Amazon AppFlow

User Guide

Amazon AppFlow: User Guide

Table of Contents

What is Amazon AppFlow?

Amazon AppFlow is a fully-managed integration service that enables you to securely exchange data between software as a service (SaaS) applications, such as Salesforce, and AWS services, such as Amazon Simple Storage Service (Amazon S3) and Amazon Redshift. For example, you can ingest contact records from Salesforce to Amazon Redshift or pull support tickets from Zendesk to an Amazon S3 bucket.

In addition to this User Guide, you can also refer to the Amazon AppFlow API Reference.

Amazon AppFlow enables you to do the following:

- **Get started quickly** — Create data flows to transfer data between a source and destination in minutes, without writing any code.
- **Keep your data in sync** — Run flows on demand or on a schedule to keep data in sync across your SaaS applications and AWS services.
- **Bring your data together** — Aggregate data from multiple sources so that you can train your analytics tools more effectively and save money.
- **Keep track of your data** — Use Amazon AppFlow flow management tools to monitor what data has moved where and when.
- **Keep your data secure** — Security is a top priority. We encrypt your data at rest and in transit.
- **Transfer data privately** — Amazon AppFlow integrates with AWS PrivateLink to provide private data transfer over AWS infrastructure instead of public data transfer over the internet.

For a list of Amazon AppFlow Regions, see Amazon AppFlow Regions and Endpoints in the *AWS General Reference*.

Related AWS services

You can use the following services with Amazon AppFlow.

AWS CloudTrail

Amazon AppFlow is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon AppFlow. CloudTrail captures all API calls for Amazon AppFlow as events. The calls captured include calls from the Amazon AppFlow console and code calls to the Amazon AppFlow API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon AppFlow. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon AppFlow, the IP address from which the request was made, who made the request, when it was made, and additional details. For more information, see Logging Amazon AppFlow API calls with AWS CloudTrail in the *Amazon AppFlow User Guide*.

AWS CloudFormation

AWS CloudFormation provides a common language for you to model and provision AWS and third party application resources in your cloud environment. AWS CloudFormation allows you to use programming languages or a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts. This gives you a single source of truth for your AWS and third party resources. Amazon AppFlow now supports AWS CloudFormation for creating and configuring Amazon AppFlow resources along with the rest of your AWS infrastructure—in

a secure, efficient, and repeatable way. For more information, see AWS::AppFlow::ConnectorProfile and AWS::AppFlow::Flow in the *AWS CloudFormation User Guide*.

Amazon EventBridge

Amazon AppFlow integrates with Amazon EventBridge to receive events from Amazon AppFlow sources such as Salesforce. This enables you to publish events ingested by Amazon AppFlow to a partner event bus in Amazon EventBridge. Amazon AppFlow supports the ingestion of Salesforce Platform events and Change Data Capture events. You can configure rules in Amazon EventBridge to match patterns from events such as those from Salesforce, and then route them to AWS services such as AWS Lambda, AWS Step Functions, Amazon Simple Queue Service, and others. You can also use Amazon AppFlow's private data transfer option to ensure that events don't get exposed to the public internet during transfers between AWS and Salesforce, improving security and minimizing risks of Internet-based attack vectors. For more information, see the Amazon EventBridge documentation page in the *Amazon AppFlow User Guide*.

AWS Identity and Access Management (IAM)

IAM is an AWS service that helps an administrator securely control access to AWS resources. Amazon AppFlow integrates with the IAM service so that you can control who in your organization has access to Amazon AppFlow. As an AWS root user or an IAM user with administrator access, you can add one or more users to your AWS account. You can also grant different levels of access to new and existing users. You can grant access using predefined identity-based policies, or you can create your own custom policy. For more information, see AWS Identity and Access Management for Amazon AppFlow in the *Amazon AppFlow User Guide*.

Setting up Amazon AppFlow

Prerequisites

This section provides a list of the prerequisites for getting started with Amazon AppFlow.

- **AWS account setup** — If you don't have an AWS account, you must create one. For more information, see How to create and activate a new AWS account.
- **SaaS application setup** — You must verify that you have the required information about the source and destination applications, and that they meet the relevant configuration requirements. For application-specific requirements and setup instructions, see Supported source and destination applications (p. 9).
- **AWS CloudFormation OAuth** — If you want to use AWS CloudFormation to create a connector profile for connectors that implement OAuth (such as Salesforce, Slack, Zendesk, and Google Analytics), you must fetch the access and refresh tokens. You can do this by implementing your own UI for OAuth, or by retrieving them from elsewhere. Alternatively, you can use the Amazon AppFlow console to create the connector profile, and then use that connector profile in the flow creation AWS CloudFormation template.
- **Data encryption** — Amazon AppFlow encrypts your data and connection details during transit and at rest. For more information, see Data protection in Amazon AppFlow (p. 71). When you configure a flow, you specify an AWS Key Management Service CMK to use for encryption. You can choose the AWS managed customer master key (CMK) that Amazon AppFlow creates by default, named **AWSDefaultEncryptionKey**, or you can choose a customer managed CMK that you create. To create a CMK, see Creating symmetric CMKs in the *AWS Key Management Service Developer Guide*. For examples of how to set IAM permissions for KMS access, see Amazon AppFlow policy examples.
- **Identity and access management** — If you access AWS as an IAM user, your administrator must grant you the permissions required to create and run flows. For more information, see Identity and access management for Amazon AppFlow.

General information for all applications

This section provides a list of general information that applies to all supported source and destination applications.

Source and destination API limits

The API calls that Amazon AppFlow makes to data sources and destinations count against any API limits for that application. For example, if you set up an hourly flow that pulls 5 pages of data from Salesforce, Amazon AppFlow will make a total of 120 daily API calls (24x5=120). This will count against your 24-hour Salesforce API limit. Exact API limits can vary depending on your licensing with the SaaS application.

IP address ranges

Amazon AppFlow operates from the AWS IP address ranges shown in the *Amazon Web Services General Reference Guide*. Configuring a flow connection with an incorrect URL, URI, or IP address range can return a bad gateway error. If you encounter this error, we recommend deleting your connection and creating a new one with the correct URL, URI, or IP address range. For instructions on how to create a new connection for your SaaS application, see Supported source and destination applications (p. 9).

Note
You can't use IP allow listing in your S3 bucket policy to deny access to any other IP addresses besides Amazon AppFlow IP addresses. This is because Amazon AppFlow uses a VPC endpoint when placing data in your Amazon S3 buckets. For more information about Amazon AppFlow Regions and endpoints, see Amazon AppFlow Regions and Endpoints in the *AWS General Reference*.

Schema changes

Amazon AppFlow only supports the automatic import of newly created Salesforce fields into Amazon S3 without requiring the user to update their flow configurations. For other source applications, Amazon AppFlow does not currently support schema changes, but you can edit your flow to reload the fields and update your mapping. For more information on how to edit a flow, see Edit an Amazon AppFlow flow (p. 64).

Note
If the source or destination fields in a flow's configuration are deleted from the source or destination application (including Salesforce), then the flow run will fail. To prevent failed flows, we recommend that you edit your flows to remove deleted fields from the mapping.

Getting started with Amazon AppFlow

This tutorial provides a hands-on introduction to Amazon AppFlow. You create and configure a flow to move data between a data source and a data destination.

Tasks

Prerequisites

Ensure you have reviewed the Prerequisites (p. 3) for getting started with Amazon AppFlow.

Step 1: Create a flow

Provide basic information for your flow.

To create a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow. A valid flow name is a combination of alphanumeric characters and the following special characters: !@#.-_.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then select an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value. The following basic restrictions apply to tags:

   - Maximum number of tags per resource – 50
   - For each resource, each tag key must be unique, and each tag key can have only one value.
   - Maximum key length – 128
   - Unicode characters in UTF-8
   - Use letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
   - Tag keys and values are case-sensitive.
   - The `aws:` prefix is reserved for AWS use. If a tag has a tag key with this prefix, then you can't edit or delete the tag's key or value. Tags with the `aws:` prefix do not count against your tags per resource limit.
6. Choose **Next**.

Step 2: Configure the flow

Provide information about the source and destination for your flow.

To configure the flow

1. For **Source details**, select the source and provide the requested information. For example, provide connection information and select objects or events. For more information, look up your source application on the Supported source and destination applications (p. 9) page where you can find application-specific connection instructions.

   Note
   To successfully configure a connection for a flow, the user or role you use to create the flow must have permission to use the `UseConnectorProfile` permission-only action for the connection (connectorprofile) that you choose for the flow. This permission is included in the AmazonAppFlowFullAccess managed policy. If you are using a custom policy, you must add the permission to the policy and specify the connectorprofile resource in the policy.
2. For **Destination details**, select the destination and provide the requested information about the location. For more information, look up your destination application on the Supported source and destination applications (p. 9) page where you can find application-specific connection instructions.
3. For **Flow trigger**, choose how to trigger the flow. The following are the flow trigger options:

   - **Run on demand** - Run the flow manually.
   - **Run on event** - Run the flow based on the specified change event.
     - This option is available only for SaaS applications that provide change events. You must choose the event when you choose the source.
   - **Run on schedule** - Run the flow on the specified schedule and transfer the specified data.
     - You can choose either full or incremental transfer for schedule-triggered flows.
     - When you select full transfer, Amazon AppFlow transfers a snapshot of all records at the time of the flow run from the source to the destination.
     - When you select incremental transfer, Amazon AppFlow transfers only the records that have been added or changed since the last successful flow run. You can also select a timestamp field to specify how Amazon AppFlow identifies new or changed records. For example, if you have a **Created Date** timestamp field, choose this to instruct Amazon AppFlow to transfer only newly-created records (and not changed records) since the last successful flow run. The first flow in a schedule-triggered flow will pull 30 days of past records at the time of the first flow run.
     - The scheduling frequency depends on the frequency supported by the source application.
4. Choose **Next**.

   Tip
   Attempting a connection with an expired user login can return a 'status code 400' error. If you encounter this error, we recommend creating a new connection and deleting the old one, or using an existing connection with valid credentials. For more information on setting up a connection, look up your source application on the Supported source and destination applications (p. 9) page.

Step 3: Map data fields

Map the fields in the source objects to fields in the destination. This determines how data is transferred from the source to the destination.

To map data fields

1. For **Mapping method**, choose how to map the fields and complete the field mapping. The following are the field mapping options:
   - **Manually map fields** - Use the Amazon AppFlow user interface to specify the field mapping. To map all fields, choose **Source field name**, **Bulk actions**, **Map all fields directly**. Otherwise, select one or more fields from **Source field name**, **Source fields**, and then choose **Map fields directly**.
   - **Upload a .csv file with mapped fields** - Use a comma-separated values (CSV) file to specify the field mappings. Each line in the CSV file contains the source field name, followed by a comma, which is followed by the destination field name. For more information on how to create the CSV file for upload, see the note that follows this procedure.
2. (Optional) To add a formula that concatenates fields, select two fields from **Mapped fields** and then choose **Add formula**.
3. (Optional) To mask or truncate field values, select one or more fields from **Mapped fields** and then choose **Modify values**.
4. (Optional) For **Validations**, add validations to check whether a field has bad data. For each field, choose the condition that indicates bad data and what action Amazon AppFlow should take when a field in a record is bad.
5. Choose **Next**.

Tip
When manually mapping between a source and destination, you must select compatible fields and be sure not to exceed the number of records supported by the destination. For more information on supported record quotas, see Quotas for Amazon AppFlow in the *Amazon AppFlow User Guide*.

Note
When creating a CSV file to upload to Amazon AppFlow, you must specify each source field and destination field pair in a single line separated by a comma. For example, if you want to map source fields SF1, SF2, and SF3 to destination fields DFa, DFb, and DFc respectively, the CSV file should contain three lines as follows:
SF1, DFa
SF2, DFb
SF3, DFc
Save your file with a .csv extension and then upload this file to import the mapping into Amazon AppFlow.

Step 4 (Optional): Add filters

Specify a filter to determine which records to transfer. Amazon AppFlow enables you to filter data fields by adding multiple filters and by adding criteria to a filter.

Note
When you select field names with string values, OR logic allows you to combine two or more criteria into a broader condition. When you add multiple filters, AND logic allows you to combine your filters into a narrower condition.

To add filters

1. To add a filter, choose **Add filter**, select the field name, select a condition, and then specify the criteria.
2. (Optional) To add further criteria to your filter, choose **Add criteria**. Depending on the field and the condition, you can add up to 10 criteria per filter.

3. (Optional) To add another filter, choose **Add filter** again. You can create up to 10 filters to specify which data fields you want to use in your flow. Amazon AppFlow will implement each filter in the order in which you specify them, and transfer only the records that meet all filter criteria.
4. To remove a filter, choose **Remove** next to the filter.
5. When you are finished adding filters, choose **Next**.

Step 5: Review and create

Review the information for your flow. To change the information for a step, choose **Edit**. When you are finished, choose **Create flow**.

Tip
If the flow creation fails, review the error message and confirm that all required fields have been entered, and that the user or role you are using has permission to the `UseConnectorProfile` action for the connection selected for the flow.

Supported source and destination applications

SaaS applications supported by Amazon AppFlow

Select an application from the following list to learn more about its setup requirements.

Amazon S3

The following are the requirements and connection instructions for using Amazon Simple Storage Service (Amazon S3) with Amazon AppFlow.

> **Note**
> You can use Amazon S3 as a source or a destination.

Topics

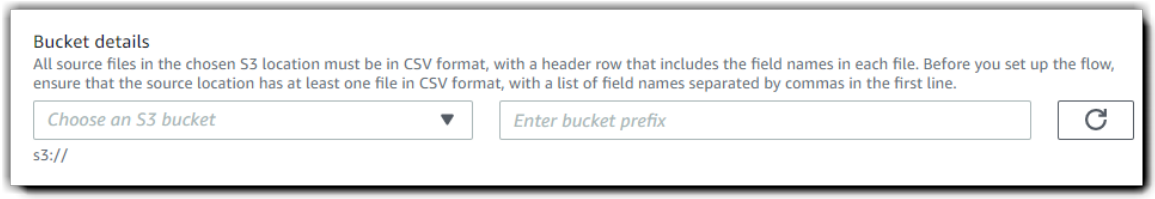
Requirements

- Your S3 buckets must be in the same AWS Region as your console and flow.
- If you use Amazon S3 as a source, all source files in the chosen S3 bucket must be in CSV format, with a header row that includes the field names in each file. Before you set up the flow, ensure that the source location has at least one file in CSV format, with a list of field names separated by commas in the first line. You must place the CSV file inside a folder in the S3 bucket.
- If the chosen input type is JSONL, you must place the JSONL files inside a folder in the S3 bucket
- Each source file should not exceed 125 MB in size. However, you can upload multiple CSV/JSONL files in the source location, and Amazon AppFlow will read from all of them to transfer data over a single flow run. You can check for any applicable destination data transfer limits in Quotas for Amazon AppFlow (p. 99).
- Amazon AppFlow does not support cross-account access to S3 buckets in order to prevent unauthorized access and potential security concerns.

Connection instructions

To use Amazon S3 as a source or destination while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Amazon S3** from the **Source name** or **Destination name** dropdown list.
8. Under **Bucket details**, select the S3 bucket that you're retrieving from or adding to. You can specify a prefix, which is equivalent to specifying a folder within the S3 bucket where your source files are located or records are to be written to the destination.



Now that you are connected to your S3 bucket, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

> **Tip**
> If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 10) section above.

Notes

- When you use Amazon S3 as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per minute.
- When you use Amazon S3 as a destination, the following additional settings are available.

Setting name	Description
Data format preference	• You can specify your preferred file format for the input file(s). The following options are currently available: CSV, JSONL • You can specify your preferred file format for the transferred records. The following options are currently available: JSONL (default), CSV, or Apache Parquet. **Note** If you choose Parquet as the format for your destination file in Amazon S3, the option to aggregate all records into one file per flow run will not be available. When choosing Parquet, Amazon AppFlow will write the output as string, and not declare the data types as defined by the source.
Data transfer preference	• You can choose between aggregation, and no aggregation of records. • By default, Amazon AppFlow transfers data into multiple files per flow run. • Alternatively, you can choose to aggregate all transferred data into one file per run flow.
Filename preference	• You can choose to add a timestamp to the filename. • Your filename will end with the file creation timestamp in YYYY-MM-DDThh:mm:sss format. • The creation date is in UTC time.
Folder structure preference	• You can choose to place the file in a timestamped folder. • You can choose your preferred level of granularity (year, month, week, day, or minute). • The granularity that you choose determines the naming format of the folder. • The timestamp is in UTC time.

Related resources

- Amazon Simple Storage Service User Guide
- Amazon AppFlow now supports new data formats for ingesting files into Amazon S3 in the AWS *What's new* blog
- Video: How to insert new Salesforce records with data in Amazon S3 using Amazon AppFlow
- Video: How to transfer data from Slack to Amazon S3 using Amazon AppFlow
- Video: How to transfer data from Google Analytics to Amazon S3 using Amazon AppFlow
- Video: How to transfer data from Zendesk Support to Amazon S3 using Amazon AppFlow

Amazon Redshift

The following are the requirements and connection instructions for using Amazon Redshift with Amazon AppFlow.

Note
You can use Amazon Redshift as a destination only.

Topics

Requirements

You must provide Amazon AppFlow with the following:

- The name and prefix of the S3 bucket that Amazon AppFlow will use when moving data into Amazon Redshift.
- The user name and password of your Amazon Redshift user account.
- The JDBC URL of your Amazon Redshift cluster. For more information, see Finding your cluster connection string in the *Amazon Redshift Cluster Management Guide*.

You must also do the following:

- Ensure that you enter a correct JDBC connector and password when configuring your Redshift connections. An incorrect JDBC connector or password can return an '[Amazon](500310)' error.
- Create an AWS Identity and Access Management (IAM) role that grants `AmazonS3ReadOnlyAccess` and access to the `kms:Decrypt` action (see the following example). This allows Amazon Redshift to access the encrypted data that Amazon AppFlow stored in the S3 bucket. Attach the role to your cluster.

  For more information, see Create an IAM role in the *Amazon Redshift Getting Started Guide Guide*.

```
{
    "Effect": "Allow",
    "Action": "kms:Decrypt",
    "Resource": "*"
}
```

- Ensure that your cluster is publicly accessible. For more information, see How to make a private Redshift cluster publicly accessible in the AWS Knowledge Center.
- Ensure that your Amazon Redshift cluster is accessible from Amazon AppFlow IP address ranges in your Region.

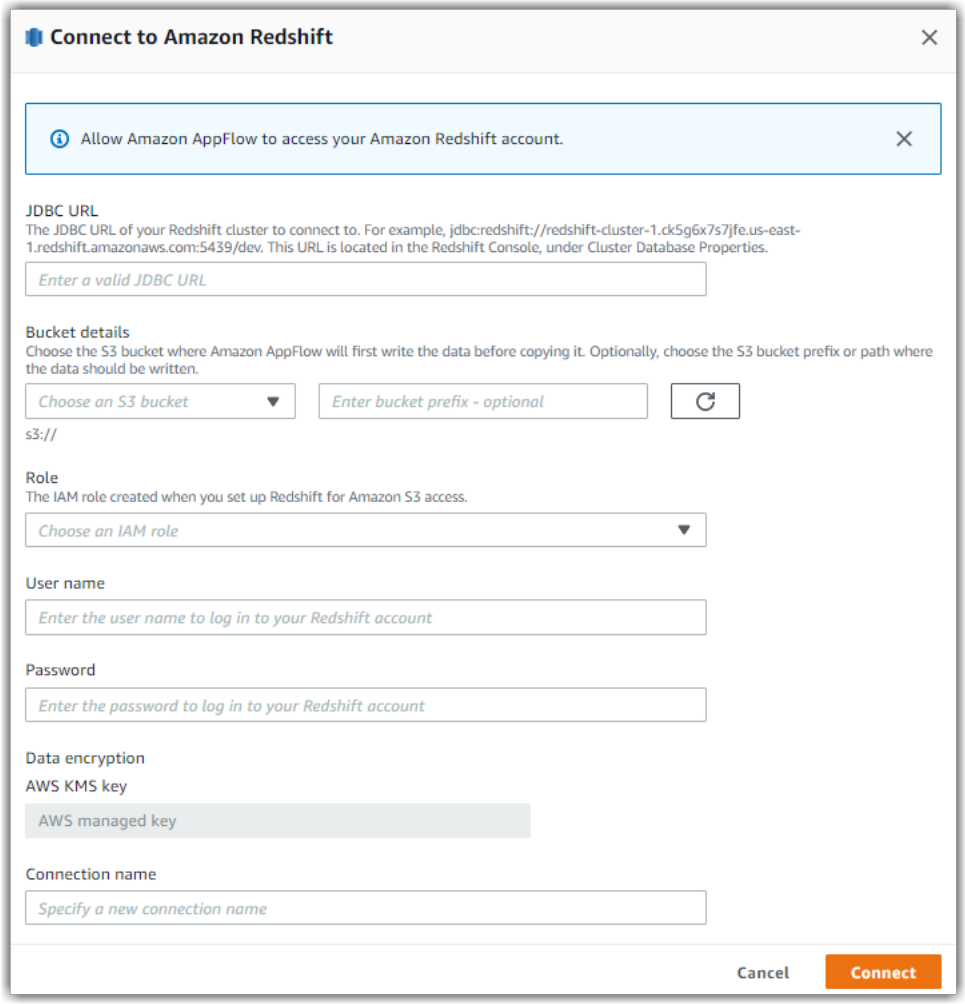
Connection instructions

To ensure that your Amazon Redshift cluster is accessible from Amazon AppFlow IP address ranges in your Region

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. Choose the cluster to modify.
3. Choose the link next to VPC security groups to open the Amazon Elastic Compute Cloud (Amazon EC2) console.
4. On the **Inbound Rules** tab, be sure that all Amazon AppFlow IP CIDR blocks for your region and the port of your Amazon Redshift cluster are allowed.

To connect to Amazon Redshift while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Amazon Redshift** from the **Destination name** list.
8. Choose **Connect** to open the **Connect to Amazon Redshift** dialog box.
    a. Under **JDBC URL**, enter your access key ID.
    b. Under **Bucket details**, select the S3 bucket where Amazon AppFlow will write data before copying it.
    c. Under **Role**, select the IAM role that you created when you set up Amazon Redshift for Amazon S3 access.
    d. Under **User name**, enter the user name that you use to log into Amazon Redshift.
    e. Under **Password**, enter the password that you use to log into Amazon Redshift.
    f. Under **Data encryption**, enter your AWS KMS key.
    g. Under **Connection name**, specify a name for your connection.
9. Choose **Connect**.



Now that you are connected to Amazon Redshift, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 12) section.

Notes

- The default port for Amazon Redshift is 5439, but your port might be different. To find the Amazon AppFlow IP CIDR block for your region, see AWS IP address ranges in the *Amazon Web Services General Reference*.
- Amazon AppFlow currently supports the insert action when transferring data into Amazon Redshift, but not the update or upsert action.

Related resources

- Finding your cluster connection string in the *Amazon Redshift Cluster Management Guide*
- How to make a private Redshift cluster publicly accessible in the AWS Knowledge Center
- Create an IAM role in the *Amazon Redshift Getting Started Guide Guide*

- Workaround to extract Salesforce data using Amazon AppFlow and upsert it to Amazon Redshift tables hosted on private subnet using data APIs in the Amazon AppFlow GitHub Page

Amazon EventBridge

The following are the requirements and connection instructions for using Amazon EventBridge with Amazon AppFlow.

> **Note**
> You can use Amazon EventBridge as a destination only.

Topics

Requirements

Amazon AppFlow integrates with Amazon EventBridge to receive events from Salesforce. When you configure a flow that responds to Salesforce events, you can choose Amazon EventBridge as a destination. This enables Salesforce events received by Amazon AppFlow to be routed directly to a partner event bus.

- To configure Amazon EventBridge integration in Amazon AppFlow, you must first create a flow with Amazon EventBridge as the destination and then specify the partner event source.
- Before you can activate the flow, you must go to Amazon EventBridge to associate the partner event source with the event bus. After you complete this association and activate the flow, Salesforce events start flowing to the Amazon EventBridge event bus.

Connection instructions

To create a flow with Amazon EventBridge as the destination

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow** and enter a name for your flow.
3. For **Source details**, choose **Salesforce** as the source and select **Salesforce Events** with the specific event name.
4. For **Destination details**, choose Amazon EventBridge as the destination and one of the following partner event sources:

   - **Existing partner event source** - Amazon AppFlow displays a list of existing partner event sources that are available to you.
   - **New partner event source** - Amazon AppFlow creates a new partner event source on your behalf. If you choose this option, the partner event source name generated by Amazon AppFlow appears in a dialog box. (Optional) You can modify this name if needed.

   > **Note**
   > The actual call to Amazon EventBridge API operations for creating this partner event source happens only when you choose **Create flow** in step 11 of this procedure.

5. For **Large event handling**, specify the S3 bucket where you want Amazon AppFlow to send large event information.
6. Ensure that **Run flow on event** is selected in the **Flow trigger** section. This setting ensures that the flow is executed when a new Salesforce event occurs.
7. For field mapping, choose **Map all fields directly**. Alternatively, you can choose the fields that you're interested in using from the **Source field name** list.
8. Choose **Next**.
9. (Optional) Configure filters for data fields in Amazon AppFlow.
10. Choose **Next**.
11. Review the settings and then choose **Create flow**.

To associate the partner event source with the event bus in Amazon EventBridge

1. Open the **Partner event sources** view in the Amazon EventBridge console at https://console.aws.amazon.com/events/home?#/partners/.
2. Choose the partner event source that you created.
3. Choose **Associate with event bus**.
4. Validate the name of the partner event bus.
5. Choose **Associate**.
6. Return to Amazon AppFlow and choose **Activate flow** to activate the flow.

Notes

- Events are limited to 256 KB. For events larger than 256 KB, Amazon AppFlow doesn't send the full event to Amazon EventBridge. Instead, the event payload contains a pointer to an S3 bucket, where you can get the full event.
- Events should be enabled in Salesforce and also in Amazon AppFlow for the destination to receive them. The destination service receives all such events configured for your account. If you need to filter the kinds of events that you want to process, or send different events to different targets, you can use content-based filtering with event patterns.

Related resources

- Receiving events from a SaaS partner in the *Amazon EventBridge* documentation
- Amazon AppFlow now supports Amazon EventBridge as a destination in the AWS *What's new* blog
- Building Salesforce integrations with Amazon EventBridge and Amazon AppFlow in the AWS *Compute* blog

Amazon Honeycode

The following are the requirements and connection instructions for using Amazon Honeycode with Amazon AppFlow.

Note
You can use Amazon Honeycode as a destination only. Amazon Honeycode is only available as an AppFlow destination in the AWS US West (Oregon) Region.

Topics

Connection instructions

To create a flow with Amazon Honeycode as the destination

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow** and enter a name for your flow.
3. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
4. Choose **Next**.
5. For **Source details**, choose one of the supported sources such as Salesforce, and provide the requested information.
6. For **Destination details**, choose Amazon Honeycode as the destination. If you are connecting to Amazon Honeycode for the first time, follow the instructions to complete the OAuth workflow and create a connection profile.
7. Select the workbook and table that are enabled in your account. You can select only one workbook and one table at a time.
8. Specify an error handling option to determine what action Amazon AppFlow takes if it can't write a record to the destination. If data can't be transferred to Amazon Honeycode, Amazon AppFlow writes that data to the Amazon S3 location of your choice. You can also choose to **Stop the current flow run** or **Ignore and continue the flow run**.
9. Choose a trigger for your flow. When using Amazon Honeycode as a destination, the **Run on demand** and **Run flow on schedule** options are available.
10. Choose **Next**.
11. For field mapping, choose **Map all fields directly**. Alternatively, you can manually select the fields that you want to use from the **Source field name** list.
12. (Optional) Under **Validations - optional**, add validations to check whether a field has bad data. For each field, choose the condition that indicates bad data and what action Amazon AppFlow should take when a field in a record is bad.
13. Choose **Next**.
14. (Optional) Specify a filter to determine which records to transfer. To add a filter, choose **Add filter**, select the field name, select a condition, and then specify the criteria.
15. Choose **Next**.
16. Review the settings and then choose **Create flow**.

Notes

- This integration with Amazon Honeycode currently supports the *append* functionality only. You can add new records to existing workbooks and tables, but you cannot update existing records at this time.

Related resources

- Amazon Honeycode User Guide

Amazon Lookout for Metrics

The following are the requirements and connection instructions for using Amazon Lookout for Metrics with Amazon AppFlow.

Note
You can use Amazon Lookout for Metrics as a destination only.

Topics

Requirements

- To get access to Amazon Lookout for Metrics, you must first be added to the allow list. To request access, see Amazon Lookout for Metrics Preview. For more information about the service, see Amazon Lookout for Metrics.

Setup instructions

To create a flow with Amazon Lookout for Metrics as the destination

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow** and enter a name for your flow.
3. Under **Data encryption**, choose **Customize encryption settings (advanced)** then select an existing customer managed key (CMK) or create a new one. The default AWS managed CMK is not supported when using Amazon Lookout for Metrics as a destination.
4. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
5. Choose **Next**.
6. For **Source details**, choose a supported source and provide the requested information.
7. For **Destination details**, choose Amazon Lookout for Metrics as the destination for your time-series data.
8. When using Amazon Lookout for Metrics as a destination, only the **Run flow on schedule** option is available. Specify the appropriate schedule settings, such as the frequency, start date, and start time. You can also enter an end date (optional).

   Amazon Lookout for Metrics currently supports the following scheduling options:

   - If the source supports minutes: you can run the flow every 5 or 10 minutes by selecting **5** or **10** from the **Every** dropdown list.
   - If the source supports hours: you can run the flow once an hour by selecting **1** from the **Every** dropdown list.
   - If the source supports days: you can run the flow once a day by selecting **1** from the **Every** dropdown list.
9. Choose **Next**.
10. Under **Source to destination field mapping**, go to the **Source field name** dropdown list and choose **Map all fields directly**. Alternatively, you can manually select the fields that you want to use from the list.

Note
A timestamp field is not required in your data. However, in order to use the anomaly detection feature of Amazon Lookout for Metrics, you need at least one measure or numeric column with values changing over time.

11. (Optional) Under **Validations - optional**, add validations to check whether a field has bad data. For each field, choose the condition that indicates bad data and what action Amazon AppFlow should take when a field in a record is bad.
12. Choose **Next**.
13. (Optional) Specify a filter to determine which records to transfer. To add a filter, choose **Add filter**, select the field name, select a condition, and then specify the criteria.
14. Choose **Next**.
15. Review the settings and then choose **Create flow**.

Notes

- The default AWS managed CMK is not supported when using Amazon Lookout for Metrics as a destination.
- The following sources are supported when using Amazon Lookout for Metrics as a destination:
  - Amplitude
  - Dynatrace
  - Google Analytics
  - Infor Nexus
  - Marketo
  - Salesforce
  - ServiceNow
  - Singular
  - Trend Micro
  - Veeva
  - Zendesk
- Amazon Lookout for Metrics currently supports the following scheduling options:
  - If the source supports minutes: you can run the flow every 5 or 10 minutes
  - If the source supports hours: you can run the flow once an hour
  - If the source supports days: you can run the flow once a day

Related resources

- Amazon Lookout for Metrics service page
- Amazon Lookout for Metrics Preview

Amplitude

The following are the requirements and connection instructions for using Amplitude with Amazon AppFlow.

Note
You can use Amplitude as a source only.

Topics

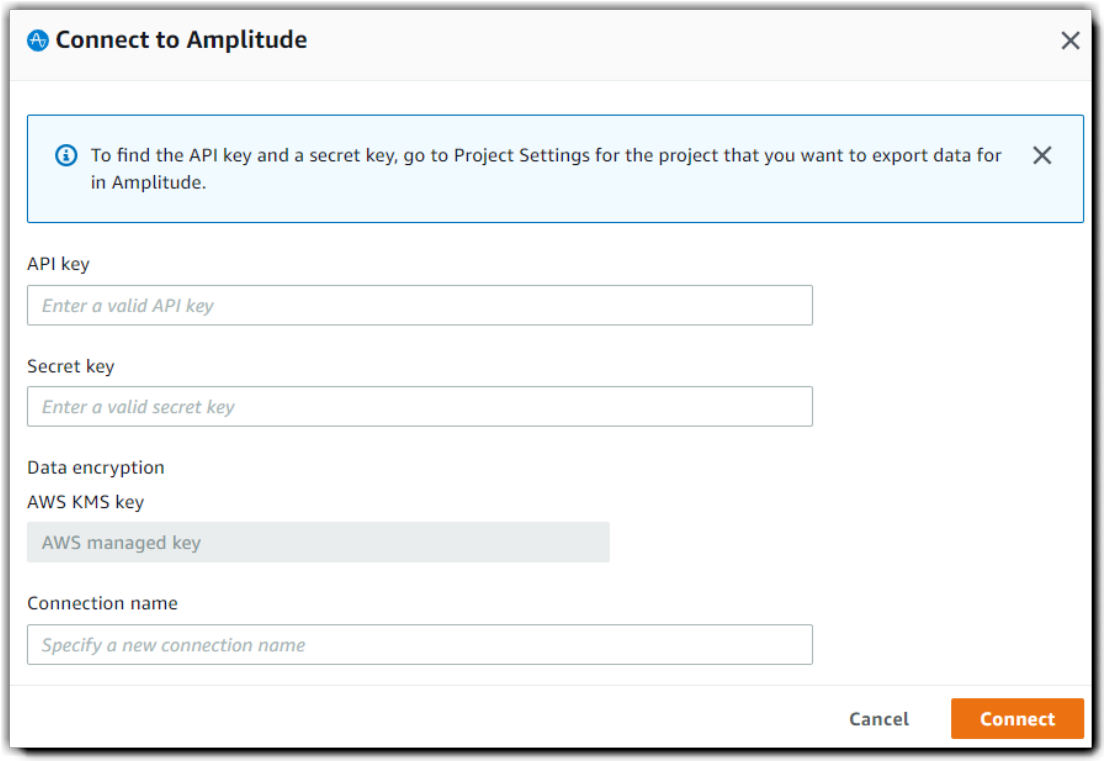
Requirements

You must provide Amazon AppFlow with the API key and secret key for the project with the data that you want to transfer. Your API key can be found on the Settings page of the Amplitude dashboard. For more information about how to retrieve this information from Amplitude, see Settings in the Amplitude documentation.

Connection instructions

To connect to Amplitude while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Amplitude** from the **Source name** dropdown list.
8. Choose **Connect** to open the **Connect to Amplitude** dialog box.
    a. Under **API key**, enter your API key.
    b. Under **Secret key**, enter your secret key.
    c. Under **Data encryption**, enter your AWS KMS key.
    d. Under **Connection name**, specify a name for your connection.
    e. Choose **Connect**.



9. You will be redirected to the Amplitude login page. When prompted, grant Amazon AppFlow permissions to access your Amplitude account.

Now that you are connected to your Amplitude account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 20).

Notes

- When you use Amplitude as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per day.
- Amplitude can process 25 MB of data as part of a single flow run.

Related resources

- Settings in the Amplitude documentation
- Breaking Data Silos with Amazon AppFlow and Amplitude from *Inside Amplitude*

Datadog

The following are the requirements and connection instructions for using Datadog with Amazon AppFlow.

Note
You can use Datadog as a source only.

Topics

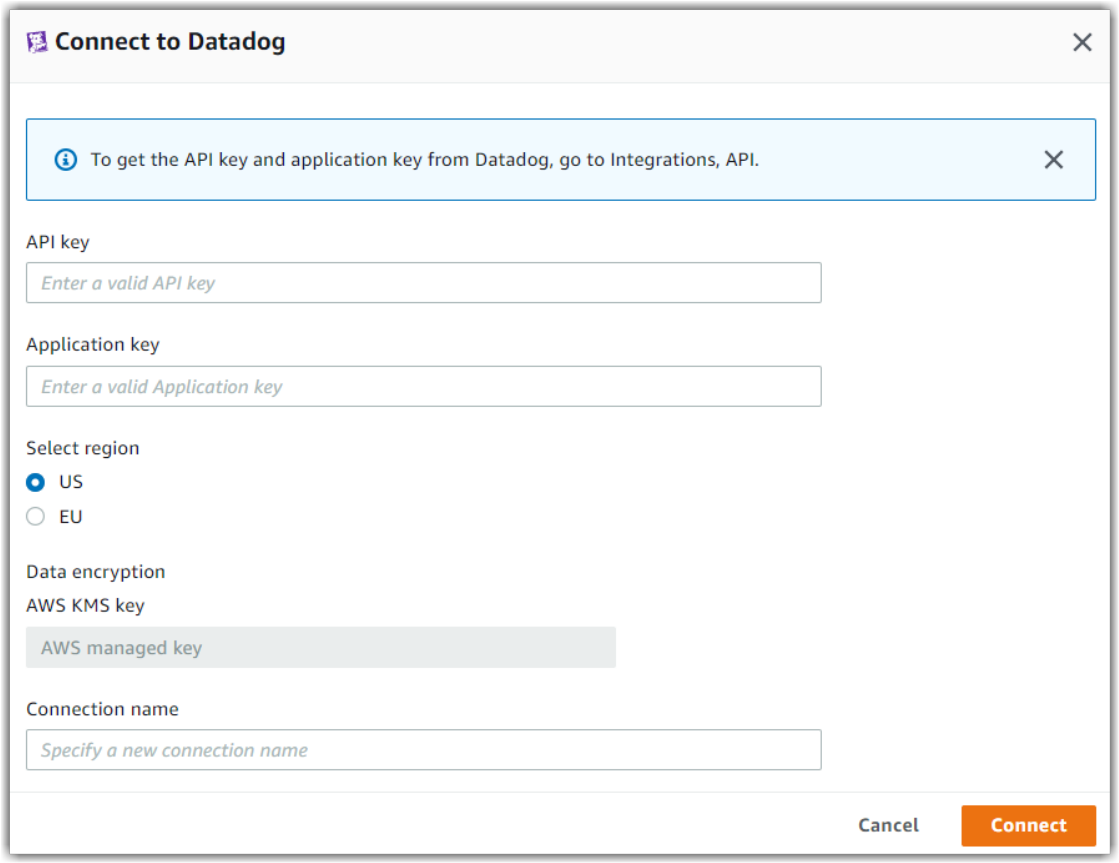
Requirements

- You must provide Amazon AppFlow with an API key and an application key. For more information about how to retrieve your API key and application key, see the API and Application Keys information in the Datadog documentation.
- You must configure your flow with a date range and query filter.

Connection instructions

To connect to Datadog while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Datadog** from the **Source name** dropdown list.
8. Choose **Connect** to open the **Connect to Datadog** dialog box.
    a. Under **API key**, enter your API key.
    b. Under **Application key**, enter your application key.
    c. Under **Select region**, select the region for your instance of Datadog.
    d. Under **Data encryption**, enter your AWS KMS key.
    e. Under **Connection name**, specify a name for your connection.
    f. Choose **Connect**.



9. You will be redirected to the Datadog login page. When prompted, grant Amazon AppFlow permissions to access your Datadog account.

Now that you are connected to your Datadog, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 22) section.

Notes

- When you use Datadog as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per minute.

Related resources

- API and Application Keys information in the *Datadog* documentation

Dynatrace

The following are the requirements and connection instructions for using Dynatrace with Amazon AppFlow.

Note
You can use Dynatrace as a source only.

Topics

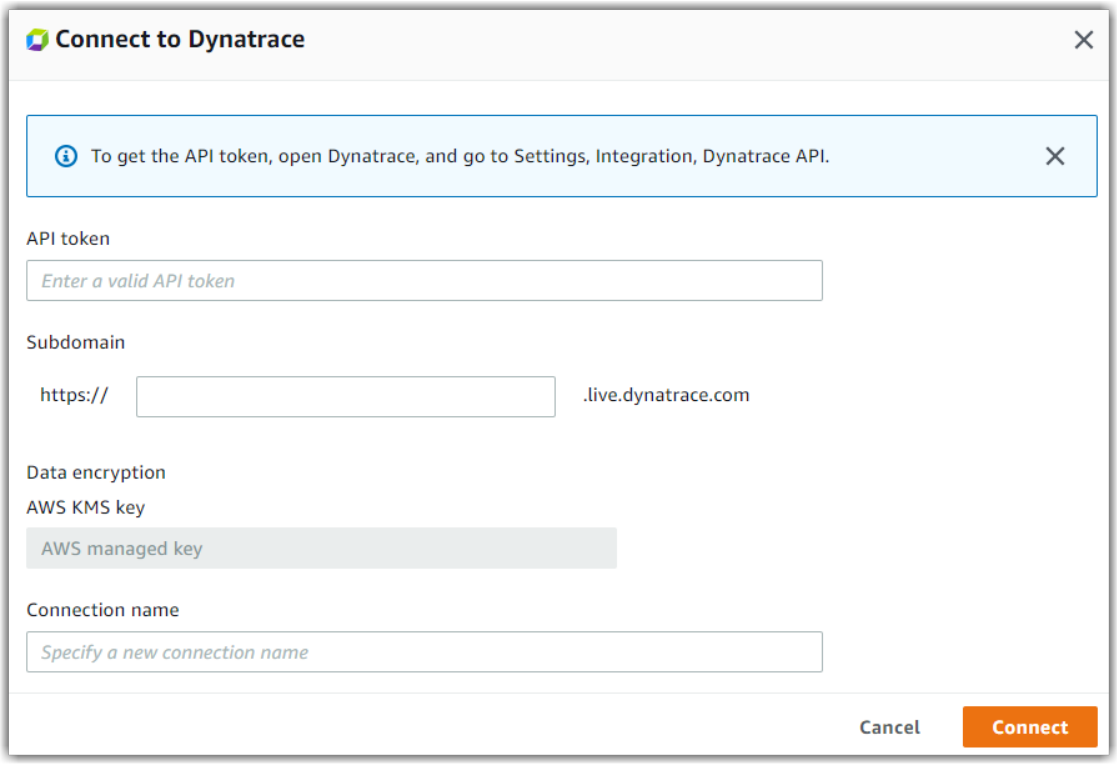
Requirements

- You must provide Amazon AppFlow with an API token. For more information about how to retrieve or generate an API token to use with Amazon AppFlow, see the Access tokens instructions in the Dynatrace documentation.
- You must configure your flow with a date filter with a date range that does not exceed 30 days.

Connection instructions

To connect to Dynatrace while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Dynatrace** from the **Source name** dropdown list.
8. Choose **Connect** to open the **Connect to Dynatrace** dialog box.
    a. Under **API token**, enter your API token.
    b. Under **Subdomain**, enter the subdomain for your instance of Dynatrace.
    c. Under **Data encryption**, enter your AWS KMS key.
    d. Under **Connection name**, specify a name for your connection.
    e. Choose **Connect**.



9. You will be redirected to the Dynatrace login page. When prompted, grant Amazon AppFlow permissions to access your Dynatrace account.

Now that you are connected to your Dynatrace account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 24).

Notes

- When you use Dynatrace as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per minute.

Related resources

- Access tokens instructions in the Dynatrace documentation
- Dynatrace API documentation for more information about the types of data you can extract from Dynatrace
- Dynatrace is launch partner of Amazon AppFlow – a service for easy and secure data transfer from *Dynatrace Resources*

Google Analytics

The following are the requirements and connection instructions for using Google Analytics with Amazon AppFlow.

Note
You can use Google Analytics as a source only.

Topics

Requirements

You must log in to the Google API Console at https://console.developers.google.com and do the following:

- Activate the Analytics API.
- Create a new app named **AppFlow**. Set the user type as **Internal**. Add the scope for read only access and add `amazon.com` as an authorized domain.
- Create a new OAuth 2.0 client. Set the application type as **Web application**.
- Set the authorized JavaScript origins URL to `https://console.aws.amazon.com/`.
- Set the authorized redirect URL as follows:
  - https://console.aws.amazon.com/appflow/oauth for the us-east-1 Region
  - https://*region*.console.aws.amazon.com/appflow/oauth for all other Regions
- Provide Amazon AppFlow with your client ID and client secret. After you provide them, you are redirected to the Google login page. When prompted, grant Amazon AppFlow permissions to access your Google Analytics account.

For more information, see Management API - Authorization in the Google Analytics documentation.

Connection instructions

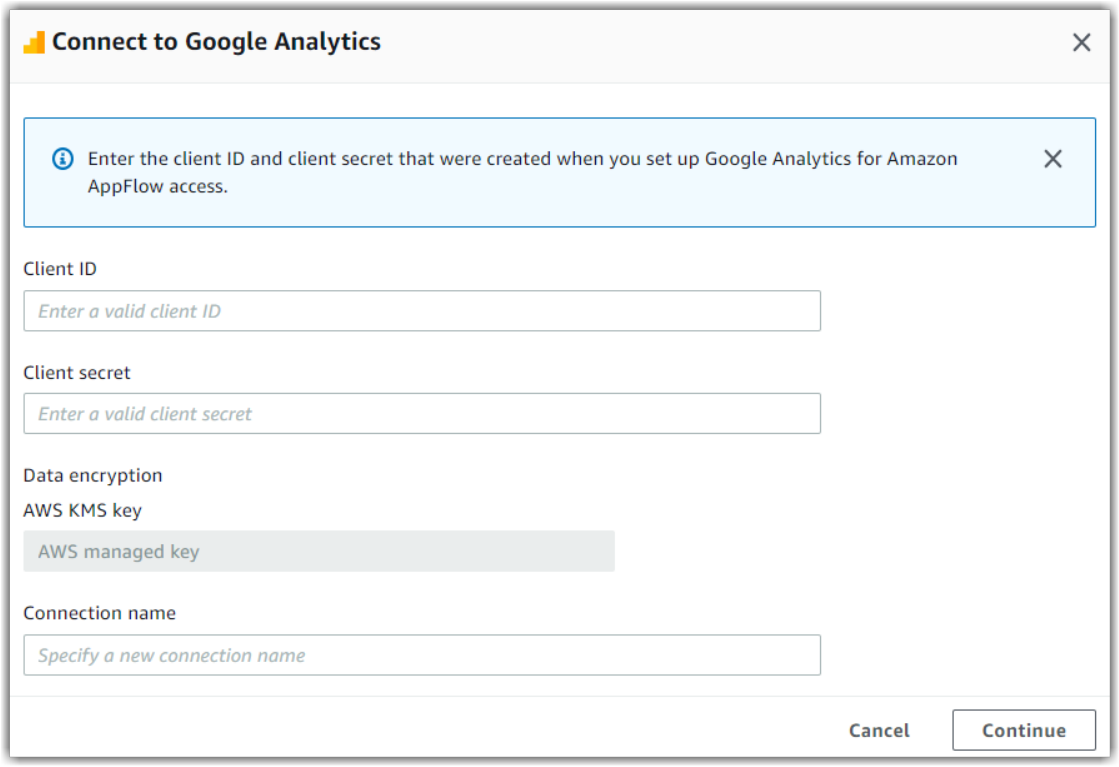
To connect to Google Analytics while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Google Analytics** from the **Source name** dropdown list.
8. Choose **Connect** to open the **Connect to Google Analytics** dialog box.
   a. Under **Client ID**, enter your client ID.
   b. Under **Client secret**, enter your client secret.
   c. Under **Secret access key**, enter your secret access key.

d. Under **Data encryption**, enter your AWS KMS key.

e. Under **Connection name**, specify a name for your connection.

f. Choose **Continue**.



9. You will be redirected to the Google Analytics login page. When prompted, grant Amazon AppFlow permissions to access your Google Analytics account.

Now that you are connected to your Google Analytics account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 26) section.

Notes

- When you use Google Analytics as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per day.
- Google Analytics can process 9 dimension and 10 metrics (including custom ones) as part of a single flow run.
- If you choose Google Analytics, you can only specify JSON as the data format for the Amazon S3 destination file.
- You can import custom dimensions and metrics from Google Analytics into Amazon S3. To specify custom dimensions or metrics, choose the **upload a .csv file with mapped field** option in the **Map data fields** step of the flow configuration. In the source field name in the CSV file, specify the custom dimension or the metric as `ga:dimension`*`XX`* or `ga:metric`*`XX`*, with *`XX`* containing the actual index (numerical value) that you provided to Google Analytics.

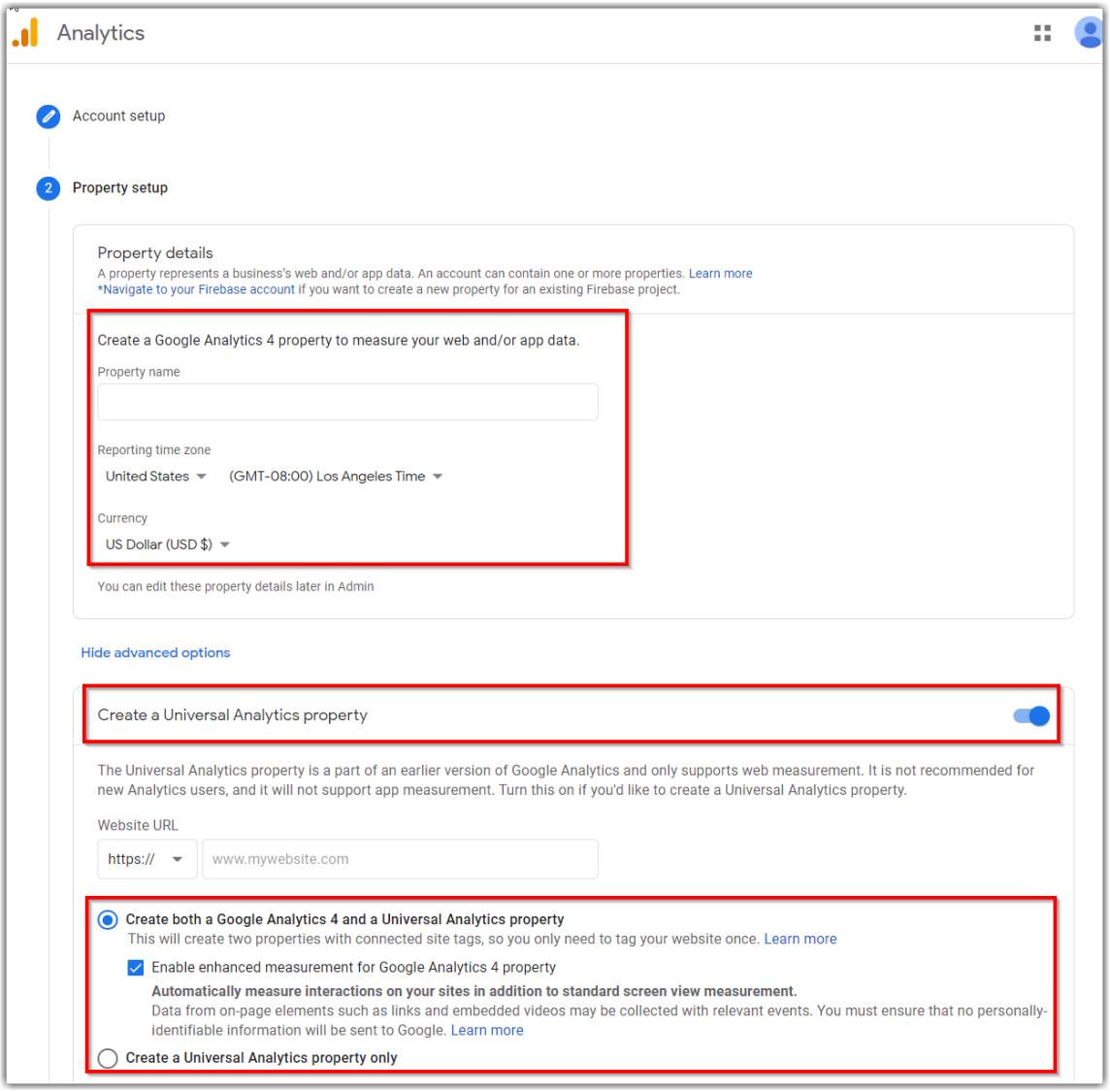
The following is an example row in the CSV file:

```
ga:dimension24|DIMENSION, PriceDimension
```

This imports the custom dimension in Google Analytics to a field named `PriceDimension` in the destination Amazon S3 file.

Note
The option to specify custom dimensions and metrics is available only when you upload a CSV file with mapped fields, and not when you manually map fields using the console.

- Google Analytics 4 properties are not yet supported. When you create a property in Google Analytics, you must select **Create both a Google Analytics 4 and a Universal Analytics Property** or **Create a Universal Analytics Property only**, as shown in the following screenshot. For more information, see Create a Property in the Google Analytics documentation.



Related resources

- Management API - Authorization in the Google Analytics documentation
- Create a Property in the Google Analytics documentation
- Analyzing Google Analytics data with Amazon AppFlow and Athena in the *AWS Big Data Blog*
- Video: How to transfer data from Google Analytics to Amazon S3 using Amazon AppFlow

Infor Nexus

The following are the requirements and connection instructions for using Infor Nexus with Amazon AppFlow.

Note
You can use Infor Nexus as a source only.

Topics

Requirements

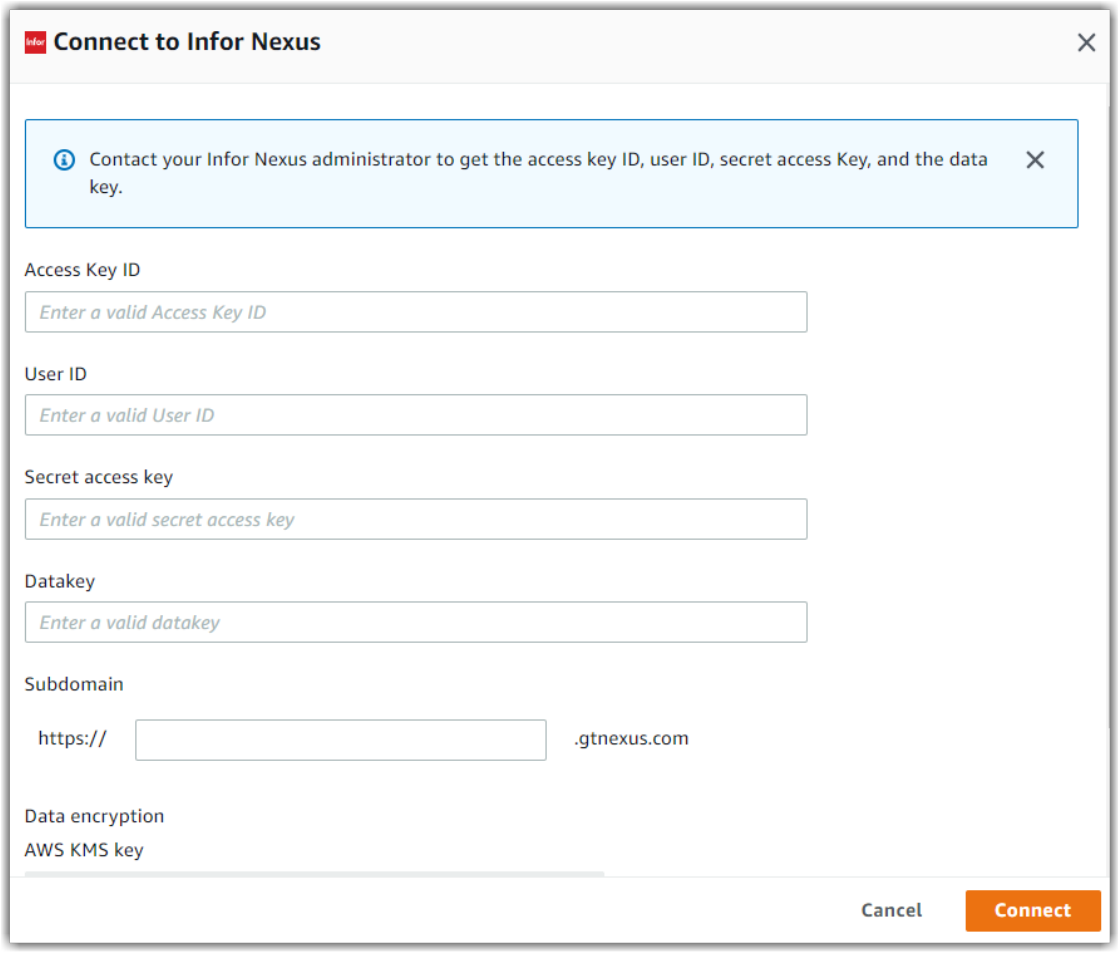
- Amazon AppFlow uses hash-based message authentication (HMAC) to connect to Infor Nexus.
- You must provide Amazon AppFlow with your access key ID, user ID, secret access key, and data key. To retrieve this information, contact your Infor Nexus administrator.

Connection instructions

To connect to Infor Nexus while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Infor Nexus** from the **Source name** dropdown list.
8. Choose **Connect** to open the **Connect to Infor Nexus** dialog box.
    a. Under **Access Key ID**, enter your access key ID.
    b. Under **User ID**, enter your Infor Nexus user ID.
    c. Under **Secret access key**, enter your secret access key.
    d. Under **Datakey**, enter your data key.
    e. Under **Subdomain**, enter the subdomain for your instance of Infor Nexus.
    f. Under **Data encryption**, enter your AWS KMS key.
    g. Under **Connection name**, specify a name for your connection.

h. Choose **Connect**.



9. You will be redirected to the Infor Nexus login page. When prompted, grant Amazon AppFlow permissions to access your Infor Nexus account.

Now that you are connected to your Infor Nexus account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 29) section.

Notes

- When you use Infor Nexus as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per minute.

Marketo

The following are the requirements and connection instructions for using Marketo with Amazon AppFlow.

Note
You can use Marketo as a source or destination.

Topics

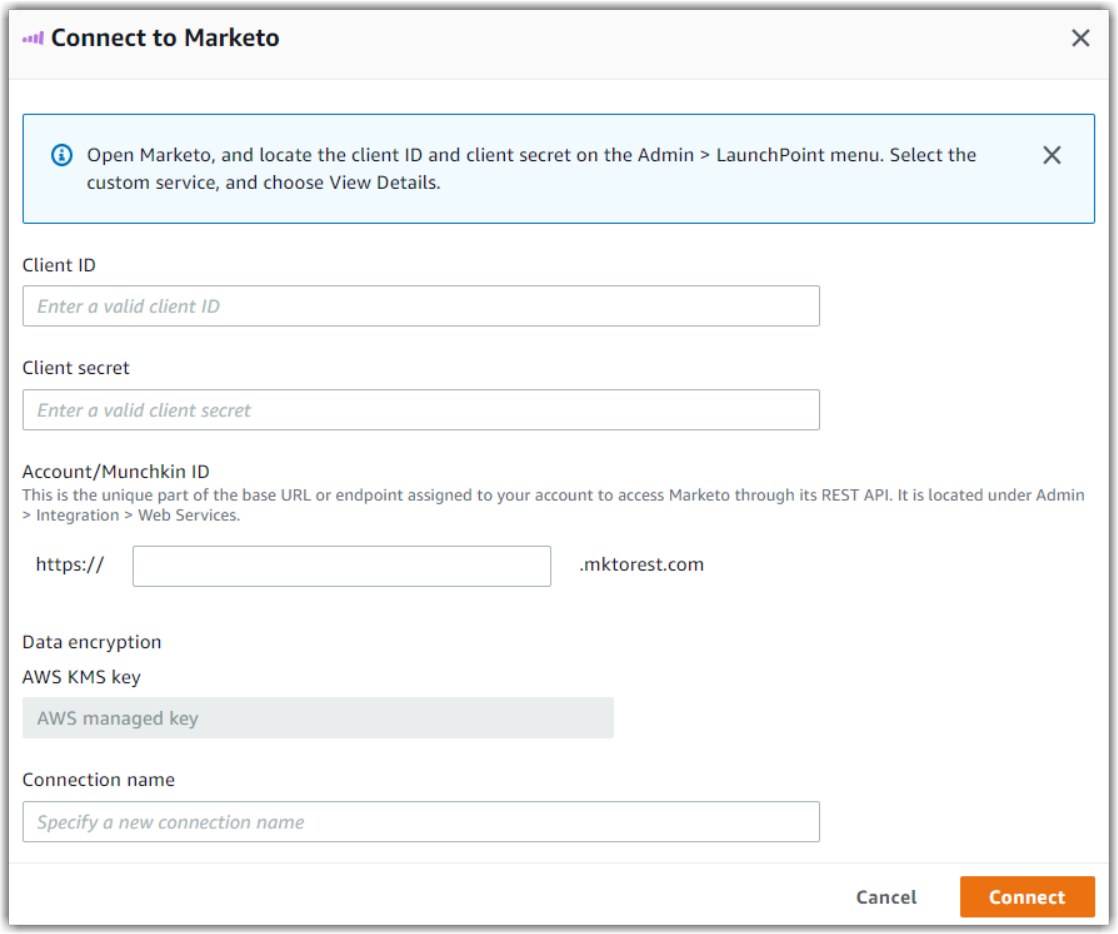
Requirements

You must provide Amazon AppFlow with your client ID and client secret. For more information about how to retrieve your client ID and client secret, see Credentials for API Access in the Marketo documentation.

Connection instructions

To connect to Marketo while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings**. Then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag**, and then enter the key name and value.
6. Choose **Next**.
7. Choose **Marketo** from the **Source name** or **Destination name** dropdown list.
8. Choose **Connect** to open the **Connect to Marketo** dialog box.
    a. Under **Client ID**, enter your Marketo client ID.
    b. Under **Client secret**, enter your client secret.
    c. Under **Account/Munchkin ID**, specify the unique part of the base URL or endpoint assigned to your Marketo account.
    d. Under **Data encryption**, enter your AWS KMS key.
    e. Under **Connection name**, specify a name for your connection.
    f. Choose **Connect**.



9. You will be redirected to the Marketo login page. When prompted, grant Amazon AppFlow permissions to access your Marketo account.

Now that you are connected to your Marketo account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in Requirements (p. 31).

Notes

- When you use Marketo as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per hour.
- Depending on your instance, Marketo might queue requests for data extraction. This can result in longer flow run times. If you want to avoid queueing, contact your Marketo administrator for assistance. We recommend that you avoid running concurrent flows using Marketo if your use case does not benefit from it.
- Depending on your Marketo instance, you can submit more than one bulk import request (with limitations). Each request is added as a job to be processed in a First-In-First-Out (FIFO) queue. A maximum of two jobs are processed at the same time. A maximum of ten jobs are allowed in the queue at any given time, including the two currently being processed. If you exceed the ten job

maximum, a 1016: Too many imports error is returned. If you want to avoid queueing, contact your Marketo administrator for assistance.

- There is a soft quota of 1 GB per flow when extracting data from Marketo. If you need to process more records in a single flow, you can submit a request to Amazon AppFlow through the Amazon AppFlow support channel. For more information, see Creating a support case in the *AWS Support User Guide*.

Related resources

- Credentials for API Access in the Marketo documentation
- API Limits with Marketo in the Marketo documentation
- Error Codes with Marketo in the Marketo documentation
- Video: Introduction to the Marketo Connector in Amazon AppFlow

Salesforce

The following are the requirements and connection instructions for using Salesforce with Amazon AppFlow.

Note
You can use Salesforce as a source or destination.

Topics

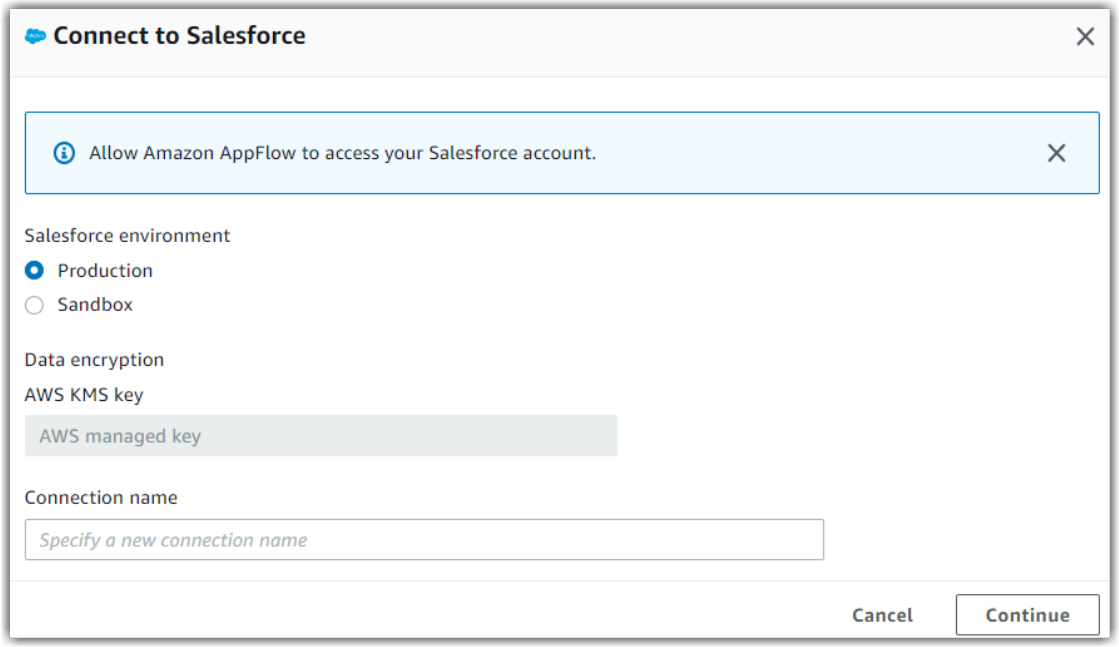
Requirements

- Your Salesforce account must be enabled for API access. API access is enabled by default for the Enterprise, Unlimited, Developer, and Performance editions.
- Your Salesforce account must allow you to install connected apps. If this functionality is disabled, contact your Salesforce administrator. After you create a Salesforce connection in Amazon AppFlow, verify that the connected app named **Amazon AppFlow Embedded Login App** is installed in your Salesforce account.
- The refresh token policy for the **Amazon AppFlow Embedded Login App** must be set to **Refresh token is valid until revoked**. Otherwise, your flows will fail when your refresh token expires. For more information on how to check and edit the refresh token policy, see Manage OAuth Access Policies for a Connected App in the Salesforce documentation.
- You must enable change data capture in Salesforce to use event-driven flow triggers. For more information on how to enable this, see Select Objects for Change Notifications in the User Interface in the Salesforce documentation.
- If your Salesforce app enforces IP address restrictions, you must grant access to the addresses used by Amazon AppFlow. For more information, see AWS IP address ranges in the *Amazon Web Services General Reference*.
- To create private connections using AWS PrivateLink, you must enable both `Manager Metadata` and `Manage External Connections` user permissions in your Salesforce account. Private connections are currently available in the us-east-1 and us-west-2 AWS Regions.

Connection instructions

To connect to Salesforce while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Salesforce** from the **Source name** or **Destination name** dropdown list.
8. Choose **Connect** or **Connect with PrivateLink** to open the **Connect to Salesforce** dialog box.
    a. Under **Salesforce environment**, choose **Production** to log into your developer account.
    b. Under **Data encryption**, enter your AWS KMS key.
    c. Under **Connection name**, specify a name for your connection.
    d. Choose **Continue**.



9. You will be redirected to the Salesforce login page. When prompted, grant Amazon AppFlow permissions to access your Salesforce account.
10. After you log in, you will see the objects that you enabled in your Salesforce account in the **Choose Salesforce object** dropdown list.

Now that you are connected to your Salesforce account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 33) section above.

Use a global connected app with Amazon AppFlow

- You can use your own global connected app for Salesforce with Amazon AppFlow APIs. For instructions on how to create a connected app in Salesforce, see Create a global connected app in Salesforce (p. 37).
- To use your own global connected app, you need to pass on the clientId, clientSecret, and secrets manager ARN to Amazon AppFlow.
  - The following example shows a sample secrets manager entry with application credentials for Salesforce:

```
{
    "clientCredsARN": "arn:aws:secretsmanager:region:SecretID:secret:Secret_Key",
    "Name": "Salesforce",
    "VersionId": "db83aeb0-e995-480a-81f3-8805b0bf2b79",
    "SecretString": "{\"clientId\":\"sampleClientId\",\"clientSecret\":
\"sampleClientSecret\"}"
}
```

  - This example shows how you can call the `ConnectorProfile` API with an access token, refresh token, and credentials ARN:

```
{
    "connectorProfileName": "testSalesforceProfileNew",
    "kmsArn": null,
    "connectorType": "Salesforce",
    "connectionMode": "Public",
    "connectorProfileConfig": {
        "connectorProfileProperties": {
            "salesforce": {
            "instanceUrl": "InstanceURL",
            "isSandboxEnvironment": false
            }
        }
     }
},
"connectorProfileCredentials": {
    "salesforce": {
    "clientCredsARN": "arn:aws:secretsmanager:region:SecretID:secret:Secret_Key",**
    "accessToken": "testAccessToken",
    "refreshToken": "testRefreshToken",
    "oauthRequest": {
        "authCode": null,
        "redirectUri": null
                    }
                   }
   }
```

- You must attach a resource policy to the secrets manager and the KMS key which is used encrypt the secret. This resource policy allows Amazon AppFlow to read the secret and use it.
  - The following is the policy to be attached for the KMS key. Replace the *placeholder* with your own information.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "appflow.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "<KMS key ARN>"
    }
  ]
}
```

Additionally, Amazon AppFlow supports adding confused deputy protection to this KMS key policy. To learn about the confused deputy problem and mitigations, refer to our Amazon S3 documentation. The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in your AWS KMS key to prevent the confused deputy problem. Replace *Account ID* with your AWS account ID and *Resource ARNs* with a list of ARNs for any connector profiles created with the client credentials secret. Additionally you may use wildcards in the aws:SourceAccount key (*). For example, you can replace *Resource ARNs* with `arn:aws:appflow:`*region*`:`*accountId*`:*` to give access to all created resources created on your behalf.

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal":    {
                "Service": "appflow.amazonaws.com"
                },
            "Action": [
            "kms:Encrypt",
            "kms:GenerateDataKey",
            "kms:Decrypt"
            ],
            "Resource": "<KMS key ARN>",
            "Condition": {
                "StringEquals": {
                       "aws:SourceAccount":"<Account ID>"
                },
                "ArnLike": {
                "aws:SourceArn":"<Resource ARNs>"
                }
            }
        }
    ]
}
```

- The following is the policy to be attached for the secret. Replace the *placeholder* with your own information.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "appflow.amazonaws.com"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "<Secret ARN>"
    }
  ]
}
```

Create a global connected app in Salesforce

Follow these instructions to create a connected app in Salesforce if you haven't done so already.

To create a global connected app in Salesforce

1. Log in to Salesforce with an account that has administrator rights, and go to **Setup**.
2. In the navigation pane under **Platform Tools**, expand **Apps** and choose **App Manager**.
3. Choose **New Connected App** in the upper-right corner, and enter the following information for your connected app:
   - The name of your connected app, such as **`"Amazon AppFlow Embedded Login App"`**.
   - The API name for your connected app. This is auto-generated and can be edited, if needed.
   - The contact email address for Salesforce to use if they need to contact you about your connected app.
   - The logo image URL and icon, if you have one. This is optional.
   - A brief description to specify what the connected app is for, such as **`"Application which handles interaction between Salesforce and Amazon AppFlow console"`**.
4. Select the **Enable OAuth Settings** check box.
5. In the **Callback URL** text field, enter the URLs for your console for the stages and Regions in which you will use the connected app. Enter these URLs on separate lines.
6. Select the **Require Secret for Web Server Flow** check box.
7. In the **Available OAuth Scopes** list, select the following items and then choose **add** to move them to the **Selected OAuth Scopes** list. You can customize this list as needed.
   - Access and manage your data (api)
   - Access custom permissions (custom_permissions)
   - Access your basic information (id, profile, email, address, phone)
   - Allow access to your unique identifier (openid)
   - Perform requests on your behalf at any time (refresh_token, offline_access)
8. Choose **Save**.

To retrieve the client ID and client secret for use in your OAuth flow, you can view your connected app in Salesforce by choosing **Apps** and then **App Manager**, and then selecting the connected app that you created.

For more information on connected apps in Salesforce, see Connected Apps in the Salesforce documentation.

Notes

- If you are transferring more than 1 million Salesforce records, you cannot choose any Salesforce compound field. Amazon AppFlow uses Salesforce bulk APIs for the transfer, which does not allow the transfer of compound fields.

- Amazon AppFlow only supports the automatic import of newly created Salesforce fields into Amazon S3 without requiring the user to update their flow configurations.
- When you use Salesforce as a source, you can import 15 GB of data as part of a single flow run. To transfer over 15 GB of data, you can split your workload into multiple flows by applying the appropriate filters to each flow. Salesforce records are typically 2 KB in size, but can be up to 4 KB. Therefore, 15 GB would be approximately 7.5 million Salesforce records.
- When you use Salesforce as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per minute.
- Amazon AppFlow added support for Salesforce API version 50.0 on January 19th, 2021. Flows associated with all Salesforce connections created after this date will use Salesforce API version 50.0, while flows for all previously created connections will use Salesforce API version 47.0.
- Amazon AppFlow supports Change Data Capture Events and Platform events from Salesforce.
- When you use Salesforce as a destination, the following additional settings are available:

Setting name	Description
Insert new records	• This is the default data transfer option. • When you choose this setting, Amazon AppFlow inserts your source data into the chosen Salesforce object as a new record.
Update existing records	• When you choose this setting, Amazon AppFlow uses your source data to update existing records in Salesforce. For every source record, Amazon AppFlow looks for a matching record in Salesforce based on your criteria. You can specify matching criteria on the Map data fields page. To do so, select a field in the source application and map it to a Salesforce record ID field using the dropdown list. • When a matching record is found, Amazon AppFlow updates the record in Salesforce. If no matching record is found, Amazon AppFlow ignores the record or fails the flow per your chosen error handling option. You can specify your error handling preferences on the Configure flow page. • Please note that you must use the upsert operation in order to update existing records using an external id field. The standard update operation does not support use of an external id field.
Upsert records	• When you choose this setting, Amazon AppFlow performs an upsert operation in Salesforce. For every source record, Amazon AppFlow looks for a matching record in Salesforce based on your criteria. You can specify matching criteria on the Map data fields page. To do so, select a field in the source application and map it to a Salesforce external field using the dropdown list. • When a matching record is found, Amazon AppFlow updates the record in Salesforce. If no matching record is found, Amazon AppFlow inserts the data as a new record. Any errors in performing the operation are handled per your chosen error handling option. You can specify your error handling preferences on the Configure flow page.

Related resources

- Amazon AppFlow now supports new Salesforce integrations in the AWS *What's new* blog
- Amazon AppFlow now supports private data transfers between AWS and Salesforce in the AWS *What's new* blog

- Building Salesforce integrations with EventBridge and Amazon AppFlow in the *AWS Compute* blog
- Building Secure and Private Data Flows Between AWS and Salesforce Using Amazon AppFlow in the *AWS Partner Network (APN)* blog
- Using Amazon AppFlow to Achieve Bi-Directional Sync Between Salesforce and Amazon RDS for PostgreSQL in the *AWS Partner Network (APN)* blog
- Salesforce Private Connect Demo in the Salesforce documentation
- Manage OAuth Access Policies for a Connected App in the Salesforce documentation
- Select Objects for Change Notifications in the User Interface in the Salesforce documentation
- AWS IP address ranges in the *Amazon Web Services General Reference*
- Video: How to insert new Salesforce records with data in Amazon S3 using Amazon AppFlow

Salesforce Pardot

The following are the requirements and connection instructions for using Pardot with Amazon AppFlow.

Note
You can use Pardot as a source only.

Topics

Requirements

- Your Salesforce account must be enabled for API access. API access is enabled by default for Enterprise, Unlimited, Developer, and Performance editions.
- Your Salesforce account must allow you to install connected apps. If this option is disabled, contact your Salesforce administrator.
- After you create a Pardot connection in Amazon AppFlow, verify that the connected app named *Amazon AppFlow Pardot Embedded Login App* is installed in your Salesforce account. For instructions on how to create a connected app in Salesforce, see Create a global connected app in Salesforce (p. 37). For more information about connected apps in Salesforce, see Connected Apps in the Salesforce documentation.
- The refresh token policy for the **Amazon AppFlow Pardot Embedded Login App** must be set to **Refresh token is valid until revoked**. Otherwise, your flows will fail when your refresh token expires.
- If your Pardot app enforces IP address restrictions, you must grant access to the addresses used by Amazon AppFlow. For more information, see AWS IP address ranges in the *Amazon Web Services General Reference*.

Pardot version support

Amazon AppFlow supports Pardot version 4 only. If you are still using version 3, you must upgrade to version 4 to use Amazon AppFlow. For more information, see Transitioning from version 3 to version 4 in the Pardot documentation.

Authentication and Pardot business ID

- Amazon AppFlow supports authentication via OAuth2 with Pardot. For more information, see Authentication Via Salesforce OAuth in the Pardot documentation.

- You must have the Pardot Business Unit ID that you are trying to authenticate with. To find the Pardot Business Unit ID in Salesforce, go to **Setup** and enter **`Pardot Account Setup`** in the **Quick Find** box. Your Pardot Business Unit ID begins with *0Uv* and is 18 characters long. If you cannot access the Pardot account setup information, ask your Salesforce administrator to provide you with the Pardot Business Unit ID.

Setup instructions

To connect to Pardot while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Pardot** from the **Source name** dropdown list.
8. Choose **Connect** to open the **Connect to Pardot** dialog box. If you are connecting to Pardot for the first time, follow the instructions to complete the OAuth workflow and create a connection profile.
9. You will be redirected to the Pardot login page. When prompted, grant Amazon AppFlow permissions to access your Pardot account.

Now that you are connected to your Pardot account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

> **Tip**
> If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 39) section.

Notes

- When you use Pardot as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per minute.
- You can connect Amazon AppFlow to your Pardot sandbox account in addition to your Pardot production account.
- Amazon AppFlow inherits quotas from Pardot. Quotas are enforced on daily requests and concurrent requests at the customer level. *Pardot Pro* customers are allocated 25,000 API requests a day. *Pardot Ultimate* customers can make up to 100,000 API requests a day. These limits reset at the beginning of the day based on your account time zone settings. Any request that exceeds these quotas results in an error code 122. Amazon AppFlow handles these error codes transparently.

Related resources

- Transitioning from version 3 to version 4 in the Pardot documentation
- Connected Apps in the Salesforce documentation
- Authentication Via Salesforce OAuth in the Pardot documentation

SAP OData

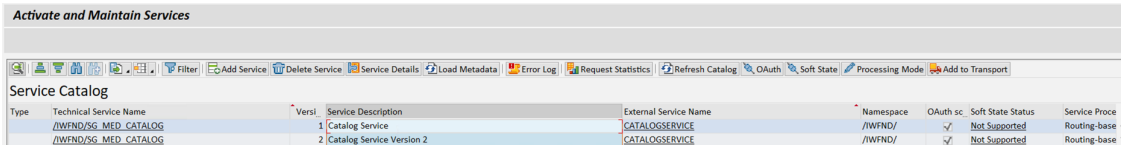
Amazon AppFlow SAP OData connector provides data pull support for the OData APIs exposed by SAP S/4HANA and SAP on premise. The following are the requirements and connection instructions for using SAP OData with Amazon AppFlow.

Note
You can use SAP OData as a source only.

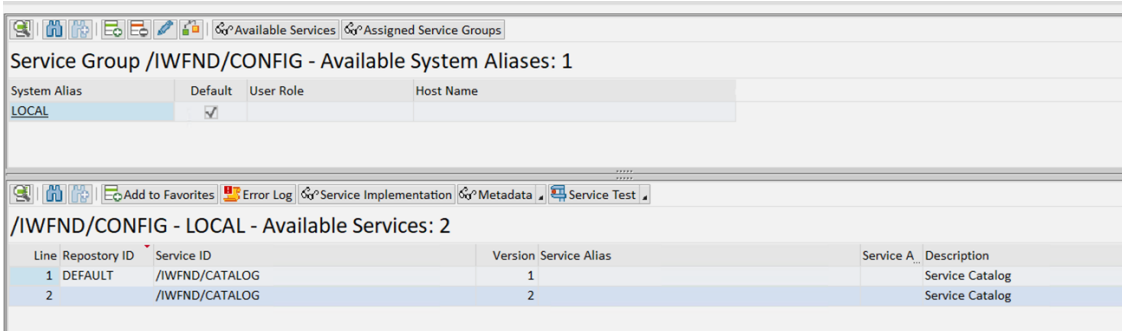
Topics

Requirements

- Your SAP NetWeaver stack version must be 7.40 SP02 or above.
- You must enable catalog service for service discovery.
  - **OData V2.0:** The OData V2.0 catalog service(s) can be enabled in your SAP Gateway via transaction **/IWFND/MAINT_SERVICE** .



  - **OData V4.0:** The OData V4.0 catalog services can be enabled in your SAP Gateway environment by publishing the service groups **/IWFND/CONFIG** or as described in the SAP documentation relevant to your gateway version.



- You must enable OData V2.0/V4.0 services in your SAP Gateway. The OData V2.0 services can be enabled via transaction **/IWFND/MAINT_SERVICE** and V4.0 services can be published via transaction **/IWFND/V4_ADMIN**.
- Your SAP OData service must support client side pagination/query options such as **$top** and **$skip**. It must also support system query option **$count**.
- Appflow supports following authentication mechanisms:
  - **Basic** - Supported for OData V2.0 and OData V4.0
  - **OAuth 2.0** - Supported for only OData V2.0. You must enable OAuth 2.0 for the OData service and register the OAuth client per SAP documentation and set the authorized redirect URL as follows:
    - https://console.aws.amazon.com/appflow/oauth for the us-east-1 Region
    - https://region.console.aws.amazon.com/appflow/oauth for all other Regions

- You must enable secure setup for connecting over HTTPS.
- You must provide required authorization for the user in SAP to discover the services and extract data using SAP OData services. Please refer to the security documentation provided by SAP.

Private Connection Requirements

- You need to create VPC Endpoint Service for your SAP OData instance running in a VPC. This VPC endpoint service must have Amazon AppFlow service principal **appflow.amazonaws.com** as allowed principal and must be available in **at least more than 50% AZs in a region**.
- When creating connection using OAuth, your **Authorization Code URL** must be reachable by the network from where the connection is being setup. This is because OAuth connection involves browser interaction with SAP Login Page which cannot happen over AWS PrivateLink. The network from where the connection is being setup must be connected to SAP OData instance running in a VPC so that hostname of authorization code url can be resolved. Alternately, you can choose to make your Authorization Code URL available over public internet so that console user interaction can happen from any network.
- For OAuth, in addition to **Application Host URL**, your **Authorization Tokens URL** must also be available behind VPC Endpoint Service to fetch Access/Refresh tokens over private network.
- For OAuth, you must set your OAuthCode expiry to at least 5 minutes.

Setup instructions

To connect to SAP OData while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **SAP OData** from the **Source name** dropdown list.
8. Choose **Connect** or **Connect with PrivateLink** to open the **Connect to SAP OData** dialog box.

    a. Under **Application Host URL**, enter your Application host url. This application host url must be accessible over public internet for non PrivateLink connection.

    b. Under **Application Service Path**, enter your catalog service path. e.g. **/sap/opu/odata/iwfnd/catalogservice;v=2**. Appflow doesn't accept specific object path.

    c. Under **Port Number**, enter your port number.

    d. Under **Client Number**, enter your 3 digit client number. Acceptable values are [001-999]. e.g. **010**

    e. Under **Logon Language**, enter your two character logon language. e.g. **EN**.

    f. (Optional) To use private connection for data transfer, under **AWS PrivateLink service name**, enter your VPC Endpoint (PrivateLink) service name. e.g. **com.amazonaws.vpce.us-east-1.vpce-svc-xxxxxxxxxxxxx**

    g. Select your preferred Authentication Mode.

    - If Basic,

        i. Under **User name**, enter your useraname.

        ii. Under **Password**, enter your password.

- If OAuth2,

  i. Under **Authorization Code URL**, enter your authorization code URL.

  ii. Under **Authorization Tokens URL**, enter your authorization token URL.

  iii. Under **OAuth Scopes**, enter your OAuth scopes separated by space. e.g. **/IWFND/SG_MED_CATALOG_0002 ZAPI_SALES_ORDER_SRV_0001**

  iv. Under **Client ID**, enter your client id .

  v. Under **Client Secret**, enter your client secret .

h. Under **Connection name**, specify a name for your connection.

i. Choose **Connect**.

j. If using OAuth, you will be redirected to the SAP login page. When prompted, grant Amazon AppFlow permissions to access your SAP account.

k. If connection type is **Private**, then

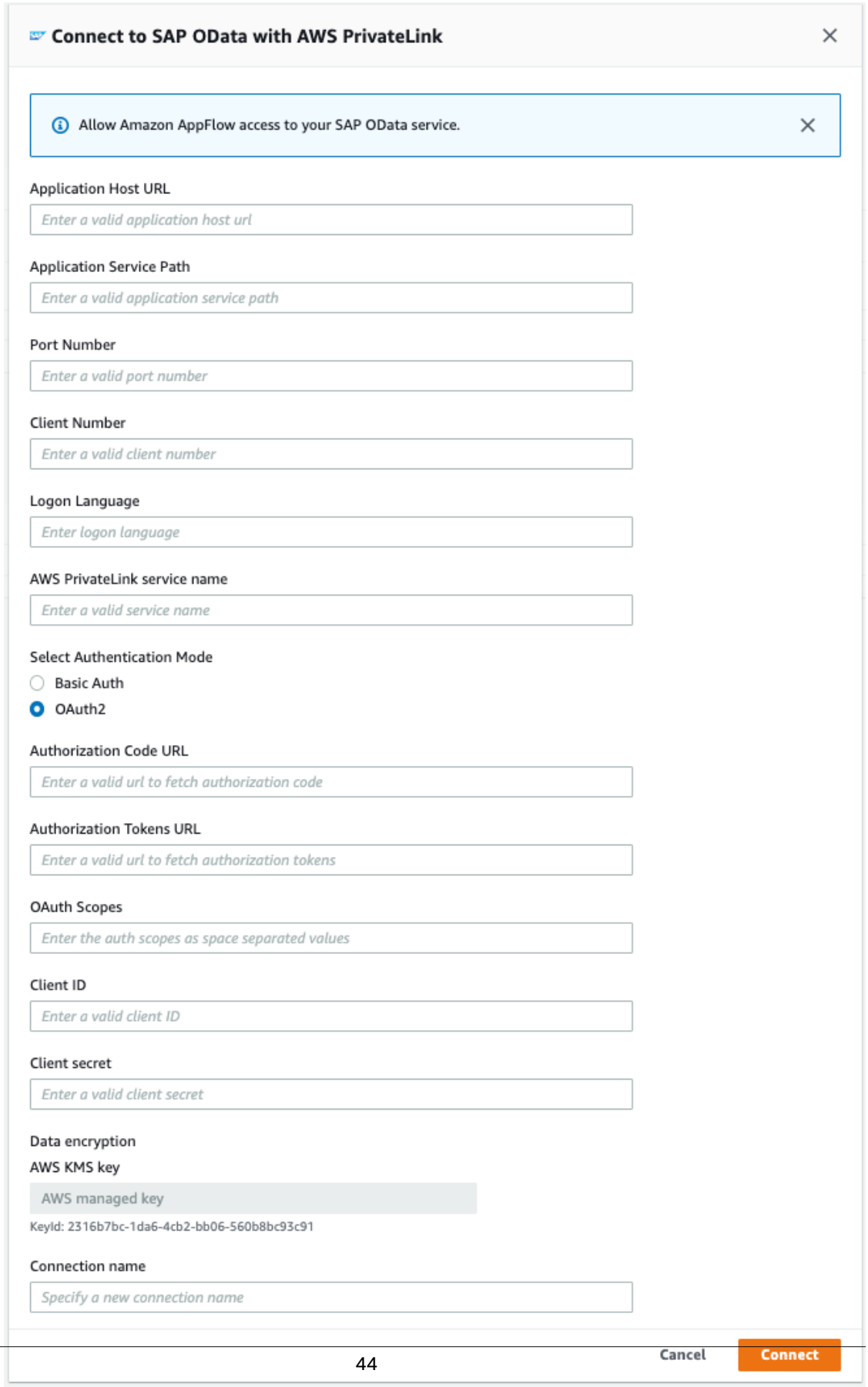
  i. AppFlow creates AWS PrivateLink Endpoint (if not already present) connection to your VPC Endpoint Service before any metadata/data transfer calls can be made to your SAP OData instance over private network. AWS PrivateLink Endpoint creation can take 3-5 minutes, and until its created, profile status would be PENDING.

  Important
  While the connection status is PENDING, you will not be prompted to list and choose SAP OData object.

  ii. (Optional) If your VPC Endpoint Service has **Acceptance Required** setting set to true. You will need to accept the connection in the AWS account which has VPC Endpoint service for AWS PrivateLink endpoint provisioning to start.

  iii. Once the AWS PrivateLink Endpoint connection is established, AppFlow fetches (only for OAuth) access/refresh tokens using the authCode, make a test connection call over private network and finally change connection status from PENDING to CREATED. At this point, you will be prompted to list and choose SAP OData object.

  iv. If for any reason private connection creation fails, connection status would change to FAILED.


Connect to SAP OData with AWS PrivateLink
Allow Amazon AppFlow access to your SAP OData service.
Application Host URL
Enter a valid application host url
Application Service Path
Enter a valid application service path
Port Number
Enter a valid port number
Client Number
Enter a valid client number
Logon Language
Enter logon language
AWS PrivateLink service name
Enter a valid service name
Select Authentication Mode
Basic Auth
OAuth2
Authorization Code URL
Enter a valid url to fetch authorization code
Authorization Tokens URL
Enter a valid url to fetch authorization tokens
OAuth Scopes
Enter the auth scopes as space separated values
Client ID
Enter a valid client ID
Client secret
Enter a valid client secret
Data encryption
AWS KMS key
AWS managed key
KeyId: 2316b7bc-1da6-4cb2-bb06-560b8bc93c91
Connection name
Specify a new connection name
Cancel
Connect

Now that you are connected to your SAP OData account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements section above.

Notes

- Appflow does not support SAP ODP based OData extraction or header based pagination e.g. (prefer=odata.maxpagesize).
- Appflow does not support intrinsic delta tokens provided by SAP for incremental data pulls.
- When you use SAP OData as a source, you can run schedule-triggered flows at a maximum frequency of one flow runs per minute.
- If you have a private ConnectorProfile for a VPC endpoint service, and you try to create another private ConnectorProfile for the same VPC endpoint service, AppFlow will re-use the already created private connection, and thus you would not need to wait for private connection provisioning to complete to list and choose SAP OData object.
- AppFlow allows at max 1000 flow executions at a time per AWS account. If you choose to run multiple flows against the same SAP OData instance, you need to accordingly scale your instance.

Related resources

- Setting up SAP Gateway in *SAP* documentation.

ServiceNow

The following are the requirements and connection instructions for using ServiceNow with Amazon AppFlow.

Note
You can use ServiceNow as a source only.

Topics

Requirements

- You must provide Amazon AppFlow with your ServiceNow user name, password, and instance name.
- Verify that you have admin roles. For more information, see Roles in the ServiceNow documentation.

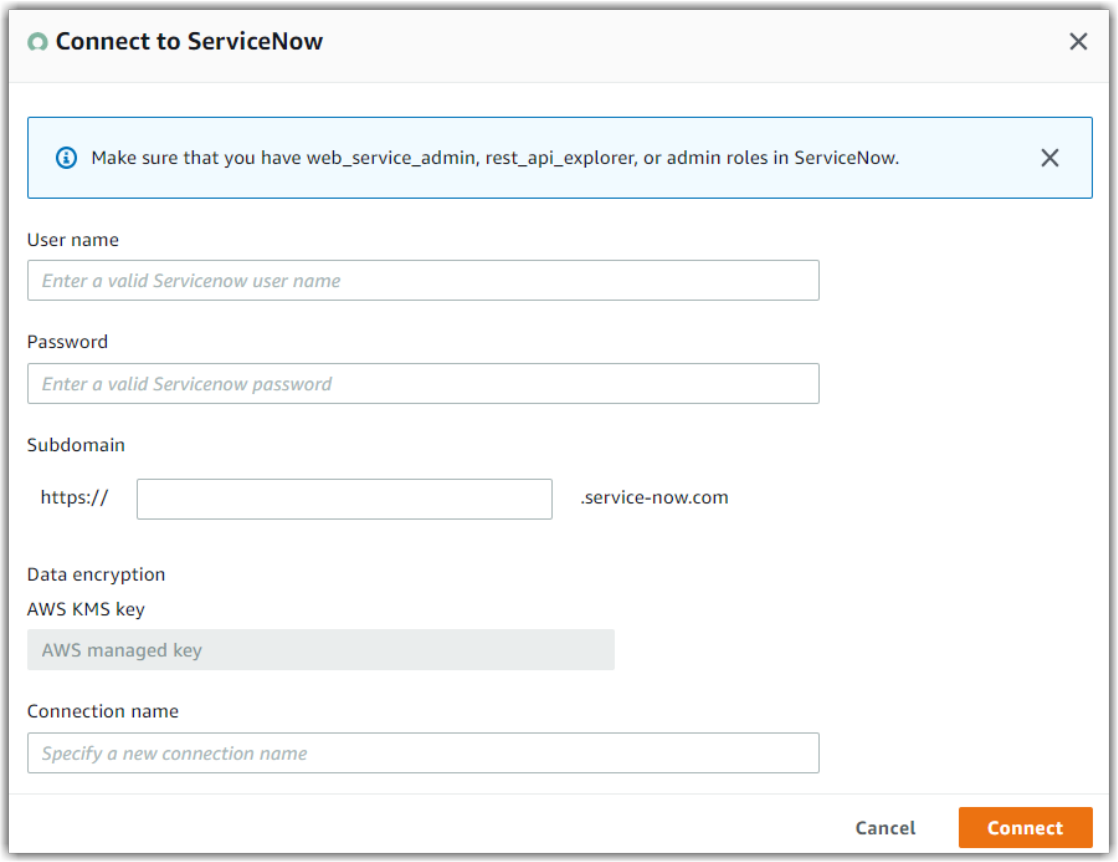
Connection instructions

To connect to ServiceNow while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.

2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **ServiceNow** from the **Source name** dropdown list.
8. Choose **Connect** to open the **Connect to ServiceNow** dialog box.

    a. Under **User name**, enter your ServiceNow user name.
    b. Under **Password**, enter the password for that account.
    c. Under **Subdomain**, specify the instance of ServiceNow you want to connect to.
    d. Under **Data encryption**, enter your AWS KMS key.
    e. Under **Connection name**, specify a name for your connection.
    f. Choose **Connect**.



9. Once connected, you can choose the ServiceNow object.

Now that you are connected to your ServiceNow account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 45) section.

Notes

- Once you are connected to your ServiceNow instance, you can select the relevant objects from ServiceNow by using the dropdown list. Given the amount of data being available via ServiceNow, the dropdown list may take some time to fully populate. Amazon AppFlow will list all tables available (including custom ones) and you can map the source fields to the destination fields during flow setup.
- You can run your flows either on demand, or on schedule, which enables you to integrate your ServiceNow data with AWS services.
- When you use ServiceNow as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per minute.
- ServiceNow can process up to 100,000 records as part of a single flow run.

Related resources

- Roles in the *ServiceNow* documentation

Singular

The following are the requirements and connection instructions for using Singular with Amazon AppFlow.

> **Note**
> You can use Singular as a source only.

Topics

Requirements

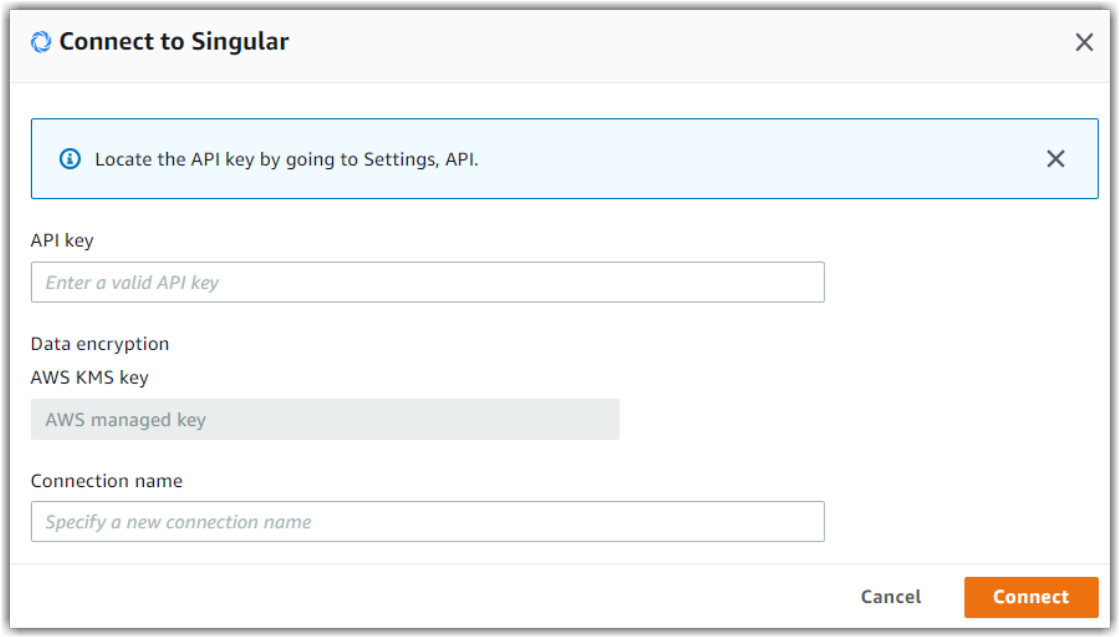
- You must provide Amazon AppFlow with an API key. For more information about retrieving your client ID and client secret, see Authentication in the Singular documentation.
- The date range for the flow cannot exceed 30 days.
- The flow cannot return more than 100,000 records.

Connection instructions

To connect to Singular while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.

5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Singular** from the **Source name** dropdown list.
8. Choose **Connect** to open the **Connect to Singular** dialog box.
   a. Under **API key**, enter your API key.
   b. Under **Data encryption**, enter your AWS KMS key.
   c. Under **Connection name**, specify a name for your connection.
   d. Choose **Connect**.



9. You will be redirected to the Singular login page. When prompted, grant Amazon AppFlow permissions to access your Singular account.

Now that you are connected to your Singular account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 47) section.

Notes

- When you use Singular as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per hour.

Related resources

- Authentication in the Singular documentation
- Load all your paid marketing with Amazon AppFlow. No code required. from Singular

Slack

The following are the requirements and connection instructions for using Slack with Amazon AppFlow.

Note
You can use Slack as a source only.

Topics

Requirements

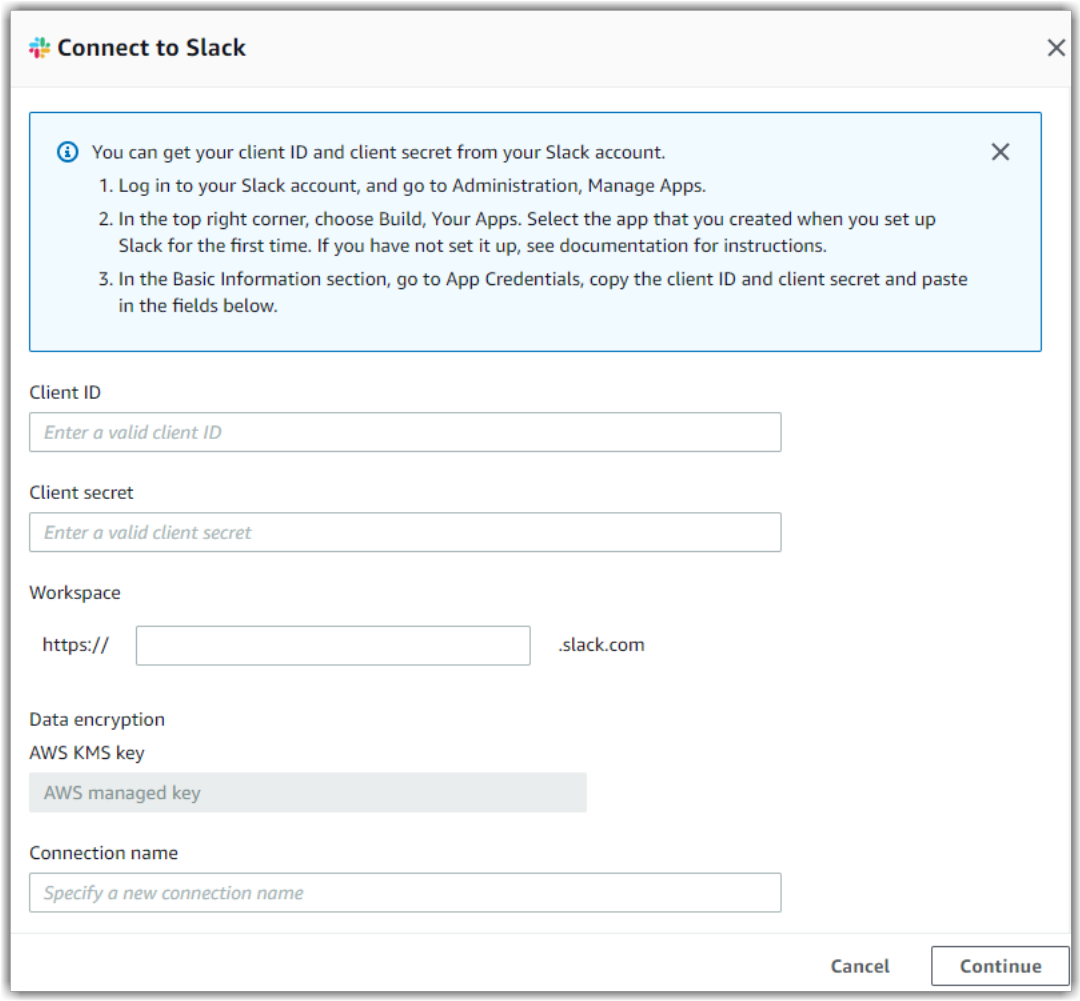
- To create a Slack connection in Amazon AppFlow, you must note your client ID, client secret, and Slack instance name. To retrieve your client ID and secret from Slack, you first must create a Slack App if you haven't already. For more information about how to create an App and then retrieve your client ID and secret, see the Slack documentation.
- Set the redirect URL as follows:
  - https://console.aws.amazon.com/appflow/oauth for the us-east-1 Region
  - https://*region*.console.aws.amazon.com/appflow/oauth for all other Regions
- Set the following user token scopes:
  - `channels:history`
  - `channels:read`
  - `groups:history`
  - `groups:read`
  - `im:history`
  - `im:read`
  - `mpim:history`
  - `mpim:read`

Connection instructions

To connect to Slack while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Slack** from the **Source name** dropdown list.
8. Choose **Connect** to open the **Connect to Slack** dialog box.
   a. Under **Client ID**, enter your Slack client ID.

b. Under **Client secret**, enter your Slack client secret.
c. Under **Workspace**, enter the name of your Slack instance.
d. Under **Data encryption**, enter your AWS KMS key.
e. Under **Connection name**, specify a name for your connection.
f. Choose **Continue**.



9. You will be redirected to the Slack login page. When prompted, grant Amazon AppFlow permissions to access your Slack account.

Now that you are connected to your Slack account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 49) section.

Notes

- When you use Slack as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per minute.

Related resources

- Retrieve your client ID and secret in the Slack documentation
- New – Announcing Amazon AppFlow (dataflow: Slack, S3, Athena, QuickSight) in the *AWS News* blog
- Video: How to transfer data from Slack to Amazon S3 using Amazon AppFlow

Snowflake

The following are the requirements and connection instructions for using Snowflake with Amazon AppFlow.

Note
You can use Snowflake as a destination only.

Topics

Requirements

- Amazon AppFlow uses the Snowflake COPY command to move data using an S3 bucket. To configure the integration, see Configuring Secure Access to Amazon S3 in the Snowflake documentation.
- You must also add access to the `kms:Decrypt` action so that Snowflake can access the encrypted data that Amazon AppFlow stored in the Amazon S3 bucket.

```
{
    "Effect": "Allow",
    "Action": "kms:Decrypt",
    "Resource": "*"
}
```

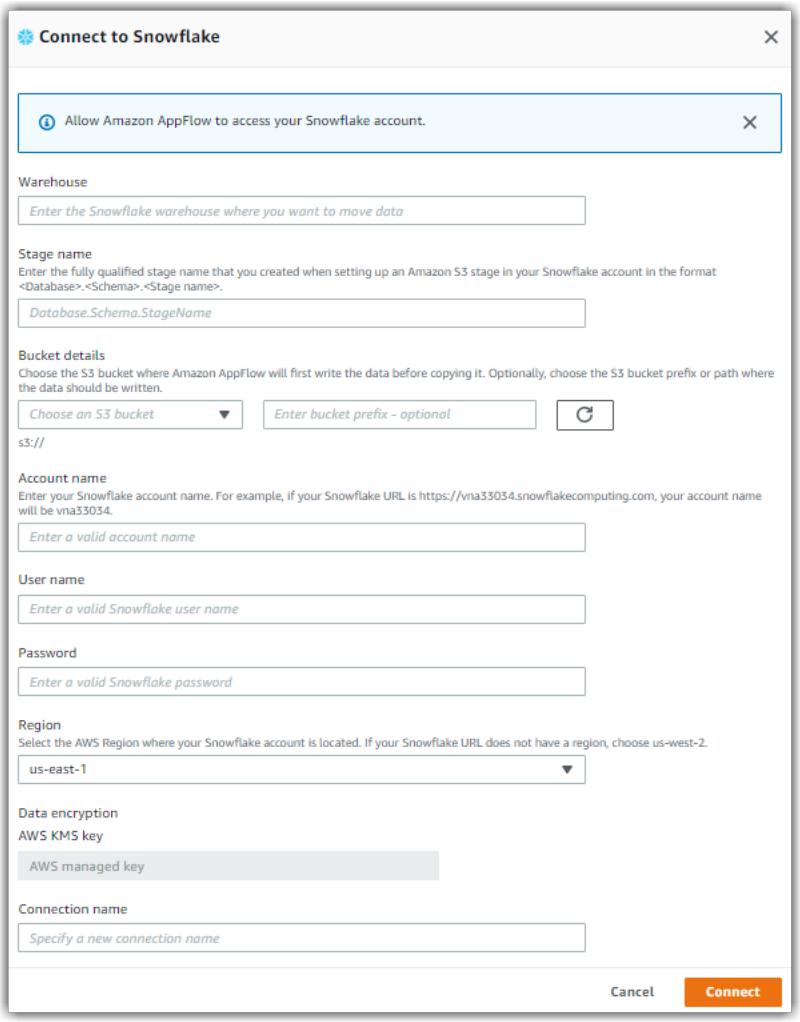
- You must provide Amazon AppFlow with the following information:
  - the name of the stage and the S3 bucket for the stage
  - the user name and password for your Snowflake account
  - the S3 bucket prefix
  - the warehouse that you want to move data to

Connection instructions

To connect to Snowflake while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.

7. Choose **Snowflake** from the **Destination name** dropdown list.
8. Choose **Connect** or **Connect with PrivateLink** to open the **Connect to Snowflake** dialog box.
   a. Under **Warehouse**, enter the Snowflake warehouse that you want to move data to.
   b. Under **Stage name**, enter the Amazon S3 stage name in the following format: <Database> <Schema> <Stage name>
   c. Under **Bucket details**, select the S3 bucket where Amazon AppFlow will write data prior to copying it.
   d. Under **Account name**, enter your Snowflake account name. You can find your account name in the URL of your Snowflake instance. For example, if your Snowflake URL is https://vna33034.snowflakecomputing.com, your account name is vna33034.
   e. Under **User name**, enter the user name you use to log into Snowflake.
   f. Under **Data encryption**, enter your AWS KMS key.
   g. Under **Connection name**, specify a name for your connection.
   h. Choose **Connect**.



Now that you are connected to your Snowflake account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 51) section.

Related resources

- Configuring Secure Access to Amazon S3 in the Snowflake documentation

Trend Micro

The following are the requirements and connection instructions for using Tend Micro with Amazon AppFlow.

Note
You can use TrendMicro as a source only.

Topics

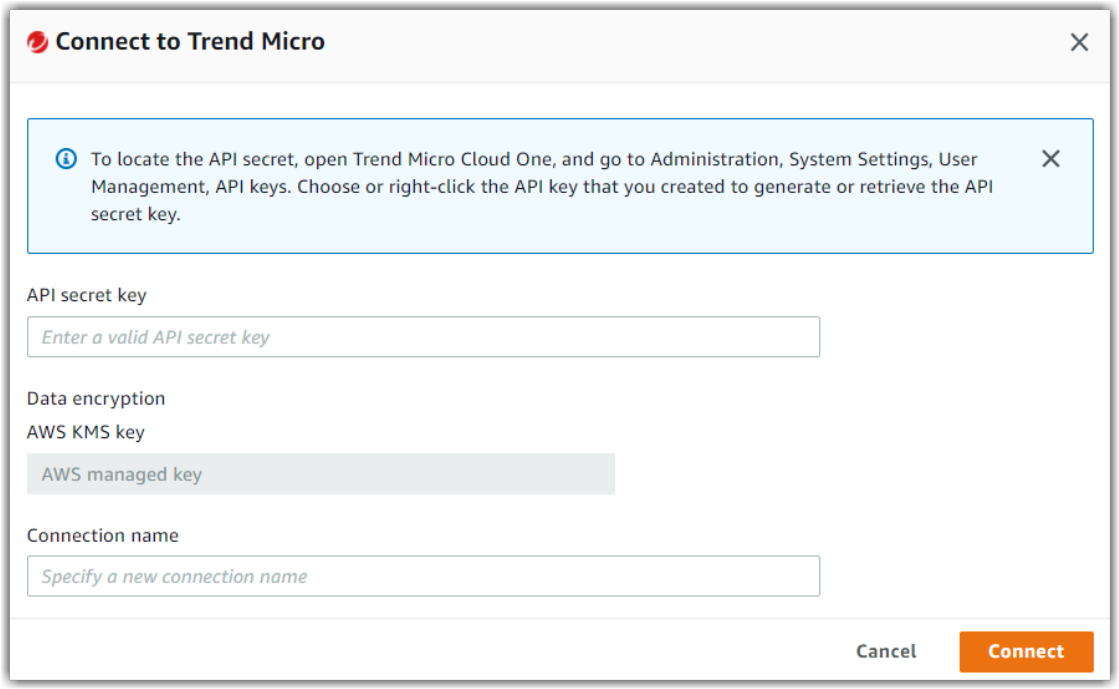
Requirements

You must provide Amazon AppFlow with an API secret. For more information about how to generate or retrieve an API secret from Trend Micro, see Create and Manage API Keys in the *Trend Micro* documentation.

Connection instructions

To connect to Trend Micro while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Trend Micro** from the **Source name** dropdown list.
8. Choose **Connect** or **Connect with PrivateLink** to open the **Connect to Trend Micro** dialog box.
    a. Under **API secret key**, enter your API secret key.
    b. Under **Data encryption**, enter your AWS KMS key.
    c. Under **Connection name**, specify a name for your connection.
    d. Choose **Connect**.



Now that you are connected to your Trend Micro account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 53) section.

Notes

- When you use Trend Micro as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per hour.

Related resources

- Trend Micro Integrates with Amazon AppFlow from Trend Micro
- Create and Manage API Keys in the Trend Micro documentation

Upsolver

The following are the requirements and connection instructions for using Upsolver with Amazon AppFlow.

Note
You can use Upsolver as a destination only.

Topics

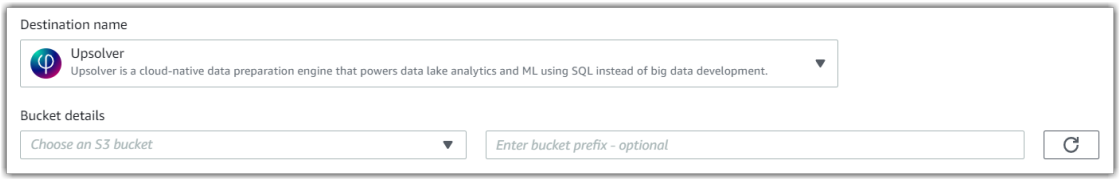
Requirements

- You must create an Amazon AppFlow data source in the Upsolver user interface. This will create an S3 bucket in your AWS account where Amazon AppFlow will send data.
- Alternatively, you can create an Amazon S3 bucket through the Amazon S3 console. The bucket name must begin with `upsolver-appflow`.

Setup instructions

To connect to Upsolver while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Upsolver** from the **Destination name** dropdown list.
8. Under **Bucket details**, select the S3 bucket in which you will place your data. You can specify a prefix, which is equivalent to specifying a folder within the S3 bucket where your source files are located or records are to be written to the destination.



Now that you are connected to your Amazon S3 bucket, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Now that you are connected to your S3 bucket, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

> **Tip**
> If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 55).

Notes

- You can configure Amazon AppFlow flows with Upsolver as the destination, and send data from any supported source to the integrated Upsolver Amazon S3 bucket. The data is then available for downstream processing in Upsolver.

Related resources

- Amazon AppFlow data source from the Upsolver documentation

Veeva

The following are the requirements and connection instructions for using Veeva with Amazon AppFlow.

Note
You can use Veeva as a source only.

Topics

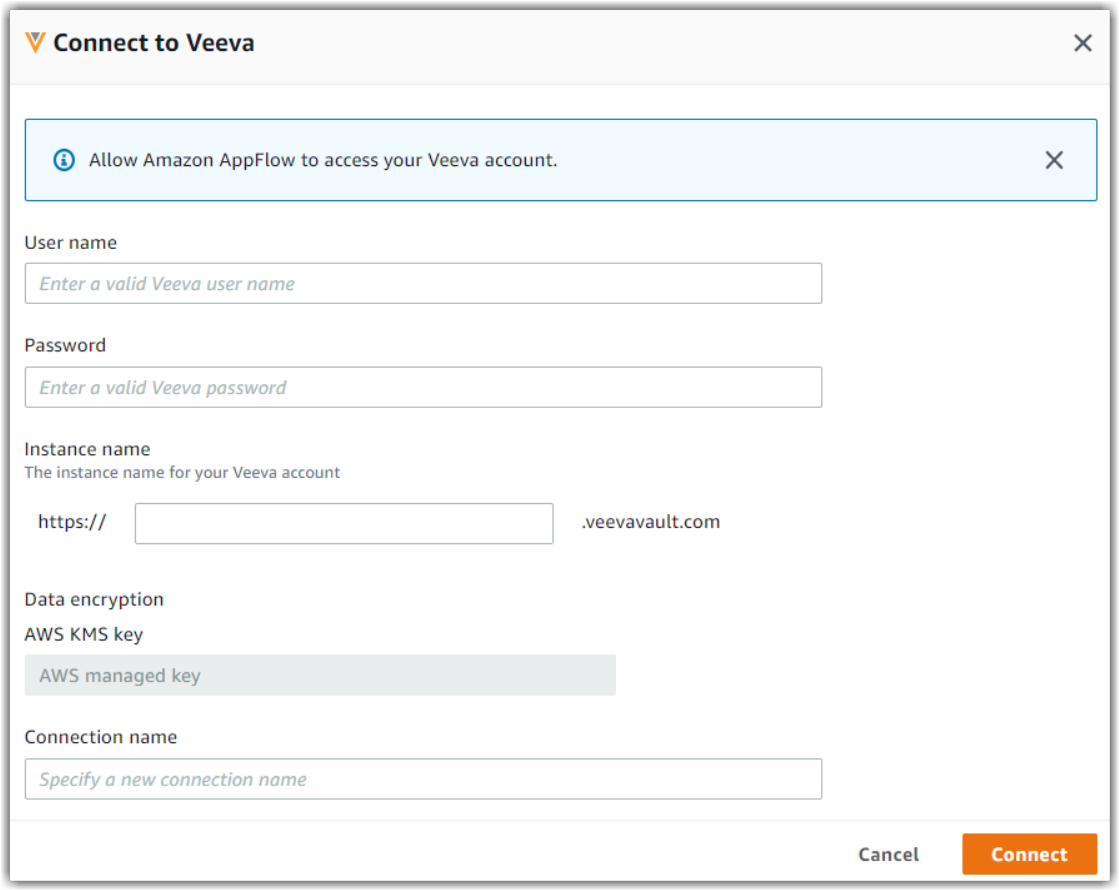
Requirements

- You must provide Amazon AppFlow with your user name, password, and Veeva instance name.
- Your user account must have API access. For more information, see API access permissions in the Veeva documentation.

Connection instructions

To connect to Veeva while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Veeva** from the **Source name** dropdown list.
8. Choose **Connect** to open the **Connect to Veeva** dialog box.
    a. Under **User name**, enter the user name you use to log into Veeva.
    b. Under **Password**, enter your secret key.
    c. Under **Instance name**, enter the name of your Veeva instance.
    d. Under **Data encryption**, enter your AWS KMS key.
    e. Under **Connection name**, specify a name for your connection.
    f. Choose **Connect**.



Now that you are connected to your Veeva account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 56) section above.

Extract Veeva VAULT documents with Amazon AppFlow

You can use Amazon AppFlow to extract documents from Veeva VAULT. Follow the steps below to configure a flow to extract documents.

Step 1: Create a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.

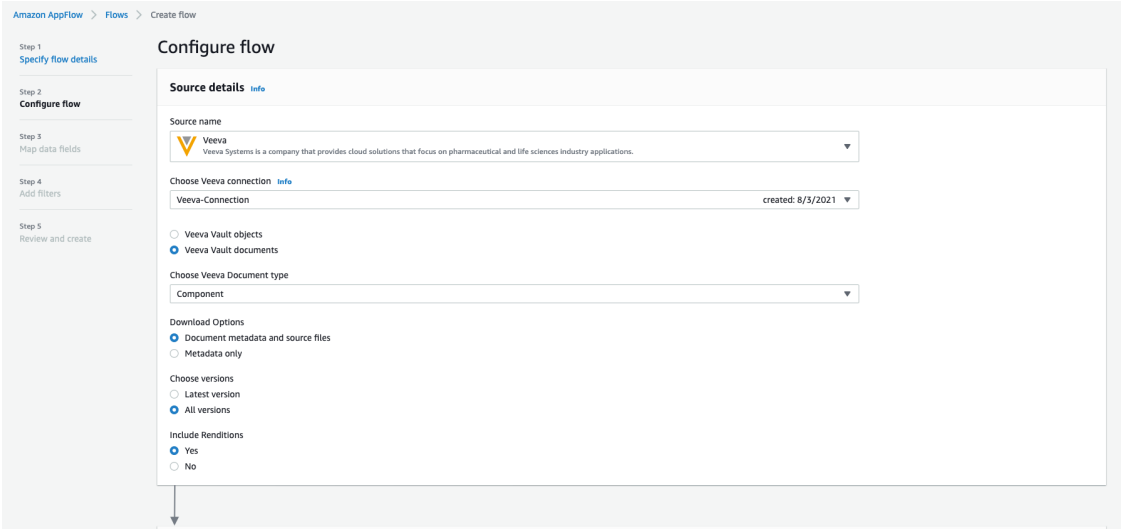
6. Choose **Next**.

Step 2: Configure the flow

1. Choose **Veeva VAULT** from the **Source name** dropdown list.
2. Choose a Veeva VAULT connection from already existing connections or create a new connection.
3. Choose **Veeva VAULT documents** from the radio options.
4. Choose a **Veeva VAULT document type** from the dropdown.
5. Choose **Document metadata and source files** option to extract source files along with associated metadata. Choose **Metadata only** option to only download Metadata. By default Metadata only is selected.
6. If you select **Document metadata and source files.**

    a. Choose **versions** of the document you want to extract, By default only latest version of document is extracted, You can select all versions to be extracted.

    b. Choose **Renditions** options if required, By default Renditions are not included.



7. Choose a destination from drop down menu.

    Note
    Currently Amazon AppFlow only supports Amazon S3 as a destination for document extraction.

8. Choose a **Bucket Name** and **Bucket Prefix**.
9. Select a trigger to run flow. You can select **Run on demand** or **Run on Schedule**to run the flow. If you choose a scheduled trigger,you can run flows at a maximum frequency of one flow run **per hour**.
10. Choose **Next**.

Step 3: Map data fields

1. You can choose a mapping method either to **Manually map the fields** or **Upload .csv file with mapped fields** to map fields from source to destination.
2. If you choose to **Manually map the fields** choose the fields from drop down list.
3. Options like **Add formula**, **Modify Values** and **Validations** are not supported for Veeva VAULT document extraction.

4. Choose **Next**.

Step 4 (Optional): Add filters

Specify a filter to determine which records to transfer. Amazon AppFlow enables you to filter data fields by adding multiple filters and by adding criteria to a filter. If you want to filter the documents by **Document subtype** or **Document Classification** you can add the appropriate filters here.

1. Based on the selected field names choose appropriate filter condition.
2. Choose **Next**.

Step 5: Review and create

- Review the information for your flow. To change the information for a step, choose **Edit**. When you are finished, choose **Create flow**.

Notes

- When you use Veeva as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per minute.

Related resources

- API access permissions in the Veeva Product Support Portal

Zendesk

The following are the requirements and connection instructions for using Zendesk with Amazon AppFlow.

> **Note**
> You can use Zendesk as a source or a destination.

Topics

Requirements

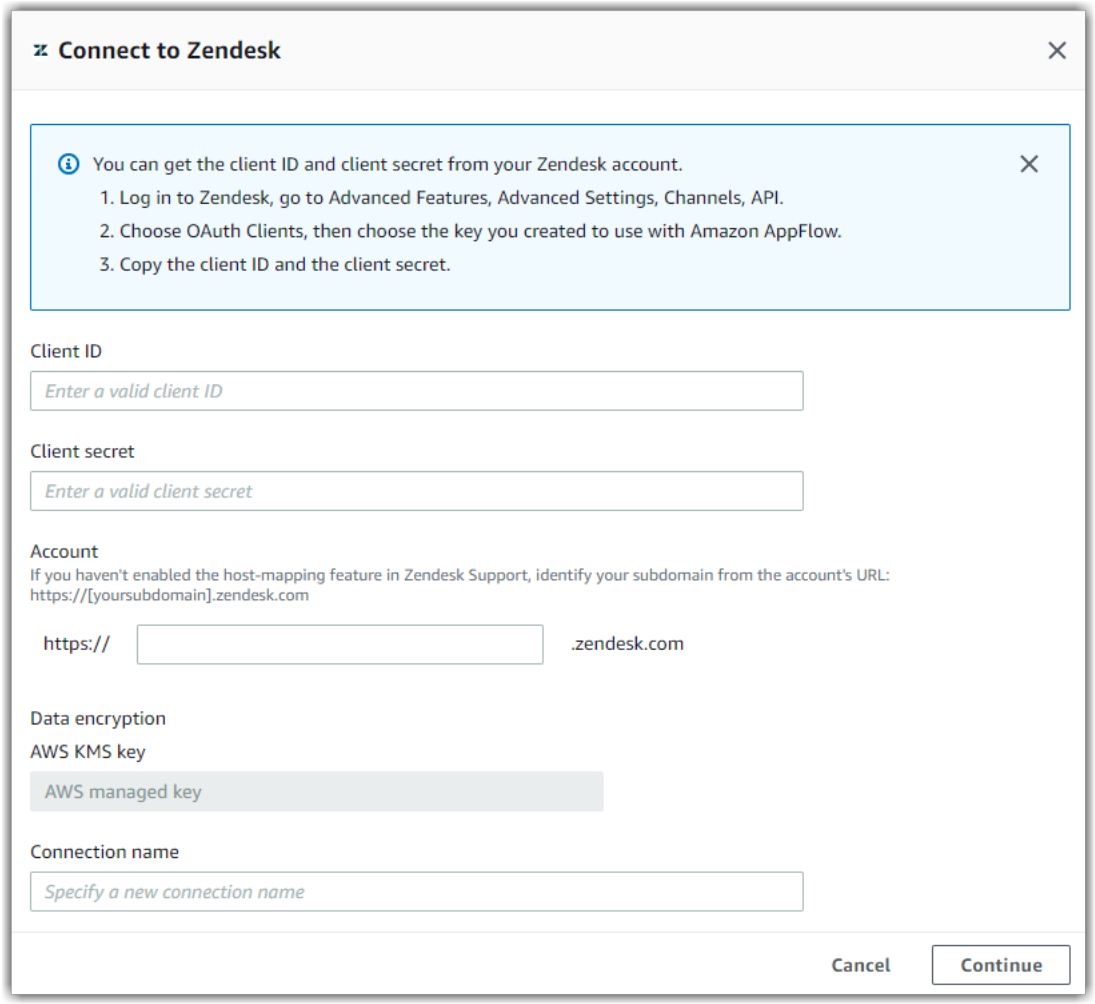
- To use Amazon AppFlow, you need to register the application to generate OAuth credentials that your application can use to authenticate API calls to Zendesk. This is done in Zendesk Support.
- In Zendesk, you must create an OAuth client with the following settings:
  - Unique identifier: aws_integration_to_Zendesk
  - Redirect URL: https://console.aws.amazon.com/appflow/oauth (us-east-1) or https://*region*.console.aws.amazon.com/appflow/oauth (all other Regions)

For more information, see Setting up the Amazon AppFlow integration with Zendesk in the Zendesk documentation.

Connection instructions

To connect to Zendesk while creating a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. Choose **Create flow**.
3. For **Flow details**, enter a name and description for the flow.
4. (Optional) To use a customer managed CMK instead of the default AWS managed CMK, choose **Data encryption**, **Customize encryption settings** and then choose an existing CMK or create a new one.
5. (Optional) To add a tag, choose **Tags**, **Add tag** and then enter the key name and value.
6. Choose **Next**.
7. Choose **Zendesk** from the **Source name** or **Destination name** dropdown list.
8. Choose **Connect** to open the **Connect to Zendesk** dialog box.
    a. Under **Client ID**, enter your Zendesk client ID.
    b. Under **Client secret**, enter your Zendesk client secret.
    c. Under **Account**, enter the name of your instance of Zendesk.
    d. Under **Data encryption**, enter your AWS KMS key.
    e. Under **Connection name**, specify a name for your connection.
    f. Choose **Continue**.



Now that you are connected to your Zendesk account, you can continue with the flow creation steps as described in Getting started with Amazon AppFlow (p. 5).

Tip
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 59).

Notes

- When you use Zendesk as a source, you can run schedule-triggered flows at a maximum frequency of one flow run per minute.
- When you use Zendesk as a destination, the following additional settings are available:

Setting name	Description
Insert new records	• This is the default data transfer option.

Setting name	Description
	• When you choose this setting, Amazon AppFlow inserts your source data into the chosen Zendesk object as a new record.
Update existing records	• When you choose this setting, Amazon AppFlow uses your source data to update existing records in Zendesk. For every source record, Amazon AppFlow looks for a matching record in Zendesk based on your criteria. You can specify matching criteria on the Map data fields page. To do so, select a field in the source application and map it to a Zendesk record ID or external field using the dropdown list. • When a matching record is found, Amazon AppFlow updates the record in Zendesk. If no matching record is found, Amazon AppFlow ignores the record or fails the flow per your chosen error handling option. You can specify your error handling preferences on the Configure flow page.
Upsert records	• When you choose this setting, Amazon AppFlow performs an upsert operation in Zendesk. For every source record, Amazon AppFlow looks for a matching record in Zendesk based on your criteria. You can specify matching criteria on the Map data fields page. To do so, select a field in the source application and map it to a Zendesk external field using the dropdown list. • When a matching record is found, Amazon AppFlow updates the record in Zendesk. If no matching record is found, Amazon AppFlow inserts the data as a new record. Any errors in performing the operation are handled per your chosen error handling option. You can specify your error handling preferences on the Configure flow page.

Related resources

- Setting up the Amazon AppFlow integration with Zendesk in the Zendesk documentation
- Building great customer experiences with Zendesk and AWS from Zendesk
- Video: How to transfer data from Zendesk Support to Amazon S3 using Amazon AppFlow

Amazon AppFlow flows

With Amazon AppFlow, a *flow* transfers data between a source and a destination. Amazon AppFlow supports a variety of AWS services and SaaS applications as sources or destinations.

A *data mapping* determines how data from the source is placed in the destination. You can map the fields in each source object to fields in the destination. You can concatenate multiple fields in a source object to a single field in the destination. You can mask the values of sensitive fields so that the destination field contains only an asterisk (*). You can also truncate fields to a fixed length.

A *filter* controls which data records are transferred to the destination. Amazon AppFlow transfers only the records that meet the filter criteria.

A *trigger* determines how a flow runs. The following are the supported flow trigger types:

- **Run on demand** — Users manually run the flow as needed.
- **Run on event** — Amazon AppFlow runs the flow in response to an event from a SaaS application.
- **Run on schedule** — Amazon AppFlow runs the flow on a recurring schedule.

When a flow is run, Amazon AppFlow verifies that the data is available in the source, processes the data according to the flow configuration, and transfers the processed data to the destination.

To create a flow

Follow the directions in Getting started with Amazon AppFlow (p. 5).

To view flow details

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane, select **Flows**.
3. Select the name of the flow from the **Flow name** column.
4. To view information about the source and destination for the flow, see **Flow details**.
5. To view information about how data is mapped between the source and destination, see **Data field settings**.
6. To view information about the runs for the flow, choose **Execution history**.

To work with a flow

- Activate an Amazon AppFlow flow (p. 63)
- Edit an Amazon AppFlow flow (p. 64)
- Delete an Amazon AppFlow flow (p. 64)
- Flow triggers (p. 64)
- Private Amazon AppFlow flows (p. 66)
- Flow notifications (p. 67)

Activate an Amazon AppFlow flow

Flows are configured with a flow trigger that determines how a flow runs. Flows can be run on a schedule, based on an event, or on demand.

If the flow trigger is a schedule or an event, you must activate a flow after you save it and you can deactivate it as needed. If the flow trigger is run on demand, you must run the flow each time you want to transfer the data.

To activate a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane, select **Flows**.
3. Select the name of the flow from the **Flow name** column.
4. Choose **Run flow**.

Edit an Amazon AppFlow flow

After you create a flow, you can change the field mappings, trigger type, and filters. You cannot change the flow name, source, or destination. The changes apply only to flow runs that occur after you save your changes.

To edit a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane, select **Flows**.
3. From the **Flow name** column, select the name of the flow.
4. (Optional) To edit the field mapping, choose **Data field settings**, **Edit data fields**. When you are finished making changes, choose **Save**.
5. (Optional) To edit the validations, choose **Data field settings**, **Edit validations**. When you are finished making changes, choose **Save**.
6. (Optional) To edit the filters, choose **Filters**, **Edit filters**. When you are finished making changes, choose **Save**.

Delete an Amazon AppFlow flow

After you delete a flow, it no longer runs.

To delete a flow

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane, select **Flows**.
3. From the **Flow name** column, select the name of the flow.
4. Choose **Actions**, **Delete flow**.
5. When prompted for confirmation, type **`delete`** and then choose **Delete**.

Flow triggers

A *trigger* determines how a flow runs. The following are the supported flow trigger types:

- **Run on demand** — Users manually run the flow as needed.

- **Run on event** — Amazon AppFlow runs the flow in response to an event from an SaaS application.
- **Run on schedule** — Amazon AppFlow runs the flow on a recurring schedule.

On demand flows

You can manually run on-demand flows as needed. You must run this type of flow each time you want to transfer the data. For more information, see Activate an Amazon AppFlow flow (p. 63).

Event-triggered flows

Amazon AppFlow runs event-triggered flows based on a specified change event in the source application.

This option is available only for SaaS applications that provide change events. You must choose the event when you choose the source.

Schedule-triggered flows

Amazon AppFlow runs schedule-triggered flows based on the schedule that you specify during flow setup. The scheduling frequency depends on the frequency supported by the source application.

You can choose either full or incremental data transfer for schedule-triggered flows.

Full transfer

When you select full transfer, Amazon AppFlow transfers a snapshot of all records at the time of the flow run from the source to the destination.

Incremental transfer

When you select incremental transfer, Amazon AppFlow transfers only the records that have been added or changed since the last successful flow run. You can also select a source timestamp field to specify how Amazon AppFlow identifies new or changed records. For example, if you have a *Created Date* timestamp field, choose this to instruct Amazon AppFlow to transfer only newly-created records (and not changed records) since the last successful flow run. The first schedule-triggered flow will pull 30 days of past records at the time of the first flow run.

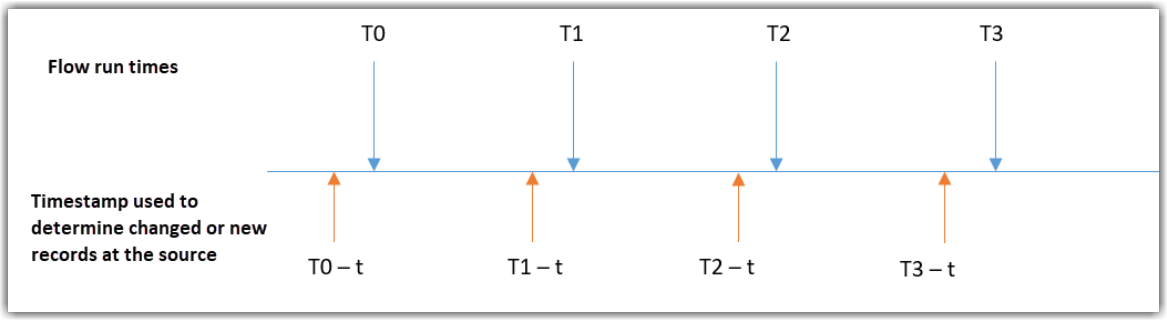
> **Tip**
> To transfer records created or modified over a different time range other than the past 30 days at the time of the first flow run, set up the flow to be triggered on demand. You can then use the filter option to pull records over the desired time range. After the on-demand flow runs and pulls the initial set of records, edit the flow to be triggered on schedule so that subsequent flow runs transfer incremental data.

Offset option

Optionally, you can add a time offset (*t*) to the time range for the incremental transfer. The flow run will import records that were created or changed between the previous flow run and the specified offset prior to the current flow run. This feature can be used to accommodate any latencies in the source systems in timestamping changes to records. By choosing a sufficiently large offset, you can avoid missing records that changed in the source application close to the run time of the scheduled flow.

If a schedule-triggered flow runs at time instances *T0*, *T1*, *T2*, and so on, then records that are new or have changed between *T0 minus t* and *T1 minus t* will be imported from the source at *T1*, and those that have changed between *T1 minus t* and *T2 minus t* will be imported from the source at *T2*.

The total offset value can be longer than the schedule interval (for example, *t* can be longer than *T1 minus T0*), but it must be less than 10 hours. The default value is 0.



- The flow run at *T0* transfers records that changed between *T0 minus 30 days* and *T0 minus t* in the source application.
- The flow run at *T1* transfers records that changed between *T0 minus t* and *T1 minus t* in the source application.
- The flow run at *T2* transfers records that changed between *T1 minus t* and *T2 minus t* in the source application.
- The flow run at *T3* transfers records that changed between *T2 minus t* and *T2 minus t* in the source application.

Private Amazon AppFlow flows

With Amazon AppFlow, you can create private flows between AWS services and supported software as a service (SaaS) applications. Private flows use AWS PrivateLink to route data over AWS infrastructure without exposing it to the public internet.

The following SaaS applications are integrated with AWS PrivateLink:

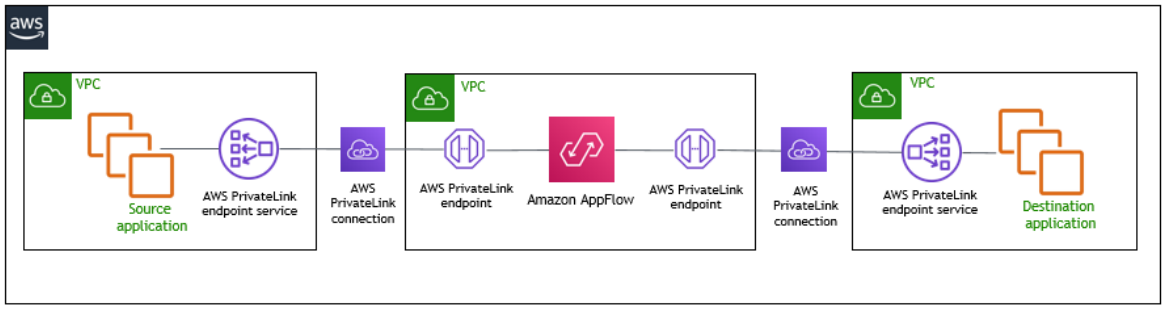
- Salesforce
- Singular
- Snowflake
- Trend Micro

> **Note**
> Your SaaS account must be enabled for AWS PrivateLink access. Please check with the administrator for the SaaS application.

When you create a connection using AWS PrivateLink, Amazon AppFlow creates the VPC endpoint service configuration for you. When you no longer need the endpoint service configuration, Amazon AppFlow deletes it.

> **Note**
> Amazon AppFlow makes metadata API calls to populate a list of objects and fields in the console over the public endpoints. However, the actual data transfer during the flow run happens over Amazon VPC endpoints powered by AWS PrivateLink.

The following diagram illustrates the components of a private flow.



Flow notifications

Note
Amazon CloudWatch Events and Amazon EventBridge are both the same underlying service and API. Changes you make in either CloudWatch Events or EventBridge will appear in each console.

Amazon AppFlow is integrated with Amazon CloudWatch Events to publish events related to the status of a flow. The following flow events are published to your default event bus.

- **AppFlow Start Flow Run Report**: This event is published at the start of a flow run.
- **AppFlow End Flow Run Report**: This event is published when a flow run is complete.
- **AppFlow Event Flow Report**: This event is generated every five minutes for an event-triggered flow, and provides a count of event triggers over the five minute interval.
- **AppFlow Event Flow Deactivated**: This event is generated when Amazon AppFlow deactivates an event-triggered flow due to a failure. The deactivation reason is specified in the event payload.
- **AppFlow Scheduled Flow Deactivated**: This event is generated when Amazon AppFlow deactivates a schedule-triggered flow due to a failure. The deactivation field is specified in the event payload.

You can access these events in your CloudWatch Event Console by creating an appropriate rule.

To configure the Event Source in CloudWatch

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Events**, **Rules**.
3. Choose **Create rule** to create a new rule, or select an existing rule and choose **Actions, Edit**.
4. For **Event Source**, do the following:
   a. Choose **Event Pattern**.
   b. Choose **Build event pattern to match events by service**.
   c. For **Service Name**, choose **Appflow**.
   d. For **Event Type**, select the flow event name.
5. Alternatively, choose **Edit** to edit the event source, and enter the following (replacing the *placeholder text* with the flow event name):

```
{
  "source": [
    "aws.appflow"
  ],
  "detail-type": [
    "flow event name"
  ]
```

```
}
```

For further details on using CloudWatch, see the Amazon CloudWatch User Guide.

Common fields

All event payloads include the following common fields:

account

The 12 digit number identifying the AWS account.

detail-type

The name of the event. See the preceding list of flow events for more information.

id

The unique value generated for every event.

region

The AWS region where the event originated.

resources

The ARNs (AWS Resource Numbers) that identify the resources involved in the event.

source

"aws.appflow".

time

The event timestamp.

version

The flow version. By default, this is set to 0 (zero) in all events.

Flow event detail fields

The following fields are available as part of the flow event details:

created-by

The ARN of the user who created the flow.

destination

The details of the destination connector for the flow.

destination-object

The destination object chosen in the flow.

flow-arn

The ARN of the flow.

flow-name

The name of the flow selected at the time of the flow creation.

source

The details of the source connector for the flow.

source-object

The source object chosen in the flow.

trigger-type

The flow trigger.

The following table shows the additional event field details.

Name of the flow event	Field	Description
AppFlow Start Flow Run Report	start-time	The timestamp of the start of the flow run.
AppFlow Start Flow Run Report, **AppFlow End Flow Run Report**	incremental-transfer-time-range	The start and end timestamps that Amazon AppFlow sent to the source application, indicating the time range for the incremental record transfer. This is available only for schedule-triggered flows.
AppFlow Event Flow Deactivated, **AppFlow Scheduled Flow Deactivated**	deactivation-reason	The reason for deactivation.
AppFlow Event Flow Deactivated, **AppFlow Scheduled Flow Deactivated**	deactivation-time	The time at which the flow was deactivated.
AppFlow Event Flow Report	status-report	The count of event triggers received from the source, and the timestamp of the five minute interval over which this count was calculated. This is available only for event-triggered flows.
AppFlow End Flow Run Report	end-time	The timestamp of the flow run completion.
AppFlow End Flow Run Report	num-of-records-processed	The number of records from the source that were processed by Amazon AppFlow.
AppFlow End Flow Run Report	num-of-record-failures	The number of records that could not be inserted into the destination.
AppFlow End Flow Run Report	data-processed	The volume of data (in bytes) that was processed.

Name of the flow event	Field	Description
AppFlow End Flow Run Report	status	The status that indicates if the flow run failed or was successful.
AppFlow End Flow Run Report	error	The reason for flow run failure in the event of a failed flow.

Security in Amazon AppFlow

Amazon AppFlow provides a secure platform that enables you to move data bi-directionally between AWS services and software as a service (SaaS) applications, with availability in multiple Regions and built-in redundancy.

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon AppFlow, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations

This documentation helps you understand how to apply the shared responsibility model when using Amazon AppFlow. It shows you how to configure Amazon AppFlow to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon AppFlow resources.

Contents

Data protection in Amazon AppFlow

The AWS shared responsibility model applies to data protection in Amazon AppFlow. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.

- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with Amazon AppFlow or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption at Rest

When you configure an SaaS application as a source or destination, you create a connection. This includes information required for connecting to the SaaS applications, such as authentication tokens, user names, and passwords. Amazon AppFlow securely stores your connection data, encrypting it using AWS Key Management Service (AWS KMS) customer master keys (CMK) and then storing it in AWS Secrets Manager.

When you delete a connection, all its metadata is permanently deleted.

When you use Amazon S3 as a destination, you can choose either an AWS managed CMK or a customer managed CMK for encrypting the data in the S3 bucket using Amazon S3 SSE-KMS.

Encryption in Transit

When you configure a flow, you can choose either an AWS managed CMK or a customer managed CMK. When executing a flow, Amazon AppFlow stores data temporarily in an intermediate S3 bucket and encrypts it using this key. This intermediate bucket is deleted after 7 days, using a bucket lifecycle policy.

Amazon AppFlow secures all data in transit using Transport Layer Security (TLS) 1.2.

With some of the SaaS applications that are a supported source or destination, you can create a connection that does not send traffic over the public internet. For more information, see Private Amazon AppFlow flows (p. 66).

Key Management

Amazon AppFlow provides both AWS managed and customer managed CMKs for encrypting connection data and data stored in Amazon S3 when it is a destination. We recommend that you use a customer managed CMK, as it puts you in full control over your encrypted data. When you choose a customer managed CMK, Amazon AppFlow attaches a resource policy to the CMK that grants it access to the CMK.

Connection credentials

Amazon AppFlow stores the encrypted credentials that are used to connect to flow source and destination applications in your AWS Secrets Manager account. These credentials include OAuth tokens,

Application and API keys, and passwords. To create a new connection, grant the following permissions to any custom IAM policies.

Note
The `AmazonAppFlowFullAccess` policy includes these permissions.

```
{
          "Sid": "SecretsManagerCreateSecretAccess",
          "Effect": "Allow",
          "Action": "secretsmanager:CreateSecret",
          "Resource": "*",
          "Condition": {
               "StringLike": { "secretsmanager:Name": "appflow!*"
            },
            "ForAnyValue:StringEquals": {
                  "aws:CalledVia": [
                        "appflow.amazonaws.com"
                    ]
               }
          }
},
{
          "Sid": "SecretsManagerPutResourcePolicyAccess",
          "Effect": "Allow",
          "Action": [
               "secretsmanager:PutResourcePolicy"
          ],
          "Resource": "*",
          "Condition": {
               "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                         "appflow.amazonaws.com"
                    ]
          },
          "StringEqualsIgnoreCase": {
               "secretsmanager:ResourceTag/aws:secretsmanager:owningService": "appflow"
          }
     }
}
```

Identity and access management for Amazon AppFlow

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Amazon AppFlow resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Amazon S3 Bucket Policies for Amazon AppFlow (p. 89)
- AWS managed policies for Amazon AppFlow (p. 92)
- Troubleshooting Amazon AppFlow identity and access (p. 95)

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Amazon AppFlow.

Service user – If you use the Amazon AppFlow service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Amazon AppFlow features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Amazon AppFlow, see Troubleshooting Amazon AppFlow identity and access (p. 95).

Service administrator – If you're in charge of Amazon AppFlow resources at your company, you probably have full access to Amazon AppFlow. It's your job to determine which Amazon AppFlow features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Amazon AppFlow, see How Amazon AppFlow works with IAM (p. 77).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Amazon AppFlow. To view example Amazon AppFlow identity-based policies that you can use in IAM, see Identity-based policy examples for Amazon AppFlow (p. 85).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see Signing in to the AWS Management Console as an IAM user or root user in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the AWS Management Console, use your password with your root user email address or your IAM user name. You can access AWS programmatically using your root user or IAM users access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 signing process in the *AWS General Reference*.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then

securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see Managing access keys for IAM users in the *IAM User Guide*. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the *IAM User Guide*.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for Amazon AppFlow in the *Service Authorization Reference*.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. You can sign in as the root user or an IAM user, or you can assume an IAM role. When you then make a request, AWS evaluates the related identity-based or resource-based policies. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-

based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How Amazon AppFlow works with IAM

Before you use IAM to manage access to Amazon AppFlow, learn what IAM features are available to use with Amazon AppFlow.

IAM features you can use with Amazon AppFlow

IAM feature	Amazon AppFlow support
Identity-based policies (p. 78)	Yes
Resource-based policies (p. 79)	No
Policy actions (p. 79)	Yes
Policy resources (p. 80)	Yes
Policy condition keys (p. 81)	Partial
ACLs (p. 82)	No
ABAC (tags in policies) (p. 82)	Yes
Temporary credentials (p. 82)	Yes
Principal permissions (p. 82)	Yes
Service roles (p. 83)	No
Service-linked roles (p. 83)	No

To get a high-level view of how Amazon AppFlow and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for Amazon AppFlow

Supports identity-based policies	Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Other required permissions in identity-based policies for Amazon AppFlow

Because Amazon AppFlow always encrypts data at rest and in motion, ensure that the user that is creating and running a flow has the following AWS KMS permissions in your identity-based policies.

Required AWS KMS permission	Description
kms:ListKeys	Controls permission to view the key ID and Amazon Resource Name (ARN) of all customer master keys (CMKs) in the account.
kms:DescribeKey	Controls permission to view detailed information about a CMK.

Required AWS KMS permission	Description
kms:ListAliases	Controls permission to view the aliases that are defined in the account. Aliases are optional friendly names that you can associate with CMKs.
kms:CreateGrant	Controls permission to add a grant to a CMK. You can use grants to add permissions without changing the key policy or IAM policy.
kms:ListGrants	Controls permission to view all grants for a CMK.

For more information about AWS Key Management Service (AWS KMS), see What is AWS KMS in the *AWS Key Management Service Developer Guide*.

For the complete list of AWS services that are integrated with AWS KMS, see AWS Service Integration.

Identity-based policy examples for Amazon AppFlow

To view examples of Amazon AppFlow identity-based policies, see Identity-based policy examples for Amazon AppFlow (p. 85).

Resource-based policies within Amazon AppFlow

Supports resource-based policies	No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

Policy actions for Amazon AppFlow

Supports policy actions	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also

some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Amazon AppFlow actions, see Actions defined by Amazon AppFlow in the *Service Authorization Reference*.

Policy actions in Amazon AppFlow use the following prefix before the action.

```
appflow
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "appflow:CreateConnectorProfile",
      "appflow:CreateFlow"
         ]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action.

```
"Action": "appflow:Describe*"
```

To view examples of Amazon AppFlow identity-based policies, see Identity-based policy examples for Amazon AppFlow (p. 85).

Policy resources for Amazon AppFlow

Supports policy resources	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of Amazon AppFlow resource types and their ARNs, see Resources defined by Amazon AppFlow in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by Amazon AppFlow.

An Amazon AppFlow connector profile has the following Amazon Resource Name (ARN) format.

```
arn:${Partition}:appflow:${Region}:${Account}:connectorprofile/${connector-profile-name}
```

An Amazon AppFlow flow has the following ARN format.

```
arn:${Partition}:appflow:${Region}:${Account}:flow/${flow-name}
```

For more information about the format of ARNs, see Amazon Resource Names (ARNs).

For example, to specify the `test-flow` flow in your statement, use the following ARN.

```
"Resource": "arn:aws:appflow:us-east-1:123456789012:flow/test-flow"
```

To specify all flows that belong to a specific account, use the wildcard (*).

```
"Resource": "arn:aws:appflow:us-east-1:123456789012:flow/*"
```

Some Amazon AppFlow actions, such as those for creating resources, cannot be performed on a specific resource. In those cases, you must use the wildcard (*).

```
"Resource": "*"
```

Many Amazon AppFlow API actions involve multiple resources. For example, `DescribeConnectorProfiles` returns a list of details for specified connector profiles that are accessible by the currently logged in AWS account. So an IAM user must have permissions to view those connector profiles. To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [
      "resource1",
      "resource2"
```

To see a list of Amazon AppFlow resource types and their ARNs, see Resources defined by Amazon AppFlow in the *IAM User Guide*. To learn about actions with which you can specify the ARN of each resource, see Actions defined by Amazon AppFlow.

Policy condition keys for Amazon AppFlow

Supports policy condition keys	Partial

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

Amazon AppFlow does not provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

Access control lists (ACLs) in Amazon AppFlow

Supports ACLs	No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Attribute-based access control (ABAC) with Amazon AppFlow

Supports ABAC (tags in policies)	Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

For more information about ABAC, see What is ABAC? in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with Amazon AppFlow

Supports temporary credentials	Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switching to a role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for Amazon AppFlow

Supports principal permissions	Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for Amazon AppFlow in the *Service Authorization Reference*.

Service roles for Amazon AppFlow

Supports service roles	No

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

Service-linked roles for Amazon AppFlow

Supports service-linked roles	No

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

Managing permissions for Amazon AppFlow users

The following sections walk you through managing permissions for Amazon AppFlow users. You can learn how to add new users, grant access for existing users, change access levels, and create custom access policies.

Topics

Adding a new user

The following procedure shows how to add a new user to Amazon AppFlow and grant a permissions policy to the new user.

To add a new user

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane, choose **Users**, and then choose **Create user**. This takes you directly to the **User** page on the IAM console.
3. Choose **Add user**.
4. Enter a user name for the new user.
5. Choose **AWS Management Console Access**. This allows the user to sign in based on the password that you assign.

6. Choose an auto-generated or custom password.

   Tip
   We recommend that you select the option that requires the user to reset the password.
7. Choose **Next: Permissions**.
8. Under **Set permissions**, choose **Attach existing policies directly**.
9. Search for one of the predefined Amazon AppFlow policies.
   - Alternatively, you can create your own policy (p. 85).
   - You can also attach policies for other AWS services at this time, if needed. For example, enter `S3` in the search box to see available policies for accessing Amazon S3.
10. Choose **Next: Tags**. Adding tags is optional.
11. Choose **Next: Review** and review your choices.
12. Choose **Create user** to create the user and view their security credentials, which you can now download. This is the last time these credentials will be available to download. However, you can create new credentials at any time.
13. Choose the **Send email** link to send login instructions to the new user.

    Tip
    We recommend that you send the password in a separate email.

Granting access for existing users

The following procedure shows how to grant Amazon AppFlow access to an existing IAM user.

To grant access

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane, choose **Users**, and then choose **Create user**. This takes you directly to the **User** page on the IAM console.
3. Choose the user who requires Amazon AppFlow permissions.
4. Choose **Add permissions**.
5. Under **Set permissions**, choose **Attach existing policies directly**.
6. Search for one of predefined Amazon AppFlow policies.
   - Alternatively, you can create your own policy (p. 85).
7. Choose **Next: Review** to review the permissions that you added.
8. Choose **Add permissions**.

Changing access levels

The following procedure shows how to change the Amazon AppFlow access level of an existing IAM user.

To change access levels for existing users

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane, choose **Users**, and then choose **Create user**. This takes you directly to the **User** page on the IAM console.
3. Choose the user whose Amazon AppFlow access you want to change.
4. Choose **X** next to the existing policy that you want to delete.
5. Choose **Detach**.

6. Choose **Add permissions** to add a new policy.
7. Under **Set permissions**, choose **Attach existing policies directly**.
8. Search for one of predefined Amazon AppFlow policies.
   - Alternatively, you can create your own policy (p. 85).
9. Choose **Next: Review** to review the permissions you have added.
10. Choose **Add permissions**.

Creating custom access policies

You can create a custom IAM policy and assign it to a user, group, or role.

In the action `Action` of your custom policy statement, you can specify the desired permissible actions for Amazon AppFlow. To see a full list of Amazon AppFlow actions, see Actions defined by Amazon AppFlow in the *Service Authorization Reference*.

To learn more about the Amazon AppFlow-specific resources, actions, and condition context keys used in IAM permissions policies, see Actions, resources, and condition keys for Amazon AppFlow in the *IAM User Guide*.

To create a custom access policy

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. Choose **Policies**.
3. Choose **Create policy**.
4. In the visual editor, choose Amazon AppFlow as the service, and follow the instructions to add specific permissions to the policy that you create.

Identity-based policy examples for Amazon AppFlow

By default, IAM users and roles don't have permission to create or modify Amazon AppFlow resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform actions on the resources that they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating policies on the JSON tab in the *IAM User Guide*.

Topics

Policy best practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete Amazon AppFlow resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started using AWS managed policies** – To start using Amazon AppFlow quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get started using permissions with AWS managed policies in the *IAM User Guide*.
- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant least privilege in the *IAM User Guide*.
- **Enable MFA for sensitive operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.

Example 1: Allow IAM users full administrator access to Amazon AppFlow

This policy example provides full access to Amazon AppFlow, to all AWS services that are available as flow sources or destinations, and to AWS Key Management Service (AWS KMS).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "appflow:*",
            "Resource": "*"
        },
        {
            "Sid": "ListRolesForRedshift",
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "*"
        },
        {
            "Sid": "KMSListAccess",
            "Action": [
                "kms:ListKeys",
                "kms:DescribeKey",
                "kms:ListAliases"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Sid": "KMSGrantAccess",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "appflow.*.amazonaws.com"
                },
                "Bool": {
```

```
                "kms:GrantIsForAWSResource": "true"
            }
        }
    },
    {
        "Sid": "KMSListGrantAccess",
        "Effect": "Allow",
        "Action": [
            "kms:ListGrants"
        ],
        "Resource": "*",
        "Condition": {
            "StringLike": {
                "kms:ViaService": "appflow.*.amazonaws.com"
            }
        }
    },
    {
        "Sid": "S3ReadAccess",
        "Effect": "Allow",
        "Action": [
            "s3:ListAllMyBuckets",
            "s3:ListBucket",
            "s3:GetBucketLocation",
            "s3:GetBucketPolicy"
        ],
        "Resource": "*"
    },
    {
        "Sid": "S3PutBucketPolicyAccess",
        "Effect": "Allow",
        "Action": [
            "s3:PutBucketPolicy"
        ],
        "Resource": "arn:aws:s3:::appflow-*"
    },
    {
        "Sid": "SecretsManagerCreateSecretAccess",
        "Effect": "Allow",
        "Action": "secretsmanager:CreateSecret",
        "Resource": "*",
        "Condition": {
            "StringLike": {
                "secretsmanager:Name": "appflow!*"
            },
            "ForAnyValue:StringEquals": {
                "aws:CalledVia": [
                    "appflow.amazonaws.com"
                ]
            }
        }
    },
    {
        "Sid": "SecretsManagerPutResourcePolicyAccess",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:PutResourcePolicy"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals": {
                "aws:CalledVia": [
                    "appflow.amazonaws.com"
                ]
            },
            "StringEqualsIgnoreCase": {
```

```
                    "secretsmanager:ResourceTag/aws:secretsmanager:owningService":
 "appflow"
                }
            }
        }
    ]
}
```

Example 2: Allow IAM users read-only access to Amazon AppFlow

This policy example provides read-only access to Amazon AppFlow.

For definitions of each action, see Actions defined by Amazon AppFlow.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
            "appflow:DescribeConnectors",
            "appflow:DescribeConnectorProfiles",
            "appflow:DescribeFlows",
            "appflow:DescribeFlowExecution",
            "appflow:DescribeConnectorFields",
            "appflow:ListConnectorFields",
            "appflow:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Example 3: Grant access to permission-only actions

If you use a custom policy to grant users permission to use Amazon AppFlow instead of the managed policies provided, you need to include specific permissions for the user or role to perform specific actions. For example, if the user or role needs to add or update a flow, the policy attached to the user or role must include permission to use the `UseConnectorProfile` permission-only action so that the user has permission to use the connection specified for the flow. You can specify that the user is allowed to use all connector profiles, or only a specific connector profile. The following example policy statement demonstrates how to grant access only to a specific connector profile by specifying the ARN to the connector profile named *test-profile* in the account 123456789012. You can modify this policy statement and include it in a custom policy for your environment, but this statement grants permission only to use the connector profile. The user or role needs additional permissions to perform other Amazon AppFlow actions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowConnectionProfile",
            "Effect": "Allow",
            "Action": "appflow:UseConnectorProfile",
            "Resource": "arn:aws:appflow:us-east-1:123456789012:connectorprofile/test-
profile"
        }
    ]
```

```
}
```

Example 4: Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon S3 Bucket Policies for Amazon AppFlow

By default, all Amazon S3 buckets and objects are private. Only the resource owner, the AWS account that created the bucket, can access the bucket and any objects that it contains. However, the resource owner can choose to grant access permissions to other resources and users by writing an access policy.

If you want to create or modify an Amazon S3 bucket to be used as a source or destination in a flow, you must further modify the bucket policy. To read from or write to an Amazon S3 bucket, Amazon AppFlow must have the the following permissions. Amazon AppFlow automatically attaches the required permissions to a bucket when you select an Amazon S3 bucket as either the source or destination in a flow in the Amazon AppFlow console. If using the Amazon AppFlow SDK these policies must be added manually.

Amazon AppFlow Required Amazon S3 Policies

Amazon AppFlow requires a permission policy with the following attributes:

- The statement SID

- The bucket name
- The service principal name for Amazon AppFlow.
- The resources required for Amazon AppFlow: the bucket and all of its contents
- The required actions that Amazon AppFlow needs to take, which varies depending on if the bucket is used as a source or destination

The following policy allows Amazon AppFlow to access an Amazon S3 bucket used as the source in a flow. It contains all of the necessary actions Amazon AppFlow needs to read objects from the specified bucket.

Amazon S3 bucket policy

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Sid": "AllowAppFlowSourceActions",
            "Principal": {
                "Service": "appflow.amazonaws.com"
            },
            "Action": [
                "s3:ListBucket",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::myBucketName",
                "arn:aws:s3:::myBucketName/*"
            ]
        }
    ]
}
```

The following policy allows Amazon AppFlow to access an Amazon S3 bucket used as the destination in a flow. It contains all of the necessary actions Amazon AppFlow needs to put objects into an Amazon S3 bucket.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Sid": "AllowAppFlowDestinationActions",
            "Principal": {
                "Service": "appflow.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:AbortMultipartUpload",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::myBucketName",
                "arn:aws:s3:::myBucketName/*"
            ]
        }
    ]
```

```
}
```

Cross-service confused deputy prevention

The Confused Deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform that action in AWS. Cross-service impersonation is one means of creating a confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The called service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to do. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that Amazon AppFlow gives another service to the resource. If you use both global condition context keys, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must use the same account ID when used in the same policy statement.

The value of `aws:SourceArn` must be the resource that is accessing the Amazon S3 bucket. The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. For Amazon AppFlow, these will be the ARNs of the flows created with Amazon S3 as a source or destination. If you would like to specify multiple different flows, you may use a list of different ARNs for the `aws:SourceArn` context key. Additionally, you may use the `aws:SourceArn` global context condition key with wildcards (*). For example, `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`. The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in Amazon S3 to prevent the confused deputy problem when Amazon S3 is the destination (Note the Amazon AppFlow console automatically populates the `aws:SourceAccount` condition key to its Amazon S3 policy put in the your Amazon S3 bucket during flow creation).

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Sid": "AllowAppFlowDestinationActions",
            "Principal": {
                "Service": "appflow.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:AbortMultipartUpload",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::myBucketName",
                "arn:aws:s3:::myBucketName/*"
            ],
            "Condition" : {
                "StringEquals" : {
                    "aws:SourceAccount" : "myAccountId"
                },
                "ArnLike" : {
                    "aws:SourceArn": ["arn:aws:appflow::myAccountId:flow/flow-name-1",
                                      "arn:aws:appflow::myAccountId:flow/flow-name-2"]
                }
            }
        }
```

```
    ]
}
```

Cross-service confused deputy prevention for DescribeConnectorEntity

As part of its DescribeConnectorEntity API, Amazon AppFlow will make calls to Amazon S3 in order to get information about specific objects in a customer's Amazon S3 bucket. The DescribeConnectorEntity API is invoked either through the direct usage of the Amazon AppFlow SDK, or via the Amazon AppFlow console when using an Amazon S3 bucket as the source during flow creation. This API requires the following permissions:

- `S3:GetObject`
- `S3:ListBucket`

These calls are not associated with a particular resource. As such, when using `aws:SourceArn` in a bucket policy granting these permissions to Amazon AppFlow, one should use the global context condition key with wildcard if planning to use Amazon AppFlow's console or DescribeConnectorEntity API with the Amazon S3 bucket the policy is attached to.

AWS managed policies for Amazon AppFlow

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ViewOnlyAccess** AWS managed policy provides read-only access to many AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the *IAM User Guide*.

AWS managed policy: AmazonAppFlowFullAccess

You can attach the `AmazonAppFlowFullAccess` policy to your IAM identities.

This policy grants administrative permissions that allow you to view, create, update, run, and delete flows, and also to list, create, and delete connections. In addition, this policy grants access to the API actions that are required to configure other AWS services as a source or destinations. This policy also provides access to AWS Key Management Service to allow use of customer managed CMKs for encryption. It does not grant the ability to add other users.

> **Note**
> This policy automatically grants read and write permissions to S3 buckets with an `appflow-` prefix only. You will not have access rights to any other S3 buckets without this prefix.

Permissions details

This policy includes the following permissions.

- `appflow` – Allows principals to have full access to Amazon AppFlow. This is required so that you can view, create, update, run, and delete flows, in addition to list, create, and delete connections.
- `iam` – Allows principals to list IAM roles from Amazon RDS. This is required so that you can use Amazon RDS as a flow destination.
- `s3` – Allows principals to access buckets, bucket locations, and bucket policies for Amazon Simple Storage Service (Amazon S3). This is required so that you can use Amazon S3 as a flow source or destination (or use it to support the use of another source or destination).
- `kms` – Allows principals to view the key ID and Amazon Resource Name (ARN) of all the customer master keys (CMKs) in the account, view detailed information about a CMK, view the aliases that are defined in the account, and add a grant to a CMK. This is required so that you can use customer managed CMKs for encryption.
- `secretsmanager` – Allows principals to create secrets in Secrets Manager. This is required so that Amazon AppFlow can store the encrypted credentials that you use to connect to flow source and destination applications in your Secrets Manager account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "appflow:*",
            "Resource": "*"
        },
        {
            "Sid": "ListRolesForRedshift",
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "*"
        },
        {
            "Sid": "KMSListAccess",
            "Action": [
                "kms:ListKeys",
                "kms:DescribeKey",
                "kms:ListAliases"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Sid": "KMSGrantAccess",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "appflow.*.amazonaws.com"
                },
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        },
        {
```

```
            "Sid": "KMSListGrantAccess",
            "Effect": "Allow",
            "Action": [
                "kms:ListGrants"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "appflow.*.amazonaws.com"
                }
            }
        },
        {
            "Sid": "S3ReadAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "s3:GetBucketPolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3PutBucketPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::appflow-*"
        },
        {
            "Sid": "SecretsManagerCreateSecretAccess",
            "Effect": "Allow",
            "Action": "secretsmanager:CreateSecret",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "secretsmanager:Name": "appflow!*"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "appflow.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "SecretsManagerPutResourcePolicyAccess",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:PutResourcePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "appflow.amazonaws.com"
                    ]
                },
                "StringEqualsIgnoreCase": {
                    "secretsmanager:ResourceTag/aws:secretsmanager:owningService":
"appflow"
                }
            }
        }
```

```
    ]
}
```

AWS managed policy: AmazonAppFlowReadOnlyAccess

You can attach the `AmazonAppFlowReadOnlyAccess` policy to your IAM identities.

This policy grants read-only permissions that allow you to view flows and connections in an AWS account. This policy doesn't allow you to create or delete flows or connections, and it doesn't grant the ability to add other users or grant access to other AWS services.

Permissions details

This policy includes the following permissions.

- `appflow` – Allows principals to describe and list resources from Amazon AppFlow. This is required so that Amazon AppFlow users can view connectors, connector profiles, flows, and their associated metadata.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "appflow:DescribeConnectors",
                "appflow:DescribeConnectorProfiles",
                "appflow:DescribeFlows",
                "appflow:DescribeFlowExecution",
                "appflow:DescribeConnectorFields",
                "appflow:ListConnectorFields",
                "appflow:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon AppFlow updates to AWS managed policies

View details about updates to AWS managed policies for Amazon AppFlow since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Amazon AppFlow Document history page.

Change	Description	Date
Amazon AppFlow started tracking changes	Amazon AppFlow started tracking changes for its AWS managed policies.	03/26/2021

Troubleshooting Amazon AppFlow identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Amazon AppFlow and IAM.

Topics

I am not authorized to perform an action in Amazon AppFlow

The `AccessDeniedException` error appears when a user doesn't have permission to use Amazon AppFlow or the Amazon AppFlow API operations.

In this case, your administrator must update the policy to allow you access.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to Amazon AppFlow.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Amazon AppFlow. However, the action requires the service to have permissions granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary asks her administrator to update her policies to allow her to perform the `iam:PassRole` action.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

> **Important**
> Do not provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see Managing access keys in the *IAM User Guide*.

I'm an administrator and want to allow others to access Amazon AppFlow

To allow others to access Amazon AppFlow, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in Amazon AppFlow.

To get started right away, see Creating your first IAM delegated user and group in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my Amazon AppFlow resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Amazon AppFlow supports these features, see How Amazon AppFlow works with IAM (p. 77).
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

Compliance validation for Amazon AppFlow

Third-party auditors assess the security and compliance of AWS services as part of multiple AWS compliance programs, such as SOC, PCI, FedRAMP, and HIPAA.

To learn whether Amazon AppFlow or other AWS services are in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.

  Note
  Not all services are compliant with HIPAA.

- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in Amazon AppFlow

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in Amazon AppFlow

As a managed service, Amazon AppFlow is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access Amazon AppFlow through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Quotas for Amazon AppFlow

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas cannot be increased.

Flows

Your AWS account has the following quotas related to Amazon AppFlow.

- Number of flows per account: 1,000
- Number of flow runs per month: 10 million
- Number of concurrent flow runs at any time: 1000

Flow runs

Amazon AppFlow can process up to 100 GB of data as part of a single flow run. However, the following source applications place quotas on the amount of data they can process:

- Amplitude: 25 MB of data per flow run.
- Marketo:
  - Data import from Marketo: 1 GB per flow run. To transfer over 1 GB of data, you can split your workload into multiple flows by applying the appropriate filters for each flow.
  - Data export to Marketo: You can insert up to 500 MB of records into Marketo in a single flow run. If your source is Amazon S3, each CSV file cannot exceed 125 MB in size. However, you can drop multiple CSV files (each less than 125 MB) into the source bucket or folder, and Amazon AppFlow will transfer all the data to Marketo in a single flow run.
- Salesforce:
  - Data import from Salesforce: 15 GB per flow run. To transfer over 15 GB of data, you can split your workload into multiple flows by applying the appropriate filters to each flow.
  - Events from Salesforce: Amazon AppFlow currently uses a third-party library which is allocated a fixed buffer size. Amazon AppFlow's buffer size is currently set to 1 MB. If there is a sudden surge of events on a single event channel (such as `AccountChangeEvent`) that exceeds the buffer size, this might lead to events being dropped. Please create a support case in the AWS Managent Console if this happens. For more information, see Creating a support case.
  - Data export to Salesforce: You can insert, update, or upsert up to 500 MB of records into Salesforce in a single flow run. If your source is Amazon S3, each CSV file cannot exceed 125 MB in size. However, you can drop multiple CSV files (each less than 125 MB) into the source bucket or folder, and Amazon AppFlow will transfer all the data to Salesforce in a single flow run.
- ServiceNow: 100,000 records per flow run.
- Google Analytics: 9 dimensions and 10 metrics per flow run
- Amazon EventBridge: Events are limited to 256 KB. If your event exceeds this size, Amazon AppFlow publishes a summary event with a pointer to the S3 bucket where you can get the full event.

Flow frequency

Amazon AppFlow can run schedule-triggered flows up to once per minute. However, the following source applications place quotas on how frequently you can run a schedule-triggered flow:

- Amazon S3: Maximum frequency of one flow run per minute

- Amplitude: Maximum frequency of one flow run per day
- Datadog: Maximum frequency of one flow run per minute
- Dynatrace: Maximum frequency of one flow run per minute
- Google Analytics: Maximum frequency of one flow run per day
- Infor Nexus: Maximum frequency of one flow run per minute
- Marketo: Maximum frequency of one flow run per hour
- Salesforce: Maximum frequency of one flow run per minute
- Salesforce Pardot: Maximum frequency of one flow run per minute
- ServiceNow: Maximum frequency of one flow run per minute
- Singular: Maximum frequency of one flow run per hour
- Slack: Maximum frequency of one flow run per minute
- Trend Micro: Maximum frequency of one flow run per hour
- Veeva: Maximum frequency of one flow run per minute
- Zendesk: Maximum frequency of one flow run per minute

Source and destination API limits

The API calls that Amazon AppFlow makes to data sources and destinations count against any API limits for that application. For example, if you set up an hourly flow that pulls five pages of data from Salesforce, Amazon AppFlow will make a total of 120 daily API calls (24x5=120). This will count against your 24-hour Salesforce API limit. The exact SalesForce API limit in this example would vary depending on your edition and number of licenses.

Amazon AppFlow API limits

There is a soft quota of 100 connector profiles per AWS account. If you need more connector profiles than this quota allows, you can submit a request to the Amazon AppFlow team through the Amazon AppFlow support channel.

Logging Amazon AppFlow API calls with AWS CloudTrail

Amazon AppFlow is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon AppFlow. CloudTrail captures all API calls for Amazon AppFlow as events. The calls captured include calls from the Amazon AppFlow console and code calls to the Amazon AppFlow API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon AppFlow. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon AppFlow, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Amazon AppFlow information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon AppFlow, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Amazon AppFlow, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions
- Receiving CloudTrail Log Files from Multiple Accounts

All actions are logged by CloudTrail and are documented in the Amazon AppFlow API Reference. For example, calls to the `CreateFlow`, `CreateConnectorProfile` and `TagResource` API actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Understanding Amazon AppFlow log file entries

A trail is a configuration that enables delivery of events as log files to an S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following is an example of a CloudTrail log entry generated when you view the details of a flow using the Amazon AppFlow console. Amazon AppFlow does not log the response elements, as they could contain sensitive data.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Richard"
    },
    "eventTime": "2020-04-23T17:08:09Z",
    "eventSource": "appflow.amazonaws.com",
    "eventName": "DescribeFlows",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "flowNames": ["my-flow"]
    },
    "responseElements": {
    },
    "requestID": "ba96f0cf-4c4a-4e42-95b5-d6c69EXAMPLE",
    "eventID": "cce710cd-d1f8-44b3-8bd1-75184EXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Document history for user guide

The following table describes the important changes in each release of the *Amazon AppFlow User Guide* from April 22nd, 2020, onward.

update-history-change	update-history-description	update-history-date
Support for Marketo as a destination (p. 103)	You can now use Marketo as a destination. For more information, see Marketo.	May 25, 2021
Updated IAM documentation (p. 103)	The *Amazon AppFlow User Guide* now includes an enhanced IAM documentation chapter, and has started tracking changes for its AWS managed policies.	March 26, 2021
Support for Zendesk as a destination (p. 103)	You can now use Zendesk as a destination. For more information, see Zendesk.	March 22, 2021
API support for Amazon Lookout for Metrics (p. 103)	The *Amazon AppFlow API Reference* now includes the following data type for Amazon Lookout for Metrics: LookoutMetricsDestinationProperties.	February 24, 2021
API support for Amazon Honeycode (p. 103)	The *Amazon AppFlow API Reference* now includes the following data types for Amazon Honeycode: HoneycodeConnectorProfileCredentials, HoneycodeConnectorProfileProperties, HoneycodeDestinationProperties, and HoneycodeMetadata.	February 24, 2021
API support for Amazon Connect Customer Profiles (p. 103)	The *Amazon AppFlow API Reference* now includes the following data types for Amazon Connect Customer Profiles: CustomerProfilesDestinationProperties and CustomerProfilesMetadata.	February 24, 2021
Application-specific User Guide pages (p. 103)	The *Amazon AppFlow User Guide* now includes application-specific pages with requirements, instructions, notes, and related resources for each supported source and destination. For more information, see SaaS applications supported by Amazon AppFlow.	January 6, 2021
Support for Salesforce Pardot as a source (p. 103)	You can now use Salesforce Pardot as a source. For more	December 18, 2020

	information, see Salesforce Pardot.	
Support for Amazon Lookout for Metrics as a destination (p. 103)	You can now use Amazon Lookout for Metrics as a destination. For more information, see Amazon Lookout for Metrics.	December 8, 2020
Schedule-triggered flow settings (p. 103)	You can now specify a time offset when configuring incremental data transfer for schedule-triggered flows. For more information, see Incremental transfer.	December 4, 2020
Support for Amazon Honeycode as a destination (p. 103)	You can now use Amazon Honeycode as a destination. For more information, see Amazon Honeycode.	December 1, 2020
Support for Upsolver as a destination (p. 103)	You can now use Upsolver as a destination. For more information, see Upsolver.	November 20, 2020
Support for Salesforce global connected apps (p. 103)	You can use your own global connected app for Salesforce with Amazon AppFlow APIs. For more information, see Use a global connected app with Amazon AppFlow.	November 10, 2020
Support for updating records in Salesforce (p. 103)	You can now update existing records when you use Salesforce as a destination. For more information, see Salesforce, Notes.	October 21, 2020
Support for Google Analytics custom dimensions and metrics (p. 103)	You can now import custom dimensions and metrics from Google Analytics into Amazon S3. For more information, see Google Analytics, Notes.	October 21, 2020
Support for upserting and inserting records in Salesforce (p. 103)	You can now insert new records or upsert records when you use Salesforce as a destination. For more information, see Salesforce, Notes.	October 5, 2020
Schedule-triggered flow settings (p. 103)	You can now choose from additional settings when you set up a schedule-triggered flow. For more information, see Getting started with Amazon AppFlow, Step 2: Configure flow.	October 5, 2020

AWS CloudFormation support (p. 103)	Amazon AppFlow now supports AWS CloudFormation. For more information, see Related AWS services, AWS CloudFormation.	September 17, 2020
Support for Amazon EventBridge as a destination (p. 103)	Amazon AppFlow now supports Amazon EventBridge as a flow destination. For more information, see Amazon EventBridge.	August 26, 2020
Amazon AppFlow API Reference (p. 103)	You can now reference the API operations used with Amazon AppFlow. For more information, see the Amazon AppFlow API Reference.	August 26, 2020
Support for new data formats (CSV, Parquet) (p. 103)	You can now specify your preferred file format for transferred records when using Amazon S3 as a destination. For more information, see Amazon S3, Notes.	August 14, 2020
Improved filter support (p. 103)	You can now add criteria to your filters and apply multiple filters to a flow. For more information, see Add filters.	August 10, 2020
Connect over PrivateLink to Salesforce (p. 103)	Amazon AppFlow now supports connections over PrivateLink. For more information, see Private Amazon AppFlow flows.	July 22, 2020
CloudWatch integration documentation (p. 103)	Amazon AppFlow now supports CloudWatch Event integration. For more information, see Flow notifications.	July 17, 2020
Additional Amazon S3 destination settings (p. 103)	When you use Amazon S3 as a destination, you can now add timestamps to file names or place files in a timestamped folder. For more information, see Amazon S3, Notes.	July 10, 2020
IAM managed policies (p. 103)	Amazon AppFlow now supports IAM managed policies. For more information, see Identity and access management for Amazon AppFlow.	July 3, 2020
Google Analytics service quota (p. 103)	When you use Google Analytics as a source, you can include up to 9 dimensions and 10 metrics per flow run. For more information, see Quotas for Amazon AppFlow.	June 23, 2020

Initial release (p. 103)	Initial release of the Amazon AppFlow User Guide.	April 22, 2020