Digital User Engagement Events Database

Implementation Guide

Digital User Engagement Events Database: Implementation Guide

Table of Contents

Start analyzing the real-time stream of engagement data from Amazon SES and Amazon Pinpoint

AWS Implementation Guide

AWS Solutions Builder Team

June 2020

This implementation guide discusses architectural considerations and configuration steps for deploying the Digital User Engagement Events Database solution in the Amazon Web Services (AWS) Cloud. It includes a link to an AWS CloudFormation template that launches, configures, and runs the AWS services required to deploy this solution using AWS best practices for security and availability.

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

Overview

Customers want to stay connected to their favorite businesses and brands. They loyally follow the latest products, news, and promotions through a variety of online and offline channels. They expect businesses and brands to understand them uniquely and communicate to them with relevant and timely messaging. Rising to meet these expectations, modern data-driven marketers look to data to understand their customers to deliver the right message, on the right channel, at the right time. These marketers require messaging tools that can execute across multiple channels at scale and analytics tools to gain insights from customer engagement.

Amazon Simple Email Service (Amazon SES) and Amazon Pinpoint provide customers powerful tools to orchestrate and deliver communications using email, SMS, voice, and mobile push channels. In addition to providing rich dashboards showing aggregate engagement data, both Amazon SES and Amazon Pinpoint allow you to stream engagement events in real-time to Amazon Kinesis. These events include email sends, email opens, email clicks, email bounces, email spam complaints, SMS sends, SMS failures, SMS opt outs, and custom application events.

The Digital User Engagement Events Database solution is a reference implementation that automatically provisions and configures the AWS services necessary to start analyzing the real-time stream of engagement data from Amazon SES and Amazon Pinpoint using Amazon Athena. The deployed event database follows best practices and can be queried directly by data analysts or pulled into visualization tools like Amazon QuickSight to create custom dashboards.

Cost

You are responsible for the cost of the AWS services used while running this solution. As of the date of publication, the cost for running this solution with default settings in the US East (N. Virginia) Region is approximately **$40 a month**. The cost estimate includes the cost of Amazon Pinpoint or Amazon Simple Email Service (Amazon SES) to send email messages, and the cost of Amazon Kinesis Data Firehose, Amazon Simple Storage Service (Amazon S3), AWS Lambda, Amazon Athena, AWS Glue, and Amazon CloudWatch Events. The estimate assumes sending one million email messages a day through Amazon Pinpoint or Amazon SES, and executing 5 Amazon Athena queries daily that query a month's worth of event data.

Prices are subject to change. For full details, see the pricing webpage for each AWS service you will be using in this solution.

Architecture Overview

Deploying this solution with the **default parameters** builds the following environment in the AWS Cloud.

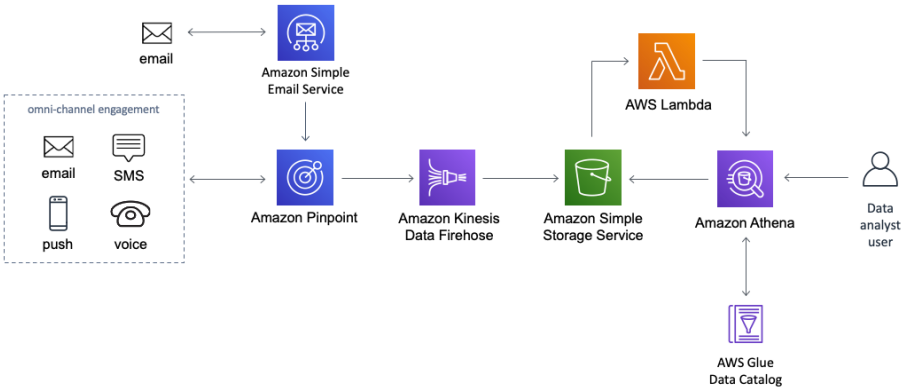

Figure 1: Digital User Engagement Events Database

The AWS CloudFormation template configures an Amazon Pinpoint event stream and an Amazon Simple Email Service (Amazon SES) configuration set to use Amazon Kinesis Data Firehose to store event data in Amazon Simple Storage Service (Amazon S3). The Amazon S3 data schema is stored in an AWS Glue Data Catalog enabling data queries with Amazon Athena. If configured, the AWS CloudFormation template will either use an existing Amazon Pinpoint project or create a new one. Optionally, it will configure existing Amazon SES configuration sets to use the Amazon Pinpoint project as an event destination, allowing for all email events to be collected into the event database.

There is a constant flow of real-time data moving from Amazon SES and Amazon Pinpoint through Kinesis Data Firehose and persisted in Amazon S3. Kinesis Data Firehose is configured to convert these events from JSON to parquet format, and then Snappy compressed for queries against Amazon S3 to be more performant using Amazon Athena.

The schema created in the AWS Glue Data Catalog follows data partitioning best practices requiring the new partitions to be added as new data is persisted in Amazon S3. The AWS CloudFormation template configures the Amazon S3 bucket with an Lambda function trigger that automatically adds new table partitions as files are saved.

Solution Components

Amazon Athena database

The deployed Amazon Athena database contains all engagement data from Amazon Pinpoint and Amazon Simple Email Service (Amazon SES) configuration sets. A single table is constructed that matches the event schema from the Amazon Pinpoint event stream. Individual views are created for each event type, which allows the data to be easily consumed by end users.

Refer to the full data dictionary that describes each view at the field level in Appendix A (p. 13). Customers can use the Amazon Pinpoint Events API to report custom events from their applications. To create views for these custom events, refer to Appendix E (p. 41). Reference queries using these views can be found in Appendix B (p. 35), which demonstrates query best practices including how to use the partition field **ingest_timestamp**.

Amazon Kinesis Data Firehose

This solution deploys a single reusable Amazon Kinesis Data Firehose and configures it to be used by the Amazon Pinpoint project's event stream. This Kinesis Data Firehose can be used in multiple Amazon SES configuration sets and Amazon Pinpoint projects at the same time.

To configure additional Amazon Pinpoint projects, refer to Appendix C (p. 39). To configure additional Amazon Simple Email Service configuration sets, refer to Appendix D (p. 40).

Design Considerations

Regional Deployment

This solution uses Amazon Pinpoint and Amazon Simple Email Service (Amazon SES), which are currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where Amazon Pinpoint and Amazon SES are available. For the most current service availability by AWS Region, see AWS service offerings by region.

AWS CloudFormation Template

The Digital User Engagement Events Database solution uses AWS CloudFormation to automate its deployment in the AWS Cloud. This solution includes the following AWS CloudFormation template, which you can download before deployment.

View Template **digital-user-engagement-events-database.template:** Use this template to launch this solution and all associated components. The default configuration deploys Amazon Pinpoint, Amazon Simple Email Service (Amazon SES), Amazon Kinesis, Amazon Simple Storage Service (Amazon S3), AWS Glue, AWS Lambda, Amazon Athena, Amazon CloudWatch, and AWS Identity and Access Management (IAM), but you can customize the template based on your specific network needs.

Automated Deployment

Before you launch the automated deployment, please review the architecture, configuration, storage security, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy this solution into your account.

Time to deploy: Approximately 5 minutes

Prerequisites

There are no required prerequisites for this solution; however, if you wish to deploy this solution to an existing Amazon Pinpoint project or utilize existing Amazon Simple Email Service (Amazon SES) configuration sets, then you must make some customizations before you launch.

Prepare an existing Amazon Pinpoint project for this solution

Use the following procedure to remove the event stream configuration of an existing Amazon Pinpoint project.

1. Navigate to the Amazon Pinpoint console.
2. In the **All projects** section, choose the project you wish to configure.
3. In the navigation pane, select **Settings**, **Event stream**.
4. If the event stream is currently enabled, select **Edit** (on the upper-right corner of the **Services** card).
5. Uncheck **Stream to Amazon Kinesis** and choose **Save** to disable the configuration.
6. In the navigation pane, select **Settings**, **General settings**, and note the **Project ID**. You will need this identifier in Step 1 (p. 8).

   Note
   Enter the project identifier as the value for the **Amazon Pinpoint Project ID** parameter. In Amazon Pinpoint, a *project* is the same as an *application*. This solution uses the term *Project ID* instead of *Application ID*.

Gather the existing Amazon SES configuration sets names

To configure Amazon Simple Email Service (Amazon SES) to use the same events database, complete the following steps.

1. Navigate to the Amazon Simple Email Service console.
2. In the navigation pane, select **Configuration Sets**.
3. Make note of all of the **Configuration Set Names** that you want this solution to update in order to report the events in the events database. These names will be used in Step 1 (p. 8).

What We'll Cover

The procedure for deploying this architecture on AWS consists of the following steps. For detailed instructions, follow the links for each step.

Step 1. Launch the stack (p. 8)

- Launch the AWS CloudFormation template into your AWS account.
- Review the template parameters and adjust, if necessary.

Step 2. Verify with Amazon Athena (p. 9)

- Verify that the **all_events** table successfully deployed.
- Verify that the individual event views successfully deployed.

Step 1. Launch the stack

This automated AWS CloudFormation template deploys the Digital User Engagement Event Database solution in the AWS Cloud.

Note
You are responsible for the cost of the AWS services used while running this solution. See the Cost (p. 2) section for more details. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Sign in to the AWS Management Console and use the button below to launch the AWS CloudFormation template.

   Launch Solution

   You can also download the template as a starting point for your own implementation.
2. The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the console navigation bar.

   Note
   This solution uses Amazon Pinpoint and Amazon Simple Email Service (Amazon SES), which are currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where both Amazon Pinpoint and Amazon SES are available. For the most current availability by AWS Region, see AWS service offerings by Region.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.

Parameter	Default	Description
Amazon Pinpoint Project ID (Optional)	*<optional input>*	The project ID is only applicable if deploying this solution to update an existing Amazon

Parameter	Default	Description
		Pinpoint project. Leaving this blank will cause the solution to create a new Amazon Pinpoint project. **Note:** If specifying an existing Amazon Pinpoint project, ensure that the event stream configuration has been removed according to the Prerequisites (p. 7) section.
Amazon Pinpoint Project Name	`My Pinpoint Project`	If the **Amazon Pinpoint Project ID (Optional)** parameter is left blank, provide a name for this parameter and this solution will create an Amazon Pinpoint project using that name.
Existing Amazon Simple Email Service Configuration Set Names	*<optional input>*	Comma delimited list of existing Amazon SES set names for this solution to update and send email events to the new events database. Leave blank to not update any configuration sets. **Note:** If applicable, remove any Amazon Pinpoint event destinations currently configured in the configuration sets specified.
Amazon Athena / AWS Glue Database Name	`due_eventdb`	Name of the database available in AWS Glue and Amazon Athena where the required schemas are registered. Only lower case and the underscore (_) characters are allowed.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template might create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **CREATE_COMPLETE** in approximately 5 minutes.

   Note
   In addition to the `AthenaPartitionLambda` AWS Lambda function used to repartition the Amazon Athena table, two additional Lambda functions, `CustomResourceHelper` and `CustomBucketNameHelper`, run only during initial configuration or when resources are updated or deleted.
   When running this solution, you will see all three Lambda functions in the AWS Console, but only the `AthenaPartitionLambda` Lambda function is regularly active. Do not delete the `CustomResourceHelper` or `CustomBucketNameHelper` functions—they are necessary to manage associated resources.

Step 2. Verify with Amazon Athena

After the events database stack launch completes, verify that the database, table, and views were set up correctly using Amazon Athena.

1. Navigate to the Amazon Athena console.
2. Under the **Database** dropdown, select the database name you provided earlier in Step 1 (p. 8).

3. Verify that the **all_events** table was created. This table contains all event data emitted from Amazon SES and Amazon Pinpoint.
4. Verify that the views found in the data dictionary in Appendix A (p. 13) were all created successfully.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. For more information about security on AWS, visit the AWS Security Center.

Security Groups

By default, the Amazon Simple Storage Service (Amazon S3) buckets this solution creates are encrypted with S3-SSE AES 256 encryption. The Amazon Kinesis Data Firehose delivery streams are not encrypted. For end-to-end encryption, we recommend enabling server-side encryption. For more information, see Data Protection in Amazon Kinesis Data Firehose in the *Amazon Kinesis Data Firehose Developer Guide*.

Additional Resources

AWS services

- AWS CloudFormation
- Amazon Pinpoint
- Amazon Simple Email Service
- Amazon Kinesis
- Amazon Simple Storage Service
- AWS Glue
- AWS Lambda
- Amazon Athena
- Amazon CloudWatch

Appendix A: Events database data dictionary

This appendix contains the full data dictionary of all the views generated by this solution. Each view represents a distinct event that is generated by Amazon Pinpoint and Amazon Simple Email Service (Amazon SES).

campaign_send

Data view representing the campaign send event from Amazon Pinpoint. A campaign send event is generated for every endpoint when Amazon Pinpoint executes the campaign.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When

Field Name	Data Type	Description
		specified in the WHERE clause, Amazon Athena scans the data only from that partition.

email_click

Data view representing the email click event from Amazon Pinpoint or Amazon SES. The click event is generated when a recipient has received the message and clicked a link in it.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
message_id	`String`	The unique ID of the message. Amazon Pinpoint automatically generates this ID when it accepts the message.
message_send_timestamp	`Timestamp`	The date and time when the message was sent.
from_address	`String`	The email address that sent the message.
destination	`Array<String>`	An array that contains the email addresses that the message was sent to.
subject	`String`	The subject line of the email.

Field Name	Data Type	Description
ip_address	`String`	IP address of the originating click event.
user_agent	`String`	User Agent of the browser of the originating click event.
link	`String`	Link that clicked on originating the event.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon SES, this map contains MessageTags from the email send. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

email_complaint

Data view representing the email complaint event from Amazon Pinpoint or Amazon SES. A complaint event is generated when a recipient received the message, and then reported the message to their email provider as spam (for example, by using the "Report Spam" feature of their email client).

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment

Field Name	Data Type	Description
		number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
message_id	`String`	The unique ID of the message. Amazon Pinpoint automatically generates this ID when it accepts the message.
message_send_timestamp	`Timestamp`	The date and time when the message was sent.
from_address	`String`	The email address that sent the message.
destination	`Array<String>`	An array that contains the email addresses that the message was sent to.
complained_email_address	`String`	Email address that generated the complaint event.
feedback_id	`String`	The unique ID of the complaint event.
user_agent	`String`	The value of the **User-Agent** field from the feedback report. This indicates the name and version of the system that generated the report.
complaint_feedback_type	`String`	The value of the **Feedback-Type** field from the feedback report received from the ISP. This contains the type of feedback.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon SES, this map contains MessageTags from the email send. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

email_delivered

Data view representing the email delivered event from Amazon Pinpoint or Amazon SES. A delivery event is generated when the message was delivered to the recipient.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
message_id	`String`	The unique ID of the message. Amazon Pinpoint automatically generates this ID when it accepts the message.
message_send_timestamp	`Timestamp`	The date and time when the message was sent.
from_address	`String`	The email address that sent the message.
destination	`Array<String>`	An array that contains the email addresses that the message was sent to.
subject	`String`	The subject line of the email.
smtp_response	`String`	The SMTP response message of the remote ISP that accepted the email from Amazon SES. This message varies by email, by receiving mail server, and by receiving ISP.

Field Name	Data Type	Description
reporting_mta	`String`	The host name of the Amazon SES mail server that sent the mail.
recipients	`Array<String>`	A list of the intended recipients of the email to which the delivery notification applies.
processing_time_millis	`Integer`	The time taken to send the message, shown as Unix time in milliseconds.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon SES, this map contains MessageTags from the email send. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

email_hardbounce

Data view representing the email hardbounce event from Amazon Pinpoint or Amazon SES. A hard bounce is generated when a permanent issue prevented Amazon Pinpoint or Amazon Simple Email Service from delivering the message.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.

Field Name	Data Type	Description
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
message_id	`String`	The unique ID of the message. Amazon Pinpoint automatically generates this ID when it accepts the message.
message_send_timestamp	`Timestamp`	The date and time when the message was sent.
from_address	`String`	The email address that sent the message.
destination	`Array<String>`	An array that contains the email addresses that the message was sent to.
bounce_type	`String`	The type of bounce, as determined by Amazon SES. For more information, see Bounce Types.
bounce_sub_type	`String`	The subtype of the bounce, as determined by Amazon SES. For more information, see Bounce Types.
feedback_id	`String`	A unique ID for the bounce.
reporting_mta	`String`	The value of the **Reporting-MTA** field from the DSN. This is the value of the MTA that attempted to perform the delivery, relay, or gateway operation described in the DSN.
bounced_recipient_email_address	`String`	The email address of the recipient. If a DSN is available, this is the value of the **Final-Recipient** field from the DSN.
bounced_recipient_action	`String`	The value of the **Action** field from the DSN. This indicates the action performed by the Reporting-MTA as a result of its attempt to deliver the message to this recipient.

Field Name	Data Type	Description
bounced_recipient_status	`String`	The value of the **Status** field from the DSN. This is the per-recipient transport-independent status code that indicates the delivery status of the message.
bounced_recipient_diagnostic_code	`String`	The status code issued by the reporting MTA. This is the value of the **Diagnostic-Code** field from the DSN. This field may be absent in the DSN (and therefore also absent in the JSON).
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon SES, this map contains MessageTags from the email send. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

email_open

Data view representing the email open event from Amazon Pinpoint or Amazon SES. An open event is generated when the recipient received the message and opened it.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.

Field Name	Data Type	Description
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
message_id	`String`	The unique ID of the message. Amazon Pinpoint automatically generates this ID when it accepts the message.
message_send_timestamp	`Timestamp`	The date and time when the message was sent.
from_address	`String`	The email address that sent the message.
destination	`Array<String>`	An array that contains the email addresses that the message was sent to.
subject	`String`	The subject line of the email.
ip_address	`String`	IP address of the originating click event.
user_agent	`String`	User agent of the browser of the originating click event.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon SES, this map contains MessageTags from the email send. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

email_rejected

Data view representing the email reject event from Amazon Pinpoint or Amazon SES. An open event is generated when the recipient received the message and opened it.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
message_id	`String`	The unique ID of the message. Amazon Pinpoint automatically generates this ID when it accepts the message.
message_send_timestamp	`Timestamp`	The date and time when the message was sent.
from_address	`String`	The email address that sent the message.
destination	`Array<String>`	An array that contains the email addresses that the message was sent to.
subject	`String`	The subject line of the email.
reject_reason	`String`	Reason for the message to be rejected.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon SES, this map contains MessageTags from the email send. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.

Field Name	Data Type	Description
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

email_send

Data view representing the email send event from Amazon Pinpoint or Amazon SES. A send event is generated when the message was accepted by Amazon Pinpoint or Amazon SES and attempted to deliver it to the recipient.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
message_id	`String`	The unique ID of the message. Amazon Pinpoint automatically generates this ID when it accepts the message.
message_send_timestamp	`Timestamp`	The date and time when the message was sent.
from_address	`String`	The email address that sent the message.

Field Name	Data Type	Description
destination	`Array<String>`	An array that contains the email addresses that the message was sent to.
subject	`String`	The subject line of the email.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon SES, this map contains MessageTags from the email send. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

email_softbounce

Data view representing the email softbounce event from Amazon Pinpoint or Amazon SES. A soft bounce is generated when a temporary issue prevented Amazon Pinpoint or Amazon Simple Email Service from delivering the message.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For

Field Name	Data Type	Description
		standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
message_id	`String`	The unique ID of the message. Amazon Pinpoint automatically generates this ID when it accepts the message.
message_send_timestamp	`Timestamp`	The date and time when the message was sent.
from_address	`String`	The email address that sent the message.
destination	`Array<String>`	An array that contains the email addresses that the message was sent to.
bounce_type	`String`	The type of bounce, as determined by Amazon SES. For more information, see Bounce Types.
bounce_sub_type	`String`	The subtype of the bounce, as determined by Amazon SES. For more information, see Bounce Types.
feedback_id	`String`	A unique ID for the bounce.
reporting_mta	`String`	The value of the **Reporting-MTA** field from the DSN. This is the value of the MTA that attempted to perform the delivery, relay, or gateway operation described in the DSN.
bounced_recipient_email_address	`String`	The email address of the recipient. If a DSN is available, this is the value of the **Final-Recipient** field from the DSN.
bounced_recipient_action	`String`	The value of the **Action** field from the DSN. This indicates the action performed by the Reporting-MTA as a result of its attempt to deliver the message to this recipient.
bounced_recipient_status	`String`	The value of the **Status** field from the DSN. This is the per-recipient transport-independent status code that indicates the delivery status of the message.

Field Name	Data Type	Description
bounced_recipient_diagnostic_code	`String`	The status code issued by the reporting MTA. This is the value of the **Diagnostic-Code** field from the DSN. This field may be absent in the DSN (and therefore also absent in the JSON).
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon SES, this map contains MessageTags from the email send. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

email_unsubscribe

Data view representing the email unsubscribe event from Amazon Pinpoint or Amazon SES. The unsubscribe event is generated when a recipient received the message and clicked an unsubscribe link in it.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For

Field Name	Data Type	Description
		standard campaigns, this value is 0.
aws_account_id	`String`	The ID of the AWS account that sent the message.
message_id	`String`	The unique ID of the message. Amazon Pinpoint automatically generates this ID when it accepts the message.
message_send_timestamp	`Timestamp`	The date and time when the message was sent.
from_address	`String`	The email address that sent the message.
destination	`Array<String>`	An array that contains the email addresses that the message was sent to.
subject	`String`	The subject line of the email.
ip_address	`String`	IP address of the originating unsubscribe event.
user_agent	`String`	User agent of the browser of the originating unsubscribe event.
link	`String`	Unsubscribe link that clicked on originating the event.
unsubscribelinktag	`Array<String>`	Link tag used to decorate the unsubscribe link.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon SES, this map contains MessageTags from the email send. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

journey_send

Data view representing the journey send event from Amazon Pinpoint. A journey send event is generated for every endpoint that receives an email from an Amazon Pinpoint journey.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
journey_run_id	`String`	The unique ID of the journey run that generated the event. Amazon Pinpoint generates and assigns this ID automatically to each new run of a journey.
journey_send_status	`String`	Indicates the delivery status of the message that's associated with the event.
journey_id	`String`	The unique ID of the journey that generated the event.
journey_activity_id	`String`	The unique ID of the journey activity that generated the event.
aws_account_id	`String`	The ID of the AWS account that was used to send the message.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

sms_buffered

Data view representing the sms buffered event from Amazon Pinpoint. An SMS buffered event is generated when the message has been received and is still in process of being delivered to the recipient.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
sender_request_id	`String`	A unique ID that's associated with the request to send the SMS message.
destination_phone_number	`String`	The phone number that you attempted to send the message to.
record_status	`String`	Additional information about the status of the message.
iso_country_code	`String`	The country that's associated with the recipient's phone number, shown in ISO 3166-1 alpha-2 format.
number_of_message_parts	`String`	The number of message parts that Amazon Pinpoint created in order to send the message.
message_id	`String`	The unique ID that Amazon Pinpoint generates when it accepts the message.
message_type	`String`	The type of message. Possible values are **Promotional** and **Transactional**.
price_in_millicents_usd	`Double`	The amount that AWS charged you to send the message. This

Field Name	Data Type	Description
		price is shown in thousandths of a United States cent.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

sms_failure

Data view representing the sms failure event from Amazon Pinpoint. A SMS failure event is generated when Amazon Pinpoint wasn't able to deliver the message to the recipient.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
sender_request_id	`String`	A unique ID that's associated with the request to send the SMS message.

Field Name	Data Type	Description
destination_phone_number	`String`	The phone number that you attempted to send the message to.
record_status	`String`	Additional information about the status of the message.
iso_country_code	`String`	The country that's associated with the recipient's phone number, shown in ISO 3166-1 alpha-2 format.
number_of_message_parts	`String`	The number of message parts that Amazon Pinpoint created in order to send the message.
message_id	`String`	The unique ID that Amazon Pinpoint generates when it accepts the message.
origination_phone_number	`String`	The phone number that sent the message.
price_in_millicents_usd	`Double`	The amount that AWS charged you to send the message. This price is shown in thousandths of a United States cent.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

sms_optout

Data view representing the sms optout event from Amazon Pinpoint. A SMS buffered event is generated when a customer received the message and replied by sending the opt-out keyword (usually "STOP").

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon

Field Name	Data Type	Description
		Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
sender_request_id	`String`	A unique ID that's associated with the request to send the SMS message.
destination_phone_number	`String`	The phone number that you attempted to send the message to.
record_status	`String`	Additional information about the status of the message.
iso_country_code	`String`	The country that's associated with the recipient's phone number, shown in ISO 3166-1 alpha-2 format.
number_of_message_parts	`String`	The number of message parts that Amazon Pinpoint created in order to send the message.
message_id	`String`	The unique ID that Amazon Pinpoint generates when it accepts the message.
message_type	`String`	The type of message. Possible values are **Promotional** and **Transactional**.
origination_phone_number	`String`	The phone number that sent the message.
price_in_millicents_usd	`Double`	The amount that AWS charged you to send the message. This price is shown in thousandths of a United States cent.

Field Name	Data Type	Description
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

sms_success

Data view representing the sms success event from Amazon Pinpoint. A SMS buffered event is generated when the message was successfully delivered to the recipient.

Field Name	Data Type	Description
event_timestamp	`Timestamp`	The time when the event was reported, shown as Unix time in milliseconds.
arrival_timestamp	`Timestamp`	The time when the event was received by Amazon Pinpoint, shown as Unix time in milliseconds.
application_id	`String`	Amazon Pinpoint project ID used to send the campaign.
endpoint_id	`String`	The ID of the endpoint that the campaign was sent to.
pinpoint_campaign_id	`String`	The unique ID of the campaign that sent the message.
pinpoint_treatment_id	`String`	If the message was sent using an A/B test campaign, this value represents the treatment number of the message. For standard campaigns, this value is `0`.
aws_account_id	`String`	The ID of the AWS account that sent the message.
sender_request_id	`String`	A unique ID that's associated with the request to send the SMS message.

Field Name	Data Type	Description
destination_phone_number	`String`	The phone number that you attempted to send the message to.
record_status	`String`	Additional information about the status of the message.
iso_country_code	`String`	The country that's associated with the recipient's phone number, shown in ISO 3166-1 alpha-2 format.
number_of_message_parts	`String`	The number of message parts that Amazon Pinpoint created in order to send the message.
message_id	`String`	The unique ID that Amazon Pinpoint generates when it accepts the message.
message_type	`String`	The type of message. Possible values are **Promotional** and **Transactional**.
origination_phone_number	`String`	The phone number that sent the message.
price_in_millicents_usd	`Double`	The amount that AWS charged you to send the message. This price is shown in thousandths of a United States cent.
message_tags	`Map<String, String>`	Custom user-defined event map useful for tracking events from API calls. For events generated by Amazon Pinpoint, this map contains custom context values from the API call.
ingest_timestamp	`Timestamp`	Table partition matching the location of the event in the Amazon S3 bucket. When specified in the WHERE clause, Amazon Athena scans the data only from that partition.

Appendix B: Reference Amazon Athena queries

Below are reference queries that can be run in Athena. To run these queries:

1. Navigate to the Amazon Athena console.
2. Under the **Database** dropdown, select the database name you provided earlier in Step 1 (p. 8).
3. Copy and paste the below queries into one of the query tabs and select **Run query**.

> **Note**
> The following queries take advantage of the Athena table partition field **ingest_timestamp**. This field maps directly to Amazon S3 file paths and helps the query engine determine which files need to be scanned for results. This best practice will increase performance and decrease Athena query costs. See Partitioning Data topic in the *Athena User Guide* for more details.

Reference SMS queries

Query: SMS cost analysis

Find the total SMS cost and number of Amazon Pinpoint campaign sends over the last 30 days. All SMS messages sent via the API are designated as `'No Campaign'`.

```
SELECT
  COALESCE(pinpoint_campaign_id, 'No Campaign') as campaign,
  SUM(price_in_millicents_usd) as cost_in_millicents,
  COUNT(*) as sends
FROM sms_buffered
WHERE ingest_timestamp > current_timestamp - interval '30' day
GROUP BY pinpoint_campaign_id
```

Query: SMS cost by Amazon Pinpoint context parameter

Find the cost of all SMS messages via API with a custom Amazon Pinpoint context parameter (ex: "context":{"mycustomerid":"4"}) over the last 30 days.

```
SELECT SUM(price_in_millicents_usd) FROM sms_buffered
WHERE message_tags['mycustomerid'] = '4'
AND ingest_timestamp > current_timestamp - interval '30' day
```

Reference email queries

Query: All subject lines sent to an email address

Find all emails sent to a particular email address, returning the subject line and the send timestamp, over the last 30 days.

```
SELECT
 a.event_timestamp as WhenSent,
 a.subject as EmailSubject,
 CASE WHEN b.event_timestamp IS NULL THEN 0 ELSE 1 END as DidOpen
FROM email_send a
LEFT JOIN email_open b
 ON a.message_id = b.message_id
WHERE contains(a.destination, 'example_address@example.com')
 AND a.ingest_timestamp > current_timestamp - interval '30' day
ORDER BY a.event_timestamp DESC
```

Query: Email engagement analysis by subject line

Find total sends, opens, clicks, and unsubscribes, grouping by email subject line, over the last 30 days.

```
SELECT
 subject, COUNT(*) as sends,
 (SELECT COUNT(*) FROM email_open WHERE email_open.subject = email_send.subject AND
 ingest_timestamp > current_timestamp - interval '30' day) AS NumOpens,
 (SELECT COUNT(*) FROM email_click WHERE email_click.subject = email_send.subject AND
 ingest_timestamp > current_timestamp - interval '30' day) AS NumClicks,
 (SELECT COUNT(*) FROM email_unsubscribe WHERE email_unsubscribe.subject =
 email_send.subject AND ingest_timestamp > current_timestamp - interval '30' day) AS
 NumUnsubs

FROM email_send
WHERE ingest_timestamp > current_timestamp - interval '30' day
GROUP BY subject
ORDER BY COUNT(*) DESC
```

Query: Transactional email count by Amazon SES tag or Amazon Pinpoint context

Find all email sends by SES message tag (ex: "EmailTags": [{"Name": "mycustomerid", "Value": "4"}]) or custom Amazon Pinpoint context attribute (ex: "context": {"mycustomerid":"4"}) over the last 30 days.

```
SELECT COUNT(*) FROM email_send
WHERE message_tags['mycustomerid'] = '4'
AND ingest_timestamp > current_timestamp - interval '30' day
```

Query: Email address engagement analysis

Find the last email engagement (open or click) for each email address sent at least 10 emails in the last 6 months.

```
WITH dataset AS (

  SELECT
   a.destination,
   COUNT(a.arrival_timestamp) num_sends,
   MAX(b.arrival_timestamp) max_delivered_timestamp,
   MIN(b.arrival_timestamp) min_delivered_timestamp,
   MAX(c.arrival_timestamp) max_open_timestamp,
   MAX(d.arrival_timestamp) max_click_timestamp
  FROM email_send a
  LEFT JOIN email_delivered b ON a.message_id = b.message_id
```

```
  LEFT JOIN email_open c ON a.message_id = c.message_id
  LEFT JOIN email_click d ON a.message_id = d.message_id
  WHERE a.ingest_timestamp > current_timestamp - interval '6' month
  GROUP BY a.destination
)

SELECT
  to_address,
  num_sends,
  max_delivered_timestamp,
  min_delivered_timestamp,
  GREATEST(max_open_timestamp, max_click_timestamp) as last_engagement
FROM dataset
CROSS JOIN UNNEST(destination) as t(to_address)
WHERE min_delivered_timestamp < current_timestamp - interval '3' month
 AND num_sends > 10
```

Reference Amazon Pinpoint campaign queries

Query: Campaign engagement analysis

Find total sends, opens, clicks, and unsubscribes, grouping by email Amazon Pinpoint campaign ID, over the last 30 days.

```
SELECT
 pinpoint_campaign_id,
 (SELECT COUNT(*) FROM email_send WHERE email_send.pinpoint_campaign_id =
 campaign_send.pinpoint_campaign_id) AS NumSends,
 (SELECT COUNT(*) FROM email_open WHERE email_open.pinpoint_campaign_id =
 campaign_send.pinpoint_campaign_id) AS NumOpens,
 (SELECT COUNT(*) FROM email_click WHERE email_click.pinpoint_campaign_id =
 campaign_send.pinpoint_campaign_id) AS NumClicks,
 (SELECT COUNT(*) FROM email_hardbounce WHERE email_hardbounce.pinpoint_campaign_id =
 campaign_send.pinpoint_campaign_id) AS NumHardBounces,
 (SELECT COUNT(*) FROM email_softbounce WHERE email_softbounce.pinpoint_campaign_id =
 campaign_send.pinpoint_campaign_id) AS NumSoftBounces,
 (SELECT COUNT(*) FROM email_unsubscribe WHERE email_unsubscribe.pinpoint_campaign_id =
 campaign_send.pinpoint_campaign_id) AS NumUnsubs

FROM campaign_send
WHERE ingest_timestamp > current_timestamp - interval '30' day
GROUP BY pinpoint_campaign_id
```

Reference operational queries

Query: Bounce and complaint monitoring

Monitor sends, hard bounces, complaints, bounce rates, and complaint rates by hour for the last 30 days.

```
SELECT DATE_TRUNC('hour', a.ingest_timestamp) time_window,
  COUNT(a.message_id) total_sends,
  COUNT(b.message_id) total_hardbounces,
  COUNT(c.message_id) total_complaints,
  CAST(COUNT(b.message_id) as double) / CAST(COUNT(a.message_id) as double) bounce_rate,
  CAST(COUNT(c.message_id) as double) / CAST(COUNT(a.message_id) as double) complaint_rate
```

```
FROM email_send a
LEFT JOIN email_hardbounce b ON a.message_id = b.message_id
LEFT JOIN email_complaint c ON a.message_id = c.message_id
WHERE a.ingest_timestamp > current_timestamp - interval '30' day
GROUP BY 1
ORDER BY 1 DESC
```

Query: Open, click, unsubscribe monitoring

Monitor sends, opens, clicks, unsubscribes, open rates, click rates, unsubscribe rates by hour for the last 30 days.

```
SELECT DATE_TRUNC('hour', a.ingest_timestamp) time_window,
  COUNT(a.message_id) total_sends,
  COUNT(b.message_id) total_opens,
  COUNT(c.message_id) total_clicks,
  COUNT(c.message_id) total_unsubs,
  CAST(COUNT(b.message_id) as double) / CAST(COUNT(a.message_id) as double) open_rate,
  CAST(COUNT(c.message_id) as double) / CAST(COUNT(a.message_id) as double) click_rate,
  CAST(COUNT(d.message_id) as double) / CAST(COUNT(a.message_id) as double) unsub_rate
FROM email_send a
LEFT JOIN email_open b ON a.message_id = b.message_id
LEFT JOIN email_click c ON a.message_id = c.message_id
LEFT JOIN email_unsubscribe d ON a.message_id = d.message_id
WHERE a.ingest_timestamp > current_timestamp - interval '30' day
GROUP BY 1
ORDER BY 1 DESC
```

Appendix C: Configuring additional Amazon Pinpoint projects

This solution configures a single Amazon Pinpoint project's event stream to route events into the events database. You can use the following procedure to configure additional Amazon Pinpoint projects.

1. Navigate to the AWS CloudFormation console.
2. On the **Stacks** page, choose the stack created for this solution.
3. On the stack details page, choose the **Outputs** tab and, under the **Key** column, locate `PinpointEventStreamFirehoseName` and `PinpointEventStreamFirehoseRoleName`. These keys identify the Kinesis Data Firehose and Amazon Identity and Access Management (IAM) role names used below.
4. Navigate to the Amazon Pinpoint console.
5. In the **All projects** section, select your project.
6. In the navigation pane, select **Settings**, **Event Stream**.
7. Select **Edit** (on the upper-right corner of the **Services** card).
8. If the event stream is not already configured, check the box for **Stream to Amazon Kinesis**.
9. Choose **Send events to Amazon Kinesis Data Firehose stream** and select the `PinpointEventStreamFirehoseName` name from the dropdown.
10. Choose **Use an existing role** and select the `PinpointEventStreamFirehoseRoleName` name from the dropdown.
11. Choose **Save**.

Appendix D: Updating Amazon SES configuration sets

Amazon Simple Email Service (Amazon SES) can be manually configured to send events to the events database by using the AWS Command Line Interface (AWS CLI). To utilize the Amazon SES ability to export email events, you must set up and utilize configuration sets in your sending activities. To update a configuration set to send events to the events database, complete the following steps in the AWS CLI. For more information, see the AWS Command Line Interface User Guide.

1. Navigate to the AWS CloudFormation console.
2. On the **Stacks** page, choose the stack created for this solution.
3. On the stack details page, choose the **Outputs** tab and, under the **Key** column, locate `PinpointProjectArn`. This key identifies the Amazon Pinpoint project Amazon Resource Name (ARN) that this solution configured.
4. Open the AWS CLI and run the following command. Replace *`<project-arn>`* with the `PinpointProjectArn` value. Replace the *`<config-set-name>`* with the name of the Amazon SES configuration set you want to update.

```
PINPOINT_PROJECT_ARN=<project-arn>
SES_CONFIG_SET_NAME=<config-set-name>

aws sesv2 create-configuration-set-event-destination --configuration-
set-name $SES_CONFIG_SET_NAME --event-destination-name event-
database --event-destination '{"Enabled":true,"MatchingEventTypes":
["SEND","REJECT","BOUNCE","COMPLAINT","DELIVERY","OPEN","CLICK","RENDERING_FAILURE"],"PinpointDestina
{"ApplicationArn":'"$PINPOINT_PROJECT_ARN"'}}'
```

Appendix E: Create views for custom events

Customers can use the Amazon Pinpoint Events API to report custom events. These events are emitted by the Amazon Pinpoint project into Amazon Kinesis Data Firehose and will be available in the events database for analysis. To create views for these custom events, complete the following steps.

1. Navigate to the Amazon Athena console.
2. Under the **Database** dropdown, select the database name you provided earlier in Step 1 (p. 8).
3. Modify the example query below. Replace the italicized items to match the custom event.

```
CREATE OR REPLACE VIEW custom_view_name AS
          SELECT
          from_unixtime((event_timestamp / 1000)) event_timestamp
          , from_unixtime((arrival_timestamp / 1000)) arrival_timestamp
          , application.app_id application_id
          , client.client_id endpoint_id
          , awsaccountid aws_account_id
          , attributes['MyCustomEventAttribute1'] as custom_attr_1
          , attributes['MyCustomEventAttribute2'] as custom_attr_2
          … continue as necessary …

          , ingest_timestamp
          FROM
          all_events
          WHERE (event_type = 'custom_event_name')
```

   Note
   The views take advantage of the Athena table partition field **ingest_timestamp**. This field maps directly to Amazon S3 file paths and helps the query engine determine which files must be scanned for results. This best practice increases performance and decreases Athena query costs. Create your views with this field from the **all_events** table. For more details, see Amazon Athena Partitioning Data in the *Amazon Athena User Guide*.
4. Paste the modified query into one of the query tabs and select **Run query**.

Appendix F: Collection of Operational Metrics

This solution includes an option to send anonymous operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When enabled, the following information is collected and sent to AWS:

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each solution deployment
- **Timestamp:** Data-collection timestamp
- **S3PartitionTriggers:** Number of Amazon S3 triggers processed by the solution as data is written from Amazon Kinesis Data Firehose.

Note that AWS will own the data gathered via this survey. Data collection will be subject to the AWS Privacy Policy. To opt out of this feature, modify the AWS CloudFormation template mapping section as follows:

```
MetricsMap:
    Send-Data:
      SendAnonymousData: "Yes"
```

to

```
MetricsMap:
    Send-Data:
      SendAnonymousData: "No"
```

Source Code

You can visit our GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.

Document Revisions

Date	Change	
June 2020	Initial release	

Notices

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