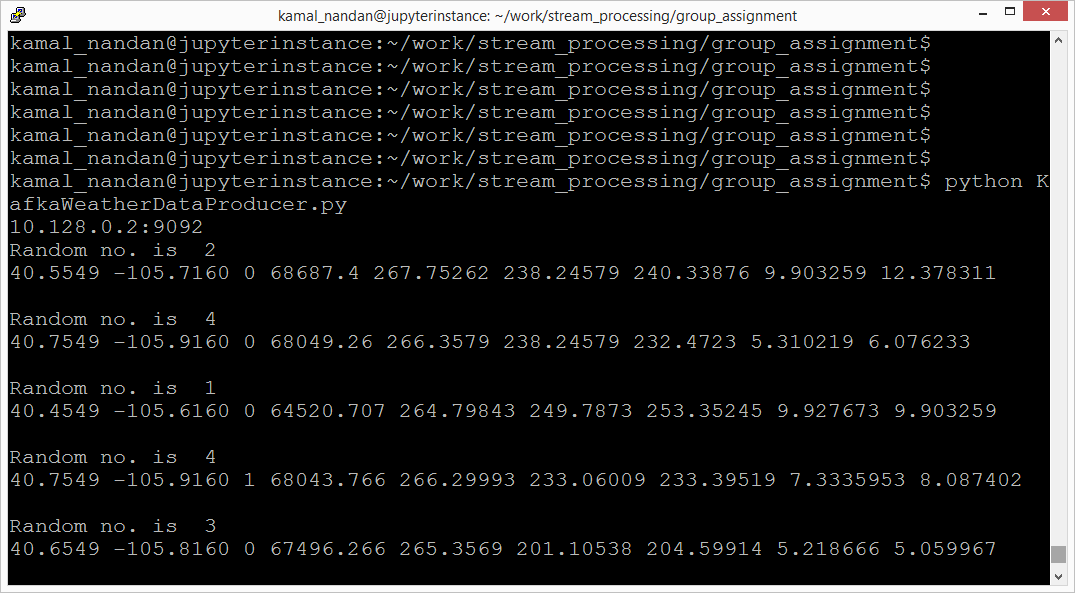
**Screenshots and steps to run the program:**

1. Open “groupe.conf” configuration file and set the required parameters such as kafka broker host, port and topics etc.

If you want email notification for simulation of notification purposes, provide SMTP details too.



1. Launch the script “KafkaWeatherDataProducer.py” by typing the following command and pressing enter:
   * python KafkaWeatherDataProducer.py



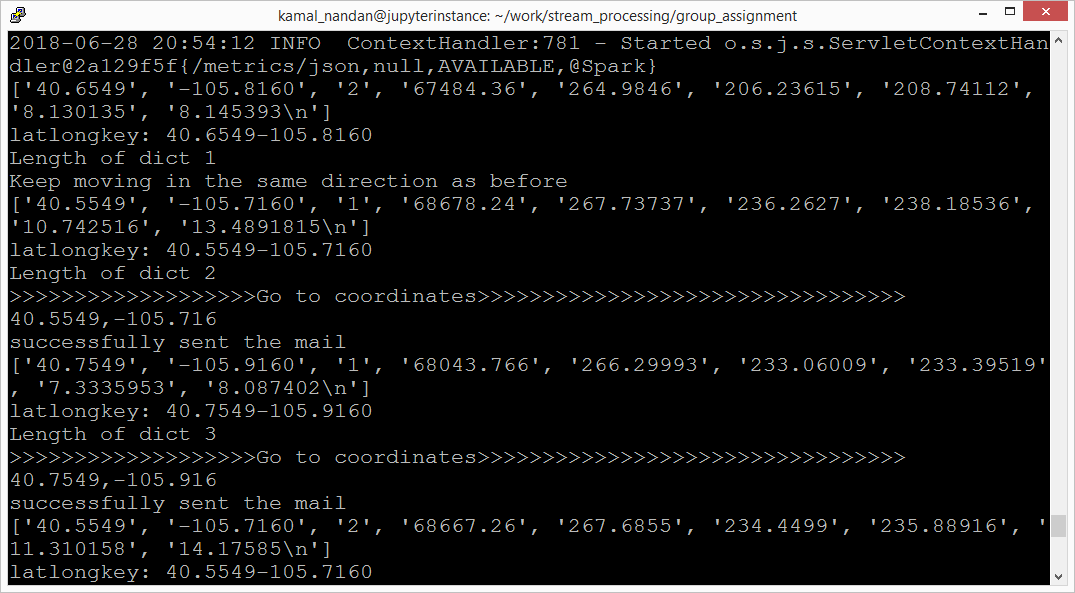
This script is a simulator for IoT devices that will be sending the weather data to the Kafka broker. Once this script starts, it starts reading the weather data records randomly from different files and starts sending the data into the Kafta topic.

1. Open another terminal, browse into the same folder as before and run the shell script “./run\_kafka\_consumer.sh” by typing the following command:
   * **./run\_kafka\_consumer.sh**

Doing the above will launch the python script called **“ConsumeWeatherDataFromKafka.py”** which will start pulling the weather data from kafka topic in a streaming fashion and run the required algorithm, determine the most favorable direction in which the boat should move.

**Important note:** In real life, the stream-processor would be sending the data to a database from where it would be picked up by the REST engine and published on to the on-board devices, on the boat, that would have subscribed to the notifications. For now we are simulating the notifications by sending emails.

In the following screenshot, we see the “KafkaCosnumer” in action:



We see in the above screenshots that the Consumer script would run its algorithm and determine the best co-ordinate to which the boat must move to and send notification to the subscriber.

To simulate the sending of notification, as a make-shift arrangement, we are sending notifications over email.

1. To stop the above scripts, press control-c at both the terminals

**Future scope for improvement:**

1. REST interface to which the on-board devices would have subscribed.
2. More granularity in notification