

AWS Streaming Data Solution for Amazon Kinesis

AWS Implementation Guide

Daniel Pinheiro

August 2020



Copyright (c) 2020 by Amazon.com, Inc. or its affiliates.

AWS Streaming Data Solution for Amazon Kinesis is licensed under the terms of the Apache License Version 2.0 available at <https://www.apache.org/licenses/LICENSE-2.0>

Contents

Overview	4
Cost	5
Architecture overview.....	6
Solution components.....	9
Components for the CloudFormation template using API Gateway, Kinesis Data Streams, and AWS Lambda	9
Components for the CloudFormation template using Amazon EC2, KPL, Kinesis Data Streams, and Kinesis Data Analytics.....	10
Custom resources	14
AWS CloudFormation templates.....	14
Automated deployment	15
Prerequisites.....	15
Deployment for the aws-streaming-data-solution-for-kinesis-using-api-gateway-and-lambda CloudFormation template	15
Deployment for the aws-streaming-data-solution-for-kinesis-using-kpl-and-kinesis-data-analytics CloudFormation template	17
Security	20
IAM roles	21
Security groups	21
Auditing.....	21
Additional resources.....	21
Appendix A: Kinesis Producer Library	23
Appendix B: Uninstall the solution	24
Using the AWS Management Console.....	24
Using AWS Command Line Interface.....	24
Deleting the Amazon S3 buckets.....	24
Deleting the CloudWatch Logs	25

Appendix C: Collection of operational metrics.....	25
Source code.....	27
Revisions	27

About this guide

This implementation guide describes architectural considerations and configuration steps for deploying AWS Streaming Data Solution for Amazon Kinesis in the Amazon Web Services (AWS) Cloud. It includes links to [AWS CloudFormation](#) templates that launch and configure the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for IT architects, developers, and DevOps professionals who want to get started quickly with the core streaming services available in the AWS Cloud.

Overview

Developers want to build a high performance, security compliant, and easy-to-manage streaming solution. This requires them to understand best practices related to access control, logging, and optimal configurations, which are provided by this AWS Streaming Data Solution for Amazon Kinesis Implementation Guide.

[Amazon Kinesis](#) offers key capabilities to cost-effectively process streaming data at any scale, along with the flexibility to choose the tools that best suit the requirements of your application. Streaming data use cases follow a similar pattern where data flows from data producers through streaming storage and data consumers to storage destinations. *Sources* continuously generate data, which is delivered via the *ingest* stage to the *stream storage* layer, where it's durably captured and made available for *streaming processing*. The stream processing layer processes the data in the stream storage layer and sends the processed information to a specified *destination*. For example, developers can leverage [Amazon Kinesis Data Analytics](#) applications that use machine learning queries to perform complex analysis on data, relying on the history of the data in the stream to find unusual patterns. Similarly, developers can use [Amazon Kinesis Data Streams](#) to continuously capture gigabytes of data per second from hundreds of thousands of sources such as website clickstreams, database event streams, financial transactions, social media feeds, and IT logs.

Several combinations of AWS streaming data services are provided to help customers meet their use cases (from streaming extract-transform-load (ETL) to real-time analytics). The AWS Streaming Data Solution for Amazon Kinesis helps developers more easily configure the core AWS services necessary to capture, store, process, and deliver streaming data. To provide flexibility to adapt to different use cases and business needs, this solution offers two AWS CloudFormation templates. Developers can choose the template that most closely aligns to their needs to deploy.

One option is designed to capture data from non-AWS environments such as mobile clients, enable throttling at the API level, and leverage [AWS Lambda](#) to handle errors for Amazon Kinesis streams. This layer of abstraction enables you to implement custom authentication approaches for data producers, control quotas for specific producers, and change the target Kinesis stream. This template uses AWS Lambda as the data consumer, which is best suited for use cases that don't require internal state like filtering, business event processing, and data cleansing. AWS Lambda offers a small surface area for error scenarios and is simple to scale and operate.

The second option leverages [Apache Flink](#) and provides a fully managed service to handle backups for snapshots, a Kinesis Data Analytics implementation of an Apache Flink

Savepoint, automatically. This option also supports the [Amazon Kinesis Producer Library](#) (KPL). This template uses direct ingestion from [Amazon Elastic Compute Cloud](#) (Amazon EC2) to Kinesis Data Streams. This template is best suited when you have a level of control over the code that is writing to the Kinesis Data Streams. This control enables you to implement cost optimizations through buffering on the data producer and has fewer resources to manage compared to alternatives. Kinesis Data Analytics for Apache Flink is used as the data consumer, which is best suited when your use case requires capabilities like durable application and exactly-once processing. It's very efficient at processing high volume data streams with low latency and high availability. Use cases that benefit from these capabilities include streaming extract-transform-load (ETL), real-time analytics, predictive analytics, and machine learning.

Both options offered in this solution are configured to apply best practices to monitor using dashboards and alarms, and to protect data. Developers can easily customize either option in this solution by adding code to streaming applications or by modifying the preconfigured defaults for *streaming processing*.

This guide provides infrastructure and configuration information for planning and deploying the AWS Streaming Data Solution for Amazon Kinesis in the AWS Cloud.

Cost

You are responsible for the cost of the AWS services used while running this solution. This solution offers two AWS CloudFormation templates, enabling developers to choose the template that most closely aligns to their needs. At the date of publication, the monthly cost for running this solution (using either AWS CloudFormation template) and publishing 100 records/second in the US East (N. Virginia) Region are described in the following tables. The Option 1 table provides a cost estimate to deploy the `aws-streaming-data-solution-for-kinesis-using-api-gateway-and-lambda` AWS CloudFormation template that uses Amazon API Gateway, Amazon Kinesis Data Streams (with [enhanced monitoring](#) disabled), and AWS Lambda.

Table for Option 1: Cost estimate for running the solution using the CloudFormation template that deploys API Gateway, Kinesis Data Streams, and AWS Lambda

AWS service	Dimensions	Total cost (per month)
Amazon Kinesis Data Streams	1 shard	\$10.95
	100 records (4 KB) / second	\$3.68
	168 hours data retention	\$14.60
Amazon API Gateway	2,678,400 requests / month (1/sec)	\$10.50

AWS service	Dimensions	Total cost (per month)
AWS Lambda	2,678,400 requests / month (1/sec)	\$3.33
TOTAL:		\$43.06 / month

The Option 2 table provides a cost estimate to deploy the `aws-streaming-data-solution-for-kinesis-using-kpl-and-kinesis-data-analytics` AWS CloudFormation template that uses Amazon Elastic Compute Cloud (Amazon EC2), KPL, Kinesis Data Streams, Kinesis Data Analytics, and Amazon CloudWatch.

Table for Option 2: Cost estimate for running the solution using the CloudFormation template that deploys Amazon EC2, KPL, Kinesis Data Streams, Kinesis Data Analytics, and CloudWatch

AWS service	Dimensions	Total cost (per month)
Amazon Kinesis Producer Library (KPL)	EC2 instance (t3.small) 730 hours / month	\$15.18
Kinesis Data Streams	1 shard	\$10.95
	100 records (4 KB) / second	\$3.68
	168 hours data retention	\$14.60
Kinesis Data Analytics	1 processing unit	\$80.30
	50 GB running application storage	\$5.00
TOTAL:		\$129.71 / month

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

Architecture overview

This solution automatically configures the core AWS services necessary to easily capture, store, process, and deliver streaming data. Two options are available, depending on your use case and business requirements.

Regardless of the combination chosen, this solution leverages [best practices](#) for monitoring and data protection. Review the following two options that are available and choose the one that meets your use case. Each option includes demo code and application that is deployed with the solution. You can easily customize or replace the demo code for your business needs.

Option 1: Deploy the CloudFormation template using API Gateway, Kinesis Data Streams, and AWS Lambda

Deploying the `aws-streaming-data-solution-for-kinesis-using-api-gateway-and-lambda` AWS CloudFormation template builds the following environment in the AWS Cloud.

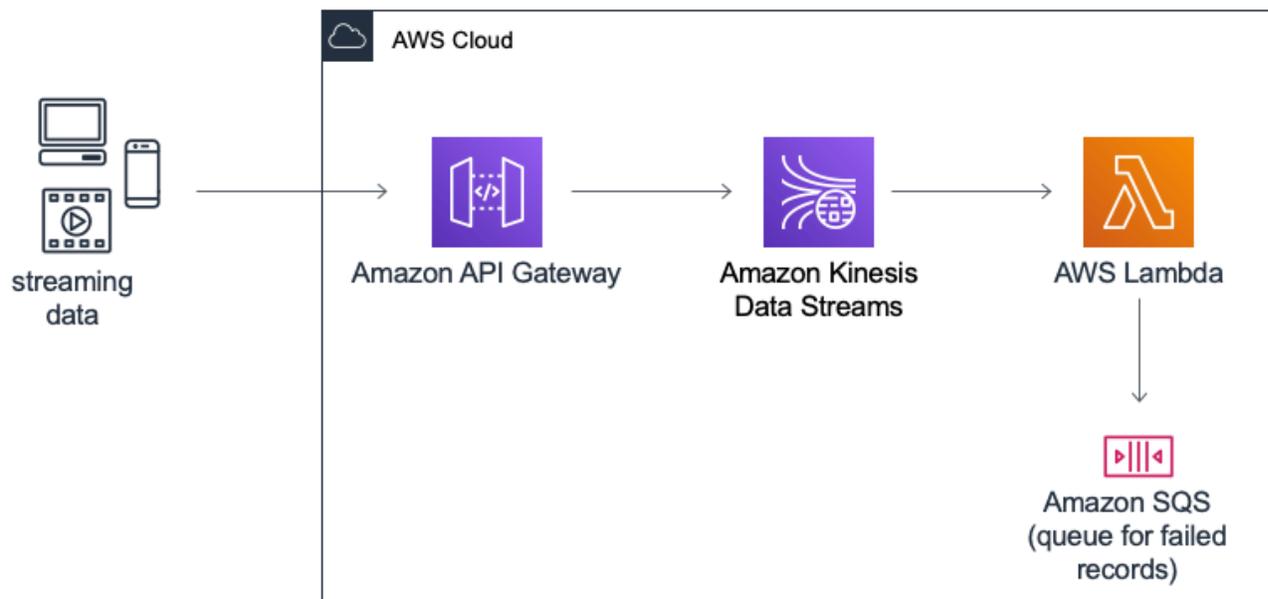


Figure 1: AWS CloudFormation template using Amazon API Gateway, Kinesis Data Streams, and AWS Lambda architecture on AWS

The AWS CloudFormation template deploys an architecture to ingest streaming data using an [Amazon API Gateway](#) and to process the incoming data using an AWS Lambda function. An API Gateway REST API acts as a proxy to [Amazon Kinesis Data Streams](#), enabling either an individual data record or a list of data records to be added. These records are stored in Kinesis Data Streams. An AWS Lambda function processes the records from the data stream. Errors that occur during AWS Lambda processing and records that cannot be processed (failed records) by AWS Lambda are annotated and the events are stored in [Amazon Simple Queue Service](#) (Amazon SQS). An Amazon SQS queue stores metadata for failed batch records and Lambda errors, providing information to enable customers to retrieve these records and determine the next steps to resolve them.

Note: For information about how AWS services handle errors, refer to [Error handling](#) in the *AWS Lambda Developer Guide* and [AWS Lambda Supports Failure-Handling Features](#).

Option 2: Deploy the CloudFormation template using Amazon EC2, KPL, Kinesis Data Streams, Kinesis Data Analytics, and CloudWatch

Deploying the `aws-streaming-data-solution-for-kinesis-using-kpl-and-kinesis-data-analytics` AWS CloudFormation template builds the following environment in the AWS Cloud.

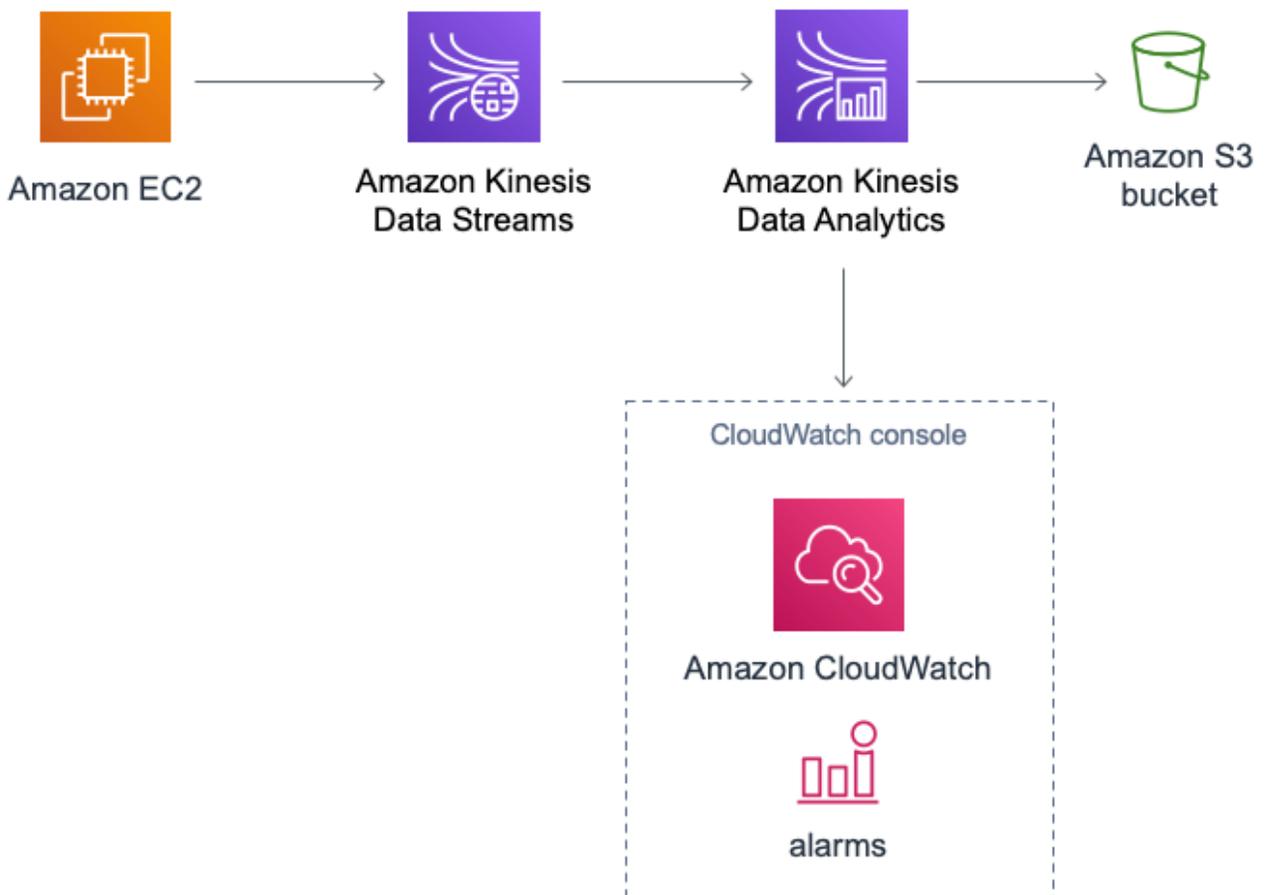


Figure 2: AWS CloudFormation template using Amazon EC2, KPL, Kinesis Data Streams, Kinesis Data Analytics, and CloudWatch architecture on AWS

This AWS CloudFormation template deploys an architecture to ingest streaming data using KPL and to process the incoming data using Amazon Kinesis Data Analytics. In this solution, a sample application publishes data to the data stream.

An Amazon Elastic Compute Cloud (Amazon EC2) instance generates data using KPL. Kinesis Data Streams stores the incoming streaming data and Kinesis Data Analytics processes the incoming records and saves the processed data in an [Amazon Simple Storage Service](#) (Amazon S3) bucket. An [Amazon CloudWatch](#) dashboard monitors application

health, progress, resource utilization, and specific events and errors. CloudWatch alarms are set on essential metrics for Kinesis Data Analytics. For information on essential metrics including recommended alarms, refer to [Using CloudWatch Alarms with Amazon Kinesis Data Analytics for Apache Flink](#) in the *Kinesis Data Analytics Developer Guide*.

Note: This solution can be configured to use an existing Amazon Virtual Private Cloud (Amazon VPC), allowing access to private resources during execution (such as your private databases). By default, Kinesis Data Analytics does not launch into a customer's VPC, but you can enable your applications to access Amazon Relational Database Service (Amazon RDS) using this solution. For information about connecting to a virtual private cloud in your account, refer to [Configuring Kinesis Data Analytics](#) in the *Kinesis Data Analytics Developer Guide*.

Solution components

Components for the CloudFormation template using API Gateway, Kinesis Data Streams, and AWS Lambda

Demo consumer application

A *consumer* is an application that processes data from a Kinesis data stream. The `aws-streaming-data-solution-for-kinesis-using-api-gateway-and-lambda` template includes a demo consumer application, which is a Node.js function that logs the data being published. The source code is available from the solution's [GitHub repository](#) and can be customized to your business needs.

CloudWatch dashboards and alerts

This solution deploys an Amazon CloudWatch dashboard to monitor the health of the data stream. You can easily customize the dashboards and alerts using Amazon CloudWatch or the source code from the [GitHub repository](#).

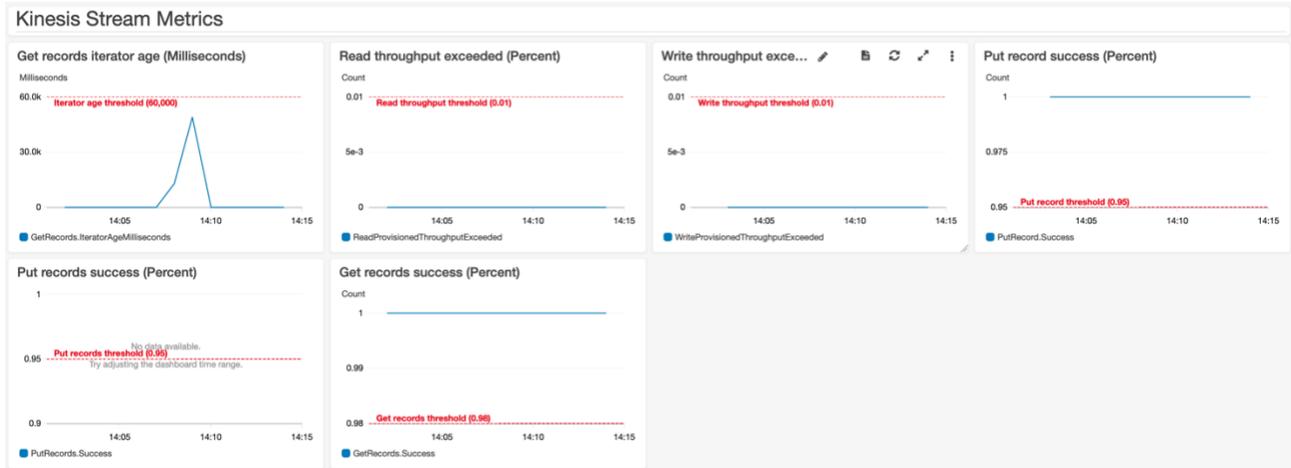


Figure 3: Kinesis Data Streams metrics on the CloudWatch dashboard

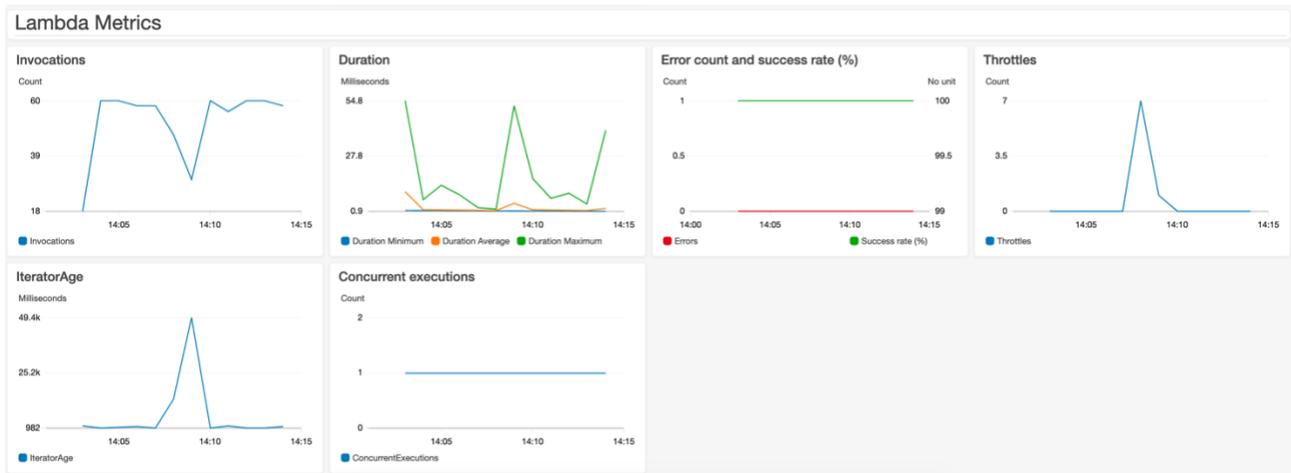


Figure 4: AWS Lambda metrics on the CloudWatch dashboard

Components for the CloudFormation template using Amazon EC2, KPL, Kinesis Data Streams, and Kinesis Data Analytics Demo producer application

A *producer* is an application that puts user data records into a Kinesis data stream (also called data ingestion). The `aws-streaming-data-solution-for-kinesis-using-kpl-and-kinesis-data-analytics` AWS CloudFormation template includes a demo producer application, which is implemented using the Amazon Kinesis Producer Library (KPL). It is configured to write 100 records per second to the data stream. The source code is available from the solution’s [GitHub repository](#) and can be customized to your business needs. For information about the demo producer application and customizing the demo application or replacing it with your own application, refer to the KPL demo [README.md file](#) in the GitHub repository.

Note: This application uses the same schema that is provided in [Getting Started with Amazon Kinesis Data Analytics](#) in the *Amazon Kinesis Data Analytics Developer Guide*.

Demo consumer application

A *consumer* is an application that processes data from a Kinesis data stream. This template also includes a demo consumer application, which is a Java application for Amazon Kinesis Data Analytics. This app demonstrates how to use Apache Flink sources, sinks, and operators. The source code is available from the solution's [GitHub repository](#) and can be customized to your business needs. For information about the demo consumer application and customizing the demo application or replacing it with your own application, refer to the Kinesis Data Analytics and Apache Flink demo [README.md file](#) in the GitHub repository.

For more examples on applications in Kinesis Data Analytics, refer to [Kinesis Data Analytics for Apache Flink Examples](#) in the *Kinesis Data Analytics Developer Guide*.

CloudWatch dashboards and alerts

This solution deploys an Amazon CloudWatch dashboard to monitor the health, progress, resource utilization, and specific events and errors of the data streaming activities. You can also view Log Insights statistics on the CloudWatch dashboard which provides information on Kinesis Data Analytics application logs.

You can easily customize the dashboards and alerts using CloudWatch or the source code from the [GitHub repository](#).

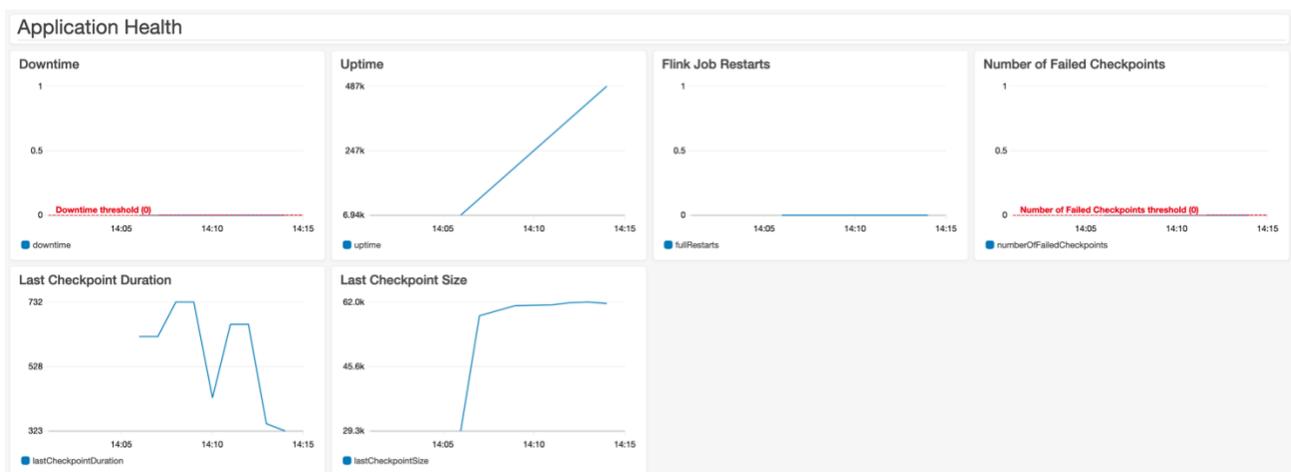


Figure 5: Application Health on the CloudWatch dashboard

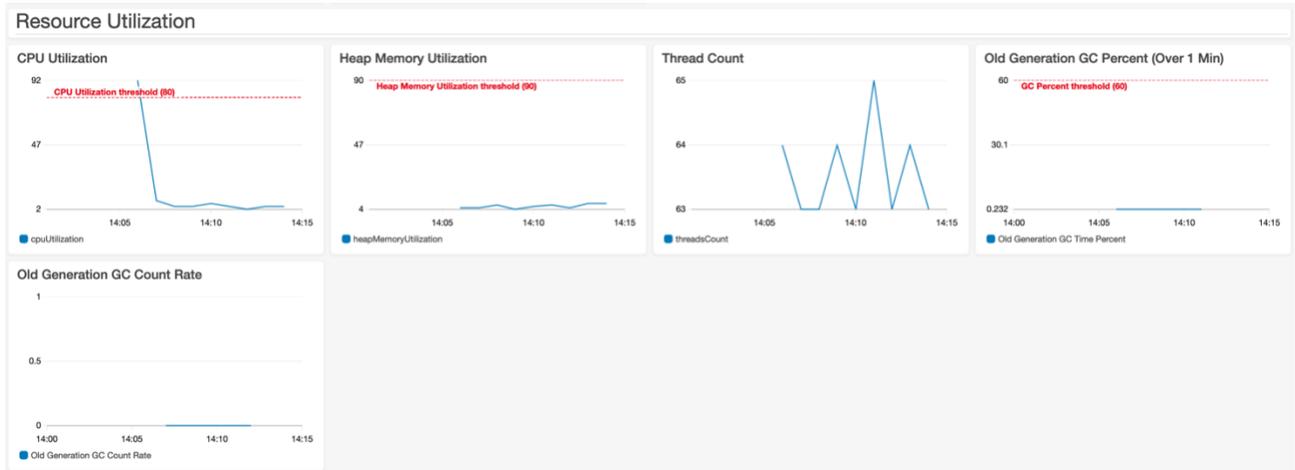


Figure 6: Resource Utilization on the CloudWatch dashboard

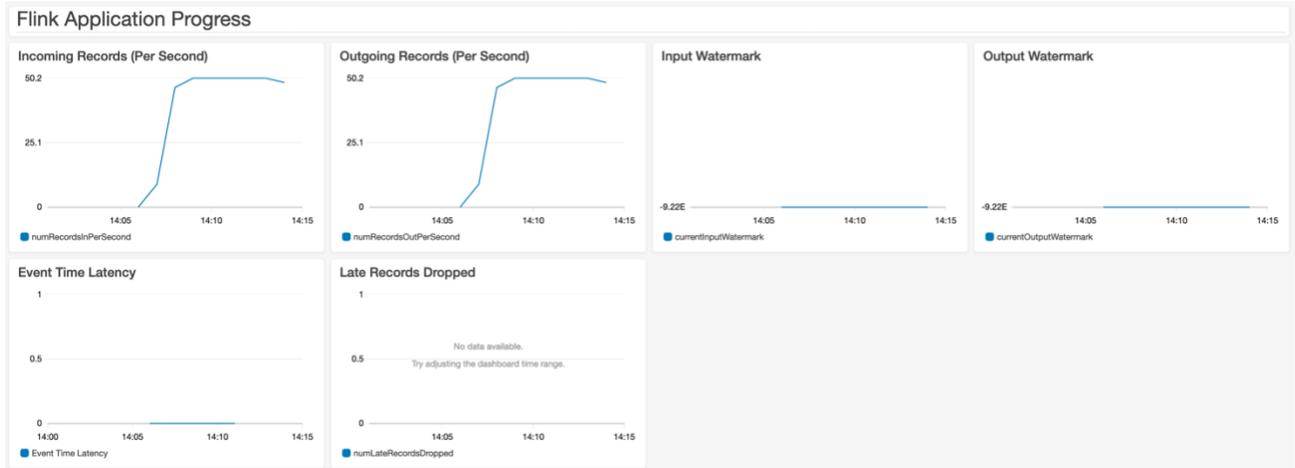


Figure 7: Fink Application Progress on the CloudWatch dashboard

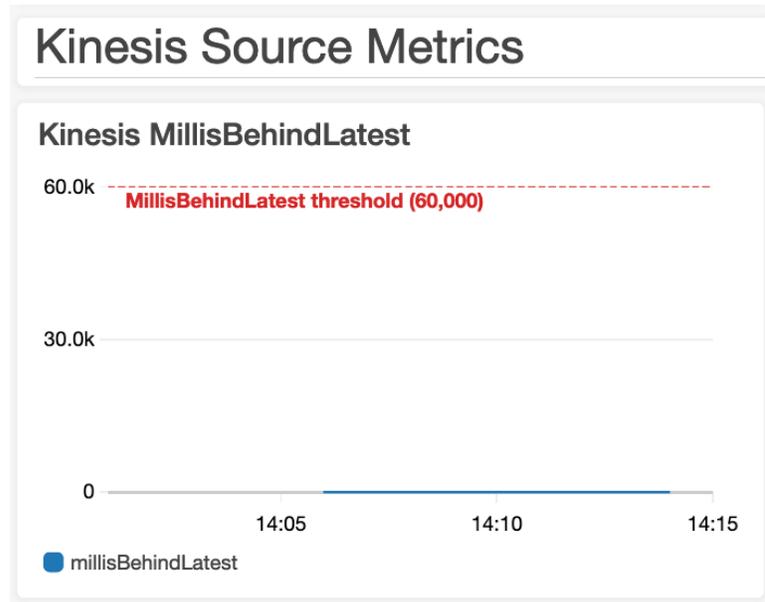


Figure 8: Kinesis Source Metrics on the CloudWatch dashboard

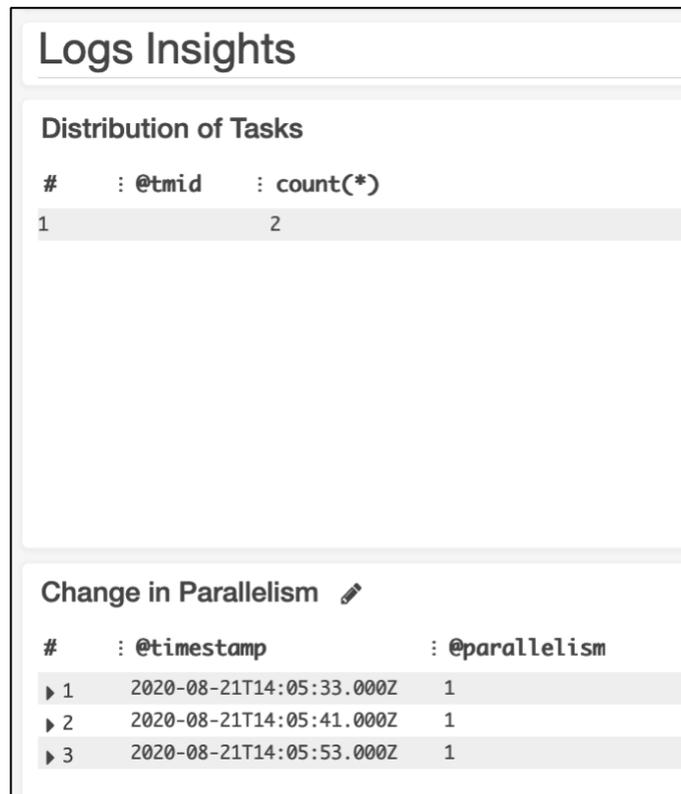


Figure 9: Logs Insights on the CloudWatch dashboard

Custom resources

Both AWS CloudFormation templates provided in this solution support the [enhanced monitoring](#) feature in Amazon Kinesis Data Streams. When enhanced monitoring is enabled, Kinesis Data Streams sends shard-level data to Amazon CloudWatch for an additional cost.

The `aws-streaming-data-solution-for-kinesis-using-kpl-and-kinesis-data-analytics` AWS CloudFormation template uses the VPC configuration capability in Kinesis Data Analytics.

However, since these features are not supported in Amazon Kinesis services (as of the date of this publication), the solution provides AWS Lambda functions that implement custom resources to enable these features. The custom resources are managed in the same stack as the other solution components.

AWS CloudFormation templates

This solution uses AWS CloudFormation to automate the deployment of the AWS Streaming Data Solution for Amazon Kinesis in the AWS Cloud. It includes the following CloudFormation templates, which you can download before deployment and customize to meet your needs:

[View template](#)

aws-streaming-data-solution-for-kinesis-using-api-gateway-and-lambda.template:

Use this template to launch this solution using Amazon API Gateway, Amazon Kinesis Data Streams, and AWS

Lambda. The default configuration deploys API Gateway REST APIs, Kinesis Data Streams, AWS Lambda functions, and an Amazon Simple Queue Service queue. You can also customize the template based on your specific needs.

[View template](#)

aws-streaming-data-solution-for-kinesis-using-kpl-and-kinesis-data-analytics.template:

Use this template to launch this solution using Kinesis Producer Library, Amazon Elastic Compute Cloud

(Amazon EC2), and Amazon Kinesis Data Analytics. The default configuration deploys an Amazon EC2 instance, Kinesis Data Streams, Kinesis Data Analytics, Amazon CloudWatch, and an Amazon Simple Storage Service bucket. You can also customize the template based on your specific needs.

Automated deployment

Prerequisites

Choose one of the following AWS CloudFormation templates to deploy, then follow the step-by-step instructions for your selected template:

- Deploy the `aws-streaming-data-solution-for-kinesis-using-api-gateway-and-lambda.template` [AWS CloudFormation template](#) using Amazon API Gateway, Amazon Kinesis Data Streams, and AWS Lambda
- Deploy the `aws-streaming-data-solution-for-kinesis-using-kpl-and-kinesis-data-analytics.template` [AWS CloudFormation template](#) using Amazon Elastic Compute Cloud (Amazon EC2), Amazon Kinesis Producer Library (KPL), Kinesis Data Streams, Amazon Kinesis Data Analytics, and Amazon CloudWatch

Deployment for the `aws-streaming-data-solution-for-kinesis-using-api-gateway-and-lambda` CloudFormation template

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

Time to deploy: Approximately five minutes

Launch the stack

Note: You are responsible for the cost of the AWS services used while running this solution. Refer to the [Cost](#) section for more details. For full details, refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and use the button to the right to launch the `aws-streaming-data-solution-for-kinesis-using-api-gateway-and-lambda` AWS CloudFormation template. Alternatively, you can [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 the solution in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.

A blue rectangular button with rounded corners containing the text "Launch Solution" in white.

4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to [IAM and STS Limits](#) in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Amazon API Gateway configuration		
Steady-state requests per second (<i>ThrottlingRateLimit</i>)	100	The number of steady-state requests per second that the proxy API Gateway permits. The allowed range is from 1 to 10000.
Burst requests per second (<i>ThrottlingBurstLimit</i>)	50	The number of burst requests per second that the proxy API Gateway permits. The allowed range is from 0 to 5000.
Amazon Kinesis Data Streams configuration		
Number of open shards (<i>ShardCount</i>)	2	The number of shards that the stream uses. The allowed range is from 1 to 100 shards.
Data retention period (hours) (<i>RetentionHours</i>)	24	The number of hours that data records stored in shards will remain accessible. The allowed range is from 24 to 168 hours.
Enable enhanced (shard-level) metrics (<i>EnableEnhancedMonitoring</i>)	false	Choose whether to enable enhanced monitoring for shard-level metrics. This function is disabled by default.
AWS Lambda consumer configuration		
Largest number of records that will be read from the stream at once (<i>BatchSize</i>)	100	The maximum number of records to retrieve in a single batch. The allowed range is from 1 to 10000.
Number of batches to process from each shard concurrently (<i>ParallelizationFactor</i>)	1	The number of batches to process from each shard concurrently. The allowed range is from 1 to 10.
Maximum number of times to retry when the function returns an error (<i>MaxRetryAttempts</i>)	3	The maximum number of retries when the AWS Lambda function returns an error.

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 will 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 receive a **CREATE_COMPLETE** status in approximately five minutes.

The demo application that is included with this solution can be customized or replaced to meet your business needs. The source code is available from the solution's [GitHub repository](#). For information about the demo producer application and customizing the demo application or replacing it with your own application, refer to the [README.md file](#) in the GitHub repository.

Note: This solution includes the `solution-helper` Lambda function, which runs only during initial configuration or when resources are updated or deleted. When you run this solution, you will notice the Lambda functions in the AWS console. While it may not appear active, do not delete the `solution-helper` function because it is necessary to manage associated resources.

Deployment for the aws-streaming-data-solution-for-kinesis-using-kpl-and-kinesis-data-analytics CloudFormation template

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

Time to deploy: Approximately 10 minutes

Deployment overview

Use the following steps to deploy this solution on AWS. For detailed instructions, follow the links for each step.

[Step 1. Launch the stack](#)

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

[Step 2. Post-configuration steps](#)

Step 1. Launch the stack

Note: You are responsible for the cost of the AWS services used while running this solution. Refer to the [Cost](#) section for more details. For full details, refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and use the button to the right to launch the `aws-streaming-data-solution-for-kinesis-using-kpl-and-kinesis-data-analytics` AWS CloudFormation template. Alternatively, you can [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 the solution in a different AWS Region, use the Region selector in the console navigation bar.



Note: This template uses Amazon Kinesis Data Analytics, which is not currently available in all AWS Regions. You must launch this solution in an AWS Region where Kinesis Data Analytics is available. For the most current availability by Region, refer to the [AWS Service Region Table](#).

3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, see [IAM and STS Quotas](#) in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Amazon Kinesis Producer Library (KPL) configuration		
VPC where the KPL instance should be launched (<i>ProducerVpcId</i>)	<i><Requires input></i>	VPC where the KPL instance is launched.
Subnet where the KPL instance should be launched (<i>ProducerSubnetId</i>)	<i><Requires input></i>	Subnet where the KPL instance is launched (the subnet requires access to Kinesis Data Streams, either via IGW or NAT).
Amazon Machine Image for the KPL instance (<i>ProducerAmiId</i>)	<code>/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2</code>	Amazon Machine Image (AMI) ID for the KPL instance.
Amazon Kinesis Data Streams configuration		
Number of open shards (<i>ShardCount</i>)	2	The number of shards that the stream uses. The allowed range is from 1 to 100 shards.
Data retention period (hours) (<i>RetentionHours</i>)	24	The number of hours that data records stored in shards will remain accessible. The allowed range is from 24 to 168 hours.

Parameter	Default	Description
Enable enhanced (shard-level) metrics (<i>EnableEnhancedMonitoring</i>)	false	Choose whether to enable enhanced monitoring for shard-level metrics. This function is disabled by default.
Amazon Kinesis Data Analytics configuration		
Monitoring log level (<i>LogLevel</i>)	INFO	The level of detail of the CloudWatch logs for an application. The available options include <code>DEBUG</code> , <code>ERROR</code> , <code>INFO</code> , and <code>WARN</code> . For information about choose a log level, refer to Application Monitoring Levels in the <i>Amazon Kinesis Data Analytics Developer Guide</i> .
Monitoring metrics level (<i>MetricsLevel</i>)	OPERATOR	The granularity of the CloudWatch logs for an application. The available options include <code>APPLICATION</code> , <code>OPERATOR</code> , <code>PARALLELISM</code> , and <code>TASK</code> . For information about choosing a metrics level, refer to Application Monitoring Levels in the <i>Amazon Kinesis Data Analytics Developer Guide</i> .
Enable service-triggered snapshots (<i>EnableSnapshots</i>)	true	Choose whether snapshots are enabled for an application.
Enable automatic scaling (<i>EnableAutoScaling</i>)	true	Choose whether the Kinesis Data Analytics service can increase the parallelism of the application in response to increased throughput.
Comma-separated list of subnet ids for VPC connectivity (<i>ApplicationSubnetIds</i>)	<Optional input>	If subnet IDs are provided, then security groups must also be included.
Comma-separated list of security groups ids for VPC connectivity (<i>ApplicationSecurityGroupIds</i>)	<Optional input>	If security group IDs are provided, then subnets must also be included.

- Choose **Next**.
- On the **Configure stack options** page, choose **Next**.
- On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
- 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 receive a **CREATE_COMPLETE** status in approximately 10 minutes.

Note: This solution includes the `solution-helper` Lambda function, which runs only during initial configuration. This function is only created if you enable the collection of operational metrics.

Step 2. Post-configuration tasks

By default, the demo producer and demo consumer applications will not run after the stacks are created. Take the following steps to enable them.

1. Sign in to the [Amazon Kinesis console](#) and, from the left menu pane, select **Analytics applications**.
2. On the **Amazon Kinesis Data Analytics** page, select `kda<application-name>`.
3. Choose **Actions** then choose **Run application**.
4. Navigate to the AWS Systems Manager console and, from the left menu pane under **Instances and Nodes**, select **Session Manager**.
5. On the **AWS Systems Manager** page, choose **Start session**.
6. On the **Start a session** page, select the `<ec2-instance-id>` for the KPL instance and choose **Start session**.

Refer to the AWS CloudFormation **Outputs** tab for the Amazon EC2 instance ID.

7. In the console window, run the following command to start the demo producer application. (Replace `<stream-name>`, `<aws-region>`, and `<seconds>` with your specific information).

```
sudo java -jar /tmp/aws-kpl-demo-1.0.0.jar <stream-name> <aws-  
region> <seconds>
```

The demo application that is included with this solution can be customized or replaced to meet your business needs. The source code is available from the solution's [GitHub repository](#). For information about the demo producer application and customizing the demo application or replacing it with your own application, refer to the [README.md file](#) in the GitHub repository.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared 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, visit [AWS Cloud Security](#).

IAM roles

AWS Identity and Access Management (IAM) roles enable customers to assign granular access policies and permissions to services and users in the AWS Cloud. This solution creates IAM roles for communication between services. For more information, refer to [Providing Access to an AWS Service](#) in the *IAM User Guide*.

Security groups

This solution creates a security group for the Amazon Kinesis Producer Library (KPL) instance so that it can communicate with the Amazon Kinesis endpoint. This security group does not allow any inbound traffic, and the instance can only be accessed via [AWS Systems Manager Session Manager](#).

Auditing

Each AWS service included in this solution is integrated with [AWS CloudTrail](#), which captures all API calls. For more details, refer to the following documentation.

- [Logging Kinesis Data Analytics API Calls with AWS CloudTrail](#)
- [Logging calls to Amazon API Gateway APIs with AWS CloudTrail](#)
- [Logging AWS Lambda API calls with AWS CloudTrail](#)
- [Logging Amazon Kinesis Data Streams API Calls with AWS CloudTrail](#)

Additional resources

AWS services

- [Amazon API Gateway](#)
- [Amazon CloudWatch](#)
- [Amazon Elastic Compute Cloud](#)
- [Amazon Kinesis Data Analytics](#)
- [Amazon Kinesis Data Streams](#)
- [Amazon Simple Storage Service](#)
- [AWS CloudFormation](#)
- [AWS Identity and Access Management](#)
- [AWS Lambda](#)
- [Amazon Simple Queue Service](#)

AWS documentation

[Developing Producers Using the Amazon Kinesis Producer Library](#)

Best practices for monitoring and data protection:

- [Viewing Kinesis Data Analytics Metrics and Dimensions](#)
- [Using CloudWatch Alarms with Amazon Kinesis Data Analytics for Apache Flink](#)
- [Security in Amazon Kinesis Data Analytics](#)
- [Security in Amazon Kinesis Data Streams](#)
- [Using AWS Lambda with Amazon Kinesis](#)

Appendix A: Kinesis Producer Library

The CloudFormation resources are created from AWS CDK components, and the resources contain the template code that you can use to customize this solution. For more information about setting up your project and customizing this solution, refer to the [README.md file](#) in GitHub.

The Amazon Kinesis Producer Library (KPL) is used in the AWS CloudFormation template using Amazon Elastic Compute Cloud (Amazon EC2), Amazon Kinesis Data Analytics, and Amazon CloudWatch. The KPL is an easy-to-use, highly configurable library that helps you write to Amazon Kinesis Data Streams. It acts as an intermediary between the producer application code and the Kinesis Data Streams API actions. It also integrates seamlessly with the Kinesis Client Library (KCL) and submits Amazon CloudWatch [metrics](#) on your behalf to provide visibility into producer performance.

The sample producer application included in this solution uses the following [configuration](#):

Parameter	Value	Description
MaxConnections	1	The maximum number of open connections to the backend. HTTP requests are sent in parallel over multiple connections. The allowed range is from 1 to 256.
RequestTimeout	60000	The maximum elapsed time (in milliseconds) from when an HTTP request initiates and when all responses are received. The request times out if the elapsed time is exceeded.
RecordMaxBufferedTime	2000	The maximum amount of buffer time (in milliseconds) that a record is allowed to be buffered before it is sent.
AggregationEnabled	false	Choose whether to enable aggregation. With aggregation, multiple user records are packed into a single KinesisRecord . If disabled, each user record is sent in its own KinesisRecord .

Although the default settings should work for most use cases, you may want to change some of them to tailor the behavior of the **KinesisProducer** to your needs. For more information about configuration parameters, see the sample properties file on the [Amazon Kinesis Producer GitHub repository](#).

Appendix B: Uninstall the solution

You can uninstall the AWS Streaming Data Solution for Amazon Kinesis using the AWS Management Console or the AWS Command Line Interface (AWS CLI). However, the Amazon Simple Storage Service (Amazon S3) bucket and Amazon CloudWatch Logs created by this solution must be manually deleted.

Using the AWS Management Console

1. Sign in to the [AWS CloudFormation console](#).
2. On the **Stacks** page, select this solution stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, see [What Is the AWS Command Line Interface](#) in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <cloudformation-stack-name>
```

Replace *<cloudformation-stack-name>* with the name of your CloudFormation stack.

Deleting the Amazon S3 buckets

This solution is configured to retain the Amazon S3 buckets if you decide to delete the AWS CloudFormation stack to prevent against accidental data loss. After uninstalling the solution, you can manually delete the S3 buckets if you do not need to retain the data. Use the following procedure to delete the Amazon S3 buckets.

1. Sign in to the [Amazon S3 console](#).
2. Choose **Buckets** from the left navigation pane.
3. Locate the *<stack-name>* S3 buckets.
4. Select one of the S3 buckets and choose **Delete**.

Repeat the steps until you have deleted all the *<stack-name>* S3 buckets.

Alternatively, you can configure the AWS CloudFormation template to delete the Amazon S3 buckets automatically. Prior to deleting the stack, change the deletion behavior in the AWS CloudFormation [DeletionPolicy attribute](#).

Deleting the CloudWatch Logs

This solution retains the CloudWatch Logs if you decide to delete the AWS CloudFormation stack to prevent against accidental data loss. After uninstalling the solution, you can manually delete the logs if you do not need to retain the data. Use the following procedure to delete the CloudWatch Logs.

1. Sign in to the [Amazon CloudWatch console](#).
2. Choose **Log Groups** from the left navigation pane.
3. Locate the log groups created by the solution.
4. Select one of the log groups.
5. Choose **Actions** and then choose **Delete**.

Repeat the steps until you have deleted all the solution log groups.

Appendix C: 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 AWS Streaming Data solution for Amazon Kinesis deployment
- **Timestamp:** The UTC formatted timestamp of when the event occurred
- **Data:** The Region where the stack launched, request type (that is, the stack was created, updated, or deleted), and data stream details (shard count, data retention, and whether enhanced monitoring was enabled). For example:

```
{'EnhancedMonitoring': 'false', 'RetentionHours': '24', 'Region':  
'us-east-1', 'ShardCount': '2', 'RequestType': 'Create'}
```

Note that AWS owns the data gathered through this survey. Data collection is subject to the [AWS Privacy Policy](#). To opt out of this feature, modify the AWS CloudFormation template mapping section as follows:

```
"AnonymousData": {  
  "SendAnonymousData" : { "Data" : "Yes" }  
},
```

to

```
"AnonymousData": {  
  "SendAnonymousData" : { "Data" : "No" }  
},
```

Source code

Visit the [GitHub repository](#) to download the templates and scripts for this solution, and to share your customizations with others.

Revisions

Date	Change
August 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.

The AWS Streaming Data Solution for Amazon Kinesis is licensed under the terms of the Apache License Version 2.0 available at <https://www.apache.org/licenses/LICENSE-2.0>.

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.