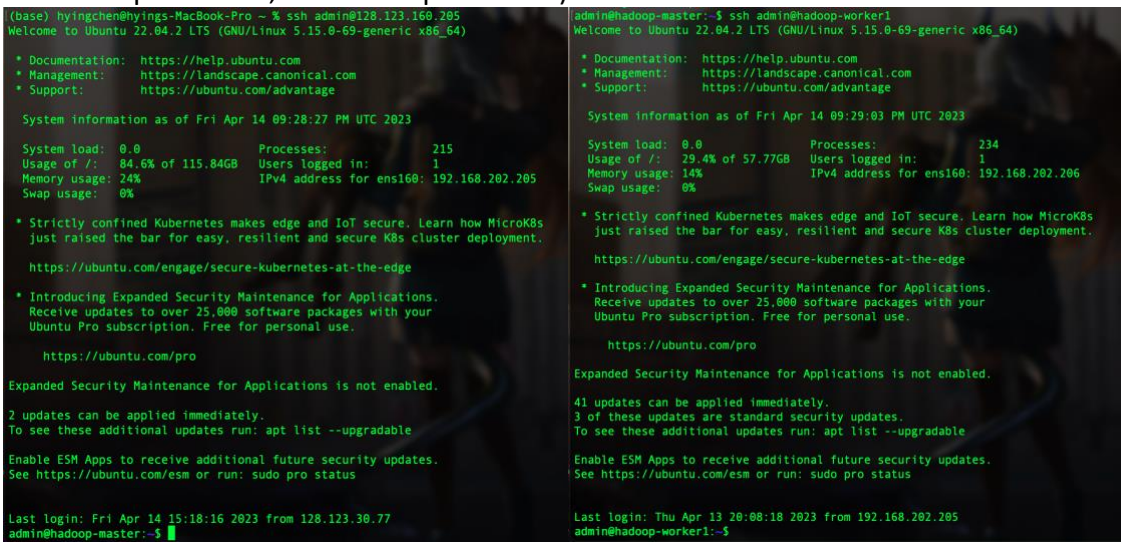
CS 496/522 Cloud Computing
Programming Assignment 1

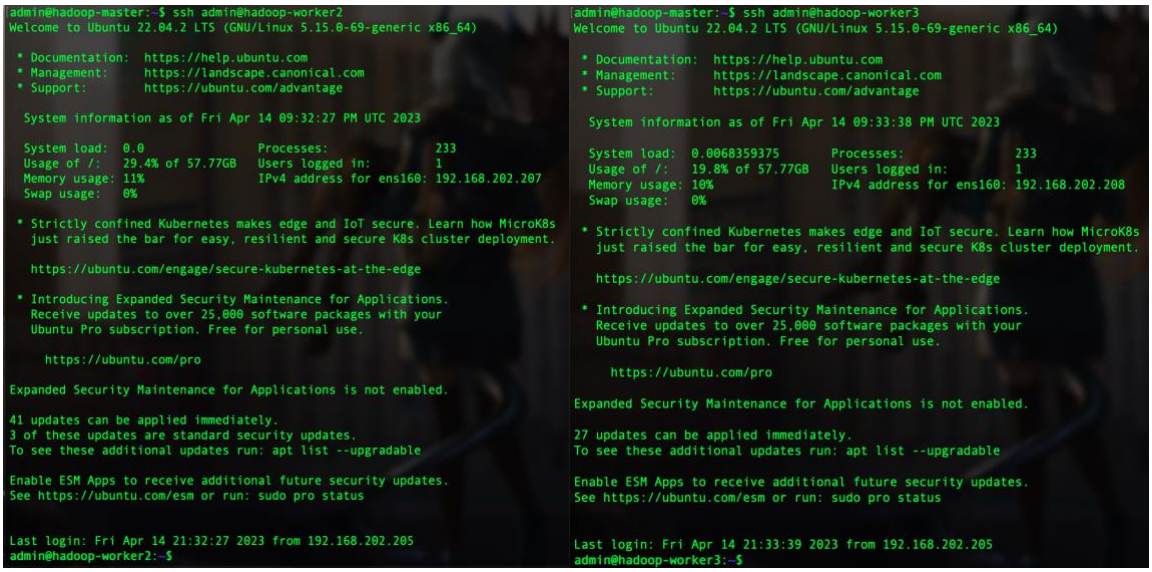
Huiying Chen
800722249

Note: Detailed walk through written in Prog1_readme.md. This document is a supplement for task tracking, results showcase, and for easy grading.

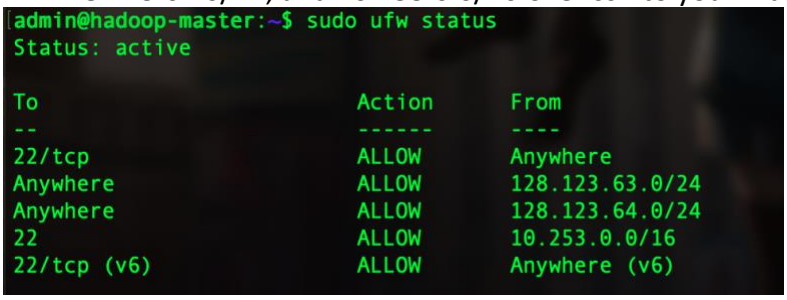
1. Setting up VMs – 20 pts

- Deploy 4 nodes (1 light and 3 heavy) with specs; (done)
- Deploy Ubuntu-server (v.22) on all four nodes (done)
- Do the OS and system setup
    - Setup the IP address configuration; create and establish connectivity between the nodes (I have renamed my cluster nodes to be hadoop-master, hadoop-worker1, hadoop-worker2, and hadoop-worker3)

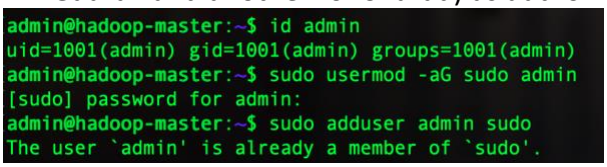


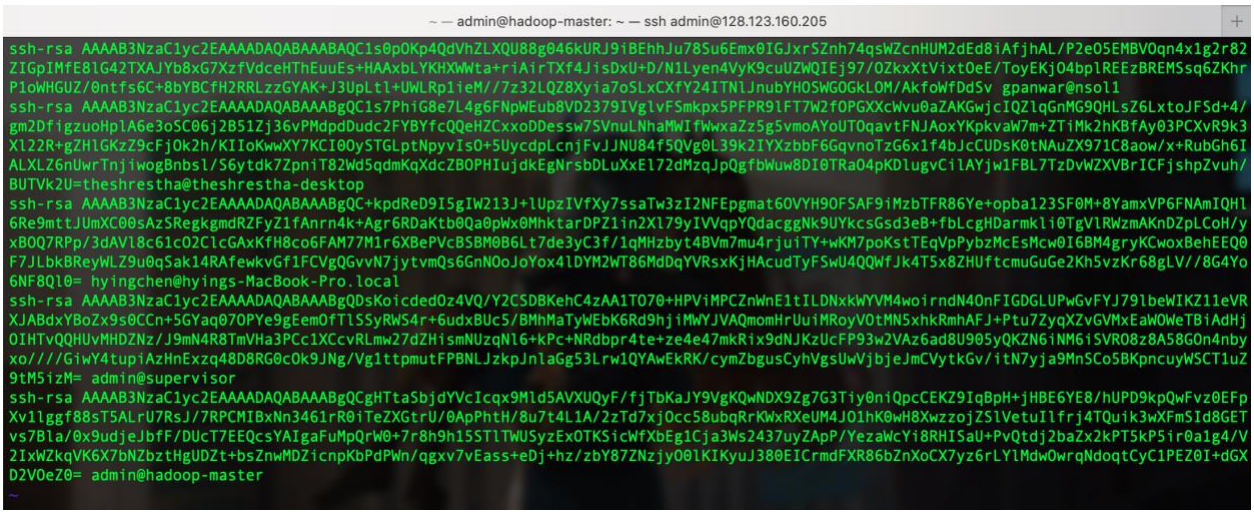


- Setup firewall protection; Open Firewall access from 128.123.63.0/24, 128.123.64.0/24, and 10.253.0.0/16 over ssh to your master node.

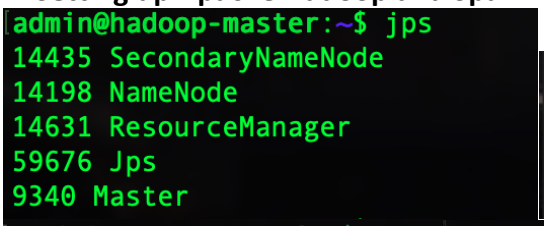


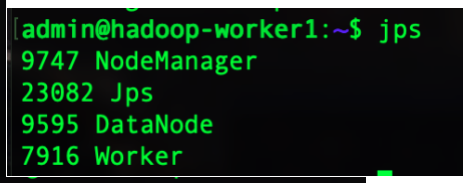
- Create user named admin and give them sudo access. Add two ssh keys (one for Gaurav and another for Sharad) as authorized keys for the admin user:

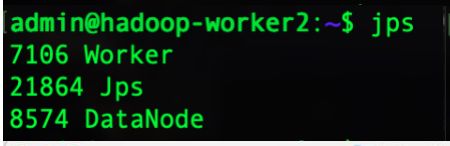


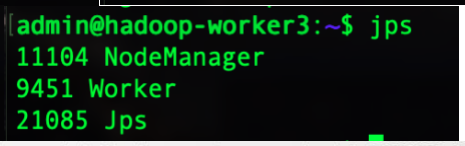


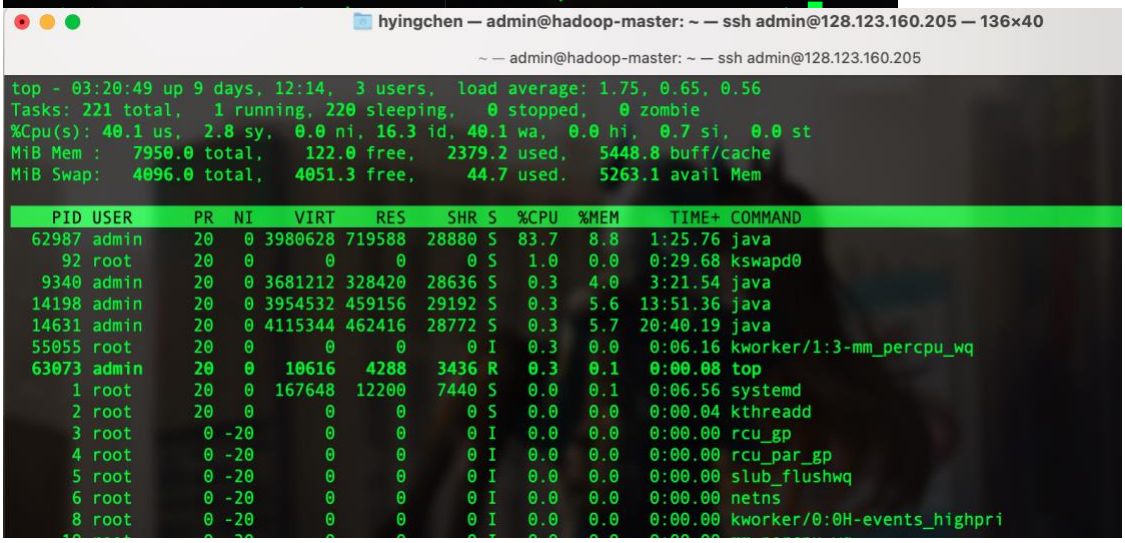
2. Setting up Apache Hadoop and Spark frameworks – 20 pts





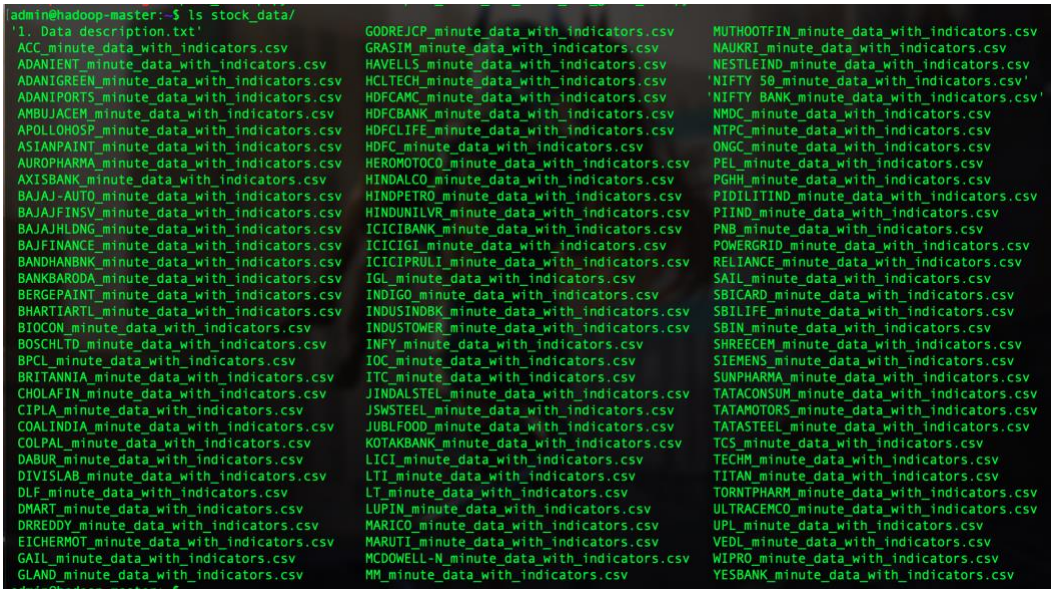






3. Performing the map-reduce and spark tasks – 30 pts; Clear and well-presented output – 10 pts

- Retrieve data available using scp



- **Find the maximum number of transactions in a day for all companies in a user supplied given time window.**

>> Logic:

1) Read arguments of start date and end date
2) Construct dataframe structure
3) Read csv files as spark dataframes for all companies then combine them into a single one
4) Select necessary columns and stock transaction data between start date and end date
5) **Sum 'volume' as 'daily_trade_count', groupby 'date'** (extract date info from timestamp as date)
6) Find the **max daily_trade_count value** and output

>> Code developed in spark_stock_max_trans_all_given_win.py

Running Sample:

```spark-submit spark_stock_max_trans_all_given_win.py -start <start_date> -end <end_date>```

>> spark-submit spark_stock_max_trans_all_given_win.py -start 01/01/2016 -end 12/31/2016 >spark_output/output_stock_max_trans_all_given_win.log

>> check full log file at attached output files

- **Maximum stock deviation is defined as (highest price for a day – lowest price for a day)/lowest price for a day. Find the stock that had the highest maximum stock deviation in a day among all stocks and what was the corresponding value for a given time window.**

>> Logic:

1) Read arguments of start date and end date
2) Construct dataframe structure
```

3) Read csv files as spark dataframes, extracting company_name from file_name, then assign the value as 'company' into the corresponding dataframe
4) Combine all dataframes into a single one
5) Select necessary columns and stock transaction data between start date and end date
6) Calculate **'high'-'low' value as 'deviation', groupby 'company', 'date'** (extract date info from timestamp as date)
7) **Find the max 'deviation' in a day as 'max_deviation'**
8) Find the **max 'max_deviation'** within time window
9) Select corresponding company name and date info with the max 'max_deviation' value and output

>> Code developed in spark_stock_max_deviation_all_given_win.py

Running Sample:

```spark-submit spark_stock_max_deviation_all_given_win.py -start <start_date> -end <end_date>```

>> spark-submit spark_stock_max_deviation_all_given_win.py -start 01/01/2016 -end 12/31/2016 >spark_output/output_stock_max_deviation_all_given_win.log

>> check full log file at attached output files

- **Find the maximum sell price in a day for a given company in the entire data set (No time window required).**

>> Logic:

1) Read arguments of company name
2) Construct dataframe structure
3) Read the csv file of given company as a spark dataframe
4) Select necessary columns
5) Find the **max 'high' value for the given company**
6) Find the corresponding date and output

>> Code developed in spark_stock_max_sell_by_given_company.py

Running Sample:

```spark-submit spark_stock_max_sell_by_given_company.py -company <company_name>```

>> spark-submit spark_stock_max_sell_by_given_company.py -company ADANIENT >spark_output/output_stock_max_sell_given_company.log

>> check full log file at attached output files

5. Documentation and README for running and grading the code. - 20 pts.

>> Check attached files