**Week 5: Automating the Stock Market Monitoring and Prediction System with Azure DevOps**

**Prerequisites:**

1. **Azure DevOps Account**: Ensure you have a project set up in Azure DevOps.
2. **Azure Databricks Workspace**: Access an Azure Databricks workspace where your stock market monitoring notebooks and clusters are hosted.
3. **Service Principal or Personal Access Token (PAT) for Azure Databricks**: Generate a PAT in Databricks for authentication.
4. **Databricks CLI Installed and Configured**: Install the Databricks CLI on your local machine or CI agent for pipeline integration.

**Step 1: Set up the Databricks CLI**

1. **Install Databricks CLI**:

pip install databricks-cli

1. **Configure the Databricks CLI**:

databricks configure --token

* + **Databricks Host URL**: Use the URL of your Azure Databricks workspace.
  + **Token**: Generate a Personal Access Token (PAT) in Databricks for authentication.

**Step 2: Create an Azure DevOps Pipeline**

1. **Create a YAML Pipeline**: In your Azure DevOps project, create a new pipeline using YAML configuration. This pipeline will automate deployment and execution of your stock market monitoring and prediction notebooks.
2. **Add Variables**: In Azure DevOps, navigate to **Pipelines > Library** and add the following variables for Databricks configuration:
   * DATABRICKS\_HOST: The URL of your Azure Databricks workspace.
   * DATABRICKS\_TOKEN: The Personal Access Token.

**Step 3: Azure DevOps YAML Pipeline Example**

Here’s an example of an azure-pipelines.yml file for automating the deployment of the stock market monitoring system:

trigger:

- main

pool:

vmImage: 'ubuntu-latest'

variables:

DATABRICKS\_HOST: 'https://<databricks-instance>.azuredatabricks.net'

DATABRICKS\_TOKEN: $(databricksToken)

steps:

# Step 1: Install Python and Databricks CLI

- task: UsePythonVersion@0

inputs:

versionSpec: '3.x'

addToPath: true

- script: |

pip install databricks-cli

displayName: 'Install Databricks CLI'

# Step 2: Configure Databricks CLI

- script: |

databricks configure --host $(DATABRICKS\_HOST) --token $(DATABRICKS\_TOKEN)

displayName: 'Configure Databricks CLI'

env:

DATABRICKS\_HOST: $(DATABRICKS\_HOST)

DATABRICKS\_TOKEN: $(DATABRICKS\_TOKEN)

# Step 3: Upload Stock Monitoring Notebook to Databricks Workspace

- script: |

databricks workspace import ./notebooks/StockMarket\_notebook.py /Shared/StockMarket\_notebook -l PYTHON

displayName: 'Upload Notebook to Databricks Workspace'

# Step 4: Run Databricks Notebook

- script: |

JOB\_ID=$(databricks runs submit --json-file run\_config.json | jq -r '.run\_id')

echo "Job ID: $JOB\_ID"

databricks runs wait --run-id $JOB\_ID

displayName: 'Run Stock Market Monitoring Notebook'

**Explanation of the Pipeline**

1. **Trigger**: The pipeline automatically triggers when changes are pushed to the main branch.
2. **Pool**: Uses the latest Ubuntu image for the build environment.
3. **Install Python and Databricks CLI**: Installs Python and the Databricks CLI.
4. **Configure Databricks CLI**: Configures the CLI using DATABRICKS\_HOST and DATABRICKS\_TOKEN environment variables.
5. **Upload Notebook**: Uploads the StockMarket\_notebook.py to the Databricks workspace under /Shared/ directory.
6. **Run Notebook**: Submits the notebook for execution using the JSON file (run\_config.json), which defines cluster configurations and notebook parameters.

**Step 4: Run Databricks Notebook with JSON Config File**

A JSON configuration file (e.g., run\_config.json) defines the notebook parameters and cluster settings for running the stock market monitoring notebook.

**Sample run\_config.json**:

{

"run\_name": "Stock Market Notebook Run",

"new\_cluster": {

"spark\_version": "10.4.x-scala2.12",

"node\_type\_id": "Standard\_DS3\_v2",

"num\_workers": 2

},

"notebook\_task": {

"notebook\_path": "/Shared/StockMarket\_notebook",

"base\_parameters": {

"param1": "value1",

"param2": "value2"

}

}

}

**Summary of the Pipeline:**

* **Step 1**: Installs Python and Databricks CLI.
* **Step 2**: Configures the Databricks CLI using the host URL and token for authentication.
* **Step 3**: Uploads the StockMarket\_notebook.py to the Databricks workspace.
* **Step 4**: Runs the uploaded notebook using the configuration in run\_config.json, which includes cluster specifications and parameters for monitoring stock data.

**Key Points:**

* **Databricks CLI**: Used for interacting with Databricks to upload notebooks and run jobs.
* **Azure DevOps Variables**: Sensitive information such as tokens should be stored in Azure DevOps variable groups or as secrets.
* **Run Configuration**: The JSON file specifies the cluster details and parameters for notebook execution, ensuring proper monitoring of stock market data.