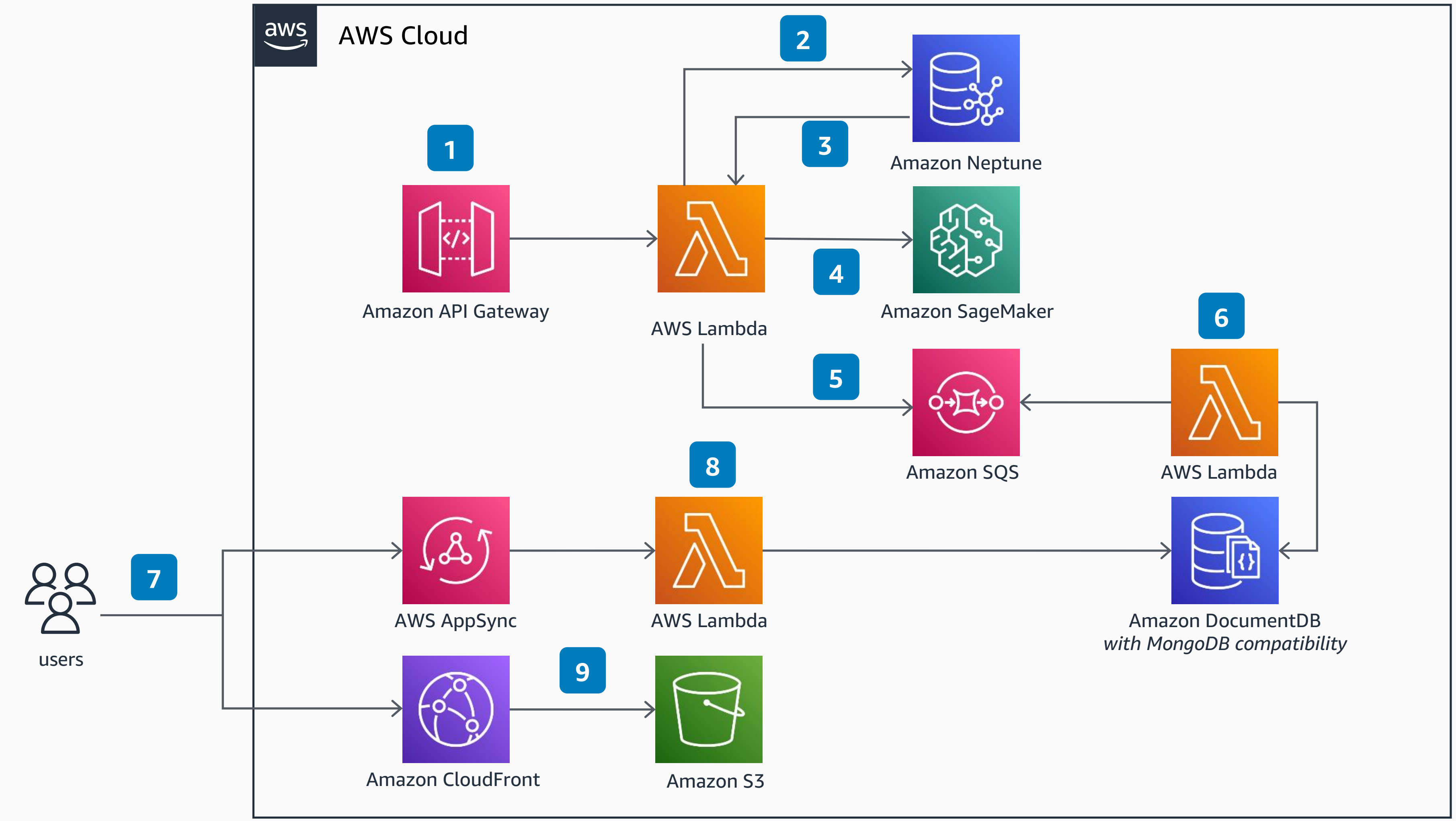
Guidance for Near Real-Time Fraud Detection with Graph Neural Network on AWS

A full managed GNN-based near real-time fraud detection solution

This is a blueprint architecture for near real-time fraud detection using graph databases Amazon Neptune, Amazon SageMaker and Deep Graph Library (DGL) to construct a heterogeneous graph from tabular data and train a Graph Neural Network (GNN) model to detect fraudulent transactions in the IEEE-CIS fraud detection dataset.



1 Use **Amazon API Gateway** to host HTTP APIs for near real-time fraud detection services.

2 Use **AWS Lambda** functions as an HTTP API backend. The functions process the new transactions as graph data then store them in a graph database such as **Amazon Neptune**.

3 Query the sub-graph of the requested transactions from **Amazon Neptune**.

4 Use an **Amazon SageMaker** endpoint to predict the fraudulent possibility of transactions with pre-trained GNN models.

5 Send the predicated results to **Amazon Simple Queue Service** (Amazon SQS) to be consumed by business analysis systems.

6 Use **AWS Lambda** functions to poll the predicated results from **Amazon SQS**, then store them in **Amazon DocumentDB**.

7 Business analysts access the business dashboard, which uses **Amazon CloudFront** and **Amazon Simple Storage Service** (Amazon S3) to host a static website, and **AWS AppSync** and **AWS Lambda** as a backend.

8 Use **AWS Lambda** functions as an **AWS AppSync** resolver to fetch the data from **Amazon DocumentDB**.

9 **Amazon CloudFront** uses origin access identity (OAI) to securely access the static web files on **Amazon S3**.

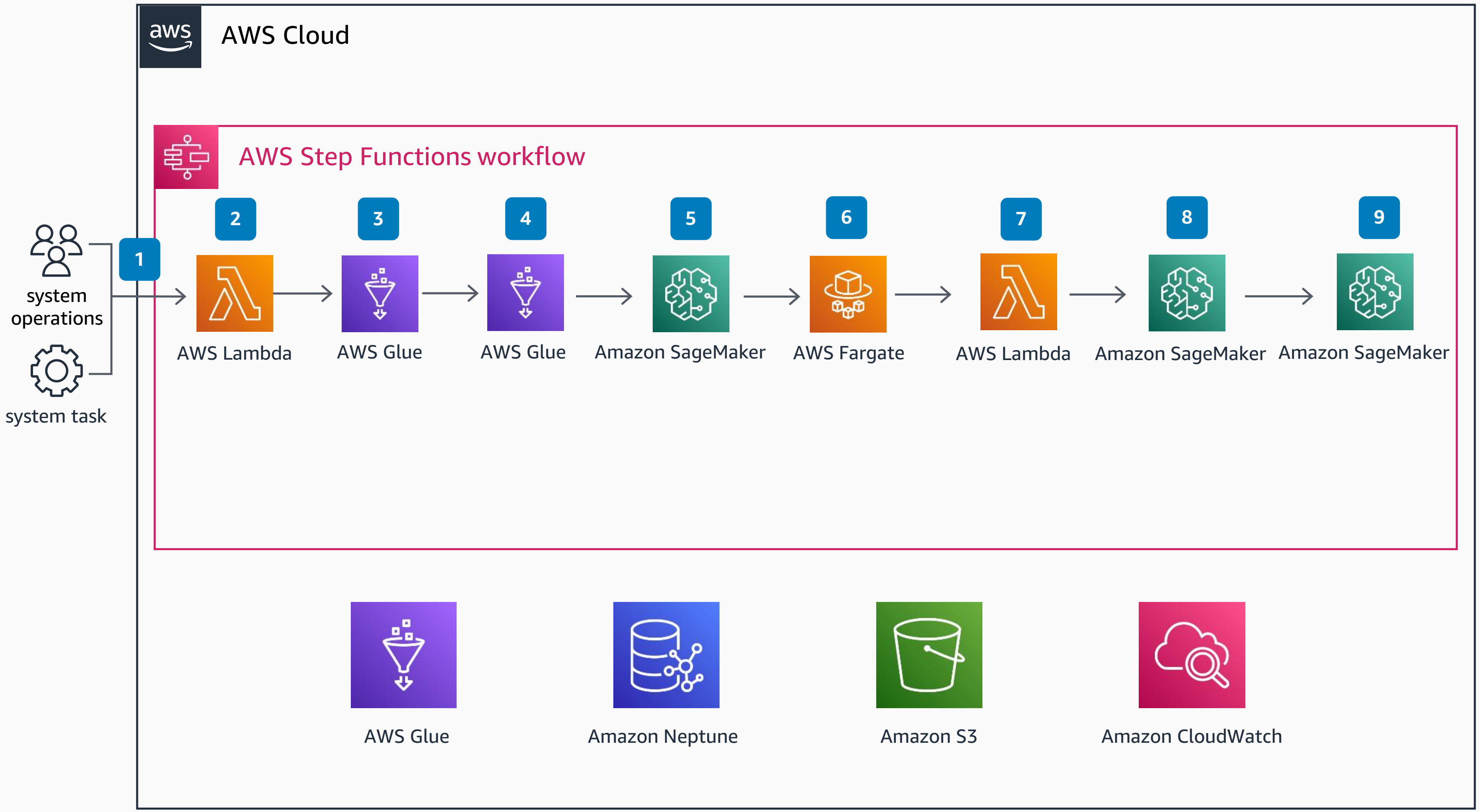
Reviewed for technical accuracy December 15, 2022

AWS Reference Architecture

Guidance for Near Real-time Fraud Detection with Graph Neural Network on AWS

A fully-managed GNN-based near real-time fraud detection solution

This architecture is a blueprint for near real-time fraud detection using graph database services Amazon Neptune, Amazon SageMaker, and DGL to construct a heterogeneous graph. The tabular data is used to train a GNN model to detect fraudulent transactions in the IEEE-CIS Fraud detection dataset.



1 System operations or a periodic system task initiates the model training workflow.

2 Use **Lambda** function to ingest the raw dataset to Amazon S3.

3 Use **AWS Glue** crawler to crawl the raw dataset to populate the Data Catalog.

4 Use **AWS Glue** extract, transform, load (ETL) job to transform the tabular dataset to a heterogeneous graph dataset, then save it to **Amazon S3**.

5 Use the **SageMaker** training job to train the Graph Neural Network (GNN)-based fraud detection model with Deep Graph Library (DGL).

6 Use **AWS Fargate** with **Amazon Elastic Container Service** (Amazon ECS) to load the graph dataset from **Amazon S3** into fully-managed graph database service, **Neptune**.

7 Use **Lambda** to package the GNN model and custom code as the model in **SageMaker**.

8 Create an endpoint configuration of **SageMaker**.

9 Create or update an endpoint using the endpoint configuration in **SageMaker**.

Reviewed for technical accuracy December 15, 2022

AWS Reference Architecture