CREATING FOR CI/CD WITH PYTEST USING GITHUB CODESPACES, GITHUB ACTIONS AND AWS CODEBUILD

1) WHY TEST YOUR LOGIC/CODE?

BRIEF SUMMARY

Testing your code or logic is a vital step data engineers take to ensure that their code or logic works before being pushed to production.

Whether you are deploying for a DevOps team (Building Applications), DataOps team (Creating data pipelines for ETL (Extract, Transform & Load) applications) or MLOps team (Building machine learning algorithms), it is vital to ensure that your code or logic works.

WHY IT IS IMPORTANT?

Why it is vital to test your code or logic is to ensure that it works to prevent future problems after deployment and the best way to test your code or logic is to create CI/CD pipelines.

WHAT ARE CI/CD PIPELINES?

CI stands for **Continuous Integration**, and there are several ways to test your code or logic using continuous integration. With cloud-based developer environments like **GitHub code spaces**, we will use the latter to test our code or logic using a **MAKE FILE**. I prefer using cloud developer environments because they have everything we need to test our code or logic.

CD stands for **Continuous Delivery** and we will use **GitHub Actions** using a **YML file** to test then we will clone our repo in AWS (Amazon Web Services) using the **AWS CloudShell** before testing in **AWS CodeBuild** which is used to push our logic or code to production.

This will ensure that any change we make in our developer environment will automatically create a change in deployment in AWS CodeBuild and the YML file makes it possible because without it, we cannot replicate what we have from GitHub Actions.

Creating CI/CD test is one interesting skill of a data engineer. It makes sure that the code you generate to build data pipelines, build applications, or develop machine learning algorithms does not fail after deployment.

2) STRUCTURE FOR TESTING?

A) CREATE A VIRTUAL ENVIRONMENT

What is a Virtual Environment and why is it important?

B) CREATE THE requirement.txt file

The requirements file will contain all the installations for used for the test.

C) CREATE THE make file

What is the make file what does function mean in the make file.

D) CREATE THE py files before the test.py files

E) IMPORT THE TESTING csv FILES AND CREATE A FOLDER FOR THE csv FILES.

3) **DO THE** MAKE install, test, lint, format All, **AND MAKE EVERYTHING WORK.**

4) TEST USING GITHUB ACTIONS

5) CLONE IN AWS CloudShell AND CONTINOUSLY DEPLOY INTO AWS CodeBuild.