Xavier Goovaerts
Jeremy Moore
COSC 480 Cloud Computing
Fall 2019

DRONE PROJECT - PART VI

REPOSITORY: https://github.com/xfgoovaerts/DroneProjectPartOne

WHAT WE GOT TO WORK:

- Created delete.py script which will delete the most recent flight after receiving user confirmation.
- Modified delete.py such that providing a flight id as an argument will delete that specific flight (after receiving user confirmation).
- Created list.py script to list the last 10 flights sorted by time.
- Modified fly.py such that, if the last argument is an 'r', the flight id will not be reset and the previous flight will resume.
- Added the two most recent flights to the top of the leaderboard in the Leaderboard.php file.
- We already had DSE reading the Cassandra node IP addresses from a text file.

WHAT WE TRIED HARD BUT DIDN'T GET TO WORK:

- All assigned tasks were completed.

Xavier Goovaerts
Jeremy Moore
COSC 480 Cloud Computing
Fall 2019

DRONE PROJECT - PART VI

Xavier Goovaerts
Jeremy Moore
COSC 480 Cloud Computing
Fall 2019

DRONE PROJECT - PART VI

2. CONNECT TO AWS CLUSTER FROM PYTHON SCRIPT

At the moment, we are using the native Cassandra python drivers to connect to the DSE instance by importing the following classes:

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

These can easily be converted to the DSE python driver using the instructions found here: https://docs.datastax.com/en/developer/python-dse-driver/1.1/getting_started/

Because the tutorial we followed included setting credentials on the cluster, the connect command must include an **auth_provider**. Example code is shown below:

```
def connect_to_db():
    global session

    auth_provider=PlainTextAuthProvider(
        username='cassandra',
        password='eagles29')

    cluster = Cluster(
        auth_provider=auth_provider,
        contact_points=['3.230.244.15', '3.228.63.63',
        '3.231.140.68'])

    try:
        session = cluster.connect('competition')
        print('Connected to Cassandra cluster.')

    except:
        print('Cannot connect to database. Exiting ...')
        exit(1)
```

The contact points are the public IP addresses of the AWS EC2 instances running each node of the cluster.

The try/except isn’t strictly necessary, but without it the flight would continue without being recorded in the database and the console output would include a series of errors as queries execution is executed.

We thought it would be better if the program quit, so flights are not inadvertently flown without being recorded.

Xavier Goovaerts
Jeremy Moore
COSC 480 Cloud Computing
Fall 2019

DRONE PROJECT - PART VI