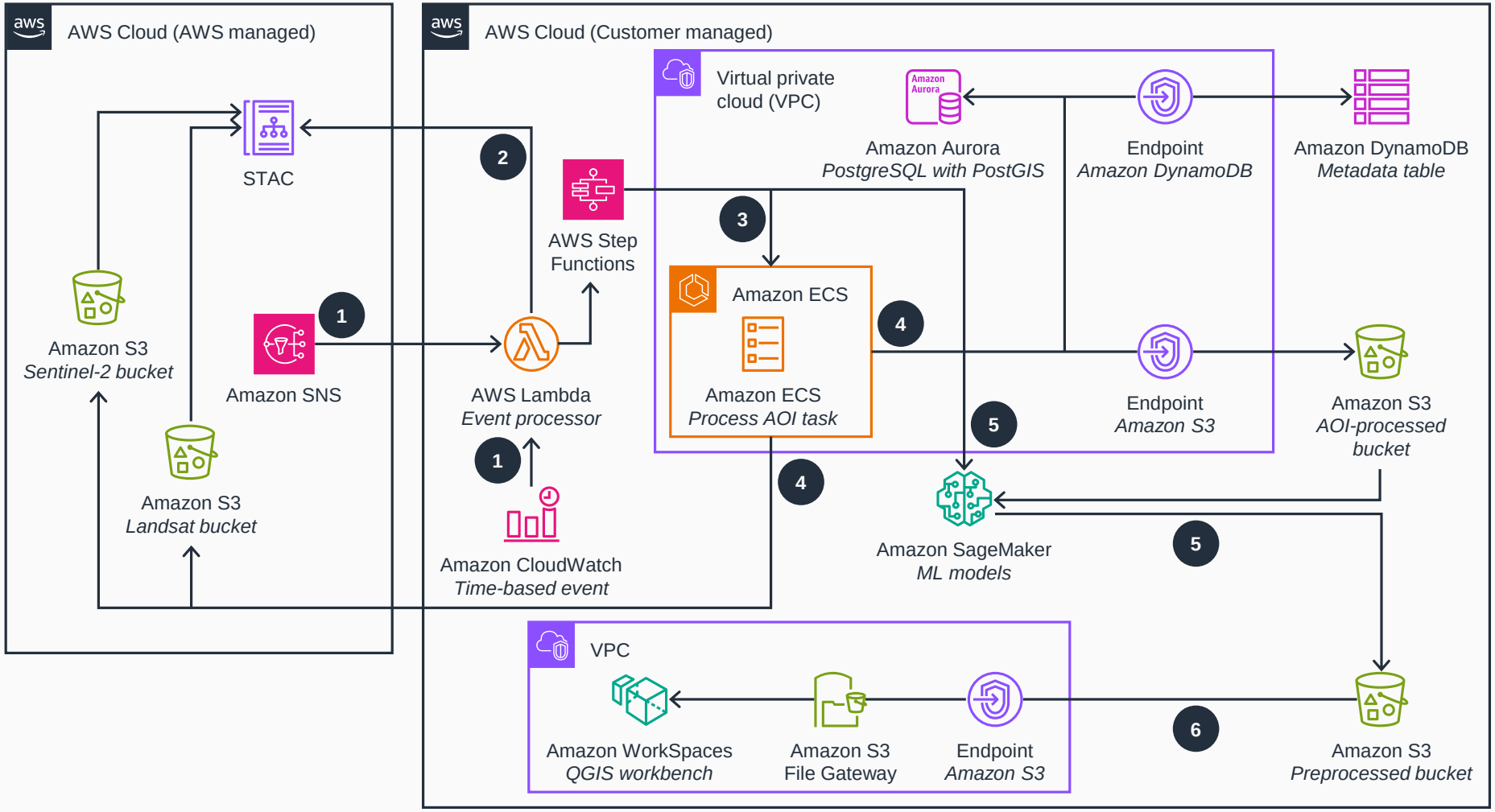
Guidance for Scaling Geospatial Data Lakes with Earth on AWS

This architecture diagram shows how to build scalable geospatial data repositories on AWS. It simplifies the design of geospatial data pipelines, making it faster to access raw data by integrating datasets managed by the Registry of Open Data on AWS. As a result, you won't need to store this data in your own data lake.



1. Invoke a data ingestion pipeline based on new scene detection. Subscribe to **Amazon Simple Notification Service (Amazon SNS)** topics for managed datasets with appropriate filters, and configure time-based ingestion rules using **Amazon CloudWatch**.

2. **AWS Lambda** queries the SpatioTemporal Asset Catalog (STAC) API for respective datasets to get product details. **Lambda** then invokes the data processing pipeline through **AWS Step Functions**.

3. **Step Functions** orchestrates the processing tasks. Parallel or sequential processing can be configured based on task characteristics and requirements.

4. **Amazon Elastic Container Service (Amazon ECS)** runs the following containerized tasks:
   - Downloads the products from datasets hosted on the Registry of Open Data on AWS.
   - Processes the tiles (through crop and geomosiac operations) to the area of interest and stores them in an **Amazon Simple Storage Service (Amazon S3)** automated optical inspection (AOI)–processed bucket.
   - Builds metadata and stores it in **Amazon DynamoDB**.
   - Stores vector data in **Amazon Aurora PostgreSQL-Compatible Edition** with PostGIS extensions.

5. **Step Functions** invokes the next processing task with **Amazon SageMaker**. Machine learning (ML) models on **SageMaker** perform cloud removal and band math and store the data in an **Amazon S3** preprocessed bucket.

6. QGIS, a geographic information system (GIS) software, is hosted on **Amazon WorkSpaces**. The QGIS workbench is used for visualizing or further processing.

Reviewed for technical accuracy July 2, 2024

AWS Reference Architecture