**||DOCKER||**

Docker Installation:

• $ sudo apt-get update -y

• $ sudo apt-get upgrade -y

• $ sudo apt-get remove docker docker-engine docker.io containerd runc

• $ sudo apt install docker.io

• $ sudo systemctl start docker

• $ sudo systemctl status docker

• $ sudo systemctl enable docker

• $ sudo systemctl status docker

• $ docker –version

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

Creating Dockerfile & Docker Image

• cd database/

• touch Dockerfile

• sudo vi Dockerfile

FROM mysql:latest

ENV MYSQL\_DATABASE my\_DB

ENV MYSQL\_ROOT\_PASSWORD root123

ENV MYSQL\_USER test

ENV MYSQL\_PASSWORD admin123

EXPOSE 3306

• sudo docker build -t database .

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

• cd backend/

• git clone https://github.com/peyyala7hills/new\_chatapp.git

• sudo vi Dockerfile

FROM python:3.6

RUN apt-get update

RUN apt-get install --yes python3-pip

RUN mkdir /new\_chatapp

COPY /new\_chatapp-1 /new\_chatapp

WORKDIR /new\_chatapp

RUN pip3 install -r requirements.txt

RUN pip3 install mysqlclient

EXPOSE 8000

WORKDIR /new\_chatapp/fundoo/

ENTRYPOINT python manage.py migrate && python3 manage.py runserver 0.0.0.0:8000

• sudo docker build -t backend .

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

• cd frontend/

• Sudo vi Dockerfile

FROM ubuntu

RUN apt-get update

RUN apt-get install -y nginx

COPY nginx.conf /etc/nginx/sites-available/fundoo

RUN ln -s /etc/nginx/sites-available/fundoo /etc/nginx/sites-enabled

RUN rm -rf /etc/nginx/sites-enabled/default

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

• sudo vi nginx.conf

server {

listen 80;

server\_name \_default;

#location = /favicon.ico { access\_log off; log\_not\_found off; }

location /static/ {

root /new\_chatapp/fundoo;

}

location / {

include proxy\_params;

proxy\_pass http://backend:8000;

}

}

• sudo docker build -t frontend .

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

Creating Docker Container:

Note: Manually running the containers

Manually linking database to backend and backend to frontend

• sudo docker run -it --name database -d database-image-id

• sudo docker run -it --name backend --link database -d backend-imge-id

• sudo docker run -it --name frontend -p 80:80 --link backend -d frontend-image-id

• sudo docker ps

• to get inside container

• sudo docker exec -it containerName/containerID /bin/bash

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

other commands:

• to delete image

sudo docker rmi -f imageID/imageName

• to delete container

sudo docker rm -f containerID/contaierName

• to check the IP Port of container

sudo docker exec -it containerID/containerName cat /etc/hosts

#######################################################################################Docker-compose file################################################

We are using 3 services in one file.

sudo vi docker-compose.yml

version: '2'

services:

database:

build:

dockerfile: Dockerfile

context: /home/ubuntu/database

image: database

container\_name: databasecompose

backend:

build:

dockerfile: Dockerfile

context: /home/ubuntu/backend

image: backend

container\_name: backendcompose

links:

- database

depends\_on:

* database

frontend:

build:

dockerfile: Dockerfile

context: /home/ubuntu/frontend

image: frontend

container\_name: frontendcompose

ports:

- 80:80

links:

- backend

depends\_on:

* backend

#========================= Central Management =================================

# Beats is configured under central management, you can define most settings

# from the Kibana UI. You can update this file to configure the settings that

# are not supported by Kibana Beats management.

management:

enabled: true

period: 1m0s

events\_reporter:

period: 30s

max\_batch\_size: 1000

access\_token: ${management.accesstoken}

kibana:

protocol: http

host: 34.170.156.9:5601

ssl: null

timeout: 10s

ignoreversion: true

blacklist:

output: console|file

#================================ General =====================================

# The name of the shipper that publishes the network data. It can be used to group

# all the transactions sent by a single shipper in the web interface.

#name:

# The tags of the shipper are included in their own field with each

# transaction published.

#tags: ["service-X", "web-tier"]

# Optional fields that you can specify to add additional information to the

# output.

#fields:

# env: staging

#================================ Logging =====================================

# Sets log level. The default log level is info.

# Available log levels are: error, warning, info, debug

#logging.level: debug

# At debug level, you can selectively enable logging only for some components.

# To enable all selectors use ["\*"]. Examples of other selectors are "beat",

# "publish", "service".

#logging.selectors: ["\*"]

#============================== Xpack Monitoring ===============================

# filebeat can export internal metrics to a central Elasticsearch monitoring

# cluster. This requires xpack monitoring to be enabled in Elasticsearch. The

# reporting is disabled by default.

# Set to true to enable the monitoring reporter.

#monitoring.enabled: false

# Uncomment to send the metrics to Elasticsearch. Most settings from the

# Elasticsearch output are accepted here as well.

# Note that the settings should point to your Elasticsearch \*monitoring\* cluster.

# Any setting that is not set is automatically inherited from the Elasticsearch

# output configuration, so if you have the Elasticsearch output configured such

# that it is pointing to your Elasticsearch monitoring cluster, you can simply

# uncomment the following line.

#monitoring.elasticsearch: