AWS re:Invent

ANT313

Serverless Data Prep with AWS Glue

Moataz Anany
Solutions Architect
AWS

Nitin Wagh
Solutions Architect
AWS

AWS re:Invent

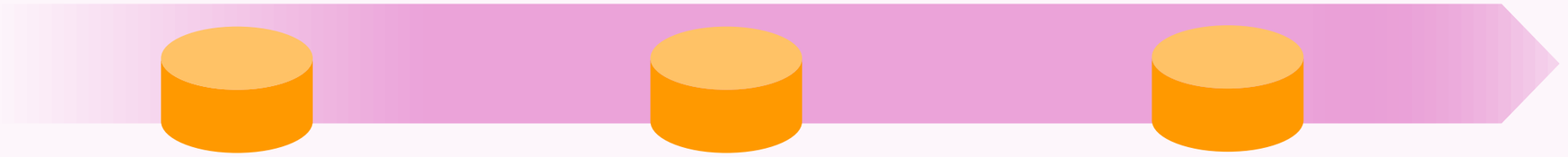
aws

Workshop map

Part I – Get ready!

1. **Learn** workshop goals
2. **Get** your AWS account ready
3. **Quick** intro to AWS Glue

You



Part II

Workshop map

Part II – Practice and learn

1. **Explore** raw dataset

2. **Create** an optimized dataset

3. **Explore** optimized dataset

4. **Set up** an AWS Glue ETL pipeline

5. **Solve** a machine learning problem

AWS Glue ETL enlightenment

Workshop map

Part I – Get ready!

1. **Learn** workshop goals

2. **Get** your AWS account ready

3. **Quick** intro to AWS Glue

Part II

You are a **data engineer at AnyCompany***

Create a dataset for
reporting and visualization

Cleanse

Transform

Optimize for reporting queries

Help solve a
machine learning problem

Data scientists at AnyCompany need to understand **passenger tipping behavior**

* a completely made-up ride-hailing startup

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

http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

Your dataset: NYC taxi trips

Original raw dataset

Green and yellow taxi + FHV

Years 2009 to 2018

~1.6Bn rows

215 files

253GB total

Simplified raw dataset

Only yellow taxi + few look-ups

Jan to March 2017

~2M rows

3 files

2.5GB+ uncompressed

Ready in a publicly-accessible Amazon Simple Storage Service (Amazon S3) bucket

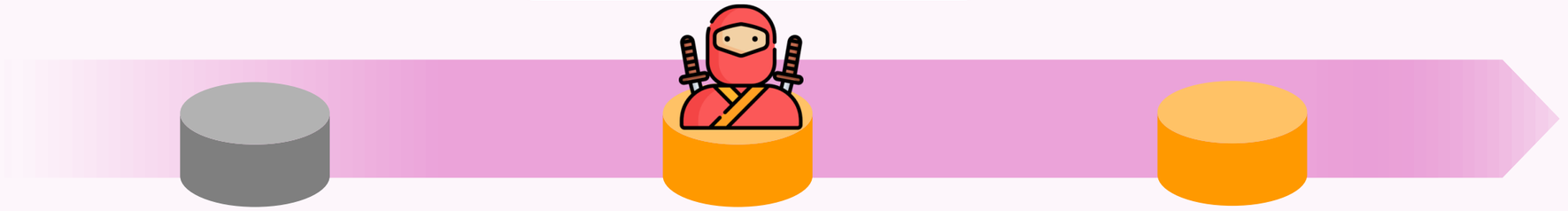
Workshop map

Part I – Get ready!

1. **Learn** workshop goals

2. **Get** your AWS account ready

3. **Quick** intro to AWS Glue



Part II

Navigate to the ANT313 workshop website

Use Firefox or Chrome. Keep the website open in a separate tab

https://bit.ly/ant313-workshop

Hands-on

I.2: Get your AWS account ready

Key resources AWS CloudFormation will create

1. A new data lake S3 bucket
2. Necessary AWS Identity and Access Management (IAM) policies and roles for AWS Glue, Amazon Athena, and Amazon SageMaker
3. An AWS Glue development endpoint
4. A number of named queries in Athena

Finally, the **NYC taxi trips raw dataset is copied** into your S3 bucket

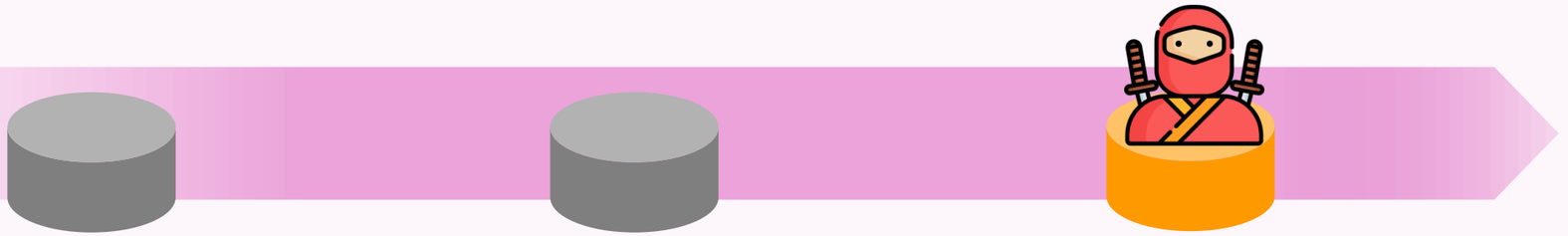
Workshop map

Part I – Get ready!

1. **Learn** workshop goals

2. **Get** your AWS account ready

3. **Quick** intro to AWS Glue



Part II

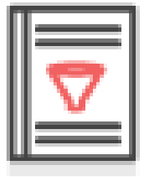
AWS Glue automates the undifferentiated heavy lifting of ETL

Discover Automatically discover and categorize your data making it immediately searchable and queryable across data sources

Develop Generate code to clean, enrich, and reliably move data between various data sources

Deploy Run your jobs on a serverless, fully managed, scale-out environment. No compute resources to manage

AWS Glue components



Data Catalog

- Hive Metastore compatible with enhanced functionality
- Crawlers automatically extract metadata and create tables
- Integrated with Athena, Amazon Redshift Spectrum

Job authoring

- Auto-generates ETL code
- Build on open frameworks – Python and Spark
- Developer-centric – editing, debugging, sharing

Job execution

- Run jobs on a serverless Spark platform
- Provides flexible scheduling
- Handles dependency resolution, monitoring, and alerting

Workshop map

Part II – Practice and learn

1. **Explore** raw dataset

2. **Create** an optimized dataset

3. **Explore** optimized dataset

4. **Set up** an AWS Glue ETL pipeline

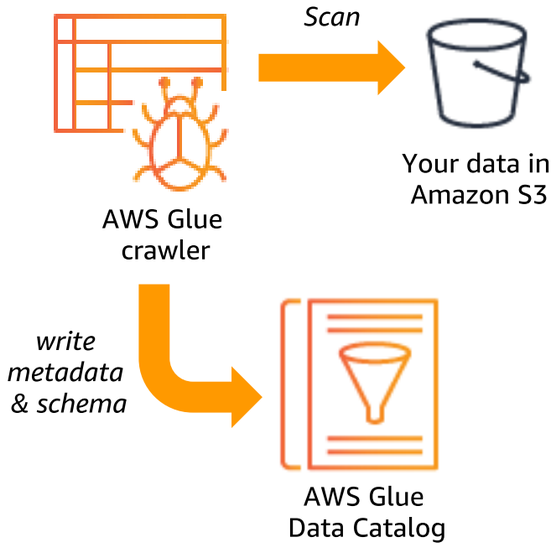
5. **Solve** a machine learning problem

AWS Glue ETL enlightenment

AWS Glue Crawlers and the AWS Glue Data Catalog

A crawler . . .

- **Samples and classifies** your data
- Extracts metadata
- **Infers schema** and partitioning format
- **Creates tables** in your account's **AWS Glue Data Catalog**



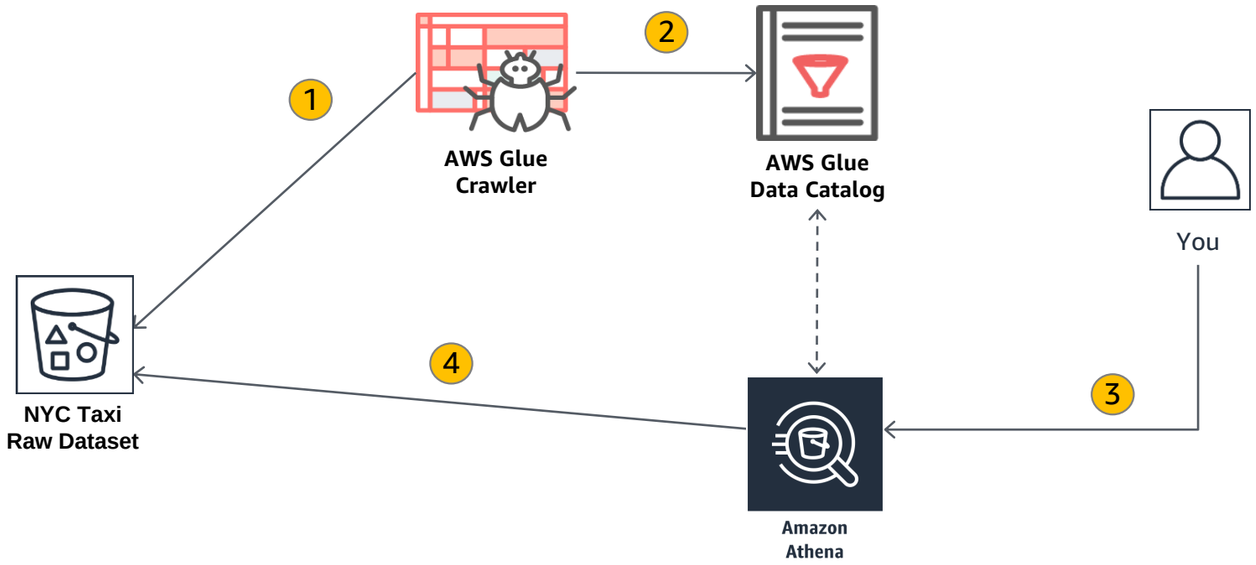
Why query with Amazon Athena?

- **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

Amazon Athena

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



Hands-on

II.1.1: Catalog raw data with an AWS Glue crawler

Hands-on

II.1.2: Explore table schema and metadata in AWS Glue Data Catalog

Hands-on

II.1.3: Query raw data with Amazon Athena

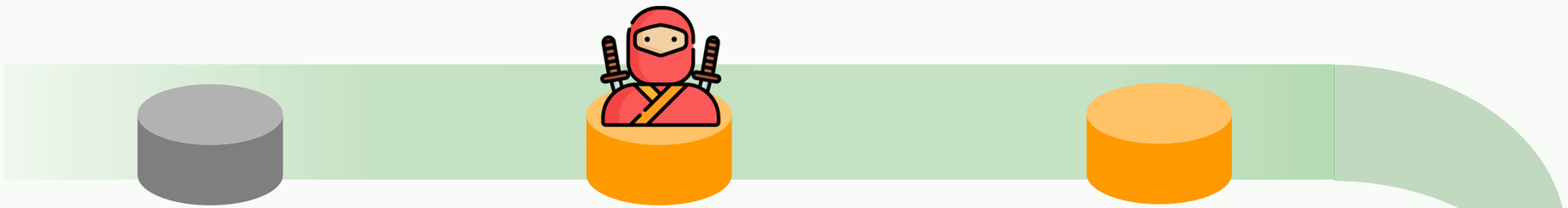
Workshop map

Part II – Practice and learn

1. **Explore** raw dataset

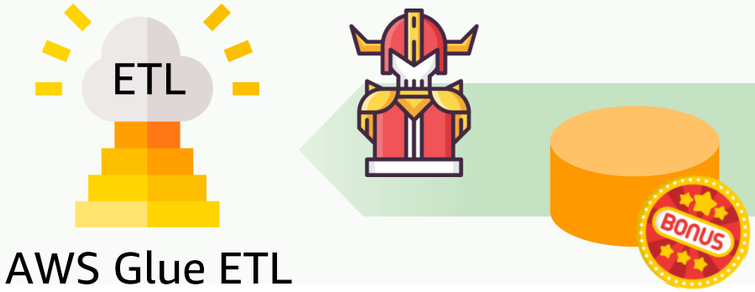
2. **Create** an optimized dataset

3. **Explore** optimized dataset



4. **Set up** an AWS Glue ETL pipeline

5. **Solve** a machine learning problem



AWS Glue ETL enlightenment

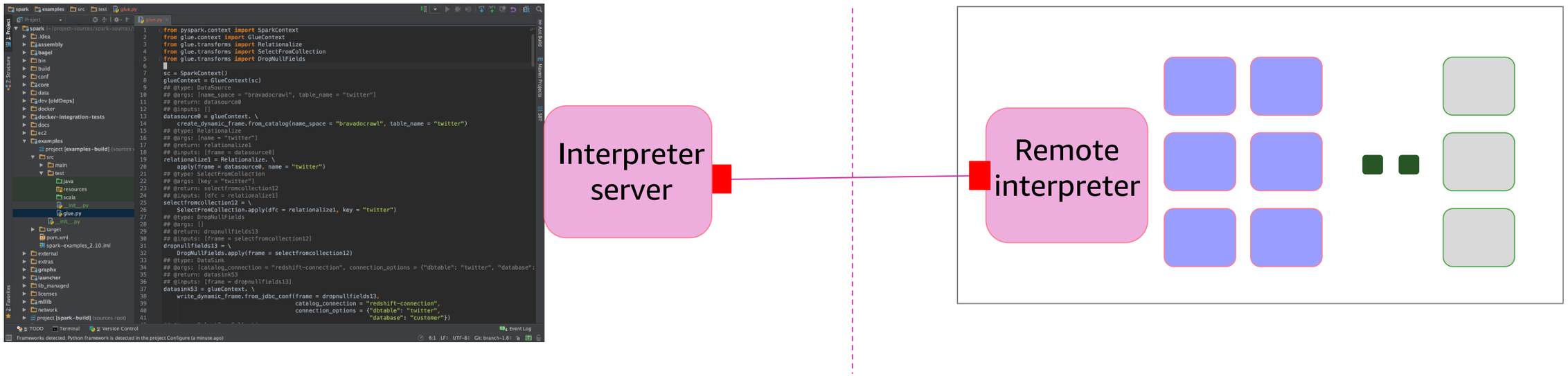
Interactive ETL development with **AWS Glue** and **Amazon SageMaker**

What is Amazon SageMaker?

- 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?



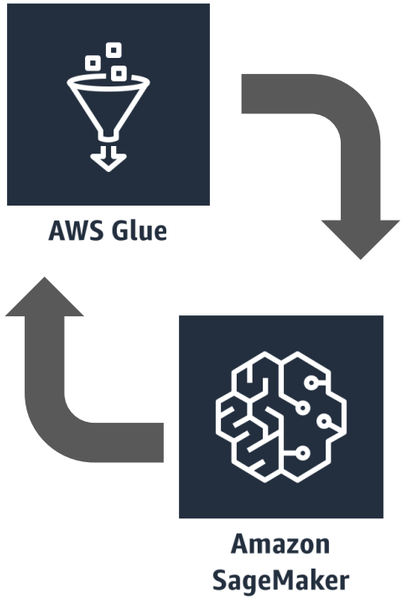
Connect your IDE to an AWS Glue development endpoint

Environment to **interactively develop,** debug, and test ETL code

Interactive ETL development with **AWS Glue** and **Amazon SageMaker**

AWS Glue enables you to

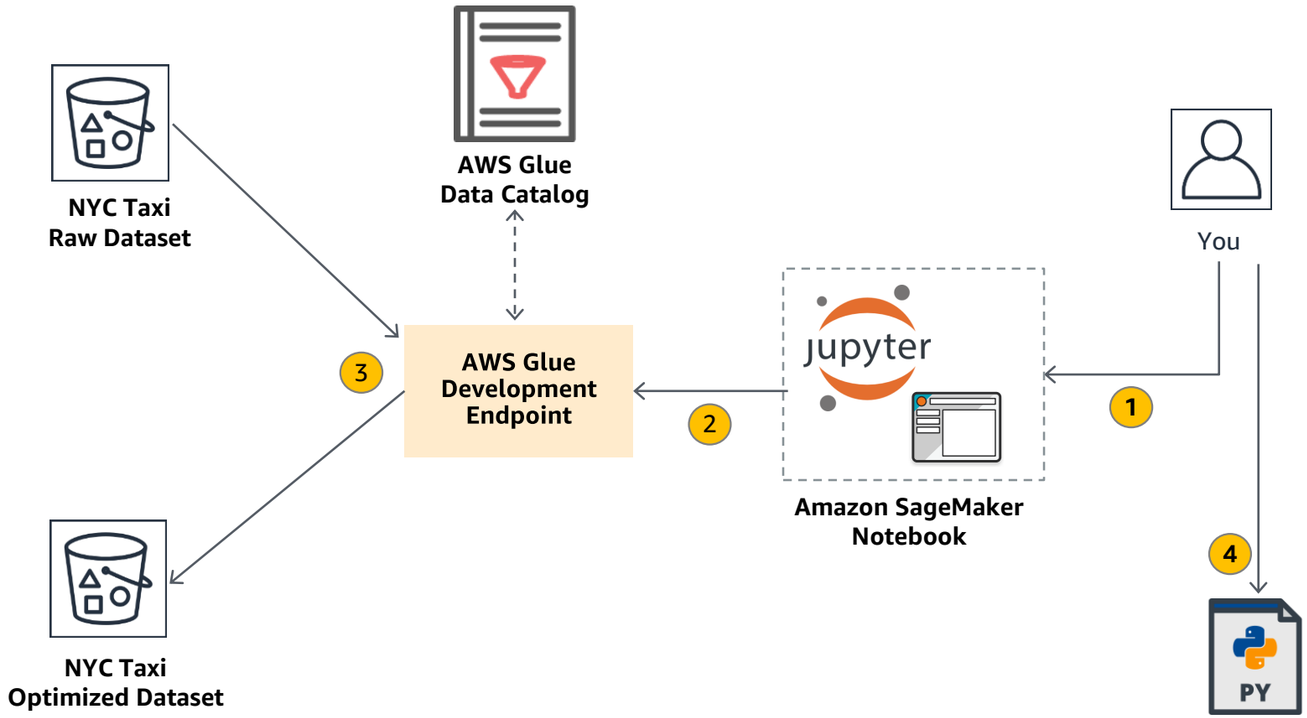
- **Create** an **Amazon SageMaker notebook** environment
- **Connect** it to an **AWS Glue dev endpoint**
- **And, finally, access** your **Jupyter** notebook



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



Hands-on

II.2.1: Create an Amazon SageMaker notebook instance

Hands-on

II.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)

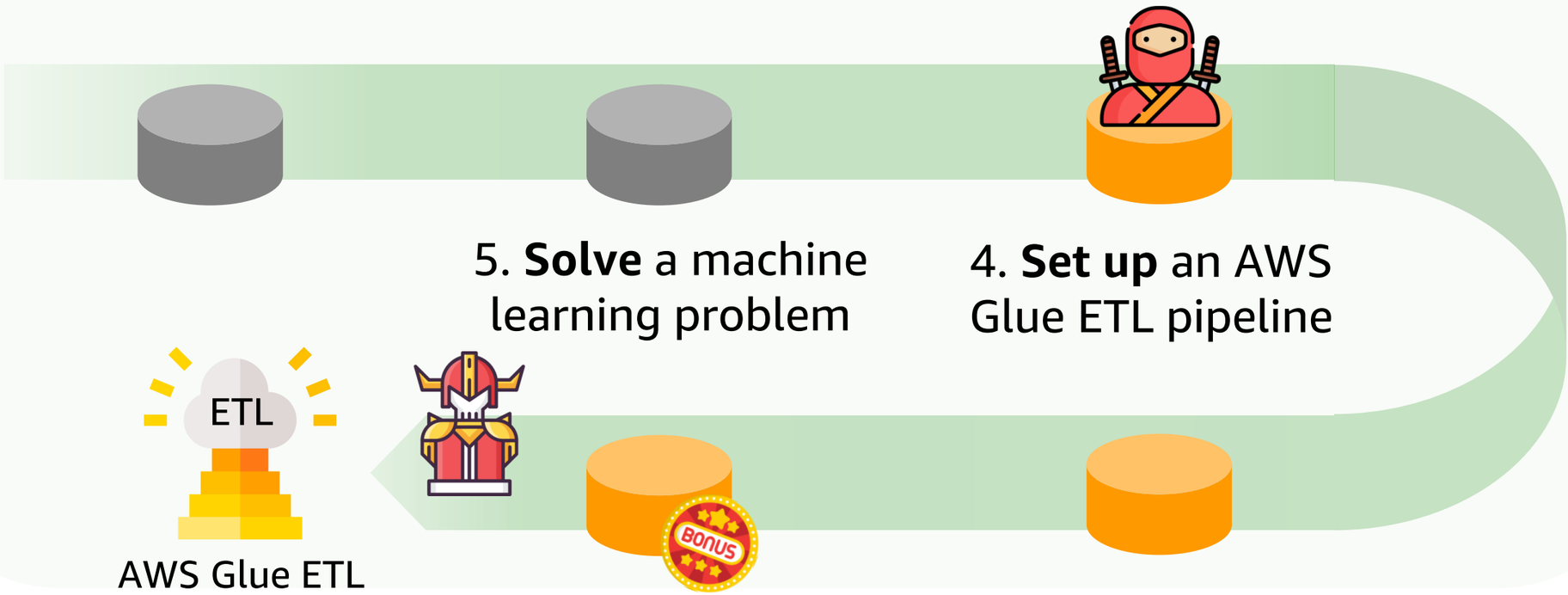
Workshop map

Part II – Practice and learn

1. **Explore** raw dataset

2. **Create** an optimized dataset

3. **Explore** optimized dataset



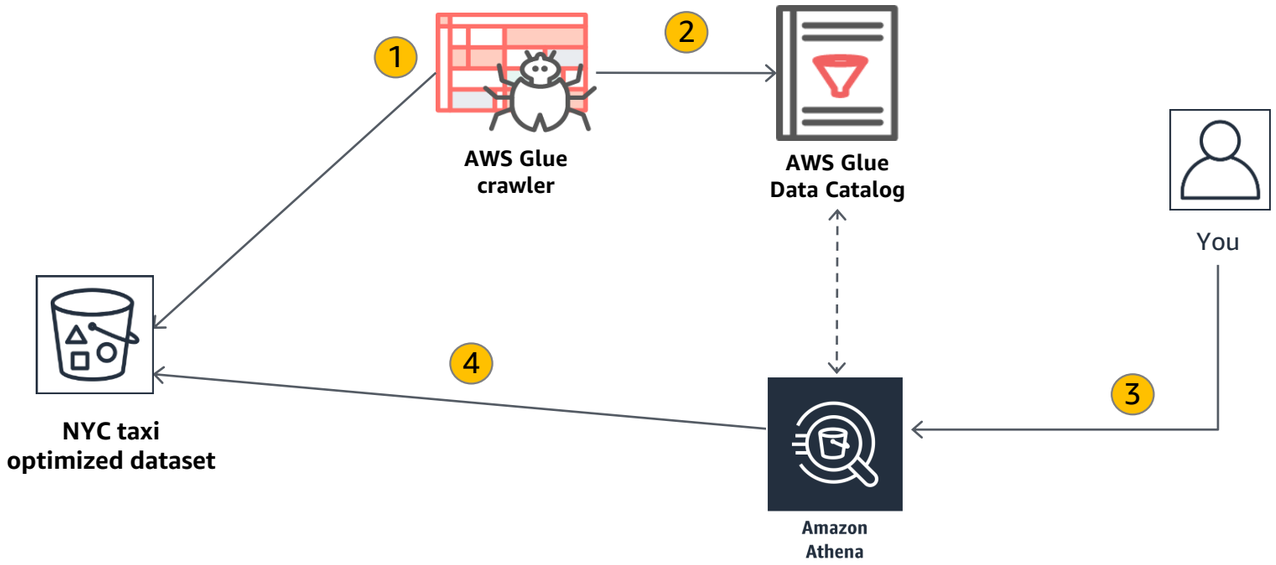
4. **Set up** an AWS Glue ETL pipeline

5. **Solve** a machine learning problem

AWS Glue ETL enlightenment

Concepts in action

1. Crawler crawls **optimized dataset** in 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



Hands-on

II.3.1: Catalog optimized data with an AWS Glue crawler

Hands-on

II.3.2: Query optimized data with Amazon Athena

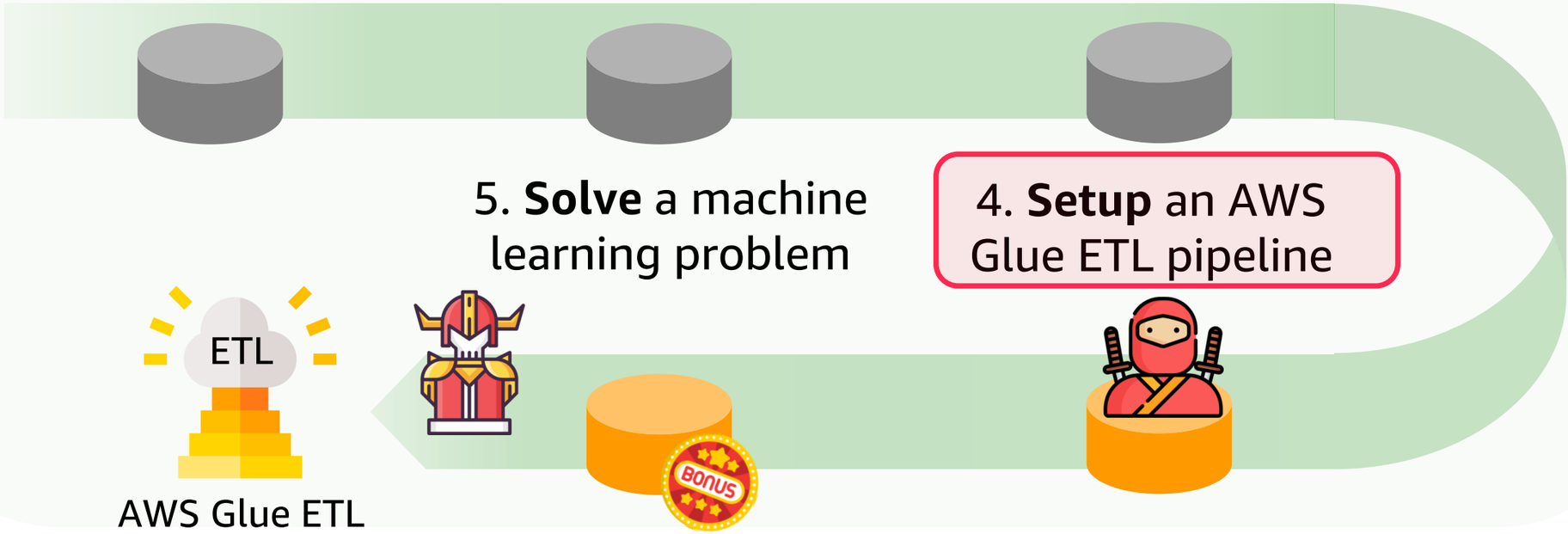
Workshop map

Part II – Practice and learn

1. **Explore** raw dataset

2. **Create** an optimized dataset

3. **Explore** optimized dataset



5. **Solve** a machine learning problem

4. **Setup** an AWS Glue ETL pipeline

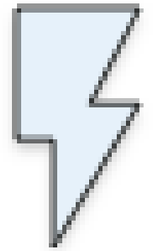
AWS Glue ETL enlightenment

What is an AWS Glue job?

An AWS Glue job encapsulates the business logic that performs extract, transform, and load (ETL) work

- A **core building block** in your production ETL pipeline
- Provide your PySpark ETL script, or **have one auto-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

Triggers . . .

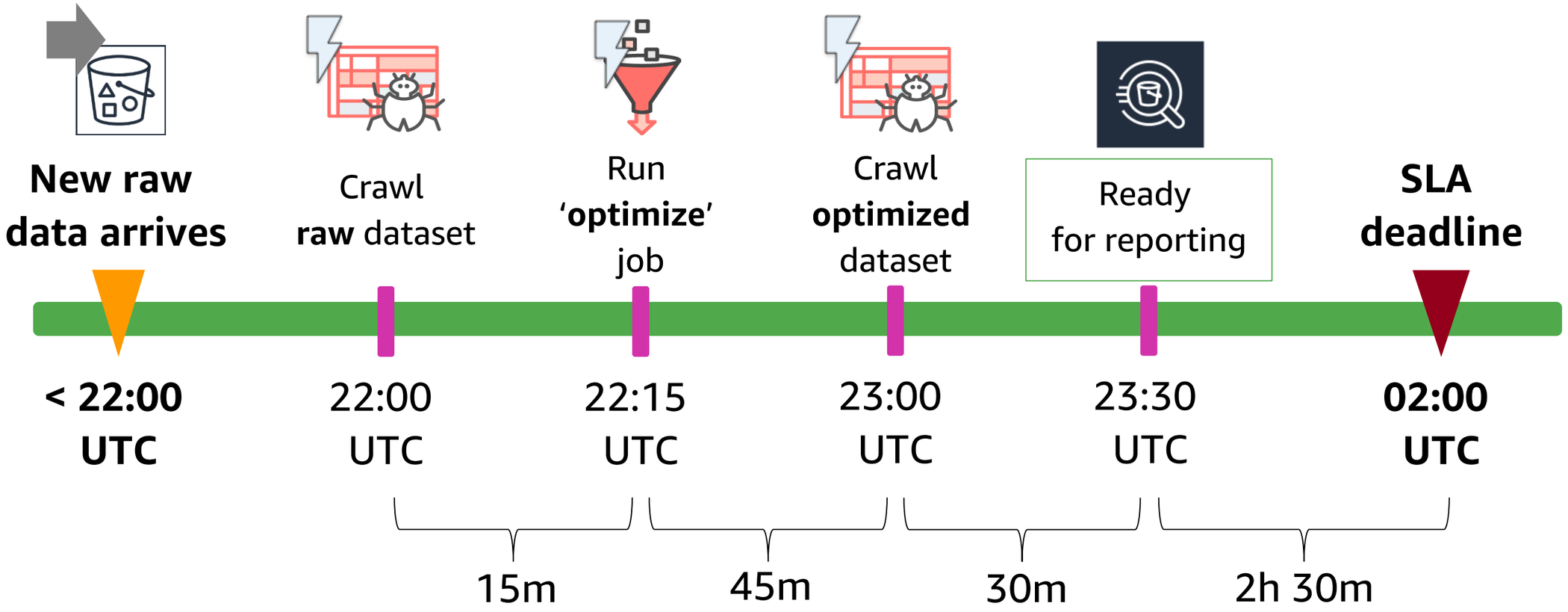
- Can be used to **chain** multiple AWS Glue jobs in a series
- Can start **multiple jobs at once**
- Can be **scheduled**, **on-demand,** or based on **job events**
- Can **pass unique parameters** to customize AWS Glue job runs

Three ways to set up an AWS Glue ETL pipeline

- **Schedule**-driven
- **Event**-driven
- **State machine**–driven

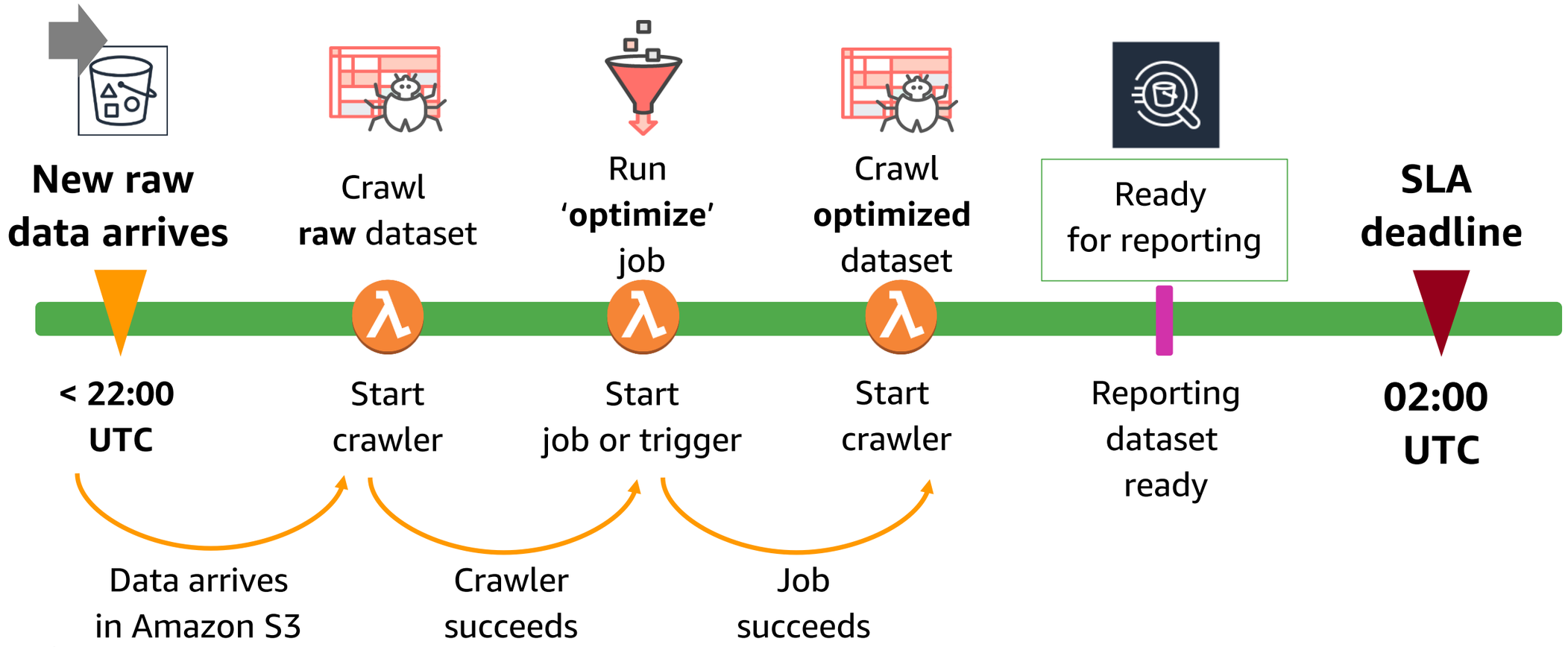
Schedule-driven AWS Glue ETL pipeline

We work our way backwards from a daily SLA deadline



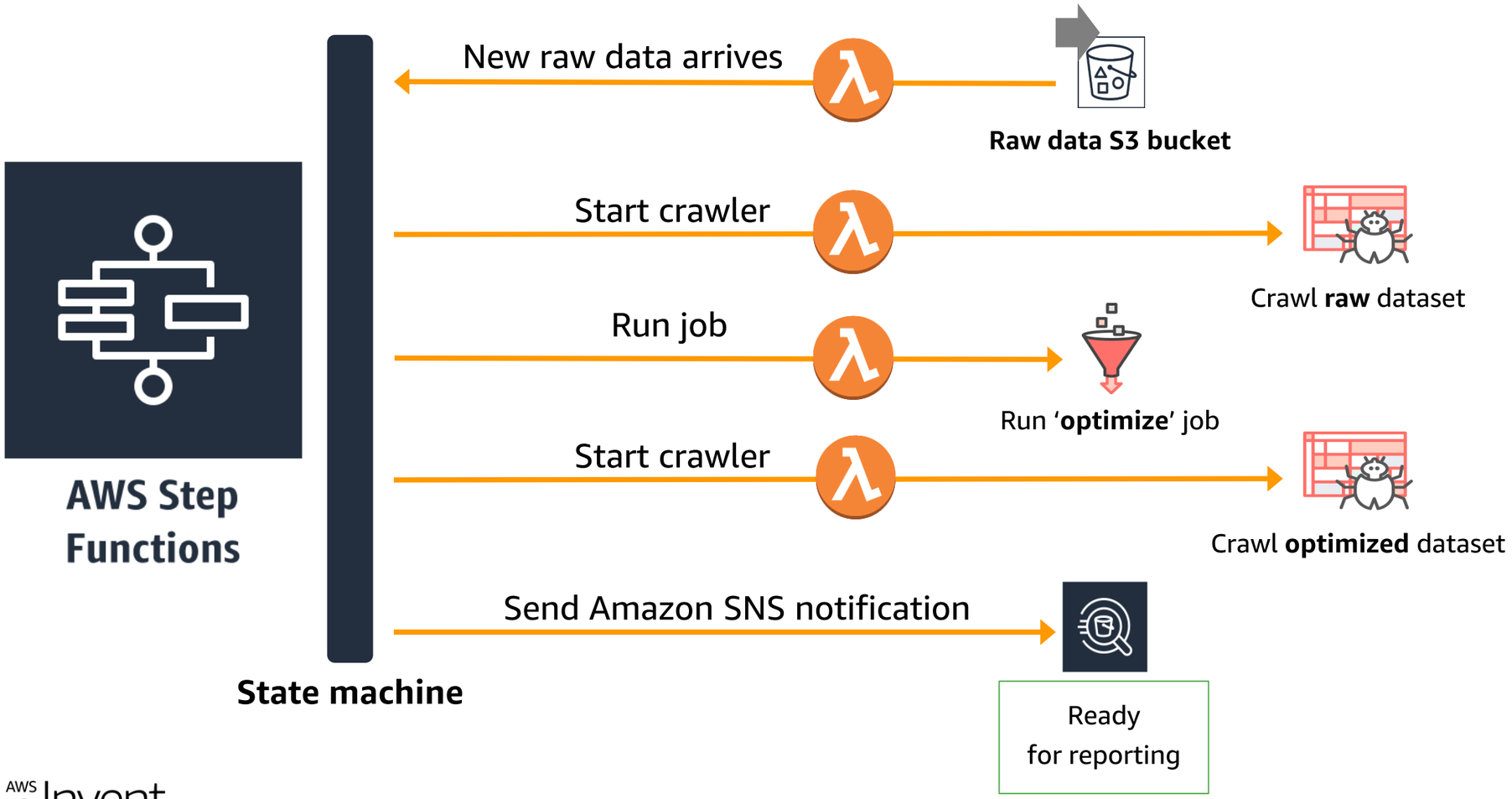
Event-driven AWS Glue ETL pipeline

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



State machine–driven AWS Glue ETL pipeline

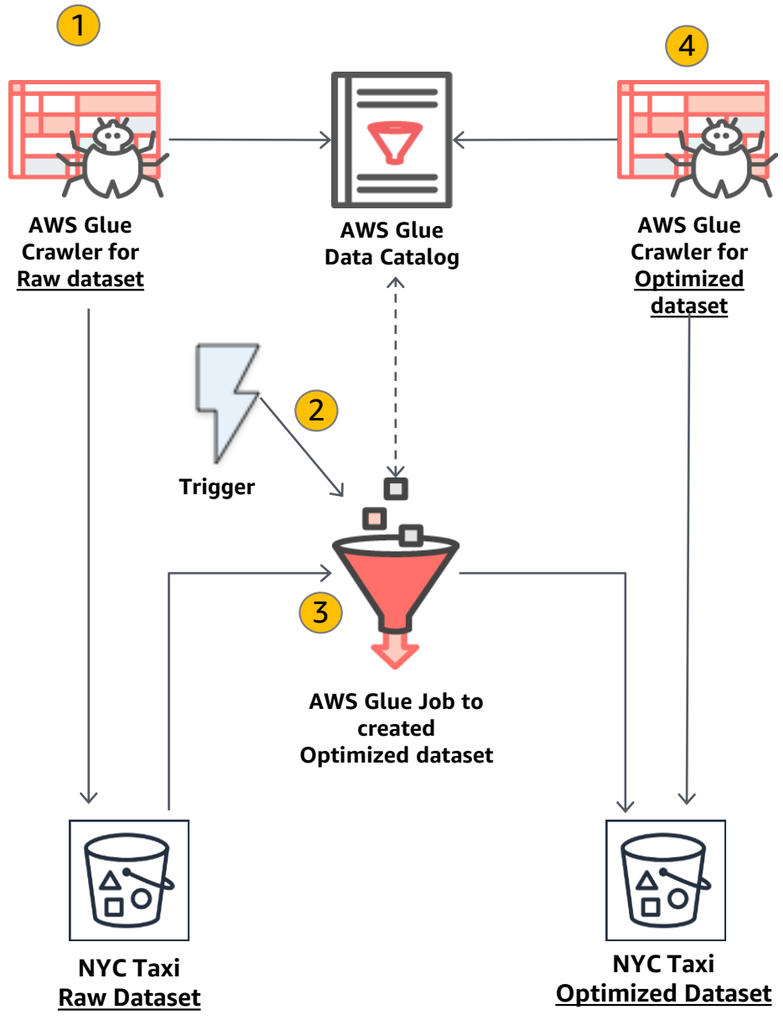
Let **AWS Step Functions** drive the pipeline



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



Hands-on

II.4.1: Schedule AWS Glue crawlers

Hands-on

II.4.2: Create an AWS Glue job

Hands-on

II.4.3: Create an AWS Glue trigger to run jobs

Hands-on

Advanced: Try out AWS Glue job bookmarks

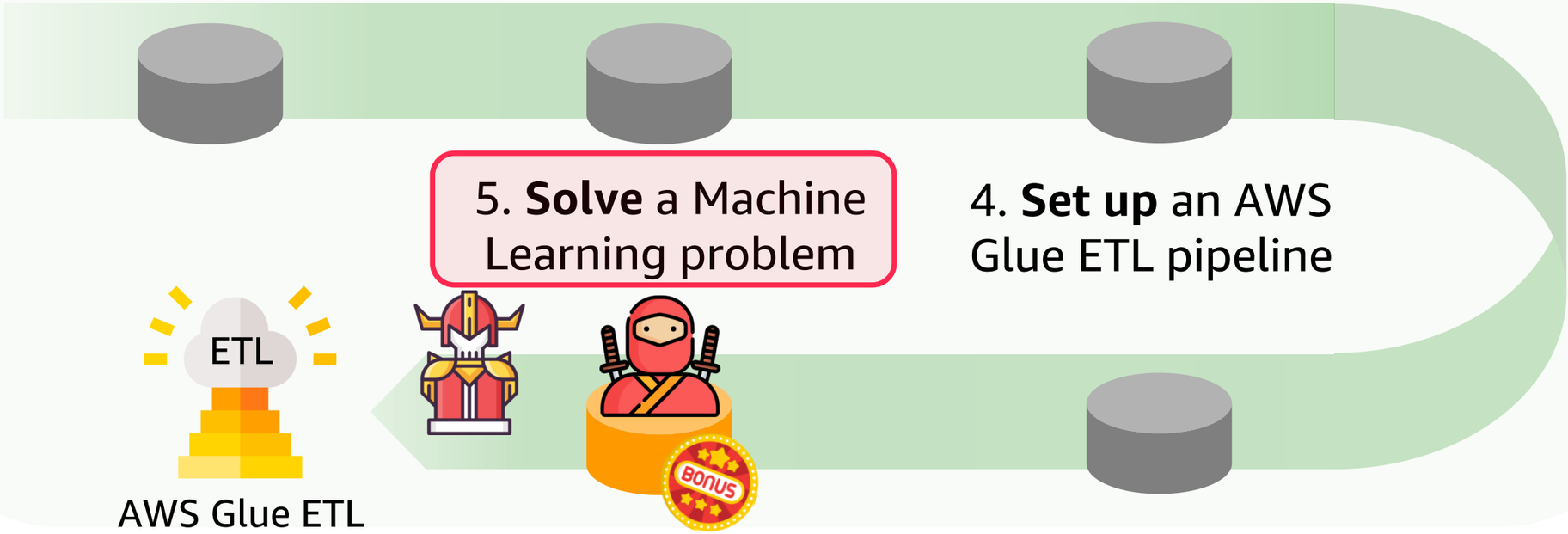
Workshop map

Part II – Practice and learn

1. **Explore** raw dataset

2. **Create** an optimized dataset

3. **Explore** optimized dataset

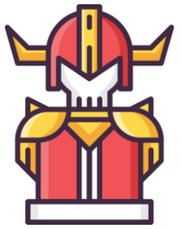


4. **Set up** an AWS Glue ETL pipeline

5. **Solve** a Machine Learning problem

AWS Glue ETL enlightenment

ML problem definition



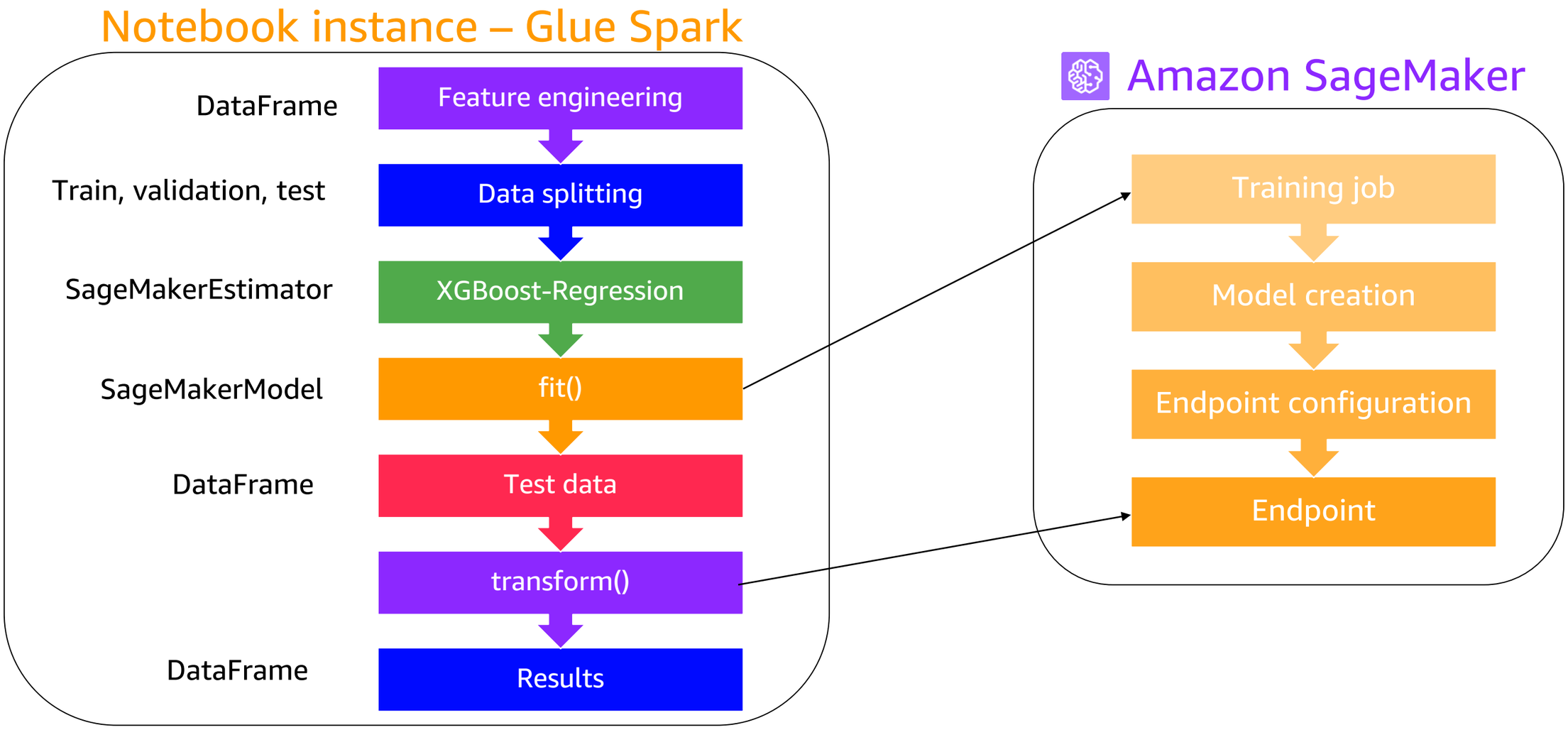
Tip given to taxi drivers is substantial part of their income

They can also serve as loose metric for service quality that can be useful for taxi companies

What are influences for higher tips (for example, taxi starting and ending in richer part of cities?)

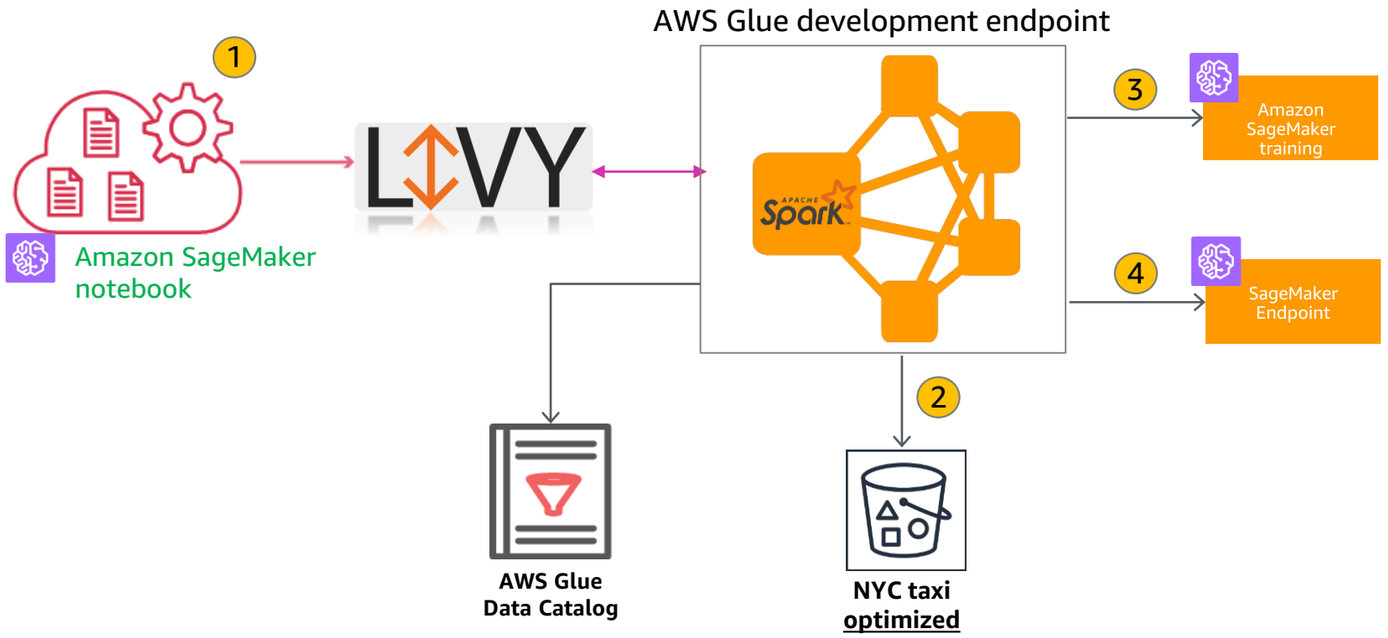
We present analysis of factors affecting tips and use these factors to predict tips

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



Hands-on

II.5.1: 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 data set into Train and Test
- Launch Spark **XGBoostEstimator** (Observe hyperparameters)
- Perform Prediction and observe accuracy

You made it!

Before you go . . .

- Visit the “**Account clean-up**” section for instructions on cleaning-up your AWS account

- Tell us how we did: Please fill out the survey

Thank you

Thank you!

Moataz Anany
Solutions Architect
AWS

Nitin Wagh
Solutions Architect
AWS

AWS re:Invent

aws

Please complete the session survey in the mobile app.

AWS re:Invent

aws