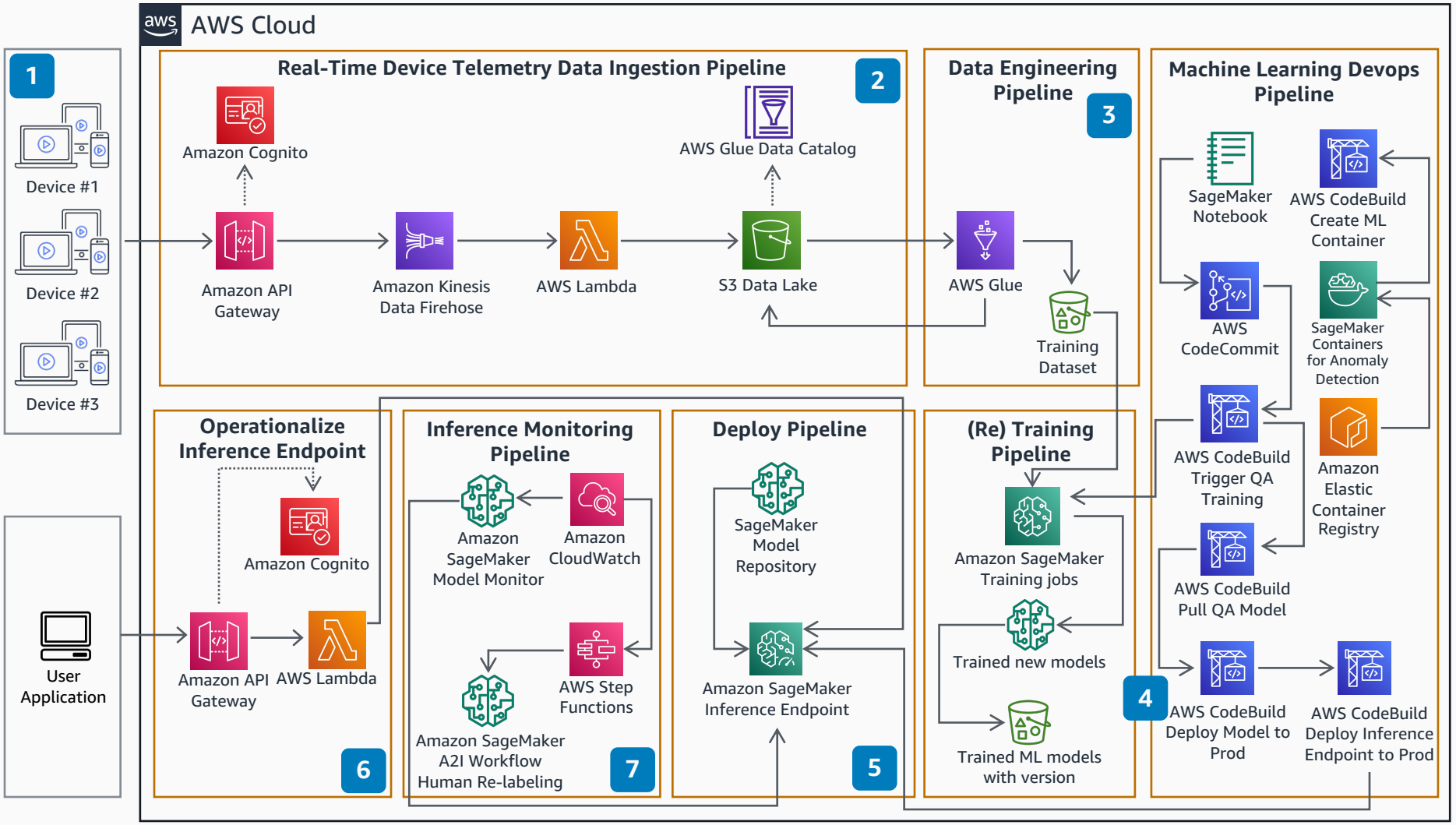
Build your own Anomaly Detection ML Pipeline

This end-to-end ML pipeline detects anomalies by ingesting real-time, streaming data from various network edge field devices, performing transformation jobs to continuously run daily predictions/inferences, and retraining the ML models based on the incoming newer time series data on a daily basis. Note that Random Cut Forest (RCF) is one of the machine learning algorithms for detecting anomalous data points within a data set and is designed to work with arbitrary-dimensional input.



1. Device telemetry data is ingested from the field devices on a near real-time basis by calls to the API via **Amazon API Gateway**. The requests get authenticated/authorized using **Amazon Cognito**.

2. **Amazon Kinesis Data Firehose** ingests the data in real time, and invokes **AWS Lambda** to transform the data into parquet format. **Kinesis Data Firehose** will automatically scale to match the throughput of the data being ingested.

3. The telemetry data is aggregated on an hourly basis and re-partitioned based on the year, month, date, and hour using **AWS Glue** jobs. The additional steps like transformations and feature engineering are performed for training the Anomaly Detection ML Model using **AWS Glue** jobs. The training data set is stored on **Amazon S3 Data Lake**.

4. The training code is checked in an **AWS CodeCommit** repo which triggers a Machine Learning DevOps (MLOps) pipeline using **AWS CodePipeline**. **CodePipeline** builds the **Amazon SageMaker** training and inference containers, triggers the **SageMaker** training job using the specified training dataset, deploys the trained model in the testing environment, and upon approval, deploys the model into production using **SageMaker** inference endpoints.

5. The ML models generated by training jobs are registered in the **SageMaker** Model Repository. The deploy pipeline selects the best ML model to deploy using **SageMaker** hosting.

6. Classify if the telemetry data is an anomaly or not via HTTP(s) API using **Amazon API Gateway** and **Lambda** functions. The **Lambda** function invokes the **SageMaker** endpoint to predict the anomaly.

7. The inference quality is monitored using **SageMaker** Model Monitor. The requests with ambiguous prediction scores are sent for re-labeling using **Amazon CloudWatch** events, triggering the **SageMaker** A2I workflow using **AWS Step Functions**.

Reviewed for technical accuracy June 1, 2021

AWS Reference Architecture