

Installation Instructions for Windows

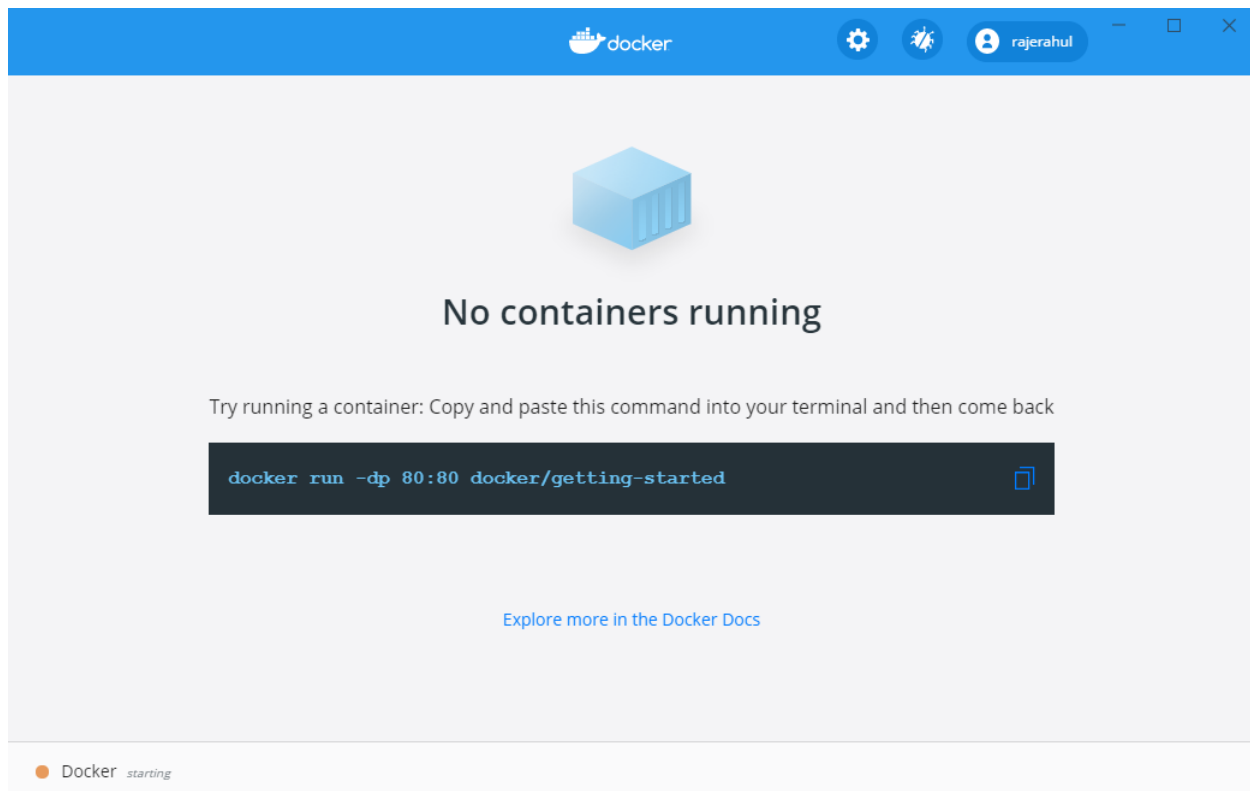
The document contains detailed instructions to install the recommendation system project on a Windows OS.

Step 1: Install Docker

Make sure that you have 'docker for Windows' installed.

If it's not already installed, here is a link for installation:

<https://docs.docker.com/docker-for-windows/install/>

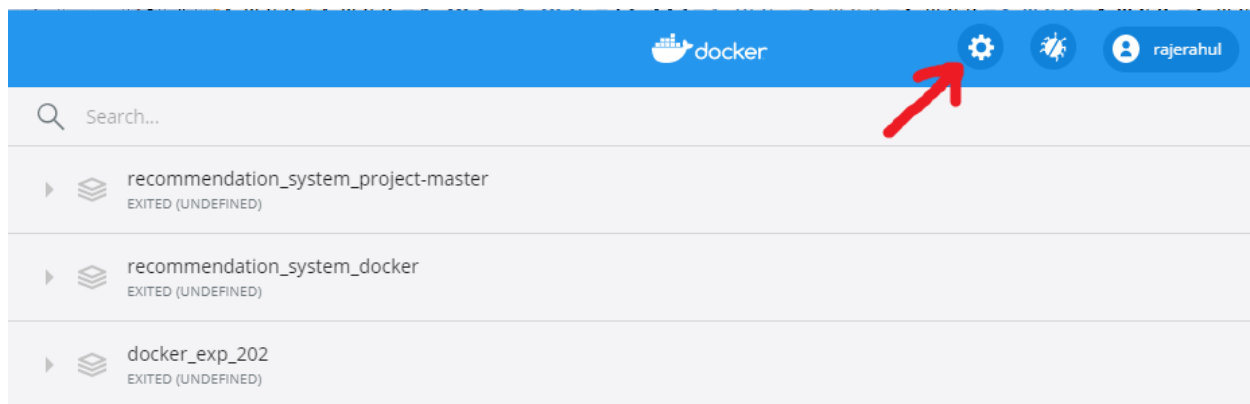


This is the docker dashboard after installation.

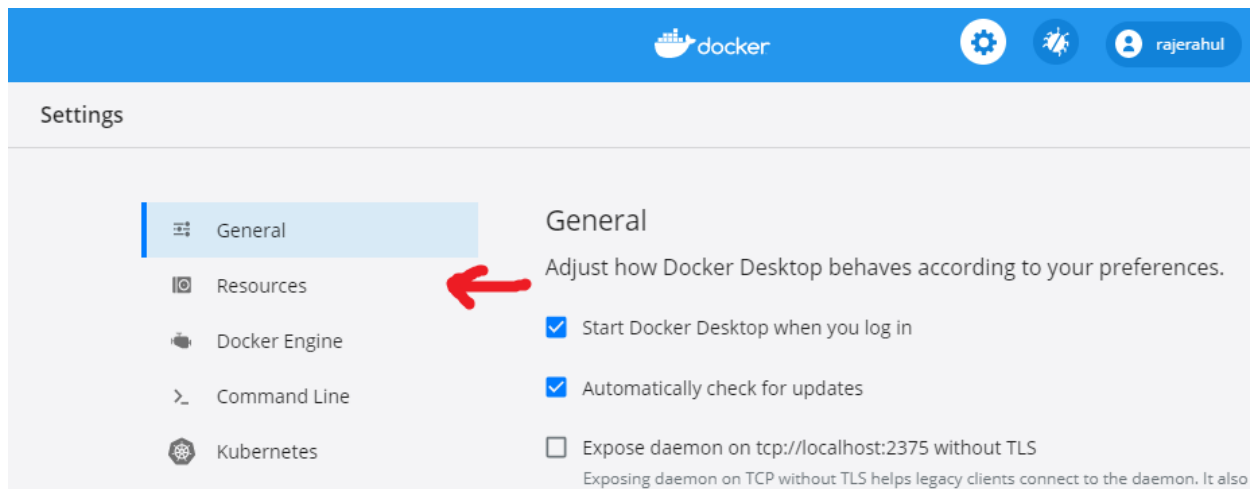
Step 2: Docker File Sharing

Docker may deny running a container if it doesn't have a file sharing access from docker. To give file sharing access to our project folder:

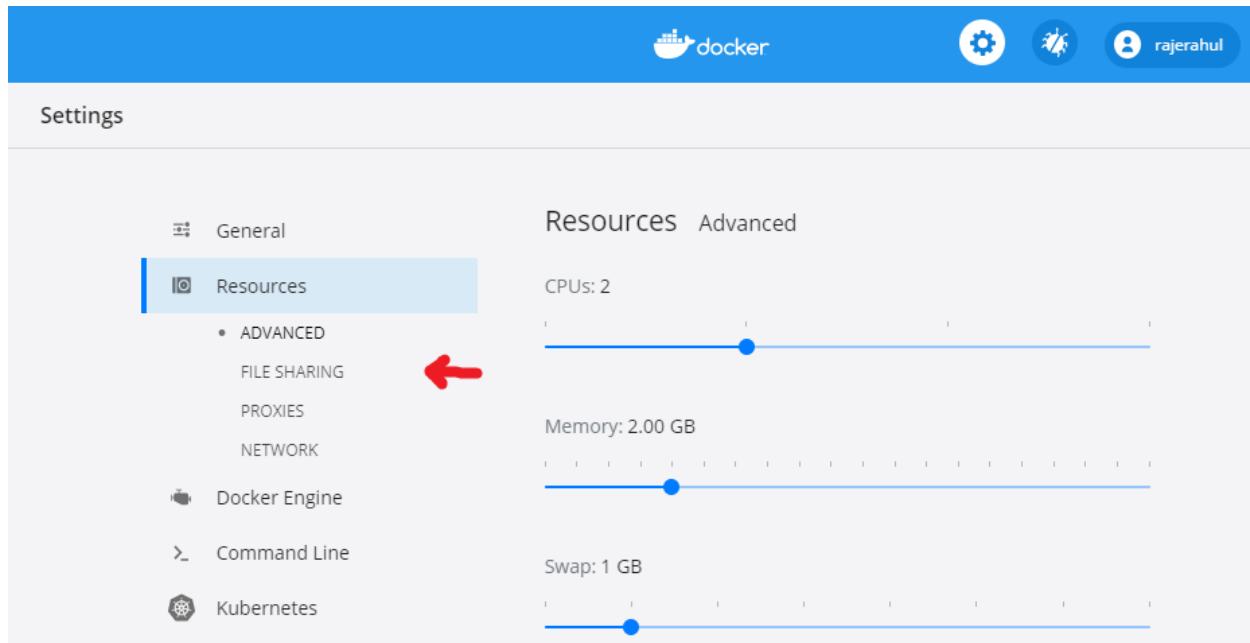
1. Go to settings on Docker dashboard.



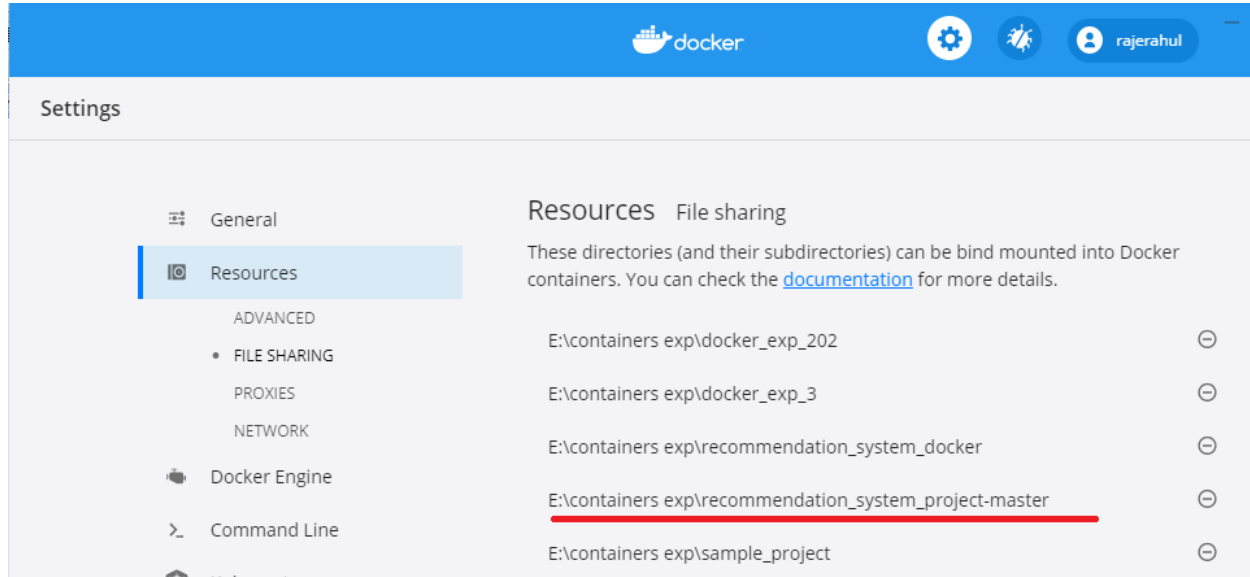
2. Go to resources



3. Go to File Sharing



4. Add project folder path



Select 'Apply and Restart' after adding the path.

Step 3: Build the project image

To build the docker image, open command prompt in the project folder and type the command.

Command: **docker-compose build**

```
C:\Windows\System32\cmd.exe - docker-compose build
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

E:\containers exp\recommendation_system_project-master>docker-compose build
mongo uses an image, skipping
mongo-express uses an image, skipping
Building web
Step 1/7 : FROM python:3
--> 659f826fabf4
Step 2/7 : ENV PYTHONUNBUFFERED 1
--> Using cache
--> d741d0714352
Step 3/7 : RUN mkdir /code
--> Using cache
--> a71695ab01c8
Step 4/7 : WORKDIR /code
--> Using cache
--> 2244f41f88e7
Step 5/7 : ADD . /code
--> 1464df292ed5
Step 6/7 : COPY . /code
--> d90b9c2d4ebc
Step 7/7 : RUN pip install -r requirements.txt
--> Running in 0ac4e0a5886a
Collecting Django==2.2
  Downloading Django-2.2-py3-none-any.whl (7.4 MB)
Collecting djongo==1.3
  Downloading djongo-1.3.0.tar.gz (329 kB)

Stored in directory: /root/.cache/pip/wheels/95/13/60/31c9d5f4cd012e491aeac154ef8b0ec964916523623eb02f0b
Building wheel for odfpy (setup.py): started
Building wheel for odfpy (setup.py): finished with status 'done'
Created wheel for odfpy: filename=odfpy-1.4.1-py2.py3-none-any.whl size=160692 sha256=2315b156649d0d24d0b920c90b46f696
518a150e0a5f552a37ec4a4b572c274f
Stored in directory: /root/.cache/pip/wheels/ea/af/da/2bdd7308f7b334429a558df1e36d31864cd19c07ede92ddf0e
Building wheel for pyyaml (setup.py): started
Building wheel for pyyaml (setup.py): finished with status 'done'
Created wheel for pyyaml: filename=PyYAML-5.3.1-cp38-cp38-linux_x86_64.whl size=572455 sha256=e02ef6c373272ad6a19eb989
4d8e80112d31c7925d0c488e59625738fa688b4a
Stored in directory: /root/.cache/pip/wheels/13/90/db/290ab3a34f2ef0b5a0f89235dc2d40fea83e77de84ed2dc05c
Building wheel for et-xmlfile (setup.py): started
Building wheel for et-xmlfile (setup.py): finished with status 'done'
Created wheel for et-xmlfile: filename=et_xmlfile-1.0.1-py3-none-any.whl size=8915 sha256=931f168dcd2e7cec17a6c6565092
d99eee7b292efd777f73fb852d14c72e8fbb
Stored in directory: /root/.cache/pip/wheels/6e/df/38/abda47b884e3e25f9f9b6430e5ce44c47670758a50c0c51759
Successfully built djongo bson diff-match-patch markuppy odfpy pyyaml et-xmlfile
Installing collected packages: sqlparse, pytz, Django, six, python-dateutil, bson, pymongo, dataclasses, djongo, gunicorn, dnspython, django-crispy-forms, markuppy, defusedxml, odfpy, xlrd, xlwt, jdcals, et-xmlfile, openpyxl, pyyaml, tablib, diff-match-patch, django-import-export
Successfully installed Django-2.2 bson-0.5.8 dataclasses-0.6 defusedxml-0.6.0 diff-match-patch-20181111 django-crispy-forms-1.9.1 django-import-export-2.0.2 djongo-1.3.0 dnspython-1.16.0 et-xmlfile-1.0.1 gunicorn-20.0.4 jdcals-1.4.1 markuppy-1.14 odfpy-1.4.1 openpyxl-3.0.4 pymongo-3.10.1 python-dateutil-2.8.1 pytz-2020.1 pyyaml-5.3.1 six-1.15.0 sqlparse-0.2.4 tablib-2.0.0 xlrd-1.2.0 xlwt-1.3.0
Removing intermediate container 0ac4e0a5886a
--> e752e1d7f9d5
Successfully built e752e1d7f9d5
Successfully tagged recommendation_system_project-master_web:latest

E:\containers exp\recommendation_system_project-master>
```

Step 4: Run the image

You can run *docker images* to check if the new image has been added. Now we need to run this image.

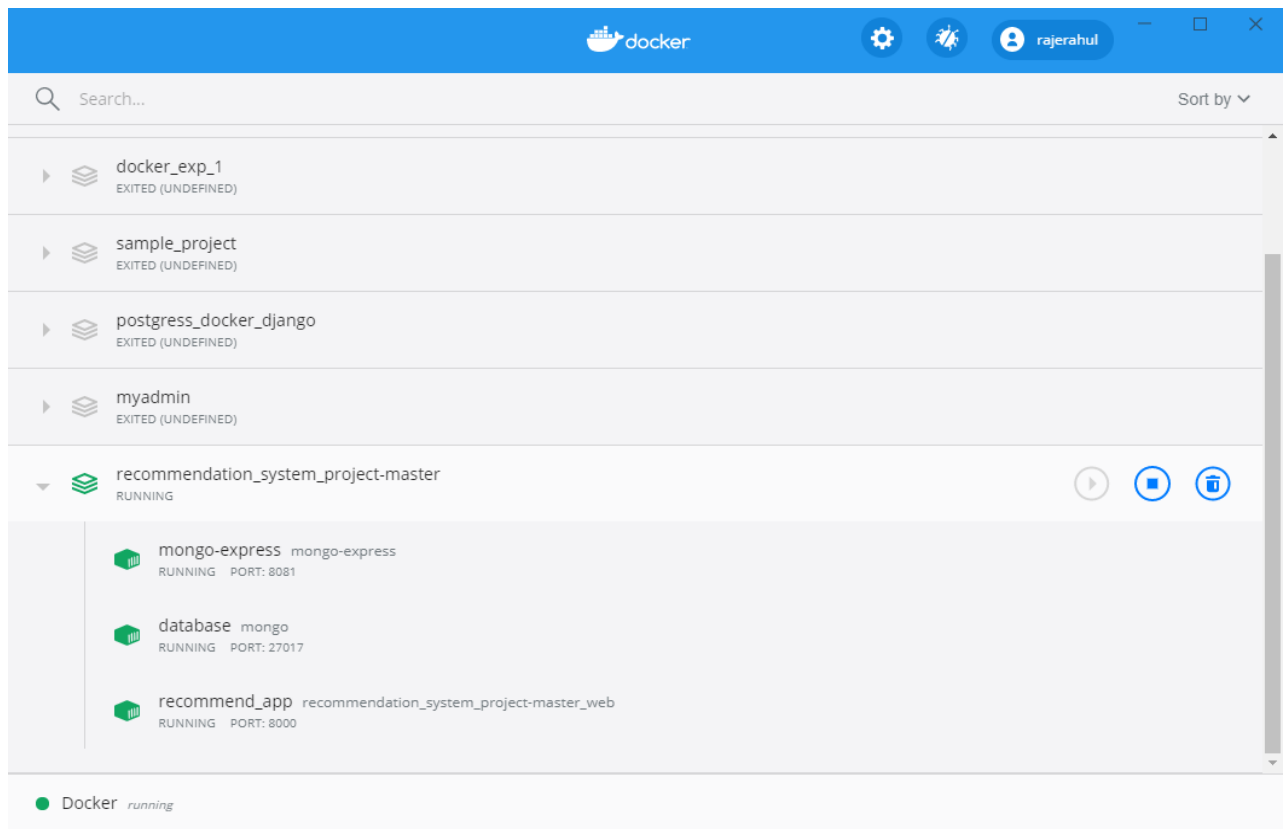
Command: **docker-compose up**

```
C:\Windows\System32\cmd.exe - docker-compose up
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

E:\containers exp\recommendation_system_project-master>docker-compose up
Starting database ... done
Starting mongo-express ... done
Starting recommend_app ... done
Attaching to database, mongo-express, recommend_app
database | 2020-06-26T11:40:32.404+0000 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS
1.0 specify --sslDisabledProtocols 'none'
database | 2020-06-26T11:40:32.407+0000 W ASIO [main] No TransportLayer configured during NetworkInterface
startup
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] MongoDB starting : pid=1 port=27017 dbpath=/
data/db 64-bit host=e66679005a08
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] db version v4.2.8
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] git version: 43d25964249164d76d5e04dd6cf38f6
111e21f5f
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.1.1 11 Sep 2018
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] allocator: tcmalloc
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] modules: none
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] build environment:
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] distmod: ubuntu1804
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] distarch: x86_64
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] target_arch: x86_64
database | 2020-06-26T11:40:32.408+0000 I CONTROL [initandlisten] options: { net: { bindIp: "*" } }
mongo-express | Waiting for mongo:27017...
database | 2020-06-26T11:40:32.410+0000 I STORAGE [initandlisten] Detected data files in /data/db created by t
he 'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'.
database | 2020-06-26T11:40:32.410+0000 I STORAGE [initandlisten]
database | 2020-06-26T11:40:32.410+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is stro
v
C:\Windows\System32\cmd.exe - docker-compose up
database | 2020-06-26T11:40:33.896+0000 I NETWORK [listener] connection accepted from 172.28.0.2:44114 #4 (3 c
onnections now open)
database | 2020-06-26T11:40:34.000+0000 I SHARDING [ftdc] Marking collection local.oplog.rs as collection versi
on: <unsharded>
recommend_app | Watching for file changes with StatReloader
recommend_app | Performing system checks...
recommend_app | System check identified no issues (0 silenced).
database | 2020-06-26T11:40:34.923+0000 I NETWORK [listener] connection accepted from 172.28.0.4:46358 #5 (4 c
onnections now open)
database | 2020-06-26T11:40:34.924+0000 I NETWORK [conn5] received client metadata from 172.28.0.4:46358 conn5
: { driver: { name: "PyMongo", version: "3.10.1" }, os: { type: "Linux", name: "Linux", architecture: "x86_64", version:
"4.19.76-linuxkit" }, platform: "CPython 3.8.3.final.0" }
database | 2020-06-26T11:40:34.927+0000 I NETWORK [listener] connection accepted from 172.28.0.4:46360 #6 (5 c
onnections now open)
database | 2020-06-26T11:40:34.928+0000 I NETWORK [conn6] received client metadata from 172.28.0.4:46360 conn6
: { driver: { name: "PyMongo", version: "3.10.1" }, os: { type: "Linux", name: "Linux", architecture: "x86_64", version:
"4.19.76-linuxkit" }, platform: "CPython 3.8.3.final.0" }
database | 2020-06-26T11:40:34.933+0000 I SHARDING [conn6] Marking collection recommendation_system_db.django_m
igrations as collection version: <unsharded>
recommend_app | June 26, 2020 - 11:40:34
recommend_app | Django version 2.2, using settings 'recommendation_system.settings'
recommend_app | Starting development server at http://0.0.0.0:8000/
recommend_app | Quit the server with CONTROL-C.
```

Do not close this window, minimize it and keep the containers running in the background.

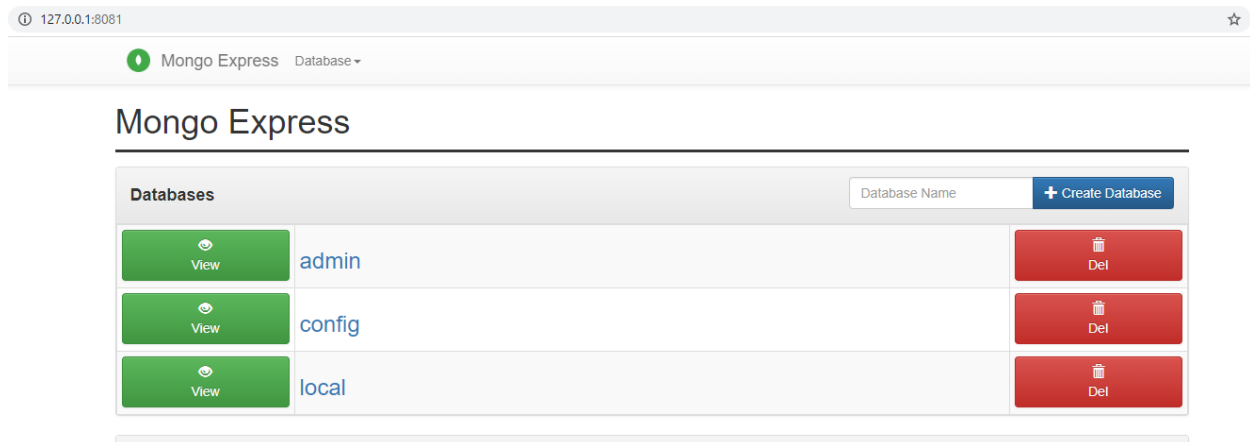
You can check into your docker desktop dashboard to find that the containers are running.



Step 5: Running script.bat

At this point, it is possible to access the web application. But the database has not yet been migrated and imported.

Go to **localhost:8081** to access Mongo-Express.



Here, we can observe that we are yet to add our application database into the mongo container.

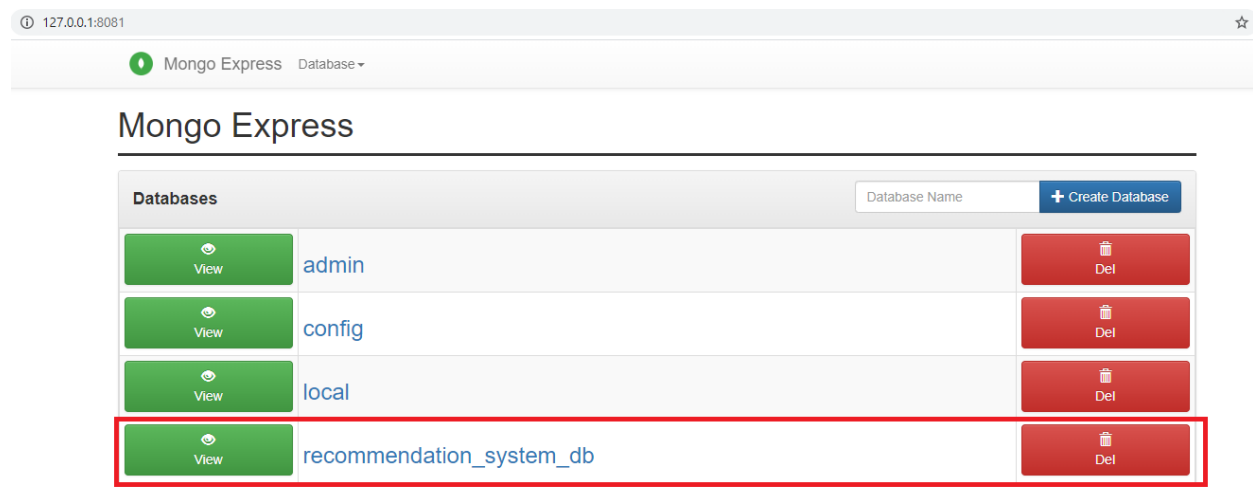
In order to add the system database, we need to run the **script.bat** file in the project folder. Just double-click on the file and it will migrate and import everything into the running containers.

Make sure that the containers are 'up' and not 'exited' when executing script.bat

```
C:\WINDOWS\system32\cmd.exe

Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying recommendation_webapp.0001_initial... OK
Applying recommendation_webapp.0002_auto_20200610_1655... OK
Applying recommendation_webapp.0003_anime2... OK
Applying recommendation_webapp.0004_delete_anime... OK
Applying sessions.0001_initial... OK
copy movies
copy books
copy animes
import movie.json
2020-06-26T11:58:31.862+0000    connected to: mongodb://localhost/
2020-06-26T11:58:32.934+0000    39515 document(s) imported successfully. 0 document(s) failed to import.
import book.json
2020-06-26T11:58:33.109+0000    connected to: mongodb://localhost/
2020-06-26T11:58:34.107+0000    32581 document(s) imported successfully. 0 document(s) failed to import.
import anime2.json
2020-06-26T11:58:34.285+0000    connected to: mongodb://localhost/
2020-06-26T11:58:34.944+0000    15044 document(s) imported successfully. 0 document(s) failed to import.
all imports done successfully!
Go to http://127.0.0.1:8080/recommendations_webapp_home/ to try the application
Press any key to continue . . .
```

All the migrations are done. You can verify by accessing Mongo-Express again.



The 'recommendation_system_db' has been added.

127.0.0.1:8081/db/recommendation_system_db/

Mongo Express Database: recommendation_system_db

| | | | | | |
|------|--------|--------|--------|------------------------------|-----|
| View | Export | [JSON] | Import | django_admin_log | Del |
| View | Export | [JSON] | Import | django_content_type | Del |
| View | Export | [JSON] | Import | django_migrations | Del |
| View | Export | [JSON] | Import | django_session | Del |
| View | Export | [JSON] | Import | recommendation_webapp_anime2 | Del |
| View | Export | [JSON] | Import | recommendation_webapp_book | Del |
| View | Export | [JSON] | Import | recommendation_webapp_movie | Del |

You can also verify that the Movie, Anime and Books collections have been imported into the database.

Step 6: Check the output

To access the application, go to:

Localhost:8000/recommendations_webapp_home/



Preferred browsers:

Chrome, Opera, Firefox.

For best experience, if your OS or browsers has a side bar get the side bar at the bottom.