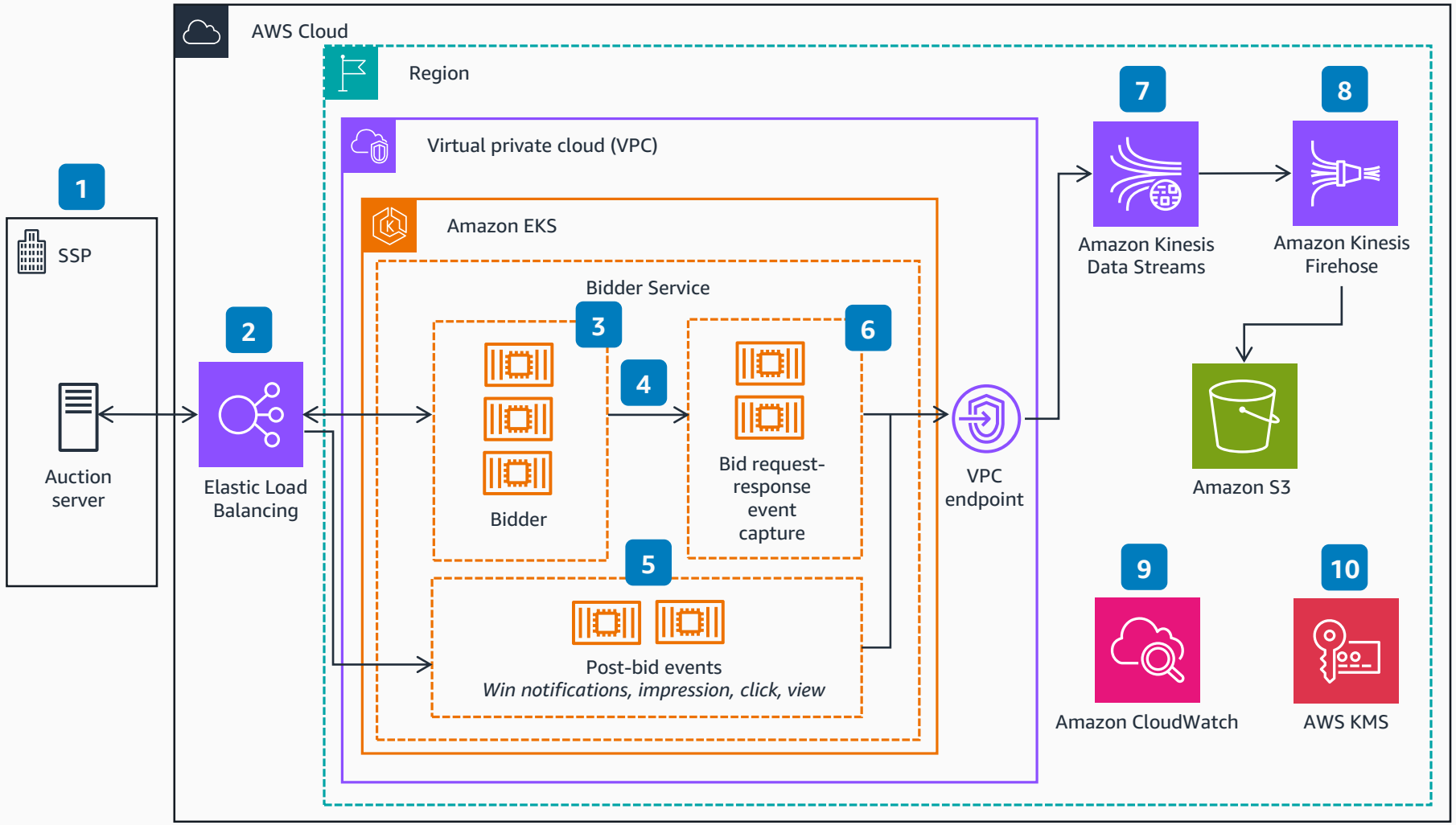
Guidance for Capturing Advertising OpenRTB (Real-Time Bidding) Events for Analytics on AWS

This architecture enables you to capture OpenRTB bid events for both near real-time and batch analytics.



1. The supply-side platform (SSP) receives an ad request from a publisher and launches an auction.
2. An OpenRTB bid request is sent to a DSP public endpoint that is configured on an **Elastic Load Balancer**.
3. The bidder application service receives the bid request. This service runs on **Amazon Elastic Kubernetes Service (Amazon EKS)** within an **Amazon Virtual Private Cloud (VPC)**.
4. The bid request-response event capture service is hosted on a different container pod in the same cluster. To reduce latency of the bid response, combine the publishing of the bid request and response event as a single call per bid asynchronously to the bid request-response event capture service endpoint.
5. Post bid events capture service is hosted on a separate container pod that exposes the service to SSPs. This service is used to receive post bid events.
6. Implement the event capture service in Java to take advantage of **Amazon Kinesis** Producer Library (KPL). KPL simplifies implementation of an asynchronous producer application and reduces costs for sending data to the **Amazon Kinesis Data Streams** API.
7. The event messages are routed to **Kinesis Data Streams** through a dedicated VPC endpoint.
8. **Amazon Kinesis Data Firehose** consumes these aggregated records and deaggregates and sends individual events to **Amazon Simple Storage Service (Amazon S3)** for long-term storage and analytics.
9. **Amazon CloudWatch** captures application logs for traceability.
10. **AWS Key Management Service (AWS KMS)** stores and manages encryption keys used for securing persisted data in **Kinesis Data Streams** and **Amazon S3**.

Reviewed for technical accuracy June 14, 2023

AWS Reference Architecture