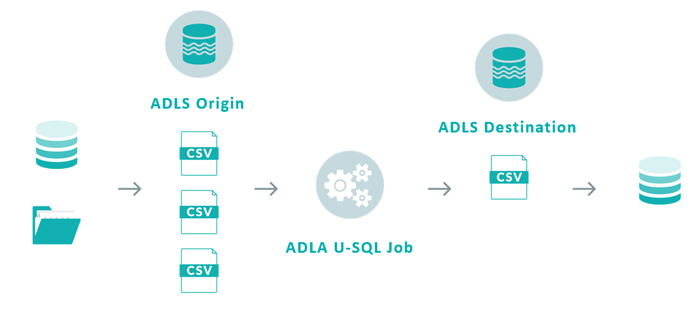
**1. Azure Data Lake Analytics**

Azure Data Lake is an on-demand scalable cloud-based storage and analytics service. It can be divided in two connected services, Azure Data Lake Store (ADLS) and Azure Data Lake Analytics (ADLA). ADLS is a cloud-based file system which allows the storage of any type of data with any structure, making it ideal for the analysis and processing of unstructured data.

Azure Data Lake Analytics is a parallelly-distributed job platform which allows the execution of U-SQL scripts on Cloud. The syntax is based on SQL with a twist of C#, a general-purpose programming language first released by Microsoft in 2001.

The general idea of ADLA is based on the following schema:



Text files from different sources are stored in Azure Data Lake Store and are joined, manipulated and processed in Azure Data Lake Analytics. The results of the operation are dumped into another location in Azure Data Lake Store.

ADLA jobs can only read and write information from and to Azure Data Lake Store. Connections to other endpoints must be complemented with a data-orchestration service such as Data Factory.

**2. HDInsight**

Azure HDInsight is a cloud service that allows cost-effective data processing using open-source frameworks such as Hadoop, Spark, Hive, Storm, and Kafka, among others.

Using Apache Sqoop, we can import and export data to and from a multitude of sources, but the native file system that HDInsight uses is either Azure Data Lake Store or Azure Blob Storage.

Of all Azure’s cloud-based ETL technologies, HDInsight is the closest to an IaaS, since there is some amount of cluster management involved. Billing is on a per-minute basis, but activities can be scheduled on demand using Data Factory, even though this limits the use of storage to Blob Storage.

**3. Databricks**

Azure Databricks is the latest Azure offering for data engineering and data science. Databricks’ greatest strengths are its zero-management cloud solution and the collaborative, interactive environment it provides in the form of notebooks.

Databricks is powered by Apache Spark and offers an API layer where a wide span of analytic-based languages can be used to work as comfortably as possible with your data: R, SQL, Python, Scala and Java. The Spark ecosystem also offers a variety of perks such as Streaming, MLib, and GraphX.

Data can be gathered from a variety of sources, such as Blob Storage, ADLS, and from ODBC databases using Sqoop.

**4. Azure ETL showdown**

Let’s look at a full comparison of the three services to see where each one excels:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **ADLA** | **HDInsight** | **Databricks** |
| Pricing | Per Job | Per Cluster Time | Per Cluster Time (VM cost + DBU processing time) |
| Engine | Azure Data Lake Analytics | Apache Hive or Apache Spark | Apache Spark, optimized for Databricks since founders were creators of Spark |
| Default Environment | Azure Portal, Visual Studio | Ambari (HortonWorks), Zeppelin if using Spark | Databricks Notebooks, RStudio for Databricks |
| De Facto Language | U-SQL, (Microsoft) | HiveQL, open source | R, Python, Scala, Java, SQL, mostly open-source languages |
| Integration with Data Factory | Yes, to run U-SQL | Yes, to run MapReduce jobs, Pig, and Spark scripts | Yes, to run notebooks, or Spark scripts (Scala, Python) |
| Scalability | Easy, based on Analytics Units | Not scalable, requires cluster shutdown to resize | Easy to change machines and allows autoscaling |
| Testing | Tedious, each query is a paid script execution, and always generates output file (Not interactive) | Easy, Ambari allows interactive query execution (if Hive). If using Spark, Zeppelin | Very easy, notebook functionality is extremely flexible |
| Setup and managing | Very easy as computing is detached from user | Complex, we must decide cluster types and sizes | Easy, Databricks offers two main types of services and clusters can be modified with ease |
| Sources | Only ADLS | Wide variety, ADLS, Blob and databases with sqoop | Wide variety, ADLS, Blob, flat files in cluster and databases with sqoop |
| Migratability | Hard, every U-SQL script must be translated | Easy as long as new platform supports MapReduce or Spark | Easy as long as new platform supports Spark |
| Learning curve | Steep, as developers need knowledge of U-SQL and C# | Flexible as long as developers know basic SQL | Very flexible as almost all analytic-based languages are supported |
| Reporting services | Power BI | Tableau, Power BI (if using Spark), Qlik | Tableau, open-source packages such as ggplot2, matplotlib, bokeh, etc. |

**5. Use case in all three platforms**

Now, let’s execute the same functionality in the three platforms with similar processing powers to see how they stack up against each other regarding duration and pricing:

In this case, let’s imagine we have some HR data gathered from different sources that we want to analyse. On the one hand, we have a .CSV containing information about a list of employees, some of their characteristics, the employee source and their corresponding performance score.

On the other hand, from another source, we’ve gathered a .CSV that tells us how much we’ve invested in recruiting for each platform (Glassdoor, Careerbuilder, Website banner ads, etc).

Our goal is to build a fact table that aggregates employees and allows us to draw insights from their performance and their source, to pursue better recruitment investments.