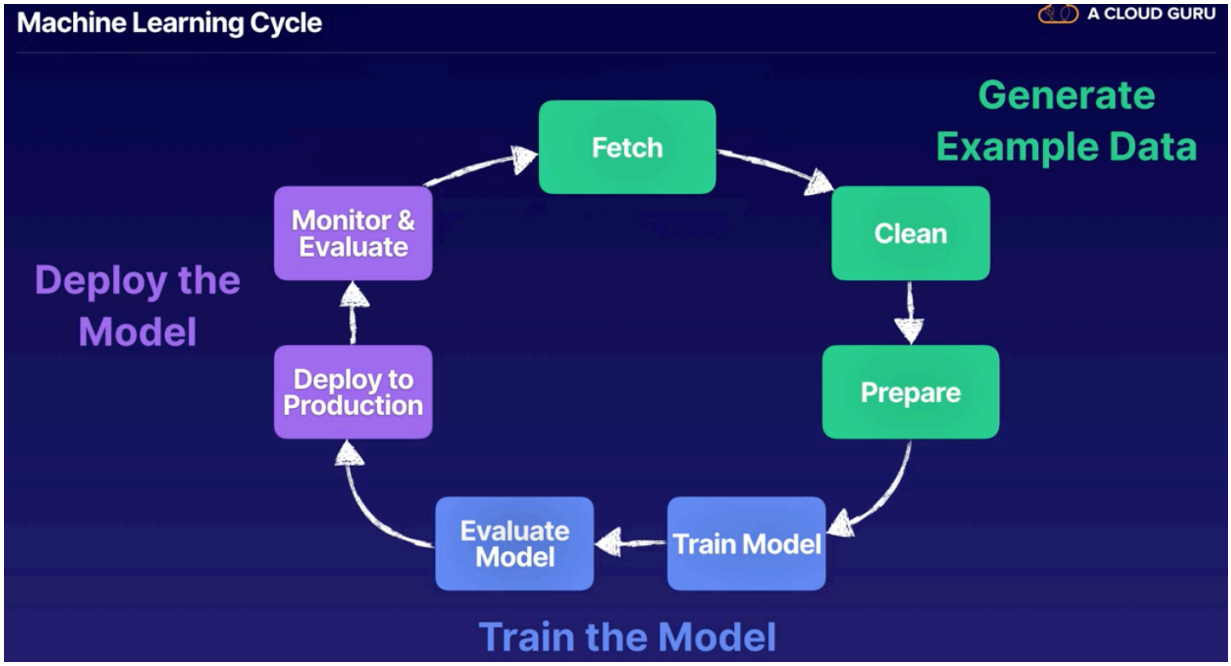
CloudGuru - 1 - Data collection

https://acloud.guru/course/aws-certified-machine-learning-specialty/learn/bc7319b8-616e-f972-ec23-d2b49728a81d/c068feca-3ae9-ce9b-0b5a-a7eca7f6f895/watch



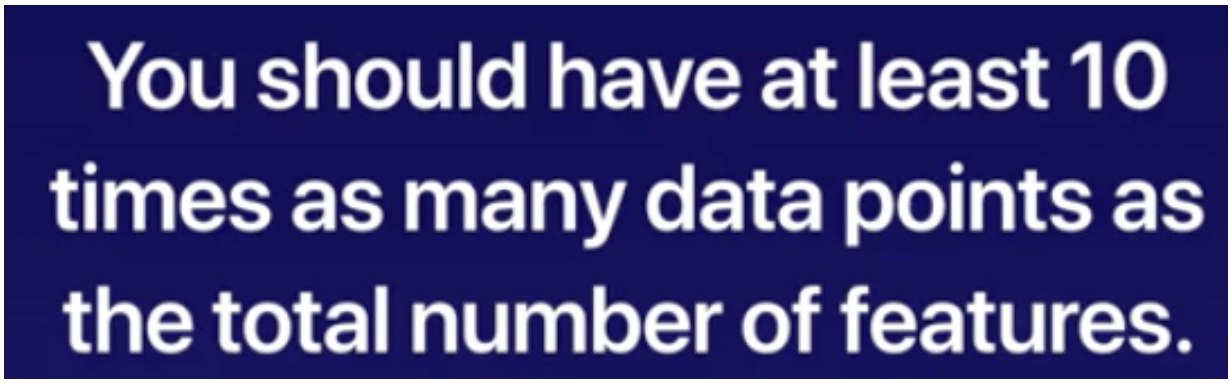
Data mostly comes unstructured, not prepared,...

A model would not infer anything good from noisy/crapy data

Good Data

Traits of good data	Traits of bad data	Why
Large datasets.	Small datasets (less than 100 rows).	Generally, more data means better model training.
Precise attribute types, feature rich.	Useless attributes, not needed for solving problem at hand.	Models need to train on important features.
Complete fields, no missing values.	Missing values, null fields.	Models can skew results when data points are missing.
Values are consistent.	Inconsistent values.	Models like clean and consistent data.
Solid distribution of outcomes.	Lots of positive outcome, few negative outcomes.	Models cannot learn with skewed distributions of outcomes.
Fair sampling	Biased sampling	Models will skew results with biased data.

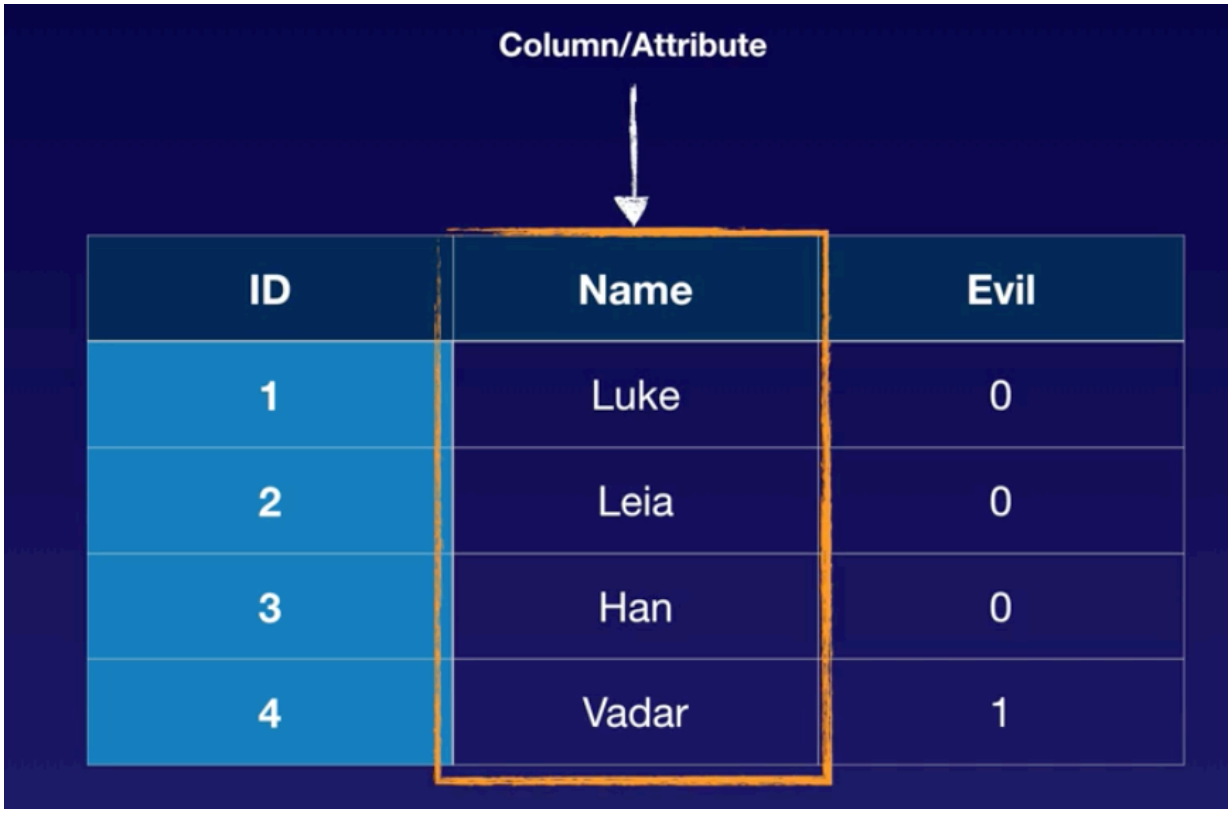
How much data do we need?



DATA Terminology

dataset = input data = training/testing data

column = attribute = feature

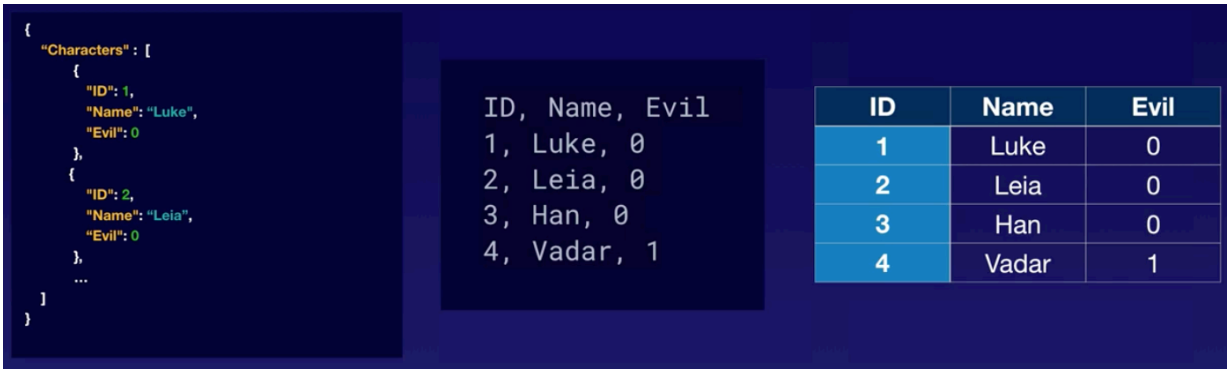


row = observation = sample = data point

ID	Name	Evil
1	Luke	0
2	Leia	0
3	Han	0
4	Vadar	1

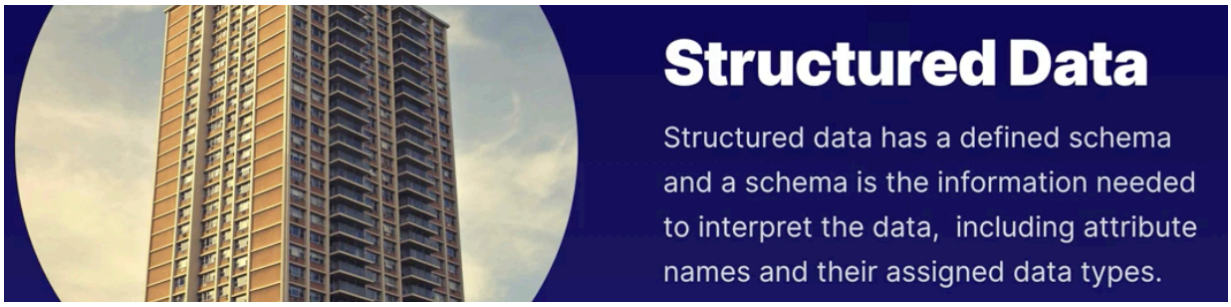
Row/Observation →

Other formats: JSON, CSV,...



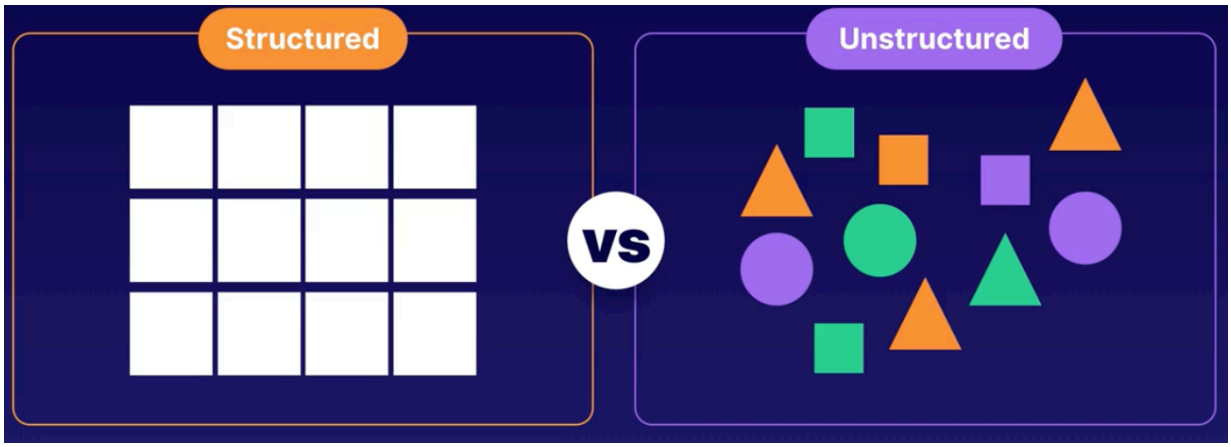
Also images, video, audio,... not just text

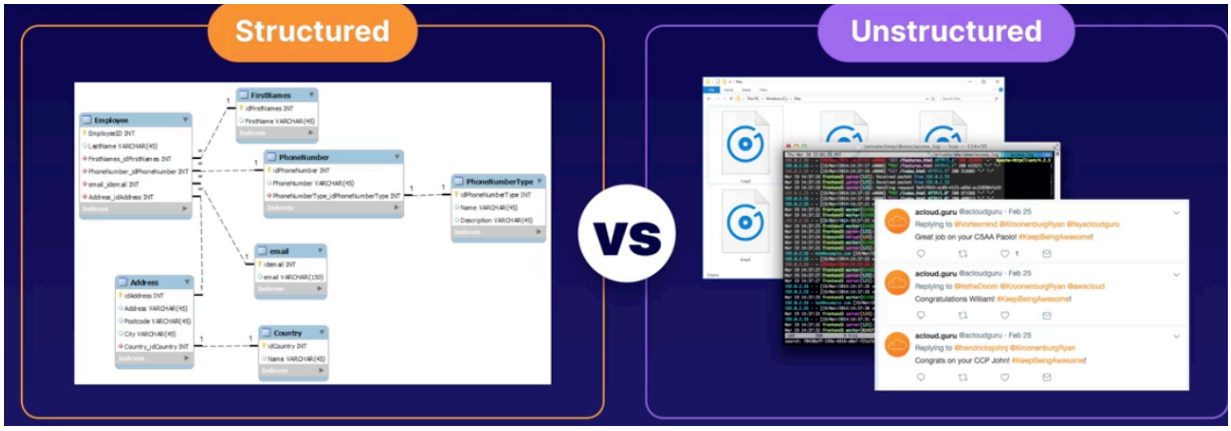
Structured data: SQL,...
It's how we would love our data



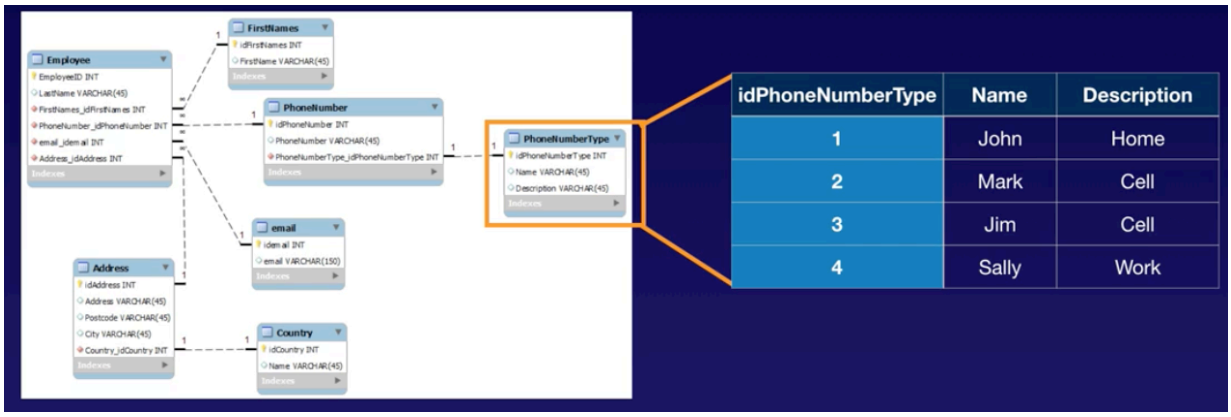
But majority of data is unstructured, with no defined schema
=> PDF, images, video, audio,.. log files

Some other data is semi-structured
NoSQL, JSON, XML, JSON



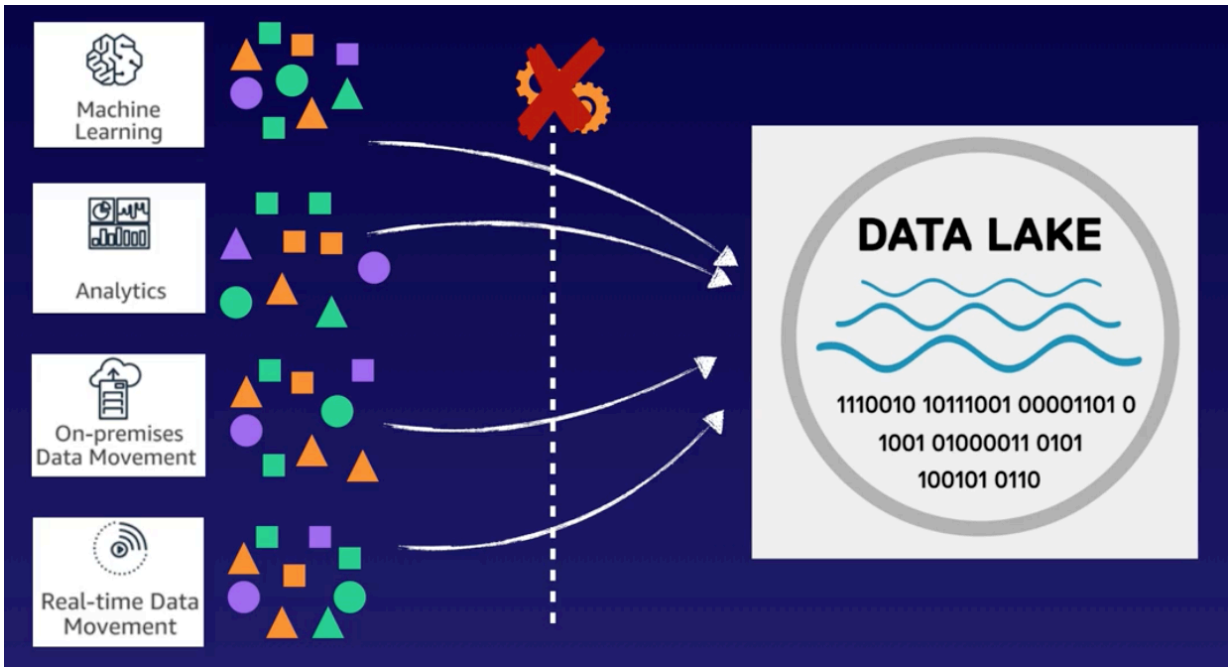


Where is data stored:
 - structured => transactional DB when’re changes re rolled back if a transaction is not successful



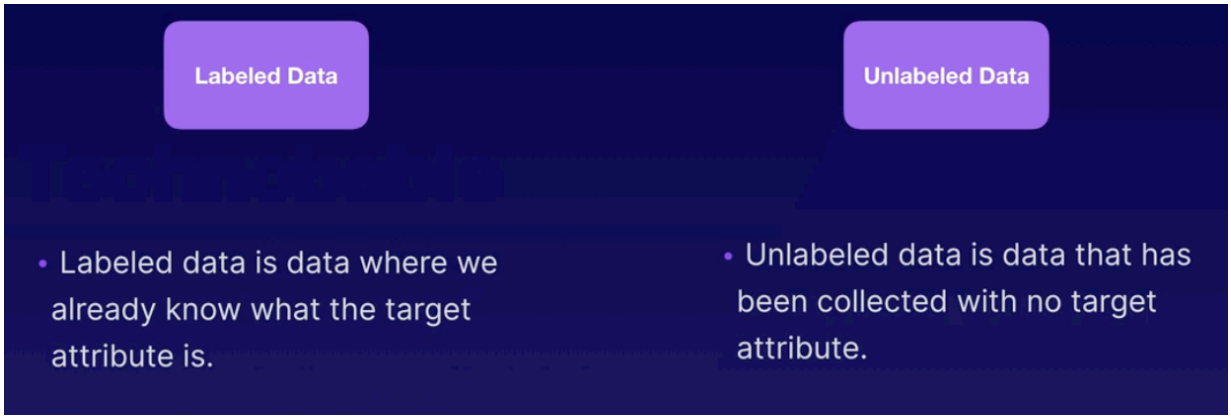
Other data store: **Data Warehouse**
Collects from many sources and congrats them together, and make them avail to BI tools
=> PROCESSING done before storing
Can store petabytes/heabytes of data and allow us to analyze more easily

Lastly, **Data Lakes** store VAST amount of data
There is **NOT processing** done. One stop shop for dumping our data (for historical data or data we are not sure what to do with but that is important)



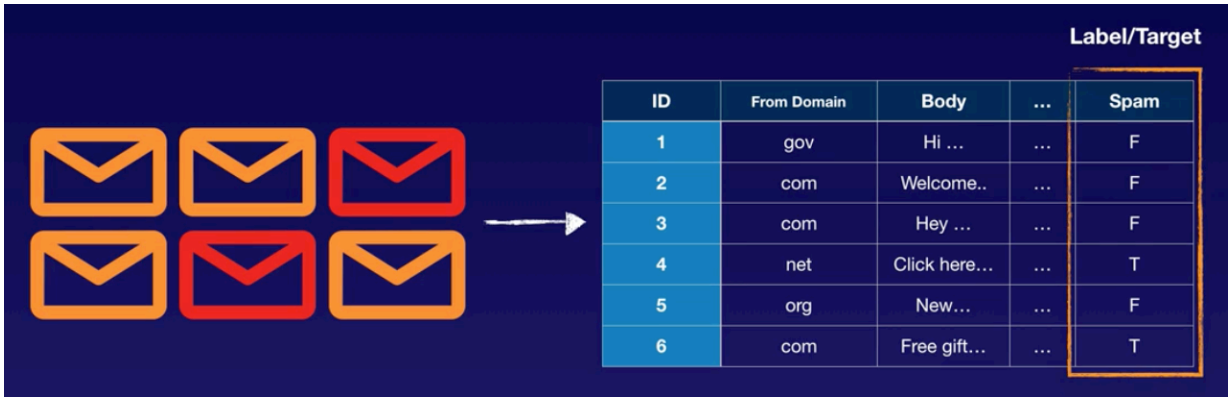
Databases	• Traditional relational databases • Transactional • Strict defined schema
Data Warehouses	• Processing done on import (schema-on-write) • Data is classified/stored with user in mind • Ready to use with BI tools (query and analysis)
Data Lakes	• Processing done on export (schema-on-read) • Many different sources and formats • Raw data may not be ready for use

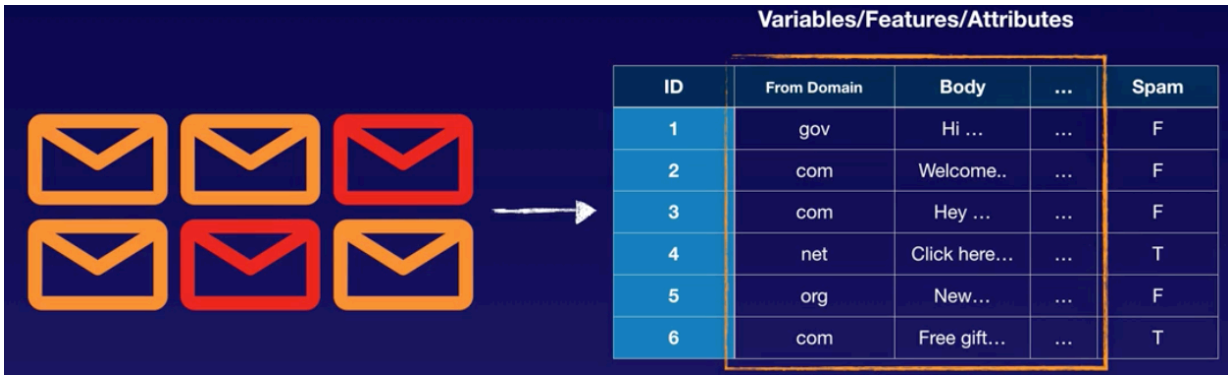
Labeled/Unlabeled data



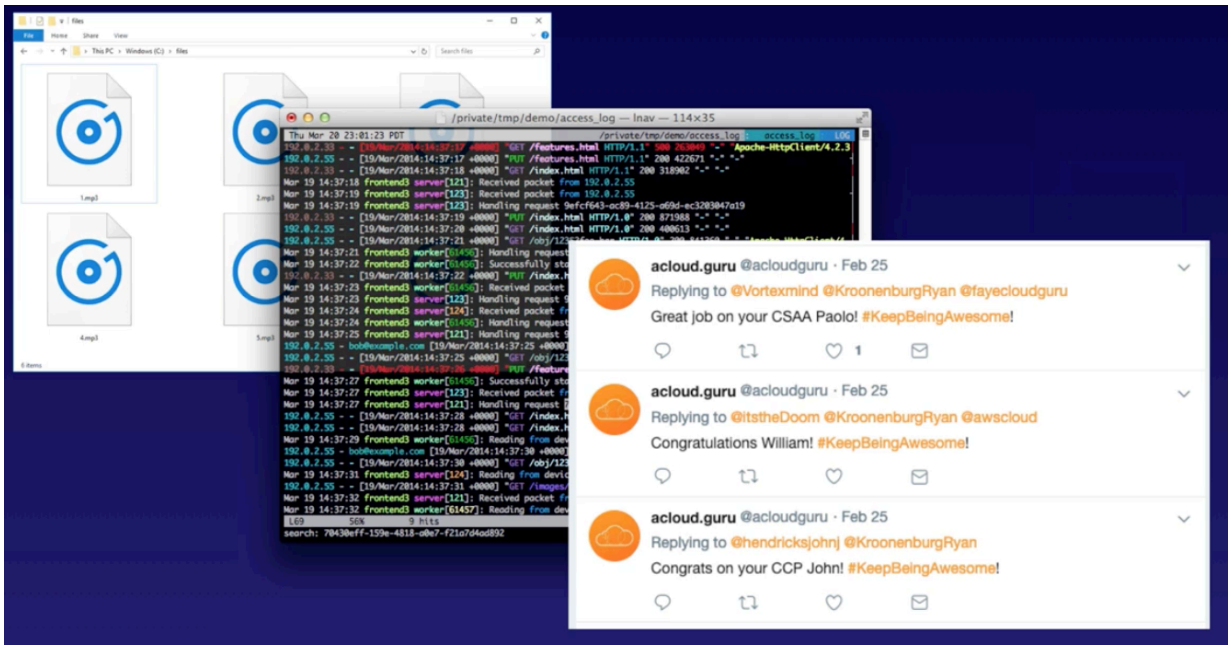
Example of labelled day on email with what is spam or not

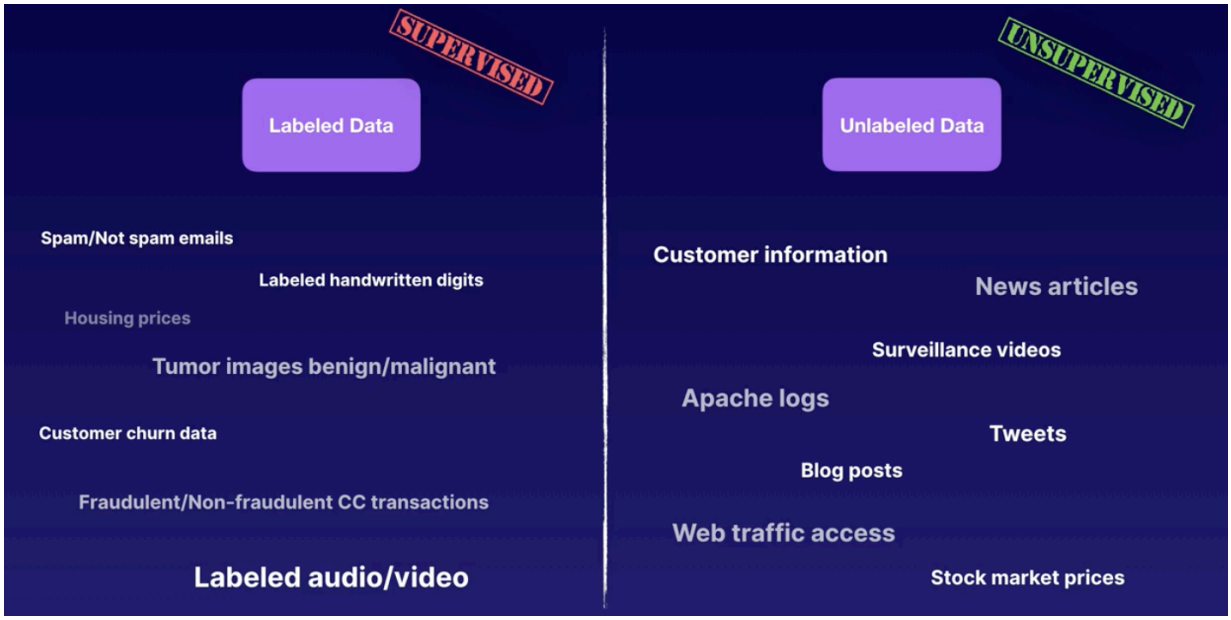
Spam is the label (or target attribute)



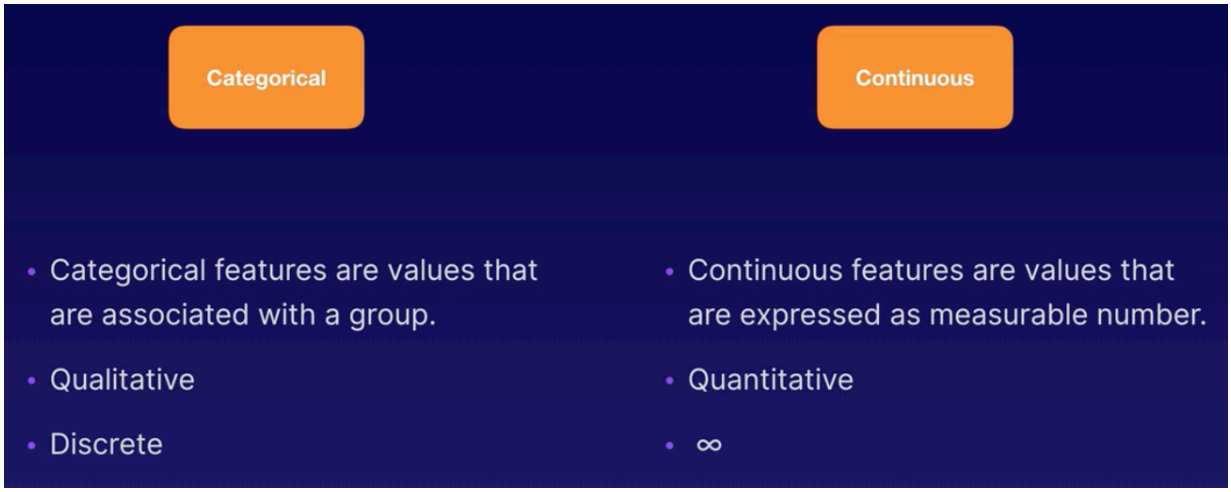


Unlabelled data: social media streams, log files, video, audio,...
There is no answer or tag associated to them





Categorical vs Continuous



Top to remember:

Group/Category

ID	Height	Weight	...	Breed
1	26	6	...	Cairn Terrier
2	82	45	...	Airedale Terrier
3	38	9	...	Fox Terrier
4	40	20	...	Bull Terrier
5	36	6.5	...	Fox Terrier
6	55	20	...	Airedale Terrier
7	55	30	...	Bull Terrier
8	31	7.5	...	Cairn Terrier

Group/Category

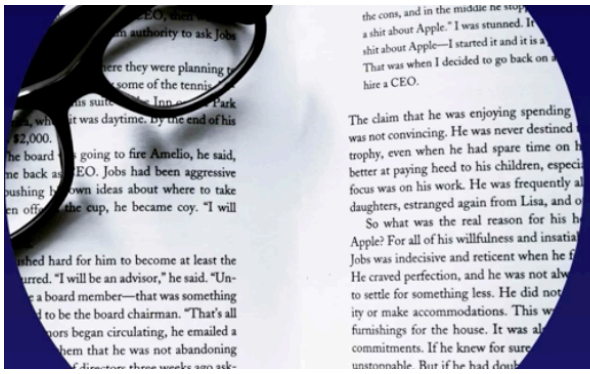
ID	From Domain	Body	...	Spam
1	gov	Hi ...	...	F
2	com	Welcome..	...	F
3	com	Hey ...	...	F
4	net	Click here...	...	T
5	org	New...	...	F
6	com	Free gift...	...	T

If the attribute falls into a group or category, the attribute is categorical.

Continuous

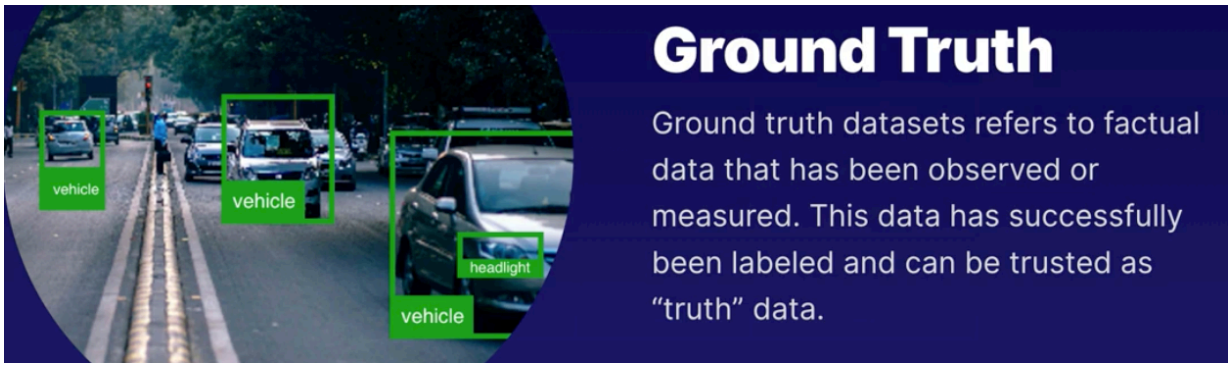
ID	Bedrooms	Square Feet	...	Price
1	2	1250	...	$125,000
2	3	1600	...	$165,000
3	1	975	...	$98,000
4	2	1400	...	$145,000
5	4	2500	...	$200,000
6	1	1075	...	$105,000

If you can place the attribute value on a number line, the attribute is continuous.

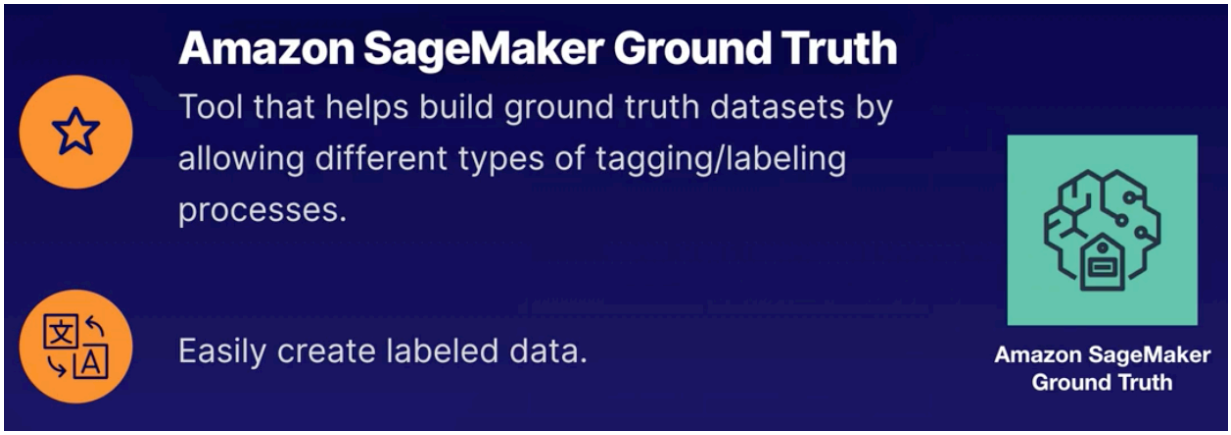


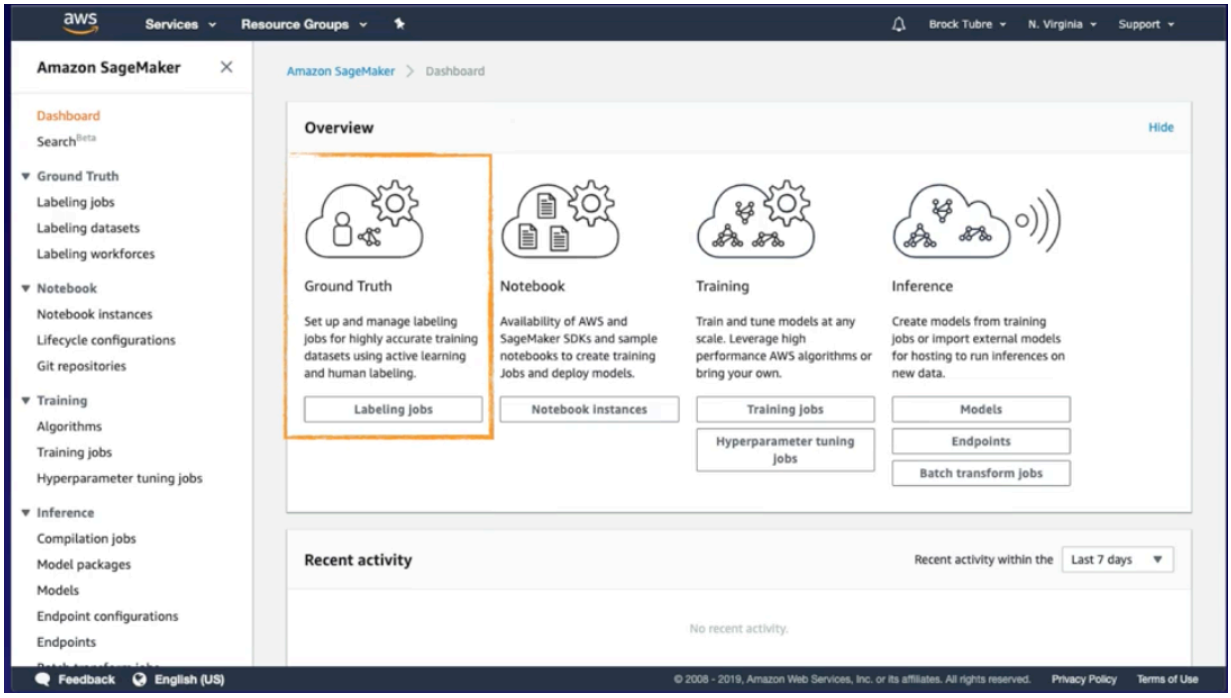
Text Data (Corpus Data)

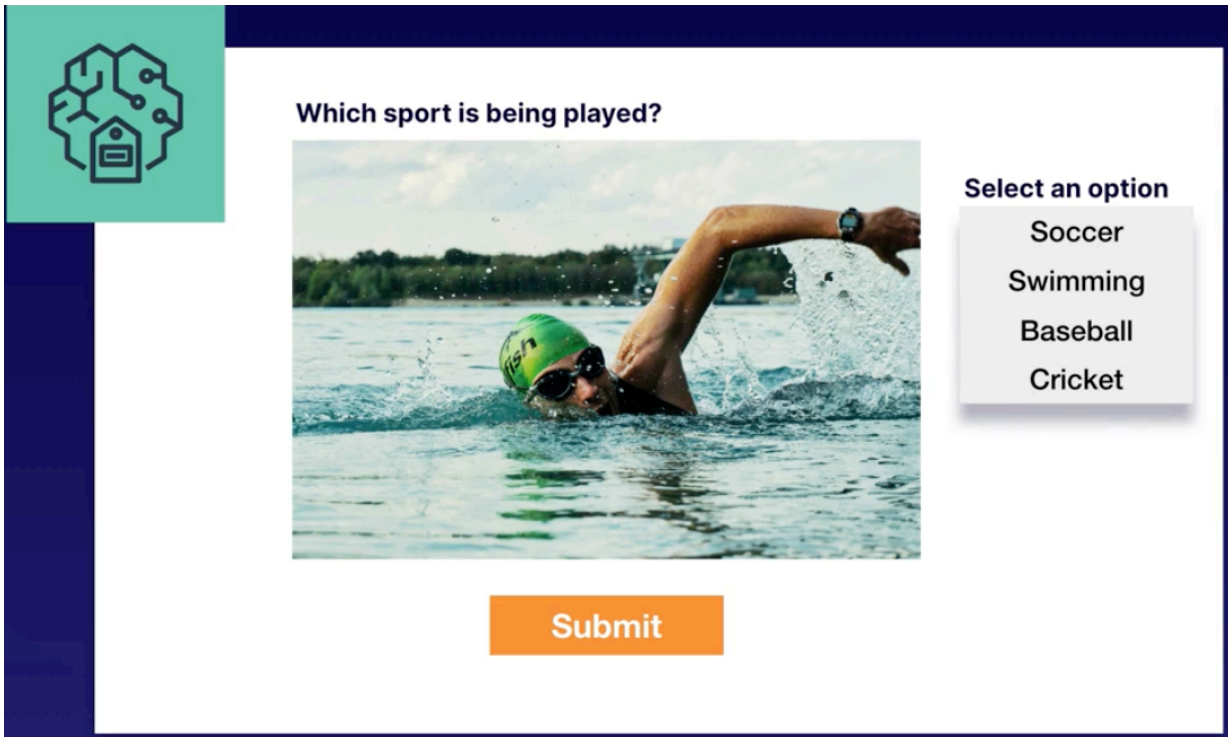
These are datasets collected from text. Used in Natural Language Processing (NLP), speech recognition, text to speech, and more.



SameMaker Ground Truth - tool to enable label data







Example of Tagged image data

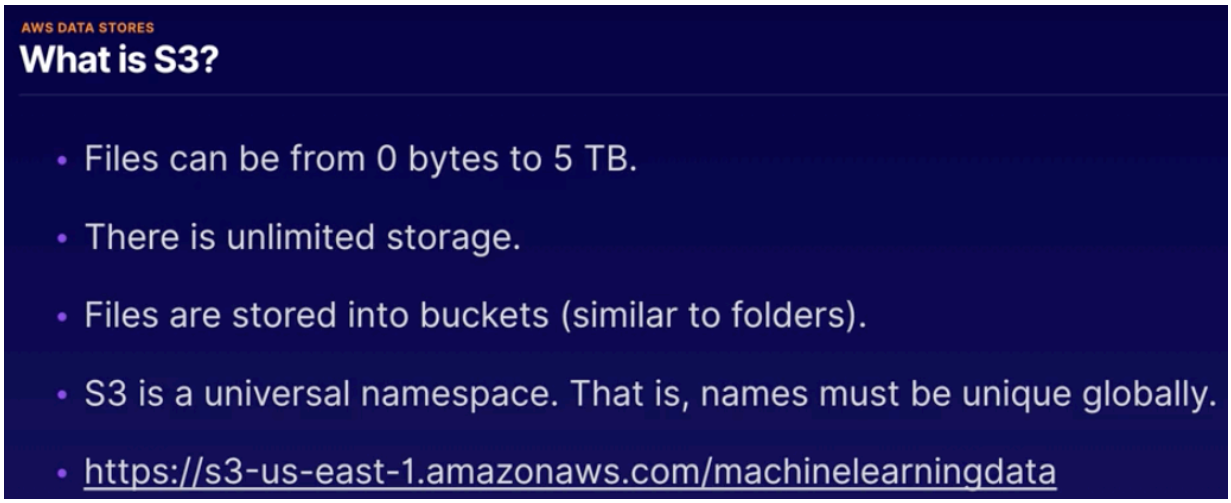
Dataset examples

Dataset Type	Example cases	Format
Image Data	Facial recognition, action recognition, object detection, handwriting and character recognition.	images, videos
Text Data	Reviews, news articles, messages, twitter and tweets, dialogs.	text, csv
Sound Data	Speech, music, other sounds.	mp3, text
Signal Data	Electrical signals, motion-tracking, chemical compounds.	text

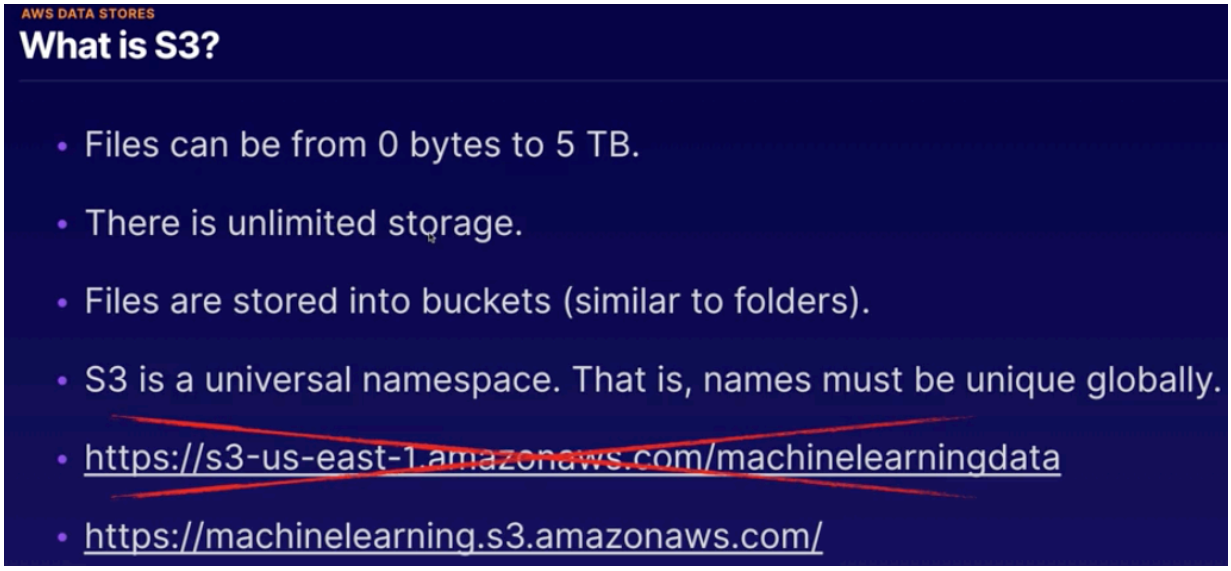
Dataset Type	Example cases	Format
Physical Data	high-energy physics, systems, astronomy, earth science.	text
Biological Data	human, animal, plants, microbes.	text
Multi-variable Data	financial, weather, census, transit, internet, games.	csv, text

AWS Data Stores and services to store data

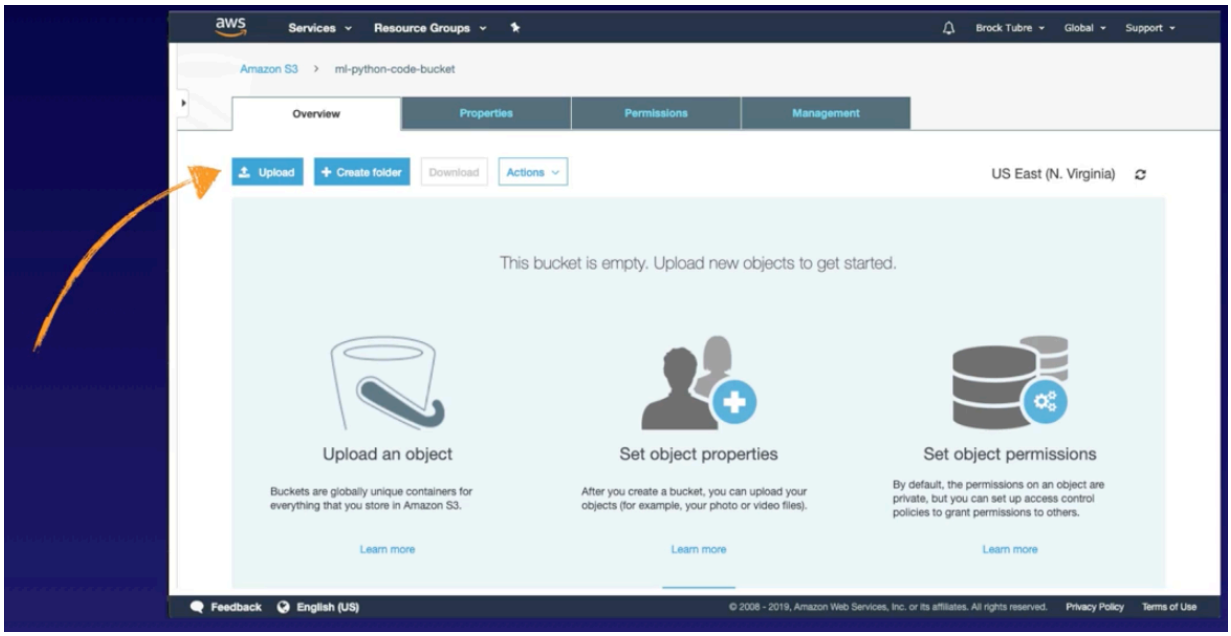
S3



Sept 2020: virtual hosted style coming

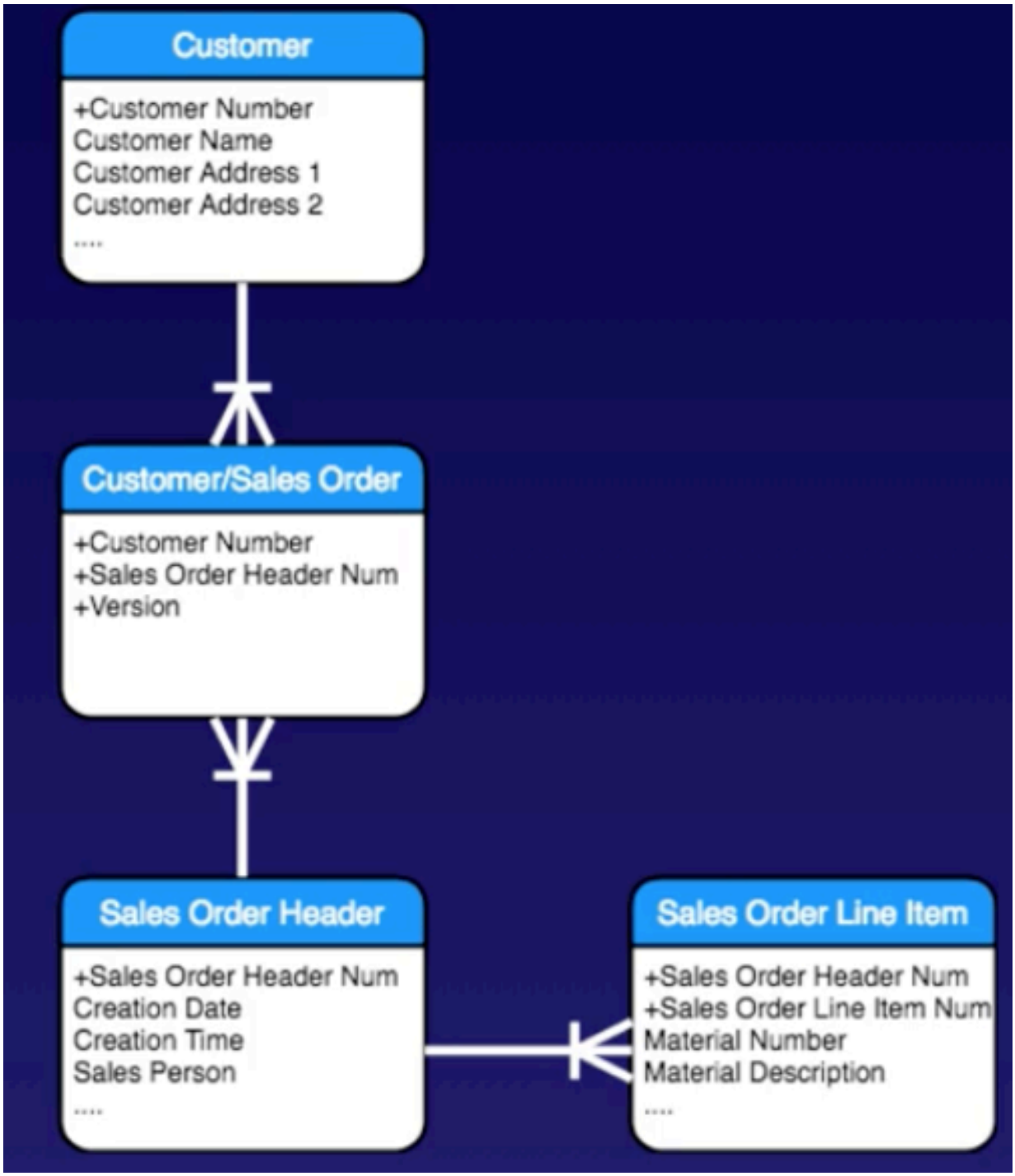


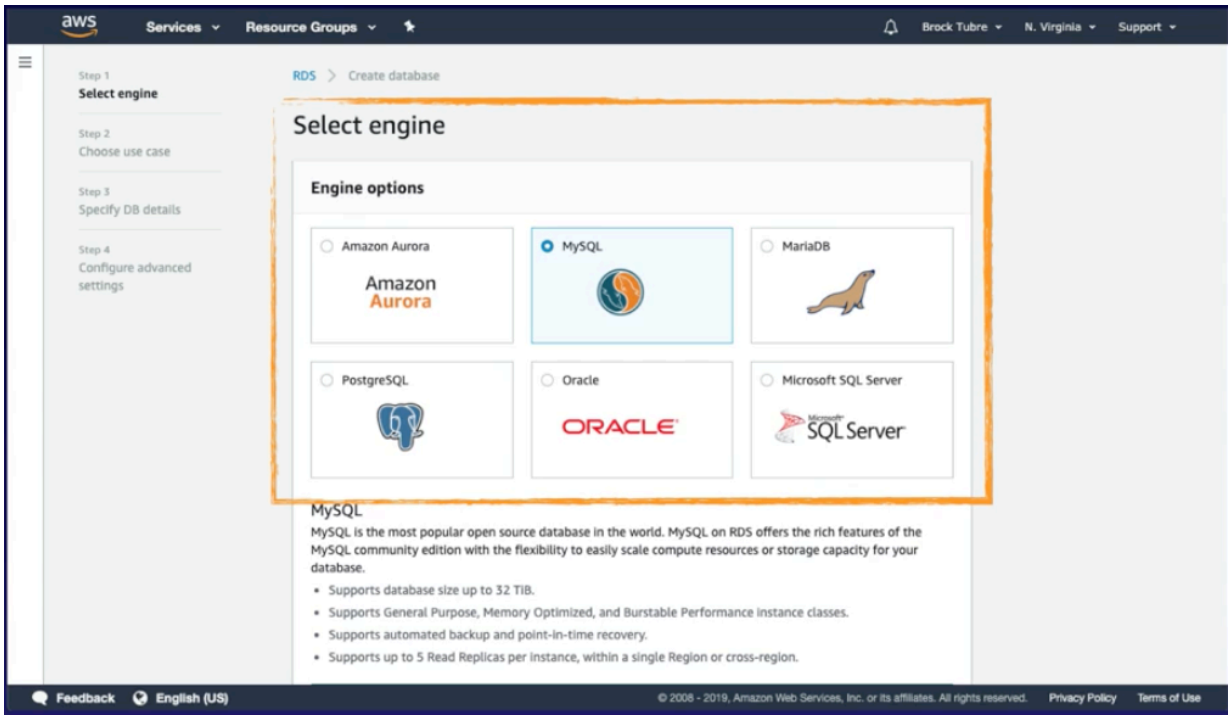
How to upload:
- via console



- via SDK
- via CLI

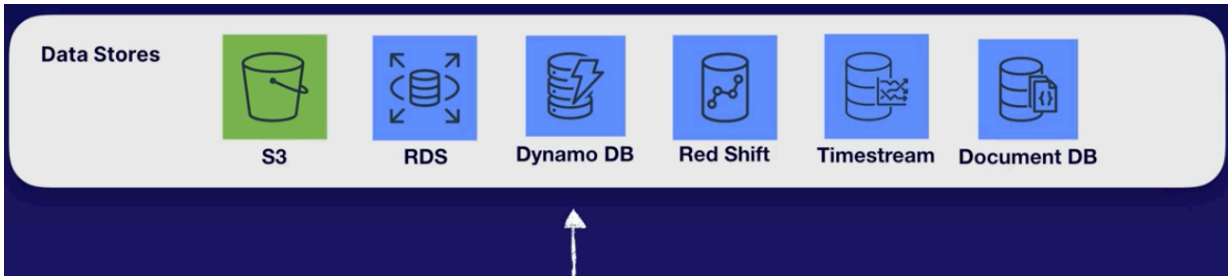
RDS: Relational DB Service


Customer
+Customer Number
Customer Name
Customer Address 1
Customer Address 2
....
Customer/Sales Order
+Customer Number
+Sales Order Header Num
+Version
Sales Order Header
+Sales Order Header Num
Creation Date
Creation Time
Sales Person
....
Sales Order Line Item
+Sales Order Header Num
+Sales Order Line Item Num
Material Number
Material Description
....

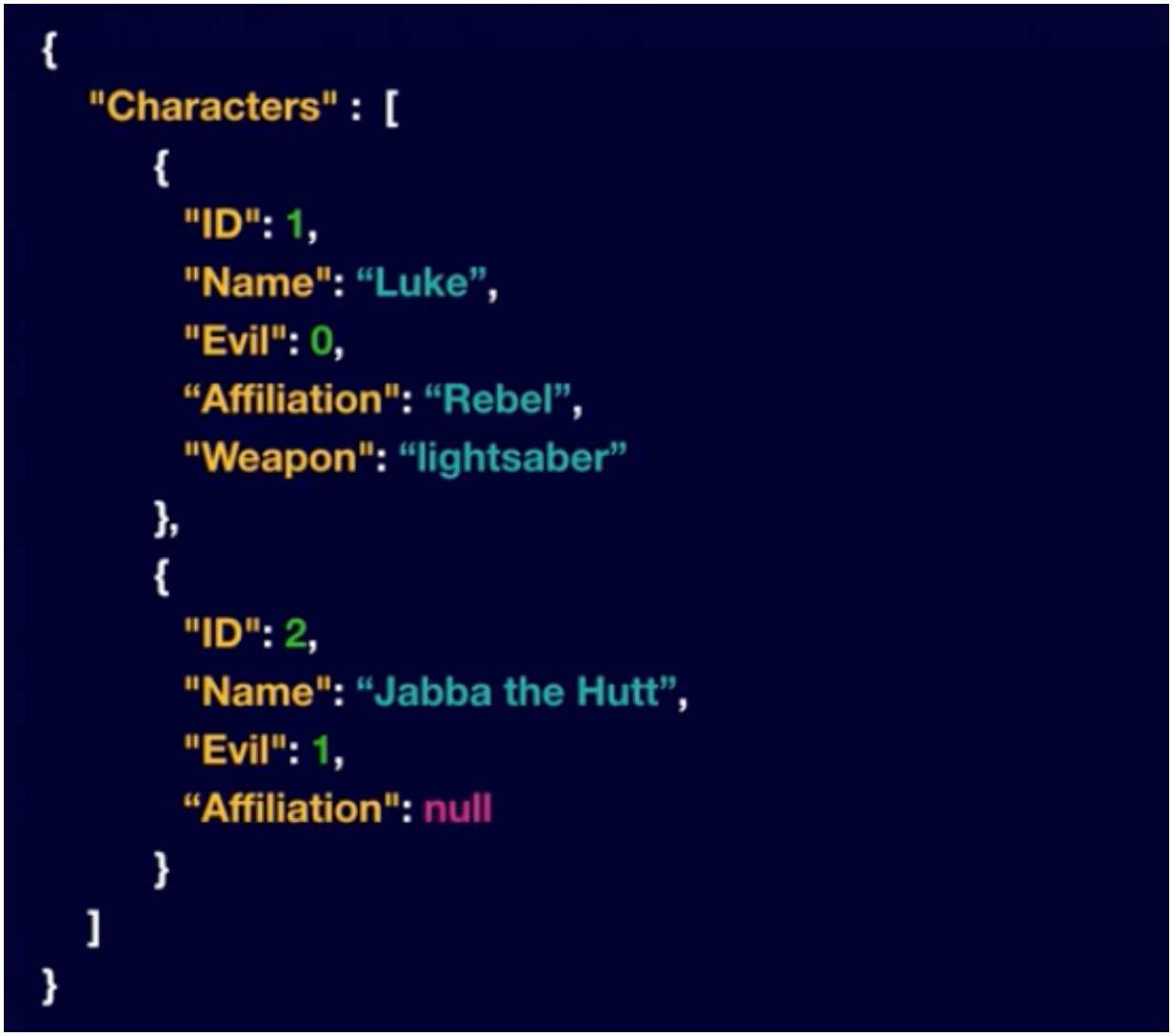


No SQL DataStore:

For schema less data or unstructured



Example: JSON data



In dynamo DB, this json dataset is the TABLE, and the first property is the TABLE Name

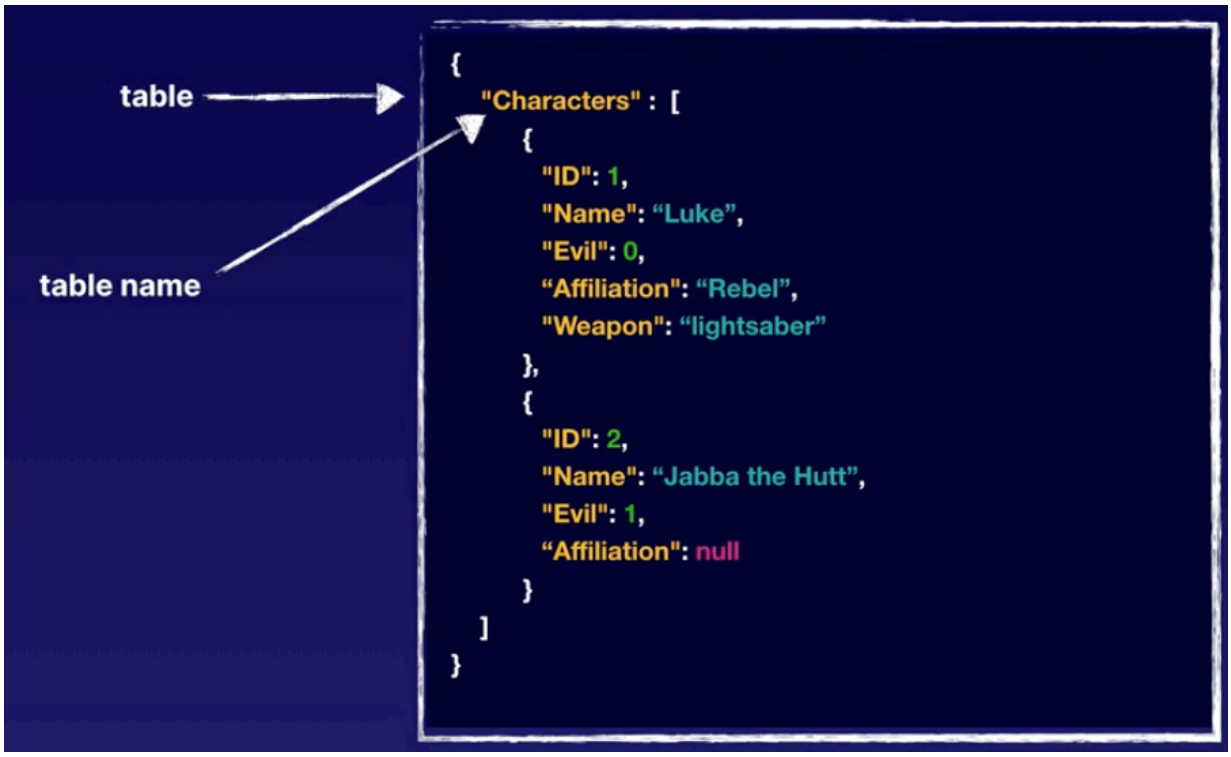
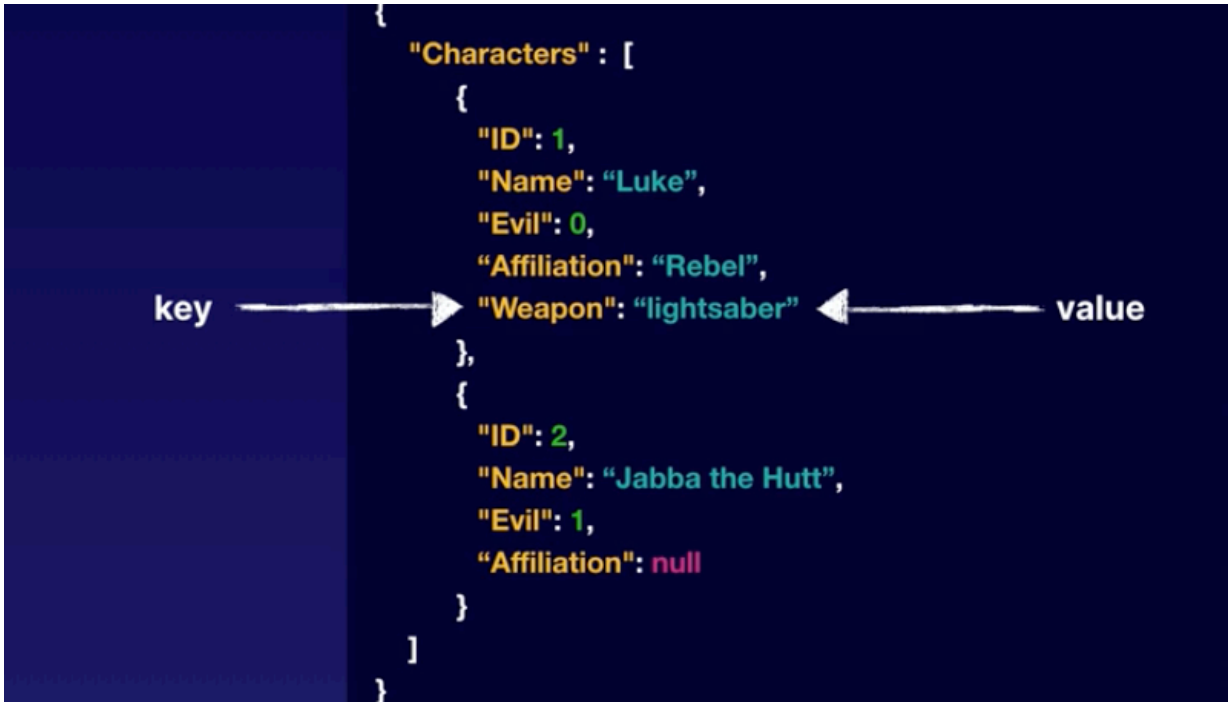
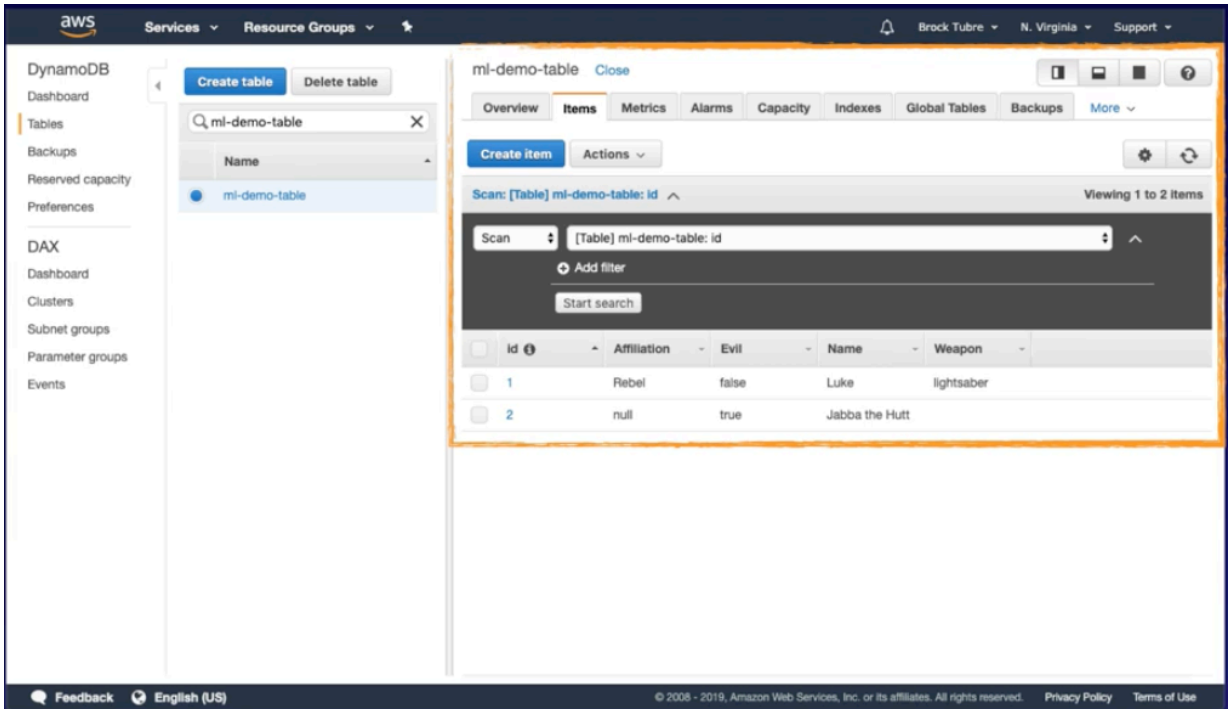


table
table name
{
"Characters" : [
{
"ID": 1,
"Name": “Luke”,
"Evil": 0,
“Affiliation": “Rebel”,
"Weapon": “lightsaber”
},
{
"ID": 2,
"Name": “Jabba the Hutt”,
"Evil": 1,
“Affiliation": null
}
]
}


{
"Characters" : [
{
"ID": 1,
"Name": “Luke”,
"Evil": 0,
“Affiliation": “Rebel”,
"Weapon": “lightsaber”
},
{
"ID": 2,
"Name": “Jabba the Hutt”,
"Evil": 1,
“Affiliation": null
}
]
}
item


{
"Characters" : [
{
"ID": 1,
"Name": “Luke”,
"Evil": 0,
“Affiliation": “Rebel”,
"Weapon": “lightsaber”
},
{
"ID": 2,
"Name": “Jabba the Hutt”,
"Evil": 1,
“Affiliation": null
}
]
}
key
value

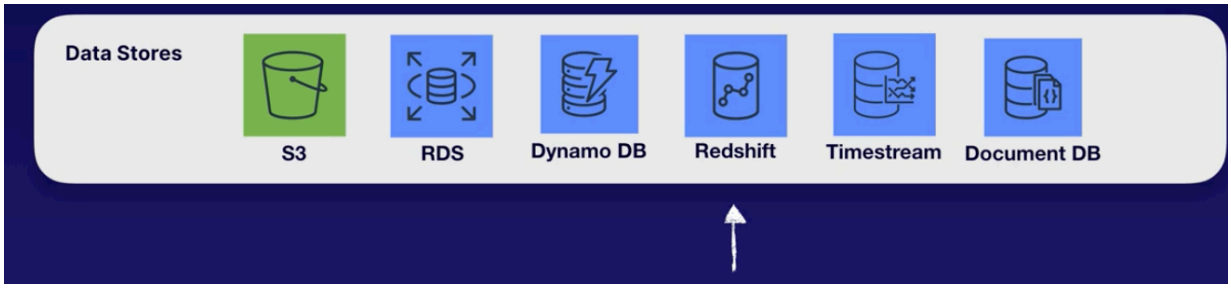
Showing the items in the console:

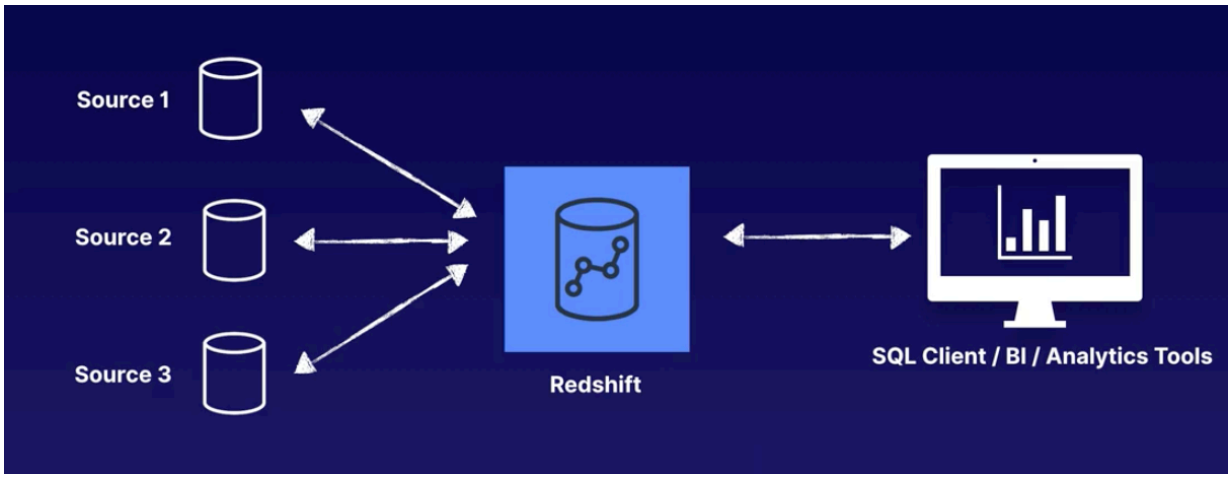


Amazon Redshift:

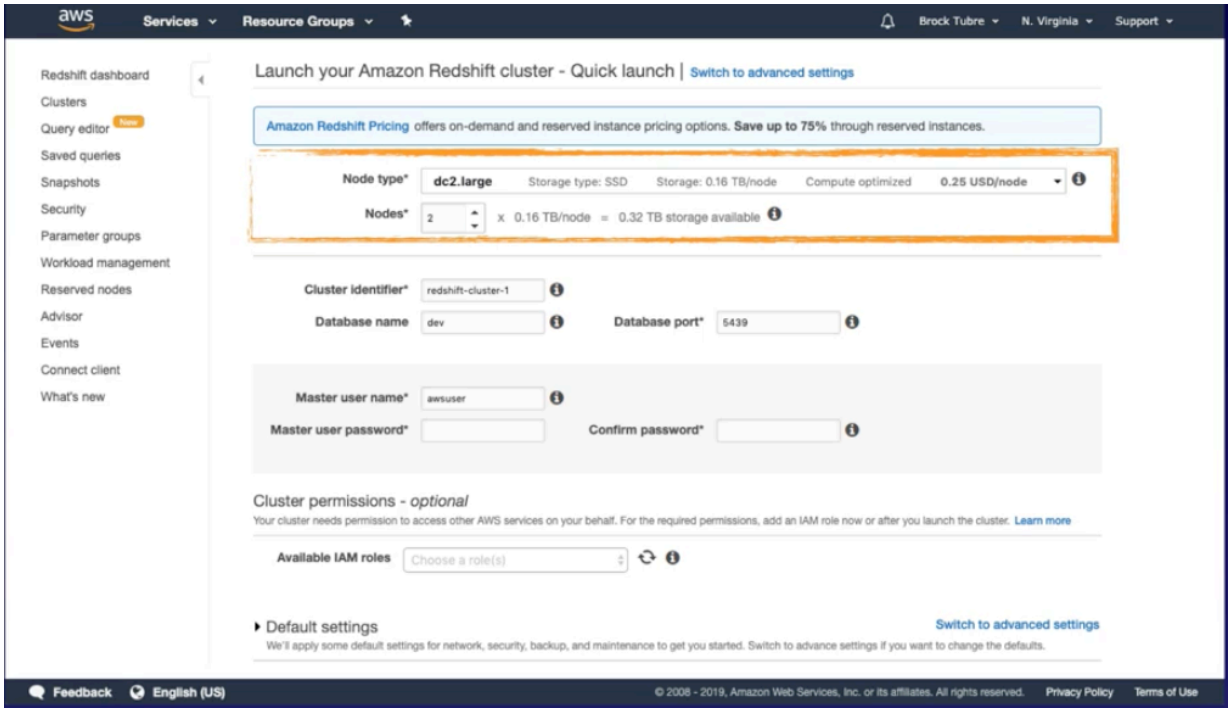
Fully managed clustered peta bytes data warehouse solutions that congregates data from S3, DynamoDB,... and can handle any data type

Once in Redshift, can use SQL like tools or BI tools to query it

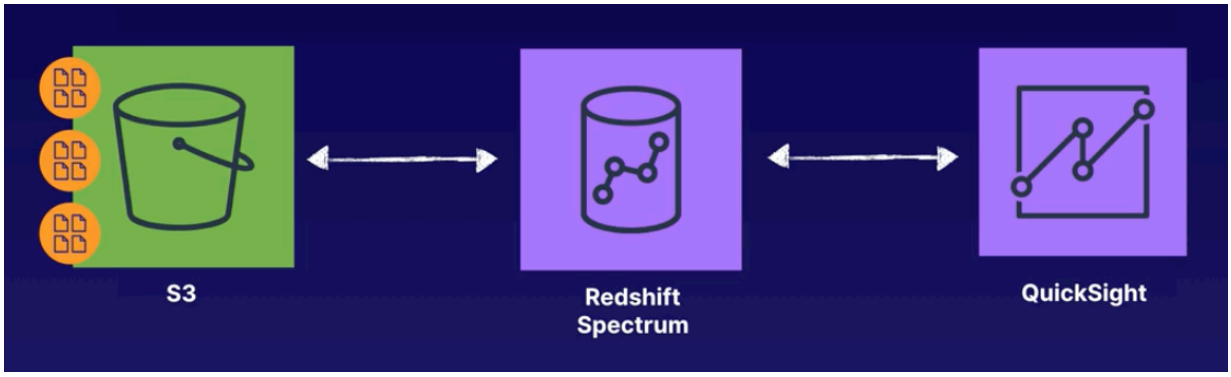




In console, we can set # of nodes

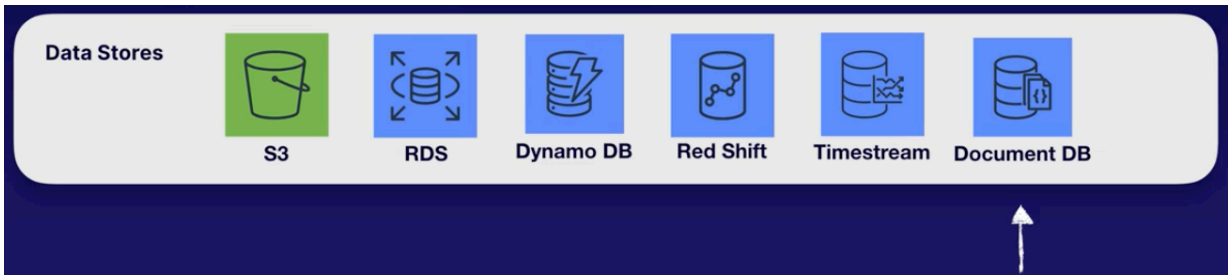


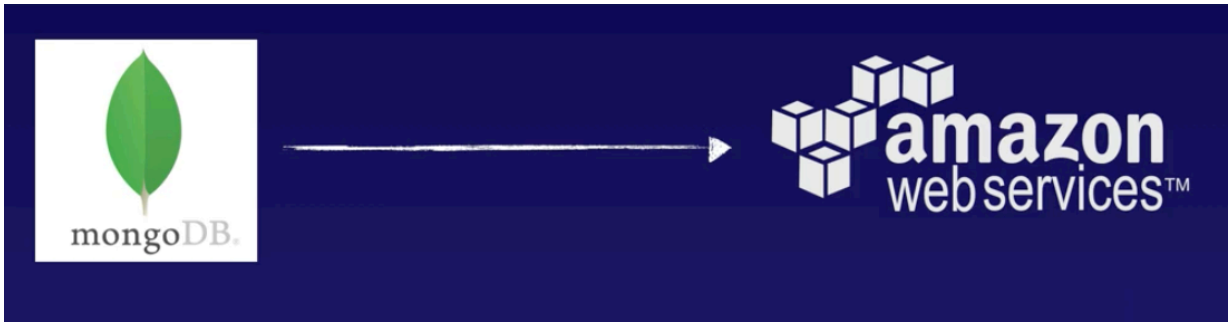
Redshift spectrum can query data from S3 and use tools like Quicksight to visualize it



New **TimeStream**, announced in 2018
To handle time series data
Will probably Not be in the exam

Document DB
Place to migrate MongoDB data
Probably not be in the exam





These services at used at a high level in the exam
Can still go more in depth - solutions architect developer course will provide more details.

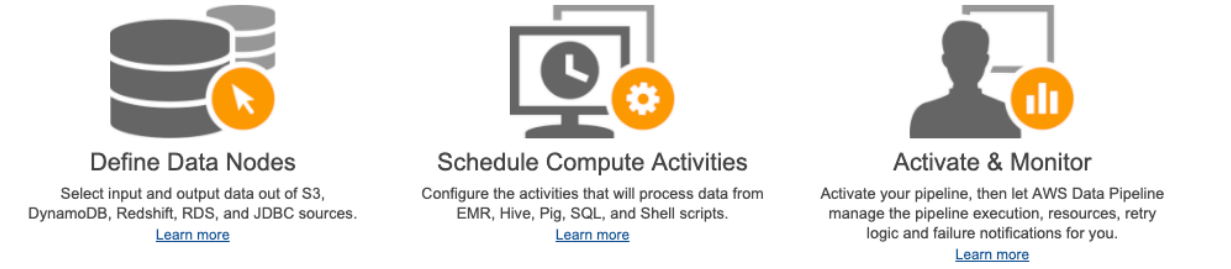
AWS Migration Tools
To help get our data into S3 so it's ready fo ML process

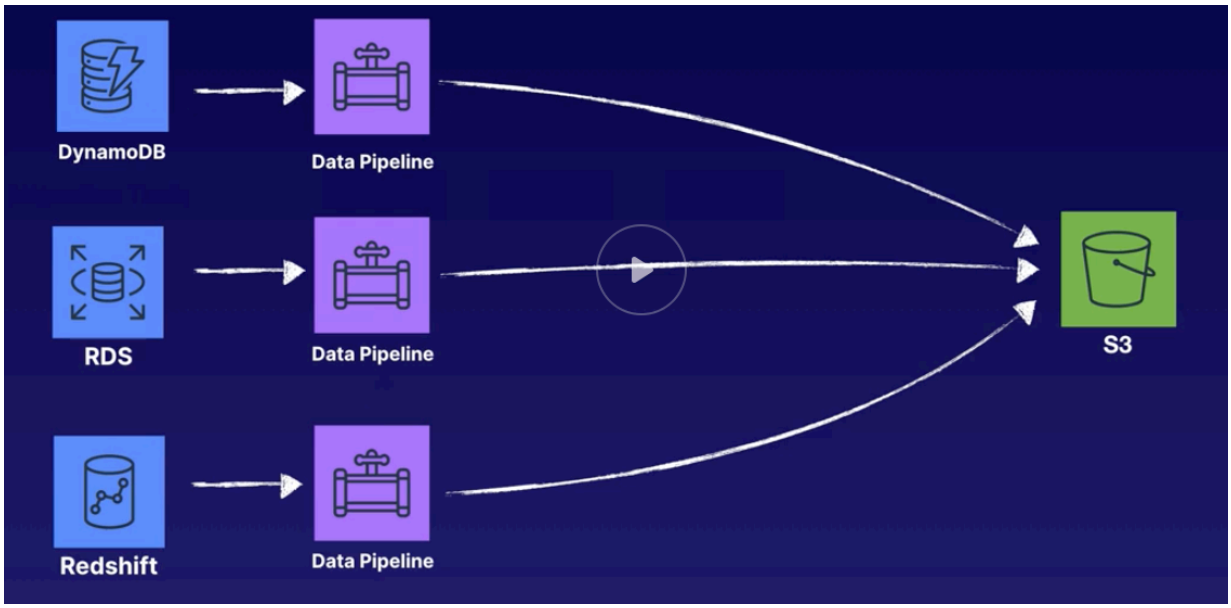
Can be a questions: If data in one location, what is the best tool or strategy to get it into another location

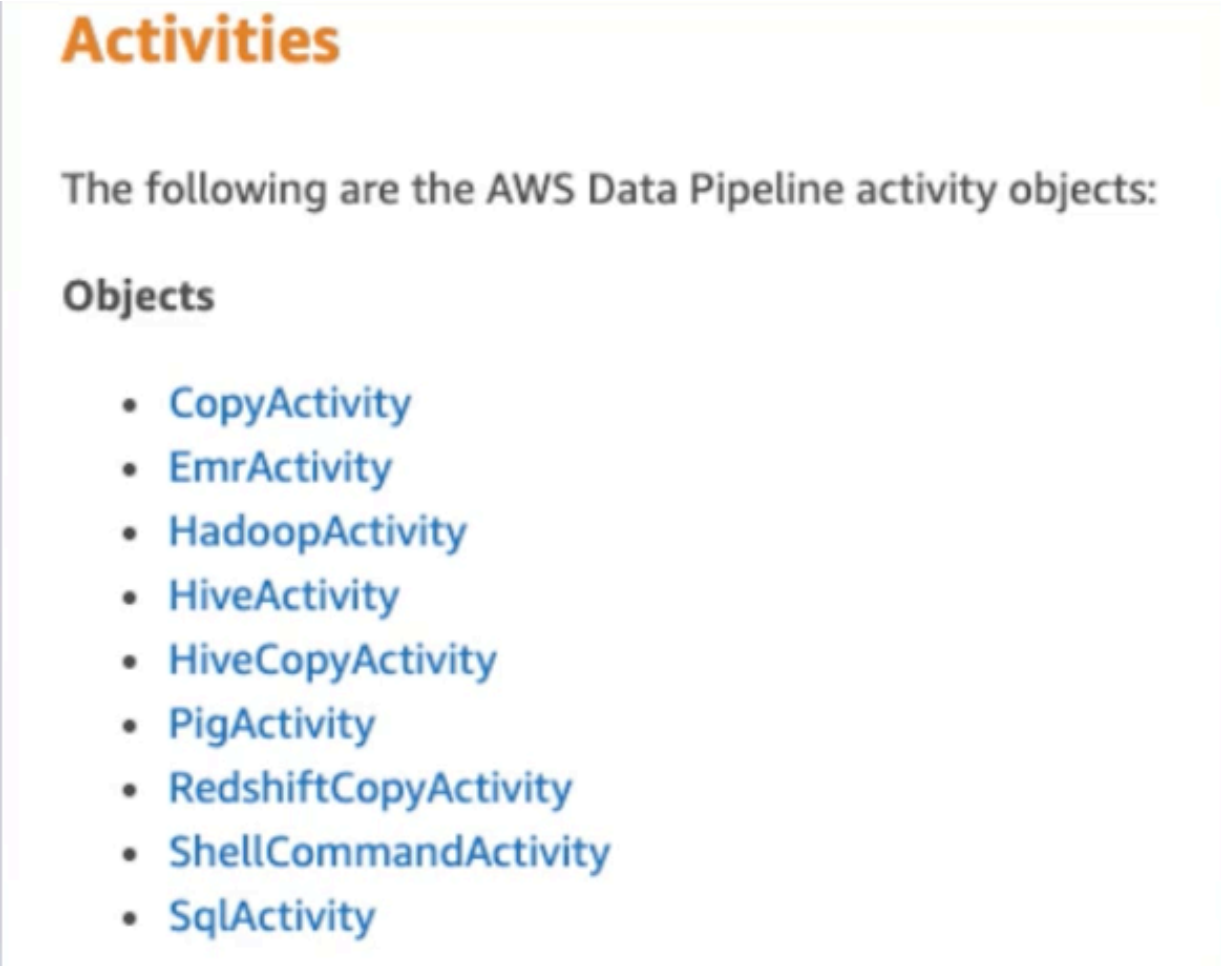
Data Pipeline


Migration Tools
Data Pipeline
DMS
AWS Glue


AWS Data Pipeline
AWS Data Pipeline helps you move, integrate, and process data across AWS compute and storage resources, as well as your on-premises resources. AWS Data Pipeline supports integration of data and activities across multiple AWS regions.
Get started now

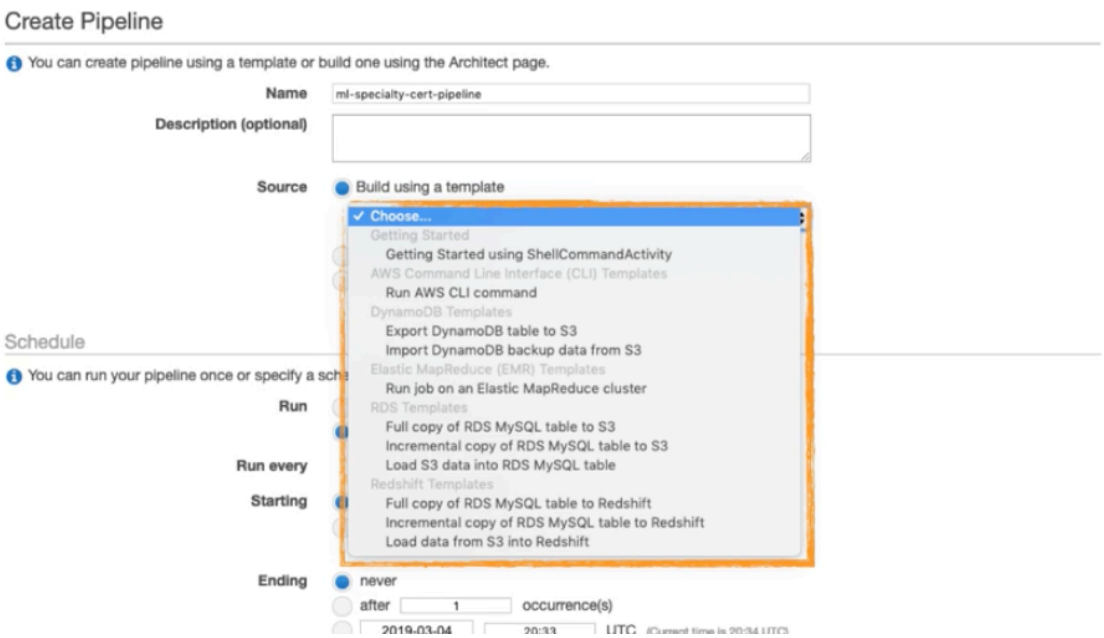

Define Data Nodes
Select input and output data out of S3, DynamoDB, Redshift, RDS, and JDBC sources.
Learn more
Schedule Compute Activities
Configure the activities that will process data from EMR, Hive, Pig, SQL, and Shell scripts.
Learn more
Activate & Monitor
Activate your pipeline, then let AWS Data Pipeline manage the pipeline execution, resources, retry logic and failure notifications for you.
Learn more


DynamoDB
Data Pipeline
RDS
Data Pipeline
Redshift
Data Pipeline
S3

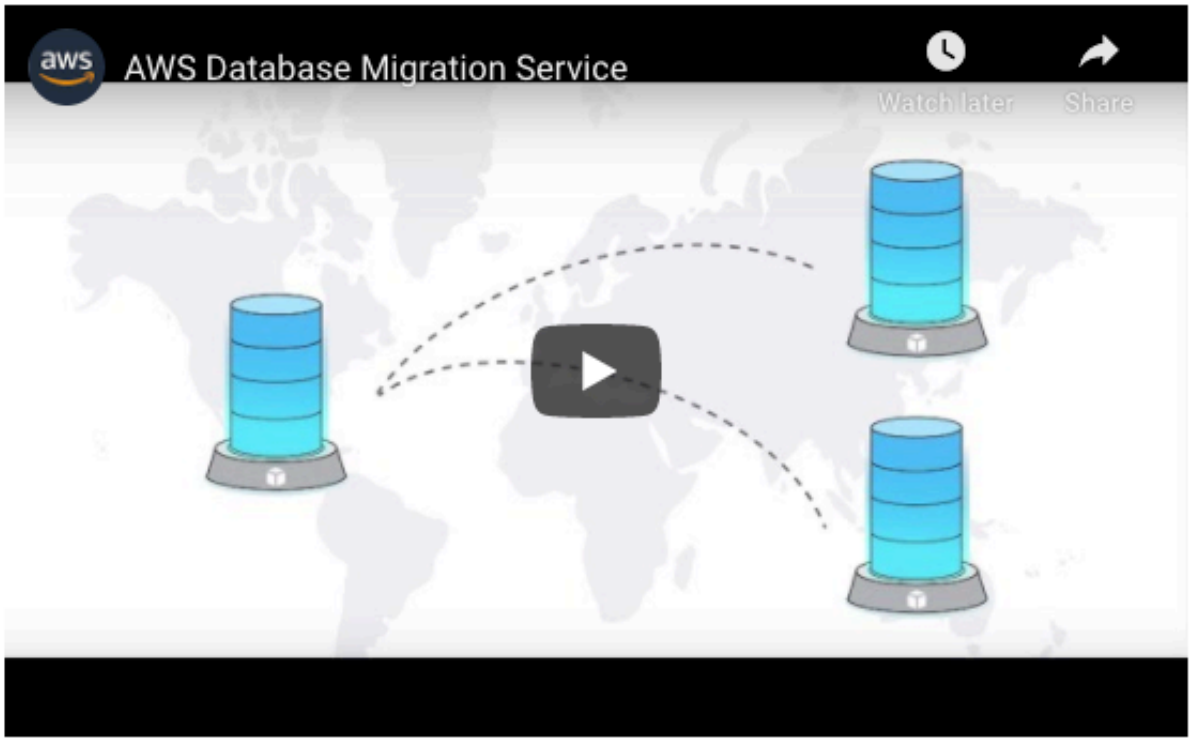


Can use these activities, specify datasource and how we want to migrate the data.
Can run on demand or on schedule
Can also be used as a transformation tool
We can manage the pipeline execution, the resources (EC2) and retry

AWS has some templates already available:



DMS - Database Migration Service



AWS Database Migration Service helps you migrate databases to AWS quickly and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database. The AWS Database Migration Service can migrate your data to and from most widely used commercial and open-source databases.

Use cases Learn more

Homogeneous database migrations

You create a migration task with connections to the source and target databases, then start the migration with the click of a button. AWS Database Migration Service takes care of the rest.

Heterogenous database migrations

Use the AWS Schema Conversion Tool to convert the source schema and code to match that of the target database, and then use the AWS Database Migration Service to migrate data from the source database to the target database.

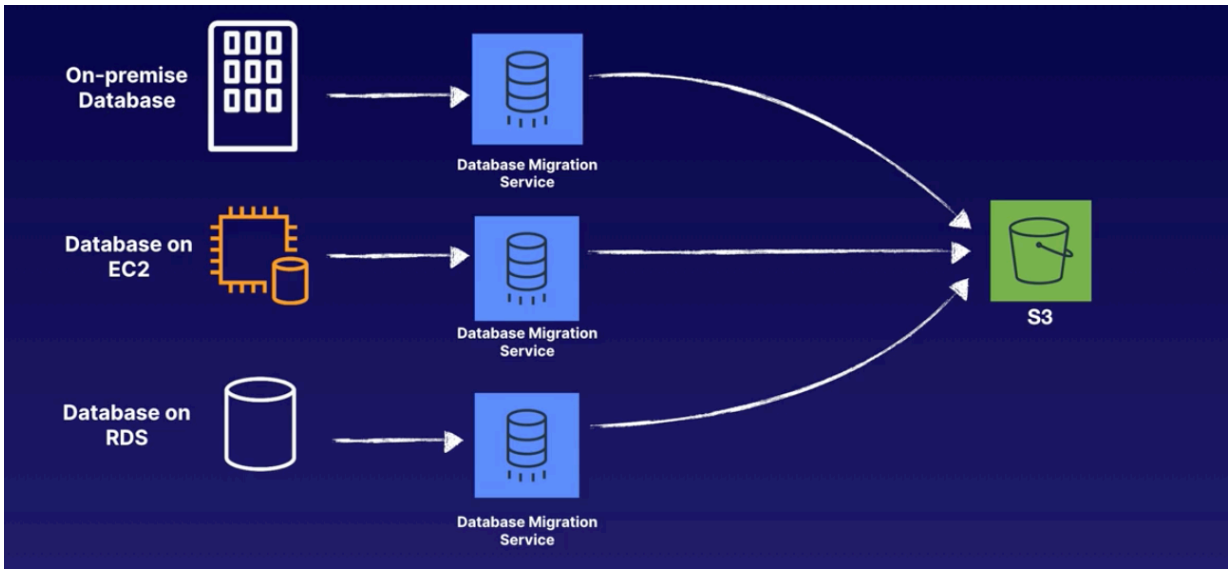
Database consolidation

You can use AWS Database Migration Service to consolidate multiple source databases into a single target database. This can be done for homogeneous and heterogeneous migrations, and you can use this feature with all supported database engines.

Continuous data replication

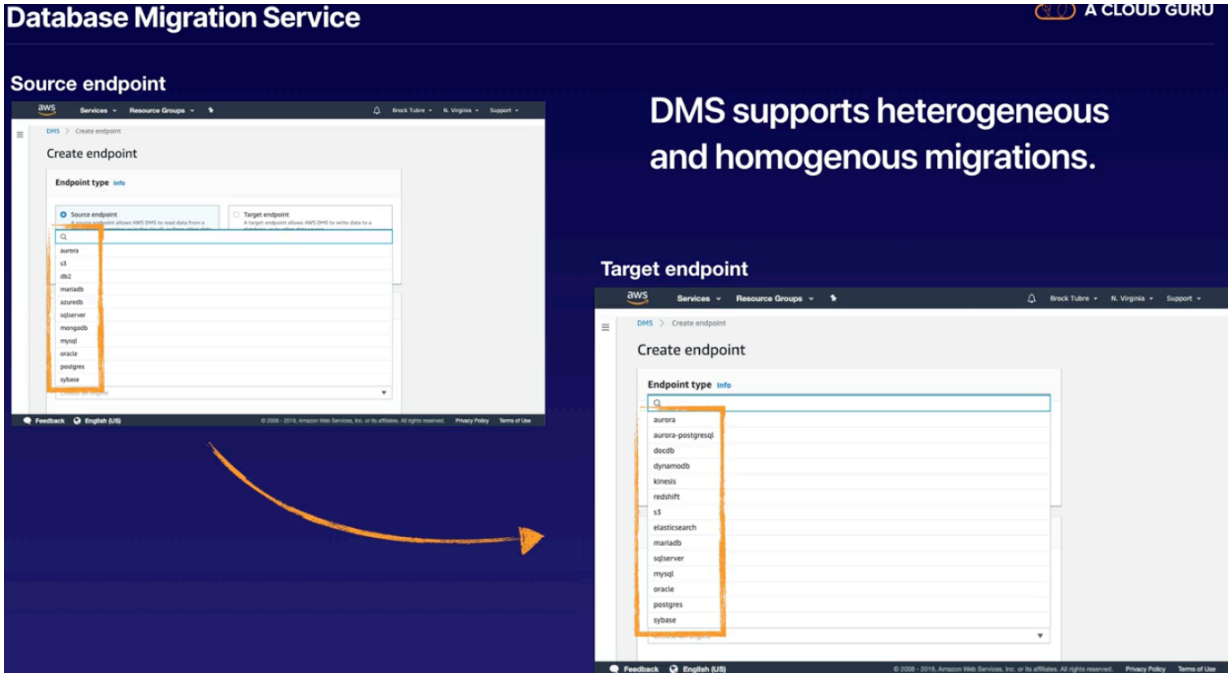
Continuous data replication has a multitude of use cases including Disaster Recovery instance synchronization, geographic database distribution and Dev/Test environment synchronization. You can use DMS for both homogeneous and heterogeneous data replications for all supported database engines.

To transfer data between 2 relational DBs, but we can also output data to S3
Can do SQL Server to S3, or SQL Server to MySQL -> heterogenous migration

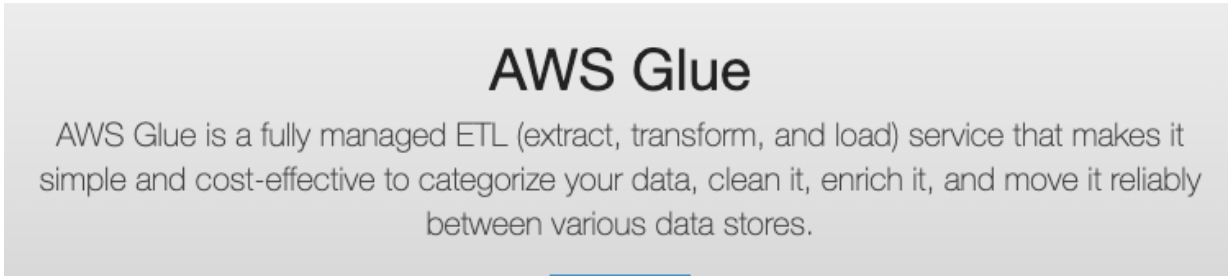


DMS does NOT transfer any transformation
Other than just column name change

Can set up dat source, and pick target



Last Migration Service: AWS Clue



Build your AWS Glue Data Catalog

AWS Glue automatically stores metadata in a central data catalog. It can create table definitions for many common data stores, including, S3 buckets, web logs, and AWS databases. AWS Glue recognizes, infers, organizes, and classifies your data.

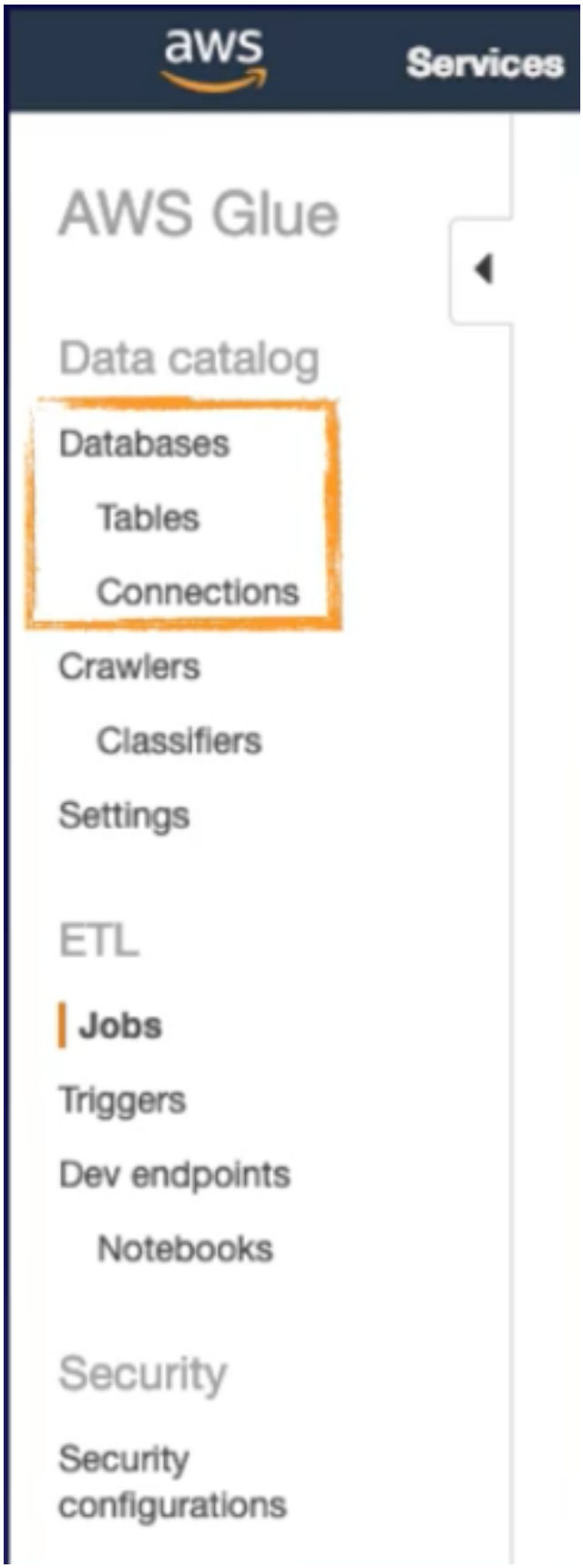
Generate and edit transformations

PySpark transformation scripts are auto generated using source and target metadata. You can store customized versions to transform your data to meet your business needs. AWS Glue provides an environment to modify your jobs.

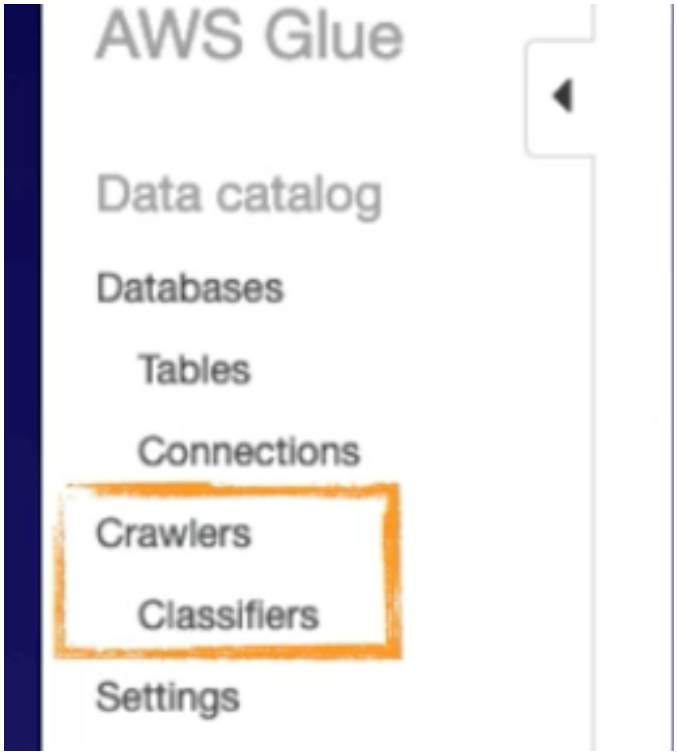
Schedule and run your jobs

AWS Glue runs your ETL jobs in a serverless environment. You don't need to set up the infrastructure, you just use Amazon's infrastructure and pay for the resources you use. You can define triggers to run jobs based on a schedule or event. AWS Glue enables you to monitor your jobs.

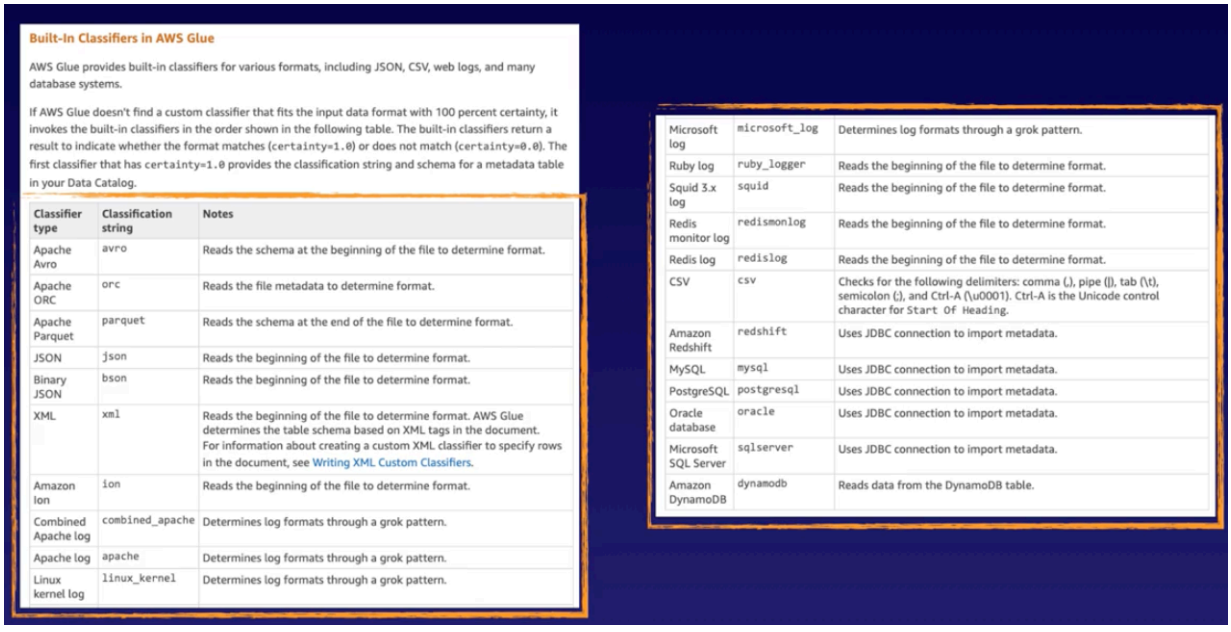
When using Glue, it creates Tables that are related to metadata


aws
Services
AWS Glue
Data catalog
Databases
Tables
Connections
Crawlers
Classifiers
Settings
ETL
Jobs
Triggers
Dev endpoints
Notebooks
Security
Security configurations

To create Tables, we set up a Crawler, that goes out, looks at the data and determine the schema associated with that data



Within crawlers, several classifiers that work at hierarchy level and try to infer what type of data it is.
So works for schema less or JSON data



Built-In Classifiers in AWS Glue

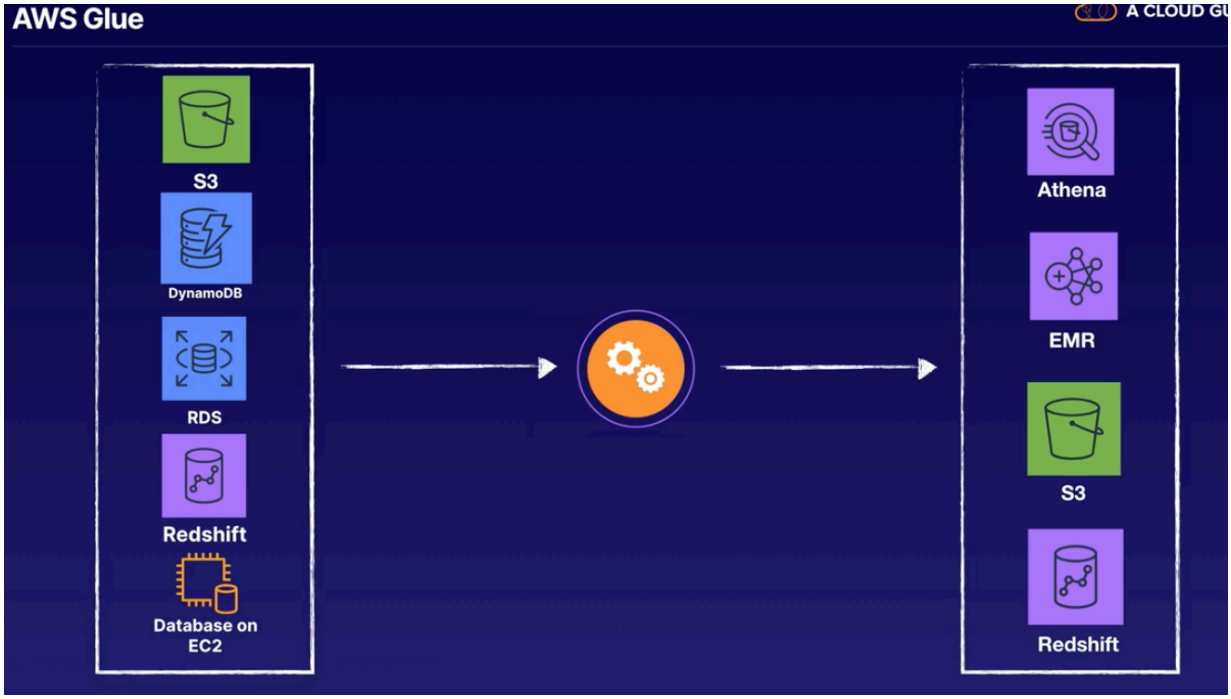
AWS Glue provides built-in classifiers for various formats, including JSON, CSV, web logs, and many database systems.

If AWS Glue doesn't find a custom classifier that fits the input data format with 100 percent certainty, it invokes the built-in classifiers in the order shown in the following table. The built-in classifiers return a result to indicate whether the format matches (`certainty=1.0`) or does not match (`certainty=0.0`). The first classifier that has `certainty=1.0` provides the classification string and schema for a metadata table in your Data Catalog.

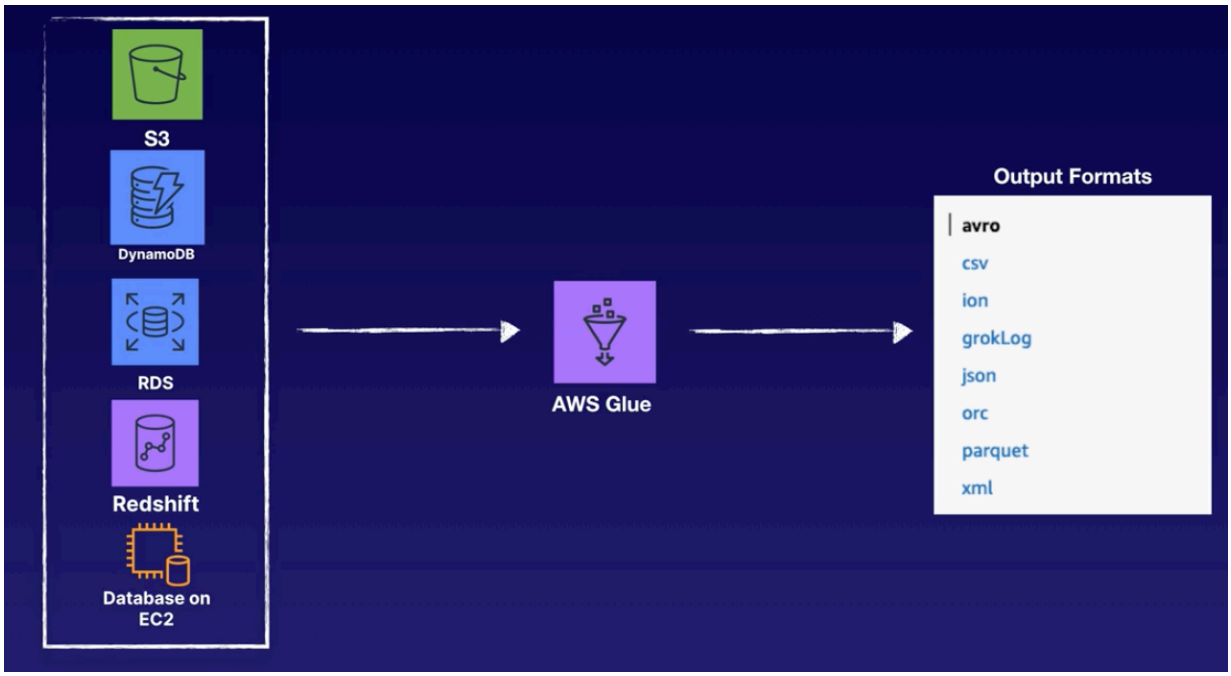
Classifier type	Classification string	Notes
Apache Avro	`avro`	Reads the schema at the beginning of the file to determine format.
Apache ORC	`orc`	Reads the file metadata to determine format.
Apache Parquet	`parquet`	Reads the schema at the end of the file to determine format.
JSON	`json`	Reads the beginning of the file to determine format.
Binary JSON	`bson`	Reads the beginning of the file to determine format.
XML	`xml`	Reads the beginning of the file to determine format. AWS Glue determines the table schema based on XML tags in the document. For information about creating a custom XML classifier to specify rows in the document, see Writing XML Custom Classifiers.
Amazon Ion	`ion`	Reads the beginning of the file to determine format.
Combined Apache log	`combined_apache`	Determines log formats through a grok pattern.
Apache log	`apache`	Determines log formats through a grok pattern.
Linux kernel log	`linux_kernel`	Determines log formats through a grok pattern.
Microsoft log	`microsoft_log`	Determines log formats through a grok pattern.
Ruby log	`ruby_logger`	Reads the beginning of the file to determine format.
Squid 3.x log	`squid`	Reads the beginning of the file to determine format.
Redis monitor log	`redismonlog`	Reads the beginning of the file to determine format.
Redis log	`redislog`	Reads the beginning of the file to determine format.
CSV	`csv`	Checks for the following delimiters: comma (,), pipe (|), tab (\t), semicolon (;), and Ctrl-A (\u0001). Ctrl-A is the Unicode control character for `Start Of Heading`.
Amazon Redshift	`redshift`	Uses JDBC connection to import metadata.
MySQL	`mysql`	Uses JDBC connection to import metadata.
PostgreSQL	`postgresql`	Uses JDBC connection to import metadata.
Oracle database	`oracle`	Uses JDBC connection to import metadata.
Microsoft SQL Server	`sqlserver`	Uses JDBC connection to import metadata.
Amazon DynamoDB	`dynamodb`	Reads data from the DynamoDB table.

To note we can create specific classifiers that specify our data

So with a crawler and classifier, we can pull the data, map it to an appropriate schema and export to various sources like S3



We can also change the format to any ML format

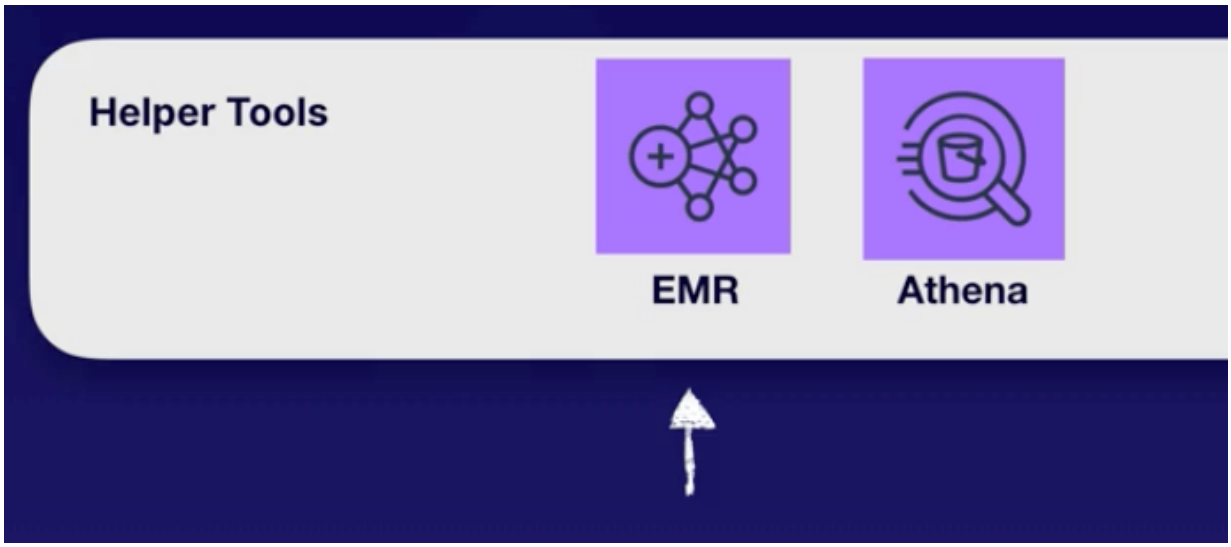


Choose the right approach for migration.
Pull from a DB and export to S3

Datasource	Migration Tool	Why
PostgreSQL RDS instance with training data.	AWS Data Pipeline	Specify SqlActivity query and places the output into S3.
Unstructured log files in S3.	AWS Glue	Create custom classifier and output results into S3.
Clustered Redshift data.	AWS Data Pipeline AWS Glue	Use the **unload** command to return results of a query to CSV file in S3. Create Data catalog describing data and load it into S3.
On-premise MySQL instance with training data.	AWS Database Migration Service	DMS can load data in CSV format onto S3.

AWS Helper Tools

Amazon EMR

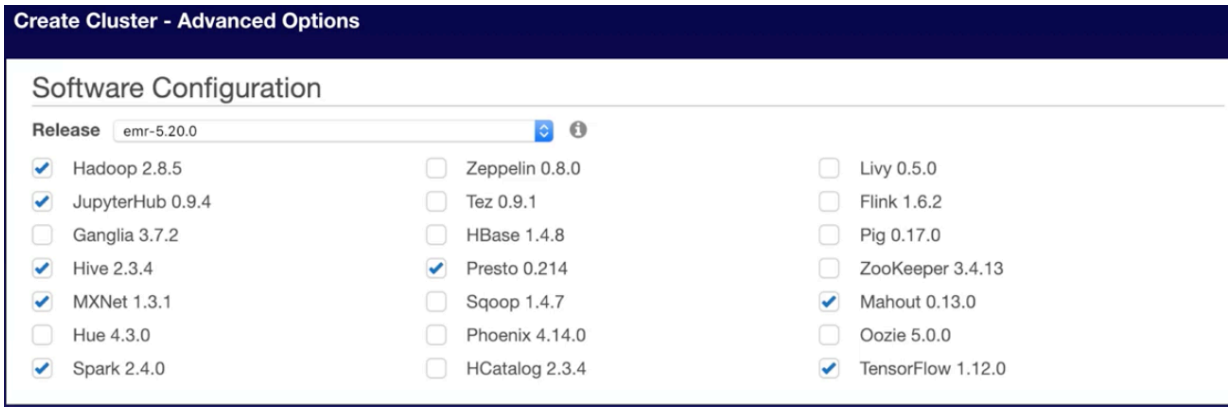


Fully managed Hadoop Cluster ecosystem that runs on multiple EC2 instances

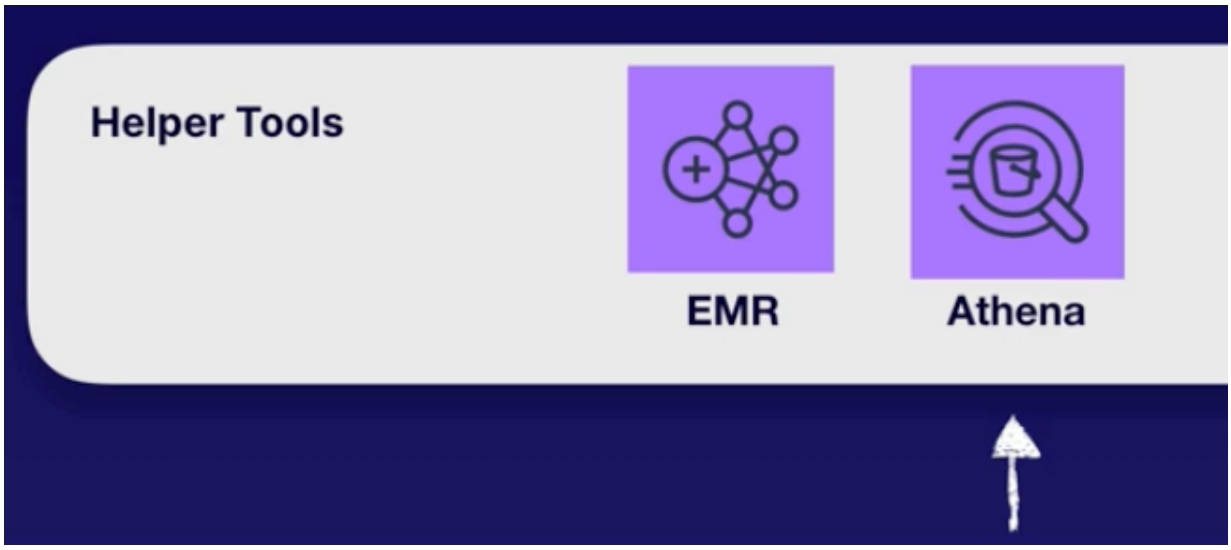
Pick and choose different frameworks we want to use in a cluster
To run different workload across multiple EC2 for petabytes worth of data

We can use EMR to store vast amount of files in a different file system, to use as an Input or Training data
If data is already in EMR, we can use frameworks/services to migrate the data into S3, or use AWS Data Pipeline to load the data into S3

When creating a cluster, we can use different frameworks:

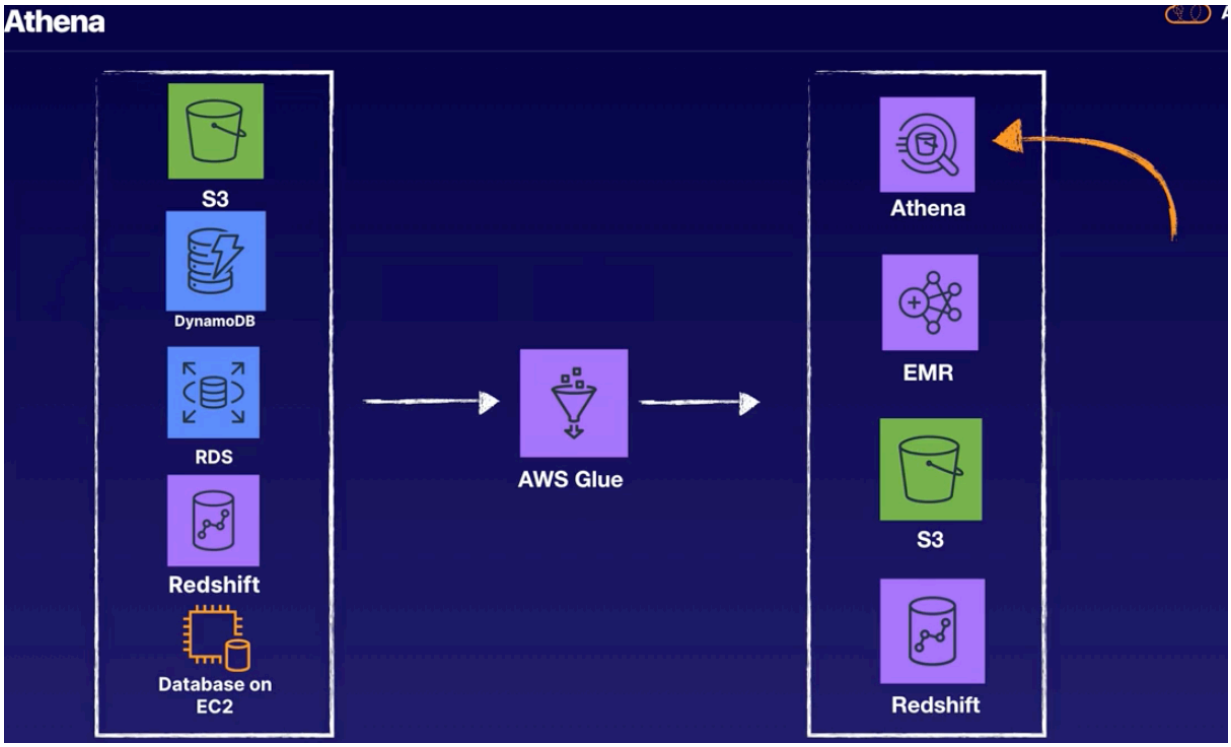


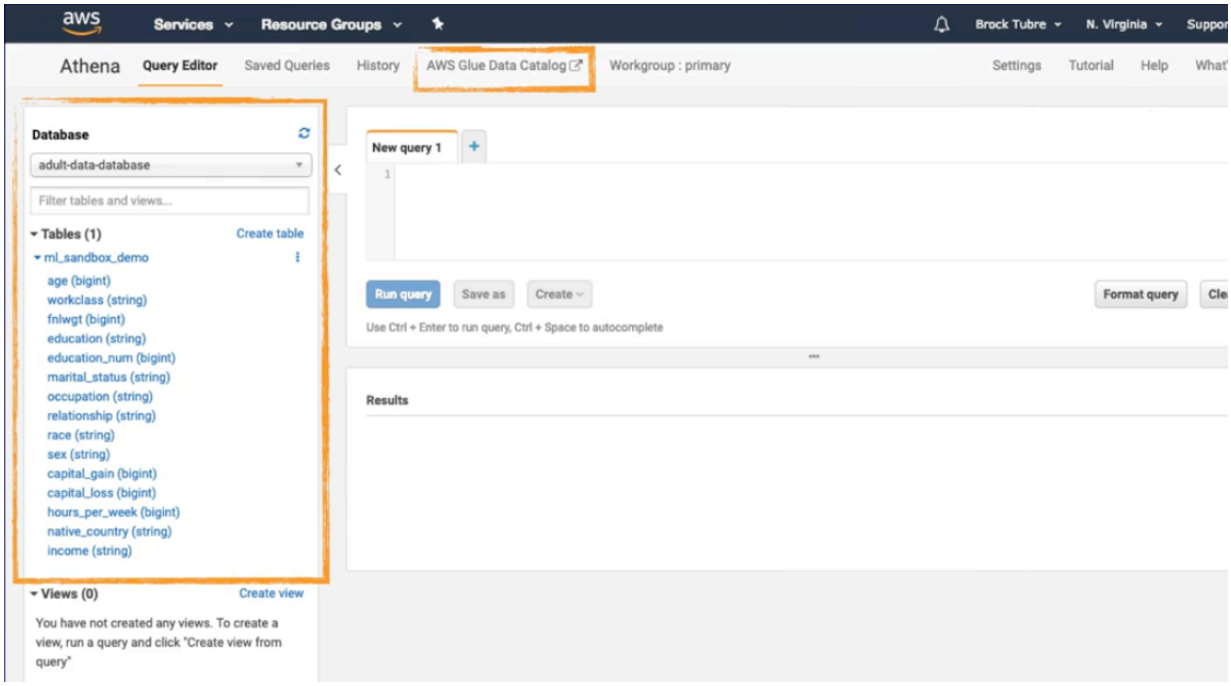
Amazon Athena

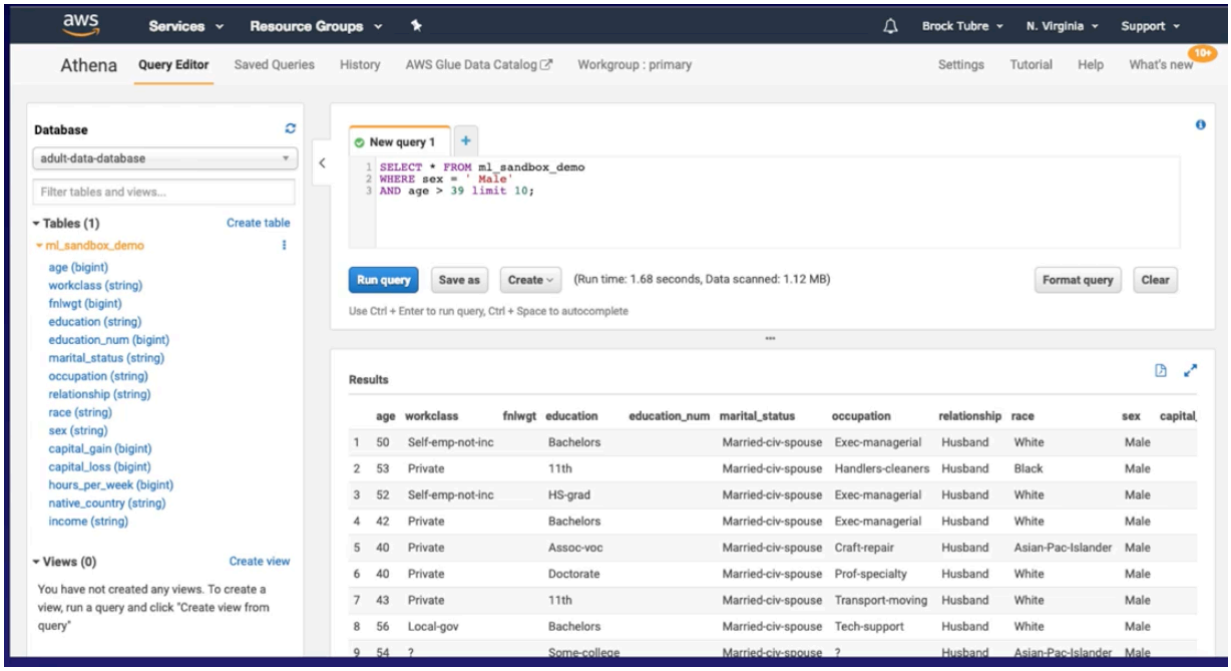


Serveless platform to run SQL queries on S3

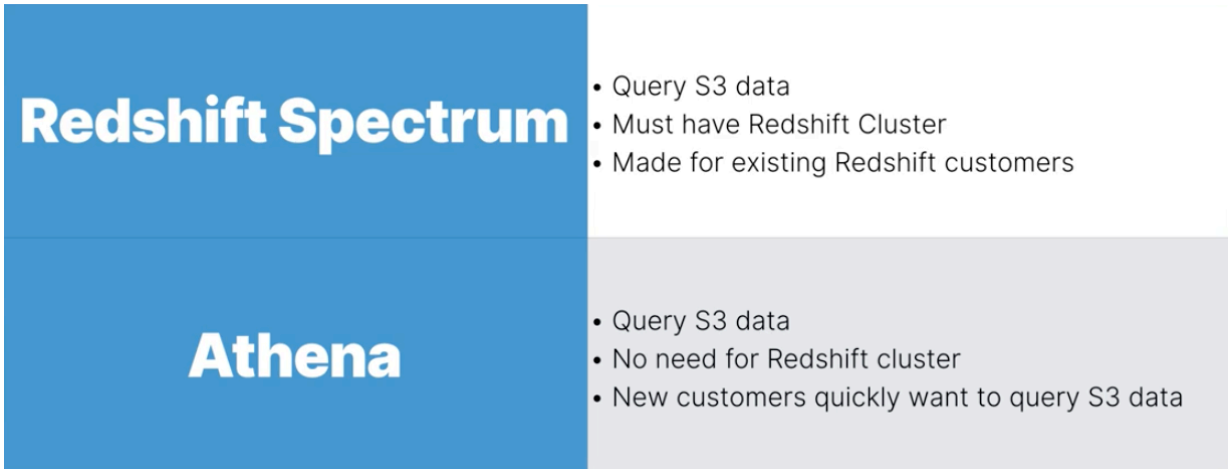
We can set up a table within data catalog within AWS Glue, and use Athena to query S3


Athena
S3
DynamoDB
RDS
Redshift
Database on EC2
AWS Glue
Athena
EMR
S3
Redshift


aws
Services
Resource Groups
Brock Tubre
N. Virginia
Athena
Query Editor
Saved Queries
History
AWS Glue Data Catalog
Workgroup : primary
Settings
Tutorial
Help
Database
adult-data-database
Filter tables and views...
Tables (1)
Create table
ml_sandbox_demo
age (bigint)
workclass (string)
fnlwgt (bigint)
education (string)
education_num (bigint)
marital_status (string)
occupation (string)
relationship (string)
race (string)
sex (string)
capital_gain (bigint)
capital_loss (bigint)
hours_per_week (bigint)
native_country (string)
income (string)
Views (0)
Create view
You have not created any views. To create a view, run a query and click "Create view from query"
New query 1
Run query
Save as
Create
Format query
Use Ctrl + Enter to run query, Ctrl + Space to autocomplete
Results



Difference between Athena and Redshift spectrum



Overall, S3 is our one stop shop for Data store for SageMaker


S3
SageMaker