

Implementation Guide

Guidance for Clickstream Analytics on AWS



Guidance for Clickstream Analytics on AWS: Implementation Guide

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Guidance Overview	1
Features and benefits	2
Use cases	3
Terms and concepts	3
Architecture overview	6
Architecture diagram	6
Guidance end-to-end architecture	6
Ingestion module	7
Data processing module	9
Data modeling module	10
Reporting module	12
Analytics Studio	12
AWS Well-Architected pillars	14
Operational excellence	14
Security	14
Reliability	14
Performance efficiency	15
Cost optimization	15
Sustainability	15
Architecture details	16
Guidance components	16
Web console	16
Analytics Studio	16
SDKs	16
Data pipeline	17
AWS services in this guidance	18
Plan your deployment	21
Cost	21
Ingestion module	22
Data processing & modeling modules	25
Reporting module	12
Logging and monitoring	28
Additional features	28
Security	29

IAM Roles	29
Amazon VPC	29
Security Groups	29
Amazon CloudFront	30
Supported AWS Regions	30
Deployment	36
Prerequisites	36
Deployment in AWS Regions	36
Getting started	38
Step 1: Create a project	38
Prerequisites	38
Steps	38
Step 2: Configure data pipeline	39
Steps	39
Step 3: Integrate SDK	42
Steps	42
Generate sample data	43
Step 4: Analyze data	44
Steps	44
Pipeline management	46
Prerequisites	46
Ingestion	47
Ingestion endpoint	47
Data sink – Kafka	52
Data sink – Kinesis	53
Data sink - S3	54
Data processing	55
Data schema	55
Execution parameters	75
Processing plugin	78
Data modeling	80
Reporting	81
App registration	82
Pipeline maintenance	83
Monitoring and Alarms	83
Pipeline modification	83

Pipeline upgrade	84
SDK manual	85
Key features and benefits	85
Android SDK	86
Introduction	86
Integrate the SDK	86
Swift SDK	114
Introduction	114
Integrate the SDK	114
Data format definition	129
Preset events	133
Change log	148
Web SDK	148
Introduction	148
Integrate the SDK	148
Get started	150
Data format definition	157
Preset events	161
Event attributes	166
Google Tag Manager integration	175
Change logs	176
Sample project	176
References	176
Flutter SDK	176
Introduction	176
Integrate the SDK	177
Data format definition	184
Preset events	188
Event attributes	188
Change log	188
References	189
ReactNative SDK	189
Integrate the SDK	189
Data format definition	197
Preset events	197
Event attributes	197

Change log	197
References	198
HTTP API	198
Request endpoint	198
API Specification	198
Request code example	206
Verification data reported successfully	209
User identifier	209
User ID	209
Device ID	210
User Pseudo ID	211
Migrate from third-party SDKs	212
Introduction	212
Step 1: Integrate Clickstream Web SDK	213
Step 2: Encapsulate common data logger methods	213
Step 3: Migrate to common APIs in minutes	215
Summary	216
Analytics Studio	217
Modules	217
Terms and concepts	217
Dashboard	218
Overview	218
Acquisition report	219
Engagement report	227
Retention report	232
Device report	236
Details report	240
Custom report	241
Exploration	246
Access Exploration	246
How Exploration works	246
How to use Exploration	247
Event Analysis	249
Funnel Analysis	251
Path Analysis	254
Retention Analysis	256

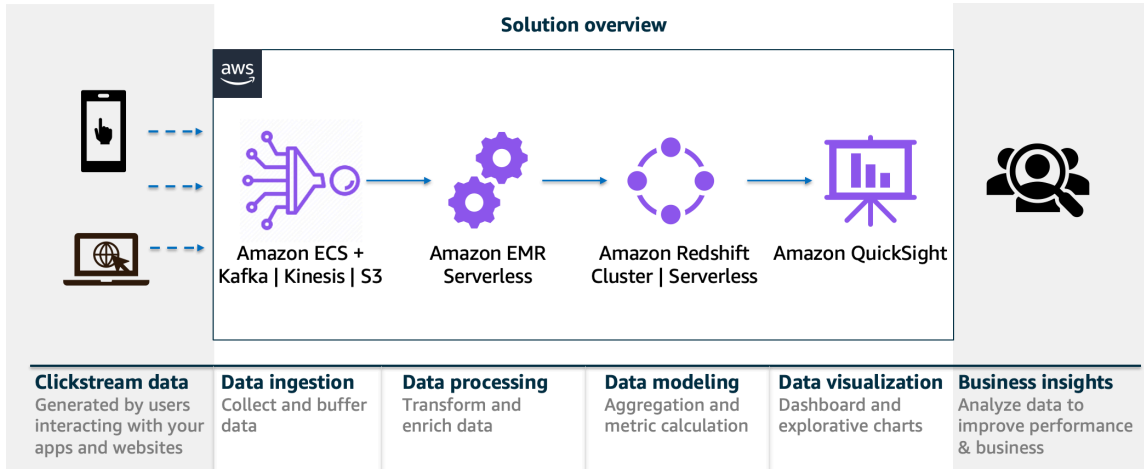
Attribution Analysis	259
Analyses	263
Access Analyses	264
How it works	264
Data management	265
Access Data Management	265
Metadata management	265
Metadata dimensions	265
Update event display name and description	268
Customize data dictionary for Event Parameter values	268
Traffic source	269
Processing	271
Configurations	272
Identity Management	279
User management	279
User roles	279
User roles management	280
Upgrade the guidance	283
Planning and Preparation	283
Upgrade Process	283
Upgrade web console stack	283
Upgrade the pipeline of project	285
Post-Upgrade Actions	285
Ingestion	285
Data Modeling	286
Frequently Asked Questions	290
General	290
Data pipeline	290
SDK	293
Analytics Studio	293
Pricing	297
Troubleshooting	298
Problem: Deployment failure due to "Invalid Logging Configuration: The CloudWatch Logs Resource Policy size was exceeded"	298
Problem: Cannot delete the CloudFormation stacks created for the Clickstream pipeline	299
Problem: Reporting stack(Clickstream-Reporting-xxx) deployment fail	300

Problem: Clickstream-DataModelingRedshift-xxxxx stack upgrade failed in UPDATE_ROLLBACK_FAILED	300
Problem: Can not sink data to MSK cluster, got "InvalidReplicationFactor (Broker: Invalid replication factor)" log in Ingestion Server	301
Problem: data processing job failure	301
Problem: data loading workflow failure due to meeting the 25,000 events limit in a single execution history	302
Uninstall the guidance	303
Step 1. Delete projects	303
Step 2. Delete Clickstream Analytics on AWS stack	303
Additional resources	304
Upload SSL Certificate to IAM	304
Developer guide	305
Source code	305
Contributors	306
Revisions	307
Notices	308

An end-to-end guidance to collect, ingest, analyze, and visualize clickstream data inside your web and mobile applications

The Clickstream Analytics on AWS guidance allows you to collect, ingest, process and analyze clickstream data from websites and mobile applications to drive your business growth. You can use the guidance to spin up an analytics platform that fits your organizational needs, and maintain complete ownership and control over your valuable user behavior data. The solution can be applied to various use cases such as user behavior analysis and marketing analysis to improve website and application's performance.

The guidance provides modularized and configurable components of a data pipeline so that you can choose and customize the components to accelerate the building of a Well-Architected data pipeline from weeks to minutes. The purpose-built SDKs and guidance allow you to collect client-side data from different application platforms (for example, Android, iOS, and JavaScript) to AWS. In addition, ready-to-use dashboards and explorative analytics studio enable you to derive actionable business insights easily and quickly.



Use this navigation table to quickly find answers to these questions:

If you want to ...	Read...
Know the cost for running this guidance	Cost

If you want to ...	Read...
Understand the security considerations for this guidance	Security
Know which AWS Regions are supported for this guidance	Supported AWS Regions
Get started with the guidance quickly to build an end-to-end data pipeline, send data into the pipeline, and then view the ready-to-use dashboards	Getting started
Learn the concepts related to pipeline, and how to manage a data pipeline throughout its lifecycle	Pipeline management
Practice how to customize analytics dashboard in an intuitive manner	Analytics studio

This guide is intended for IT architects, developers, DevOps, data analysts, and marketing technology professionals who have practical experience architecting in the AWS Cloud.

Features and benefits

The guidance includes the following key features and benefits:

- **Visual data pipeline builder**

You can simply define the data pipeline from a web-based UI console. The guidance will take care of the underlying infrastructure creation, required security setup, and data integrations. Each pipeline module is built with various features and designed to be loosely-coupled, making it flexible for you to customize for specific use cases.

- **Purpose-built SDKs**

The SDKs are optimized for collecting data from Android, iOS, and JavaScript platforms, which automatically collect common events (for example, first visit, screen view), support built-in local cache, retry, and verification mechanisms to ensure high completeness of data transmission.

- **Out-of-the-box dashboard**

The guidance offers a dozen of built-in visualizations (for example, acquisition, engagement, retention, and user demographic) and exploratory reporting templates (for example, user details, event details), powering various critical business analytics use cases such as user behavior analytics, marketing analytics, and product analytics.

Use cases

Clickstream data play a pivotal role in numerous online business analytics use cases, and the Clickstream Analytics on AWS can be applied to the following:

- **User behavior analytics**

Clickstream data in user behavior analytics provides insights into the sequential and chronological patterns of user interactions on a website or application, helping businesses understand user navigation, preferences, and engagement levels to enhance the overall product experience and drive product innovation.

- **First-party customer data platform (CDP)**

Clickstream data, together with other business data sources (for example, order history, and user profile), allow customers to create a first-party customer data platform that offers a comprehensive view of their users, enabling businesses to personalize customer experiences, optimize customer journeys, and deliver targeted marketing messages.

- **Marketing analytics**

Clickstream data in marketing analytics offers detailed information about users' click paths and interactions with marketing campaigns, enabling businesses to measure campaign effectiveness, optimize marketing strategies, and enhance conversion rates.

Terms and concepts

This section describes key concepts and defines terminology specific to this guidance:

Project

A project in this solution is the top-level entity, like a container, that groups your apps and data pipeline for collecting and processing clickstream data. One project contains one data pipeline, and can have one or more apps registered to it.

Data pipeline

A data pipeline is deployed into one AWS region, which means all the underlining resources are created in one AWS region. A data pipeline in this solution contains four modules:

- **Ingestion server:** a web service that provides an endpoint to collect data through HTTP requests, and sink the data in streaming services or S3.
- **Data processing:** a module that transforms raw data to the solution schema and enriches data with additional dimensions.
- **Data modeling:** a module that aggregates data to calculate metrics for business analytics.
- **Reporting:** a module that creates metrics and out-of-the-box visualizations in QuickSight.

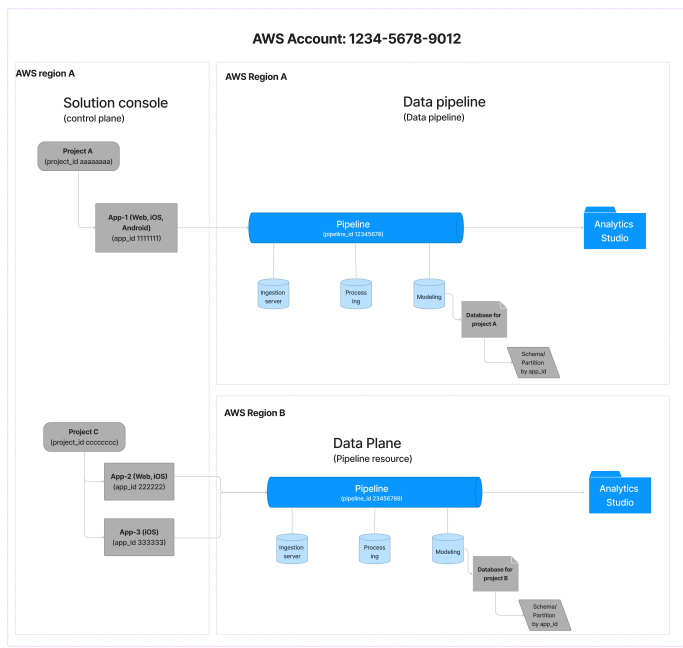
App

An app in this guidance can represent an application in your business, which might be built on one or multiple platforms (for example, Android, iOS, and Web).

Analytics Studio

Analytics Studio is a web console for business or data analysts to view dashboards, query, and manage clickstream data.

Below is a diagram to help you better understand those concepts and their relationship with each other in the AWS context.



For a general reference of AWS terms, see the [AWS glossary](#) in the *AWS General Reference*.

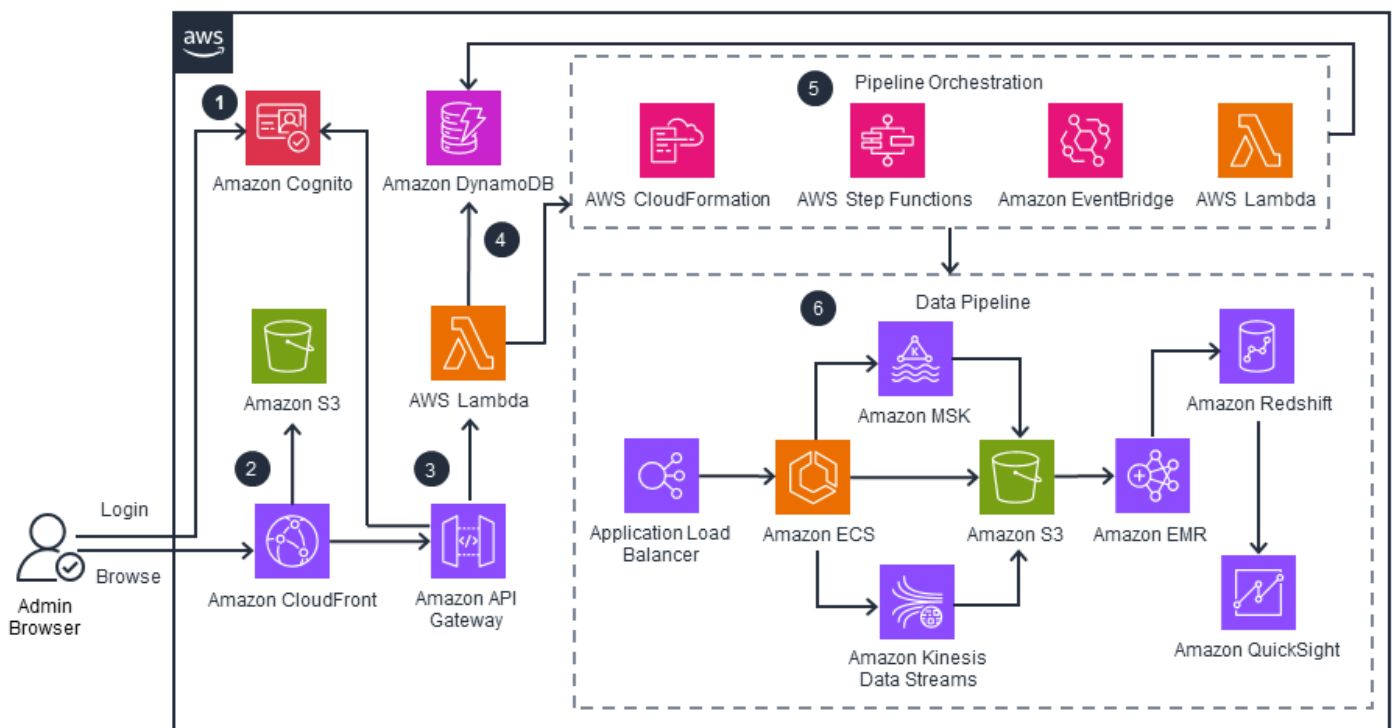
Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this guidance.

Architecture diagram

Guidance end-to-end architecture

Deploying this guidance with the default parameters builds the following environment in AWS:



Clickstream Analytics on AWS architecture

This guidance deploys the AWS CloudFormation template in your AWS Cloud account and completes the following settings.

1. An [Amazon Cognito](#) user pool or OpenID Connect (OIDC) handles authentication.
2. [Amazon CloudFront](#) distributes the frontend web UI assets hosted in the [Amazon S3](#) bucket.
3. [Amazon API Gateway](#) manages the backend APIs and routes traffic to [AWS Lambda](#).

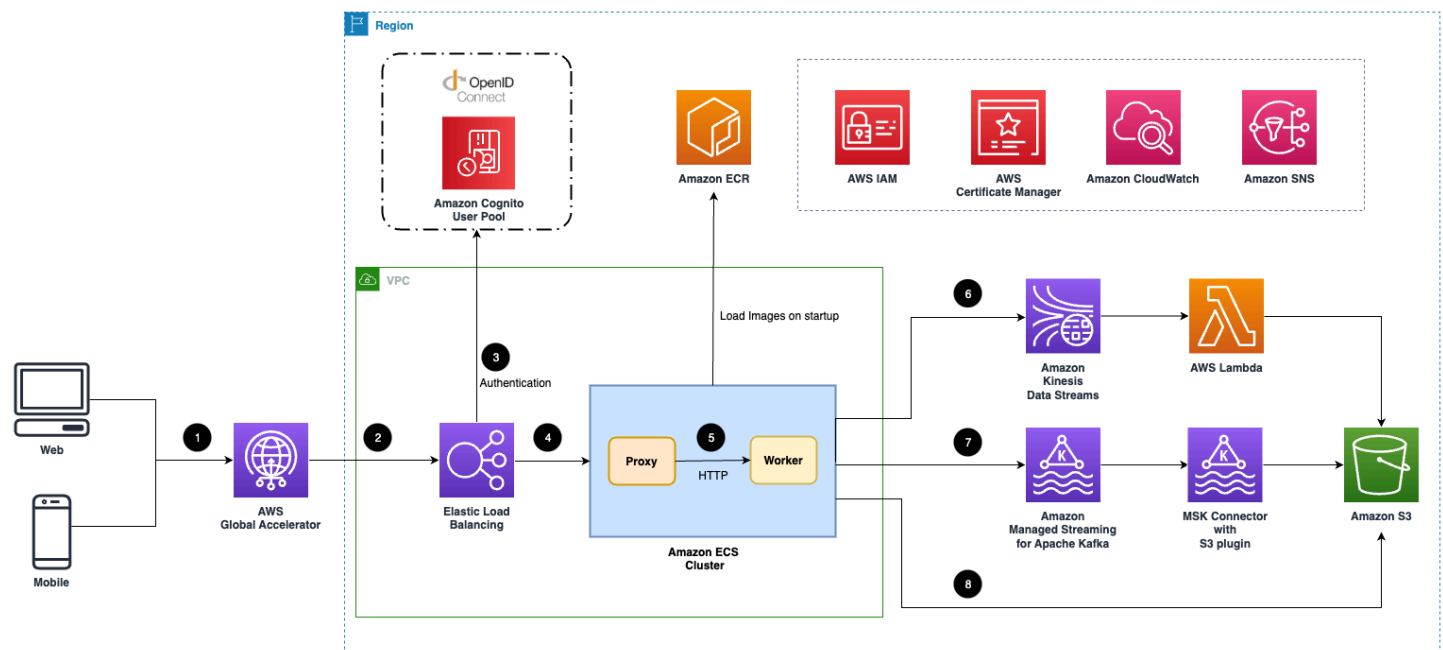
4. [Amazon DynamoDB](#) stores persistent data from the web UI console.
5. [AWS Step Functions](#), [AWS CloudFormation](#), [AWS Lambda](#) and [Amazon EventBridge](#) orchestrate the lifecycle management of data pipelines.
6. The data pipeline, consisting of [Application Load Balancer](#), [Amazon ECS](#), [Amazon Managed Streaming for Apache Kafka \(Amazon MSK\)](#), [Amazon Kinesis Data Streams](#), Amazon S3, [Amazon EMR Serverless](#), [Amazon Redshift](#), and [Quick](#) provides scalable clickstream ingestion through load-balanced processing, buffered storage, ETL, and warehouse analytics.

The key functionality of this guidance is to build a data pipeline to collect, process, and analyze their clickstream data. The data pipeline consists of four modules:

- Ingestion module
- Data processing module
- Data modeling module
- Reporting module

The following introduces the architecture diagram for each module.

Ingestion module



Ingestion module architecture

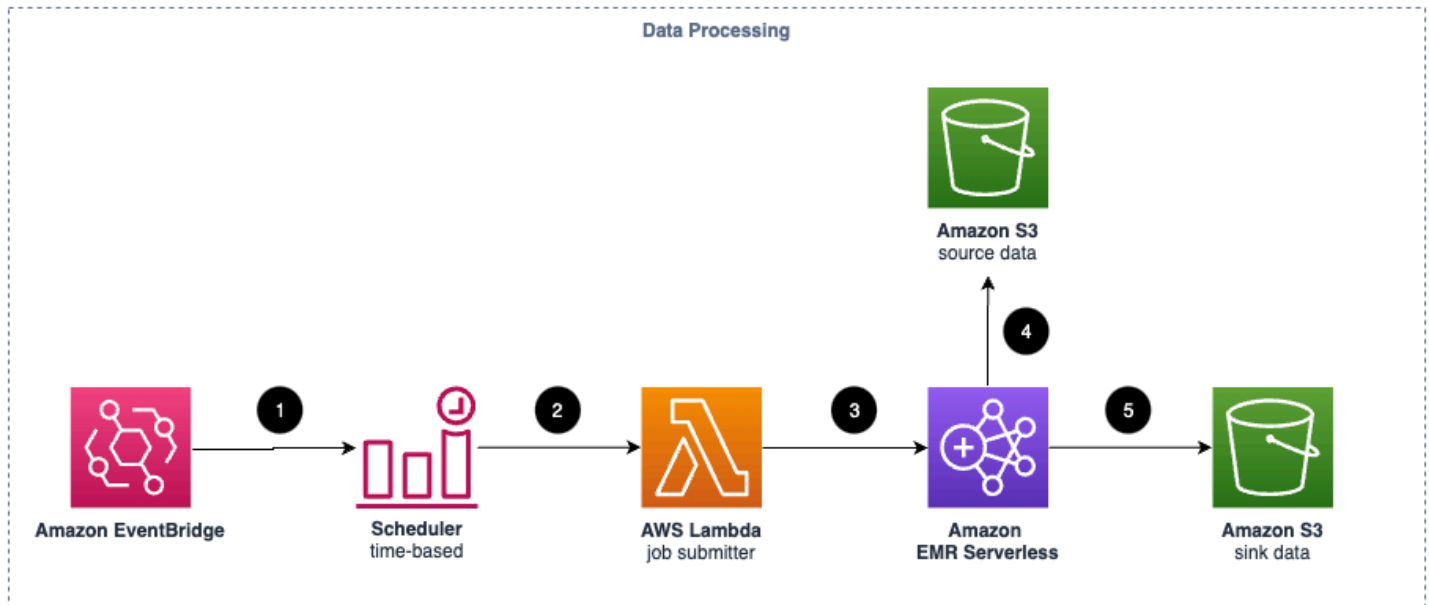
Suppose you create a data pipeline in the guidance. This guidance deploys the Amazon CloudFormation template in your AWS account and completes the following settings.

 **Note**

The ingestion module supports three types of data sinks. You can only have one type of data sink in a data pipeline.

1. (Optional) The ingestion module creates an AWS global accelerator endpoint to reduce the latency of sending events from your clients (web applications or mobile applications).
2. [Elastic Load Balancing \(ELB\)](#) is used for load balancing ingestion web servers.
3. (Optional) If you enable the authenticating feature, the ALB will communicate with the OIDC provider to authenticate the requests.
4. ALB forwards all authenticated and valid requests to the ingestion servers.
5. Amazon ECS cluster is hosting the ingestion fleet servers. Each server consists of a proxy and a worker service. The proxy is a facade of the HTTP protocol, and the worker will send the events to a data sink based on your choice.
6. If Amazon Kinesis Data Streams is used as a buffer, AWS Lambda consumes the clickstream data in Kinesis Data Streams and then sinks them to Amazon S3 in batches.
7. If Amazon MSK is used as a buffer, MSK Connector is provisioned with an S3 connector plugin that sinks the clickstream data to Amazon S3 in batches.
8. If Amazon S3 is selected as data sink, the ingestion server will buffer a batch of events and sink them to Amazon S3.

Data processing module

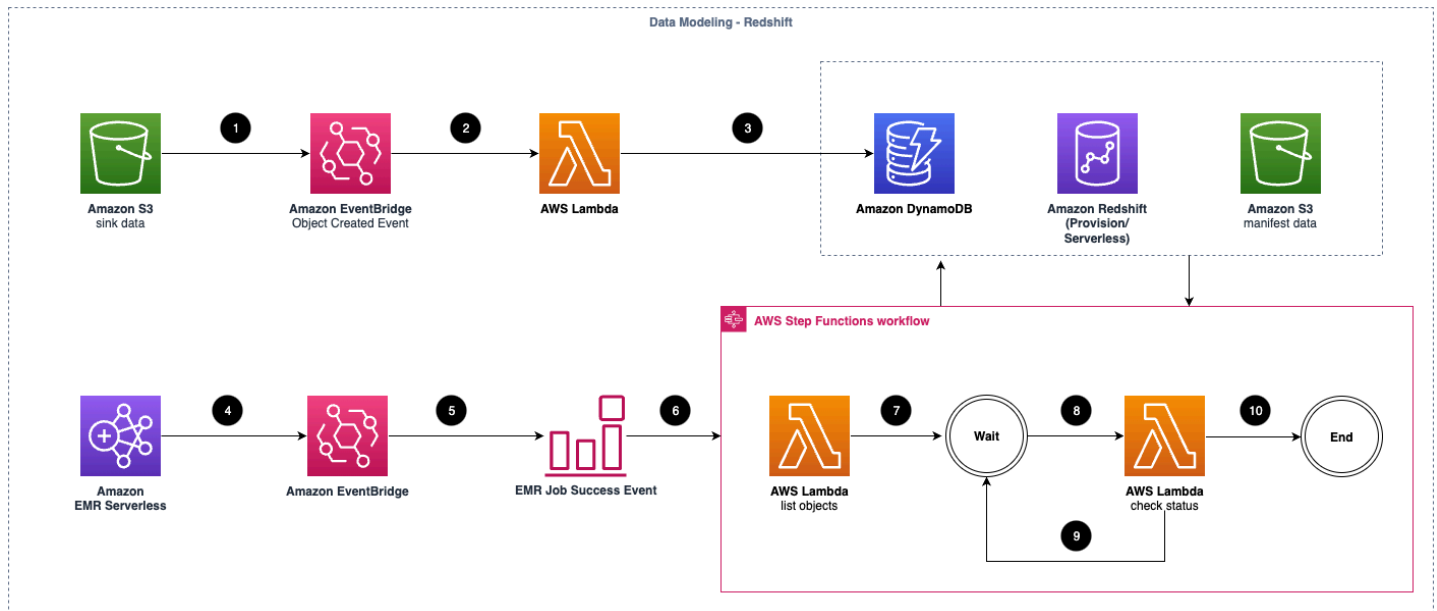


Data processing module architecture

Suppose you create a data pipeline in the guidance and enable data processing. This guidance deploys the Amazon CloudFormation template in your AWS Cloud account and completes the following settings.

1. Amazon EventBridge is used to trigger the data processing jobs periodically.
2. The configurable time-based scheduler invokes an AWS Lambda function.
3. The Lambda function kicks off an EMR Serverless application based on Spark to process a batch of clickstream events.
4. The EMR Serverless application uses the configurable transformer and enrichment plug-ins to process the clickstream data from the source S3 bucket.
5. After processing the clickstream events, the EMR Serverless application sinks the processed clickstream data to the sink S3 bucket.

Data modeling module

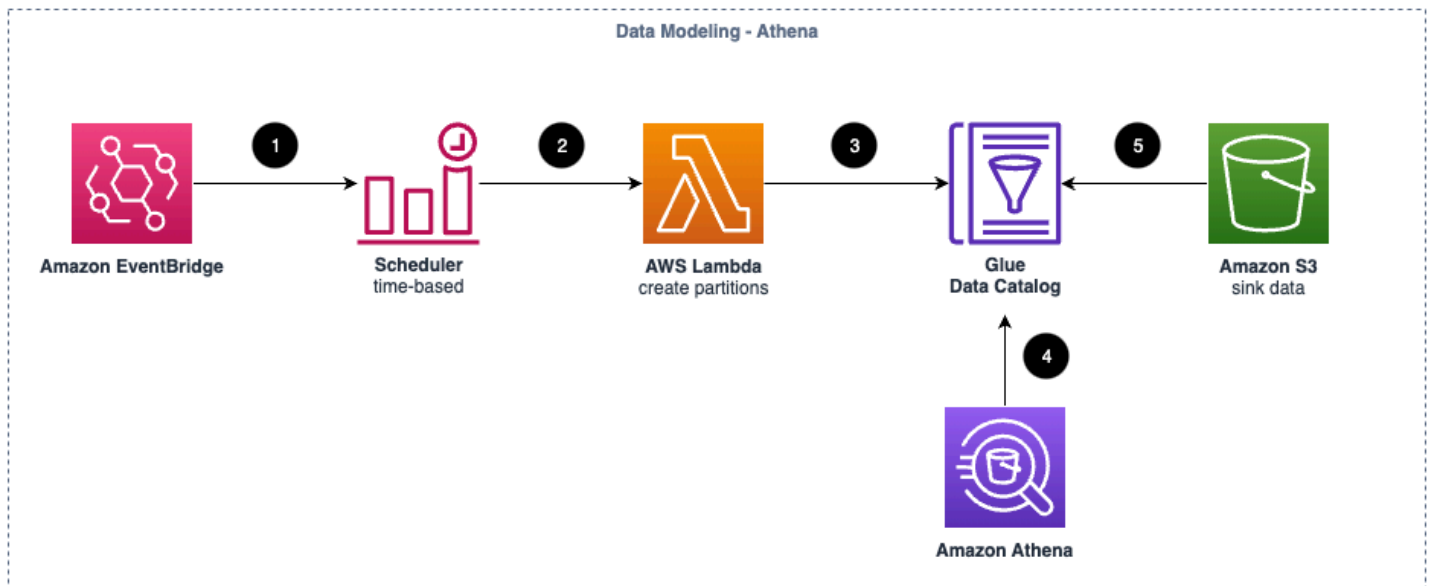


Data modeling in Redshift architecture

Suppose you create a data pipeline in the guidance and enable data modeling in Amazon Redshift. This guidance deploys the Amazon CloudFormation template in your AWS Cloud account and completes the following settings.

1. After the processed clickstream data is written in the Amazon S3 bucket, the Object Created Event is emitted.
2. An Amazon EventBridge rule is created for the event emitted in step 1, and an AWS Lambda function is invoked when the event happens.
3. The Lambda function persists the source event to be loaded in an Amazon DynamoDB table.
4. When data processing job is done, an event is emitted to Amazon EventBridge.
5. The pre-defined event rule of Amazon EventBridge processes the EMR job success event.
6. The rule invokes the AWS Step Functions workflow.
7. The workflow invokes the `list objects` Lambda function that queries the DynamoDB table to find out the data to be loaded, then creates a manifest file for a batch of event data to optimize the load performance.
8. After a few seconds, the `check status` Lambda function checks the status of the loading job.
9. If the load is still in progress, the `check status` Lambda function waits for a few more seconds.

10 After all objects are loaded, the workflow ends.

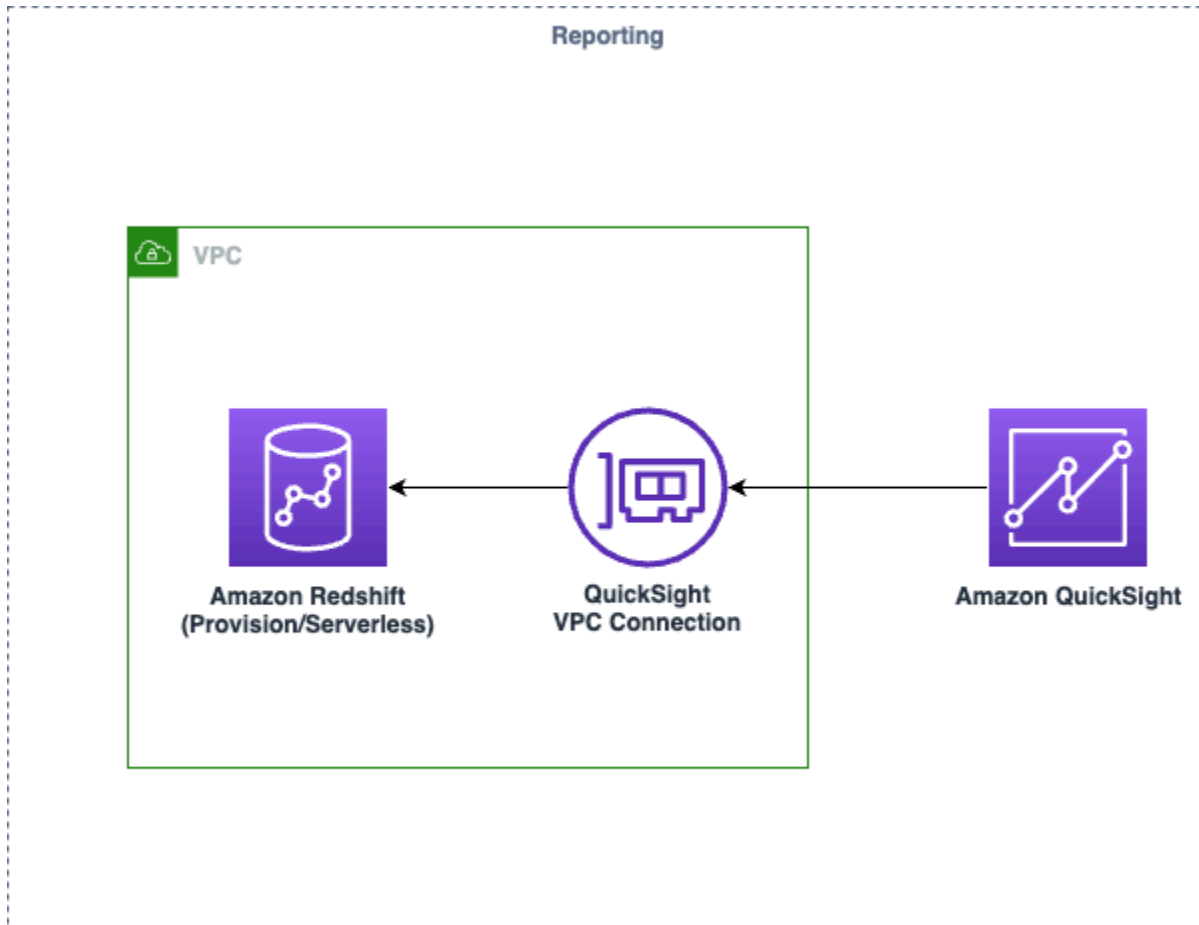


Data modeling in Athena architecture

Suppose you create a data pipeline in the guidance and enable data modeling in Amazon Athena. This guidance deploys the Amazon CloudFormation template in your AWS Cloud account and completes the following settings.

1. Amazon EventBridge invokes the data load into [Amazon Athena](#) periodically.
2. The configurable time-based scheduler invokes an AWS Lambda function.
3. The AWS Lambda function creates the partitions of the [AWS Glue](#) table for the processed clickstream data.
4. Amazon Athena is used for interactive querying of clickstream events.
5. The processed clickstream data is scanned via the Glue table.

Reporting module



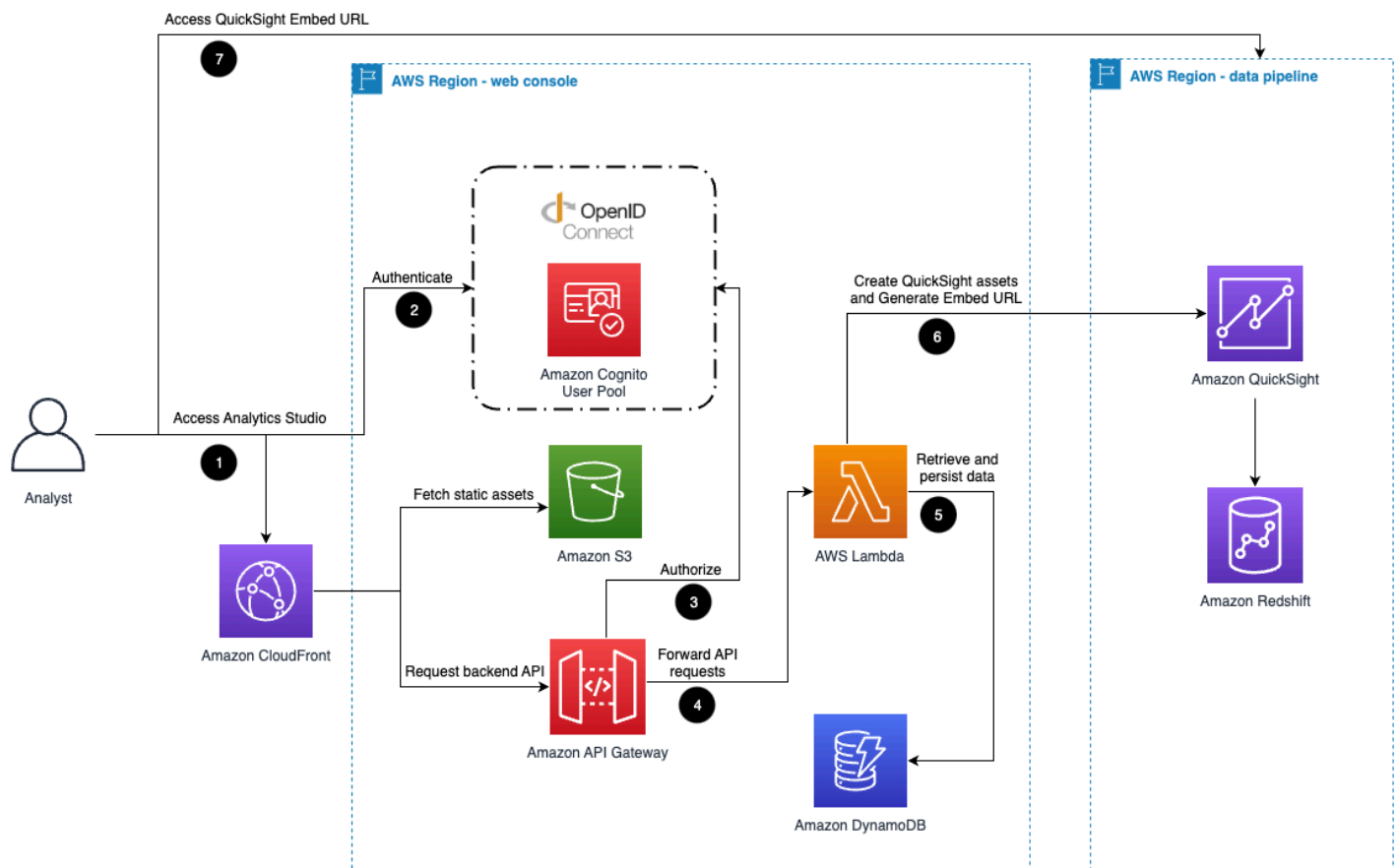
Reporting module architecture

Suppose you create a data pipeline in the guidance, enable data modeling in Amazon Redshift, and enable reporting in Quick. This guidance deploys the Amazon CloudFormation template in your AWS Cloud account and completes the following settings.

1. VPC connection in Quick is used for securely connecting your Redshift within VPC.
2. The data source, data sets, template, analysis, and dashboard are created in Quick for out-of-the-box analysis and visualization.

Analytics Studio

Analytics Studio is a unified web interface for business analysts or data analysts to view and create dashboards, query and explore clickstream data, and manage metadata.



Analytics Studio architecture

1. When analysts access Analytics Studio, requests are sent to [Amazon CloudFront](#), which distributes the web application.
2. When the analysts log in to Analytics Studio, the requests are redirected to the [Amazon Cognito](#) user pool or OpenID Connect (OIDC) for authentication.
3. [Amazon API Gateway](#) hosts the backend API requests and uses the custom Lambda authorizer to authorize the requests with the public key of OIDC.
4. API Gateway integrates with [AWS Lambda](#) to serve the API requests.
5. The AWS Lambda function uses [Amazon DynamoDB](#) to retrieve and persist the data.
6. When analysts create analyses, the Lambda function requests [Amazon QuickSight](#) to create assets and get the embed URL in the data pipeline Region.
7. The browser of analysts accesses the QuickSight embed URL to view the QuickSight dashboards and visuals.

AWS Well-Architected pillars

This guidance was designed with best practices from the [AWS Well-Architected Framework](#) which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework were applied when building this guidance.

Operational excellence

This section describes how the principles and best practices of the [operational excellence pillar](#) were applied when designing this guidance.

The Clickstream Analytics on AWS guidance pushes metrics, logs and traces to Amazon CloudWatch at various stages to provide observability into the infrastructure, Elastic load balancer, Amazon ECS cluster, Lambda functions, EMR serverless application, Step Function workflow and the rest of the guidance components. This guidance also creates the CloudWatch dashboard for each [data pipeline](#).

Security

This section describes how the principles and best practices of the [security pillar](#) were applied when designing this guidance.

- Clickstream Analytics on AWS web console users are authenticated and authorized with Amazon Cognito or OpenID Connect.
- All inter-service communications use AWS IAM roles.
- All roles used by the guidance follows least-privilege access. That is, it only contains minimum permissions required so the service can function properly.

Reliability

This section describes how the principles and best practices of the [reliability pillar](#) were applied when designing this guidance.

- Using AWS serverless services wherever possible (for example, EMR Serverless, Redshift Serverless, Lambda, Step Functions, Amazon S3, and Amazon SQS) to ensure high availability and recovery from service failure.

- Data ingested by [data pipeline](#) is stored in Amazon S3 and Amazon Redshift, so it persists in multiple Availability Zones (AZs) by default.

Performance efficiency

This section describes how the principles and best practices of the [performance efficiency pillar](#) were applied when designing this guidance.

- The ability to launch this guidance in any Region that supports AWS services in this guidance such as: Amazon S3, Amazon ECS, and Elastic load balancer.
- Using Analytics Serverless architectures removes the need for you to run and maintain physical servers for traditional compute activities.
- Automatically testing and deploying this guidance daily. Reviewing this guidance by guidance architects and subject matter experts for areas to experiment and improve.

Cost optimization

This section describes how the principles and best practices of the [cost optimization pillar](#) were applied when designing this guidance.

- The guidance uses Autoscaling Group so that the compute costs are only related to how much data is ingested and processed.
- The guidance uses serverless services such as Amazon S3, Amazon Kinesis Data Streams, Amazon EMR Serverless and Amazon Redshift Serverless so that customers only get charged for what they use.

Sustainability

This section describes how the principles and best practices of the [sustainability pillar](#) were applied when designing this guidance.

- The guidance's serverless design (using Amazon Kinesis Data Streams, Amazon EMR Serverless, Amazon Redshift Serverless and Quick) and the use of managed services (such as Amazon ECS, Amazon MSK) are aimed at reducing carbon footprint compared to the footprint of continually operating on-premises servers.

Architecture details

This section describes the components and AWS services that make up this guidance and the architecture details on how these components work together.

Guidance components

The Clickstream Analytics on AWS guidance has four components: a web console, Analytics Studio, SDKs, and the data pipeline.

Web console

This guidance provides a simple web console which allows you to create and manage projects with their data pipeline to ingest, process, analyze and visualize the Clickstream data.

Analytics Studio

Analytics Studio is a unified web interface for business analysts or data analysts to view and create dashboards, query and explore clickstream data, and manage metadata. It supports the following features:

- provide a pre-canned user lifecycle dashboard
- provide an explorative analytics model to query and analyze clickstream data
- support creating custom analysis and visualization in a drag-and-drop manner
- auto-generate metadata for clickstream data, and support metadata management

SDKs

This guidance provides native SDKs to help you easily collect and report in-app events from your applications to your Clickstream pipelines.

- [Android SDK](#)
- [Swift SDK](#)
- [Web SDK](#)
- [Flutter SDK](#)

Data pipeline

This guidance uses the web console to manage the project and its data pipeline. The data pipeline consists of four modules.

Ingestion module

The ingestion module serves as web server for ingesting the Clickstream data. It supports the following features:

- Specify the auto scaling group capability
- Specify warm pool size to scale out faster and save costs
- Support authenticate with OIDC
- Support SSL
- Support enabling AWS Global Accelerator for ELB
- Support different sink buffer, S3, KDS and MSK

Data processing module

The data processing module transforms and enriches the ingested data to guidance's data model by the Apache Spark application running in EMR serverless. It supports the following features:

- Specify the batch interval of data processing
- Specify the data refreshness age
- Provider out-of-the-box enrichment plug-ins
 - UA enrichment to parse OS, device, browser information from User Agent string of the HTTP request header
 - IP enrichment to mapping device location information (for example, city, country, region) based on the request source IP
- Support third-party transformer plug-ins
- Support third-party enrichment plug-ins

Data modeling module

The data modeling module loads the processed data into 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 QuickSight.

AWS services in this guidance

The following AWS services are included in this guidance:

AWS service	Description
Amazon Elastic Load Balancing	Core. To distribute network traffic to ingestion fleet.
Amazon ECS	Core. To run the ingestion module fleet.
Amazon EC2	Core. To provide the underlying computing resources for ingestion fleet.
Amazon ECR	Core. To host the container images used by ingestion fleet.
Amazon S3	Core. To store the ingested and processed Clickstream data. And it also stores the service logs and static web assets (frontend user interface).
AWS Global Accelerator	Supporting. To improve the availability, performance, and security of the ingestion service in AWS Regions.
AWS CloudWatch	Supporting. To monitor the metrics, logs and trace of data pipeline.

AWS service	Description
Amazon SNS	Supporting. To provide topic and email subscription notifications for the alarms of data pipeline.
Amazon Kinesis Data Streams	Supporting. To provide the ingestion buffer.
AWS Lambda	Supporting. To integrate with kinds of AWS services. For example, sink ingestion data to S3, manage the lifecycle of AWS resources.
Amazon Managed Streaming for Apache Kafka (MSK)	Supporting. To provide the ingestion buffer with Apache Kafka.
Amazon EMR Serverless	Supporting. To process the ingested data.
Amazon Glue	Supporting. To manage the data catalog of ingested data.
Amazon EventBridge	Supporting. To integrate with AWS services with events or schedule.
Amazon Redshift	Supporting. To analyze your Clickstream data in data warehouse.
Amazon Athena	Supporting. To analyze your Clickstream data in data lake.
AWS Step Functions	Supporting. To orchestrate the lifecycle management of project's pipeline. Also it manages the workflow to load data into data warehouse.
AWS Secrets Manager	Supporting. To store the credential for OIDC credentials and BI user in Redshift.
Quick	Supporting. Visual your analysis reporting of your Clickstream data.

AWS service	Description
Amazon CloudFront	Supporting. To made available the static web assets (frontend user interface) and proxy the backend in the same origin.
Amazon Cognito	Supporting. To authenticate users (in AWS Regions).
Amazon API Gateway	Supporting. To provide the backend APIs.
Amazon DynamoDB	Supporting. To store projects data.
AWS CloudFormation	Supporting. To provision the AWS resources for the modules of data pipeline.

Plan your deployment

This section describes the cost, security, and Region considerations for planning your deployment.

Cost

Important

The following cost estimations are examples and may vary depending on your environment.

You are responsible for the cost of AWS services used when running this guidance. Deploying this guidance will only create a guidance web console in your AWS account, which is completely serverless and typically can be covered within free tier.

The cost for this guidance is mostly incurred by the data pipeline. As of this revision, the main factors affecting the guidance cost include:

- **Ingestion module**, the cost depends on the size of the ingestion server and the type of the data sink you choose.
- **Data processing and modeling module** (optional), the cost depends on whether you choose to enabled this module and its relevant configurations
- **Enabled Dashboards** (optional), the cost depends on whether you choose to enabled this module and its relevant configurations
- **Additional features**

The following are cost estimations for data volumes of 10/100/1000/10000 RPS (request per second) with different data pipeline configurations. Cost estimation are provided by modules. To get a total cost for your use case, sum the cost by modules based on your actual configuration.

Important

As of this revision, the following cost is calculated with On-Demand prices in the us-east-1 Region and measured in USD.

Ingestion module

Ingestion module includes the following cost components:

- Application load balancer
- EC2 for ECS
- Data sink (Kinesis | Kafka | Direct to S3)
- S3 storage

Key assumptions include:

- Compressed request payload: 2KB (10 events per request)
- MSK configurations (m5.large * 2)
- KDS configuration (on-demand, provision)
- 10/100/1000 request per second (RPS)

RPS	ALB cost	EC2 cost	Buffer type	Buffer cost	S3 cost	Total (USD per month)
10 RPS (49GB per month)	\$28.80	\$122.00	Kinesis (On-Demand)	\$38.00	\$3.00	\$191.80
	\$28.80	\$122.00	Kinesis (Provisioned 2 shard)	\$22.00	\$3.00	\$175.80
	\$28.80	\$122.00	Amazon MSK (m5.large * 2, connector MCU * 1)	\$417.00	\$3.00	\$570.80

RPS	ALB cost	EC2 cost	Buffer type	Buffer cost	S3 cost	Total (USD per month)
	\$28.80	\$122.00	None	None	\$3.00	\$153.80
100 PRS (490GB per month)	\$53.80	\$122.00	Kinesis(O n-demand)	\$115.00	\$4.00	\$294.80
	\$53.80	\$122.00	Kinesis (Provisioned 2 shard)	\$26.00	\$4.00	\$205.80
	\$53.80	\$122.00	Amazon MSK (m5.large * 2, connector MCU * 1)	\$417.00	\$4.00	\$596.80
	\$53.80	\$122.00	None	None	\$4.00	\$179.80
1,000 RPS (4,900 GB per month)	\$262.80	\$122.00	Kinesis(O n-demand)	\$1,051.00	\$14.00	\$1,449.80
	\$262.80	\$122.00	Kinesis (Provisioned 10 shard)	\$180.00	\$14.00	\$578.80
	\$262.80	\$122.00	Amazon MSK (m5.large * 2, connector MCU * 2~3)	\$590.00	\$14.00	\$988.80

RPS	ALB cost	EC2 cost	Buffer type	Buffer cost	S3 cost	Total (USD per month)
	\$262.80	\$122.00	None	None	\$14.00	\$398.80

Data transfer

There are associated costs for data transfer from EC2 to the downstream data sink. Below is an example of data transfer costs based on 1000 RPS and a 1KB request payload.

- EC2 Network In: This does not incur any costs.
- EC2 Network Out: There are three data sink types (Amazon S3, Amazon MSK, and Amazon Kinesis Data Streams).

Data sink type	Way to access data sink	Dimensions	Total (USD)
Amazon S3	Amazon S3 gateway endpoints	The Amazon S3 gateway endpoints do not incur any costs.	\$0
Amazon MSK	N/A	Data processed cost (\$0.010 per GB in/out/between EC2 Availability Zones)	\$210
Amazon Kinesis Data Streams	NAT	NAT fixed cost: \$64 (2 Availability Zones and a NAT per Availability Zone, \$0.045 per NAT Gateway Hour). Data processed cost: \$1201 (\$0.045 per	\$1266

Data sink type	Way to access data sink	Dimensions	Total (USD)
		GB Data Processed by NAT gateways).	
Amazon Kinesis Data Streams	VPC endpoint	VPC endpoint fixed cost: \$14.62 (Availability Zones \$0.01 per AZ Hour). Data processed cost: \$267 (\$0.01 per GB Data Processed by Interface endpoints).	\$281.62

You are recommended to use a VPC endpoint for the Amazon Kinesis Data Streams data sink. For more information, refer to the [VPC endpoint](#) documentation.

Data processing & modeling modules

Data processing & modeling module include the following cost components if you enable:

- EMR Serverless
- Redshift

Key assumptions include:

- 10/100/1000 RPS
- Data processing interval: hourly/6-hourly/daily
- EMR running three built-in plugins to process data

RPS	EMR schedule interval	EMR cost	Redshift type	Redshift load cost	Redshift storage cost	S3 cost	Total (USD/ Month)
10 RPS	Hourly	\$65.50 (\$1.35/GB)	Serverless (8 based RPU)	\$172.00	\$3.40	\$0.36	\$242.26
	6-hourly	\$22.20 (\$0.45/GB)	Serverless (8 based RPU)	\$70.00	\$3.40	\$0.36	\$95.96
	Daily	\$39.00 (\$0.80/GB)	Serverless (8 based RPU)	\$31.00	\$3.40	\$0.36	\$73.76
100 RPS	Hourly	\$353.00 (\$0.72/GB)	Serverless (8 based RPU)	\$385.00	\$34.00	\$3.60	\$775.60
	6-hourly	\$179.00 (\$0.37/GB)	Serverless (8 based RPU)	\$282.00	\$34.00	\$3.60	\$498.60
	Daily	\$247.00 (\$0.50/GB)	Serverless (8 based RPU)	\$160.00	\$34.00	\$3.60	\$444.60
1,000 RPS	Hourly	\$1,260.00 (\$0.26/GB)	Serverless (16 based RPU)	\$2,325.00	\$340	\$36	\$3,961.00

Note

The term **Redshift storage cost** refers to the cost of Redshift storage incurred for one month based on the corresponding RPS. If the data is stored for more than one month, please refer to the [Redshift pricing](#).

Reporting module

Reporting module includes the following cost components if you choose to enable:

- QuickSight

Key assumptions include

- QuickSight Enterprise subscription
- Exclude Q cost
- Analytics Studio
- 10GB SPICE capacity

Daily data volume/ RPS	Cost for authors	Cost for SPICE	Total cost (USD/ Month)
All size	\$24	0	\$24

Note

The QuickSight cost applies to all your data pipelines, including the visualization managed outside the guidance.

Logging and monitoring

The guidance utilizes CloudWatch Logs, CloudWatch Metrics and CloudWatch Dashboard to implement logging, monitoring and visualizing features. The total cost is around \$14 per month and may vary based on the volume of logs and the number of metrics being monitored.

Additional features

You will be charged with additional cost only if you choose to enable the following features.

Secrets Manager

- If you enable reporting, the guidance creates a secret in Secrets Manager to store the Redshift credentials used by QuickSight visualization. **Cost:** 0.40 USD/month.
- If you enable the authentication feature of the ingestion module, you need to create a secret in Secrets Manager to store the information for OIDC. **Cost:** 0.40 USD/month.

AWS Global Accelerator

It incurs a fixed hourly charge and a per-day volume data transfer cost.

Key assumptions: Ingestion deployment in us-east-1

RPS	Fixed hourly cost	Data transfer cost	Total cost (USD/ Month)
10 RPS	\$18	\$0.60	\$18.60
100 RPS	\$18	\$6	\$24
1000 RPS	\$18	\$60	\$78

Application Load Balancer access logs

The charged cost includes the storage cost for Amazon S3, but not for the bandwidth used by Elastic Load Balancing to send log files to Amazon S3. For more information, see [Amazon S3 pricing](#).

RPS	Log size (GB)	S3 cost (USD/Month)
10 RPS	16.50	\$0.38
100 RPS	165	\$3.80
1000 RPS	1650	\$38

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This [shared responsibility model](#) reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, see [AWS Cloud Security](#).

IAM Roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. This guidance creates IAM roles that grant the guidance's AWS Lambda functions, Amazon API Gateway and Amazon Cognito or OpenID connect access to create regional resources.

Amazon VPC

This guidance optionally deploys a web console within your VPC. You can isolate access to the web console via Bastion hosts, VPNs, or Direct Connect. You can create [VPC endpoints](#) to let traffic between your Amazon VPC and AWS services not leave the Amazon network to satisfy the compliance requirements.

Security groups

The security groups created in this guidance are designed to control and isolate network traffic between the guidance components. We recommend that you review the security groups and further restrict access as needed once the deployment is up and running.

Amazon CloudFront

This guidance optionally deploys a web console hosted in an Amazon S3 bucket and Amazon API Gateway. To help reduce latency and improve security, this guidance includes an Amazon CloudFront distribution with an Origin Access Control (OAC), which is a CloudFront user that provides public access to the guidance's website bucket contents. For more information, refer to [Restricting access to an Amazon S3 origin](#) in the Amazon CloudFront Developer Guide.

Supported AWS Regions

This guidance uses services which may not be currently available in all AWS Regions. Launch this guidance in an AWS Region where required services are available. For the most current availability by Region, refer to the [AWS Regional Services List](#).

Clickstream Analytics on AWS provides two types of authentication for its web console, [Amazon Cognito User Pool](#) and [OpenID Connect \(OIDC\) Provider](#). You must choose to launch the guidance with OpenID Connect in case one of the following scenarios:

- Amazon Cognito User Pool is not available in your AWS Region.
- You already have an OpenID Connect Provider and want to authenticate against it.

Supported Regions for web console deployment

Region Name	Launch with Amazon Cognito user pool	Launch with OpenID Connect
US East (N. Virginia)	Yes	Yes
US East (Ohio)	Yes	Yes
US West (N. California)	Yes	Yes
US West (Oregon)	Yes	Yes
Africa (Cape Town)	No	Yes
Asia Pacific (Hong Kong)	No	Yes
Asia Pacific (Jakarta)	No	Yes

Region Name	Launch with Amazon Cognito user pool	Launch with OpenID Connect
Asia Pacific (Mumbai)	Yes	Yes
Asia Pacific (Osaka)	No	Yes
Asia Pacific (Seoul)	Yes	Yes
Asia Pacific (Singapore)	Yes	Yes
Asia Pacific (Sydney)	Yes	Yes
Asia Pacific (Tokyo)	Yes	Yes
Canada (Central)	Yes	Yes
Europe (Frankfurt)	Yes	Yes
Europe (Ireland)	Yes	Yes
Europe (London)	Yes	Yes
Europe (Milan)	No	Yes
Europe (Paris)	Yes	Yes
Europe (Stockholm)	Yes	Yes
Middle East (Bahrain)	No	Yes
South America (Sao Paulo)	Yes	Yes
China (Beijing) Region Operated by Sinnet	No	Yes
China (Ningxia) Region Operated by NWCD	No	Yes

This guidance provides [modular components](#) for supporting different data pipeline architecture. The data processing, and reporting modules are optional, that is, you can create a data pipeline without data processing and reporting modules if needed.

Pipeline modules availability

Region Name	Data ingestion with MSK as buffer	Data ingestion with KDS as buffer	Data ingestion with S3 as buffer	Data processing	Data modeling with Redshift Serverless	Data modeling with Provisioned Redshift	Reporting with QuickSight
US East (N. Virginia)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
US East (Ohio)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
US West (N. California)	Yes	Yes	Yes	Yes	Yes	Yes	No
US West (Oregon)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Africa (Cape Town)	Yes	Yes	Yes	No	No	No	No
Asia Pacific (Hong Kong)	Yes	Yes	Yes	Yes	No	Yes	No

Region Name	Data ingestion with MSK as buffer	Data ingestion with KDS as buffer	Data ingestion with S3 as buffer	Data processing	Data modeling with Redshift Serverless	Data modeling with Provisioned Redshift	Reporting with QuickSight
Asia Pacific (Mumbai)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Asia Pacific (Osaka)	Yes	Yes	Yes	No	No	No	No
Asia Pacific (Seoul)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Asia Pacific (Singapore)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Asia Pacific (Sydney)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Asia Pacific (Tokyo)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Canada (Central)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Europe (Frankfurt)	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Region Name	Data ingestion with MSK as buffer	Data ingestion with KDS as buffer	Data ingestion with S3 as buffer	Data processing	Data modeling with Redshift Serverless	Data modeling with Provisioned Redshift	Reporting with QuickSight
Europe (Ireland)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Europe (London)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Europe (Milan)	Yes	Yes	Yes	No	No	No	No
Europe (Paris)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Europe (Stockholm)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Middle East (Bahrain)	Yes	Yes	Yes	Yes	No	Yes	No
South America (Sao Paulo)	Yes	Yes	Yes	Yes	No	Yes	Yes
China (Beijing) Region Operated by Sinnet*	Yes	Yes	Yes	Yes	No	Yes	Yes

Region Name	Data ingestion with MSK as buffer	Data ingestion with KDS as buffer	Data ingestion with S3 as buffer	Data processing	Data modeling with Redshift Serverless	Data modeling with Provisioned Redshift	Reporting with QuickSight
China (Ningxia) Region Operated by NWCD*	Yes	Yes	Yes	Yes	No	Yes	No

Note

AWS China Regions don't support using AWS Global Accelerator to accelerate the ingestion endpoint.

Deployment

Before you launch the guidance, review the architecture, supported Regions, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the guidance into your account.

Prerequisites

Review all the [considerations](#) and make sure you have the following in the target Region you want to deploy the solution:

- At least two vacant S3 buckets.

Deployment in AWS Regions

- Clone the repository:

```
git clone https://github.com/aws-solutions-library-samples/guidance-for-clickstream-analytics-on-aws.git
cd guidance-for-clickstream-analytics-on-aws
```

- Install pnpm and dependencies:

```
npm install -g pnpm@9.15.3
pnpm install && pnpm projen && pnpm nx build @aws/clickstream-base-lib
```

- Bootstrap CDK (if not done before):

```
npx cdk bootstrap
```

- Deploy the stack:

```
cd deployment
sh solution-deploy.sh --region <aws_region> --profile <profile_name> --
email <user_email> --template-deploy
```

- Note the CloudFront URL from the outputs to access the web console.

Getting started

After [deploying the guidance](#), refer to this section to quickly learn how to leverage the Clickstream Analytics on AWS guidance to collect and analyze clickstream data from your applications. This chapter shows you how to create a serverless data pipeline to collect data from an application, and use Analytics Studio to view the out-of-the-box user lifecycle dashboard and query the clickstream data with exploration analytics.

- [Step 1: Create a project.](#)

Create a project.

- [Step 2: Configure a data pipeline.](#)

Configure a data pipeline with serverless infrastructure.

- [Step 3: Integrate SDK.](#)

Integrate SDK into your application to automatically collect data and send data to the pipeline.

- [Step 4: Analyze data.](#)

View the out-of-the-box dashboards based on the data automatically collected from your applications.

Step 1: Create a project

To get started with the Clickstream Analytics on AWS guidance, you need to firstly create a project in the guidance console. A project is like a container for all the AWS resources provisioned for collecting and analyzing the clickstream data from your apps.

Prerequisites

Make sure you have deployed the Clickstream Analytics on AWS guidance. If you haven't, please refer to the [deployment guide](#).

Steps

Following below steps to create a project.

1. Log into **Clickstream Analytics on AWS Management Console**.

2. On the **Home** page, choose **Create Project**.
3. In the window that pops up, enter a project name, for example, `quickstart`.
4. (Optional) Customize the project ID that was automatically created by guidance. To do so, click the `edit` icon and update the project ID as per your need.
5. Enter a description for your project, for example, `This is a demo project`.
6. Choose **Next**.
7. Enter an email address to receive notification regarding this project, for example, `email@example.com`, and choose **Next**.
8. Specify an environment type for this project. In this example, select `Dev`.
9. Choose **Create**. Wait until the project creation completed, and you will be directed to the **Projects** page.

We have completed all the steps of creating a project.

Step 2: Configure data pipeline


After you create a project, you need to configure the data pipeline for it. A data pipeline is a set of connected modules that collect and process the clickstream data sent from your applications. A data pipeline contains four modules of ingestion, processing, modeling and reporting. For more information, see [pipeline management](#).

Here we provide an example with steps to create a data pipeline with end-to-end serverless infrastructure.

Steps

1. Sign in to **Clickstream Analytics on AWS Management Console**.
2. In the left navigation pane, choose **Projects**, then select the project you just created in **Step 1**, choose **View Details** in the top right corner to navigate to the project homepage.
3. Choose **Configure pipeline**, and it will bring you to the wizard of creating data pipeline for your project.
4. On the **Basic information** page, fill in the form as follows:
 - AWS Region: **us-east-1**
 - VPC: select a VPC that meets the following requirements
 - At least two public subnets across two different AZs (Availability Zone)

- At least two private subnets across two different AZs
- One NAT Gateway or Instance
- Data collection SDK: **Clickstream SDK**
- Data location: select an S3 bucket. (You can create one bucket, and select it after choosing **Refresh**.)

 **Note**

- Please comply with [Security best practices for Amazon S3](#) to create and configure Amazon S3 buckets. For example, Enable Amazon S3 server access logging, Enable S3 Versioning and so on.
- If you don't have a VPC meet the criteria, you can create a VPC with VPC wizard quickly. For more information, see [Create a VPC](#).

5. Choose **Next**.

6. On the **Configure ingestion** page, fill in the information as follows:

- Fill in the **Ingestion endpoint settings** form.
 - Public Subnets: Select two public subnets in two different AZs
 - Private Subnets: Select two private subnets in the same AZs as public subnets
 - Ingestion capacity: Keep the default values
 - Enable HTTPS: Uncheck and then **Acknowledge** the security warning
 - Additional settings: Keep the default values
- Fill in the **Data sink settings** form.
 - Sink type: **Amazon Kinesis Data Stream(KDS)**
 - Provision mode: **On-demand**
 - In **Additional Settings**, change **Sink Maximum Interval** to 60 and **Batch Size** to 1000
- Choose **Next** to move to step 3.

 **Important**

Using HTTP is not a recommended configuration for production workload. This example configuration is to help you get started quickly.

7. On the **Configure data processing** information, fill in the information as follows:

- In the **Enable data processing** form, turn on **Enable data processing**
- In the **Execution parameters** form,
 - Data processing interval:
 - Select **Fixed Rate**
 - Enter **10**
 - Select **Minutes**
 - Event freshness: **35 Days**

⚠ Important

This example sets Data processing interval to be 10 minutes so that you can view the data faster. You can change the interval to be less frequent later to save cost. Refer to [Pipeline Management](#) to make changes to data pipeline.

- In the **Enrichment plugins** form, make sure the two plugins of **IP lookup** and **UA parser** are selected.
 - In the form of **Analytics engine**, fill in the form as follow:
 - Select the box for **Redshift**
 - Select the **Redshift Serverless**
 - Keep **Base RPU** as **8**
 - VPC: select the default VPC or the same one you selected previously in the last step
 - Security group: select the default security group
 - Subnet: select **three** subnets across three different AZs
 - Keep **Athena** selection as default
 - Choose **Next**.
8. On the **Reporting** page, fill in the form as follows:
- If your AWS account has not subscribed to QuickSight, please follow this [guide](#) to subscribe.
 - Toggle on the option **Enable Analytics Studio**.
 - Choose **Next**.
9. On the **Review and launch** page, review your pipeline configuration details. If everything is configured properly, choose **Create**.

We have completed all the steps of configuring a pipeline for your project. This pipeline will take about 15 minutes to create, and please wait for the pipeline status change to be **Active** in pipeline detail page.

Step 3: Integrate SDK

Once pipeline's status becomes **Active**, it is ready to receive clickstream data. Now you need to register an application to the pipeline, then you can integrate SDK into your application to enable it to send data to the pipeline.

Steps

1. Log into **Clickstream Analytics on AWS Management Console**.
2. In the left navigation pane, choose **Projects**, then select the project (quickstart) you just created in previous steps, click its title, and it will bring you to the project page.
3. Choose **+ Add application** to start adding application to the pipeline.
4. Fill in the form as follows:
 - App name: **test-app**
 - App ID: The system will generate one ID based on the name, and you can customize it if needed.
 - Description: **A test app for Clickstream Analytics on AWS guidance**
 - Android package name: leave it blank
 - App Bundle ID: leave it blank
5. Choose **Register App & Generate SDK Instruction**, and wait for the registration to be completed.
6. Select the tab **Android**, and you will see the detailed instruction of adding SDK into your application. You can follow the steps to add SDK.
7. Choose **Download the config json file** to download the config file, and keep this file open, which will be used later.

It will take about 3 ~ 5 minutes to update the pipeline with the application you just add. When you see the pipeline status become **Active** again, it is ready to receive data from your application.

We have completed all the steps of adding an application to a project.

Generate sample data

You might not have immediate access to integrate SDK with your app. In this case, we provide a Python script to generate sample data to the pipeline you just configured, so that you can view and experience the analytics dashboards.

Important

Python 3.8+ is required.

1. Clone the repository to your local environment.

```
git clone https://github.com/aws-solutions-library-samples/guidance-for-clickstream-analytics-on-aws.git
```

2. After you cloned the repository, change directory into the `examples/standalone-data-generator` project folder.
3. Install the dependencies of the project.

```
pip3 install requests
```

4. Put `amplifyconfiguration.json` into the root of `examples/standalone-data-generator` which you downloaded in **register an app** step. See the `examples/standalone-data-generator/README.md` for more information.
5. Open an terminal at this project folder location. For example, if you are using Visual Studio Code IDE, at the top of **Visual Studio Code**, click **Terminal** -> **New Terminal** to open a terminal.
6. Copy the following command and paste it to the terminal:

```
python3 create_event.py
```

Let's enter the `Enter` key in terminal to execute the program. If you see the following output, this means that the program execution is completed.

```
job finished, upload 4360476 events, cost: 95100ms
```

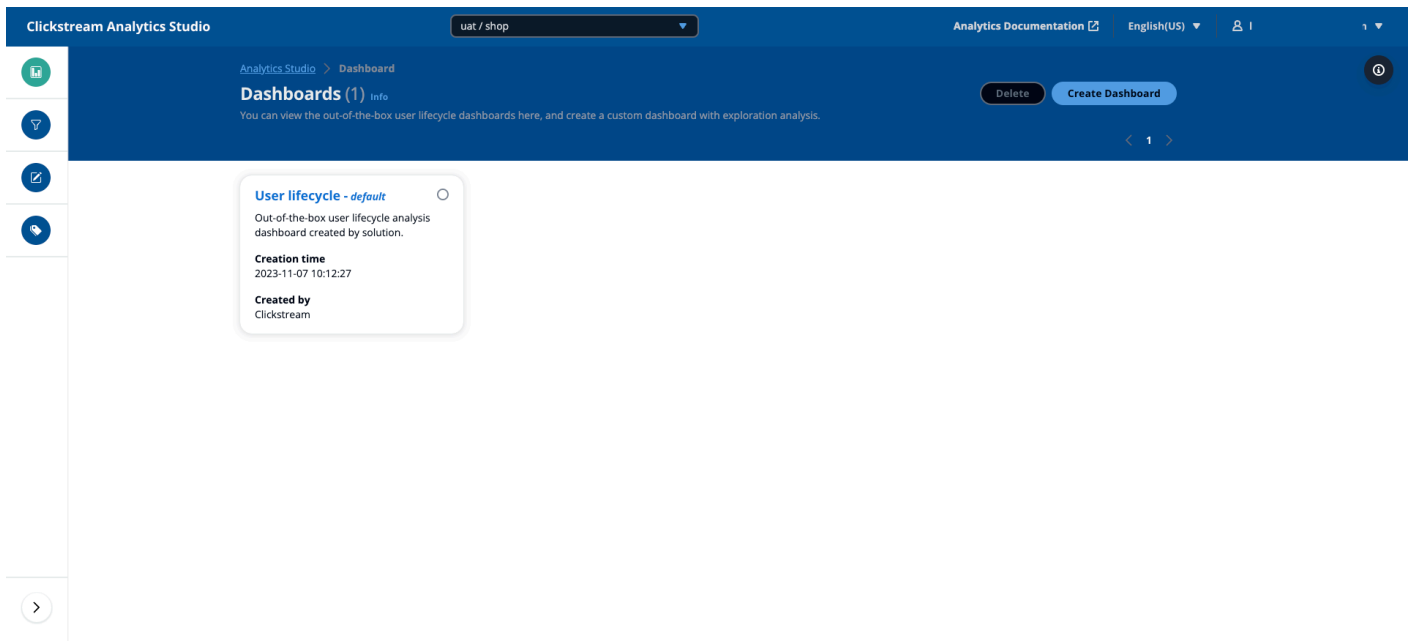
This process will take about 10 minutes with default configuration. After job is finished, you can move to next step.

Step 4: Analyze data

After your application sends data (or the sample data are sent) to the pipeline, you can go into the Analytics Studio to view dashboard and query data.

Steps

1. Log into **Clickstream Analytics on AWS Management Console**.
2. In the left navigation pane, choose **Analytics Studio**, a new tab opens in your browser.
3. In the Analytics Studio page, select the project and app you just created in the drop-down list at the top of the web page.
4. By default, you will be navigated to the **Dashboards** page. If not, choose **Dashboards** from the left navigation pane.



5. Choose **User lifecycle dashboard - default**. You can see the dashboard created by the guidance.
6. Choose **Exploration** in the left navigation pane. You can query the clickstream data by using the exploratory analytics models.

Congratulations! You have completed the getting started tutorial. You can explore the Analytic Studio or continue to learn more about this guidance later.

Pipeline management

Data pipeline is the core functionality of this guidance. In the Clickstream Analytics on AWS guidance, we define a data pipeline as a sequence of integrated AWS services that ingest, process, and model the clickstream data into a destination data warehouse for analytics and visualization. It is also designed to efficiently and reliably collect data from your websites and apps to a S3-based data lake, where it can be further processed, analyzed, and utilized for additional use cases (such as real-time monitoring, and recommendation).

To get a basic understanding of the pipeline, you can refer to [terms and concepts](#) for more information.

Prerequisites

You can configure the pipeline in all AWS Regions. For opt-in regions, you need to [enable them](#) first.

Before you start to configure the pipeline in a specific region, make sure you have the following in the target region:

- At least one Amazon VPC.
- At least two public subnets across two AZs in the VPC.
- At least two private (with NAT gateways or instances) subnets across two AZs, or at least two isolated subnets across two AZs in the VPC. If you want to deploy the guidance resources in the isolated subnets, you have to create [VPC endpoints](#) for below AWS services,
 - s3, logs, ecr.api, ecr.dkr, ecs, ecs-agent, ecs-telemetry.
 - kinesis-streams if you use KDS as sink buffer in ingestion module.
 - emr-serverless, glue if you enable data processing module.
 - redshift-data, sts, dynamodb, states and lambda if you enable Redshift as analytics engine in data modeling module.
- An Amazon S3 bucket located in the same region.
- If you need to enable Redshift Serverless as analytics engine in data modeling module, you need have subnets across at least three AZs.
- QuickSight Enterprise edition subscription is required if the reporting is enabled.

Ingestion

Ingestion module contains a web service that provides an endpoint to collect data through HTTP/HTTPS requests, which mainly is composed of Amazon Application Load Balancer and Amazon Elastic Container Service. It also supports sinking data into a stream service or S3 directly.

You can create an ingestion module with the following settings:

- [Ingestion endpoint settings](#): Create a web service as an ingestion endpoint to collect data sent from your SDKs.
- Data sink settings: Configure how the guidance sinks the data for downstream consumption. Currently, the guidance supports three types of data sink:
 - [Apache Kafka](#)
 - [Amazon S3](#)
 - [Amazon Kinesis Data Stream \(KDS\)](#)

Throttle

Currently, there is no built-in throttling feature available with this guidance. If needed, you can configure AWS WAF to implement throttling feature. Please refer to [WAF](#) documentation.

Ingestion endpoint

The guidance creates a web service as an ingestion endpoint to collect data sent from your SDKs. You can set below configurations for ingestion endpoint.

- **Network type**: Specify the scheme of the application load balancer. Choose **General** for internet-facing or **Private** for internal.
- **Public subnets (only applicable to General network type)** : Select at least two existing VPC public subnets, and the Amazon Application Load Balancers (ALBs) will be deployed in these subnets.
- **Private subnets**: Select at least two existing VPC private subnets, and the EC2 instances running in ECS will be deployed in these subnets.

Note

The availability zones where the public subnets are located must be consistent with those of the private subnets.

- **Ingestion capacity:** This configuration sets the capacity of the ingestion server, and the ingestion server will automatically scale up or down based on the utilization of the processing CPU.
- **Ingestion Capacity Unit (ICU):** A single Ingestion Compute Unit (ICU) represents billable compute and memory units, approximately 8 gigabytes (GB) of memory and 2 vCPUs. 1 ICU generally can support 4000~6000 requests per second.
- **Minimum capacity:** The minimum capacity to which the ingestion server will scale down.
- **Maximum capacity:** The maximum capacity to which the ingestion server will scale up.
- **Warm pool:** Warm pool gives you the ability to decrease latency for your applications that have exceptionally long boot time. For more information, please refer to [Warm pools for Amazon EC2 Auto Scaling](#).
- **Enable HTTPS:** Users can choose HTTPS/HTTP protocol for the Ingestion endpoint.

Warning

If you switch between enabling HTTPS and disabling HTTPS, there may be interruptions in the ingestion service.

- **Enable HTTPS:** If users choose to enable HTTPS, the ingestion server will provide HTTPS endpoint.
- **Domain name:** enter a domain name.

Note

Once the ingestion server is created, use the custom endpoint to create an alias or CNAME mapping in your Domain Name System (DNS) for the custom endpoint.

- **SSL Certificate:** User need to select an ACM certificate corresponding to the domain name that you input. If there is no ACM certificate, please refer to [create public certificate](#) to create it.

- **Disable HTTPS:** If users choose to disable HTTPS, the ingestion server will provide HTTP endpoint.

⚠ Important

Using HTTP protocol is not secure, because data will be sent without any encryption, and there are high risks of data being leaked or tampered during transmission. Please acknowledge the risk to proceed.

- **Cross-Origin Resource Sharing (CORS):** You can enable CORS to limit requests to data ingestion API from a specific domain. Note that, you need to input a complete internet address, for example, `https://www.example.com`, `http://localhost:8080`. Use comma to separate domain if you have multiple domain for this setting.

⚠ Warning

CORS is a mandatory setting if you are collecting data from a website. If you do not set value for this parameter, the ingestion server to reject all the requests from Web platform.

- **Additional Settings**
 - **Request path:** User can input the path of ingestion endpoint to collect data, the default path is `"/collect"`.
 - **AWS Global Accelerator:** User can choose to create an accelerator to get static IP addresses that act as a global fixed entry point to your ingestion server, which will improve the availability and performance of your ingestion server. Note that additional charges apply.
 - **Authentication:** User can use OIDC provider to authenticate the request sent to your ingestion server. If you plan to enable it, please create an OIDC client in the OIDC provider then create a secret in AWS Secret Manager with information:
 - issuer
 - token endpoint
 - User endpoint
 - Authorization endpoint
 - App client ID
 - App Client Secret

The format is like:

```
{
  "issuer": "xxx",
  "userEndpoint": "xxx",
  "authorizationEndpoint": "xxx",
  "tokenEndpoint": "xxx",
  "appId": "xxx",
  "appSecret": "xxx"
}
```

In the OIDC provider, you need to add `https://<ingestion server endpoint>/oauth2/idpresponse` to "Allowed callback URLs".

If you need to obtain the authentication token directly without inputting credential (username/password) manually, you can refer to [alb headless authentication client code](#) to set up your client to obtain the authentication token automatically.

Warning

If you switch between enabling Authentication and disabling Authentication, there may be interruptions in the ingestion service.

- **Access logs:** ALB supports delivering detailed logs of all requests it receives. If you enable this option, the guidance will automatically enable access logs for you and store the logs into the S3 bucket you selected in previous step.

Important

The bucket must have a bucket policy that grants Elastic Load Balancing permission to write the access logs to the bucket. For details, refer to [Step 2: Attach a policy to your S3 bucket](#).

Below is an example policy for the bucket in Regions available before August 2022.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/
AWSLogs/123456789012/*"
    }
  ]
}

```

You need to replace `elb-account-id` with the ID of the AWS account for Elastic Load Balancing in your Region:

- US East (N. Virginia) – 127311923021
- US East (Ohio) – 033677994240
- US West (N. California) – 027434742980
- US West (Oregon) – 797873946194
- Africa (Cape Town) – 098369216593
- Asia Pacific (Hong Kong) – 754344448648
- Asia Pacific (Jakarta) – 589379963580
- Asia Pacific (Mumbai) – 718504428378
- Asia Pacific (Osaka) – 383597477331
- Asia Pacific (Seoul) – 600734575887
- Asia Pacific (Singapore) – 114774131450
- Asia Pacific (Sydney) – 783225319266
- Asia Pacific (Tokyo) – 582318560864
- Canada (Central) – 985666609251
- Europe (Frankfurt) – 054676820928
- Europe (Ireland) – 156460612806
- Europe (London) – 652711504416
- Europe (Milan) – 635631232127
- Europe (Paris) – 009996457667

- Middle East (Bahrain) – 076674570225
- South America (São Paulo) – 507241528517
- China (Beijing) – 638102146993
- China (Ningxia) – 037604701340

Data sink – Kafka

This data sink will stream the clickstream data collected by the ingestion endpoint into a topic in a Kafka cluster. Currently, guidance support Amazon Managed Streaming for Apache Kafka (Amazon MSK) or a self-hosted Kafka cluster.

Amazon MSK

- **Select an existing Amazon MSK cluster.** Select an MSK cluster from the drop-down list, and the MSK cluster needs to meet the following requirements:
 - MSK cluster and this guidance need to be in the same VPC
 - Enable **Unauthenticated access** in Access control methods
 - Enable **Plaintext** in Encryption
 - Set **auto.create.topics.enable** as true in MSK cluster configuration. This configuration sets whether MSK cluster can create topic automatically.
 - The value of **default.replication.factor** cannot be larger than the number of MKS cluster brokers

Note

If there is no MSK cluster, the user needs to create an MSK Cluster following above requirements.

- **Topic:** The user can specify a topic name. By default, the guidance will create a topic with “project-id”.

Self-hosted Kafka

Users can also use self-hosted Kafka clusters. To integrate the guidance with Kafka clusters, provide the following configurations:

- **Broker link:** Enter the brokers link of Kafka cluster that you wish to connect to. The Kafka cluster needs to meet the following requirements:
 - The Kafka cluster and this guidance need to be in the same VPC.
 - At least two Kafka cluster brokers are available.
- **Topic:** User can specify the topic for storing the data
- **Security Group:** This VPC security group defines which subnets and IP ranges can access the Kafka cluster.

Connector

Enable guidance to create Kafka connector and a custom plugin for this connector. This connector will sink the data from Kafka cluster to S3 bucket.

Additional Settings

- **Sink maximum interval:** Specifies the maximum length of time (in seconds) that records should be buffered before streaming to the AWS service.
- **Batch size:** The maximum number of records to deliver in a single batch.

Data sink – Kinesis

This data sink will stream the clickstream data collected by the ingestion endpoint into KDS. The guidance will create a KDS in your AWS account based on your specifications.

Provision mode

Two modes are available: **On-demand** and **Provisioned**

- **On-demand:** In this mode, KDS shards are provisioned based on the workshop automatically. On-demand mode is suited for workloads with unpredictable and highly-variable traffic patterns.

- **Provisioned:** In this mode, KDS shards are set at creation. The provisioned mode is suited for predictable traffic with capacity requirements that are easy to forecast. You can also use the provisioned mode if you want fine-grained control over how data is distributed across shards.
 - **Shard number:** With the provisioned mode, you must specify the number of shards for the data stream. For more information, please refer to [provisioned mode](#).

Additional settings

- **Sink maximum interval:** You can specify the maximum interval (in seconds) that records should be buffered before streaming to the AWS service.
- **Batch size:** You can specify the maximum number of records to deliver in a single batch.

Data sink - S3

In this option, clickstream data is buffered in the memory of ingestion Server, then sink into a S3 bucket. This option provides the best cost-performance in case real-time data consumption is not required.

Note

Unlike Kafka and KDS data sink, this option buffers data in the ingestion server and responses 200 code to SDK client before sink into S3, so there is chance data could be lost while ingestion server fails and auto-scaled machine is in the process of creation. But it is worth to note that this probability is very low because of the High-availability design of the guidance.

- **Buffer size:** Specify the data size to buffer before sending to Amazon S3. The higher buffer size may be lower in cost with higher latency, while the lower buffer size will be faster in delivery with higher cost. Min: 1 MiB, Max: 50 MiB
- **Buffer interval:** Specify the maximum interval (in seconds) for saving buffer to S3. The higher interval allows more time to collect data and the size of data may be bigger. The lower interval sends the data more frequently and may be more advantageous when looking at shorter cycles of data activity. Min: 60 Seconds, Max: 3600 Seconds

Data processing

Clickstream Analytics on AWS provides an inbuilt data schema to parse and model the raw event data sent from your web and mobile apps, which makes it easy for you to analyze the data in analytics engines (such as RedShift and Athena).

Data Processing module includes two functionalities:

- **Transformation:** Extract the data from files sank by ingestion module, then parse each event data and transform them to guidance data model.
- **Enrichment:** Add additional dimensions/fields to event data.

This chapter includes:

- [Data schema](#)
- [Configure execution parameters](#)
- [Configure custom plugins](#)

Data schema

This article explains the data schema and format in Clickstream Analytics on AWS. This guidance uses an **event-based** data model to store and analyze clickstream data. Every activity (such as click and view) on the clients is modeled as an event with dimensions, and each dimension represents a parameter of the event. Dimensions are common for all events.

You can use JSON objects to store custom event parameters as key-value pairs into special dimensions, which helps you to collect information that is specific for your business. Those JSON objects are stored in special data types, which allow you to extract the values in the analytics engines.

Database and table

For each project, the guidance creates a database with name of `<project-id>` in Amazon Redshift and Athena. Each app will have a schema with name of `app_id`. In Athena, all tables are added partitions of `app_id`, year, month, and day. Based on the event data, the guidance's data-processing module creates the following four base tables:

- **event-v2:** This table stores event data. Each record represents an individual event.

- **user-v2:** This table stores the latest user attributes. Each record represents a visitor (pseudonymous user).
- **item-v2:** This table stores event-item data. Each record represents an event that is associated with an item.
- **session:** This table stores session data. Each record represents a session for each pseudonymous user.

Columns

Each column in the tables represents a specific parameter for an event, user, or item. Some parameters are nested within a `Super` field in Amazon Redshift or a `Map` field in Athena. Those fields (such as `custom_parameters`, `user_properties`) contain parameters that are repeatable. The following table describes the fields.

Event table fields

Field Name	Data Type - Redshift	Data Type - Athena	Description
event_timestamp	TIMESTAMP	TIMESTAMP	The timestamp (in microseconds, UTC) when the event was logged on the client.
event_id	VARCHAR	STRING	Unique ID for the event.
event_time_msec	BIGINT	BIGINT	The time in UNIX timestamp format (microseconds) when the event was logged on the client.
event_name	VARCHAR	STRING	The name of the event.
event_value	DOUBLE PRECISION	FLOAT	The value of the event's "value" parameter.

Field Name	Data Type - Redshift	Data Type - Athena	Description
event_value_currency	VARCHAR	STRING	The currency of the value associated with the event.
event_bundle_sequence_id	BIGINT	BIGINT	The sequential ID of the bundle in which these events were uploaded.
ingest_timestamp	BIGINT	BIGINT	Timestamp offset between collection time and upload time in micros.
device.mobile_brand_name	VARCHAR	STRING	The device brand name.
device.mobile_model_name	VARCHAR	STRING	The device model name.
device.manufacturer	VARCHAR	STRING	The device manufacturer name.
device.carrier	VARCHAR	STRING	The device network provider name.
device.network_type	VARCHAR	STRING	The network_type of the device, e.g., WIFI, 5G
device.operating_system	VARCHAR	STRING	The operating system of the device.
device.operating_system_version	VARCHAR	STRING	The OS version.

Field Name	Data Type - Redshift	Data Type - Athena	Description
device.vendor_id	VARCHAR	STRING	IDFV (present only if IDFA is not collected).
device.advertising_id	VARCHAR	STRING	Advertising ID/IDFA.
device.system_language	VARCHAR	STRING	The OS language.
device.time_zone_offset_seconds	BIGINT	BIGINT	The offset from GMT in seconds.
device.ua_browser	VARCHAR	STRING	The browser in which the user viewed content, derived from User Agent string
device.ua_browser_version	VARCHAR	STRING	The version of the browser in which the user viewed content, derive from User Agent
device.ua_device	VARCHAR	STRING	The device in which user viewed content, derive from User Agent.
device.ua_device_category	VARCHAR	STRING	The device category in which user viewed content, derive from User Agent.
device.ua_os	VARCHAR	STRING	The operating system of the device in which user viewed content, derive from User Agent.

Field Name	Data Type - Redshift	Data Type - Athena	Description
device.ua_os_version	VARCHAR	STRING	The operating system version of the device category in which user viewed content, derive from User Agent.
device.ua	SUPER	MAP	The parsed User Agent in key-value pairs
device.screen_width	VARCHAR	STRING	The screen width of the device.
device.screen_height	VARCHAR	STRING	The screen height of the device.
device.viewport_width	VARCHAR	STRING	The screen width of the browser viewport.
device.viewport_height	VARCHAR	STRING	The screen height of the browser viewport.
geo.continent	VARCHAR	STRING	The continent from which events were reported, based on IP address.
geo.sub_continent	VARCHAR	STRING	The subcontinent from which events were reported, based on IP address.
geo.country	VARCHAR	STRING	The country from which events were reported, based on IP address.

Field Name	Data Type - Redshift	Data Type - Athena	Description
geo.region	VARCHAR	STRING	The region from which events were reported, based on IP address.
geo.metro	VARCHAR	STRING	The metro from which events were reported, based on IP address.
geo.city	VARCHAR	STRING	The city from which events were reported, based on IP address.
geo.locale	VARCHAR	STRING	The locale information obtained from device.
traffic_source_name	VARCHAR	STRING	Name of the marketing campaign that acquired the user when the events were reported.
traffic_source_medium	VARCHAR	STRING	Name of the medium (paid search, organic search, email, etc.) that acquired the user when the events were reported.
traffic_source_campaign	VARCHAR	STRING	The marketing campaign (derive from utm_campaign) associated with the event.

Field Name	Data Type - Redshift	Data Type - Athena	Description
traffic_source_content	VARCHAR	STRING	The marketing campaign content (derive from utm_content) associated with the event.
traffic_source_term	VARCHAR	STRING	The marketing campaign term (derive from utm_term) associated with the event.
traffic_source_campaign_id	VARCHAR	STRING	The marketing campaign id (derive from utm_id) associated with the event.
traffic_source_click_id	VARCHAR	STRING	The click id associated with the event.
traffic_source_click_platform	VARCHAR	STRING	The platform of the click id associated with the event.
traffic_source_channel_group	VARCHAR	STRING	The channel group (assigned by traffic classification rules) associated with the event.

Field Name	Data Type - Redshift	Data Type - Athena	Description
traffic_source_category	VARCHAR	STRING	The source category (i.e., Search, Social, Video, Shopping) based on the traffic source associated with the event.
user_first_touch_time_msec	BIGINT	BIGINT	The time in UNIX timestamp format (microseconds) when the user first touch the app or website.
app_package_id	VARCHAR	STRING	The package name or bundle ID of the app.
app_version	VARCHAR	STRING	The app's versionName (Android) or short bundle version.
app_title	VARCHAR	STRING	The app's name.
app_id	VARCHAR	STRING	The App ID (created by this guidance) associated with the app.
app_install_source	VARCHAR	STRING	The store from which user installed the app.
platform	VARCHAR	STRING	The data stream platform (Web, IOS or Android) from which the event originated.

Field Name	Data Type - Redshift	Data Type - Athena	Description
project_id	VARCHAR	STRING	The project id associated with the app.
screen_view_screen_name	VARCHAR	STRING	The screen name associated with the event.
screen_view_screen_id	VARCHAR	STRING	The screen class id associated with the event.
screen_view_screen_unique_id	VARCHAR	STRING	The unique screen id associated with the event
screen_view_previous_screen_name	VARCHAR	STRING	The previous unique screen id associated with the event.
screen_view_previous_screen_id	VARCHAR	STRING	The previous unique screen id associated with the event.
screen_view_previous_screen_unique_id	VARCHAR	STRING	The previous unique screen id associated with the event.
screen_view_entrances	BOOLEAN	BOOLEAN	Whether the screen is the entrance view of the session.
page_view_page_referrer	VARCHAR	STRING	The referrer page url.
page_view_page_referrer_title	VARCHAR	STRING	The referrer page title.

Field Name	Data Type - Redshift	Data Type - Athena	Description
page_view_previous_time_msec	BIGINT	BIGINT	The timestamp of the previous page_view event.
page_view_engagement_time_msec	BIGINT	BIGINT	The previous page_view duration in milliseconds.
page_view_page_title	VARCHAR	STRING	The title of the webpage associated with the event.
page_view_page_url	VARCHAR	STRING	The url of the webpage associated with the event.
page_view_page_url_path	VARCHAR	STRING	The url path of the webpage associated with the event.
page_view_page_url_query_parameters	SUPER	MAP	The query parameters in key-value pairs of the page url associated with the event.
page_view_hostname	VARCHAR	STRING	The host name of the web page associated with the event.
page_view_latest_referrer	VARCHAR	STRING	The url of the latest external referrer.
page_view_latest_referrer_host	VARCHAR	STRING	The hostname of the latest external referrer.

Field Name	Data Type - Redshift	Data Type - Athena	Description
page_view_entrances	BOOLEAN	BOOLEAN	Whether the page is the entrance view of the session.
app_start_is_first_time	BOOLEAN	BOOLEAN	Whether the app start is a new app launch.
upgrade_previous_app_version	VARCHAR	STRING	Previous app version before app upgrade event.
upgrade_previous_os_version	VARCHAR	STRING	Previous os version before OS upgrade event.
search_key	VARCHAR	STRING	The name of the keyword in the URL when user perform search on web site.
search_term	VARCHAR	STRING	The search content in the URL when user perform search on web site.
outbound_link_classes	VARCHAR	STRING	The content of class in tag that associated with the outbound link.
outbound_link_domain	VARCHAR	STRING	The domain of href in tag that associated with the outbound link.

Field Name	Data Type - Redshift	Data Type - Athena	Description
outbound_link_id	VARCHAR	STRING	The content of id in tag that associated with the outbound link.
outbound_link_url	VARCHAR	STRING	The content of href in tag that associated with the outbound link.
outbound_link	BOOLEAN	BOOLEAN	Whether the link is outbound link or not.
user_engagement_time_msec	BIGINT	BIGINT	The user engagement duration in milliseconds.
user_id	VARCHAR	STRING	The unique ID assigned to a user through setUserId() API.
user_pseudo_id	VARCHAR	STRING	The pseudonymous id generated by SDK for the user.
session_id	VARCHAR	STRING	The session id associated with the event.
session_start_time_msec	BIGINT	BIGINT	The start time in UNIX timestamp of the session.
session_duration	BIGINT	BIGINT	The duration the session lasts, in milliseconds.

Field Name	Data Type - Redshift	Data Type - Athena	Description
session_number	BIGINT	BIGINT	Number of the sessions generated from the client.
scroll_engagement_time_msec	BIGINT	BIGINT	The engagement time on the web page until user scroll.
sdk_error_code	VARCHAR	STRING	The error code generated by SDK when an event is invalid in some way.
sdk_error_message	VARCHAR	STRING	The error message generated by SDK an event is invalid in some way.
sdk_version	VARCHAR	STRING	The version of the SDK.
sdk_name	VARCHAR	STRING	The name of the SDK.
app_exception_message	VARCHAR	STRING	The exception message when the app crashes or throws an exception.
app_exception_stack	VARCHAR	STRING	The exception stack trace when the app crashes or throws an exception.
custom_parameters_json_str	VARCHAR	STRING	All the custom event parameters stored in key-value pairs.

Field Name	Data Type - Redshift	Data Type - Athena	Description
custom_parameters	SUPER	MAP	All the custom event parameters stored in key-value pairs.
process_info	SUPER	MAP	Store information about the data processing.
created_time	TIMESTAMP	TIMESTAMP	Store information about the data processing.

User table fields

Field Name	Data Type - Redshift	Data Type - Athena	Description
event_timestamp	BIGINT	STRING	The timestamp of when the user attributes was collected.
user_id	VARCHAR	STRING	The unique ID assigned to a user through <code>setUserId()</code> API.
user_pseudo_id	VARCHAR	STRING	The pseudonymous id generated by SDK for the user.
user_properties	SUPER	ARRAY	Properties of the user.
user_properties_json_str	VARCHAR	STRING	Properties of the user.

Field Name	Data Type - Redshift	Data Type - Athena	Description
first_touch_timest amp	BIGINT	BIGINT	The time (in microseconds) at which the user first opened the app or visited the site.
first_visit_date	Date	Date	Date of the user's first visit.
first_referer	VARCHAR	STRING	The first referer detected for the user.
first_traffic_source	VARCHAR	STRING	The the network source that acquired the user that was first detected for the user, e.g., Google, Baidu
first_traffic_sour ce_medium	VARCHAR	STRING	The medium of the network source that acquired the user that was first detected for the user, e.g., paid search, organic search, email, etc
first_traffic_sour ce_campaign	VARCHAR	STRING	The name of the marketing campaign that acquired the user that was first detected for the user.

Field Name	Data Type - Redshift	Data Type - Athena	Description
first_traffic_source_content	VARCHAR	STRING	The marketing campaign content that acquired the user that was first detected for the user.
first_traffic_source_term	VARCHAR	STRING	The keyword of the marketing ads that acquired the user that was first detected for the user.
first_traffic_source_campaign_id	VARCHAR	STRING	The id of the marketing campaign that acquired the user that was first detected for the user.
first_traffic_source_clid_platform	VARCHAR	STRING	The click id platform of the marketing campaign that acquired the user that was first detected for the user.
first_traffic_source_clid	VARCHAR	STRING	The click id of the marketing campaign that acquired the user that was first detected for the user.
first_traffic_source_channel_group first_traffic_source_category	VARCHAR	STRING	The channel group of the traffic source that acquired the user that was first detected for the user.

Field Name	Data Type - Redshift	Data Type - Athena	Description
first_app_install_source	VARCHAR	STRING	The source category (i.e., Search, Social, Video, Shopping) based on the traffic source that acquired the user for the first time.
process_info	SUPER	MAP	The install channel for the user, e.g., Google Play Store information about the data processing.
created_time	TIMESTAMP	TIMESTAMP	Store information about the data processing.

Session table fields

Field Name	Data Type - Redshift	Data Type - Athena	Description
event_timestamp	TIMESTAMP	STRING	The timestamp of when the event occurred.
user_pseudo_id	VARCHAR	STRING	The pseudonymous ID generated by the SDK for the user.
session_id	VARCHAR	STRING	The ID assigned to a session.
user_id	VARCHAR	STRING	The unique ID assigned to a user

Field Name	Data Type - Redshift	Data Type - Athena	Description
			through the <code>setUserId()</code> API.
<code>session_number</code>	BIGINT	INT	The sequence number of the session in the client.
<code>session_start_time_msec</code>	BIGINT	BIGINT	The start time of the session in milliseconds.
<code>session_source</code>	VARCHAR	STRING	The traffic source of the session.
<code>session_medium</code>	VARCHAR	STRING	The traffic source medium of the session.
<code>session_campaign</code>	VARCHAR	STRING	The traffic source campaign of the session.
<code>session_content</code>	VARCHAR	STRING	The traffic source content of the session.
<code>session_term</code>	VARCHAR	STRING	The traffic source term of the session.
<code>session_campaign_id</code>	VARCHAR	STRING	The traffic source campaign ID of the session.
<code>session_clid_platform</code>	VARCHAR	STRING	The platform of the CLID (Click ID) of the session.

Field Name	Data Type - Redshift	Data Type - Athena	Description
session_clid	VARCHAR	STRING	The CLID (Click ID) of the session.
session_channel_group	VARCHAR	STRING	The traffic source channel group of the session.
session_source_category	VARCHAR	STRING	The traffic source category of the session source.
process_info	SUPER	MAP	Additional data processing information.
created_time	TIMESTAMP	STRING	The timestamp of when the session data was created.

Item table fields

Field Name	Data Type - Redshift	Data Type - Athena	Description
event_timestamp	TIMESTAMP	STRING	The timestamp of when the event occurred.
event_id	VARCHAR	STRING	The ID of the event.
event_name	VARCHAR	STRING	The name of the event.
platform	VARCHAR	STRING	The platform associated with the event.

Field Name	Data Type - Redshift	Data Type - Athena	Description
user_pseudo_id	VARCHAR	STRING	The pseudonymous ID generated by the SDK for the user.
user_id	VARCHAR	STRING	The unique ID assigned to a user through the <code>setUserId()</code> API.
item_id	VARCHAR	STRING	The ID of the item.
name	VARCHAR	STRING	The name of the item.
brand	VARCHAR	STRING	The brand of the item.
currency	VARCHAR	STRING	The currency associated with the item price.
price	DOUBLE PRECISION	DOUBLE	The price of the item.
quantity	DOUBLE PRECISION	DOUBLE	The quantity of the item in the event.
creative_name	VARCHAR	STRING	The name of the creative associated with the item.
creative_slot	VARCHAR	STRING	The slot of the creative associated with the item.
location_id	VARCHAR	STRING	The ID of the location associated with the item.

Field Name	Data Type - Redshift	Data Type - Athena	Description
category	VARCHAR	STRING	The category of the item.
category2	VARCHAR	STRING	The second category of the item.
category3	VARCHAR	STRING	The third category of the item.
category4	VARCHAR	STRING	The fourth category of the item.
category5	VARCHAR	STRING	The fifth category of the item.
custom_parameters_ json_str	VARCHAR	STRING	The JSON string representation of custom parameters.
custom_parameters	SUPER	MAP	Additional custom parameters.
process_info	SUPER	MAP	Additional process information.
created_time	TIMESTAMP	STRING	The timestamp of when the item data was created.

Execution parameters

Execution parameters control how the transformation and enrichment jobs are orchestrated.

Parameters

You can configure the following **Execution parameters** after you turn on **Enable data processing**.

Parameter	Description	Values
Data processing interval/ Fixed Rate	Specify the interval to batch the data for data processing by fixed rate	1 hour 12 hours 1 day
Data processing interval/ Cron Expression	Specify the interval to batch the data for data processing by cron expression	cron(0 * * ? *) cron(0 0,12 * ? *) cron(0 0 * ? *)
Event freshness	Specify the days after which the guidance will ignore the event data. For example, if you specify 3 days for this parameter, the guidance will ignore any event which arrived more than 3 days after the events are triggered	3 days 5 days 30 days

Cron expression syntax

Syntax

cron(minutes hours day-of-month month day-of-week year)

For more information, refer to [Cron-based schedules](#).

Config Spark job parameters

By default, the Clickstream pipeline automatically adjusts EMR job parameters based on the dataset volume that requires processing. In most of time, you do not need to adjust the EMR job parameters, but if you want to override the EMR job parameters, you can put `spark-config.json` file in S3 bucket to set your job parameters.

To add your customized the EMR job parameters, you can add a file `s3://{PipelineS3Bucket}/{PipelineS3Prefix}{ProjectId}/config/spark-config.json` in the S3 bucket.

Please replace `{PipelineS3Bucket}`, `{PipelineS3Prefix}`, and `{ProjectId}` with the values of your data pipeline. These values are found in the `Clickstream-DataProcessing-<uuid>` stack's Parameters.

Also, you can get these values by running the below commands,

```
stackNames=$(aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
UPDATE_COMPLETE --no-paginate | jq -r '.StackSummaries[].StackName' | grep
Clickstream-DataProcessing | grep -v Nested)

echo -e "$stackNames" | while read stackName; do
    aws cloudformation describe-stacks --stack-name $stackName | jq
'.Stacks[].Parameters' | jq 'map(select(.ParameterKey == "PipelineS3Bucket"
or .ParameterKey == "PipelineS3Prefix" or .ParameterKey == "ProjectId"))'
done
```

Here is an example of the file `spark-config.json`:

```
{
  "sparkConfig": [
    "spark.emr-serverless.executor.disk=200g",
    "spark.executor.instances=16",
    "spark.dynamicAllocation.initialExecutors=16",
    "spark.executor.memory=100g",
    "spark.executor.cores=16",
    "spark.network.timeout=10000000",
    "spark.executor.heartbeatInterval=10000000",
    "spark.shuffle.registration.timeout=120000",
    "spark.shuffle.registration.maxAttempts=5",
    "spark.shuffle.file.buffer=2m",
    "spark.shuffle.unsafe.file.output.buffer=1m"
  ],
  "inputRePartitions": 2000
}
```

Please make sure your account has enough `emr-serverless` quotas, you can view the quotas via `emr-serverless-quotas` in the Region `us-east-1`. For more configurations, please refer to [Spark job properties](#) and application [worker config](#).

Processing plugin

There are two types of plugins: **transformer** and **enrichment**. You can choose to have only one **transformer**, and zero or multiple **enrichment**.

Built-in plugins

Below plugins are provided by Clickstream Analytics on AWS.

Plugin name	Type	Description
UaEnrichment	enrichment	User-agent enrichment, use ua_parser Java library to enrich User-Agent in the HTTP header to ua_browser, ua_browser_version, ua_os, ua_os_version, ua_device
IpEnrichment	enrichment	IP address enrichment, use GeoLite2 data by MaxMind to enrich IP to city, continent, country

The UaEnrichment uses [UA Parser](#) to parse user-agent in Http header

The IpEnrichment plugin uses [GeoLite2-City data](#) created by MaxMind, available from <https://www.maxmind.com>

Custom plugins

You can add custom plugins to transform raw event data or enrich the data for your need.

Note

To add custom plugins, you must develop your own plugins firstly, see [Develop Custom Plugins](#).

To add your plugins, choose **Add Plugin**, which will open a new window, in which you can upload your plugins.

1. Enter the plugin **Name** and **Description**
2. Choose **Plugin Type**
3. **Enrichment:** Plugin to add fields into event data collected by SDK (both Clickstream SDK or third-party SDK)
4. **Transformation:** A plugin used to transform a third-party SDK's raw data into guidance built-in schema
5. Upload plugin java JAR file
6. (Optional) Upload the dependency files if any
7. **Main function class:** fill the full class name of your plugin class name, e.g. com.company.sol.CustomTransformer

Develop Custom Plugins

The simplest way to develop custom plugins is making changes based on our example project.

1. Clone/Fork the example project.

```
git clone https://github.com/aws-solutions-library-samples/guidance-for-clickstream-analytics-on-aws.git
cd examples/custom-plugins
```

- For enrichment plugin, please refer to the example: custom-enrich/
 - For transformer plugin, please refer to the example: custom-sdk-transformer/
2. Change packages and classes name as you desired.
 3. Implement the method `public Dataset<row> transform(Dataset<row> dataset)` to do transformation or enrichment.
 4. (Optional) Write test code.
 5. Run gradle to package code to jar `./gradlew clean build`.
 6. Get the jar file in build output directory `./build/libs/`.

Data modeling

Once the data pipeline processes the event data, you can load the data into an analytics engine for data modeling, such as Redshift or Athena, where data will be aggregated and organized into different views (such as event, device, session), as well as calculated metrics that are commonly used. Below are the preset data views this guidance provides if you choose to enable data modeling module.

You can choose to use Redshift or Athena, or both.

Note

We recommended you select both, that is, using Redshift for hot data modeling and using Athena for all-time data analysis.

You can set below configurations for Redshift.

- **Redshift Mode:** Select Redshift serverless or provisioned mode.
 - **Serverless mode**
 - **Base RPU:** RPU stands for Redshift Processing Unit. Amazon Redshift Serverless measures data warehouse capacity in RPUs, which are resources used to handle workloads. The base capacity specifies the base data warehouse capacity Amazon Redshift uses to serve queries and is specified in RPUs. Setting higher base capacity improves query performance, especially for data processing jobs that consume a lot of resources.
 - **VPC:** A virtual private cloud (VPC) based on the Amazon VPC service is your private, logically isolated network in the AWS Cloud.

Note

If you place the cluster within the isolated subnets, the VPC must have VPC endpoints for S3, Logs, Dynamodb, STS, States, Redshift and Redshift-data service.

- **Security Group:** This VPC security group defines which subnets and IP ranges can access the endpoint of Redshift cluster.
- **Subnets:** Select at least three existing VPC subnets.

Note

We recommend using private subnets to deploy in accordance with the security best practices.

Provisioned mode

- **Redshift Cluster:** With a provisioned Amazon Redshift cluster, you build a cluster with node types that meet your cost and performance specifications. You have to set up, tune, and manage Amazon Redshift provisioned clusters.
- **Database user:** The guidance requires permissions to access and create database in Redshift cluster. By default, it grants Redshift Data API with the permissions of the admin user to execute the commands to create DB, tables, and views, as well as loading data.
- **Data range:** Considering the cost performance issue of having Redshift to save all the data, we recommend that Redshift save hot data and that all data are stored in S3. It is necessary to delete expired data in Redshift on a regular basis.
- **Athena:** Choose Athena to query all data on S3 using the table created in the AWS Glue Data Catalog.

Reporting

Once the data are processed and modeled by the data pipeline, you can enable the Analytics Studio for the pipeline, which will allow the guidance create out-of-the-box dashboards in QuickSight, provide advanced analytics model for user to query their clickstream data, and data management functionalities.

Note

To enable this module, your AWS account needs to have subscription in QuickSight. If it hasn't, please follow this [sign up for Quick](#) to create a subscription first.

You need to make the following configuration for Reporting.

- **Create sample dashboard in QuickSight:** Enabling this feature allows the guidance to create sample dashboards in your QuickSight account.

- **QuickSight user:** Select an admin user for the guidance to create QuickSight resources. (Only required for AWS China Regions)

App registration

After the data pipeline transitions to an **Active** state, register your application(s) with the pipeline to enable the reception of clickstream data. After the application is added, you can integrate the SDK into your application to send data to the pipeline. Complete the following steps to register an application to a data pipeline.

Steps

1. Sign in to Clickstream Analytics on AWS Console.
2. In the left navigation pane, choose **Projects**, then select a project you want to register app, click its title, and it will bring you to the project page.
3. Click on **+ Add application** to start adding application to the pipeline.
4. Complete the form by filling in the following fields:
 - **App name:** Provide a name for your app.
 - **App ID:** The system will generate one ID based on the name, which you can customize if needed.
 - **App reporting time zone:** Specify a reporting time zone for your app. This time zone is used by the preset dashboard to calculate daily metrics.
 - **Description:** Provide a description for your app.
 - **Android package name:** Provide a package name if your app has an Android client.
 - **App Bundle ID:** Provide the bundle ID if your app has an iOS client.
5. Choose **Register App & Generate SDK Instruction**, and wait for the registration to be completed.
6. You should now see tabs for SDK integration for different platforms. Choose the relevant tab to view detailed instructions on adding the SDK to your application. Follow the provided steps to integrate the SDK. For certain platforms, you can choose the **Download the config json file** button to obtain the configuration file.

The pipeline update with the newly added application takes approximately 3 to 5 minutes. After the pipeline status returns to **Active**, it is ready to receive data from your application.

Pipeline maintenance

This guidance provides three features to help you manage and operate the data pipeline after it gets created.

Monitoring and Alarms

The guidance collects metrics from each resource in the data pipeline and creates monitoring dashboards in CloudWatch, which provides you a comprehensive view into the pipeline status. It also provides a set of alarms that will notify project owner if anything goes abnormal.

Following are steps to view monitoring dashboards and alarms.

Monitoring dashboards

To view monitoring dashboard for a data pipeline, follows below steps:

1. Go to project detail page.
2. Choose project id or **View Details**, which will direct to the pipeline detail page.
3. Select the "**Monitoring**" tab.
4. In the tab, choose **View in CloudWatch**, which will direct you to the monitoring dashboard.

Alarms

To view alarms for a data pipeline, follows below steps:

1. Go to project detail page.
2. Choose project id or **View Details**, which will direct to the pipeline detail page.
3. Select the "**Alarms**" tab.
4. In the tab, you can view all the alarms. You can also choose **View in CloudWatch**, which will direct you to CloudWatch alarm pages to view alarm details.
5. You can also enable or disable an alarm by selecting the alarm then choosing **Enable** or **Disable**.

Pipeline modification

You are able to modify some configuration the data pipeline after it created, follow below steps to update a pipeline.

1. Go to project detail page.
2. Choose project id or **View Details**, which will direct to the pipeline detail page.
3. In the project details page, choose **Edit**, which will bring you to the pipeline creation wizard page. Note that some configuration are in disable mode, which means they cannot be updated after creation.
4. If needed, update those configuration options which are editable.
5. After editing the configuration, choose **Next** until you reach last page, and choose **Save**.

You will see pipeline is in Updating status.

Pipeline upgrade

For detailed procedure, see [upgrade the guidance](#).

SDK manual

Clickstream Analytics on AWS provides purpose-built software development kits (SDKs), which can make it easier for you to report events to the data pipeline created in the guidance. Currently, the guidance supports the following platforms:

- [Android](#)
- [Swift](#)
- [Web](#)
- [Flutter](#)
- [ReactNative SDK](#)

In addition, HTTP API is also provided to collect clickstream data from other platforms, such as server, through HTTP request.

Key features and benefits

- **Automatic data collection.** Clickstream SDKs provide built-in capabilities to automatically collect common events, such as screen view, session, and user engagement, so that you only need to focus on recording business-specific events.
- **Ease of use.** Clickstream SDKs provide multiple APIs and configuration options to simplify the event reporting and attribute setting process.
- **Cross-platform analytics.** Clickstream SDKs are consistent in event data structure, attribute validation rules, and event sending mechanism, so that data can be normalized in the same structure for cross-platform analytics.

Note

All Clickstream SDKs are open source under Apache 2.0 License in [Github](#). You can customize the SDKs if needed. All contributions are welcome.

Android SDK

Introduction

Clickstream Android SDK can help you easily collect in-app click stream data from Android devices to your AWS environments through the data pipeline provisioned by this guidance.

The SDK is based on the Amplify for Android SDK Core Library and developed according to the Amplify Android SDK plug-in specification. In addition, the SDK provides features that automatically collect common user events and attributes (for example, screen view and first open) to accelerate data collection for users.

Platform support

Clickstream Android SDK supports Android 4.1 (API level 16) and later.

Integrate the SDK

1 Include the SDK

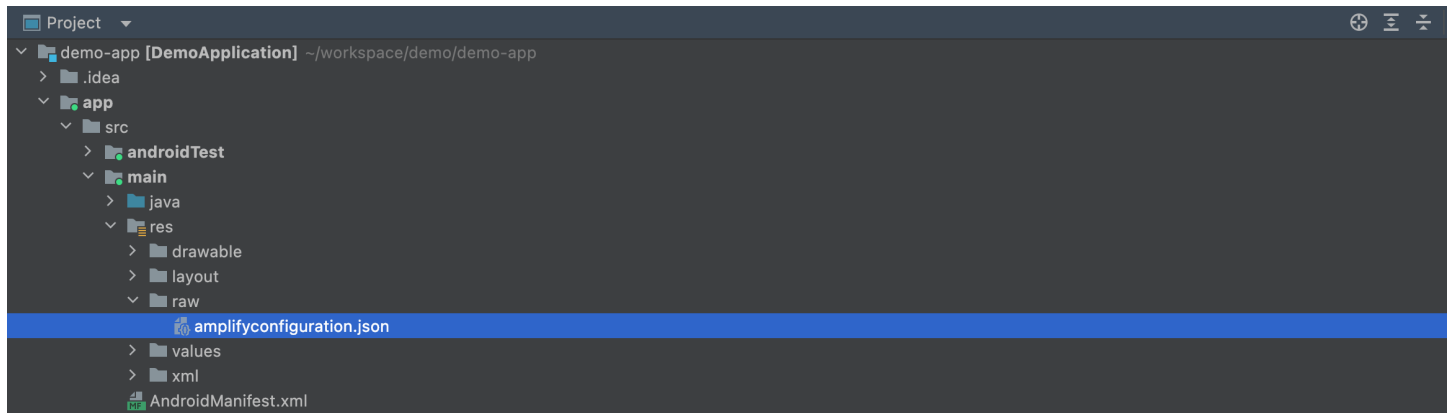
Add the Clickstream SDK dependency to your app module's `build.gradle` file. For example,

```
dependencies {  
    implementation 'software.aws.solution:clickstream:0.13.0'  
}
```

If needed, you can synchronize your project with [the latest version](#).

2 Configure parameters

Find the `res` directory under your `project/app/src/main`, and manually create a `raw` folder in the `res` directory.



Download your `amplifyconfiguration.json` file from your clickstream control plane, and paste it to the raw folder. The JSON file is like:

```
{
  "analytics": {
    "plugins": {
      "awsClickstreamPlugin": {
        "appId": "appId",
        "endpoint": "https://example.com/collect",
        "isCompressEvents": true,
        "autoFlushEventsInterval": 10000,
        "isTrackAppExceptionEvents": false
      }
    }
  }
}
```

In the file, your `appId` and `endpoint` are already set up. The explanation for each property is as follows:

- **appId (Required):** the app id of your project in web console.
- **endpoint (Required):** the endpoint url you will upload the event to AWS server.
- **isCompressEvents:** whether to compress event content when uploading events, and the default value is `true`
- **autoFlushEventsInterval:** event sending interval, and the default value is `10s`
- **isTrackAppExceptionEvents:** whether auto track exception event in app, and the default value is `false`

3 Initialize the SDK

It is recommended that you initialize the SDK in your application's `onCreate()` method.

Note

The initialization code needs to run in the main thread.

```
import software.aws.solution.clickstream.ClickstreamAnalytics;

public void onCreate() {
    super.onCreate();

    try{
        ClickstreamAnalytics.init(getApplicationContext());
        Log.i("MyApp", "Initialized ClickstreamAnalytics");
    } catch (AmplifyException error){
        Log.e("MyApp", "Could not initialize ClickstreamAnalytics", error);
    }
}
```

4 Start using

Record event

Add the following code where you need to report an event.

```
import software.aws.solution.clickstream.ClickstreamAnalytics;
import software.aws.solution.clickstream.ClickstreamEvent;

// for record an event with custom attributes
ClickstreamEvent event = ClickstreamEvent.builder()
    .name("button_click")
    .add("category", "shoes")
    .add("currency", "CNY")
    .add("value", 279.9)
    .build();
ClickstreamAnalytics.recordEvent(event);

// for record an event directly
```

```
ClickstreamAnalytics.recordEvent("button_click");
```

Add global attribute

1. Add global attributes when initializing the SDK. The following example code shows how to add traffic source fields as global attributes when initializing the SDK.

```
import software.aws.solution.clickstream.ClickstreamAnalytics;

ClickstreamAttribute globalAttributes = ClickstreamAttribute.builder()
    .add(ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_SOURCE, "amazon")
    .add(ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_MEDIUM, "cpc")
    .add(ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_CAMPAIGN, "summer_promotion")
    .add(ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_CAMPAIGN_ID,
        "summer_promotion_01")
    .add(ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_TERM, "running_shoes")
    .add(ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_CONTENT, "banner_ad_1")
    .add(ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_CLID, "amazon_ad_123")
    .add(ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_CLID_PLATFORM, "amazon_ads")
    .add(ClickstreamAnalytics.Attr.APP_INSTALL_CHANNEL, "Amazon Store")
    .build();

ClickstreamConfiguration configuration = new ClickstreamConfiguration()
    .withAppId("your appId")
    .withEndpoint("http://example.com/collect")
    .withInitialGlobalAttributes(globalAttributes);

ClickstreamAnalytics.init(getApplicationContext(), configuration);
```

2. Add global attributes after initializing the SDK.

```
import software.aws.solution.clickstream.ClickstreamAnalytics;
import software.aws.solution.clickstream.ClickstreamAttribute;

ClickstreamAttribute globalAttribute = ClickstreamAttribute.builder()
    .add(ClickstreamAnalytics.Attr.APP_INSTALL_CHANNEL, "Amazon Store")
    .add("level", 5.1)
    .add("class", 6)
    .add("isOpenNotification", true)
    .build();

ClickstreamAnalytics.addGlobalAttributes(globalAttribute);
```

It is recommended to set global attributes when initializing the SDK. Global attributes will be included in all events that occur after it is set.

Delete global attribute

```
import software.aws.solution.clickstream.ClickstreamAnalytics;
ClickstreamAnalytics.deleteGlobalAttributes("level");
```

Login and logout

```
import software.aws.solution.clickstream.ClickstreamAnalytics;
```

```
// when user login success
ClickstreamAnalytics.setUserId("UserId");
```

```
// when user logout
ClickstreamAnalytics.setUserId(null);
```

Add user attribute

```
import software.aws.solution.clickstream.ClickstreamAnalytcs;
import software.aws.solution.clickstream.ClickstreamUserAttribute;

ClickstreamUserAttribute clickstreamUserAttribute = ClickstreamUserAttribute.builder()
    .add("_user_age", 21)
    .add("_user_name", "carl")
    .build();
ClickstreamAnalytics.addUserAttributes(clickstreamUserAttribute);
```

The currently logged-in user's attributes will be cached in disk. The next time the app launches, you don't need to set all of the user's attributes again. You can use the same API `ClickstreamAnalytics.addUserAttributes()` to update the the current user's attributes when they change.

⚠ Important

If your application is already published and most users have already logged in, please manually set the user attributes once when integrating the Clickstream SDK for the first time. This ensures that subsequent events contain user attributes.

Record event with items

You can add the following code to log an event with an item.

```
import software.aws.solution.clickstream.ClickstreamAnalytics;
import software.aws.solution.clickstream.ClickstreamItem;

ClickstreamItem item_book = ClickstreamItem.builder()
    .add(ClickstreamAnalytics.Item.ITEM_ID, "123")
    .add(ClickstreamAnalytics.Item.ITEM_NAME, "Nature")
    .add(ClickstreamAnalytics.Item.ITEM_CATEGORY, "book")
    .add(ClickstreamAnalytics.Item.PRICE, 99)
    .add("book_publisher", "Nature Research")
    .build();

ClickstreamEvent event = ClickstreamEvent.builder()
    .name("view_item")
    .add(ClickstreamAnalytics.Attr.VALUE, 99)
    .add(ClickstreamAnalytics.Attr.CURRENCY, "USD")
    .add("event_category", "recommended")
    .setItems(new ClickstreamItem[] {item_book})
    .build();

ClickstreamAnalytics.recordEvent(event);
```

For more information about logging more attribute in an item, please refer to Item attributes.

⚠ Important

Only pipelines from version 1.1.0 can handle items with custom attribute. ITEM_ID is a required attribute. If this attribute is not set, the item will be discarded.

Record screen view events manually

By default, the SDK will automatically track the **preset `_screen_view` event** when Activity triggers **`onResume`**.

You can manually record screen view events whether automatic screen view tracking is enabled, add the following code to record a screen view event with two attributes.

- **SCREEN_NAME:** Required. Your screen's name.
- **SCREEN_UNIQUE_ID:** Optional. Set the hashcode of your Fragment or View. If you do not set this attribute, the SDK will set a default value based on the current Activity's **`hashCode`**.

```
import software.aws.solution.clickstream.ClickstreamAnalytics;

ClickstreamEvent event = ClickstreamEvent.builder()
    .name(ClickstreamAnalytics.Event.SCREEN_VIEW)
    .add(ClickstreamAnalytics.Attr.SCREEN_NAME, "HomeFragment")
    .add(ClickstreamAnalytics.Attr.SCREEN_UNIQUE_ID,
        String.valueOf(HomeFragment.hashCode()))
    .build();
ClickstreamAnalytics.recordEvent(event);
```

Send event immediately

```
// for send event immediately.
ClickstreamAnalytics.flushEvent();
```

Disable SDK

You can disable the SDK if needed. After disabling the SDK, the SDK will not handle the logging and sending of any events. Of course you can enable the SDK when you need to continue logging events.

Please note that the disable and enable code needs to be run in the main thread.

```
import software.aws.solution.clickstream.ClickstreamAnalytics;

// disable SDK
ClickstreamAnalytics.disable();

// enable SDK
ClickstreamAnalytics.enable();
```

Other configuration

In addition to the required appId and endpoint, you can configure other information to get more customized usage when initializing the SDK:

```
import software.aws.solution.clickstream.ClickstreamAnalytics;

// config the SDK after initialize.
ClickstreamAnalytics.getClickStreamConfiguration()
    .withAppId("your appId")
    .withEndpoint("https://example.com/collect")
    .withAuthCookie("your authentication cookie")
    .withSendEventsInterval(10000)
    .withSessionTimeoutDuration(1800000)
    .withTrackScreenViewEvents(false)
    .withTrackUserEngagementEvents(false)
    .withTrackAppExceptionEvents(false)
    .withLogEvents(true)
    .withCustomDns(CustomOkhttpDns.getInstance())
    .withCompressEvents(true);

ClickstreamAnalytics.init(getApplicationContext(), configuration);
```

Here is an explanation of each method.

Method name	Parameter type	Required	Default value	Description
withAppId()	String	true	N/A	The app ID of your application in the web console.
withEndpoint()	String	true	N/A	The endpoint path you will upload the event to Clickstream ingestion server.
withAuthCookie()	String	false	N/A	Your auth cookie for AWS application load

Method name	Parameter type	Required	Default value	Description
				balancer auth cookie.
withSendEventsInterval()	long	false	1800000	Event sending interval in milliseconds.
withSessionTimeoutDuration()	long	false	5000	The duration of the session timeout in milliseconds.
withTrackScreenViewEvents()	boolean	false	true	Whether to auto-record screen view events.
withTrackUserEngagementEvents()	boolean	false	true	Whether to auto-record user engagement events.
withTrackAppExceptionEvents()	boolean	false	true	Whether to auto-record app exception events.
withLogEvents()	boolean	false	false	Whether to automatically print event JSON for debugging events.
withCustomDns()	String	false	N/A	The method for setting your custom DNS.

Method name	Parameter type	Required	Default value	Description
withCompressEvents()	boolean	false	true	Whether to compress event content by gzip when uploading events.

By default, we will use the configurations in **amplifyconfiguration.json** file. If you add a custom configuration, the added configuration items will override the default values.

You can also add all the configuration parameters you need in the init method without using the **amplifyconfiguration.json** file.

Configuration update

You can update the default configuration after initializing the SDK, below are the additional configuration options you can customize.

```
import software.aws.solution.clickstream.ClickstreamAnalytics;

// config the SDK after initialize.
ClickstreamAnalytics.getClickStreamConfiguration()
    .withAppId("your appId")
    .withEndpoint("https://example.com/collect")
    .withAuthCookie("your authentication cookie")
    .withTrackScreenViewEvents(false)
    .withTrackUserEngagementEvents(false)
    .withTrackAppExceptionEvents(false)
    .withLogEvents(true)
    .withCustomDns(CustomOkhttpDns.getInstance())
    .withCompressEvents(true);
```

Debug events

You can follow the steps below to view the event raw JSON and debug your events.

1. Enable the **withLogEvents** configuration when initializing the SDK.

```
import software.aws.solution.clickstream.ClickstreamAnalytics;
```

```
// log the event in debug mode.
ClickstreamConfiguration configuration = new ClickstreamConfiguration()
    ...
    .withLogEvents(true);
ClickstreamAnalytics.init(getApplicationContext(), configuration);
```

2. Integrate the SDK and launch your app by Android Studio, then open the **Logcat** window.
3. Input EventRecorder to the filter, and you will see the JSON content of all events recorded by Clickstream Android SDK.

Configure custom DNS

```
import software.aws.solution.clickstream.ClickstreamAnalytics;

// config custom DNS.
ClickstreamAnalytics.getClickStreamConfiguration()
    .withCustomDns(CustomOkhttpDns.getInstance());
```

If you want to use custom DNS for network request, you can create your CustomOkhttpDns which implementation okhttp3.Dns, then configure `.withCustomDns(CustomOkhttpDns.getInstance())` to make it work. Here is an [example code](#).

Data format definition

Data types

Clickstream Android SDK supports the following data types:

Data type	Range	Example
int	-214748364 ~ 2147483647	12
long	-9223372036854775808 ~ 9223372036854775807	26854775808
double	4.9E-324 ~ 1.7976931 348623157E308	3.14
boolean	true, false	true

Data type	Range	Example
String	1024 characters maximum	"Clickstream"

Naming rules

1. The event name and attribute name cannot start with a number, and only contain uppercase and lowercase letters, numbers, and underscores. In case of an invalid event name, it will throw `IllegalArgumentException`. In case of an invalid attribute name or user attribute name, it will discard the attribute and record error.
2. Do not use `_` as prefix in an event name or attribute name, because the `_` prefix is reserved for the guidance.
3. The event name and attribute name are case sensitive, so `Add_to_cart` and `add_to_cart` will be recognized as two different event names.

Event and attribute limitation

To improve the efficiency of querying and analysis, we apply limitations to event data as follows:

Item	Recommended	Maximum	Strategy	Error code
Event name invalid	N/A	N/A	discard event, print log and record <code>_clickstream_error</code> event	1001
Length of event name	less than 25 characters	50 characters	discard event, print log and record <code>_clickstream_error</code> event	1002
Length of event attribute name	less than 25 characters	50 characters	discard the attribute, print log and record	2001

Item	Recommended	Maximum	Strategy	Error code
			error in event attribute	
Attribute name invalid	N/A	N/A	discard the attribute, print log and record error in event attribute	2002
Length of event attribute value	less than 100 characters	1024 characters	discard the attribute, print log and record error in event attribute	2003
Event attribute per event	less than 50 attributes	500 event attributes	discard the attribute, print log and record error in event attribute	2004
User attribute number	less than 25 attributes	100 user attributes	discard the attribute , print log and record _clickstream_error event	3001
Length of user attribute name	less than 25 characters	50 characters	discard the attribute , print log and record _clickstream_error event	3002

Item	Recommended	Maximum	Strategy	Error code
User attribute name invalid	N/A	N/A	discard the attribute , print log and record <code>_clickstream_error</code> event	3003
Length of user attribute value	less than 50 characters	256 characters	discard the attribute , print log and record <code>_clickstream_error</code> event	3004
Item number in one event	under 50 items	100 items	discard the item, print log and record error in event attribute	4001
Length of item attribute value	under 100 characters	256 characters	discard the item, print log and record error in event attribute	4002
Custom item attribute number in one item	under 10 custom attributes	10 custom attributes	discard the item, print log and record error in event attribute	4003
Length of item attribute name	under 25 characters	50 characters	discard the item, print log and record error in event attribute	4004

Item	Recommended	Maximum	Strategy	Error code
Item attribute name invalid	--	--	discard the item, print log and record error in event attribute	4005

Important

- The character limits are the same for single-width character languages (for example, English) and double-width character languages (for example, Chinese).
- The limitation for event attribute per event involves both common attributes and preset attributes.
- If the attribute or user attribute with the same name is added more than twice, the latest value will apply.

Preset events

Automatically collected events

Event name	Triggered	Event attributes
_first_open	when the user launches an app the first time after installation	N/A
_session_start	when a user first open the app or a user returns to the app after 30 minutes of inactivity period	_session_id _session_start_timestamp
_screen_view	when a new screen opens	_screen_name _screen_id _screen_unique_id

Event name	Triggered	Event attributes
		_previous_screen_name _previous_screen_id _previous_screen_unique_id _entrances _previous_timestamp _engagement_time_msec
_user_engagement	when a user navigates away from current screen and the screen is in focus for at least one second	_engagement_time_msec
_app_start	every time the app goes to visible	_is_first_time (when it is the first _app_start event after the application starts, the value is true)
_app_end	every time the app goes to invisible	N/A
_profile_set	when the addUserAttributes() or setUserId() API is called	N/A
_app_exception	when the app crashes	_exception_message _exception_stack
_app_update	when the app is updated to a new version and launched again	_previous_app_version

Event name	Triggered	Event attributes
_os_update	when device operating system is updated to a new version	_previous_os_version
_clickstream_error	event_name is invalid or user attribute is invalid	_error_code _error_message

Session definition

In Clickstream Android SDK, we do not limit the total time of a session. As long as the time between the next entry of the app and the last exit time is within the allowable timeout period, the current session is considered to be continuous.

The **_session_start** event triggered when the app open for the first time, or the app was open to the foreground and the time between the last exit exceeded **session_time_out** period. The following are session-related attributes.

- **_session_id**: It is calculated by concatenating the last 8 characters of `uniqueId` and the current millisecond, for example: `dc7a7a18-20230905-131926703`.
- **_session_duration**: We calculate the `_session_duration` by minus the current event create timestamp and the session's `_session_start_timestamp`. This attribute will be added in every event during the session.
- **_session_number**: The auto increment number of session in current device, and the initial value is 1.
- **Session timeout duration**: By default, it is 30 minutes, which can be customized through the configuraton update API.

Screen view definition

In Clickstream Android SDK, we define the `_screen_view` as an event that records a user's browsing path of screen. When a screen transition started, the `_screen_view` event will be recorded if any of the following conditions is met:

- No screen was previously set.
- The new screen name differs from the previous screen title.

- The new screen id differs from the previous screen id.
- The new screen unique id differs from the previous screen unique id.

This event listens for Activity's `onResume` lifecycle method to judgment the screen transition. In order to track screen browsing path, we use `_previous_screen_name` , `_previous_screen_id` and `_previous_screen_unique_id` to link the previous screen. In addition, there are some other attributes in screen view event.

- **`_screen_unique_id`**: We calculate the screen unique id by getting the current screen's hashcode, for example: "126861252".
- **`_entrances`**: The first screen view event in a session is 1, others is 0.
- **`_previous_timestamp`**: The timestamp of the previous `_screen_view` event.
- **`_engagement_time_msec`**: The previous page last engagement milliseconds.

When the app goes to the background for more than 30 minutes and then is opened again, a new session will be generated, the previous screen information will be cleared, and a new screen view event will be sent.

User engagement definition

In Clickstream Android SDK, we define the `user_engagement` as an event that records the screen browsing time, and we only send this event when user leave the screen and the screen has focus for at least one second.

We define that users leave the screen in the following situations.

- When the user navigates to another screen.
- The user moves the app screen to the background.
- The user exit the app, or kill the process of app.

`engagement_time_msec`: We calculate the milliseconds from when a screen is visible to when the user leave the screen.

Event attributes

Sample event structure

```
{
  "event_type": "add_to_cart",
  "event_id": "460daa08-0717-4385-8f2e-acb5bd019ee7",
  "timestamp": 1667877566697,
  "device_id": "f24bec657ea8eff7",
  "platform": "Android",
  "os_version": "10",
  "make": "Samsung",
  "brand": "Samsung",
  "model": "TAS-AN00",
  "locale": "zh_CN_#Hans",
  "carrier": "CDMA",
  "network_type": "Mobile",
  "screen_height": 2259,
  "screen_width": 1080,
  "zone_offset": 28800000,
  "system_language": "zh",
  "country_code": "CN",
  "sdk_version": "0.7.1",
  "sdk_name": "aws-solution-clickstream-sdk",
  "app_version": "1.0",
  "app_package_name": "com.notepad.app",
  "app_title": "Notepad",
  "app_id": "notepad-4a929eb9",
  "user": {
    "_user_id": {
      "value": "312121",
      "set_timestamp": 1667877566697
    },
    "_user_name": {
      "value": "carl",
      "set_timestamp": 1667877566697
    },
    "_user_first_touch_timestamp": {
      "value": 1667877267895,
      "set_timestamp": 1667877566697
    }
  },
  "attributes": {
    "event_category": "recommended",
    "currency": "CNY",
    "_session_id": "dc7a7a18-20221108-031926703",
    "_session_start_timestamp": 1667877566703,
  }
}
```

```

    "_session_duration": 391809,
    "_session_number": 1,
    "_screen_name": "ProductDetailActivity",
    "_screen_unique_id": "126861252"
  }
}

```

All user attributes will be stored in user object, and all custom and global attributes in attributes object.

Common attribute

Attribute name	Description	It is generated...	It is used to or for...
app_id	clickstream app id	generated when clickstream app create from guidance control plane	identify the events for your apps
unique_id	the unique id for user	generated from <code>UUID.randomUUID().toString()</code> during the SDK first initialization. It will be changed after user relogin to another user who never login, and when user relogin to the previous user in same device. The <code>unique_id</code> will reset to the previous user's <code>unique_id</code> .	identity different users and associate the behavior of logging in and not logging in
device_id	the unique id for device	generated from <code>Settings.System.get</code>	distinguish different devices

Attribute name	Description	It is generated...	It is used to or for...
		<pre>tString(context.getContentResolver(), Settings.Secure.ANDROID_ID)</pre> <p>If Android ID is null or "", we will use UUID instead.</p>	
event_type	event name	set by developer or SDK	distinguish different event types
event_id	the unique id for event	<pre>generated from UUID.randomUUID().toString()</pre> <p>when the event is created</p>	distinguish each event
timestamp	event create timestamp	<pre>generated from System.currentTimeMillis()</pre> <p>when the event is created</p>	data analysis needs
platform	the platform name	for Android device, it is always "Android"	data analysis needs
os_version	the platform version code	<pre>generated from Build.VERSION.RELEASE</pre>	data analysis needs

Attribute name	Description	It is generated...	It is used to or for...
make	the manufacturer of the device	generated from <code>Build.MANUFACTURER</code>	data analysis needs
brand	the brand of the device	generated from <code>Build.BRAND</code>	data analysis needs
model	the model of the device	generated from <code>Build.MODEL</code>	data analysis needs
carrier	the device network operator name	generated from <code>TelephonyManager.getNetworkOperatorName()</code> , the default is "UNKNOWN"	data analysis needs
network_type	the current device network type	generated from <code>android.net.ConnectivityManager</code> , it can be "Mobile", "WIFI" or "UNKNOWN"	data analysis needs
screen_height	the absolute height of the available display size in pixels	generated from <code>applicationContext.getResources().displayMetrics.heightPixels</code>	data analysis needs

Attribute name	Description	It is generated...	It is used to or for...
screen_width	the absolute width of the available display size in pixels	generated from <code>applicationContext.getResources().displayMetrics.widthPixels</code>	data analysis needs
zone_offset	the device raw offset from GMT in milliseconds	generated from <code>java.util.Calendar.get(Calendar.ZONE_OFFSET)</code>	data analysis needs
locale	the default locale (language, country and variant) for this device of the Java Virtual Machine	generated from <code>java.util.Locale.getDefault()</code>	data analysis needs
system_language	the device language code	generated from <code>java.util.Locale.getLanguage()</code> , and its default is value "UNKNOWN"	data analysis needs
country_code	country/region code for this device	generated from <code>java.util.Locale.getCountry()</code> and its default value is "UNKNOWN"	data analysis needs

Attribute name	Description	It is generated...	It is used to or for...
sdk_version	clickstream SDK version	generated from BuildConfig.VERSION_NAME	data analysis needs
sdk_name	clickstream SDK name	it is always "aws-solution-clickstream-sdk"	data analysis needs
app_version	the app version name of user's app	generated from android.content.pm.PackageManager.getVersionName(), and its default value is "UNKNOWN"	data analysis needs
app_package_name	the app package name of user's app	generated from android.content.pm.PackageManager.getPackageName(), and its default value is "UNKNOWN"	data analysis needs
app_title	the display name of user's app	generated from android.content.pm.ApplicationLabel(appInfo)	data analysis needs

User attributes

Attribute name	Description
_user_id	Reserved for user id that is assigned by app
_user_ltv_revenue	Reserved for user lifetime value
_user_ltv_currency	Reserved for user lifetime value currency
_user_first_touch_timestam	The time (in microseconds) when the user first opened the app or visited the site, and it is included in every event in user object

Event attributes

Attribute name	Data type	Auto track	Description
_traffic_source_source	String	false	Reserved for traffic source source. Name of the network source that acquired the user when the event were reported. Example: Google, Facebook, Bing, Baidu.
_traffic_source_medium	String	false	Reserved for traffic medium. Use this attribute to store the medium that acquired user when events were logged.
_traffic_source_campaign	String	false	Reserved for traffic source campaign. Use this attribute to store the campaign of your traffic source.

Attribute name	Data type	Auto track	Description
			Example: summer_sale, holiday_specials
_traffic_source_campaign_id	String	false	Reserved for traffic source campaign id. Use this attribute to store the campaign id of your traffic source. Example: campaign_1, campaign_2.
_traffic_source_term	String	false	Reserved for traffic source term. Use this attribute to store the term of your traffic source. Example: running_shoes, fitness_tracker
_traffic_source_content	String	false	Reserved for traffic source content. Use this attribute to store the content of your traffic source. Example: banner_ad_1, text_ad_2
_traffic_source_clid	String	false	Reserved for traffic source clid. Use this attribute to store the clid of your traffic source. Example: amazon_ad_123, google_ad_456

Attribute name	Data type	Auto track	Description
_traffic_source_clid_platform	String	false	Reserved for traffic source clid platform. Use this attribute to store the clid platform of your traffic source. Example: amazon_ads, google_ads
_app_install_channel	String	false	Reserved for install source, it is the channel for app was downloaded.
_traffic_source_name	String	false	Reserved for traffic name. Use this attribute to store the marketing campaign that acquired user when events were logged.
_session_id	String	true	Added in all events.
_session_start_timestamp	long	true	Added in all events.
_session_duration	long	true	Added in all events.
_session_number	int	true	Added in all events.
_screen_name	String	true	Added in all events.
_screen_unique_id	String	true	Added in all events.

Item attributes

Attribute name	Data type	Auto track	Description
id	String	False	The ID of the item
name	String	False	The name of the item
brand	String	False	The brand of the item
currency	String	False	The currency of the item
price	String	False	The price of the item
quantity	String	False	The quantity of the item
creative_name	String	False	The creative name of the item
creative_slot	String	False	The creative slot of the item
location_id	String	False	The location id of the item
category	String	False	The category of the item
category2	String	False	The category2 of the item
category3	String	False	The category3 of the item
category4	String	False	The category4 of the item
category5	String	False	The category5 of the item

You can use the above preset item attributes, of course, you can also add custom attributes to an item. In addition to the preset attributes, an item can add up to 10 custom attributes.

Change log

[GitHub change log](#).

Sample project

[Sample Android Project for SDK integration](#).

Reference link

[Source code](#)

[Project issues](#)

Swift SDK

Introduction

Clickstream Swift SDK can help you easily collect in-app click stream data from iOS devices to your AWS environments through the data pipeline provisioned by this guidance.

The SDK is based on the Amplify for Swift Core Library and developed according to the Amplify Swift SDK plug-in specification. In addition, the SDK is equipped with features that automatically collect common user events and attributes (for example, screen view, first open) to simplify data collection for users.

Platform Support

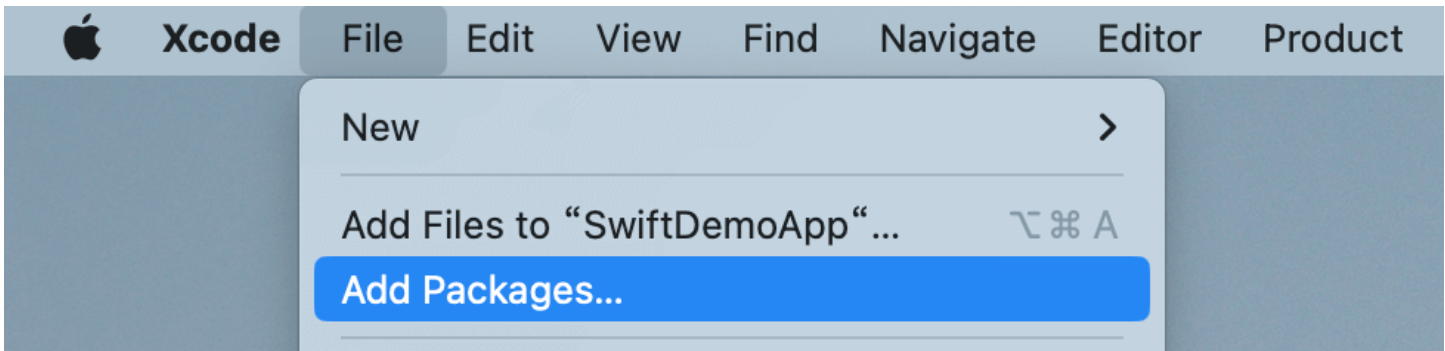
Clickstream Swift SDK supports iOS 13+.

Clickstream Swift SDK requires Xcode 13.4 or higher to build.

Integrate the SDK

1 Add Package

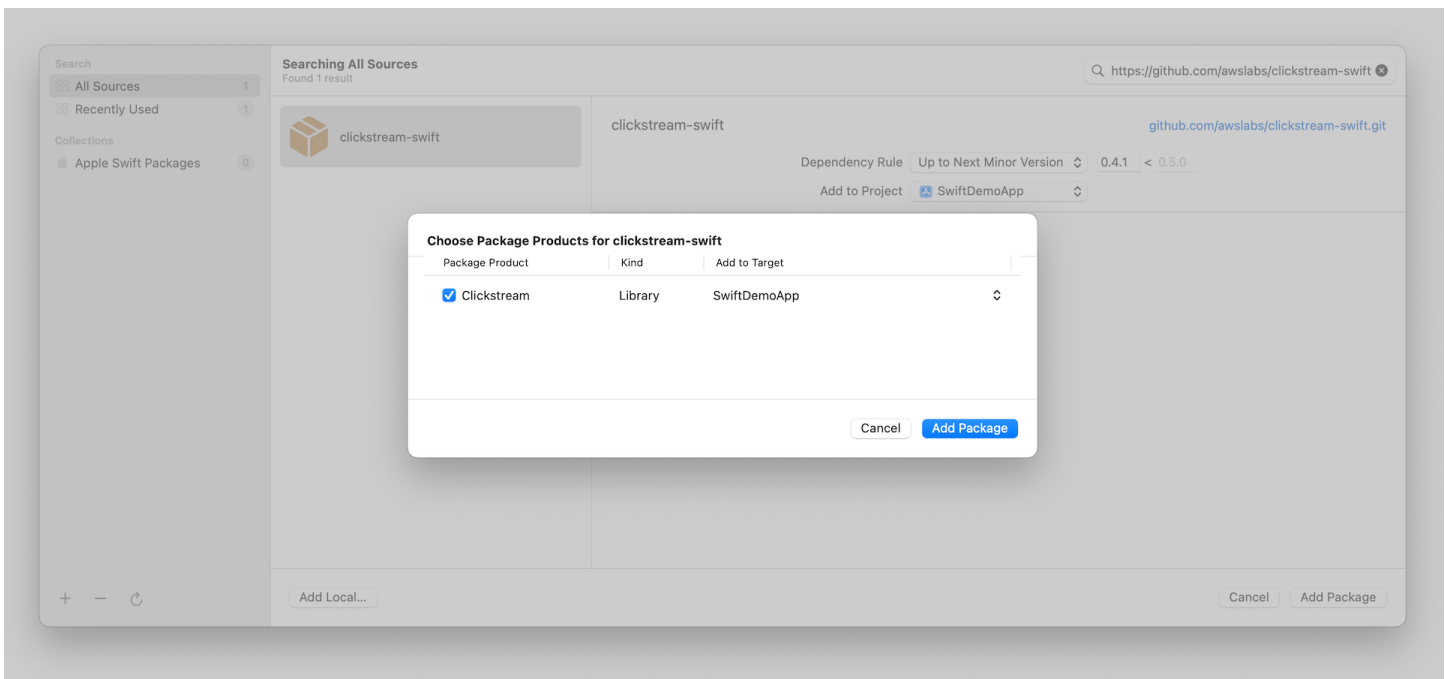
The guidance uses Swift Package Manager to distribute Clickstream Swift SDK. Open your project in Xcode and select **File > Add Packages**.



1. Copy the Clickstream Swift SDK GitHub repository URL and paste it into the search bar.

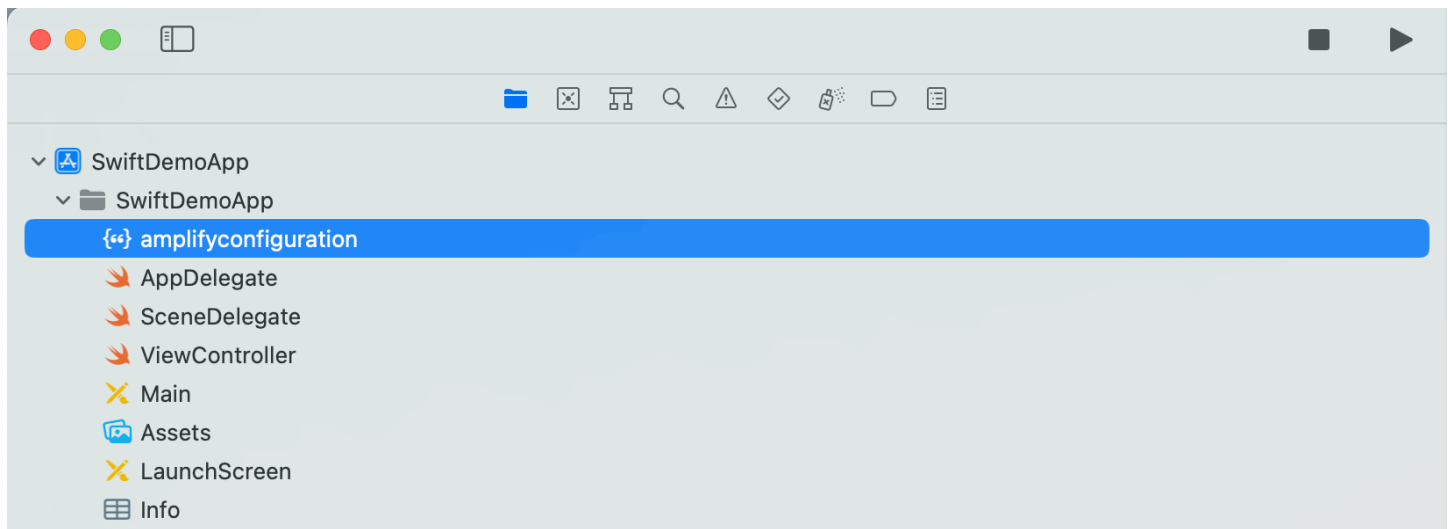
```
https://github.com/awslabs/clickstream-swift
```

2. Check the rules for the version of the SDK that you want Swift Package Manager to install, it is recommended to choose **Up to Next Major Version**, then click **Add Package**.
3. Keep the Clickstream product checked as default.
4. Choose **Add Package** again to finish the package installation.



2 Parameter configuration

Download your `amplifyconfiguration.json` file from your Clickstream guidance control plane. Copy and paste it to your project root folder:



The JSON file will be as follows:

```
{
  "analytics": {
    "plugins": {
      "awsClickstreamPlugin ": {
        "appId": "appId",
        "endpoint": "https://example.com/collect",
        "isCompressEvents": true,
        "autoFlushEventsInterval": 10000,
        "isTrackAppExceptionEvents": false
      }
    }
  }
}
```

In the file, your appId and endpoint are already configured. The explanation for each property is as follows:

- **appId (Required):** the app id of your project in web console.
- **endpoint (Required):** the endpoint url you will upload the event to AWS server.
- **isCompressEvents:** whether to compress event content when uploading events, and the default value is true.
- **autoFlushEventsInterval:** event sending interval, and the default value is 10s.
- **isTrackAppExceptionEvents:** whether auto track exception event in app, and the default value is false .

3 Initialize the SDK

Once you have configured the parameters, you need to initialize it in AppDelegate's `didFinishLaunchingWithOptions` lifecycle method to use the SDK.

Swift

```
import Clickstream
...
func application(_ application: UIApplication, didFinishLaunchingWithOptions
  launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
  // Override point for customization after application launch.
  do {
    try ClickstreamAnalytics.initSDK()
  } catch {
    assertionFailure("Fail to initialize ClickstreamAnalytics: \(error)")
  }
  return true
}
```

Objective-C

```
@import Clickstream;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions {
    NSError *error = nil;
    [ClickstreamObjc initSDKAndReturnError:&error];
    if (error) {
        NSLog(@"Fail to initialize ClickstreamAnalytics: %@",
error.localizedDescription);
    }
    return YES;
}
```

SwiftUI configuration

If your project is developed with SwiftUI, you need to create an application delegate and attach it to your App through `UIApplicationDelegateAdaptor`.

```
@main
struct YourApp: App {
```

```
@UIApplicationDelegateAdaptor(AppDelegate.self) var appDelegate
var body: some Scene {
    WindowGroup {
        YourView()
    }
}
}
```

Clickstream Swift SDK depends on method swizzling to automatically record screen views. SwiftUI apps must record screen view events manually, and disable automatic tracking of screen view events by setting `configuration.isTrackScreenViewEvents(false)` when the SDK is initialized.

5 Start using

Record event

Add the following code where you need to report an event.

Swift

```
import Clickstream

// for record an event with custom attributes
let attributes: ClickstreamAttribute = [
    "category": "shoes",
    "currency": "CNY",
    "value": 279.9
]
ClickstreamAnalytics.recordEvent("button_click", attributes)

// for record an event directly
ClickstreamAnalytics.recordEvent("button_click")
```

Objective-C

```
@import Clickstream;

// for record an event with custom attributes
NSDictionary *attributes =@{
    @"category": @"shoes",
    @"currency": @"CNY",
```

```

    @"value": @12.34
  };
  [ClickstreamObjc recordEvent:@"button_click" :attributes];

```

Add global attribute

1. Add global attributes when initializing the SDK.

The following example code shows how to add traffic source fields as global attributes when initializing the SDK.

Swift

```

import Clickstream

let configuration = ClickstreamConfiguration()
    .withAppId("your appId")
    .withEndpoint("https://example.com/collect")
    .withInitialGlobalAttributes([
        ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_SOURCE: "amazon",
        ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_MEDIUM: "cpc",
        ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_CAMPAGN: "summer_promotion",
        ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_CAMPAGN_ID: "summer_promotion_01",
        ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_TERM: "running_shoes",
        ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_CONTENT: "banner_ad_1",
        ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_CLID: "amazon_ad_123",
        ClickstreamAnalytics.Attr.TRAFFIC_SOURCE_CLID_PLATFORM: "amazon_ads",
        ClickstreamAnalytics.Attr.APP_INSTALL_CHANNEL: "App Store"
    ])
try ClickstreamAnalytics.initSDK(configuration)

```

Objective-C

```

#import Clickstream;

NSDictionary *globalAttributes = @{
    Attr.TRAFFIC_SOURCE_SOURCE: @"amazon",
    Attr.TRAFFIC_SOURCE_MEDIUM: @"cpc",
    Attr.TRAFFIC_SOURCE_CAMPAGN: @"summer_promotion",
    Attr.TRAFFIC_SOURCE_CAMPAGN_ID: @"summer_promotion_01",
    Attr.TRAFFIC_SOURCE_TERM: @"running_shoes",
    Attr.TRAFFIC_SOURCE_CONTENT: @"banner_ad_1",

```

```

Attr.TRAFFIC_SOURCE_CLID: @"amazon_ad_123",
Attr.TRAFFIC_SOURCE_CLID_PLATFORM: @"amazon_ads",
Attr.APP_INSTALL_CHANNEL: @"App Store",
};
ClickstreamConfiguration *configuration = [[[[[ClickstreamConfiguration alloc] init]
withAppId:@"your appId"]
withEndpoint:@"https://example.com/collect"]
withInitialGlobalAttributesObjc:globalAttributes];
[ClickstreamObjc initWithSDK:configuration error: &error];

```

2. Add global attributes after initializing the SDK.

Swift

```

import Clickstream

let globalAttribute: ClickstreamAttribute = [
    ClickstreamAnalytics.Attr.APP_INSTALL_CHANNEL: "App Store",
    "class": 6,
    "level": 5.1,
    "isOpenNotification": true,
]
ClickstreamAnalytics.addGlobalAttributes(globalAttribute)

// for delete an global attribute
ClickstreamAnalytics.deleteGlobalAttributes("level")

```

Objective-C

```

#import Clickstream;

NSDictionary *attributes = @{
    Attr.APP_INSTALL_CHANNEL: @"App Store",
    @"class": @6,
    @"level": @5.1,
    @"isOpenNotification": @YES
};
[ClickstreamObjc addGlobalAttributes :attributes];

// for delete an global attribute
[ClickstreamObjc deleteGlobalAttributes: @[@"level"]];

```

It is recommended to set global attributes when initializing the SDK, global attributes will be included in all events that occur after it is set.

Login and logout

Swift

```
import Clickstream

// when user login usccess.
ClickstreamAnalytics.setUserId("userId")

// when user logout
ClickstreamAnalytics.setUserId(nil)
```

Objective-C

```
@import Clickstream;

// when user login usccess.
[ClickstreamObjc setUserId:@"userId"];

// when user logout
[ClickstreamObjc setUserId:NULL];
```

Add user attribute

Swift

```
import Clickstream

let userAttributes : ClickstreamAttribute=[
    "_user_age": 21,
    "_user_name": "carl"
]
ClickstreamAnalytics.addUserAttributes(userAttributes)
```

Objective-C

```
@import Clickstream;

NSDictionary *userAttributes =@{
```

```
@"_user_age": @21,  
  @"user_name": @"carl"  
};  
[ClickstreamObjc addUserAttributes:userAttributes];  
;
```

Current login user's attributes will be cached in disk, so the next time app launch you don't need to set all user's attribute again, of course you can use the same API `ClickstreamAnalytics.addUserAttributes()` to update the current user's attribute when it changes.

Important

If your application is already published and most users have already logged in, please manually set the user attributes once when integrate the Clickstream SDK for the first time to ensure that subsequent events contains user attributes.

Record event with items

You can add the following code to log an event with an item.

Swift

```
import Clickstream  
  
let attributes: ClickstreamAttribute = [  
    ClickstreamAnalytics.Item.ITEM_ID: "123",  
    ClickstreamAnalytics.Item.CURRENCY: "USD",  
    "event_category": "recommended"  
]  
  
let item_book: ClickstreamAttribute = [  
    ClickstreamAnalytics.Item.ITEM_ID: 123,  
    ClickstreamAnalytics.Item.ITEM_NAME: "Nature",  
    ClickstreamAnalytics.Item.ITEM_CATEGORY: "book",  
    ClickstreamAnalytics.Item.PRICE: 99.9,  
    "book_publisher": "Nature Research"  
]  
ClickstreamAnalytics.recordEvent("view_item", attributes, [item_book])
```

Objective-C

```
@import Clickstream;

NSDictionary *attributes = @{
    Attr.VALUE: @99.9,
    Attr.CURRENCY: @"USD",
    "event_category": @"recommended"
};
NSDictionary *item_book = @{
    ClickstreamItemKey.ITEM_ID: @123,
    ClickstreamItemKey.ITEM_NAME: @"Nature",
    ClickstreamItemKey.ITEM_CATEGORY: @"book",
    ClickstreamItemKey.PRICE: @99.9,
    "book_publisher": @"Nature Research"
};
[ClickstreamObjc recordEvent:@"view_item" :attributes, @[item_book]];
```

For more information about logging more attributes in an item, refer to item attributes.

Important

Only pipelines from version 1.1.0 can handle items with custom attribute.
ITEM_ID is required attribute, if not set the item will be discarded.

Record screen view events manually

By default, SDK will automatically track the preset `_screen_view` event when `ViewController` triggers `viewDidAppear`.

You can manually record screen view events whether automatic screen view tracking is enabled, add the following code to record a screen view event with two attributes.

- **SCREEN_NAME:** Required. Your screen's name.
- **SCREEN_UNIQUE_ID:** Optional. Set the unique value of your `ViewController` or `UIView`. If you do not set, SDK will set a default value based on the current `ViewController`'s `hashValue`.

Swift

```
import Clickstream
```

```
ClickstreamAnalytics.recordEvent(ClickstreamAnalytics.EventName.SCREEN_VIEW, [  
    ClickstreamAnalytics.Attr.SCREEN_NAME: "HomeView",  
    ClickstreamAnalytics.Attr.SCREEN_UNIQUE_ID: "your screen uniqueId"  
])
```

Objective-C

```
@import Clickstream;  
  
NSDictionary *attributes = @{  
    Attr.SCREEN_NAME: @"HomeView",  
    Attr.SCREEN_UNIQUE_ID: @"your screen uniqueId"  
};
```

Send event immediately

Swift

```
import Clickstream  
// for send event immediately. ClickstreamAnalytics.flushEvents()
```

Objective-C

```
@import Clickstream;  
// for send event immediately.  
[ClickstreamObjc flushEvents];
```

Disable SDK

You can disable the SDK if needed. After disabling the SDK, the SDK will not handle the logging and sending of any events. Of course you can enable the SDK when you need to continue logging events.

Swift

```
import Clickstream  
  
// disable SDK ClickstreamAnalytics.disable()  
  
// enable SDK  
ClickstreamAnalytics.enable()
```



```

withLogEvents:TRUE]
withCompressEvents:TRUE]
withSendEventInterval: 10000]
withSessionTimeoutDuration: 1800000]
withTrackScreenViewEvents:TRUE]
withTrackUserEngagementEvents:TRUE]
withTrackAppExceptionEvents:TRUE]
withAuthCookie: @"your auth cookie"]
withInitialGlobalAttributesObjc:@{Attr.TRAFFIC_SOURCE_SOURCE: @"amazon"}];
[ClickstreamObjc initWithSDK:configuration error: &error];

```

Here is an explanation of each option:

Method name	Parameter type	Required	Default value	Description
withAppId()	String	true	--	The app ID of your application in the web console.
withEndpoint()	String	true	--	The endpoint path where you will upload the event to the Clickstream ingestion server.
withLogEvents()	Bool	false	false	Whether to automatically print an event JSON for debugging events .
withCompressEvents()	Bool	false	true	Whether to compress event content by gzip when uploading events.

Method name	Parameter type	Required	Default value	Description
withSendEventsInterval()	Int	false	10,000	Event sending interval in milliseconds.
withSessionTimeoutDuration()	Int64	false	1,800,000	The duration of the session timeout in milliseconds.
withTrackScreenViewEvents()	Bool	false	true	Whether to auto-record screen view events.
withTrackUserEngagementEvents()	Bool	false	true	Whether to auto-record user engagement events.
withTrackAppExceptionEvents()	Bool	false	true	Whether to auto-record app exception events.
withAuthCookie()	String	false	--	Your auth cookie for AWS application load balancer auth cookie.

Configuration update

After initializing the SDK, you can use the following code to customize the configuration of the SDK.

Swift

```
import Clickstream

// config the sdk after initialize.
do {
    var configuration = try ClickstreamAnalytics.getClickstreamConfiguration()
    configuration.withAppId("your appId")
                  .withEndpoint("https://example.com/collect")
                  .withLogEvents(true)
                  .withCompressEvents(true)
                  .withTrackAppExceptionEvents(true)
                  .withTrackScreenViewEvents(true)
                  .withTrackUserEngagementEvents(true)
                  .withAuthCookie("your authentication cookie")
} catch {
    print("Failed to config ClickstreamAnalytics: \(error)")
}
```

Objective-C

```
@import Clickstream;

// config the sdk after initialize.
ClickstreamConfiguration *configuration = [ClickstreamObjc
    getClickstreamConfigurationAndReturnError:&error];
configuration = [configuration withAppId:@"your appId"]
    withEndpoint:@"https://example.com/collect"]
    withLogEvents:TRUE]
    withCompressEvents:TRUE]
    withTrackScreenViewEvents:TRUE]
    withTrackUserEngagementEvents:TRUE]
    withTrackAppExceptionEvents:TRUE];
```

Debug events

You can follow the steps below to view the event raw JSON and debug your events.

1. Set the `isLogEvents` option when `with true` in debug mode.
2. Integrate the SDK and launch your app by Xcode, then open the log panel.
3. Input **EventRecorder** to the filter, and you will see the JSON content of all events recorded by Clickstream Swift SDK.

Data format definition

Data type

Clickstream Swift SDK supports the following data types:

Data type	Range	Example
Int	-214748364 ~ 2147483647	12
Int64	-9223372036854775808 ~ 9223372036854775807	26854775808
Double	-2.22E-308 ~ 1.79E+308	3.14
Boolean	true, false	true
String	1024 characters maximum	"Clickstream"

Naming rules

1. The event name and attribute name cannot start with a number, and only contain uppercase and lowercase letters, numbers, and underscores. In case of an invalid event name, it will throw `precondition failure`, In case of an invalid attribute name or user attribute name, it will discard the attribute and record error.
2. Do not use `_` as prefix in an event name or attribute name, because the `_` prefix is reserved for the guidance.
3. The event name and attribute name are case sensitive, so `Add_to_cart` and `add_to_cart` will be recognized as two different event names.

Event and attribute limitation

To improve the efficiency of querying and analysis, we apply limitations to event data as follows:

Item	Recommended	Maximum (hard limit)	Strategy	Error code
Event name invalid	N/A	N/A	discard event, print log and record_clickstream_error event	1001
Length of event name	less than 25 characters	50 characters	discard event, print log and record_clickstream_error event	1002
Length of event attribute name	less than 25 characters	50 characters	discard the attribute, print log and record error in event attribute	2001
Attribute name invalid	N/A	N/A	discard the attribute, print log and record error in event attribute	2002
Length of event attribute value	less than 100 characters	1024 characters	discard the attribute, print log and record error in event attribute	2003
Event attribute per event	less than 50 attributes	500 event attributes	discard the attribute, print log and record	2004

Item	Recommended	Maximum (hard limit)	Strategy	Error code
			error in event attribute	
User attribute number	less than 25 attributes	100 user attributes	discard event, print log and record_clickstream_error event	3001
Length of user attribute name	less than 25 characters	50 characters	discard event, print log and record_clickstream_error event	3002
User attribute name invalid	N/A	N/A	discard event, print log and record_clickstream_error event	3003
Length of user attribute value	less than 50 characters	256 characters	discard event, print log and record_clickstream_error event	3004
Item number in one event	less than 50 items	100 items	discard the item, print log and record error in event attribute	4001

Item	Recommended	Maximum (hard limit)	Strategy	Error code
Length of item attribute value	less than 100 characters	256 characters	discard the item, print log and record error in event attribute	4002
Custom item attribute number in one item	less than 10 custom attributes	10 custom attributes	discard the item, print log and record error in event attribute	4003
Length of item attribute name	less than 25 characters	50 characters	discard the item, print log and record error in event attribute	4004
Item attribute name invalid	N/A	N/A	discard the item, print log and record error in event attribute	4005

Rules:

- The character limits are the same for single-width character languages (for example, English) and double-width character languages (for example, Chinese).
- The limitation of event attribute per event involves both common attributes and preset attributes.
- If the attribute or user attribute with the same name is added more than twice, the latest value will apply.
- All errors that exceed the limit will be recorded `_error_code` and `_error_message` in the event attributes.

Preset events

Automatically collected events

Event name	Triggered	Event attributes
_first_open	when the user launches an app the first time after installation	N/A
_session_start	when a user first opens the app or a user returns to the app after 30 minutes of inactivity period	_session_id _session_start_timestamp
_screen_view	when a new screen opens	_screen_name _screen_id _screen_unique_id _previous_screen_name _previous_screen_id _previous_screen_unique_id _entrances _previous_timestamp _engagement_time_msec
_user_engagement	when user navigates away from current screen and the screen is in focus for at least one second	_engagement_time_msec
_app_start	every time the app goes to visible	_is_first_time (when it is the first _app_start event after

Event name	Triggered	Event attributes
		the application starts, the value is true)
<code>_app_end</code>	every time the app goes to invisible	
<code>_profile_set</code>	when the <code>addUserAttributes()</code> or <code>setUserId()</code> API is called	
<code>_app_exception</code>	when the app crashes	<code>_exception_message</code> <code>_exception_stack</code>
<code>_app_update</code>	when the app is updated to a new version and launched again	<code>_previous_app_version</code>
<code>_os_update</code>	when device operating system is updated to a new version	<code>_previous_os_version</code>
<code>_clickstream_error</code>	<code>event_name</code> is invalid or user attribute is invalid	<code>_error_code</code> <code>_error_message</code>

Session definition

In Clickstream Swift SDK, we do not limit the total time of a session. As long as the time between the next entry of the app and the last exit time is within the allowable timeout period, the current session is considered to be continuous.

The **`_session_start`** event triggered when the app open for the first time, or the app was open to the foreground and the time between the last exit exceeded **`session_time_out`** period. The following are session-related attributes.

- **`_session_id`**: It is calculated by concatenating the last 8 characters of `uniqueId` and the current millisecond, for example: `dc7a7a18-20230905-131926703`.

- **_session_duration:** We calculate the `_session_duration` by minus the current event create timestamp and the session's `_session_start_timestamp`. This attribute will be added in every event during the session.
- **_session_number:** The auto increment number of session in current device, and the initial value is 1.
- **Session timeout duration:** By default, it is 30 minutes, which can be customized through the configuration update API.

Screen view definition

In Clickstream Swift SDK, we define the `_screen_view` as an event that records a user's browsing path of screen. When a screen transition started, the `_screen_view` event will be recorded if any of the following conditions is met:

- No screen was previously set.
- The new screen name differs from the previous screen title.
- The new screen id differs from the previous screen id.
- The new screen unique id differs from the previous screen unique id.

This event listens for `UIViewController`'s `onViewDidAppear` lifecycle method to determine the screen transition.. In order to track screen browsing path, we use `_previous_screen_name` , `_previous_screen_id` and `_previous_screen_unique_id` to link the previous screen. In addition, there are some other attributes in screen view event.

- `_screen_unique_id:` We calculate the screen unique id by getting the current screen's hashcode, for example: "5260751568".
- `_entrances:` The first screen view event in a session is 1, others is 0.
- `_previous_timestamp:` The timestamp of the previous `_screen_view` event.
- `_engagement_time_msec:` The previous page last engagement milliseconds.

When the app goes to the background for more than 30 minutes and then opened again, a new session will be generated, the previous screen information will be cleared, and a new screen view event will be sent.

User engagement definition

In Clickstream Swift SDK, we define the `user_engagement` as an event that records the screen browsing time, and we only send this event when user leave the screen and the screen has focus for at least one second.

We define that users leave the screen in the following situations.

- When the user navigates to another screen.
- The user moves the app screen to the background.
- The user exit the app, or kill the process of app.

engagement_time_msec: We calculate the milliseconds from when a screen is visible to when the user leave the screen.

Event attributes

Sample event structure

```
{
  "app_id": "Shopping",
  "app_package_name": "com.company.app",
  "app_title": "Shopping",
  "app_version": "1.0",
  "brand": "apple",
  "carrier": "UNKNOWN",
  "country_code": "US",
  "device_id": "A536A563-65BD-49BE-A6EC-6F3CE7AC8FBE",
  "device_unique_id": "",
  "event_id": "91DA4BBE-933F-4DFA-A489-8AEFBC7A06D8",
  "event_type": "add_to_cart",
  "hashCode": "63D7991D",
  "locale": "en_US",
  "make": "apple",
  "model": "iPhone 14 Pro",
  "network_type": "WIFI",
  "os_version": "16.4",
  "platform": "iOS",
  "screen_height": 2556,
  "screen_width": 1179,
  "sdk_name": "aws-solution-clickstream-sdk",
```

```

"sdk_version": "0.4.1",
"system_language": "en",
"timestamp": 1685082174195,
"unique_id": "0E6614B7-2D2C-4774-AB2F-B0A9E6C3BFAC",
"zone_offset": 28800000,
"user": {
  "_user_city": {
    "set_timestamp": 1685006678437,
    "value": "Shanghai"
  },
  "_user_first_touch_timestamp": {
    "set_timestamp": 1685006678434,
    "value": 1685006678432
  },
  "_user_name": {
    "set_timestamp": 1685006678437,
    "value": "carl"
  }
},
"attributes": {
  "event_category": "recommended",
  "currency": "CNY",
  "_session_duration": 15349,
  "_session_id": "0E6614B7-20230526-062238846",
  "_session_number": 3,
  "_session_start_timestamp": 1685082158847,
  "_screen_name": "ProductDetailViewController",
  "_screen_unique_id": "5260751568"
}
}

```

All user attributes will be stored in user object, and all custom and global attributes in attributes object.

Common attribute

Attribute name	Description	It is generated ...	It is used to or for ...
hashCode	the AnalyticsEvent Object's hash code	generated from String(format: "%08X", hasher.co	distinguish different events

Attribute name	Description	It is generated ...	It is used to or for ...
		mbine(eventjson))	
app_id	clickstream app id	generated when clickstream app create from guidance control plane	identify the events for your apps
unique_id	the unique id for user	generated from UUID().uuidString during the SDK first initialization. It will be changed after user relogin to another user who never login, and when user relogin to the previous user in same device. The unique_id will reset to the previous user's unique_id .	identity different users and associate the behavior of logging in and not logging in

Attribute name	Description	It is generated ...	It is used to or for ...
device_id	the unique id for device	generated from <code>UIDevice.current.identifierForVendor?.uuidString ?? UUID().uuidString</code> . It will be changed after app is reinstalled.	distinguish different devices
device_unique_id	the device advertising id	generated from <code>ASIdentifierManager.shared().advertisingIdentifier.uuidString ?? ""</code>	distinguish different devices
event_type	event name	set by user or SDK	distinguish different event types
event_id	the unique id for event	generated from <code>UUID().uuidString</code> when the event is created	distinguish each event

Attribute name	Description	It is generated ...	It is used to or for ...
timestamp	event create timestamp	generated from <code>Date().timeIntervalSinceSince1970 * 1000</code> when the event is created	data analysis needs
platform	the platform name	for iOS device, it is always "iOS"	data analysis needs
os_version	the iOS operating system version	generated from <code>UIDevice.current.systemVersion</code>	data analysis needs
make	the manufacturer of the device	for iOS device, it is always "apple"	data analysis needs
brand	the brand of the device	for iOS device, it is always "apple"	data analysis needs
model	the model of the device	generated from mapping of device identifier	data analysis needs
carrier	the device network operator name	generated from <code>CTTelephonyNetworkInfo().serviceSubscriberCellularProviders?.first?.value</code> , and its default value is "UNKNOWN"	data analysis needs

Attribute name	Description	It is generated ...	It is used to or for ...
network_type	the current device network type	generated from <code>NWPathMonitor</code> , it can be "Mobile", "WIFI" or "UNKNOWN"	data analysis needs
screen_height	the absolute height of the available display size in pixels	generated from <code>UIScreen.main.bounds.size.height * UIScreen.main.scale</code>	data analysis needs
screen_width	the absolute width of the available display size in pixels	generated from <code>UIScreen.main.bounds.size.width * UIScreen.main.scale</code>	data analysis needs
zone_offset	the device raw offset from GMT in milliseconds	generated from <code>TimeZone.current.secondsFromGMT() * 1000</code>	data analysis needs
locale	the default locale (language, country and variant) for this device of the Java Virtual Machine	generated from <code>Locale.current</code>	data analysis needs

Attribute name	Description	It is generated ...	It is used to or for ...
system_language	the device language code	generated from <code>Locale.current.languageCode</code> , and its default is value "UNKNOWN"	data analysis needs
country_code	country/region code for this device	generated from <code>Locale.current.regionCode</code> and its default value is "UNKNOWN"	data analysis needs
sdk_version	clickstream SDK version	generated from <code>PackageInfo.version</code>	data analysis needs
sdk_name	clickstream SDK name	it is always "aws-solution-clickstream-sdk"	data analysis needs
app_version	the app version name of user's app	generated from <code>Bundle.main.infoDictionary["CFBundleShortVersionString"] ?? ""</code>	data analysis needs

Attribute name	Description	It is generated ...	It is used to or for ...
app_package_name	the app package name of user's app	generated from Bundle.main.infoDictionary["CFBundleIdentifier"] ?? ""	data analysis needs
app_title	the display name of user's app	generated from Bundle.main.infoDictionary["CFBundleName"] ?? ""	data analysis needs

User attributes

Attribute name	Description
_user_id	Reserved for user id that is assigned by app
_user_ltv_revenue	Reserved for user lifetime value
_user_ltv_currency	Reserved for user lifetime value currency
_user_first_touch_timestamp	The time (in microseconds) when the user first opened the app or visited the site, and it is included in every event in user object

Event attributes

Attribute name	Data type	Auto track	Description
_traffic_source_source	String	false	Reserved for traffic source source. Name of the network source that acquired the user when the event were reported. Example: Google, Facebook, Bing, Baidu.
_traffic_source_medium	String	false	Reserved for traffic medium. Use this attribute to store the medium that acquired user when events were logged.
_traffic_source_campaign	String	false	Reserved for traffic source campaign. Use this attribute to store the campaign of your traffic source. Example: summer_sale, holiday_specials
_traffic_source_campaign_id	String	false	Reserved for traffic source campaign id. Use this attribute to store the campaign id of your traffic source. Example: campaign_1, campaign_2.
_traffic_source_term	String	false	Reserved for traffic source term. Use this attribute to

Attribute name	Data type	Auto track	Description
			store the term of your traffic source. Example: running_shoes, fitness_tracker
_traffic_source_content	String	false	Reserved for traffic source content. Use this attribute to store the content of your traffic source. Example: banner_ad_1, text_ad_2
_traffic_source_clid	String	false	Reserved for traffic source clid. Use this attribute to store the clid of your traffic source. Example: amazon_ad_123, google_ad_456
_traffic_source_clid_platform	String	false	Reserved for traffic source clid platform. Use this attribute to store the clid platform of your traffic source. Example: amazon_ads, google_ads
_app_install_channel	String	false	Reserved for install source, it is the channel for app was downloaded.

Attribute name	Data type	Auto track	Description
_traffic_source_name	String	false	Reserved for traffic name. Use this attribute to store the marketing campaign that acquired user when events were logged.
_traffic_source_source	String	false	Reserved for traffic source. Name of the network source that acquired the user when the event were reported.
_channel	String	false	The channel for app was downloaded
_session_id	String	true	Added in all events.
_session_start_timestamp	long	true	Added in all events.
_session_duration	long	true	Added in all events.
_session_number	int	true	Added in all events.
_screen_name	String	true	Added in all events.
_screen_unique_id	String	true	Added in all events.

Item attributes

Attribute name	Data type	Required	Description
id	string	False	The id of the item

Attribute name	Data type	Required	Description
name	string	False	The name of the item
brand	string	False	The brand of the item
currency	string	False	The currency of the item
price	number	False	The price of the item
quantity	string	False	The quantity of the item
creative_name	string	False	The creative name of the item
creative_slot	string	False	The creative slot of the item
location_id	string	False	The location id of the item
category	string	False	The category of the item
category2	string	False	The category2 of the item
category3	string	False	The category3 of the item
category4	string	False	The category4 of the item
category5	string	False	The category5 of the item

You can use the listed preset item attributes, or you can add custom attributes to an item. In addition to the preset attributes, an item can add up to 10 custom attributes.

Change log

For more information, see [change logs on GitHub](#).

Sample project

[Sample iOS Project for SDK integration](#)

Reference link

[Source code](#)

[Project issue](#)

[API Documentation](#)

[ClickstreamObjc Api Reference](#)

Web SDK

Introduction

The Clickstream Web SDK can help you easily collect click stream data from browser to your AWS environments through the data pipeline provisioned by this solution.

The SDK is based on the amplify-js SDK core library and developed according to the amplify-js SDK plug-in specification. In addition, the SDK is equipped with features that automatically collect common user events and attributes (for example, page view and first open) to simplify data collection for users.

Integrate the SDK

Using NPM

1. Include SDK

```
npm install @aws/clickstream-web
```

2. Initialize the SDK

You need to configure the SDK with default information before using it. To do this, firstly copy your initial code from your guidance web console. The initial code is like:

```
import { ClickstreamAnalytics } from '@aws/clickstream-web';

ClickstreamAnalytics.init({
  appId: "your appId",
  endpoint: "https://example.com/collect",
});
```

Then, add the code to your your app's root entry point, for example `index.js/app.tsx` in React or `main.ts` in Vue/Angular.

In the code, `appId` and `endpoint` are already set up. Alternatively, you can manually add this code snippet and replace the values of `appId` and `endpoint` after you registered app to a data pipeline in the guidance web console.

Using JS File

1. Download the `clickstream-web.min.js` from the assets in [GitHub Release](#) page, and then copy it into your project.
2. Add the following initial code into your `index.html`.

```
<script src="path to your clickstream-web.min.js"></script>
<script>
  window.ClickstreamAnalytics.init({
    appId: 'your appId',
    endpoint: 'https://example.com/collect',
  })
</script>
```

You can find the `appId` and `endpoint` in the application detail page of the guidance web console.

To lazy load the SDK, use the `async` attribute and place the `ClickstreamAnalytics.init()` method after `window.onload` or `DOMContentLoaded`.

Start using

Record event

Add the following code where you need to record event.

```
import { ClickstreamAnalytics } from '@aws/clickstream-web';
// record event with attributes
ClickstreamAnalytics.record({
  name: 'button_click',
  attributes: {
    category: 'shoes',
    currency: 'CNY',
    value: 279.9,
  }
});

//record event with name
ClickstreamAnalytics.record({ name: 'buttonClick' });
```

Add global attribute

- Add global attributes when initializing the SDK. The following example code shows how to add traffic source fields as global attributes when initializing the SDK.

```
import { ClickstreamAnalytics, Attr } from '@aws/clickstream-web';

ClickstreamAnalytics.init({
  appId: "your appId",
  endpoint: "https://example.com/collect",
  globalAttributes:{
    [Attr.TRAFFIC_SOURCE_SOURCE]: 'amazon',
    [Attr.TRAFFIC_SOURCE_MEDIUM]: 'cpc',
    [Attr.TRAFFIC_SOURCE_CAMPAGN]: 'summer_promotion',
    [Attr.TRAFFIC_SOURCE_CAMPAGN_ID]: 'summer_promotion_01',
    [Attr.TRAFFIC_SOURCE_TERM]: 'running_shoes',
    [Attr.TRAFFIC_SOURCE_CONTENT]: 'banner_ad_1',
    [Attr.TRAFFIC_SOURCE_CLID]: 'amazon_ad_123',
    [Attr.TRAFFIC_SOURCE_CLID_PLATFORM]: 'amazon_ads',
  }
});
```

- Add global attributes after initializing the SDK.

```
ClickstreamAnalytics.setGlobalAttributes({
  _traffic_source_medium: "Search engine",
  level: 10,
});
```

It is recommended to set global attributes when initializing the SDK, and global attributes will be included in all events that occur after it is set. You also can remove a global attribute by setting its value to null.

Login and logout

```
import { ClickstreamAnalytics } from '@aws/clickstream-web';

// when user login success.
ClickstreamAnalytics.setUserId("1234");

// when user logout
ClickstreamAnalytics.setUserId(null);
```

Add user attribute

```
ClickstreamAnalytics.setUserAttributes({
  userName: "carl",
  userAge: 22
});
```

Current login user's attributes will be cached in localStorage, so the next time the browser opens you don't need to set up all user's attributes again. You can also use the same api `ClickstreamAnalytics.setUserAttributes()` to update the current user's attributes in case of any changes.

Important

If your application is already published and most users have already logged in, please manually set the user attributes once when integrating the Clickstream SDK for the first time to ensure that subsequent events contain user attributes.

Record event with items

You can add the following code to log an event with an item.

```
import { ClickstreamAnalytics, Item, Attr } from '@aws/clickstream-web';

const itemBook: Item = {
  id: '123',
  name: 'Nature',
  category: 'book',
  price: 99,
  book_publisher: 'Nature Research',
};

ClickstreamAnalytics.record({
  name: 'view_item',
  attributes: {
    [Attr.CURRENCY]: 'USD',
    [Attr.VALUE]: 99,
    event_category: 'recommended',
  },
  items: [itemBook],
});
```

For more information about logging more attributes in an item, refer to item attributes.

Important

Only pipelines from version 1.1.0 can handle items with custom attribute. **ITEM_ID** is required attribute, if not set the item will be discarded.

Send event immediate in batch mode

In batch mode, you can still send an event immediately by setting the **isImmediate** attribute to true, as shown in the following code.

```
import { ClickstreamAnalytics } from '@aws/clickstream-web';

ClickstreamAnalytics.record({
  name: 'button_click',
  isImmediate: true,
```

```
});
```

Other configurations

In addition to the required `appId` and `endpoint`, you can configure other information for customization purposes:

```
import { ClickstreamAnalytics, SendMode, PageType } from '@aws/clickstream-web';

ClickstreamAnalytics.init({
  appId: "your appId",
  endpoint: "https://example.com/collect",
  sendMode: SendMode.Batch,
  sendEventsInterval: 5000,
  isTrackPageViewEvents: true,
  isTrackUserEngagementEvents: true,
  isTrackClickEvents: true,
  isTrackSearchEvents: true,
  isTrackScrollEvents: true,
  isTrackPageLoadEvents: true,
  isTrackAppStartEvents: true,
  isTrackAppEndEvents: true,
  pageType: PageType.SPA,
  isLogEvents: false,
  authCookie: "your auth cookie",
  sessionTimeoutDuration: 1800000,
  idleTimeoutDuration: 120000,
  searchKeyWords: ['product', 'class'],
  domainList: ['example1.com', 'example2.com'],
});
```

Each option is explained below:

Name	Required	Default value	Description
<code>appId</code>	true	N/A	the app id of your application in the web console
<code>endpoint</code>	true	N/A	the endpoint path where you will

Name	Required	Default value	Description
			upload the event to Clickstream ingestion server
sendMode	false	Immediate	there are two ways to send events: Immediate and Batch
sendEventsInterval	false	5,000	event sending interval in milliseconds, only works in Batch mode
isTrackPageViewEvents	false	true	whether to auto record page view events in the browser
isTrackUserEngagementEvents	false	true	whether to auto record user engagement events in the browser
isTrackClickEvents	false	true	whether to auto record link click events in the browser
isTrackSearchEvents	false	true	whether to auto record search result page events in the browser
isTrackScrollEvents	false	true	whether to auto record page scroll events in the browser

Name	Required	Default value	Description
pageType	false	SPA	the website type: SPA for single page application, and multiPage App for multiple page application. This attribute works only when the value of attribute <code>isTrackPageViewEvents</code> is true.
isLogEvents	false	false	whether to print out event json in the web console for debugging
authCookie	false	--	your auth cookie for AWS application load balancer auth cookie
sessionTimeoutDuration	false	1,800,000	the duration for session timeout in milliseconds
searchKeyWords	false	--	the customized keywords to trigger the <code>_search</code> event. By default, it supports <code>q</code> , <code>s</code> , <code>search</code> , <code>query</code> and <code>keyword</code> query parameters.

Name	Required	Default value	Description
domainList	false	--	the domain list can be configured if a website crosses multiple domains. The <code>_outbound</code> attribute of the <code>_click</code> event will be true when a link leads to a website that's not a part of your configured domain.

Configuration update

You can update the default configuration after initializing the SDK. The following are additional configuration options that you can customize.

```
import { ClickstreamAnalytics } from '@aws/clickstream-web';

ClickstreamAnalytics.updateConfigure({
  isLogEvents: true,
  authCookie: 'your auth cookie',
  isTrackPageViewEvents: false,
  isTrackUserEngagementEvents: false,
  isTrackClickEvents: false,
  isTrackScrollEvents: false,
  isTrackSearchEvents: false,
});
```

Debug events

You can follow the steps below to view the event raw json and debug your events.

1. Use `ClickstreamAnalytics.init()` API to set the `isLogEvents` attribute to true in debug mode.

2. Integrate the SDK and launch your web application in a browser, and then open the Inspection page and switch to console tab.
3. Enter EventRecorder to Filter, and you will see the JSON content of all events recorded by Clickstream Web SDK.

Data format definition

Data type

Clickstream Web SDK supports the following data types:

Data type	Range	Example
number	5e-324~1.79e+308	12, 26854775808, 3.14
boolean	true false	true
string	max 1024 characters	"Clickstream"

Naming rules

1. The event name and attribute name cannot start with a number, and only contains uppercase and lowercase letters, numbers, and underscores. If the event name is invalid, the SDK will record `_clickstream_error` event; if the attribute or user attribute name is invalid, the attribute will be discarded and the SDK also records `_clickstream_error` event.
2. Do not use `_` as prefix in an event name or attribute name, because the `_` prefix is reserved for the guidance.
3. The event name and attribute name are case sensitive, so `Add_to_cart` and `add_to_cart` will be recognized as two different event names.

Event and attribute limitation

In order to improve the efficiency of querying and analysis, we apply limits to event data as follows:

Error code	Name	Suggestion	Hard limit	Strategy
1001	Invalid event name	N/A	N/A	discard event, print log and record <code>_clickstream_error</code> event
1002	Length of event name	Less than 25 characters	50 characters	discard event, print log and record <code>_clickstream_error</code> event
2001	Length of event attribute name	Less than 25 characters	50 characters	discard the attribute, print log and record error in event attribute
2002	Attribute name invalid	N/A	N/A	discard the attribute, print log and record error in event attribute
2003	Length of event attribute value	Less than 100 characters	1024 characters	discard the attribute, print log and record error in event attribute
2004	Event attribute per event	Less than 50 attributes	500 event attributes	discard the attribute that exceed, print log and record error in event attribute

Error code	Name	Suggestion	Hard limit	Strategy
3001	User attribute number	Less than 25 attributes	100 user attributes	discard the attribute that exceed, print log and record <code>_clickstream_error</code> event
3002	Length of user attribute name	Less than 25 characters	50 characters	discard the attribute, print log and record <code>_clickstream_error</code> event
3003	User attribute name invalid	N/A	N/A	discard the attribute, print log and record <code>_clickstream_error</code> event
3004	Length of User attribute value	Less than 50 characters	256 characters	discard the attribute, print log and record <code>_clickstream_error</code> event
4001	Item number in one event	Less than 50 items	100 items	discard the item, print log and record error in event attribute
4002	Length of item attribute value	Less than 100 characters	256 characters	discard the item, print log and record error in event attribute

Error code	Name	Suggestion	Hard limit	Strategy
4003	Custom item attribute number in one item	Less than 10 custom attributes	10 custom attributes	discard the item, print log and record error in event attribute
4004	Length of item attribute name	Less than 25 characters	50 characters	discard the item, print log and record error in event attribute
4005	Item attribute name invalid	N/A	N/A	discard the item, print log and record error in event attribute

Important

- The character limits are the same for single-width character languages (for example, English) and double-width character languages (for example, Chinese).
- The limit of event attribute per event includes common attributes and preset attributes.
- If the attribute or user attribute with the same name is added more than twice, the latest value will apply.
- All errors that exceed the limit will be recorded `_error_code` and `_error_message` these two attribute in the event attributes.

Preset events

Automatically collected events

Event name	Triggered	Event Attributes
_first_open	the first time a user launches the site in a browser	
_session_start	when a user first visits the site or a user returns to the website after 30 minutes of inactivity period, Learn more	<ol style="list-style-type: none"> 1._session_id 2._session_start_timestamp
_page_view	when new page is opens, Learn more	<ol style="list-style-type: none"> 1._page_referrer 2._page_referrer_title 3._entrances 4._previous_timestamp 5. _engagement_time_msec
_user_engagement	when user navigates away from current webpage and the page is in focus for at least one second, Learn more	<ol style="list-style-type: none"> 1._engagement_time_msec
_app_start	every time the browser goes to visible	<ol style="list-style-type: none"> 1. _is_first_time(when it is the first _app_start event after the application starts, the value is true)
_app_end	every time the browser goes to invisible	
_profile_set	when the addUserAttributes() or setUserId() api called	

Event name	Triggered	Event Attributes
_scroll	the first time a user reaches the bottom of each page (that is, when a 90% vertical depth becomes visible)	_engagement_time_msec
_search	each time a user performs a site search, indicated by the presence of a URL query parameter, by default we detect q, s, search, query and ke query parameters	_search_key (the keyword name) _search_term (the search content)
_click	each time a user clicks a link that leads away from the current domain (or configured domain list)	<ol style="list-style-type: none"> 1._link_classes(the content of class in tag <a>) 2._link_domain (the domain of href in tag <a>) 3._link_id (the content of id in tag <a>) 4._link_url (the content of href in tag <a>) 5._outbound (if the domain is not in configured domain list, the attribute value is true)

Event name	Triggered	Event Attributes
_page_load	each time a new page loaded, and browser PerformanceObserver triggered navigation event.	<ol style="list-style-type: none"> 1. duration 2. deliveryType 3. nextHopProtocol 4. renderBlockingStatus 5. startTime 6. redirectStart 7. redirectEnd 8. workerStart 9. fetchStart 10domainLookupStart 11domainLookupEnd 12connectStart 13secureConnectionStart 14connectEnd 15requestStart 16firstInterimResponseStart 17responseStart 18responseEnd 19ransferSize 20encodedBodySize 21decodedBodySize 22responseStatus 23unloadEventStart 24unloadEventEnd 25domInteractive 26domContentLoadedEventStart 27domContentLoadedEventEnd

Event name	Triggered	Event Attributes
		28domComplete 29loadEventStart 30loadEventEnd 31type 32edirectCount 33activationStart 34criticalCHRestart 35serverTiming For more detail you can refer PerformanceNavigationTiming and toJsonMethod
_clickstream_error	event_name is invalid or user attribute is invalid	1. _error_code 2. _error_message

Session definition

In Clickstream Web SDK, there is no limit to the total time of a session. As long as the time between the next entry of the browser and the last exit time is within the allowable timeout period, the current session is considered to be continuous.

The `_session_start` event is initiated when the website opens for the first time, or the browser opens to the foreground and the time between the last exit exceeded `session_time_out` period, and the following are session-related attributes.

- `_session_id`: We calculate the session id by concatenating the last 8 characters of `uniqueid` and the current millisecond, for example, `dc7a7a18-20230905-131926703`.
- `_session_duration` : We calculate the session duration by minus the current event create timestamp and the session's `_session_start_timestamp`, this attribute will be added in every event during the session.

- `_session_number` : It indicates the auto increment number of session in current browser, and has an initial value 1.
- Session timeout duration: By default, it is 30 minutes, which can be customized through the [configuration](#) API.

Page view definition

In Clickstream Web SDK, the `_page_view` refers to an event that records a user's browsing path of page. When a page transition started, the `_page_view` event will be recorded if any of the following conditions is met:

- No page was previously set.
- The new page title differs from the previous page title.
- The new page URL differs from the previous page URL.

This event listens for `pushState`, `popState` in history, and `replaceState` of window to determine the page transition. In order to track page browsing path, we use `_page_referrer` (last page URL) and `page_referrer_title` to link the previous page. There are some other attributes in page view event.

- `_entrances`: It is 1 for the first page view event in a session. Otherwise, it is 0.
- `_previous_timestamp`: The timestamp of the previous `_page_view` event.
- `_engagement_time_msec`: The previous page last engagement in milliseconds.

When the page goes to invisible for more than 30 minutes and then is opened again, a new session will be generated, the previous page URL will be cleared, and a new page view event will be sent.

User engagement definition

In Clickstream Web SDK, the `_user_engagement` refers to an event that records the page browsing time. This event is sent only when a user leaves the page and the page has focus for at least one second.

We define that users leave the page in the following situations.

- When the user navigates to another page under the current domain.

- When the user clicks a link that leads away from the current domain.
- When the user clicks another browser tab or minimizes the current browser window.
- When the user closes the website tab or closes the browser application.

engagement_time_msec: We calculate the milliseconds from when the current page is visible to when the user leaves the current page excluding the idle time in between.

Event attributes

Sample event structure

```
{
  "unique_id": "c84ad28d-16a8-4af4-a331-f34cdc7a7a18",
  "event_type": "add_to_cart",
  "event_id": "460daa08-0717-4385-8f2e-acb5bd019ee7",
  "timestamp": 1667877566697,
  "device_id": "f24bec657ea8eff7",
  "platform": "Web",
  "make": "Google Inc.",
  "locale": "zh_CN",
  "screen_height": 1080,
  "screen_width": 1920,
  "viewport_height": 980,
  "viewport_width": 1520,
  "zone_offset": 28800000,
  "system_language": "zh",
  "country_code": "CN",
  "sdk_version": "0.2.0",
  "sdk_name": "aws-solution-clickstream-sdk",
  "host_name": "https://example.com",
  "app_id": "appId",
  "items": [{
    "id": "123",
    "name": "Nike",
    "category": "shoes",
    "price": 279.9
  }],
  "user": {
    "_user_id": {
      "value": "312121",
      "set_timestamp": 1667877566697
    }
  }
}
```

```

    },
    "_user_name": {
      "value": "carl",
      "set_timestamp": 1667877566697
    },
    "_user_first_touch_timestamp": {
      "value": 1667877267895,
      "set_timestamp": 1667877566697
    }
  },
  "attributes": {
    "event_category": "recommended",
    "currency": "CNY",
    "_session_id": "dc7a7a18-20221108-031926703",
    "_session_start_timestamp": 1667877566703,
    "_session_duration": 391809,
    "_session_number": 1,
    "_latest_referrer": "https://amazon.com/s?k=nike",
    "_latest_referrer_host": "amazon.com",
    "_page_title": "index",
    "_page_url": "https://example.com/index.html"
  }
}

```

All user attributes will be stored in user object, and all custom and global attributes in attributes object.

Common attributes

Attribute name	Data type	Description	How to generate	Usage and purpose
hashCode	string	the event object's hash code	calculated by library @aws-crypto/sha256-js	distinguish different events
app_id	string	the app_id for your app	generated by clickstream guidance when you register an	identify the events for your apps

Attribute name	Data type	Description	How to generate	Usage and purpose
			app to a data pipeline	
unique_id	string	the unique id for user	generated from <code>uuidV4()</code> by the SDK first initialization. It will be changed if user logout and then login to a new user. When user re-login to the previous user in the same browser, the <code>unique_id</code> will be reset to the same previous <code>unique_id</code> .	the unique id to identity different users and associating the behavior of logged-in and not logged-in
device_id	string	the unique id for device	generated from <code>uuidV4()</code> when the website is first open, then the <code>uuid</code> will stored in <code>localStorage</code> and will never be changed	distinguish different devices
event_type	string	event name	set by developer or SDK	distinguish different events type

Attribute name	Data type	Description	How to generate	Usage and purpose
event_id	string	the unique id for event	generated from <code>uuidV4()</code> when the event create	distinguish different events
timestamp	number	event create timestamp in millisecond	generated from <code>new Date().getTime()</code> when event create	data analysis needs
platform	string	the platform name	for browser is always Web	data analysis needs
make	string	the browser make	generated from <code>window.navigator.product</code> or <code>window.navigator.vendor</code>	data analysis needs
screen_height	number	the screen height pixel	generated from <code>window.screen.height</code>	data analysis needs
screen_width	number	the screen width pixel	generated from <code>window.screen.width</code>	data analysis needs
viewport_height	number	the website viewport height pixel	generated from <code>window.innerHeight</code>	data analysis needs

Attribute name	Data type	Description	How to generate	Usage and purpose
viewport_width	number	the website viewport width pixel	generated from <code>window.innerWidth</code>	data analysis needs
zone_offset	number	the device raw offset from GMT in milliseconds.	generated from <code>currentTime.getTimezoneOffset()*60000</code>	data analysis needs
locale	string	the default locale(language, country and variant) for the browser	generated from <code>window.navigator.language</code>	data analysis needs
system_language	string	the browser language code	generated from <code>window.navigator.language</code>	data analysis needs
country_code	string	country/region code for the browser	generated from <code>window.navigator.language</code>	data analysis needs
sdk_version	string	clickstream sdk version	generated from <code>package.json</code>	data analysis needs
sdk_name	string	clickstream sdk name	this will always be <code>aws-solution-clickstream-sdk</code>	data analysis needs

Attribute name	Data type	Description	How to generate	Usage and purpose
host_name	string	the website hostname	generated from window.location.hostname	data analysis needs

User attributes

Attribute name	Description
_user_id	Reserved for user id that is assigned by app
_user_ltv_revenue	Reserved for user lifetime value
_user_ltv_currency	Reserved for user lifetime value currency
_user_first_touch_timestamp	Added to the user object for all events. The time (in milliseconds) when the user first visited the website.

Event attributes

Attribute name	Data type	Auto track	Description
_traffic_source_source	String	false	Reserved for traffic source source. Name of the network source that acquired the user when the event were reported. Example: Google, Facebook, Bing, Baidu

Attribute name	Data type	Auto track	Description
_traffic_source_medium	String	false	Reserved for traffic medium. Use this attribute to store the medium that acquired user when events were logged. Example: Email, Paid search, Search engine.
_traffic_source_campaign	String	false	Reserved for traffic source campaign. Use this attribute to store the campaign of your traffic source. Example: summer_sale, holiday_specials
_traffic_source_name	String	false	Reserved for traffic name. Use this attribute to store the marketing campaign that acquired user when events were logged. Example: Summer promotion.
_traffic_source_campaign_id	String	false	Reserved for traffic source campaign id. Use this attribute to store the campaign id of your traffic source. Example: campaign_1, campaign_2

Attribute name	Data type	Auto track	Description
_traffic_source_term	String	false	Reserved for traffic source term. Use this attribute to store the term of your traffic source. Example: running_shoes, fitness_tracker.
_traffic_source_content	String	false	Reserved for traffic source content. Use this attribute to store the content of your traffic source. Example: banner_ad_1, text_ad_2.
_traffic_source_clid	String	false	Reserved for traffic source clid. Use this attribute to store the clid of your traffic source. Example: amazon_ad_123, google_ad_456.
_traffic_source_clid_platform	String	false	Reserved for traffic source clid platform. Use this attribute to store the clid platform of your traffic source. Example: amazon_ads, google_ads
_session_id	String	true	Added in all events.

Attribute name	Data type	Auto track	Description
_session_start_timestamp	number	true	Added in all events. The value is millisecond.
_session_duration	number	true	Added in all events. The value is millisecond.
_session_number	number	true	Added in all events.
_page_title	String	true	Added in all events.
_page_url	String	true	Added in all events.
_latest_referrer	String	true	Added in all events. The last off-site url.
_latest_referrer_host	String	true	Added in all events. The last off-site domain name.

Item attributes

Attribute name	Data type	Required	Description
id	string	False	The id of the item
name	string	False	The name of the item
brand	string	False	The brand of the item
price	number	False	The price of the item
quantity	string	False	The quantity of the item

Attribute name	Data type	Required	Description
creative_name	string	False	The creative name of the item
creative_slot	string	False	The creative slot of the item
location_id	string	False	The location id of the item
category	string	False	The category of the item
category2	string	False	The category2 of the item
category3	string	False	The category3 of the item
category4	string	False	The category4 of the item
category5	string	False	The category5 of the item

You can use the listed preset item attributes, and you can also add custom attributes to an item. In addition to the preset attributes, an item can add up to 10 custom attributes.

Google Tag Manager integration

1. Download the Clickstream SDK template file (.tpl) from the SDK Release Page.
2. Refer to the Google Tag Manager Import Guide for instructions on importing the .tpl file as a custom template in your tag manager console.
3. Refer to the Use your new tag to add ClickstreamAnalytics tag to your container.
4. The ClickstreamAnalytics tag currently supports four tag types:
 - Initialize SDK

- Record Custom Event
- Set User ID
- Set User Attribute

Important

Please ensure that you initialize the SDK tag first before use other ClickstreamAnalytics tag types.

Change logs

For more information, see the [change logs on GitHub](#).

Sample project

[Sample Web Project for SDK integration](#)

References

[Source code](#)

[Project issue](#)

Flutter SDK

Introduction

Clickstream Flutter SDK can help you easily collect in-app click stream data from mobile devices to your AWS environments through the data pipeline provisioned by this guidance.

The SDK relies on the [Clickstream Android SDK](#) and [Clickstream Swift SDK](#). Therefore, Flutter SDK also supports automatically collecting common user events and attributes (for example, session start, first open). In addition, we've added easy-to-use APIs to simplify data collection in Flutter apps.

Platform Support

Android: 4.1 (API level 16) and later

iOS: 13 and later

Integrate the SDK

1. Include SDK

```
flutter pub add clickstream_analytics
```

After completion, rebuild your Flutter application.

```
flutter run
```

2. Initialize the SDK

Copy your configuration code from your clickstream guidance web console, and the configuration code should be as follows. You can also manually add this code snippet and replace the values of `appId` and `endpoint` after you registered app to a data pipeline in the Clickstream Analytics guidance console.

```
import 'package:clickstream_analytics/clickstream_analytics.dart';

final analytics = ClickstreamAnalytics();
analytics.init(
  appId: "your appId",
  endpoint: "https://example.com/collect"
);
```

Important

- Your `appId` and `endpoint` are already set up.
- We only need to initialize the SDK once after the application starts. It is recommended to do it in the main function of your App.
- We can use `bool result = await analytics.init()` to get the boolean value of the initialization result.

3. Start using

Record event

Add the following code where you need to record event.

```
import 'package:clickstream_analytics/clickstream_analytics.dart';

final analytics = ClickstreamAnalytics();

// record event with attributes
analytics.record(name: 'button_click', attributes: {
  "event_category": "shoes",
  "currency": "CNY",
  "value": 279.9
});

// record event with name
analytics.record(name: "button_click");
```

Add global attributes

a. Add global attributes when initializing the SDK.

The following example code shows how to add traffic source fields as global attributes when initializing the SDK.

```
analytics.init({
  appId: "your appId",
  endpoint: "https://example.com/collect",
  globalAttributes: {
    Attr.TRAFFIC_SOURCE_SOURCE: "amazon",
    Attr.TRAFFIC_SOURCE_MEDIUM: "cpc",
    Attr.TRAFFIC_SOURCE_CAMPAGN: "summer_promotion",
    Attr.TRAFFIC_SOURCE_CAMPAGN_ID: "summer_promotion_01",
    Attr.TRAFFIC_SOURCE_TERM: "running_shoes",
    Attr.TRAFFIC_SOURCE_CONTENT: "banner_ad_1",
    Attr.TRAFFIC_SOURCE_CLID: "amazon_ad_123",
    Attr.TRAFFIC_SOURCE_CLID_PLATFORM: "amazon_ads",
    Attr.APP_INSTALL_CHANNEL: "amazon_store"
  }
});
```

b. Add global attributes after initializing the SDK.

```
analytics.addGlobalAttributes({
  Attr.TRAFFIC_SOURCE_MEDIUM: "Search engine",
  "level": 10
});
```

It is recommended to set global attributes after each SDK initialization, and global attributes will be included in all events that occur after it is set.

Delete global attributes

```
analytics.deleteGlobalAttributes(["level"]);
```

Login and logout

```
// when user login success.
analytics.setUserId("userId");

// when user logout
analytics.setUserId(null);
```

Add user attributes

```
analytics.setUserAttributes({
  "userName": "carl",
  "userAge": 22
});
```

Current login user's attributes will be cached in disk, so the next time app launches you don't need to set all user's attribute again, of course you can use the same api `analytics.setUserAttributes()` to update the current user's attribute when it changes.

⚠ Important

If your application is already published and most users have already logged in, please manually set the user attributes once when integrating the Clickstream SDK for the first time to ensure that subsequent events contain user attributes.

Record event with items

You can add the following code to log an event with an item, and you can add custom item attribute in the attributes Map. In addition to the preset attributes, an item can add up to 10 custom attributes.

```
var itemBook = ClickstreamItem(  
    id: "123",  
    name: "Nature",  
    category: "book",  
    price: 99,  
    attributes: {  
        "book_publisher": "Nature Research"  
    }  
);  
  
analytics.record(  
    name: "view_item",  
    attributes: {  
        Attr.VALUE: 99,  
        Attr.CURRENCY: "USD",  
        "event_category": "recommended"  
    },  
    items: [itemBook]  
);
```

For logging more attributes in an item, please refer to [item attributes](#).

⚠ Important

Only pipelines from version 1.1.0 can handle items with custom attribute. **item id** is a required attribute. If not set, the item will be discarded.

Record Screen View events manually

By default, SDK will automatically track the **preset _screen_view** event when Android Activity triggers **onResume** or iOS **ViewController** triggers **viewDidAppear**.

You can also manually record screen view events whether automatic screen view tracking is enabled, add the following code to record a screen view event with two attributes:

- **SCREEN_NAME** Required. Your screen's name.
- **SCREEN_UNIQUE_ID** Optional. Set the id of your Widget. If you do not set, the SDK will set a default value based on the hashcode of the current Activity or **ViewController**.

```
analytics.recordScreenView(  
    screenName: 'Main',  
    screenUniqueId: '123adf',  
    attributes: { ... }  
);
```

Other configurations

In addition to the required `appId` and `endpoint`, you can configure other information to get more customized usage when initializing the SDK:

```
final analytics = ClickstreamAnalytics(); analytics.init(  
    appId: "your appId",  
    endpoint: "https://example.com/collect", isLogEvents: false,  
    isCompressEvents: false, sendEventsInterval: 10000, isTrackScreenViewEvents: true,  
    isTrackUserEngagementEvents: true, isTrackAppExceptionEvents: false, authCookie: "your  
    auth cookie", sessionTimeoutDuration: 1800000,  
    globalAttributes: {  
        "_traffic_source_medium": "Search engine",  
    }  
);
```

Here is an explanation of each option:

Name	Required	Default value	Description
appId	true	N/A	the app id of your application in control plane
endpoint	true	N/A	the endpoint path you will upload the event to Clickstream ingestion server
isLogEvents	false	false	whether to print out event json in console for debugging events
isCompressEvents	false	true	whether to compress event content by gzip when uploading events
sendEventsInterval	false	10,000	event sending interval in milliseconds
isTrackScreenViewEvents	false	true	whether auto record screen view events in app
isTrackUserEngagementEvents	false	true	whether auto record user engagement events in app
isTrackAppExceptionEvents	false	false	whether auto track exception event in app

Name	Required	Default value	Description
authCookie	false	N/A	your auth cookie for AWS application load balancer auth cookie
sessionTimeoutDuration	false	1,800,000	the duration for session timeout in milliseconds
globalAttributes	false	--	the global attributes when initializing the SDK

Configuration update

You can update the default configuration after initializing the SDK. The following are additional configuration options that you can customize.

```
final analytics = ClickstreamAnalytics();
analytics.updateConfigure(
    appId: "your appId",
    endpoint: "https://example.com/collect",
    isLogEvents: true,
    isCompressEvents: false,
    isTrackScreenViewEvents: false,
    isTrackUserEngagementEvents: false,
    isTrackAppExceptionEvents:
    false, authCookie: "test cookie");
```

Send event immediately

```
final analytics = ClickstreamAnalytics();
analytics.flushEvents();
```

Disable SDK

You can disable the SDK in the scenario you need. After disabling the SDK, the SDK will not handle the logging and sending of any events. You can enable the SDK when you need to continue logging events.

```
final analytics = ClickstreamAnalytics();

// disable SDK
ClickstreamAnalytics.disable();

// enable SDK
ClickstreamAnalytics.enable();
```

Debug events

You can complete the following steps to view the event raw JSON and debug your events.

1. Using **analytics.updateConfigure()** API and set the **isLogEvents** attributes with **true** in debug mode, for example:

```
// log the event in debug mode.
analytics.updateConfigure(isLogEvents: true);
```

2. Integrate the SDK and launch your app.
 - a. For Android application logs, we can see the logs directly in the terminal window. You can also use filters in the Android Studio **Logcat** window to view logs.
 - b. For iOS application logs, launch it via Xcode and open the log panel to see it.
3. Input **EventRecorder** to the filter, and you will see the JSON content of all events recorded by Clickstream Flutter SDK.

Data format definition

Data types

Clickstream Flutter SDK supports the following data types.

Data type	Range	Example
int	-9223372036854775808 ~ 9223372036854775807	12
double	5e-324 ~ 1.79e+308	3.14
bool	true, false	true

Data type	Range	Example
String	max 1024 characters	"Clickstream"

Naming rules

1. The event name and attribute name cannot start with a number, and must contain only uppercase and lowercase letters, numbers, and underscores. In case of an invalid attribute name or user attribute name, it will discard the attribute and record error.
2. Do not use `_` as a prefix in an event name or attribute name. The `_` prefix is reserved for the guidance.
3. The event name and attribute name are case-sensitive, so `Add_to_cart` and `add_to_cart` will be recognized as two different event names.

Event and attribute limitation

To improve the efficiency of querying and analysis, we apply limits to event data as follows:

Name	Suggestion	Hard limit	Strategy	Error code
Event name invalid	N/A	N/A	discard event, print log, and record <code>_clickstream_errorevent</code>	1001
Length of event name	under 25 characters	50 characters	discard event, print log, and record <code>_clickstream_errorevent</code>	1002
Length of event attribute name	under 25 characters	50 characters	discard the attribute, print log, and record error in event attribute	2001

Name	Suggestion	Hard limit	Strategy	Error code
Attribute name invalid	N/A	N/A	discard the attribute, print log, and record error in event attribute	2002
Length of event attribute value	under 100 characters	1024 characters	discard the attribute, print log, and record error in event attribute	2003
Event attribute per event	under 50 attributes	500 event attributes	discard the attribute, print log, and record error in event attribute	2004
User attribute number	under 25 attributes	100 user attributes	discard event, print log, and record_clickstream_errorevent	3001
Length of User attribute name	under 25 characters	50 characters	discard event, print log, and record_clickstream_errorevent	3002
User attribute name invalid	N/A	N/A	discard event, print log, and record_clickstream_errorevent	3003

Name	Suggestion	Hard limit	Strategy	Error code
Length of User attribute value	under 50 characters	256 characters	discard event, print log, and record_clickstream_erronevent	3004
Item number in one event	under 50 items	100 items	discard the attribute, print log, and record error in event attribute	4001
Length of item attribute value	under 100 characters	256 characters	discard the attribute, print log, and record error in event attribute	4002
Custom item attribute number in one item	under 10 custom attributes	10 custom attributes	discard the attribute, print log, and record error in event attribute	4003
Length of item attribute name	under 25 characters	50 characters	discard the attribute, print log, and record error in event attribute	4004
Item attribute name invalid	N/A	N/A	discard the attribute, print log, and record error in event attribute	4005

⚠ Important

- The character limits are the same for single-width character languages (e.g., English) and double-width character languages (e.g., Chinese).
- The limit of event attribute per event include preset attributes.
- If the attribute or user attribute with the same name is added more than twice, the latest value will apply.
- All errors that exceed the limit will be recorded `_error_code` and `_error_message` these two attribute in the event attributes.

Preset events

For Android, refer to [Android SDK preset events](#).

For iOS, refer to [Swift SDK preset events](#).

Event attributes

For Android, refer to [Android SDK event attributes](#).

For iOS, refer to [Swift SDK event attributes](#).

Change log

For more information, see [GitHub change log](#).

Native SDK version dependencies

Flutter SDK Version	Android SDK Version	Swift SDK Version
0.3.0 ~ 0.4.0	0.12.0	0.11.0
0.2.0	0.10.0	0.9.1
0.1.0	0.9.0	0.8.0

References

[Source code](#)

[Project issue](#)

ReactNative SDK

Clickstream React NativeSDK can help you easily collect in-app click stream data from mobile devices to your AWS environments through the data pipeline provisioned by this guidance.

The SDK relies on the [Clickstream Android SDK](#) and [Clickstream Swift SDK](#). Therefore, React Native SDK also supports automatically collect common user events and attributes (e.g., session start, first open) . In addition, we've added easy-to-use APIs to simplify data collection in React Native apps.

Platform Support

Android: 4.1 (API level 16) and later

iOS: 13 and later

Integrate the SDK

1. Include SDK

```
npm install @aws/clickstream-react-native
```

After completion, you need to install the pod dependencies for iOS:

```
cd ios pod install;
```

2. Initialize the SDK

Copy your configuration code from your clickstream guidance web console, and the configuration code should be as follows. You can also manually add this code snippet and replace the values of appId and endpoint after you registered app to a data pipeline in the Clickstream Analytics guidance console

```
import { ClickstreamAnalytics } from '@aws/clickstream-react-native';
```

```
ClickstreamAnalytics.init({
  appId: 'your appId',
  endpoint: 'https://example.com/collect',
});
```

Important

- Your appId and endpoint are already set up in it. We only need to initialize the SDK once after the application starts.
- Make sure you call `ClickstreamAnalytics.init()` as early as possible in your application's life-cycle. And make sure the SDK is initialized when calling other APIs.
- We can use `const result = await ClickstreamAnalytics.init()` to get the boolean value of the initialization result.

3. Start using.

Record event

Add the following code where you need to record event.

```
import { ClickstreamAnalytics } from '@aws/clickstream-react-native';

// record event with attributes
ClickstreamAnalytics.record({
  name: 'button_click',
  attributes: {
    event_category: 'shoes',
    currency: 'CNY',
    value: 279.9,
  },
});

// record event with name
ClickstreamAnalytics.record({name: 'button_click'});
```

Add global attributes

1. Add global attributes when initializing the SDK.

The following example code shows how to add traffic source fields as global attributes when initializing the SDK.

```
import { ClickstreamAnalytics, Attr } from '@aws/clickstream-react-native';

ClickstreamAnalytics.init({
  appId: "your appId",
  endpoint: "https://example.com/collect",
  globalAttributes: {
    [Attr.TRAFFIC_SOURCE_SOURCE]: 'amazon',
    [Attr.TRAFFIC_SOURCE_MEDIUM]: 'cpc',
    [Attr.TRAFFIC_SOURCE_CAMPAIN]: 'summer_promotion',
    [Attr.TRAFFIC_SOURCE_CAMPAIN_ID]: 'summer_promotion_01',
    [Attr.TRAFFIC_SOURCE_TERM]: 'running_shoes',
    [Attr.TRAFFIC_SOURCE_CONTENT]: 'banner_ad_1',
    [Attr.TRAFFIC_SOURCE_CLID]: 'amazon_ad_123',
    [Attr.TRAFFIC_SOURCE_CLID_PLATFORM]: 'amazon_ads',
    [Attr.APP_INSTALL_CHANNEL]: 'amazon_store',
  }
});
```

2. Add global attributes after initializing the SDK

```
import { ClickstreamAnalytics, Attr } from '@aws/clickstream-react-native';

ClickstreamAnalytics.setGlobalAttributes({
  [Attr.TRAFFIC_SOURCE_MEDIUM]: "Search engine",
  level: 10,
});
```

It is recommended to set global attributes after each SDK initialization, and global attributes will be included in all events that occur after it is set.

Delete global attribute

```
ClickstreamAnalytics.deleteGlobalAttributes(['level', '_traffic_source_medium']);
```

Login and logout

```
import { ClickstreamAnalytics } from '@aws/clickstream-react-native';
```

```
// when user login success.  
ClickstreamAnalytics.setUserId("userId");  
  
// when user logout  
ClickstreamAnalytics.setUserId(null);
```

Add user attribute

```
analytics.setUserAttributes({  
  "userName": "carl",  
  "userAge": 22  
});
```

Current login user's attributes will be cached in disk, so the next time app launches you don't need to set all user's attribute again, of course you can use the same api `ClickstreamAnalytics.setUserAttributes()` to update the current user's attribute when it changes.

Important

If your application is already published and most users have already logged in, please manually set the user attributes once when integrating the Clickstream SDK for the first time to ensure that subsequent events contain user attributes.

Record event with items

You can add the following code to log an event with an item.

```
import { ClickstreamAnalytics, Item, Attr } from '@aws/clickstream-react-native';  
  
const itemBook: Item = {  
  id: '123',  
  name: 'Nature',  
  category: 'book',  
  price: 99,  
  book_publisher: "Nature Research",  
};  
ClickstreamAnalytics.record({  
  name: 'view_item',  
  attributes: {  
    [Attr.VALUE]: 99,  
  },  
});
```

```
[Attr.CURRENCY]: 'USD',
  event_category: 'recommended',
},
items: [itemBook],
});
```

For logging more attributes in an item, please refer to [item attributes](#).

Important

Only pipelines from version 1.1.0 can handle items with custom attribute. **item id** is a required attribute. If not set, the item will be discarded.

Record Screen View events manually

By default, SDK will automatically track the **preset _screen_view** event when Android Activity triggers **onResume** or iOS **ViewController** triggers **viewDidAppear**.

You can also manually record screen view events whether automatic screen view tracking is enabled, add the following code to record a screen view event with two attributes:

- **SCREEN_NAME** Required. Your screen's name.
- **SCREEN_UNIQUE_ID** Optional. Set the id of your Widget. If you do not set, the SDK will set a default value based on the hashcode of the current Activity or **ViewController**.

```
import { ClickstreamAnalytics } from '@aws/clickstream-react-native';

ClickstreamAnalytics.record({
  name: ClickstreamAnalytics.Event.SCREEN_VIEW,
  attributes: {
    [ClickstreamAnalytics.Attr.SCREEN_NAME]: 'HomeComponet',
    [ClickstreamAnalytics.Attr.SCREEN_UNIQUE_ID]: '123adf',
  },
});
```

Record Screen Views for React Navigation

Here's an example of globally logging React Native screen view events when using React Navigation 6.x

For other version of React Navigation, you can refer to official documentation: [Screen tracking for analytics](#).

Send event immediately

```
import { ClickstreamAnalytics } from '@aws/clickstream-react-native';

ClickstreamAnalytics.flushEvents();
```

Other configurations

In addition to the required `appId` and `endpoint`, you can configure other information to get more customized usage when initializing the SDK:

```
import { ClickstreamAnalytics } from '@aws/clickstream-react-native';

ClickstreamAnalytics.init({
  appId: 'your appId',
  endpoint: 'https://example.com/collect',
  isLogEvents: true,
  isCompressEvents: true,
  isTrackScreenViewEvents: false,
  isTrackUserEngagementEvents: true,
  isTrackAppExceptionEvents: true,
  sendEventsInterval: 15000,
  sessionTimeoutDuration: 1800000,
  authCookie: 'your auth cookie',
  globalAttributes: {
    _traffic_source_medium: 'Search engine',
  },
});
```

Here is an explanation of each option:

Name	Required	Default value	Description
<code>appId</code>	true	N/A	the app id of your application in control plane

Name	Required	Default value	Description
endpoint	true	N/A	the endpoint path you will upload the event to Clickstream ingestion server
isLogEvents	false	false	whether to print out event json in console for debugging events
isCompressEvents	false	true	whether to compress event content by gzip when uploading events
isTrackScreenViewEvents	false	true	whether auto record screen view events in app
isTrackUserEngagementEvents	false	true	whether auto record user engagement events in app
isTrackAppExceptionEvents	false	false	whether auto track exception event in app
sendEventsInterval	false	10,000	event sending interval in milliseconds
sessionTimeoutDuration	false	1,800,000	the duration for session timeout in milliseconds
authCookie	false	N/A	your auth cookie for AWS application load balancer auth cookie

Name	Required	Default value	Description
globalAttributes	false	--	the global attributes when initializing the SDK

Configuration update

You can update the default configuration after initializing the SDK. The following are additional configuration options that you can customize.

```
import { ClickstreamAnalytics } from '@aws/clickstream-react-native';

ClickstreamAnalytics.updateConfigure({
  appId: 'your appId',
  endpoint: 'https://example.com/collect',
  isLogEvents: true,
  authCookie: 'your auth cookie',
  isCompressEvents: true,
  isTrackScreenViewEvents: false,
  isTrackUserEngagementEvents: false,
  isTrackAppExceptionEvents: false,
});
```

Disable SDK

You can disable the SDK in the scenario you need. After disabling the SDK, the SDK will not handle the logging and sending of any events. You can enable the SDK when you need to continue logging events.

```
import { ClickstreamAnalytics } from '@aws/clickstream-react-native';

// disable SDK
ClickstreamAnalytics.disable();

// enable SDK
ClickstreamAnalytics.enable();
```

Debug events

You can complete the following steps to view the event raw JSON and debug your events.

1. Enable the **isLogEvents** configuration when initializing the SDK.

```
import { ClickstreamAnalytics } from '@aws/clickstream-react-native';

ClickstreamAnalytics.init({
  appId: 'your appId',
  endpoint: 'https://example.com/collect',
  isLogEvents: true,
});
```

2. After configuring **isLogEvents:true**, when you record an event, you can see the event raw json in AndroidStudio Logcat or Xcode debug console by filter **EventRecorder**.

Data format definition

Refer to [Web SDK Data format](#) definition.

Preset events

For Android, refer to [Android SDK preset events](#).

For iOS, refer to [Swift SDK preset events](#).

Event attributes

For Android, refer to [Android SDK event attributes](#).

For iOS, refer to [Swift SDK event attributes](#).

Change log

For more information, see [GitHub change log](#).

Native SDK version dependencies

Flutter SDK Version	Android SDK Version	Swift SDK Version
0.1.0 ~ 0.2.0	0.12.0	0.11.0

Sample project

[Sample React Native Project](#) for SDK integration.

References

[Source code](#)

[Project issue](#)

HTTP API

This section introduces how to send your clickstream data directly to the Clickstream ingestion server via HTTP requests. The Clickstream data processing module will correctly process your data simultaneously by following the guidelines below. Then, you can visually analyze them in the subsequent report module.

Request endpoint

After creating the application in the solution web console, you will get the **Server Endpoint** and **App ID** on the details page. For example:

- **Server Endpoint:** `https://example.com/collect`
- **App ID:** `my_app`

API Specification

- The app ID in the query parameters must be one of the applications you create for the project in the solution web console. Otherwise, the server will respond to HTTP status code 403.
- The request body contains four parts: common attributes, items, user, and attributes. The public attributes require the `event_type`, `event_id`, `timestamp`, and `app_id`; the rest are optional parameters.
- The total size of the body of a single request cannot exceed 1MB. Otherwise, the HTTP status code 413 will return.

Request method

POST

Request headers

Parameter name	Required	Example	Description
Content-Type	YES	application/json; charset=utf-8	Content type
X-Forwarded-For	NO	101.188.67.134	Source IP address, it's required if you forward the client requests to clickstream servers from your servers
User-Agent	NO	Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Mobile Safari/537.36	User-Agent
cookie	NO	your auth cookie	The authentication token for your request. Please refer to server side configuration .

Request query parameters

Parameter name	Required	Example	Description
appld	YES	test_app	The app ID for your application is created in the guidance web console
platform	NO	Android/iOS/Web/...	Distinguish between platforms

Parameter name	Required	Example	Description
event_bundle_sequence_id	NO	1	Request sequence number, an incrementing integer starting from 1
hashCode	NO	478acd09	The first eight digits of the sha256 calculation result of the request body string
compression	NO	gzip	Request body compression method. Currently, only gzip is supported. Keeping it absent means no compression

Request body

The request body is an array structure that contains the JSON string of one or more events. For example:

```
[
  {
    "event_type": "button_click",
    "event_id": "460daa08-0717-4385-8f2e-acb5bd019ee7",
    "timestamp": 1667877566697,
    "app_id": "your appId",
    "attributes": {
      "productName": "shoes",
      "Price": 99.9
    }
  },
  {
    "event_type": "item_view",
    "event_id": "c6067c1c-fd8d-4fdb-bfaf-cc1212ca0195",
```

```

    "timestamp": 1667877565698,
    "app_id": "your appId",
    "attributes": {
      "productName": "book",
      "Price": 39.9
    }
  }
]

```

Event attributes

Attributes name	Required	Data Type	Example	Description
event_type	YES	String	button_click	Event type
event_id	YES	String	460daa08-0717-4385-8f2e-acb5bd019ee7	Event's unique ID, we recommend using UUID to generate
timestamp	YES	Long	1667877566697	The timestamp when the event was generated, in milliseconds
app_id	YES	String	shopping_dev	The corresponding id when creating the application in the Clickstream web console
platform	NO	String	Android/iOS/Web/...	Device platform
os_version	NO	String	10	System version
unique_id	NO	String	c84ad28d-16a8-4af4	Unique ID to identify

Attributes name	Required	Data Type	Example	Description
			-a331-f34 cdc7a7a18	different users and associate the behavior of logged-in and not logged-in
device_id	NO	String	f24bec657 ea8eff7	Distinguish between different devices
make	NO	String	Samsung	Device manufactory
brand	NO	String	Samsung	Device brand
model	NO	String	S23 Ultra	Device model
carrier	NO	String	CDMA	Device network operator name
network_type	NO	String	Mobile	Current device network type
locale	NO	String	zh_CN	Local information
system_language	NO	String	zh	Device language code
country_code	NO	String	CN	Device country code
zone_offset	NO	int	2880000	Device's raw offset from GMT in milliseconds
screen_height	NO	int	1920	Screen height in pixels

Attributes name	Required	Data Type	Example	Description
screen_width	NO	int	1080	Screen width in pixels
viewport_height	NO	int	540	App viewport height
viewport_width	NO	int	360	App viewport width
sdk_version	NO	String	1.2.3	SDK version
sdk_name	NO	String	aws-solution-clickstream-sdk	SDK name
app_package_name	NO	String	com.example.app	User's Application package name
app_version	NO	String	1.1.0	Application version number
app_title	NO	String	shopping	Application name

Attributes name	Required	Data Type	Example	Description
items	NO	Object	[{ "id": "b011ddc3-632f-47c b-a68a-ad83678ecfe d", "name": "Classic coat-rack", "category": "housewares", "price": 167 }]	Item list, Supports uploading multiple items at one time. A maximum of 100 items can be uploaded at one time For the item quantity limit, please refer to Event and Attribute Limitation For the supported attributes of the item, please refer to item attribute

Attributes name	Required	Data Type	Example	Description
user	NO	Object	<pre>{ "_user_id": { "value": "0202d0e1", "set_time stamp": 1695006816345 }, "username": { "value": "carl", "set_time stamp": 1695006816345 } }</pre>	<p>User attributes. Each attribute key is the user attribute name. Each attribute contains an object. The object contains two attributes: value: The value of the user attribute. set_timestamp: The timestamp millisecond value when setting the attribute. Up to 100 user attributes can be added to an event. For specific restrictions, please refer to: Event and Attribute Limitations</p>

Attributes name	Required	Data Type	Example	Description
attributes	NO	Object	{ "productName": "book", "Price": 39.9 }	Custom attributes. Up to 500 custom attributes can be added to an event, and the attribute name must meet the naming rules .

Request response

If the HttpStatusCode status code returned by the request is 200, the request is considered successful, other status codes are failures, and the request does not return any other content.

HttpCode

Code	Description
200	Request successful
403	Request failed. Please check if appId and endpoint match, if configured with authentication, please check whether the authentication cookie is correct
413	Request failed. The request body exceeds 1MB

Request code example

cURL:

```
curl --location 'https://example.com/collect?
appId=test_release&platform=Android&event_bundle_sequence_id=1' \
```

```
--header 'Content-Type: application/json; charset=utf-8' \
--header 'X-Forwarded-For: 101.188.67.134' \
--data '[{"event_type":"button_click","event_id":"460daa08-0717-4385-8f2e-
acb5bd019ee7","timestamp":1667877566697,"app_id":"your appId","attributes":
{"productName":"shoes","Price":99.9}},{"event_type":"item_view","event_id":"c6067c1c-
fd8d-4fdb-bfaf-cc1212ca0195","timestamp":1667877565698,"app_id":"your
appId","attributes":{"productName":"book","Price":39.9}}]'
```

C# HttpClient:

```
var client = new HttpClient();
var request = new HttpRequestMessage(HttpMethod.Post, "https://example.com/collect?
appId=test_release&platform=Android&event_bundle_sequence_id=1");
request.Headers.Add("X-Forwarded-For", "101.188.67.134");
var content = new StringContent("[{\"event_type\": \"button_click\", \"event_id\":
\"460daa08-0717-4385-8f2e-acb5bd019ee7\", \"timestamp\": 1667877566697, \"app_id\": \"your
appId\", \"attributes\": {\"productName\": \"shoes\", \"Price\": 99.9}}, {\"event_type
\": \"item_view\", \"event_id\": \"c6067c1c-fd8d-4fdb-bfaf-cc1212ca0195\", \"timestamp
\": 1667877565698, \"app_id\": \"your appId\", \"attributes\": {\"productName\": \"book\",
\"Price\": 39.9}}]", null, "application/json; charset=utf-8");
request.Content = content;
var response = await client.SendAsync(request);
response.EnsureSuccessStatusCode();
Console.WriteLine(await response.Content.ReadAsStringAsync());
```

Java Okhttp:

```
OkHttpClient client=new OkHttpClient().newBuilder()
    .build();
MediaType mediaType=MediaType.parse("application/json; charset=utf-8");
RequestBody body=RequestBody.create(mediaType,"[\"event_type\": \"button_click
\", \"event_id\": \"460daa08-0717-4385-8f2e-acb5bd019ee7\", \"timestamp\": 1667877566697,
\"app_id\": \"your appId\", \"attributes\": {\"productName\": \"shoes\", \"Price\": 99.9}},
{\"event_type\": \"item_view\", \"event_id\": \"c6067c1c-fd8d-4fdb-bfaf-cc1212ca0195\",
\"timestamp\": 1667877565698, \"app_id\": \"your appId\", \"attributes\": {\"productName\":
\"book\", \"Price\": 39.9}}]");
Request request=new Request.Builder()
    .url("https://example.com/collect?
appId=test_release&platform=Android&event_bundle_sequence_id=1")
    .method("POST", body)
    .addHeader("Content-Type", "application/json; charset=utf-8")
    .addHeader("X-Forwarded-For", "101.188.67.134")
    .build();
```

```
Response response=client.newCall(request).execute();
```

JavaScript Fetch:

```
var myHeaders = new Headers();
myHeaders.append("Content-Type", "application/json; charset=utf-8");
myHeaders.append("X-Forwarded-For", "101.188.67.134");

var raw = "[{\"event_type\":\"button_click\",\"event_id\":\"460daa08-0717-4385-8f2e-acb5bd019ee7\",\"timestamp\":1667877566697,\"app_id\":\"your appId\",\"attributes\":{\"productName\":\"shoes\",\"Price\":99.9}},{\"event_type\":\"item_view\",\"event_id\":\"c6067c1c-fd8d-4fdb-bfaf-cc1212ca0195\",\"timestamp\":1667877565698,\"app_id\":\"your appId\",\"attributes\":{\"productName\":\"book\",\"Price\":39.9}}]";

var requestOptions = {
  method: 'POST',
  headers: myHeaders,
  body: raw,
  redirect: 'follow'
};

fetch("https://example.com/collect?
appId=test_release&platform=Android&event_bundle_sequence_id=1", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

Python Request:

```
import requests

url = "https://example.com/collect?
appId=test_release&platform=Android&event_bundle_sequence_id=1"

payload = "[{\"event_type\":\"button_click\",\"event_id\":
\"460daa08-0717-4385-8f2e-acb5bd019ee7\",\"timestamp\":1667877566697,\"app_id\":\"your
appId\",\"attributes\":{\"productName\":\"shoes\",\"Price\":99.9}},{\"event_type
\":\"item_view\",\"event_id\":\"c6067c1c-fd8d-4fdb-bfaf-cc1212ca0195\",\"timestamp
\":1667877565698,\"app_id\":\"your appId\",\"attributes\":{\"productName\":\"book\",
\"Price\":39.9}}]"

headers = {
  'Content-Type': 'application/json; charset=utf-8',
  'X-Forwarded-For': '101.188.67.134'
```

```
}  
  
response = requests.request("POST", url, headers=headers, data=payload)  
  
print(response.text)
```

Verification data reported successfully

If you enabled data processing, you can query the **event**, **event_parameter**, **user**, **item** or **ingestion_events** table in Athena directly through SQL.

Moreover, if you enabled Redshift in data modeling, you can query the **event**, **event_parameter**, **user**, or **item** table in Redshift directly through SQL.

User identifier

When you perform data analysis, you usually need to select an appropriate user identifier for analysis based on your business analysis scenario. This will help you improve the accuracy of your analysis, especially in funnel, retention, session and other analysis scenarios.

Clickstream Analytics on AWS guidance mainly contains three types of IDs:

- User ID
- Device ID
- User Pseudo ID

This section introduces how to use these three IDs respectively, and you will learn in detail how to use User ID and User Pseudo ID to correlate user behavior.

User ID

User ID is usually a unique identifier that describes the user in your business database, which is relatively more accurate and unique.

- When the user is not registered or logged in, the value of User ID is empty.
- The SDK provides the `ClickstreamAnalytics.setUserId("your user id")` method to set the User ID. When logging out, set null/nil to clear the User ID.
- The User ID is stored in the `user_id` field in the user table.

Device ID

The guidance uses Device ID to identify user devices.

- The Device ID will be automatically generated when the app is launched for the first time after integrating the SDK.
- The Device ID may not be the unique identifier of the device. Usually the Device ID may be regenerated after the user uninstalls the app or clears the cache on the web page.
- The Device ID is stored in the `device_id_list` field in the user table.

The following table will introduce how the SDK on each end generates the Device ID.

SDK Types	Generate Rules	Storage Location	Is Unique
Android SDK	By default, <code>AndroidId</code> is used as the Device ID. If the <code>AndroidId</code> cannot be obtained, a random UUID is used instead	Stored in <code>SharedPreferences</code> key-value pair file	Usually, the <code>AndroidId</code> will not change even if the app is uninstalled and reinstalled. If using the UUID as Device ID, it will change after the user uninstalls and reinstalled the app
Swift SDK	If your app has been authorized to obtain IDFA, use IDFA as the Device ID. Otherwise, use IDFV as the Device ID. If IDFV cannot be obtained, use a random UUID	<code>UserDefault</code> key-value pair file	Usually, the IDFA does not change even if the app is uninstalled and reinstalled. When using IDFV or UUID, the Device ID will change after the user uninstalls and reinstalled the app
Web SDK	By default, a random UUID is used as the device ID	In the browser's <code>localStorage</code>	Device ID will be regenerated after

SDK Types	Generate Rules	Storage Location	Is Unique
			user clears browser cache

User Pseudo ID

Clickstream Analytics on AWS guidance uses User Pseudo ID to correlate logged-in and non-logged-in behavior on the same device.

- User Pseudo IDs are generated from random UUIDs in all SDKs.
- User Pseudo ID will only be reassigned when a new user logs in on the current device. When switching to a user who has already logged in on the current device, it will revert to the User Pseudo ID of the previous user.
- The User Pseudo ID is stored in the `user_pseudo_id` field of the user table.

The following table lists the correspondence between Device ID, User ID, and User Pseudo ID under various scenarios.

Sequence	Events	Device ID	User ID	User Pseudo ID
1	Install App	S	--	1
2	Use the App	S	--	1
3	Logged in user A	S	A	1
4	Use the App	S	A	1
5	Sign out and view	S	--	1
6	Logged in user B	S	B	2
7	Use the App	S	B	2
8	Sign out and view	S	--	2

Sequence	Events	Device ID	User ID	User Pseudo ID
9	Logged in user A	S	A	1
10	Use the App	S	A	1
11	Sign out and view	S	--	1
12	Logged in user C	S	C	3
13	Use the App	S	C	3

As shown in the table, you can count all the behavioral events of user A on device S when user A did not log in and after logging in twice by looking for `user_pseudo_id=1`. Additionally, you can use User ID to join user clickstream data with data from your business systems to build a more complete customer data platform.

Note

When the user uninstalls the app or clears the browser cache, the relationship between the original User Pseudo ID and User ID will be cleared on the device, and a new User Pseudo ID will be generated on the device.

Migrate from third-party SDKs

Introduction

This article provides a best practice for you to migrate from a third-party SDK to Clickstream SDK. If you already have an SDK in your app or website, and you want to replace it with Clickstream SDK, we recommend you adopt this practice, which allow you to achieve a smooth migration with the following benefits:

- Minimum code changes
- Reuse existing data tracking codes
- Quick implementation time

- Dual measurement to ensure data completeness

In summary, we recommend you create one overarching analytic logger function that encapsulates all the event logging methods from both legacy SDK and Clickstream SDK, so that you have one API to log event data to multiple destinations. Once satisfied with the data, you can easily update the function to disable the legacy SDK data logging.

To make it easier to understand, this example uses Clickstream Web SDK to replace Firebase Web SDK (GA4 SDK). Assuming you have integrated Firebase Web SDK into your website, follow the steps below.

Step 1: Integrate Clickstream Web SDK

1. Include SDK

```
npm install @aws/clickstream-web
```

2. Initialize the SDK

Copy your configuration code from your clickstream guidance web console. We recommend you add the code to your app's root entry point, for example `index.js/app.tsx` in React or `main.ts` in Vue/Angular. The configuration code should look as follows.

```
import { ClickstreamAnalytics } from '@aws/clickstream-web';

ClickstreamAnalytics.init({
  appId: "your appId",
  endpoint: "https://example.com/collect",
});
```

Step 2: Encapsulate common data logger methods

When integrating multiple data analysis SDKs, it is strongly recommended that you encapsulate all event-logging methods in one function. Processing data logging codes of different SDKs in the same place can make the code concise and easy for you to maintain. Below is an example of our encapsulation that you can copy directly into your project.

```
import { ClickstreamAnalytics } from "@aws/clickstream-web";
```

```
import { getAnalytics, logEvent, setUserProperties, setUserId } from "firebase/
analytics";

export const AnalyticsLogger = {

  log(eventName, attributes, items) {
    attributes = attributes ?? {}
    const [{"items": items, ...mAttributes} = attributes;

    // Clickstream SDK
    ClickstreamAnalytics.record({
      name: eventName,
      attributes: mAttributes,
      items: items
    })

    //Firebase SDK
    const analytics = getAnalytics();
    logEvent(analytics, eventName, attributes);
  },

  setUserAttributes(attributes) {
    // Clickstream SDK
    ClickstreamAnalytics.setUserAttributes(attributes);

    // Firebase SDK
    const analytics = getAnalytics();
    setUserProperties(analytics, attributes);
  },

  setUserId(userId) {
    //Clickstream SDK
    ClickstreamAnalytics.setUserId(userId)

    //Firebase SDK
    const analytics = getAnalytics();
    setUserId(analytics, userId);
  },
}
```

We need to encapsulate three APIs `log()`、`setUserAttributes()` and `setUserId()`. When we invoke the `AnalyticsLogger.log('testEvent')` method, both Clickstream and Firebase SDK will log the event, so we only need to call the `AnalyticsLogger` API when you need to log event data.

Step 3: Migrate to common APIs in minutes

For log events

```
onSignedUp(user) {
  let attributes = {
    _user_id: user.id,
    username: user.username,
    email: user.email,
  };
  -- logEvent(analytics, 'sign_up', attributes);
  ++ AnalyticsLogger.log('sign_up', attributes);
}
```

For the events log API, we need to get the event name and attributes for the events log API and pass them into the new API. Of course, you can also use the "Replace in File" feature to make quick changes, as shown in the image below.

The screenshot shows an IDE interface with a "Replace in Files" dialog box. The dialog box contains the following text:

```
Replace All
Replace 13 occurrences of
'logEvent(analytics, '
across 2 files with 'AnalyticsLogger.log('?
```

Below the dialog box, the IDE shows a list of files with line numbers:

- AnalyticsHandler.js 156
- AnalyticsHandler.js 175
- AnalyticsHandler.js 209
- AnalyticsHandler.js 244**
- AnalyticsHandler.js 304
- AnalyticsHandler.js 339
- AnalyticsHandler.js 377
- AnalyticsHandler.js 408
- AnalyticsHandler.js 439
- AnalyticsHandler.js 491

The code editor shows the following code snippet:

```
AnalyticsHandler.js src/analytics
242     ],
243   }
244   logEvent(analytics, eventName: "add_to_cart", attributes);
245 },
246
```

For log user attributes

```
userSignedIn(user) {
  -- setUserId(analytics, user.id);
```

```
++ AnalyticsLogger.setUserId(user.id);
   let attributes = {
       _user_id: user.id,
       username: user.username,
       email: user.email,
   };
-- setUserProperties(analytics, attributes);
++ AnalyticsLogger.setUserAttributes(attributes);
}
```

For user id, replace `setUserId()` with `AnalyticsLogger.setUserId()` .

For user attributes,

replace `setUserProperties()` with `AnalyticsLogger.setUserAttributes()` .

Summary

In summary, it is easy to get Clickstream SDK and Firebase SDK to work together. After these three steps, your data will be uploaded to Clickstream Analytics and Firebase, these two SDKs will work well together and will not influence each other. After you are satisfied with the data, you only need to modify the `AnalyticsLogger` file to remove or disable another SDK smoothly.

Analytics Studio

Analytics Studio is a unified web interface for business analysts or data analysts to create and view dashboards, query and explore clickstream data, and manage metadata.

Modules

Below are the modules included in the Analytics Studio.

- **Dashboard.** View the out-of-the-box dashboard and custom dashboards (if any).
- **Exploration.** Use advanced analytics models to query clickstream data beyond the out-of-the-box dashboard.
- **Analyses.** Create and modify dashboards, as well as manage datasets.
- **Data management.** View and manage the metadata for clickstream data.

Terms and concepts

This section describes key concepts and terms used in Analytics Studio.

Event. Clickstream data generated by a user action within an app or website (such as clicking, and visiting a page).

Preset event. Events that are automatically collected by clickstream SDK, usually with a name starting with an underscore '_'.

Custom event. Events that are defined and collected by an app owner. The collection timing and business purpose of the event varies by apps.

Event parameter. Event parameters are used to describe the various dimensions of information at the time an event occurred. They fall into two categories: public parameters and private parameters.

Public parameter. The parameters that all events include, such as user information (`user_id`), and device information (app version, device model, etc.)

Private parameter. The parameters that are unique to certain events, such as user-customized parameters.

User attribute. User attribute is used to record the property of a user. They fall into two categories: preset attributes collected by SDK presets, such as '_first_visit_date'; custom attributes, that is, user attributes reported by the user themselves, such as 'email_address'.

Dashboard

Overview

Clickstream Analytics on AWS collects data from your websites and apps to create dashboards that derive insights. You can use dashboards to monitor traffic, investigate data, and understand your users and their activities.

Depends on your pipeline configuration, the time for data to be available in your dashboard varies. By default, the metrics in the out-of-the-box dashboards are calculated on a daily basis, you should see the data and metrics from previous day (according to your app's time zone).

View dashboards

Use this procedure to view dashboards:

1. Go to **Clickstream Analytics on AWS Console**, in the **Navigation Bar**, choose **Analytics Studio**.
2. In the Analytics Studio page that opens, select the project and app you just created in the drop-down list at the top of the web page.
3. Choose the **User lifecycle - default** dashboard.

Reports

The dashboard contains a set of reports throughout the user lifecycle, which aim to help you understand how people use your website or app, from acquisition to retention.

Report name	What it is
Acquisition	Summarizes key metrics about new users, and provides detail view user profile
Engagement	Summarizes key metrics about user engagements and sessions

Report name	What it is
Retention	Summarizes key metrics about active users and user retentions
Device	Summarizes key metrics about the devices users are using to access your apps and websites, and provides detail view of each device
Details	This report allows you to query and view user's attributes and the events the user performed.

Custom report

If you want to investigate certain pieces of data further, you can write SQL to create views in Redshift or Athena, then add dataset into QuickSight to create visualization. Refer to [this example](#) to learn how to create a customize report with Redshift.

Create Dashboard

You can create a custom dashboard to save the result of exploration query. Below are the steps:

1. Select **Create Dashboard** button on the top-left.
2. Fill in a name as **Dashboard name** .
3. Fill in a description for the dashboard.
4. Enter a sheet name, then click on the + button on the right. You can add multiple sheets in one dashboard.
5. You can remove a sheet by click on the X button on the sheet name.
6. Select **Create** button.
7. After the dashboard was created, you can select the dashboard to save query results.

Acquisition report

You can use the acquisition report to get insights into how new users arrive at your website or app for the first time, as well as the sources of everyday traffic.

Note

This article describes the default report. You can customize the report by applying filters or comparisons or by changing the dimensions, metrics, or charts in QuickSight. For more information, refer to [Visualizing data in Quick](#).

View the report

1. Access the dashboard for your application. Refer to [Access dashboard](#).
2. In the dashboard, choose the sheet with name of Acquisition.

Data sources

Acquisition report is created based on the following QuickSight datasets:

QuickSight dataset	Redshift view / table	Description
User_User_View-<app>-<project>	clickstream_acquisition_day_user_view_cnt	This dataset stores data on the number of new users and number of active users on your websites or apps for each day.
Day_Traffic_Source_User-<app>-<project>	clickstream_acquisition_day_traffic_source_user	This dataset stores data on the number of new users per each traffic source type for each day.
Day_User_Acquisition-<app>-<project>	clickstream_acquisition_day_user_acquisition	This dataset stores data on the number of new users, number of active users, number of sessions, number of engaged sessions, and number of events per each traffic source type for each day.

QuickSight dataset	Redshift view / table	Description
Country_New_User_Acquisition- <code><app></code> - <code><project></code>	clickstream_acquisition_country_new_user	This dataset stores data on the number of new users per each country and city for each day.

Dimensions

The Acquisition report includes the following dimensions.

Dimension	Description	How it's calculated
First user traffic source	The source of the traffic that acquires new users to your websites or apps (for example, google, baidu, and bing).	Traffic source is populated from utm parameters in page_url (i.e., utm_source) or traffic-source preserved attribute (i.e., _traffic_source_source), or derived from referrer url (only for web).
First user traffic medium	The medium of the traffic that acquires new users to your websites or apps (for example, organic, paid search)	Traffic medium is populated from utm parameters in page_url (i.e., utm_medium) or traffic-source preserved attribute (i.e., _traffic_source_medium), or derived from referrer url (only for web).
First user traffic campaign	The name of a promotion or marketing campaign that acquires new users to your websites or apps.	Traffic campaign is populated from utm parameters in page_url (i.e., utm_campaign) and traffic-source preserved attribute (i.e., _traffic_

Dimension	Description	How it's calculated
		source_campaign), or derived from referrer url (only for web).
First user traffic source / Medium First	The combination of traffic source and medium that acquires new users to your websites or apps.	Same as above for traffic source and traffic medium.
user traffic channel group	Channel groups are rule-based definitions of the traffic sources. The name of the traffic channel group that acquires new users to your websites or apps.	Traffic channel group is derived based on the traffic source and medium.
First user traffic clid platform	The name of platform for click id (auto-tagging from advertisement platform) that acquires new users to your websites or apps.	Traffic source clid platform is populated from clid parameter in page_url and traffic-source preserved attribute (i.e., _traffic_source_clid_platform).
First user app install source	The name of app store that acquires new users to your apps, for example, App Store, Google Store.	App install source is from traffic-source preserved attribute (i.e., _app_install_channel).
Session traffic source	The traffic source that acquires users into a new session on your websites or apps (for example, google, baidu, and bing)	Traffic source is populated from utm parameters in page_url (i.e., utm_source) or traffic-source preserved attribute (i.e., _traffic_source_source), or derived from referrer url (only for web).

Dimension	Description	How it's calculated
Session traffic medium	The traffic medium that acquires users into a new session on your websites or apps (for example, organic, paid search)	Traffic medium is populated from utm parameters in page_url (i.e., utm_medium) or traffic-source preserved attribute (i.e., _traffic_source_medium), or derived from referrer url (only for web).
Session traffic campaign	The name of a promotion or marketing campaign that acquires users into a new session on your websites or apps.	Traffic campaign is populated from utm parameters in page_url (i.e., utm_campaign) and traffic-source preserved attribute (i.e., _traffic_source_campaign), or derived from referrer url (only for web).
Session traffic Source / Medium	The combination of traffic source and medium that acquires users into a new session on your websites or apps.	Same as above for traffic source and traffic medium.
Session traffic channel group	Channel groups are rule-based definitions of the traffic sources. The name of the traffic channel group that acquires users into a new session on your websites or apps.	Traffic channel group is derived based on the traffic-source and medium.

Dimension	Description	How it's calculated
Session traffic clid platform	The name of platform for click id (auto-tagging from advertisement platform) that acquires users into new session on your websites or apps.	Traffic clid platform is populated from clid parameter in page_url and traffic-source preserved attribute (i.e., _traffic_source_clid_platform).
Geo country	The country where users are when they are using your websites or apps.	Geo location information is inferred based on user IP address.
Geo city	The city where the users are when they are using your websites or apps.	Geo location information is inferred based on user IP address.

Metrics

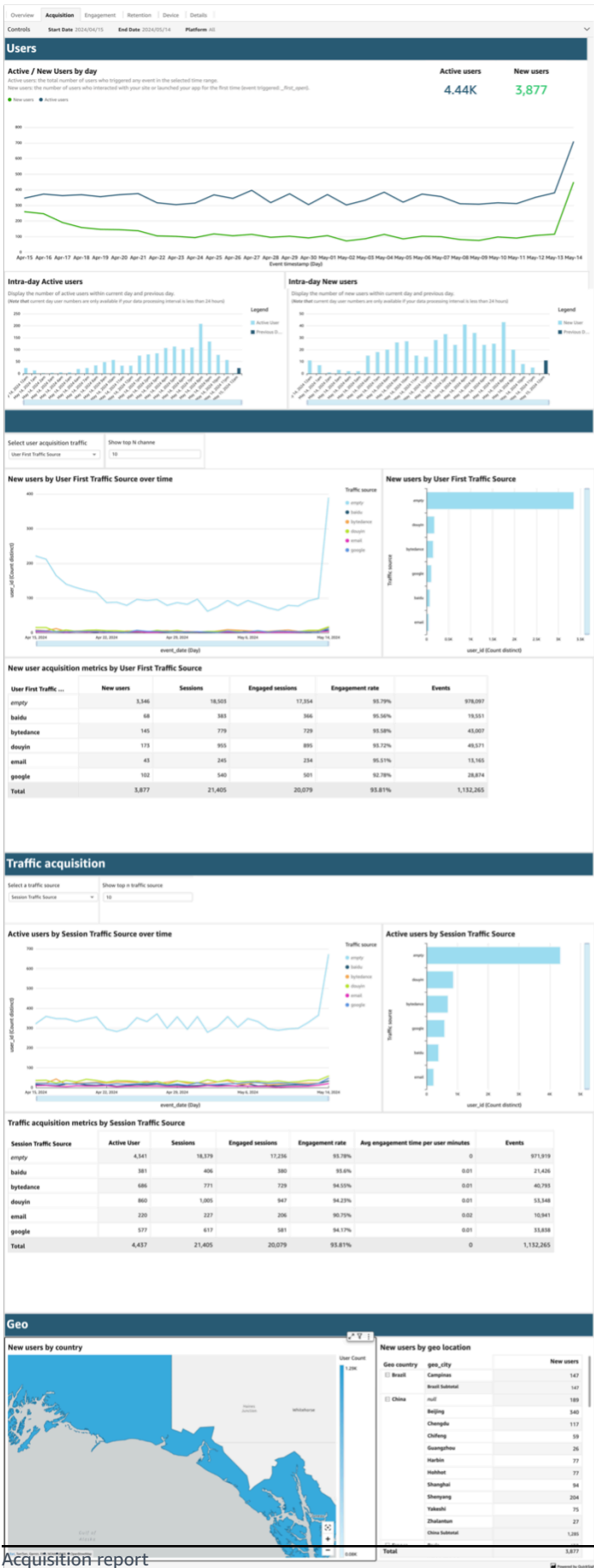
The Acquisition report includes the following metrics:

Metric	Description	How it's calculated
New users	The number of users who interacted with your site or launched your app for the first time (event triggered: _first_open).	Count distinct user_id or user_pseudo_id (if user_id is not available) when event_name equals '_first_open'.
Active users	The number of distinct users who triggered any event in the selected time range.	Count distinct user_id or user_pseudo_id (if user_id is not available) at any event
Sessions	The number of sessions users created.	Count distinct session_id.
Engaged Session	The number of sessions that lasted 10 seconds or longer,	Count distinct session_id if the session is engaged.

Metric	Description	How it's calculated
	or had 1 or more page or screen views.	
Engaged Rate	The percentage of sessions that were engaged sessions.	Engaged sessions / total sessions.
Events	The number of times users triggered an event.	Count event_id
Avg_engagement_time_per_user	The average time per user that your website was in focus in a user's browser or an app was in the foreground of a user's device.	Total user engagement durations / Number of active users Count distinct user_id or user_pseudo_id (if user_id is not available) when event_name equals '_first_open'.

Sample dashboard

Below image is a sample dashboard for your reference.



Engagement report

You can use the Engagement report to get insights into how users interact with your websites and apps. It shows metrics around user engagement levels, activities performed, and which pages/screens are most visited.

Note

This article describes the default report. You can customize the report by applying filters or comparisons or by changing the dimensions, metrics, or charts in QuickSight. For more information, refer to [Visualizing data in QuickSight](#).

View the report

1. Access the dashboard for your application. Refer to [Access dashboard](#).
2. In the dashboard, choose the sheet with name of Engagement.

Data sources

Engagement report are created based on the following QuickSight datasets:

QuickSight dataset	Redshift view / table	Description
Engagement_KPI-<app>-<project>	clickstream_engagement_kpi	This dataset stores data on Engagement KPIs per day.
Day_Event_View_Engagement-<app>-<project>	clickstream_engagement_day_event_view	This dataset stores data on the number of events and number of view events for each day
Event_Name-<app>-<project>	clickstream_engagement_event_name	This dataset stores data on the number of events per event name for each user for each day.

QuickSight dataset	Redshift view / table	Description
Page_Screen_View-<app>-<project>	clickstream_engagement_page_screen_view	This dataset stores data on the number of views per each page or screen for each day.
Page_Screen_View_Detail-<app>-<project>	clickstream_engagement_page_screen_detail_view	This dataset stores data on the view event per page title/page url or screen name/screen id for each user for each day.

Dimensions

The report includes the following dimensions:

Dimension	Description	How it's calculated
Event name	Name of the event triggered by users	Derived from the event name you set for an event with Clickstream SDK or HTTP API.
Page title	Title of the web page	Page title derives from the title tag in your HTML.
Page URL path	The path in the web page URL	Page path derives from the value after the domain. For example, if someone visits <code>www.example.com/books</code> , then <code>example.com</code> is the domain and <code>/books</code> is the page path.
Screen name	Title of the screen	Screen name derives from the name you set for a screen using clickstream SDK or HTTP API .

Dimension	Description	How it's calculated
Screen class	The class name of the screen	Screen class derives from the class name of the UIViewController or Activity that is currently in focus.

Metrics

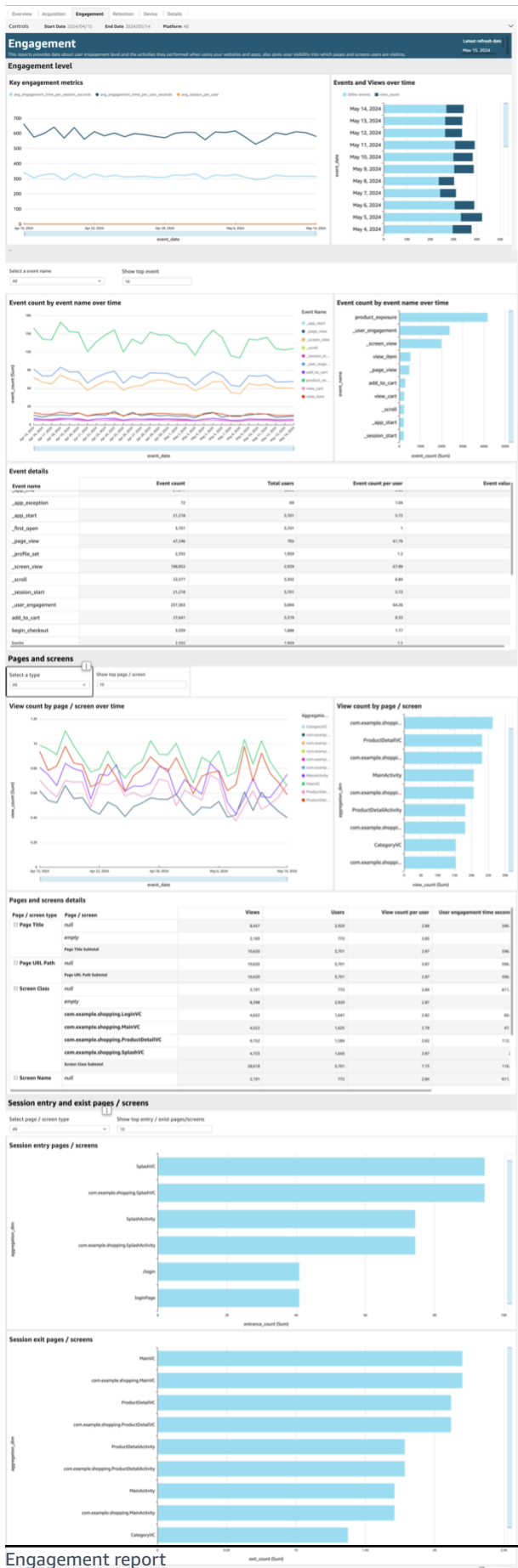
The report includes the following metrics:

Metric	Description	How it's calculated
Avg_session_per_user	The average number of sessions per active user.	Total number of sessions / total number of active users
Avg_engagement_time_per_user_minute	The average time per user that your website was in focus in a user's browser or an app was in the foreground of a user's device.	Total user engagement durations / Number of active users
Avg_engagement_time_per_session_minute	The average time per session that your website was in focus in a user's browser or an app was in the foreground of a user's device	Total user engagement durations / Number of sessions
Active users	Number of active users that had page_view or screen_view events. .	Count distinct user_id or user_pseudo_id (if user_id is not available) when event_name is '_page_view' or '_screen_view'.
Event count	The number of times users triggered a '_page_view' or '_screen_view' event.	Count event_id when event_name is '_page_view' or '_screen_view'.

Metric	Description	How it's calculated
Event count per user	Average event count per user	Event count / Active users

Sample dashboard

Below image is a sample dashboard for your reference.



Retention report

You can use the Retention report to get insights into how frequently and for how long users engage with your website or mobile app after their first visit. This helps you understand how effective your product is at attracting users to return after their initial visit.

Note

This article describes the default report. You can customize the report by applying filters or comparisons or by changing the dimensions, metrics, or charts in QuickSight. For more information, refer to [Visualizing data in QuickSight](#).

View the report

1. Access the dashboard for your application. Refer to [Access dashboard](#).
2. In the dashboard, choose the sheet with name of Retention.

Data sources

Retention report are created based on the following QuickSight dataset:

QuickSight dataset	Redshift view / table	Description
User_New_Return-<app>-<project>	clickstream_retention_user_new_return	This dataset stores data on the number of new users and number of returning users for each day
Retention_View-<app>-<project>	clickstream_retention_view_v3	This dataset stores data on the cohort of user return by their first visit date and days since their first visit.
Event_Overtime-<app>-<project>	clickstream_retention_event_overtime	This dataset stores data on daily event counts.

QuickSight dataset	Redshift view / table	Description
DAU_MAU-<app>-<project>	clickstream_retention_dau_wau	This dataset stores the user_ids of active users for each day.
Lifecycle_Weekly_View-<app>-<project>	clickstream_retention_dau_wau	This dataset stores data on the number of new users, number of retained users, number of return users, and number of lost users for each week.

Metrics

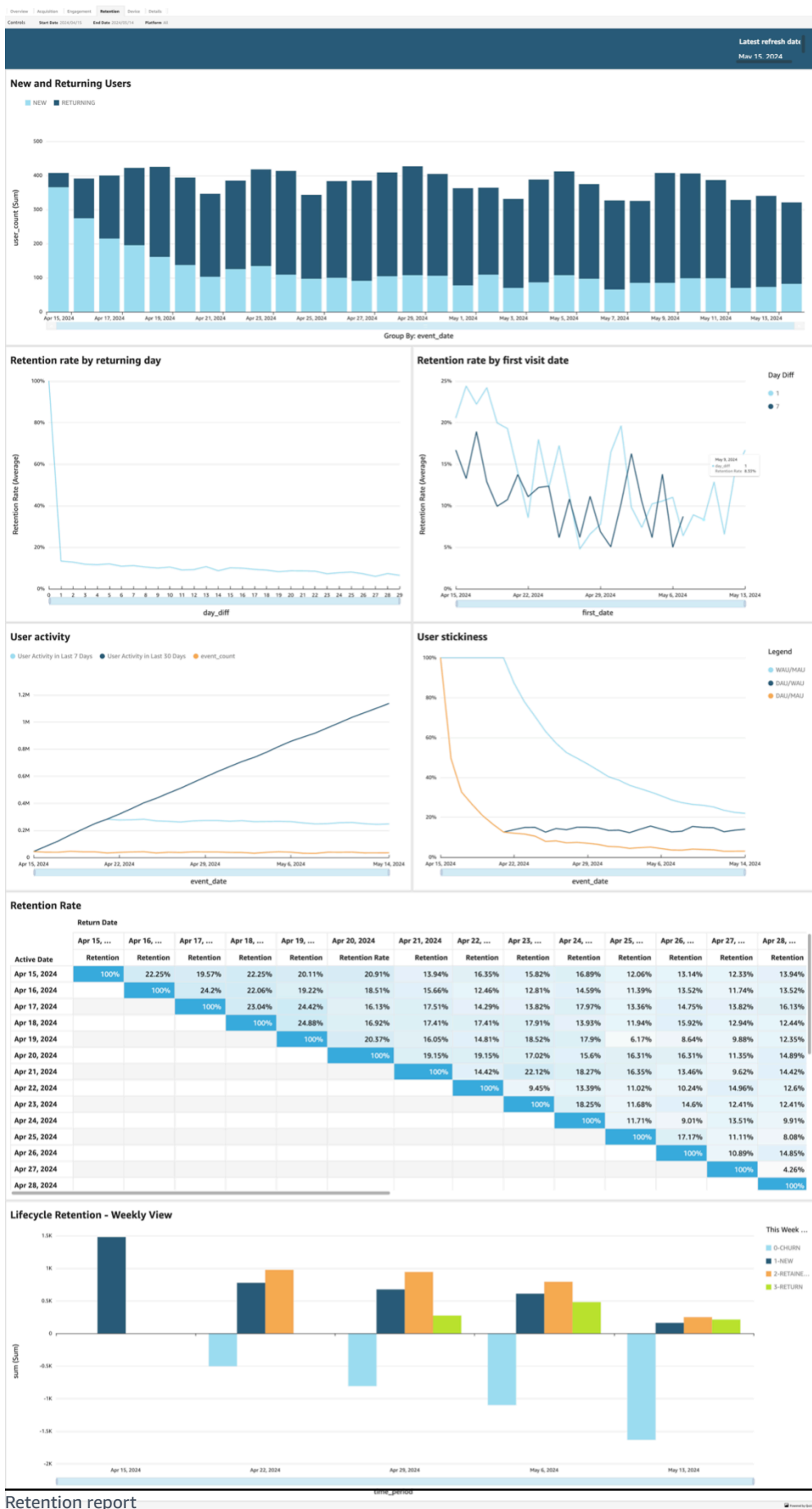
The report includes the following metrics:

Metric	Definition	How it's calculated
New users	The number of users who visited your website or app for the first time.	Count distinct user_id or user_pseudo_id (if user_id is not available) when event_name equals '_first_open'.
Returning users	The number of users who have visited your website or app before.	Count distinct user_id or user_pseudo_id (if user_id is not available) when user_first_visit_date does not equal event_date.
Returning rate	The percentage of users returned over total users in that cohort.	Returning users / total users.

Metric	Definition	How it's calculated
Lifecycle - New	Number of new users in the week.	Number of users that triggered no events in previous weeks.
Lifecycle - Retained	Number of users are active in this week and were active in last week.	Number of users that triggered events in previous week and this week.
Lifecycle - Return	Number of users visited before and were not active in previous week, but are active in this week (excludes new users).	Number of users that triggered events in at least two weeks before and are active in this week.
Lifecycle - Churn	Number of users were active in previous week, but are not active in this week.	Number of users that triggered events in previous week but triggered no event this week.

Sample dashboard

Below image is a sample dashboard for your reference.



Device report

The Device Report provides insights into the devices, operating systems, and browsers used by your app or website users. This data helps you understand your user base and optimize for the most common configurations.

Note

This article describes the default report. You can customize the report by applying filters or comparisons or by changing the dimensions, metrics, or charts in QuickSight. For more information, refer to [Visualizing data in Quick](#).

View the report

1. Access the dashboard for your application. Refer to [Access dashboard](#).
2. In the dashboard, choose the sheet with name of Device.

Data sources

Device reports are created based on the following QuickSight dataset:

Retention report are created based on the following QuickSight dataset:

QuickSight dataset	Redshift view / table	Description
Device-<app>-<project>	clickstream_device_user_device_view	This dataset stores data on the device information per each user for each day.
Crash_Rate-<app>-<project>	clickstream_device_crash_rate	This dataset stores data on the crash rate for each app version for each day.

Dimensions

The report includes the following dimensions:

Dimension	Description	How it's calculated
App version	The version of the app	Derived from the <code>app_version</code> field automatically collected by Clickstream SDK or manually set by HTTP API.
Device	The device name associated with the event.	For mobile app, it is derived from <code>model_name</code> field automatically collected by Clickstream SDK or manually set in HTTP request. For web, it is the <code>device_ua_device</code> that parsed from User Agent.
Device screen resolution	The screen resolution of the device	Concatenation of <code>screen_width</code> and <code>screen_height</code> values
Browser	The name of the browser	From the value of <code>device_ua_browser</code> field which parsed from the User Agent.
Operating system	The operating system of the device.	For mobile app, it is derived from <code>platform</code> field automatically collected by Clickstream SDK or manually set in HTTP request. For web, it is the <code>device_ua_operating_system</code> that parsed from User Agent.
Operating system version	The operating system version of the device.	For mobile app, it is derived from <code>os_version</code> field automatically collected by Clickstream SDK or manually set in HTTP request. For web, it is the <code>device_ua_operatin</code>

Dimension	Description	How it's calculated
		g_system_version that parsed from User Agent.

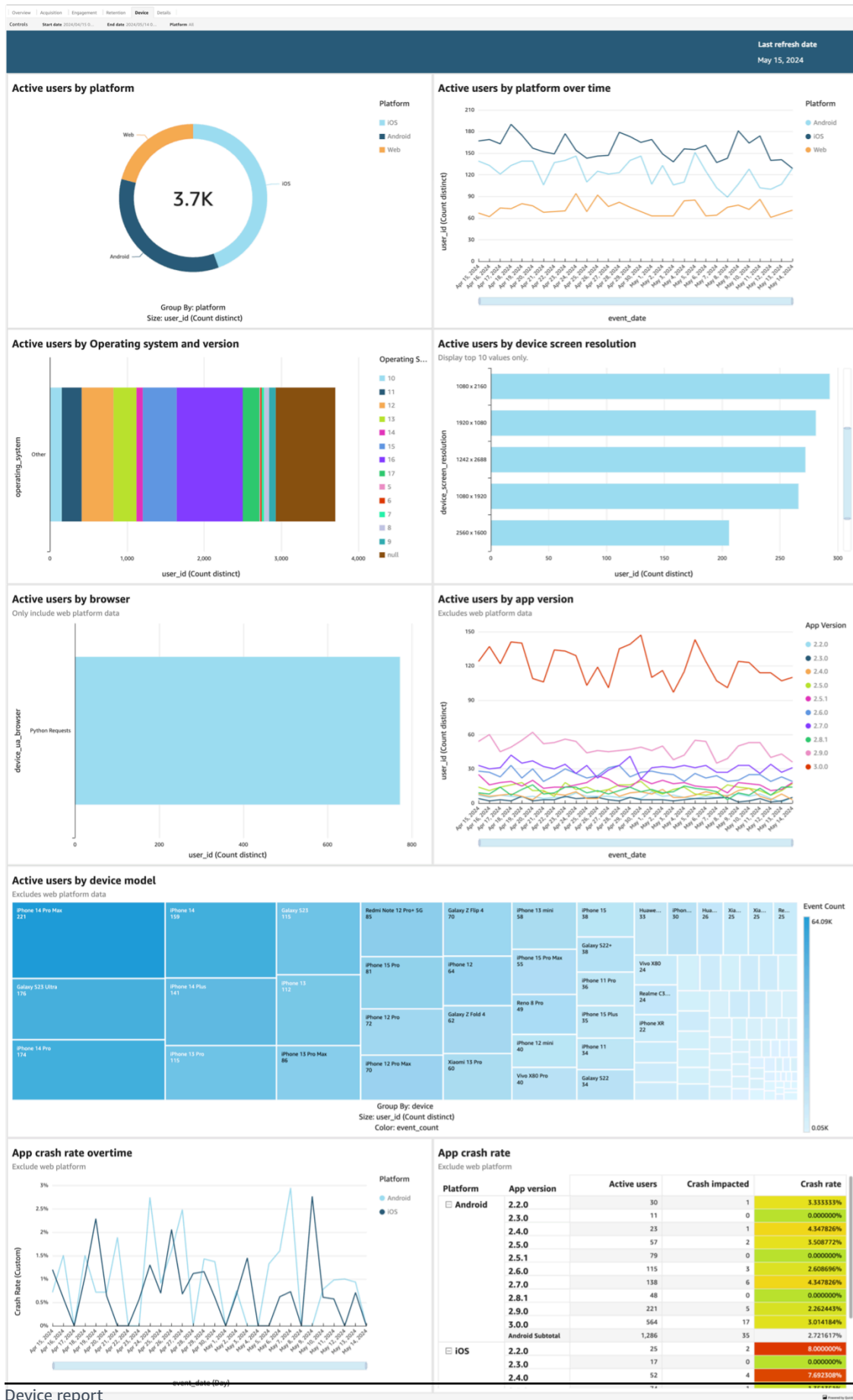
Metrics

The report includes the following metrics:

Metric	Definition	How it's calculated
Active users	Number of active users that had triggered any event.	Count distinct user_id or user_pseudo_id (if user_id is not available).
Crash rate	The percentage of the users experiencing app.	Number of distinct user with '_app_exception' events / Total active users

Sample dashboard

Below image is a sample dashboard for your reference.



Details report

You can use the Details report to view common and custom dimensions for individual events, query all user attributes for a specific user, and see the events that a particular user has performed

Note

This article describes the default report. You can customize the report by applying filters or comparisons or by changing the dimensions, metrics, or charts in QuickSight. For more information, refer to [Visualizing data in Quick](#).

View the report

1. Access the dashboard for your application. Refer to [Access dashboard](#).
2. In the dashboard, choose the sheet with name of Details.

Data sources

User reports are created based on the following QuickSight dataset:

QuickSight dataset	Redshift view / table	Description
Event_View-<app>-<project>	clickstream_event_view_v3	This datasets stores all the raw event data joined with user attributes and session attributes.

Dimensions

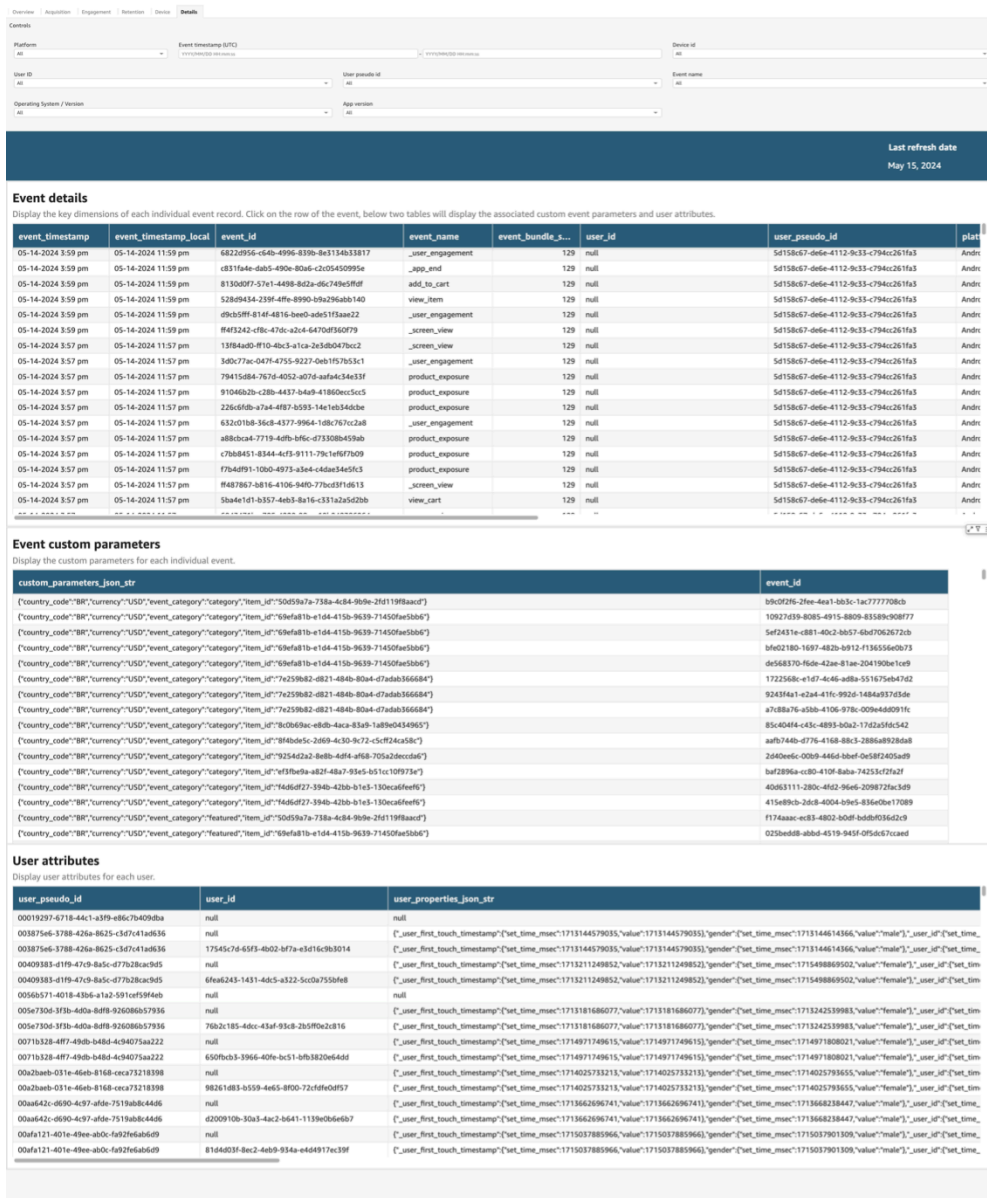
This report includes all the available dimensions for event, user, and session tables. Please refer to [Data Schema](#) for each dimensions.

Metrics

This report does not any metrics by default.

Sample dashboard

Below image is a sample dashboard for your reference.



Custom report

One of the key benefits of this guidance is that you have complete control over the clickstream data collected from your apps and websites. You have complete flexibility to analyze the data for your specific business needs. This article illustrates the steps of creating a custom report with an example of creating funnel analysis by using Redshift Serverless as analytics engine and QuickSight as reporting tools.

There are two options to create custom analyses. The following examples will guide you through the detailed steps.

Option 1- create custom analysis with preset QuickSight dataset

The guidance had created a `Event_View-<app>-<project>` dataset which stores all the raw event data, includes both common and custom dimension, joined with user attributes and session attributes in QuickSight, you can use these datasets to create custom analysis based on your specific requirements.

For example, let's say you have a custom event `view_item` that records when a user views an item's detail page. This custom event has a custom attribute `event_category` to track the promotion spot from which the user came to the item detail page. Now, you want to create a custom analysis that uses a pie chart to show the percentage of each `event_category` that led user to item detail pages.

1. Go to **Analyze module** in Clickstream Analytics on AWS web console.
2. Click on **New analysis** in the top-right corner.
3. Select `Event_View-<app>-<project>` dataset.
4. Select the **USE IN ANALYSIS** button.
5. Since the `event_category` is a custom parameter for the `view_item` event, we need to extract it from the `custom_parameters_json_str` field, which is a JSON string, we can use `parseJSON` function to extract values from it. Select **+ CALCULATED FIELD** button to add a calculated field.
6. Input **Event Category** as the name for the calculated field, and input `parseJson({custom_parameters_json_str}, "$.event_category")` as the formula, then select **Save**.
7. In the analysis author console, you can see a new field called Event Category appear in the Data panel.
8. In the Visuals panel, click on **+ ADD** button, and select Pie chart.
9. Drag the Event Category field to **GROUP/COLOR**, drag `event_id` to **VALUE**.
10. In the Filter panel, add a filter to only include `event_name` equals `view_item`.
11. Now you should be able to see a bar chart shows the percentage of each `event_category` that led to `view_item` events.
12. You can format the analysis according to your need then publish it as dashboard.
13. To enable your custom analysis to appear in the Dashboards module of the Clickstream Analytics Studio, you need to add the dashboard into the Shared folder with name of the `<project-`

id>_<app_id>, which was pre-created by the guidance. After you added the custom dashboard into the Shared folder, it will automatically display in the Clickstream Analytics Studio.

Option 2 - create custom view in Redshift and import to QuickSight

Part 1 - Dataset preparation

1. Open **Redshift Serverless dashboard**.
2. Choose the workgroup starting with `clickstream-<project-id>` created by the guidance.
3. Choose **Query data**. You will be directed to the Redshift Query Editor.
4. In the **Editor** view on the Redshift Query Editor, right click on the workgroup with name of `clickstream-<project-id>`. In the prompted drop-down, select **Edit connection**, and you will be asked to provide connection parameters. Follow this [guide](#) to use an appropriate method to connect.

Important

Read and write permissions are required for the database (with name as `<project-id>`) to create custom view or table. For example, you can use Admin user to connect to the cluster or workgroup. If you don't know the password for the Admin user, you can reset the admin password in the Redshift Console. For more information, refer to [Security and connections in Amazon Redshift Serverless](#).

5. If it is the first time you access the query editor, you will be prompted to configure the account. Choose **Config account** to open query editor.
6. Add a new SQL editor, and make sure you selected the correct workgroup and schema.
7. Create a new view for funnel analysis. In this example, we used below SQL.

```
CREATE OR REPLACE VIEW {{schema}}.clickstream_funnel_view as
SELECT
platform,
COUNT(DISTINCT step1_id) AS session_start_users,
COUNT(DISTINCT step2_id) AS page_view_users,
COUNT(DISTINCT step3_id) AS scroll_users
FROM (
SELECT
    platform,
    user_pseudo_id AS step1_id,
```

```
    event_timestamp AS step1_timestamp,  
    step2_id,  
    step2_timestamp,  
    step3_id,  
    step3_timestamp  
FROM  
    {{schema}}.clickstream_event_base_view  
LEFT JOIN (  
SELECT  
    user_pseudo_id AS step2_id,  
    event_timestamp AS step2_timestamp  
FROM  
    {{schema}}.clickstream_event_base_view  
WHERE  
    event_name = '_page_view')  
ON  
    user_pseudo_id = step2_id  
    AND event_timestamp < step2_timestamp  
LEFT JOIN (  
SELECT  
    user_pseudo_id AS step3_id,  
    event_timestamp AS step3_timestamp  
FROM  
    {{schema}}.clickstream_event_base_view  
WHERE  
    event_name= '_scroll' )  
ON  
    step3_id = step2_id  
    AND step2_timestamp < step3_timestamp  
WHERE  
event_name = '_session_start' )  
group by  
platform
```

8. Go to QuickSight console, choose **Dataset**, and then choose **New dataset**.
9. In the New Dataset page, choose **Redshift Manual connect** to add dataset, and fill in the prompted form with the following parameters.
 - **Data source name:** clickstream-funnel-view-<project-id>
 - **Connection type:** select VPC connections / VPC Connection for Clickstream pipeline <project-id>
 - **Database server:** input the endpoint url of the serverless workgroup, which you can find on the workgroup console.

- **Port:** 5439
- **Database name:** <project-id>
- **User name:** name of the user you used to created the custom view in previous steps
- **Password:** password of the user you used to created the custom view in previous steps

10. Validate the connection, and then choose **Create data source**.

11. Choose the view from Redshift as data source - "**clickstream_funnel_view**", then

- Schema: select notepad
- Tables: clickstream_funnel_view

 **Note**

When prompted to select Import to SPICE or Directly query your data, select **Directly query your data** for this example.

- Choose **Edit/Preview data** to preview the data. Once you're familiar with the data, choose **PUBLISH & VISUALIZE** at the top-right.

Part 2 - Create visualizations in QuickSight

1. When prompted, select a layout for your visualization.
2. Choose "**+Add**" at the top-left of the screen then choose "**Add visual**".
3. Select a Visual type at the bottom-left of the screen, in this example, select **Vertical bar chart**.
4. In the Field wells, select platform as X axis, login_user, add_button_click_users, and note_create_users as Value.

Now you can publish this analysis as dashboard or continue to format it. For more information, see [Visualizing data in Amazon QuickSight](#).

Part 3 - Add the custom dashboard to Analytics Studio

To enable your custom dashboard to appear in the Dashboards module of the Analytics Studio, you need to add the dashboard into the Shared folder with the name of <project-id>_<app_id>, which was pre-created by the guidance.

Exploration

The Exploration module provides a collection of advanced analytics models with flexible and easy-to-use query method, allowing you to derive deeper insights from your clickstream data.

When you want to explore the clickstream data in more details, you can use explorations to:

- perform ad hoc queries
- focus on the most relevant data by using filters on event parameters and user attributes
- group metrics by dimension to make it easy to compare
- easily switch visualization type and drill down or up into data
- view and export summary data into the excel or csv
- save the explorative analytics results into a dashboard for share or regular use

Access Exploration

You can create an ingestion module with the following settings:

To access Explorations, follow below steps:

1. Go to Clickstream Analytics on AWS Console, in the **Navigation Bar**, choose **Analytics Studio**.
2. In the Analytics Studio page that opens, choose **Explorations** in the left navigation panel.

How Exploration works

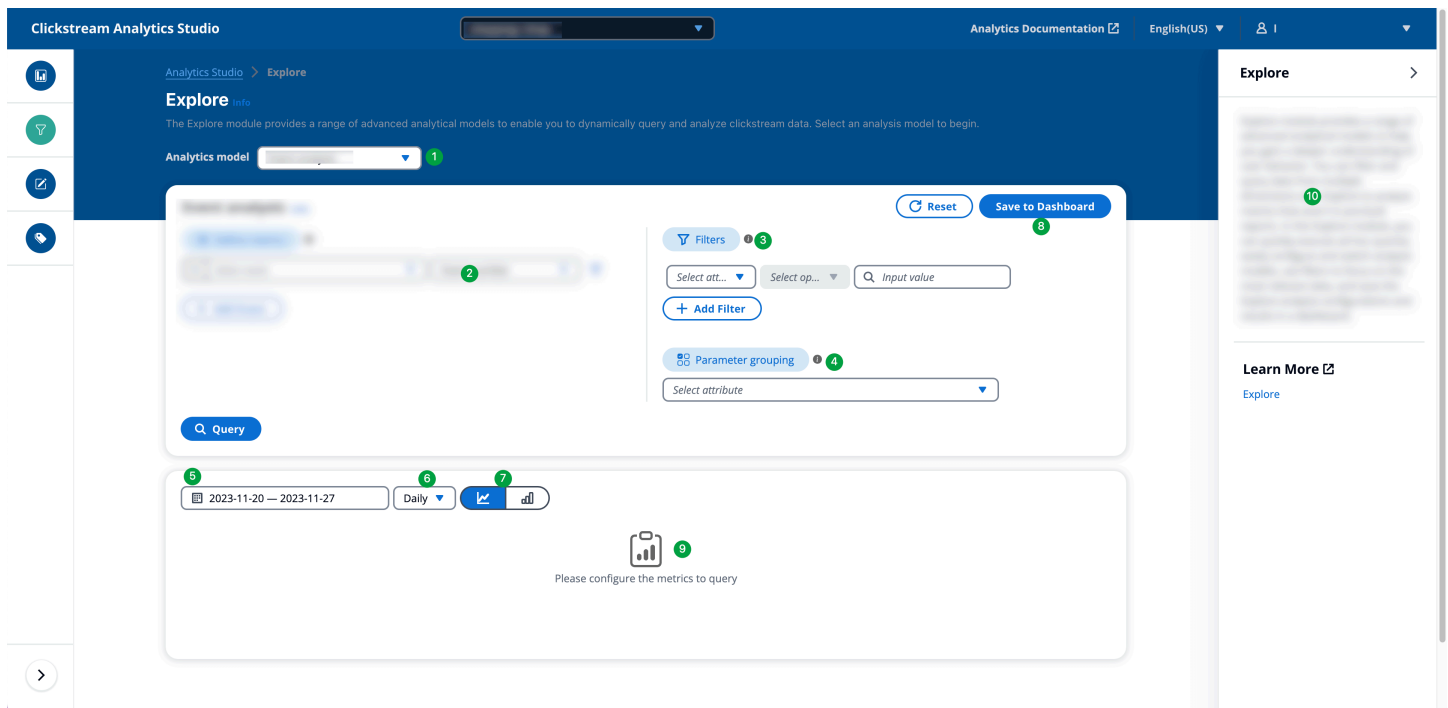
Currently, exploration provides 4 analytics models:

Report name	Description
Event Analysis	Event analysis is used to study the frequency of certain behavioral events. You can conduct multi-dimensional analysis of user behavior through custom metrics, groupings, filters, and various visual charts.
Funnel Analysis	Funnel analysis, or conversion analysis, is mainly used to analyze the conversion status

Report name	Description
	<p>of users in a specified process. The model first breaks down the entire process into steps and then counts the conversion rate from each step to the next. It can be used to measure the performance of each step. Common usage scenarios include analyzing registration conversion rates, purchase conversion rates, etc.</p>
Path Analysis	<p>Path analysis refers to the distribution of the behavior of a group of users after sorting them in sequential order. It is mainly used to analyze and record the distribution ratio of users between pages or events. For example, check how many customers clicked on the product list after opening the app, how many customers viewed the product detail page after visiting landing page, how many customers searched for the product, etc.</p>
Retention Analysis	<p>Retention analysis supports configuring initial event and returning event to calculate the retention or attrition rate of target user groups. It also supports setting associated attributes for initial event and return event.</p>
Attribution Analysis	<p>Attribution analysis allows you to understand the importance of different touchpoints for a specific goal in your website and apps.</p>

How to use Exploration

Explorations interface consists of the following components:



- **Analytics Model.** A drop-down list to select or switch analytics model.
- **Model Configuration.** Specifies the configurations for the analytics model, such as select event, adding filters. Each model might have different configuration.
- **Global Filters.** Filters that apply to all the metrics that defined in the Model Configuration. You can add multiple filters and adjust the filter relationship (i.e., 'And' or 'Or').
- **Grouping.** Group the analytics result by specified parameter's value, make it easy for you can compare the metrics at a dimension.
- **Analysis Time Range.** Specifies the time range for the analysis.
- **Aggregation Granularity.** Specifies the level of granularity to display metrics, such by day, week, or month.
- **Visual Type.** Specifies the chart type of the visualization.
- **Save to Dashboard.** Save the exploration analysis
- **Result Display Area.** Show visualization and detail data.
- **Help Panel.** Display additional helpful info when click "Info" icon.

Event Analysis

Event analysis is used to study the frequency of certain behavioral events. You can conduct multi-dimensional analysis of user behavior through custom metrics, groupings, filters, and various visual charts.

Use cases

Event analysis are commonly used when analyzing user behaviors, for example:

- Query on user usage of certain product functions, such as adding favorite, video playback, and view live stream;
- Compare different groups of user behaviors, such as the number of logins per country;
- Compare different channel's effectiveness, such as sign-up rate per traffic source.

Key concept

- **Metric:** perform aggregation on a selected event, such as the number of events, or the number of distinct users generating the event.

How to use event analysis

1. Select an event, and select the aggregation method for the metric.
2. Add filter to the event by clicking the icon next to the metric.
3. Select an event parameter or user attribute as filter. You can add multiple filters by clicking on the filter icon. You can also configure the filter relationship by choosing AND or OR.
4. Repeat above step to add more metric if needed.
5. If needed, configure global filter by selecting event parameter or user attributes. Similar to event filter, you can add multiple global filters and configure the filter relationship.
6. If needed, configure grouping by selecting an event parameter or user attribute.
7. Choose **Query** to start the analysis.
8. Adjust the data granularity, such as Daily, Weekly, Monthly, if needed.
9. Adjust query time range if needed.
10. Choose **Save to Dashboard** to save the analysis to a Dashboard, enter a name, description, and select a dashboard and sheet.

Example

Calculate the daily page views (PV) and active user count (UV) on the web from different countries over the past month, requiring active users to have a session duration of at least 30,000 milliseconds.

Steps

1. Select the **Event Analysis** model.
2. In the left **Define Metrics** area, choose `_page_view` as the metric for calculating events and select Event number as the metric type.
3. Click the **+ Add Event** button to add another metric. Choose `_app_end` as the metric for calculating events and select User number as the metric type.
4. Click the filter icon to the right of `_app_end` to add a event filter condition:
 - Filter property: Session / Session duration(msec)
 - Operation: >=
 - Value: 30000 (the unit of Session duration(msec) is millisecond)
5. Configure a global filter in the right **Filters** area:
 - Choose Event preset / Platform as the filter property.
 - Operation: =
 - Value: Web
6. In the right **Attribute Grouping** area, configure grouping by selecting Geography/Country.
7. In the time selector at the bottom, choose Past Month and click **OK**.
8. Click the **Save to Dashboard** button in the top right corner. In the pop-up dialog, enter:
 - Chart Name: PV and UV
 - Chart Description: PV and UV on the web over the past month (at least 30 seconds)
 - Choose a Dashboard: Select a dashboard. (You need to create a dashboard first. For more information, see [Create dashboard.](#))
 - Choose a Worksheet: Select a worksheet.
 - Click **OK**.

All configurations are shown below:

Funnel Analysis

Funnel analysis, or conversion analysis, is mainly used to analyze the conversion status of users in a specified process. The model first breaks down the entire process into steps and then counts the conversion rate from each step to the next. It can be used to measure the performance of each step.

Use cases

Funnel analysis is commonly used when analyzing the following user behaviors:

- Analysis of the conversion rate of the key process in a product such as order to purchase rate, and registration completion rate;
- Analysis of the conversion rate of promotion such as conversion rate of different in-app promotion spot;
- Analysis of marketing channels's effectiveness such as purchase rate of new users brought by different ad campaigns.

Key concepts

- **Metric:** The entity used for funnel analysis, such as event number or user number.

- **Funnel:** A funnel is a sequence of events that represents a process. It contains at least two events, and each event represents a step in the funnel.
- **Funnel window:** A funnel window refers to the time for the user to complete the entire process. It is considered a successful conversion only when the user completes all the selected steps within the set window period.

How to use funnel analysis

1. Select a metric type.
 - a. User number: calculate the number of distinct users passing through the entire funnel.
 - b. Event number: calculate the number of completions of the entire funnel.
2. Configure the funnel window.
 - a. Custom: you can define any duration as the funnel window.
 - b. The day: complete the funnel within the same date of the first step.
3. Select event for as the step. If needed, choose **+Add Step** to add more steps. You can add up to 10 steps.
4. Choose the filter button to filter the event. Only the events meeting the filter criterial will be considered as passing through the funnel. You can add multiple filters to one event.
5. If needed, configure global filter by selecting event parameter or user attributes. Similar to event filter, you can add multiple global filters and configure the filter relationship.
6. If needed, configure grouping by selecting an event parameter or an user attribute.

Note

The Funnel visualization does not support grouping. If you need to group funnel result, please select bar chart.

1. If you want to only apply the grouping on the first event, toggle on Apply grouping to first step only. If this option is not selected, the grouping will apply to all the steps in the funnel, which means all the events should have parameter or attributes that used to group.
2. Choose **Query** to start the analysis.
3. Adjust the data granularity, such as Daily, Weekly, Monthly, if needed.
4. Adjust query time range if needed.

5. Choose **Save to Dashboard** to save the analysis to a Dashboard. Enter a name, description, and select a dashboard and sheet.

Example

Calculate the conversion rate of users on the web from opening the website -> viewing the product details page -> adding to the shopping cart -> making a payment over the past week.

1. Select the **Funnel Analysis** model.
2. Choose User number as the metric.
3. In the left **Define Funnel** area, choose The Day as the funnel window.
4. Choose `_session_start`, `view_item`, `add_to_cart`, `purchase` as funnel events.
5. Configure a global filter in the right **Filters** area:
 - Choose Event preset / Platform as the filter property.
 - Operation: =
 - Value: Web
6. Click **Query**.

All configurations are as shown in the image below:

The screenshot displays the Clickstream Analytics Studio interface for configuring a Funnel Analysis. The top navigation bar shows the current workspace as 'shopping / shop' and includes links to 'Analytics Documentation' and 'English(US)'. The main configuration area is titled 'Funnel analysis' and includes a 'Reset' button and a 'Save to Dashboard' button.

The configuration is divided into several sections:

- Analytics model:** Set to 'Funnel analysis' (1).
- Metrics:** Set to 'User number' (2).
- Funnel window:** Set to 'The Day' (3).
- Funnel steps:** A list of four steps: 1. '_session_start', 2. 'view_item', 3. 'add_to_cart', and 4. 'purchase' (4).
- Filters:** A filter is configured with the property 'other.pl...', the operation '=', and the value 'Web' (5).
- Query:** A 'Query' button is visible (6).

At the bottom of the interface, there is a 'Daily' dropdown menu, a 'Last 7 day' range selector, and a 'Please configure the metrics to query' message.

Path Analysis

Path analysis refers to the distribution of the behavior of a group of users after sorting them in sequential order. It is mainly used to analyze and record the distribution ratio of users between pages or events. For example, you may want to check how many customers clicked on the product list after opening the app, how many customers viewed the product detail page after visiting landing page, how many customers searched for the product.

Use cases

Path analysis is commonly used when analyzing user navigation pattern.

- Analyze the behavioral path distribution of users after entering the product or arriving at landing page
- Analyze the transition steps or screen within a specific processes (such as registration, login, payment) or product modules

Key concepts

- **Metric:** the entity used for path analysis, such as event number or user number.
- **Session:** the length of the path analysis session. Only the events happened within the session can form a path, events happened in different path will not be counted in the same path.
- **Node:** a node in the path can be an event, a screen, or a page.
- **Starting node:** the starting point of the path analysis.

How to use path analysis

1. Select a metric type.
 - a. User number: calculate the number of distinct users passing through the entire path.
 - b. Event number: calculate the number of completion of the entire path.
2. Configure the session.
 - a. Session ID: Use the session generated by the SDK as the session for path analysis. Only the events happened with the same session ID could be counted as a path.
 - b. Custom: you can define any duration as the funnel window.
3. Specify the type of node for the path.

- a. Event name: use event as a node, for example, `_page_view`, `screen_view`.
 - b. Screen name: use the name of a screen in the App as a node.
 - c. Screen ID: use the class ID of a screen in the App as a node.
 - d. Page title: use the page title as a node.
 - e. Page url: use the page url as a node.
4. Set a starting node. The start node will be the first node of the path. Only events happened after the starting node will be counted into the path.
 - a. Show all nodes: display all nodes that follow the starting node.
 - b. Show selected nodes: display the selected nodes only, other nodes could be grouped as 'others'.
 5. Select nodes to participate in the analysis. Only nodes selected will be highlighted as nodes in the path. You can add up to 10 nodes.
 6. Specify if you want to include other nodes in the path. If this option is toggled on, all the nodes that are not selected will be counted and display as "Other" in the path.
 7. Specify if you want to merge consecutive nodes. If this option is toggled on, nodes that are repeated continuously in the same session will be merged into a single node.
 8. If needed, configure global filter by selecting event parameter or user attributes. Similar to event filter, you can add multiple global filters and configure the filter relationship.
 9. Choose **Query** to start the analysis.
 - 10 Adjust query time range if needed.
 - 11 Choose **Save to Dashboard** to save the analysis to a Dashboard. Enter a name, description, and select a dashboard and sheet.

Example

Calculate the distribution of events triggered by customers on the web after the session starts, focusing on login, registration, product exposure, search, viewing products, adding to cart, starting checkout, and placing an order, as well as all other events.

1. Select the **Path Analysis** model.
2. Choose User number as the metric.
3. Choose Session ID as the session definition.

4. Choose Event as the node type.
5. In the left **Select Nodes** area, choose `_session_start` as the starting node.
6. Choose `sign_up`, `login`, `product_exposure`, `search`, `view_item`, `add_to_cart`, `begin_checkout`, `purchase` in sequence as path nodes.
7. Enable **Include Other Events**.
8. Enable **Merge Consecutive Events**.
9. Configure a global filter in the right **Filters** area:
 - Choose `other.platform` as the filter property.
 - Operation: `=`
 - Value: `Web`
10. Choose **Query**.

All configurations are as shown in the image below:

The screenshot displays the Clickstream Analytics Studio interface for configuring a Path analysis. The main configuration panel is titled "Path analysis" and includes the following settings:

- Analytics model:** Path analysis (1)
- Metrics:** User number (2)
- Session definition:** Session ID (3)
- Node type:** Event name (4)
- Set start node:** `_session_start` (5)
- Select nodes to participate in the analysis:** A list of nodes including `_session_start`, `sign_up`, `login`, `product_exposure`, `search`, `view_item`, `add_to_cart`, `begin_checkout`, and `purchase`. The `login` node is highlighted with a green circle (6).
- Filters:** A filter configured with property `other.pl...`, operation `=`, and value `Web` (9).
- Including Other Events:** Yes (7)
- Merge Consecutive Events:** Yes (8)
- Query:** A button labeled "Query" (10)

Retention Analysis

Retention rate is a common metric used to assess the user stickiness for an app or website.

Retention means that the user returns to your app or website again some time after they used your app. In addition to the standard retention metrics in the default dashboard, Retention Analysis

module allows you to select a start event and returning event to customize a retention or attrition rate of a target user group.

Use cases

Retention analysis is commonly used to understand how well your app or website is doing in terms of retaining users.

- Calculate new user retention rate to measure the effectiveness of traffic channel;
- Calculate the active user retention rate to measure the effectiveness for a promotion campaign;
- Compare the repurchase rate for different groups of users to identify the most valuable customers.

Key concept

- **Start:** the event indicates that users start using the app or website.
- **Revisit:** the event indicates that users returning to the app or website.
- **Associated parameter:** Associated parameter are used to keep the value of a parameter consistent between the starting event and the return event. For example, promotion campaign name, page title, or product titles must be the same values for both starting event and return event.

Note


The two associated parameter must both have values, and the value types must be consistent.

- **Retention Rate:** retention rate refers to the relationship between the number of users who perform the specified starting event on the start date (or week, or month depending on the granularity selection) and the number of the same users who perform the specified returning event on the the return date (or week, or month).

How to use retention analysis

1. Select a **Start** event, you can add filter to the event by clicking the filter icon.

2. Select a **Revisit** event, you can add filter to the event by clicking the filter icon.
3. If needed, you can toggle on **Associate parameter**, then select parameters for both starting event and return event.
4. Repeat above step to add more metric if needed.
5. If needed, configure global filter by selecting event parameter or user attributes. Similar to event filter, you can add multiple global filters and configure the filter relationship.
6. If needed, configure grouping by selecting an event parameter or user attribute.
7. Choose **Query** to start the analysis.
8. Adjust the data granularity, such as Daily, Weekly, Monthly, if needed.
9. Specify query time range.

 **Note**

The start time will be the starting point (i.e., day 0) for the retention analysis, and the retention rate % will be calculated against the number of users who performed the specified starting event on the start date (or week, or month depends on the granularity selection).

10. Choose **Save to Dashboard** to save the analysis to a Dashboard. Enter a name, description, and select a dashboard and sheet.

Example

Calculate the retention rate of new customers who downloaded from different app markets on Android one week ago.

1. Select the **Retention Analysis** model.
2. Choose `_first_open` as the start event.
3. Choose `_app_start` as the return event.
4. Configure a global filter in the right **Filters** area:
 - Choose Event preset / Platform as the filter property.
 - Operation: =
 - Value: Android

- In the right **Attribute Grouping** area, configure grouping by selecting Application / App install source.
- Click **Query**.

All configurations are as shown in the image below:

Attribution Analysis

Attribution analysis is an analytics technique to assign credit to the touchpoint in user conversion journeys for a conversion goal. It allows you to understand the importance of different touchpoints for a specific goal in your website and apps.

Use cases

Attributions analysis are commonly used when analyzing the contribution of specific touchpoints, for example:

- Identify the most important promotion slots within an app/website in terms of their contribution to a purchase goal;
- Identify the most important traffic channels for an app/website based on the contribution to a conversion goal.

Key concept

- **Conversion goal:** A quantifiable metric that the app owner wants to achieve, e.g., number of purchase, purchase value, registration user number.
- **Touchpoint:** Events that app owner designed in the user journey to drive user towards the conversion goal, e.g., page_view, product exposure, button_click.
- **Attribution models:** Attribution models are a set of rules or data-driven algorithms used to determine how conversions are assigned to touchpoints on the conversion path. There is no one-fits-all model, choose one base on your scenario. Clickstream Analytics on AWS supports the following models:

Models	Definition	Applicable scenerio	Consideration
First-Touch Attribution	The first attribution touchpoint in completing a goal event receives 100% contribution.	For example, in the early testing phase of the homepage flash sales, the initial traffic source plays the most important roles.	Amplifies the value of the traffic source, underestimates the output of other touchpoints.
Last-Touch Attribution	The last attribution touchpoint in completing a goal event receives 100% contribution.	Helps understand which touchpoint led to the final decision for a transaction.	May underestimate certain touchpoints that are not closed to conversion event, but avoids the bias towards high-traffic touchpoints seen in the first-touch model.
Linear Attribution	All attribution touchpoints in completing a goal event evenly share the contribution	Treats each touchpoint equally, some touchpoints may consistently serve as intermediary touchpoints.	Tends to favor touchpoints clicked frequently by users, amplifying their value (e.g., in search

Models	Definition	Applicable scenerio	Consideration
	(each gets an equal share).	Evaluate if certain touchpoints with long chains can be eliminated or optimized.	and recommend ations).
Position-Based Attribution	The first and last attribution touchpoints each receive 40% contribution, while the remaining positions evenly share the remaining 20%.	Aims to distribute value across all touchpoints but emphasizes the importance of the first and last touchpoints. Acknowledges the significance of these two touchpoints compared to others.	Subject to human bias factors.

How to use attribution analysis

1. Select an event and metric as conversion goal, you can add filter. Metric types includes:
 - a. Event number: number of conversion times
 - b. SumGroup: sum a value from a numerical parameter that are associated with the event selected as conversion goal
2. Select touchpoints events, you can add filter(s) to each touchpoint.
3. Click on Query to start calculation
4. You can change the time range or attribution model according to your need, which will automatically re-run the query.
5. Analysis result is displayed in a table with the following columns:
 - a. Touchpoint Name: The touchpoint event name for attribution.

- b. **Total Trigger Count:** Number of times the touchpoint has been triggered within the specified conversion window.
 - c. **Number of Triggers with Conversion:** Within the conversion window, number of time the touchpoint occurred simultaneously with the conversion goal.
 - d. **Contribution (number/sum...value):** The contribution value of this touchpoint attributed to the conversion goal based on the selected model.
 - e. **Contribution Rate:** After calculating through the attribution model, the percentage contribution of this touchpoint to the overall total. The calculation logic is the conversion goal metrics under the current attributed event divided by the sum of all conversion goals.
6. If needed, you can toggle on Associate parameter, then select parameters for both starting event and return event.
 7. Repeat above step to add more metric if needed.
 8. If needed, configure global filter by selecting event parameter or user attributes. Similar to event filter, you can add multiple global filters and configure the filter relationship.
 9. If needed, configure grouping by selecting an event parameter or user attribute.
 10. Choose Query to start the analysis.
 11. Adjust the data granularity, such as Daily, Weekly, Monthly, if needed.
 12. Specify query time range.

Example

Calculate the retention rate of new customers who downloaded from different app markets on Android one week ago.

1. Select the **Attribution Analysis** model.
2. Choose purchase as the conversion event, choose sumGroup by [event]value as conversion metric
3. Choose The day as conversion window.
4. Select view_live as touchpoint event, add filter of live_id = live_1.
5. Repeat step 4 to add touchpoints for the rest of three live channel.
6. Click **Query** .

All configurations are as shown in the image below:

The screenshot shows the 'Attribution analysis' configuration page. At the top, the 'Analytics model' is set to 'Attribution analysis'. The main configuration area is divided into two columns. The left column, 'Define metrics', is highlighted with a red box and contains the following settings:

- Set conversion goal ID:** 'purchase' (dropdown), 'Sum by [event] value' (dropdown).
- Set conversion window ID:** 'The Day' (dropdown).
- Select touchpoint event ID:** Four rows, each with 'view_live' (dropdown), '[event] <...' (dropdown), '=', and 'input value' (input field). Each row has a corresponding 'live,1' through 'live,4' button.
- Buttons:** '+ Add Touch Point' and 'Query' (highlighted with a red box).

The right column, 'Filters', contains a search bar with 'input value' and a '+ Add Filter' button. At the bottom, the 'Date range' is set to 'Last 1 month' and the 'Attribution model' is set to 'First touch' (highlighted with a red box). A message at the bottom center reads 'Please configure metrics to query'.

Analyses

Analyses module allows you to create and modify dashboards based on the clickstream datasets in a drag-and-drop approach. It provides greater flexibility for users to create business-specific metrics and visualizations. You can use the module to:

- create dashboard that are not provided in preset dashboard or not supported by explorations.
- make changes to the custom dashboard saved from exploration analysis, such as adding calculation fields to calculate custom metrics, adjust visual types etc.
- join clickstream data with external datasets, such as adding item master data to enrich clickstream datasets.

Access Analyses

To access Analyses, follow below steps:

1. Go to **Clickstream Analytics on AWS Console**, in the **Navigation Bar**, click on "**Analytics Studio**", a new tab will be opened in your browser.
2. In the Analytics Studio page, click the **Analyses** icon in the left navigation panel.

How it works

Analyses module is essentially the author interface of QuickSight, in which you have the admin access to all the QuickSight functionalities, for example, create analysis, add or manage datasets, publish and share dashboards.

Note

Only the user with Administrator or Analyst role can access this module.

The guidance automatically added the following datasets for each project and app, which contains all fields of event, user, and session tables, making it easy for you do custom analysis.

Dataset name	Description
Event_View_app_name_project_name	Event data that includes all public event parameters

To create a custom analysis, you can follow below QuickSight documentation to prepare data and create visualization:

1. [Connecting to data](#)
2. [Preparing data](#)
3. [Visualizing data](#)

Data management

Data Management helps you manage your clickstream data. This module provides the following features:

1. **Metadata management.** The guidance automatically scans the data in your clickstream database to generate metadata, allowing you to view names and descriptions of events, event parameters, and user attributes, as well as update their display names, descriptions, and data dictionaries.
2. **Traffic source configuration.** You can update the traffic source categorization rules in this tab.

Access Data Management

Follow below steps:

1. Go to **Clickstream Analytics on AWS Console**, in the **Navigation Bar**, choose **Analytics Studio**.
2. In the Analytics Studio page that opens, choose **Data Management** in the left navigation panel.

Note

Only the user with Administrator or Analyst role can modify the metadata, such as display name, description.

Metadata management

The guidance automatically scans clickstream data to generate metadata and then stored them in Redshift on a daily basis. There are three types of metadata:

- **Event:** metadata describes clickstream events.
- **Event Parameter:** metadata describes clickstream event parameters.
- **User Attribute:** metadata describes user attributes. Below tables list all the dimensions included in each type of the metadata.

Metadata dimensions

Below tables list all the dimensions included in each type of the metadata.

Event

Dimension name	Description
Event name	The name of the event reported from SDK
Display name	The display name of the event. By default, it is the same as Event name, user can customize the display name.
Description	The description name of the event reported from SDK. User can customize the display name. For the event automatically collected by the clickstream SDK, the guidance has pre-populated description
Source	Describe how the event was collected , Preset indicates the event is automatically collected by SDK, Custom indicates the event is defined and collected by app owner
Platform	Describe which platform the event was collected from, i.e., from Android, Web or iOS
Data volume last day	Describe how much data was collected in last day (in UTC timezone)
SDK version	Describe the version of the SDK that collected the event
Associate preset parameters	The preset event parameters associated with the event
Associate custom parameters	The custom event parameters associated with the event

Event parameters

Dimension name	Description
Parameter name	The name of the event event parameter reported from SDK
Display name	The display name of the event parameter. By default, it is the same as Parameter name, user can customize the display name.
Description	The description name of the event parameter reported from SDK. User can customize the display name. For the event automatically collected by the clickstream SDK, the guidance has pre-populated description
Source	Describe how the event parameter was collected, Preset indicates the event parameter is automatically collected by SDK, Custom indicates the event parameter is defined and collected by app owner
Data type	Describe the data type of the event parameter value, e.g., int, string,
Associate event	The event that the event parameters associate d with.
Dictionary	The unique value for the event parameters, user can customize the display value.

User attribute

Dimension name	Description
Attribute name	The name of the user attribute reported from SDK

Dimension name	Description
Display name	The display name of the user attribute. By default, it is the same as Attribute name, user can customize the display name.
Description	The description name of the user attribute reported from SDK. User can customize the display name. For the user attribute automatically collected by the clickstream SDK, the guidance has pre-populated description
Source	Describe how the event parameter was collected, Preset indicates the user attribute is automatically collected by SDK, Custom indicates the user attribute is defined and collected by app owner
Data type	Describe the data type of the user attribute value, e.g., int, string,

Update event display name and description

1. Click on the **Events** page, select any event, for example, view_item.
2. Click on the column labeled **Display Name**, enter a name, such as View Product Details, and then click confirm.
3. Go back to Exploration, and in the filter dropdown, you can see view_item now displayed as View Product Details.

Follow the same steps to update display name and description for Event Parameters and User Attributes.

Customize data dictionary for Event Parameter values

1. Click on the **Event Properties** page, and in the search box, select an event parameter, such as _entrances.

2. The detailed information for the property should open automatically, choose the **Dictionary** page.
3. Click on the column labeled **Display Value**, enter a new value, such as Non-First Entry for 0, and enter First Entry for 1, then click confirm.
4. Go back to Exploration, and in the filter dropdown, you can see `_entrances` displayed as Non-First Entry and First Entry.

Follow the same steps to customize data dictionary for User Attributes values.

Traffic source

Traffic source describes the channel through which the users arrive at your website or application, such as paid ads, marketing campaigns, search engines, and social networks. This article describes how the guidance collects and processes traffic-source data.

Traffic-source data fields

Traffic source includes the following key dimension to describe how the users arrives at your website or app.:

1. Source - where the traffic originates (e.g., google, baidu, bing)
2. Medium - the methods by which users arrive at your site/app (medium, e.g., organic, cpc/ppc, email).
3. Campaign - the specific marketing efforts you use to drive that traffic (e.g., campaign, creative format, keywords).
4. Auto-tagged Click ID - the parameter generated and appended by ad platform automatically when ad are showed and clicked. (e.g., gclid).

Clickstream Analytics on AWS uses below dimensions and fields to track traffic-source data when sending events.

Dimension	Clickstream SDK preserved attributes	UTM parameters in page_url (Web only)	Field in data schema
Source	<code>_traffic_source_source</code>	<code>utm_source</code>	<code>traffic_source_source</code>

Dimension	Clickstream SDK preserved attributes	UTM parameters in page_url (Web only)	Field in data schema
Medium	_traffic_source_medium	utm_medium	traffic_source_medium
Campaign Name	_traffic_source_campaign	utm_campaign	traffic_source_campaign
Campaign ID	_traffic_source_id	utm_id	traffic_source_id
Campaign Term	_traffic_source_term	utm_term	traffic_source_term
Campaign Content	_traffic_source_content	utm_content	traffic_source_content
Auto-tagged Click ID	_traffic_source_clid	(*)clid	traffic_source_clid
Auto-tagged Click ID Platform	_traffic_source_clid_platform	N/A	traffic_source_clid_platform
App Install Source	_app_install_channel	N/A	app_install_source

Below two dimensions are enriched from above traffic-source data fields and added to each event

Dimension	Field in data schema	Description
Source Category	__traffic_source_category	Categories based on traffic_source_source and referral domain, including Search, Social, Shopping, Video, and Internal
Channel Group	_traffic_source_channel_group	A channel group is a set of channels, which are rule-based categories of your traffic sources, for example, paid search, paid social.

Processing

During data processing, traffic-source field values are populated into dimension values for each event and attributed to users and sessions. Below describes the details steps.

Step 1 - Extract traffic-source data

1. If the preset traffic-source attributes are set with values, data process module will map their values them into corresponding traffic-source data fields. For example, map the value of `_traffic_source_source` to `traffic_source_source` field.
2. (For web only) If the preset traffic-source attributes have no values, data processing module will map the `utm_parameters` (e.g., `utm_source`) and auto-tagged click id in the `page_url` fields into corresponding traffic-source data fields. For example, map the value of `utm_source` to `traffic_source_source` field.
3. (For web only) If the source dimension are still blank after above steps, data processing module will check if there is value in `page_view_latest_referrer` field, and look up source value from the Source Category mapping table based on the domain of the referrer, if no source is matched, it will use the top-level domain name as the value of the `traffic_source_source` dimension.

Step 2 - Derive source category

Data processing module uses a Source Category mapping table (to classify the source into different categories (i.e., search, shopping, video, social). For example, source with values of "google" or "bing" will be classified into Search category.

Step 3 - Derive channel group

Data processing module uses a set of predefined rules to categorize the traffics into different groups (e.g., direct, paid search, organic search) based on the key traffic-source dimensions (mainly the source. medium, and campaign).

Step 4 - Populate traffic source dimensions for user and session tables

While process the traffic-source for each event, the data processing module populate traffic source dimension for each user and session.

1. **User:** If there are traffic-source data in the first meaningful events (e.g., `first_open`, `page_view`, `app_start`, `app_end`) for the first time user visit your website or apps, those

traffic-source dimension will be assigned to corresponding user traffic-source attributes, i.e., `first_traffic_source`, `first_traffic_medium`.

- Session:** When user initiate a new session, the data processing module derives traffic-source dimension for the session from the traffic-source dimensions of the first meaningful events in the session (e.g., `first_open`, `page_view`, `app_start`, `app_end`).

Configurations

Clickstream Analytics on AWS allows you to configure the channel group rules and source category mapping to customize the traffic source processing to meet your analytics needs.

Channel group definitions and rules

Below are the default channel groups and the rules that the guidance uses to categorize the traffic:

Order	Channel	Description	Evaluation rules
1	Direct	Direct is the channel by which users arrive at your site/app via a saved link or by entering your URL.	1. <code>traffic_source_category</code> , <code>traffic_source_source</code> , <code>traffic_source_medium</code> , <code>traffic_source_campaign</code> , <code>traffic_source_content</code> , <code>traffic_source_term</code> , <code>traffic_source_campaign_id</code> , <code>traffic_source_clid</code> are all blank/ (not set), (none) AND 2. <code>latest_referrer</code> is blank
2	Paid Search	Paid Search is the channel by which users arrive at your site/app via ads on search-engine sites	1. <code>traffic_source_category</code> is Search AND (2. <code>traffic_source_medium</code> matches regex <code>^(.cp. </code>

Order	Channel	Description	Evaluation rules
		like Bing, Baidu, or Google.	ppc retargeting paid.*)\$ OR clid is not none/blank).
3	Organic Search	Organic Search is the channel by which users arrive at your site/app via non-ad links in organic-search results.	1. traffic_source_category is Search AND (2. medium is blank or none or exactly matches organic).
4	Paid Social	Paid Social is the channel by which users arrive at your site/app via ads on social sites like Facebook and Twitter.	1. traffic_source_category is Social AND 2. traffic_source_medium matches regex <code>^(cp. ppc retargeting paid.*)\$</code> OR clid is not none/blank.
5	Organic Social	Organic Social is the channel by which users arrive at your site/app via non-ad links on social sites like Facebook or Twitter.	1. traffic_source_category is Social OR 2. traffic_source_medium is one of ("social", "social-network", "social-media", "sm", "social network", "social media")

Order	Channel	Description	Evaluation rules
6	Paid Video	Paid Video is the channel by which users arrive at your site/app via ads on video sites like TikTok, Vimeo, and YouTube.	<p>1. traffic_source_category is Video (i.e., traffic_source_source OR latest_referrer_host matches a list of video sites) AND (2. traffic_source_medium matches regex <code>^(cp. ppc retargeting paid.*)\$</code> OR clid is not none/blank).</p>
7	Organic Video	Organic Video is the channel by which users arrive at your site/app via non-ad links on video sites like YouTube, TikTok, or Vimeo.	<p>1. traffic_source_category is Video OR 2. traffic_source_medium matches regex <code>^(.video.)\$</code>.</p>

Order	Channel	Description	Evaluation rules
8	Paid Shopping	Paid Shopping is the channel by which users arrive at your site/app via paid ads on shopping sites like Amazon or ebay or on individual retailer sites.	<p>1. traffic_source_category is Shopping AND (2. traffic_source_medium matches regex <code>^(cp. ppc retargeting paid.)\$</code></p> <p><i>OR</i> <i>clid is not none/blank OR traffic_source_campaign matches regex <code>^(([\^a-df-z] ^)shop shopping).*\$</code></i></p>
9	Organic Shopping	Organic Shopping is the channel by which users arrive at your site/app via non-ad links on shopping sites like Amazon or ebay.	<p>1. traffic_source_category is Shopping OR</p> <p>2. traffic_source_campaign matches regex <code>^(([\^a-df-z] ^)shop shopping).*\$</code> .</p>
10	Paid Other	Paid Other is the channel by which users arrive at your site/app via ads, but not through an ad identified as Search, Social, Shopping, or Video.	<p>1. traffic_source_category is none AND 2. traffic_source_medium matches regex <code>^(cp. ppc retargeting paid.*)\$</code>.</p>

Order	Channel	Description	Evaluation rules
11	Email	Email is the channel by which users arrive at your site/app via links in email.	1. traffic_source_source contains "mail" OR 2. traffic_source_medium contains "mail" OR 3. latest_referrer_host start with "mail".
12	SMS	SMS is the channel by which users arrive at your site/app via links from text messages.	1. traffic_source_source exactly matches sms OR 2. traffic_source_medium exactly matches "sms" .
13	Audio	Audio is the channel by which users arrive at your site/app via ads on audio platforms (e.g., podcast platforms).	1.traffic_source_medium exactly matches audio
14	Mobile Push Notifications	Mobile Push Notifications is the channel by which users arrive at your site/app via links in mobile-device messages when they're not actively using the app.	1. traffic_source_medium ends with "push" OR 2.traffic_source_medium contains "mobile" or "notification" .

Order	Channel	Description	Evaluation rules
15	Referral	Referral is the channel by which users arrive at your site via non-ad links on other sites/apps (e.g., blogs, news sites).	1. latest_referrer is not none AND traffic_source_category is none AND 2. latest_referrer_host is not Internal Domain .
16	Internal	Traffic from specified internal domain.	1. latest_referrer_host is one of the Internal domains.
17	Unassigned	Traffic that can not be assigned to a channel group.	All others

To create and edit channel group, go to the Data Management > Traffic Source tab > Channel group.

1. Create a new group.

- Click on the Add new group button.
- Fill in the Group name, Description, and Condition.
- Click on Reorder, adjust the evaluation sequence to by clicking on the Upper arrow or Down arrow, then click the Apply.

2. Edit a channel group

- Select a channel group
- Click the action button, and select View details
- Update the channel group then click on Confirm.
- Click on the Reorder
- Adjust the sequence of the channel group by clicking on the Upper arrow and Down arrow.
- Click Apply to save the order.

Source category mapping table

Clickstream Analytics on AWS uses a source category mapping table to classify some known sources into the categories of Search, Social, Shopping, and Video. You can also add Internal category for the traffic source coming from internal source. Below are the description for the columns in the mapping table.

Column	Description	Example
Domain	The host name of the referral URL.	google.com, baidu.com
Source name	The name for the traffic source.	google, baidu
Category	Category for the source.	Search, Shopping, Social, Video, and Internal
Keyword pattern	The keyword parameter name in the referral url, only for Search domain.	q, query, keyword

To create and edit source category, go to the Data Management > Traffic Source tab > Source category.

1. Create a new category.

- Click on the Add new category button. Or you can select an existing category, then click on Action>Copy to new.
- Fill in the Domain, Source name, and Category.
- If it is Search, fill in keyword pattern, you can add multiple.
- Click on Confirm

2. Edit a channel group

- Select a category record
- Click the action button, and select View details
- Update the record then click on Confirm.

Identity Management

Clickstream Analytics on AWS supports a built-in Cognito user pool or third-party OpenID Connect (OIDC) for user management based on your [deployment type](#).

User management

If you use built-in Cognito for user management, you can find the [Cognito user pool](#) starting with `userPoolIDC9497E0` in your deployment Region. When you deploy the web console of the guidance, a user with the required email address will be created as the first user with administrator permission. For more information about user management, refer to [Managing users in your user pool](#). You can also follow [Adding user pool sign-in through a third party](#) to add federated third-party providers, such as SAML and OIDC.

If you are using an OIDC provider, you need to follow the documentation of the OIDC provider to manage users.

User roles

There are four different types of roles that you can assign to users:

Role	Description
Administrator	Have full access to the guidance, including identity management
Operator	Manage projects, alarms, and plug-ins
Analyst	View and update in Analytics Studio
Analyst Reader	View in Analytics Studio

The specific features for roles are shown in the following table:

Feature	Administrator	Operator	Analyst	Analyst Reader
Project management	Read/Write	Read/Write	Read*	Read*
Operation and Alarm	Read/Write	Read/Write	Read*	Read*
Plugin Management	Read/Write	Read/Write	Read*	Read*
Identity Management	Read/Write	None	None	None
Analytics Studio - Dashboards	Read/Write	None	Read/Write	Read
Analytics Studio - Exploration	Read/Write	None	Read/Write	Read
Analytics Studio - Analyses	Read/Write	None	Read/Write	None
Analytics Studio - Data Management	Read/Write	None	Read/Write	Read

* Accessable through the API only

User role management

By default, the authenticated users do not have a role in the guidance. You have two options to manage the user roles in the guidance:

Option 1:

Choose **System - Users** in the web console of the guidance as Administrator user. Then, add, update, or remove the user roles. This setting has precedence over other settings.

Option 2:

Choose **Setting** in **System - Users** in the web console of the guidance as Administrator user. Configure the roles of the guidance mapping to the groups or roles in your OIDC provider.

By default, the guidance supports mapping [group information from the Cognito user pool](#) to multiple roles in the guidance with the following rules:

Group name in Cognito	Solution role
ClickstreamAdmin	Administrator
ClickstreamOperator	Operator
ClickstreamAnalyst	Analyst
ClickstreamAnalystReader	Analyst Reader

For example, you create a group named ClickstreamAnalyst, then add users in the user pool to that group. After those users log in to the guidance, the user has an analyst role to access Analyst Studio.

The guidance supports mapping multiple groups to a single system role, with various group names separated by commas. For example, by modifying the **Operator Role Name: Group1,Group2**, both user groups can be mapped to the **Operator** role of the system.

If you need to support other OIDC providers, modify **User Role Json Path**.

Example: Modify **User Role Json Path** to `$.payload.realm_access.roles`. It can support the mapping of Keycloak roles to solution roles, where the token format of Keycloak is as follows:

```
{
  "exp": 1701070445,
  "iat": 1701063245,
  "auth_time": 1701062050,
  "jti": "4a892061-56e1-4997-a5f3-84a5d38215f0",
  "iss": "https://keycloak.xxxx.cn/auth/realms/xxx",
  "aud": "P*****Y",
  "sub": "29563a2d-****-43bb-b861-c163da7fe984",
  "typ": "ID",
  "azp": "P*****Y",
  "session_state": "4df36df4-****-4e53-9c1a-43e6d27ffbb9",
```

```
"at_hash": "P*****Y",
"acr": "0",
"sid": "4df36df4-****-4e53-9c1a-43e6d27ffbb9",
"email_verified": false,
"realm_access": {
  "roles": [
    "role1",
    "role2",
    "role3",
  ]
},
"preferred_username": "your name",
"email": "your-name@example.com"
}
```

Upgrade the guidance

Note

Be advised that upgrading directly from version 1.0.x to 1.1.6(+) version is not supported. It is necessary to upgrade to version 1.1.5 first.

Note

By upgrading the web console from earlier 1.1 versions before 1.1.6, you could continue to view the dashboards of the project. However, you could not explore the existing Clickstream data due the changes of data schemas. If you wish to continue to use the Explorations, you will need to upgrade the data pipeline as well and migrate the existing data to new data schemas (if you want to explore historical data).

Planning and Preparation

1. **Backup of Modified QuickSight Analysis and Dashboard:** The guidance upgrade may involve modifying the out-of-the-box analysis and dashboard. In this case, you can back them up following [this documentation](#).
2. **Data Processing Interval** (only applicable to upgrade from v1.0.x): The pipeline upgrade will take about 20 minutes. Make sure no data processing job is running while upgrading the existing pipeline. You can update the existing pipeline to increase the interval and view whether there are running jobs of the EMR Serverless application in the console.

Upgrade Process

Upgrade the web console stack

1. Log in to [AWS CloudFormation console](#), select your existing web console stack, and choose **Update**.
2. Select **Replace current template**.
3. Under **Specify template**:

- Select Amazon S3 URL.
- Refer to the table below to find the link for your deployment type.
- Paste the link in the Amazon S3 URL box.
- Choose **Next**.

Template	Description
Use Cognito for authentication	Deploy as public service in AWS Regions
Use Cognito for authentication with custom domain	Deploy as public service with custom domain in AWS Regions
Use OIDC for authentication	Deploy as public service in AWS Regions
Use OIDC for authentication with custom domain	Deploy as public service with custom domain in AWS Regions
Use OIDC for authentication within VPC	Deploy as private service within VPC in AWS Regions
Use OIDC for authentication with custom domain in AWS China	Deploy as public service with custom domain in AWS China Regions
Use OIDC for authentication within VPC in AWS China	Deploy as private service within VPC in AWS China Regions

4. Under **Parameters**, review the parameters for the template and modify them as necessary. Refer to [Deployment](#) for details about the parameters.
5. Choose **Next**.
6. On the **Configure stack options** page, choose **Next**.
7. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template might create (IAM) resources.
8. Choose **View change set** and verify the changes.
9. Choose **Execute change set** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive an UPDATE_COMPLETE status after a few minutes.

Upgrade the pipeline of project

Important

If you encounter any issues during the upgrade process, refer to [Troubleshooting](#) for more information.

1. Log in to the web console of the guidance.
2. Go to **Projects**, and choose the project to be upgraded.
3. Choose project id or **View Details** button.
4. In the project details page, choose the **Upgrade** button. You will be prompted to confirm the upgrade action.
5. Choose **Confirm**.

You can view the status of the pipeline in the guidance console in the **Status** column. After a few minutes, you can receive an Active status.

Post-Upgrade Actions

This section provides instructions for post-upgrade actions.

Ingestion

As of version 1.1.7, this guidance uses [launch templates](#). After upgrading the data ingestion module, complete the following steps to replace the Amazon EC2 instances used by Amazon ECS with the new launch template configuration.

1. Increase the desired task number by [updating the Amazon ECS service](#).
2. After the newly added Amazon ECS tasks have started successfully, [manually stop the old tasks](#).
3. Manually [terminate the old Amazon EC2 instances](#).

Data Modeling

Upgrade the Data Schema and Out-of-the-box Dashboards

The guidance automatically and asynchronously upgrades the views and materialized views used by the dashboard after upgrading the pipeline of the project. The duration of the update depends on the workload of the Redshift cluster and the existing data volume, and can take minutes to hours. You can track the progress in the **Redshift Schemas** section in the **Processing** tab of the Pipeline Detail page. If the post-configuration job fails, you can access the execution of the workflow through its link and rerun the job via **Actions - Redrive or New** execution with the input unchanged.

Migrate the existing data (only applicable when upgrading from version earlier than v1.1.6)

1. Open [Redshift query editor v2](#). For more information, refer to [Working with query editor v2](#) to log in and query data using Redshift query editor v2.

Note

The data migration process is CPU-intensive and **will incur additional cost**. Before starting the migration, ensure that the load on your Redshift is low. It's also advisable to consider temporarily increasing the RPU's of Redshift Serverless or the cluster size when migrating large volumes of data.

In our benchmark, we migrated 100 million events in 25 minutes using 32 RPU's of Redshift Serverless.

- Average number of events per day: ****10 million****
- Total events for 30 days: ****300 million****
- RedShift RPU: ****32 RPU's****
- Total duration: ****4 hours 45 minutes****
- Total cost: ****\$47.77****

2. Select the Serverless workgroup or provisioned cluster, `<project-id>-><app-id>->Tables`, and make sure tables for the appId are listed there.
3. Create a new SQL Editor, select your project's schema.

- Execute below SQL in editor. Customize the date range as desired, and execute the following SQL in the editor to migrate events from the past 30 days, or any number of days up to the present, to the new tables.

```
-- please replace `<app-id>` with your actual app id
-- update the day range based on your need
CALL "<app-id>".sp_migrate_data_v2(30);
```

- Wait for the SQL to complete. The execution time depends on the volume of data in events table.
- Execute the following SQL to check the stored procedure execution log; ensure there are no errors. If there are any interruptions, timeouts, or other errors, you can re-execute step 4 to continue the data migration.

```
-- please replace `<app-id>` with your actual app id

SELECT * FROM "<app-id>".clickstream_log WHERE log_name = 'sp_migrate_event_to_v2'
ORDER BY log_date DESC;

SELECT * FROM "<app-id>".clickstream_log WHERE log_name = 'sp_migrate_user_to_v2'
ORDER BY log_date DESC;

SELECT * FROM "<app-id>".clickstream_log WHERE log_name = 'sp_migrate_item_to_v2'
ORDER BY log_date DESC;

SELECT * FROM "<app-id>".clickstream_log WHERE log_name =
'sp_migrate_session_to_v2' ORDER BY log_date DESC;

SELECT * FROM "<app-id>".clickstream_log WHERE log_name = 'sp_migrate_data_to_v2'
ORDER BY log_date DESC;
```

- Populate the event data to clickstream_event_base_view table.

```
-- please replace `<app-id>` with your actual app id
-- update the day range (30 days in below example based on your need
CALL "<app-id>".clickstream_event_base_view_sp(NULL, NULL,24*30);
```

Note

It is recommended to refresh the `clickstream_event_base_view` in batches, especially in the following scenarios:

- When there are new event load jobs coming in before the migration job completes.
- When there are new event load jobs coming in before the migration job completes.

Refreshing the data in batches needs to be done based on the event timestamp. Call the following stored procedure multiple times, in order from old to new event timestamps using below SQL command.

```
call "schema".clickstream_event_base_view(start_event_timestamp,
end_event_timestamp, 1);
```

For example, to refresh data between 2024-05-10 00:00:00 and 2024-05-12 00:00:00, execute the following SQL:

```
call "schema".clickstream_event_base_view_sp(TIMESTAMP 'epoch' +
1715270400 * INTERVAL '1 second', TIMESTAMP 'epoch' + 1715443200 *
INTERVAL '1 second', 1);
```

8. Follow this guide to calculate metrics for the new preset dashboard based on the migrated data.
9. If your applications no longer use the legacy tables and views, run the SQLs below to clean them to save the storage of Redshift.

```
-- please replace `<app-id>` with your actual app id

DROP TABLE "<app-id>".event CASCADE;

DROP TABLE "<app-id>".item CASCADE;

DROP TABLE "<app-id>".user CASCADE;

DROP TABLE "<app-id>".event_parameter CASCADE;

DROP PROCEDURE "<app-id>".sp_migrate_event_to_v2(nday integer);

DROP PROCEDURE "<app-id>".sp_migrate_item_to_v2(nday integer);

DROP PROCEDURE "<app-id>".sp_clear_expired_events(retention_range_days integer);
```

```
DROP PROCEDURE "<app-id>".sp_migrate_data_to_v2(nday integer);  
  
DROP PROCEDURE "<app-id>".sp_migrate_user_to_v2();  
  
DROP PROCEDURE "<app-id>".sp_migrate_session_to_v2();  
  
DROP PROCEDURE "<app-id>".sp_clear_item_and_user();
```

Frequently Asked Questions

General

Q: What is Clickstream Analytics on AWS?

A Guidance that enables customers to build clickstream analytic system on AWS easily. This guidance automates the data pipeline creation per customers' configurations with a visual pipeline builder, and provides SDKs for web and mobiles apps (including iOS, and Android) to help customers to collect and ingest client-side data into the data pipeline on AWS. After data ingestion, the guidance allows customers to further enrich and model the event data for business users to query, and provides built-in visualizations (for example, acquisition, engagement, retention) to help them generate insights faster.

Data pipeline

Q: When do I choose Amazon Redshift Serverless for data modeling?

If the data pipeline meets the below criteria, Amazon Redshift Serverless is preferred.

- The data processing interval is equal to or larger than one hour.
- The report querying and other usage, such as ETL, is intensive for a few hours at most.
- Redshift is not used for streaming ingestion.
- The estimated cost is lower than the provisioned cluster.

Q: I already enable data modeling on Redshift, so why can't I see the schema and tables created by this guidance in the Redshift query editor?

This guidance creates a separate database and schema within your Amazon Redshift cluster for storing and processing clickstream events. By default, the schema, tables, and views are only owned by the user who created them and are not visible to other users who log into the Redshift query editor.

You could use the superusers or the admin of Redshift to view them.

- For provisioned Redshift, you could use admin or the Database user specified when configuring the data pipeline.

- For Redshift serverless, the schema and tables are created by an IAM role managed by the guidance; there is no default password for this user. You could edit admin credentials for the Redshift serverless namespace.

Once you view the schema and tables in the query editor, you can grant permissions to other Redshift users.

Q: How do I monitor the health of the data pipeline for my project?

You can open the built-in [observability dashboard](#) to view the key metrics of your data pipeline.

The dashboard displays metrics for different components of your data pipeline, including data ingestion, processing, and modeling.

- **Data Ingestion - Server**
 - **Server Request Counts:** The total requests received by the ingestion servers in the given period. You can use it to calculate the request per second (RPS) of the ingestion servers.
 - **Server Response Time:** The average response time in seconds of the ingestion servers in the given period.
 - **Server (ECS) Tasks:** The number of tasks/instances running for the ingestion servers.
- **Data Ingestion - Sink - Kinesis Data Stream** (available when enabling KDS as ingestion sink)
 - **Kinesis Throttled and Failed Records:** The total putting records of KDS were throttled or failed in the given period.
 - **Kinesis to S3 Lambda Error count:** The total error count in a given period when sinking records in KDS to S3.
 - **Kinesis to S3 Lambda success rate (%):** The success rate of sinking KDS records to S3.
- **Data Processing** (available when enabling data processing)
 - **Data Processing Job success rate (%):** The success rate of a data processing job in a given period.
 - **Data Processing Row counts:** The chart contains four metrics.
 - **source count:** The raw request count of the ingestion server received in the batch data processing.
 - **flatted source count:** The SDK sends multiple clickstream events in a request in batch. It's the total clickstream events in the processed source requests.

- **sink count:** The total number of valid clickstream events that are transformed and enriched in data processing, and sink to S3 again.
- **corrupted count:** The total invalid or unprocessable events in the data processing batch. You can check the corrupted file log in the bucket with path `clickstream/<project id>/data/pipeline-temp/<project id>/job-data/etl_corrupted_json_data/jobName=<emr serverless run job id>/` configured for your data pipeline.
- **Data Modeling** (available when enabling data modeling on Redshift)
 - **'Load data to Redshift tables' workflow:** The success or failure count of the workflow loading processed data into Redshift in the given period.
 - **File max-age:** The maximum age of processed files located in S3 is not loaded to Redshift. There is a built-in alarm that will be triggered when the max-age exceeds the data processing interval.
 - **Redshift-Serverless ComputeCapacity** (available when using Redshift serverless): The RPU usage of the Redshift serverless workgroup. If the used RPU count always reaches the maximum RPU number of the Redshift serverless, it means there are insufficient compute resources for the workload in Redshift.

Q: How do I re-run a failed data processing job?

Occasionally, the data processing job fails. You can re-run the failed job to reprocess the data in that given period. The steps are:

1. Locate the failed job in **EMR Studio - Applications - Clickstream-<project id>**.
2. Choose **Clone**.
3. Keep all parameters unchanged, and choose **Submit job run**.

Q: How do I resume a failed data-loading workflow?

This guidance uses a workflow named `ClickstreamLoadDataWorkflow`, orchestrated by AWS Step Functions to load the processed data into Amazon Redshift. The workflow uses a DynamoDB table to record the files to be loaded that are processed by the data processing job. Any failure won't lose any data for loading into Redshift. It's safe to execute the workflow again to resume the loading workflow after it fails.

Q: How do I recalculate historical events for out-of-the-box dashboards?

The metrics in the out-of-the-box dashboards are calculated on a daily basis. If you need to recalculate the metrics in case of there are changes to your historical data, you could manually reschedule the workflow to re-calculate the metrics. Follow these steps:

1. Open the Step Functions service in the AWS console for the region where your data pipeline is located.
2. Find the state machine named RefreshMaterializedViewsWorkflowRefreshMVStateMachine. If you have multiple projects in the same region, check the tags of the state machine to ensure it belongs to the project for which you want to recalculate the metrics.
3. Start a new execution with the following input JSON. You need to change the refreshStartTime and refreshEndTime values to the date range for the data you want to recalculate.

```
{
  "refreshEndTime": 171540689200,
  "refreshStartTime": 1711929600000
}
```

SDK

Q: Could I use other SDK to send data to the pipeline created by this guidance?

Yes. The guidance supports using third-party SDK to send data to the pipeline. Note that, if you want to enable data processing and modeling module when using a third-party SDK to send data, you need to provide an transformation plugin to map third-party SDK's data structure to guidance data schema. Please refer to [Custom plugin](#) for more details.

Analytics Studio

Q: Why is the Analytics Studio is not available?

Possible reasons are:

- The version of the pipeline is not v1.1 or higher. You can try upgrading the pipeline and wait for the upgrade to complete before trying again.
- The reporting module is not enabled on the pipeline.

Q: How can I modify the default dashboard?

You are not allowed to modify the default dashboard directly. However, you can create a new analysis from the default dashboard and then create a new dashboard from the analysis that you copied. Below are the steps to create analysis from the default dashboard:

1. In Analytics Studio, choose the Analyzes module, and choose **Dashboards**.
2. Open the default dashboard with the name of `Clickstream Dashboard - - .`
3. Choose the **Share** icon and choose **Share Dashboard** in the upper right corner.
4. In the new window, turn on Allow "save as" in the **Save as Analysis** column for "ClickstreamPublishUser" (scroll the window to the right if you don't see the column).
5. Go back to the Dashboard, refresh the webpage, and you should be able to see the **Save as** button in the upper right corner.
6. Choose **Save as**, enter a name for the analysis, and choose **SAVE**. Now you should be able to see a new analysis in the Analyzes, with which now you can edit and publish a new dashboard.

Q: How to speed up the loading of the default dashboard?

You can accelerate report loading by converting QuickSight datasets to SPICE mode. Here are the steps to do this:

1. Purchase SPICE capacity through the QuickSight console. The required capacity depends on your data volume; it is recommended to enable the auto-purchase option.
2. Open the guidance console, select the target project, click the pipeline **Active** status on the pipeline details page, and then open the stack detail link of **Reporting** in the CloudFormation console.
3. Click the Update button and select the **Use existing template** option.
4. Find the **Enable QuickSight SPICE Import Mode** parameter and change its value to yes. Keep other parameters unchanged.
5. Click next to complete the stack update. Once the update is complete, you can start using it.

Note

1. Enabling SPICE will incur additional cost, refer to QuickSight pricing page for more details.

2. By default, the guidance refreshes data in SPICE at 6 AM in your dashboard's time zone every day. You can manually update the schedule in QuickSight

Q: How to implement a dedicated Redshift for Analytics Studio?

Redshift supports sharing data across different Redshift clusters, allowing you to use a dedicated Redshift cluster for Analytics Studio to achieve better query performance and cost optimization.

Before implementing Amazon Redshift data sharing, please note the following:

- You can share data between the same cluster types, as well as between provisioned clusters and serverless clusters.
- Only Ra3 type clusters and Redshift serverless support data sharing.

Taking Redshift serverless as an example of data sharing, follow these operational steps:

1. Create a Redshift serverless as the data consumer.
2. Run SQL in the producer Redshift database (The project database configured in the Clickstream guidance) to create a data share and grant consumer permissions:

```
-- Create Data sharing
CREATE DATASHARE <data share name> SET PUBLICACCESSIBLE FALSE;
ALTER DATASHARE <data share name> ADD SCHEMA <schema>;
ALTER DATASHARE <data share name> ADD ALL TABLES IN SCHEMA <schema>;

-- Grant the Data sharing to the consumer Redshift.

GRANT USAGE ON DATASHARE <data share name> TO NAMESPACE '<consumer namespace id>';
```

Replace **<data share name>** with the name you want to share, **<schema>** with the schema you want to share, and **<consumer namespace id>** with the consumer Redshift serverless namespace ID.

3. Run following SQLs in the consumer Redshift database:

```
-- Create database
CREATE DATABASE <new database name> WITH PERMISSIONS FROM DATASHARE <data share name>
OF NAMESPACE '<source namespace id>';
```

```

-- Create bi user
CREATE USER bi_user PASSWORD '<strong password>';

GRANT USAGE ON DATABASE "<new database name>" TO bi_user;

GRANT USAGE ON SCHEMA "<new database name>". "<schema>" TO bi_user;

GRANT SELECT ON ALL TABLES IN SCHEMA "<new database name>". "<schema>" TO bi_user;
-- Grant permission to data api role

CREATE USER "IAMR:<data api role name>" PASSWORD DISABLE;
GRANT USAGE ON DATABASE "<new database name>" TO "IAMR:<data api role name>";
GRANT USAGE ON SCHEMA "<new database name>". "<schema>" TO "IAMR:<data api role name>";
GRANT SELECT ON ALL TABLES IN SCHEMA "<new database name>". "<schema>" TO "IAMR:<data
  api role name>";

-- Test bi_user permission (optional)

SET SESSION AUTHORIZATION bi_user;
SELECT CURRENT_USER;
SELECT * FROM "<new database name>". "<schema>". "event_v2" limit 1;

```

Replace **<new database name>** with the database name in the consumer Redshift (it can be different from the original database name), replace **<source namespace id>** with the producer Redshift serverless namespace ID, and replace **<data api role name>** with the name of Data Api Role, which can be obtained from the output `RedshiftDataApiRoleArn` of the Reporting stack.

4. Create a new secret for the BI user in Secrets Manager, specifying the value as plaintext like below:

```
{"username":"bi_user","password":"<strong password>"}
```

The key name should be like: **/clickstream/reporting/user/bi_user**.

5. Go to Cloudformation in AWS console, update the reporting stack to use the consumer Redshift:

- Redshift Endpoint Url (Required): Consumer Redshift access endpoint
- Redshift Default database name (Required): dev
- Redshift Database Name (Required): <new database name>
- Parameter Key Name (Required): <key name>

- **Comma Delimited Security Group Ids (Optional):** The security group for VPC connection to access Redshift
- **Comma Delimited Subnet Ids (Optional):** The subnet IDs for the consumer Redshift

Pricing

Q: How will I be charged and billed for the use of this guidance?

The guidance is free to use, and you are responsible for the cost of AWS services used while running this guidance. You pay only for what you use, and there are no minimum or setup fees. Refer to the Cost section for detailed cost estimation.

Troubleshooting

This section provides troubleshooting instructions for deploying and using the guidance.

Problem: Deployment failure due to "Invalid Logging Configuration: The CloudWatch Logs Resource Policy size was exceeded"

If you encounter a deployment failure due to creating CloudWatch log group with an error message like the one below,

```
Cannot enable logging. Policy document length breaking Cloudwatch Logs Constraints, either < 1 or > 5120 (Service: AmazonApiGatewayV2; Status Code: 400; Error Code: BadRequestException; Request ID: xxx-yyy-zzz; Proxy: null)
```

Resolution:

[CloudWatch Logs resource policies are limited to 5120 characters](#). The remediation is merging or removing useless policies, then updating the resource policies of CloudWatch logs to reduce the number of policies.

Below is a sample command to reset resource policy of CloudWatch logs:

```
aws logs put-resource-policy --policy-name AWSLogDeliveryWrite20150319 \  
--policy-document '  
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AWSLogDeliveryWrite2",  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "delivery.logs.amazonaws.com"  
      },  
      "Action": [  
        "logs:CreateLogStream",  
        "logs:PutLogEvents"  
      ]  
    }  
  ]  
}
```

```
    ],
    "Resource": [
    ],
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "<your AWS account id>"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:logs:<AWS region>:<your AWS account id>:*"
      }
    }
  }
]
}
```

Problem: Cannot delete the CloudFormation stacks created for the Clickstream pipeline

If you encounter a failure with an error message like the one below when deleting the CloudFormation stacks created for the Clickstream pipeline,

```
Role arn:aws:iam::<your AWS account id>:role/<stack nam>-
ClickStreamApiStackActionSta-<random suffix> is invalid or cannot be
assumed
```

Resolution:

It results from deleting the web console stack for this guidance before the CloudFormation stacks are made for the Clickstream pipeline.

Please create a new IAM role with the identical name mentioned in the above error message and trust the CloudFormation service with sufficient permission to delete those stacks.

Note

You can delete the IAM role after successfully removing those CloudFormation stacks.

Problem: Reporting stack(Clickstream-Reporting-xxx) deployment fail

Reporting stack deployment failed with message like

```
Connection attempt timed out
```

And it happened when creating DataSource(AWS::QuickSight::DataSource).

Resolution:

Login guidance web console and click "Retry" button in pipeline detail information page.

Problem: Clickstream-DataModelingRedshift-xxxxx stack upgrade failed in UPDATE_ROLLBACK_FAILED

When upgrading from 1.0.x to the latest version, if the CloudFormation stack Clickstream-DataModelingRedshift-xxxxx is in the UPDATE_ROLLBACK_FAILED state, you need to manually fix it by following the steps below.

Resolution:

1. In the Cloudformation **Resource** tab, find the Lambda Function name with logical ID CreateApplicationSchemasCreateSchemaForApplicationsFn.
2. Update the fn_name and aws_region in below script and execute it in a shell terminal. Make sure you have AWS CLI installed and configured.

```
aws_region=<us-east-1> # replace this with your AWS region, and remove '<', '>'
fn_name=<fn_name_to_replace> # replace this with actual function name in step 1 and
remove '<', '>'

cat <<END | > ./index.mjs
export const handler = async (event) => {
  console.log('No ops!')
  const response = {
    Status: 'SUCCESS',
    Data: {
      DatabaseName: '',
```

```
    RedshiftBIUsername: ''
  }
};
return response;
};
END

rm ./noops-lambda.zip > /dev/null 2>&1
zip ./noops-lambda.zip ./index.mjs

aws lambda update-function-code --function-name ${fn_name} \
  --zip-file fileb://./noops-lambda.zip \
  --region ${aws_region} | tee /dev/null
```

3. In the Cloudformation web console, choose **Stack actions** and **Continue update rollback**.
4. Wait until the stack status is UPDATE_ROLLBACK_COMPLETE.
5. Retry the upgrade from the guidance web console.

Problem: Can not sink data to MSK cluster, got "InvalidReplicationFactor (Broker: Invalid replication factor)" log in Ingestion Server

If you notice that data can not be sunk into S3 through MSK cluster, and the error message in log of Ingestion Server (ECS) worker task is as below:

```
Message production error: InvalidReplicationFactor (Broker: Invalid replication factor)
```

Resolution:

This is caused by replication factor larger than available brokers, please edit the MSK cluster configuration, set `default.replication.factor` not larger than the total number of brokers.

Problem: data processing job failure

If the data processing job implemented by EMR serverless fails with the below errors:

- `IOException: No space left on device`

```
Job failed, please check complete logs in configured logging destination. ExitCode:
1. Last few exceptions: Caused by: java.io.IOException: No space left on device
Exception in thread "main" org.apache.spark.SparkException:
```

- **ExecutorDeadException**

```
Job failed, please check complete logs in configured logging destination. ExitCode:
1. Last few exceptions: Caused by: org.apache.spark.ExecutorDeadException:
The relative remote executor(Id: 34), which maintains the block data to
fetch is dead. org.apache.spark.shuffle.FetchFailedException Caused by:
org.apache.spark.SparkException: Job aborted due to stage failure: ShuffleMapStage
```

- **Could not find CoarseGrainedScheduler**

```
Job failed, please check complete logs in configured logging destination.
ExitCode: 1. Last few exceptions: org.apache.spark.SparkException: Could not find
CoarseGrainedScheduler.
```

You need to tune the EMR job default configuration. For more information, refer to [configure execution parameters](#).

Problem: data loading workflow failure due to meeting the 25,000 events limit in a single execution history

It's caused by the large volume of data to be loaded or the Redshift load being very high. You could mitigate this error by increasing the compute resources of Redshift (for example, RPU for Redshift serverless) or reducing the [data processing interval](#). Then [restart the data-loading workflow](#).

Uninstall the guidance

You will encounter an IAM role missing error if you delete Clickstream Analytics on AWS main stack before you delete the stacks created for Clickstream projects. Clickstream Analytics on AWS console launches additional CloudFormation stacks for the Clickstream pipelines. We recommend you delete projects before uninstalling the guidance.

Step 1. Delete projects

1. Go to the Clickstream Analytics on AWS console.
2. In the left sidebar, choose **Projects**.
3. Select the project to be deleted.
4. Choose the **Delete** button in the upper right corner.
5. Repeat steps 3 and 4 to delete all your projects.

Step 2. Delete Clickstream Analytics on AWS stack

1. Go to the [CloudFormation console](#).
2. Find the CloudFormation stack of the guidance.
3. Delete the CloudFormation Stack of the guidance.
4. (Optional) Delete the S3 bucket created by the guidance.
 - a. Choose the CloudFormation stack of the guidance, and select the **Resources** tab.
 - b. In the search bar, enter DataBucket. It shows all resources with the name DataBucket created by the guidance. You can find the resource type **AWS::S3::Bucket**, and the **Physical ID** field is the S3 bucket name.
 - c. Go to the S3 console, and find the S3 bucket with the bucket name. **Empty** and **Delete** the S3 bucket.

Additional resources

Upload SSL Certificate to IAM

Upload the SSL certificate by running the AWS CLI command `upload-server-certificate` similar to the following:

```
aws iam upload-server-certificate --path /cloudfront/ \  
--server-certificate-name YourCertificate \  
--certificate-body file://Certificate.pem \  
--certificate-chain file://CertificateChain.pem \  
--private-key file://PrivateKey.pem
```

Replace the file names and `Your Certificate` with the names for your uploaded files and certificate. You must specify the `file://` prefix in the `certificate-body`, `certificate-chain` and `private-key` parameters in the API request. Otherwise, the request fails with a `MalformedCertificate: Unknown error message`.

Note

You must specify a path using the `--path` option. The path must begin with `/cloudfront` and must include a trailing slash (for example, `/cloudfront/test/`).

After the certificate is uploaded, the AWS command `upload-server-certificate` returns metadata for the uploaded certificate, including the certificate's Amazon Resource Name (ARN), friendly name, identifier (ID), and expiration date.

To view the uploaded certificate, run the AWS CLI command `list-server-certificates`:

```
aws iam list-server-certificates
```

For more information, see [uploading a server certificate](#) to IAM.

Developer guide

This section provides the source code for the guidance.

Source code

Visit our GitHub repository to download the [source code](#) for this guidance. The Clickstream Analytics on AWS template is generated using the [AWS Cloud Development Kit \(AWS CDK\) \(CDK\)](#). Refer to the [README.md](#) file for additional information.

Contributors

- Chen, Haiyun
- Deng, Mingtong
- Li, Jingnan
- Li, Min
- Liu, Yong
- Luo, Robin
- Qiao, Wei
- Qian, Yang
- Que, Mingfei
- Zhu, Mengxin
- Zhu, Xiaowei

Revisions

Publication date: *August 2023*

Check the [CHANGELOG.md](#) file in the GitHub repository to see all notable changes and updates to the software. The changelog provides a clear record of improvements and fixes for each version.

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 AWS current 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. AWS responsibilities and liabilities 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.

Clickstream Analytics on AWS is licensed under the terms of the [Apache License, Version 2.0](#).