MATILLION MASTERCLASS

Let the Data Do The Talking...

ANALYTICS
WITH ANAND
LET THE DATA DO THE TALKING

Course Introduction – Matillion Masterclass

What does the course offer ?

- ❖ Extensively cover Matillion as a cloud native data integration platform and its offering
- ❖ Understand the Architecture of Matillion
- ❖ Practical explanation of how various component work in Matillion
- ❖ Mini Project Demonstrating practical implementation of Matillion as a Comprensive cloud native ETL Solution

Audience ?
Data Engineers, ETL Engineers, Data Analyst and anyone with willingness to learn Matillion

Course Prequisite

- ❖ Recommended to have previous ETL and AWS knowledge but not essential

Course Takeaways

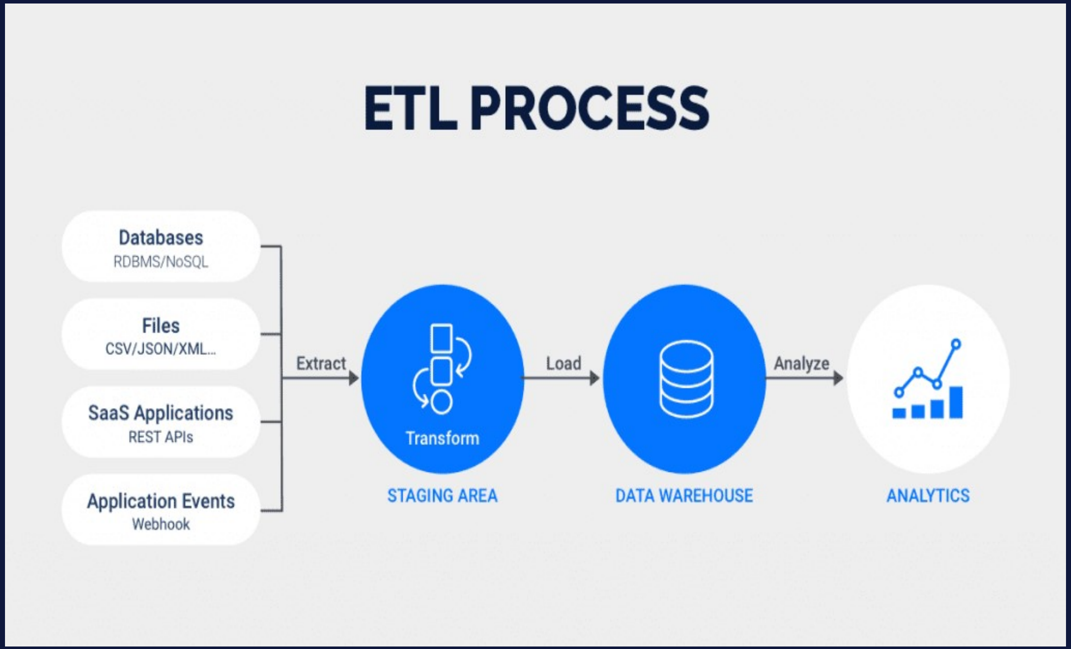
- ❖ Matillion Architecture and how it works
- ❖ Practical knowledge and hands-on experience of Matillion

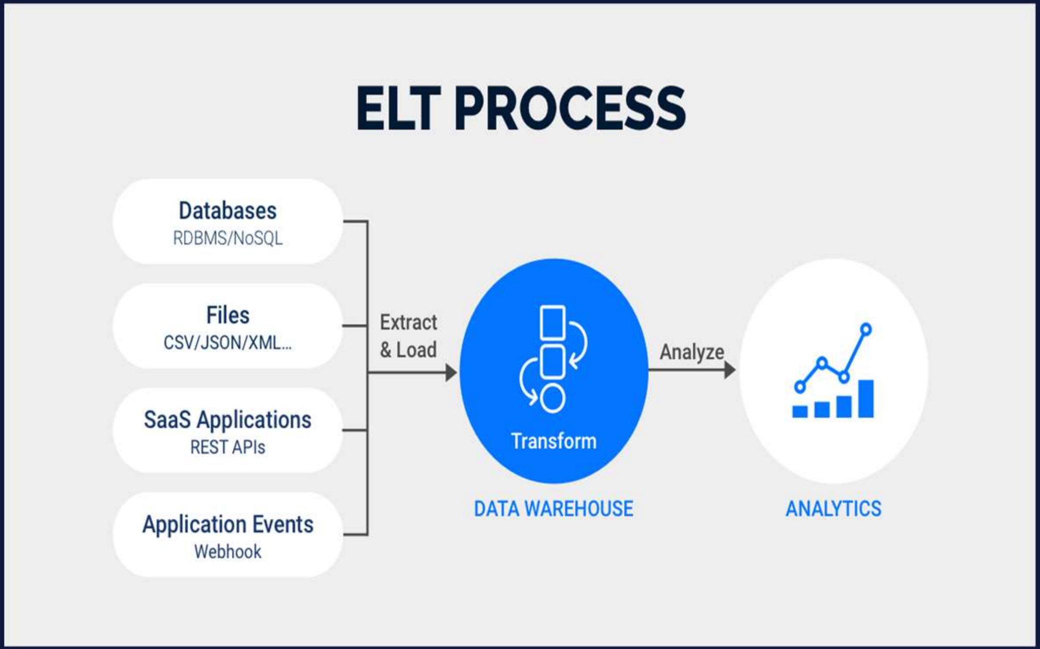
Section 2 – Matillion

- What is ETL / ELT ?
- Matillion Overview
- Competitors
- Product Offering
- Architecture
- Matillion Setup

What is ETL / ELT ?

ETL and ELT are two data integration approaches that are commonly used to move data from one system to another. ETL stands for Extract, Transform, Load, while ELT stands for Extract, Load, Transform. The key difference between these two approaches is the order in which data is transformed.





Introduction to Matillion

- What is Matillion?
  - Matillion is a cloud-native data integration platform that enables organizations to extract, load and transform (ELT) data from various sources into their data warehouses, such as Amazon Redshift, Google BigQuery, and Snowflake. It is designed to help businesses accelerate their data integration processes and achieve faster time-to-insight
  - Matillion provides a visual, drag-and-drop interface that allows users to easily create ETL pipelines without writing any code. It also offers a range of pre-built connectors for popular data sources, including Salesforce, Google Analytics, and Facebook Ads
  - Matillion is a popular choice among data engineers, data analysts, and other data professionals looking to streamline their data integration workflows and increase their data agility.

Matillion Product Offering

Matillion ETL

Matillion ETL is purpose-built for the cloud to migrate and continually load all of your data. It transform your data into an analytics-ready state directly within your cloud data environment.

- ❖ Matillion ETL for Amazon Redshift
- ❖ Matillion ETL for Snowflake
- ❖ Matillion ETL Delta lake on Databricks
- ❖ Matillion ETL for Google BigQuery
- ❖ Matillion for Synapse

Matillion Data Loader

Matillion Data Loader is a no-code, SaaS-based data integration tool that extracts data from almost any data source and ingests it via batch loading or change data capture (CDC) into your cloud data platforms. This is a lightweight product for quickly and easily migrating data from on-premises or cloud-based sources to a cloud data warehouse. It provides a simple, wizard-driven interface for transferring data without the need for complex ETL processes.

Overall, Matillion's product offerings are focused on helping organizations with cloud-based data integration, providing powerful visual interfaces and pre-built connectors to common data sources.

Matillion ETL

- **Unlocks the power of your data warehouse**: Matillion ETL pushes down data transformations to your cloud data warehouse, and processes millions of rows in seconds, with real-time feedback.
- **Offers a modern, browser-based environment**: Including features such as collaboration, version control, full-featured graphical job development, and dozens of read, write, join, and transform components.
- **Is fast to set up**: Launch and be developing ETL jobs within minutes

Important features

- Drag-and-drop browser interface. Build sophisticated, powerful ETL/ELT jobs.
- Push-down ELT technology uses the power of your data warehouse to process complex joins over millions of rows in seconds.
- Live feedback, validation, and data previews, inside the application, as you build your ETL/ELT jobs.
- Collaboration baked-in. Build jobs together in different locations.
- Version control, import/export, and server-side undo.
- Over 80 out-of-the-box connectors to popular online services that you can connect to and pull your data from, not to mention the ability to create your own connector.
- An easy-to-use user interface and basic functions so you can build your first job in minutes.
- In-client support.
- Enterprise features such as generated documentation.

Data Sources

- Amazon Web Services (AWS) services, including Amazon S3, Amazon Redshift, Amazon RDS, Amazon DynamoDB, Amazon EMR, and Amazon Aurora
- Google Cloud Platform (GCP) services, including Google BigQuery, Google Cloud Storage, and Google Cloud SQL
- Microsoft Azure services, including Azure Blob Storage, Azure Data Lake Storage, and Azure SQL Database
- Databases such as MySQL, PostgreSQL, Oracle, Microsoft SQL Server, and MongoDB
- Salesforce, SAP, NetSuite, and other enterprise applications
- Social media platforms such as Facebook, Twitter, and LinkedIn
- Web services and APIs, including REST and SOAP APIs
- Flat files such as CSV, Excel, and JSON

Destination

- Amazon Redshift
- Google BigQuery
- Snowflake
- Microsoft Azure Synapse Analytics
- Deltalake on Databricks

Competitors

Matillion operates in the highly competitive data integration and ETL market. Some of the main competitors of Matillion include:

- ❖ Talend
- ❖ Informatica
- ❖ SnapLogic
- ❖ Alooma
- ❖ Fivetran
- ❖ Alteryx
- ❖ AWS
- ❖ Microsoft
- ❖ Qlik

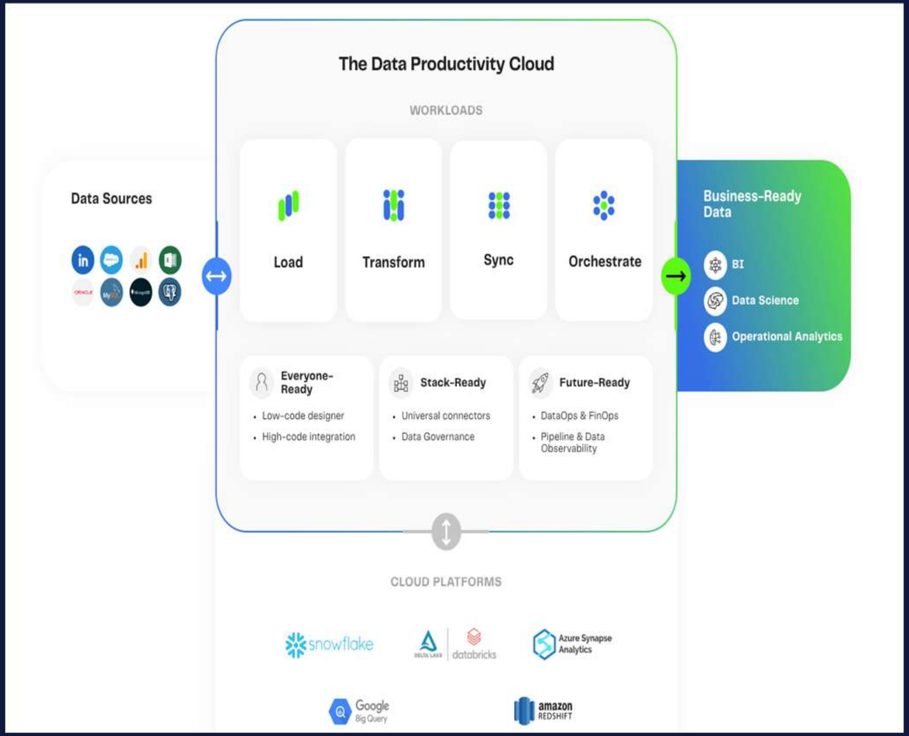
These companies, along with several others, offer similar data integration and ETL capabilities to Matillion and are considered its main competitors in the market

Matillion – The Data Productivity Cloud

Matillion Data Productivity Cloud is a cloud-based platform that helps data teams move, transform, and orchestrate data pipelines faster and more easily. It is a unified platform that can be used by both coders and non-coders, and it is designed to scale to meet the needs of any organization.

Some of the key benefits of Matillion Data Productivity Cloud include:

- ❖ Simplify and automate data movement: Matillion makes it easy to connect to a wide variety of data sources, and it can automate the process of moving data between them. This can save data teams a significant amount of time and effort.
- ❖ Bridge the skills gap for data transformation: Matillion provides a drag-and-drop interface that makes it easy for non-coders to transform data. This can help organizations to close the skills gap in their data teams and make it easier to get data into production.
- ❖ Handle the complexity of pipeline orchestration: Matillion provides a powerful orchestration engine that can manage complex data pipelines. This can help organizations to scale their data pipelines and ensure that they are reliable and efficient.



Key Components

Matillion instance: This is the main component of the Matillion architecture and represents the server or virtual machine that runs the Matillion software. The Matillion instance can be launched on a cloud computing platform such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP). The Matillion instance is responsible for running Matillion ETL jobs, managing data connections, and providing access to the Matillion user interface.

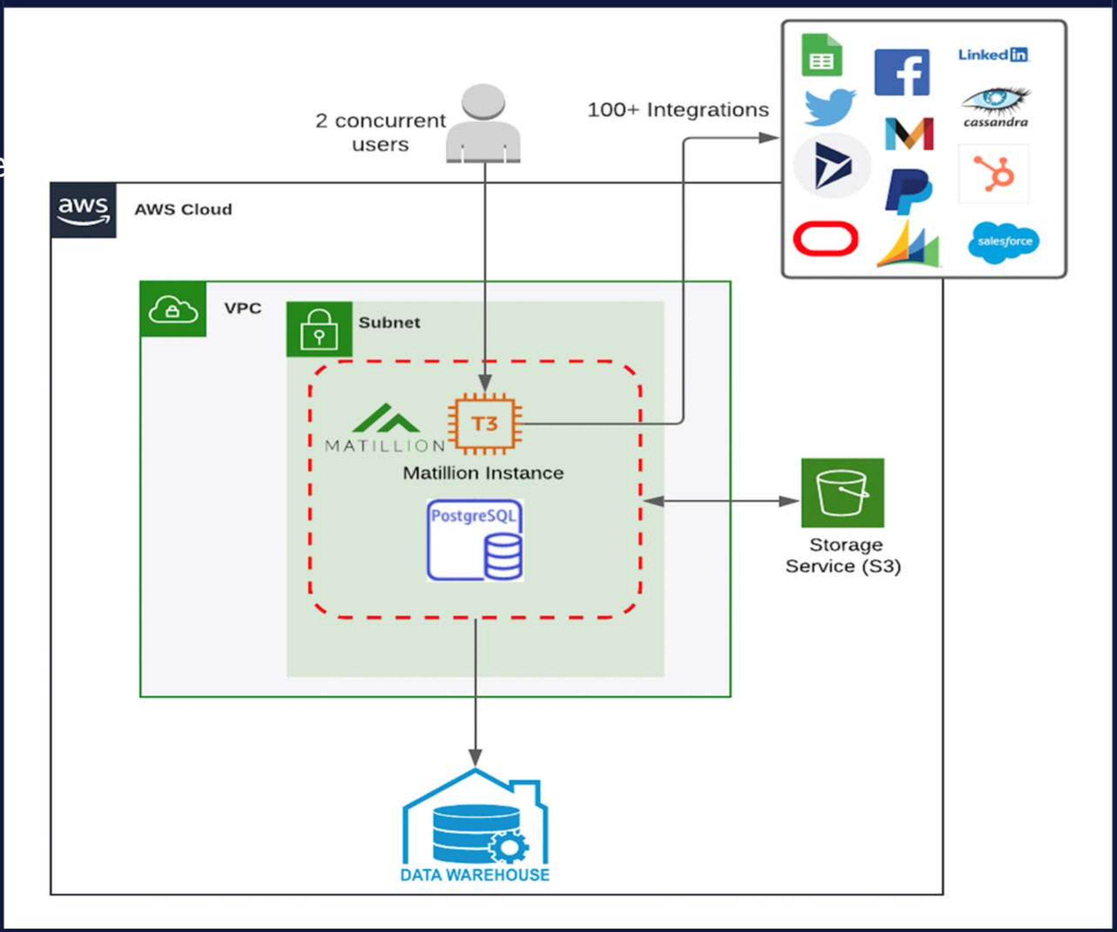
Cloud data warehouse: This is the target data warehouse where the data will be loaded after being transformed by Matillion ETL. Matillion supports a range of cloud data warehouses, including Amazon Redshift, Google BigQuery, and Snowflake.

Source data: This is the data that will be extracted by Matillion ETL from a range of data sources, including databases, flat files, and cloud-based applications.

Matillion Architecture – Part 1

Single Node

- This is the simplest Matillion ETL implementation. It consists of a single Matillion server that runs all of the ETL processes. This architecture is easy to set up and manage, but it can be limited in terms of scalability and performance.
- Typically used when trialling Matillion ETL
- Simplest and easiest deployment type
- Included Resources
  - t3.medium EC2 Instance running Matillion
  - IAM Role (attached to EC2 Instance)

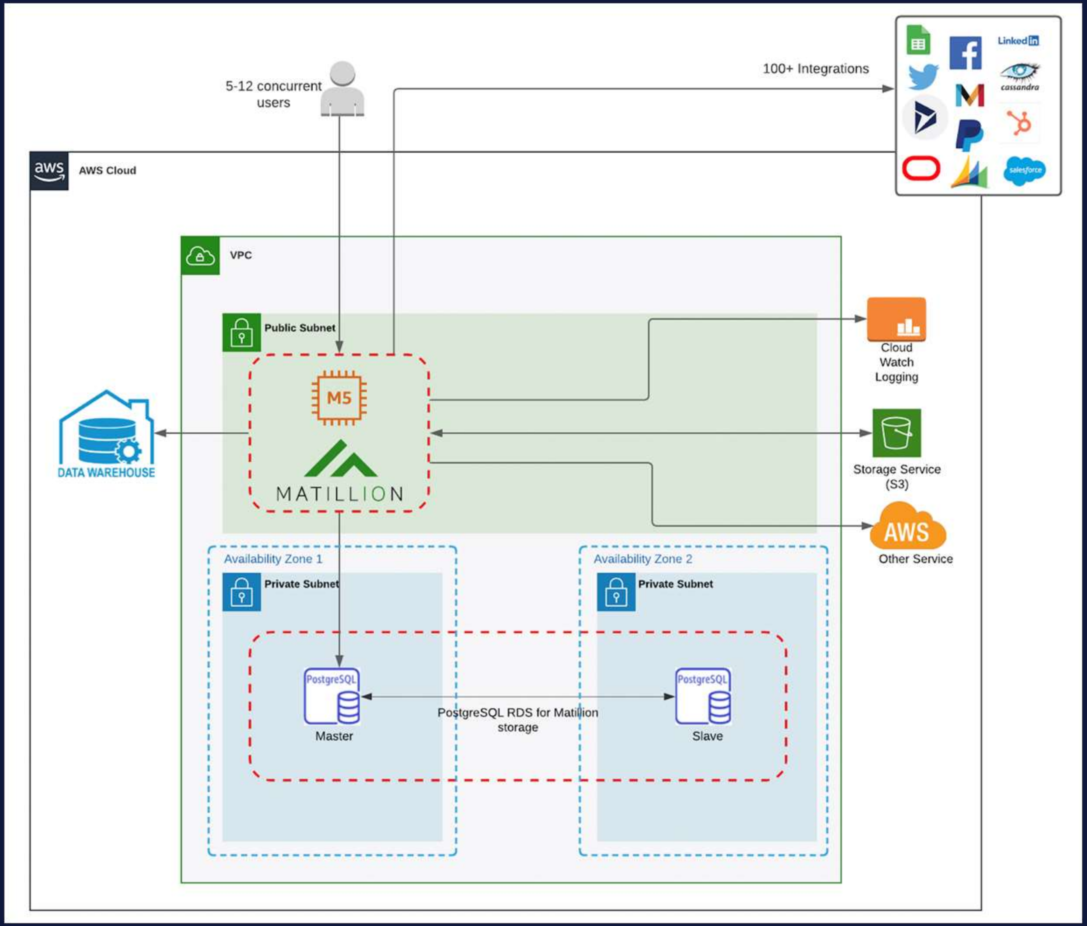


Matillion Architecture – Part 1 - Contd

Single node & RDS architecture:

- This architecture adds a Relational Database Service (RDS) database to the single node architecture. The RDS database is used to store Matillion metadata, such as job configurations and data lineage. This architecture provides a more scalable and reliable solution than the single node architecture, but it is also more complex to set up and manage

- Included Resources:
  - m5.large|m5.xlarge EC2 Instance running Matillion
  - IAM Role (attached to EC2 Instance)
  - Postgres RDS Instance with Multi-AZ Failover
  - Cloudwatch-based logging



Matillion Architecture – Part 2

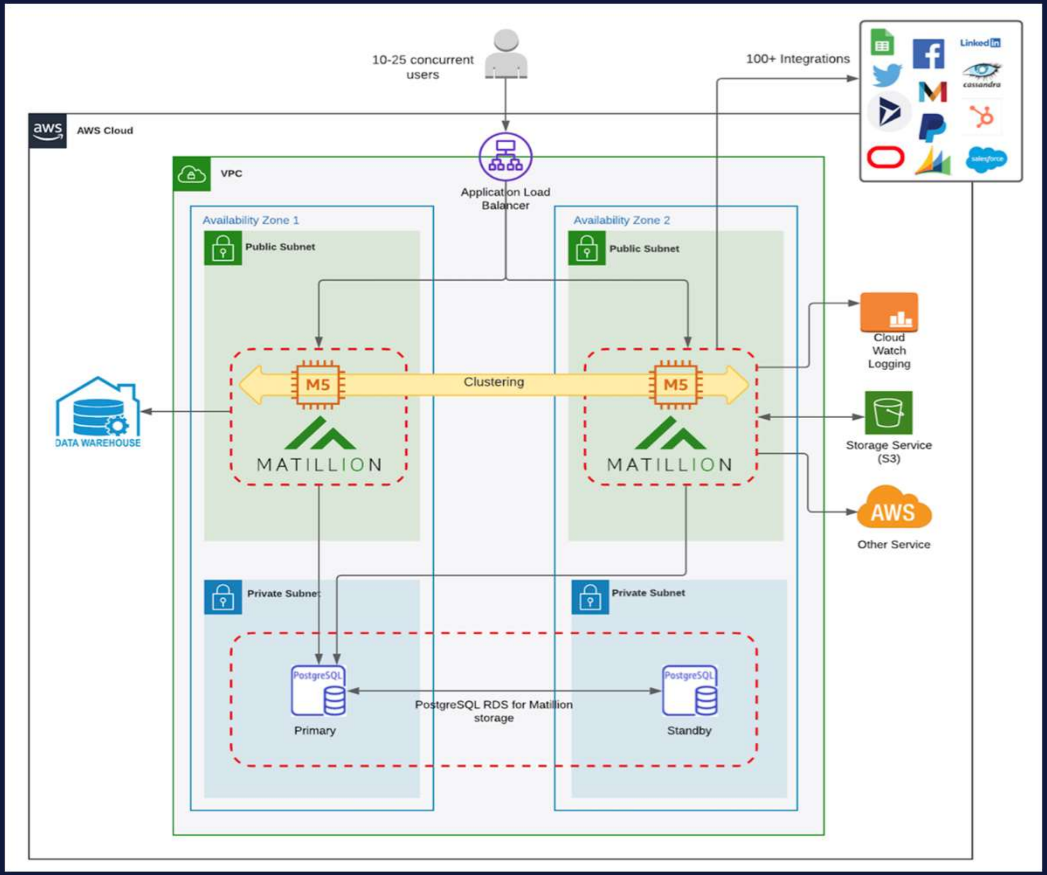
Clustered deployment type or Multinode

- ❖ Multiple Matillion instance in different availability zone
- ❖ External Postgres DB used for failover
- ❖ Load balance used to direct
- ❖ Typically used when no downtime

Included Resources

- 2x m5.large|m5.xlarge EC2 Instances running Matillion in a Cluster
- IAM Role (attached to EC2 Instance)
- Postgres RDS Instance with Multi-AZ Failover
- Application Load Balancer
- Cloudwatch-based logging

Note: This requires that an LDAP or Active Directory Based Directory Service already exists in your cloud platform account.



Certification Centric Summary – ETL Basic 15%

- Environment variables can be shared between multiple jobs
- Matillion instance needs to be running for all the scheduled job to complete
- The search tab exists on the Matillion ETL interface in the lower-right panel. Through this tab, the user can perform a search (partial match) on any term
- Job variables will override any environment variables of the same name within that specific job
- Version locking allows you to protect a version of your project, This feature can prevent accidental changes to important versions such as a "production" or "live" version
- Schedular in matillion can only launch orchestration job
- Environment variables can be shared between different jobs while job variables have a scope of a job
- Grey color connector is uncondition connector means execution goes to next component irrespective of success or failure
- Green connector is for success and red is for failure
- Data source in query component allows to select specific worksheet
- 10 is the maximum number of version recommended by Matillion
- Main working version is called "Default" version

Initial Setup

What you need ?

- AWS Account – Free Tier Usage - https://aws.amazon.com/free
- Matillion Hub Account - https://hub.matillion.com/
- Snowflake Free Tier - https://signup.snowflake.com/
- Microsoft Azure Free Account – https://portal.azure.com

………and follow the instruction in labs

Section 4 – Matillion Data Loader

What is Matillion Data Loader ?

Matillion Data Loader is a no-code data loading platform that helps data teams extract and load data faster and easier. It is a SaaS platform that specializes in data loading tasks, extracting data from data sources and loading that data into your preferred cloud data platform destinations.

Matillion Data Loader

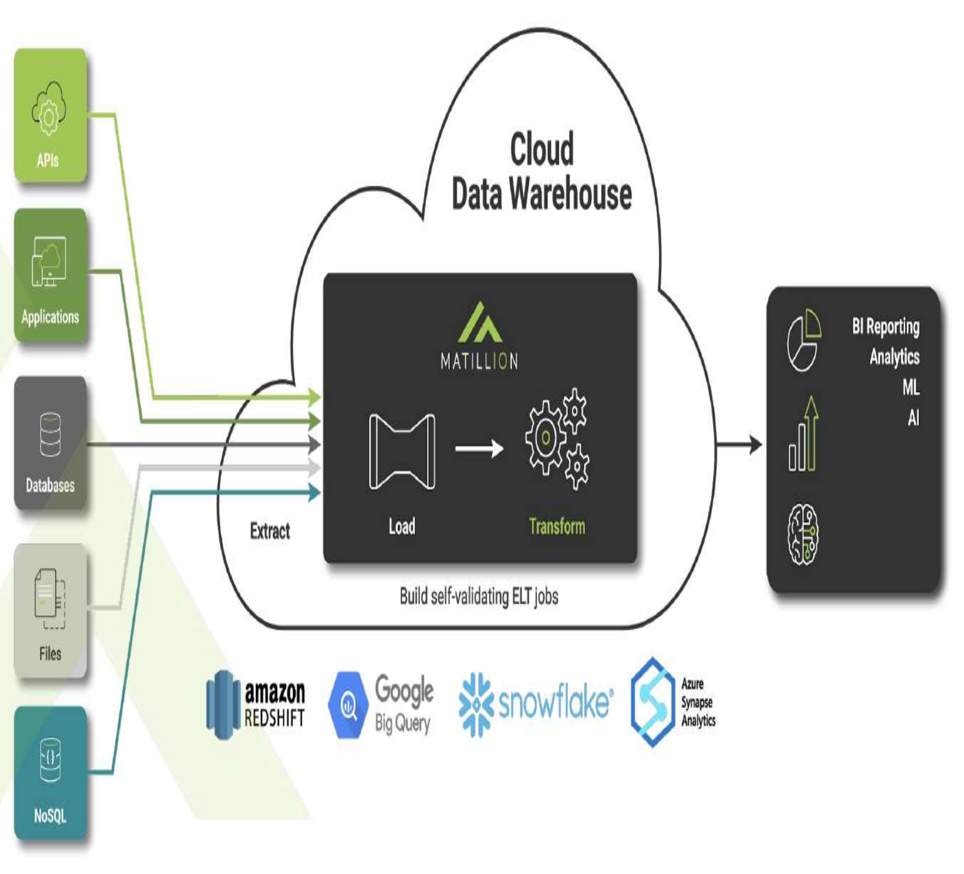
Here's a high-level overview of how it works:

Data Extraction: Matillion Data Loader connects to the source database and extracts the data.

Data Loading: It offers two methods for loading data:

- **Batch Loading**: A SaaS-based, incremental data loading experience. It enables users to extract data from popular sources and load that data into cloud data platform destinations at specific time intervals.
- **Change Data Capture (CDC) Loading**: Captures real-time change events from enabled source databases and loads the "change data" into cloud data platform destinations in near real time.

Integration with Matillion ETL: Data Loader seamlessly integrates with Matillion ETL, which is a comprehensive extract, transform, and load (ETL) tool.



Benefits of Matillion Data Loader (MDL)

No-code pipeline creation: It enables you to design data pipelines without the need for coding. Single platform for batch and change data capture pipelines: Users can manage both batch loading and change data capture (CDC) pipelines in a single platform.

Increase productivity: By extracting and loading data at speed and scale with Matillion Data Loader.

Accelerate time to insights: Through self-serve, no-code data loading.

Get data business ready faster: Through pre-built transformations.

Scale easily across the business: With predictable, usage-based pricing (and start out for free).

Simplify data architecture and security: With a single data loading platform and unique hybrid SaaS integration

Section 5 – Matillion UI

- Look and Feel
- Project Menu Options
- Admin Menu options
- Scope of options

Project Menu Options

- Look and Feel
- Project Menu Options
- Admin Menu options

Project Menu Options

- Manage Project
- Manage Credentials
- Manage Query Profiles
- Manage Extract Profiles
- Manage Oauth
- Manage Sequences
- Manage Versions
- Manage API Profiles Wizards
- Manage Error Reporting

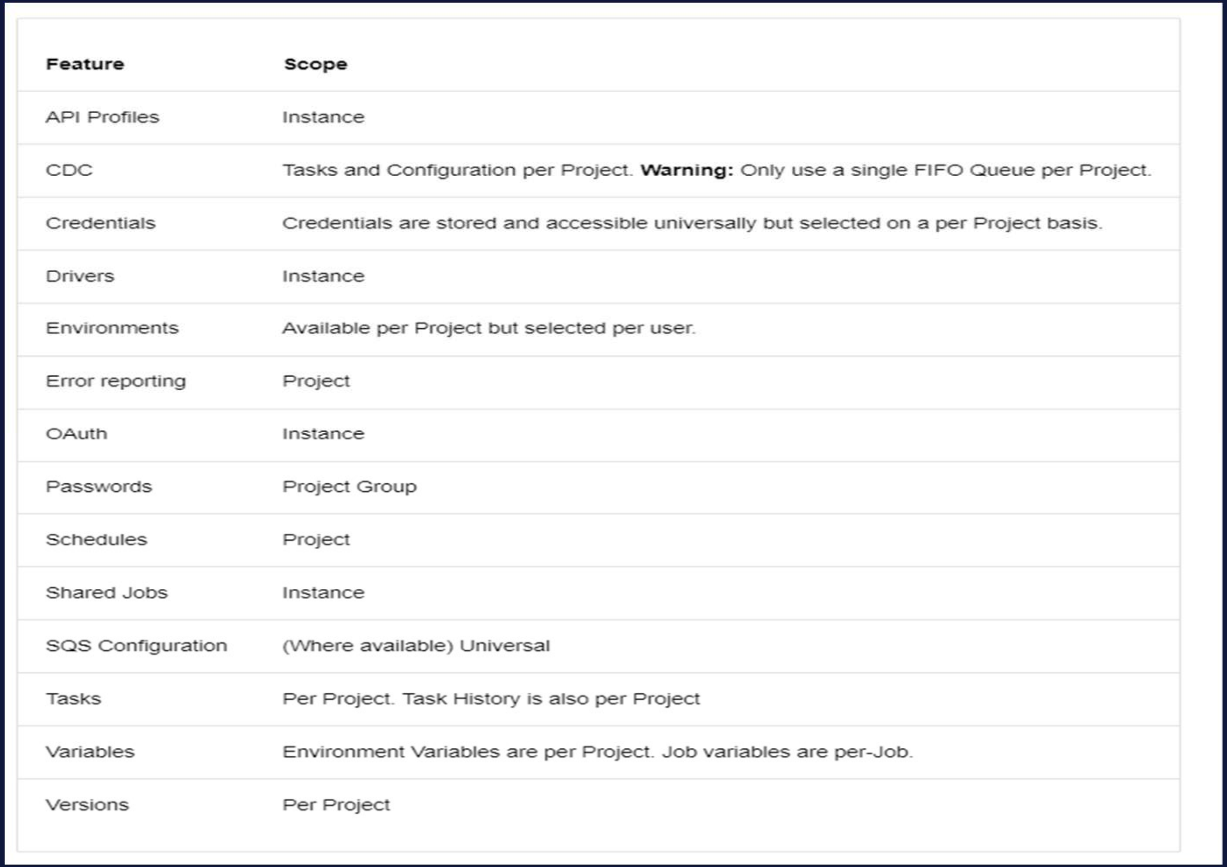
Admin Menu Options

- Manage Project
- Manage Credentials
- Manage Query Profiles
- Manage Extract Profiles
- Manage Oauth
- Manage Sequences
- Manage Versions
- Manage API Profiles Wizards
- Manage Error Reporting

Certification Centric Summary – Project Menu 10%

- There are no practical limits to the number of projects you can create
- Jobs, environments, and variables can be exported from one Matillion ETL instance to be imported into another using the Import/Export project menu
- API Query Profiles, Shared Jobs and OAuth are all set at the instance level – please check the scope of project menu options
- An API is described in a "Profile" definition, which is a collection of XML files describing the API and mapping it to tables, rows, and columns.
- Schedules are created against a project, and users can set up multiple schedules.
- Matillion export and import is done via json file exchange
- Project Group and Project Name are mandatory fields when creating a new project
- Job, Variable and Environment Variable can be exported from the project menu
- Matillion ETL project is a logical grouping of configuration settings and resources
- Disable All feature of Schedule job can be used to disable all the job at once

Certification Centric Summary – Project Menu Feature Scope



Feature	Scope
API Profiles	Instance
CDC	Tasks and Configuration per Project. **Warning:** Only use a single FIFO Queue per Project.
Credentials	Credentials are stored and accessible universally but selected on a per Project basis.
Drivers	Instance
Environments	Available per Project but selected per user.
Error reporting	Project
OAuth	Instance
Passwords	Project Group
Schedules	Project
Shared Jobs	Instance
SQS Configuration	(Where available) Universal
Tasks	Per Project. Task History is also per Project
Variables	Environment Variables are per Project. Job variables are per-Job.
Versions	Per Project

Section 6 – Variables

Variables are name-value pairs stored within each environment. Variables can be used in all sorts of parameters and expressions to allow the user to pass and centralize environment specific configuration. They're also easily accessible through Python and Bash scripts where they are substituted in at run time.

- Matillion ETL supports Environment Variables, Automatic Variables, Job Variables, and Grid Variables
- Variables can be referenced with the syntax: ${<variable name>}.
- When a job is run, variables are resolved by first consulting job variables, then environment variables, then system variables. Thus, if a job variable and environment variable of the same name exists, the job variable will be preferentially used

Types of Variables

- **Environment**: These variables are Global and can be used across multiple jobs of a project.
- **Job**: The scope of these variables is a job. Job variables will override any environment variables of the same name within that specific job. These variables can be used across multiple components in the Job
- **Grid**: Grid Variables are a special type of Job Variable that can be declared in the Matillion ETL. Grid Variables are 2D arrays that hold scalar values. Headers for columns of the grid can be defined within Matillion but are separate from the data held in that grid.
- **Automatic**: These variables are internally generated. These include environment variables like: project_group_name,project_name

Variables - Behavior

Variables may be referred to as Copied and Shared. This refers to their "branch behaviour" inside a job. A "branch" in this context means a divergence of connectors within a job, giving a branched structure. Iterator components are also examples of branching, albeit with a different aesthetic. The branch behaviour describes how a variable behaves when updated during a job run:

- Copied variables can be updated within one branch of a job without updating variables of the same name in other branches. A branch using a copied variable will begin by taking its default value, even if this variable has been updated in another, parallel branch.

- Conversely, Shared variables are updated in a job-wide fashion. As such, regardless of branches, if a Shared variable is updated in one branch, all other branches will use that updated value.

Note: In previous versions of Matillion ETL, Copied and Shared variables have been referred to as Local and Global, respectively, and can be thought of synonymously when reading the documentation.

Section 7 – Jobs

- Type of Jobs
- Job Component
- Practical usage of job components

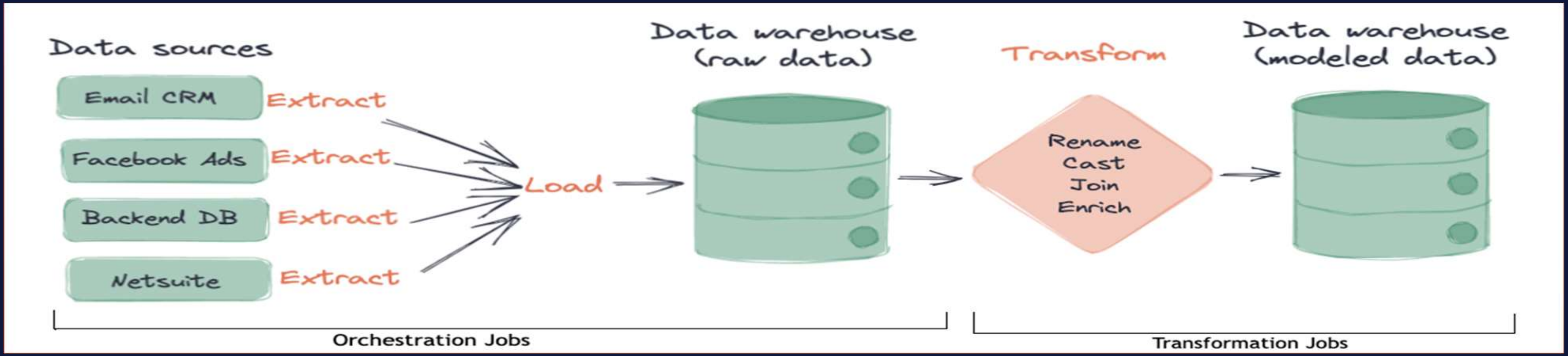
Jobs

Jobs are Matillion ETL's main way of designing, organizing, and executing workflows.

Types of Jobs:

Orchestration jobs are primarily concerned with DDL statements (especially creating, dropping, and altering resources), loading data from external sources.

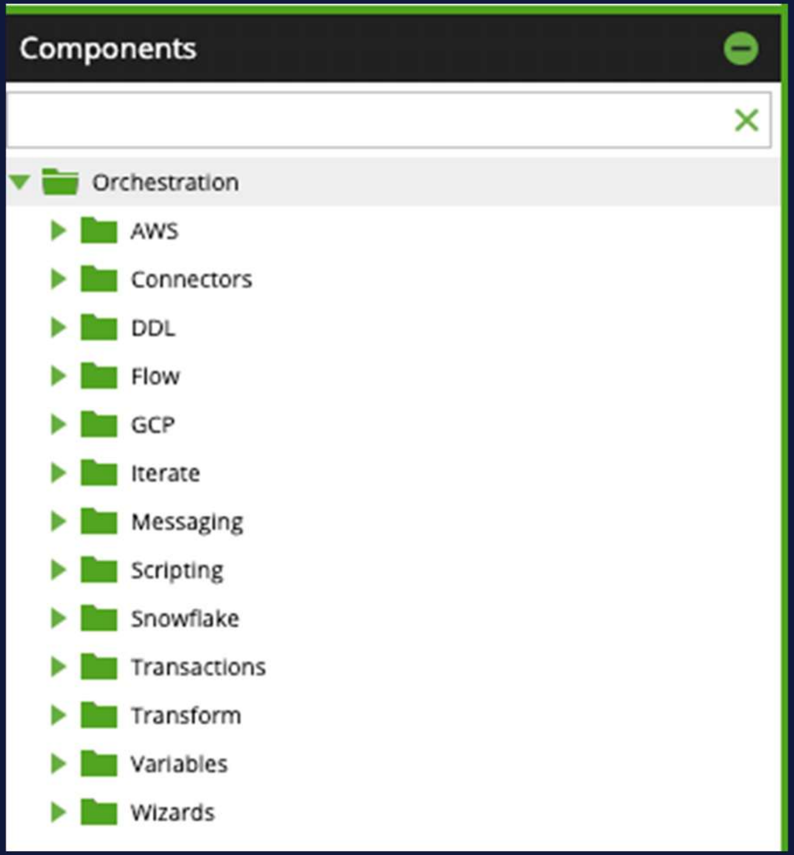
Transformation jobs are used for transforming data that already exists within tables. This includes filtering data, changing data types, and removing rows.



Orchestration job Component

High Level Categories

- Data Staging Components – Connectors, Data Transfers, Query
- DDL – Alter Table, Delete Tables, Refresh External Tables
- Flow – And, Or, End Failure, End Success, Retry
- Iterator – File, Fixed, Grid, Loop, Table
- Messaging – SQS/SNS Message, PubSub, Webhook Post
- Transaction – Begin, Rollback, Commit
- Scripting – Bash, Python Scripting, Run DBT, Run Notebook
- Variables – Append to Grid, Table Metadata to Grid
- Wizards – S3 Load Generator, Salesforce Incremental Load



Orchestration job Component

Type of Connector

- Query – API Query, Facebook Query, Salesforce Query
- Extract : API Extract, Instragram Extract, Recurly Extract
- Load: S3 Load, Azure Load, Dynamo Load
- Incremental Load: JDBC Incremental Load, Salesforce Incremental Load
- Output: Salesforce Output, Intercom Output, Oracle Output

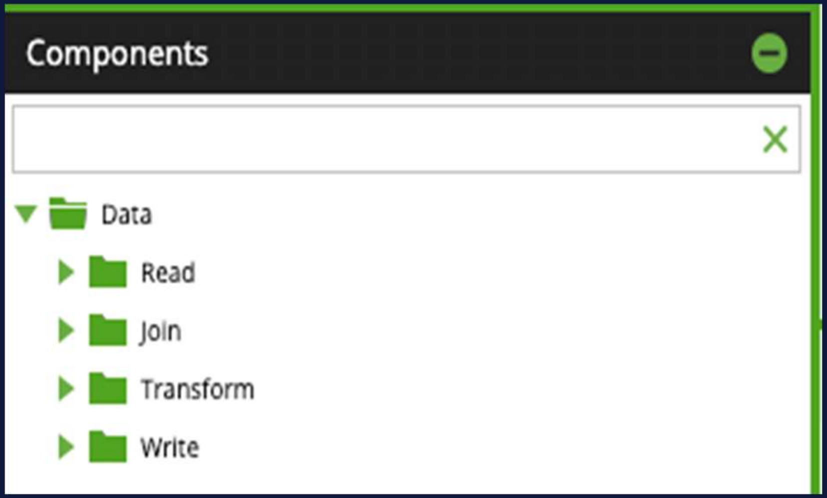
Note: Component properties differ between connectors. However, there are some common themes that should help the process feel familiar. Below are a few of the more common properties that illustrate the core usage of connector components.

Data Source,Data Selection,Data Source Filter,Limit,OAuth,(Staging) Location,Target Table

Transformation jobs Component

High Level Categories

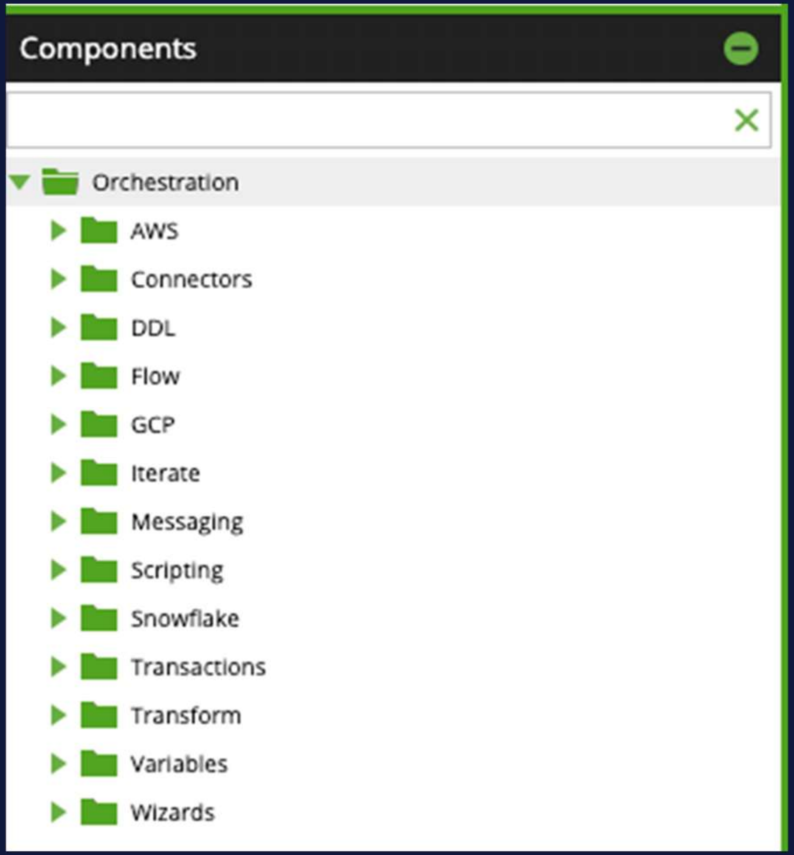
- READ – Fixed Flow, Generate Sequence, Table Input, Mutlitable Input
- JOIN – Except, Intersect, Join, Unite
- TRANSFORM – Aggregate, Calculator, Convert Type, Rank
- WRITE – Create View, Rewrite Table, Table Output, Table Update



Orchestration job Component – Deep Dive

High Level Categories

- Data Staging Components – Connectors, Data Transfers, Query
- DDL – Alter Table, Delete Tables, Refresh External Tables
- Flow – And, Or, End Failure, End Success, Retry
- Iterator – File, Fixed, Grid, Loop, Table
- Scripting – Bash, Python Scripting
- Variables – Append to Grid, Table Metadata to Grid
- Wizards – S3 Load Generator, Salesforce Incremental Load



Components

- Orchestration
  - AWS
  - Connectors
  - DDL
  - Flow
  - GCP
  - Iterate
  - Messaging
  - Scripting
  - Snowflake
  - Transactions
  - Transform
  - Variables
  - Wizards

Orchestration Job – Data Staging Components Practicals

For all the loading scenarios – our choice of destination is Snowflake

- Load Data from S3 Bucket
- Load Data from Excel File
- Load Data from Database Query
- Load Data from API
- Load Data from Gmail
- Data Transfer

Certification Centric Summary – Orchestration & Transformation Job - 75%

- Variable can be referred in component using the syntax ${<VariableName>}
- Environment variables can be shared between multiple jobs
- Matillion ETL Versioning creates a copy of all the job at that point of time
- Fixed Iterator can iterate over fixed values and loop iterator over simple sequence
- Calculated or derived columns can be added by using calculator component
- SQL Script component allows you to write your own SQL code in an Orchestration job
- Data Staging components can retrieve data from other services and load that data into a table
- Data Staging components can also be interchangeably called as data stagers, connectors, query components or integrations
- All jobs start in 'Auto Commit' mode. All work is committed immediately as the job runs. This default behaviour can be changed using the Begin, Commit, and Rollback transaction components
- Start component comes as a default when creating orchestration job and there needs to be atleast one per orchestration job
- Row count and few other attributes can be captured using export tab on the particular component
- Set operation can be carried out using transformation job component – unite, intersect, join components can be used for the same
- Rollback component only affects DML statement and not DDL statements
- Convert Type component is used to cast data types
- Detect Change components indicator type C=Changed D=Deleted I=Identical N=New
- SQL Script component – cannot have select statement in orchestration job
- To see value of variable it can be printed via print command in python script component
- IF component has three connectors - Blue ="true" Orange="false" Red="Error"

Section 8 – What are Slowly Changing Dimensions ?

A Slowly Changing Dimension (SCD) is a dimension that stores and manages both current and historical data over time in a data warehouse12. It is considered and implemented as one of the most critical ETL tasks in tracking the history of dimension records

Slowly Changing Dimensions (SCDs) are classified into several types, each with a different way of handling changes in the data. Here are the common types of SCDs with examples:

1. **Type 0** - Fixed Dimension: No changes allowed, dimension never changes. For example, an employee's date of joining a company would not change and would be retained as is.

2. **Type 1** - No History: Update record directly, there is no record of historical values, only current state. For instance, if a customer's address changes, the new address overwrites the old one.

3. **Type 2** - Row Versioning: A new record is added to the table whenever a change occurs, allowing full history to be maintained. For example, if a product's price changes, the old record is retained, and a new record is created with the new price.

4. **Type 3** - Add New Attribute: A new attribute is added to the table to track changes. For instance, if an employee's designation changes, a new column could be added to track the previous designation

5. **Type 4** - Add History Table: A separate history table is maintained for tracking changes.

6. **Type 6** - Combined Approach: This is a combination of Type 2 (for tracking historical changes) and Type 3 (for tracking the previous value of a changed attribute).

Section 8 - Dimension SCD Type 2 in Matillion

- **Type 2 dimensions are always created as a new record. If a detail in the data changes, a new row will be added to the table with a new primary key. However, the natural key would remain the same in order to map a record change to one another. Type 2 dimensions are the most common approach to tracking historical records.**

Example:

ADDR_KEY	CUSTOMERID	ADDRESSID	ADDRESSTYPE	ACTIVE_FLAG	EFF_START_DT DATE	EFF_END_DT
1	C1	A1	Main Office	N	01/01/2023	14/01/2024
2	C1	A1	Shipping	Y	15/01/2024	

Section 9 – Data Modelling Process

- **Model a Business Process**: Be clear about which business process is being modeled. There are many different types of Fact tables, each with its own purpose: Transactional Fact tables, Factless Fact tables, and Derived tables.

- **Set up Transactional Fact Tables**: Transactional Fact tables store data at the most granular level available for the process being modeled. It is important to store single values at the lowest possible grain.

- **Consider Factless Fact Tables**: For business processes that focus on events, there may be no measures beyond the information provided by the Dimensions.

- **Create Derived Tables**: Derived tables are useful for storing aggregated or calculated data.

- **Error Handling**: Make your jobs more resilient to errors. Implement a retry pattern for transient faults and make jobs idempotent.

- **Logging Errors**: Matillion logs any errors encountered by the application in the Server Logs. Use the Auto Debug feature on the Data Stager components with a verbosity setting that allows you to capture much more information to help with troubleshooting.

Project : Buidling a Data Warehouse

- Follow the labs

Section 10 - What is Amazon Redshift ?

It is a fully managed, petabyte-scale data warehouse service in the cloud provided by Amazon Web Services (AWS). It allows you to access and analyze data without all of the configurations of a provisioned data warehouse. You can load data and start querying right away in the query editor or in your favorite Business Intelligence (BI) tool.

Architecture: https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html

Performance: https://docs.aws.amazon.com/redshift/latest/dg/c_challenges_achieving_high_performance_queries.html

Columnar : https://docs.aws.amazon.com/redshift/latest/dg/c_columnar_storage_disk_mem_mgmnt.html

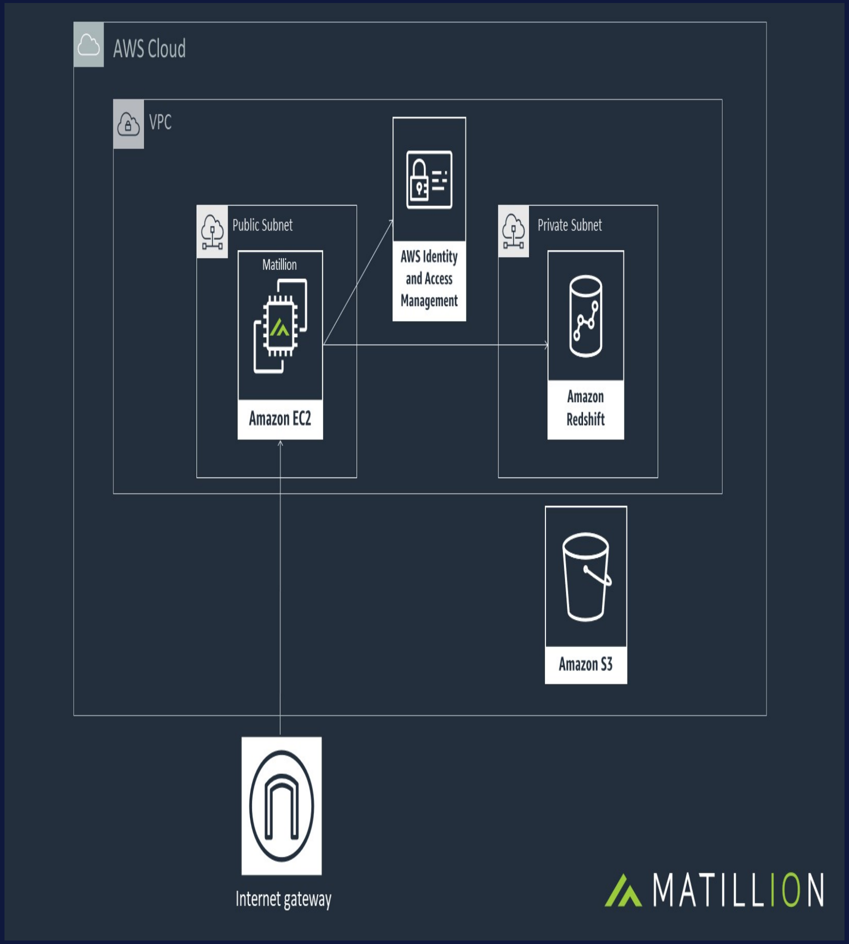
Amazon Redshift and Matillion

- Recommendation is that Matillion is launched in the same region as Amazon Redshift, in either the same VPC or in a peered VPC. Matillion can either be launched as an Amazon Machine Image (AMI), which you can fit into your existing architecture or as a Cloud Formation template which an spin up the required resources in conjunction with spinning up Matillion from the AMI.

- An S3 Bucket will be required in the same region as Amazon Redshift. Matillion uses S3 to stage your business data to before it is copied into Amazon Redshift. You can also use S3 as a data lake if you wish to have a copy of your data outside of Amazon Redshift.

- Matillion's access to AWS resources is controlled by the IAM role which the Matillion instance is running as. This can give Matillion access to a number of AWS services that will streamline your data processes, such as, SNS, SQS, Cloudwatch, RDS, S3 and Amazon Redshift.

- A security group for the Matillion instance controls who has access to that instance on which ports. We recommend only opening ports 80, 443 and 22 and to restrict these to known IP addresses.



Section 11 - Best Practises Matillion

- **"ETL" or "E-L-T"** : ETL has performance and I/O bottleneck. Hence, if the transformation happens in the cloud then massive performance gain can be achieved
- **SSH Matillion instance**: The instance should be accessible via ssh is the first check we need to perform after launch
- **Regular EBS Backup**: Backup of the Matillion instance needs to be done on regular basis
- **Python Component**: This component should not be used for transformation and neither to be used for bulk loading
- **Data Model**: Prior to starting the transformation – data model using standard approach should be in place
- **Staging**: It's a good practise to create temporary stage data prior to loading into target table. The temporary data retention is recommended for troubleshooting
- **Order or write operation**: The order of write operation should be maintained in the orchestration job
- **Handwritten DML**: Writing DML operation by hand is not a good practise. Instead appropriate job component usage is advisable

Section 11 - Best Practises Matillion

There's no official naming convention set by Matillion

General principles:

- **Clarity**: Names should be descriptive and easily understandable by anyone familiar with the project.
- **Conciseness**: Avoid overly long names, but ensure sufficient detail to differentiate objects.
- **Consistency**: Use the same naming scheme throughout your project for object types (jobs, variables, mappings, etc.).
- **Avoid special characters**: Stick to alphanumeric characters, underscores, and hyphens to ensure compatibility with different tools and platforms.

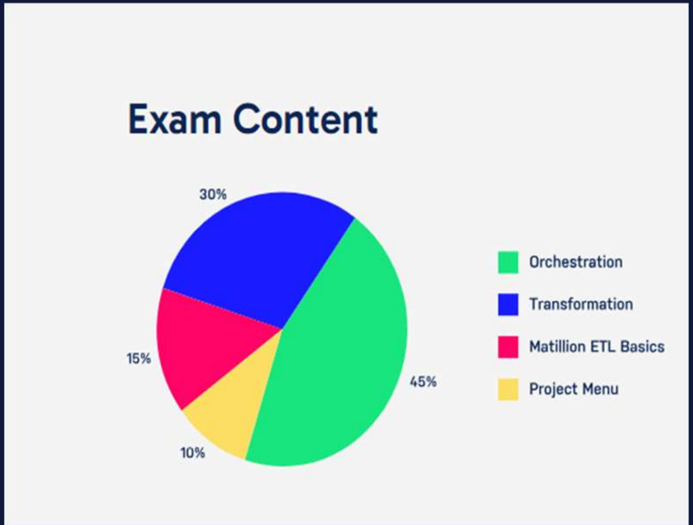
Jobs:

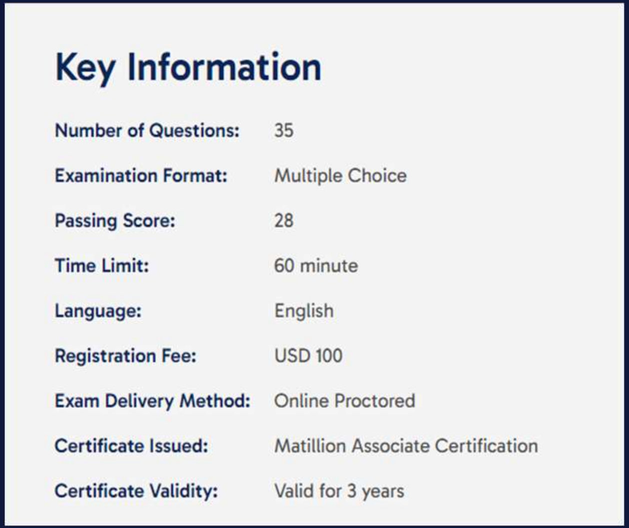
- Start with a prefix denoting the purpose or data source (e.g., ODS_Load, CRM_ETL, STG_, DIM_, FACT_).
- Include the specific action (e.g., _FullLoad, _Transform)

Variables:

- Prefix with the type of variable e.g. jv_ ~ Job Variable, ev ~ Environment Variable, gv ~ Grid Variable and all followed by underscore (_) & variable Name in Camel Case
- Provide default value per environment

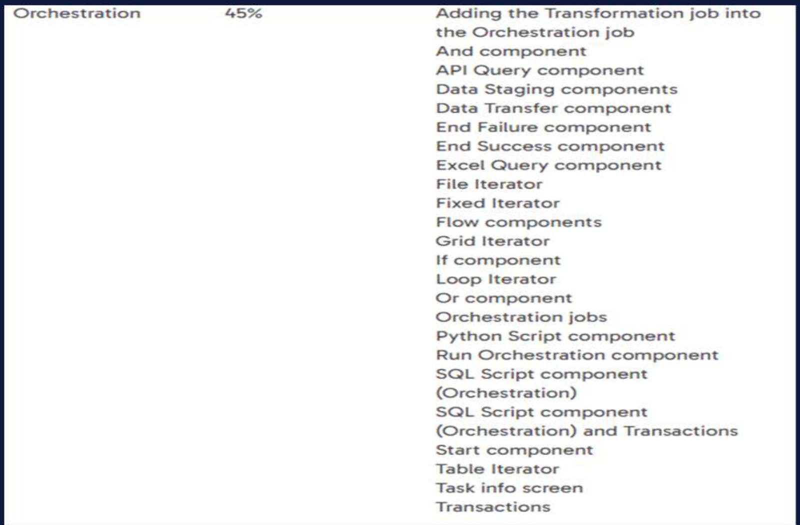
Section 12: Matillion Associate Certification



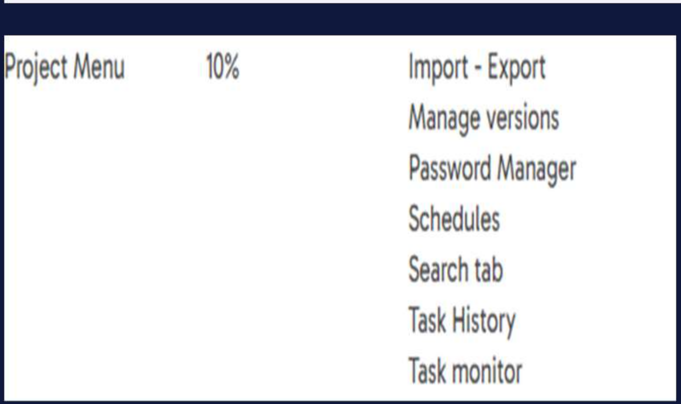


Key Information

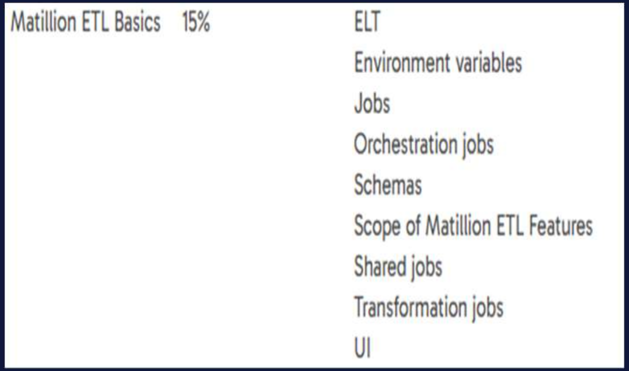
Number of Questions:	35
Examination Format:	Multiple Choice
Passing Score:	28
Time Limit:	60 minute
Language:	English
Registration Fee:	USD 100
Exam Delivery Method:	Online Proctored
Certificate Issued:	Matillion Associate Certification
Certificate Validity:	Valid for 3 years



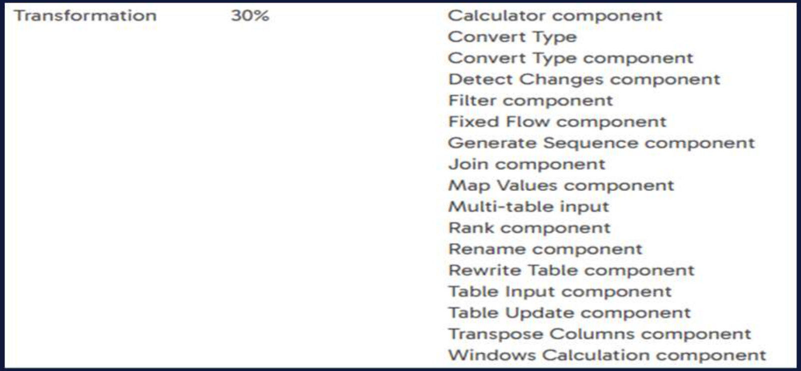
Orchestration	45%	Adding the Transformation job into the Orchestration job
		And component
		API Query component
		Data Staging components
		Data Transfer component
		End Failure component
		End Success component
		Excel Query component
		File Iterator
		Fixed Iterator
		Flow components
		Grid Iterator
		If component
		Loop Iterator
		Or component
		Orchestration jobs
		Python Script component
		Run Orchestration component
		SQL Script component (Orchestration)
		SQL Script component (Orchestration) and Transactions
		Start component
		Table Iterator
		Task info screen
		Transactions



Project Menu	10%	Import - Export
		Manage versions
		Password Manager
		Schedules
		Search tab
		Task History
		Task monitor



Matillion ETL Basics	15%	ELT
		Environment variables
		Jobs
		Orchestration jobs
		Schemas
		Scope of Matillion ETL Features
		Shared jobs
		Transformation jobs
		UI



Transformation	30%	Calculator component
		Convert Type
		Convert Type component
		Detect Changes component
		Filter component
		Fixed Flow component
		Generate Sequence component
		Join component
		Map Values component
		Multi-table input
		Rank component
		Rename component
		Rewrite Table component
		Table Input component
		Table Update component
		Transpose Columns component
		Windows Calculation component