

Week 5: CI/CD Automation with Azure DevOps – Execution Steps

Objective:

- 1.To automate the weekly energy monitoring workflow using Azure DevOps, which:
- 2.Loads smart home energy usage data.
- 3.Transforms and checks for overuse.
- 4.Triggers alerts for devices consuming more than a set threshold.
- 5.Automates execution via Azure DevOps pipelines.

Pre-requisites:

Before you begin, make sure you have:

-Authentication: SSH Key or Personal Access Token (PAT) configured.

-Tools Installed:

- Git
- Python 3.x
- VS Code or any preferred IDE

-Local Project Folder contains:

-Python script: alert_threshold.py

-Azure DevOps pipeline config file: azure-pipelines.yml

Step-by-Step Execution

Step 1: Create Python Script & YAML in Local Folder

- alert_threshold.py

```
import pandas as pd
```

```
def load_data():
```

```
    # Simulated smart home energy usage
```

```
    data = {
```

```
        'device_id': ['AC01', 'WM02', 'FR03'],
```

```
        'date': ['2025-07-01'] * 3,
```

```
        'kwh_used': [8.5, 12.4, 9.3]
```

```
    }
```

```
    return pd.DataFrame(data)
```

```
def transform_data(df):
```

```
    return df[df['kwh_used'] > 10]
```

```
def alert_high_usage(df):
```

```
    for _, row in df.iterrows():
```

```
        print(f"ALERT: {row['device_id']} used {row['kwh_used']} kWh on  
{row['date']}!")
```

```
def run_elt():
```

```
    df = load_data()
```

```
    high_usage = transform_data(df)
```

```
    alert_high_usage(high_usage)
```

```
if __name__ == "__main__":
```

run_elt()

- **azure-pipelines.yml**

trigger:

- main

pool:

vmImage: 'ubuntu-latest'

steps:

- task: UsePythonVersion@0

inputs:

versionSpec: '3.x'

- script: |

pip install pandas

python Smart_Home_Energy_Usage_Tracker/

displayName: 'Successfully Run'

- script: |

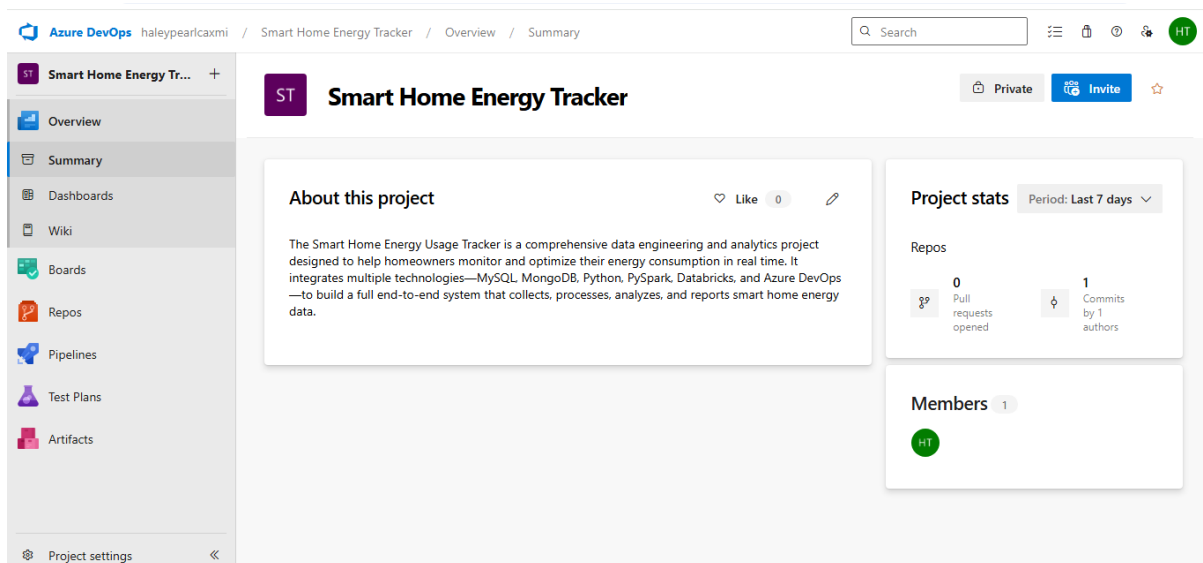
pip install pandas

python alert_threshold.py

displayName: 'Run ELT + Alert Script'

Step 2: Create Azure DevOps Project

1. Go to [Azure DevOps Portal](#).
2. Click "New Project".
3. Enter project name and visibility.
4. Click "Create".



Step 3: Push Local Code to Azure Repo via SSH

```
cd "C:\Users\User\Desktop\Hexaware\Phase-2 Role Based Training [Data Engineering]\Capstone Project\Smart Home Energy Usage Tracker"
```

```
git init
```

```
git add .
```

```
git commit -m "Initial commit"
```

```
git remote add origin
```

```
"git@ssh.dev.azure.com:v3/haleypcrlcaxmi/Smart%20Home%20Energy%20Tracker/Smart%20Home%20Energy%20Tracker"
```

```
git push -u origin main
```

```
C:\Users\User\Desktop\Hexaware\Phase-2 Role Based Training [Data Engineering]\Capstone Project\Smart Home Energy Usage Tracker>notepad C:\Users\User\.ssh\id_rsa.pub
C:\Users\User\Desktop\Hexaware\Phase-2 Role Based Training [Data Engineering]\Capstone Project\Smart Home Energy Usage Tracker>
C:\Users\User\Desktop\Hexaware\Phase-2 Role Based Training [Data Engineering]\Capstone Project\Smart Home Energy Usage Tracker>notepad C:\Users\User\.ssh\id_rsa.pub
C:\Users\User\Desktop\Hexaware\Phase-2 Role Based Training [Data Engineering]\Capstone Project\Smart Home Energy Usage Tracker>git push -u origin main
Enumerating objects: 60, done.
Counting objects: 100% (60/60), done.
Delta compression using up to 2 threads
Compressing objects: 100% (47/47), done.
Writing objects: 100% (60/60), 56.41 KiB | 1013.00 KiB/s, done.
Total 60 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Analyzing objects... (60/60) (173 ms)
remote: Validating commits... (1/1) done (1 ms)
remote: Storing packfile... done (150 ms)
remote: Storing index... done (43 ms)
remote: Updating refs... done (201 ms)
To ssh.dev.azure.com:v3/haleypearlca/mi/Smart%20Home%20Energy%20Tracker/Smart%20Home%20Energy%20Tracker
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

The screenshot shows the Azure DevOps web interface for the 'Smart Home Energy Tracker' repository. The left sidebar contains navigation options: Overview, Boards, Repos, Files (selected), Commits, Pushes, Branches, Tags, Pull requests, Advanced Security, Pipelines, and Project settings. The main area displays the 'Files' view for the 'main' branch. It shows a directory structure with folders for Week 1 through Week 5, and files like 'SmartHome SQL part.sql', 'SmartHome-SensorLogs', 'Data Preprocessing.py', 'devices.csv', 'energy_logs.csv', 'energy_per_device_numpy', 'room_energy_summary_p', 'rooms.csv', 'peak_offpeak_usage', and 'top energy devices'. A table of recent commits is also visible, showing commits by Haley Pearl Caxmi for each week.

Name	Last change	Commits
Week 1	7m ago	89cca1ef Initial commit Haley Pearl Caxmi
Week 2	7m ago	89cca1ef Initial commit Haley Pearl Caxmi
Week 3	7m ago	89cca1ef Initial commit Haley Pearl Caxmi
Week 4	7m ago	89cca1ef Initial commit Haley Pearl Caxmi
Week 5	7m ago	89cca1ef Initial commit Haley Pearl Caxmi
Smart_Home_Energy_Tracker.pdf	7m ago	89cca1ef Initial commit Haley Pearl Caxmi

Step 4: Configure and Run Azure Pipeline

1. In Azure DevOps, go to Pipelines → Create Pipeline.

2. *Select:*

Code source: Azure Repos Git

Your repository

Choose: "Existing Azure Pipelines YAML file"

3. *Specify:*

Branch: main

YAML Path: /Week 5/azure-pipelines.yml

4. Click Continue → Run.

ST Smart Home Energy Tr... +

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Library

Test Plans

Artifacts

Project settings <<

✓ Connect

✓ Select

Configure

Review

New pipeline

Configure your pipeline

Python package

Create and test a Python package on multiple Python versions.

Python to Linux Web App on Azure

Build your Python project and deploy it to Azure as a Linux Web App.

Starter pipeline

Start with a minimal pipeline that you can customize to build and deploy your code.

Existing Azure Pipelines YAML file

Select an Azure Pipelines YAML file in any branch of the repository.

.NET Core Function App to Windows on Azure

Build a .NET Core function app and deploy it to Azure as a Windows function App.

.NET Desktop

Build and run tests for .NET Desktop or Windows classic desktop solutions.

Android

Build your Android project with Gradle.

Ant

ST Smart Home Energy Tr... +

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Library

Test Plans

Artifacts

Project settings <<

✓ Connect

✓ Select

Configure

Review

New pipeline

Configure your pipeline

Python package

Create and test a Python package on multiple Python versions.

Python to Linux Web App on Azure

Build your Python project and deploy it to Azure as a Linux Web App.

Starter pipeline

Start with a minimal pipeline that you can customize to build and deploy your code.

Existing Azure Pipelines YAML file

Select an Azure Pipelines YAML file in any branch of the repository.

.NET Core Function App to Windows on Azure

Build a .NET Core function app and deploy it to Azure as a Windows function App.

.NET Desktop

Build and run tests for .NET Desktop or Windows classic desktop solutions.

Android

Build your Android project with Gradle.

Ant

Select an existing YAML file

Select an Azure Pipelines YAML file in any branch of the repository.

Branch

main

Path

/Week 5/azure-pipelines.yml

Select a file from the dropdown or type in the path to your file

Smart Home Energy Tracker

Cancel

Continue

ST Smart Home Energy Tr... +

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Library

Test Plans

Artifacts

Project settings <<

✓ Connect

✓ Select

✓ Configure

Review

New pipeline

Review your pipeline YAML

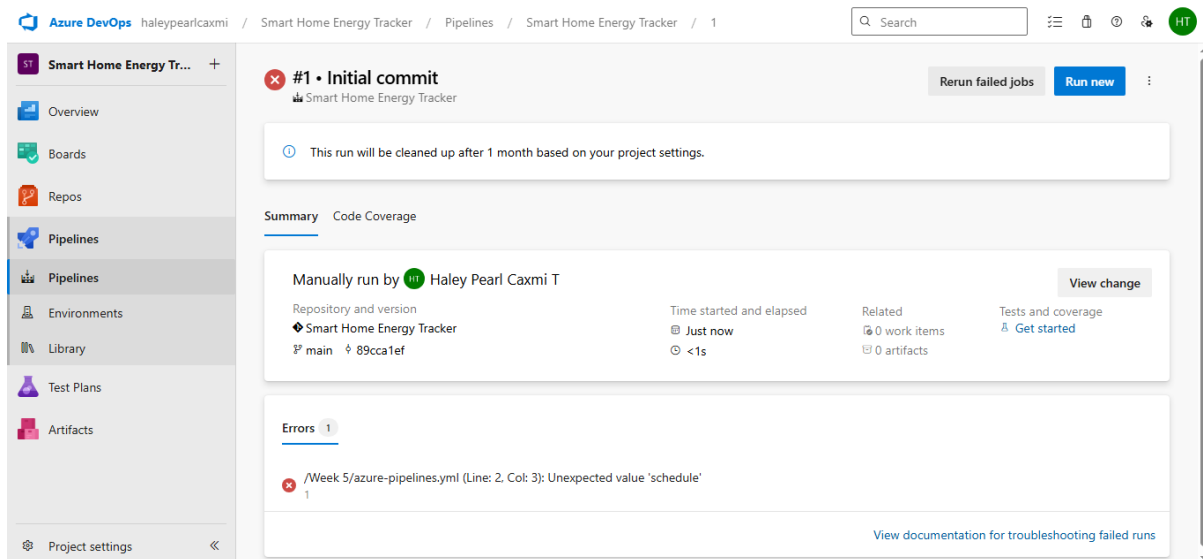
Variables

Run

Smart Home Energy Tracker / Week 5/azure-pipelines.yml

Show assistant

```
1 trigger:
2   - schedule:
3     - cron: "0 6 * * 1" # Every Monday at 6 AM
4     - displayName: Weekly Run
5   - branches:
6     - include:
7       - main
8
9 pool:
10  - vmImage: 'ubuntu-latest'
11
12 variables:
13   - dataPath: 'data/sensor_logs.csv'
14   - reportPath: 'output/weekly_report.csv'
15   - alertThreshold: 10 # kWh per device per day
16
17 steps:
18   - task: UsePythonVersion@0
```



Final Output

1. Azure DevOps pipeline automatically runs:

- Sets up Python environment.
- Installs dependencies (pandas).
- Executes `elt_energy_alert.py`.

2. Log displays:

- "Successfully Run"
- "Run ELT + Alert Script"

If energy consumption > 10 kWh for any device, alert message appears:

- ALERT: WM02 used 12.4 kWh on 2025-07-01!