**Processing Big Data with Spark and Airflow**

**Part 1: PySpark**

A new folder named "bitnami" was created. The command prompt window was opened and the "bitnami" folder was navigated to. The docker-compose.yml file was downloaded using the provided curl command.

A screenshot of a computer program

Description automatically generated

Containers were created using the "docker-compose up" command in a Terminal window. The successful pulling of the containers was demonstrated with a screenshot of Docker Desktop.

A screenshot of a computer

Description automatically generated

The departuredelays.csv file was downloaded, and the Docker copy command was used to transfer this file to the bitnami\_spark\_1 container.

A screen shot of a computer

Description automatically generated

PySpark was opened using the appropriate command in the CLI window.

A screenshot of a computer program

Description automatically generated

The correct package to initiate a PySpark session was imported in the CLI window.

A computer screen shot of a blue screen

Description automatically generated

A PySpark session named "spark" was defined with the appName argument set to "Assignment19.3".

A screenshot of a computer

Description automatically generated

A variable named "assignment19\_3\_data" was defined to hold the path to the departuredelays.csv file.

A screenshot of a computer screen

Description automatically generated

A dataframe, "df", containing all entries from the departuredelays.csv file was defined.

A blue screen with white text

Description automatically generated

Using the createOrReplaceTempView method on the "df" dataframe, a view named "assignment19\_3\_table" was created.

A screenshot of a computer

Description automatically generated

An SQL query was inputted to select the first 15 flights from Philadelphia International Airport (PHL) to Dallas Fort Worth International Airport (DFW) that had delays greater than 150 minutes.

A screenshot of a computer

Description automatically generated

Another SQL query was used to select the first 10 flights with a distance of fewer than 200 miles.

A screenshot of a computer

Description automatically generated

Finally, an SQL query was made to select the first 10 flights with a distance greater than 600 miles.

A screenshot of a computer

Description automatically generated

**Part 2: Airflow**

An empty folder named "airflow\_assignment" was created on the local machine. The "airflow\_assignment" folder was navigated to in a Terminal window, and the Airflow file was pulled using the given curl command.

A computer screen shot of a blue screen

Description automatically generated

The generated docker-compose.yaml file was opened using VS Code. Under "environment", the AIRFLOW\_\_CORE\_\_LOAD\_EXAMPLES variable was set to false.

A screenshot of a computer program

Description automatically generated

The command "docker-compose up" was run in the Terminal window, starting the Airflow Docker container. The running Airflow Docker containers were shown with a screenshot from Docker Desktop.

A screenshot of a computer

Description automatically generated

<https://localhost:8080/> was accessed using a web browser.

A screenshot of a computer

Description automatically generated

A successful login to Airflow using the default credentials was shown with a screenshot.

A screenshot of a computer

Description automatically generated

Inside the "airflow\_assignment" folder's "dags" directory, the "module19\_assignment.py" file was created using VS Code.

A screenshot of a computer

Description automatically generated

The required libraries for the DAG were imported using the given command.

A screenshot of a computer

Description automatically generated

The DAG was defined in the "module19\_assignment.py" file using the provided lines of code.

A screenshot of a computer

Description automatically generated

A Python function named "square()" was defined in the "module19\_assignment.py" file.

A screenshot of a computer program

Description automatically generated

A DAG object was defined in the "module19\_assignment.py" file using the provided code.

A screen shot of a computer

Description automatically generated

A task, "t1", was defined in the "module19\_assignment.py" file.

A screen shot of a computer

Description automatically generated

<https://localhost:8080/> was accessed to confirm the correct configuration of the DAG. The correct configuration was displayed in a screenshot of the Airflow UI.

A screenshot of a computer

Description automatically generated

The DAG was started, and its Graph View was accessed. Logs for the DAG were opened and viewed.

A screenshot of a computer

Description automatically generated