AWS Well-Architected Framework

Video Streaming Advertising Lens

Video Streaming Advertising Lens: AWS Well-Architected Framework

Table of Contents

Video Streaming Advertising Lens - AWS Well-Architected Framework

Publication date: **December 9, 2025** (*Document revisions*)

This paper describes the Video Streaming Advertising (VSA) Lens for the AWS Well-Architected Framework. The lens explores how to review and improve your cloud-based architectures and better understand the impact of design decisions. We present general design principles and specific best practices aligned to the six pillars of the Well-Architected Framework.

Introduction

The AWS Well-Architected Framework helps cloud architects build secure, high-performing, resilient, and efficient infrastructure for their applications and workloads. The AWS Well-Architected Framework is based on six pillars. The pillars are operational excellence, security, reliability, performance efficiency, cost optimization and sustainability. AWS Well-Architected provides a consistent approach for customers and AWS Partners to evaluate architectures, remediate risks, and implement designs that deliver business value.

In this lens, we focus on how to design, architect, and deploy your advertising workloads in the AWS Cloud. We define components, explore common workload scenarios, and outline design principles that help you apply the AWS Well-Architected Framework. We recommend that you begin designing your architecture by considering the best practices and questions from the AWS Well-Architected Framework whitepaper.

Operational challenges with the advertising workloads are:

- High traffic volumes with tens of millions of transactions per second.
- Low latency application and data retrieval responses with single-digit millisecond response time SLA.
- Rapid changes in traffic volumes and associated fluctuations in compute and network infrastructure.
- Data transfer is a significant part of overall operational costs and a focus area.
- Very low revenue (and profit margin) per transaction drives the focus on cost. Cost efficiency is the dominant design principle.

- End-to-end network latency impacts time available for response processing. Roundtrip latency of under 300 milliseconds is required to meet industry trading service-level objectives (SLOs).
- The use of 3rd party ISVs (independent software vendors) requires responses under two milliseconds to meet end-to-end processing SLAs.
- Rapid traffic changes require stateless and flexible infrastructure to facilitate automated up and down scaling of platform.
- Flexible supply and demand reduces redundancy requirements.
- Handling massive data volumes (100+ petabytes) and high throughput (up to 1+ trillion events per week).
- Verify data privacy compliance with regulations like GDPR and CCPA while processing large amounts of user data.
- Providing low-latency reporting and ad-hoc querying capabilities for campaign performance analysis and business intelligence.
- Scaling data processing pipelines cost-effectively as data volumes and business complexity grow rapidly.
- Enabling advanced use cases like machine learning for advertising performance measurement and attribution modeling.
- Maintaining operational efficiency with managed services as engineering team sizes may not keep up with data growth.
- Achieving high availability and resilience in data pipelines to support critical advertising workloads.
- Address alternate solution option for addressable targeting to deliver personalized ads, due to significant signal loss of traditional data points like third-party cookies, mobile IDs (IDFA for iOS and the Android Advertising ID (AAID)), hashed emails, and IP addresses, driven by privacy regulations.
- In addition to these individual challenges, organizations face multiple interconnected challenges:
  - Protecting user privacy while complying with evolving regulations like GDPR and CCPA
  - Combating sophisticated ad fraud and verifying brand safety through robust content moderation
  - Adapting to the phaseout of third-party cookies by developing alternative measurement approaches using first-party data and modeling
  - Managing data quality and secure collaboration across multiple solutions and partners

- Effectively using AI/ML technologies for fraud detection, content moderation, and cross-system measurement while maintaining transparency and human oversight

- Three main categories of digital advertising fraud: placement fraud, traffic fraud, and action fraud. These fraudulent activities employ both automated and manual methods, targeting different aspects of the advertising solution.

  - *Placement fraud* involves manipulating ad placements through techniques like malvertising, ad stacking, fake websites, domain spoofing, and ad injection.

Does What (Sub-Type of Fraud)		Who (Fraudster)	to Whom (Victim)	How (Objective)
Stuffing	Keyword Stuffing Placement Stuffing Stacking	Dishonest Publisher	Advertiser	• A dishonest publisher stacks multiple ad placements over one another—all advertisers are charged for the impressions/clicks even though only the top ad is visible to the user(s).
Domain Spoofing/Fake Sites		Fraudster	Advertiser/User	• A fraudster mounts/publishes a copy of a well-known websites to mislead potential advertisers and/or users into thinking that they are dealing or interacting with the legitimate website.
Malicious Toolbar/ Malicious Adware		Fraudster/ Dishonest Publisher	Advertiser/User/ Premium Publisher	• A user inadvertently installs a malicious tool-bar in their browser, which then allows the fraudster to inject ad windows into websites the user visits, thus artificially increasing the number of impressions on the shown/injected ads.
Ad/Content Injection		Dishonest Publisher/ Deceitful ISP/ Malicious Competitor Publisher	Advertiser/User/ Publisher	• A dishonest publisher installs ad-injection scripts on their website to show more ads per webpage and increase their per-impression and/or per-click revenue. • A deceitful Internet Service Provider injects ads of their own choosing into the websites that their customers visit, (Only possible in case of unencrypted client-server communication).

Placement fraud

  - *Traffic fraud* focuses on artificially inflating visitor numbers and clicks using bots or human labor.

Does What (Sub-Type of Fraud)		Who (Fraudster)	to Whom (Victim)	How (Objective)
Click Fraud	Impression Fraud Publisher Click Inflation Advertiser Competition Clicks	Dishonest Publisher/ Malicious Competitor Advertiser	Advertiser	• One way a malicious advertiser can financially hurt its competitor(s) is by generating a large number of fake impressions and/or clicks on the competitor' ads (e.g., by hiring 'human farms' or web bots).

Traffic fraud

  - *Action fraud* includes falsifying conversions, manipulating re-targeting data, and various forms of affiliate fraud.

Does What (Sub-Type of Fraud)		Who (Fraudster)	to Whom (Victim)	How (Objective)
Conversion Fraud		Dishonest Publisher/ Malicious Adver-tiser/Fraudster	Advertiser/User	• To complete a purchase, the user is typically required to fill out a form by providing personal information such as name, address and credit card number. A dishonest publisher could conduct conversion fraud is by filling these forms with fake or stolen customer information so as to inflate the number of conversions coming through their site.
Re-targeting Fraud		Fraudster/ Dishonest Publisher	Advertiser	• A fraudster or dishonest publisher can mislead an advertiser into believing that fake users (typically a group of 'pre-trained' web bots) are prospective purchasers and encourage them to put a higher bid price on impressions generated by these bots ultimately increasing their ad revenue.
Affiliate Fraud	Malware and Adware Cookie stuffing	Nefarious Affiliate	User/ Advertiser	• A nefarious affiliate creates a site/webpage that looks like the home page of a legitimate website and is hosted on a look-alike DNS/URL domain. During interaction with this fake site, the user's browsers gets stuffed with the nefarious affiliate's cookies, or the user ends up inadvertently downloading malware. Besides hurting the user and the advertiser, the existence of such a 'fake' site could also hurt the reputation and the profit of the legitimate (impersonated) publisher.
	URL Hijacking		User/Advertiser/ Impersonated Publisher	

Action fraud

This lens specifies best practices that address the unique characteristics of building and operating advertising workloads in the cloud. They are based on our experience with industry developers and operations teams. It provides guidance on how to design and operate your environment addressing the operational challenges.

This document is intended for those in technology roles, such as chief technology officers (CTOs), Technical directors, architects, developers, and operations team members. After reading this document, you will understand AWS best practices and recommended strategies to use when designing and operating architectures for advertising workloads.

Lens availability

Custom lenses extend the best practice guidance provided by AWS Well-Architected Tool. AWS WA Tool allows you to create your own custom lenses, or to use lenses created by others that have been shared with you.

To begin reviewing your advertising workload, download and import the Video Streaming Advertising Lens into AWS Well-Architected Tool from the public AWS Well-Architected custom lens GitHub repository.

Definitions

Digital advertising (ads) refers to marketing through online channels, like websites and streaming content, as well as linear channels, like cable television. Digital ads span media formats, including text, image, audio, and video. They can help you achieve a variety of business goals, ranging from brand awareness to customer engagement. They can also help you launch new products and drive repeat sales.

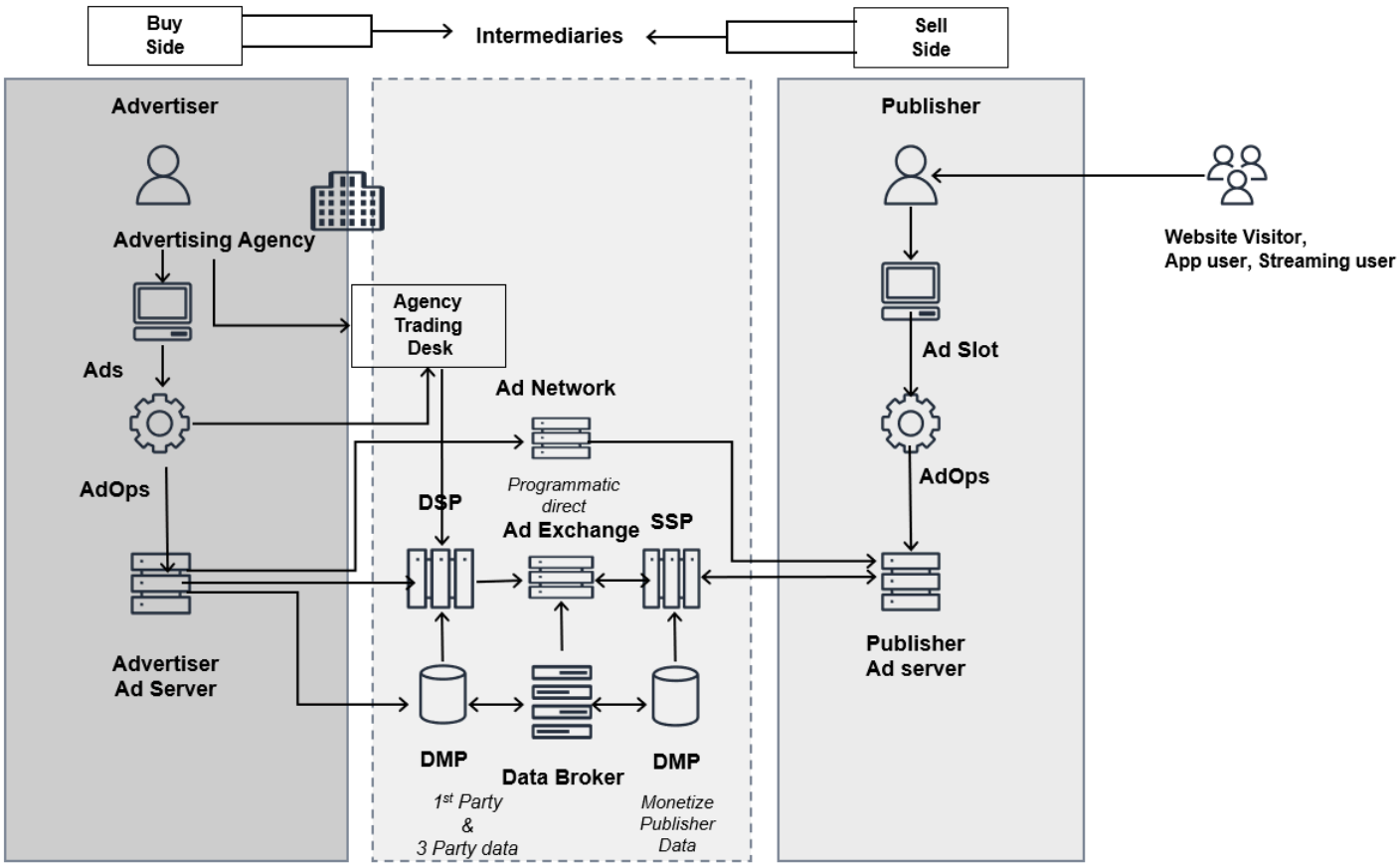
There are multiple types of advertising formats in digital advertising. The most common types are:

- **Search advertising:** Search ads, also called search engine marketing (SEM), appear in search engine results pages (SERPs). These are typically text ads that appear above or alongside organic search results.
- **Display advertising:** Ads that use text and visual elements. These can include images or animations and can appear on websites, apps, and devices. They appear in or alongside the content of a website.
- **Online video advertising:** Ads that use a video format. Out-stream video ads appear in places similar to display ads, like on websites, apps, and devices. In-stream video ads appear before, during, or after video content.
- **Streaming media advertising:** Also known as over-the-top (OTT), these are a specific type of video ad that appears in streaming media content delivered over the Internet without satellite or cable.
- **Audio advertising:** In the context of digital advertising, audio ads are ads that play before, during, or after online audio content, such as streaming music or podcasts.
- **Social media advertising:** Ads that appear on social media platforms.

Scope

This lens focuses on programmatic advertising, specifically online video advertising and streaming media advertising. This document provides best practices applicable to supply-side platforms (SSPs), ad exchanges, demand-side platforms, and related programmatic advertising platforms. It complements the best practices for ad supported content monetization (a part of the Streaming Media Lens for publishers).

The following programmatic advertising lifecycle describes the complex end-to-end interaction between a user, the publication used, the selling and buying platforms, and the advertiser.



Online video advertising logical environment

On the Buy Side (or the advertiser), the personas are:

- **Advertiser:** A person, organization, or company that places free or paid promotion of a specific product, event, service in a public medium to attract potential or new customers
- **Agency:** An organization that, on behalf of clients, plans marketing and advertising campaigns, drafts and produces advertisements, and places advertisements in the media. In interactive advertising, agencies often use third party technology (ad servers) and may place advertisements with publishers, ad networks and other industry participants.
- **DSP:** Demand-side platforms provide centralized (aggregated) media buying from multiple sources, including ad exchanges, ad networks, and sell-side platforms, often bidding in real time. While there is some similarity between a DSP and an ad network, DSPs are differentiated from ad networks because they do not provide standard campaign management services, publisher services, or direct publisher relationships.

On the Sell Side (or the publisher), the personas are:

- **Publisher:** A person or company that makes content (in any form) available for consumption, for free or for sale
- **Ad network:** Ad networks provide an outsourced sales capability for publishers and a means to aggregate inventory and audiences from numerous sources in a single buying opportunity for media buyers. Ad networks may provide specific technologies to enhance value to both publishers and advertisers, including unique targeting capabilities, creative generation, and optimization. Ad networks' business models and practices may include features that are similar to those offered by ad exchanges.
- **SSP:** Sell-side platforms provide outsourced media selling and ad network management services for publishers. Sell-side platform and ad networks business models and practices are similar. Sell-side platforms are typically differentiated from ad networks in not providing services for advertisers.
- **Ad server:** An ad server is a technological platform that manages, delivers, and tracks online ads. Ad servers are used by publishers, advertisers, and ad networks.

The technology Intermediary layer that connects the Buy and Sell Sides is:

- **Ad exchange:** An ad exchange provides a sales channel to publishers and ad networks, as well as aggregated inventory to advertisers. They bring a technology platform that facilitates automated auction-based pricing, selling, and buying in real-time.
- **Data management platform (DMP):** A DMP is an aggregator of third-party data. This data is made available on a commercial basis to enrich the ad supply of publishers, expand the attributes available for audience segmentation, and expand targeting by advertisers. The data is also available to advertisers to expand the attributes available to their users and widen campaign definitions.
- **Customer data platform (CDP):** A CDP is a data solution that centralizes customer related data, so that retailers can link and bridge multi-channel customer journeys through one identity profile, which assists publishers to understand the customer needs demands and make better decisions.

Additional industry definitions

- **Beacon:** Tracking pixels or tags that are used to measure ad performance, attribution, and analytics.

- **Click:** A metric that is counted every time someone clicks on an ad, even if the person doesn't reach the website (such as when it's temporarily unavailable).
- **Fill rate:** Ad impressions generated in the ad space divided by the total inventory available on that ad space.
- **Impression:** An ad view is counted every time a creative (ad) is served to a user. The impression is considered one of the basic campaign performance indicators.
- **Industry advertising bureau (IAB):** An advertising business organization that develops industry standards, conducts research, and provides legal support for the online advertising industry.
- **OpenRTB:** OpenRTB is a standard of communication between buyers and sellers of real-time bidding (RTB) advertising, allowing advertising platforms (demand-side platforms, supply-side platforms, and ad exchanges) to speak the same language when conducting online media transactions.
- **Real-time bidding (RTB):** The process of purchasing and selling digital ad space through real-time auctions that occur in the time it takes a webpage to load
- **Programmatic advertising:** The buying and selling of online media via automated systems (advertising platforms).
- **VAST (video ad-serving template):** An XML schema developed by the IAB that allows in-stream video ads to be served from video ad servers and played in video players across a number of websites, or publishers, and on numerous devices (for example, desktop computers, mobile devices, or tablets).

List of AWS services

The following are the AWS services part of the implementation best practices referenced.

AWS services referenced in this lens
AWS Systems Manager
AWS CloudFormation
Amazon CloudWatch
AWS OpenSearch
AWS X-Ray

AWS services referenced in this lens
AWS Step Functions
AWS Lambda
AWS Identity and Access Management
AWS Certificate Manager
Elastic Load Balancing
Network Load Balancer
Application Load Balancer
AWS WAF
Amazon API Gateway
Amazon Virtual Private Cloud
Amazon AppFlow
AWS PrivateLink
AWS Transfer Family
AWS Data Exchange
Amazon Kinesis
Amazon CloudFront
AWS Network Firewall
AWS Direct Connect
AWS Shield
Amazon EventBridge

AWS services referenced in this lens
Amazon Simple Notification Service (SNS)
Amazon Simple Queue Service (SQS)
AWS Resilience Hub
AWS Fault Injection Service (AWS FIS)
Amazon ElastiCache
Amazon EC2 Auto Scaling
Amazon Elastic Kubernetes Service (EKS)
AWS Control Tower
AWS Managed Microsoft AD
AWS KMS
Amazon S3
Amazon Relational Database Service (Amazon RDS)
Amazon DynamoDB
Amazon Aurora
Amazon Route 53
Amazon DynamoDB Global Tables
AWS CodePipeline
Amazon Elastic Container Service (ECS)
AWS Elastic Disaster Recovery (DRS)
AWS Config

AWS services referenced in this lens
AWS Backup
AWS Key Management Service (KMS)
Amazon Elastic Compute Cloud (EC2)
Amazon Elastic Block Store (EBS)
Amazon Elastic File System (EFS)
Amazon SageMaker AI
Amazon Managed Streaming for Apache Kafka (MSK)
Amazon S3 Express One Zone
AWS Lambda@Edge
Quick
Amazon Athena
AWS Cost Explorer
Amazon Cost and Usage Reports (CUR)
AWS Pricing Calculator
AWS Graviton
Amazon Fargate
Amazon Kinesis Data Streams
AWS Serverless services
AWS Customer Carbon Footprint Tool
Amazon EBS volumes

AWS services referenced in this lens
AWS Compute Optimizer
AWS Trusted Advisor
AWS Data Transfer Cost Estimator
AWS Global Accelerator

Design principles

The following design principles can help you achieve and maintain efficient advertising workloads in the cloud.

- **Design for low latency and rapid changes in traffic volume:** Build a scalable architecture with automated scaling capacity to enable rapid increases and decreases of traffic gracefully. Use load balancers to distribute traffic to new nodes available to scale horizontally. Implement auto scaling based on application node metrics. Cache application data content, when possible, to reduce response latency and the load on database clusters. Use containerized workloads and prebuilt container images for fast scaling and predictable performance. Choose server hardware optimized for memory and CPU for ultra-low latency needs.
- **Design for rapid growth, very large data volumes, high QPS (queries per second), and low latency transactions across multiple Regions:** Build a scalable distributed database for transactions while optimizing it for fast writes. Consider use of a distributed NoSQL database that can handle high write throughput with linear scalability. Consider compression techniques to optimize storage and an appropriate caching strategy to reduce database load for user profiles, target segments, and creatives. Use streaming services for ingestion and transportation of event data. Set up auto scaling databases to handle traffic spikes. Implement a data archive strategy to purge old ad impressions data to less expensive storage. Monitor database performance metrics, including latency, timeouts, and saturation, to identify and fix bottlenecks.
- **Design for cost optimization to reduce costs while maintaining performance:** The key to cost optimization for advertising workloads is to minimize costs while you maintain performance and reliability. Optimization efforts should focus on minimizing unnecessary traffic charges and providing sufficient but not excessive capacity through auto scaling. The main drivers of advertisement costs are data transfer, compute, storage, and networking. Considering the large amounts of traffic involved in campaigns, even small changes can yield significant cost optimization.
- **Design for scaled, cost-optimized, and performant data pipelines:** Use event-driven architecture and pre-compute data where possible to reduce real-time compute needs. Design for horizontal scalability rather than optimizing within a single host. For large-scale systems, consider distributed service-oriented architecture to allow components to innovate independently. Use caching solutions at scale, with multiple replicas for high availability and data recovery. Implement real-time introspection and debugging capabilities for system transparency. When dealing with massive data volumes, consider batching and direct writing to object storage instead of using intermediary services. Use a combination of real-time streaming

and batch analytics processing for data processing, based on specific use cases. Implement a centralized data lake for aggregating data from various sources with a data catalog. Use tiered storage solutions to balance cost and performance for data storage. Reuse compute resources like analytics clusters, to reduce costs and startup times. Retain all event-level data and logs for less than 30 days.

- **Design for privacy-enhanced collaboration:** Consider the following strategies to address third-party signal loss through linked first-party data, browser and OS-mediated data, and unlinked first-party data. Use linked first-party data design, where advertiser and publisher data are connected through privacy-enhancing technologies (PETs), enabling personalized customer engagement and targeting while maintaining privacy. Use browser and OS-mediated data design where web browsers or operating systems act as intermediaries for user data collection and sharing. It protects user privacy by limiting direct access to personal information, instead of providing aggregated or anonymized data to advertisers and websites. Use contextual targeting to deliver relevant ads without relying on user data and seller-defined audience solutions which allow publishers to create targetable segments based on their first-party data. Implement first-party and third-party identity matching and identity transcoding that use trusted and compliant runtime environments like data clean rooms and confidential computing environments.
  Design for robust access controls, audit trails, and secured data exchange mechanisms for your organization and partners. Design for alignment on file formats, APIs, communication protocols, and data schemas, ideally with standardized industry formats as defined by the Interactive Advertising Bureau (IAB).
- **Design for advertising regulatory, security, and privacy needs:** Design for robust access control and network security measures, complemented by industry-standard ad fraud detection software and anti-fraud tools. Design for monitoring and assessment of ads to facilitate policy compliance and malicious content detection, while maintaining strong consumer privacy protection, and managing consent and data rights. Consider regulatory standards involved for ad formats and creative standards (IAB), measurement standards (MRC, IAB), privacy and consent standards (IAB, regional regulations for example, China, CCPA, and GDPR), programmatic advertising standards (IAB, openRTB consortium), brand safety and fraud prevention (TAG, IAB), identity and attribution (U-ID2.0, rampID, ID5, MTA) and specific ad standards published by Google, Meta, and Amazon.

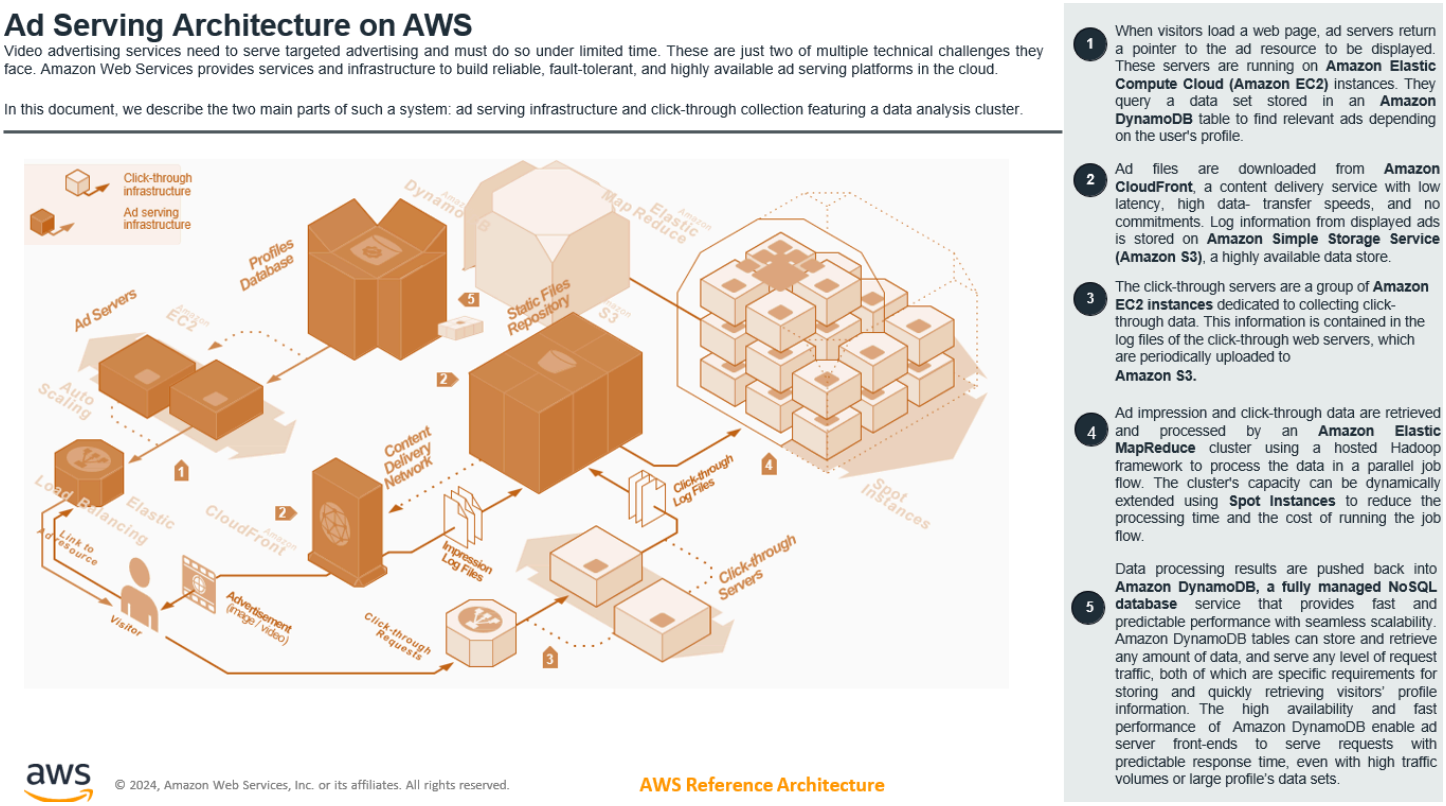
Scenarios

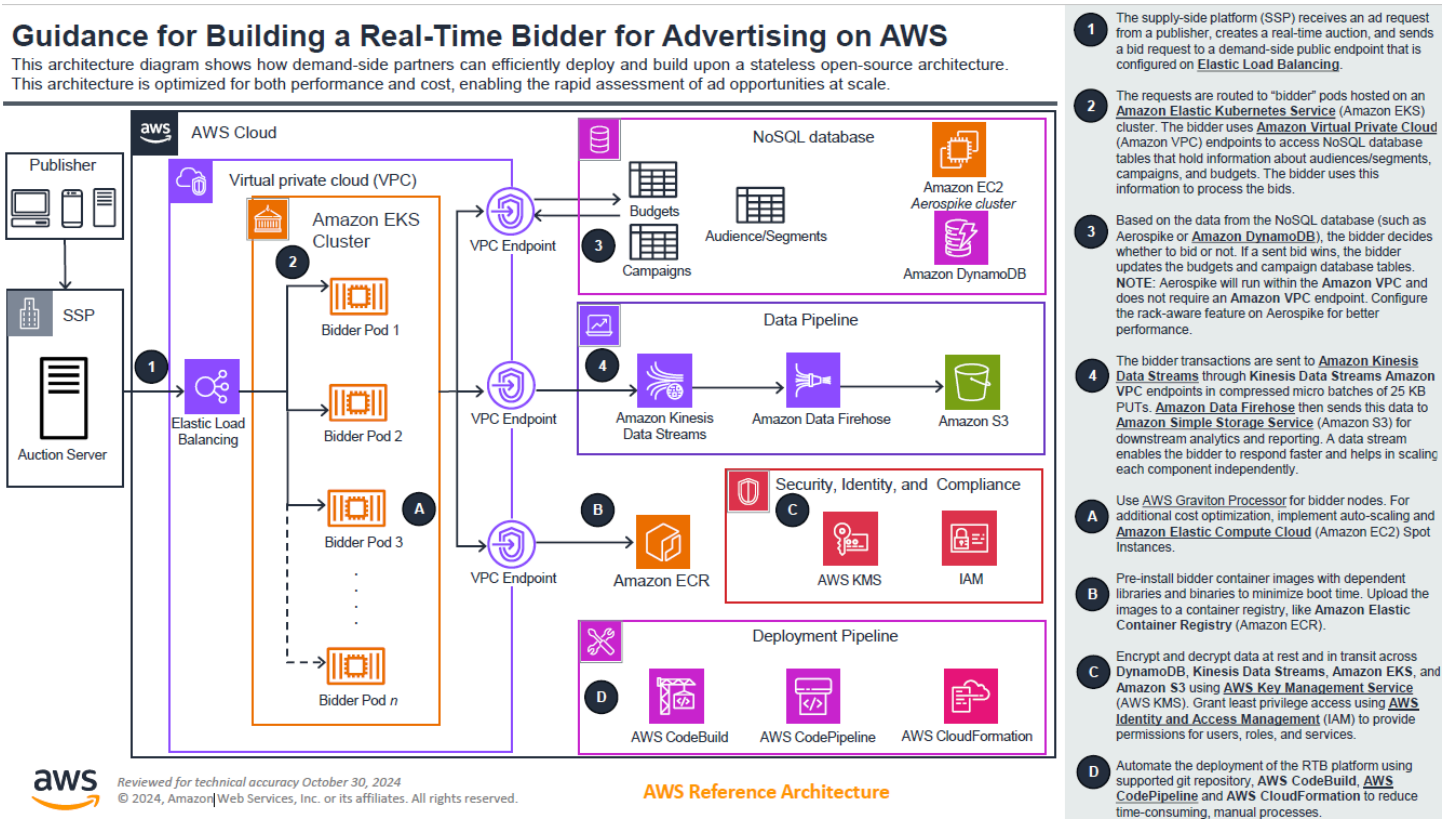
The following are common scenarios that can influence the design and architecture of your advertising workloads on AWS. This section covers the common drivers for the design, along with prescribed reference architecture.

Scenarios

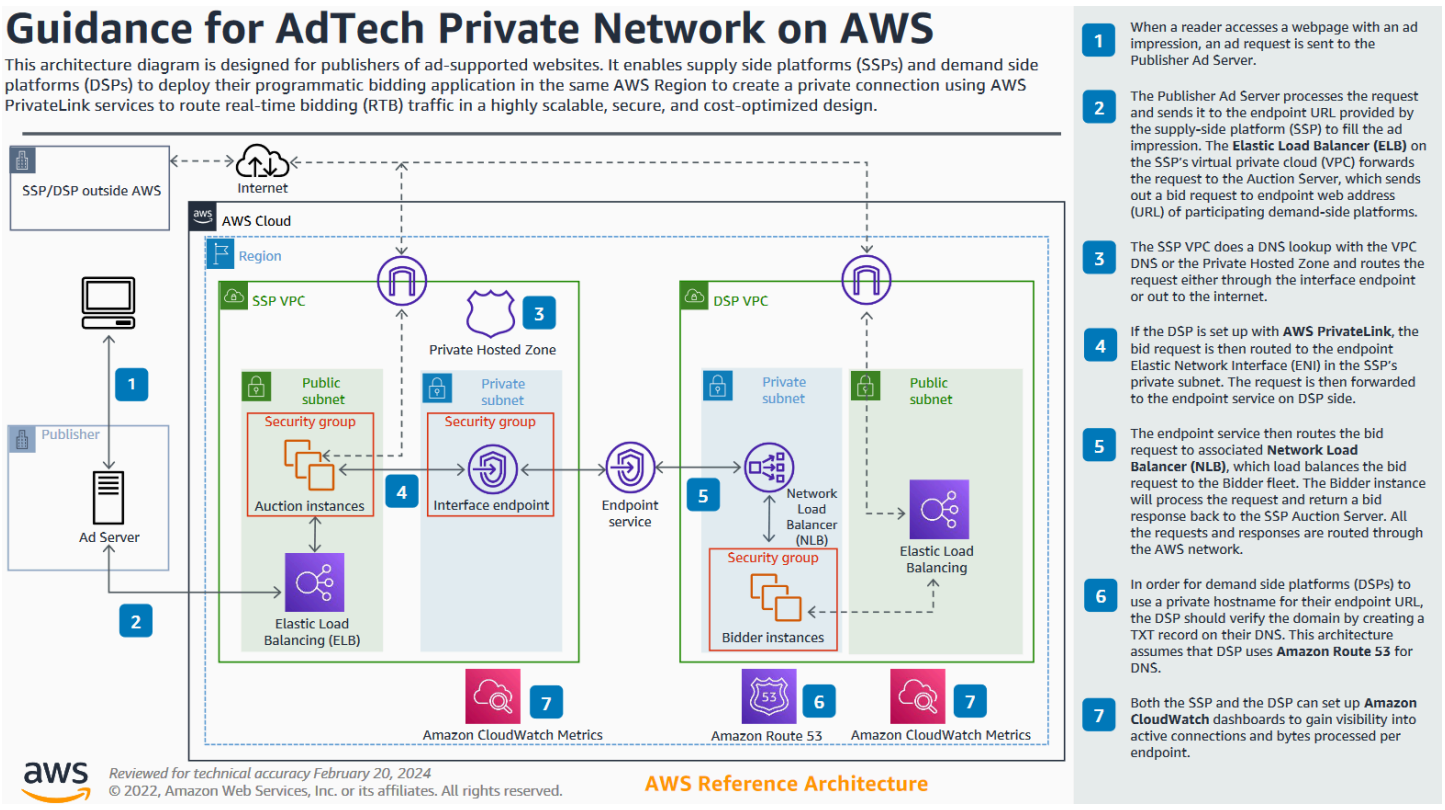
Programmatic advertising workloads

The petabyte-scale, millisecond latency world of programmatic advertising brings significant challenges for performance, throughput, latency, and costs to advertising platforms. The key platforms in programmatic advertising are ad servers, ad exchanges and SSPs, and DSPs.





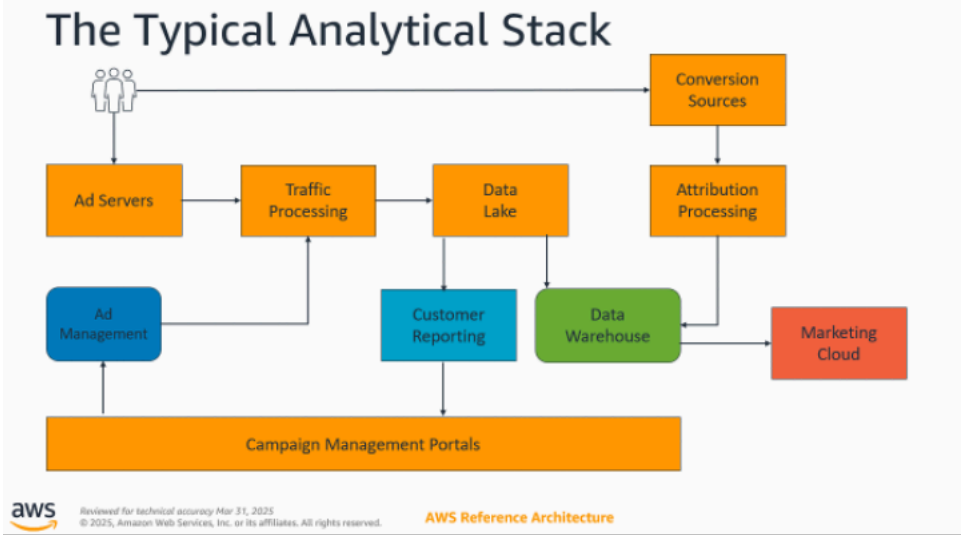
For additional detail, see Guidance for Building a Real Time Bidder for Advertising on AWS.



For additional detail, see Guidance for AdTech Private Network on AWS.

Data pipelines architecture

The typical analytical logical architecture is depicted as follows:

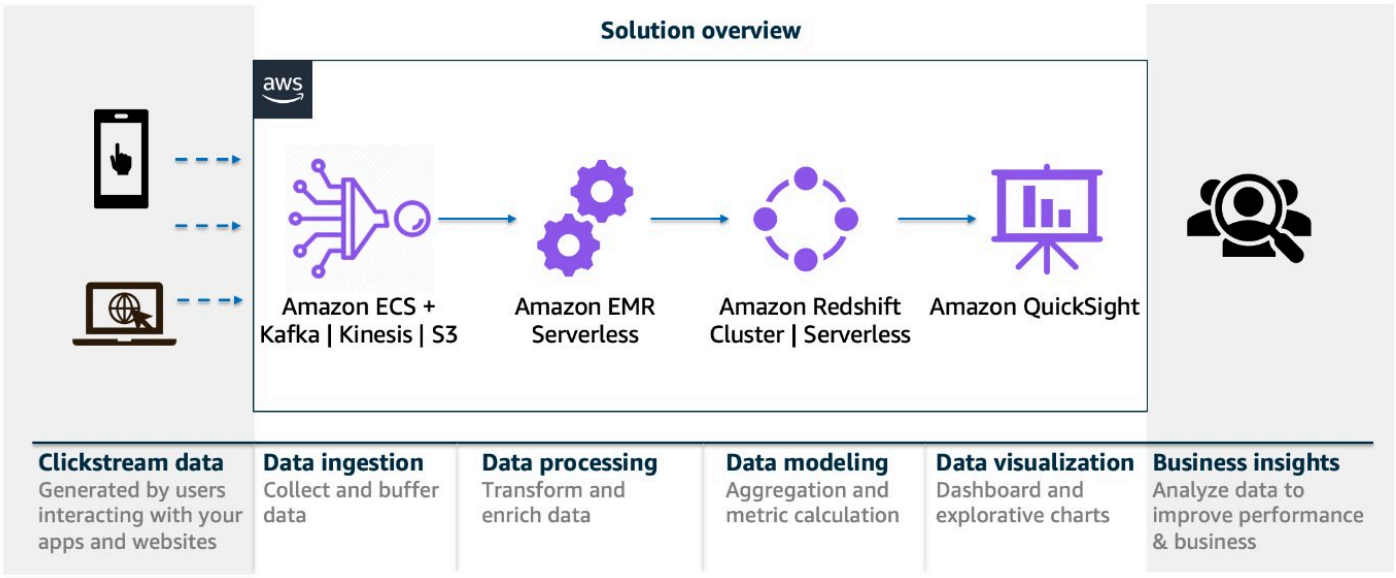


The key objectives and best practices for this architecture are as follows:

- **Data ingestion from source and data standardization:** Involves preprocessing and moving of events and ad server data from sources and storing them in a landing zone. The key objectives are to preprocess data to common format, control data quality, ingest and store data in landing zone, and separate data for downstream systems. Best practices are to aggregate data in a single Region for cost reduction, certify data completeness and register new data, convert to standard format before storing into the landing zone, and store raw data for max 30 days for compliance alignment.
- **Measurement, exploration, and data science:** Processing anonymized, aggregated data for machine learning and business intelligence. The key objectives are to prepare data for downstream consumption, enrich data with dimensions, and transform for loading into fact spaces. Best practices are to use micro-batches for scale, avoid raw data, store aggregates, preprocess for downstream use, implement data catalog and partitioning for efficient querying.
- **Advertising data warehouse (ADW):** An engine for querying of data when necessary. The key objectives are to support research questions, serve intake as data source for ML workloads, and store partitioned campaign metrics. Best practice is to load aggregates and dimensions from your data lake.

- **Reporting:** Campaign and ads performance reporting. The key objectives are to support low latency (one to two seconds), handle continually updated data, and support multiple reporting granularity. Best practice is to use indexes or search engine like OpenSearch.
- **Attribution and event-level data:** Processing and storing event-level data that may include transaction details, behavioral data and PII. The key objectives are to enable event-level data storage or querying, integrate with analytical pipelines and process in prescribed geographic regions. Best practices are to use near real-time stores like DynamoDB, use aggregates instead of PII where possible, and implement data handling per regulations.

Data pipeline solution guidance



caption text

This solution provides the implementation details for a data pipline that uses AWS services.

Ingestion module

The ingestion module serves as web server for ingesting the Clickstream data.

Data processing module

The data processing module transforms and enriches the ingested data to solution's data model by the Apache Spark application running in EMR serverless.

Data modeling module

The data modeling module loads the processed data in a lake house. It supports the following features:

- Support both provisioned Redshift and Redshift Serverless as data warehouse
  - Users can specify the time range for storing data in Redshift
  - Users can specify the interval to update user dimension table
- Support use Athena to query the data in data lake

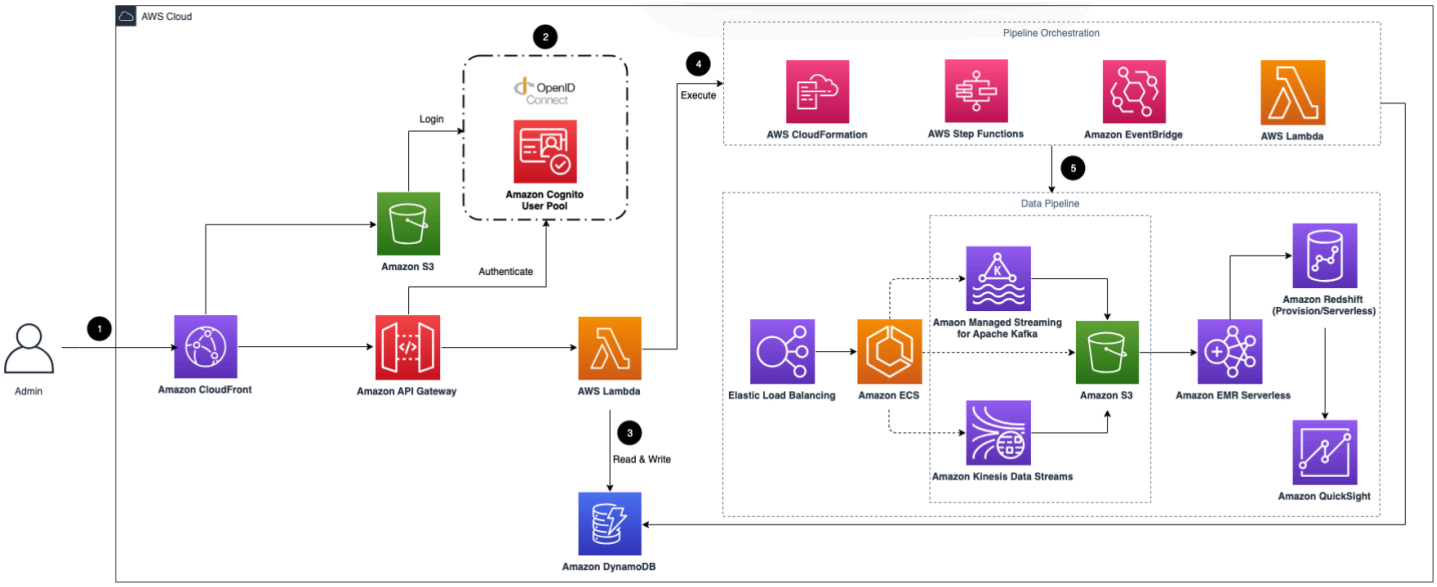
Reporting module

The reporting module creates a secure connection to the data warehouse and provisions the out-of-the-box dashboards in business intelligence Amazon Quick.

For more information, see Guidance for Clickstream Analytics on AWS.

Clickstream analytics on AWS

Clickstream analytics on AWS collects, ingests, analyzes, and visualizes clickstream data from your websites and mobile applications. Clickstream data is critical for analyzing user behavior, customer data, and marketing campaigns. This data derives insights into the patterns of user interactions on a website or application, better understanding of user navigation, preferences, and engagement levels to drive product innovation and optimize marketing investments.

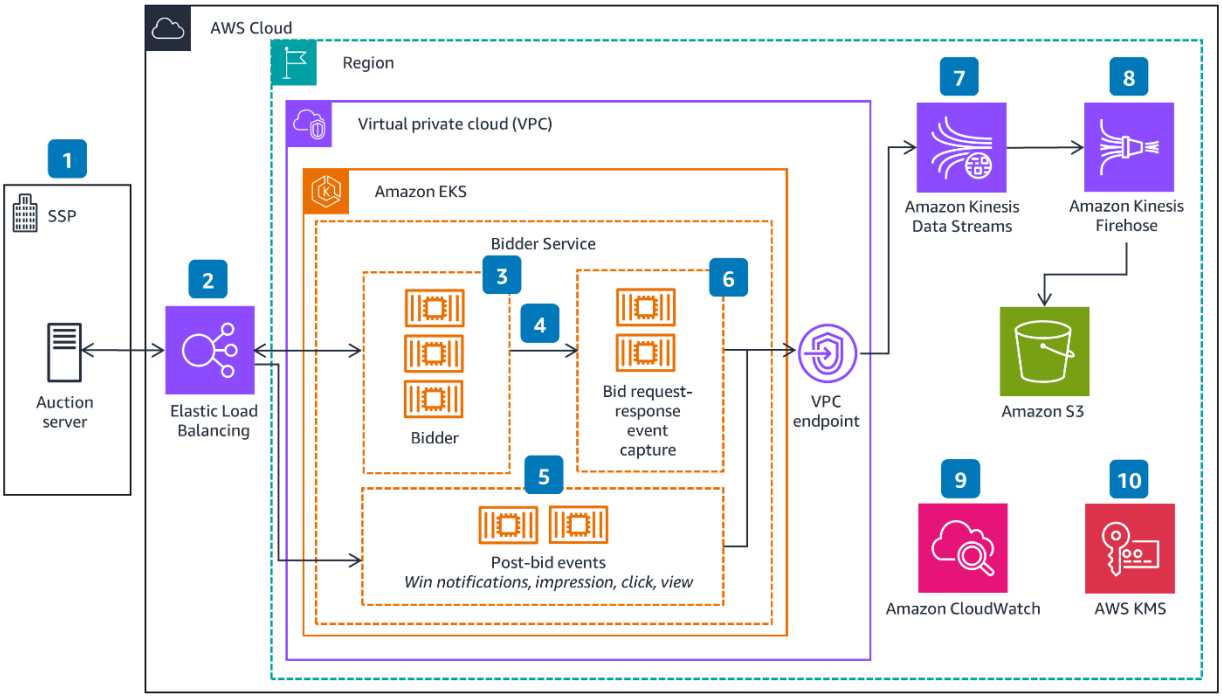


- **Step 1:** Amazon CloudFront distributes the frontend web user interface (UI) assets hosted in the Amazon Simple Storage Service (Amazon S3) bucket and the backend APIs hosted with Amazon API Gateway and AWS Lambda.
- **Step 2:** The Amazon Cognito user pool, or OpenID Connect (OIDC), is used for authentication.
- **Step 3:** The web UI console uses Amazon DynamoDB to store persistent data.
- **Step 4:** AWS Step Functions, AWS CloudFormation, **Lambda**, and Amazon EventBridge orchestrate the lifecycle management of data pipelines.
- **Step 5:** The data pipeline is provisioned based on the configurations specified by the user in the web console. It consists of Application Load Balancer (ALB), Amazon Elastic Container Service (Amazon ECS), Amazon Managed Streaming for Apache Kafka (Amazon MSK), Amazon Kinesis Data Streams, **Amazon S3**, Amazon EMR Serverless, Amazon Redshift, and Amazon Quick Sight.

For more information, see Guidance for Clickstream Analytics on AWS.

RTB event capture solution guidance

This guidance assists ad-tech companies capture open real-time bidding (OpenRTB) events and establish a foundation for near real-time and batch analytics. During a programmatic advertising transaction, a demand-side platform (DSP) service generates a series of events. Capturing these events assists ad-tech companies keep their budgets updated and understand signals for optimizing the bid response. By tracking events, such as a successful win bid, advertisers can better measure the effectiveness of their campaigns. They can also analyze these events to make informed decisions for future bids.



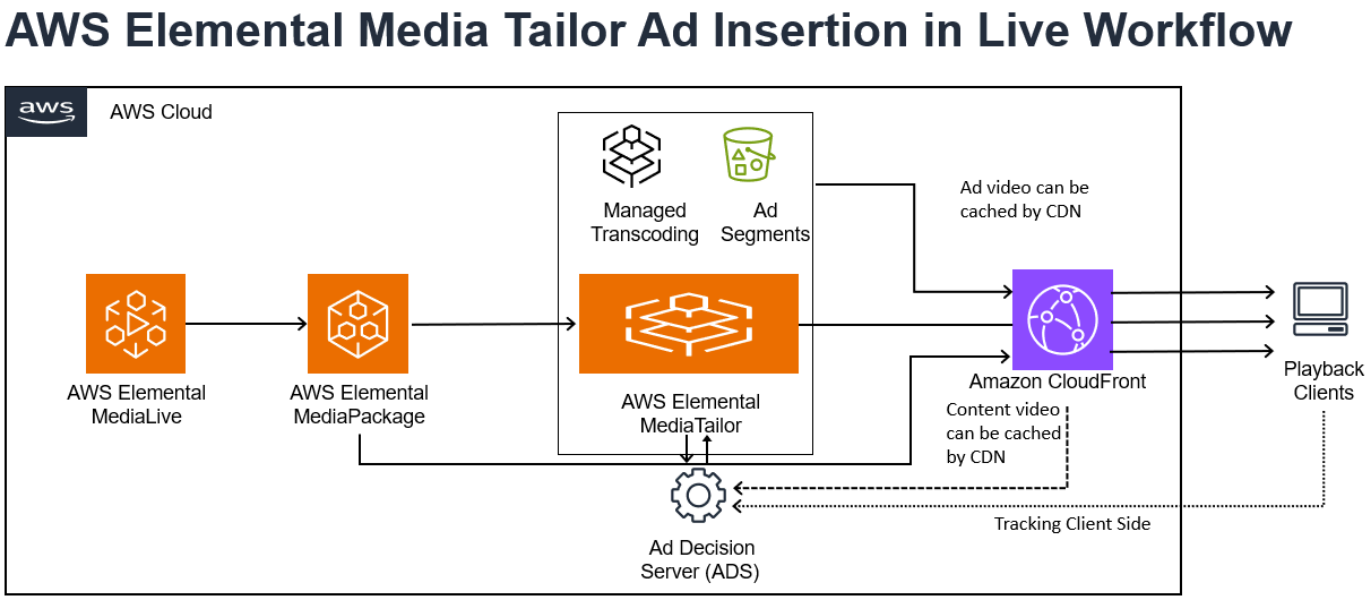
- **Step 1:** The supply-side platform (SSP) receives an ad request from a publisher and launches an auction.
- **Step 2:** An OpenRTB bid request is sent to a DSP public endpoint that is configured on an Elastic Load Balancer.
- **Step 3:** The bidder application service receives the bid request. This service runs on Amazon Elastic Kubernetes Service (Amazon EKS) within an Amazon Virtual Private Cloud (VPC).
- **Step 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.
- **Step 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.
- **Step 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.
- **Step 7:** The event messages are routed to **Kinesis Data Streams** through a dedicated VPC endpoint

- **Step 8:** Amazon Data Firehose consumes these aggregated records and de-aggregates and sends individual events to Amazon Simple Storage Service (Amazon S3) for long-term storage and analytics.
- **Step 9:** Amazon CloudWatch captures application logs for traceability.
- **Step 10:** AWS Key Management Service (AWS KMS) stores and manages encryption keys used for securing persisted data in Kinesis Data Streams and Amazon S3.

Additional reference available here

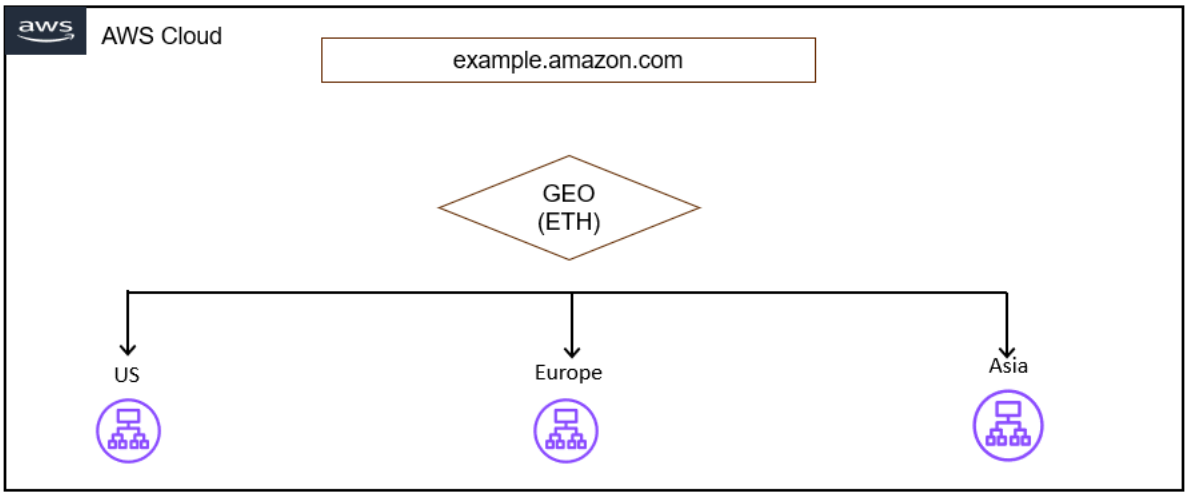
Multi-Region advertising workloads

The section expands on ad decision service (ADS) invoked by the publisher ad insertion workflow provided in the following diagrams. This comprises of SSP, ad network, Ad exchange, DSP, and DMP systems as part of the direct and programmatic ad serving process.



ADS intake traffic distribution

1. Using Route 53 geolocation or geo proximity routing policies, the intake traffic from publisher can be routed to the ADS regional workloads with a continent based on data residency requirements for that continent. The example referenced in the following diagram shows intake traffic routed to US, Europe, and Asia based on the geolocation of the intake origin from the publisher.

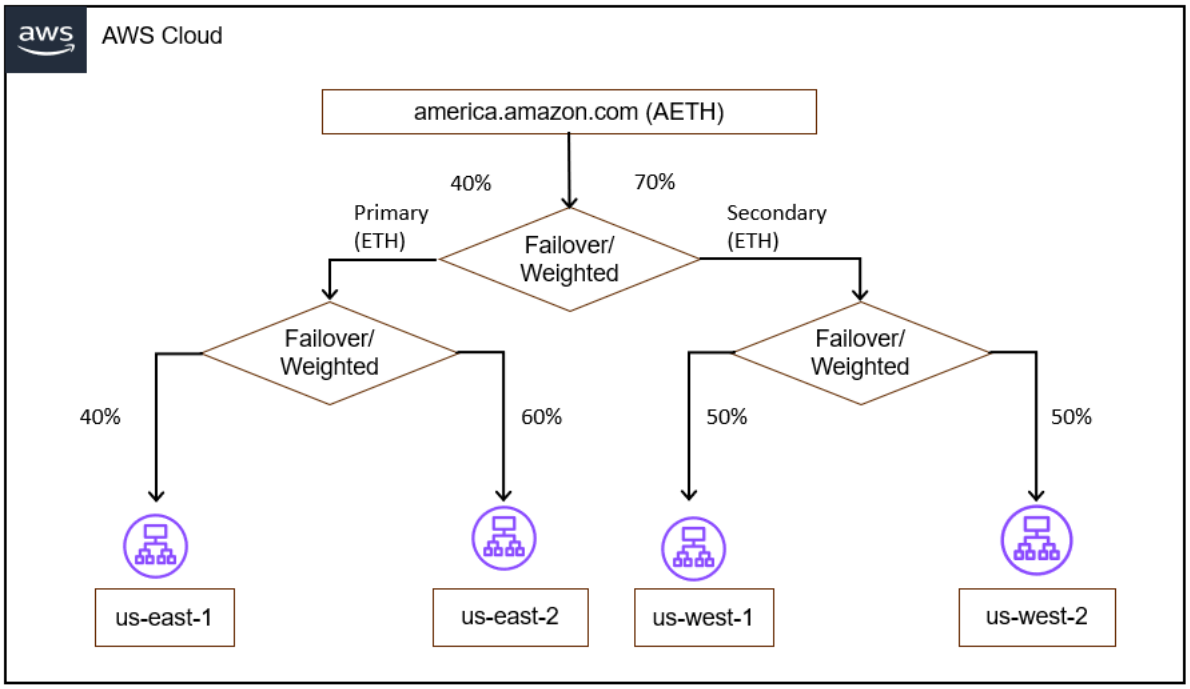


2. Furthermore, for traffic routing within a content across the various AWS Regions, you can use the weighted routing policy of Amazon Route 53. This can be also combined with the failover routing policy of Route 53 to implement resiliency feature.

   For example, in the following diagram for the intake traffic routed to the America continent:

   a. Traffic is distributed between us-east and us-west regions in ratio of 70:30. In case of a Regional failure, traffic will be routed 100 % to the other region.

   b. The same combination of weighted is implemented to split traffic between us-east-1 and us-east-2 in the ratio of 40:60, & between us-west-1 and us-west-2 in the ratio of 50:50.



c. Traffic pattern between an ad network and DSP for programmatic bidding is local within the boundaries of a Region due to latency concerns.

d. DSP patterns are replicated at a Regional boundary, and the calls from the ad network to DSP are made over the internet through RTB protocol to DSP load balancers.

Data management considerations

Low-latency data (real-time or near real-time): Bid data, user data, ad impression data, and real-time campaign performance data (clicks, impressions, and conversions). This data needs to be processed and acted upon within milliseconds to facilitate optimal ad delivery, real-time bidding, and accurate tracking of user interactions.

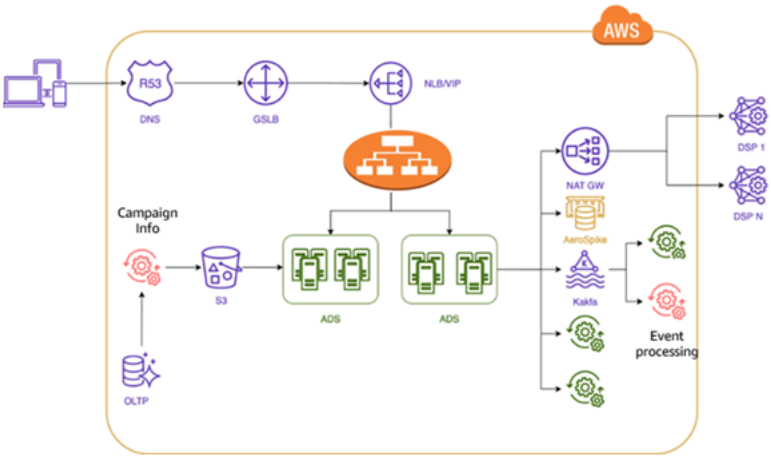
Medium-latency data (near real-time or batch processing): User behavior data, audience segmentation data, campaign optimization data, and attribution data. This data can be processed in near real-time (within minutes or hours) or in batches, as it is used for audience targeting, campaign optimization, and attribution analysis.

High-latency data (batch processing or offline): Historical campaign data, third-party data, and ad creative data. This data can be processed in batches or offline, as it is typically used for analysis, reporting, and long-term decision-making rather than real-time ad delivery or optimization.

Data is confined within a continent for data residency requirements and replicated cross-AZ and cross-Region within the continent.

Campaign data for frequency capping perspective is replicated globally.

The ad serving workload is replicated in each Region with data stores for local and global storage.



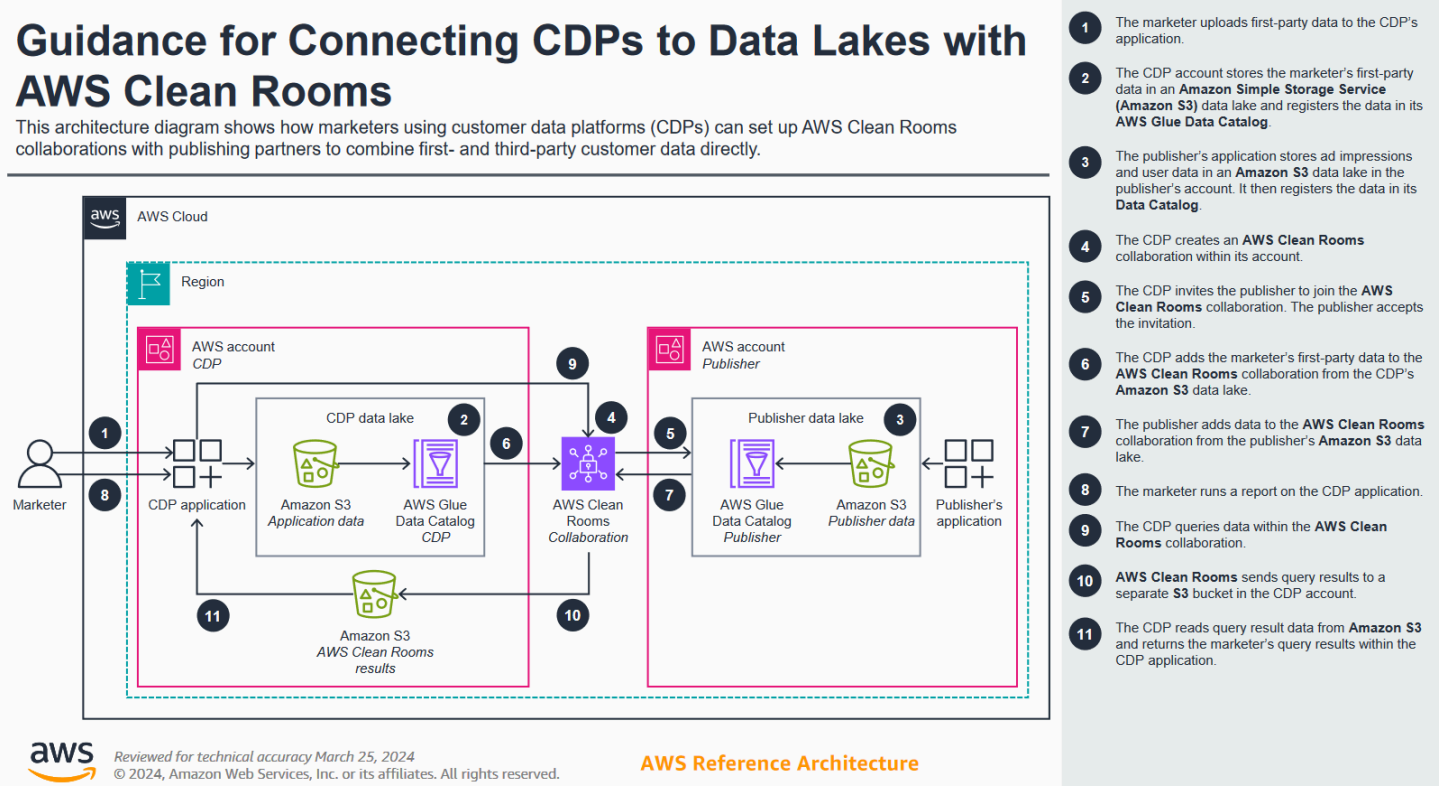
Ad serving workload at a Regional level with interaction with 1: many DSPs at a Region

Using privacy-enhanced collaboration

This scenario covers the best practices for collaboration between advertising, publishers and agencies, to improve campaign planning, activation, and measurement while protecting consumer data privacy.

AWS Clean Rooms implementation connecting marketing and publisher customer data platform

This guidance shows you how to use customer data platforms (CDPs) to set up a collaboration between first-party marketing data and third-party data from a publishing partner. By using an AWS Clean Rooms collaboration, CDPs can facilitate the connection between separate data lakes on AWS. Marketers can upload their data to the CDP application, then use the application to run reports from the compiled data, and activate their audiences.



Browser and OS-mediated campaigns

There are three privacy sandbox APIs that have dependencies to back-end services that the adtech companies need to host on a public cloud.

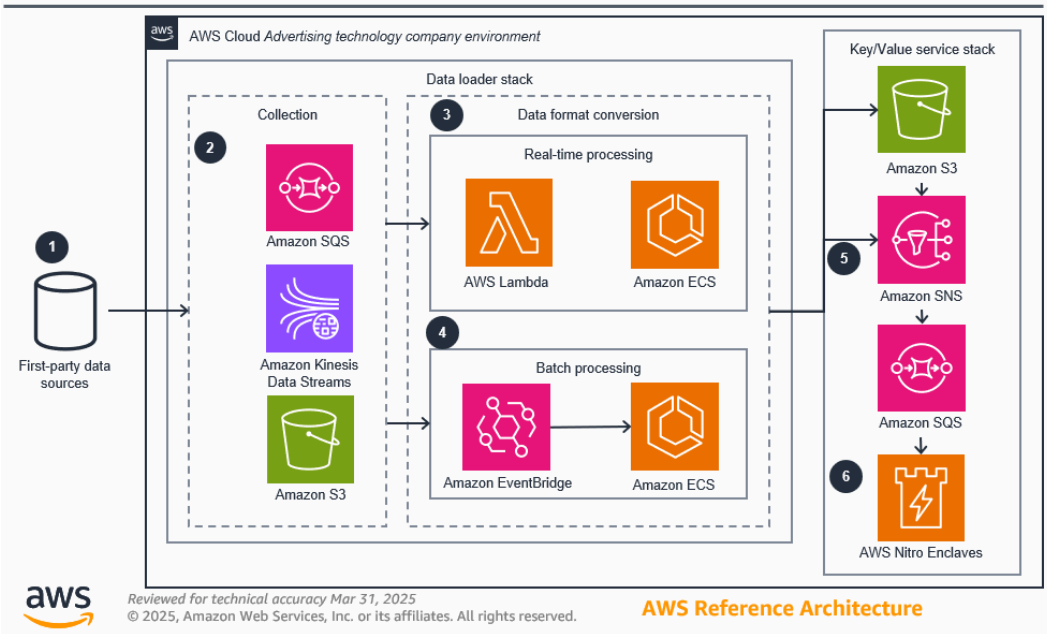
- **Topics API:** With this API, each publisher or the adtech acting on behalf of the publisher can use the topics API to add a particular browser or device to an interest group. Chrome will not share the unique identifier of the device or browser. But a publisher can use first party cookies to collect signals around user interaction on their property and identify them being part of 1 or more interest groups. This is then available to the bidders. On the demand side, DSP's will only get to see the interest groups and they have to decide to bid or not and how much to bid for based on this.
- **Protected audience API:** The associated cloud hosted service is a Key value server. During the auction or bidding process SSP's and DSP's respectively need to access data that is to be used in auction logic or bidding logic. Adtech customers have to run a key value server on Nitro enclaves to store and retrieve data to be compatible with protected audience API.
- **Measurement API:** Private aggregation and attribution reporting API and the associated cloud service is private aggregation service.

Hosting protected audience API

This guidance demonstrates how to deploy the Google Chrome Privacy Sandbox Key or value service within a trusted execution environment (TEE) on AWS. The key or value service allows implementers to fetch real-time signals to inform remarketing to custom audiences through the protected audience API (PAAPI). This real-time data assists ad buyers determine how to bid and assists sellers to pick winning bids in a privacy-enhanced way. This guidance intends to simplify the implementation of the Key/Value service while optimizing cost and latency.

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

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.

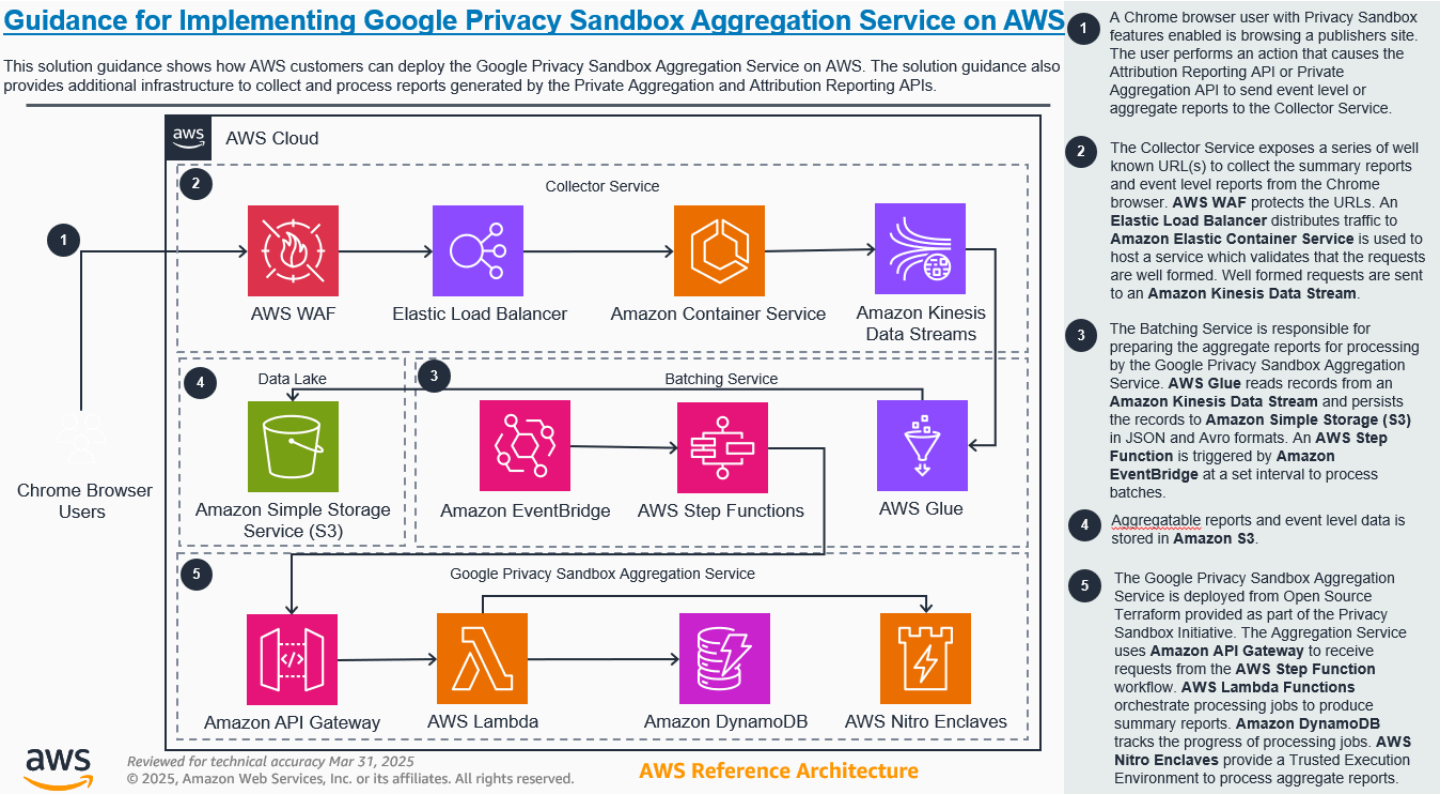


1. Your first-party data likely exists in multiple formats, and your system of records is part of your existing real-time bidding application. To support ad targeting through Privacy Sandbox APIs, your first-party data needs to be generated or copied to Privacy Sandbox's Protected Audience API Key/Value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

For additional details, see Guidance for Implementing Google Privacy Sandbox Key/Value Service on AWS.

Measurement API

This guidance demonstrates how to deploy the Google Privacy Sandbox Aggregation Service within a trusted execution environment (TEE) using AWS services. The Aggregation Service can be used to produce event or aggregate campaign measurement data through the Privacy Sandbox Attribution Reporting API (ARA) or Private Aggregation API.



For additional details, see [Guidance for Implementing the Google Privacy Sandbox Aggregation Service on AWS].

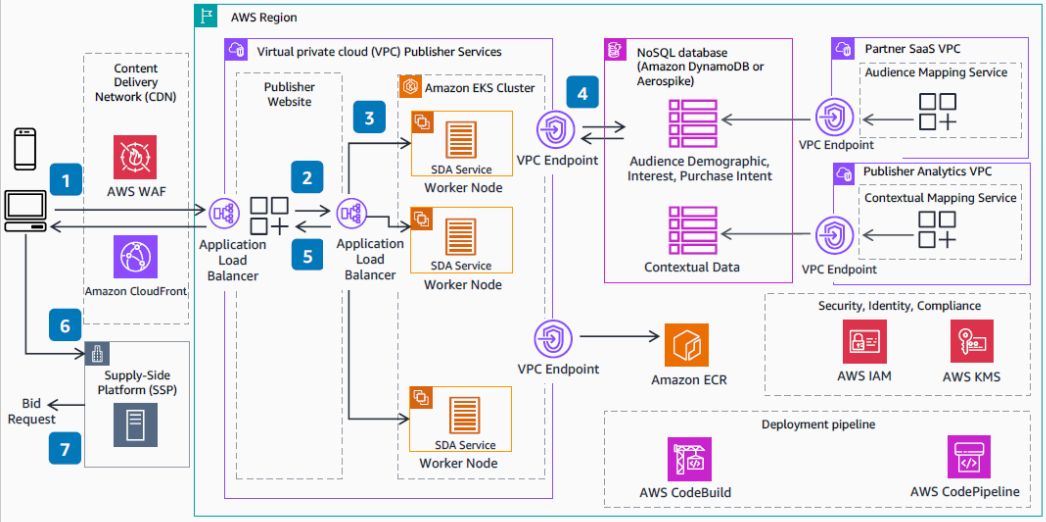
Activating seller-defined audiences

This guidance 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 to industry standard taxonomies, returning the associated SDA identifications for activation through real-time bidding (RTB).

Guidance for Activating Seller Defined Audiences on AWS

Page 1 of 2

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).

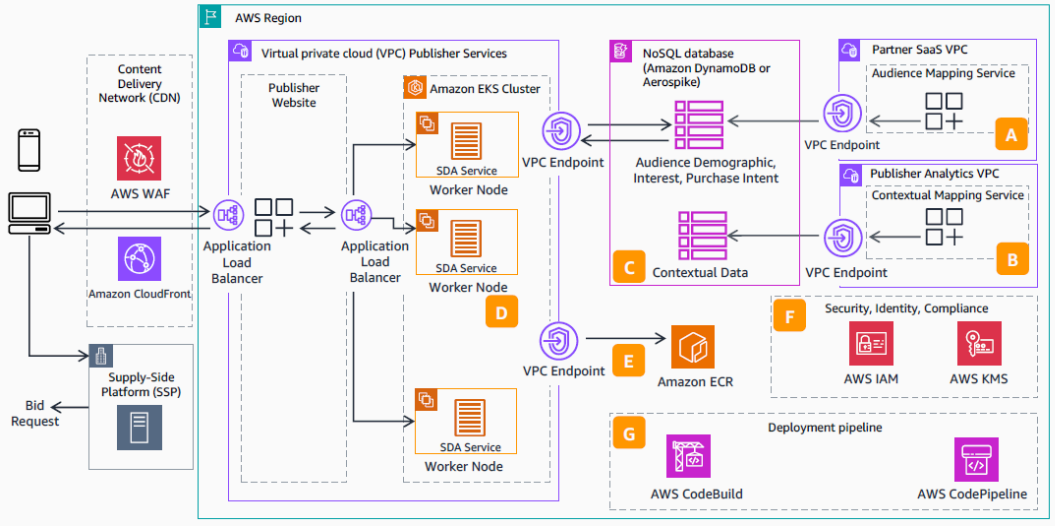


AWS Reference Architecture

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.

Guidance for Activating Seller Defined Audiences on AWS

Page 2 of 2



AWS Reference Architecture

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.

For additional details, see Guidance for Activating Seller Defined Audiences on AWS.

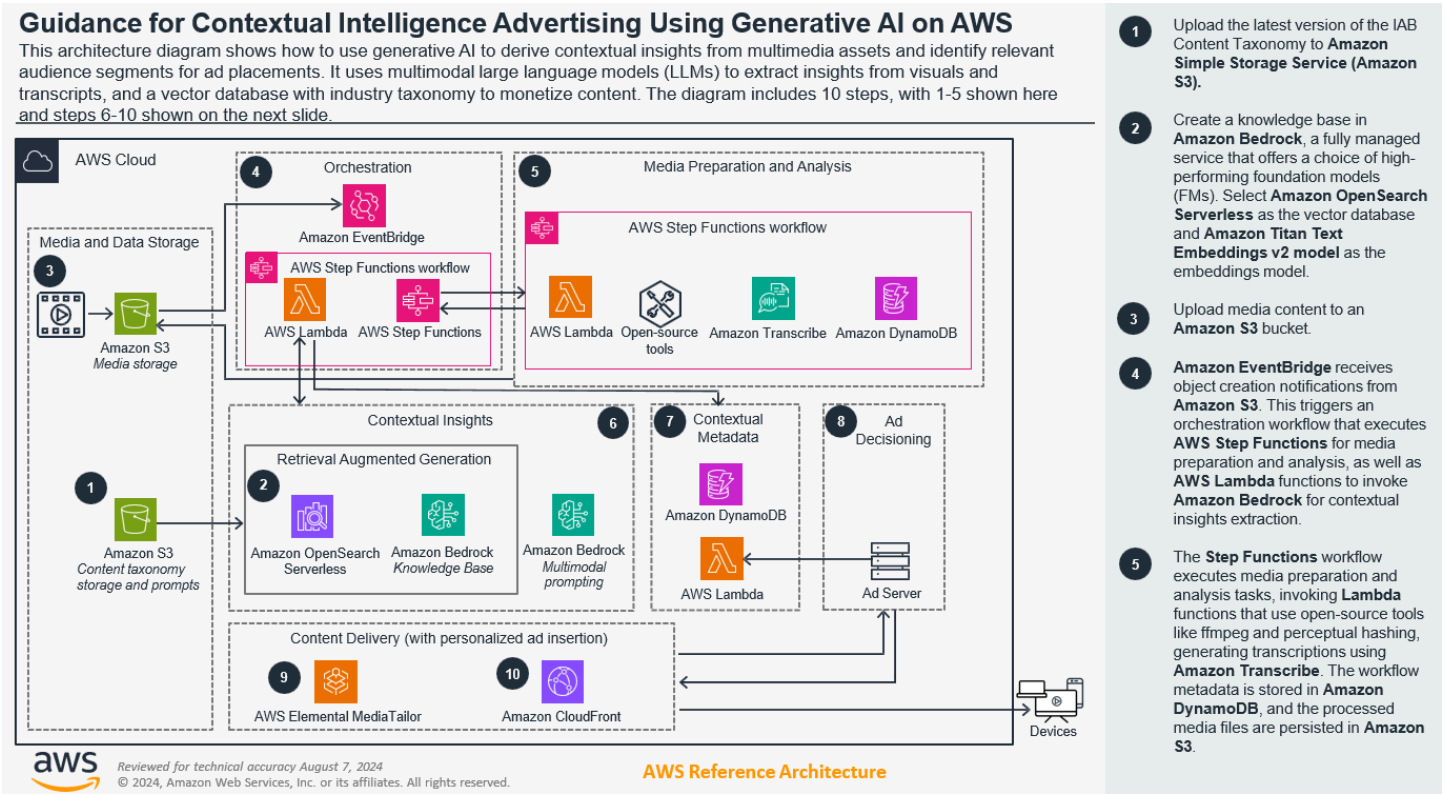
Ad intelligence, measurement, and security

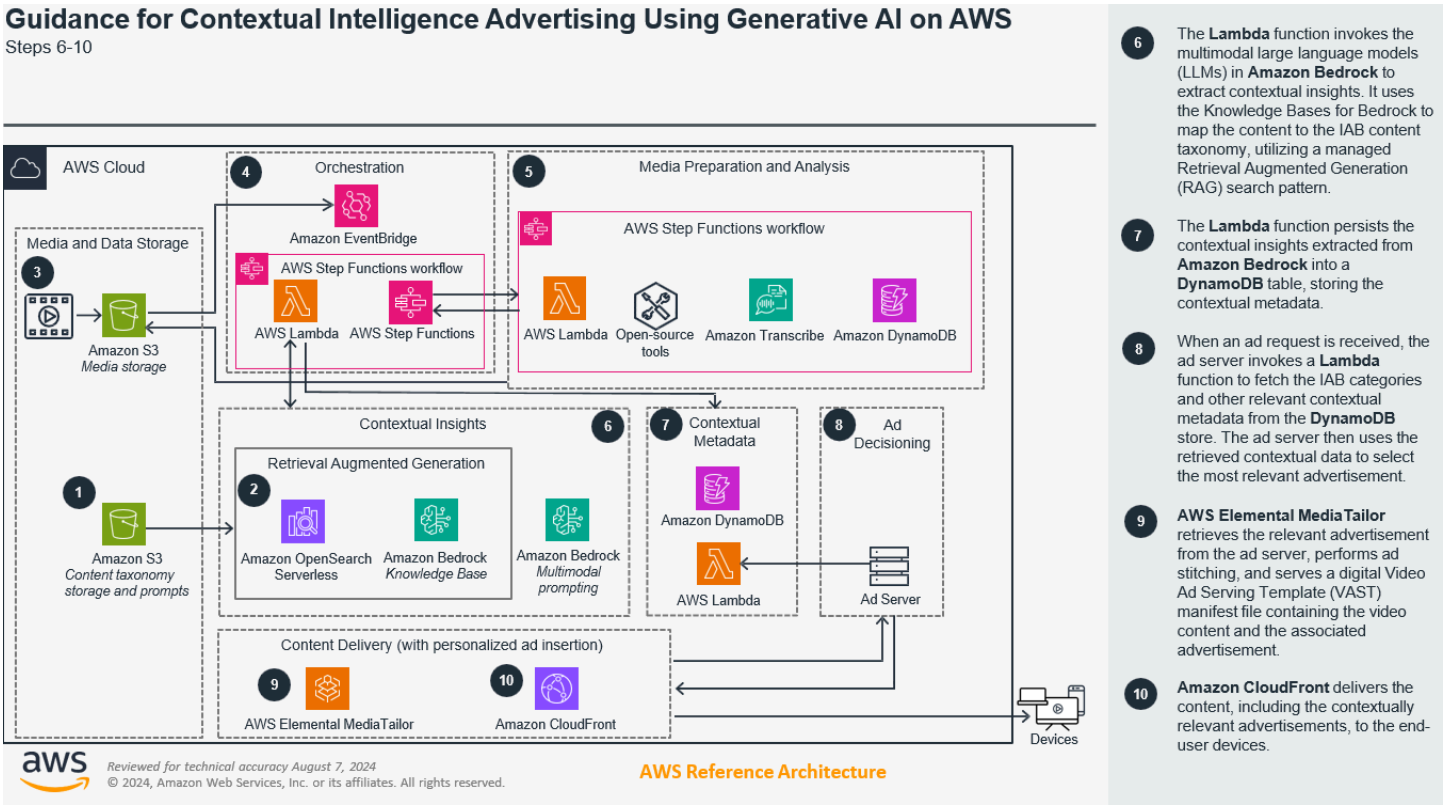
This scenario covers the solution guidance for digital ad content creation with brand safety, verification, and fraud detections.

Contextual advertising

This guidance shows how to use generative AI to derive contextual insights from multimedia assets and identify relevant audience segments for targeted advertising placements. By analyzing video, audio, and text, you can identify the most relevant audience segments and deliver personalized advertising experiences. Moreover, the use of multimodal large language models (LLMs) facilitates the extraction of insights from both visual and transcript components so you can align your media with the right advertising opportunities and monetize content more effectively.

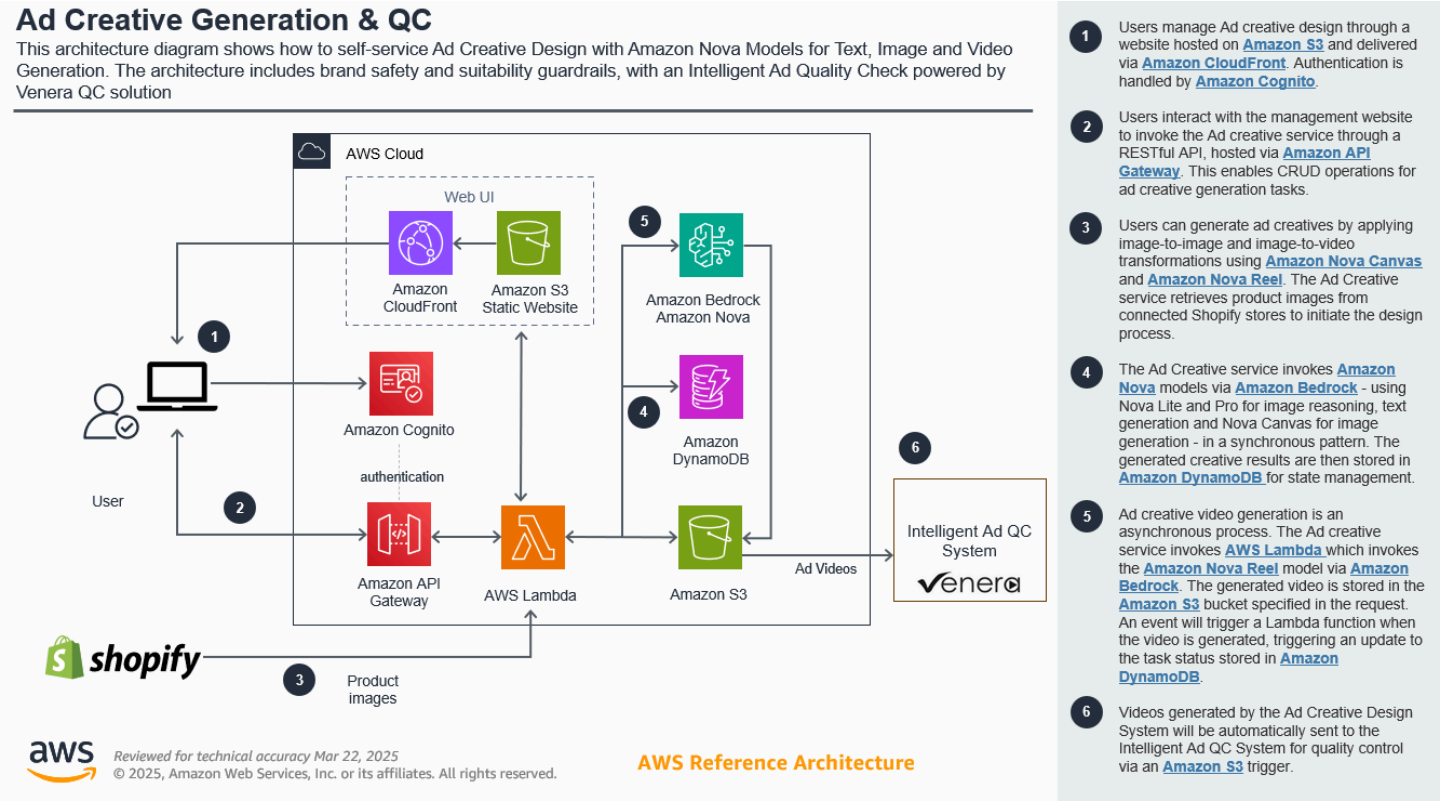
For additional details, see [Guidance for Contextual Intelligence Advertising Using Generative AI on AWS](#).





Generative AI-generated ad creative and moderation

The solution guidance demonstrates the Intelligent Ad QC solution leveraging the power of AWS AI and generative AI to enable media publishers to efficiently manage advertising content at scale, facilitate alignment with brand and audience requirements, and deliver a high-quality viewer experience - ultimately improving the monetization of their ad-supported streaming services.



Fraud protection using the HUMAN AWS Marketplace

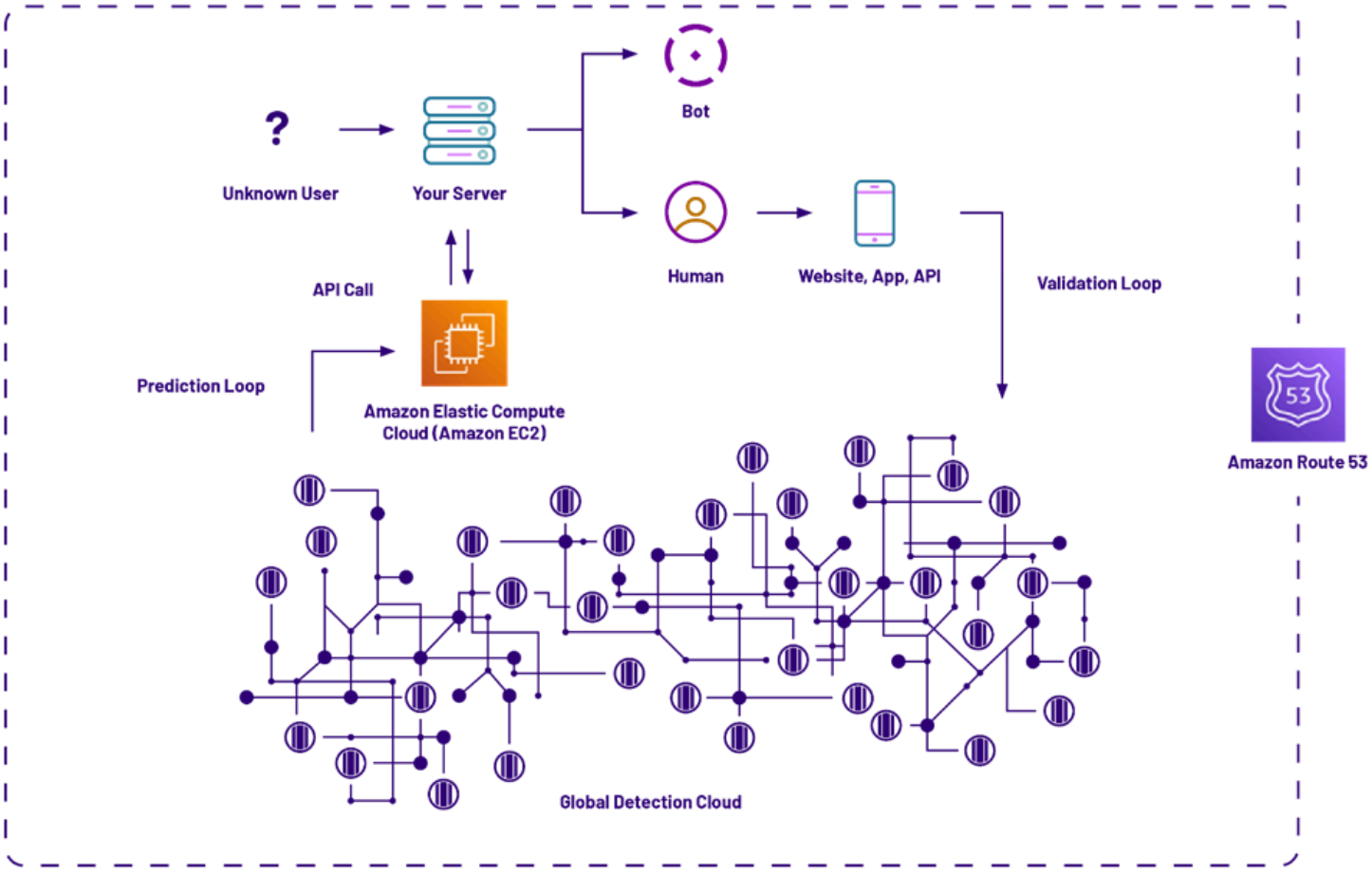
The HUMAN AWS Marketplace solution provides protection from ad fraud to stakeholders in the digital advertising solution. The scalability, availability, and efficiency of Amazon Web Services (AWS) enable the HUMAN technology to help prevent ad fraud, no matter where it originates—within 12 milliseconds or less as the page loads.

Prediction services are deployed on Amazon Elastic Compute Cloud (Amazon EC2) instances, and Amazon Route 53 is used for the HUMAN authoritative domain name system (DNS).

The HUMAN fraud-protection solution on AWS makes it simple for DSPs, SSPs, and businesses to protect their digital investments. By deploying the HUMAN solution, AWS customers can benefit from the following:

- **Solution verification:** The HUMAN solution assists DSPs and SSPs improve inventory quality by using genuine ad servers and verifying ad fraud is not part of their advertising solution. DSPs and SSPs have the means to provide their customers with proof their solutions are trustworthy and free from abuse.

- **Trustworthy data:** When fraudulent activity is avoided, businesses can rely on their data when making business decisions. With accurate insight into an advertising campaign's performance, businesses do not need to waste advertising budget on fraudulent impressions.
- **Maintain reputation:** By avoiding fraud before it enters their solution, HUMAN assists publishers provide a trustworthy experience for visitors to their site and advertisers, supporting their brand integrity.
- **Optimize return:** By delivering only verified human impressions, DSPs and SSPs can deliver greater value for their customers through better-performing inventory, verifying their resources, time, technology, and budget are used effectively.
- **Increase trust in the digital marketplace:** By protecting against phishing, malvertising, and other ad fraud, HUMAN provides DSPs, SSPs, publishers, and advertisers with a way to verify the experience is consistent with expectation, restoring trust and transparency to digital advertising in every industry.
- **Take advantage of the latest technology:** With the visibility of HUMAN into the internet and through its network effect, customers know they are receiving the latest abuse protection techniques for their technology solutions—one that keeps pace with the speed of innovation.



For additional details, see [How HUMAN Advertising Intelligence Solutions Help Protect Against Ad Fraud in the Ad Tech Industry].

Operational excellence

The operational excellence pillar provides guidance on running and monitoring systems to deliver business value and continually improving supporting processes and procedures.

Operational excellence includes the ability to support development, and run workloads effectively. It also helps to gain insight into operations, so that supporting processes and procedures can be improved continuously to deliver business value. For more information, see the Operational Excellence Pillar whitepaper.

Design principles

- **Automate operational processes:** Advertising workloads require a high degree of automation to ensure consistent, reliable, and scalable operations. By automating advertising-specific processes, you can reduce the risk of human errors that can lead to ad serving failures, data quality issues, or compliance violations.
- **Optimize for observability:** Advertising workloads generate massive amounts of data that must be collected, processed, and analyzed in real-time. By optimizing for observability, you can rapidly identify and resolve issues, continuously improve performance, and make data-driven decisions to optimize your advertising business.

Organization

ADVOPS01: What organizational mechanisms do you have to support your advertising outcomes?

Effective organizational mechanisms are important for supporting successful advertising outcomes. It is important to assess the processes, roles, responsibilities, and performance management practices in place within the organization to ensure the efficient and reliable delivery of advertising workloads.

Best practices

- ADVOPS01-BP01 Assess trade-offs between ad serving architecture options and associated risks

- ADVOPS01-BP02 Create RACI matrices that define the roles and responsibilities for each key advertising process like infrastructure monitoring
- ADVOPS01-BP03 Establish performance metrics by defining key performance indicators (KPIs) and service-level objectives (SLOs)
- ADVOPS01-BP04 Establish data governance and compliance operations

ADVOPS01-BP01 Assess trade-offs between ad serving architecture options and associated risks

When designing the ad serving infrastructure, evaluate the trade-offs between different architectural approaches and their associated risks. This includes considering factors such as performance, scalability, availability, security, and cost to determine the optimal solution.

Implementation guidance

- Assess the performance and scalability requirements of your ad serving workload, including peak traffic patterns and seasonal fluctuations. Evaluate architectures that can dynamically scale, such as serverless or containerized approaches.
- Analyze the availability and reliability needs of your ad serving infrastructure, ensuring that your architecture includes redundancy and fault tolerance mechanisms to maintain high uptime.
- Evaluate the security risks associated with your ad serving workload, such as bot attacks and ad fraud, and implement appropriate controls like web application firewalls and rate limiting.

Key AWS services

Key AWS services

- AWS Lambda
- AWS Fargate
- Amazon ECS
- Amazon EKS
- Amazon CloudFront
- Amazon Route 53
- AWS WAF

- Application Load Balancer
- Amazon Virtual Private Cloud

Resources

- Building Applications with Serverless Architectures

ADVOPS01-BP02 Create RACI matrices that define the roles and responsibilities for each key advertising process like infrastructure monitoring

When designing advertising workloads, define roles and set clear expectations for each stakeholder for seamless key advertising processes. By implementing this best practice, organizations can leverage RACI (responsible, accountable, consulted, and informed) matrices to establish a robust framework for accountability and decision-making.

Implementation guidance

By creating comprehensive RACI matrices, organizations can establish accountability, decision-making authority, and communication requirements for each step of the ad-serving workflow. This level of clarity assists in blocking confusion, gaps, or overlaps in responsibilities. This clarity also verifies that stakeholders understand their roles and how they contribute to the overall success of the advertising operations.

For data management specifically, implement the following approach:

1. Establish data classification and handling processes:
   - Classify advertising data based on criticality and latency requirements (real-time bidding data, campaign configuration data, historical analytics data)
   - Define data retention policies for each classification (for example, bid data retained for 30 days, campaign data for one year)
   - Implement data lineage tracking using AWS Glue Data Catalog to document data origins, transformations, and dependencies across the advertising pipeline
2. Structure teams around data criticality:
   - **Real-time data operations team:** Responsible for sub-100ms data like bidding, user profiles, and fraud detection

- **Campaign data management team:** Handles near real-time data for configurations, targeting, and budgets
- **Analytics and reporting team:** Manages batch processing for historical data and business intelligence

3. Define data ownership with specific domains:
   - Assign data stewards for specific advertising domains such as:
     - Bid management domain (bid requests, responses, auction data)
     - User profile domain (demographic data, behavioral signals, privacy preferences)
     - Campaign domain (creative assets, targeting parameters, budget configurations)
     - Analytics domain (performance metrics, attribution data, reporting datasets)
   - Document domain-specific quality standards and governance responsibilities in the RACI matrix
4. Implement data governance using AWS services:
   - Use AWS Organizations with service control policies (SCPs) to enforce data residency requirements for different Regions
   - Configure IAM roles with least-privilege permissions aligned to team responsibilities (for example, real-time team with write access to bidding data, read-only for analytics)
   - Deploy AWS Control Tower guardrails to help block unauthorized cross-account or cross-Region data transfers
   - Implement AWS Config rules to continuously audit compliance with data governance policies
5. Establish data management processes:
   - Data cataloging: Use AWS AWS Glue Data Catalog to maintain a comprehensive inventory of advertising datasets with metadata, ownership, and classification
   - Quality monitoring: Implement automated data quality checks using AWS Glue DataBrew to validate incoming data against defined schemas and business rules
   - Workflow automation: Create AWS Step Functions workflows for data handoffs between teams, with validation checkpoints and approval gates for critical data transitions

Resources

- AWS Organizations for team boundaries: Implement multi-account strategies with SCPs to enforce separation of duties between data teams as described in AWS Organizations Best Practices

- Data governance implementation: Follow the framework in AWS Data Governance Whitepaper to establish controls specific to advertising data domains
- Team collaboration on data assets: Use Amazon DataZone to create a data marketplace where teams can discover, share, and collaborate on advertising datasets
- Automated operational procedures: Implement AWS Systems Manager runbooks for standardized data operations tasks across teams
- Compliance monitoring: Deploy AWS Config rules to continuously validate that data handling practices meet organizational and regulatory requirements
- Create a RACI or RASCI matrix for a cloud operating model

Key AWS services

- AWS Organizations
- AWS IAM
- AWS Control Tower
- AWS AWS Glue Data Catalog
- AWS Glue DataBrew
- Amazon DataZone
- AWS Systems Manager
- AWS Config
- AWS Step Functions

ADVOPS01-BP03 Establish performance metrics by defining key performance indicators (KPIs) and service-level objectives (SLOs)

Establishing a comprehensive set of operational performance metrics is critical in an advertising workload. It helps organizations to measure, monitor, and validate the performance of the advertising operations.

Implementation guidance

Define operational KPIs and establish SLOs. For example, consider the following example criteria:

- **Ad serving:** Ad serving latency

- **Infrastructure maintenance:** System uptime, maintenance task completion rate, incident response time

Key AWS services

- Amazon CloudWatch

Resources

- Improve application reliability with effective SLOs

ADVOPS01-BP04 Establish data governance and compliance operations

Advertising data management requires robust governance and compliance procedures, especially in a multi-Region environment. This best practice verifies adherence to regional data privacy laws while maintaining operational efficiency across global advertising operations.

Implementation guidance

- For data residency compliance alignment:
  - Implement landing zone controls for different geographical Regions
  - Configure data boundary controls using AWS Control Tower
  - Set up guardrails for data movement between Regions
  - Monitor and enforce data locality requirements
- For data governance:
  - Establish data classification policies for advertising data types
  - Implement data retention and archival procedures
  - Set up access controls and encryption policies
  - Configure audit logging and compliance reporting
- For regulatory compliance alignment:
  - Implement GDPR requirements for EU user data
  - Set up consent management systems
  - Monitor compliance with regional advertising laws

Key AWS services

- AWS Control Tower
- AWS Organizations
- AWS Config
- AWS CloudTrail
- Amazon Macie
- AWS Local Zones
- AWS Identity and Access Management

Resources

- Best Practices for managing data residency in AWS Local Zones using landing zone controls
- General Data Protection Regulation (GDPR) Center
- Scale across borders: build a multi-Region architecture while maintaining data residency

Prepare

ADVOPS02: How do you ensure comprehensive observability and monitoring across your advertising workload?

Advertising workloads often involve complex and distributed systems that must operate reliably and efficiently. To detect, diagnose, and resolve issues quickly, it's important to have comprehensive observability across the entire advertising workload. This includes implementing monitoring and logging solutions that provide visibility into key metrics, events, and dependencies between various components. By having a holistic view of the workload's performance and health, you can proactively identify and address problems. This approach can help ensure a reliable and predictable advertising experience for your customers.

Best practices

- ADVOPS02-BP01 Implement monitoring across each layer of your advertising stack including infrastructure, applications, and user experience

- ADVOPS02-BP02 Collect and analyze detailed metrics for successful operations and ad campaigns
- ADVOPS02-BP03 Implement centralized logging to aggregate logs from all components of your advertising stack
- ADVOPS02-BP04 Instrument your advertising application code and infrastructure to emit detailed, structured logs and metrics

ADVOPS02-BP01 Implement monitoring across each layer of your advertising stack including infrastructure, applications, and user experience

Ensuring operational excellence in advertising workloads requires a holistic approach to monitoring. This best practice emphasizes the importance of implementing comprehensive monitoring solutions that span all layers of the advertising stack. The advertising stack includes the ad-serving infrastructure, data pipelines, application performance, and user experience. By monitoring these various components, you can gain a complete understanding of the overall health and performance of your advertising workload. This understanding helps you identify and address issues, optimize resource utilization, and deliver a seamless customer experience. With a multi-layered monitoring approach, you can proactively detect and resolve problems before they impact your business.

Implementation guidance

Monitor and set KPIs and SLOs for infrastructure services using Amazon CloudWatch for services like Amazon EC2 and Amazon EBS. Set up CloudWatch Alarms for resource utilization, performance, and availability.

Resources

- Observability using native Amazon CloudWatch and AWS X-Ray for serverless modern applications
- AWS Observability Maturity Model

ADVOPS02-BP02 Collect and analyze detailed metrics for successful operations and ad campaigns

Advertising workloads can experience significant spikes in traffic and resource utilization, which can impact performance and availability. To maintain observability across these dynamic workloads, collect granular, one-second metrics with near real-time latency. Use advanced analytics, machine learning, and anomaly detection to continuously analyze this data and proactively identify issues before they impact campaigns. This level of observability and proactive issue detection improves the reliability and responsiveness of your advertising infrastructure, even during periods of high demand.

Implementation guidance

Consider the following for collecting important ad-serving metrics:

- **Granular metrics:** Collect metrics at a one-second granularity to capture spikes and fluctuations in advertising workloads. Key metrics to monitor include:
  - **Bid requests per second:** Number of bid requests received.
  - **Bid response time:** Time taken to respond to bid requests.
  - **Successful bids:** Number of successful bids placed.
  - **Bid win rate:** Percentage of bids won compared to total bids placed.
  - **Latency metrics:** Measure network latency, processing time, and database query times.

For database metrics for RTB platforms:

- **Read and write latency:** Measure the time taken for read and write operations in your databases including DynamoDB and Amazon RDS.
- **Throughput:** Monitor read and write capacity units to verify that your database can handle the load.
- **Error rates:** Track the number of failed read/write operations.
- **Connection count:** Monitor the number of active connections to the database.

Consider the following for effective analysis of ad serving insights:

- **Anomaly detection:** Use Amazon CloudWatch anomaly detection to detect anomalies in your metrics based on historical data patterns automatically. This can help identify potential issues before they impact campaigns.

Create useful alarms for monitoring and alerting. Configure CloudWatch alarms for critical metrics such as:

- **High latency:** Set alarms for when bid response times exceed a defined threshold (for example, 100ms).
- **Low bid win rate:** Initiate alerts if the bid win rate drops below a specific percentage.
- **Database latency:** Create alarms for read or write latency thresholds to ensure database performance.

Configure your notification mechanisms. Use Amazon Simple Notification Service (Amazon SNS) to send alerts to relevant stakeholders using email or SMS when alarms go off. This makes it possible for the appropriate teams to respond quickly to potential issues.

Other important considerations for observability of advertising workloads:

- **Impact on cost:** CloudWatch has charges for custom metrics, alarms, and API requests, which can add to the overall AWS costs. The cost can vary based on the number of metrics, alarms, and API calls configured. SNS has charges for the number of notifications sent, which can also contribute to the overall cost.
- **To reduce impact on cost:** Analyze the expected usage patterns and configure CloudWatch and SNS based on specific needs to optimize costs. Consider cost-optimized approaches, such as using sampling or aggregation for high-volume metrics, to reduce the number of custom metrics and API calls.
- **Impact on latency:** The monitoring and logging solutions recommended, when implemented correctly, should have minimal impact on the latency of your advertising workloads. CloudWatch provides near real-time data ingestion and processing, which helps in quickly detecting and diagnosing issues. However, it's important to verify that the monitoring and logging solutions are non-blocking and do not introduce additional latency in your critical advertising workflows.
- **To reduce impact on latency:** Implement monitoring and logging solutions using asynchronous, non-blocking approaches to minimize the impact on latency. Consider using sampling or batching techniques to reduce the number of API calls and optimize the performance of your monitoring and logging solutions.

- **Ad fraud metrics:** Monitor invalid traffic rates, bot detection rates, and suspicious activity patterns.
- **Brand safety metrics:** Track content classification accuracy, moderation response times, and policy violation rates.
- **Measurement consistency:** Monitor cross-system measurement discrepancies, attribution model performance, and conversion path integrity.

Resources

- Set up custom metrics in CloudWatch
- Monitoring metrics in an Amazon RDS instance
- Creating cross-service dashboards
- Aggregating metrics using CloudWatch
- Analyzing performance anomalies with Amazon DevOps Guru for Amazon RDS

ADVOPS02-BP03 Implement centralized logging to aggregate logs from all components of your advertising stack

To provide comprehensive visibility and operational efficiency across your advertising stack, implement a centralized logging solution. You can gain a holistic view of your system's performance and behavior by aggregating logs from all components of your advertising stack, including third-party integrations and custom applications.

Implementation guidance

Review centralized logging with opensearch to aggregate logs from all core components of the advertising workload.

Amazon OpenSearch Service

Use Amazon OpenSearch Service to aggregate logs from all core components of the advertising workload, including ad serving components like AWS Fargate tasks, Amazon EC2 instances, or AWS Lambda functions. OpenSearch provides a robust, scalable, and highly-available log aggregation solution with powerful search and analytics capabilities. Use this approach to have a consolidated view of logs across your entire advertising ecosystem, facilitating faster issue detection and resolution.

Amazon CloudWatch Logs

Alternatively, you can use Amazon CloudWatch Logs to capture and aggregate logs specifically from your ad serving components. CloudWatch Logs is a fully-managed service that makes it easy to monitor, store, and access your log files from various AWS services and on-premises sources. If your primary focus is on monitoring and analyzing the logs related to your ad serving components, CloudWatch Logs can be a suitable option.

The choice between OpenSearch and CloudWatch Logs for ad serving logs depends on your specific requirements and the overall complexity of your advertising workload. If you need a comprehensive, cross-component log aggregation and analysis solution, OpenSearch may be the preferred choice. However, if your needs are more focused on the ad serving components, CloudWatch Logs can be a simpler and more cost-effective option.

Resources

- Centralized Logging with OpenSearch

ADVOPS02-BP04 Instrument your advertising application code and infrastructure to emit detailed, structured logs and metrics

Instrument your advertising application code and infrastructure to emit detailed, structured logs and metrics to achieve comprehensive visibility into advertising workloads. Organizations can monitor all components of their workloads, define KPIs, and set up alerts for critical metrics by using observability services like Amazon CloudWatch. This structured approach enables teams to detect, diagnose, and resolve issues quickly. This approach also optimizes performance and reliability of advertising campaigns.

Implementation guidance

To gain comprehensive visibility into your advertising workload and quickly detect, diagnose and resolve issues, use the following logging strategy:

- Use Amazon CloudWatch and AWS X-Ray to capture key performance metrics, error rates, latency data, and detailed logs from ad serving infrastructure.
- Centralize all logs from the advertising stack, including third-party integrations and partner platforms, using a log aggregation solution like Amazon CloudWatch Logs.

- Implement distributed tracing with AWS X-Ray to track user journeys and identify performance bottlenecks across advertising applications and services.
- Integrate with ad tech platforms and partners to receive comprehensive event-level data like bid requests, ad impressions, and conversions to power observability and analytics.

Resources

- Observability using native Amazon CloudWatch and AWS X-Ray for serverless modern applications

Operate

ADVOPS03: How do you document runbooks to guide your teams to respond and resolve operational events that can impact your workload?

Advertising workloads often involve complex systems and processes that are susceptible to various operational events and incidents. These can include ad serving failures, data quality issues, regulatory changes, and other disruptions that can impact the performance and reliability of the advertising platform. Maintain well-documented runbooks and playbooks that guide your teams in effectively responding to and resolving these issues. You can improve the stability and resilience of your advertising workload by proactively identifying the most common operational events and incidents and establishing clear procedures for managing them.

Best practices

- ADVOPS03-BP01 Create runbooks for the most common operational events and incidents that can impact your advertising workload
- ADVOPS03-BP02 Automate runbooks to gain operational efficiency

ADVOPS03-BP01 Create runbooks for the most common operational events and incidents that can impact your advertising workload

Develop structured procedures to manage operational events and incidents in your advertising workloads. Runbooks provide step-by-step procedures for well-understood, routine operations,

while playbooks guide your response to incidents with less predictable outcomes. These documented procedures assist teams to respond consistently and effectively, reducing human error and improving operational resilience.

Implementation guidance

1. Use automation for predictable operational responses:
   - Implement auto scaling and load balancing using AWS services to handle common traffic patterns:
     - Configure Amazon EC2 Auto Scaling groups with appropriate scaling policies based on advertising traffic patterns.
     - Set up Elastic Load Balancing to distribute traffic across healthy instances.
     - Implement predictive scaling based on historical data for cyclical advertising campaigns.
     - Configure scheduled scaling for planned high-traffic events like product launches or holiday promotions.
   - For scenarios where auto scaling may not be sufficient:
     - Create runbooks for requesting additional EC2 capacity through ODCRs before anticipated high-traffic events.
     - Implement AWS Systems Manager automation documents to execute common scaling procedures.
     - Use AWS Auto Scaling for predictive scaling based on historical data patterns.
     - Configure CloudWatch alarms to trigger automated responses for common capacity issues.
2. Create purpose-built runbooks and playbooks for different advertising scenarios:
   - Infrastructure capacity runbooks:
     - Document procedures for submitting on-demand capacity requests (ODCRs) for anticipated high-traffic events
     - Create step-by-step guides for scaling resources up or down based on traffic patterns
     - Establish processes for capacity planning before major advertising campaigns
     - Define monitoring thresholds that trigger capacity management procedures
   - Ad fraud incident playbooks:
     - Document investigation procedures for detected fraud patterns (bot traffic, click fraud, impression fraud)
     - Establish clear escalation paths for high-impact fraud incidents with defined severity levels

    - Create detailed workflows for fraud investigation and mitigation, including evidence collection
    - Define recovery procedures to restore normal operations after fraud mitigation
    - Document procedures for `ads.txt` and `sellers.json` verification and maintenance
    - Implement AWS Marketplace solutions like HUMAN for automated fraud detection and prevention
    - Create operational workflows for real-time fraud detection using Amazon SageMaker AI ML models
- Brand safety incident playbooks:
    - Document procedures for content moderation escalations and violations
    - Establish workflows for emergency ad creative approval and rejection
    - Create processes for brand safety incident response, including stakeholder communication
    - Define procedures for implementing emergency blocking rules in ad serving systems
    - Implement content moderation workflows using Amazon Rekognition for image/video analysis and Amazon Comprehend for text analysis
    - Create escalation procedures for high-risk content identification with automated alerts via Amazon EventBridge
    - Document processes for regular updates to content moderation models in SageMaker AI
- Measurement anomaly runbooks:
    - Document step-by-step procedures for investigating common measurement discrepancies
    - Establish workflows for cross- data reconciliation and validation
    - Create processes for measurement system recalibration and data correction
    - Define verification steps to confirm resolution of measurement issues
- AI measurement system playbooks:
    - Document operational procedures for AI model training, validation, and deployment
    - Create monitoring workflows for detecting model drift and performance degradation
    - Establish human oversight processes for AI-based measurement systems
    - Document recovery procedures for AI system failures
    - Implement continuous improvement workflows using AWS Step Functions to orchestrate model retraining
    - Create operational procedures for managing model versions and deployments

3. Implement continuous improvement for runbooks and playbooks:
    - Review and update documentation after each incident or significant operational event
    - Conduct regular simulation exercises to validate playbook effectiveness
    - Incorporate lessons learned into revised procedures
    - Track key metrics on playbook execution efficiency and outcome effectiveness

Key AWS services

- Amazon EC2 Auto Scaling
- Elastic Load Balancing
- AWS Systems Manager
- Amazon CloudWatch
- AWS Auto Scaling
- Amazon SageMaker AI
- Amazon Rekognition
- Amazon Comprehend
- AWS Lambda
- AWS Step Functions
- Amazon EventBridge
- Amazon Route 53
- AWS Marketplace solutions (like HUMAN)

Resources

- OPS07-BP03 Use runbooks to perform procedures
- Use playbooks
- AWS Marketplace: HUMAN Fraud Protection Solutions
- How HUMAN Advertising Intelligence Solutions Help Protect Against Ad Fraud in the Ad Tech Industry
- Content Moderation Using Machine Learning on AWS
- Guidance for Content Moderation on AWS

ADVOPS03-BP02 Automate runbooks to gain operational efficiency

Document runbooks for failover procedures, capacity scaling, and incident response workflows using AWS Systems Manager documents for automation.

Implementation guidance

Consider the following example use case runbooks:

Scaling runbook

- Design step-by-step workflows for manually scaling up and down Amazon EC2 instances and increasing or decreasing managed service capacities
- Create automation scripts to initiate auto scaling actions based on predefined events
- Perform validation checks for successful scaling operations

Third-party service disruptions

- Implement multi-provider redundancy and failover mechanisms using AWS Lambda functions and Amazon API Gateway
- Use AWS X-Ray for end-to-end tracing and troubleshooting of distributed applications and third-party integrations
- Document playbooks for provider switching, data synchronization, and incident escalation using AWS Step Functions and AWS Lambda

Infrastructure capacity issues

- Implement auto scaling and load balancing using Amazon EC2 Auto Scaling and Elastic Load Balancing (ELB)
- Use AWS Auto Scaling for predictive scaling based on historical data and scheduled scaling for planned events
- Document runbooks for capacity planning, scaling procedures, and cost optimization using AWS Systems Manager Documents

Cost optimization runbook

- Procedures for reviewing resource utilization and identifying opportunities for optimization using AWS Cost Explorer
- Guidelines for selecting the most cost-effective Amazon EC2 instance types and purchasing models (like On-Demand, Reserved, or Spot) based on workload patterns
- Automation to right size Amazon EC2 instances, remove unused resources, and use AWS Savings Plans
- Processes for periodic cost reviews and budget management

Data management runbook

- Create runbooks for:
  - Data pipeline failures
  - Replication issues
  - Storage capacity management
  - Compliance violations
- Include Regional considerations.
- Document recovery procedures.

Manage operational processes

ADVOPS04: How do you manage operational processes for different data types, latency, and regulatory requirements in your workload?

Effective management of advertising data requires distinct operational processes based on data types and latency requirements. Organizations must implement appropriate procedures for handling real-time bidding data, user profiles, campaign data, and analytics while maintaining compliance with regional regulations and performance requirements.

Best practices

- ADVOPS04-BP01 Implement operational procedures based on data classification and latency requirements

ADVOPS04-BP01 Implement operational procedures based on data classification and latency requirements

Managing advertising workloads requires different operational approaches based on data latency needs. This best practice focuses on establishing specific procedures for handling low-latency data like bid requests, medium-latency data such as campaign optimization, and high-latency data including historical analytics.

Implementation guidance

For low-latency data (bid data, user profiles, real-time impressions):

- Implement multi-AZ deployments with automatic failover mechanisms to facilitate continuous availability
- Configure monitoring with short evaluation periods appropriate for detecting real-time issues
- Establish dedicated rapid-response procedures for critical alerts affecting bidding operations
- Implement circuit breakers in API calls to help block cascading failures during service degradation
- Create runbooks for emergency traffic management during extreme load conditions
- Configure auto-scaling with aggressive scaling policies to handle sudden traffic spikes
- Implement local caching strategies to reduce database load for frequently accessed data
- Set up dedicated dashboards with high-frequency metric collection for real-time monitoring

For medium-latency data (behavioral data, campaign optimization):

- Configure batch processing jobs with appropriate completion targets for campaign optimization
- Implement queue management with automated retry mechanisms for failed operations
- Set up monitoring with balanced evaluation periods suitable for near real-time operations
- Create standard incident response procedures with appropriate escalation paths
- Implement data validation checks with error handling for data quality issues
- Configure auto-scaling based on processing queue depth and scheduled campaign activities
- Set up dashboards with appropriate refresh rates for campaign management operations

For high-latency data (historical data, analytics):

- Schedule batch processing during off-peak hours to minimize impact on real-time operations
- Implement cost-optimized storage strategies with appropriate data lifecycle policies
- Configure monitoring with periodic health checks and summary reporting
- Create standard support procedures with appropriate response times for non-critical systems
- Implement automated data quality validation with notification mechanisms
- Configure resource allocation with scheduled scaling based on known processing windows
- Establish regular performance review processes with trend analysis

For specialized advertising data types:

- Fraud detection data: Implement optimized processing pipelines with appropriate monitoring and escalation procedures designed for the critical nature of fraud detection
- Content moderation data: Create workflows that balance automated screening with human review processes, with appropriate prioritization based on content risk assessment

Key AWS services

- Amazon CloudWatch
- AWS Systems Manager
- Amazon EventBridge
- Amazon Kinesis Data Streams
- Amazon Managed Service for Apache Flink

Resources

- AWS for Advertising & Marketing
- Architectural patterns for real-time analytics using Amazon Kinesis Data Streams, part 1
- Streaming architecture patterns using a modern data architecture

Security

The security pillar encompasses the ability to protect data, systems, and assets to take advantage of cloud technologies to improve your security.

The security pillar provides an overview of design principles and questions. You can find implementation guidance in the Security Pillar whitepaper.

Design principles

- **Protect the advertising pipeline:** Control access to and from your DSP (demand-side platform) using identity management and network access control. Protect data in transit using encryption.

Identity and access management

ADVSEC01: How do you manage access to your advertising platform?

Managing access to an advertising platform remains a critical security concern, as it involves protecting potentially sensitive data, financial information, and programmatic access while maintaining appropriate permissions for authorized users.

Best practices

- ADVSEC01-BP01 Implement user authentication and access control to protect bidding process and content
- ADVSEC01-BP02 Restrict DSP access to allow only authorized SSPs
- ADVSEC01-BP03 Restrict DSP outbound traffic to authorized SSPs only
- ADVSEC01-BP04 Implement authorization by setting access policies, and implement least privilege access to protect programmatic workloads

ADVSEC01-BP01 Implement user authentication and access control to protect bidding process and content

Authenticate the approved SSPs (supply-side platforms) and advertisers. Based on this authentication, DSPs can provide them with least-privileged authorization and access to the relevant resources and data.

Implementation guidance

AWS offers multiple services to provide SSPs and DSPs secured and scalable user management across all parts of the workload. Consider using Amazon Cognito to provide scalable authentication, authorization, and user management to your applications. Implementing federated identity integration with trusted identity providers can allow for ideal single sign on (SSO) for both publishers and advertisers. SSPs and DSPs can either use SAML 2.0 or OpenID Connect (OIDC) to create a trusted identity provider. From there, roles and permissions can be configured by a trusted administrator for users from the identity provider.

Additionally, you can use AWS Identity and Access Management (IAM) for fine-grained access control for users and different AWS services that may interact with advertising workloads. Enforce strict IAM policies that define permissions to help control access within AWS workloads. IAM policies define permissions for an action regardless of the method used to perform the operation.

Consider implementing role-based access control to determine which access to resources may align with a role based on business requirements. Use specific roles for different advertising services, including DSPs and SSPs, to verify that services operate with limited least privileged access.

Resources

- AWS IAM Identity Center

ADVSEC01-BP02 Restrict DSP access to allow only authorized SSPs

Provide a mechanism to control and manage third-party access to each part of your cloud network environment.

Implementation Guidance

Consider using AWS WAF to allow access for authorized IPs for traffic that arrives at your Application Load Balancer, Amazon API Gateway, and Amazon CloudFront distributions. AWS WAF

helps protect your web applications against common web exploits that may compromise security. Using AWS WAF rules, you can define a set of inspection criteria and review when incoming requests meets the set criteria. It is recommended to use AWS WAF rules to inspect incoming traffic based on several factors like source IP or originating geographic location.

Additionally, consider using AWS PrivateLink to restrict access to your AWS services. AWS PrivateLink allows for the private connection between your AWS VPCs and AWS services without exposing your network traffic to the public internet. If you cannot use AWS PrivateLink, consider using IAM to control access to your AWS services.

Resources

- Configure security groups for your Classic Load Balancer
- How do I use AWS WAF to create IP set rules to restrict IPv4 and IPv6 access?
- Update the security groups for your Network Load Balancer
- Controlling access to Amazon Kinesis Data Streams resources using IAM
- Introducing Amazon API Gateway Private Endpoints
- Use interface VPC endpoints for Amazon Kinesis Data Streams
- Private Amazon AppFlow flows
- Create a server in a virtual private cloud
- Configuring VPC endpoints as AWS Database Migration Service source and target endpoints
- Creating an interface VPC endpoint for AWS Data Exchange
- AWS PrivateLink for Amazon S3
- Considerations for AWS Glue VPC endpoints
- Amazon MSK multi-VPC private connectivity in a single Region
- Changing an Amazon MSK cluster's security group

ADVSEC01-BP03 Restrict DSP outbound traffic to authorized SSPs only

Address the risk of DSP unintentional data disclosure to SSPs that were not approved.

Implementation guidance

Consider using an Amazon Virtual Private Cloud (Amazon VPC) to restrict outgoing traffic from instances to the authorized DSP endpoints. VPCs can to define access to verify that all ports, protocols, and destination IP addresses meet your organizations security needs. Use VPC security

groups to permit access from trusted sources or specific IP ranges. Use a protocol with encryption when transmitting data to maintain data confidentiality and mitigate the risk of unauthorized access to the data.

Additionally, implement AWS Network Firewall to provide control over outbound traffic from your VPCs to approved destinations only. Network Firewall allows you to define and enforce rules to inspect and filter outgoing traffic against malware or unauthorized data exfiltration. Using Network Firewall rule groups, you can prevent data loss, meet compliance requirements, or block any known malware communications.

ADVSEC01-BP04 Implement authorization by setting access policies, and implement least privilege access to protect programmatic workloads

Address the risk of authenticated advertisers and SSPs access to data they should not reach.

Implementation guidance

Implement strong AWS Identity and Access Management (IAM) policies when you deploy a global advertising technology workload. Use the principle of least privilege, and enforce the separation of duties for good security posture. Administrative access should only be given to a small number of secured administrators.

Use IAM Access Analyzer to validate IAM policies and verify that they match IAM best practices and your organization's security standards. IAM Access Analyzer can help your organization review and removed unused or external access across your AWS resources with continuous monitoring. IAM Access Analyzer can also assist administrators by validating your IAM policies against IAM policy grammar and AWS best practices.

Data protection

ADVSEC02: How do you protect data in transit?

Protecting data in transit for advertising workloads is crucial as it involves the constant movement of potentially sensitive information including customer data, campaign metrics, and financial metrics across various portions of the network.

Best practices

- ADVSEC02-BP01 Encrypt DSP to SSP communication in transit using TLS

ADVSEC02-BP01 Encrypt DSP to SSP communication in transit using TLS

Protect data in transit by using encrypted communication channel at the network communication level.

Implementation guidance

Protecting data that is transmitted from network to network remains a top security priority. Data confidentiality, integrity, and authenticity of the supported workloads are crucial for securing sensitive information, preventing unauthorized access, and enabling reliable operations within the workload.

Use AWS PrivateLink to establish connectivity between Amazon VPCs and other services without exposing the data to the public internet. If you have on-premises resources, consider using AWS Direct Connect. Direct Connect can make it easy to establish private connectivity between an AWS datacenter and your internal network. Implementing MACsec security on your Direct Connect connection provides point-to-point encryption for your traffic.

Infrastructure protection

ADVSEC03: How do you protect availability of your programmatic advertisement platform?

Robust security measures protect advertisement workloads and can maintain desired uptime.

Best practices

- ADVSEC03-BP01 Use distributed denial of service (DDoS) protection service to maintain platform availability

ADVSEC03-BP01 Use distributed denial of service (DDoS) protection service to maintain platform availability

Deploying DDoS protection helps create strategies for robust system reliability against potential threats.

Implementation guidance

AWS Shield Standard protects against most DDoS attacks by protecting your AWS resources. AWS Shield Standard is automatically enabled to all AWS customer accounts by default. AWS Shield defends against common volumetric and exhaustion attacks and can help protect advertising endpoints such as API's and websites. AWS Shield can protect advertisement servers or DSPs APIs that may be accessed by advertisers and publishers globally.

To additional features to help you protect against DDoS attacks, consider implementing AWS Shield Advanced to provide additional DDoS protection. Shield Advanced includes continual proactive support and increased bandwidth to protect from DDoS attacks. Shield Advanced can provide advanced monitoring and protection to Amazon CloudFront distributions, Route 53 hosted zones, and Amazon ELBs.

Additionally, AWS WAF can help protect login and provider sign-in pages against credential stuffing or creation of fake accounts. By deploying AWS WAF rules, companies can implement protection against commonly deployed web-based attacks. These attacks include bad bots and SQLi. AWS WAF helps prevent those web requests from hitting your CloudFront edge distributions. You can use AWS WAF to implement bad actor deny lists, which can help prevent certain denial of service (DOS) or bad actors trying to implement malicious ad injection.

Resources

- AWS Shield Advanced overview

Data protection

ADVSEC04: How do you store and protect raw data in your DMP system?

A DMP solution often collects multiple forms of data from customers, advertising solutions, DSPs, and SSPs to transform raw data and provide insights to different portions of a cloud-based advertising workload. Protecting DMPs usually requires robust security measures for the different parts of the pipeline that ingest and transform data. Consider using Amazon S3 as a centralized data store. Amazon S3 is an object storage service that can securely store data for a range of use cases. For data protection, strict S3 bucket policies should be implemented that enforce encryption of data at rest and define strict access controls.

An enhanced layer of protection would be to enforce the use of service-side encryption (SSE) with an AWS KMS-managed key (SSE-KMS) which can assist in protecting the data but also meet potential compliance requirements. SSE-KMS allows you to maintain the ownership of the keys with the ability to revoke access depending on the job requirements to the data. By implementing SSE-KMS you have a defense in depth strategy as the consumer of the data needs access to the data on S3 and permission to use the KMS key to decrypt it. A combination of these features verify data is protected at rest.

Best practices

- ADVSEC04-BP01 Implement secure data collaboration with least privileged access and privacy controls
- Key AWS services
- Resources

ADVSEC04-BP01 Implement secure data collaboration with least privileged access and privacy controls

Implementation guidance

Raw data that is used in collaboration with SSPs, DSPs, and third-party systems need to be carefully shared to verify consumer privacy and data protection. Consider using AWS Clean Rooms, which enables more secure data collaboration without potentially exposing raw data and allows different parties to review and analyze data while maintaining strict privacy controls. With AWS Clean Rooms, you can create a more secure data clean room in minutes and collaborate with other companies to generate unique insights about advertising campaigns, investment decisions, and research and development. AWS Clean Rooms automatically encrypts service metadata at rest without requiring additional configurations. AWS Clean Rooms allows for you to have granular control on the type of information you may want to share.

Use IAM to provide least privileged access to approved parties with AWS Clean Rooms. Use IAM policies to define which users and roles can access which data, analyses, and collaborations. This allows for the precise control of how data is created, modified, and queried within AWS Clean Rooms.

Key AWS services

- AWS Clean Rooms
- AWS IAM

Resources

- Solutions for Advertising and Marketing
- AWS Clean Rooms proof of concept scoping part 1: media measurement
- How AWS Clean Rooms works with IAM

Key AWS services

- AWS S3
- AWS KMS

Resources

- Using server-side encryption with AWS KMS keys (SSE-KMS)
- Bucket policies for Amazon S3

Fraud detection

ADVSEC05: How do you detect and block fraud in your advertisement solution?

Traffic fraud in digital advertising could impact the advertising solution by affecting ad spend through fake clicks, bot traffic, and manipulation schemes. Workloads benefit from resilient and secure systems that implement defense in depth strategies for fraud prevention. Effective

protection assists advertisers optimize budgets by focusing on genuine traffic and properly represented inventory, while publishers can maintain reputation and revenue by addressing fraudulent activities on their solution.

Consider using AWS WAF to evaluate traffic to your advertising solution and filter out suspicious or un-wanted bot traffic through customizable rules. Amazon GuardDuty can also be utilized to help protect your AWS accounts, workloads, and data from threats.

Best practices

- ADVSEC05-BP01 Validate and sanitize content before running a campaign
- Key AWS services
- Resources

ADVSEC05-BP01 Validate and sanitize content before running a campaign

Content validation is essential to mitigate ad fraud and block unwanted content from reaching ad audience.

Implementation guidance

Consider using Amazon S3 which can serve as a secure, scalable storage solution for advertising content. It allows for simple management and distribution of assets. S3 can be configured with strict access controls and encryption to maintain the security of the advertisement files. Additionally, Amazon Rekognition can be utilized to analyze images and videos in advertisements, verifying they meet solution standards and don't contain inappropriate content. This AI-powered service can detect objects, scenes, and activities in visual content. For additional monitoring and auditing, consider using AWS CloudTrail to provide a record of actions taken by users, roles, or AWS services in the ad serving solution, which is essential for security analysis and compliance audits.

Key AWS services

- Amazon S3
- Amazon Rekognition
- AWS CloudTrail

Resources

- Checking object integrity in Amazon S3
- Amazon Rekognition
- AWS CloudTrail

Key AWS services

- AWS WAF
- Amazon GuardDuty
- Amazon Fraud Detector

Resources

- Amazon GuardDuty
- AWS WAF rules
- Amazon Fraud Detector FAQs

Regulatory adherence

ADVSEC06: How do you verify that you adhere to regulatory requirements for your advertising solution?

Compliance is essential for organizations as it supports organizational growth and enables organizations to operate legally across different jurisdictions. Also, compliance standards enhance data protection and security, by addressing requirements such as General Data Protection Regulation (GDPR) or NIST frameworks you verify proper data storage, handling, and deletion protocols.

Best practices

- ADVSEC06-BP01 Verify your advertising workload remains adherent to data protection regulations

ADVSEC06-BP01 Verify your advertising workload remains adherent to data protection regulations

Maintaining compliance is essential to operate and grow your solution. Data encryption is a key requirement for several compliance programs; you can utilize AWS KMS to facilitate data encryption and key management for your solution. KMS allows for the creation and management of cryptographic keys which can be used to encrypt data at rest and in transit. It simply integrates with other AWS services and maintains an audit trail for key usage. KMS also maintains validation and certifications from multiple compliance regimes including FIPS, PCI DSS, and HIPAA.

Implementation guidance

To assist with data governance consider using Amazon Macie. Macie can automatically scan and identify sensitive data across AWS environments. The service can categorize data based on content type and sensitivity level. Based on the data classification Macie provides a risk score for different datasets and storage locations. Amazon Macie can assist to meet regulatory requirements including GDPR, CCPA, HIPAA, by generating detailed reports on data types and locations for regulatory audits.

Key AWS services

- AWS KMS
- Amazon Macie

Resources

- Compliance validation for Macie
- Compliance validation for AWS Key Management Service

User privacy

ADVSEC07: How do you balance user privacy concerns with effective ad targeting?

A privacy sandbox within a web browser is designed to limit and reduce cross-site and cross-application tracking. This highlights a new age of user privacy, where there are less trackers

between a user's internet browser and the end connections designed to increase a user's privacy. There is also an improved user experience with enhanced browser security and more transparent data practices.

Best practices

- ADVSEC07-BP01 Enable secure data privacy and collaboration between advertisers while protecting user privacy

ADVSEC07-BP01 Enable secure data privacy and collaboration between advertisers while protecting user privacy

Delivering a privacy-centric advertising infrastructure assists to optimize your system to keep privacy which focuses on data minimization and secure processing.

Implementation guidance

Consider implementing AWS Clean Rooms as the service makes it simple for you and your partners to analyze and collaborate on collective datasets to gain insights without revealing underlying data to one another. AWS Clean Rooms enable you to share secure data between different parties while maintaining data privacy and control. It has configurable data access controls and differential privacy options. AWS Clean Rooms serves as a privacy-enhanced alternative to traditional cookie-based tracking, aligning well with privacy sandbox principles.

Key AWS Service

- AWS Clean Rooms

Resources

- AWS Clean Rooms FAQs

Brand safety

ADVSEC08: How do you verify brand safety and proper content moderation in your workload?

Facilitating brand safety while meeting regulatory, security, and privacy standards in for your advertising workloads is essential for multiple reasons. Legal compliance is equally critical, as adherence to regulations like GDPR and CCPA assists to avoid fines while facilitating proper handling of personally identifiable information (PII). Financial security is protected by blocking fraud and misplacement, verifying ads appear in appropriate contexts and maximizing ROI on ad spend.

Best practices

- ADVSEC08-BP01 Create guardrails and controls to maintain brand safety and content moderation within your workload
- ADVSEC08-BP02 Look for opportunities to block ad fraud and enhance transparency in your advertising solution

ADVSEC08-BP01 Create guardrails and controls to maintain brand safety and content moderation within your workload

Brand reputation protection can block brand association with inappropriate or otherwise harmful content. Having guardrails can maintain customer trust and potential business relationships while avoiding reputational damage and negative publicity.

Implementation guidance

Consider implementing Amazon SageMaker AI, with the custom model development capability of SageMaker AI, you can build, train, and deploy custom machine learning models. Designing a guardrail for brand safety could allow you to develop a model that could detect inappropriate imagery in advertisements, classify text within content for sentiment and safety, and predict the likelihood of an ad placement being brand appropriate. With the real time inference capability of SageMaker AI, you can deploy your models deemed brand safe for real time content analysis, allowing for quick decision making for your solution.

Additionally, consider using AWS Config, to assess, audit, and evaluate resource configurations within your AWS environment. Config can track changes to underlying resources with your advertising solution to verify that security settings and access controls remain compliance-aligned for brand safety.

Key AWS services

- AWS Config

- Amazon SageMaker AI

Resources

- Examples and More Information: Use Your Own Algorithm or Model
- Compliance

ADVSEC08-BP02 Look for opportunities to block ad fraud and enhance transparency in your advertising solution

DSP's need to verify their advertisers and agencies are purchasing legitimate advertising inventory across potentially multiple exchanges in real time. Consider implementing an ads.txt file, designed by IAB tech labs, is designed to enable additional transparency within the advertising solution by allowing DSPs to review legitimate companies authorized to market their advertisement inventory.

Implementation guidance

Adding an `ads.txt` file lets ad publishers declare which services can market their ad space. Retailers can verify incoming advertisement inventory against the list to verify authenticity. This aids in fraud prevention by blocking domain spoofing threats by bad actors impersonating legitimate publishers. The file also aids in protecting DSP's budgets and campaigns performance. Ads.txt may also aid in compliance by meeting certain criteria large advertisers require within their best practices.

Consider using Amazon S3 to host your `ads.txt` file for highly available and simple access. Amazon S3 allows for version control and accessible updates to the file if needed. Lastly, within Amazon S3, you can block object version deletion using S3 object lock. This defined retention period can be used as an extra layer of data protection.

Key AWS services

- Amazon S3

Resources

- Locking objects with Object Lock

Reliability

The reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it's expected to. This includes the ability to operate and test the workload through its entire lifecycle. This section provides in-depth, best practice guidance for implementing reliable workloads on AWS.

The reliability pillar provides an overview of design principles, best practices, and questions. You can find implementation guidance in the Reliability Pillar whitepaper.

Design principles

There are three key dimensions that define Reliability in the advertising industry: latency, uptime, data security. While this is similar to the range of criticality in transactional systems found in the financial industry, there are unique expectations of resiliency for SSPs, exchanges, and DSPs. We will use RPO and RTO, which are established in the reliability pillar of the Well-Architected Framework

Definitions

- **Recovery Point Objective (RPO):** The maximum amount of data loss allowed as the result of a system failure expressed in units of time.
- **Recovery Time Objective (RTO):** The maximum amount of time allowed for a system to resume its normal operations after a failure.
- **Uptime:** A measure of system reliability, expressed as the period of time a machine, typically a computer, has been continuously working and available.
- **Microservices:** An architectural pattern that arranges an application as a collection of loosely-coupled, fine-grained services communicating through lightweight protocols. One of its goals is to enable teams to develop and deploy their services independently.

Design for reliability

ADVREL01: How do you design your advertising workload service architecture around reliability?

Evaluate architectural approaches for building resilient systems through loosely coupled designs, including SOA, microservices, and event-driven patterns. Define recovery objectives, implement scalable solutions for handling demand fluctuations, and apply chaos engineering to validate system reliability and failure recovery capabilities.

Best practices

- ADVREL01-BP01 Use loosely-coupled architectures to enable graceful recovery from failures
- ADVREL01-BP02 Architect your system with appropriate recovery objectives
- ADVREL01-BP03 Architect for variable demand
- ADVREL01-BP04 Implement chaos engineering practices

ADVREL01-BP01 Use loosely-coupled architectures to enable graceful recovery from failures

Use architecture patterns like service-oriented architecture (SOA), microservices, and event-driven architecture (EDA) to recover quickly and efficiently from failure. These architectural patterns enable robust failure recovery through loosely coupled designs and enhance system resilience and component self-sufficiency.

Implementation guidance

Highly scalable and reliable workloads necessitate reusable software components that are accessible through service interfaces like APIs. Microservices take this a step further by breaking down components into smaller, simpler units. EDAs build upon and enhance microservices with an event broker, fostering greater efficiency.

Implement EDAs using services like Amazon EventBridge and Amazon Simple Notification Service (SNS) to decouple components and enable asynchronous communication. This can improve resilience by reducing hard coded dependencies and enabling retries and error handling.

Make sure that the data pipelines of the advertising system operate reliably despite unexpected failures, packet loss, or high latency. Design interactions between components in your distributed advertising system in such way that their failure makes minimal impact.

Key AWS services

- Amazon Simple Queue Service (SQS)
- AWS Step Functions

Resources

- What is EDA? - Event Driven Architecture Explained - AWS
- Avoiding insurmountable queue backlogs
- How can I prevent an increasing backlog of messages in my Amazon SQS queue?
- Amazon Simple Notification Service (SNS) | AWS News Blog
- Increasing MTBF - Availability and Beyond: Understanding and Improving the Resilience of Distributed Systems on AWS

ADVREL01-BP02 Architect your system with appropriate recovery objectives

Avoid over- or under-architecting your services by working backwards from your services' recovery objectives, striking a balance with adjacent pillars such as cost optimization and operational excellence. KPIs established in the operational excellence pillar should inform approaches to reliability.

Implementation guidance

Identify critical parts of the architecture and individually confirm their reliability and recovery point and time objectives (RPO and RTO). For example, with real-time bidding (RTB), delivery services have increased RPO and RTO requirements as compared to creative services. On close inspection, certain architectures also have variable availability and recovery requirements, operating on a spectrum from multiple layers of redundancy to entirely non-redundant. Advertising customers accept ranges from milliseconds to hours as appropriate recovery. For example, enrichment and auction layers often have the most stringent requirements, while analytics or as necessary reporting can see reduced requirements.

Key AWS services

- AWS Resilience Hub

Resources

- Establishing RPO and RTO Targets for Cloud Applications

ADVREL01-BP03 Architect for variable demand

Architect to elastically launch resources for variable demand, including the most challenging peak events, like flash crowds or thundering herds.

Implementation guidance

Depending on the advertising channel, such as retail stores, video streaming, or audio apps, loads will peak at different times in different locations. Know your historical load statistics, and adjust load testing scenarios based on historical peaks to determine how the workload performs in unexpected situations and peak demand. With Amazon CloudWatch Real-User Monitoring (RUM), you can collect and view client-side data about your web application performance from actual user sessions in near real-time. CloudWatch Synthetics are configurable scripts that run on a schedule to monitor your endpoints and APIs.

If this a new workload without historical data, load testing is part of this process. Until enough historical data is obtained, use Auto Scaling groups and Elastic Load Balancers (ELB) to meet compute demands and send requests to healthy hosts. Networking demands must also be considered and capacity planned to prevent congestion. For critical workloads, consider private AWS Direct Connect networking to connect to partners or on-premise infrastructure to provide sufficient capacity and more stable latency.

Resources

- Predictive scaling for Amazon EC2 Auto Scaling
- Guidance for AdTech Private Network on AWS

ADVREL01-BP04 Implement chaos engineering practices

Accept that "everything fails, all the time," (Dr. Werner Vogels, Amazon CTO), and safely disrupt things on your terms to discover faults and fragility so that you can later improve services.

Implementation guidance

Advertising systems have components that are sensitive to disconnects, latency, and bandwidth changes. Use tools like AWS Fault Injection Service (FIS) or open-source tools like Chaos Monkey to inject failures into your workload which simulate network disruptions or resource unavailability. Based on the results, update responses to failure scenarios, how you monitor, and

what you alert on, then adapt runbooks and playbooks before practicing failure response with relevant teams.

Key AWS services

- AWS Resilience Hub

Resources

Related documentation:

- AWS chaos engineering blogs
- Continuous integration and continuous delivery
- Leverage AWS Resilience Lifecycle Framework to assess and improve the resilience of application using AWS Resilience Hub
- [QA.NT.6] Experiment with failure using resilience testing to build recovery preparedness

Related videos:

- AWS re:Invent 2020 - Developer Keynote with Dr. Werner Vogels

Latency sensitive advertising

ADVREL02: How do your latency sensitive advertising workloads react to including throttling and rate-limiting scenarios?

Consider strategies for managing latency-sensitive advertising workloads through throttling and rate-limiting implementations. Avoid traditional retry mechanisms for fast-failing services, implement effective caching strategies, and proportionally scale across all system components to maintain consistent performance.

Best practices

- ADVREL02-BP01 To allow fast and graceful failure of latency-sensitive services, avoid exponential backing off and retry

- ADVREL02-BP02 Implement a caching strategy
- ADVREL02-BP03 Prevent scale mismatch of both internal services and external partners

ADVREL02-BP01 To allow fast and graceful failure of latency-sensitive services, avoid exponential backing off and retry

With real-time bidding systems, your workload must handle failures in latency-sensitive services. Traditional exponential backoff and retry mechanisms should be avoided. Instead, opt for fast-fail approaches and appropriate rate-limiting techniques to maintain service responsiveness.

Implementation guidance

Operating within 100 ms real-time bidding contracts, a single throttle and retry of five seconds can result in many failed bids and potentially insurmountable retry queues. Avoid this by adapting retries to fail fast. Regulate request rates using algorithms, such as token buckets, leaky buckets, or fixed window counters, or use managed service features, like Amazon API Gateway's request throttling. Rate limiting helps prevent resource exhaustion and fairly distributes resources among clients or services. Know the trade-offs: while rate limiting can be an effective way to protect a service from being overloaded, it can also potentially make the service less reliable if not implemented carefully. For example, if the rate limits are set too low, legitimate requests may be rejected or delayed, leading to reduced availability or responsiveness of the service.

Key AWS services

- Amazon API Gateway implements the token bucket algorithm to throttle requests according to account and region limits
- Amazon Simple Queue Service (Amazon SQS) and Amazon Kinesis can buffer requests to smooth out the request rate
- AWS WAF can also be used to implement rate limiting and throttle specific API consumers

Resources

Related documentation:

- Implementing layers of admission control
- API Gateway Request Throttling

ADVREL02-BP02 Implement a caching strategy

Implementing caching strategies enhances system reliability and performance. Evaluate different caching levels from client-side to server-side, and explore various caching solutions, including ElastiCache, third-party databases, and CDNs for optimizing ad payload delivery and reducing backend load.

Implementation guidance

Caching can be applied at various levels, such as client-side caching of user-profiles and server-side caching for bid enhancement. Distributed caching solutions include Amazon ElastiCache Redis or Memcached. Third-party databases such as Aerospike, Cassandra, and Scylla Cache are also commonly deployed for server-side caching. Ad Creative payloads are very effectively cached by CDNs, such as CloudFront, further reducing the load on web-servers.

Key AWS services

- Amazon ElastiCache is a fully managed in-memory data store
- Amazon API Gateway also provides a built-in caching layer
- AWS Lambda, a serverless compute service, can be used to implement caching at the application layer

Resources

Related documentation:

- Amazon ElastiCache (Memcached)
- Data Caching Across Microservices in a Serverless Architecture
- Caching for high-volume workloads with Amazon ElastiCache

ADVREL02-BP03 Prevent scale mismatch of both internal services and external partners

It's important to implement proportional scaling across all system components in advertising workloads. Balance service capacities, particularly in DSP to SSP integrations, and use pub/sub patterns to reliably distribute load and prevent service overload in microservices architectures.

Implementation guidance

Providing reliability is paramount for advertising workloads, which can be achieved by proportionally scaling all sub-components. For instance, when you integrate using a PrivateLink between DSP and SSP, your partner's requests may overwhelm your API front-end services, leading to throttling. To mitigate this when using a microservices architecture, the smaller services should drive larger capacity services, preventing them from being overwhelmed. The pub/sub pattern should also be followed wherever possible to enhance reliability through decoupled communication and load distribution across multiple subscribers. By implementing these measures, advertising workloads can maintain high availability and fault tolerance, providing a seamless and reliable experience for all stakeholders.

Key AWS services

- Amazon API Gateway
- Amazon SQS
- Amazon Kinesis
- AWS Lambda

Resources

- Avoiding overload in distributed systems by putting smaller service in control
- Pub/sub pattern

Design for single- and Multi-Region deployments

ADVREL03: How have you designed application to achieve reliability in single- and multi-Region deployments?

There are multiple strategies for achieving reliability in single and multi-Region deployments. Implement full Regional deployment using auto scaling and container orchestration. Select AWS Regions based on legal and disaster recovery requirements. Configure your databases with appropriate reliability and recovery strategies. Manage service capacity through proper resource reservation and load testing.

Focus on building resilient architectures that balance performance, compliance, and cost considerations.

Best practices

- ADVREL03-BP01 Use a full Regional deployment for compute resources through Auto Scaling groups and compute container orchestrators
- ADVREL03-BP02 Choose AWS Regions that meet your legal and disaster recovery requirements
- ADVREL03-BP03 Configure databases to span across multiple Availability Zones
- ADVREL03-BP04 Reserve appropriate capacity of services in the supported Regions

ADVREL03-BP01 Use a full Regional deployment for compute resources through Auto Scaling groups and compute container orchestrators

Deploy compute resources across multiple Availability Zones (AZs) and Regions to enhance application resilience. Implement zone-aware architectures to optimize performance and manage costs, and focus on intra-AZ communication and load balancing configurations.

Implementation guidance

Increase resiliency of real-time advertising applications by distributing resources across multiple Availability Zones or Regions, but maintain awareness of cross-AZ and cross-Region data transfer costs. When you use a full Regional deployment, implement zone-aware architectures within each Region to optimize performance and costs. When distributing resources across multiple Availability Zones for resilience, implement logic to prefer intra-AZ communication, when possible, and use features like AZ-aware load balancing to minimize cross-AZ traffic. By being zone-aware, companies can reduce costs and improve performance even when they need to operate in multiple Regions.

Key AWS services

- Amazon EC2 Auto Scaling groups can be configured to span multiple AZs
- Amazon Elastic Kubernetes Service (EKS) clusters can also be deployed across multiple AZs

Resources

- Regions and Availability Zones

- Distribute instances across Availability Zones
- EC2 Instance Meta-Data Retrieval
- Creating Kubernetes Auto Scaling Groups for Multiple Availability Zones | Containers
- Add an Availability Zone - Amazon EC2 Auto Scaling
- Simplify node lifecycle with managed node groups - Amazon EKS

ADVREL03-BP02 Choose AWS Regions that meet your legal and disaster recovery requirements

Select AWS Regions based on compliance and disaster recovery needs. It emphasizes the importance of understanding data jurisdiction requirements, particularly for advertising systems, and explains how regional choices impact both regulatory compliance (like GDPR) and system redundancy.

Implementation guidance

Depending on the resiliency design of your advertising system, some components may reside in a different Region for redundancy purposes. Consider compliance needs for your in-transit and at-rest data.

Key AWS services

- AWS Control Tower provides Region-deny capabilities
- AWS Managed Microsoft AD supports multi-Region deployment, allowing AD-aware applications and AWS services to connect to the local instances of the global directory
- AWS KMS allows you to replicate multi-Region keys into other Regions
- AWS services like Amazon S3 and Amazon RDS are designed to be resilient by spreading requests and data across multiple Availability Zones within a Region. However, for additional redundancy, you can deploy these services across multiple Regions to achieve isolation and avoid correlated failures

Resources

- Accelerate your multi-region strategy with Amazon DynamoDB: Part 1
- AWS Global Infrastructure

- Understand resiliency patterns and trade-offs to architect efficiently in the cloud
- Deny services and operations for AWS Regions of your choice with AWS Control Tower
- Design consideration for AWS Managed Microsoft Active Directory - Active Directory Domain Services on AWS
- Creating multi-Region replica keys - AWS Key Management Service
- Regional services - AWS Fault Isolation Boundaries
- Navigating GDPR Compliance on AWS

ADVREL03-BP03 Configure databases to span across multiple Availability Zones

Explore database configuration strategies for reliability and disaster recovery, such as periodic snapshots to warm standby solutions. Evaluate trade-offs between single-AZ and multi-AZ deployments, costs considerations, and specific recovery time objectives (RTO).

Implementation guidance

Carefully consider the trade-offs between disaster recovery strategies when configuring databases in multi-AZ and single-AZ deployments. While multi-AZ deployments offer high availability, they can incur significant cross-AZ data transfer costs.

For cost-sensitive workloads, consider implementing a single-AZ database cluster with the following resilience strategies:

1. **Periodic snapshots:** Implement frequent automated snapshots of your database. This approach provides point-in-time recovery capabilities with a relatively low RTO, typically in the range of 15-60 minutes, depending on the database size and recovery process.
2. **Read replicas:** Deploy read replicas in a different Availability Zone. While this incurs some cross-AZ data transfer costs, it's generally less expensive than a full multi-AZ deployment. In case of a primary Availability Zone failure, promote the read replica to become the new primary. This can reduce RTO to between five and 15 minutes.
3. **Cold standby:** Maintain a stopped database instance in another Availability Zone, and periodically update it with snapshots. This approach balances cost and recovery time, with an RTO of approximately 10-30 minutes.

For mission-critical applications, where minimal downtime is essential, consider:

1. **Warm standby:** Keep an active, scaled-down secondary database in another Availability Zone continuously updated using asynchronous replication. This approach offers a lower RTO (between one and five minutes), but at a higher cost than cold standby.

Choose the strategy that best aligns with your specific RTO requirements and budget constraints. Implement and regularly test your chosen disaster recovery process to verify that it meets your RTO targets.

For AdTech customers who require multi-region deployment for global resilience, use services like Amazon Aurora Global Database or Amazon DynamoDB global tables. These services provide Region-wide resilience with minimal impact on performance and manageable costs.

Regularly review and optimize your database architecture as your workload and requirements evolve. Always weigh the costs of potential downtime against the ongoing expenses of more resilient configurations.

Key AWS services

- Amazon Relational Database Service (Amazon RDS) provides a Multi-AZ deployment option
- Amazon DynamoDB
- Amazon Aurora
- Amazon ElastiCache

Resources

- Amazon RDS Multi-AZ
- Protect critical workload with Pod Disruption Budgets
- Using Amazon Aurora Global Database
- Amazon DynamoDB global tables
- What is Amazon Relational Database Service (Amazon RDS)?
- Multi-AZ DB instance deployments for Amazon RDS

ADVREL03-BP04 Reserve appropriate capacity of services in the supported Regions

Manage service capacity across multiple Regions. Perform regular load testing at five times your baseline RTB traffic levels to validate capacity requirements. Validate that appropriate reservations are made to handle normal operations, peak loads, and potential disruptions.

Implementation guidance

If your application is designed to scale out over multiple Regions, service could be disrupted by temporary resource constraints or other issues impacting a single Availability Zone or Region. Regularly perform load tests with at least five times the baseline of RTB traffic expectations to validate that allocated capacity meets low water mark, mean, and peak capacity projections. Based on the results of your load tests, make capacity reservation.

Key AWS services

- Amazon Route 53
- Amazon DynamoDB global tables
- Amazon S3

Resources

- AWS service quotas
- Quotas and constraints for Amazon RDS
- What to Consider when Selecting a Region for your Workloads
- Creating a Multi-Region Application with AWS Services – Part 1, Compute, Networking, and Security

Change management

ADVREL04: How do you prevent regression from changes in your application and infrastructure?

Changes to your advertising workload or its environment must be anticipated and accommodated to achieve reliable operation of the workload. Changes include those imposed on your workload such as spikes in demand, as well as those from within such as feature deployments and security patches.

Maintain reliability during application and infrastructure changes. Implement comprehensive testing (like regression, performance, and canary) in CI/CD pipelines to monitor impact on critical metrics. Additionally, use phased deployment strategies (like blue/green and rolling) to minimize service disruption and quickly recover from issues.

Best practices

- ADVREL04-BP01 Through your CI/CD pipeline, employ end-to-end regression, performance, and canary testing
- ADVREL04-BP02 Deploy new code or resources in staggered phases, separated by sufficient time, to verify that the changes are successful

ADVREL04-BP01 Through your CI/CD pipeline, employ end-to-end regression, performance, and canary testing

Integrate comprehensive testing methodologies into CI/CD pipelines for advertising workloads. Monitor key metrics like 5xx errors and latency, especially in RTB systems, and respond quickly to issues through immediate engagement and fast rollbacks.

Implementation guidance

For RTB at scale, the primary reliability metrics for availability are 5xx internal errors and elevated latency. If these metrics are breached, do not wait for impacts to ad effectiveness. Instead, fail fast and revert changes until the root cause of the issue can be identified and addressed.

Key AWS services

- AWS CodePipeline is a fully-managed continuous delivery service
- AWS Fault Injection Service is a fully-managed service that simulates real-world failures
- AWS Step Functions

Resources

- Deployment strategies
- Canary deployments
- Use CloudWatch Synthetics to Monitor Sites, API Endpoints, Web Workflows, and More
- Performing canary deployments and metrics-driven rollback with Amazon managed Service for Prometheus and Flagger
- Testing and creating CI/CD pipelines for AWS Step Functions

ADVREL04-BP02 Deploy new code or resources in staggered phases, separated by sufficient time, to verify that the changes are successful

Implement gradual, phased deployments to minimize risks and service impacts when updating systems.

Implementation guidance

When deploying new code or resources, it is possible for unintended results to occur. Various deployment strategies can be used to reduce frequency and service impact.

By making changes through a blue/green deployment methodology, you can significantly reduce the impact of any potential issues and avoid downtime.

When a blue/green deployment isn't possible, a rolling deployment methodology should be used to reduce the number of resources being modified simultaneously. With a rolling deployment, changes are made in small batches, with a pre-determined amount of buffer time between batches. If an issue occurs with the deployment, the unchanged resources can continue handling traffic, avoiding downtime.

Key AWS services

- AWS CloudFormation
- Amazon Elastic Container Service (ECS)

Resources

- Blue/Green Deployments on AWS

- Rolling deployments
- Deployment methods

Failure management

Failures are unavoidable, and every system eventually fails over time, especially in high volume advertising systems. Anticipate and manage failures before they happen through detection, testing, and quick recovery.

ADVREL05: How do you continuously evaluate the resilience of your advertising workload to meet availability and recovery requirements?

Continuous resilience evaluation of advertising workloads can be achieved through two main approaches.

Perform regular fault tolerance testing and assessment using tools like AWS Gamedays, Well-Architected Framework Reviews, and Support Countdowns. Additionally, create and test disaster recovery procedures through documented runbooks and restoration processes, which helps you quickly recover during incidents.

Best practices

- ADVREL05-BP01 Perform routine evaluation of your workload's fault tolerance capabilities
- ADVREL05-BP02 Create disaster recovery (DR) runbooks, and regularly test documented backup and restoration processes

ADVREL05-BP01 Perform routine evaluation of your workload's fault tolerance capabilities

Resiliency evaluations should not be considered a one-time effort, but a continuous part of any workload's lifecycle.

Implementation guidance

Your workload, as well as the environment (both regulatory and partner) in which it operates, is constantly changing. Make resilience a regular part of your feature delivery and operational

cadence throughout a workload's lifetime. Create a living document to track evolving processes, expectations, and improvements. Use AWS Gamedays, Well-Architected Framework Reviews, and Support Countdown engagements to improve reliability of advertising workloads. Coordinate with your various advertising partners and stakeholders to perform successful failover testing.

Key AWS services

- AWS Well-Architected Tool
- Fault Tolerance Analyzer Tool is an open-source tool that focuses specifically on identifying potential fault tolerance issues across different AWS services
- AWS Gamedays
- Support Countdowns

Resources

- AWS Countdown
- Build Your Own Game Day to Support Operational Resilience
- Best practices for handling EC2 Spot Instance interruptions
- Using the Fault Tolerance Analyzer Tool to Identify Potential Issues

ADVREL05-BP02 Create disaster recovery (DR) runbooks, and regularly test documented backup and restoration processes

Processes for backup, restoration, and failover of data should be documented and regularly tested to validate efficacy and understanding.

Implementation guidance

Advertising workloads are designed for low latency when accessing information. An unsuccessful or slow data restoration could result in negative impact to the workload. To mitigate the impact from data unavailability during a disaster, implement data backup mechanisms which can quickly make necessary data available. By documenting processes, incident response teams can address impactful events, while validation ensures that the processes will work when needed, and that team members are comfortable, and confident, in performing disaster response activities quickly.

Key AWS services

- AWS Elastic Disaster Recovery (DRS) is a service that can help design a DR solution, map applications and networks, and build and test a DR runbook
- AWS Config can be used to continuously monitor and record resource configurations
- AWS CloudFormation can detect drift in stacks that have been deployed

Resources

- Disaster recovery options in the cloud
- Orchestrate disaster recovery automation using Amazon Application Recovery Controller (ARC) and AWS Step Functions
- Testing disaster recovery

Architecture capacity

ADVREL06: How does the architecture handle challenges involved with tens to hundreds of partners running in a hybrid environment?

Identify the components in your workload which are outside your control, and the mechanisms which you have implemented to maintain your workload's availability when operations with those dependencies are disrupted.

Best practices

- ADVREL06-BP01 Architect defensively against failures

ADVREL06-BP01 Architect defensively against failures

Acknowledge that systems and services occasionally fail, and some failures will come from external partners and dependencies outside of your control.

Implementation guidance

Most advertising systems exist in hybrid configurations, with services and applications spanning across cloud and on-premise infrastructure. They use mechanisms across multiple Regions or data centers to provide high availability, scalability, and performance.

Understand the characteristics of your application components and how each component in a hybrid environment may impact your system as a whole. Be familiar with the complexity of deployment and operations across different types of environments and how that complexity can impact overall resilience.

Instead of using internet-based connections, use AWS Direct Connect where possible to provide a consistent network experience for critical workload networking requirements. Implement circuit breakers, retries, and fallbacks to gracefully handle failures from external dependencies, and prevent cascading failures within your system. Adopt a distributed architecture with loose coupling and asynchronous communication patterns to isolate failures and prevent them from propagating across the entire system.

To validate your resilience strategies and identify potential weaknesses, regularly conduct chaos engineering experiments by intentionally injecting controlled failures into your system.

Key AWS services

- Amazon Simple Queue Service (Amazon SQS)
- Amazon DynamoDB
- Amazon ElastiCache
- AWS Lambda
- Amazon API Gateway
- AWS Auto Scaling
- AWS Availability Zones and Regions
- AWS Elastic Load Balancing
- Monitoring and alerting

Resources

- Architecting for Reliability on AWS

- Implementing Microservices on AWS
- Disaster Recovery of Workloads on AWS: Recovery in the Cloud

Failure and recovery

ADVREL07: How do you handle component failures and disaster recovery?

Handle component failures and disaster recovery by designing fault-tolerant systems using cell-based architectures and distributed resources. Implement backup strategies aligned with RTO/RPO objectives. Manage data backup across multiple locations, and comply with required regulations.

Best practices

- ADVREL07-BP01 Design your workloads to withstand failures of individual components, such as compute instances, queues, databases, and caches
- ADVREL07-BP02 Implement a backup strategy which would meet RTO and RPO objectives
- ADVREL07-BP03 Back up data in multiple locations with consideration for your regulatory or legal requirements

ADVREL07-BP01 Design your workloads to withstand failures of individual components, such as compute instances, queues, databases, and caches

Build building resilient advertising systems by identifying critical components, and implement fault tolerance through cell-based architectures and distributed resources across Availability Zones.

Implementation guidance

Determine which components of your workload are in a critical path to maintain operations for real-time bidding, ad serving, and other crucial functions. Identify AWS services that provide built-in fault tolerance mechanisms which are within your workload's response time, RTO, and RPO targets. Use cell-based architectures, with resources spread across multiple availability zones, to reduce the scope of a disruptive event. Where consistent communications are necessary, implement static stability mechanisms to reduce the dependency on control plane actions.

Key AWS services

- Amazon Simple Queue Service (Amazon SQS)
- Amazon DynamoDB
- Amazon ElastiCache
- AWS Lambda
- Amazon API Gateway
- AWS Auto Scaling
- AWS Availability Zones and Regions
- AWS Elastic Load Balancing
- Monitoring and Alerting

Resources

- Reducing the Scope of Impact with Cell-Based Architecture
- Static stability using Availability Zones
- Control planes and data planes

ADVREL07-BP02 Implement a backup strategy which would meet RTO and RPO objectives

Develop comprehensive backup strategies, focusing on data classification and meeting Recovery Time Objective (RTO) and Recovery Point Objective (RPO) requirements through appropriate service selection.

Implementation guidance

Review the data related to your workload and classify the data according to usage, retention, and availability needs. Example classifications might be user profile info, campaign data, reporting data. Consider how those different data classes are used within your workload and how the availability of that data can impact your workload's operation. Use those classifications to determine the RPO and RTO requirements for your workload. Identify the AWS services that can meet your requirements, and deploy resources to the Regions or Availability Zones that can achieve your RTO and RPO targets. Test the backup and restoration process to verify that your backup and recovery strategies will work during a disruptive event.

Key AWS services

- AWS Backup
- Amazon EBS
- Amazon EC2
- Amazon Relational Database Service
- Amazon Elastic File System

Resources

- Disaster Recovery (DR) Architecture on AWS, Part II: Backup and Restore with Rapid Recovery
- Establishing RPO and RTO Targets for Cloud Applications

ADVREL07-BP03 Back up data in multiple locations with consideration for your regulatory or legal requirements

Back up data in multiple locations, and consider how consumer privacy laws may impact your data replication and storage plans.

Implementation guidance

Select AWS Regions for backup locations that satisfy your legal and business requirements. Consider how consumer privacy laws may impact your ability to replicate data which could contain personal data. Be aware of how countries where your workload operates regulate advertising and related data, and seek legal consultation when you are unsure of how regulations might apply to your workload. Use your understanding of those regulations to select AWS services and Regions. Seek legal counsel when in doubt.

Key AWS services

- AWS Backup
- AWS Key Management Service (KMS)

Resources

- Cloud security guidance

- Protecting your data with backups
- Amazon DynamoDB now helps you meet regulatory compliance and business continuity requirements through enhanced backup features in AWS Backup

Privacy

ADVREL08: How do you verify privacy while building reliable collaborations between multiple parties in your system?

Privacy-enhanced collaboration systems (such systems allow multiple parties to work together on advertising campaigns and data analysis while protecting individual user privacy and sensitive business data) require high reliability to maintain secure and consistent data sharing between parties while verifying data integrity and availability. Implementing reliable practices involves designing resilient architectures, maintaining data consistency, and verifying robust recovery mechanisms while preserving privacy. This includes redundancy in critical components, proper error handling, and maintaining service availability across different Regions.

Best practices

- ADVREL08-BP01 Design resilient architectures with privacy-preserving fault tolerance
- ADVREL08-BP02 Maintain data consistency and availability across collaboration workflows
- ADVREL08-BP03 Implement secure and privacy-preserving recovery mechanisms for collaboration workloads

ADVREL08-BP01 Design resilient architectures with privacy-preserving fault tolerance

Build resilient architectures that maintain data privacy in multi-party collaborations, focusing on fault-tolerant AWS Clean Rooms deployment, encrypted failover mechanisms, and privacy-preserving disaster recovery procedures.

Implementation guidance

- Deploy AWS Clean Rooms across multiple Regions with replicated privacy policies, differential privacy budgets, and encrypted collaboration configurations to facilitate continuous privacy-protected analytics during regional outages.
- Configure automatic failover for AWS Clean Rooms and Nitro Enclaves with cross-Region KMS key access, synchronized IAM roles, and validated privacy control restoration to maintain cryptographic isolation and data protection during service transitions.
- Implement privacy-aware error handling for data matching with encrypted retry queues, failed operation logging that preserves anonymity, and automatic termination of computations that cannot maintain privacy guarantees during processing errors.
- Deploy circuit breakers with privacy validation that fail-closed when privacy controls cannot be verified, monitor differential privacy budget exhaustion, and halt operations when cryptographic attestation fails in dependent services.
- Monitor AWS Clean Rooms privacy metrics including query result threshold compliance, privacy budget consumption rates, unauthorized access attempts, and cryptographic operation health with automated alerts for privacy policy violations.
- Use encrypted dead-letter queues for failed matching operations with privacy context preservation, secure purging policies for expired operations, and manual review processes that maintain data anonymization during failure analysis.
- Automate backup of privacy-protected datasets with cross-Region encrypted replication, privacy policy version control, differential privacy state preservation, and recovery procedures that validate privacy controls before data restoration.

Key AWS services:

- AWS Clean Rooms
- Amazon Route 53
- AWS Auto Scaling
- Amazon EventBridge

Resources

- Reliablity Design principles

- Disaster recovery best practices

ADVREL08-BP02 Maintain data consistency and availability across collaboration workflows

Data consistency and availability is critical when working with multiple stakeholders or workflows. Implement tools like versioning, logging, and health checks to verify that data remains consistent and available.

Implementation guidance

- Implement versioning for collaborative datasets and schemas.
- Use transaction logs for tracking privacy computation state.
- Configure cross-Region replication for critical data stores.
- Implement idempotency for matching operations.
- Set up health checks for collaboration service endpoints.
- Use read replicas for high-availability data access.
- Configure automated rollback procedures for failed operations.

Key AWS services

- Amazon DynamoDB
- Amazon S3
- AWS Lambda
- Amazon CloudWatch

Resources

- Guidance for Maximum Data Availability Architecture on AWS
- CAP theorem

ADVREL08-BP03 Implement secure and privacy-preserving recovery mechanisms for collaboration workloads

Ad data requires security and preservation of privacy. Implement tooling and systems that preserve secure data and avoid privacy breaches when collaborating with first and third parties, and verify that you have disaster recovery and automated backup mechanisms in place.

Implementation guidance

- Design recovery procedures that maintain data encryption throughout the process.
- Implement point-in-time recovery for privacy-protected datasets.
- Configure automated backup verification with privacy controls intact.
- Set up secure backup encryption key rotation policies.
- Establish recovery time objectives (RTOs) aligned with privacy requirements. This is because privacy requirements can significantly extend RTOs by adding mandatory verification and security steps.
- Implement secure state management for interrupted privacy computations. For example, if encryption fails during recovery, then halt the process rather than expose data; if privacy controls can't be verified then deny access until controls are restored; If secure state can't be maintained then terminate the computation safely
- Create automated disaster recovery testing procedures.

Key AWS services

- AWS Backup
- AWS KMS
- AWS Secrets Manager
- Amazon S3

Resources

- Data protection

Ad measurement and verification

ADVREL09: How do you verify reliable ad measurement and verification across digital advertising solutions?

Ad measurement and verification reliability requires robust systems to accurately track, validate, and report advertising performance metrics while accounting for fraud, viewability, and data quality issues. This includes implementing redundant verification methods, maintaining data accuracy across multiple solutions, and verifying consistent measurement capabilities even during system failures or data anomalies.

These reliability practices facilitate consistent and accurate ad measurement even during system stress or partial failures, while maintaining data quality and measurement integrity across the advertising solution.

Best practices

- ADVREL09-BP01 Implement redundant ad-verification systems with automated failover mechanisms
- ADVREL09-BP02 Establish robust data collection and validation pipelines for measurement accuracy

ADVREL09-BP01 Implement redundant ad-verification systems with automated failover mechanisms

Implement redundant ad-verification systems with automated failover capabilities. Use multiple verification providers and automated monitoring for continuous, reliable advertising measurement and validation.

Implementation guidance

- Deploy multiple third-party verification providers for cross-validation
- Implement automated failover mechanisms for measurement systems
- Use data quality checks and anomaly detection
- Maintain backup measurement methodologies

- Configure automated retry mechanisms for failed measurements
- Implement circuit breakers for degraded third-party services
- Set up monitoring and alerting for measurement system health

Key AWS services

- Amazon CloudWatch
- AWS Lambda
- Amazon EventBridge
- AWS Step Functions
- Amazon DynamoDB

ADVREL09-BP02 Establish robust data collection and validation pipelines for measurement accuracy

Build reliable data collection and validation pipelines, emphasizing real-time monitoring, automated reconciliation, and recovery procedures to maintain measurement accuracy in advertising systems.

Implementation guidance

- Implement data validation at collection points
- Set up real-time data quality monitoring
- Create automated data reconciliation processes
- Configure dead letter queues for failed events
- Implement idempotent processing for measurement events
- Establish clear data freshness SLAs
- Deploy automated data recovery procedures

Key AWS services

- Amazon Kinesis
- Amazon SQS

- Amazon S3
- AWS Glue
- Amazon EMR

Performance efficiency

The performance efficiency pillar includes the ability to use cloud computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.

The performance efficiency pillar provides an overview of design principles, best practices, and questions. You can find guidance on implementation in the **Performance Efficiency Pillar whitepaper**.

Design principles

The following design principles can help you achieve and maintain efficient advertising workloads in the cloud:

- **Design for optimized cost:** The key to cost optimization for advertising workloads is to minimize costs while you maintain a required level of performance and reliability.
- **Design for handling low latency, bursty, and spiky traffic:** Build a scalable architecture with automated scaling capacity to enable rapid increases and decreases of traffic gracefully. Cache application data content, when possible, to reduce response latency and the load on database clusters. Use containerized workloads and prebuilt container images for fast scaling and predictable performance. Choose server hardware optimized for memory and CPU for ultra-low latency needs.
- **Design for large data volumes and transactions:** Build a scalable distributed database for transactions while optimizing it for fast writes. Consider use of a distributed NoSQL database that can handle high write throughput with linear scalability. Consider compression techniques to optimize storage and an appropriate caching strategy to reduce database load for user profiles, target segments, and creatives. Use streaming services for ingestion and transportation of event data. Set up auto scaling databases to handle traffic spikes. Implement a data archive strategy to purge old ad impressions data to more cost optimized storage. Monitor database performance metrics, including latency, timeouts, and saturation, to identify and fix bottlenecks.
- **Design for data volume and query processing consideration for AWS Clean Rooms collaboration:** Large datasets can impact query performance. Consider data partition, aggregations, and filters to reduce result sets. Complex joins across multiple tables and number of collaborators can impact processing team. As a result, the optimal design for collaborators uses one to many collaboration channels, along with optimized pre-compiled query templates.

Architecture selection

The optimal solution for a particular workload varies, and solutions often combine multiple approaches. Well-Architected workloads use multiple solutions and allow different features to improve performance.

ADVPERF01: How does your architecture account for advertising workload needs like low latency and bursty traffic?

Focus on the architecture design needs for your advertising workload, including networking, compute, storage, machine learning workloads, decision between self-managed and managed services. Address the low latency, burst traffic, and scaling needs of advertising workloads

Best practices

- ADVPERF01-BP01 Design geographical affinity architecture with external entities (DSPs and SSPs)
- ADVPERF01-BP02 Use appropriate scaling to handle burst traffic with cost considerations
- ADVPERF01-BP03 Design for low latency with appropriate compute, storage, and network considerations
- ADVPERF01-BP04 Evaluate AI/ML-based architecture for optimization (like contextual advertising or scaling algorithms on event context)
- ADVPERF01-BP05 Evaluate the choice of open source-based software (self-managed) against using a fully-managed service

ADVPERF01-BP01 Design geographical affinity architecture with external entities (DSPs and SSPs)

Design for the least-network path, but keep regulatory needs in consideration. Use the AWS backbone network to improve latency.

Implementation guidance

Implement Amazon Route 53 (fail-over and geolocation routing) to route traffic to the target load balancers and compute workloads in the closest Region to the origination of intake requests.

This architecture may help align with specific compliance and residency needs. Consult with legal counsel for guidance tailored to your specific use case and jurisdiction.

Implement AWS PrivateLink on the same Region between external entities (like DSPs and SSPs) where both parties are on AWS.

For privacy-enhanced collaboration using AWS Clean Rooms, it is recommended to have collaborators in the same Region as the clean room to avoid latency with cross-Region data transfer.

Key AWS services

- Amazon Route 53 (R53)
- AWS PrivateLink
- AWS Clean Rooms

Resources

- Disaster Recovery Solutions with AWS managed services, Part 3: Multi-Site Active/Passive
- How Storygize and Sharethrough are using AWS PrivateLink to reduce costs and increase revenue

ADVPERF01-BP02 Use appropriate scaling to handle burst traffic with cost considerations

Consider start-up latency and scaling needs to handle burst traffic for networking, compute, and storage resources.

Implementation guidance

Network Load Balancer (NLB) and Application Load Balancer (ALB) scaling parameters depend upon the following parameters:

- Overall number of long-lived connections
- New TCP/TLS connections per second expected
- Data transfer in GB per second expected

NLB scaling needs are driven by elastic network interface at the Availability Zone level, whereas ALB scales across Availability Zones.

Consider Load balancer Capacity Unit (LCU) reservation, which you can use to proactively set a minimum capacity for your load balancer. This capability complements the load balancer's existing ability to auto scale based on your traffic pattern. Implement load balancers with target groups (like Auto Scaling groups).

For container workloads running on Amazon EKS, implement EKS Auto Scaling:

- Set up horizontal scaling and node scaling using either Cluster Autoscaler or Karpenter
- Set up pod scaling using horizontal pod scaling

Integrate with default Kubernetes metrics (like CPU and memory) or extensive metrics (inputs like queue lengths, CPU usage, and business metrics) using Kubernetes Event-driven Autoscaling (KEDA).

For databases like Amazon Aurora, enable storage auto scaling, which is a managed solution for storage expansion.

Key AWS services

- Amazon Network Load Balancer (NLB)
- Amazon Elastic Load Balancer (ELB)
- Amazon Elastic Kubernetes Service (EKS)
- Amazon Elastic Container Service (ECS)
- Amazon Aurora

Resources

- Auto Scaling benefits for application architecture
- Load Balancer Capacity Unit Reservation for Application and Network Load Balancers
- Autoscaling Amazon EKS services based on custom Prometheus metrics using CloudWatch Container Insights
- Autoscaling Amazon ECS services based on custom metrics with Application Auto Scaling
- How ktown4u built a custom auto scaling architecture using an Amazon Aurora mixed-configuration cluster to respond to sudden traffic spikes

ADVPERF01-BP03 Design for low latency with appropriate compute, storage, and network considerations

Use features from AWS compute, storage, and network services that cater to low latency advertising workload needs.

Implementation guidance

Consider the following guidance for compute, storage, and network:

Compute

- Use compute-optimized instances. Use benchmarking based on parameters like CPU, memory, launch time, and burst performance to choose the appropriate instance type.
- Cluster your EC2 instances into placement groups for ad serving components for the lowest possible latency between instances.

Storage

- Implement instance-attached SSD Amazon EBS volumes for lowest latency storage.
- Implement provisioned IOPS SSDs if you have an IOPS-intensive workload.
- Implement Amazon EFS for shared file storage with burst capability.
- Implement Elasticache Redis or Memcached to cache frequently accessed data.

Networking

- Implement enhanced networking for higher I/O and packet per second performance.
- Implement VPC endpoints to access AWS services within the network.

Resources

- Leveraging Amazon EKS managed node group with placement group for low latency critical applications
- New Amazon EC2 Instances (C7gd, M7gd, and R7gd) Powered by AWS Graviton3 Processor with Local NVMe-based SSD Storage
- Enhanced Networking

ADVPERF01-BP04 Evaluate AI/ML-based architecture for optimization (like contextual advertising or scaling algorithms on event context)

Use AWS services to implement a low latency, high throughput inference and MLOps framework.

Implementation guidance

- Implement low-latency, high-throughput model inference using Amazon ECS, Amazon EKS, and Amazon SageMaker AI.
- Implement an ML pipeline using Amazon SageMaker AI to build, train, and deploy machine learning models. Additionally, use Sage Maker for predictive scaling of compute based on learning from past event data.

Resources

Related documentation:

- Guidance for Machine Learning for Near Real-Time Advertising on AWS
- Guidance for Low-Latency High-Throughput Model Inference Using Amazon ECS

Related videos:

- AWS re: Invent 2020: Distributed machine learning for digital video and TV ad serving

ADVPERF01-BP05 Evaluate the choice of open source-based software (self-managed) against using a fully-managed service

Open source-based software is widely used by customers for advertising workloads. Carefully evaluate the factors for adoption of self-managed and managed services.

Implementation guidance

Adtech customers need to decide between self-managed and fully-managed services for container, databases, and analytics services in their workloads.

Evaluate the effect of both choices on performance of your workload from operational effort, infrastructure cost, customizability, high availability, and time to market. Create benchmarks for

performance using both options if needed, and choose the option that meets your performance requirements.

Key AWS services

- Amazon Elastic Kubernetes Service (EKS)
- Amazon Managed Streaming for Apache Kafka (MSK)
- Amazon DynamoDB
- Amazon Relational Database Service (Amazon RDS)
- Amazon SageMaker AI

Resources

- Migrating from self-managed Kubernetes to Amazon EKS? Here are some key considerations
- How to choose the right Amazon MSK cluster type for you
- Motivations for migration to Amazon DynamoDB
- Processing large records with Amazon Kinesis Data Streams
- Build an end-to-end MLOps pipeline using Amazon SageMaker AI Pipelines, GitHub, and GitHub Actions
- Choosing an AWS database service

Compute and hardware

The optimal compute choice for a particular workload can vary based on application design, usage patterns, and configuration settings. Architectures may use different compute choices for various components and allow different features to improve performance. Selecting a fitting compute choice for an architecture can improve performance efficiency.

ADVPERF02: How do you select and configure compute resources to optimize ISV compatibility, scaling, latency, and costs for ad workloads?

There are many compute choices to be considered for advertising workloads, with popular adtech ISV products, cloud native and machine learning needs, addressing low latency, burst traffic, and scaling design principles.

Best practices

- ADVPERF02-BP01 Evaluate compute benchmarks and compute options certified by the ISVs if applicable
- ADVPERF02-BP02 Consider containerization for scalability, low latency, and cost optimization
- ADVPERF02-BP03 Consider using low latency scaling tools like Karpenter to improve startup and scaling time
- ADVPERF02-BP04 Use a specialized instance family and features
- ADVPERF02-BP05 Evaluate ARM architecture for performance considerations by using AWS Graviton

ADVPERF02-BP01 Evaluate compute benchmarks and compute options certified by the ISVs if applicable

Evaluate ISV compatibility for running on AWS, and use the right resources based on published benchmarking results.

Implementation guidance

Aerospike's ISV product has been observed to be deployed for high-volume customer adtech workloads due to its speed at scale, real-time analytics capabilities, and strong data protection.

Databricks is a popular ISV platform used for advertising workloads due to its capabilities in big data processing, real-time capabilities and machine learning support. These facets make it well-suited for the large-scale and fast-changing needs of advertising analytics and intelligence.

Consider benchmark evaluation for Amazon EC2 Intel and Graviton instances for Aerospike and Databricks.

Resources

Related documentation:

- Running Ad Tech Workloads on AWS with Aerospike at Petabyte Scale

Related partner solutions:

- Database comparisons and performance benchmarks (Aerospike)
- Running operational workloads with Aerospike at petabyte scale in the cloud on 20 nodes
- Introducing the Well-Architected Data Lakehouse from Databricks6 Guiding Principles to Build an Effective Data Lakehouse
- Best Practices for Cost Management on Databricks

ADVPERF02-BP02 Consider containerization for scalability, low latency, and cost optimization

Adopt containerization as a strategy to operate at scale with low latency and cost optimization. Evaluate the various options of running container workloads on AWS.

Implementation guidance

Consider containerization, which helps improve application performance and helps scaling needs for adtech workloads, due to the following benefits:

- **Faster startup times:** Containers share the host OS kernel and start only the necessary processes, so they can start almost instantly compared to a full virtual machine (VM) startup. This makes scaling up and down faster.
- **Lower resource usage:** Containers require fewer resources than VMs, as there is no guest OS overhead. More efficient resource usage leads to cost optimization and the ability to run more container instances per host.
- **Portability across environments:** Container images can run on any infrastructure due to standardized runtime without need to re-optimize for different environments.
- **Scaling and availability:** Container orchestrators (for example, Amazon EKS) help to scale containerized apps, provide high availability, and improve performance under heavy loads.
- **Isolation:** Containers isolate processes and resources per application, reducing noisy neighbor issues on multi-tenant hosts for more predictable performance.
- **Utilization:** Higher density of containers per host allows full utilization of available resources, especially with auto scaling.
- **Microservices:** Decomposing monoliths into containerized microservices reduces interdependencies and allows independent scaling.

Key AWS services

- Amazon Elastic Compute Cloud (EC2)
- Amazon Elastic Kubernetes Service (EKS)
- Amazon Elastic Container Service (ECS)

Resources

- Leveraging Amazon EKS managed node group with placement group for low latency critical applications
- Amazon ECS vs Amazon EKS: making sense of AWS container services
- Under the hood: Lazy Loading Container Images with Seekable OCI and AWS Fargate
- Optimizing your Kubernetes compute costs with Karpenter consolidation

ADVPERF02-BP03 Consider using low latency scaling tools like Karpenter to improve startup and scaling time

Integrate observability metrics to initiate scaling of compute resources. Use open-source frameworks like Karpenter and KEDA, which provide for low startup latency scaling.

Implementation guidance

Karpenter (an open-source Amazon tool) for Kubernetes workloads can help with low-latency scaling and bursty traffic patterns for adtech workloads.

- **Faster node provisioning:** Karpenter can provision new nodes in a Kubernetes cluster much faster than traditional auto scaling methods, as Karpenter integrates directly with AWS APIs and can use services like Amazon EC2 Auto Scaling groups for rapid node provisioning.
- **Node pre-warming:** Although Karpenter does not support prewarmed node pools like Auto Scaling groups, you can use pod priority to maintain a pool of pre-initialized nodes. When new nodes are needed, Karpenter can quickly provision them from this pre-warmed pool, further reducing the latency associated with node provisioning.
- **Horizontal Pod Autoscaling (HPA) integration:** Karpenter can be configured to work in tandem with the Kubernetes Horizontal Pod Autoscaler (HPA). This integration allows Karpenter to provision new nodes proactively based on the HPA's scaling decisions, which makes resources available before pods start experiencing resource constraints.

- **Optimized node selection:** Karpenter can provision nodes with the appropriate instance types and resource configurations based on the requirements of the workloads. This optimization schedules pods on nodes with sufficient resources, minimizing the need for rescheduling or resource contention, which can introduce latency.
- **Parallel node provisioning:** Karpenter can provision multiple nodes in parallel, allowing it to rapidly scale out the cluster when faced with sudden spikes in demand. This parallelism helps minimize the overall latency associated with scaling operations.

Key AWS services

- Amazon Elastic Kubernetes Service (EKS)

Resources

- Manage scale-to-zero scenarios with Karpenter and Serverless
- Proactive autoscaling of Kubernetes workloads with KEDA using metrics ingested into Amazon Managed Service for Prometheus
- Scalable and Cost-Effective Event-Driven Workloads with KEDA and Karpenter on Amazon EKS

ADVPERF02-BP04 Use a specialized instance family and features

For advertising workloads, consider using a specialized instance family like compute-optimized for ad serving, storage-optimized for in-memory database, Trainium-based for machine learning (ML), and Inferentia-based for ML inferences.

Implementation guidance

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases.

The Amazon EC2 Compute Optimized instance family (C series) is a great match for compute-intensive workloads such as batch processing, media encoding, ad serving, bidding, and distributed analytics.

The Amazon EC2 Storage Optimized instance family (I series) are next-generation, storage-optimized instances designed to run applications that require high throughput and real-time latency access to data on local SSD storage. These instances help customers running real-time database workloads with Aerospike, where low latency local NVMe storage is required.

Amazon EC2 Accelerated Computing instances (powered by AWS Trainium) are purpose built for high performance, deep learning, and model training, while offering up to 50% cost-to-train savings over comparable GPU-based instances.

AWS Inferentia accelerators are designed by AWS to deliver high performance at the lowest cost in Amazon EC2 for your deep learning (DL) and generative AI inference applications.

AWS Nitro Enclaves enables customers to create isolated compute environments to further help protect and securely process highly sensitive data such as personally identifiable information (PII) and intellectual property data within their Amazon EC2 instances. Nitro Enclaves assist customers to reduce the threat surface area for their most sensitive data processing applications. Enclaves offers an isolated, hardened, and highly constrained environment to host security-critical applications. Nitro Enclaves enables a range of use cases that deal with the processing of highly sensitive data, such as securing private keys, tokenization, and multi-party collaboration. The isolation, cryptographic attestation, and integration with AWS Key Management Service capabilities of Nitro Enclaves are key features that provide customers with a practical approach to setting up multi-party collaboration.

Resources

- Choosing an AWS compute service
- Scaling distributed training with AWS Trainium and Amazon EKS
- AWS Inferentia2 builds on AWS Inferentia1 by delivering 4x higher throughput and 10x lower latency
- Introducing Unified ID 2.0 Private Operator Services on AWS Using Nitro Enclaves
- Use AWS Nitro Enclaves to perform computation of multiple sensitive datasets

ADVPERF02-BP05 Evaluate ARM architecture for performance considerations by using AWS Graviton

To address the low latency and high throughput needs of advertising workloads, consider adopting ARM architecture using AWS Graviton for improved performance and cost optimization.

Implementation guidance

Migrating to AWS Graviton processors can improve performance as a result of the following:

- **Faster processing:** Graviton uses 64-bit ARM Neoverse cores that are optimized for speed and efficiency in cloud workloads. Benchmarks show Graviton outperforming x86 instances for some workloads.
- **Lower latency:** The ARM architecture and custom memory subsystem in Graviton reduces latency for many operations compared to x86. This benefits real-time and latency-sensitive applications.
- **Improved throughput:** Graviton's support for new instructions like ARM Neon SIMD improves parallel processing throughput for workloads like video encoding and transcoding.
- **Enhanced networking:** Up to 25 Gbps of network bandwidth from the Nitro chip provides high throughput for network-intensive apps.
- **Burstable performance:** Graviton's TDP and credits system allows workloads to burst performance as needed.
- **Accelerated compression:** Hardware-based compression provided by the Nitro chip speeds up compressed workloads.
- **Caching optimizations:** Graviton optimizes cache utilization and memory access, leading to gains for memory bound workloads.

Key AWS services

- Amazon Elastic Compute Cloud (EC2)

Resources

- Optimizing for performance
- Considerations when transitioning workloads to AWS Graviton based Amazon EC2 instances
- Using Porting Advisor for Graviton

Data management

The optimal data management solution for a particular system varies based on the kind of data type (block, file, or object), access patterns (random or sequential), required throughput, frequency of access (online, offline, or archival), frequency of update (write once ready many (WORM) or dynamic), and availability and durability constraints. Well-Architected workloads use purpose-built data stores which allow different features to improve performance.

This focus area shares guidance and best practices for optimizing data storage, movement and access patterns, and performance efficiency of data stores.

Focus areas

- Data storage selection
- Data management design

Data storage selection

ADVPERF03: How do you choose the right storage services for your advertising workload?

In the fast-paced and data-intensive advertising industry, efficient storage solutions are crucial for handling large volumes of data generated from various sources, such as user interactions, advertising campaigns, and real-time bidding processes. With a wide range of purpose-built storage services available, you should understand the unique requirements of your advertising workload and choose the services that best align with your needs.

Best practices

- ADVPERF03-BP01 Choose appropriate block storage options to power your advertising workload
- ADVPERF03-BP02 Use object storage to store and analyze raw data from ad servers, DSPs, and DMP
- ADVPERF03-BP03 Use a cloud file system to store shared data between applications

ADVPERF03-BP01 Choose appropriate block storage options to power your advertising workload

Block storage is crucial for data storage in the cloud. Customers need to choose the appropriate block storage service based on different types of workloads, as well as their requirements for storage performance and stability.

Implementation guidance

Amazon EBS provides persistent block-level storage volumes for use with Amazon Elastic Compute Cloud (Amazon EC2) instances. In the advertising industry, Amazon EBS can be used to store

databases, such as MySQL or PostgreSQL, that power ad servers, bid management systems, and other critical components. Amazon EBS volumes can be easily scaled and optimized for different workload patterns, which provides high performance and reliability.

- **Volume types:** Choose the appropriate EBS volume type based on your workload. For general-purpose workloads, use GP3 volumes. For high-performance needs, consider IO2 volumes. If you need high performance, you'll need to use EC2 Instance Store. It's ephemeral block storage with a much higher performance than EBS.
- **EBS-optimized instances:** Use Amazon EBS-optimized Amazon EC2 instances to provide dedicated throughput between your instances and Amazon EBS volumes. For example, use Amazon EBS-optimized Amazon EC2 instances and provisioned IOPS volumes for real-time bidding or ad serving. workloads.
- **Encryption:** Enable encryption by default for all Amazon EBS volumes to meet security and compliance requirements.
- **Snapshot management:** Regularly create and manage Amazon EBS snapshots for backup and disaster recovery. Use AWS Data Lifecycle Manager to automate snapshot management.
- **Performance monitoring:** Use Amazon CloudWatch metrics to monitor and optimize EBS health and performance.
- **Scaling:** Leverage Amazon EBS Elastic Volumes to increase the size of Amazon EBS volumes dynamically without disrupting your applications.

Resources

- Amazon EBS volume types
- Amazon EBS volume performance
- Monitoring tools for Amazon EBS
- Automate backups with Amazon Data Lifecycle Manager
- What is Amazon Elastic Block Store?

ADVPERF03-BP02 Use object storage to store and analyze raw data from ad servers, DSPs, and DMP

Object storage can be used to store massive amounts of data while balancing cost and performance. Customers can use object storage services to build data lakes and analyze this data to uncover valuable insights and achieve business goals.

Implementation guidance

Amazon S3 is a highly scalable and durable object storage service that can store and protect any amount of data for a range of use cases. It is ideal for storing and serving static content, such as images, videos, and other media assets used in advertising campaigns. Amazon S3 also supports data lakes, which you can use to store and analyze vast amounts of raw data from various sources, including ad servers, demand-side platforms (DSPs), and data management platforms (DMPs).

- **Amazon S3 Express One Zone:** A powerful storage class for performance-critical applications, including advertising model training. Its low latency, high throughput, and cost efficiency makes it an ideal choice for real-time ad placement, machine learning for ad personalization, and interactive analytics.
- **Data partitioning:** Use multiple prefixes to partition your data, which distributes the load and improves performance. For example, instead of storing all objects under a single prefix, use multiple prefixes like `s3://bucket-name/prefix1/` and `s3://bucket-name/prefix2/`.
- **Data transfer:** Use Amazon S3 Transfer Acceleration to speed up data transfers over long distances, improving the performance of data ingestion and distribution processes.
- **Monitoring and auditing:** Use AWS CloudTrail and Amazon CloudWatch to monitor S3 access and performance metrics.
- **Storage tiering and class:** Each object in Amazon S3 has a storage class associated with it. Choosing a storage class designed for your use case lets you optimize storage costs, performance, and availability for your objects. Use the S3 Intelligent-Tiering storage class, which is designed to optimize storage costs by automatically moving data to the most cost-effective access tier when access patterns change, without operational overhead or impact on performance. S3 Intelligent-Tiering monitors access patterns and automatically moves objects that have not been accessed to lower-cost access tier.

Resources

- Getting started with S3 Express One Zone
- Setting an S3 Lifecycle configuration on a bucket
- Protecting data with server-side encryption
- Monitoring metrics with Amazon CloudWatch
- Manage Amazon S3 storage costs granularly and at scale using S3 Intelligent-Tiering

ADVPERF03-BP03 Use a cloud file system to store shared data between applications

File storage services (such as Amazon EFS) provide a simple way to set up and scale file systems and are widely used for big data and analytics workloads, media processing workflows, and content management scenarios. They are well-suited for distributed workloads and applications that need to share files across multiple EC2 instances.

Implementation guidance

Amazon EFS is a scalable and fully managed cloud file system that provides a simple, serverless way to share file data across AWS Cloud services and on-premises resources. In the advertising industry, Amazon EFS can be used to store and share log files, configuration files, and other data that needs to be accessed concurrently by multiple applications or instances. This is particularly useful for log processing and analysis pipelines, where data needs to be shared across multiple stages.

- **Performance modes:** Amazon EFS offers both General Purpose and Max I/O performance modes.
- **Throughput modes:** Choosing the correct throughput mode for your file system depends on your workload's performance requirements.
- **Cost optimization:** Use Amazon EFS lifecycle policies to automatically move infrequently accessed files to the EFS Infrequent Access storage class, reducing stor
- **High availability:** Create Amazon EFS mount targets in all availability zones to provide high availability and low latency access to your file system.
- age costs.

Resources

- Encrypting data in Amazon EFS
- Create an Amazon EFS file system and mount it on an Amazon EC2 instance using the AWS CLI
- Mounting considerations for Linux
- Managing automatic backups of Amazon EFS file systems

Data management design

ADVPERF04: How do you choose the right data management solution for your advertising workload?

Advertising workloads have different data classification, availability, and access latency needs based on their subsystems. Implement a strategy to identify an appropriate AWS data store service option, along with scaling and monitoring considerations.

Best practices

- ADVPERF04-BP01 Choose a data management strategy that matches your availability, latency, and access requirements
- ADVPERF04-BP02 Consider purpose-built and streaming databases
- ADVPERF04-BP03 Review your distributed database setup (sharding and replication) for performance, cost, and availability needs
- ADVPERF04-BP04 Enable detailed performance and observability monitoring to help tune queries and refine compute and storage
- ADVPERF04-BP05 Manage high volume user profile data
- ADVPERF04-BP06 Consider AWS Clean Rooms collaboration

ADVPERF04-BP01 Choose a data management strategy that matches your availability, latency, and access requirements

Customers need to have a clear data management strategy for their advertising workload datastores. The factors to consider are latency needs, availability needs which will help them chose the right AWS data service

Implementation guidance

The following are the most common data stores available in adtech:

- **User data:** Demographic data (age, gender, and location), behavioral data (browsing history, interests, and purchase history), and device data (device type, operating system, and browser).
- **Audience data:** Segmentation data (personas and target audiences) and geo-location data (IP addresses and GPS coordinates).

- **Campaign data:** Ad creative data (like images, videos, and text), ad placement data (websites, apps, and platforms), and campaign performance data (impressions, clicks, and conversions)
- **Inventory data:** Publisher data (website or app details and traffic data) and ad space data (ad sizes, formats, or placements)
- **Pricing and bidding data:** Bid data (bid prices and bid strategies) and auction data (bid landscape and winning bids).
- **Third-party data:** Data from Data Management Platforms (DMPs) and data from data exchanges or marketplaces.
- **Analytics and reporting data:** Conversion data (sales, leads, and actions), attribution data (tracking user journeys), and engagement data (view-through rates and dwell times)

For latency, consider the following:

- **Low-latency data (real-time or near real-time):** This data needs to be processed and acted upon within milliseconds to ensure optimal ad delivery, real-time bidding, and accurate tracking of user interactions.
  - Bid (bid requests, bid responses, and auction data)
  - User (device data, location data, and contextual data)
  - Ad impression (ad requests and ad responses)
  - Real-time campaign performance (clicks, impressions, and conversions)
- **Medium-latency data (near real-time or batch processing):** This data can be processed in near real-time (within minutes or hours) or in batches, as it is used for audience targeting, campaign optimization, and attribution analysis.
  - User behavior (browsing history and interests)
  - Audience segmentation
  - Campaign optimization (performance metrics and engagement data)
  - Attribution (user journeys and conversion paths)
- **High-latency data (batch processing or offline):** This data can be processed in batches or offline, as it is typically used for analysis, reporting, and long-term decision-making rather than real-time ad delivery or optimization.
  - Historical campaign
  - Detailed analytics and reporting
  - Third-party (from DMPs or data exchanges)

- Ad creative (images and videos)

Resources

- Architecture III: Picking the Right Data Store for Your Workload
- Amazon DynamoDB: Ad tech use cases and design patterns

ADVPERF04-BP02 Consider purpose-built and streaming databases

Purpose-built databases offer low latency and can better meet the scaling needs of advertising workloads.

Implementation guidance

Implement low-latency databases with in-memory AWS services (like Amazon DynamoDB or Apache Cassandra) or ISV products specialized for adtech (like Aerospike).

Implement medium latency data stores with an OLTP database like Amazon Aurora Global Database to implement a multi-Region availability design.

Resources

- Running Ad Tech Workloads on AWS with Aerospike at Petabyte Scale
- Use Amazon Aurora Global Database to build resilient multi-Region applications

ADVPERF04-BP03 Review your distributed database setup (sharding and replication) for performance, cost, and availability needs

Customers need to consider tradeoffs between performance, cost, and availability needs, while using features like sharding for scaling and replication for availability requirements.

Implementation guidance

Use availability zone affinity in Aerospike to allow client applications to access Aerospike nodes in the same zone, which optimizes data transfer across zones.

Distributed databases often support data partitioning or sharding, which allows you to split your data across multiple nodes or clusters. This can help distribute the load and optimize cost by reducing the need for high-performance instances or storage solutions for the entire dataset.

Carefully plan your data replication strategy across Availability Zones. While replication provides high availability and durability, replicating data across multiple Availability Zones can increase costs. Consider replicating only the essential data or implementing read replicas in different Availability Zones while keeping the primary node in a single Availability Zone.

Key AWS services

- Amazon RDS

Resources

- Architecture II: Distributed Data Stores
- Building globally distributed MySQL applications using write forwarding in Amazon Aurora Global Database
- Amazon Ads Architecture at Scale - ReInvent 2021

ADVPERF04-BP04 Enable detailed performance and observability monitoring to help tune queries and refine compute and storage

Provide access to necessary tools and metric granularity for performance debugging and compute and storage optimization, in particular because of the low latency requirements for advertising workloads.

Implementation guidance

Enable Amazon RDS enhanced monitoring, which provides deeper visibility into database performance and health. This heightened visibility helps you diagnose issues faster and optimize database workloads.

Enable Amazon EKS Container Insights to provide observability into cluster health, performance, logs, and billing for container workloads. This helps you run and optimize Kubernetes applications efficiently on Amazon EKS while reducing monitoring costs. The automated dashboards and analytics simplify troubleshooting.

Key AWS services

- Amazon CloudWatch

Resources

- Monitor real-time Amazon RDS OS metrics with flexible granularity using Enhanced Monitoring
- Optimizing AdTech end-user experiences Using Amazon CloudWatch Internet Monitor
- Tuning Amazon RDS for MySQL with Performance Insights
- Analyze Amazon Aurora MySQL Workloads with Performance Insights
- Announcing Amazon CloudWatch Container Insights with Enhanced Observability for Amazon EKS on EC2

ADVPERF04-BP05 Manage high volume user profile data

The user profile database is typically large, ranging from 100-200 million to 5 billion user profiles and contains a wide range of data about users' online activities and interactions. Hence this should be retained for a short time in the range of 30 days -1-year max, to manage data storage costs and data query latency SLO’s.

Implementation guidance

Use an in-memory database with a data cache strategy using Amazon MemoryDB.

Avoid replicating user profile data across multiple Regions due to high latency and data transfer costs. We recommend storing user profiles local to the user.

In the event of multi-Region architecture, implement synchronization between periodically (for example, once or twice a day) rather than in real-time, as users are unlikely to be in two locations at once. Advertisers also often use geotargeting, so a user's profile may only be accessed from the Region the user is located in for a particular ad campaign.

Key AWS services

- Amazon MemoryDB

Resources

- Observability best practices for Amazon Memory DB for Valkey

ADVPERF04-BP06 Consider AWS Clean Rooms collaboration

AWS Clean Rooms have limits on query result size (for example, AWS Clean Rooms has a 5GB limit), so consider using aggregations and filters to reduce result sets.

Implementation guidance

Large datasets can impact query performance. Partition data effectively.

A higher number of collaborators in a collaboration channel impacts processing time. Consider this as one of the factors for designing the collaboration framework with collaborators in play.

AWS Clean Rooms offers analysis templates work to support parameterized queries assisting in performance improvement through query reuse. Optimize queries before creating templates. Consider the choice of cryptographic operations for secure computation, as it adds to processing time.

Key AWS services

- AWS Clean Rooms

Resources

- Guidelines for the C3R encryption client

Networking and content delivery

The optimal networking solution for a workload varies based on latency, throughput requirements, jitter, and bandwidth. Physical constraints, such as user or on-premises resources, determine location options. These constraints can be offset with edge locations or resource placement.

On AWS, networking is virtualized and is available in a number of different types and configurations. This makes it easier to match your networking needs. AWS offers product features (for example, Enhanced Networking, Amazon EC2 networking optimized instances, Amazon S3 transfer acceleration, and dynamic Amazon CloudFront) to optimize network traffic.

AWS also offers networking features (for example, Amazon Route 53 latency routing, Amazon VPC endpoints, AWS Direct Connect, and AWS Global Accelerator) to reduce network jitter.

This focus area shares guidance and best practices to design, configure, and operate efficient networking and content delivery solutions in the cloud.

ADVPERF05: How do you build network architecture to provide efficient performance and improved user experience for your workload?

By using the various networking services provided by AWS, you can build a high- performance, low-latency, and highly available network architecture. For advertising workloads, a robust network architecture is particularly important, as all ad requests rely on a stable network to function properly.

Building a robust and efficient network architecture is crucial for delivering optimal performance and an exceptional user experience for advertising workloads on AWS.

Implementing best practices such as network segmentation, monitoring, automatic scaling, and load balancing further contributes to efficient performance and improved user experience.

Regularly review and optimize your network configuration, implement disaster recovery plans, and use content delivery networks like Amazon CloudFront to provide reliable advertising servies to your customers.

Best practices

- ADVPERF05-BP01 Establish private connections between your VPC and AWS services to improve performance
- ADVPERF05-BP02 Use edge services for static content caching and dynamic request acceleration to reduce latency and improve user experience
- ADVPERF05-BP03 Use load balancers to improve high availability and load distribution in your workload
- ADVPERF05-BP04 Provide dedicated network connection between your on-premises environment and AWS to offer high bandwidth and low latency

ADVPERF05-BP01 Establish private connections between your VPC and AWS services to improve performance

A private network not only enhances the overall stability and security of your system, but it also improves the latency and user experience for advertising customers.

Implementation guidance

Use AWS PrivateLink to establish private connections between your VPC and AWS services, such as Amazon S3, Amazon DynamoDB, or Amazon ElastiCache. This approach enhances security by avoiding the public internet and improves performance by reducing network hops and latency.

Resources

- Access AWS services through AWS PrivateLink
- Simplify private connectivity to Amazon DynamoDB with AWS PrivateLink
- AWS PrivateLink for Amazon S3
- AWS services that integrate with AWS PrivateLink

ADVPERF05-BP02 Use edge services for static content caching and dynamic request acceleration to reduce latency and improve user experience

Edge services can accelerate requests for static content as well as improve the response time for dynamic requests. By using the advantages of the cloud backbone network, it can maximize the efficiency and stability of access after requests enter the cloud.

Implementation guidance

If your advertising workload involves serving static content, such as images or videos, use Amazon CloudFront to cache and deliver your content from edge locations around the world. Amazon CloudFront reduces latency and improves user experience for your global audience by serving content from the nearest edge location.

Key AWS services

- Amazon CloudFront Regional Edge Caches (RECs)

- Amazon CloudFront Points of Presence (POPs)
- AWS Lambda@Edge

Resources

- Use an Amazon CloudFront distribution to serve a static website
- Ways to use CloudFront
- CloudFront configuration best practices
- Speeding up your website with Amazon CloudFront
- Customize at the edge with Lambda@Edge

ADVPERF05-BP03 Use load balancers to improve high availability and load distribution in your workload

Use the load balancing service provided by AWS to enhance the high availability of applications. In the event of disruptions that cause targets to become unhealthy, load balancers can automatically exclude unhealthy targets from traffic routing.

Implementation guidance

Elastic Load Balancing (ELB) employs various load balancing algorithms, such as round-robin, least outstanding requests, or IP hash, to distribute traffic evenly across healthy targets, which optimizes resource utilization and prevents overloading of individual targets. It supports content-based routing, which routes traffic based on the content of the request, such as the URL path or headers, efficiently handling different types of requests. ELB can offload SSL/TLS decryption and encryption from your targets, reducing the computational overhead on your application servers and improving overall performance.

Key AWS services

- Amazon Elastic Load balancer (ELB)
- Amazon Elastic Compute Cloud (EC2)

Resources

- What's the Difference Between Application, Network, and Gateway Load Balancing?

- Monitor your Application Load Balancers
- ELB Best Practices Guides

ADVPERF05-BP04 Provide dedicated network connection between your on-premises environment and AWS to offer high bandwidth and low latency

Use dedicated network connections to provide stable and high-speed data communication between the on-premises data center and the AWS Cloud. This model is also applicable for connections between multiple Regions, providing efficient and secure data communication while effectively avoiding public network noise.

Implementation guidance

For workloads that require high throughput or have strict compliance requirements, consider implementing AWS Direct Connect. AWS Direct Connect provides a dedicated network connection between your on-premises environment and AWS, offering high bandwidth, low latency, and enhanced security by bypassing the public internet.

Key AWS services

- AWS PrivateLink

Resources

- AWS Direct Connect Resiliency Recommendations
- Compliance validation for AWS Direct Connect
- Using the AWS Direct Connect Resiliency Toolkit to get started

Process and culture

When architecting workloads, there are principles and practices that you can adopt to help you better run efficient high-performing cloud workloads. This focus area offers best practices to help adopt a culture that fosters performance efficiency of cloud workloads.

ADVPERF06: How do you adjust your design patterns to maximize performance in your advertising workload?

Organizations that design advertising workloads should adopt a loosely coupled architecture and use new industry protocols.

Best practices

- ADVPERF06-BP01 Adopt a chipset-agnostic workload design for best availability of cloud resources and cost
- ADVPERF06-BP02 Optimize your intake request format (like HTTP/2 or HTTP/3) for faster processing

ADVPERF06-BP01 Adopt a chipset-agnostic workload design for best availability of cloud resources and cost

Implement an x86 chip-agnostic design for workloads to optimize the compute price of your advertising workload.

Implementation guidance

Adtech customers that use Amazon EC2 Spot Instances may have found that Spot Instance costs have swung between a preference towards AMD and Intel. As a result, implement a chipset-agnostic design, and make your design configuration-based for seamless adoption and to get the best compute price.

ADVPERF06-BP02 Optimize your intake request format (like HTTP/2 or HTTP/3) for faster processing

Use optimization in next generation networking protocols to address low latency needs for advertising workloads.

Implementation guidance

Implement HTTP/2 protocol, which offers features like multiplexing (multiple requests and responses are sent over the same TCP connection), header compression, and binary protocol. These features improve latency and throughput.

AWS services do support HTTP/2 and HTTP/3 protocols for gains in performance efficiency.

Key AWS services

- Amazon CloudFront
- Elastic Load Balancing

Resources

- New – HTTP/3 Support for Amazon CloudFront
- Application Load Balancers enables gRPC workloads with end to end HTTP/2 support

Cost optimization

The cost optimization pillar includes the ability to run systems to deliver business value at the lowest price point.

The cost optimization pillar provides an overview of design principles, best practices, and questions. You can find implementation guidance on implementation in the Cost Optimization Pillar whitepaper.

Design principles

The key to cost optimization for advertising workloads is to maintain profitable campaigns through a comprehensive system that can accurately track campaign costs against value. Optimization efforts should focus on minimizing unnecessary infrastructure costs and providing sufficient but not excessive capacity through intelligent auto scaling mechanisms. Given the large amounts of traffic involved in adverting workloads, even small optimizations can yield considerable cost savings. Implement effective anti-fraud solutions to reduce costs, and process and store data only from valid consumers, customers, and partners.

Practice Cloud Financial Management

ADVCOST01: How do you select and maintain cost-effective infrastructure for your advertising workloads?

Maintaining a cost-efficient infrastructure is essential for the profitability of advertising workloads.

Best practices

- ADVCOST01-BP01 Continually measure costs of different real-time bidding workloads, and adjust resource allocation accordingly
- ADVCOST01-BP02 Evaluate resiliency needs against the cost of downtime for ad delivery and bidding

ADVCOST01-BP01 Continually measure costs of different real-time bidding workloads, and adjust resource allocation accordingly

With fluctuations in usage over time, the costs associated with real-time bidding workloads can vary significantly. Continually monitoring costs is the best way to keep them under control.

Implementation guidance

- Set KPIs for each campaign to evaluate cost-to-revenue ratios, as this is key to measuring value generation.
- Set KPIs for billing metrics (for example, resource costs) as well as campaign metrics (for example, click-through rate or new subscribers).
- Implement cost allocation tags for resources relevant to campaign tracking.
- Use the Cost and Usage Dashboards Operations Solution (CUDOS) Dashboard as a way to quickly visualize information about RTB costs and performance.
- Use AWS Cost Explorer for one-off visualizations of cost data.
- Generate Quick dashboards that are specific to each campaign or that comprise the business as a whole.
- Configure Quick with user-configurable filters to allow users to focus on the data that matters most to them.
- Configure Quick to email dashboard reports to users on a schedule to automate and simplify the process.
- Regularly evaluate the data and report findings back to the business.
- As campaigns progress, continually re-evaluate them, and adjust resource allocation to meet value generation goals.

Key AWS services

- Amazon Athena
- AWS Data Exports

Resources

- Guidance for Deploying a Data Transfer Dashboard for AdTech on AWS

- Guidance for Capturing Advertising OpenRTB (Real-Time Bidding) Events for Analytics on AWS
- Using CUDOS Dashboard visualizations for AWS Marketplace spend visibility and optimization
- Additional dashboards
- Organizing costs using AWS Cost Categories

ADVCOST01-BP02 Evaluate resiliency needs against the cost of downtime for ad delivery and bidding

While resiliency can increase the cost of workloads, downtime can also be very expensive. It's important to understand the costs of having a resilient infrastructure against the costs of not having a resilient infrastructure.

Implementation guidance

- Quantify the cost of downtime for each campaign based on its expected revenue.
  - Analyze historical data and projections to estimate the potential revenue loss due to downtime.
  - Consider the impact on customer satisfaction and brand reputation.
- Estimate the cost of applying resiliency measures.
  - Evaluate the cost of additional resources required for multi-Regional deployments, backup, and recovery solutions
  - Use AWS tools like AWS Pricing Calculator for estimating costs of future resiliency efforts and Quick, Amazon Athena, AWS Cost and Usage Report, and AWS Cost Explorer for cost analysis and reporting.
- Compare the cost of downtime with the cost of resiliency measures.
  - If the potential lost revenue and reputation costs of downtime exceed the cost of resiliency, favor implementing resiliency measures.
  - Consider multi-regional deployments, backup and recovery solutions, and other resiliency best practices.

By following these steps, you can make informed decisions about implementing resiliency measures based on a cost-benefit analysis, using AWS tools and services to optimize your approach and ensure business continuity.

Key AWS services

- AWS Data Exports
- AWS Resilience Hub

Resources

- Stage 1: Set objectives

Cost-effective resources

ADVCOST02: How are you optimizing the costs associated with your bidding resources while providing the fastest performance?

Cost optimization isn't just about reducing costs, but about providing the performance required at the lowest possible cost.

Best practices

- ADVCOST02-BP01 Use ARM processors for faster and more cost-effective bidder nodes
- ADVCOST02-BP02 Use compression to reduce network traffic and storage costs
- ADVCOST02-BP03 Use provisioned resource allocation for campaigns with predictable capacity, and use dynamic allocation for unexpected capacity
- ADVCOST02-BP04 Use Spot Instances for cost-effective bidding-as-a-service workloads with flexible fault-tolerance mechanisms

ADVCOST02-BP01 Use ARM processors for faster and more cost-effective bidder nodes

ARM processors can combine lower costs and higher performance, which makes them a great consideration for cost optimization.

Implementation guidance

- Use AWS Compute Optimizer to identify the most cost-effective instance types for bidding workloads, and verify that ARM instances were considered.
- Use AWS Graviton instances, which are powered by ARM processors designed by AWS, for your cloud workloads running in Amazon Elastic Compute Cloud (Amazon EC2), AWS Lambda, containers, and various other services.
- Take advantage of the cost savings offered by Graviton instances, which generally cost less than comparable x86 instances.
- For custom software, recompile it for use on Graviton processors with the assistance of open-source tools like sse2neon and Porting Advisor for Graviton for compiled applications.
- For interpreted or JIT languages, they generally run as-is or with minimal modifications on Graviton processors.
- Conduct performance testing and benchmarking to verify that Graviton instances meet bidding workload requirements.

Key AWS services

- Amazon Cloudwatch
- Amazon Cost Explorer

Resources

- Use Graviton instances and containers
- How DeviceAtlas optimized Real-Time Advertising Price/Performance on AWS Graviton3
- Using Porting Advisor for Graviton
- AWS Unveils Next Generation AWS-Designed Chips

ADVCOST02-BP02 Use compression to reduce network traffic and storage costs

Using compression can reduce the amount of data transferred thus reducing network and storage costs.

Implementation guidance

- Use GZIP compression before transferring data to Amazon S3 to reduce traffic between Availability Zones and Regions, as well as traffic to the internet.
- Use snappy compression for Amazon Kinesis Data Streams to reduce the amount of data stored and transferred.
- Implement HTTP/2 for Application Load Balancers, Amazon API Gateway compression, and Amazon Managed Streaming for Apache Kafka (Amazon MSK).
- For databases, consider the following compression techniques to reduce storage costs:
  - Column-level compression
  - Table-level compression
  - Backup compression
  - Query result compression
  - Index compression
- Implement replication compression to reduce data transfer costs.
- Monitor the impact of compression on CPU utilization, and verify that the increased CPU costs do not exceed the network transfer costs saved.

Resources

- Cost-Optimizing your AWS architectures by utilizing Amazon CloudFront features
- Reduce network transfer time with connection compression in Amazon RDS for MySQL and Amazon RDS for MariaDB
- Enable payload compression for an API in API Gateway
- Custom Amazon MSK configurations
- Processing large records with Amazon Kinesis Data Streams
- What is AWS Transfer Family?

ADVCOST02-BP03 Use provisioned resource allocation for campaigns with predictable capacity, and use dynamic allocation for unexpected capacity

Provisioned capacity can provide the lowest cost per hour. However, for unpredictable workloads dynamic allocation can provide a lower overall cost of ownership.

Implementation guidance

Provisioned capacity and on-demand capacity are two different pricing models offered by various AWS services, including Amazon Kinesis Data Streams, Amazon DynamoDB, AWS Lambda, and Amazon Athena. The differences between the two models are the following:

- **Provisioned capacity:** With provisioned capacity, you reserve and pay for a specific amount of capacity in advance, regardless of whether you use it or not.
  - This model is suitable for workloads with predictable and consistent traffic patterns or when you have a baseline capacity requirement.
  - By provisioning capacity, you get dedicated resources and can achieve better performance and lower costs compared to on-demand capacity for sustained workloads.
  - Examples: DynamoDB provisioned throughput, Kinesis Data Streams provisioned capacity, Lambda provisioned concurrency, and Athena workgroup capacity.
- **On-demand capacity:** With on-demand capacity, you pay for the resources you consume on a per-use basis without any upfront commitment or reservation.
  - This model is suitable for workloads with unpredictable or bursty traffic patterns, where you don't have a consistent baseline requirement.
  - On-demand capacity provides flexibility and scalability, as you only pay for what you use, but it can be more expensive for sustained workloads compared to provisioned capacity.
  - Examples: DynamoDB on-demand capacity, Kinesis Data Streams on-demand capacity, Lambda on-demand concurrency, and Athena on-demand capacity.
- **Serverless capacity:** AWS offers technologies for running code, managing data, and integrating applications, all without managing servers.
  - Serverless technologies feature automatic scaling, built-in high availability, and a pay-for-use billing model to increase agility and optimize costs.
  - These technologies also eliminate infrastructure management tasks like capacity provisioning and patching, so you can focus on writing code that serves your customers.

- Examples: Amazon Aurora, Amazon Redshift, Amazon Neptune, Amazon OpenSearch Service, and Amazon Elasticache.

The choice between provisioned, on-demand, and serverless capacity depends on your workload characteristics, cost considerations, and performance requirements. Some general guidelines for making this choice are the following:

- If you have a predictable and consistent workload with a known baseline capacity requirement, provisioned capacity can provide better performance and cost savings for sustained usage.
- If your workload is highly variable, unpredictable, or bursty, on-demand or serverless capacity can offer more flexibility and scalability, but it may be more expensive for sustained usage.
- For short-term or temporary workloads, on-demand or serverless capacity may be more cost-effective because you don't have to pay for unused provisioned capacity.
- For long-running or mission-critical workloads with consistent traffic, provisioned capacity can provide better performance and cost savings.

Analyze your workload patterns, performance requirements, and cost considerations to determine the most suitable capacity model for your use case. Additionally, many AWS services offer auto scaling and capacity management features to help optimize resource allocation and costs based on actual usage patterns.

Resources

- Choose the data stream capacity mode
- Pricing for Provisioned Capacity
- Configuring provisioned concurrency for a function
- Serverless on AWS

ADVCOST02-BP04 Use Spot Instances for cost-effective bidding-as-a-service workloads with flexible fault-tolerance mechanisms

For workloads that can be interrupted, Spot Instances can provide high performance for a very low cost per hour.

Implementation guidance

By using Spot Instances and services like Auto Scaling groups and AWS Batch, you can achieve significant cost savings for your bidding-as-a-service workloads.

- **Spot Instance pricing:** Spot Instances are typically offered at a substantial discount compared to On-Demand Instance prices. The discount can range from 10% to 90%, depending on the instance type, region, and current demand. On average, you can expect to save around 70% on compute costs by using Spot Instances.
- **Auto scaling with Spot Instances:** By configuring your Auto Scaling groups to launch Spot Instances, you can benefit from the cost savings while maintaining the desired level of capacity and availability. Auto Scaling groups automatically replace interrupted Spot Instances, and your workload can continue running without disruption.
- **AWS Batch with Spot Instances:** For batch processing workloads, AWS Batch can use Spot Instances as the compute environment for your jobs. This can lead to significant cost savings, especially for compute-intensive or long-running batch jobs. AWS Batch automatically handles job retries and check-pointing, improving fault tolerance and efficient resource utilization.
- **Cost optimization strategies:**
  - **Instance right-sizing:** Regularly analyze your workload's performance and resource utilization to identify the most cost-effective instance types and sizes. Right-sizing your instances can lead to substantial cost savings without compromising performance.
  - **Spot Instance interruption handling:** Implement efficient strategies to handle Spot Instance interruptions, such as check-pointing long-running jobs or gracefully draining and restarting interrupted instances. This can help minimize wasted compute resources and associated costs.
  - **Spot Instance advisors:** Use AWS Spot Instance advisors or third-party tools to optimize your Spot Instance selection and bidding strategies. These tools can help you identify the most cost-effective Spot Instance pools based on historical pricing data and demand patterns.
- **Cost monitoring and optimization:** Continuously monitor your workload's cost and performance metrics using AWS Cost Explorer, AWS Trusted Advisor, and other monitoring tools. Identify cost optimization opportunities and implement them regularly to maximize your savings.

By implementing these strategies, you can potentially achieve significant cost savings while maintaining the scalability and performance of your bidding-as-a-service workloads.

It's important to note that while Spot Instances offer substantial cost savings, they are subject to interruptions based on AWS's capacity requirements. Therefore, it's crucial to implement proper

fault tolerance mechanisms and have a strategy to handle instance interruptions to ensure the reliability and availability of your bidding-as-a-service workloads.

Key AWS services

- Amazon Elastic Compute Cloud (EC2)
- AWS Fargate
- AWS Compute Optimizer

Resources

- Guidance for Building a Real Time Bidder for Advertising on AWS
- Beeswax Uses AWS to Cost-Effectively Process Millions of Bid Requests per Second
- AWS Fargate for Amazon ECS
- EC2 instance rebalance recommendations
- EC2 Fleet and Spot Fleet

Optimizing communication costs

ADVCOST03: How are you optimizing communication costs between SSPs and DSPs?

Communication between SSPs and DSPs can be significant and can lead to high communication costs.

Best practices

- ADVCOST03-BP01 Consider private communication channels between SSP and DSP
- ADVCOST03-BP02 When integrating SSPs and DSPs for programmatic advertising, co-locate the platforms
- ADVCOST03-BP03 Co-locate bidder and database nodes

ADVCOST03-BP01 Consider private communication channels between SSP and DSP

Private communication channels can help keep traffic secure while also reducing internet egress charges.

Implementation guidance

With AWS PrivateLink, you can establish secure, private communication channels between your SSPs, DSPs, and other AWS services or on-premises resources. This approach enhances security, reduces data exposure risks, and can improve performance for your programmatic advertising workloads, while simplifying your network architecture and reducing operational overhead. In cases where PrivateLink cannot be used, then Amazon VPC Peering, AWS Direct Connect, and AWS Global Accelerator can be considered.

Resources

- AWS lowers data processing charges for AWS PrivateLink
- Get started with AWS PrivateLink

ADVCOST03-BP02 When integrating SSPs and DSPs for programmatic advertising, co-locate the platforms

Keeping SSP and DSP components together can keep transactions fast while minimizing inter-AZ and inter-Region traffic charges.

Implementation guidance

When integrating SSPs and DSPs for programmatic advertising, use Network Load Balancer (NLB) to direct traffic from the SSP to the DSP within the same Availability Zone. This approach can help optimize costs while providing high performance and availability.

- **Deploy in the same Availability Zone:** Deploy your SSP and DSP components (such as bidding nodes) within the same Availability Zone based on expected traffic patterns to minimize cross-AZ data transfer costs and reduce network latency.
- **Use Network Load Balancer (NLB):** Use Network Load Balancer (NLB) to distribute traffic from the SSP to the DSP instances within the same Availability Zone. NLB is cost-effective for TCP traffic and can handle millions of requests per second.

- **Configure your NLB:** Set the cross-zone-load-balancing attribute to false, or use the appropriate routing policy to prioritize routing within the same Availability Zone. This approach routes traffic preferentially to bidder nodes within the same Availability Zone, reducing cross-AZ data transfer costs.
- **Monitor and optimize:** Regularly monitor your data transfer costs and traffic patterns across Availability Zones. Adjust your resource placement and NLB configurations as needed to optimize cost-effectiveness.
- **Use cost optimization tools:** Use AWS Cost Explorer, AWS Budgets, and AWS Cost Anomaly Detection to monitor and analyze your costs, set budgets, and receive alerts for potential cost anomalies.
- **Automate and scale:** Use AWS CloudFormation or AWS CDK to automate the provisioning and management of your SSP and DSP infrastructure, which helps you scale efficiently and consistently while maintaining cost optimization.

Resources

- Guidance for AdTech Private Network on AWS
- Announcing new AWS Network Load Balancer (NLB) availability and performance capabilities

ADVCOST03-BP03 Co-locate bidder and database nodes

Keeping bidder and database nodes together can help transactions occur quickly and can also reduce inter-AZ and inter-Region traffic charges.

Implementation guidance

To optimize costs when configuring advertising bidder nodes to communicate with database nodes within the same Availability Zone, consider the following guidance:

1. **Resource placement:** Carefully plan the placement of your bidder nodes and database nodes across Availability Zones. Co-locate bidder nodes and their corresponding database nodes within the same Availability Zone to minimize cross-AZ data transfer costs.
2. **Database configuration:** If using a managed database service like Amazon RDS, configure your database instances to use multi-AZ deployment within the same AWS Region. This separates the primary and standby database instances into separate Availability Zones, providing high availability while minimizing cross-AZ data transfer costs for your bidder nodes.

3. **Network configuration:** Configure your VPC and subnets to verify that bidder nodes and database nodes within the same AZ can communicate efficiently. Use private IP addresses, and avoid public IP addresses or internet gateways, which can incur additional data transfer costs.
4. **Caching and replication:** Implement caching strategies and read replicas for your database nodes to reduce the amount of data transfer required between bidder nodes and database nodes. This can further minimize cross-AZ data transfer costs.
5. **Monitoring and optimization:** Regularly monitor your data transfer costs and traffic patterns across AZs. Adjust your resource placement and network configurations as needed to optimize cost-effectiveness.
6. **Use cost optimization tools:** Use AWS Cost Explorer, AWS Budgets, and AWS Cost Anomaly Detection to monitor and analyze your costs, set budgets, and receive alerts for potential cost anomalies.

Key AWS services

Network Load Balancer (NLB)

Resources

- Exploring Data Transfer Costs for AWS Managed Databases

Database optimization

ADVCOST04: How are you optimizing user profile storage, access, and replication?

User profile databases tend to be large ranging from 100-200 million to 5 billion user profiles and contain a wide range of data about users' online activities and interactions. Storage and Access to this data can increase cost.

Best practices

- ADVCOST04-BP01 Consider lower cost storage for older User Profile data
- ADVCOST04-BP02 Consider multi-level caching for user profile data
- ADVCOST04-BP03 Store profiles in a single Region and replicate asynchronously

ADVCOST04-BP01 Consider lower cost storage for older User Profile data

As the 30 most recent days are most relevant, using DynamoDB can prioritize high performance for the most relevant data (typically within the last 30 days), and archiving to Amazon S3 can reduce costs for less relevant data.

For S3 profile data:

- Enable S3 Intelligent-Tiering on your bucket
- Configure lifecycle policies to transition older data
- Set up monitoring to track access patterns

1. **For DynamoDB:**
   - Implement TTL for old profile records
   - Create export jobs to move historical data to S3
   - Use S3 Lifecycle policies for long-term archival

Cost optimization best practices

- Regularly analyze data access patterns
- Use AWS Cost Explorer to track storage expenses
- Consider object size and retrieval frequency
- Implement tagging for better cost tracking

Key AWS services

- DynamoDB
- S3
- Intelligent Tiering

ADVCOST04-BP02 Consider multi-level caching for user profile data

DynamoDB Accelerator provides a powerful, cost-effective solution for caching user profile data by dramatically reducing read latency and minimizing direct database operations. By creating an in-memory caching layer, DAX can reduce DynamoDB read capacity unit (RCU) consumption, translating to significant cost savings for applications with high-frequency profile lookups. For user profile systems with repetitive access patterns, DAX automatically caches frequently retrieved items, delivering microsecond-level response times while substantially lowering infrastructure expenses.

The intelligent caching mechanism avoids redundant database queries, allowing organizations to optimize their database performance without complex manual caching implementations, making it an ideal solution for scalable, cost-conscious applications that require rapid access to user information.

Moreover, the seamless integration of DAX with existing DynamoDB architectures means minimal code changes are required to achieve these performance and cost benefits, providing an efficient path to enhanced application responsiveness and reduced operational costs.

1. Create a DAX Cluster:
    a. Select the same VPC as DynamoDB table
    b. Select node type (recommend r5.large for medium workloads)
    c. Configure cluster size (minimum 3 nodes for high availability)
    d. Set cache TTL
2. Modify application code to support DAX
3. Caching strategy implementation:
    a. Configure cache invalidation mechanisms
    b. Implement write-through or write-behind strategies
    c. Set appropriate TTL for cached items
4. Monitoring and optimization: CloudWatch metrics to track
    a. Cache hit or miss ratio
    b. Latency
    c. Consumed read capacity
    d. Error rates
    e. Recommended monitoring dashboard

5. Performance and cost optimization tuning:
    a. Adjust cluster size based on traffic
    b. Use reserved instances
    c. Implement intelligent caching
    d. Monitor and adjust regularly

Resources

- Reduce latency and cost in read-heavy applications using Amazon DynamoDB Accelerator

Key AWS services

- DynamoDB Accelerator
- ElastiCache (Redis OSS)

ADVCOST04-BP03 Store profiles in a single Region and replicate asynchronously

Generally, users will only be in one Region at a time and therefore will only be updating in one Region. As a result, schedule replication a few times a day with AWS Step Functions and AWS Lambda to meet the resiliency requirements for data while minimizing high latency and data transfer costs.

Implementation guidance

1. Develop a Lambda replication function.
2. Configure your Step Functions workflow.
3. Set up a Amazon CloudWatch event rule for scheduling.
4. Implement error handling.
5. Configure monitoring.
6. Test your replication workflow.

Key AWS services

- AWS Step Functions
- AWS Lambda

Privacy-enhanced collaboration

ADVCOST05: How are you implementing cost optimization for privacy-enhanced collaboration?

Privacy-enhanced collaboration in advertising technology empowers advertisers, publishers, and solutions to securely analyze and activate sensitive data such as user identities, behavioral signals, and conversion events, without revealing raw data. Achieving cost-efficient execution of these privacy-preserving workloads on AWS requires a dual focus: maintaining robust data protection through advanced privacy mechanisms while optimizing the underlying cloud infrastructure for financial efficiency.

Best practices

- ADVCOST05-BP01 Use cost efficient data types and configurations for collaborative data environments

ADVCOST05-BP01 Use cost efficient data types and configurations for collaborative data environments

Use efficient storage formats and streamlined query configurations to reduce unnecessary data scanning, duplication, and transfer costs in collaborative analytics environments.

Implementation guidance

- Use parquet or columnar formats with partitioning and compress datasets.
- Use standard SQL for lightweight or well-partitioned datasets.
- Avoid unnecessary cross-joins or full table scans.
- Use same-Region AWS Clean Rooms collaborations to minimize inter-Region transfer costs.

Key AWS services

- AWS Clean Rooms

Resources

- Data formats for AWS Clean Rooms
- Data Analytics Lens

Data transfer and processing

ADVCOST06: How do you monitor and manage data transfer and processing costs in your ad delivery, measurement, and anti-fraud pipelines?

In high-scale adtech environments where systems are engineered for low-latency and high-volume operations, fraudulent activity directly amplifies usage across compute, storage, and networking resources.

Best practices

- ADVCOST06-BP01 Design fraud detection pipelines to minimize redundant processing and optimize inference costs

ADVCOST06-BP01 Design fraud detection pipelines to minimize redundant processing and optimize inference costs

Design fraud detection workflows that avoid repeated evaluations by caching known outcomes, filtering threats as early as possible, and running only the necessary inference steps on cost-efficient compute resources.

Implementation guidance

- Enable AWS Cost Anomaly Detection with thresholds for each adtech microservice.
- Use Spot Instances and Managed Spot Training for SageMaker AI training jobs to identify malicious ads.

- Cache fraud evaluation results (for example, known creatives, IPs, or device IDs) using DynamoDB or ElastiCache to avoid reprocessing identical inputs.
- Implement AWS WAF rules for basic bot detection at edge (lowest cost).

Sustainability

The sustainability pillar provides design principles, operational guidance, best practices, and improvement plans to meet sustainability targets for your AWS workloads. You can find additional guidance on implementation in the Sustainability Pillar of the AWS Well-Architected Framework.

Design principles

When designing advertising workloads, consider the following principles to optimize your workloads for sustainability objectives:

- Set sustainability goals internally and with advertising partners to meet business objectives.
- Implement low-latency workloads only for time-critical business requirements.
- Use serverless computing, containerisation, and cloud-native technologies to scale resources dynamically.
- Region selection is a complex factor for implementing advertising workloads.
- Back up and archive data only when challenging to recreate.
- Establish an iterative process to review sustainability objectives and ensure workload usage is not exceeding service level agreements (SLAs).

Region selection

ADVSUS01: How do you select a region to optimize your advertising workloads for sustainability?

Select the AWS Region that optimizes sustainability and minimizes the environmental impact of your large-scale advertising workloads, considering factors such as energy mix, region efficiency, proximity to customers, and regulatory landscape.

Best practices

- ADVSUS01-BP01 Distribute data and workloads across Regions when necessary to minimize network usage and latency

ADVSUS01-BP01 Distribute data and workloads across Regions when necessary to minimize network usage and latency

When selecting regions to host workloads for sustainability, distribute data and workloads across multiple Regions to minimize network usage and latency, prioritising the most sustainable Regions available that leverage renewable energy sources. The millisecond latency of programmatic advertising workloads typically requires ad-servicing architectures be near consuming workloads. However, there is opportunity to consolidate data analysis for these workloads into fewer Regions.

Implementation guidance

- Identify the latency requirements for your workloads, and determine which AWS Regions can meet those requirements.
- From the eligible regions, select the one with the lowest carbon footprint, considering factors such as the energy mix (prioritize Regions with 100% renewable energy).
- Use AWS tools to measure and report your carbon footprint.
- Consolidate infrastructure needs for analytics workloads (real-time bidding, privacy-enhanced data collaboration, ad intelligence, and measurement) in fewer AWS Regions with 100% renewable energy.
- Use AWS services designed for energy efficiency, such as Amazon EBS gp3 volumes, Amazon EC2 Instances with AWS Graviton processors, and Amazon EC2 Tranium and Inferentia instances for AI workloads.
- Periodically review and optimize the regional distribution of workloads as new, more sustainable AWS regions become available, balancing sustainability goals with performance requirements.
- Aggregate analytical data in local regions and move the aggregates to the central reporting region when data needs to be centralized for business reasons.

Key AWS services

- AWS Customer Carbon Footprint Tool
- AWS Graviton Processors
- Amazon EBS volume types
- AWS Services by Region

Alignment to demand

ADVSUS02: How do you align SLAs with sustainability goals?

Verify that the service-level agreements (SLAs) for your advertising workloads incorporate sustainability metrics, incentives, and monitoring processes that align with and actively support your organization's environmental goals.

Best practices

- ADVSUS02-BP01 Break down system components to determine which are business critical and compare the trade-offs
- ADVSUS02-BP02 Identify redundant infrastructure and unnecessary data movement to reduce usage where possible

ADVSUS02-BP01 Break down system components to determine which are business critical and compare the trade-offs

When aligning SLAs with sustainability goals for advertising workloads, break down system components to identify business-critical elements, and evaluate trade-offs to balance SLAs with environmental objectives while minimizing waste.

Implementation guidance

- Categorize workloads by business impact, customer impact, and latency, monitor performance, and set SLA requirements accordingly to optimize resource allocation.
- For batch workloads like privacy-enhanced data collaboration, consider scheduling them to run during periods when the carbon footprint is lower, such as time of the day or week when more renewable energy is available or when demand is lower.
- For time-sensitive and business-critical workloads like real-time bidding, prioritize meeting SLA requirements, even if it means running during peak demand periods with a higher carbon footprint.

Key AWS services

- AWS Auto Scaling (Automatically scales resources)
- AWS Compute Optimizer (Recommends optimal compute resources)
- AWS Instance Scheduler (Schedules starting/stopping instances)
- AWS Spot Instances (Discounted spare compute capacity)
- AWS Graviton processors (Energy-efficient ARM processors)

ADVSUS02-BP02 Identify redundant infrastructure and unnecessary data movement to reduce usage where possible

Identify and eliminate redundant infrastructure components and unnecessary data movement within your advertising workloads, as this can help reduce resource usage, lower the overall carbon footprint, and improve sustainability-related key performance indicators (KPIs).

Implementation guidance

- Audit your advertising workload infrastructure to identify any redundant or underutilized resources, such as idle instances, oversized instances, or unnecessary data replication.
- Analyze data movement patterns and network traffic to identify opportunities for reducing data transfers, especially over long distances or between regions. Use Amazon CloudFront to cache and serve ad files closer to consumers.
- Implement auto scaling and right-sizing mechanisms to automatically adjust resource allocation based on actual workload demands, minimizing over-provisioning. For example, with real-time bidding workloads that use Amazon EKS, implement a scaling policy that is determined by the number of bids being served, which optimizes resource usage.
- Consolidate workloads and data storage where possible, reducing the overall infrastructure footprint and associated energy consumption. Implement lifecycle policies to remove old ad file assets that are no longer needed.
- Establish monitoring and reporting processes to track resource utilization, data movement, and sustainability KPIs over time, enabling continuous optimization.

Key AWS services

- AWS Trusted Advisor (Identify optimization opportunities)

- AWS Cost Explorer (Visualizes and analyzes cost/usage data)
- AWS Config (Monitors and records resource configurations)
- Amazon CloudFront (Cache and serve ad files)
- AWS CloudTrail (Logs API calls and events)
- AWS Auto Scaling (Automatically scales resources)
- AWS Lambda (Serverless computing)
- AWS Data Transfer Cost Estimator (Estimates data transfer costs)
- Amazon S3 Lifecycle (Remove unneeded ad assets)
- AWS Well-Architected Tool (Provides architecture best practices)

Data caching

ADVSUS03: What caching techniques are you using, and how do you store data in cache to reduce compute time and energy consumption?

Use caching techniques and store data in cache to reduce compute time and energy consumption for your advertising workloads.

Best practices

- ADVSUS03-BP01 Use caching techniques to prevent frequent data access

ADVSUS03-BP01 Use caching techniques to prevent frequent data access

Implement caching techniques to store frequently accessed data in cache, preventing repeated data retrieval and thereby reducing computing time and energy consumption for advertising workloads.

Implementation guidance

- Implement caching strategies for advertising content and data to minimize frequent data access and reduce computing time and energy consumption.

- Use AWS caching services like Amazon ElastiCache (for in-memory caching) and Amazon CloudFront (for content delivery network caching) to store frequently accessed data closer to the consumers, reducing latency and compute requirements.
- Consider using AWS Lambda@Edge and CloudFront Functions to run lightweight logic at edge locations, minimizing the need for data transfer to centralized servers and reducing overall energy consumption.

Key AWS services

- AWS Global Accelerator (for optimizing data transfer over the AWS Cloud)
- AWS Graviton processors (for energy-efficient compute instances)

Software and architecture

ADVSUS04: How are you implementing architectures that minimize the average resources required per unit of work?

Design and implement architectures that minimize the average resources (such as compute, storage, and network) required per unit of work for your advertising workloads.

Best practices

- ADVSUS04-BP01 Use batch processing for data cleansing and enrichment to create customer profiles
- ADVSUS04-BP02 Use serverless transaction processing

ADVSUS04-BP01 Use batch processing for data cleansing and enrichment to create customer profiles

Use batch processing for data cleansing and customer profile enrichment in advertising workloads. Schedule the batch jobs during periods of lowest carbon consumption to minimize resource usage and environmental impact.

Implementation guidance

- For workloads like privacy-enhanced data collaboration that involve data cleansing, enrichment, and customer profile creation, implement batch processing architectures to minimize resource usage.
- Use AWS services like AWS Batch and AWS Step Functions to queue up and schedule these batch jobs during periods when the carbon intensity is lower, such as times when more renewable energy is available or when demand is lower.
- Consider using AWS Graviton-based instances if supported, for batch processing workloads, if as they offer energy-efficient compute capabilities.
- Sample data sets when possible, to reduce compute, analytics, and data transfer needs.

Key AWS services

- AWS Instance Scheduler (for scheduling batch jobs during low-carbon periods)

ADVSUS04-BP02 Use serverless transaction processing

Implement serverless transaction processing, such as for ad measurement, to reduce the required unit of work and associated resource consumption for your advertising workloads. Proxy metrics, as defined in the Well-Architected Framework Sustainability Pillar, can be used to measure improvements from serverless use. For instance, instead of having long-running vCPU usage and partially-used volumes in a number of workload instances, use a serverless approach, so compute usage only occurs at the time of a transaction.

Implementation guidance

- For ad measurement workloads, use serverless architectures to minimize the required infrastructure and resources per unit of work.
- Implement services like Amazon API Gateway, AWS Glue, AWS Lambda, Amazon Kinesis Data Streams, and Amazon EMR Serverless to build event-driven, scalable, and efficient ad measurement pipelines.
- These services automatically scale up or down based on demand, improving resource utilization and reducing waste.
- Serverless architectures can help minimize idle resources, further contributing to sustainability goals.

Key AWS services

- AWS Graviton processors (for energy-efficient compute instances, if using EC2 instances)
- AWS Compute Optimizer (for optimizing resource utilization, if using EC2 instances)
- Proxy Metrics (AWS Sustainability Pillar)

Data management

ADVSUS05: How are you optimizing data storage and retrieval?

Data management involves provisioning the minimum amount of storage required to meet your workload's needs. For existing storage, understand the needs and usage of your data to set appropriate lifecycles for your data to move data to the most efficient type of storage. Delete unnecessary data, when possible, to reduce storage footprint.

Best practices

- ADVSUS05-BP01 Identify and remove redundant data across storage

ADVSUS05-BP01 Identify and remove redundant data across storage

Participants in the real-time advertising supply chain can accrue large volumes of data. Consider how you use data and data preservation as outlined in ADPERF04-BP01. Don't keep data that has no purpose, can easily be recreated, and expedite the removal of low value or short-lived data. Remove unwanted advertisement video, images, files, and any other associated data that is no longer needed.

Implementation guidance

- Optimize data handling needs based on workload requirements, and verify that it reflects the nature of the business and short-lived advertising content (delete or archive based on data class).
- Consider if duplicate ad files or versions of ad files are be being saved that can be easily recreated.

- Use Amazon S3 storage lifecycle rules to automate the expiration (and deletion) of draft ad content versions. For content that should be preserved for historical purposes, use Amazon S3 storage lifecycles to transition content to another storage class, such as Amazon Glacier.
- Amazon S3 Storage Lens can identify incomplete multipart uploads, buckets that have numerous noncurrent versions, and if lifecycle rules are not present. Storage Lens can also provide activity metrics to identify ad objects or even prefixes that are infrequently used.
- AWS Config can also identify if you have unused resources, such as EBS volumes.
- Use Amazon ECR lifecycle policies to expire old images used for real-time bidding containers.
- Evaluate how users are using data to eliminate use cases, dimensions, and queries that no longer provide value.

Hardware and services

ADVSUS06: How do you use the minimum amount and most efficient hardware to meet business needs?

Demand for advertising workloads can vary (for example, the number of transactions needed for supply ads and type of hardware that can be used for advertising use cases). Using the minimum amount and most efficient hardware reduces carbon footprint.

Best practices

- ADVSUS06-BP01 Shut down resources when not in use, and implement energy-efficient machine learning models
- ADVSUS06-BP02 Continuously monitor and right-size your AWS resources, and use the minimum resources required to meet your workload needs

ADVSUS06-BP01 Shut down resources when not in use, and implement energy-efficient machine learning models

Resources for machine learning may have real-time demands that fluctuate or not be needed at certain times, such as when data can be processed as a batch. Set machine learning workloads to respond to demand in real-time, including turning off or shutting down resources when not

needed. Use available tools to optimize the compute resources and models used for machine learning workloads.

Implementation guidance

- Organizations can use machine learning to draw insights on correlation and causation from data sets in order to optimize advertising activities. However, resources for data preparation, identity resolution, data collaboration, and creation of machine learning models do not need to run 24/7. Optimize and shut down these resources when not in use to reduce carbon emissions.
- When using Amazon SageMaker AI, customers can take multiple steps to optimize their compute usage:
  - Use Graviton-based instances when possible.
  - Amazon SageMaker AI Inference Recommender can specify the most performant instance type.
  - Inference optimization techniques can be applied to SageMaker AI models.
  - SageMaker AI can dynamically adjust the number of instances provisioned for a model in response to changes in your workload by using scaling policies.
- Use AI chips that provide the highest performance for training and inference, such as AWS Tranium and AWS Inferentia.

ADVSUS06-BP02 Continuously monitor and right-size your AWS resources, and use the minimum resources required to meet your workload needs

Monitoring workloads allows you to optimize and elastically scale your workloads to meet demand. Using serverless offerings can also help you automatically scale to reduce resource usage and improve the ability to meet sustainability targets. Consider how your requirements change based on advertising campaigns, and take advantage of the elasticity and agility of cloud to optimize your resource usage.

Implementation guidance

- Advertising SSPs and DSPs should use Amazon CloudWatch dashboards for visibility into active connections and bytes process per endpoint to drive resource usage.

- Use AWS Compute Optimizer to identify the optimal resources for workloads. For example, when using Amazon EMR to analyze ad impression and click-through data, Compute Optimizer can recommend the optimal EC2 instance types based on utilization data.
- Monitor boot time for improvements, such as pre-installing dependent libraries in container images for bidder processing.
- For downstream analytics and reporting of bidder transactions, use Amazon Kinesis Data Streams and Amazon Data Firehose to send data to Amazon S3. The use of a data stream enables faster responses and allows independent scaling for components of the real-time bidding architecture.
- Ad servers and click-through servers should be in Auto Scaling groups to automatically scale in when load is reduced.

Key AWS services

- AWS Lambda
- Amazon DynamoDB
- Karpenter (Open-Source Kubernetes cluster autoscaler built with AWS)

Process and culture

ADVSUS07: Do you have an iterative approach for sustainability updates, including in your development, testing, and deployment processes?

It is important to revisit your approach to sustainability as requirements changes and options continue to evolve. Include sustainability at each stage of the software development lifecycle and in both non-production and production environments. Continue to evaluate how you are meeting sustainability objectives and ensure your process evolves over time.

Best practices

- ADVSUS07-BP01 Incorporate an improvement process to reduce low utilization and idle resources or maximize the output from resources

ADVSUS07-BP01 Incorporate an improvement process to reduce low utilization and idle resources or maximize the output from resources

Advertising workloads are changing at a rapid rate. As changes are introduced consider which resources are the most efficient and where resources can be removed. Use automation to create and remove infrastructure as needed.

Implementation guidance

- Establish a cadence to revisit SLAs with advertising partners.
- Prioritize how to reduce use when over-provisioning is identified (for example, start with compute, then storage, then network usage).
- Continue to iterate with advertising partners on reducing the infrastructure needed for a minimum viable representation of production for testing.
- Use infrastructure as code (IaC) to set up a test environment, so they can be removed when a testing or staging environment is no longer needed but easily recreated when beneficial.

Resources

- Well-Architected Lab - Optimize Hardware Patterns and Observe Sustainability KPIs

Privacy-enhanced collaboration

ADVSUS08: How are you implementing sustainability for privacy-enhanced collaboration?

Privacy-enhanced collaboration in advertising technology enables more secure analysis of sensitive data between parties while protecting user privacy. Implementing sustainable practices for these workloads requires optimizing resource usage and energy efficiency while maintaining security and compliance requirements. This includes strategic scheduling of intensive computations, efficient data storage approaches, and smart resource allocation.

Best practices

- ADVSUS08-BP01 Optimize privacy workload processing patterns and resource allocation for sustainability

ADVSUS08-BP01 Optimize privacy workload processing patterns and resource allocation for sustainability

For privacy-enhanced collaboration, advertising workloads have specific sustainability considerations for combining first and third-party customer data directly.

Implementation guidance

- Schedule intensive privacy computations during periods of lower carbon intensity.
- Use batch processing for data cleansing and matching operations.
- Implement efficient data compression and formatting using formats such as Parquet.
- Leverage AWS Graviton processors for energy-efficient computing.
- Use serverless architectures for matching operations where possible.
- Implement auto scaling based on actual collaboration workload patterns.
- Configure Regional data aggregation before central processing to reduce transfer needs.

Key AWS services

- AWS Lambda
- AWS Graviton Processors
- AWS Auto Scaling

Resources

- Hardware and services
- AWS Clean Rooms

Ad intelligence, measurement, and security

ADVSUS09: How are you implementing sustainable practices for ad intelligence, measurement, and security?

For security, ad fraud detection and prevention systems require continuous monitoring and analysis of advertising traffic, which can be computationally intensive. Implementing sustainable practices while maintaining effective fraud prevention requires careful optimization of detection mechanisms and resource usage, especially for real-time bidding and campaign monitoring systems.

Best practices

- ADVSUS09-BP01 Optimize fraud detection systems for resource efficiency

ADVSUS09-BP01 Optimize fraud detection systems for resource efficiency

Fraud detection systems can perform efficiently and have a reduced carbon impact when using approaches such as intelligent sampling, scheduled analysis, and Regional detection.

Implementation guidance

- Use energy-efficient processing for continuous fraud monitoring. If running compute instances, select AWS Graviton processors.
- Implement intelligent sampling for fraud detection where appropriate to reduce computational overhead while meeting business requirements.
- Schedule intensive fraud pattern analysis during low-carbon periods.
- Use serverless architectures for variable detection workloads.
- Use efficient data storage patterns for fraud signals and patterns. Archive data that is not readily needed and remove data that is no longer required for compliance/security purposes.
- Use AWS Clean Rooms for measurement analysis across partners, with the ability to analyze data sets where they are, with no data movement.
- Implement caching for frequently accessed fraud detection rules.
- Configure Regional detection systems to minimize data transfer.

Key AWS services

- Amazon EC2 with Graviton processors
- AWS Lambda
- Amazon ElastiCache

- AWS Clean Rooms
- Amazon CloudWatch

Resources

- Hardware and services

Moderation and brand safety

ADVSUS10: How do you verify sustainability in content moderation and brand safety systems?

Content moderation and brand safety systems require continuous processing of ad content and placement context across multiple channels and formats. Implementing sustainable practices while maintaining effective moderation requires balancing real-time performance requirements with energy efficiency and resource optimization.

Best practices

- ADVSUS10-BP01 Optimize content moderation systems for sustainable operation

ADVSUS10-BP01 Optimize content moderation systems for sustainable operation

As content grows for organizations, optimizing content moderation systems can benefit sustainability-related key performance indicators (KPIs). Implement or build architectures that include efficient machine learning models, automated scaling, and optimized storage patterns.

Implementation guidance

- Use efficient machine learning models for content classification. Use AWS Inferentia chips when possible, for improved performance per watt.
- Implement batch processing for non-real-time moderation tasks.
- Configure regional content analysis to minimize data movement.

- Use caching strategies for frequently accessed moderation rules.
- Use energy-efficient computing resources, such as AWS Graviton, for moderation workloads.
- Implement automated scaling based on moderation demand using auto scaling rules and Amazon CloudWatch metrics.
- Optimize storage patterns for moderation results and audit trails. For workloads using Amazon S3, use Storage Lens for insights and recommendations to optimize storage use.

Key AWS services

- Amazon Rekognition
- AWS Inferentia
- Amazon SageMaker AI
- AWS Auto Scaling
- AWS CloudWatch
- Amazon ElastiCache
- Amazon S3 Storage Lens

Resources

- Optimize AI/ML workloads for sustainability: Part 1, identify business goals, validate ML use, and process data

Conclusion

This lens provided architectural guidance for designing and building efficient, cost-effective, and scalable video streaming advertising workloads in the cloud. We captured common architectures and overarching design tenets for video advertising workload needs. The whitepaper also discussed the AWS Well-Architected Framework pillars through an advertising lens, providing you with a set of questions to consider for new or existing workloads. Applying the Framework to your architecture helps you build efficient, cost effective and scalable video advertising systems.

Contributors

The following individuals and organizations contributed to this document:

- Abraham Lobo, Principal Technical Account Manager, Amazon Web Services
- Andrew Korshunenko, Senior Solutions Architect, Amazon Web Services
- Aydn Bekirov, Principal Technical Account Manager, Amazon Web Services
- Basil Badawiyeh, Senior Enterprise Support Manager, Amazon Web Services
- Ian Coleshill, Principal Solutions Architect, Amazon Web Services
- Jacky Bezalel, Senior Technical Account Manager, Amazon Web Services
- James Blatchly, Technical Account Manager, Amazon Web Services
- Jon Sukup, Senior Technical Account Manager, Amazon Web Services
- Matt Williams, Principal Solutions Architect, Amazon Web Services
- Naveen Kumar Jindal, Technical Account Manager, Amazon Web Services
- Rachana Sane, Technical Account Manager, Amazon Web Services
- Ryan Baker, Senior Technical Account Manager, Amazon Web Services
- Xiulei Zhu, Technical Account Manager, Amazon Web Services
- Arvind Raghunathan, Principal Operations Lead Solutions Architect, Amazon Web Services
- Brian Maguire, Principal Solutions Architect, Amazon Web Services
- Chris Geiger, Senior Solutions Architect, Amazon Web Services
- Derek Villavicencio, Cloud Optimization Success Solutions Architect, Amazon Web Services
- Gerry Louw, Principal Solutions Architect, Amazon Web Services
- Luke Notley, Senior Solutions Architect, Amazon Web Services
- Mahmoud Matouk, Principal Security Lead Solutions Architect, Amazon Web Services
- Nataliya Godunok, Cloud Optimization Success Solutions Architect, Amazon Web Services
- Ranjith Krishnamoorthy, Principal Solutions Architect, Amazon Web Services
- Bruce Ross, Lens Lead Solutions Architect Well-Architected, Amazon Web Services
- Stewart Matzek, Sr. Technical Writer Well-Architected, Amazon Web Services
- Madhuri Srinivasan, Sr. Technical Writer Well-Architected, Amazon Web Services
- Matthew Wygant, Sr. TPM Guidance, Amazon Web Services

Document revisions

Change	Description	Date
New lens version	New lens version with substantial best practice and content updates. New scenarios added.	December 9, 2025
Initial release	Initial release of the Video Streaming Advertising Lens.	April 3, 2025

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.