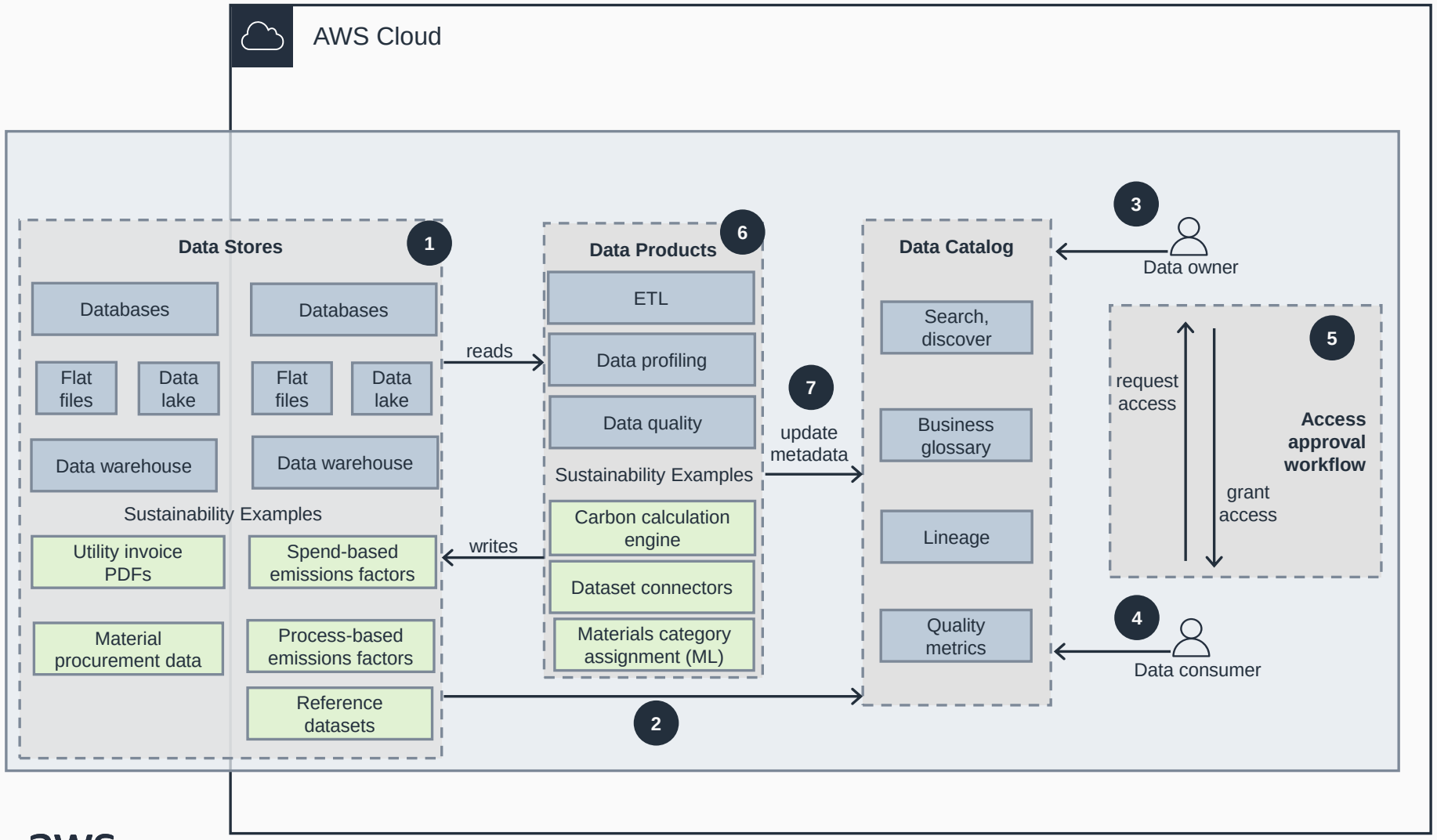
Guidance for Sustainability Data Management on AWS

This architecture diagram illustrates how sustainability applications can both consume and produce data assets, incorporating key data management concepts to quickly share and extract trusted value from data across your organization. The subsequent slides cover user access, data discovery, and automated data asset registration workflows tailored for sustainability use cases.



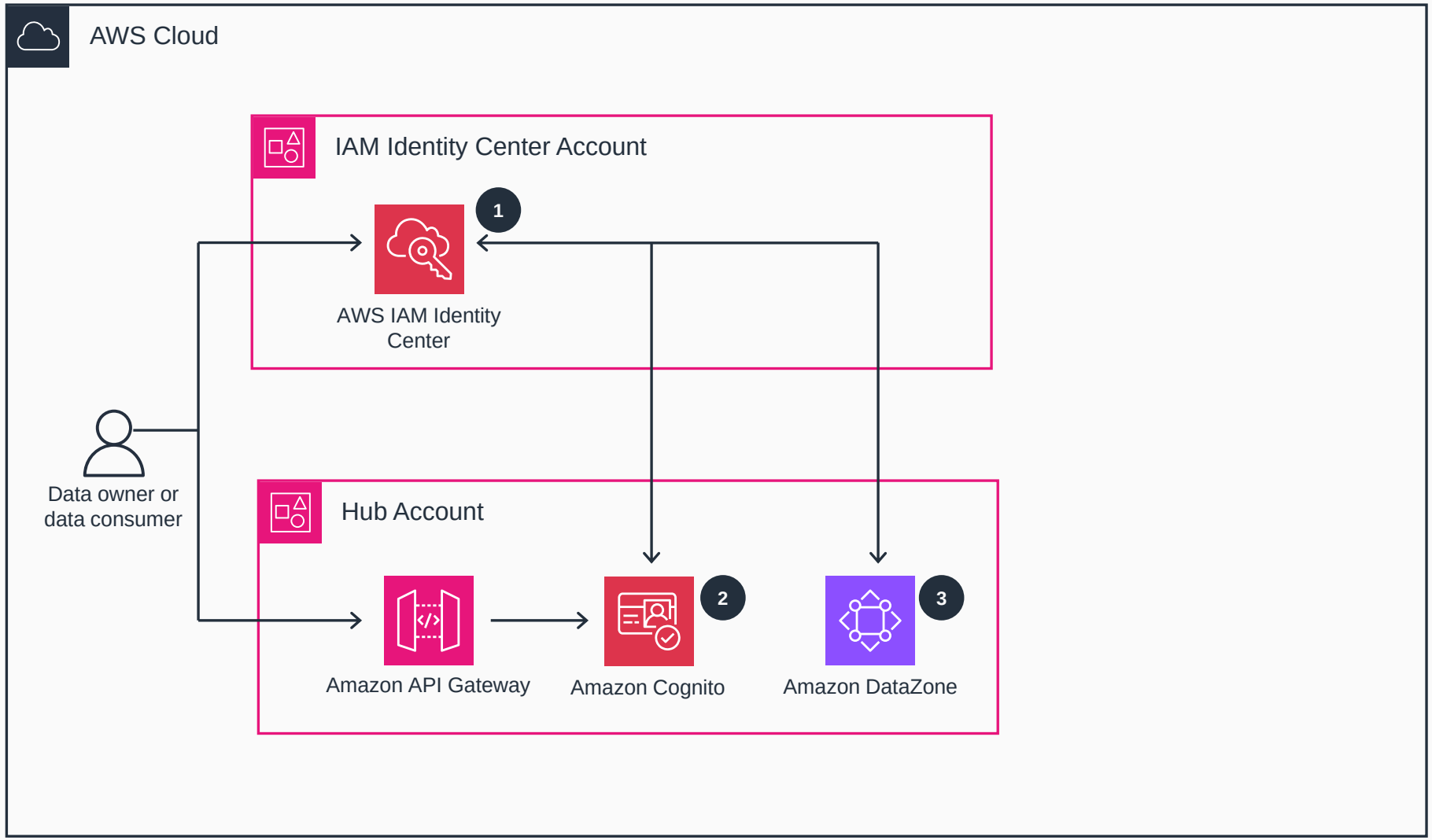
1. Data is stored in various types of data stores, within and/or outside of AWS. These data stores contain data assets that represent a physical data object (such as a database table or a file). These data stores house both source and target datasets in the data fabric.
2. Technical metadata is automatically imported into the data catalog for data assets that existed before the implementation of the data fabric.
3. The data owners maintain business metadata for their data assets in the data catalog to enrich the data with business context. For example, business context for dataset columns, tags, domain- or enterprise-wide business glossary terms.
4. The data consumers search the data catalog for data assets using technical and/or business metadata. The metadata pertaining to data quality and data lineage establishes trust in how data assets can be used.
5. The data consumers request access to the relevant data assets from the data owner, who can either grant or deny the request.
6. The data products perform export, transform, and load (ETL), data profiling, and data quality operations to create new curated data assets to enable data-driven use cases for the data consumers.
7. Data assets created by the data products are registered in the data catalog with the corresponding metadata.

Reviewed for technical accuracy April 10, 2024

AWS Reference Architecture

Guidance for Sustainability Data Management on AWS

User access to the data catalog.

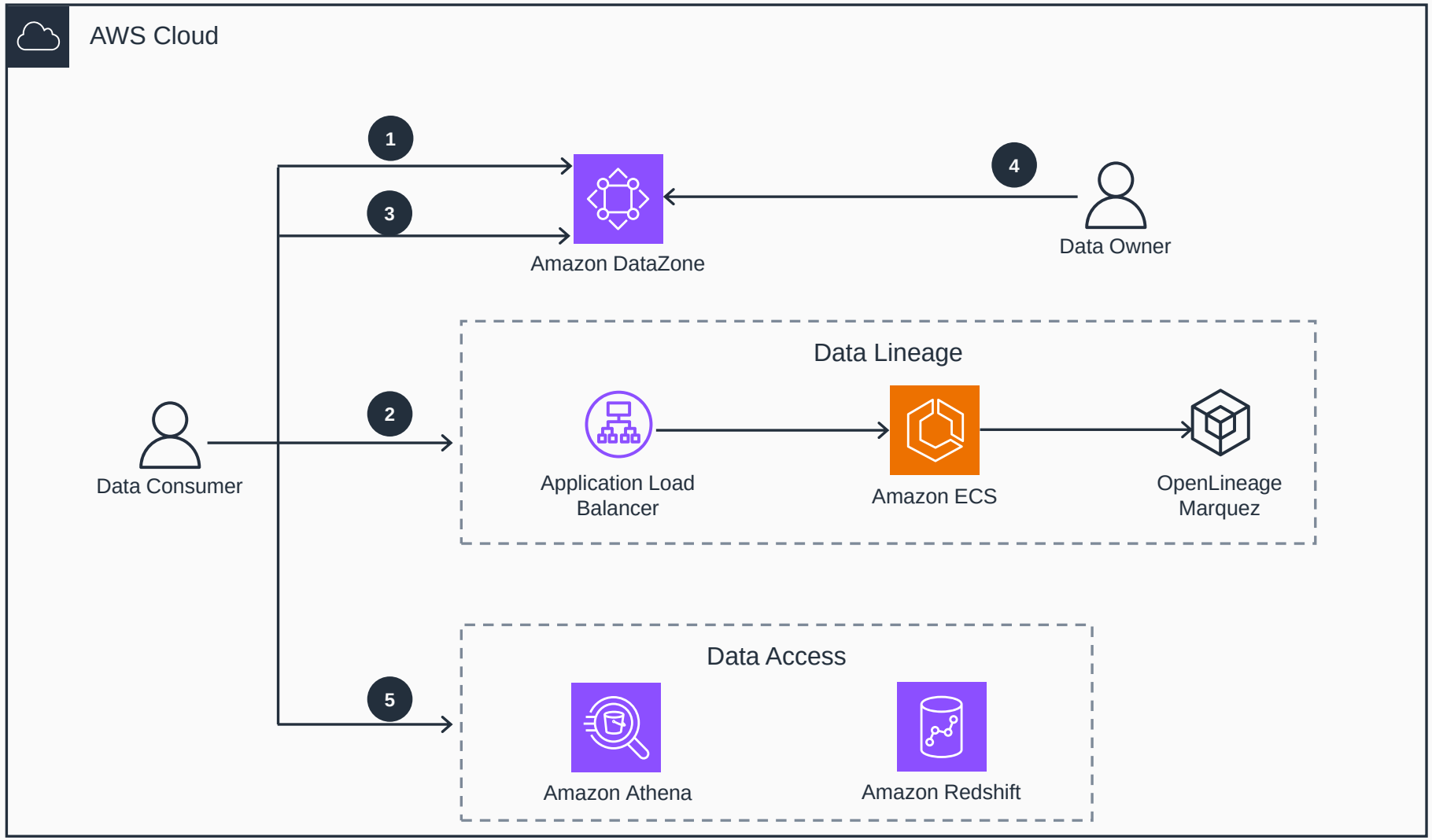


1. **AWS IAM Identity Center** manages all users for both **Amazon DataZone** and the other APIs.
2. **Amazon API Gateway** uses an **Amazon Cognito** authorizer. The corresponding user pool uses **IAM Identity Center** as its identity provider.
3. **Amazon DataZone** integrates directly with **IAM Identity Center** for user management.

Reviewed for technical accuracy April 10, 2024

Guidance for Sustainability Data Management on AWS

Search, discover, and request access to data assets in the data catalog.



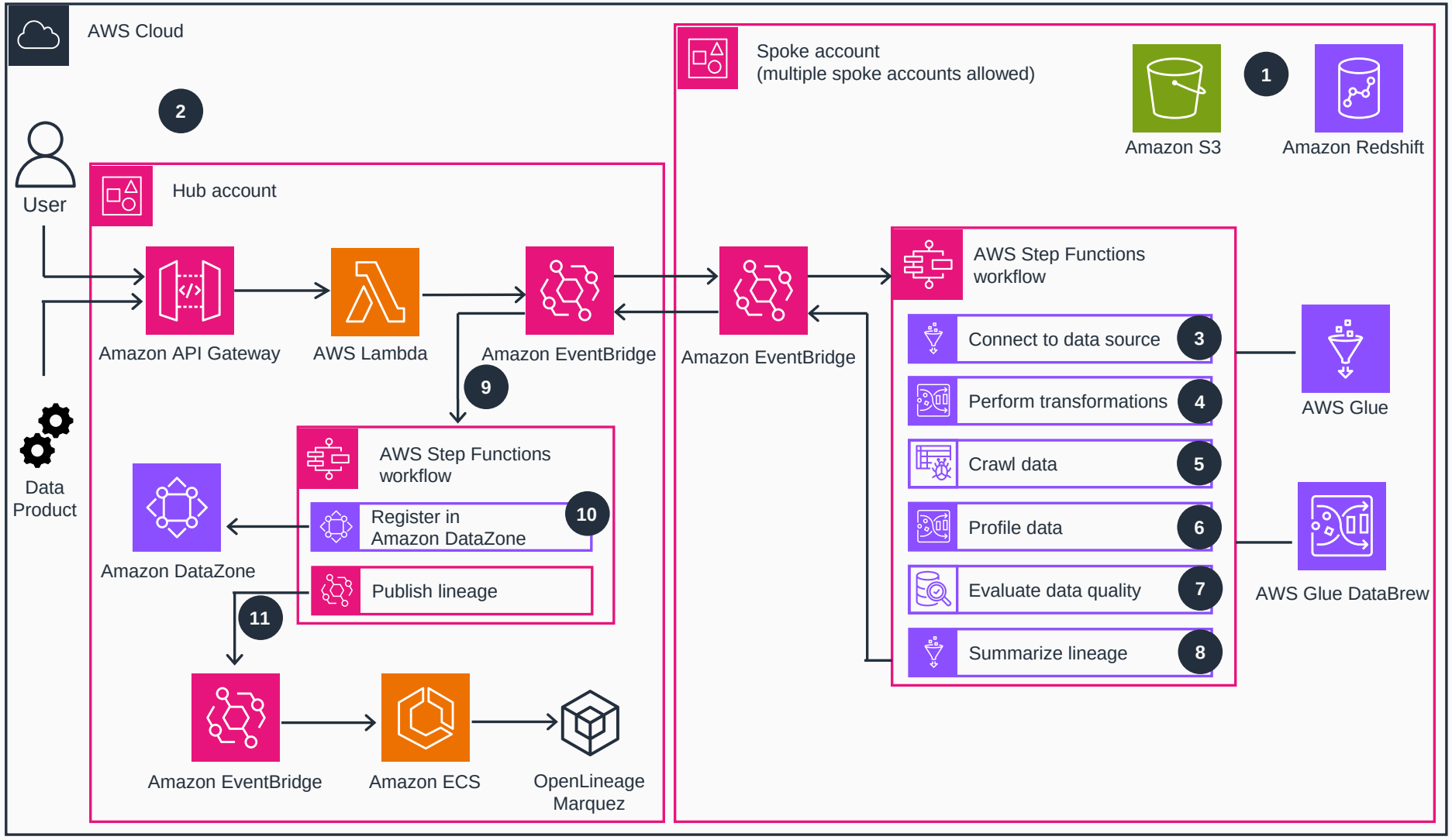
1. Users explore the data catalog through the search functionality in **Amazon DataZone**. Assets can be searched for by their associated metadata.
2. Data lineage for each asset is stored in an instance of OpenLineage Marquez. Marquez is deployed on an **Amazon Elastic Container Service (Amazon ECS)** container fronted by an Application Load balancer. Users can view the data lineage of assets through Marquez.
3. From the data catalog, the data consumer requests read-only access to a desired dataset from the data asset owner.
4. Asset owners approve or deny subscription requests to individual assets that they have published to the catalog.
5. Once an asset owner approves a user's subscription request, the user can access the asset through **Amazon Athena,** for assets registered as **Amazon Glue** tables, or through the Amazon Redshift Data API for **Amazon Redshift** tables.

Reviewed for technical accuracy April 10, 2024

AWS Reference Architecture

Guidance for Sustainability Data Management on AWS

Data asset registration with profiling, transformation, quality assertion, and lineage tracking.



1. Data is placed into **Amazon Simple Storage Service (Amazon S3)** or **Amazon Redshift**.
2. A data owner or data product invokes an **API Gateway** API backed by **AWS Lambda** in the Hub account. The API body includes information on the data location, transformation logic, profiling specifications, and data quality assertions required in future steps. The API writes an event to an **Amazon EventBridge** event bus, which replicates it to an event bus in the spoke account.
3. The event in the spoke account invokes an **AWS Step Functions** workflow. The workflow creates an **AWS Glue** connection to the **Amazon Redshift** or **Amazon S3** data source.
4. **AWS Glue DataBrew** performs data transformations through a recipe job.
5. An **AWS Glue** crawler infers the schema of the resulting dataset and creates an **AWS Glue** table.
6. An **AWS Glue DataBrew** profile job derives profile statistics against the table.
7. **AWS Glue** evaluates the data quality with user-defined assertions.
8. The resulting data lineage is summarized in the event and sent back to the hub account through **EventBridge.**
9. The **EventBridge** event bus in the hub account invokes another **Step Functions** workflow.
10. The new asset is imported into **Amazon DataZone** by creating and running a data source.
11. The lineage for the asset is published to **EventBridge**, which invokes an **Amazon ECS** deployment to register the lineage in a deployment of OpenLineage Marquez.

Reviewed for technical accuracy April 10, 2024

AWS Reference Architecture