



Documentation

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

Device-mode	Cloud-mode
<input type="radio"/> Web	<input checked="" type="checkbox"/> Web
<input type="radio"/> Mobile	<input checked="" type="checkbox"/> Mobile
<input type="radio"/> Server	<input checked="" type="checkbox"/> Server

[Amazon Kinesis Firehose](#) provides way to load streaming data into AWS. It can capture, transform, and load streaming data into Amazon Kinesis Analytics, Amazon S3, Amazon Redshift, and Amazon Elasticsearch Service, enabling near real-time analytics with existing business intelligence tools and dashboards you're already using today. It's a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security.

Getting started

To get started:

1. Create at least one Kinesis Firehose delivery stream. You can follow these [instructions](#) to create a new delivery stream.

2. Create an IAM policy.

3. Sign in to the [Identity and Access Management \(IAM\) console](#).

4. Follow [these instructions](#) to create an IAM policy on the JSON to allow Segment permission to write to your Kinesis Firehose Stream.

5. 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": [
        "firehose:PutRecord"
      ],
      "Resource": [
        "arn:aws:firehose:{region}:{account-id}:deliverystream/{stream-name}"
      ]
    }
  ]
}
```

6. Create an IAM role.

7. Follow [these instructions](#) to create an IAM role to allow Segment permission to write to your Kinesis Firehose Stream.

8. When prompted to enter an Account ID, enter 595280932656.

9. Select the checkbox to enable **Require External ID**.

10. Enter your Secret ID as the **External ID**. This can be found in Segment by navigating to your Amazon Kinesis Firehose destination in Segment, going to the Settings tab, and clicking the Secret ID setting.

Note: If you have multiple sources using Kinesis, enter one of their Secret IDs here for now and then follow the procedure outlined in the [Multiple Sources](#) section at the bottom of this doc once you've completed this step and saved your IAM role.

11. When adding permissions to your new role, find the policy you created in step 2 and attach it.

12. Create a new Kinesis Firehose Destination.

13. In the Segment source that you want to connect to your Kinesis Firehose destination, click **Add Destination**.

14. Search and select the **Amazon Kinesis Firehose** destination and enter details for [these settings options](#).

Page

Take a look to understand what the [Page method](#) does. An example call would look like:

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

Identify

Take a look to understand what the [Identify method](#) does. An example Identify call is shown below:

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

Track

Take a look to understand what the [Track method](#) does. An example Track call is shown below:

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

Event mapping

To begin using the Kinesis Firehose destination, you must first decide on which Segment events you would like to route to which Firehose delivery streams. This mapping then needs to be defined in your destination settings.

Segment Track events can map based on their **event name**. For example, if you have an event called `User Registered`, and you want these events to be published to a Firehose delivery stream called `new_users`, create a row in your destination settings that looks like this:

← Back to Destination

Edit Settings

Save

Please input the Segment **event names** or **event types** on the left and the desired Firehose delivery stream destinations on the right. This mapping is required for all events you would like in Firehose

Map Segment Events to Firehose Delivery Streams

Please input the Segment **event names** or **event types** on the left and the desired Firehose delivery stream destinations on the right. This mapping is required for all events you would like in Firehose

User Registered

►

new_users

×

↺ Add row

⊕ Add Another Stream

Any Segment **event type** (for example, Page, Track, Identify, or Screen) can also be mapped. This enables you to publish all instances of a given Segment event type to a given stream. To do this, create a row with the event type and its corresponding delivery stream:

← Back to Destination

Edit Settings

Save

Please input the Segment **event names** or **event types** on the left and the desired Firehose delivery stream destinations on the right. This mapping is required for all events you would like in Firehose

×

Map Segment Events to Firehose Delivery Streams

Please input the Segment **event names** or **event types** on the left and the desired Firehose delivery stream destinations on the right. This mapping is required for all events you would like in Firehose

page

▶

your_stream_name

×

↻

 Add row

+

 Add Another Stream

Events can be defined **insensitive to case** so `Page` will be equivalent to `page`. The delivery stream name needs to be formatted exactly as it is on AWS.

If you would like to route all events to a stream, use an `*` as the event name.

Data model

Let's say you've decided to publish your Segment track events named `User Registered` to your Kinesis Firehose delivery stream named `online_registrations`. If you send Segment the following track call:

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

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

```
firehose.putRecord({
  Record: {
    Data: JSON.stringify(msg) + '\n'
  },
  DeliveryStreamName: 'online_registrations'
});
```

Segment appends a newline character to each record to allow for easy downstream parsing.

Group

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"
});
```

Best practices

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 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-firehose-access"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "YOUR_SECRET_ID"
        }
      }
    }
  ]
}
```

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

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

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 for a Secret ID:

1 Go to the Kinesis Firehose destination settings from each of your Segment sources.

2 Click **Secret ID**.

NOTE: For security purposes, Segment sets your Segment 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.

3 Find the IAM role you created for this destination in the AWS Console in **Services > IAM > Roles**.

4 Select the role and navigate to the **Trust Relationships** tab.

5 Click **Edit trust relationship**. You should see a snippet that looks something like this:

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

6 Replace the value of `sts:ExternalId` ("YOUR_SECRET_ID") with the Secret ID value from the previous step. 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.



Real-time to batch destination sync frequency

Real-time audience syncs to Amazon Kinesis Firehose 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
Map Segment Events to Firehose Delivery Streams	<code>mixed</code> , defaults to . Please input the Segment event names or event types on the left and the desired Firehose delivery stream destinations on the right. This mapping is required for all events you would like in Firehose
AWS Kinesis Firehose Region (required)	<code>string</code> , defaults to <code>us-west-2</code> . The Kinesis Firehose AWS region key
Role Address (required)	<code>string</code> . The address of the AWS role that will be writing to Kinesis Firehose (ex: <code>arn:aws:iam::874699288871:role/example-role</code>)
Secret ID (Read-Only) (required)	<code>string</code> , defaults to <code>#SEGMENT_WORKSPACE_ID</code> . 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.

This page was last modified: 15 Nov 2023

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

Get started with Segment

Segment is the easiest way to integrate your websites & mobile apps data to over 300 analytics and growth tools.

[Request Demo](#)

or

[Create free account](#)

Website Data Collection Preferences

