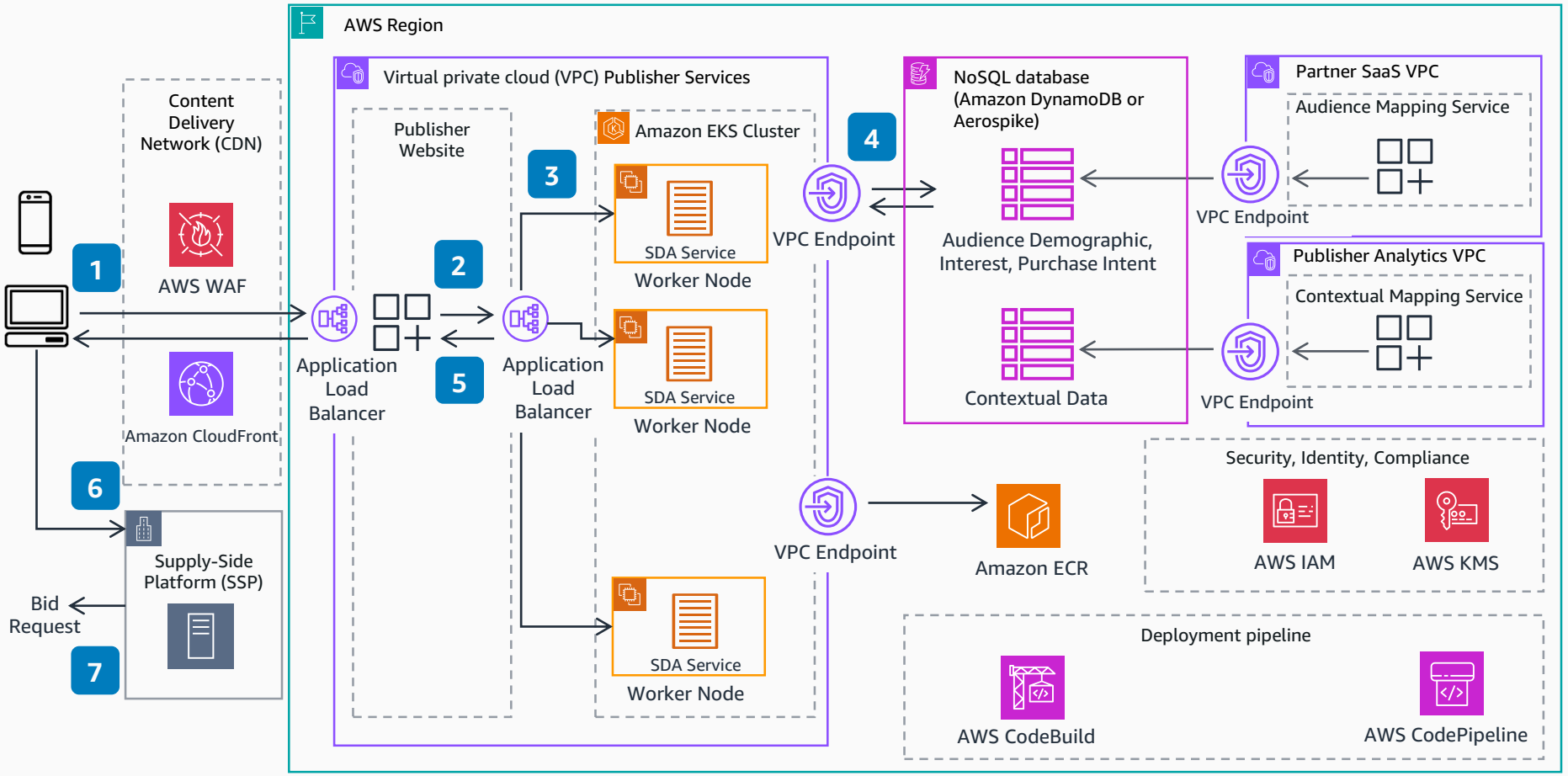
Guidance for Activating Seller Defined Audiences on AWS

This diagram shows how to activate publisher first-party data from Software as a Service (SaaS) environments that support Seller Defined Audiences (SDA). It uses page content without Personally Identifiable Information (PII) to automatically map proprietary taxonomies, returning the associated SDA IDs for activation through Real-Time Bidding (RTB).



1 A visitor's browser, mobile client, or Connected TV (CTV) device accesses publisher content containing ad impressions. The OpenRTB header bidding platform, such as prebid.js, is loaded with the page invoking the on-page data assembler. The data assembler forwards a Seller Defined Audiences (SDA) data request to an **Amazon CloudFront** distribution. The publisher's content and endpoints are protected by **AWS WAF** and **CloudFront**. **CloudFront** forwards the request to the **Application Load Balancer (ALB)** public endpoint on the publisher's **Virtual Private Cloud (VPC)** over the AWS network.

2 The publisher's web tier routes the SDA data request to the internal **ALB** private endpoint.

3 The internal **ALB** routes the SDA data request to the SDA service fleet on **Amazon Elastic Kubernetes Service (Amazon EKS)** for processing.

4 Available attributes (such as the page context classification) and user data (such as audience demographics, interest, and purchase intent) are fetched from the NoSQL database such as Aerospike or **Amazon DynamoDB**. Aerospike will run within the **VPC** and does not require a VPC endpoint. Configure the rack-aware feature on Aerospike for better performance.

5 The SDA data containing page context and audience taxonomy segment data is returned to the caller through **CloudFront**. The returned SDA data does not contain a unique ID of the user nor does it reveal a user's identity.

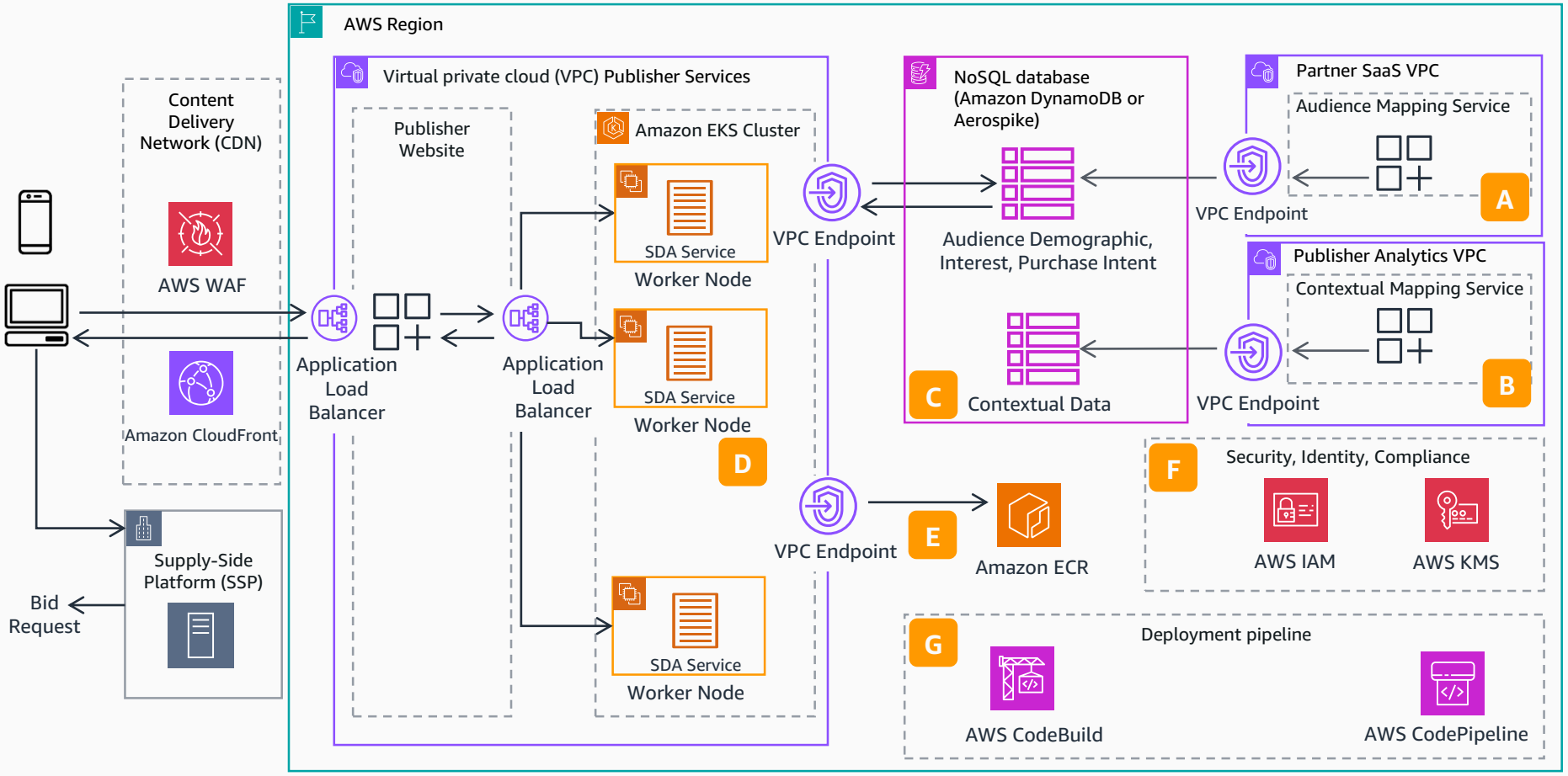
6 The on-page data assembler sets the fetched page context classification attributes in the 'site.content' top-level object. The audience related data is set within the 'user.data' top-level object. Both of these objects are configured on Prebid.js. The new segtax identifier extension, that is introduced within these objects for SDA support, determines the provided segments. In the case of site content, this identifier can be custom or the standardized IAB Tech Lab Content Taxonomy. Custom taxonomy types must be registered with IAB Tech Lab to be assigned a number. Prebid.js submits the bid request to the Supply-Side Platform (SSP).

7 The SSP parses the incoming request, resolves the data from the Prebid ortb2 object, transmits the data into the bid stream after applying the same ortb2 fields, and submits the bid request to its demand sources.

Reviewed for technical accuracy April 4, 2025

AWS Reference Architecture

Guidance for Activating Seller Defined Audiences on AWS



A **Consideration A**
The audience data in the NoSQL database is updated by an audience mapping service. This data flow occurs out of band from the RTB process. The publisher could leverage a partner to implement this service.

B **Consideration B**
The contextual data in the NoSQL database is updated by a contextual mapping service. This data flow occurs out of band from the RTB process. The publisher could utilize the Guidance for Contextual Intelligence for Advertising on AWS Guidance to build a contextual mapping service, or leverage a partner to provide this service.

C **Consideration C**
DynamoDB, Aerospike, or any other NoSQL database can be considered for storing audience and contextual data. When using **DynamoDB**, you can boost query performance by **using Amazon DynamoDB Accelerator (DAX)**, which provides in-memory acceleration to the **DynamoDB** tables.

D **Consideration D**
Use **AWS Graviton Processor** instances for bidder nodes. For additional cost optimization, implement auto-scaling.

E **Consideration E**
To minimize boot time, pre-install SDA service container images with dependent libraries and binaries. Upload the images to a container registry like **Amazon Elastic Container Registry (Amazon ECR)**.

F **Consideration F**
Encrypt and decrypt data at rest and in transit across **DynamoDB** and **Amazon EKS** using **AWS Key Management Service (AWS KMS)**. Grant least privilege access using **AWS Identity and Access Management (IAM)** to provide permissions for users, roles, and services.

G **Consideration G**
Automate the deployment of the SDA service using **AWS CodeBuild** and **AWS CodePipeline** to reduce time-consuming, manual processes.

Reviewed for technical accuracy April 4, 2025

AWS Reference Architecture