Streaming Data Solution for Amazon MSK

Implementation Guide

Streaming Data Solution for Amazon MSK: Implementation Guide

Table of Contents

Deployment framework for capturing, storing, processing, and delivering real-time streaming data

Publication date: *November 2022 (last update (p. 27): July 2022)*

The Streaming Data Solution for Amazon MSK allows you to capture, store, process, and deliver real-time streaming data. By automatically configuring the included AWS services, this solution helps you address real-time streaming use cases, for example:

- Capture high volume application log files
- Analyze website clickstreams
- Process database event streams
- Track financial transactions
- Aggregate social media feeds
- Collect IT log files
- Continuously deliver to a data lake

This solution helps accelerate your development lifecycle by minimizing or eliminating the need to model and provision resources using AWS CloudFormation, set up preconfigured Amazon CloudWatch alarms set to recommended thresholds, dashboards, and logging, and manually implement streaming data best practices. This solution is data and logic agnostic, meaning that you can start with boilerplate code and then customize it to your needs.

The solution uses templates where data flows through producers, streaming storage, consumers, and destinations. Producers continuously generate data and send it to streaming storage where it is durably captured and made available for processing by a data consumer. Data consumers process the data and then send it to a destination.

To support multiple use cases and business needs, this solution offers four AWS CloudFormation templates. You can use this solution to test new service combinations as the basis for your production environment, and to improve existing applications.

Option 1 creates a standalone Amazon Managed Streaming for Apache Kafka (Amazon MSK) cluster following best practices, such as sending broker logs to Amazon CloudWatch Logs; encryption at rest; encryption in transit among the broker nodes; and open monitoring with Prometheus activated.

Option 2 adds an AWS Lambda function that processes records in an existing Apache Kafka topic as a starting example that you can modify and customize. The Lambda service internally polls for new records or messages from the event source, and then synchronously invokes the target Lambda function.

Option 3 is intended for use cases when you must back up messages from a topic in Amazon MSK (for instance, to replay or analyze them). It uses Amazon Kinesis Data Firehose (which compresses and encrypts, minimizing the amount of storage used at the destination and increasing security) and Amazon Simple Storage Service (Amazon S3).

Option 4 showcases how to read data from an existing topic in Amazon MSK using Apache Flink, which provides exactly-once processing. It uses Amazon Kinesis Data Analytics (a fully managed service that

handles core capabilities like provisioning compute resources, parallel computation, automatic scaling, and application backups) and Amazon Simple Storage Service (Amazon S3).

All templates are configured to apply best practices to monitor functionality using dashboards and alarms, and to secure data.

This implementation guide describes architectural considerations and configuration steps for deploying the Streaming Data Solution for Amazon MSK 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.

Cost

You are responsible for the cost of the AWS services used while running this solution. As of November 2021, the monthly cost for running this solution in the US East (N. Virginia) Region, is described in the following tables.

Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution. We recommend creating a budget through AWS Cost Explorer to help manage costs.

Sample cost tables

Option 1: Deploy the AWS CloudFormation template using Amazon Managed Streaming for Apache Kafka (Amazon MSK)

The following table provides a cost estimate to deploy the `streaming-data-solution-for-msk` AWS CloudFormation template that deploys Amazon MSK.

Table for Option 1: Cost estimate for running the solution using the CloudFormation template that deploys Amazon MSK

AWS service	Dimensions	Cost per month
Amazon MSK	Broker instance type: kafka.m5.large (3 nodes) Broker storage: 1,000 GB	$468.72 $100.00
Amazon EC2	EC2 instance (t3.small) 730 hours / month	$15.18
	TOTAL:	**$583.90 per month**

Note
The templates for options 2, 3 and 4 accept the Amazon Resource Name (ARN) of the Amazon MSK cluster as a parameter, so the following cost tables only include the services created by this solution.

Option 2: Deploy the AWS CloudFormation template using Amazon MSK and AWS Lambda

The Option 2 table provides a cost estimate to deploy the `streaming-data-solution-for-msk-using-aws-lambda` AWS CloudFormation template that uses Amazon MSKand Lambda.

Table for Option 2: Cost estimate for running the solution using the CloudFormation template that deploys Amazon MSK and Lambda

AWS service	Dimensions	Cost per month
AWS Lambda	2,678,400 requests/month (1/sec) 128 MB of memory 500 ms/request	$3.33
	TOTAL:	**$3.33 per month**

Option 3: Deploy the AWS CloudFormation template using Amazon MSK, AWS Lambda, and Amazon Kinesis Data Firehose

The following table provides a cost estimate to deploy the `streaming-data-solution-for-msk-using-aws-lambda-and-kinesis-data-firehose` AWS CloudFormation template that uses Amazon MSK, AWS Lambda, Kinesis Data Firehose, and Amazon Simple Storage Service (Amazon S3).

Table for Option 3: Cost estimate for running the solution using the AWS CloudFormation template that deploys Amazon MSK, Lambda, Kinesis Data Firehose, and Amazon S3

AWS service	Dimensions	Cost per month
Lambda	2,678,400 requests/month (1/sec) 128 MB of memory 500 ms/request	$3.33
Kinesis Data Firehose	100 records (4 KB)/second	$36.34
Amazon S3	1 GB storage (Amazon S3 standard)	$0.02
	TOTAL:	**$39.69 per month**

Option 4: Deploy the AWS CloudFormation template using Amazon MSK, Amazon Kinesis Data Analytics, and Amazon S3

The following table provides a cost estimate to deploy the `streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3` AWS AWS CloudFormation template that uses Amazon MSK, Amazon Kinesis Data Analytics, and Amazon Simple Storage Service (Amazon S3).

Table for Option 4: Cost estimate for running the solution using the AWS CloudFormation template that deploys Amazon MSK, Amazon Kinesis Data Analytics, and Amazon S3

AWS service	Dimensions	Cost per month
Kinesis Data Analytics	1 processing unit 50 GB running application storage	$80.30 $5.00
Amazon S3	1 GB storage (Amazon S3 standard)	$0.02
	TOTAL:	**$85.32 per month**

Architecture overview

This solution automatically configures the core AWS services necessary to capture, store, process, and deliver streaming data.

All AWS CloudFormation resources were created using AWS Solutions Constructs.

Option 1: Deploy the AWS CloudFormation template using Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Deploying the `streaming-data-solution-for-msk` AWS CloudFormation template builds the following environment in the AWS Cloud.

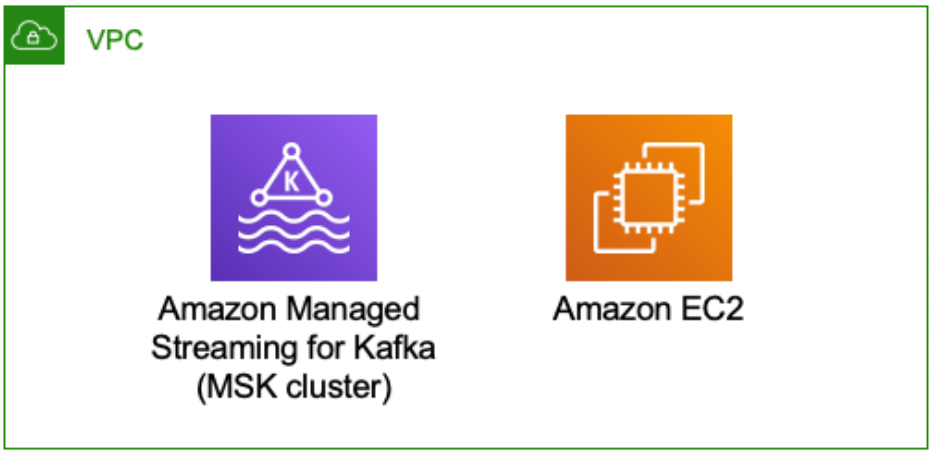

Figure 1: AWS CloudFormation template using Amazon MSK reference architecture

This AWS CloudFormation template deploys a reference architecture that includes the following:

1. An Amazon MSK cluster.
2. An Amazon EC2 instance that contains the Apache Kafka client libraries required to communicate with the MSK cluster. This client machine is located on the same VPC as the cluster, and it can be accessed via AWS Systems Manager Session Manager.

Option 2: Deploy the AWS CloudFormation template using Amazon MSK and AWS Lambda

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

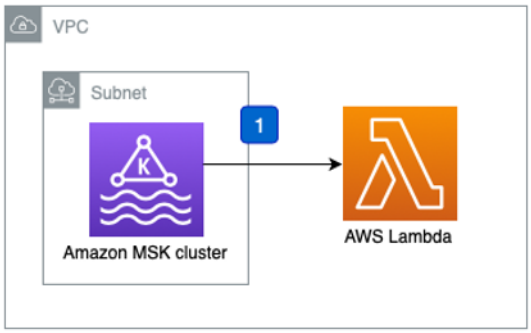

Figure 2: AWS CloudFormation template using Amazon MSK and Lambda reference architecture

This AWS CloudFormation template deploys a reference architecture that includes the following:

1. A Lambda function that processes process records in a Kafka topic. The default function is a Node.js application that logs the received messages, but it can be customized to fit your business needs.

Option 3: Deploy the AWS CloudFormation template using Amazon MSK, AWS Lambda, and Amazon Kinesis Data Firehose

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

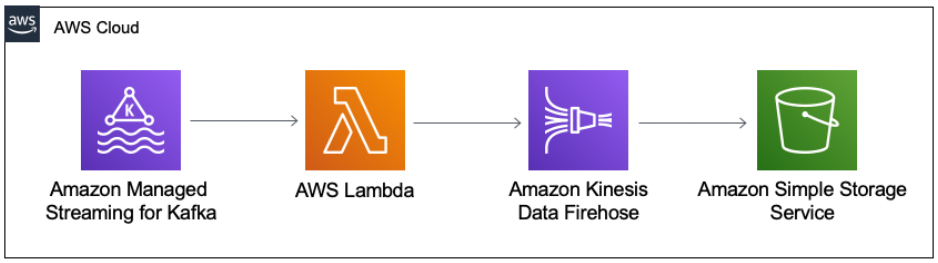

Figure 3: AWS CloudFormation template using Kinesis Data Streams, Kinesis Data Firehose, and S3 reference architecture

This AWS CloudFormation template deploys a reference architecture that does the following:

1. An AWS Lambda function that processes process records in an Apache Kafka topic.
2. A Kinesis Data Firehose delivery stream that buffers data before delivering it to the destination.
3. An Amazon S3 bucket that stores all original events from the Amazon MSK cluster.

Option 4: Deploy the AWS CloudFormation template using Amazon MSK, Amazon Kinesis Data Analytics, and Amazon S3

Deploying the `streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3` AWS CloudFormation template builds the following environment in the AWS Cloud.

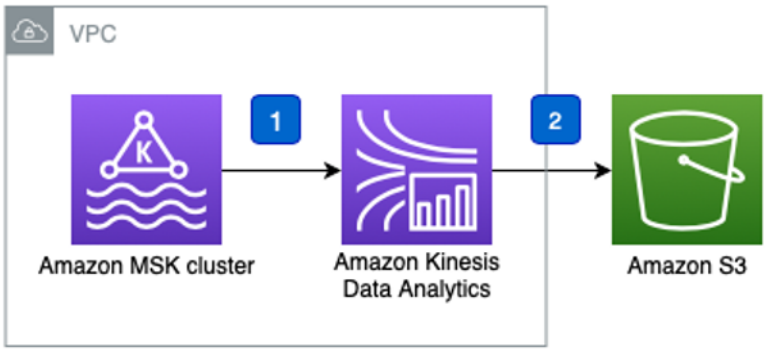

Figure 4: AWS CloudFormation template using Amazon MSK, Amazon Kinesis Data Analytics, and Amazon S3 reference architecture

This AWS CloudFormation template deploys a reference architecture that includes the following:

1. A Kinesis Data Analytics Studio notebook application that reads events from an existing topic in an Amazon MSK cluster.
2. An Amazon S3 bucket that stores the output.

Solution components

Component details for all templates.

Components for option 1: Amazon MSK

CloudWatch dashboards and alerts

Option 1 deploys an Amazon CloudWatch dashboard that monitors the health of the Amazon MSK cluster. You can customize the dashboards and alerts using Amazon CloudWatch or the source code from the solution's GitHub repository.

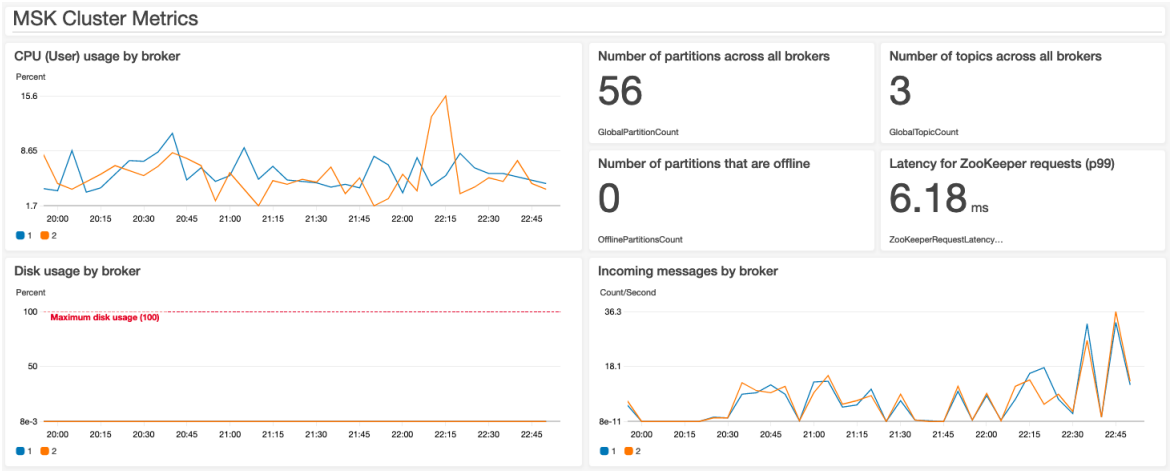

Figure 5: Amazon MSK health metrics on the CloudWatch dashboard (upper)

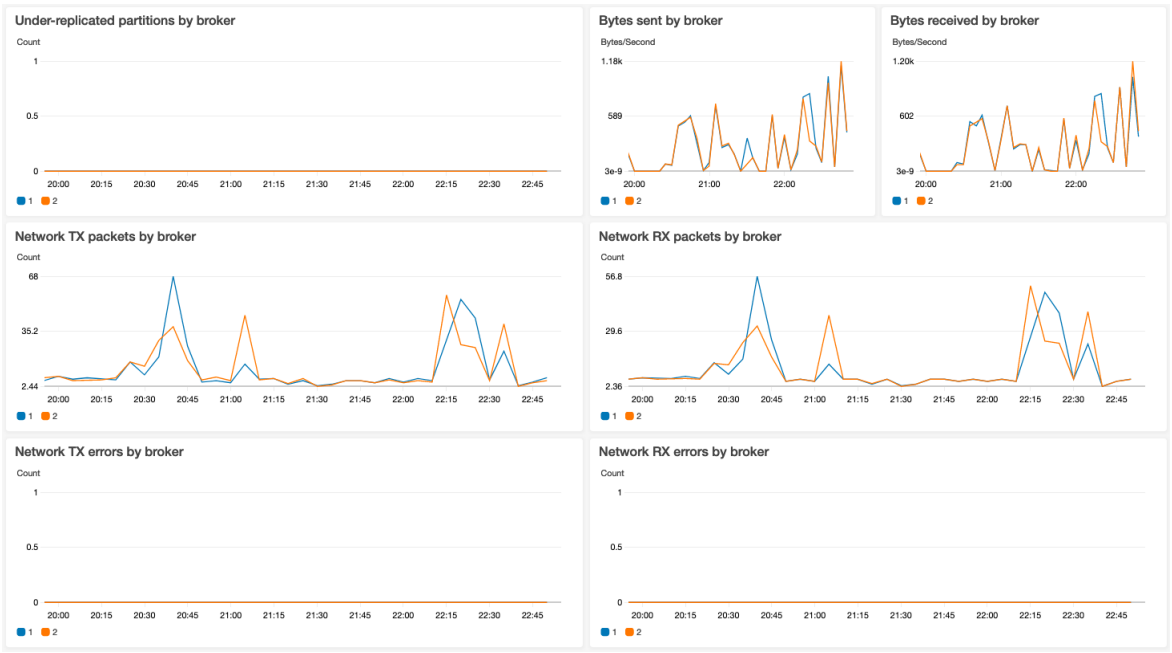

Figure 6: Amazon MSK health metrics on the CloudWatch dashboard (lower)

Components for option 4: Amazon MSK, Amazon Kinesis Data Analytics, and Amazon S3

CloudWatch dashbards and alerts

Option 4 deploys an Amazon CloudWatch dashboard that monitors the health of the Apache Flink application. You can customize the dashboards and alerts using either Amazon CloudWatch, or the source code from the solution's GitHub repository.

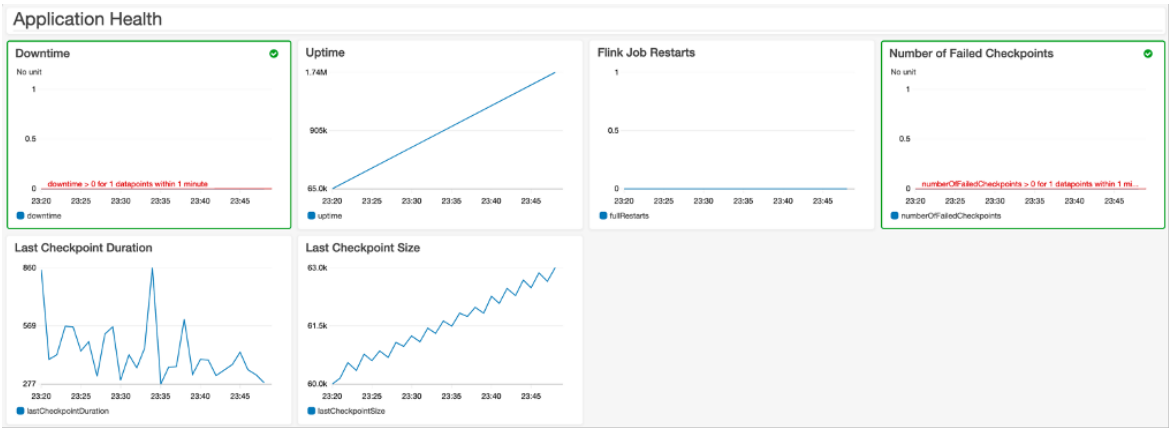

Figure 7: Application Health on the CloudWatch dashboard

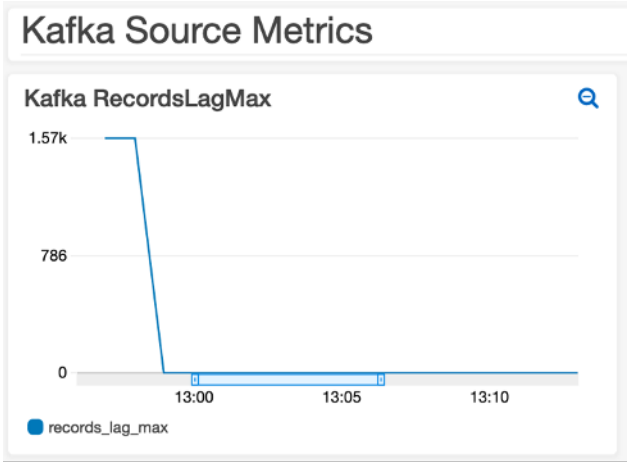

Figure 8: Kafka Source Metrics on the CloudWatch dashboard

Studio notebook

Option 4 deploys an Amazon Kinesis Data Analytics Studio notebook powered by Apache Zeppelin and Apache Flink to interactively analyze streaming data.

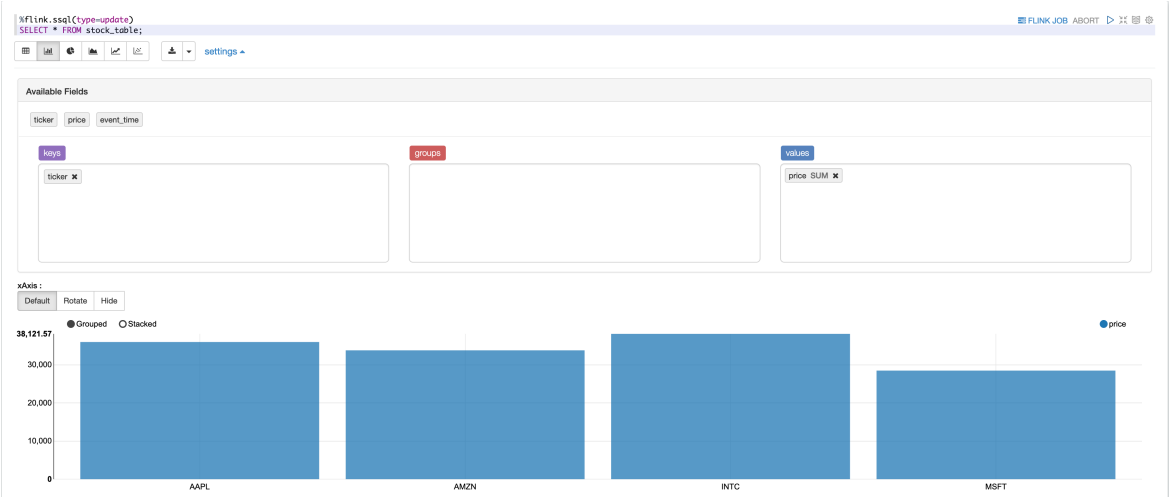

Figure 9: Example query on the Studio notebook

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 AWS security, refer to 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 MSK cluster so that it can communicate with the other solution components. This security group only includes the minimal rules required for Apache Kafka to work properly.

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 Amazon MSK API Calls with AWS CloudTrail
- Logging AWS Lambda API calls with AWS CloudTrail
- Logging Kinesis Data Analytics API Calls with AWS CloudTrail

AWS CloudFormation templates

This solution uses AWS CloudFormation to automate the deployment of the Streaming Data Solution for Amazon Amazon MSK in the AWS Cloud. You can download the following CloudFormation templates to deploy and customize to meet your needs:

Option 1: **streaming-data-solution-for-msk.template:** Use this template to launch this solution using Amazon MSK.

Option 2: **streaming-data-solution-for-msk-using-aws-lambda.template:** Use this template to launch this solution using Amazon Managed Streaming for Apache Kafka (Amazon MSK) and AWS Lambda.

Option 3: **streaming-data-solution-for-msk-using-aws-lambda-and-data-firehose.template:** Use this template to launch the solution using Amazon MSK, Lambda, and Amazon Kinesis Data Firehose.

Option 4: **streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3.template:** Use this template to launch this solution using Amazon MSK, Amazon Kinesis Data Analytics, and Amazon S3.

Automated deployment

Prerequisites

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

- **Option 1:** Deploy the `streaming-data-solution-for-msk.template` AWS CloudFormation template using Amazon Managed Streaming for Apache Kafka (Amazon MSK).
- **Option 2:** Deploy the `streaming-data-solution-for-msk-using-aws-lambda.template` AWS CloudFormation template using Amazon MSK and AWS Lambda.
- **Option 3:** Deploy the `streaming-data-solution-for-msk-using-aws-lambda-and-kinesis-data-firehose.template` AWS CloudFormation template using Amazon MSK, Lambda, and Amazon Kinesis Data Firehose.
- **Option 4:** Deploy the `streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3.template` AWS CloudFormation template using Amazon MSK, Amazon Kinesis Data Analytics, and Amazon S3.

Option 1: Deploy the streaming-data-solution-for-msk CloudFormation template

Before you launch this template, review the architecture and other considerations 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 25-30 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 (p. 14)

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

Step 2. (Optional) Create a topic that produces and consumes data (p. 16)

Launch the Stack

Note
You are responsible for the cost of the AWS services used while running this solution. Refer to the Cost (p. 3) 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 below to launch the `streaming-data-solution-for-msk.template` 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 this solution in a different AWS Region, use the Region selector in the console navigation bar.

   Note
   This template uses Amazon MSK, which is not currently available in all AWS Regions. You must launch this solution in an AWS Region where Amazon MSK is available. For the most current availability by Region, refer to the AWS Regional Services List.

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. 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
Broker configuration		
Apache Kafka version **(KafkaVersion)**	`2.8.1`	Apache Kafka version on the brokers.
Number of broker nodes **(NumberBrokerNodes)**	`3`	Number of broker nodes you want in the cluster (must be a multiple of the number of subnets).
Broker instance type **(BrokerInstanceType)**	`kafka.m5.large`	Amazon EC2 instance type that Amazon MSK uses when it creates your brokers.
Monitoring level **(MonitoringLevel)**	`DEFAULT`	Level of monitoring for the cluster. The available options include `DEFAULT`, `PER_BROKER`, `PER_TOPIC_PER_BROKER`and `PER_TOPIC_PER_PARTITION`.
Amazon EBS storage volume per broker (in GiB) (EbsVolumeSize)	`1000`	Size (in GiB) of the storage volume in each broker node. The allowed range is from `1` to `16384`.
Access control configuration		
Method Amazon MSK uses to authenticate clients	`IAM role-based authentication`	The available options are `Unauthenticated access`, `IAM role-based`

Parameter	Default	Description
(AccessControlMethod)		`authentication`, and `SASL/SCRAM authentication`.
Networking configuration		
Cluster VPC **(BrokerVpcId)**	*<Requires input>*	VPC where the cluster launch.
Cluster subnets **(BrokerSubnetIds)**	*<Requires input>*	List of subnets in which brokers are distributed (must contain between 2 and 3 items).
Client configuration		
Instance type **(ClientInstanceType)**	`t3.small`	Instance type for the client instance.
Amazon Machine Image **(ClientAmiId)**	`1`	Amazon Machine Image (AMI) ID for the client instance.

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 25 minutes.

   Note
   This solution includes the `solution-helper` Lambda function, which runs only during initial configuration. This function is only created if you start the collection of operation metrics. For details, refer to Collection of operational metrics (p. 25).

Step 2. (Optional) Create a topic that produces and consumes data

After the stack is created, you can use the Amazon EC2 client instance to interact with the Amazon MSK cluster.

1. Sign in to the Amazon MSK console and, from the left menu pane, select **Clusters**.
2. On the **Amazon MSK** page, select `kafka-cluster-`*<account-id>*.
3. Choose **View client information** then copy the values for **ZooKeeper connection** and **Bootstrap servers**.
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 *<KafkaClient>* 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 create a topic:

```
sudo su cd /home/kafka/bin
./kafka-topics.sh --create --zookeeper<zookeeper-connection-string> --replication-
factor 3 --partitions 1 --topic MyTopic
./kafka-console-producer.sh --broker-list<broker-list> --producer.config config-file --
topic MyTopic
```

Note
The client configuration file depends on the access control method selected when launching the stack. For **Unauthenticated access**, use `client-ssl.properties`; for **IAM role-based authentication**, use `client-iam.properties`; and for **SASL/SCRAM**, use `client-sasl.properties`

Option 2: Deploy the streaming-data-solution-for-msk-using-aws-lambda CloudFormation template

Before you launch this template, review the architecture and other considerations 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

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 (p. 3) 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 below to launch the `streaming-data-solution-for-msk-using-aws-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 this 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 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. 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
AWS Lambda consumer configuration		

Parameter	Default	Description
ARN of the MSK cluster **(ClusterArn)**	*<Requires input>*	ARN of the Amazon MSK cluster.
Maximum number of items to retrieve in a single batch **(BatchSize)**	`100`	The maximum number of records to retrieve in a single batch. The allowed range is from `1` to `10000`.
Name of a Kafka topic to consume **(TopicName)**	*<Requires input>*	The name of the Apache Kafka topic to consume.
Secret ARN for SASL/SCRAM authentication **(SecretArn)**	*<Optional input>*	ARN of the AWS Secrets Manager secret containing the username and password to be used for authentication with the cluster.

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.

Option 3: Deploy the streaming-data-solution-for-msk-using-aws-lambda-and-kinesis-data-firehose CloudFormation template

Before you launch this template, review the architecture and other considerations 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

Launch the Stack

Note
You are responsible for the cost of the AWS services used while running this solution. Refer to the Cost (p. 3) 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 below to launch the `streaming-data-solution-for-msk-using-aws-lambda-and-kinesis-data-firehose` 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 this 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 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. 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
AWS Lambda consumer configuration		
ARN of the MSK cluster **(ClusterArn)**	*<Requires input>*	ARN of the Amazon MSK cluster.
Maximum number of items to retrieve in a single batch **(BatchSize)**	`100`	The maximum number of records to retrieve in a single batch. The allowed range is from `1` to `10000` hours.
Name of a Kafka topic to consume **(TopicName)**	*<Requires input>*	The name of the Apache Kafka topic to consume.
Secret ARN for SASL/SCRAM authentication (SecretArn)	*<Optional input>*	ARN of the AWS Secrets Manager secret containing the username and password to be used for authentication with the cluster.
Amazon Kinesis Data Firehose configuration		
Size of the buffer (in MBs) that incoming data is buffered before delivery **(BufferingSize)**	`5`	The size to buffer incoming data before delivering to S3. The allowed range is from `1` to `128`.
Length of time (in seconds) that incoming data is buffered before delivery **(BufferingInterval)**	`300`	The amount of time to buffer incoming data before delivering to S3. The allowed range is from `60` to `900`.

Parameter	Default	Description
Compression format for delivered data in Amazon S3 **(CompressionFormat)**	`GZIP`	The format of data once it's delivered to S3. Allowed values are `GZIP`, `HADOOP_SNAPPY`, `Snappy`, `UNCOMPRESSED`, and `ZIP`.

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 ten minutes.

Option 4: Deploy the streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3 CloudFormation template

Before you launch this template, review the architecture and other considerations 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

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 (p. 3) 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 below to launch the `streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3` 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 this 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 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. 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 MSK cluster configuration		
ARN of the MSK cluster (ClusterArn)	*<Requires input>*	ARN of the Amazon MSK cluster.
Amazon Kinesis Data Analytics configuration		
Monitoring log level (LogLevel)	`INFO`	The level of detail of the CloudWatch logs for an application. The available options include `DEBUG`, `ERROR`, `INFO`, and `WARN`. For information about choosing a log level, refer to Application Monitoring Levels in the *Amazon Kinesis Data Analytics Developer Guide*.

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 ten minutes.

Step 2. Post-configuration steps

By default, the Studio notebook will not run after the stacks are created. Use the following process to start the Studio notebook.

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, go to the **Studio** tab, and select **Kda*<studio-notebook-name>***.
3. Choose **Run**.

Additional Resources

AWS services

• Amazon CloudWatch • Amazon Elastic Compute Cloud • Amazon Kinesis Data Firehose • Amazon Managed Streaming for Apache Kafka • Amazon Simple Storage Service	• AWS CloudFormation • AWS Identity and Access Management • AWS Lambda • AWS Systems Manager • Amazon Kinesis Data Analytics

AWS documentation

Best practices for monitoring and data protection:

- Security in Amazon Managed Streaming for Apache Kafka
- Using Lambda with Amazon MSK
- Controlling Access to Apache ZooKeeper
- Security in Amazon Kinesis Data Analytics
- Viewing Kinesis Data Analytics Metrics and Dimensions

Amazon MSK Labs

- The Amazon MSK Labs are a learning resource that take you through getting started, a use case example of ingesting and analyzing real-time clickstream data, and best practices for migrating your self-managed Apache Kafka cluster to Amazon MSK. They also showcase how to leverage advanced Amazon MSK features (such Cruise Control, TLS mutual authentication, and open monitoring), which can be applied to clusters created using the solution.

Uninstall the solution

You can uninstall the Streaming Data Solution for Amazon MSK using the AWS Management Console or the AWS Command Line Interface (AWS CLI). The CloudWatch dashboard (along with any changes made directly to CloudWatch) is deleted with the solution stack. 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 the solution stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to What Is the AWS Command Line Interface in the *AWS CLI User Guide*. After confirming 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

To prevent against accidental data loss, this solution is configured to retain the Amazon S3 buckets if you choose to delete the AWS CloudFormation stack. 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. Before 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.

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 activated, the following information is collected and sent to AWS:

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each Streaming Data solution for Amazon MSK deployment
- **Timestamp:** The UTC formatted timestamp of when the event occurred
- **Data:**The Region where the stack launched, request type (whether the stack was created, updated, or deleted), and details about the option chosen (for example, shard count, whether enhanced monitoring was enabled, buffering size, etc.). For example:

```
{'Pattern': 'MskStandalone', 'Version': 'v1.0.0', 'NumberOfBrokerNodes': '2', 'Region':
 'us-east-1', 'BrokerInstanceType': 'kafka.t3.small', 'MonitoringLevel': 'DEFAULT',
 '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:

1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

```
"Send" : {
"AnonymousUsage" : { "Data" : "Yes" }
},
```

   to:

```
"Send" : {
"AnonymousUsage" : { "Data" : "No" }
},
```

4. Sign in to the AWS CloudFormation console.
5. Select **Create stack**.
6. On the **Create stack** page, **Specify template** section, select **Upload a template file**.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in Launch the stack in the Automated deployment section of this guide.

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
November 2020	Initial release
January 2021	Release v1.3.0: Added support for Apache Kafka 2.7.0; added pattern for integration between Amazon MSK and Amazon Kinesis Data Analytics. For more information, refer to the CHANGELOG.md file in the GitHub repository.
April 2021	Release v1.4.0: Added new parameter that specifies the size for the storage in each of the broker nodes; Added support for partition-level monitoring. For more information, refer to the CHANGELOG.md file in the GitHub repository.
May 2021	Release v1.4.1: Added Support for Apache Kafka versions 2.8.0 and 2.6.2. For more information, refer to the CHANGELOG.md file in the GitHub repository.
July 2021	Release v1.5.0: Added support for IAM access control and SASL/SCRAM authentication; Added support for Apache Kafka version 2.7.1; Fixed location of GitHub repository for MSK Labs assets. For more information, refer to the CHANGELOG.md file in the GitHub repository.
November 2021	Release v1.6.0: Added support for clusters secured by IAM Access Control in options 2 and 3; Updated option 4 to use Amazon Kinesis Data Analytics Studio, which offers a serverless notebook to perform live data exploration. For more information, refer to the CHANGELOG.md file in the GitHub repository.
July 2022	Release v1.6.1: Security updates for the Gson package and the minimist and vm2 npm packages. For more information, refer to the CHANGELOG.md file in the GitHub repository.

Contributors

The following individuals contributed to this document:

- Tarek Abdunabi
- Daniel Pinheiro

Notices

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

AWS Streaming Data Solution for Amazon MSK is licensed under the terms of the of the Apache License Version 2.0 available at The Apache Software Foundation.