ECE-530 Cloud Computing

Homework #3: Linux Containers

Suyog Joshi – 101846426 – sjoshi@unm.edu

Spring2024

Abstract

Docker is a platform that simplifies the maintenance of highly customizable instances. It allows for quick setup and execution, often in just milliseconds, and can be used to create services accessible from anywhere in the world. For this homework assignment, we are required to create a Docker file capable of automatically building images. Additionally, we need to deploy a distributed database using Linux containers. This deployment must include at least two containers, each hosting a database instance. These instances must be interconnected, and each should store a portion of the database's data.

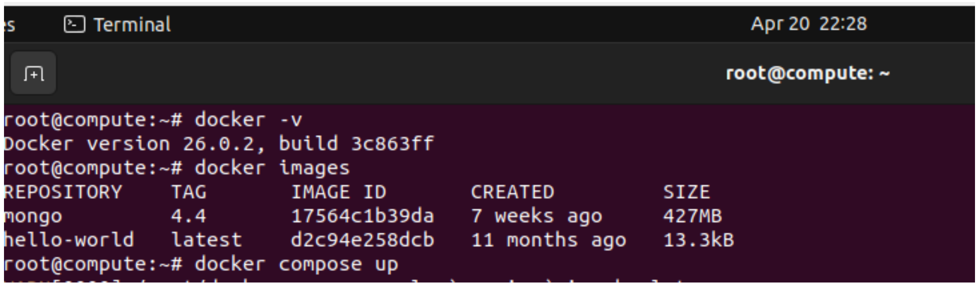
Introduction

For our deployment, we opted for Cassandra, a column-oriented NoSQL database. To construct our images, we utilized the Docker file presented in the Appendix. We soon discovered that Docker files have certain limitations in their build capabilities, particularly in terms of creating networks and generating multiple images simultaneously.

Deployment

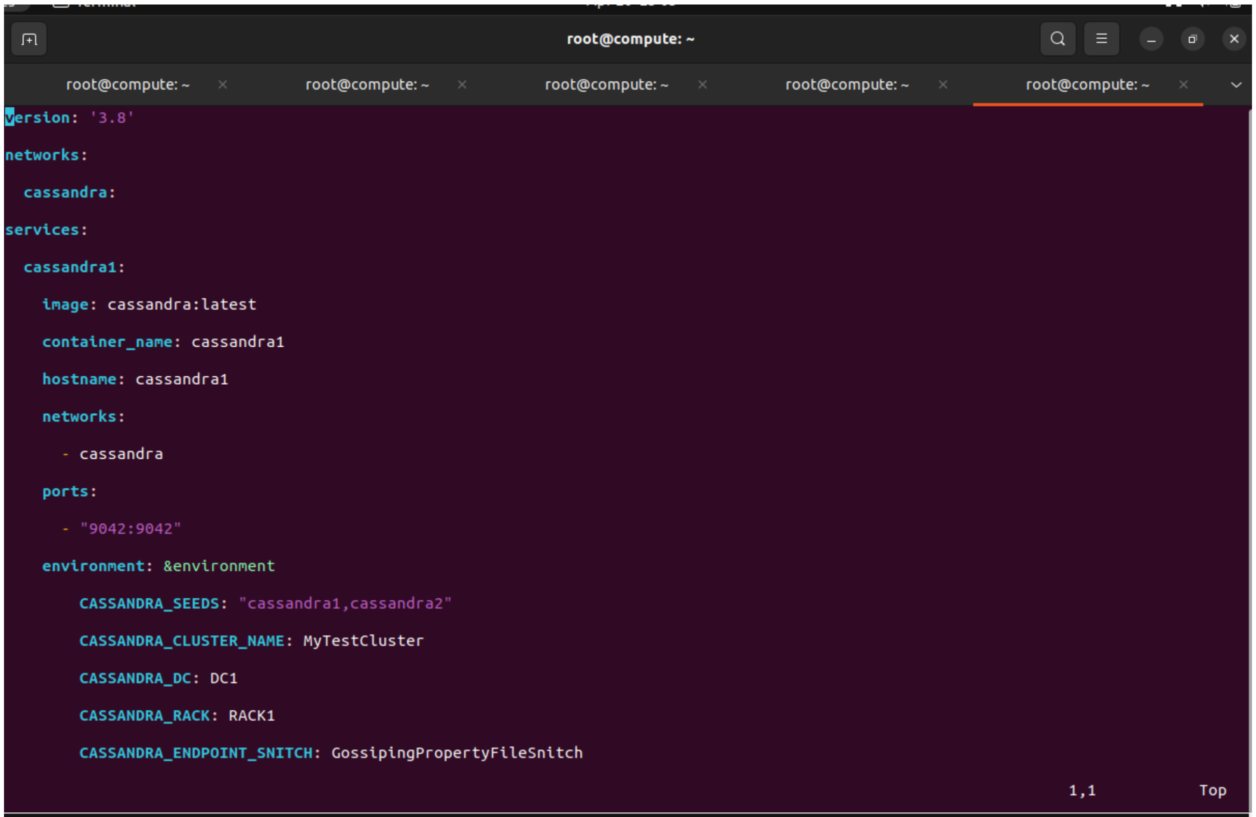
In order to use docker, we needed to install the necessary software.

We installed Docker and we can verify that from the figure below. Also, we listed the docker images before creating for this project.

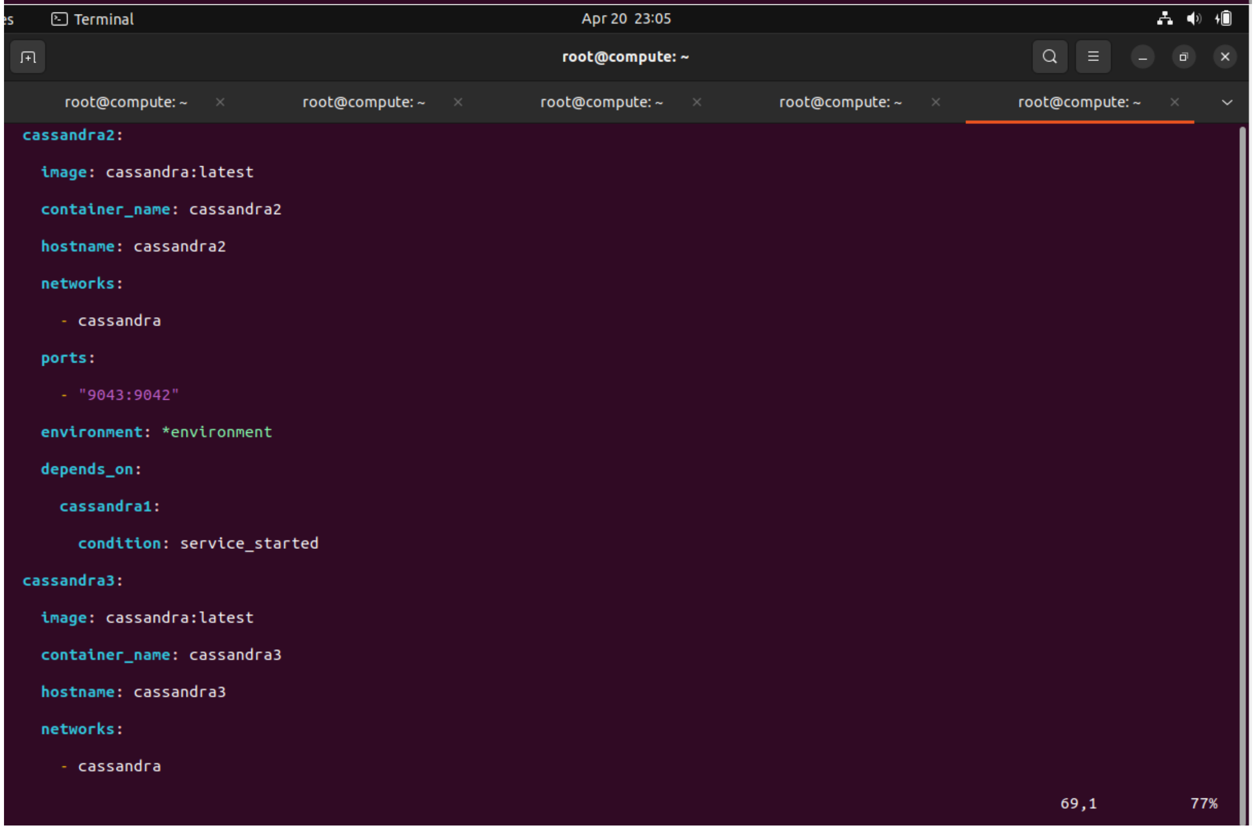


```
root@compute:~# docker -v
Docker version 26.0.2, build 3c863ff
root@compute:~# docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
mongo         4.4       17564c1b39da   7 weeks ago     427MB
hello-world   latest    d2c94e258dcb   11 months ago   13.3kB
root@compute:~# docker compose up
```

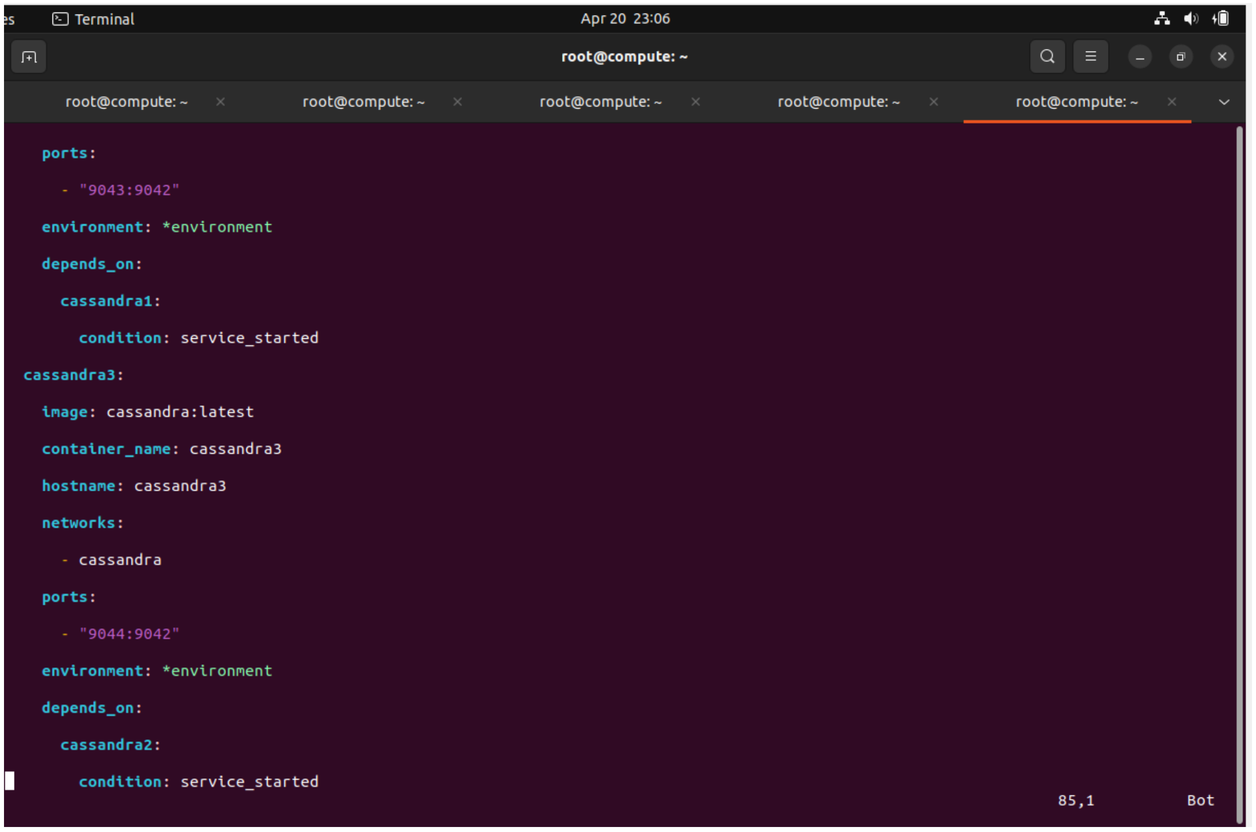
For this homework, we used 3 nodes running on a single machine. First, we created a docker-compose.yml file that describes our Cassandra cluster. The docker-compose.yml file is shown below:



```
version: '3.8'

networks:

  cassandra:

services:

  cassandra1:

    image: cassandra:latest

    container_name: cassandra1

    hostname: cassandra1

    networks:

      - cassandra

    ports:

      - "9042:9042"

    environment: &environment

        CASSANDRA_SEEDS: "cassandra1,cassandra2"

        CASSANDRA_CLUSTER_NAME: MyTestCluster

        CASSANDRA_DC: DC1

        CASSANDRA_RACK: RACK1

        CASSANDRA_ENDPOINT_SNITCH: GossipingPropertyFileSnitch

                                                                        1,1           Top
```



```
  cassandra2:

    image: cassandra:latest

    container_name: cassandra2

    hostname: cassandra2

    networks:

      - cassandra

    ports:

      - "9043:9042"

    environment: *environment

    depends_on:

      cassandra1:

        condition: service_started

  cassandra3:

    image: cassandra:latest

    container_name: cassandra3

    hostname: cassandra3

    networks:

      - cassandra

                                                                        69,1          77%
```



```
  ports:
    - "9043:9042"
  environment: *environment
  depends_on:
    cassandra1:
      condition: service_started
cassandra3:
  image: cassandra:latest
  container_name: cassandra3
  hostname: cassandra3
  networks:
    - cassandra
  ports:
    - "9044:9042"
  environment: *environment
  depends_on:
    cassandra2:
      condition: service_started
                                                                    85,1          Bot
```

First, we declared our docker compose version and created a network called cassandra to host our cluster.

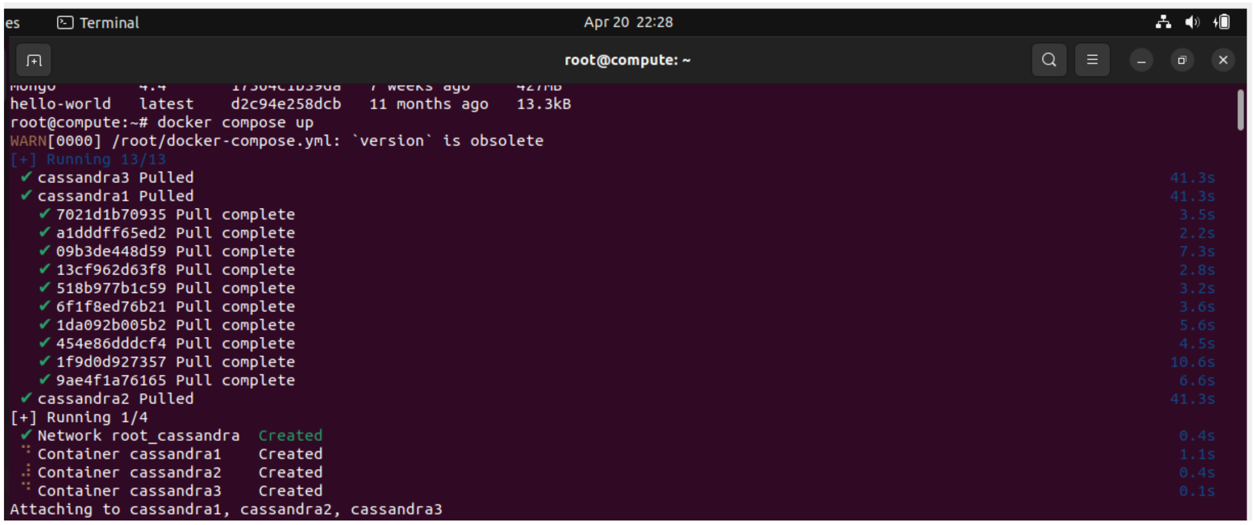
Under services, we initiate ‘cassandra1’. It's important to note that the ‘depends_on’ attributes in ‘cassandra2’ and ‘cassandra3’ prevent them from launching until the services on ‘cassandra1’ and ‘cassandra2’ have started, respectively. Additionally, we configure port forwarding so that port 9042 on our local machine is mapped to port 9042 on the container. We also connect it to the previously established Cassandra network.
We then set some environment variables needed for startup, such as declaring CASSANDRA_SEEDS to be cassandra1 and cassandra2.

The settings for the containers 'cassandra2' and 'cassandra3' are quite alike, with the primary distinction being their names.

Both containers utilize the same ‘cassandra:latest` image, assign their container names, join the same Cassandra network, and open port 9042. They also reference the same set of environment variables as 'cassandra1', using the *environment syntax. The sole variation lies in their dependencies: 'cassandra2' starts only after 'cassandra1' is up, while 'cassandra3' starts following 'cassandra2'.

To deploy the Cassandra cluster and running commands

To deploy the Cassandra cluster, use the Docker CLI in the same folder as the docker-compose.yml to run the following command:

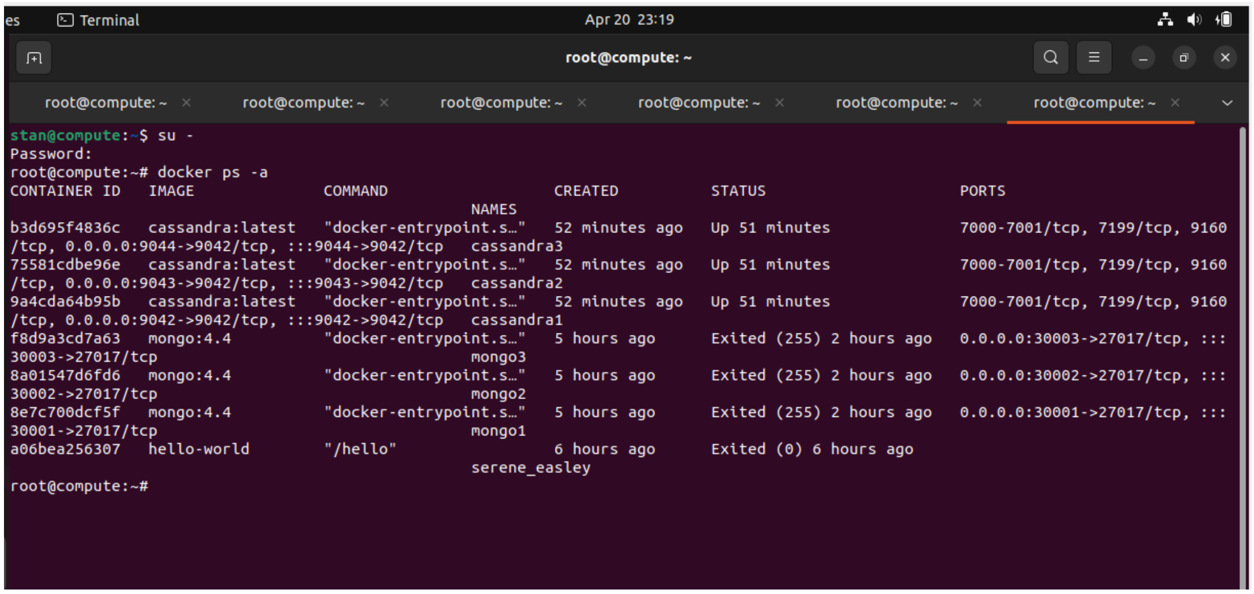


```
hello-world   latest    d2c94e258dcb   11 months ago   13.3kB
root@compute:~# docker compose up
WARN[0000] /root/docker-compose.yml: `version` is obsolete
[+] Running 13/13
 ✓ cassandra3 Pulled                                                    41.3s
 ✓ cassandra1 Pulled                                                    41.3s
   ✓ 7021d1b70935 Pull complete                                          3.5s
   ✓ a1dddff65ed2 Pull complete                                          2.2s
   ✓ 09b3de448d59 Pull complete                                          7.3s
   ✓ 13cf962d63f8 Pull complete                                          2.8s
   ✓ 518b977b1c59 Pull complete                                          3.2s
   ✓ 6f1f8ed76b21 Pull complete                                          3.6s
   ✓ 1da092b005b2 Pull complete                                          5.6s
   ✓ 454e86dddcf4 Pull complete                                          4.5s
   ✓ 1f9d0d927357 Pull complete                                         10.6s
   ✓ 9ae4f1a76165 Pull complete                                          6.6s
 ✓ cassandra2 Pulled                                                    41.3s
[+] Running 1/4
 ✓ Network root_cassandra  Created                                       0.4s
 ⠋ Container cassandra1    Created                                       1.1s
 ⠋ Container cassandra2    Created                                       0.4s
 ⠋ Container cassandra3    Created                                       0.1s
Attaching to cassandra1, cassandra2, cassandra3
```

The command to see the 3 running containers.

```
docker ps -a
```

We can see cassandra1, cassandra2 and cassandra3 that we created.

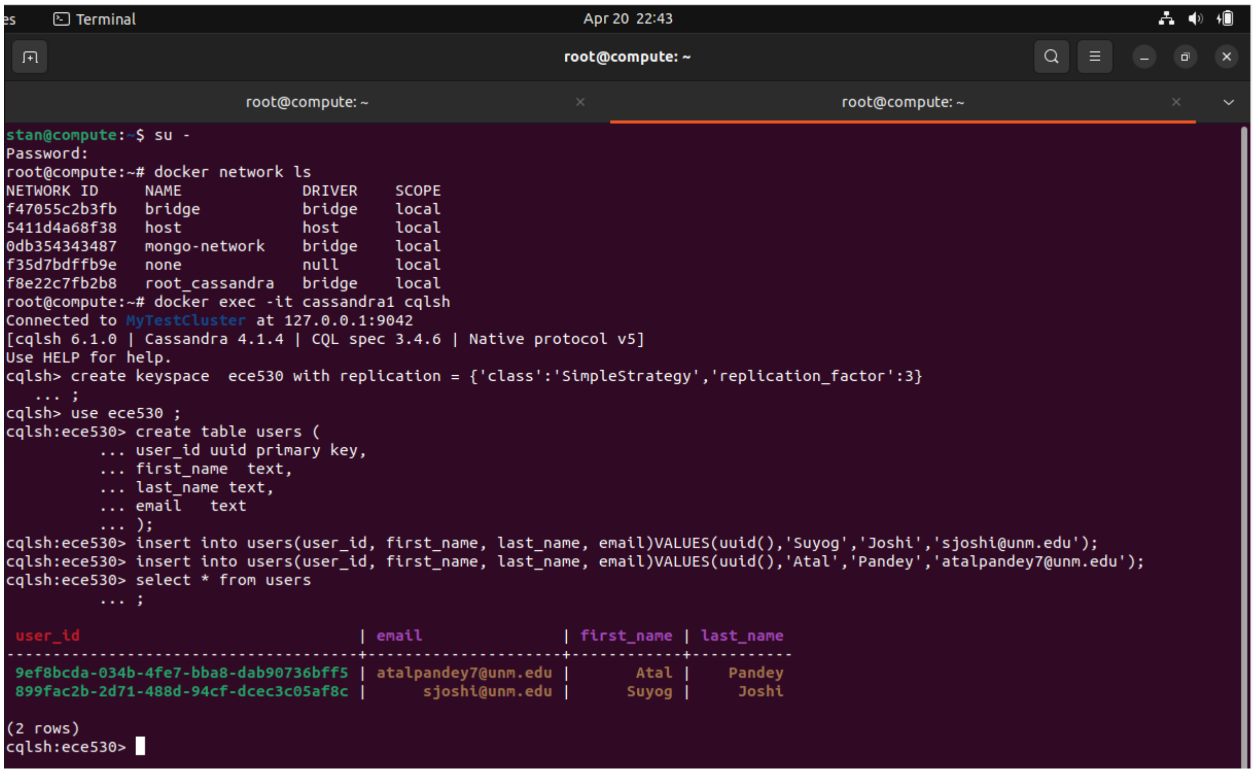


```
stan@compute:~$ su -
Password:
root@compute:~# docker ps -a
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS                     PORTS
                                                NAMES
b3d695f4836c   cassandra:latest   "docker-entrypoint.s…"   52 minutes ago   Up 51 minutes              7000-7001/tcp, 7199/tcp, 9160
/tcp, 0.0.0.0:9044->9042/tcp, :::9044->9042/tcp   cassandra3
75581cdbe96e   cassandra:latest   "docker-entrypoint.s…"   52 minutes ago   Up 51 minutes              7000-7001/tcp, 7199/tcp, 9160
/tcp, 0.0.0.0:9043->9042/tcp, :::9043->9042/tcp   cassandra2
9a4cda64b95b   cassandra:latest   "docker-entrypoint.s…"   52 minutes ago   Up 51 minutes              7000-7001/tcp, 7199/tcp, 9160
/tcp, 0.0.0.0:9042->9042/tcp, :::9042->9042/tcp   cassandra1
f8d9a3cd7a63   mongo:4.4          "docker-entrypoint.s…"   5 hours ago      Exited (255) 2 hours ago   0.0.0.0:30003->27017/tcp, :::
30003->27017/tcp                                mongo3
8a01547d6fd6   mongo:4.4          "docker-entrypoint.s…"   5 hours ago      Exited (255) 2 hours ago   0.0.0.0:30002->27017/tcp, :::
30002->27017/tcp                                mongo2
8e7c700dcf5f   mongo:4.4          "docker-entrypoint.s…"   5 hours ago      Exited (255) 2 hours ago   0.0.0.0:30001->27017/tcp, :::
30001->27017/tcp                                mongo1
a06bea256307   hello-world        "/hello"                 6 hours ago      Exited (0) 6 hours ago
                                                serene_easley
root@compute:~#
```

The data container has been created, we connected to it using the following command

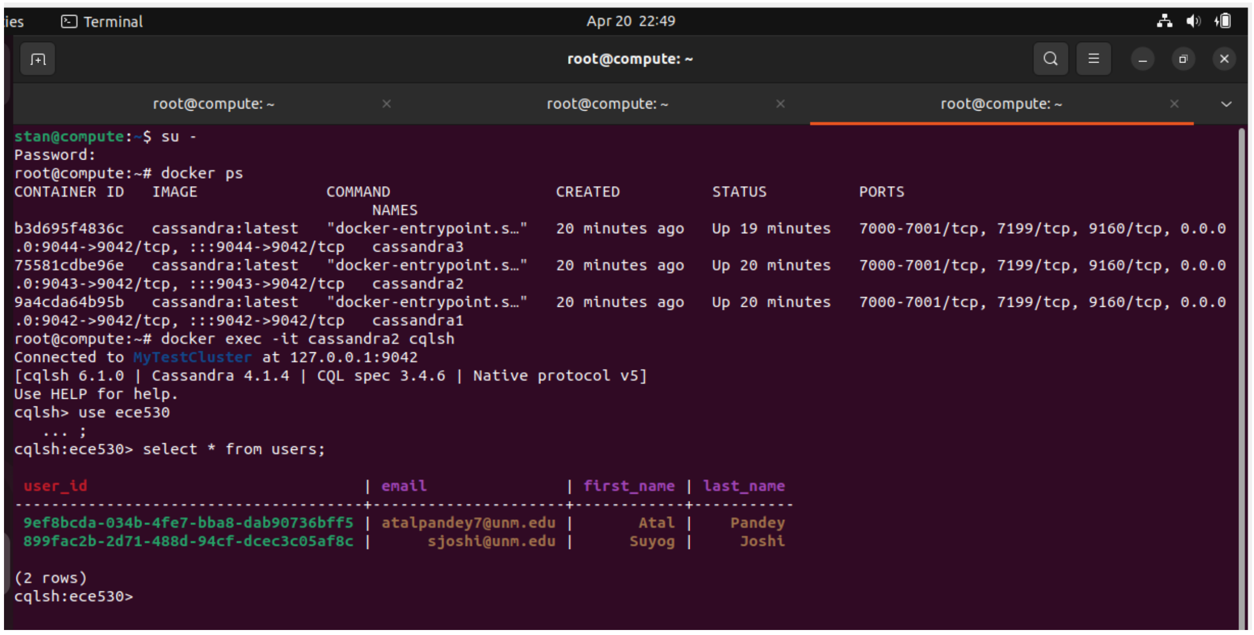
```
Docker exec -it cassandra1 cqlsh
```

This will run cqlsh, or CQL Shell, inside the container allowing us to make queries to the new Cassandra database. After that we created the a database in cassandra1 using keyspace ece530.

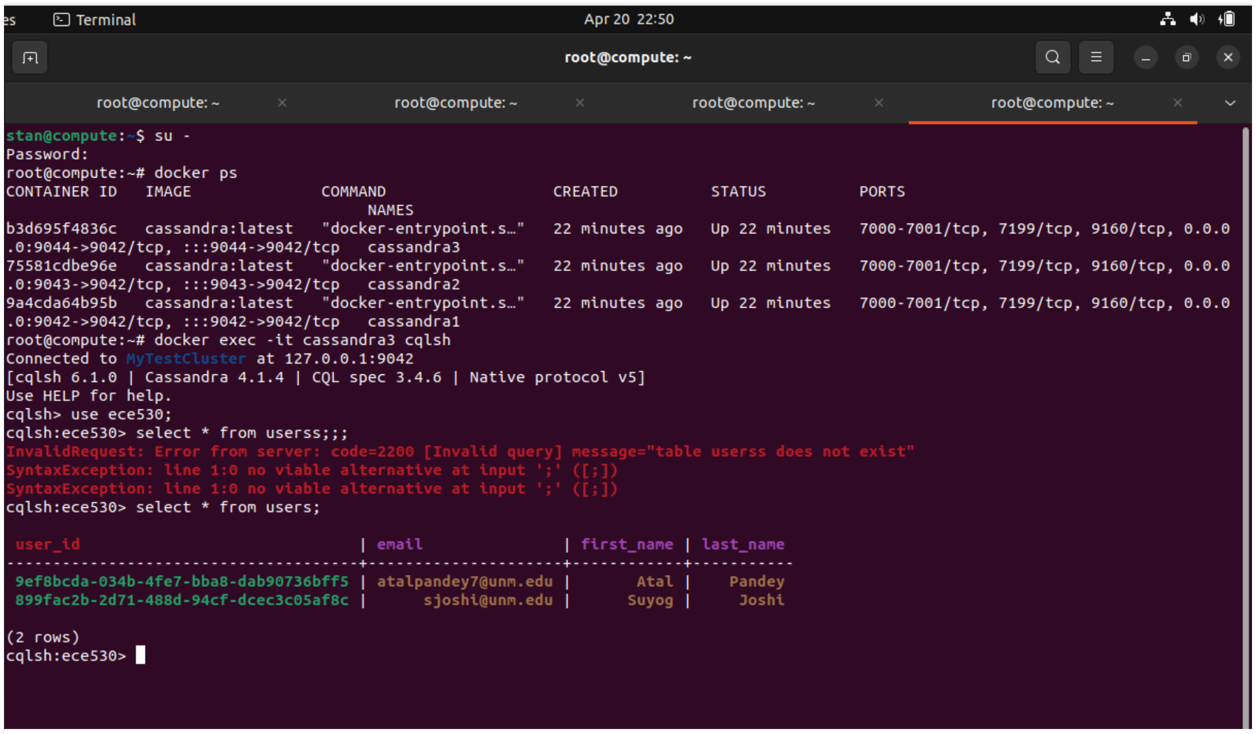


```
stan@compute:~$ su -
Password:
root@compute:~# docker network ls
NETWORK ID     NAME             DRIVER    SCOPE
f47055c2b3fb   bridge           bridge    local
5411d4a68f38   host             host      local
0db354343487   mongo-network    bridge    local
f35d7bdffb9e   none             null      local
f8e22c7fb2b8   root_cassandra   bridge    local
root@compute:~# docker exec -it cassandra1 cqlsh
Connected to MyTestCluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> create keyspace  ece530 with replication = {'class':'SimpleStrategy','replication_factor':3}
   ... ;
cqlsh> use ece530 ;
cqlsh:ece530> create table users (
          ... user_id uuid primary key,
          ... first_name  text,
          ... last_name text,
          ... email   text
          ... );
cqlsh:ece530> insert into users(user_id, first_name, last_name, email)VALUES(uuid(),'Suyog','Joshi','sjoshi@unm.edu');
cqlsh:ece530> insert into users(user_id, first_name, last_name, email)VALUES(uuid(),'Atal','Pandey','atalpandey7@unm.edu');
cqlsh:ece530> select * from users
          ... ;

 user_id                              | email               | first_name | last_name
--------------------------------------+---------------------+------------+-----------
 9ef8bcda-034b-4fe7-bba8-dab90736bff5 | atalpandey7@unm.edu |       Atal |    Pandey
 899fac2b-2d71-488d-94cf-dcec3c05af8c |      sjoshi@unm.edu |      Suyog |     Joshi

(2 rows)
cqlsh:ece530> 
```

Now we accessed this same database on from the other 2 nodes i.e. cassandra2 and cassandra3.



```
stan@compute:~$ su -
Password:
root@compute:~# docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED          STATUS          PORTS
                                     NAMES
b3d695f4836c   cassandra:latest   "docker-entrypoint.s…"   20 minutes ago   Up 19 minutes   7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0
.0:9044->9042/tcp, :::9044->9042/tcp   cassandra3
75581cdbe96e   cassandra:latest   "docker-entrypoint.s…"   20 minutes ago   Up 20 minutes   7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0
.0:9043->9042/tcp, :::9043->9042/tcp   cassandra2
9a4cda64b95b   cassandra:latest   "docker-entrypoint.s…"   20 minutes ago   Up 20 minutes   7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0
.0:9042->9042/tcp, :::9042->9042/tcp   cassandra1
root@compute:~# docker exec -it cassandra2 cqlsh
Connected to MyTestCluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> use ece530
   ... ;
cqlsh:ece530> select * from users;

 user_id                              | email                | first_name | last_name
--------------------------------------+----------------------+------------+-----------
 9ef8bcda-034b-4fe7-bba8-dab90736bff5 | atalpandey7@unm.edu |       Atal |    Pandey
 899fac2b-2d71-488d-94cf-dcec3c05af8c |      sjoshi@unm.edu |      Suyog |     Joshi

(2 rows)
cqlsh:ece530>
```

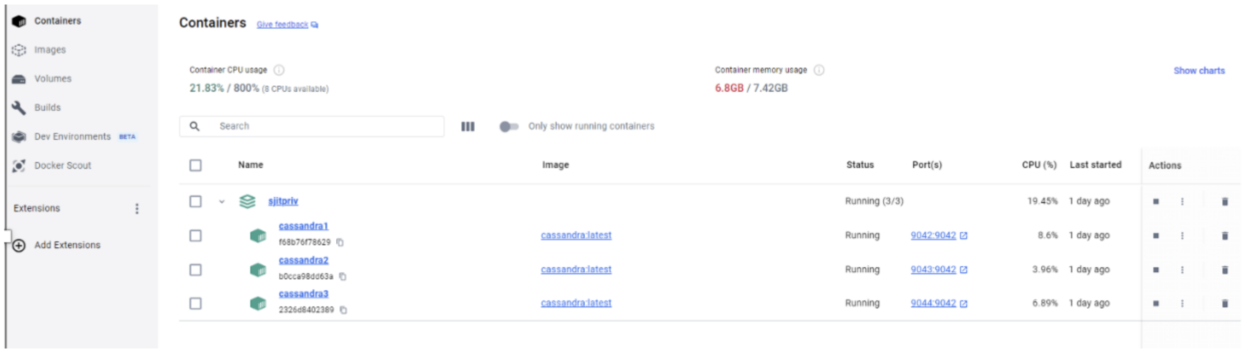


```
stan@compute:~$ su -
Password:
root@compute:~# docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED          STATUS          PORTS
                                     NAMES
b3d695f4836c   cassandra:latest   "docker-entrypoint.s…"   22 minutes ago   Up 22 minutes   7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0
.0:9044->9042/tcp, :::9044->9042/tcp   cassandra3
75581cdbe96e   cassandra:latest   "docker-entrypoint.s…"   22 minutes ago   Up 22 minutes   7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0
.0:9043->9042/tcp, :::9043->9042/tcp   cassandra2
9a4cda64b95b   cassandra:latest   "docker-entrypoint.s…"   22 minutes ago   Up 22 minutes   7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0
.0:9042->9042/tcp, :::9042->9042/tcp   cassandra1
root@compute:~# docker exec -it cassandra3 cqlsh
Connected to MyTestCluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> use ece530;
cqlsh:ece530> select * from userss;;;
InvalidRequest: Error from server: code=2200 [Invalid query] message="table userss does not exist"
SyntaxException: line 1:0 no viable alternative at input ';' ([;])
SyntaxException: line 1:0 no viable alternative at input ';' ([;])
cqlsh:ece530> select * from users;

 user_id                              | email                | first_name | last_name
--------------------------------------+----------------------+------------+-----------
 9ef8bcda-034b-4fe7-bba8-dab90736bff5 | atalpandey7@unm.edu |       Atal |    Pandey
 899fac2b-2d71-488d-94cf-dcec3c05af8c |      sjoshi@unm.edu |      Suyog |     Joshi

(2 rows)
cqlsh:ece530>
```

This satisfies our requirement for this project.

Extra Credit:

Docker Desktop is a graphical user interface (GUI) that complements Docker's command line interface (CLI), offering a more intuitive way to manage Docker containers and images. This GUI is particularly valuable for users who prefer visual aids, as it provides clear visual feedback on the status and relationships of containers, making it easier to interpret than command line output. It's also more user-friendly for those less accustomed to CLI operations, simplifying complex tasks such as network and volume management, as well as log reviews. Furthermore, Docker Desktop allows for the comprehensive management of multiple Docker resources at once, which can significantly enhance productivity and ease of use. This blend of visual simplicity and operational depth makes GUIs an essential tool for efficient Docker management.



Containers

Container CPU usage
21.83% / 800% (8 CPUs available)

Container memory usage
6.8GB / 7.42GB

Name	Image	Status	Port(s)	CPU (%)	Last started
sjitpriv		Running (3/3)		19.45%	1 day ago
cassandra1 f68b76f78629	cassandra:latest	Running	9042:9042	8.6%	1 day ago
cassandra2 b0cca98dd63a	cassandra:latest	Running	9043:9042	3.96%	1 day ago
cassandra3 2326d8402389	cassandra:latest	Running	9044:9042	6.89%	1 day ago