**CREATING A DOCKERFILE OF MONGO AND MONGO-EXPRESS AND RUN A CONTAINER**

**Step:1: -**

***Creating a Dockerfile of Mongo:***

sudo vi Dockerfile.db

FROM mongo

ENV MONGO\_INITDB\_ROOT\_USERNAME=admin \

    MONGO\_INITDB\_ROOT\_PASSWORD=password

EXPOSE 27017

🡺 save the file

***Create another Dockerfile of mongo-express:***

sudo vi Dockerfile.web

FROM mongo-express

ENV ME\_CONFIG\_MONGODB\_ADMINUSERNAME=admin \

    ME\_CONFIG\_MONGODB\_ADMINPASSWORD=password \

    ME\_CONFIG\_MONGODB\_SERVER=mongodb

EXPOSE 8081

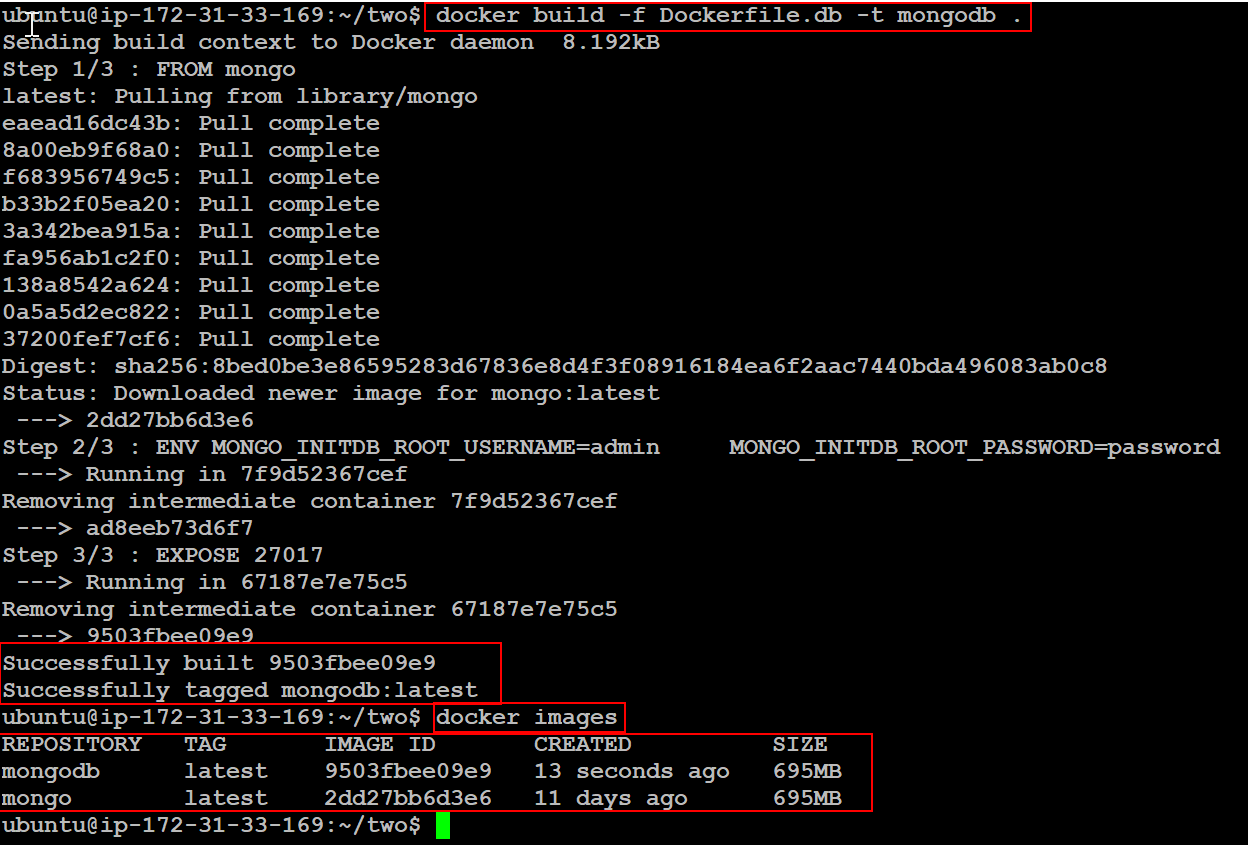
🡺 save the file.

**Step:2:-**

***Build the Dockerfiles of mongo and mongo-express:***

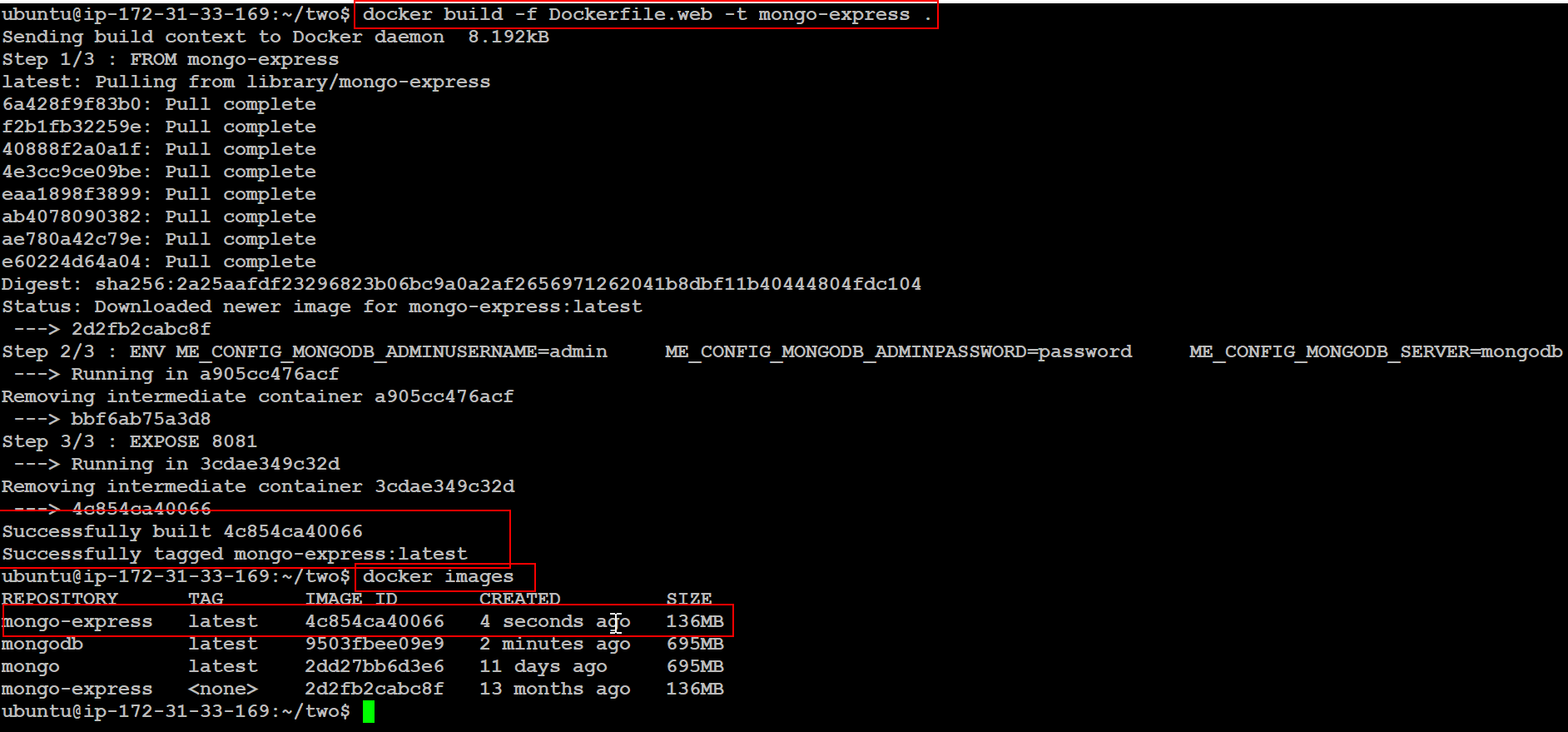
***This one for mongo:***

dockerbuild -f Dockerfile.db -t mongodb .



***This one for mongo-express:***

dockerbuild -f Dockerfile.web -t mongo-express .



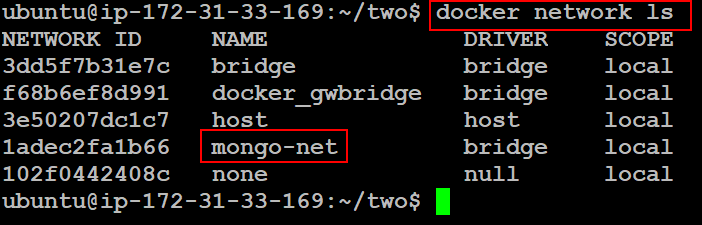
**Step:3:-**

***Create a network and connect the images with docker run:***

docker network create mongo-net

***to see the docker networks the command is:***

docker network ls



**Step:4:-**

***Create the docker containers of mongodb and mongo-express by using the build images:***

**Note: When we are creating a container of mongo and mongo-express we are attaching the network for both the container on that time only we can able to see in the browser of mongo Database.**

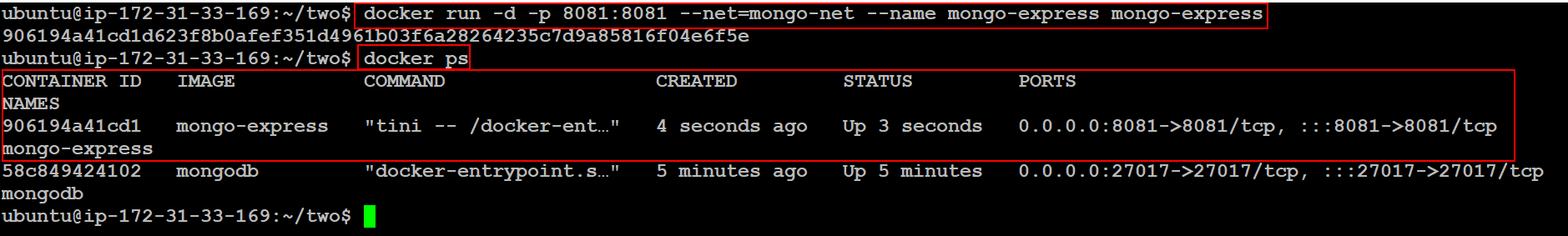
***First running the container of Mongodb:***

docker run -d -p 27017:27017 --net=mongo-net --name mongodb mongodb



***Second running the container of Mongo-express:***

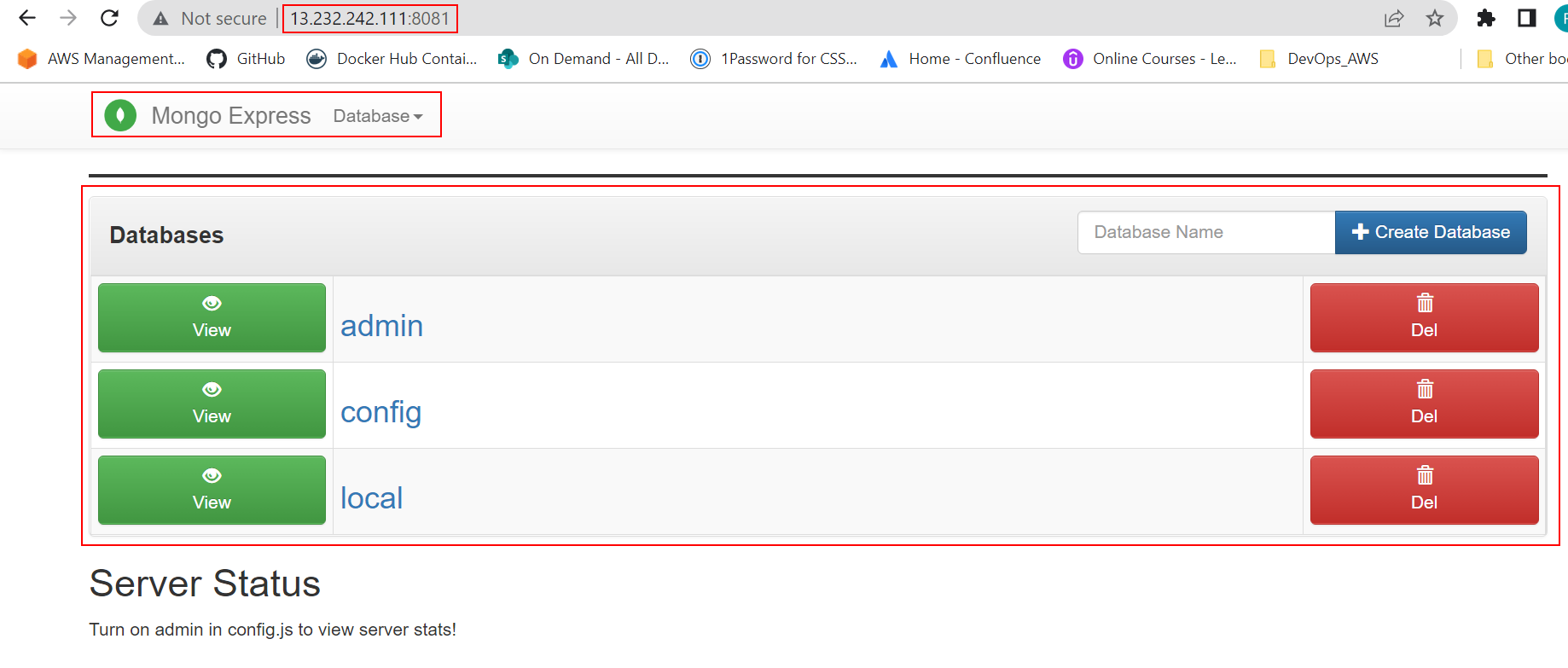
docker run -d -p 8081:8081 --net=mongo-net --name mongo-express mongo-express



**Step:5:-**

Now check the browser of using IP address of the Instance with the port of 8081

**Note: First make sure to add the specific port that need to have access to open the browser in AWS Security-Groups of Inbound-Rules.**



This is the database we have the successful access.

**[AND]**

**Step:1:-**

Without creating the Dockerfiles of mongo and mongo-express we can do to pull from DockerHub of two images and run those images with attaching of network for to create container like this:

***Pulling the images from DockerHub:***

docker pull mongo

docker pull mongo-express

**Step:2:-**

***with these images creating a container:***

docker run -d -p 27017:27017 -e MONGO\_INITDB\_ROOT\_USERNAME=admin -e MONGO\_INITDB\_ROOT\_PASSWORD=password --name mongodb --net mongo-net mongo

docker run -d -p 8081:8081 -e ME\_CONFIG\_MONGODB\_ADMINUSERNAME=admin -e ME\_CONFIG\_MONGODB\_ADMINPASSWORD=password --net mongo-net --name mongo-express -e ME\_CONFIG\_MONGODB\_SERVER=mongodb mongo-express

in here we are giving the network and environment variables as well for running a both containers to get access of Databases.

**[AND]**

**Trying with Docker-Compose file also**

**Step:1: -**

***Creating a file of docker-compose:***

mkdir two

cd two/

sudo vi docker-compose.yaml

version: '3'

services:

  mongodb:

    image: mongo:latest

    ports:

      - 27017:27017

    environment:

      - MONGO\_INITDB\_ROOT\_USERNAME=admin

      - MONGO\_INITDB\_ROOT\_PASSWORD=password

  mongo-express:

    image: mongo-express:latest

    restart: always

    depends\_on:

      - mongodb

    ports:

      - 8082:8081

    environment:

      - ME\_CONFIG\_MONGODB\_ADMINUSERNAME=admin

      - ME\_CONFIG\_MONGODB\_ADMINPASSWORD=password

      - ME\_CONFIG\_MONGODB\_SERVER=mongodb

🡺 save the file.

**Step:2: -**

***To create the container by using a docker-compose command:***

***docker-compose up*** 🡪 It will show the details every step and also those steps we can able to see by using the command of docker logs

***docker-compose up -d*** 🡪 it will show just the containers are created that’s it.

