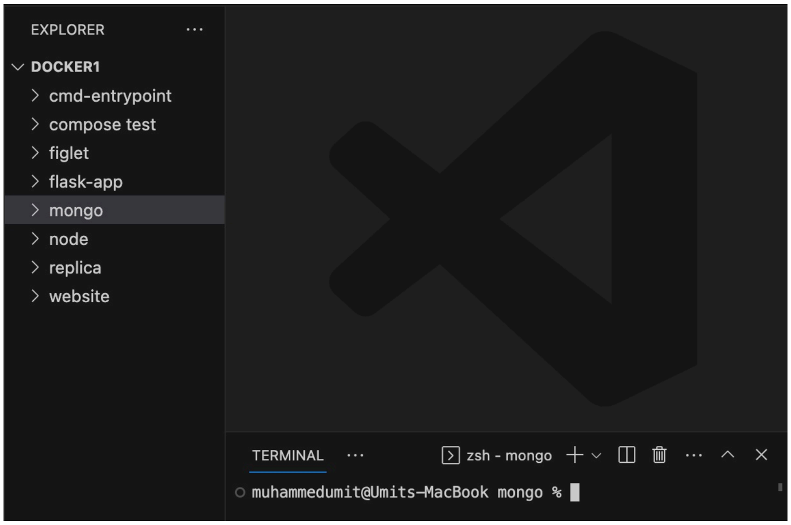
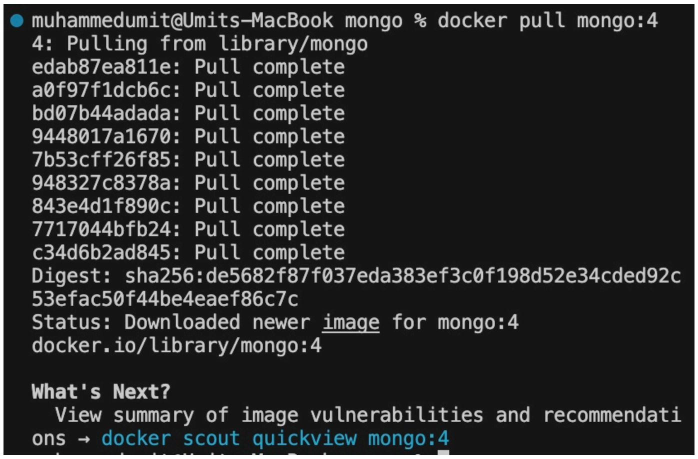
Docker Compose with Mongo and Mongo-Express

I create a mongo folder on VS Code and opened integrated terminal.



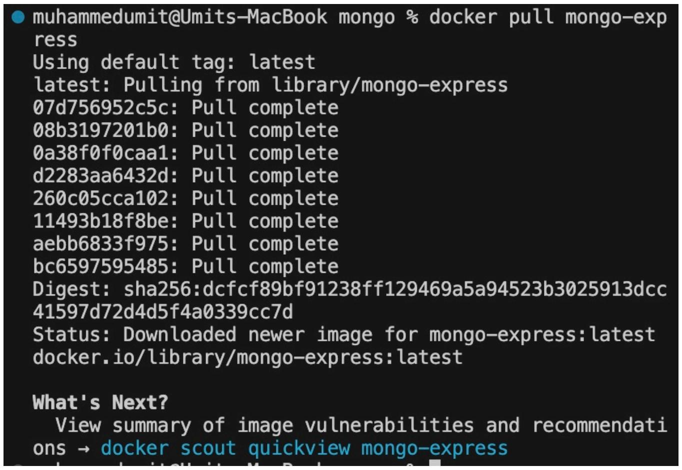
```
muhammedumit@Umits-MacBook mongo %
```

I download mongo version 4 by using ‘docker pull mongo:4’ command.



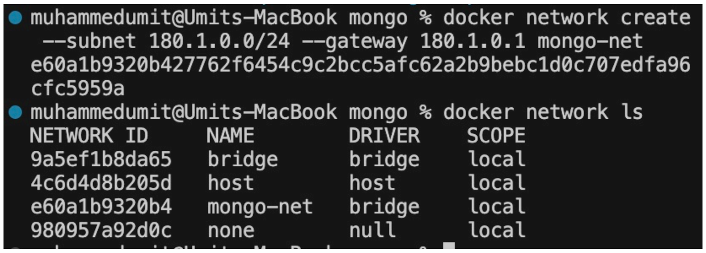
```
muhammedumit@Umits-MacBook mongo % docker pull mongo:4
4: Pulling from library/mongo
edab87ea811e: Pull complete
a0f97f1dcb6c: Pull complete
bd07b44adada: Pull complete
9448017a1670: Pull complete
7b53cff26f85: Pull complete
948327c8378a: Pull complete
843e4d1f890c: Pull complete
7717044bfb24: Pull complete
c34d6b2ad845: Pull complete
Digest: sha256:de5682f87f037eda383ef3c0f198d52e34cded92c
53efac50f44be4eaef86c7c
Status: Downloaded newer image for mongo:4
docker.io/library/mongo:4

What's Next?
  View summary of image vulnerabilities and recommendati
ons → docker scout quickview mongo:4
```

I download mongo-express the latest version by using ‘docker pull mongo-express’ command.



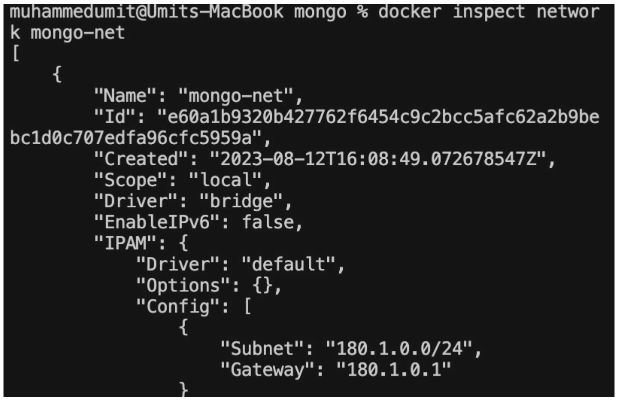
```
muhammedumit@Umits-MacBook mongo % docker pull mongo-exp
ress
Using default tag: latest
latest: Pulling from library/mongo-express
07d756952c5c: Pull complete
08b3197201b0: Pull complete
0a38f0f0caa1: Pull complete
d2283aa6432d: Pull complete
260c05cca102: Pull complete
11493b18f8be: Pull complete
aebb6833f975: Pull complete
bc6597595485: Pull complete
Digest: sha256:dcfcf89bf91238ff129469a5a94523b3025913dcc
41597d72d4d5f4a0339cc7d
Status: Downloaded newer image for mongo-express:latest
docker.io/library/mongo-express:latest

What's Next?
  View summary of image vulnerabilities and recommendati
ons → docker scout quickview mongo-express
```

I create a network for the database containers by using ‘docker network create —subnet 180.1.0.0/24 —gateway 180.1.0.1 mongo-net’ command.



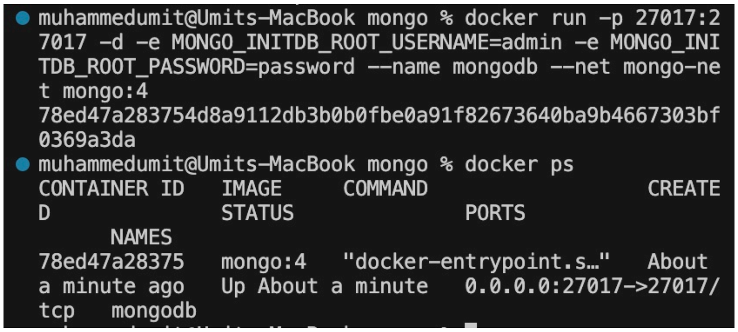
```
● muhammedumit@Umits-MacBook mongo % docker network create
   --subnet 180.1.0.0/24 --gateway 180.1.0.1 mongo-net
  e60a1b9320b427762f6454c9c2bcc5afc62a2b9bebc1d0c707edfa96
  cfc5959a
● muhammedumit@Umits-MacBook mongo % docker network ls
  NETWORK ID     NAME        DRIVER    SCOPE
  9a5ef1b8da65   bridge      bridge    local
  4c6d4d8b205d   host        host      local
  e60a1b9320b4   mongo-net   bridge    local
  980957a92d0c   none        null      local
```

In order to inspect the network to see the CIDR and gateway, I use ‘docker inspect network mongo-net’ command.



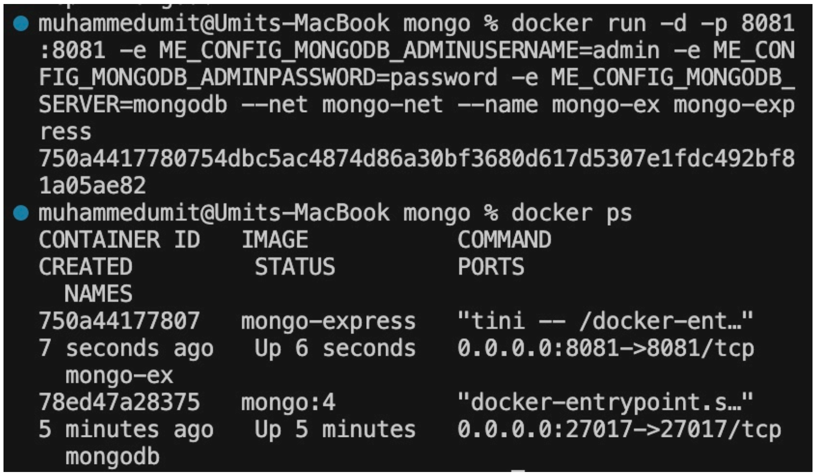
```
muhammedumit@Umits-MacBook mongo % docker inspect networ
k mongo-net
[
    {
        "Name": "mongo-net",
        "Id": "e60a1b9320b427762f6454c9c2bcc5afc62a2b9be
bc1d0c707edfa96cfc5959a",
        "Created": "2023-08-12T16:08:49.072678547Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "180.1.0.0/24",
                    "Gateway": "180.1.0.1"
                }
```

Mongo works 27017 ports. -d detached mode, -e environment, —network will be ‘mongo-net’. The code is ‘ docker run -p 27017:27017 -d -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INITDB_ROOT_PASSWORD=password —name mongodb —net mongo-net mongo:4’



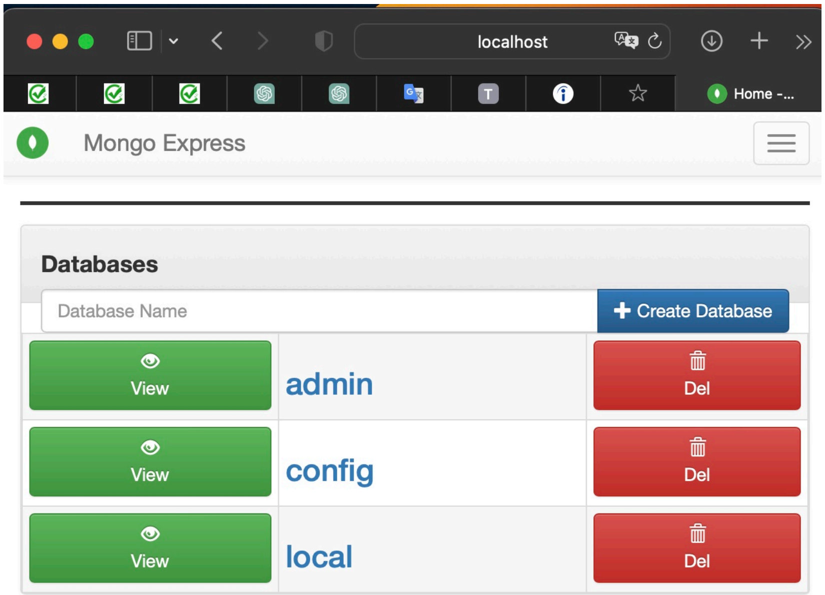
```
● muhammedumit@Umits-MacBook mongo % docker run -p 27017:2
  7017 -d -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INI
  TDB_ROOT_PASSWORD=password --name mongodb --net mongo-ne
  t mongo:4
  78ed47a283754d8a9112db3b0b0fbe0a91f82673640ba9b4667303bf
  0369a3da
● muhammedumit@Umits-MacBook mongo % docker ps
  CONTAINER ID   IMAGE     COMMAND                  CREATE
  D              STATUS              PORTS
        NAMES
  78ed47a28375   mongo:4   "docker-entrypoint.s…"   About
  a minute ago   Up About a minute   0.0.0.0:27017->27017/
  tcp   mongodb
```

Mongo-express uses port 8081. -d detached mode, -e environment, —network will be ‘mongo-net’. The code is ‘docker run -d -p 8081:8081 -e ME_CONFIG_MONGODB_ADMINUSERNAME=admin -e ME_CONFIG_MONGODB_ADMINPASSWORD=password -e ME_CONFIG_MONGODB_SERVER=mongodb --net mongo-net --name mongo-ex mongo-express’.



```
● muhammedumit@Umits-MacBook mongo % docker run -d -p 8081
  :8081 -e ME_CONFIG_MONGODB_ADMINUSERNAME=admin -e ME_CON
  FIG_MONGODB_ADMINPASSWORD=password -e ME_CONFIG_MONGODB_
  SERVER=mongodb --net mongo-net --name mongo-ex mongo-exp
  ress
  750a4417780754dbc5ac4874d86a30bf3680d617d5307e1fdc492bf8
  1a05ae82
● muhammedumit@Umits-MacBook mongo % docker ps
  CONTAINER ID   IMAGE           COMMAND
  CREATED         STATUS         PORTS
    NAMES
  750a44177807   mongo-express   "tini -- /docker-ent…"
  7 seconds ago   Up 6 seconds   0.0.0.0:8081->8081/tcp
    mongo-ex
  78ed47a28375   mongo:4         "docker-entrypoint.s…"
  5 minutes ago   Up 5 minutes   0.0.0.0:27017->27017/tcp
    mongodb
```

I use mongo-express to connect to mongo server. I can see databases, create databases.. by using localhost:8081.

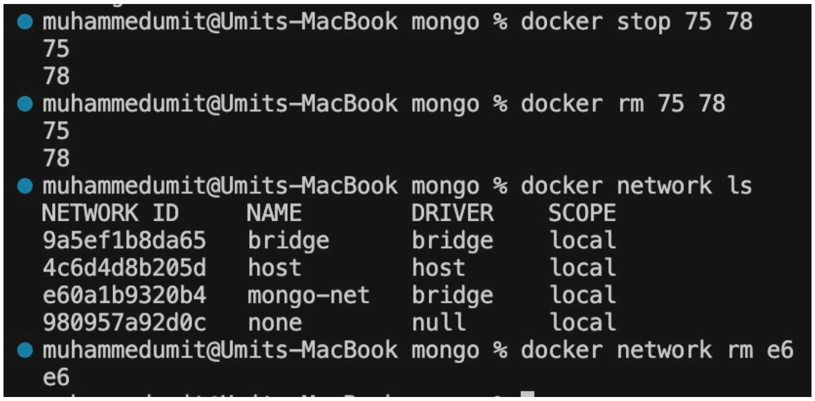


Databases

Database Name | + Create Database

View	admin	Del
View	config	Del
View	local	Del

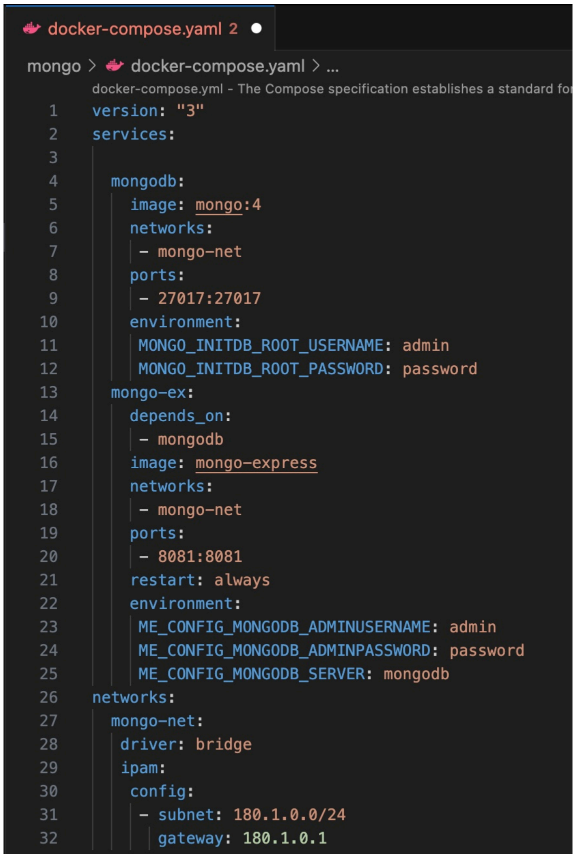
I remove the containers and network.



```
● muhammedumit@Umits-MacBook mongo % docker stop 75 78
  75
  78
● muhammedumit@Umits-MacBook mongo % docker rm 75 78
  75
  78
● muhammedumit@Umits-MacBook mongo % docker network ls
  NETWORK ID     NAME        DRIVER    SCOPE
  9a5ef1b8da65   bridge      bridge    local
  4c6d4d8b205d   host        host      local
  e60a1b9320b4   mongo-net   bridge    local
  980957a92d0c   none        null      local
● muhammedumit@Umits-MacBook mongo % docker network rm e6
  e6
```

Now I will create the same structure on the previous step, this time I use Docker Compose

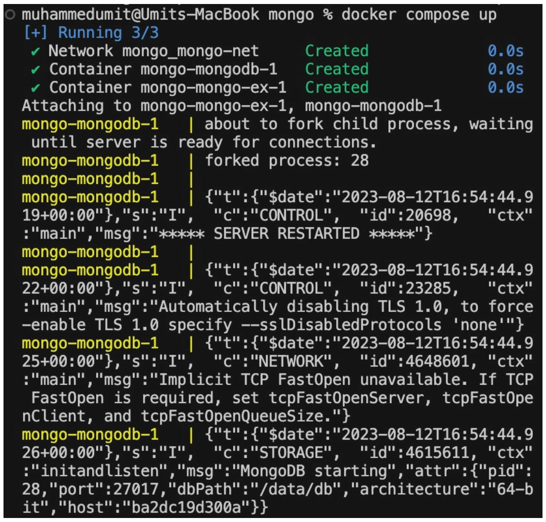
I create a file named ‘docker-compose.yaml’ and write the code below.


docker-compose.yaml 2 ●
mongo > docker-compose.yaml > ...
docker-compose.yml - The Compose specification establishes a standard for

```yaml
version: "3"
services:

  mongodb:
    image: mongo:4
    networks:
     - mongo-net
    ports:
     - 27017:27017
    environment:
     MONGO_INITDB_ROOT_USERNAME: admin
     MONGO_INITDB_ROOT_PASSWORD: password
  mongo-ex:
    depends_on:
     - mongodb
    image: mongo-express
    networks:
     - mongo-net
    ports:
     - 8081:8081
    restart: always
    environment:
     ME_CONFIG_MONGODB_ADMINUSERNAME: admin
     ME_CONFIG_MONGODB_ADMINPASSWORD: password
     ME_CONFIG_MONGODB_SERVER: mongodb
networks:
  mongo-net:
   driver: bridge
   ipam:
    config:
     - subnet: 180.1.0.0/24
       gateway: 180.1.0.1
```

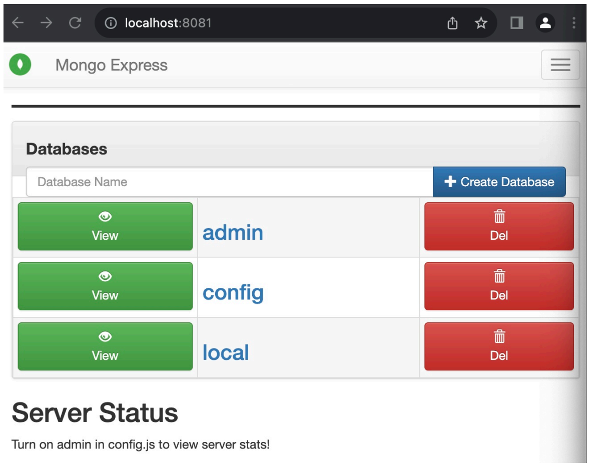
The same structure I used before. The differences are;
depends_on: mongodb meaning is that wait for mongodb to start and then start running.
Restart:always meaning is if mongo-express gives an error, don’t stop running.

Now, ‘docker compose up’



```
○ muhammedumit@Umits-MacBook mongo % docker compose up
 [+] Running 3/3
  ✔ Network mongo_mongo-net     Created              0.0s
  ✔ Container mongo-mongodb-1   Created              0.0s
  ✔ Container mongo-mongo-ex-1  Created              0.0s
Attaching to mongo-mongo-ex-1, mongo-mongodb-1
mongo-mongodb-1   | about to fork child process, waiting
 until server is ready for connections.
mongo-mongodb-1   | forked process: 28
mongo-mongodb-1   |
mongo-mongodb-1   | {"t":{"$date":"2023-08-12T16:54:44.9
19+00:00"},"s":"I",  "c":"CONTROL",  "id":20698,   "ctx"
:"main","msg":"***** SERVER RESTARTED *****"}
mongo-mongodb-1   |
mongo-mongodb-1   | {"t":{"$date":"2023-08-12T16:54:44.9
22+00:00"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx"
:"main","msg":"Automatically disabling TLS 1.0, to force
-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
mongo-mongodb-1   | {"t":{"$date":"2023-08-12T16:54:44.9
25+00:00"},"s":"I",  "c":"NETWORK",  "id":4648601, "ctx"
:"main","msg":"Implicit TCP FastOpen unavailable. If TCP
 FastOpen is required, set tcpFastOpenServer, tcpFastOpe
nClient, and tcpFastOpenQueueSize."}
mongo-mongodb-1   | {"t":{"$date":"2023-08-12T16:54:44.9
26+00:00"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx"
:"initandlisten","msg":"MongoDB starting","attr":{"pid":
28,"port":27017,"dbPath":"/data/db","architecture":"64-b
it","host":"ba2dc19d300a"}}
```

You can see mongo by using mongo net port 8081.



Databases

View	admin	Del
View	config	Del
View	local	Del

Server Status

Turn on admin in config.js to view server stats!