Continental Tire

Project Central

Contents

Modern Data Analytics Platform (MoDAP)

Introduction

The Modern Data Analytics Platform is a composition of cloud services from Amazon Web Services to enable developers to build and schedule data pipelines, deploy machine learning models and store data from different sources. It abstracts the complexity of using AWS services.

The key benefits of MoDAP are:

- Infrastructure Abstraction: Users of the Modern Data Analytics Platform are not required to have knowledge about AWS. Everything is abstracted from the users.
- Scalability: The MoDAP can dynamically scale compute resources based on demand and utilization.
- Cost Efficiency: Due to the scaling capabilities only required resources are used. This leads to a high-cost efficiency.
- Reduced Time to Market: Creating Data Engineering Pipelines can be achieved in minutes.

To realize the value held in the data generated by Continental, and the increasing agility and lower costs of experimentation by Data Scientists and Business Analysts, MoDAP implemented a DataLake in the AWS S3. S3 storage was chosen because of its high availability (99.99%), durability (eleven 9's), scalability, and native integration with other AWS services.

The DataLake enables query capabilities and serves new data science use cases by allowing the self-discovery of information through a Data Catalog. Data transformed into optimized usable format is searchable and can be queried using standard SQL (Athena) and BI tools (Tableau).

MODAP users can author their own ETL jobs using Apache Airflow for job orchestration and scheduling. To accelerate the flow of DAG submission, a CI/CD pipeline with DAG testing and automated deployment will be developed.

Architecture

Storage

The data to ingest and process is not time-critical according to the current use cases and may be processed over a timeframe of some minutes starting from the time it gets ingested into the Raw bucket. The AWS services employed are on-demand serverless solutions when possible, to reduce the operational burden to the development team and incur costs only when the service is being used.

MoDAP will ingest and store data into a centralized data platform. S3 was chosen as the storage platform due to:

- its scalability and durability
- ease of integration with other AWS services
- access control management
- encryption native capabilities
- serverless operation

As the data goes through extraction, transformation, and load process it is stored in separate buckets according to the stage of the pipeline, or the sensitivity / categorization of the data.

1. Raw: Data ingested from different sources in the original (raw) data format. The goal is to load all different sources into one single S3 bucket and categorize/partition them based on meta data information like data provider, region source, vehicle type, sensor type etc.
2. Normalized: The raw data e.g., csv files are getting normalized. Thereby all the files are going to have the same schema which is a requirement for some following AWS services. Also, the files are getting compressed e.g. csv → csv.gz.
3. Transformed: This bucket will store source/input data for Apache Airflow DAGs. The normalized data get transformed to a columnar data format for performance increase and cost reduction such as Apache Parquet (csv.gz → .parquet).
4. Analytics: This bucket will store results from Apache Airflow DAGs. The data source of Airflow DAGs can be the transformed bucket or even the analytics bucket itself.

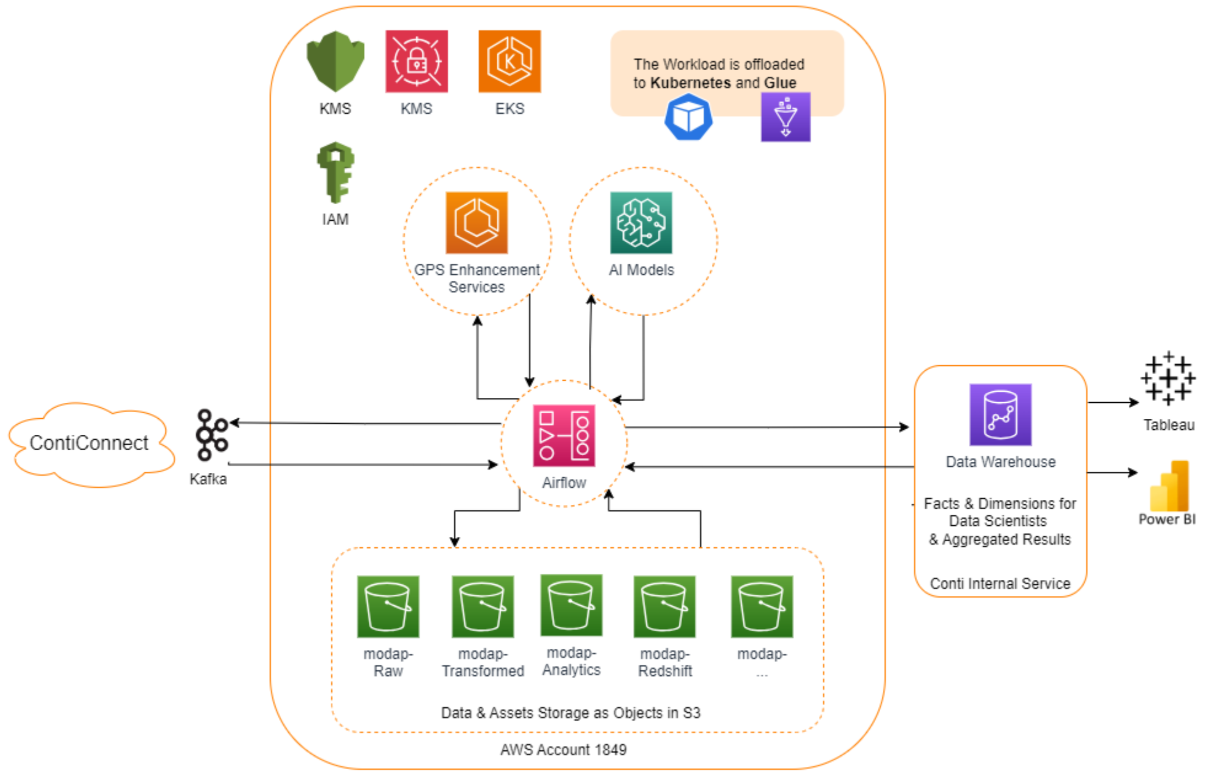


Figure 1: MoDAP General Architecture

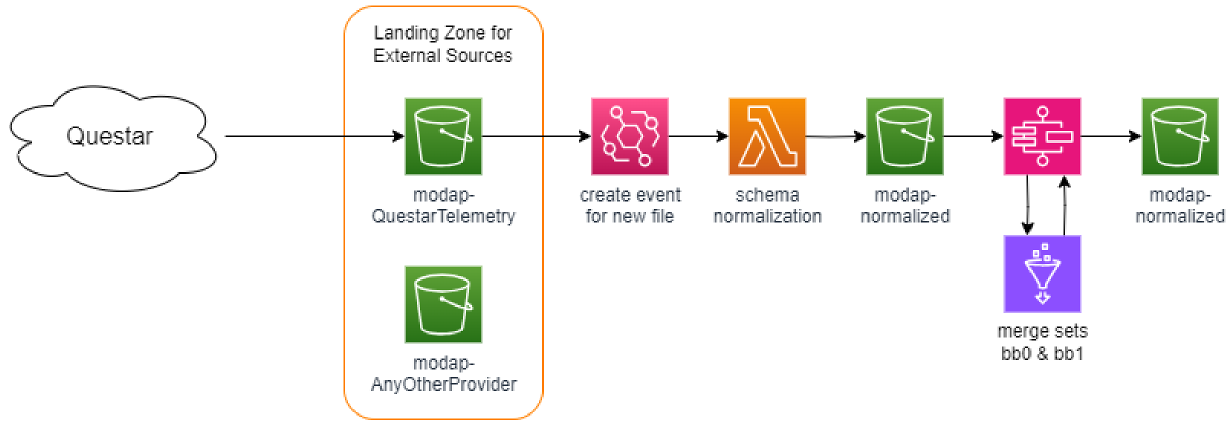


Figure 2: Questar Integration Architecture

MoDAP Development

MoDAP was developed using Infrastructure as Code (IaC) and AWS Cloud Development Kit (CDK). The AWS Cloud Development Kit (AWS CDK) is an open-source software development framework developed by Amazon Web Services (AWS) for defining and provisioning cloud infrastructure resources using familiar programming languages. The AWS CDK aims to improve the experience of working with Infrastructure as Code by providing higher-level, reusable constructs that enable developers to create and manage AWS resources more efficiently and with less boilerplate code compared to traditional configuration files like AWS CloudFormation templates.

Workflow

MoDAP uses the Apache Managed Workflows for AirFlow (MWAA) service to orchestrate new data science project workflows.

The MoDAP MWAA is setup via infrastructure as code using CDK stacks. One stack was implemented to set up the required local runner and database images and to push them to ECR manually. Additional stacks were implemented to create the pipeline and the MWAA environment.

- The MWAA code pipeline stack creates the following resources:
- the MWAA bucket to store DAGs (Directed Acyclic Graphs) with corresponding permissions and KMS key
- the artifact bucket for storage of pipeline resources with corresponding permissions and KMS key
- the codebuild project to be executed in a pipeline stage
- a build project role with corresponding permissions
- the MWAA code pipeline with stages source, test/build and infra
- a pipeline role with corresponding permissions

Data Quality

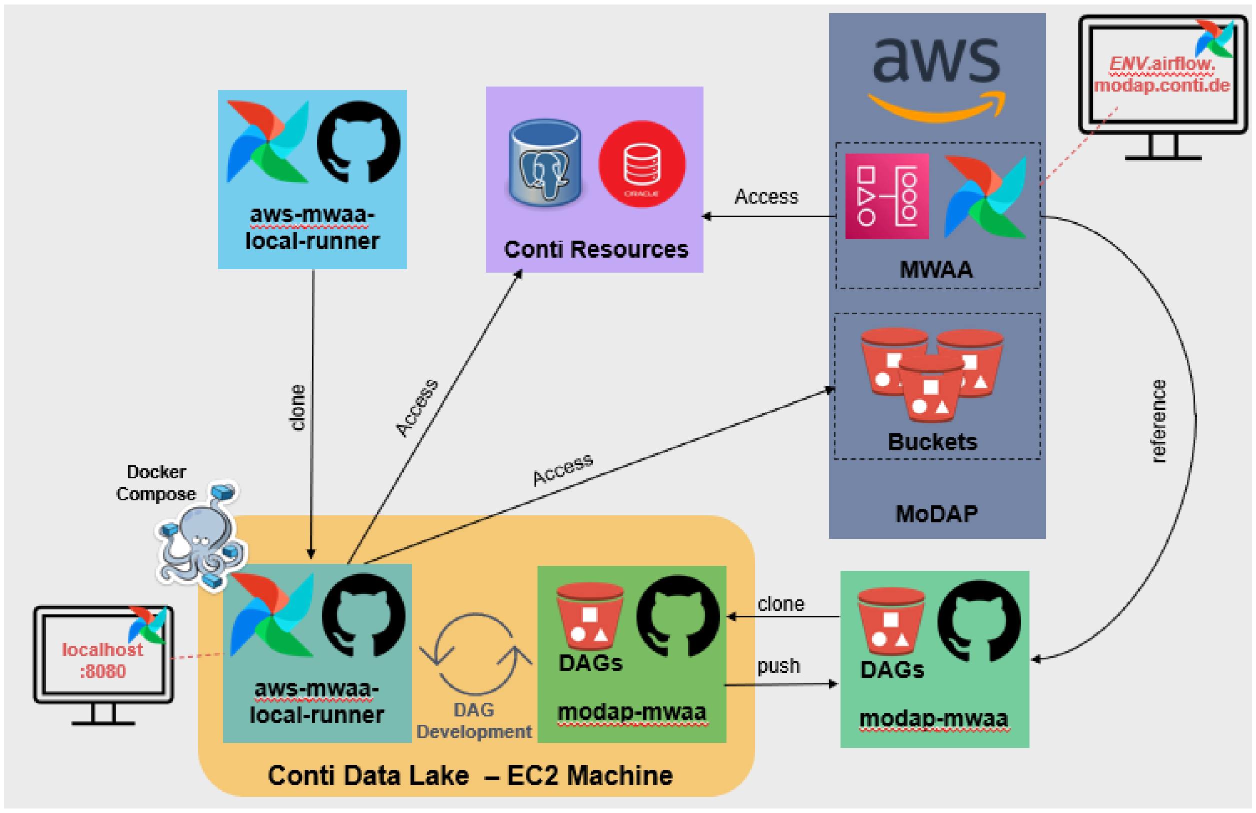
Data Quality recipes were created using AWS Glue DataBrew and CDK. AWS Glue DataBrew is a visual data preparation tool that makes it easier for data analysts and data scientists to clean and normalize data to prepare it for analytics and machine learning (ML). The following actions were implemented for the Questar datasets: blackbox0 and blackbox1:

- A CDK Construct was implemented
- DataBrew RuleSets were implemented with column statistics. Each RuleSet contained one or more Rule Properties (e.g., percentage of missing values for a specific column).
- DataBrew RuleSets were implemented to check for dataset integrity. Each RuleSet contained one or more Rule Properties (e.g., number of rows and columns in dataset, timestamp column format)

End User Experience

The process of developing DAGs and deploying them to MoDAP consists of two independent Airflow instances aws-mwaa-local-runner and MWAA (Managed Workflows Apache Airflow).

The figure below shows the components and processes which are also described underneath.



- aws-mwaa-local-runner: Is a containerized airflow instance which is running on your Conti Data Lake machine. It can be accessed with adfs-cli Conti and MoDAP resources the same way as MWAA in MoDAP. If you start with the development of your DAG, you will initially interact with aws-mwaa-local-runner.
- Conti Data Lake - EC2 Machine: Needs to be set up by the developer. It is the development environment for your data pipelines. Runs an airflow instance (aws-mwaa-local-runner) with Docker Compose. From here you can push after development of the DAG to modap-mwaa repository which holds all the developed DAGs where the common Airflow instance is fetching from.
- MoDAP - MWAA & Buckets: MoDAP has Managed Workflows for Apache Airflow (MWAA) deployed which is an Airflow instance managed by AWS. AWS S3 Buckets for different purposes are available and can be accessed by MWAA as well as aws-mwaa-local-runner. MWAA fetches some airflow resources like DAGs, requirements.txt and plugins from the Github repository modap-mwaa.
- Conti Resources: PostgreSQL DB like CTTA AL can be accessed from MWAA and aws-mwaa-local-runner in the same way. This guarantees that if the access was working during development in the Data Lake EC2 Machine, it will also work in MWAA.