Week 5 – Automation via Azure DevOps (Theoretical)

Objective

Automate the **Retail Sales Performance Dashboard** analysis using Azure DevOps Pipelines. The pipeline should run weekly, execute the analysis scripts, generate sales insights, and highlight underperforming stores automatically.

Steps to Achieve This

Step 1: Create a Project in Azure DevOps

1. Go to dev.azure.com and sign in with your account.
2. Click on **New Project**.
3. Provide a project name (e.g., *RetailSalesAnalysis*).
4. Choose visibility (Public/Private).
5. Click **Create** to set up the project.

Step 2: Push Code to Azure Repos

1. Inside the project, navigate to the **Repos** section.
2. Copy the Git URL provided.
3. Clone the repository locally:
4. git clone <repo_url>
5. Add the following files into the repository:
    - Python/PySpark scripts for sales analysis.
    - requirements.txt file with dependencies.
    - Cleaned data (CSV/Delta).
    - azure-pipelines.yml (pipeline definition file).
6. Commit and push the files back to Azure Repos.

Step 3: Create a New Pipeline

1. In Azure DevOps, go to **Pipelines** → **New Pipeline**.
2. Select the repository (Azure Repos Git or GitHub).

3. Choose the branch where scripts are stored.
4. Select **Starter Pipeline** or point to the uploaded azure-pipelines.yml.

Step 4: Define Pipeline Stages in YAML

The YAML pipeline should include:

1. Checkout code from the repository.
2. Setup Python environment (install Python 3.x on the hosted agent).
3. Install dependencies listed in requirements.txt.
4. Pull cleaned data from the repo/storage.
5. Run sales analysis script to generate store performance insights.
6. Save results into CSV or log file.
7. Publish artifact (make the report downloadable).
8. (Optional) Upload results to Azure Storage for dashboard consumption.

Step 5: Configure Scheduling

1. Open the azure-pipelines.yml file.
2. Add **cron schedules** to automate runs.
    - Weekly: Run every Sunday night.
3. Example snippet:

```
schedules:
  # Weekly Run - Every Sunday at 18:00 UTC (Monday 11:30 PM IST)
  - cron: "0 18 * * 0"
    displayName: "Weekly Retail Sales Analysis"
    always: true
    branches:
      include:
        - main
    batch: true
```

Step 6: Output Reports

1. The pipeline runs the analysis script which generates:
    - Store-level sales summary.
    - Profit/revenue by product category.
    - List of top 5 lowest performing stores.
2. The report is saved as a CSV file.
3. The CSV is published as a **pipeline artifact**.
4. Team members can download the report from the pipeline summary.

Step 7: Alerts and Notifications

1. Script logs underperforming stores in the console output.
2. Azure DevOps logs capture this information.
3. (Optional) Use Azure DevOps **Notifications/Extensions** to send email alerts listing the lowest performing stores.