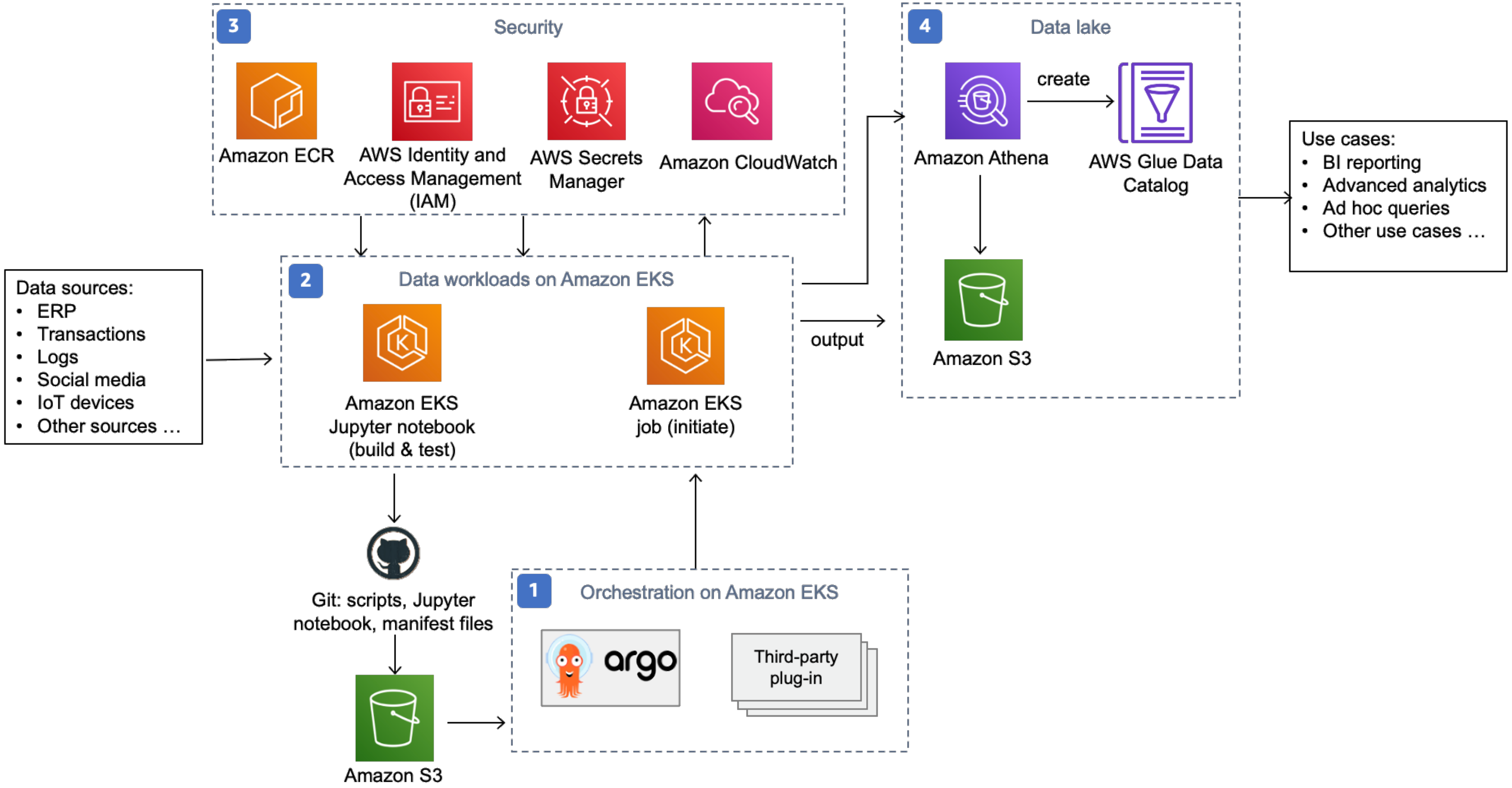
SQL-Based ETL with Apache Spark on Amazon EKS

This solution provides declarative data processing support, codeless ETL capabilities, and workflow orchestration automation to help your business users access their data and create meaningful insights without the need for manual IT processes. To deploy this solution using the available AWS CloudFormation template, select **Deploy with AWS**.



1. A customizable and flexible workflow management layer (the Orchestration on **Amazon EKS** group) includes the Argo Workflows plug-in. This plug-in provides a web-based tool to orchestrate your ETL jobs without the need to write code.

2. A secure data processing workspace is configured to unify data workloads in the same **Amazon Elastic Kubernetes Service (Amazon EKS)** cluster. This workspace contains a second web-based tool, JupyterHub, for interactive job builds and testing. You can either develop Jupyter notebook using a declarative approach to specify ETL tasks or programmatically write your ETL steps using PySpark. This workspace also provides Spark job automations that are managed by the Argo Workflows tool.

3. A set of security functions are deployed in the solution. **Amazon Elastic Container Registry (Amazon ECR)** maintains and secures a data processing framework Docker image. The **AWS Identity and Access Management (IAM) roles for service accounts (IRSA)** feature on **Amazon EKS** provides token authorization with fine-grained access control to other AWS services. Jupyter fetches login credentials from **AWS Secrets Manager** into **Amazon EKS** on-the-fly. **Amazon CloudWatch** monitors applications on **Amazon EKS** using the activated **CloudWatch Container Insights** feature.

4. The analytical workloads on the **Amazon EKS** cluster outputs data results to an **Amazon S3** data lake. A data schema entry (metadata) is created in an **AWS Glue Data Catalog** via **Amazon Athena**.

Deploy with AWS

Reviewed for technical accuracy July 29, 2021

Deployable AWS Reference Implementation