

Databricks Implementation on AWS for a global online retailer

Our Customer, an US based online marketplace and clothing reseller. They have large AWS infrastructure spread across multiple accounts and multiple pipelines within US Regions resulting in significant business risks.



Challenges

Data at the Retailer was inherently distributed due to various factors such as service-oriented architecture, multi-account cloud infrastructure, and a sprawling set of data ingestion tools. This led to many challenges including the lack of a single source of truth, scalability, security, and data incompleteness. Our customer was using a high storage configuration, computing tools and languages to manage and process the data. This led to high costs and operational overhead.

Solution

- Our team built the Databricks unified platform for data engineering, machine learning and analytics. The platform was built on underlying core AWS infrastructure, networking and security services, and was compliant with enterprise cloud security standards.
- The whole platform was built as Infrastructure as a Code (IaaS) using Terraform & DevOps tools.
- A host of foundational services were built and integrated to enable and complete the data & AI/ML ecosystem.
- Workflow orchestration was set up using Airflow, CI/CD process, Databricks MLflow, and MLOps capabilities including experiment tracking, model registry & serving.
- A series of sessions were conducted to educate customer IT teams about AWS best practices and guidelines to prevent security vulnerabilities and operate on the Databricks platform.

Innovation

- Automated intervention, no manual activity
- Planning, Designing and Architecting Setup with IAAC

Benefits

- Centralized and unified platform for data engineering, machine learning and analytics.
- Fully self managed CICD process.
- Moving to Well Architected Framework and implement best practices
- Automation

Result Highlights

Complete Setup with IaaS

100% Self Managed CICD

Centralized Platform for Data, ML and Analytics