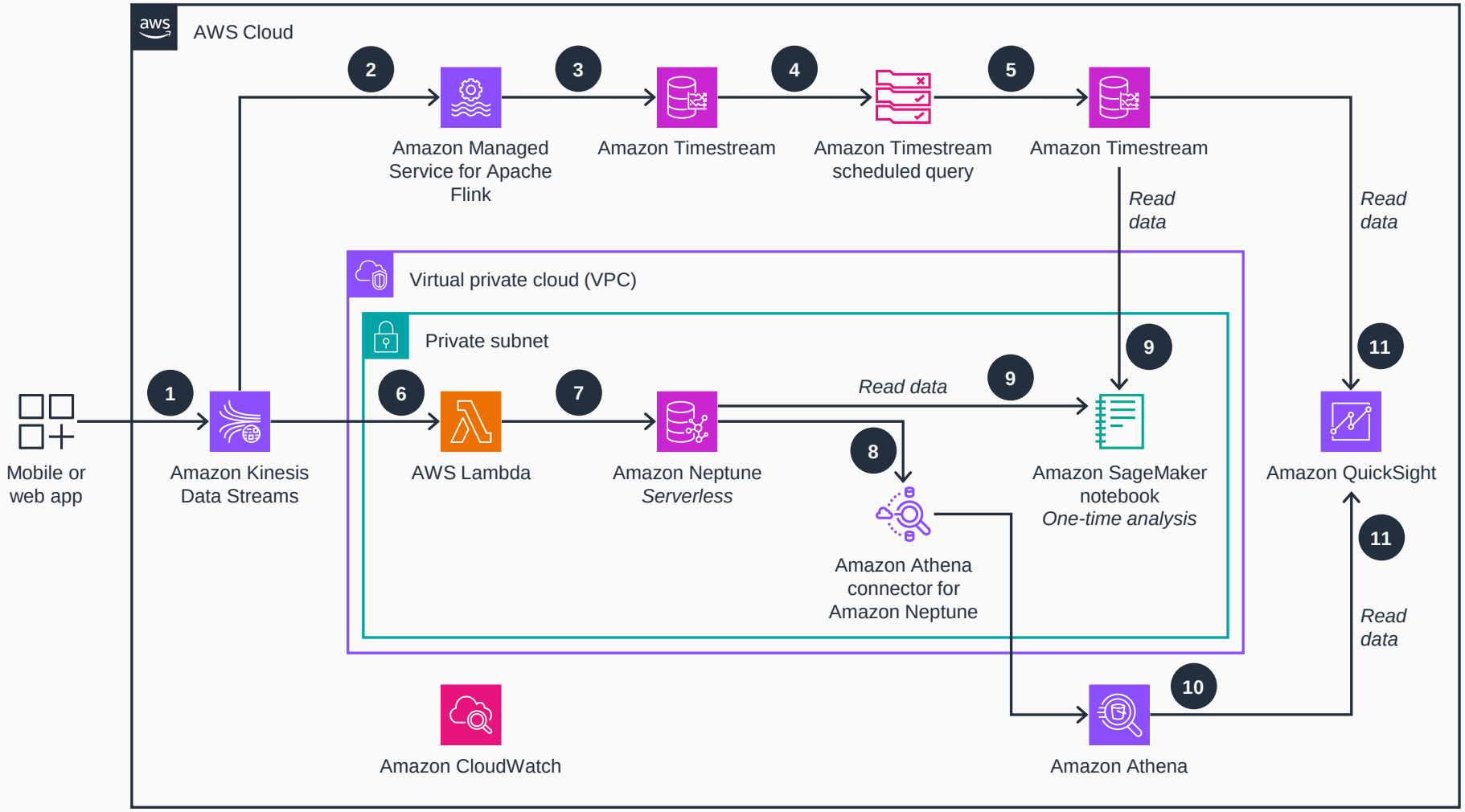
Guidance for Transactional Fraud Detection on AWS

This architecture diagram shows how to set up a serverless workflow designed to capture streaming transactions, identify potential fraudulent activities using Amazon Timestream, and conduct further impact analysis using Amazon Neptune.



1. The applications sends real-time transaction data to a data ingestion API.
2. Analyze the transaction data stream using an adapter between **Amazon Kinesis Data Streams** and **Amazon Timestream**. The adapter is deployed as an **Amazon Managed Service for Apache Flink** application.
3. Send the analyzed stream to a **Timestream** transaction table.
4. A **Timestream** scheduled query identifies aggregate metrics, such as aggregated high-value transactions made by an account in the last 5 minutes.
5. The aggregated transaction metrics are stored in another **Timestream** table.
6. Analyze the transaction data stream with an adapter between **Kinesis Data Streams** and **Amazon Neptune** based on **AWS Lambda**.
7. Store graph data in a **Neptune** database for macroanalysis.
8. **Neptune** data is made available an **Amazon Athena** connector for **Neptune**.
9. Analyze the data points and graphs using a custom script (one-time analysis) with an **Amazon SageMaker** notebook.
10. **Athena** provides federated access for downstream systems.
11. Visualize the suspected fraud accounts and their network of accounts by using **Amazon QuickSight** dashboards.

Reviewed for technical accuracy October 11, 2024

AWS Reference Architecture