

Day 17 Task: Docker Project for DevOps Engineers.

Dockerfile

A Dockerfile is a script that contains a set of instructions used to build a Docker image. It is a simple text file that contains all the commands, in order, needed to build a specific image. Once the file is created, it can be used to build an image that can be run as a container.

The basic structure of a Dockerfile is a series of commands, with each command being executed in the order specified. Each command creates a new layer in the image, and the final image consists of all the layers stacked together.

The most common commands used in a Dockerfile include:

FROM: This command specifies the base image that the new image will be built on top of. The base image can be an official image from the Docker Hub or a custom image from a private repository.

RUN: This command runs a command in the shell of the container. It is used to install packages, create directories, or perform any other task that needs to be done during the image creation process.

COPY: This command copies files or directories from the host machine to the container.

ENV: This command sets environment variables in the container.

EXPOSE: This command informs Docker that the container listens on the specified network ports at runtime.

CMD: This command specifies the command that will be run when a container is created from the image.

ENTRYPOINT: This command specifies the command that will be run when a container is created from the image, but unlike CMD it can be overridden by the command that runs at the container creation.

WORKDIR: This command sets the current working directory for any RUN, CMD, ENTRYPOINT, COPY, and ADD instructions that follow it in the Dockerfile.

USER: This command sets the user name or UID to use when running the image and for any RUN, CMD, and ENTRYPOINT instructions that follow it in the Dockerfile.

By using these commands, developers can specify all the steps needed to create a container image, and the Docker engine will handle the rest.

Task :

- **Create a Dockerfile for a simple web application (e.g. a Node.js or Python app)**

Step 1: make a directory and Clone the repository in your system using command `git clone <remote url>`

```

ubuntu@ip-172-31-90-202:~$ mkdir projects
ubuntu@ip-172-31-90-202:~$ cd projects/
ubuntu@ip-172-31-90-202:~/projects$ git clone https://github.com/shreys7/django-todo.git
Cloning into 'django-todo'...
remote: Enumerating objects: 288, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 288 (delta 0), reused 2 (delta 0), pack-reused 285
Receiving objects: 100% (288/288), 121.85 KiB | 4.87 MiB/s, done.
Resolving deltas: 100% (151/151), done.
ubuntu@ip-172-31-90-202:~/projects$

```

Step 2: View the contents of the repository which is cloned and change the directory to that repository

Step 3 : Install python3 and after that install django

```

ubuntu@ip-172-31-90-202:~/projects$ cd django-todo/
ubuntu@ip-172-31-90-202:~/projects/django-todo$ pip install django
Command 'pip' not found, but can be installed with:
sudo apt install python3-pip
ubuntu@ip-172-31-90-202:~/projects/django-todo$ sudo apt install python3-pip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done

```

```

ubuntu@ip-172-31-90-202:~/projects/django-todo$ pip3 install django
Defaulting to user installation because normal site-packages is not writeable
Collecting django
  Downloading Django-4.1.5-py3-none-any.whl (8.1 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.1/8.1 MB 44.1 MB/s eta 0:00:00
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.3-py3-none-any.whl (42 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.8/42.8 KB 5.8 MB/s eta 0:00:00
Collecting asgiref<4,>=3.5.2
  Downloading asgiref-3.6.0-py3-none-any.whl (23 kB)
Installing collected packages: sqlparse, asgiref, django
  WARNING: The script sqlformat is installed in '/home/ubuntu/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The script django-admin is installed in '/home/ubuntu/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed asgiref-3.6.0 django-4.1.5 sqlparse-0.4.3
ubuntu@ip-172-31-90-202:~/projects/django-todo$

```

Step 4 : Migrate python3 manage.py and run the server

```

ubuntu@ip-172-31-90-202:~/projects/django-todo$ python3 manage.py migrate
System check identified some issues:

WARNINGS:
todos.Todo: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
  HINT: Configure the DEFAULT_AUTO_FIELD setting or the TodosConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.

Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, todos
Running migrations:
  Applying auth.0012_alter_user_first_name_max_length... OK

```

```

ubuntu@ip-172-31-90-202:~/projects/django-todo$ python3 manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
todos.Todo: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
  HINT: Configure the DEFAULT_AUTO_FIELD setting or the TodosConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.

System check identified 1 issue (0 silenced).
January 22, 2023 - 12:04:24
Django version 4.1.5, using settings 'todoApp.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.

```

- Build the image using the Dockerfile and run the container

Step 1: Install the docker

```
ubuntu@ip-172-31-90-202:~/projects/django-todo$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (20.10.12-0ubuntu4).
0 upgraded, 0 newly installed, 0 to remove and 32 not upgraded.
ubuntu@ip-172-31-90-202:~/projects/django-todo$
```

Step 2 : Create a file called dockerfile and add configurations to it

```
ubuntu@ip-172-31-90-202:~/projects/django-todo$ vi dockerfile
```

```
FROM python:3
RUN pip install django==3.2

COPY . .

RUN python manage.py migrate

CMD ["python", "manage.py", "runserver", "0.0.0.0:8001"]
```

Step 3: Build the dockerfile using command `sudo docker build -t <app name>`

```
ubuntu@ip-172-31-90-202:~/projects/django-todo$ vi dockerfile
ubuntu@ip-172-31-90-202:~/projects/django-todo$ sudo docker build . -t todo-app
Sending build context to Docker daemon 579.1kB
Step 1/5 : FROM python:3
3: Pulling from library/python
0beef03cdalf: Already exists
f049f75f014e: Already exists
56261d0e6b05: Already exists
9bd150679dbd: Already exists
5b282ee9da04: Already exists
03f027d5e312: Already exists
db6ee1ace097: Pull complete
0a86d528f1ea: Pull complete
4cfb032ae58b: Pull complete
Digest: sha256:a3c0c6766535f85f18e7304d3a0111de5208d73935bcf1b024217005ad5ce195
Status: Downloaded newer image for python:3
----> b44268c8cbc0
Step 2/5 : RUN pip install django==3.2
----> Running in bd3dcf7492ee
```

Step 4 : Now Run dockerfile using command `sudo docker run -p <port num> <image num>` and to run to file in background use command `sudo docker run -d -p <port num> <image num>` , check it using command `docker ps`

```
ubuntu@ip-172-31-90-202:~$ sudo docker run -d -p 8001:8001 7a2a558341ec
a74a42cb908111f212571b1f95ed2715839de4db6b206f949cc2f21bd50de1a9
docker: Error response from daemon: driver failed programming external connectivity on endpoint pensive_black (a32aeada9aaf4d44da4d220ad1f594a36e8a080fe38cbf51): Bind for 0.0.0.0:8001 failed: port is already allocated.
ubuntu@ip-172-31-90-202:~$ docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED        STATUS        PORTS                               NAMES
4dbba77f0bd9   7a2a558341ec       "python manage.py ru..." 11 minutes ago Up 11 minutes  0.0.0.0:8001->8001/tcp, :::8001->8001/tcp  affectionate_elbakyan
3ad6df2dd6a2   7a2a558341ec       "python manage.py ru..." 35 minutes ago Up 35 minutes  0.0.0.0:8000->8000/tcp, :::8000->8000/tcp  admiring_payne
ubuntu@ip-172-31-90-202:~$
```

Note : U have to add port number in AWS ec2 Security Groups in Inbound Rule which your using port number, here I m using 8001 so I have add it.

sgr-070e4e9386a775d6d

Custom TCP ▼ TCP 8001 Custom ▼

0.0.0.0/0 X

0.0.0.0/0 X

Q

Delete

- **Verify that the application is working as expected by accessing it in a web browser**

← → ↻ Not secure | 54.234.158.20:8001/todos/

Gmail YouTube Maps Course | Introductio... (2383) Linux Full Co...

Todo List

Do laundry Add

- ☐ Send Resume Google now !
- ☐ Hacktoberfest Updates

- **Push the image to a public or private repository (e.g. Docker Hub)**

Step 1: Create a account in Docker Hub

Step 2: After creating account go to your terminal and login to docker using command docker login and entre your username & password of your docker hub

```
ubuntu@ip-172-31-90-202:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://docs.docker.com/docker-hub/sign-up/
Username: basanagoudapatil
Password:
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-172-31-90-202:~$ docker image
```

Step 3 : Check for docker images

```
ubuntu@ip-172-31-90-202:~/projects/django-todo$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
basanagoudapatil/7a2a558341ec  latest     7a2a558341ec  About an hour ago  984MB
todo-app            latest     7a2a558341ec  About an hour ago  984MB
python              3.9        b05aba0a4b15  4 days ago     915MB
python              3          b44268c8cbc0  4 days ago     932MB
jenkins/jenkins     latest     682e28744673  4 days ago     469MB
hello-world         latest     feb5d9fea6a5  16 months ago  13.3kB
django              latest     eb40dcf64078  6 years ago     436MB
ubuntu@ip-172-31-90-202:~/projects/django-todo$
```

Step 4 : Push a image in docker hub using command docker push <image_name>

basanagoudapatil Search by repository name All Content Create repository

basanagoudapatil / 7a2a558341ec
Contains: Image | Last pushed: a few seconds ago

Not Scanned ☆ 0 📄 0 Public

