# #018 .dockerignore

## Introduction

this is part 18 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.

## Docker ignore

if you use git before (if not I will cover it somewhere in the journey) there is .gitignore file where when you push your files to github he will ignore to push files listed in .gitignore same here for docker.

what we should ignore ? maybe some environments files , cached files , node_modules folder .
and for security reasons , let's say you need to safe some api_key locally or login credentials.



## Download app_018

```
(base) in ~
> cd Documents/DevOpsJourney/app_013/
(base) (master)in ~/Documents/DevOpsJourney/app_013
> l
Dockerfile  app.py  requirements.txt
(base) (master)in ~/Documents/DevOpsJourney/app_013
>
```

if you follow part 9

```
cd location/DevOpsJourney/
git pull
cd app_018/
```

replace location with where you put the DevOpsJourney

if your new go to part 9 and do same steps will download old lecture files and new one.
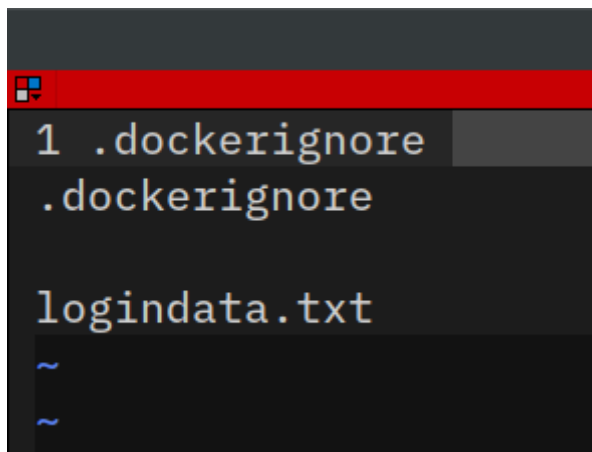
---

# lab

```
(base) (master)in ~/Documents/DevOpsJourney/app_018
> ls
Dockerfile  app.py  logindata.txt  requirements.txt
(base) (master)in ~/Documents/DevOpsJourney/app_018
> cat logindata.txt
email:password
(base) (master)in ~/Documents/DevOpsJourney/app_018
>
```

if you clone the repo and go to app_018 you should see our login file already there.

let's create our .dockerignore file
i use vim to it , inside folder app_018

```
vim .dockerignore
```

as you see i put .dockerignore and logindata.txt
so when we are going to build he will not copy those to our image that we will build
so it's stay locally.
some patterns you may find help full

```
.git
*.json
location/*.txt
```

if your project will be uploaded to git you should ignore the .git folder , usually you not gonna see it because every folder start with . will be hidden in linux you can run

```
ls -a
```

to see all the files in directory.

*.json , * mean all so here I am telling him ignore all json files

location/*.txt mean ignore all the txt files inside a folder (location is folder name)

---

# build and see

```
(base) (master)in ~/Documents/DevOpsJourney/app_018
> docker image build -t app_018 .
Sending build context to Docker daemon    5.12kB
Step 1/8 : FROM python:3.9-rc-alpine
 ---> a206803e6cb1
Step 2/8 : RUN mkdir /app
 ---> Using cache
 ---> cccdcd7a8e00
Step 3/8 : WORKDIR /app
 ---> Using cache
 ---> 1c43ee4da9d4
Step 4/8 : COPY requirements.txt requirements.txt
 ---> Using cache
 ---> b40b0b4697be
Step 5/8 : RUN pip install -r requirements.txt
 ---> Using cache
 ---> 2a9f6ab4dcdb
Step 6/8 : COPY . .
 ---> 5741fc9a8a93
Step 7/8 : LABEL maintainer="Omar ElKhatib"
 ---> Running in a69e6766a7cd
Removing intermediate container a69e6766a7cd
 ---> 6576a22aa852
Step 8/8 : CMD python run app.py
 ---> Running in 57401dc7a277
Removing intermediate container 57401dc7a277
 ---> fad47b82469b
Successfully built fad47b82469b
Successfully tagged app_018:latest
(base) (master)in ~/Documents/DevOpsJourney/app_018
>
```

```
docker image build -t app_018 .
```

always when you follow me watch out for where i am running my commands ,
here I am inside ~/Documents/DevOpsJourney/app_018 so to build I need to use .
(mean Dockerfile is here in this folder)

```
                                                              /bin/bash
                                                                   018
(base) (master)in ~/Documents/DevOpsJourney/app_018
> docker run --rm --name app_018 -exec -it app_018 sh
/app # ls
Dockerfile        app.py              requirements.txt
/app #
```

```
docker run --rm --name app_018 -exec -it app_018 sh
ls
```

as we can see the image does not have the logindata.txt inside , cool!

as challange try to not copy Dockerfile also