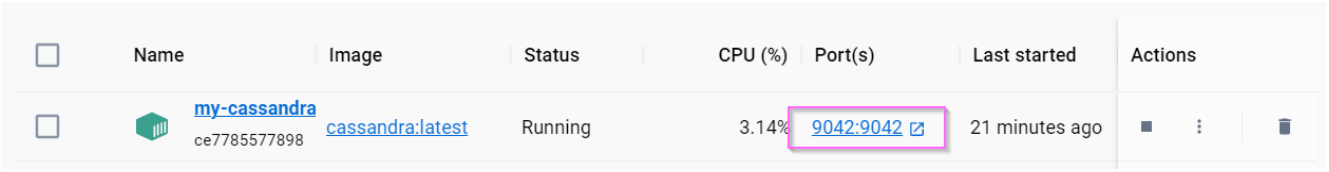
Cassandra Local Installation

Overview

1. Install Docker Desktop (https://www.docker.com/pricing/)
2. Create a folder that you will run your test script out of (e.g. c:\dev\cassandra-test)
3. Follow the quickstart guide to install the cassandra container: https://cassandra.apache.org/_/quickstart.html
    1. Note that in the instructions above, it may not perform the correct port mapping. Use this to start the container with the necessary open ports:

```
docker run --name my-cassandra -p 9042:9042 -d cassandra:latest
```

4. Once it's running, you should see the following active port mapping in docker desktop:

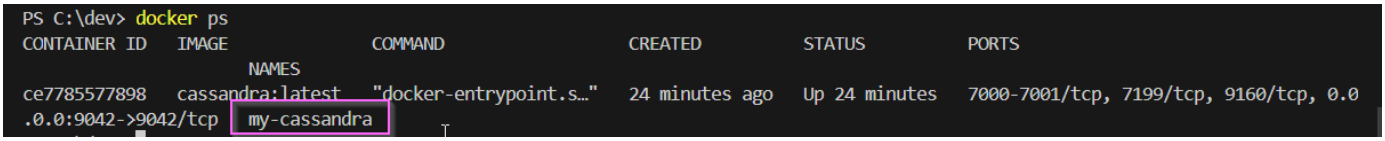


5. Install the cassandra driver for python using pip. Make sure you run this in your c:\dev\cassandra-test folder.

```
pip install cassandra-driver
```

6. Open the csl command line:

```
docker ps (this gives you the container name)
docker exec -it my-cassandra cqlsh
```



7) Enter the following in the cql command line, and hit return until all commands are processed.

```
CREATE KEYSPACE IF NOT EXISTS store WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' :
'1' };

-- Create a table
CREATE TABLE IF NOT EXISTS store.shopping_cart (
userid text PRIMARY KEY,
item_count int,
last_update_timestamp timestamp
);

-- Insert some data
INSERT INTO store.shopping_cart
(userid, item_count, last_update_timestamp)
VALUES ('9876', 2, toTimeStamp(now()));
INSERT INTO store.shopping_cart
(userid, item_count, last_update_timestamp)
VALUES ('1234', 5, toTimeStamp(now()));
```

8. Run "Select * from store.shopping_cart" in the cql command line. Ensure that the data you inserted is returned.

9. Create the following app.py script in your c:\dev\cassandra-test folder

```
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider

def main():
    # Configure the connection:
    cluster = Cluster(
        ['127.0.0.1'],  # Localhost (change if your IP is different)
        port=9042  # Default Cassandra port
    )

    # Connect to the cluster:
    try:
        session = cluster.connect()
        print("Connected to Cassandra cluster.")

        rows = session.execute("SELECT * FROM store.shopping_cart;")
        for row in rows:
            print(f'id: {row.userid}, item count: {row.item_count}')

    except Exception as e:
        print("Failed to connect to Cassandra cluster:", e)
    finally:
        cluster.shutdown()

if __name__ == "__main__":
    main()
```

Validate that the script returns a the values from the table you created.

Notes

- This example assumes that you're using the default configuration with no authentication. If your Cassandra setup requires authentication, you'll need to use the `PlainTextAuthProvider` for username and password, as hinted in the import statement.
- If you're running Cassandra with a different configuration (e.g., a different port or requiring authentication), you'll need to adjust the connection settings accordingly.
- Ensure that your Docker container with Cassandra is running when you execute this script.

Running the script should result in a message indicating that you've connected to the Cassandra cluster. If there are any errors, they will likely relate to the Cassandra cluster not being accessible under the specified configuration, so double-check your Docker and Cassandra settings.