**DOCKER CONTAINERIZATION**

To deploy a Java / Python or any application, we either need a physical machine or virtual machine (environment) that can be on a cloud.

We have to install all the software, networking, memory setup and database creation, along with our application. The entire process of this is time taking and expensive, that’s where we will be using Containerization or Containers come in.

Containers can be very quickly started and containers can be moved across environments and machines, very easily while maintaining the same level of consistency.

Containerization is the process of bringing together all the components that are required for our application to work along with the application itself and we launch containers that can be easily moved across environments or scaled up and scaled down.

Docker is the most popular containerization tool that follows OCI (Open Container Initiative) standard.

Instead of deploying our application as war / jar / dll / Django code, Once we dockerise the project, we will be deploying an image. This docker image is used to launch the application.

From this docker image, we will create docker containers, which can be deployed across environments as required.

Microsoft Windows [Version 10.0.22631.3593]

(c) Microsoft Corporation. All rights reserved.

C:\Users\kamal>docker --version

Docker version 25.0.3, build 4debf41

C:\Users\kamal>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

C:\Users\kamal>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

From the below code

-d : detached mode

-p : publishing a port (6666 🡪 3306) From 6666 it will be redirected to3306 on docker container

--name : container name

Environment (env) : used by MySQL server.

C:\Users\kamal>docker run --name mysql-django-container --env MYSQL\_DATABASE=clinicalsDb --env MYSQL\_ROOT\_PASSWORD=Jerrick2911 --publish 6666:3306 --detach docker.io/library/mysql

Unable to find image 'mysql:latest' locally

latest: Pulling from library/mysql

fcbdc4090331: Pull complete

95eb5073c36f: Pull complete

5e5ba0e6412a: Pull complete

ee4654eb29b7: Pull complete

1d2218160b86: Pull complete

bbe712936412: Pull complete

4a6efa152609: Pull complete

0f7c3b67bb2b: Pull complete

7ccd17e83c8a: Pull complete

04bf2c116556: Pull complete

Digest: sha256:4a4e5e2a19aab7a67870588952e8f401e17a330466ecfc55c9acf51196da5bd0

Status: Downloaded newer image for mysql:latest

cf5bfe0c99dc1ab498e2b3c83de0115f4299505dbc75d4e449cd309583db70cd

C:\Users\kamal>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

mysql latest e9387c13ed83 4 weeks ago 578MB

C:\Users\kamal>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

cf5bfe0c99dc mysql "docker-entrypoint.s…" 27 seconds ago Up 20 seconds 33060/tcp, 0.0.0.0:6666->3306/tcp mysql-django-container

C:\Users\kamal>docker exec -it mysql-django-container bash

bash-5.1# ls

afs boot docker-entrypoint-initdb.d home lib64 mnt proc run srv tmp var

bin dev etc lib media opt root sbin sys usr

bash-5.1#

bash-5.1# mysql -uroot -p

Enter password: Jerrick2911

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 9

Server version: 8.4.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| clinicalsDb |

| information\_schema |

| mysql |

| performance\_schema |

| sys |

+--------------------+

5 rows in set (0.00 sec)

mysql>

mysql> use clinicalsDb

Database changed

mysql> show tables

-> ;

Empty set (0.01 sec)

mysql>

**CREATE A requirements.txt FILE USING freeze**

C:\Users\kamal\django\_projects\clinicalsProject>pip freeze > requirments.txt

C:\Users\kamal\django\_projects\clinicalsProject>dir

Volume in drive C is Local Disk

Volume Serial Number is 9CC0-39AA

Directory of C:\Users\kamal\django\_projects\clinicalsProject

30/05/2024 23:41 <DIR> .

29/05/2024 23:06 <DIR> ..

27/05/2024 23:04 <DIR> .idea

25/05/2024 00:00 <DIR> clinicalsApp

24/05/2024 23:52 <DIR> clinicalsProject

24/05/2024 13:22 694 manage.py

30/05/2024 23:41 568 requirments.txt

24/05/2024 19:42 <DIR> static

24/05/2024 13:23 <DIR> templates

24/05/2024 13:23 <DIR> \_\_pycache\_\_

2 File(s) 1,262 bytes

8 Dir(s) 21,304,041,472 bytes free

C:\Users\kamal\django\_projects\clinicalsProject>

In settings.py, change the HOST to the name, which was provided when creating the image (mysql-django-container)

A screenshot of a computer

Description automatically generated

PS C:\Users\kamal\django\_projects\clinicalsProject> ls

Directory: C:\Users\kamal\django\_projects\clinicalsProject

Mode LastWriteTime Length Name

---- ------------- ------ ----

d----- 27/05/2024 23:04 .idea

d----- 25/05/2024 00:00 clinicalsApp

d----- 24/05/2024 23:52 clinicalsProject

d----- 24/05/2024 19:42 static

d----- 24/05/2024 13:23 templates

d----- 24/05/2024 13:23 \_\_pycache\_\_

-a---- 31/05/2024 00:00 191 Dockerfile.txt

-a---- 24/05/2024 13:22 694 manage.py

-a---- 30/05/2024 23:41 568 requirments.txt

**CREATE A DockerFile AND SAVE IT IN THE PROJECT FOLDER**

FROM python:3.12.3

WORKDIR /clinicaldjangoapps

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY . .

EXPOSE 8000

CMD ["python","manage.py","runserver","0.0.0.0:8000"]

A screenshot of a computer

Description automatically generated

**EXECUTE THIS CMD IN THE FOLDER WHERE THE Dockerfile and requirements.txt is available.**

PS C:\Users\kamal\django\_projects\clinicalsProject> docker build -f Dockerfile -t clinicals\_app .

[+] Building 444.8s (10/10) FINISHED docker:default

=> [internal] load build definition from Dockerfile 0.1s

=> => transferring dockerfile: 230B 0.0s

=> [internal] load metadata for docker.io/library/python:3.12.3 2.7s

=> [internal] load .dockerignore 0.0s

=> => transferring context: 2B 0.0s

=> [internal] load build context 0.1s

=> => transferring context: 14.99kB 0.1s

=> [1/5] FROM docker.io/library/python:3.12.3@sha256:3966b81808d864099f802080d897cef36c01550472ab3955fdd716d1c665acd6 135.4s

=> => resolve docker.io/library/python:3.12.3@sha256:3966b81808d864099f802080d897cef36c01550472ab3955fdd716d1c665acd6 0.1s

=> => sha256:891494355808bdd3db5552f0d3723fd0fa826675f774853796fafa221d850d42 24.05MB / 24.05MB 6.3s

=> => sha256:6582c62583ef22717db8d306b1d6a0c288089ff607d9c0d2d81c4f8973cbfee3 64.14MB / 64.14MB 27.8s

=> => sha256:3966b81808d864099f802080d897cef36c01550472ab3955fdd716d1c665acd6 2.14kB / 2.14kB 0.0s

=> => sha256:76831bb094456f98613115be37cc8d4f9ab0d128467726ed3160044c80546437 2.01kB / 2.01kB 0.0s

=> => sha256:12e5ab9d51c883bedc4db6c7cbc49f3fa97aef4e98dc205bf1c67e21fd9cb6f4 7.10kB / 7.10kB 0.0s

=> => sha256:c6cf28de8a067787ee0d08f8b01d7f1566a508b56f6e549687b41dfd375f12c7 49.58MB / 49.58MB 22.1s

=> => sha256:bf2c3e352f3d2eed4eda4feeed44a1022a881058df20ac0584db70c138b041e2 211.21MB / 211.21MB 59.3s

=> => sha256:a99509a323905a80628005e4f3bc26ac15ebaf3ffdb08a9646a7f2d110ab38f9 6.39MB / 6.39MB 29.0s

=> => extracting sha256:c6cf28de8a067787ee0d08f8b01d7f1566a508b56f6e549687b41dfd375f12c7 27.2s

=> => sha256:d46a03def8d9bd846f37fee0edc9fcb7400d68166e99f039cd87b99948af6005 22.69MB / 22.69MB 36.5s

=> => sha256:4429b810e09ef699c3034e49b60f85add3610968983af403fb3181255a5b639f 243B / 243B 29.5s

=> => sha256:2a4ca5af09fa6cc704c04ba61e304cbf3d0fb2c55f52bf8c8dcba14651216c17 2.77MB / 2.77MB 31.4s

=> => extracting sha256:891494355808bdd3db5552f0d3723fd0fa826675f774853796fafa221d850d42 6.7s

=> => extracting sha256:6582c62583ef22717db8d306b1d6a0c288089ff607d9c0d2d81c4f8973cbfee3 16.3s

=> => extracting sha256:bf2c3e352f3d2eed4eda4feeed44a1022a881058df20ac0584db70c138b041e2 44.7s

=> => extracting sha256:a99509a323905a80628005e4f3bc26ac15ebaf3ffdb08a9646a7f2d110ab38f9 4.5s

=> => extracting sha256:d46a03def8d9bd846f37fee0edc9fcb7400d68166e99f039cd87b99948af6005 5.4s

=> => extracting sha256:4429b810e09ef699c3034e49b60f85add3610968983af403fb3181255a5b639f 0.0s

=> => extracting sha256:2a4ca5af09fa6cc704c04ba61e304cbf3d0fb2c55f52bf8c8dcba14651216c17 2.8s

=> [2/5] WORKDIR /clinicaldjangoapps 5.8s

=> [3/5] COPY requirements.txt . 1.4s

=> [4/5] RUN pip install -r requirements.txt 262.2s

=> [5/5] COPY . . 0.7s

=> exporting to image 35.3s

=> => exporting layers 35.2s

=> => writing image sha256:2746d846071190f12c47386dceb2740360b53ca235aa57cd0eea0ee61f907b59 0.0s

=> => naming to docker.io/library/clinicals\_app 0.0s

View build details: docker-desktop://dashboard/build/default/default/fol8pma4ps237opu0x1vw9zcy

What's Next?

View a summary of image vulnerabilities and recommendations → docker scout quickview

PS C:\Users\kamal\django\_projects\clinicalsProject> docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

clinicals\_app latest 2746d8460711 2 minutes ago 1.5GB

mysql latest e9387c13ed83 4 weeks ago 578MB

PS C:\Users\kamal\django\_projects\clinicalsProject>

A computer screen shot of an object

Description automatically generated

**CHANGED THE CMD IN Dockerfile**

FROM

CMD ["py","-m","manage","runserver","0.0.0.0:8000"]

TO

CMD ["python","manage.py","runserver","0.0.0.0:8000"]

PS C:\Users\kamal\django\_projects\clinicalsProject> docker build -f Dockerfile -t clinicals\_app .

[+] Building 16.6s (11/11) FINISHED docker:default

=> [internal] load build definition from Dockerfile 0.1s

=> => transferring dockerfile: 232B 0.1s

=> [internal] load metadata for docker.io/library/python:3.12.3 9.3s

=> [auth] library/python:pull token for registry-1.docker.io 0.0s

=> [internal] load .dockerignore 0.3s

=> => transferring context: 2B 0.1s

=> [1/5] FROM docker.io/library/python:3.12.3@sha256:3966b81808d864099f802080d897cef36c01550472ab3955fdd716d1c665acd6 0.0s

=> [internal] load build context 1.0s

=> => transferring context: 14.41kB 0.9s

=> CACHED [2/5] WORKDIR /clinicaldjangoapps 0.0s

=> CACHED [3/5] COPY requirements.txt . 0.0s

=> CACHED [4/5] RUN pip install -r requirements.txt 0.0s

=> [5/5] COPY . . 3.3s

=> exporting to image 0.7s

=> => exporting layers 0.6s

=> => writing image sha256:dd6eb944872999c2baa1bb0a9d9ced4c6e7d2ffeaab70e697bb2f2c164761584 0.0s

=> => naming to docker.io/library/clinicals\_app 0.1s

View build details: docker-desktop://dashboard/build/default/default/f6vhld2z14yxe316t4aa24tad

What's Next?

View a summary of image vulnerabilities and recommendations → docker scout quickview

PS C:\Users\kamal\django\_projects\clinicalsProject> docker run -t --name=clinicals-app --link mysql-django-container:mysql -p 10555:8000 clinicals\_app

docker: Error response from daemon: Conflict. The container name "/clinicals-app" is already in use by container "9360566999d787e89b997d5155068d5ce5f601af8eb9d330a25202c1146720eb". You have to remove (or rename) that container to be able to reuse that name.

See 'docker run --help'.

PS C:\Users\kamal\django\_projects\clinicalsProject> docker run -t --name=clinicals-appln --link mysql-django-container:mysql -p 10555:8000 clinicals\_app

Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

You have 19 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, clinicalsApp, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.

May 30, 2024 - 23:29:05

Django version 5.0.4, using settings 'clinicalsProject.settings'

Starting development server at http://0.0.0.0:8000/

Quit the server with CONTROL-C.

MIGRATIONS FROM THE CONTAINER

PS C:\Users\kamal> docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

63c282a08665 clinicals\_app "python manage.py ru…" 4 minutes ago Up 4 minutes 0.0.0.0:10555->8000/tcp clinicals-appln

cf5bfe0c99dc mysql "docker-entrypoint.s…" About an hour ago Up About an hour 33060/tcp, 0.0.0.0:6666->3306/tcp mysql-django-container

PS C:\Users\kamal> docker exec -it clinicals-appln bash

root@63c282a08665:/clinicaldjangoapps# ls

Dockerfile \_\_pycache\_\_ clinicalsApp clinicalsProject manage.py requirements.txt static templates

root@63c282a08665:/clinicaldjangoapps# python manage.py migrate

Operations to perform:

Apply all migrations: admin, auth, clinicalsApp, contenttypes, sessions

Running migrations:

Applying contenttypes.0001\_initial... OK

Applying auth.0001\_initial... OK

Applying admin.0001\_initial... OK

Applying admin.0002\_logentry\_remove\_auto\_add... OK

Applying admin.0003\_logentry\_add\_action\_flag\_choices... OK

Applying contenttypes.0002\_remove\_content\_type\_name... OK

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 auth.0012\_alter\_user\_first\_name\_max\_length... OK

Applying clinicalsApp.0001\_initial... OK

Applying sessions.0001\_initial... OK

root@63c282a08665:/clinicaldjangoapps#

FROM THE BROWSER

localhost:10555

A screenshot of a video

Description automatically generated

A close-up of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

IN THE CMD PROMPT (WHILE LOADING THE HOME PAGE)

A screen shot of a computer code

Description automatically generated