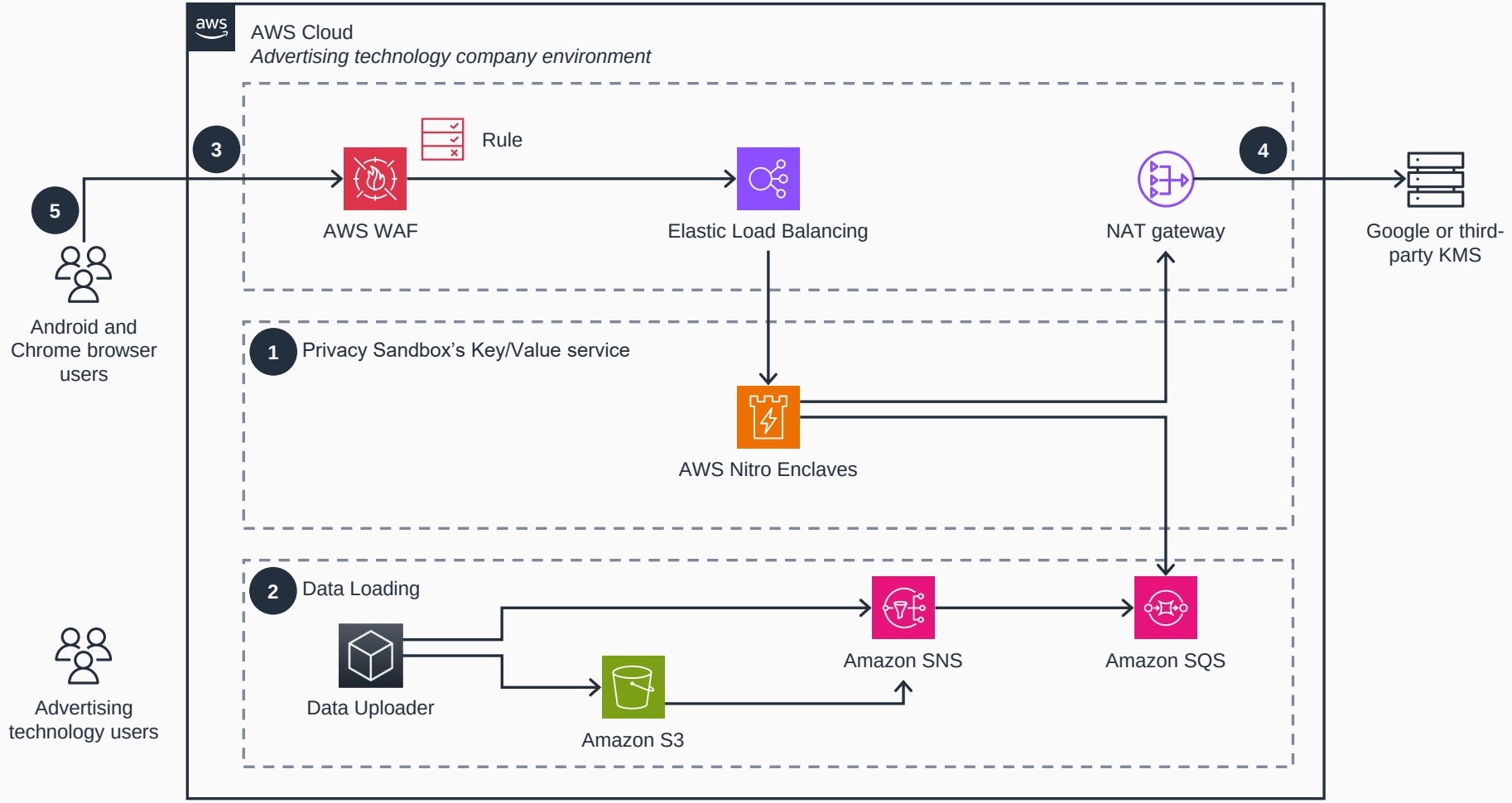
Guidance for Implementing Google Privacy Sandbox Key/Value Service on AWS

Overview

This architecture diagram shows how advertising technology companies can deploy Privacy Sandbox's Protected Audience API Key/Value service within a trusted runtime environment using AWS services. This slide details steps 1-3 of the overview.



Key/Value Service Deployment

1. Privacy Sandbox's Protected Audience API Key/Value service is deployed using a Google-provided terraform in your **Amazon Virtual Private Cloud (Amazon VPC)** environment. This Key/Value service runs within **Amazon Elastic Compute Cloud (Amazon EC2)**, and an **Amazon EC2 Auto Scaling** group scales the service horizontally to meet the workload demand. The data is secured in an **AWS Nitro Enclaves** trusted runtime environment. The communication between the parent Auto Scaling group and the enclave is done using a secure local channel.

2. Deploy additional AWS resources using a Google-provided terraform that provisions services and resources to load Key/Value data (in the form of delta or snapshot files) into Privacy Sandbox's Key/Value service application. Files are generated using the Privacy Sandbox data command line interface (CLI) application, which is hosted on **Amazon Elastic Container Service (Amazon ECS)**. The Key/Value service subscribes to data updates using **Amazon Simple Notification Service (Amazon SNS)** and **Amazon Simple Storage Service (Amazon S3)** event notifications. For low-latency updates, the Key/Value service listens to an **Amazon SNS** topic by long polling a subscribed **Amazon Simple Queue Service (Amazon SQS)** queue. VPC endpoints powered by **AWS PrivateLink** provide private connectivity between the Key/Value service VPC and other AWS resources.

Functional Flow

3. During the Privacy Sandbox auction, an end user's Chrome or Android browser (enabled with Privacy Sandbox features) sends a lookup request to the Key/Value service application's endpoint URL. **AWS WAF** protects the application from common web exploits, and **Elastic Load Balancing** distributes the traffic to healthy application endpoints.

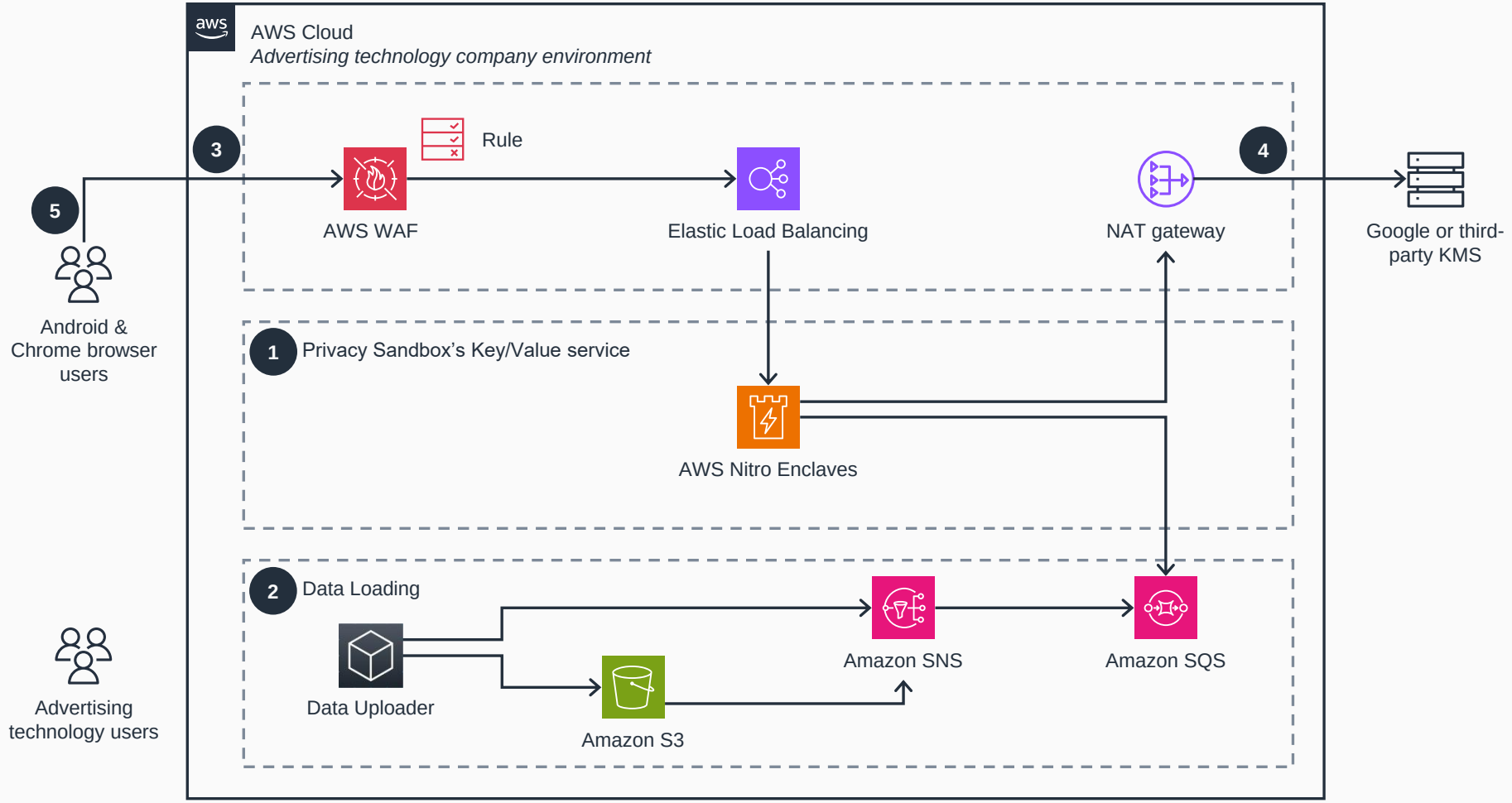
Reviewed for technical accuracy May 20, 2024

AWS Reference Architecture

Guidance for Implementing Google Privacy Sandbox Key/Value Service on AWS

Overview

This architecture diagram shows how advertising technology companies can deploy Privacy Sandbox's Protected Audience API Key/Value service within a trusted runtime environment using AWS services. This slide details steps 4-5 of the overview.



4 The Key/Value service application has secure outbound internet access through a NAT gateway to Google's key management system, operated by a coordinator. The Key/Value service obtains a private key from Google's key management system to decrypt the requests from the client device. Note that this is not **AWS Key Management Service (AWS KMS)**.

5 The Key/Value service application looks up the matching data for the keys and returns it in encrypted form to the user's Chrome or Android browser.

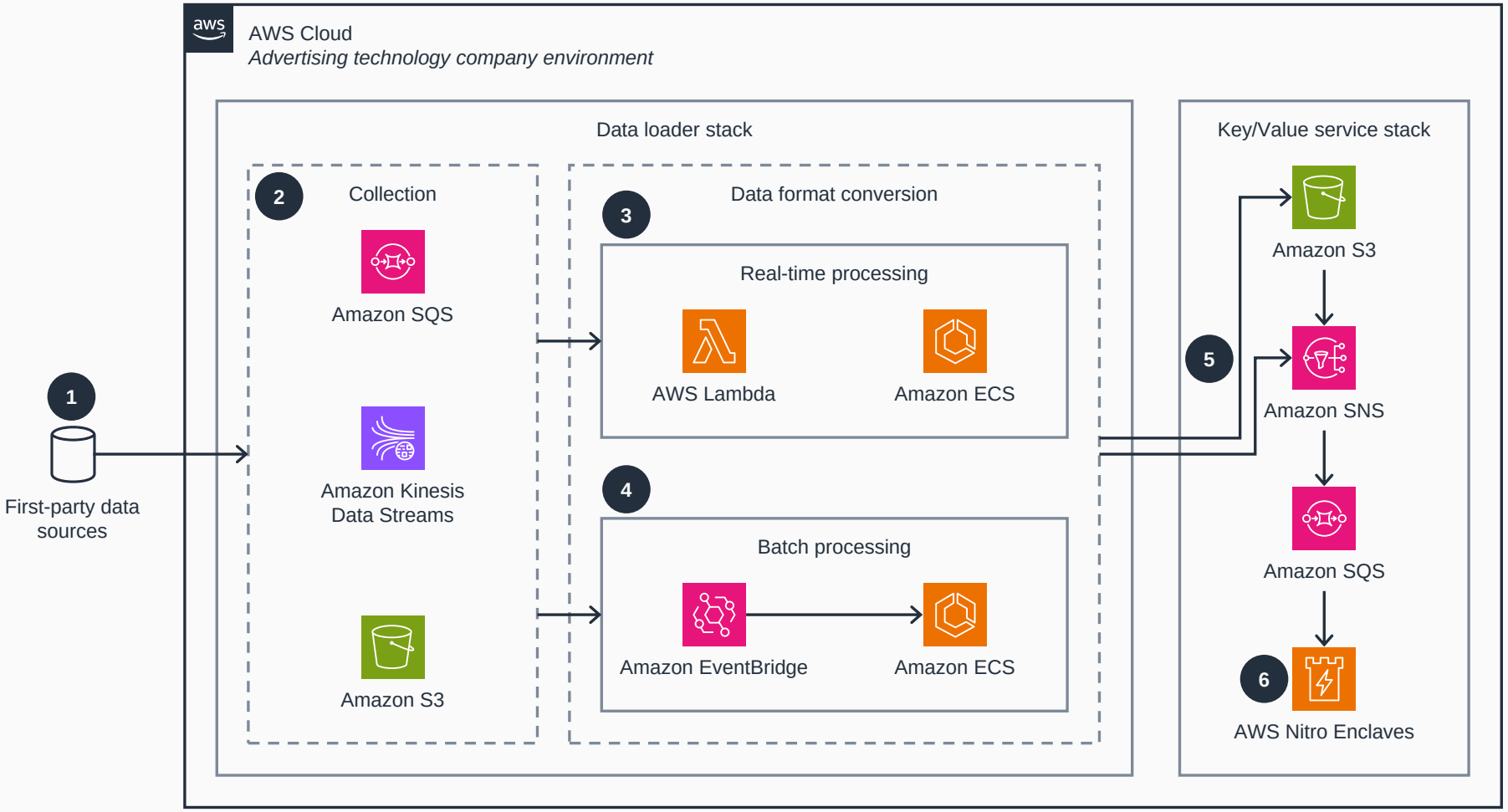
Reviewed for technical accuracy May 20, 2024

AWS Reference Architecture

Guidance for Implementing Google Privacy Sandbox Key/Value Service on AWS

Data Loading

This architecture diagram provides a closer view of the data loading component and demonstrates patterns for ingesting the first-party data needed for real-time auction and bidding into Privacy Sandbox's Protected Audience API Key/Value service.



1. Your first-party data likely exists in multiple formats, and your system of records is part of your existing real-time bidding application. To support ad targeting through Privacy Sandbox APIs, your first-party data needs to be generated or copied to Privacy Sandbox's Protected Audience API Key/Value service.

2. Data to be stored in the Key/Value service is first sent to one of the three data collection services. **Amazon SQS** supports an API-to-API-based integration pattern. **Amazon S3** supports a file-based integration pattern. **Amazon Kinesis Data Streams** supports an enhanced fan-out pattern for loading data to the Key/Value service.

3. Real-time data flowing in through **Amazon SQS** and **Kinesis Data Streams** is processed by a data formatter application running in a Docker container and is hosted in **Amazon ECS**. Alternatively, you can also use an **AWS Lambda** function to process the data formatter application. The application uses **AWS SDK** and Google's data CLI tool to receive and convert the incoming format to delta format, which the Key/Value service can read.

4. Batch data files uploaded to **Amazon S3** generate an event invocation and run an **Amazon EventBridge** rule, which in turn implements an **Amazon ECS** task. The **Amazon ECS** task runs the data formatter application, which uses **AWS SDK**, **AWS Command Line Interface (AWS CLI)**, and Google's data CLI tool to receive and convert the incoming format to delta format.

5. The data formatter application sends data to an **S3** bucket and **Amazon SNS** for consumption by the Key/Value service.

6. The Key/Value service is deployed separately, as explained in the overview architecture diagrams, using an infrastructure-as-code terraform stack. This stack provisions AWS resources that listen to incoming data rows by means of **Amazon S3** or **Amazon SNS**, and it loads them into the Key/Value service running on **Nitro Enclaves**.

Reviewed for technical accuracy May 20, 2024

AWS Reference Architecture