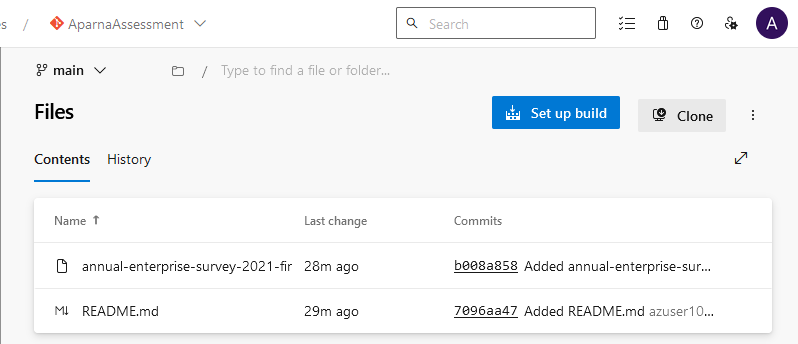
**APARNA BHARTI**

**DATA ENGINEERING BATCH-1**

**AZURE ASSESSMENT**

**Que 1) Create Azure Devops Environment and configure Azure Devops Git Repository ,configure on your local git to implement this upload few test files on the same.**

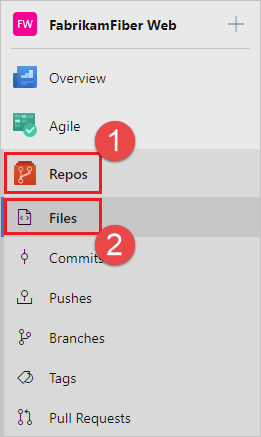
Azure Devops Environment

****

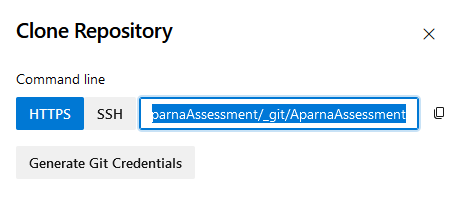
To work with a Git repo, you clone it to your computer. Cloning a repo creates a complete local copy of the repo for you to work with. Cloning also downloads all commits and branches in the repo and sets up a named relationship with the repo on the server.

**Steps to clone a repository into your local - :**

1 ) From your web browser, open the team project for your organisation in Azure DevOps and select **Repos** > **Files**. If you don't have a team project, create one now.

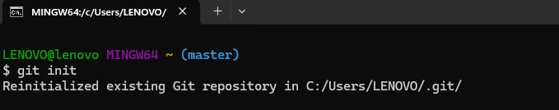


2 ) Select **Clone** in the upper-right corner of the **Files** window and copy the clone URL.

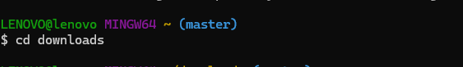


3 ) Run below commands in Git Bash

* The **git init** command creates a new Git repository. It can be used to convert an existing, unversioned project to a Git repository or initialise a new, empty repository.



* Switch your directory to the folder where you want your data to be cloned



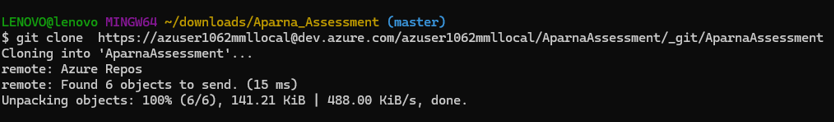
* The mkdir (make directory) command in the Microsoft Windows is **used to make a new directory**.



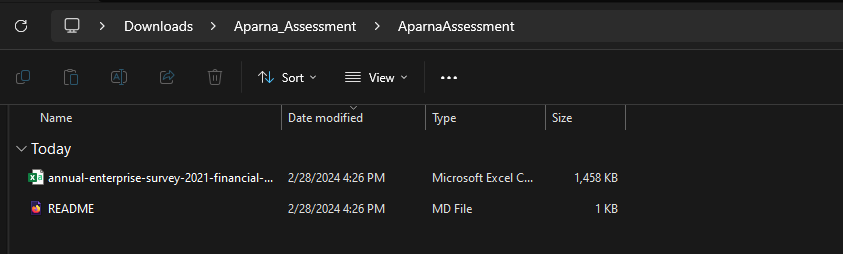
* Switch your directory to the folder that you have initially created



* Open the Git command window (Git Bash on Git for Windows). Then, browse to the folder where you want the code from the repo stored on your computer. Run git clone followed by the path copied from the **Clone URL**

****

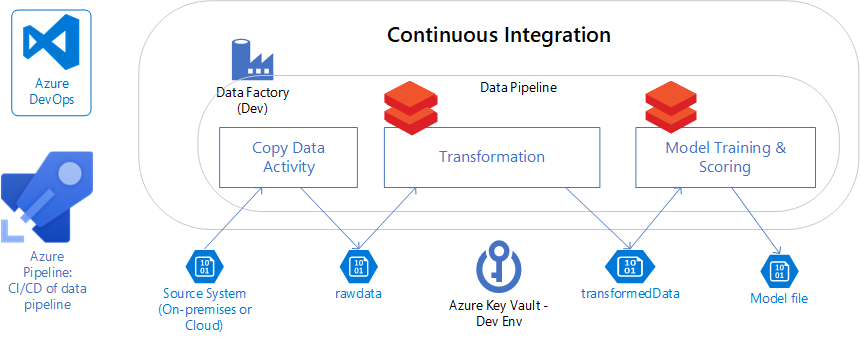
4 ) A new Folder named “AparnaAssessment” got created into local where Git repository got cloned.



**Que2) Leverage the practises of CICD Using Azure Data Engineering and explain the architecture of the Azure synapse .**

## **Azure CI CD Pipelines**

CI-CD Pipelines are part of the bigger picture of the DevOps framework, where continuous Integration and Continuous Deployment work together. Software companies are shifting their goals to ensure that bugs and errors are caught beforehand in the delivery process. So, CI-CD works together to make this process happen quickly and smoothly.



**Here's a breakdown of how to implement CI/CD practices within your Azure Data Engineering workflow:**

**1. Setting Up the Foundation:**

* **Version Control System:** Begin by integrating a version control system like Git into your workflow. This allows you to track code changes, revert to previous versions, and collaborate effectively with other data engineers. Azure DevOps offers seamless integration with Git repositories.
* **Development Environment:** Choose your preferred development environment within Azure Data Factory (ADF) or Azure Synapse Workspace. Both offer features like code editing, debugging, and version control integration.

**2. Building the CI/CD Pipeline:**

* **Azure DevOps Pipelines:** Utilise Azure DevOps Pipelines to create an automated CI/CD pipeline for your data engineering tasks. This pipeline will orchestrate the various stages of the process, including:
  + **Source Control:** Access code and configurations stored in your Git repository.
  + **Compilation and Unit Testing:** Compile your code (e.g., Python, SQL) and run unit tests to ensure its functionality. Integrations with testing frameworks like Pytest or NUnit are available through Azure DevOps.
  + **Artefact Creation:** Package your code and configurations into deployable artefacts, such as Data Factory pipelines (JSON), Notebook files, or DACPACs (for database deployments).

**3. Implementing Release Management:**

* **Multi-Stage Environments:** Create separate development, testing, and production environments within Azure Synapse or Data Factory. This allows you to test and validate changes before deploying them to production.
* **Deployment:** Integrate deployment tasks within your pipeline to deploy the created artefacts to the respective environments. This may involve using ADF's deployment triggers or deploying notebooks/DACPACs through Azure DevOps tasks.
* **Testing and Approvals:** Implement automated and manual testing stages as needed. For instance, use Azure Data Factory's built-in testing framework for pipelines or run custom tests within notebooks. Consider incorporating manual approval gates before deploying to production for critical changes.

**4. Continuous Improvement:**

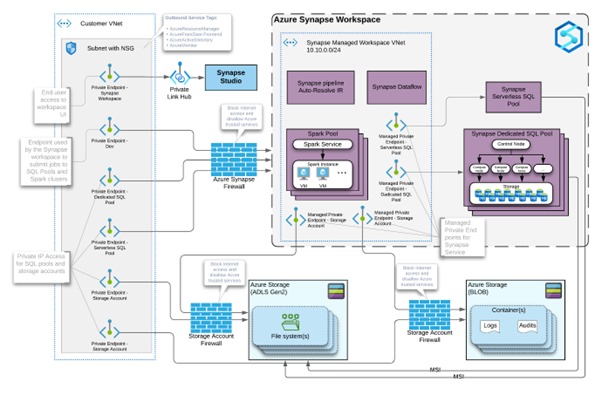
* **Monitoring and Logging:** Monitor your pipeline executions through Azure DevOps dashboards and logs. This helps identify any issues during the build, deployment, or testing stages.
* **Feedback and Optimization:** Analyse the monitoring data and feedback from stakeholders to continuously improve your CI/CD process. This might involve optimising pipeline performance, adding new tests, or refining your deployment strategy.

**Benefits of CI/CD for Azure Data Engineering:**

* **Increased Efficiency:** Automation streamlines the software development lifecycle, saving time and effort compared to manual deployments.
* **Improved Quality:** Robust testing throughout the pipeline reduces the risk of bugs and ensures data quality in production.
* **Enhanced Collaboration:** Version control and shared pipelines facilitate collaboration, clear communication, and clear ownership of code changes.
* **Faster Time to Market:** CI/CD allows for faster release cycles and quicker delivery of new data features and functionalities.

**Architecture of the Azure synapse**

Azure Synapse Analytics is **a unified analytics platform that brings together data integration, enterprise data warehousing, and big data analytics**. It gives you the freedom to query data on your terms, using either serverless or dedicated options—at scale.



**Azure Synapse Analytics architecture within Microsoft Azure, focusing on its key components and their interactions:**

**Data Storage:**

* **Azure Data Lake Storage Gen2 (ADLS Gen2):** This serves as the central storage repository for your data in Azure Synapse. It's a highly scalable and secure data lake solution capable of handling structured, semi-structured, and unstructured data formats.

**Compute Resources:**

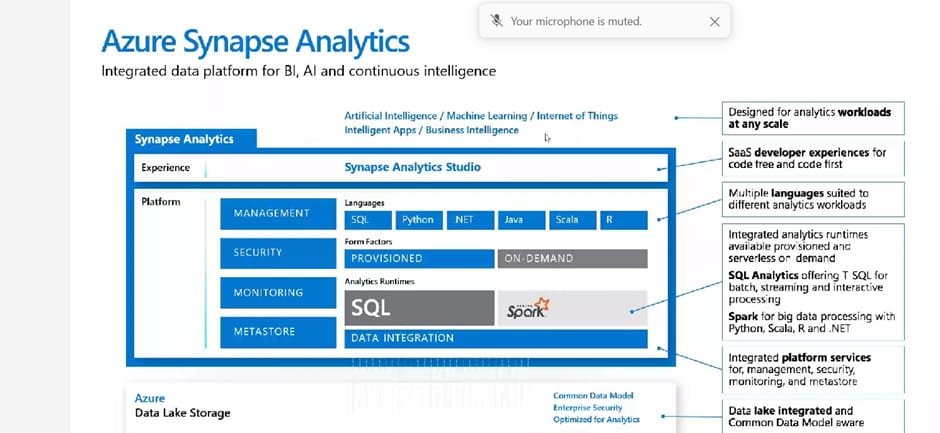
* **Dedicated SQL Pools (formerly SQL Data Warehouse):** These offer a massively parallel processing (MPP) architecture designed to handle complex queries and aggregations on large datasets efficiently. They leverage columnar storage for data compression and optimised retrieval.
* **Serverless SQL Pools:** This cost-effective option allows you to perform ad-hoc analysis and exploration directly within your ADLS Gen2 data lake. They utilise T-SQL for querying, similar to Dedicated SQL Pools, but operate on a serverless model, scaling resources automatically based on your needs.
* **Apache Spark Pools:** This component provides a powerful big data processing engine suitable for tasks like data extraction, transformation, loading (ETL), data cleansing, data science, and machine learning. It supports various programming languages such as Scala, Python, Spark SQL, and .NET (C#).

**Integration and Orchestration:**

* **Synapse Pipelines:** This acts as the central orchestration engine for data integration and transformation workflows. It allows you to design and execute pipelines visually or through code, automating tasks like data movement, manipulation, and cleansing within your Synapse workspace.

**User Experience:**

* **Synapse Studio:** This web-based workspace serves as the unified interface for interacting with all components within Azure Synapse. It provides functionalities for development, monitoring, management, collaboration, and data exploration.

****

**Additional Components:**

* **Synapse Notebooks:** These are interactive notebooks specifically designed for data exploration, analysis, and visualisation within Synapse Studio. They integrate seamlessly with other Synapse components, allowing you to combine code, data exploration, and results in a single environment.
* **Azure Data Explorer (Kusto Explorer):** This is a fast, web-based tool for interactive exploration and visualization of data stored in Azure Data Lake Storage or Synapse Analytics. It allows users to gain insights and create reports.

**Key Points:**

* **Unified Architecture:** Azure Synapse features a unified architecture, meaning all components are designed to work together seamlessly, offering a single platform for data warehousing, big data analytics, and data integration needs.
* **Scalability:** The platform offers scalability across storage and compute resources. You can independently scale your data storage in ADLS Gen2 and choose between dedicated or serverless compute options (Dedicated SQL pools and Serverless SQL pools) based on your performance and cost requirements.
* **Flexibility:** This architecture allows for flexible deployment models based on your specific requirements. You can choose dedicated or serverless compute options, leverage various programming languages for data processing, and utilise different tools for data exploration and visualisation.