**Project Deployment and Execution in Databricks**

This document provides step-by-step instructions for deploying and executing the Spark jobs in a Databricks cluster.

**Pre-requisites**

Before proceeding with the deployment and execution of the Spark jobs, make sure you have the following:

* A Databricks account with appropriate permissions to create and run jobs.
* A Databricks cluster with the appropriate runtime version and configuration.
* Access to the source data and the location where the Delta table will be created.

**Deployment**

Follow these steps to deploy the project to your Databricks cluster:

1. Open the Databricks workspace and navigate to the "Workspace" tab.
2. Create a new notebook by clicking the "Create" button and selecting "Notebook" from the dropdown menu.
3. Give the notebook a name and select the appropriate language (e.g., Python).
4. Copy and paste the code for the Spark jobs into the notebook.
5. Save the notebook.

**Execution**

Follow these steps to execute the Spark jobs in your Databricks cluster:

1. Open the notebook that contains the Spark jobs.
2. Make sure that the necessary libraries and packages are installed by running the appropriate installation commands in the notebook.
3. Run the notebook to execute spark job.

**To Execute Spark job in standalone mode:**

1. Copy paste the code to job.py

2. Execute: spark-submit job.py

**Job Configuration**

To configure the Spark jobs, you can modify the parameters and arguments passed to the functions in the notebook. For example, you can change the name or location of the Delta table, or specify different options for reading and writing data.

**Job Scheduling**

To schedule the Spark jobs to run at specific times or intervals, you can create a job in Databricks. Follow these steps to create a job:

1. Open the Databricks workspace and navigate to the "Jobs" tab.
2. Click the "Create Job" button and fill in the job details, such as the name, schedule, and cluster configuration.
3. In the "Notebooks" section, select the notebook that contains the Spark jobs.
4. In the "Parameters" section, specify any parameters or arguments that need to be passed to the Spark jobs.
5. Save the job.

Once the job is created, it will run at the scheduled times or intervals and execute the Spark jobs in the specified notebook. You can monitor the job status and view the job logs in the Databricks UI.