Getting Started

What is Segment?
How Segment Works
Getting Started Guide
A Basic Segment Installation
Planning a Full Installation
A Full Segment Installation
Sending Data to Destinations
Testing and Debugging
What's Next
Use Cases

Guides

Connections

Unify

Engage

Privacy

Protocols

Segment App

API

Partners

Glossary

Config API

Help

Connection Modes

Device-mode	Cloud-mode
⊘ Web	✓ Web
⊘ Mobile	✓ Mobile
⊘ Server	✓ Server

Amazon Kinesis enables you to build custom applications that process or analyze streaming data for specialized needs. Amazon Kinesis Streams can continuously capture and store terabytes of data per hour from hundreds of thousands of sources such as website clickstreams, financial transactions, social media feeds, IT logs, and location-tracking events.

Amazon Kinesis is not compatible with IP Allowlisting

For more information, see the IP Allowlisting documentation.

Getting Started

To get started:

1. Create a Kinesis stream. Follow these instructions in order to create a new AWS Kinesis Stream.
    - A stream is composed of multiple shards, each of which provides a fixed unit of capacity. The total capacity of the stream is the sum of the capacities of its shards. Each shard corresponds to 1 MB/s of write capacity and 2 MB/s of read capacity. See the Amazon Kinesis Developer Guide for more information on estimating number of shards needed for your stream.
2. Create an IAM policy.
    1. Sign in to the Identity and Access Management (IAM) console.
    2. Follow these instructions to Create an IAM policy to allow Segment permission to write to your Kinesis Stream.
    3. Select the **Create Policy from JSON** option and use the following template policy in the `Policy Document` field. Be sure to change the `{region}`, `{account-id}` and `{stream-name}` with the applicable values.

    ```
    {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": [
                   "kinesis:PutRecord",
                   "kinesis:PutRecords",
                   "iam:SimulatePrincipalPolicy"
               ],
               "Resource": [
                   "arn:aws:kinesis:{region}:{account-id}:stream/{stream-name}",
                   "arn:aws:iam::{account-id}:role/{role-name}"
               ]
           }
       ]
    }
    ```

    - **NOTE:** A previous version of this policy document only granted `PutRecord` access, which could slow down Kinesis write times by disallowing file batching. Substitute the updated policy document above to grant Kinesis `PutRecords` (plural) and allow batching. We've also requested `iam:SimulatePrincipalPolicy`, which will allow us to verify that the IAM Role has the appropriate Kinesis permissions without invoking the Kinesis API.
3. Create an IAM role.
    1. Follow these instructions to Create an IAM role to allow Segment permission to write to your Kinesis Stream.
    2. When prompted to enter an Account ID, enter `595280932656`. Make sure to enable **Require External ID** and enter your Segment Source ID as the External ID. This can be found by navigating to **Settings > API Keys** from your Segment source homepage.
        - **NOTE:** If you have multiple sources using Kinesis, enter one of their source IDs here for now and then follow the procedure outlined in the Multiple Sources section once you've completed this step and saved your IAM role.
    3. When adding permissions to your new role, find the policy you created in step 2 and attach it.
4. Create a new Kinesis destination.
    1. In the Segment source that you want to connect to your Kinesis destination, click **Add Destination**. Search and select the **Amazon Kinesis** destination.

2. Enter the **Role Address**, **Stream Region**, **Stream Name**, and **Secret ID**.

   - **NOTE:** For security purposes, Segment sets your Workspace ID as your Secret ID. If you're using a Secret ID different from your Workspace ID, reach out to our support team so they can change it to make your account more secure.

Page

If you're not familiar with the Segment Specs, take a look to understand what the Page method does. An example call would look like:

```
  analytics.page();
```

Identify

If you're not familiar with the Segment Specs, take a look to understand what the Identify method does. An example call would look like:

```
analytics.identify('97980cfea0085', {
  email: 'gibbons@example.com',
  name: 'John Gibbons'
});
```

Track

If you're not familiar with the Segment Specs, take a look to understand what the Track method does. An example call would look like:

```
analytics.track("User Registered", {
  checkinDate: new Date(),
  myCoolProperty: "foobar",
});
```

Data Model

Let's say you're connecting your Segment customer data stream to Kinesis Stream `arn:aws:kinesis:{region}:{account-id}:stream/{stream-name}`. If you send Segment the following in a `track` call:

```
{
  "userId": "user_1",
  "event": "User Registered",
  "properties": {
    "plan": "Pro Annual",
    "account_type" : "Facebook"
  }
}
```

The Segment Kinesis destination issues a `PutRecord` request with the following parameters:

```
const payload = {
  Data: JSON.stringify(msg.json()),
  StreamName: this.settings.stream,
  PartitionKey: this.settings.useMessageId ? msg.field('messageId') : msg.userId() || msg.anonymousId()
}
const request = kinesis.putRecord(payload)
```

Segment uses the `messageId` or the `userId || anonymousId` as the `PartitionKey`. The partition key is used by Amazon Kinesis to distribute data across shards. Amazon Kinesis segregates the data records that belong to a stream into multiple shards, using the partition key associated with each data record to determine which shard a given data record belongs to.

Group

If you’re not familiar with the Segment Specs, take a look to understand what the Group method does.

An example group call is shown below:

```
analytics.group("0e8c78ea9d9dsasahjg", {
  name: "group_name",
  employees: 3,
  plan: "enterprise",
  industry: "Technology"
});
```

Troubleshooting

When you get started, Segment recommends using any of the open source Kinesis tailing utility to validate that data is flowing correctly.

Best Practices

Updating IAM role permissions for encryption

Extra permissions need to be added to the IAM role if using at-rest encryption on the Kinesis stream. An updated role policy like below resolves issues when submitting PutRecords into Kinesis stream using encryption:

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": ["kms:GenerateDataKey"],
  "Resource": "${aws_kms_key.kinesis_key.arn}"
 }, {
  "Effect": "Allow",
  "Action": ["kinesis:PutRecord", "kinesis:PutRecords"],
  "Resource": ["${aws_kinesis_stream.kinesis1.arn}"]
 }]
}
```

Multiple Sources

If you have multiple sources using Kinesis/Firehose, you have two options:

Attach multiple sources to your IAM role

To attach multiple sources to your IAM role:

1. Find the IAM role you created for this destination in the AWS Console in **Services > IAM > Roles**.
2. Select the role and navigate to the **Trust Relationships** tab.
3. Click **Edit trust relationship**. You’ll see a snippet that looks something that looks like this:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::595280932656:role/customer-kinesis-access"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "YOUR_SEGMENT_SOURCE_ID"
        }
      }
    }
  ]
}
```

4. Replace that snippet with the following, and replace the contents of the array with all of your source IDs.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::595280932656:role/customer-kinesis-access"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": ["YOUR_SEGMENT_SOURCE_ID", "ANOTHER_SOURCE_ID", "A_THIRD_SOURCE_ID"]
        }
      }
    }
  ]
}
```

Update IAM to Support PutRecords

The Kinesis destination defaults to use PutRecords. A previous version of the IAM policy document only grants `PutRecord` access, which slows down Kinesis write times and degrades data deliverability. Substitute the updated policy document above to grant Kinesis `PutRecords` (plural) and allow batching, like this:

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "kinesis:PutRecord",
               "kinesis:PutRecords",
               "iam:SimulatePrincipalPolicy"
           ],
           "Resource": [
               "arn:aws:kinesis:{region}:{account-id}:stream/{stream-name}",
               "arn:aws:iam::{account-id}:role/{role-name}"
           ]
       }
   ]
}
```

After you update the IAM policy, Segment systems default to use PutRecords for more efficient data transmission. This is a zero-downtime change and doesn't impact your data other than increasing the deliverability success rate.

Use secret ID

If you have many sources using Kinesis that it's impractical to attach all of their IDs to your IAM role, you can instead opt to set a secret ID. To set this value:

1. Go to **Connections > Destinations > Amazon Kinesis** for each of your Segment sources.

2. Click **Secret ID**.

   NOTE: For security purposes, Segment sets your Workspace ID as your Secret ID. If you're using a Secret ID different from your Workspace ID, reach out to our support team so they can change it and make your account more secure.

3. Find the IAM role you created for this destination in the AWS Console in **Services > IAM > Roles**.
4. Click on the role and navigate to the **Trust Relationships** tab.
5. Click **Edit trust relationship**. You should see a snippet that looks something that looks like this:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::595280932656:role/customer-kinesis-access"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "YOUR_SEGMENT_SOURCE_ID"
        }
      }
    }
  ]
}
```

6. Replace the value of `sts:ExternalId` (`"YOUR_SEGMENT_SOURCE_ID"`) with your Secret ID. In the case of requiring the use of multiple secretIds, replace the `sts:ExternalId` setting above with:

```
"sts:ExternalId": ["A_SECRET_ID", "ANOTHER_SECRET_ID"]
```

Engage

You can send computed traits and audiences generated using Engage to this destination as a **user property**. To learn more about Engage, schedule a demo.

For user-property destinations, an identify call is sent to the destination for each user being added and removed. The property name is the snake_cased version of the audience name, with a true/false value to indicate membership. For example, when a user first completes an order in the last 30 days, Engage sends an Identify call with the property `order_completed_last_30days: true`. When the user no longer satisfies this condition (for example, it's been more than 30 days since their last order), Engage sets that value to `false`.

When you first create an audience, Engage sends an Identify call for every user in that audience. Later audience syncs only send updates for users whose membership has changed since the last sync.

ℹ **Real-time to batch destination sync frequency**

Real-time audience syncs to Amazon Kinesis may take six or more hours for the initial sync to complete. Upon completion, a sync frequency of two to three hours is expected.

Settings

Segment lets you change these destination settings from the Segment app without having to touch any code.

SETTING	DESCRIPTION

SETTING	DESCRIPTION
AWS Kinesis Stream Region *(required)*	`string`, defaults to `us-west-2` . The Kinesis Stream's AWS region key
Role Address	`string`. The address of the AWS role that will be writing to Kinesis (ex: arn:aws:iam::874699288871:role/example-role)
Secret ID (Read-Only) *(required)*	`string`, defaults to `#SEGMENT_WORKSPACE_ID` . The External ID to your IAM role. This value is read-only. Reach out to support if you wish to change it. This value is also a secret and should be treated as a password.
AWS Kinesis Stream Name *(required)*	`string`. The Kinesis Stream Name
Use Segment Message ID	`boolean`, defaults to `FALSE` . You can enable this option if you want to use the Segment generated `messageId` for the **Partition Key**. If you have issues with too many `provisionedthroughputexceededexceptions` errors, this means that your Segment events are not being evenly distributed across your buckets as you do not have even user event distribution (*default partition key is* `userId` *or* `anonymousId`). This option should provide much more stable and even distribution.

This page was last modified: 02 Aug 2024

Need support?

Questions? Problems? Need more info? Contact Segment Support for assistance!

Visit our Support page

Help improve these docs!

Edit this page

Request docs change

Was this page helpful?

Yes

No