Serverless Data Lake on AWS

Thanh Nguyen
Senior Solutions Architect

① Workshop Lab Guide >> Get Ready!

You

1.1. **Learn** Goals & Architecture

1.2. **Get** your AWS Account ready

1.3. **Quick Setup** Serverless Data Lake

AWS Account: Glue, Athena, QuickSight

Step-by-Step Lab Guide

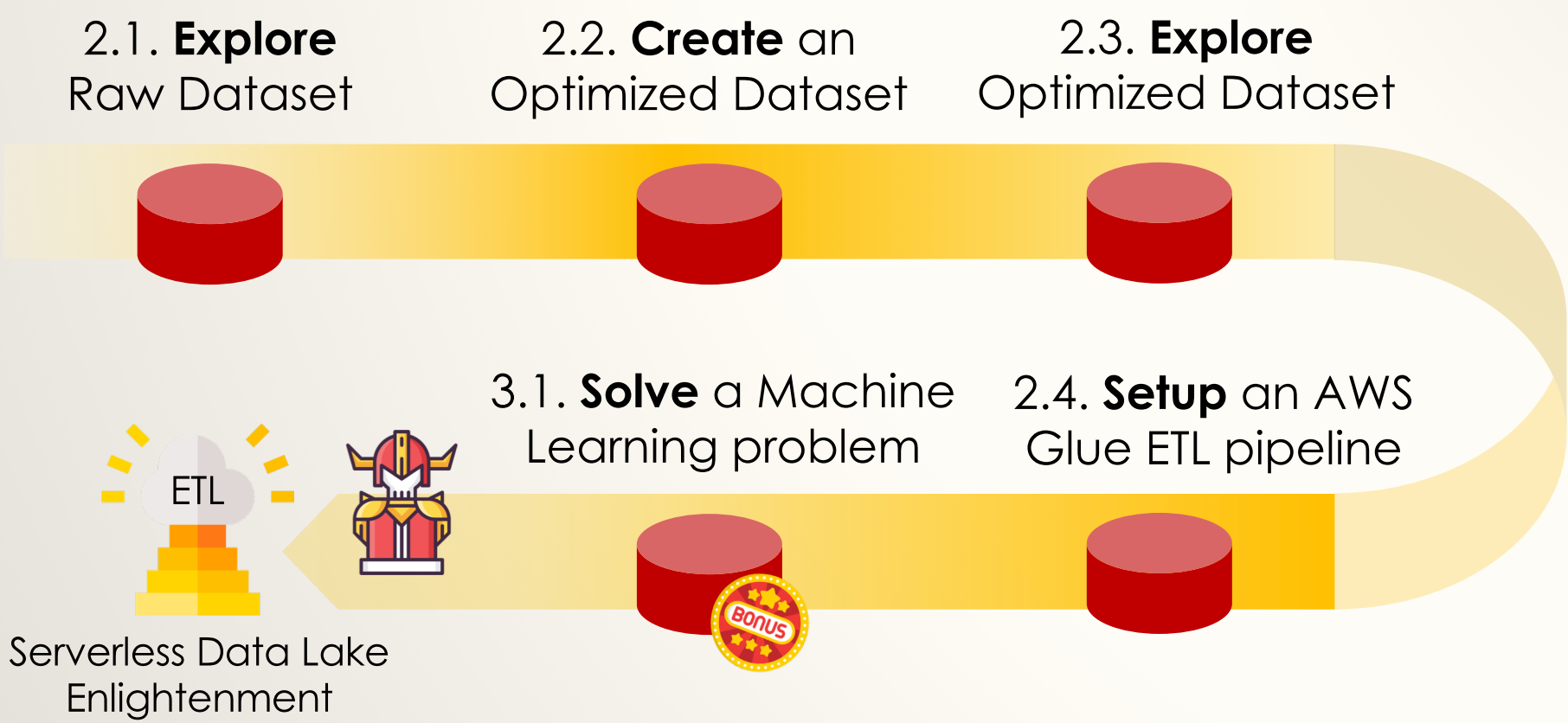
AWS CloudFormation: provision your AWS Cloud resources

Amazon SageMaker ETL Notebook

NYC-Taxi Create Optimized Dataset Job

Amazon SageMaker XGBoost ETL Notebook

② Workshop Lab Guide >> Practice & Learn



① Workshop Lab Guide >> Get Ready!

- **Data Analytics** at Unicorn-Taxi-Company
- Your dataset: **NYC Taxi Trips**
- **Simplified Raw Dataset** vs. **Original Raw Dataset**

Data Analytics at Unicorn-Taxi Company

Create a Dataset for
Reporting and Visualization

Cleanse

Transform

Optimize for reporting queries

Help solve a
Machine Learning problem

Unicorn-Taxi's Data Scientists need to understand **passenger tipping behavior**

Your dataset: **NYC Taxi Trips**

Yellow and Green taxi trips

- Pick-up and drop-off **Dates/Times**
- Pick-up and drop-off **Locations**
- Trip **Distances**
- Itemized **Fares**
- Tip **Amount**
- Driver-reported **Passenger Counts**

For-Hire Vehicle (FHV)

- Pick-up and drop-off **Dates/Times**
- Pick-up and drop-off **Locations**
- Dispatching base

https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Your dataset: **NYC Taxi Trips**

Original Raw Dataset

Green and Yellow Taxi + FHV

Years 2009 to 2018

~1.6Bn rows

215 files

253 GB total

Simplified Raw Dataset

Only Yellow Taxi + few look-ups

Jan to March 2017

~2M rows

3 files

2.5 GB+ uncompressed

Ready in a publicly accessible Amazon S3 bucket

① Workshop Lab Guide >> Get Ready!

- Navigate to the [Serverless Data Lake Workshop website]

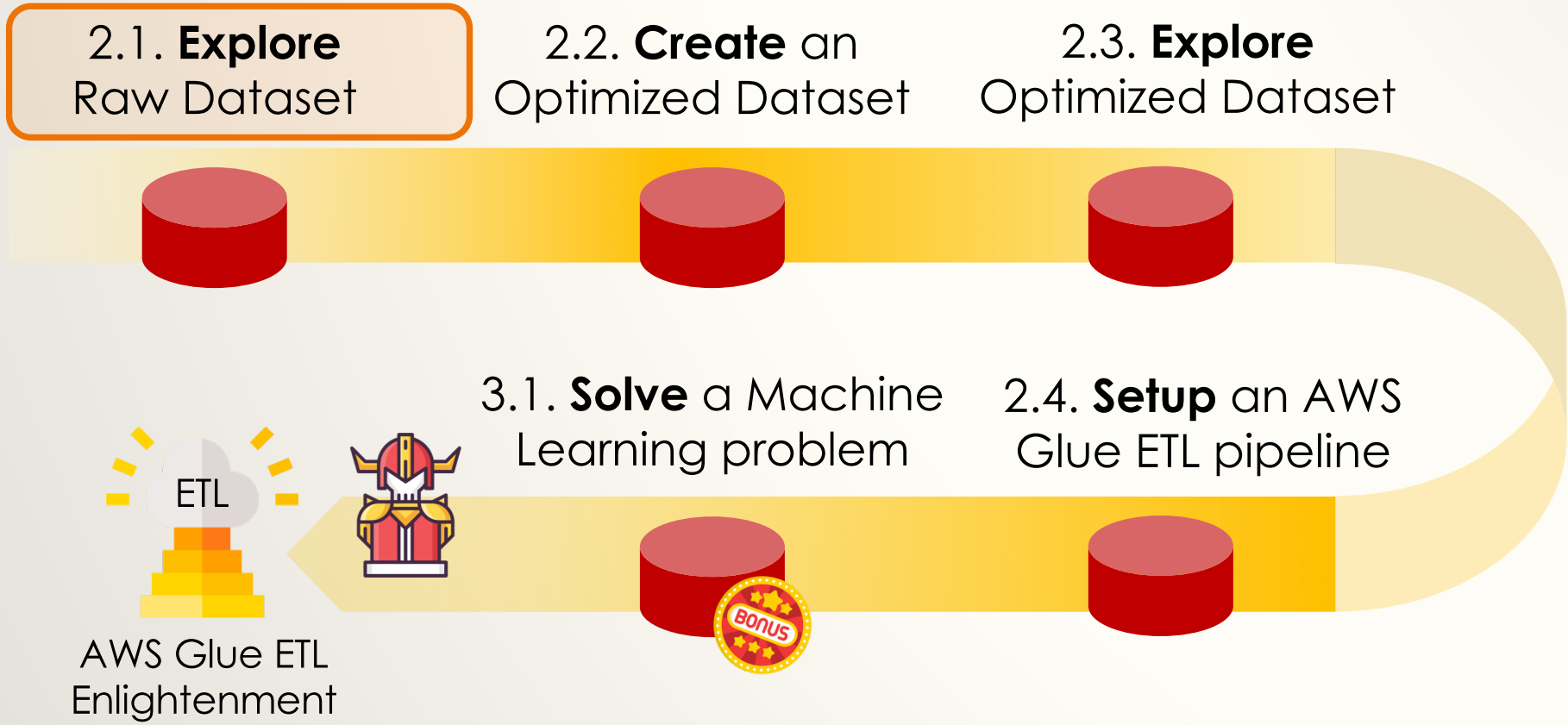
① Workshop Lab Guide >> Get Ready!

Setup Prerequisite Resources using AWS CloudFormation

1. A new Data Lake **Amazon S3** bucket
2. Necessary **IAM** Policies & Roles for **AWS Glue**, **Amazon Athena**, and **Amazon SageMaker**
3. An **AWS Glue Development Endpoint**
4. A number of named queries in **Amazon Athena**

Finally, the **NYC Taxi Trips Raw Dataset is copied** into your Amazon S3 bucket

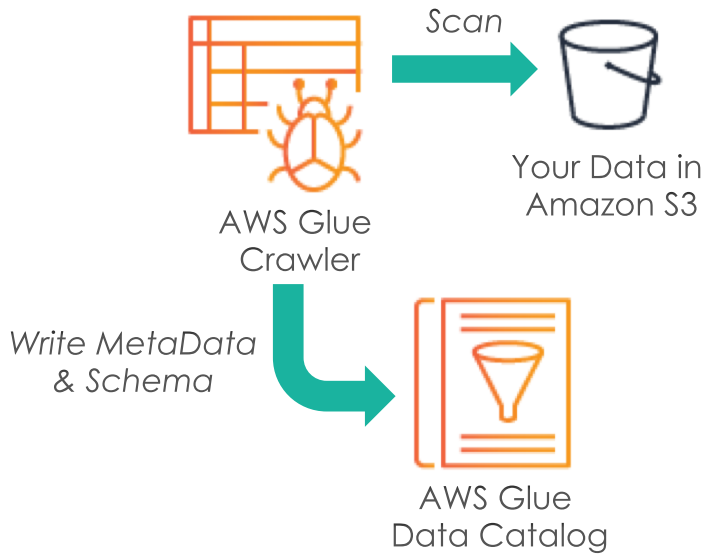
② Workshop Lab Guide - Practice & Learn



AWS Glue Crawlers and the **AWS Glue Data Catalog**

Crawler …

- **Samples and Classifies** your data
- Extracts MetaData
- **Infers Schema** and partitioning format
- **Creates Tables** in your account's **AWS Glue Data Catalog**

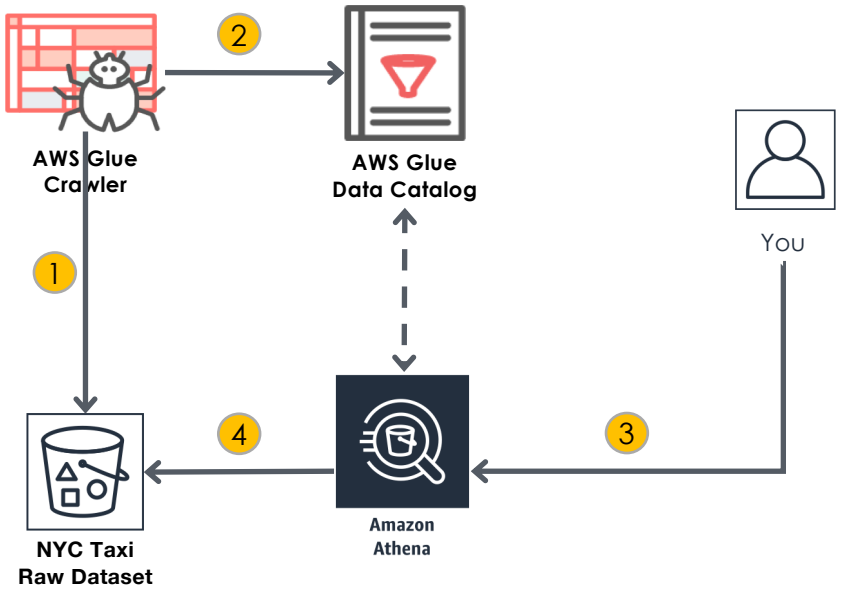


Why Query with **Amazon Athena?**

- **Interactive** query service
- Makes it easy to analyze data in Amazon S3 using **standard SQL**
- Out-of-the-box **integrated with AWS Glue Data Catalog**
- Serverless

Concepts in Action

1. Crawler crawls **Raw Dataset** in Amazon S3 Bucket
2. Crawler writes metadata into AWS Glue Data Catalog
3. You query **Raw Dataset** in Athena
4. Athena uses Schema definition to read Raw Dataset from S3 and returns results



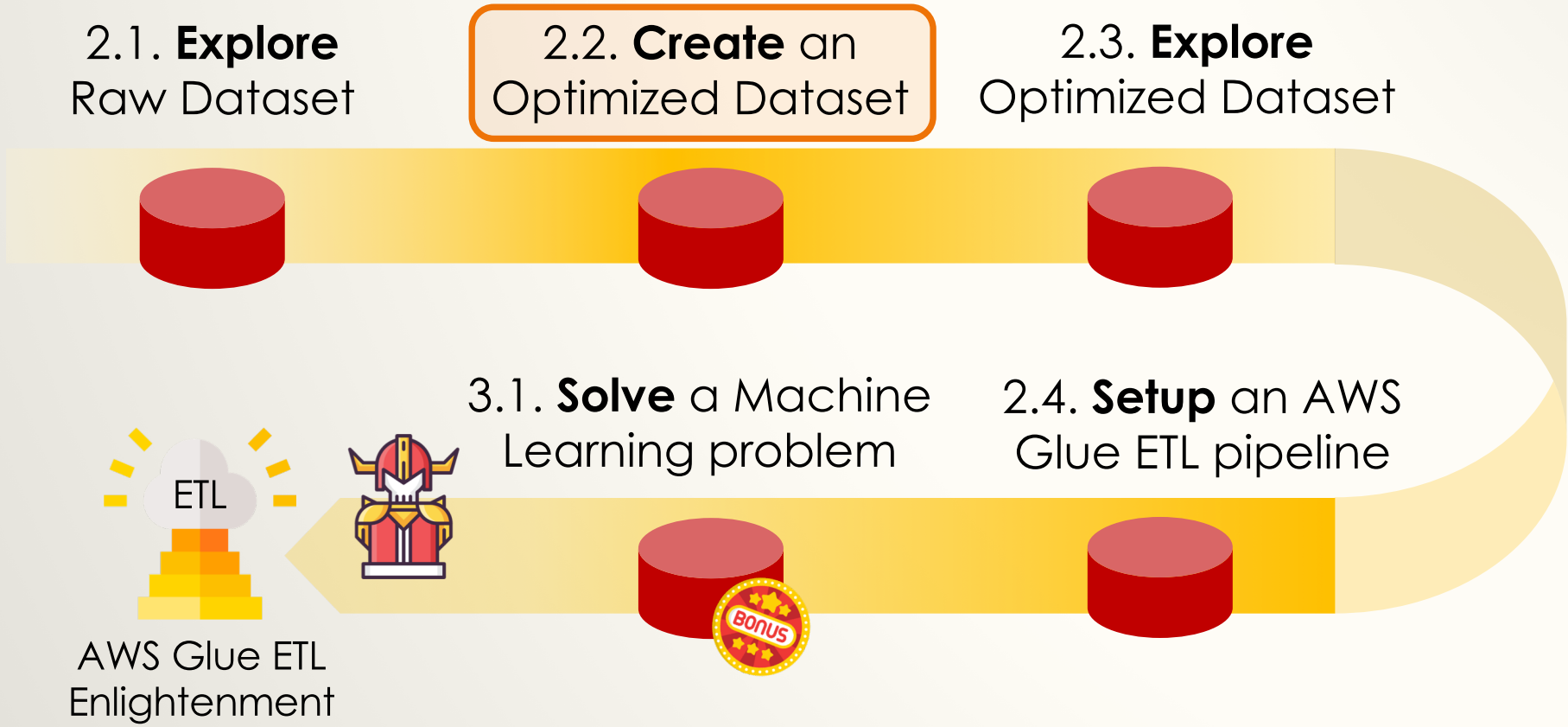
Hand-ons

2.1.1. Catalog Raw Data with an AWS Glue Crawler

2.1.2. Explore Table Schema and Metadata in AWS Glue Data Catalog

2.1.3. Query Raw Data with Amazon Athena

② Workshop Lab Guide - Practice & Learn



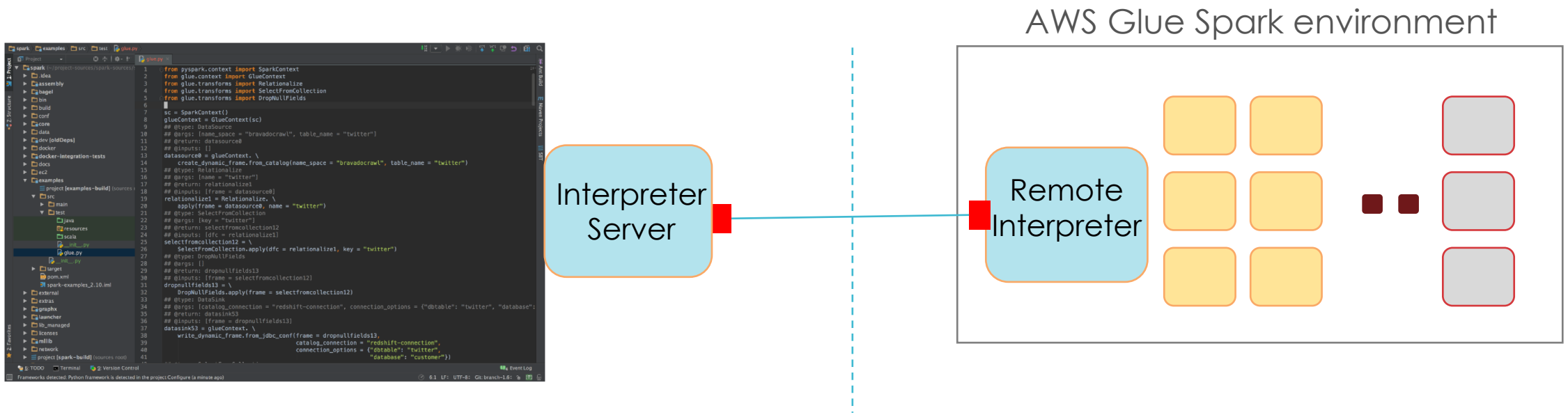
Interactive ETL Development with **Glue** & **SageMaker**

What is Amazon SageMaker?

- … is a **fully-managed ML Platform**
- … enables you to build, train, and deploy **Machine Learning Models** at any scale
- … provides a **Jupyter Notebook environment**

Amazon SageMaker

What is an AWS Glue Development Endpoint?



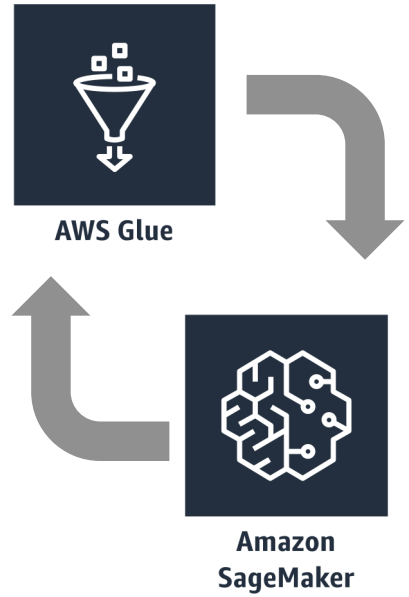
Connect your IDE to an AWS Glue Development Endpoint.

Environment to **interactively develop,** debug, and test ETLcode.

Interactive ETL Development with **Glue** & **SageMaker**

AWS Glue enables you to

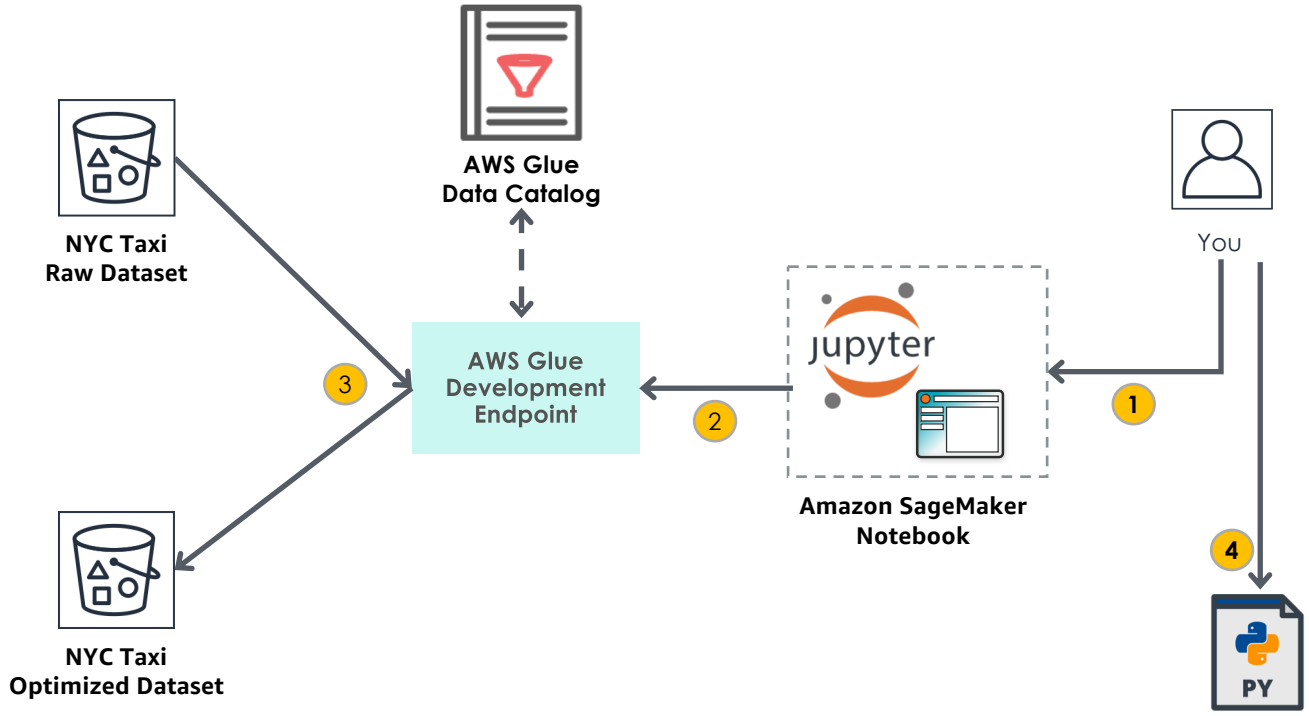
- ❑ … **create** an **Amazon SageMaker Notebook** environment
- ❑ … **connect** it to an **AWS Glue Dev Endpoint**
- ❑ … **and finally, access** your **Jupyter** notebook



All without leaving the AWS Glue console

Concepts in Action

1. You author ETL code in an Amazon SageMaker Notebook
2. ETL code runs on AWS Glue Dev Endpoint
3. ETL code…
   a. Reads Raw Dataset
   b. Applies transformations
   c. Writes Optimized Dataset back to your Amazon S3 bucket
4. You use your ETL code in notebook to create a script file



Hand-ons

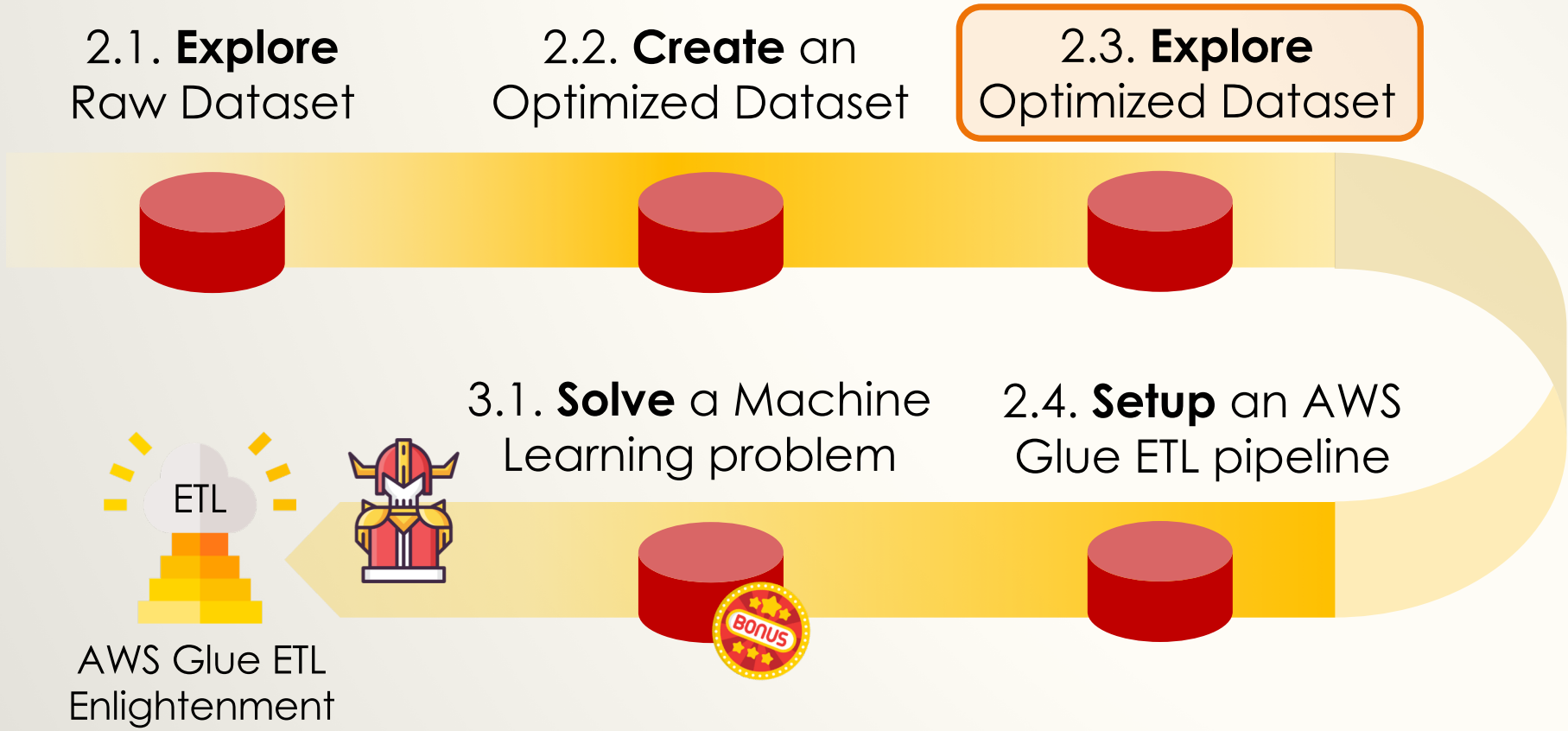
2.2.1. Create an Amazon SageMaker Notebook instance

2.2.2. Interactively author and run ETL code in Jupyter

Transformations we'll apply to Raw Data

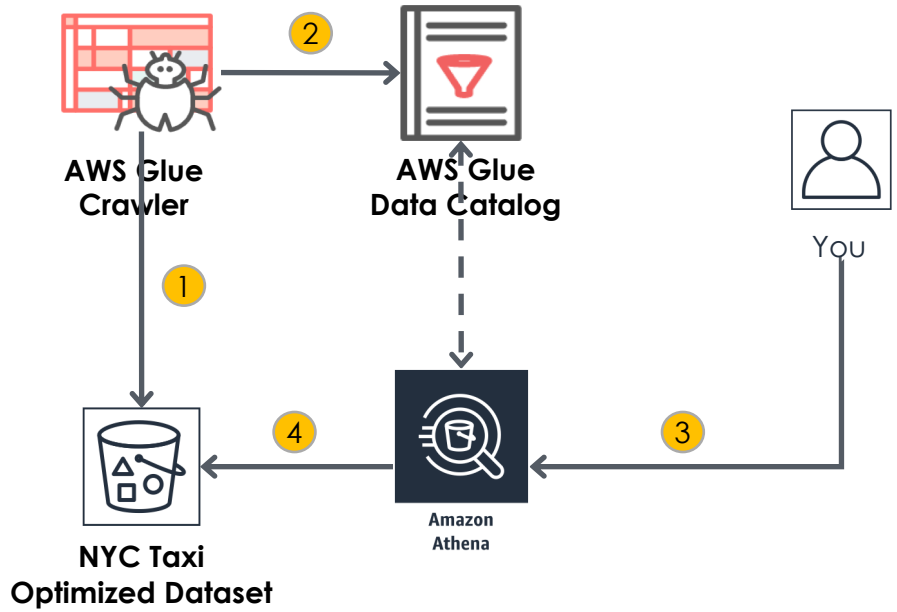
- Join NYC Trips Dataset with look-up tables (denormalization)
- Create new Timestamp columns for pick-up & drop-off
- Drop unnecessary columns
- Partition the dataset
- Convert to a columnar file format (parquet)

② Workshop Lab Guide - Practice & Learn



Concepts in Action

1. Crawler crawls **optimized dataset** in Amazon S3 Bucket
2. Crawler writes metadata into AWS Glue Data Catalog
3. You query **optimized dataset** in Athena
4. Athena uses schema definition to read data from Amazon S3 and returns results

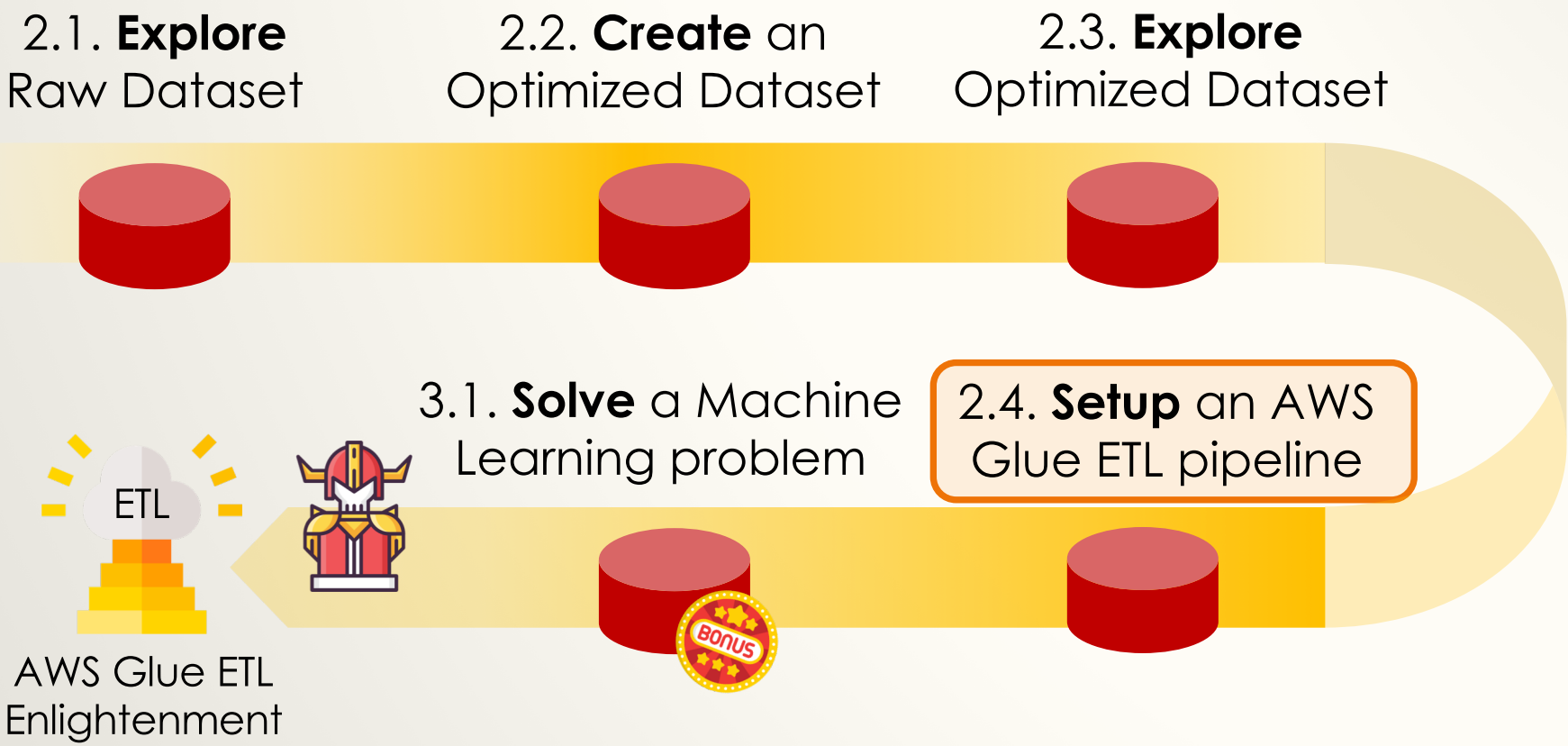


Hand-ons

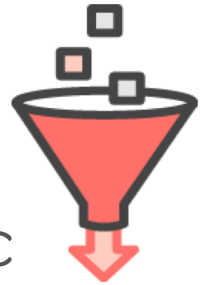
2.3.1. Catalog Optimized Data with an AWS Glue Crawler

2.3.2. Query Optimized Data with Amazon Athena

② Workshop Lab Guide - Practice & Learn



What is an **AWS Glue Job?**



An AWS Glue Job encapsulates the business logic that performs extrac transform, and load (ETL) work.

- A **core building block** in your production ETL pipeline
- Provide your PySpark ETL script or **have one automatically generated**
- Supports a **rich set of built-in AWS Glue transformations**
- Jobs can be **started, stopped, monitored**

What is an **AWS Glue trigger?**

Triggers are the 'Glue' in your AWS Glue ETL pipeline.

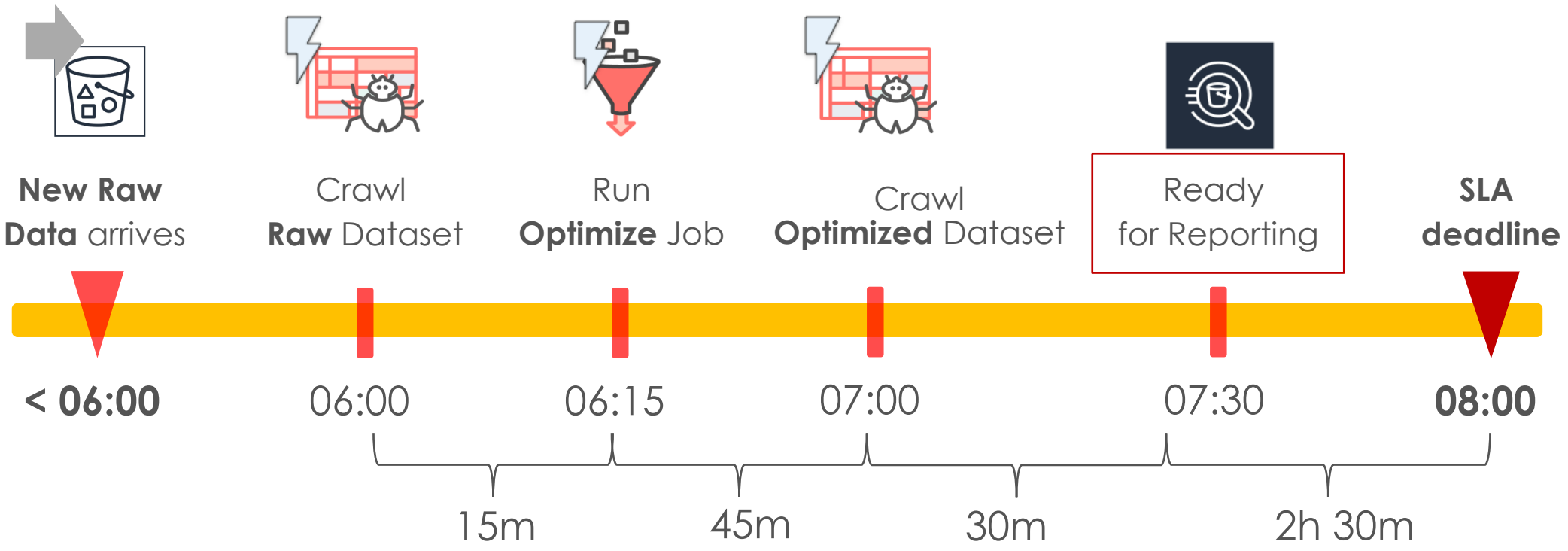
- Can be used to **chain** multiple AWS Glue Jobs in a series
- Can start **multiple Jobs at once**
- Can be **Scheduled**, **On-demand**, or based on **Job Events**
- Can **pass unique Parameters** to customize AWS Glue job runs

Three ways to set up an AWS Glue ETL pipeline

- **Schedule**-driven
- **Event**-driven
- **State Machine**-driven

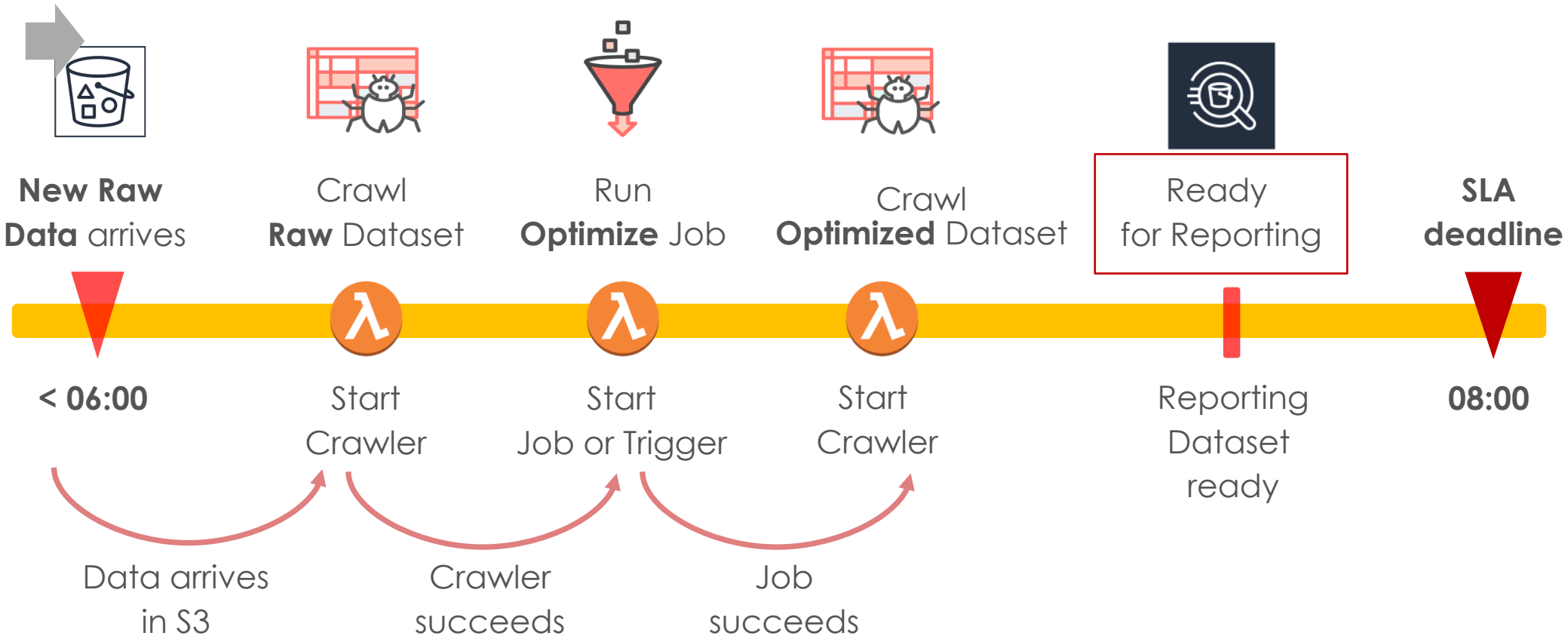
Schedule-driven AWS Glue ETL pipeline

We work our way backwards from a daily SLA deadline



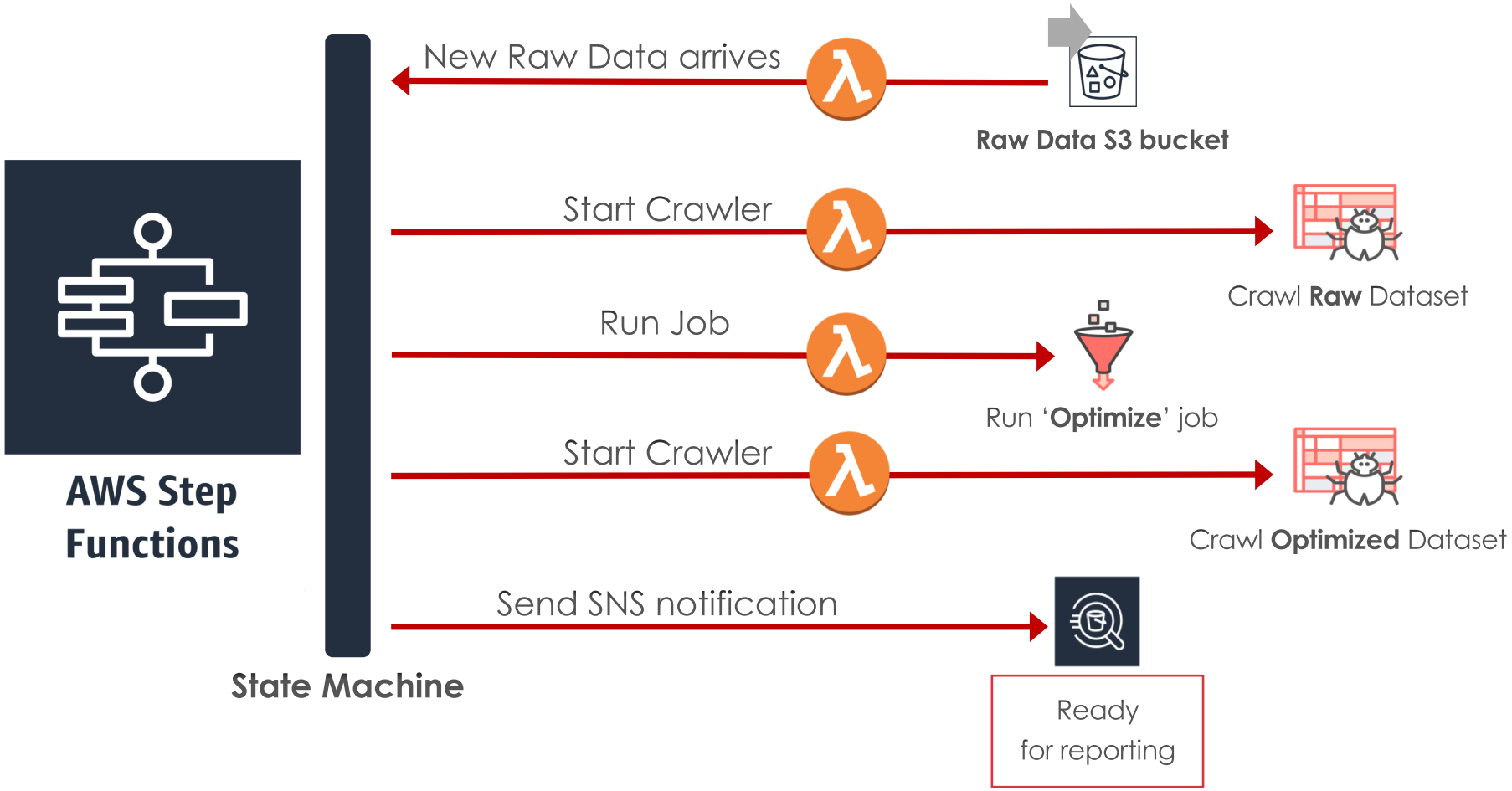
Event-driven AWS Glue ETL pipeline

Let **Amazon CloudWatch Events** and **AWS Lambda** drive the pipeline.



State Machine-driven AWS Glue ETL pipeline

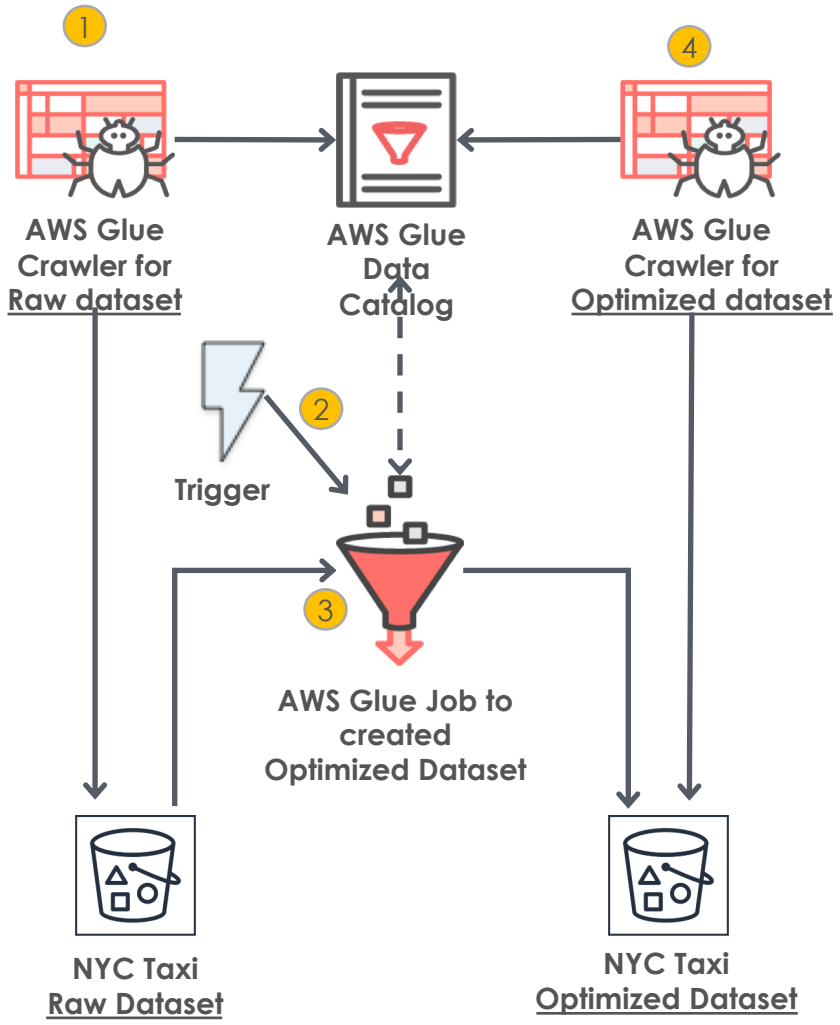
- Let **AWS Step Functions** drive the pipeline.



Concepts in Action

We'll build a **Schedule-driven** pipeline.

1. Scheduled crawler runs on **Raw Dataset** and updates AWS Glue Data Catalog
2. AWS Glue job starts based on **Trigger(s)**
3. Job reads **Raw Dataset**, applies transformations, and writes **Optimized Dataset** to your S3 bucket
4. Scheduled crawler runs on **Optimized Dataset** and updates **Data Catalog**



Hand-ons

2.4. Setup an AWS Glue ETL pipeline

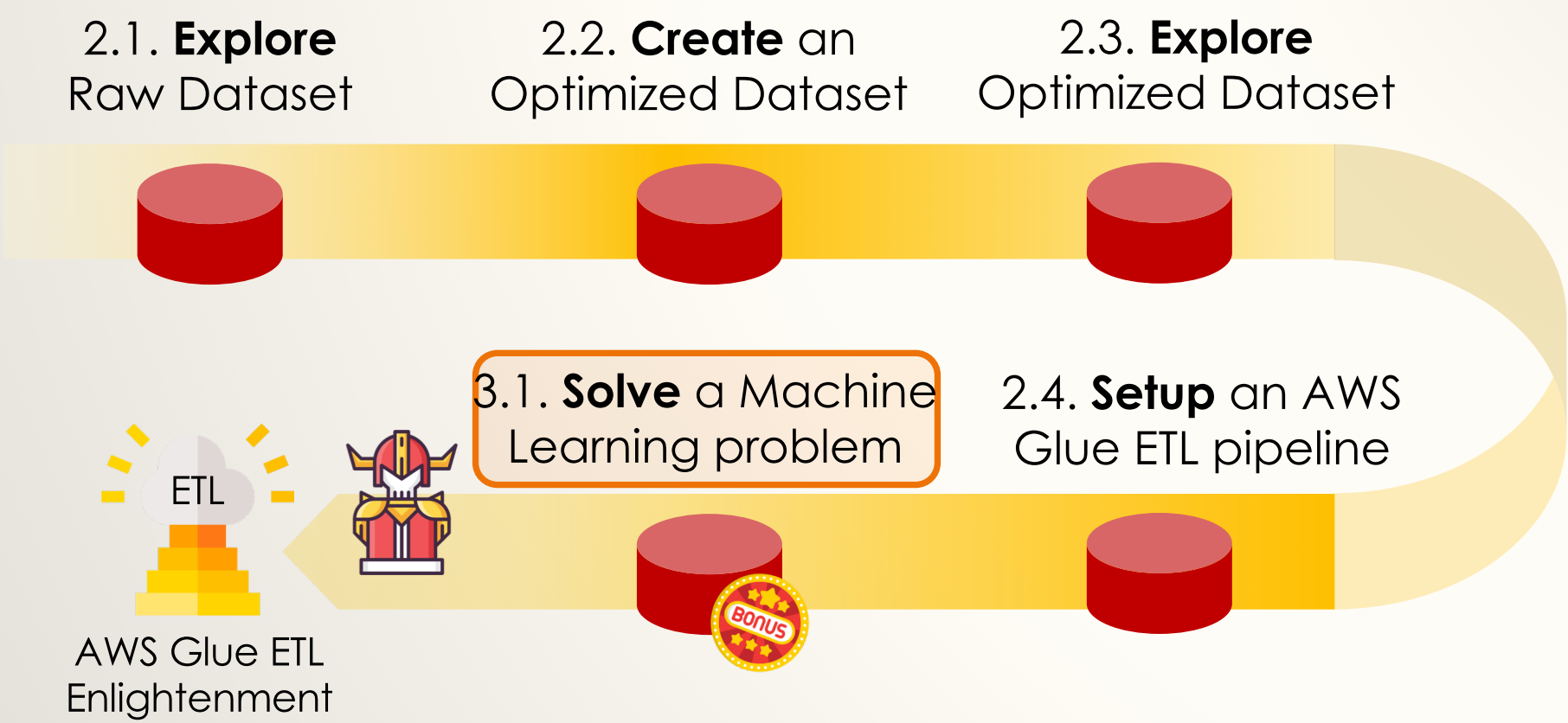
2.4.1. Schedule AWS Glue Crawlers

2.4.2. Create an AWS Glue Job

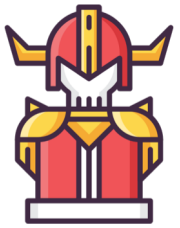
2.4.3. Create an AWS Glue trigger to run Jobs

2.4.4. AWS Glue Job Bookmarks

② Workshop Lab Guide - Practice & Learn



ML Problem Definition



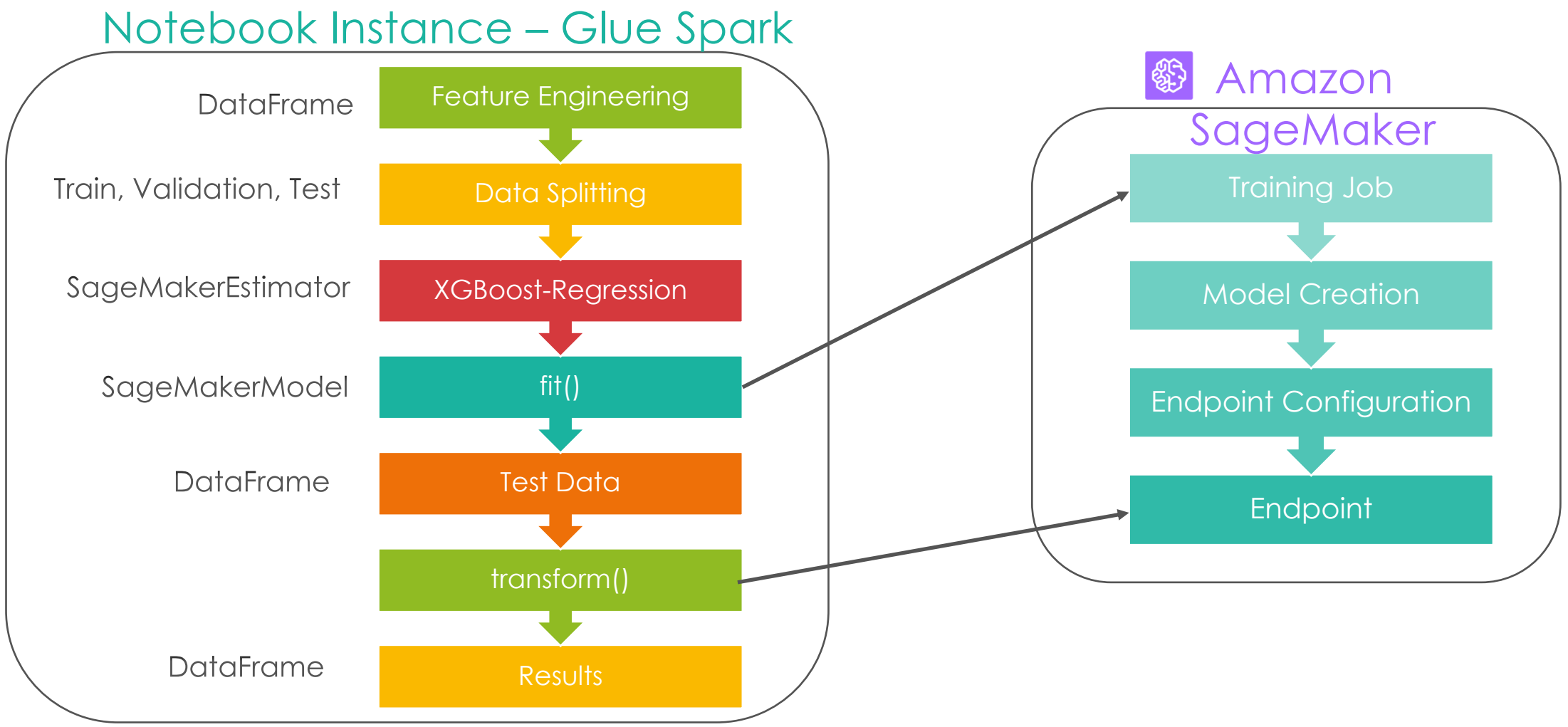
Tip given to Taxi Drivers is substantial part of their Income

They can also serve as loose metric for Service Quality that can be useful for taxi companies

What are influences for higher tips (e.g., Taxi starting and ending in Richer part of cities?)

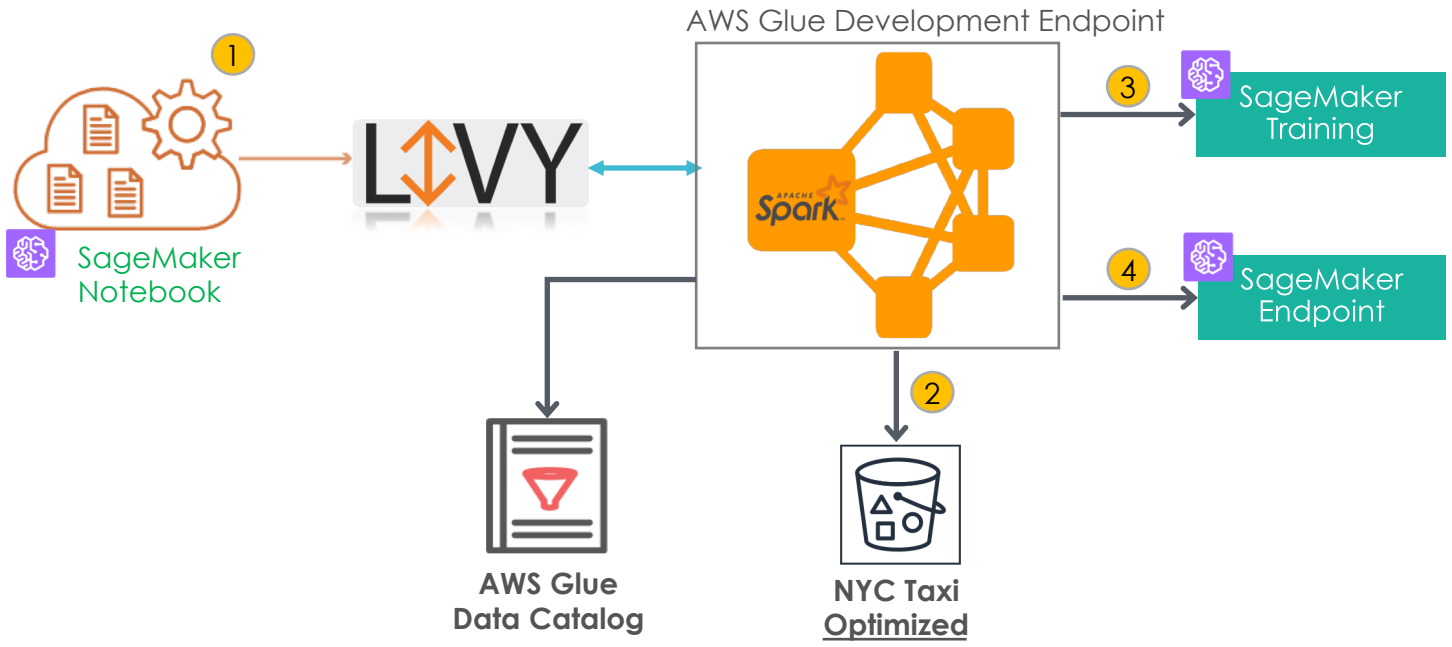
We present analysis of factors affecting Tips and use these factors to predict Tips

ML Architecture



Concepts in Action

1. Launch Notebook
2. Read Optimized Dataset
3. Train the ML Model
4. Host the trained model and perform Prediction



Hand-ons

3.1. Interactively develop a Machine Learning model in Jupyter

Run ML Notebook interactively

- Initialize variables and import spark libraries
- Retrieve Optimized NYC Taxi trips Dataset
- Observe features against target label (tip_amount)
- Perform feature engineering
- Split feature engineered dataset into Train and Test
- Launch Spark **XGBoostEstimator** (Observe hyperparameters)
- Perform Prediction and observe accuracy

You made it!