# CI/CD Automation with Azure DevOps

## Objective

To automate the data processing, report generation, and monitoring system for the Smart Home Energy Usage Tracker project using Azure DevOps CI/CD pipelines. This ensures all workflows (data fetching → cleaning → ETL → reporting → alerting) run automatically with continuous integration, deployment, and monitoring.

## EPIC: Automate Smart Energy Reporting and Monitoring

This Epic covers the full automation of energy data handling through CI/CD pipelines in Azure DevOps. It ensures end-to-end integration of Python, PySpark, and Databricks scripts into a single automated workflow that runs weekly, generates reports, and triggers alerts for abnormal energy usage.

A screenshot of a computer

AI-generated content may be incorrect.

## Feature 1: Setup Azure DevOps CI/CD Infrastructure

Establish the Azure DevOps environment — including repositories, pipelines, and environments — to manage continuous integration and deployment for the project.

A screenshot of a computer

AI-generated content may be incorrect.

### User Story 1.1: Configure Source Control and Repository

As a DevOps Engineer, I want to create and organize repositories in Azure Repos so that all scripts and configurations are version-controlled and easily accessible.

Acceptance Criteria: Repositories created for Python, PySpark, and Databricks scripts with access control.

A screenshot of a computer

AI-generated content may be incorrect.

#### Tasks:

T1.1.1: Create Azure DevOps Project and Repositories

T1.1.2: Define folder structure for source code and YAML pipelines

T1.1.3: Configure access control and permissions for contributors

### User Story 1.2: Build CI Pipeline for Script Validation

As a DevOps Engineer, I want to implement a CI pipeline that automatically validates and tests scripts to ensure code quality before deployment.

A screenshot of a computer

AI-generated content may be incorrect.

#### Tasks:

T1.2.1: Create YAML-based CI pipeline in Azure Pipelines

T1.2.2: Integrate Python linting tools (Pylint, pytest)

T1.2.3: Configure artifact publishing to pipeline storage

### User Story 1.3: Setup Continuous Deployment (CD) Pipeline

As a DevOps Engineer, I want a CD pipeline that deploys validated scripts to Databricks automatically for ETL and reporting.

A screenshot of a computer

AI-generated content may be incorrect.

#### Tasks:

T1.3.1: Define deployment stages in YAML

T1.3.2: Integrate Databricks CLI or API for script uploads

T1.3.3: Configure automatic trigger after CI success

T1.3.4: Test deployment execution logs

## Feature 2: Automate Weekly Data Report Generation

Design and automate a data pipeline that runs weekly to fetch, process, and summarize smart home energy data, producing reports automatically.

A screenshot of a computer

AI-generated content may be incorrect.

### User Story 2.1: Automate Data Ingestion and Cleaning

As a DevOps Engineer, I want to automate fetching and cleaning of energy data from MySQL and MongoDB so that weekly reports use accurate data.

A screenshot of a computer

AI-generated content may be incorrect.

#### Tasks:

T2.1.1: Develop Python script for weekly data fetching

T2.1.2: Integrate Pandas/Numpy code for data cleaning

T2.1.3: Schedule automated run in CI/CD pipeline

### User Story 2.2: Execute Databricks ETL for Data Aggregation

As a Data Engineer, I want to automate Databricks ETL jobs for daily and weekly energy summaries.

A screenshot of a computer

AI-generated content may be incorrect.

#### Tasks:

T2.2.1: Deploy Databricks notebook using CD pipeline

T2.2.2: Parameterize notebook for dynamic data ranges

T2.2.3: Save output results in Azure Blob Storage

### User Story 2.3: Automate Weekly Report Generation

As a Data Analyst, I want to generate automated weekly reports summarizing total and per-device energy usage.

A screenshot of a computer

AI-generated content may be incorrect.

#### Tasks:

T2.3.1: Create Python or PowerShell script for report generation

T2.3.2: Integrate report generation step in CD pipeline

T2.3.3: Store generated reports in Azure Storage or DevOps Artifacts

## Feature 3: Configure Alerting and Monitoring

Implement a monitoring and alerting system to track pipeline performance and detect abnormal energy usage in real time.

A screenshot of a computer

AI-generated content may be incorrect.

### User Story 3.1: Setup Energy Usage Alerts

As a Data Analyst, I want alerts when device energy usage exceeds 10 kWh/day, so I can identify high-consuming devices.

#### Tasks:

T3.1.1: Define threshold rules in DevOps variables

T3.1.2: Create Azure Logic App or Monitor alert flow

T3.1.3: Configure alert notification to stakeholders

### User Story 3.2: Monitor Pipeline Executions

As a Project Manager, I want visibility into the CI/CD pipelines to track their execution status and failures.

A screenshot of a computer

AI-generated content may be incorrect.

#### Tasks:

T3.2.1: Configure Azure DevOps dashboard with pipeline widgets

T3.2.2: Add metrics for success/failure rate and execution time

### User Story 3.3: Implement Pipeline Failure Notifications

As a DevOps Engineer, I want instant notifications when a pipeline fails, so I can take immediate corrective action.

A screenshot of a computer

AI-generated content may be incorrect.

#### Tasks:

T3.3.1: Add failure detection trigger in YAML

T3.3.2: Integrate Azure Notification System for alerts

T3.3.3: Test failure notifications during dry runs

**All Tasks:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

* **A screenshot of a computer

  AI-generated content may be incorrect.**

## Expected Deliverables

* Fully configured Azure DevOps project and repositories
* YAML CI/CD pipelines for energy data automation
* Weekly report automation and alert system
* Monitoring dashboards and notification system
* Documentation of pipeline setup and execution logs