

## Amazon Kinesis

Amazon Kinesis is a platform on AWS that enables real-time processing of streaming data at scale. It's designed to handle large volumes of data generated from various sources such as website clickstreams, application logs, IoT devices, and more.

There are several services within the Amazon Kinesis suite:

1. **Amazon Kinesis Data Streams:** This service allows you to build custom applications that process or analyze streaming data in real-time.
2. **Amazon Kinesis Data Firehose:** This service is used to capture, transform, and load streaming data into AWS data stores or analytics services without needing to write custom code.
3. **Amazon Kinesis Data Analytics:** It enables you to analyze streaming data in real-time with standard SQL queries, without having to learn new programming languages or processing frameworks.
4. **Amazon Kinesis Video Streams:** This service allows you to stream video from connected devices to AWS for analytics, machine learning, and other processing.

Each of these services provides different capabilities for managing and analyzing streaming data, allowing developers to build real-time applications, perform analytics, and gain insights from their data streams.

## Use cases of AWS Kinesis:

Amazon Kinesis offers a versatile platform for real-time data processing, making it valuable across various industries and use cases:

1. **Real-time Analytics:** Businesses can analyze streaming data to gain insights into customer behavior, market trends, and operational efficiency in real-time. For example, analyzing website clickstreams to personalize content or detect anomalies.
2. **IoT Data Processing:** Internet of Things (IoT) devices generate vast amounts of data that need to be processed in real-time. Kinesis can ingest, process, and analyze IoT data streams to monitor device performance, detect failures, or trigger automated actions.
3. **Log and Event Data Processing:** Kinesis can handle high-volume log and event data streams from applications, servers, and infrastructure. It enables real-time monitoring, troubleshooting, and alerting for systems and applications.
4. **Fraud Detection and Security Monitoring:** By analyzing streaming data from various sources such as transactions, user activities, and system logs, organizations can detect and respond to security threats and fraudulent activities in real-time.
5. **Clickstream Analysis and Personalization:** Online retailers and content providers can use Kinesis to analyze clickstream data in real-time to optimize website navigation.

## To begin with the Lab:

1. Login to AWS Console and navigate to Kinesis. Choose this service accordingly.

The screenshot shows the Amazon Kinesis dashboard. At the top, there's a purple header bar with the Kinesis logo and a star icon. Below it, the text "Work with Real-Time Streaming Data". Underneath, a section titled "Top features" lists "Kinesis Data Streams" and "Kinesis Data Analytics".

2. From Kinesis dashboard, choose Kinesis data streams and click on create data stream.

The screenshot shows the "Amazon Kinesis services" page. It features a central heading "Collect, process, and analyze data streams in real time." Below this, there are three service cards: "Kinesis Data Streams" (Collect and store data streams), "Amazon Data Firehose" (Process and deliver data streams), and "Managed Apache Flink" (Process and analyze streaming data). To the right, a "Get started" sidebar offers options like "Kinesis Data Streams" (selected), "Amazon Data Firehose - new", and "Managed Apache Flink". A prominent orange "Create data stream" button is located at the bottom of this sidebar. On the far right, there's a "Pricing (Asia Pacific (Mumbai))" section and a "Blueprints" section.

3. First give it a name.

The screenshot shows the "Create data stream" configuration page. It has a title "Create data stream" with an "Info" link. Below it is a section titled "Data stream configuration". In this section, there is a "Data stream name" field containing the value "demostream". A note below the field states: "Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods."

4. Then select it to provisioned and for provisioned shards keep the capacity to 1.

## Data stream capacity Info

### Capacity mode

#### On-demand

Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.

#### Provisioned

Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.

### Provisioned shards

The total capacity of a stream is the sum of the capacities of its shards. Enter number of provisioned shards to see total data stream capacity.

1

Shard estimator

Minimum: 1, Maximum available: 200, Account quota limit: 200. Request shard quota increase [↗](#)

### Total data stream capacity

Shard capacity is determined by the number of provisioned shards. Each shard ingests up to 1 MiB/second and 1,000 records/second and emits up to 2 MiB/second. If writes and reads exceed capacity, the application will receive throttles.

### Write capacity

#### Maximum

1 MiB/second and 1,000 records/second

### Read capacity

#### Maximum

2 MiB/second

 Provisioned mode has a fixed-throughput pricing model. See [Kinesis pricing for Provisioned mode ↗](#)

## 5. Then just create your stream.

### Data stream settings

You can edit the settings after the data stream has been created and is in the active status.

Setting	Value	Editable after creation
Capacity mode	Provisioned	 Yes
Provisioned shards	1	 Yes
Data retention period	1 day	 Yes
Server-side encryption	Disabled	 Yes
Monitoring enhanced metrics	Disabled	 Yes
Tags	-	 Yes
Data stream sharing policy	No policy	 Yes

Cancel

Create data stream

## 6. In some time, you will see that it is in active state.

7. In order to send and receive data from the stream, you have to use the API, the application programming interface. Or you can read and write the data from programs. So, there are a lot of languages such as Dotnet, Java that have the software development kit in place that can work with Amazon Kinesis.
8. Now you need to download two zip files from GitHub one is to Send data and other for Receiving date. After downloading and unzipping it you need to open the Send data program in Visual Studio.
9. In this program remember to change the name of stream with your given name if you are using a different name.

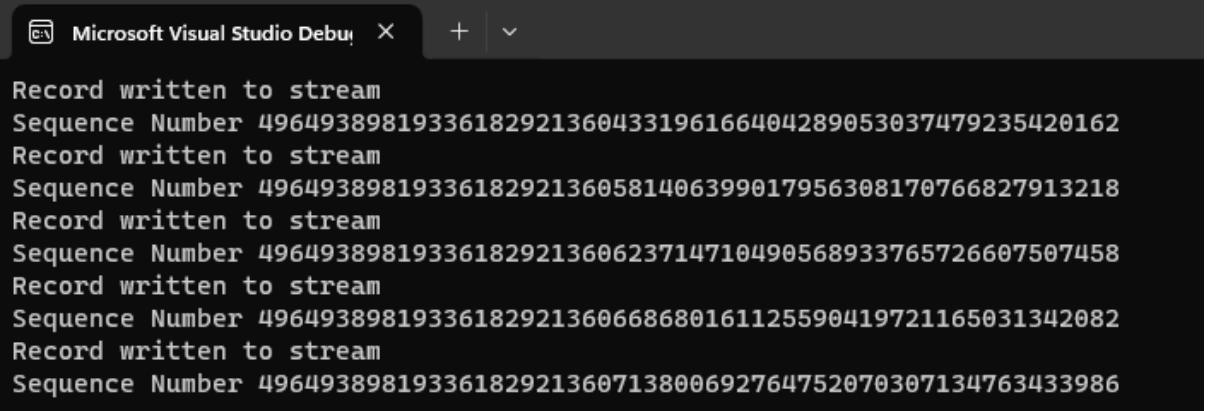
```

1  using Amazon;
2  using Amazon.Kinesis;
3  using Kinesis_SendData;
4  using Amazon.Kinesis.Model;
5  using Newtonsoft.Json;
6  using System.Text;
7
8  var kinesisClient = new AmazonKinesisClient(RegionEndpoint.APSouth1);
9
10 for (int i = 1; i <= 5; i++)
11 {
12     Logdata data=new Logdata();
13     data.logId = i;
14     data.logDetails= $"LogDetails{i}";
15
16     // We need to convert our data into bytes
17     byte[] RequestInBytes = Encoding.UTF8.GetBytes(JsonConvert.SerializeObject(data));
18
19     using (MemoryStream ms = new MemoryStream(RequestInBytes))
20     {
21         var recordRequest = new PutRecordRequest()
22         {
23             StreamName = "demostream",
24             PartitionKey = Convert.ToString(data.logId),
25             Data = ms
26         };
27
28         var response = await kinesisClient.PutRecordAsync(recordRequest);
29         Console.WriteLine("Record written to stream");
30         Console.WriteLine($"Sequence Number {response.SequenceNumber}");
31     }
32 }
33
34 
```

10. After that you should be having an IAM user in place on which you need to add the Kinesis permission.
11. Just add Kinesis full access with your user and configure it on your local machine.

Name	Type	Used as
<a href="#">AmazonKinesisFullAccess</a>	AWS managed	Permissions policy

12. Once this is all done you need to run your program. After your program has executed successfully you will receive this message.



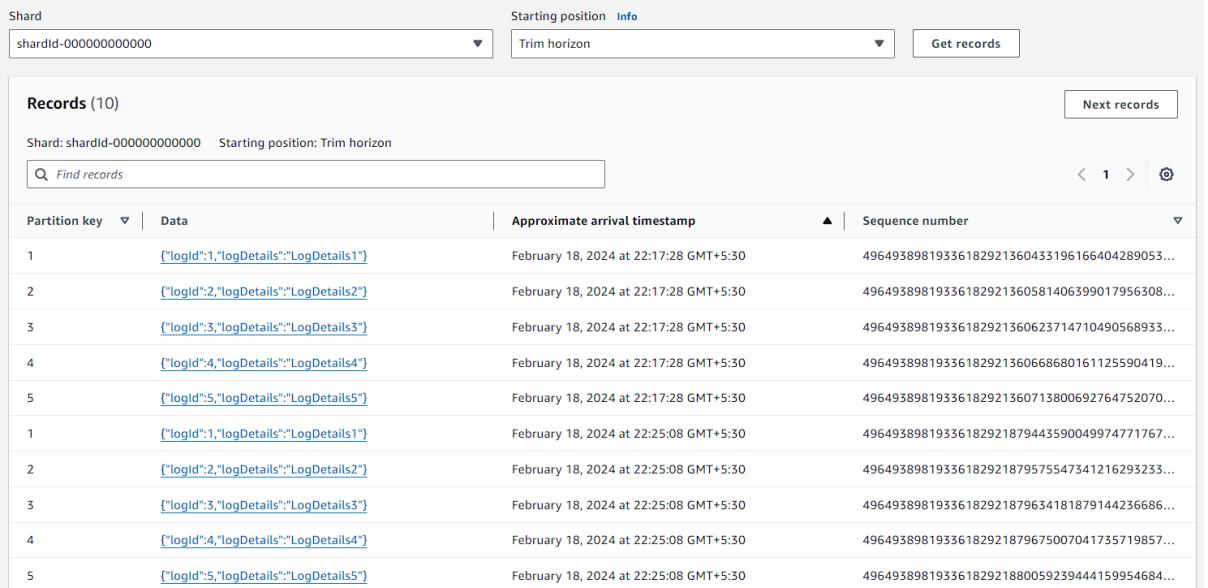
Record written to stream  
Sequence Number 49649389819336182921360433196166404289053037479235420162  
Record written to stream  
Sequence Number 49649389819336182921360581406399017956308170766827913218  
Record written to stream  
Sequence Number 49649389819336182921360623714710490568933765726607507458  
Record written to stream  
Sequence Number 49649389819336182921360668680161125590419721165031342082  
Record written to stream  
Sequence Number 49649389819336182921360713800692764752070307134763433986

13. After that go back to Kinesis on the console. Now from the options available in Kinesis click on data viewer.



14. Now in the data viewer choose your shard and starting position then click on get records.

15. You will see your records here. I had run the two times that is why there are 10 records otherwise there will only be 5 records when you will run this program for the first time.



Partition key	Data	Approximate arrival timestamp	Sequence number
1	{"logId":1,"logDetails":"LogDetails1"}	February 18, 2024 at 22:17:28 GMT+5:30	49649389819336182921360433196166404289053...
2	{"logId":2,"logDetails":"LogDetails2"}	February 18, 2024 at 22:17:28 GMT+5:30	49649389819336182921360581406399017956308...
3	{"logId":3,"logDetails":"LogDetails3"}	February 18, 2024 at 22:17:28 GMT+5:30	49649389819336182921360623714710490568933...
4	{"logId":4,"logDetails":"LogDetails4"}	February 18, 2024 at 22:17:28 GMT+5:30	49649389819336182921360668680161125590419...
5	{"logId":5,"logDetails":"LogDetails5"}	February 18, 2024 at 22:17:28 GMT+5:30	49649389819336182921360713800692764752070...
1	{"logId":1,"logDetails":"LogDetails1"}	February 18, 2024 at 22:25:08 GMT+5:30	49649389819336182921879443590049974771767...
2	{"logId":2,"logDetails":"LogDetails2"}	February 18, 2024 at 22:25:08 GMT+5:30	49649389819336182921879575547341216293233...
3	{"logId":3,"logDetails":"LogDetails3"}	February 18, 2024 at 22:25:08 GMT+5:30	49649389819336182921879634181879144236686...
4	{"logId":4,"logDetails":"LogDetails4"}	February 18, 2024 at 22:25:08 GMT+5:30	49649389819336182921879675007041735719857...
5	{"logId":5,"logDetails":"LogDetails5"}	February 18, 2024 at 22:25:08 GMT+5:30	4964938981933618292188005923944159954684...

16. Now open the Receive data program in Visual Studio. Again, remember to change the name of the stream.

17. Once you have done that just click on run.

The screenshot shows the Visual Studio IDE interface. The menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a Search bar. The toolbar has icons for file operations like Open, Save, and Print. The status bar at the bottom right shows Ln: 13 Ch: 23 SPC CRLF.

The code editor window displays the `Program.cs` file under the `Kinesis-ReceiveData` project. The code is written in C# and uses the Amazon Kinesis SDK. It demonstrates how to describe a stream, get shard iterators, and read from a Kinesis stream. The code includes imports for Amazon, Amazon.Kinesis, Amazon.Kinesis.Model, Kinesis\_ReceiveData, Newtonsoft.Json, System.Net, and System.Text. It creates an `AmazonKinesisClient`, sends a `DescribeStreamRequest` for a stream named "demostream", and then iterates over the shards to send a `GetShardIteratorRequest` for each shard, specifying `ShardIteratorType.TRIM_HORIZON`.

18. Now you will see the records if your program has executed successfully.

The screenshot shows a terminal or log window with a black background and white text. It displays five groups of log entries, each consisting of five lines labeled 1 through 5. The labels are repeated five times in a vertical sequence. The log entries likely represent the data being processed by the Kinesis stream.

```
1LogDetails1
2LogDetails2
3LogDetails3
4LogDetails4
5LogDetails5
1LogDetails1
2LogDetails2
3LogDetails3
4LogDetails4
5LogDetails5
|
```