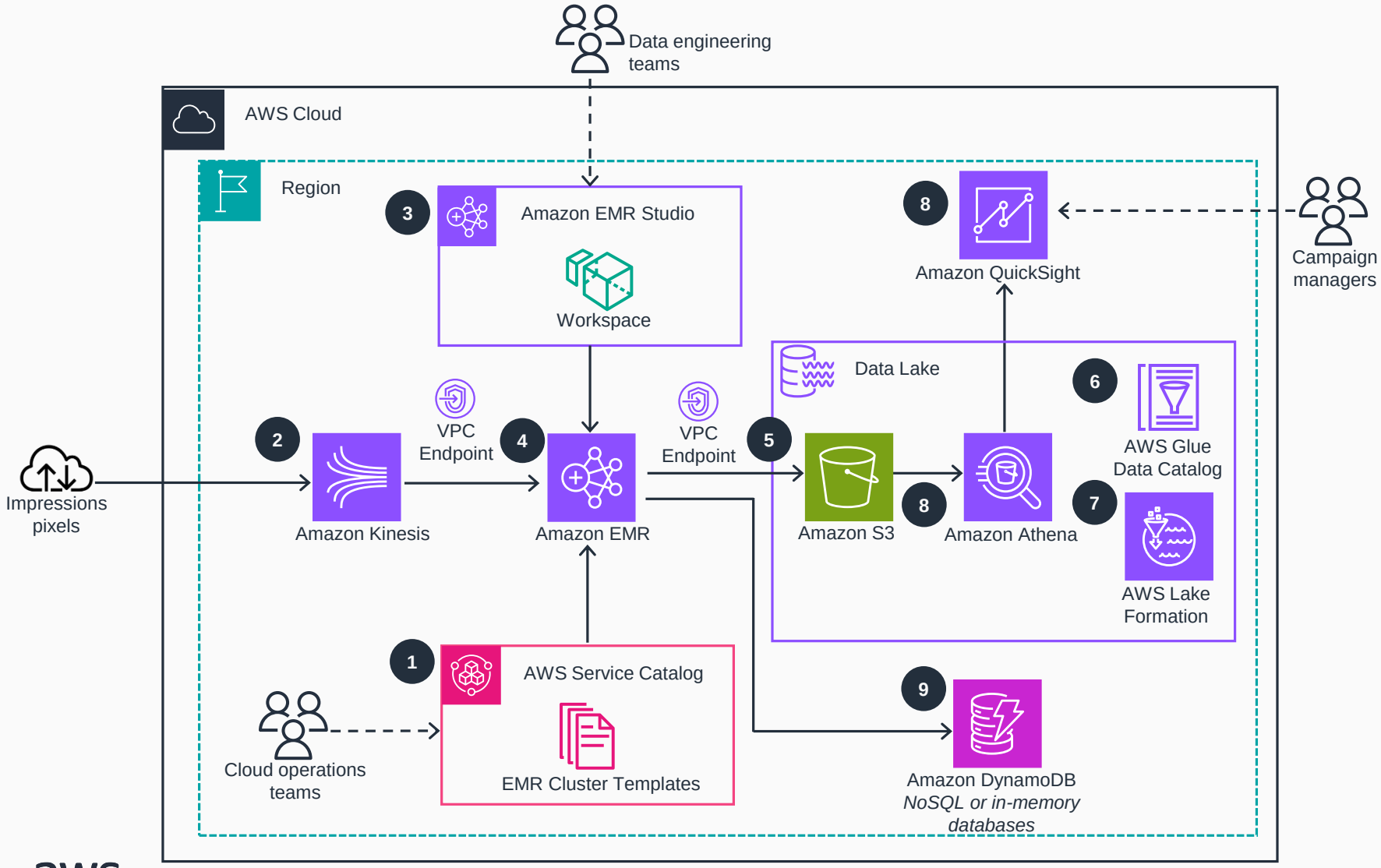
Guidance for Implementing Near Real-Time Analytics with Spark Streaming on AWS

This architecture diagram illustrates how advertising technology (AdTech) companies can build a self-service analytics platform for developing and deploying streaming data processing pipelines on big data workloads. Steps 1-5 are outlined here. For details on steps 6-9, refer to the next slide.



1. Cloud operations teams develop **Amazon EMR** cluster templates in **AWS CloudFormation** according to their desired specifications (such as instance types and network configurations) and publish the templates as products in the **AWS Service Catalog** for self-service provisioning.

2. Bid events or pixels on web ads capture user impressions and sends the data to an **Amazon Kinesis Data Streams** endpoint.

3. Data engineering teams log in to their workspaces in **Amazon EMR** Studio. Here, they self-provision **Amazon EMR** clusters. Alternatively, they attach existing clusters to develop Spark streaming applications, like bid validation or impression measurement, using interactive notebooks.

4. A Spark streaming application runs on an **Amazon EMR** cluster. It continuously ingests raw bid or impression event data from **Kinesis Data Streams**. The application transforms the data. It then stores the transformed data in an **Amazon Simple Storage Service** (Amazon S3) data lake. This process enables near real-time operational reporting. You can choose provisioned **Amazon EMR** clusters for the most flexibility in cost optimization or serverless **Amazon EMR** clusters to simplify deployment and cluster management.

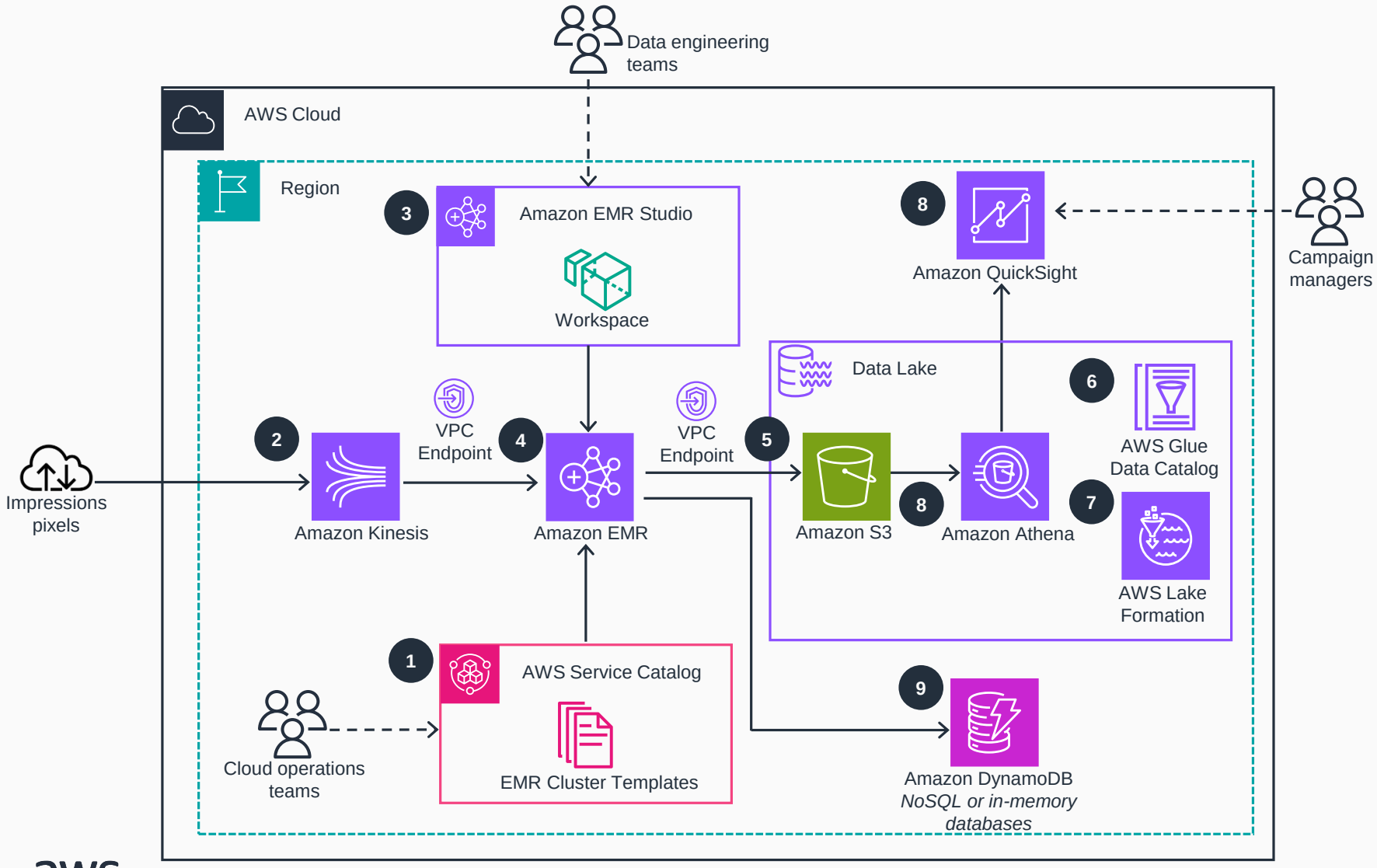
5. **Amazon S3** stores data in partitioned folders. The data can be compressed and in columnar format or in other open table formats like Apache Iceberg.

Reviewed for technical accuracy April 12, 2024

AWS Reference Architecture

Guidance for Implementing Near Real-Time Analytics with Spark Streaming on AWS

Steps 6-9



6 All database and table metadata is registered within an **AWS Glue** Data Catalog, so data can be queried by multiple AWS services like **Amazon Athena** or **Amazon SageMaker**.

7 (Optional) Data lake administrators can register the Data Catalog with **AWS Lake Formation** to provide more granular access controls and centralize user management.

8 Users can run SQL queries against curated clickstream or impression data in **Amazon S3** in near real-time with **Athena** and visualize dashboards with **Amazon QuickSight**.

9 In addition to the **Amazon S3** data lake, **Amazon EMR** workloads can write data to NoSQL databases like **Amazon DynamoDB** or in-memory databases like Aerospike. This supports read workloads requiring fast performance on a large scale, such as bid filtering or operational reporting.

Reviewed for technical accuracy April 12, 2024

AWS Reference Architecture