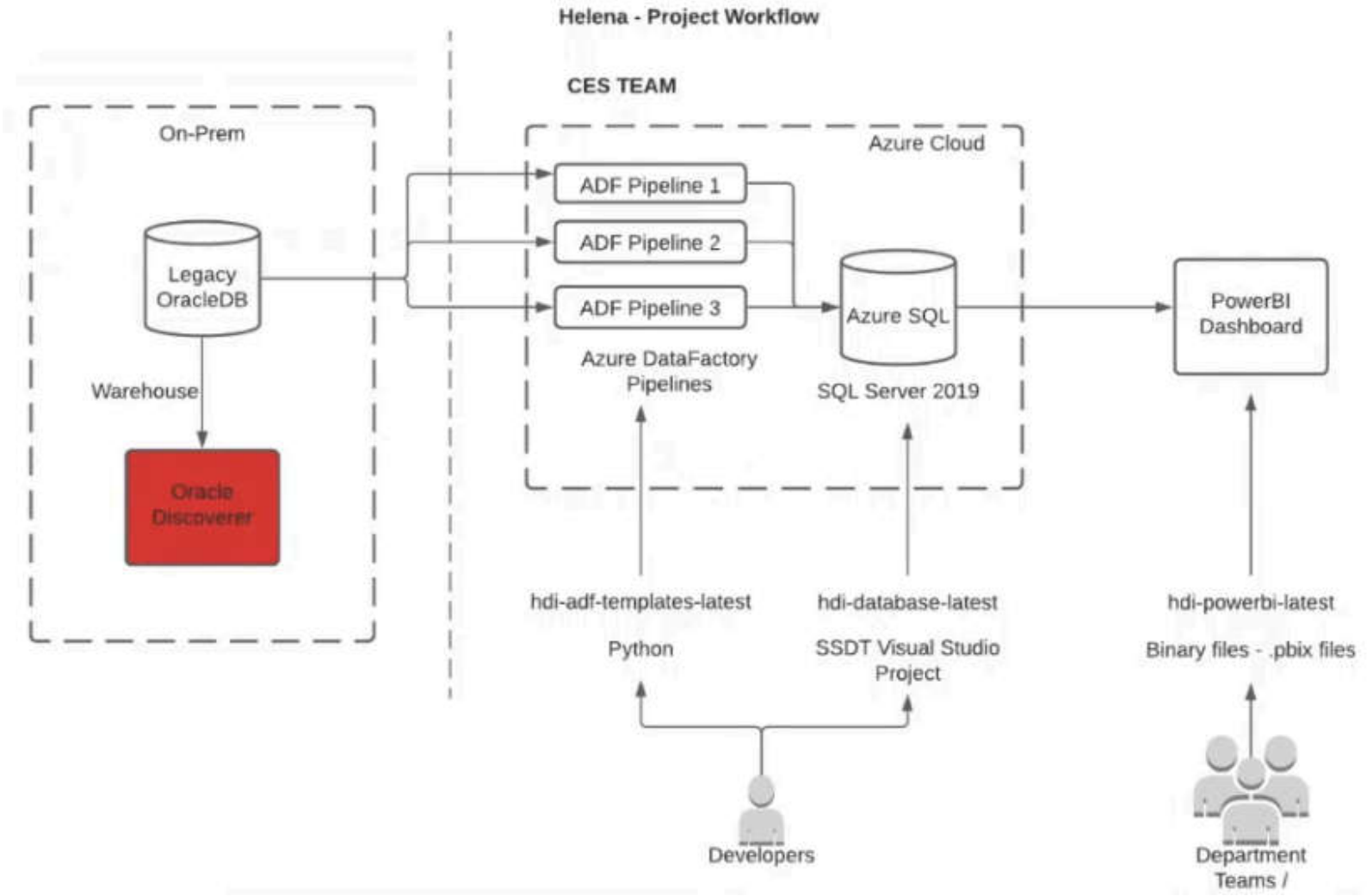
Project 1.0



Introduction:

A Client had **Legacy OrableDB** that was **on-prem** with **SQL tables** to run **analytics**. There was a need of a service to run analytics hence the client had chosen **Azure Cloud** with **ADF (Azure Data Factory)**.

Business Requirement:
The business requirement was to pull the data from the on-prem DB, store on Azure Database SQL & run analytics.

Environment: Azure

ADF Pipelines - the pipelines required source : **Legacy OracleDB** & destination : **Azure SQL DB server**.

ADF Pipelines are the **data pipelines**.
One of the pipelines was to fetch the OracleDB a specific table's data and store into Azure SQL (**Extract and Load**).

To connect the OracleDB to the Azure SQL, one needs to use **Linked Services** of **ADF**.

Data Engineer Developers usually provide us with the python scripts for the stored procedure, copy the data, lookup etc, we store them onto Azure Repos and directly use as an artifact to deply onto **ADF** through the pipelines.

We had 4 environments: Dev, QA, UAT & Prod
Promoto to UAT, Prod

UAT stands for User Acceptance Testing that includes multiple functionalities;

Functional Testing: It verifies and validates all the features, functions work according to the specified requirements and design documents.

Usability Testing: It focuses on evaluating the user interface (UI), ease of use & overall user experience.

Regression Testing: It ensures the new changes or updates should not impact negatively on the existing functionalities. It also involves retesting of the previously tested features to confirm they still work correctly.

Performance Testing: It assesses the performance of the features in various conditions like normal usage patterns, peak loads & stress conditions & identifies the bottlenecks & areas of optimization.

Security Testing: It aims to identify the vulnerabilities and weaknesses in authentication, authorization, data protection & other security measurement parameters

Compatibility Testing: It verifies the software words correctly though different devices, browsers, platforms, operating systems etc

Project 2.0

Helena Javascript Azure Functions Backend CI/CD Project

Practical https://youtu.be/14vQUd-oktg

Code

https://github.com/azdevops00/Private_Repos_200/tree/Helena_Azure_Fucntions_Backend_CI/CD_Code_201

Steps

1 Resource Group

`Helena_Resource_Group1`

2 Function App

`Helena_Function_App1`

3 Project > Repository > `Helena_Java_Backend_Repo1`

4 Build Pipeline

`Helena_Java_Backend_CI_Classic_P1`

5 Release Pipeline

`Helena_Java_Backend_CD_Classic_P1`

Project 3.0

.Net Azure WebApp Frontend CI/CD Project

GITHUB

https://github.com/azdevops00?tab=repositories

Download & install (Build Tool) : **.NET SDK x64**

https://dotnet.microsoft.com/en-us/download

DEVELOPERS RESPONSIBILITIES

 # 1. Create a local directory

```
mkdir project
cd project
```

 # 2. Create a solution file locally (Project is a set of files, Solution is a set of Projects)

```
dotnet new sln -o HelloWorldApp
cd HelloWorldApp
ls
    HelloWorldApp.sln
```

 # 3. Create an MVC project locally

```
dotnet new mvc -n HelloWorldApp.Web
ls
    HelloWorldApp.sln
    HelloWorldApp.Web
```

```
cd HelloWorldApp.Web
ls
    HelloWorldApp.Web.csproj
```

 # 4. Add the solution to the project

```
cd ..
dotnet sln add HelloWorldApp.Web/HelloWorldApp.Web.csproj
```

 # 5.Build & test the project locally (.NET supports debug & release configuration for builds)

```
dotnet restore
dotnet build
dotnet build --configuration debug
dotnet build --configuration release

cd HelloWorldApp.Web
dotnet bin/Release/net8.0/HelloWorldApp.Web.dll
        http://localhost:5000 Run in browser
```

 # 6. Create an output of the project

```
cd ..
dotnet publish -o c:/out
```

 # 7. Create an output of the Debug/Release configuration

```
dotnet publish -c release -o c:/out
cd c:/out
dotnet HelloWorldApp.Web.dll
        http://localhost:5000 Run in browser
```

 # 8. Initialise HelloWorldApp dir into local GIT Repository

```
cd ..
cd Users/Aasuz/..project/HelloWorldApp
git init
ls -a
    .git/  HelloWorldApp.Web/  HelloWorldApp.sln

code .
Press "F1", type "add gitignore" Enter, "VisualStudio"
Enter

git status
git add .
git status
git commit -m '1st Commit'
git status
```

DEVOPS RESPONSIBILITIES