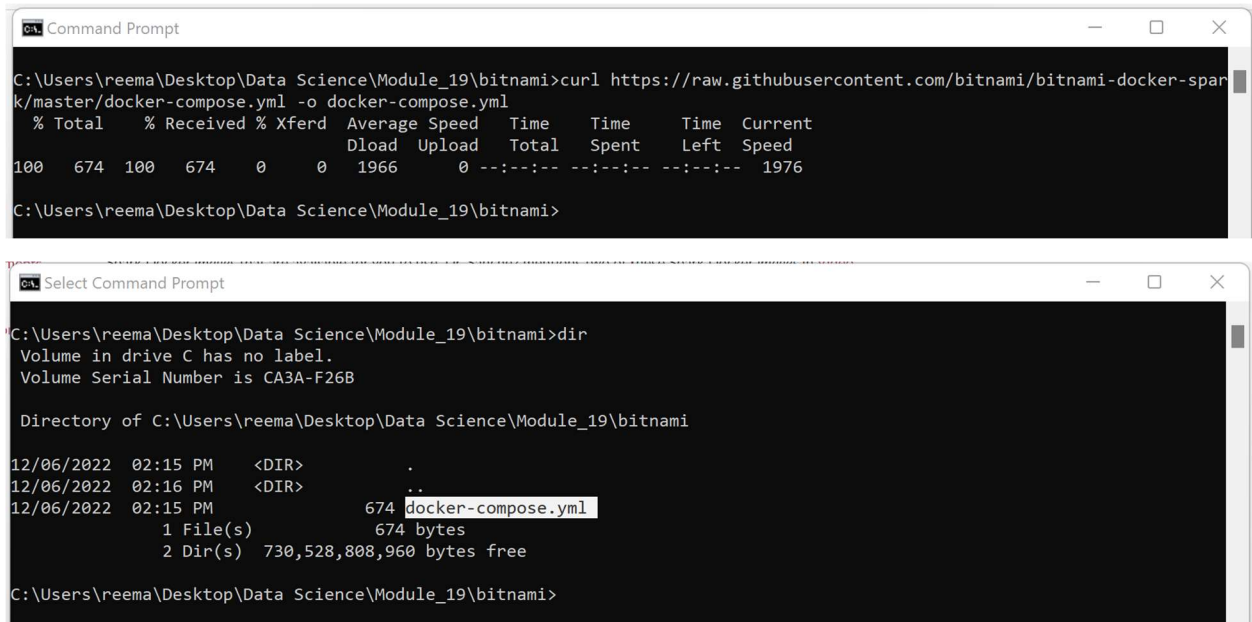


PySpark

1. Deployed Docker for Spark. Pulled docker *image* & listed contents to make sure docker-compose.yml file is present.



```
C:\Users\reema\Desktop\Data Science\Module_19\bitnami>curl https://raw.githubusercontent.com/bitnami/bitnami-docker-spark/master/docker-compose.yml -o docker-compose.yml
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 674    100 674    0    0  1966      0 --:--:-- --:--:-- --:--:-- 1976

C:\Users\reema\Desktop\Data Science\Module_19\bitnami>

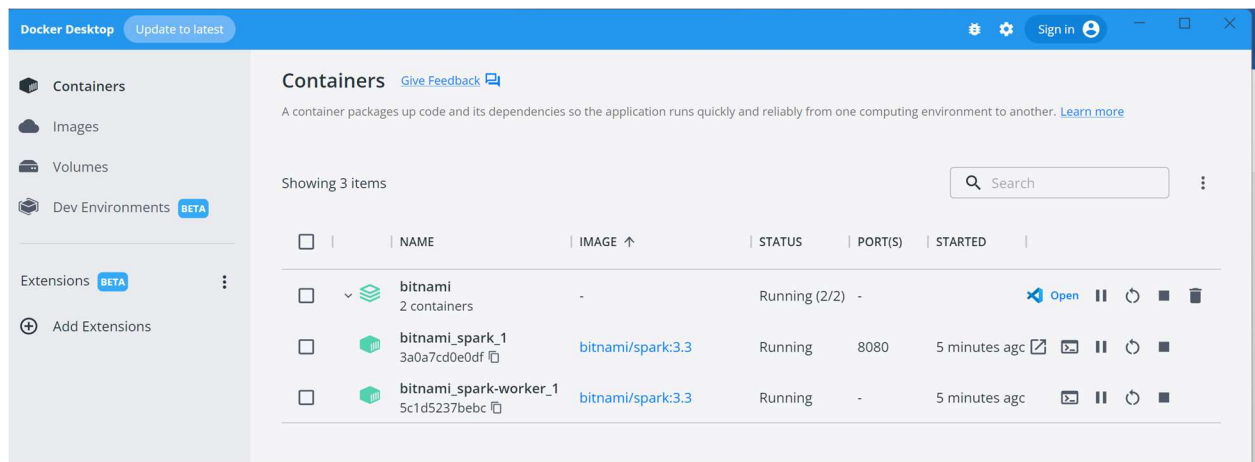
C:\Users\reema\Desktop\Data Science\Module_19\bitnami>dir
Volume in drive C has no label.
Volume Serial Number is CA3A-F26B

Directory of C:\Users\reema\Desktop\Data Science\Module_19\bitnami

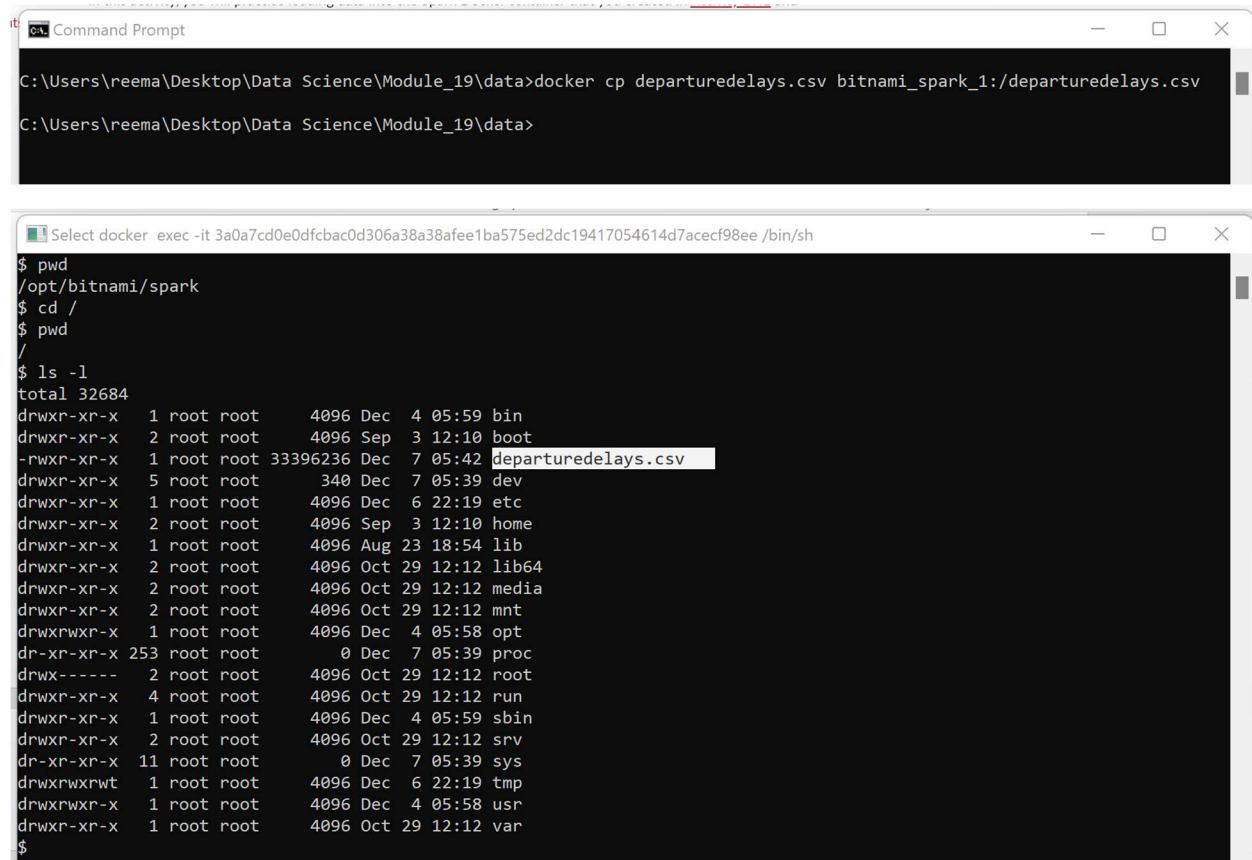
12/06/2022  02:15 PM  <DIR>          .
12/06/2022  02:16 PM  <DIR>          ..
12/06/2022  02:15 PM                674  docker-compose.yml
               1 File(s)                674 bytes
               2 Dir(s)  730,528,808,960 bytes free

C:\Users\reema\Desktop\Data Science\Module_19\bitnami>
```

2. Docker is up & running for Spark.



3. Copied the departeddelays.csv file to the bitnami_spark_1 container.



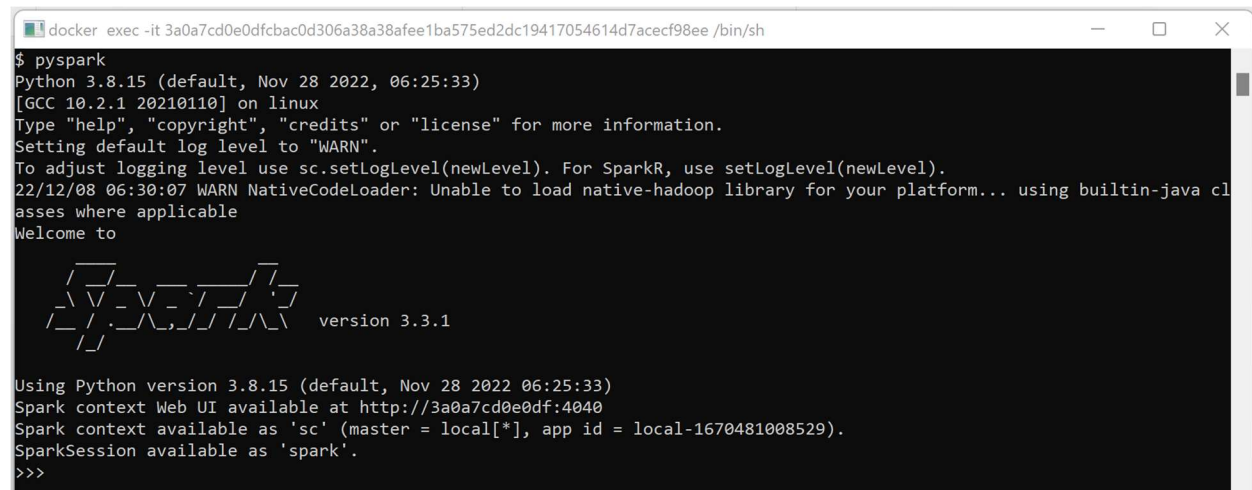
The first screenshot shows a Windows Command Prompt window with the following command and output:

```
C:\Users\reema\Desktop\Data Science\Module_19\data>docker cp departeddelays.csv bitnami_spark_1:/departuredelays.csv
C:\Users\reema\Desktop\Data Science\Module_19\data>
```

The second screenshot shows a terminal window inside a Docker container with the following commands and output:

```
$ pwd
/opt/bitnami/spark
$ cd /
$ pwd
/
$ ls -l
total 32684
drwxr-xr-x 1 root root 4096 Dec 4 05:59 bin
drwxr-xr-x 2 root root 4096 Sep 3 12:10 boot
-rwxr-xr-x 1 root root 33396236 Dec 7 05:42 departeddelays.csv
drwxr-xr-x 5 root root 340 Dec 7 05:39 dev
drwxr-xr-x 1 root root 4096 Dec 6 22:19 etc
drwxr-xr-x 2 root root 4096 Sep 3 12:10 home
drwxr-xr-x 1 root root 4096 Aug 23 18:54 lib
drwxr-xr-x 2 root root 4096 Oct 29 12:12 lib64
drwxr-xr-x 2 root root 4096 Oct 29 12:12 media
drwxr-xr-x 2 root root 4096 Oct 29 12:12 mnt
drwxrwxr-x 1 root root 4096 Dec 4 05:58 opt
dr-xr-xr-x 253 root root 0 Dec 7 05:39 proc
drwx----- 2 root root 4096 Oct 29 12:12 root
drwxr-xr-x 4 root root 4096 Oct 29 12:12 run
drwxr-xr-x 1 root root 4096 Dec 4 05:59/sbin
drwxr-xr-x 2 root root 4096 Oct 29 12:12/srv
dr-xr-xr-x 11 root root 0 Dec 7 05:39/sys
drwxrwxrwt 1 root root 4096 Dec 6 22:19/tmp
drwxrwxr-x 1 root root 4096 Dec 4 05:58/usr
drwxr-xr-x 1 root root 4096 Oct 29 12:12/var
$
```

4. Screenshot for successfully opening PySpark.



The screenshot shows a Docker terminal window with the following output:

```
docker exec -it 3a0a7cd0e0dfcbac0d306a38a38afee1ba575ed2dc19417054614d7acecf98ee /bin/sh
$ pyspark
Python 3.8.15 (default, Nov 28 2022, 06:25:33)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/12/08 06:30:07 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to

  ____      _
 / ___|    / \
| |  | |  / _ \
| |  | | / ___ \
| |  | || |_) |
| |  | || |_) |
| |  | || |_) |
|_|  |_| \____/

version 3.3.1

Using Python version 3.8.15 (default, Nov 28 2022 06:25:33)
Spark context Web UI available at http://3a0a7cd0e0df:4040
Spark context available as 'sc' (master = local[*], app id = local-1670481008529).
SparkSession available as 'spark'.
>>>
```

5. Imported Spark session & successfully defined it.

```
Select docker exec -it 3a0a7cd0e0dfcbac0d306a38a38afee1ba575ed2dc19417054614d7acecf98ee /bin/sh
$ pyspark
Python 3.8.15 (default, Nov 28 2022, 06:25:33)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/12/08 06:30:07 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Welcome to

  ____      _
 / ___|    / \
| |  | |  / _ \
| |  | | / ___ \
| |  | || |___| \
 \___|_||_|___|_|
                    version 3.3.1

Using Python version 3.8.15 (default, Nov 28 2022 06:25:33)
Spark context Web UI available at http://3a0a7cd0e0df:4040
Spark context available as 'sc' (master = local[*], app id = local-1670481008529).
SparkSession available as 'spark'.
>>> from pyspark.sql import SparkSession
>>>

docker exec -it 3a0a7cd0e0dfcbac0d306a38a38afee1ba575ed2dc19417054614d7acecf98ee /bin/sh
>>> spark = (SparkSession
...     .builder
...     .appName("Assignment19.3")
...     .getOrCreate())
22/12/08 06:37:29 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
>>>
```

6. Successfully defined the assignment19_3_data variable.

```
Select docker exec -it 3a0a7cd0e0dfcbac0d306a38a38afee1ba575ed2dc19417054614d7acecf98ee /bin/sh
^
SyntaxError: invalid character in identifier
>>> spark = (SparkSession
...     .builder
...     .appName("Assignment19.3")
...     .getOrCreate())
22/12/08 06:37:29 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
>>> assignment19_3_data="/departuredelays.csv"
>>>
```

7. Successfully defined the 'df' dataframe that contains all of the entries in the departuredelays.csv file.

```
Select docker exec -it 3a0a7cd0e0dfcbac0d306a38a38afee1ba575ed2dc19417054614d7acecf98ee /bin/sh
>>> spark = (SparkSession
...     .builder
...     .appName("Assignment19.3")
...     .getOrCreate())
22/12/08 06:37:29 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
>>> assignment19_3_data="/departuredelays.csv"
>>> df = (spark.read.format("csv")
...     .option("interSchema", "true")
...     .option("header", "true")
...     .load(assignment19_3_data))
>>>
```

8. Created a view of the assignment19_3_table *dataframe*.

```
docker exec -it 3a0a7cd0e0dfcbac0d306a38a38afee1ba575ed2dc19417054614d7acecf98ee /bin/sh
>>> df.createOrReplaceTempView("assignment19_3_table")
>>>
```

9. Ran below SQL queries in Spark.

a) Displayed the first 15 flights from PHL to DFW that had a delay of greater than 150 minutes.

```
docker exec -it 3a0a7cd0e0dfcbac0d306a38a38afee1ba575ed2dc19417054614d7acecf98ee /bin/sh
>>> spark.sql("""SELECT date,delay, origin, destination
... FROM assignment19_3_table
... WHERE delay > 150 AND ORIGIN = 'PHL' AND DESTINATION = 'DFW'
... ORDER BY CAST(delay AS INT) DESC""").show(15)
+-----+-----+-----+-----+
| date|delay|origin|destination|
+-----+-----+-----+-----+
|01141620| 1177| PHL| DFW|
|02200820| 741| PHL| DFW|
|02141850| 316| PHL| DFW|
|01022039| 295| PHL| DFW|
|03131830| 280| PHL| DFW|
|01011425| 279| PHL| DFW|
|01110820| 277| PHL| DFW|
|03171140| 249| PHL| DFW|
|02231425| 242| PHL| DFW|
|03281830| 222| PHL| DFW|
|02190933| 199| PHL| DFW|
|02211425| 198| PHL| DFW|
|01081855| 193| PHL| DFW|
|02211620| 192| PHL| DFW|
|03251425| 182| PHL| DFW|
+-----+-----+-----+-----+
only showing top 15 rows
>>>
```

b) Displayed the first 10 flights that have a distance of less than 200 miles

```
docker exec -it 3a0a7cd0e0dfcbac0d306a38a38afee1ba575ed2dc19417054614d7acecf98ee /bin/sh
>>> spark.sql("""SELECT date,delay, distance, origin, destination
... FROM assignment19_3_table
... WHERE distance < 200
... ORDER BY CAST( distance AS INT) DESC """).show(10)
+-----+-----+-----+-----+-----+
| date|delay|distance|origin|destination|
+-----+-----+-----+-----+-----+
|03051944| -4| 199| CVG| DTW|
|03041715| -7| 199| CVG| DTW|
|03051715| -1| 199| CVG| DTW|
|03021944| 3| 199| CVG| DTW|
|03031944| -6| 199| CVG| DTW|
|03041944| -4| 199| CVG| DTW|
|03051140| -4| 199| CVG| DTW|
|03021715| 0| 199| CVG| DTW|
|03061944| -2| 199| CVG| DTW|
|03031715| -2| 199| CVG| DTW|
+-----+-----+-----+-----+-----+
only showing top 10 rows
>>>
```

c) Displayed the first 10 flights that have a distance greater than 600 miles

```
docker exec -it 3a0a7cd0e0dfcbac0d306a38a38afee1ba575ed2dc19417054614d7acecf98ee /bin/sh
>>> spark.sql("""SELECT date, delay, distance, origin, destination
... FROM assignment19_3_table
... WHERE distance > 600
... ORDER BY CAST( distance AS INT) DESC """).show(10)
+-----+-----+-----+-----+-----+
| date | delay | distance | origin | destination |
+-----+-----+-----+-----+-----+
| 01090900 | -3 | 4330 | JFK | HNL |
| 01050900 | 98 | 4330 | JFK | HNL |
| 01080900 | 14 | 4330 | JFK | HNL |
| 01020900 | 1 | 4330 | JFK | HNL |
| 01040900 | 111 | 4330 | JFK | HNL |
| 01060900 | -2 | 4330 | JFK | HNL |
| 01070900 | 3 | 4330 | JFK | HNL |
| 01010900 | 6 | 4330 | JFK | HNL |
| 01110900 | -4 | 4330 | JFK | HNL |
| 01030900 | 784 | 4330 | JFK | HNL |
+-----+-----+-----+-----+-----+
only showing top 10 rows
>>>
```