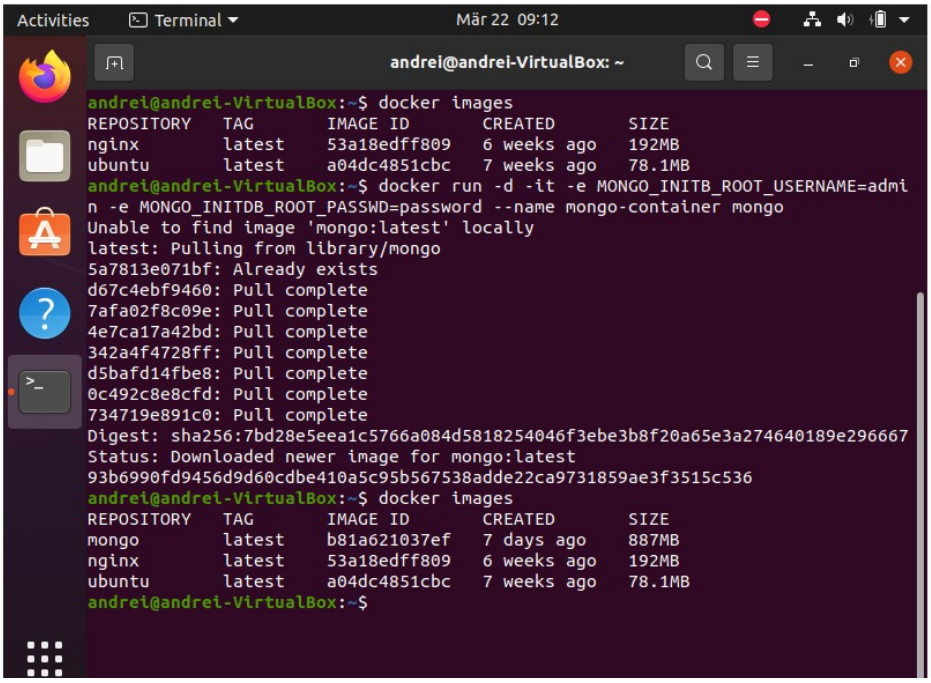
Задание 1.

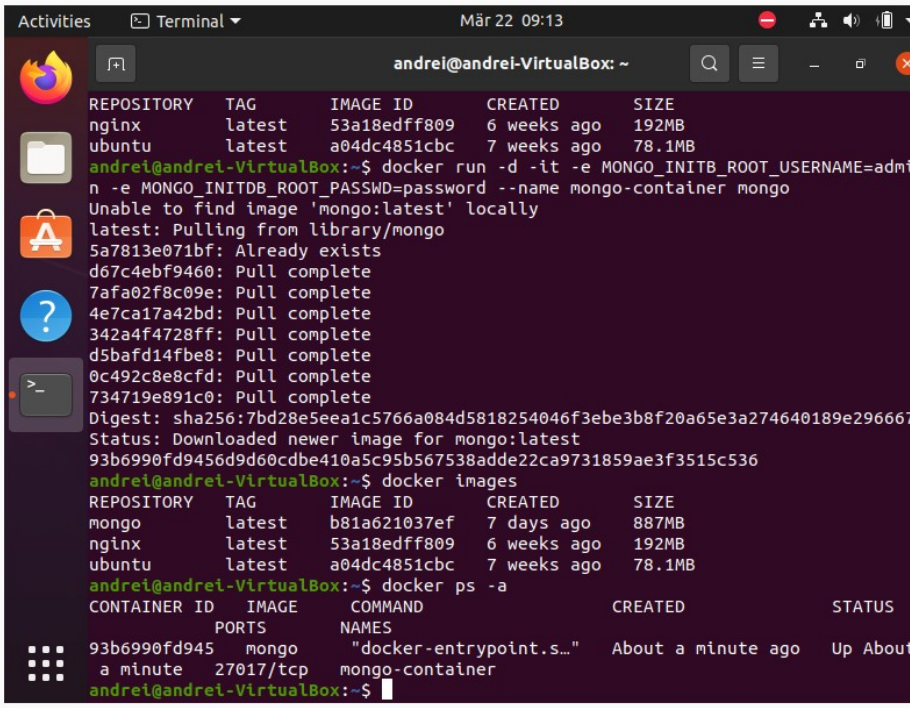
Запускаем контейнер mongo-container на основе образа mongo

```
docker run -d -it -e MONGO_INITDB_ROOT_USERNAME=admin \
      -e MONGO_INITDB_ROOT_PASSWD=password \
      --name mongo-container mongo
```



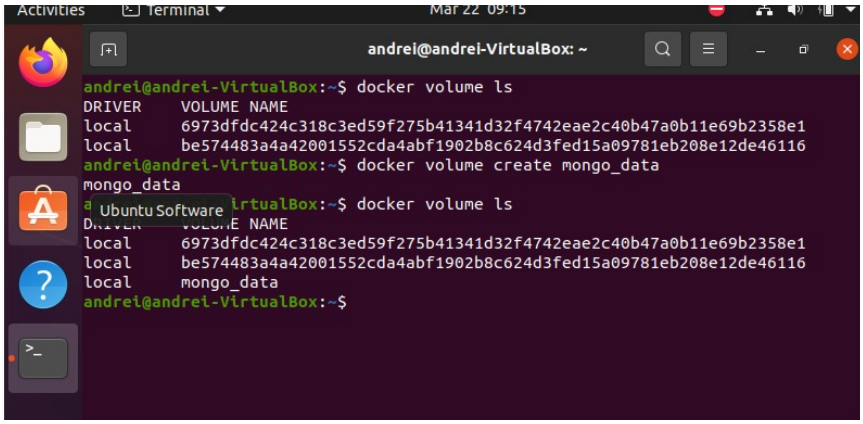
Проверяем, что контейнер mongo-container запущен

```
docker ps -a
```



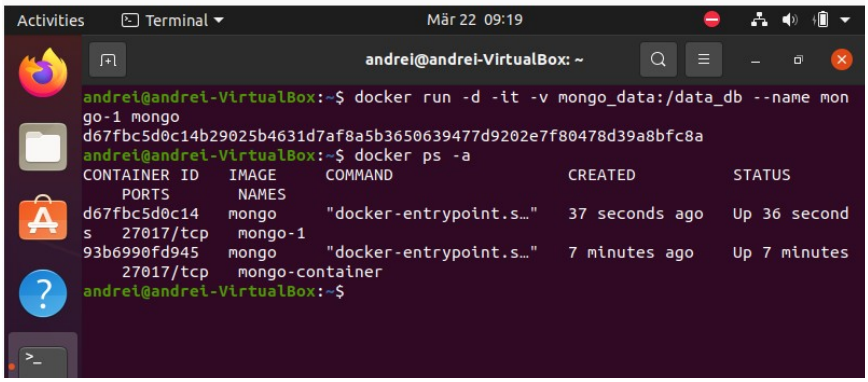
Создаем том mongo_data и проверяем его

```
docker volume create mongo_data
docker volume ls
```



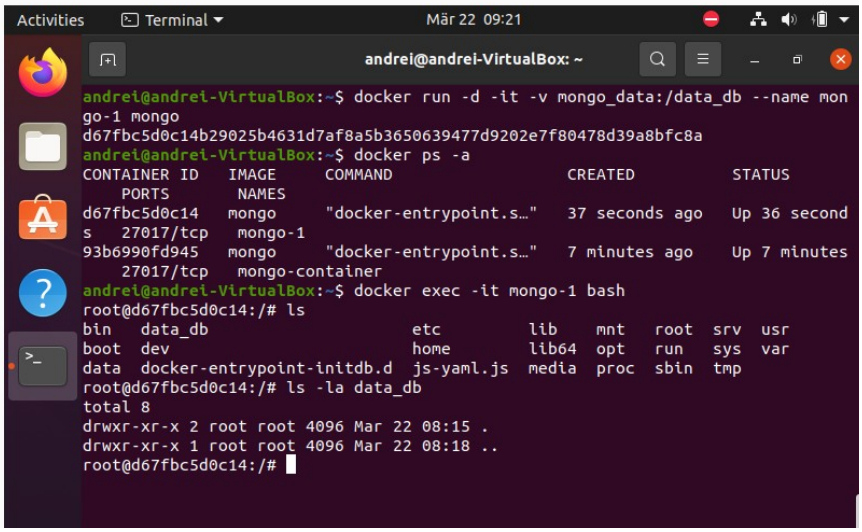
Запускаем новый контейнер и подключаем к нему том mongo_data

```
docker run -d -it –v mongo_data:/data_db ––name mongo-1 mongo
```



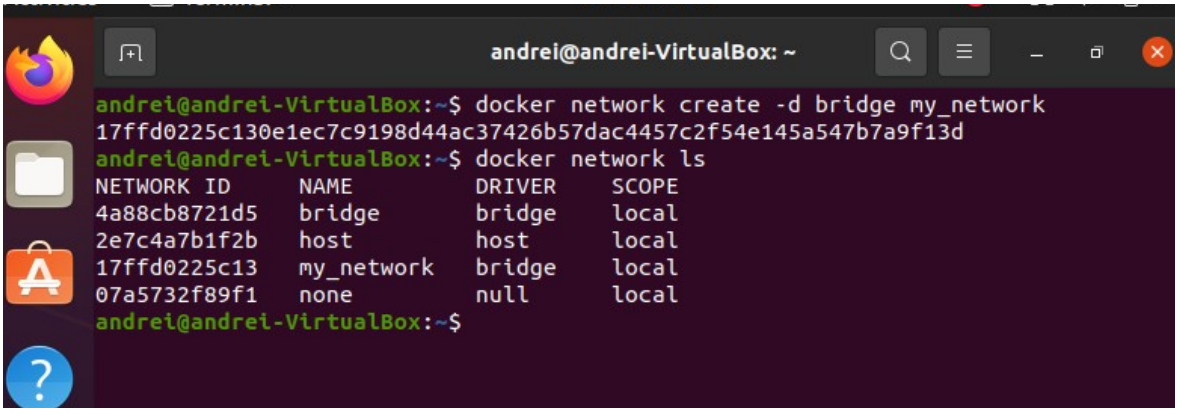
Заходим в контейнер mongo-1 и проверяем подключение docker volume

```
docker exec it mongo-1 bash
```



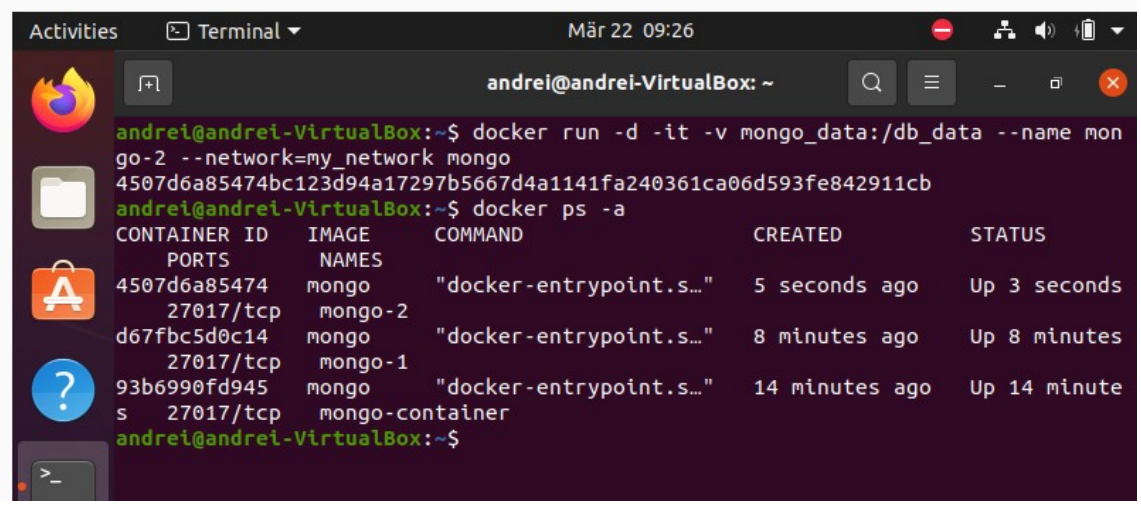
Создаем новую сеть

docker network create -d bridge my_network



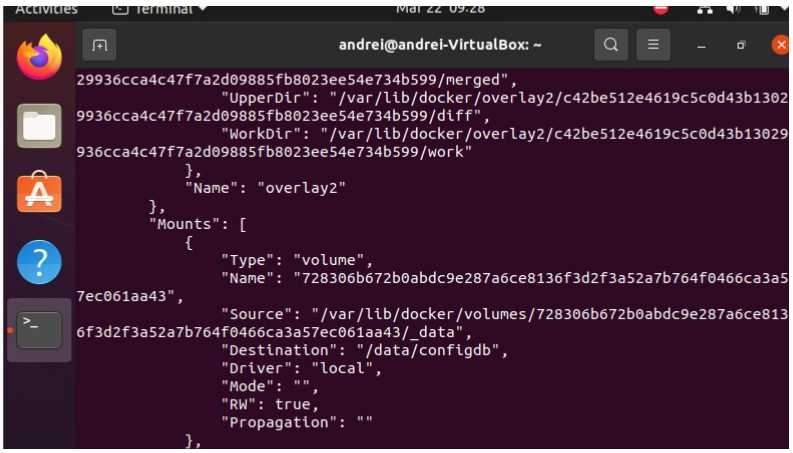
Создаем контейнер mongo-2 из образа mongo и подключим к нему my_network и том mongo_data

**docker run -d -it -v mongo_data:/db_data –-name mongo-2 **
–-network=my_network mongo

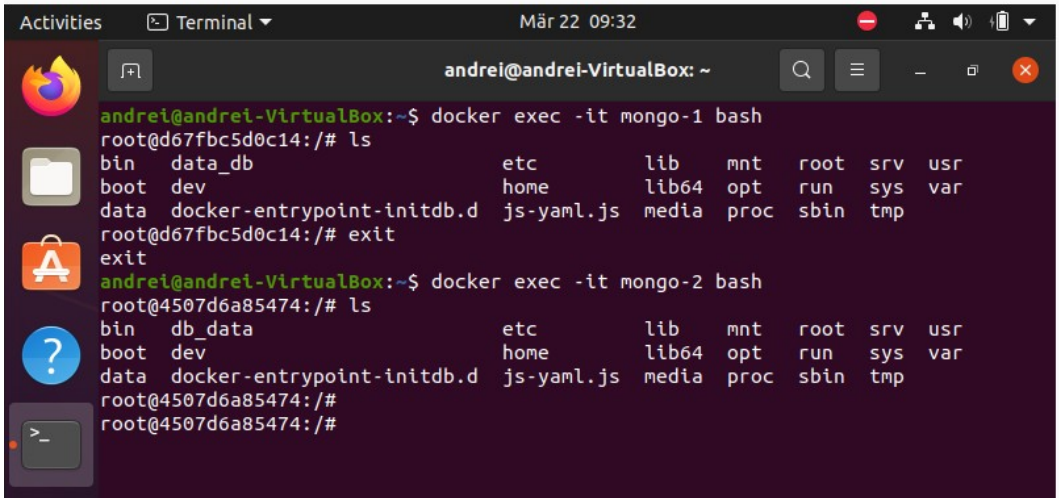


Проверяем контейнер mongo-2, его сеть my_network и docker volume mongo_data

docker inspect mongo-2



Запущенные контейнеры **mongo-1** и **mongo-2** смотрят на один том mongo_data.

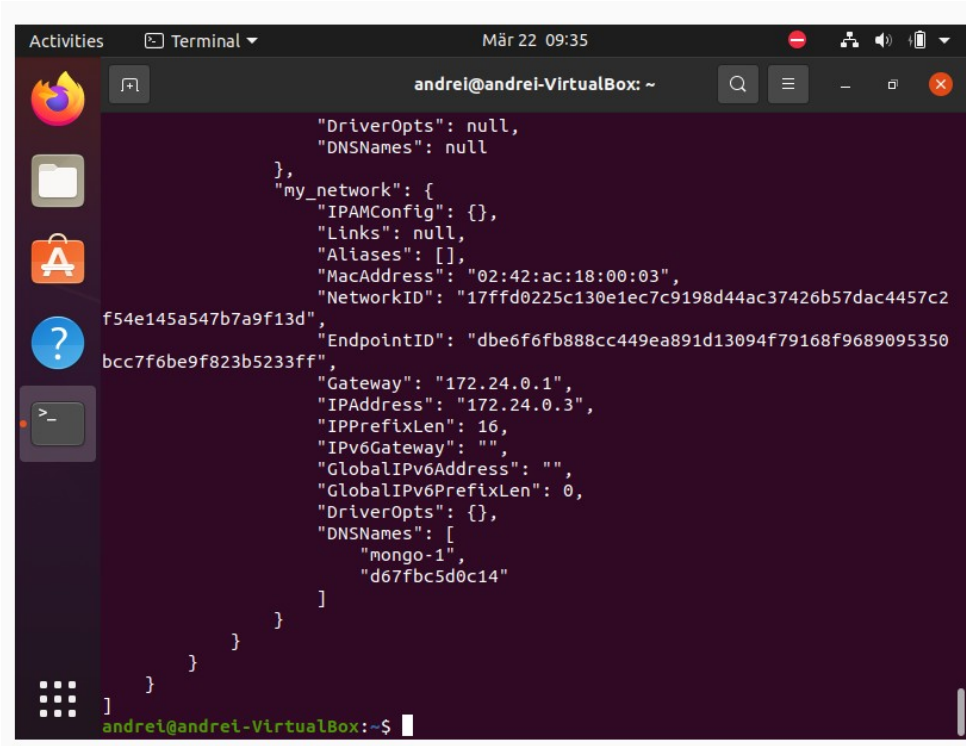


Подключим контейнер mongo-1 также к сети my_network

docker network connect my_network mongo-1

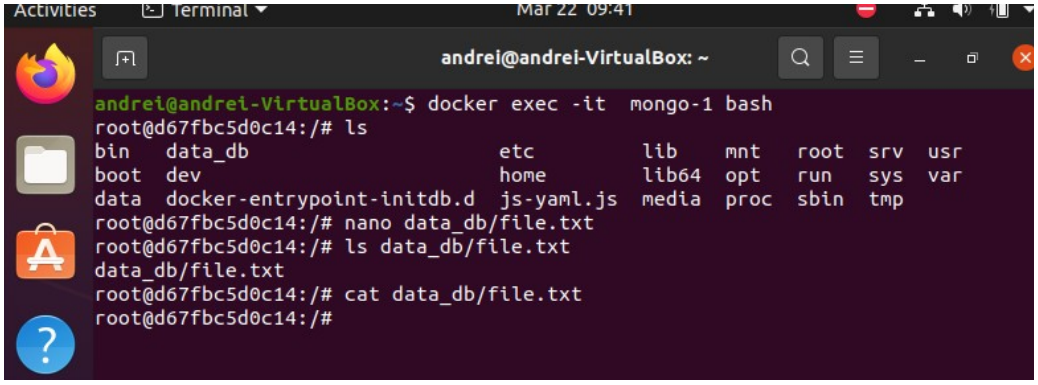
и выполняем проверку

docker inspect mongo-1

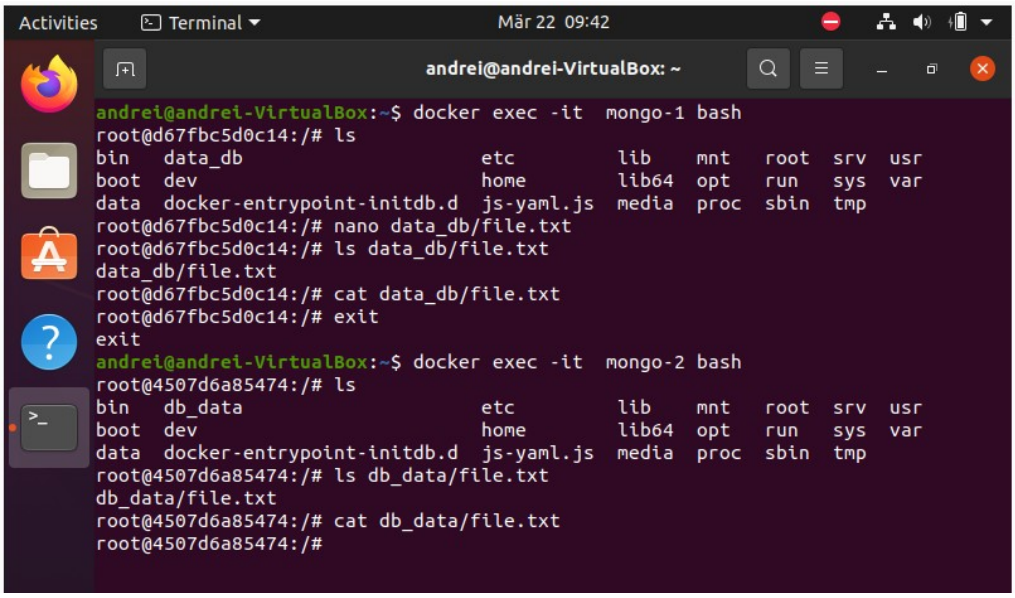


Создаем в контейнере mongo-1, в подключенном volume, file.txt

docker exec -it mongo-1 bash
nano data_db/file.txt

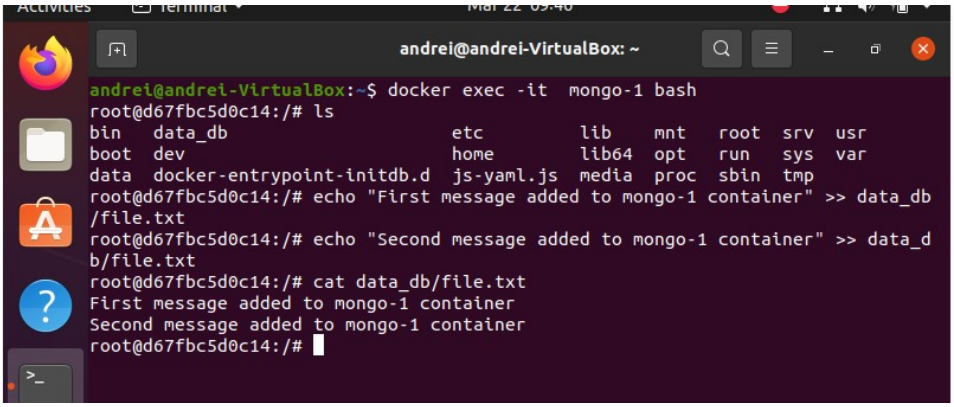


Файл file.txt появляется и в контейнере mongo-2



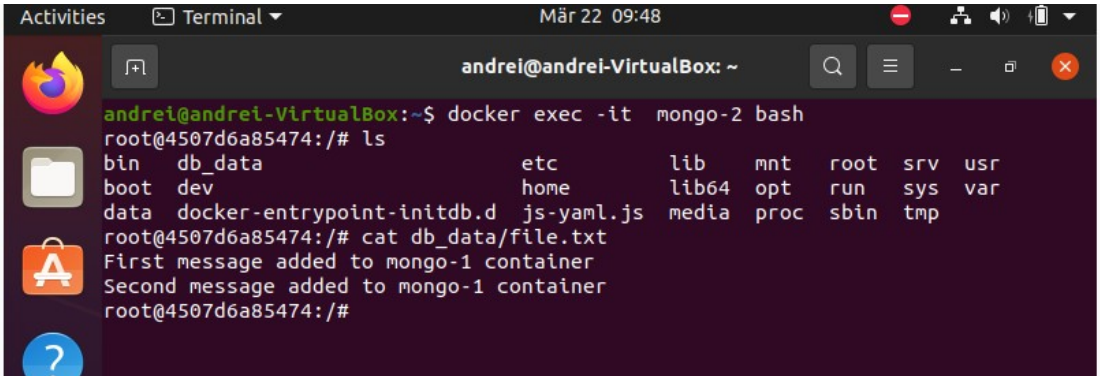
Записываем несколько сообщений в созданный файл file.txt в mongo-1

docker exec -it mongo-1 bash
echo “First message added to mongo-1 container” >> data_db/file.txt
echo “Second message added to mongo-1 container” >> data_db/file.txt



Проверяем содержание файла file.txt в контейнере mongo-2

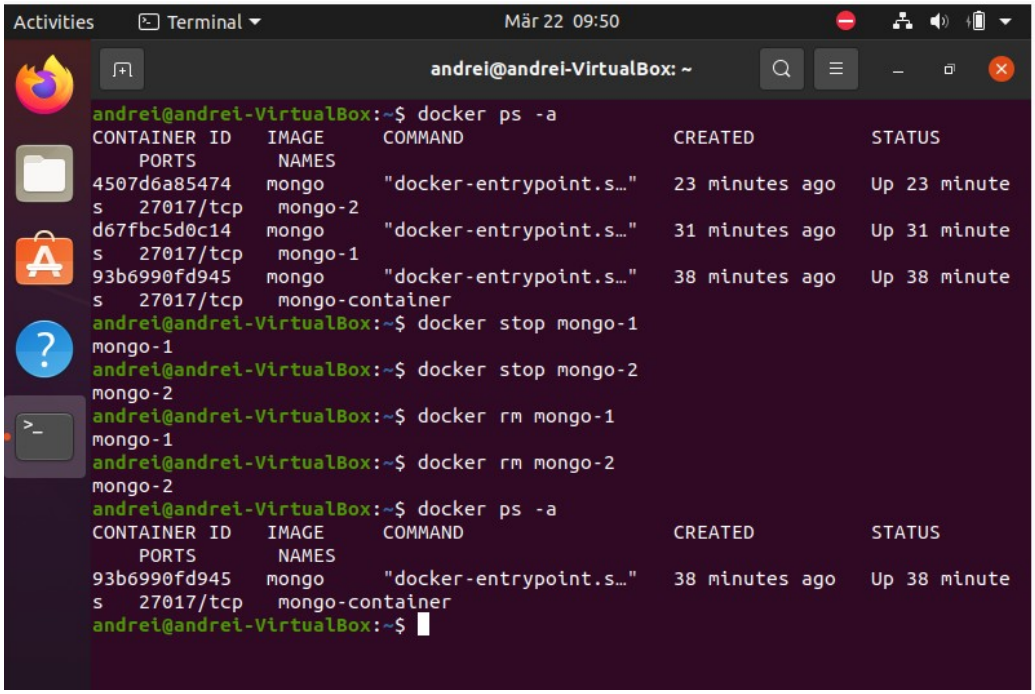
docker exec -it mongo_vol_container2 bash
cat db_data/file.txt



Останавливаем контейнеры и затем удаляем их

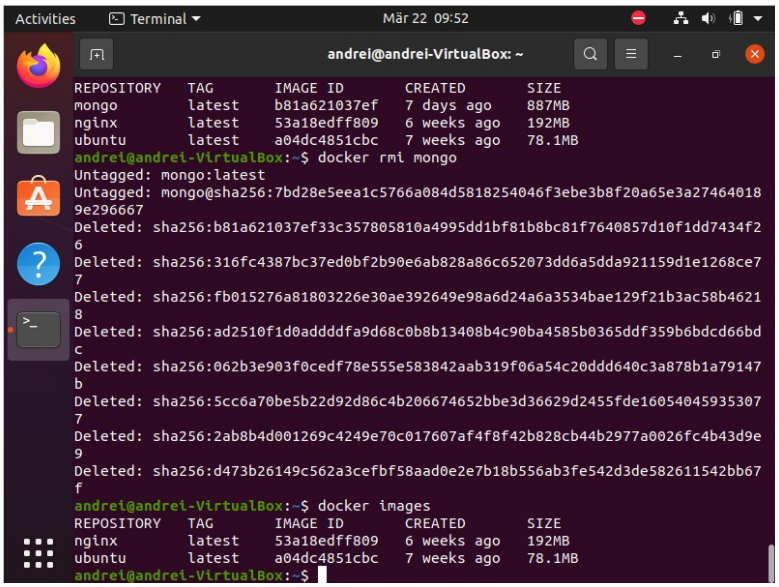
docker stop mongo-1
docker stop mongo-2

docker rm mongo-1
docker rm mongo-2



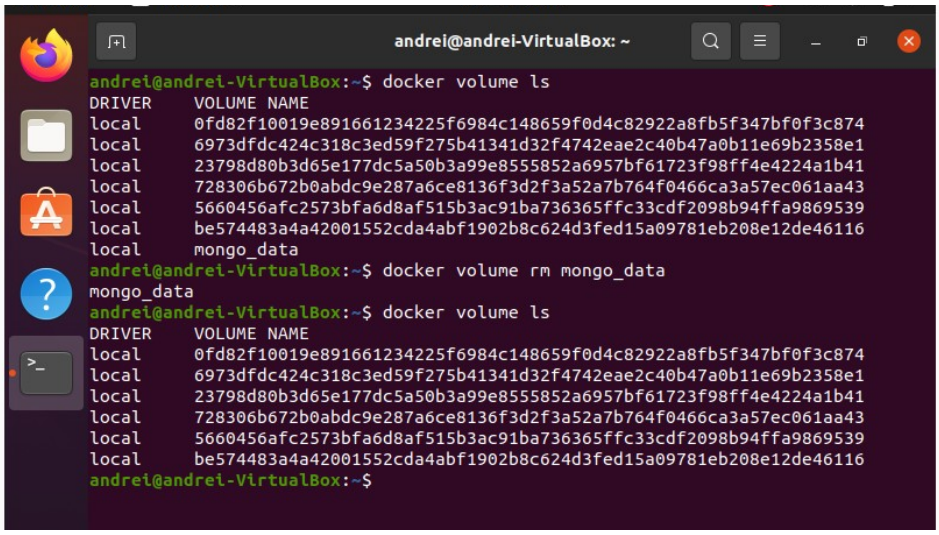
Удаляем образ mongo

docker rmi mongo



Удаляем том mongo_data

docker volume rm mongo_data



Удаляем network

docker network rm my_network