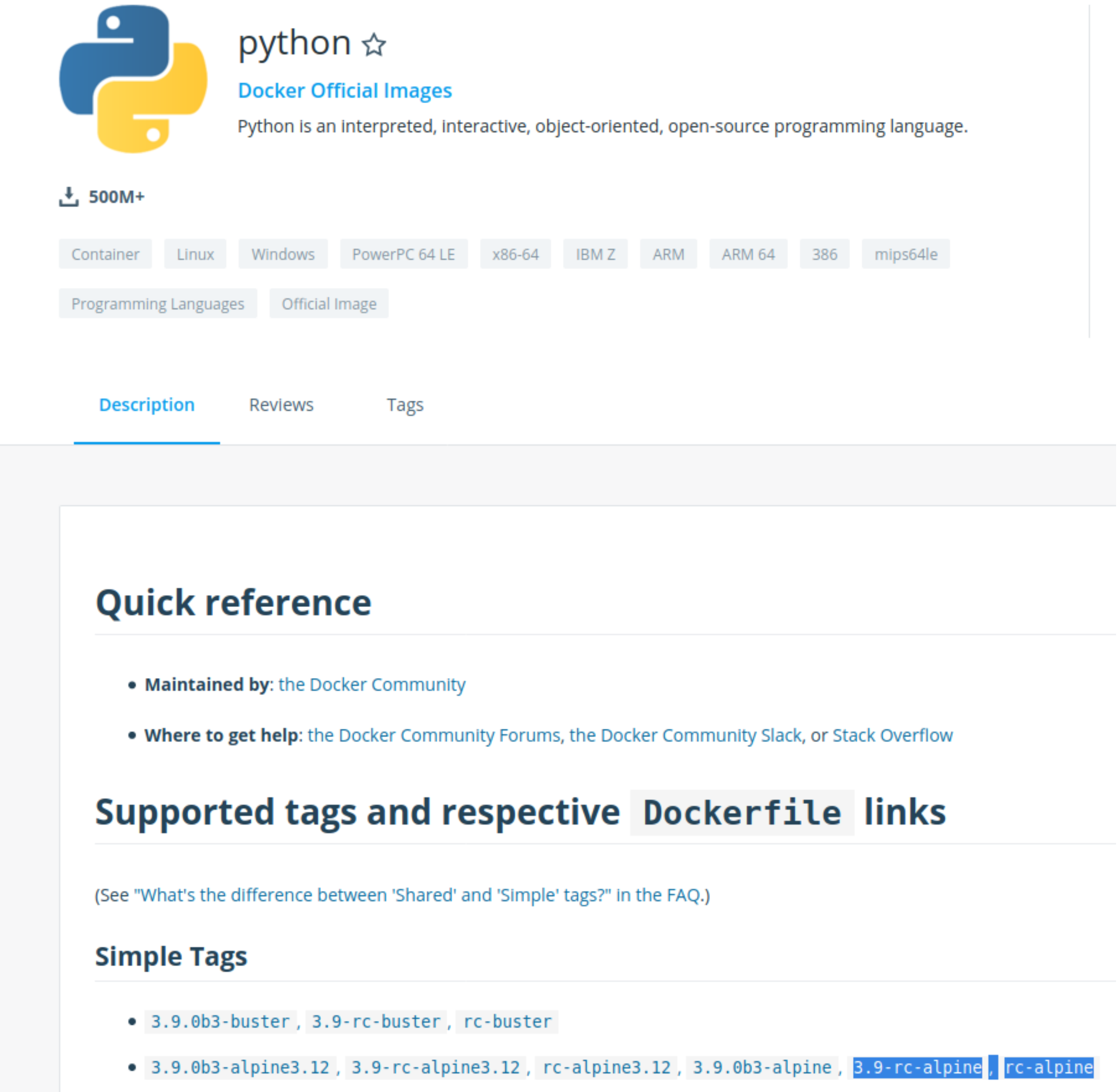
A Docker File is a simple text file with instructions on how to build your images. and he apply the commands from top to down step by step.
And it's good to mention that every step stored as a layer , so if you change something on line 4 , the line 1 to 3 will be cached so he will not execute them.
So the builds will get faster and faster in case of any change.

let's take a look of what inside our Dockerfile in the project that I made special for this part (app_009)



```
1 Dockerfile
   1 FROM python:3.9-rc-alpine
   2
   3 RUN mkdir /app
   4 WORKDIR /app
   5
   6 COPY requirements.txt requirements.txt
   7 RUN pip install -r requirements.txt
   8
   9 COPY . .
  10 LABEL maintainer="Omar ElKhatib"
  11
  12 CMD python app.py
~
~
```

let's start break it line by line :
in line 1 I am asking docker to go and get the image of python with tag 3.9-rc-alpine



python ☆

Docker Official Images

Python is an interpreted, interactive, object-oriented, open-source programming language.

500M+

Container Linux Windows PowerPC 64 LE x86-64 IBM Z ARM ARM 64 386 mips64le

Programming Languages Official Image

## Quick reference

- **Maintained by**: the Docker Community
- **Where to get help**: the Docker Community Forums, the Docker Community Slack, or Stack Overflow

## Supported tags and respective `Dockerfile` links

(See "What's the difference between 'Shared' and 'Simple' tags?" in the FAQ.)

### Simple Tags

- `3.9.0b3-buster`, `3.9-rc-buster`, `rc-buster`
- `3.9.0b3-alpine3.12`, `3.9-rc-alpine3.12`, `rc-alpine3.12`, `3.9.0b3-alpine`, `3.9-rc-alpine`, `rc-alpine`

in line 3 I ask him to RUN mkdir
RUN executes the command(s) that you give in a new layer and creates a new image. This is mainly used for installing a new package.

in line 4 , WORKDIR mean that from now all things I do it will be excuted inside the app folder . like cd but it doesn't change until I change the WORKDIR again.

in line 6 COPY mean copy the requirements text from the folder to the image in app folder (COPY FROM TO)

in line 7 it's will run the line "pip install -r requirements.txt" it's an python cmd to install the dependencies listed in the requirements text. (requirements.txt in the app folder inside the image)

in line 9 COPY . . , it means that copy all the folder containing to the app folder ( . mean the folder that I am inside)

in line 10 LABEL is an META DATA , so it give more information about the docker image.

in line 12 CMD is the default command to be run by the entrypoint. It sets default command and/or parameters, however, we can overwrite those commands or pass in and bypass the default parameters from the command line when docker runs. python app.py is to run the script that I made.