# Table of Contents

What is Amazon AppFlow? .......................................................... 1
Use cases ................................................................................. 1
Related AWS services ............................................................. 2
Overview .................................................................................. 4
  Supported Amazon AppFlow Connectors ................................. 4
  General information ............................................................. 4
Getting started .................................................................... 6
  Prerequisites ......................................................................... 6
Tutorial: Transfer data between applications ......................... 8
  Prerequisites ......................................................................... 9
Step 1: Upload data to Amazon S3 ........................................ 10
  (Optional) Download sample data ....................................... 11
  Create an S3 bucket ............................................................ 11
  Create a folder .................................................................... 11
  Upload data ......................................................................... 12
  Additional resources .......................................................... 12
Step 2: Connect to an application .......................................... 12
  Prerequisites ......................................................................... 12
  Create a connection ............................................................ 13
  Additional resources .......................................................... 13
Step 3: Transfer data from Amazon S3 to a SaaS destination .... 13
  Prerequisites ......................................................................... 14
  Create a flow ....................................................................... 14
  Run a flow ........................................................................... 15
  View transferred data ........................................................ 16
  (Optional) Edit flow to add validations ............................... 16
Step 4: Transfer data from a SaaS source to Amazon S3 ........ 17
  Prerequisites ......................................................................... 18
  Change data capture in Salesforce ...................................... 18
  Create a flow ....................................................................... 18
  Run a flow (event-triggered or on-demand) ......................... 20
  View transferred data ........................................................ 21
Step 5: Clean up .................................................................... 21
  Clean up in Amazon S3 ........................................................ 21
  Clean up in Amazon AppFlow .............................................. 22
  Clean up in Salesforce ........................................................ 22
Supported applications .......................................................... 23
  Amazon Connect .................................................................. 23
    Amazon Connect support .................................................. 24
    Transferring data to Amazon Connect ............................... 24
  Amazon EventBridge .......................................................... 24
    Requirements .................................................................... 25
    Connection instructions .................................................... 25
    Notes ................................................................................ 26
    Related resources ............................................................ 26
  Amazon Honeycode .......................................................... 26
    Setup instructions ............................................................ 26
    Notes ................................................................................ 27
    Related resources ............................................................ 27
  Amazon Lookout for Metrics ............................................. 27
    Requirements .................................................................... 28
    Setup instructions ............................................................ 28
    Notes ................................................................................ 29
    Related resources ............................................................ 29
<table>
<thead>
<tr>
<th>Supported destinations</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Redshift</td>
<td>29-34</td>
</tr>
<tr>
<td>Amazon Redshift support</td>
<td>29</td>
</tr>
<tr>
<td>Before you begin</td>
<td>30</td>
</tr>
<tr>
<td>Connecting to Amazon Redshift</td>
<td>31</td>
</tr>
<tr>
<td>Transferring data to Amazon Redshift</td>
<td>34</td>
</tr>
<tr>
<td>Amazon S3</td>
<td>34</td>
</tr>
<tr>
<td>Requirements</td>
<td>34</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>34</td>
</tr>
<tr>
<td>Notes</td>
<td>35</td>
</tr>
<tr>
<td>Related resources</td>
<td>36</td>
</tr>
<tr>
<td>Amplitude</td>
<td>36</td>
</tr>
<tr>
<td>Requirements</td>
<td>36</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>36</td>
</tr>
<tr>
<td>Notes</td>
<td>37</td>
</tr>
<tr>
<td>Related resources</td>
<td>38</td>
</tr>
<tr>
<td>Datadog</td>
<td>38</td>
</tr>
<tr>
<td>Requirements</td>
<td>38</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>38</td>
</tr>
<tr>
<td>Notes</td>
<td>39</td>
</tr>
<tr>
<td>Related resources</td>
<td>39</td>
</tr>
<tr>
<td>Dynatrace</td>
<td>40</td>
</tr>
<tr>
<td>Requirements</td>
<td>40</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>40</td>
</tr>
<tr>
<td>Notes</td>
<td>41</td>
</tr>
<tr>
<td>Related resources</td>
<td>41</td>
</tr>
<tr>
<td>Facebook Ads</td>
<td>42</td>
</tr>
<tr>
<td>Facebook Ads support</td>
<td>42</td>
</tr>
<tr>
<td>Before you begin</td>
<td>42</td>
</tr>
<tr>
<td>Connecting to the Marketing API</td>
<td>43</td>
</tr>
<tr>
<td>Transferring data from the Marketing API</td>
<td>44</td>
</tr>
<tr>
<td>Supported objects</td>
<td>44</td>
</tr>
<tr>
<td>Supported destinations</td>
<td>44</td>
</tr>
<tr>
<td>Google Ads</td>
<td>45</td>
</tr>
<tr>
<td>Google Ads support</td>
<td>45</td>
</tr>
<tr>
<td>Before you begin</td>
<td>45</td>
</tr>
<tr>
<td>Connecting to Google Ads</td>
<td>46</td>
</tr>
<tr>
<td>Transferring data from Google Ads</td>
<td>47</td>
</tr>
<tr>
<td>Supported objects</td>
<td>47</td>
</tr>
<tr>
<td>Supported destinations</td>
<td>47</td>
</tr>
<tr>
<td>Google Analytics</td>
<td>48</td>
</tr>
<tr>
<td>Requirements</td>
<td>48</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>48</td>
</tr>
<tr>
<td>Notes</td>
<td>49</td>
</tr>
<tr>
<td>Related resources</td>
<td>51</td>
</tr>
<tr>
<td>Infor Nexus</td>
<td>51</td>
</tr>
<tr>
<td>Requirements</td>
<td>51</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>51</td>
</tr>
<tr>
<td>Notes</td>
<td>52</td>
</tr>
<tr>
<td>Jira Cloud</td>
<td>52</td>
</tr>
<tr>
<td>Jira Cloud support</td>
<td>53</td>
</tr>
<tr>
<td>Before you begin</td>
<td>53</td>
</tr>
<tr>
<td>Connecting to Jira Cloud</td>
<td>55</td>
</tr>
<tr>
<td>Transferring data from Jira Cloud</td>
<td>55</td>
</tr>
<tr>
<td>Supported objects</td>
<td>56</td>
</tr>
<tr>
<td>Supported destinations</td>
<td>57</td>
</tr>
<tr>
<td>Marketo</td>
<td>57</td>
</tr>
<tr>
<td>Requirements</td>
<td>58</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>58</td>
</tr>
<tr>
<td>Notes</td>
<td>59</td>
</tr>
<tr>
<td>Related resources</td>
<td>60</td>
</tr>
<tr>
<td>Mixpanel</td>
<td>60</td>
</tr>
<tr>
<td>Mixpanel support</td>
<td>60</td>
</tr>
<tr>
<td>Before you begin</td>
<td>60</td>
</tr>
<tr>
<td>Connecting to Mixpanel</td>
<td>61</td>
</tr>
<tr>
<td>Transferring Data From Mixpanel</td>
<td>61</td>
</tr>
<tr>
<td>Supported objects</td>
<td>62</td>
</tr>
<tr>
<td>Supported destinations</td>
<td>64</td>
</tr>
<tr>
<td>Salesforce</td>
<td>65</td>
</tr>
<tr>
<td>Requirements</td>
<td>65</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>65</td>
</tr>
<tr>
<td>Additional flow settings for Salesforce</td>
<td>69</td>
</tr>
<tr>
<td>Notes</td>
<td>72</td>
</tr>
<tr>
<td>Related resources</td>
<td>72</td>
</tr>
<tr>
<td>Salesforce Marketing Cloud</td>
<td>73</td>
</tr>
<tr>
<td>Salesforce Marketing Cloud support</td>
<td>73</td>
</tr>
<tr>
<td>Before you begin</td>
<td>73</td>
</tr>
<tr>
<td>Connecting to Salesforce Marketing Cloud</td>
<td>74</td>
</tr>
<tr>
<td>Transferring data from Salesforce Marketing Cloud</td>
<td>75</td>
</tr>
<tr>
<td>Supported objects</td>
<td>75</td>
</tr>
<tr>
<td>Supported destinations</td>
<td>76</td>
</tr>
<tr>
<td>Salesforce Pardot</td>
<td>76</td>
</tr>
<tr>
<td>Requirements</td>
<td>76</td>
</tr>
<tr>
<td>Setup instructions</td>
<td>77</td>
</tr>
<tr>
<td>Notes</td>
<td>77</td>
</tr>
<tr>
<td>Related resources</td>
<td>78</td>
</tr>
<tr>
<td>SAP OData</td>
<td>78</td>
</tr>
<tr>
<td>SAP OData support</td>
<td>78</td>
</tr>
<tr>
<td>Before you begin</td>
<td>78</td>
</tr>
<tr>
<td>Connecting to SAP</td>
<td>80</td>
</tr>
<tr>
<td>Transferring data from SAP OData</td>
<td>83</td>
</tr>
<tr>
<td>Advanced capabilities</td>
<td>84</td>
</tr>
<tr>
<td>Notes</td>
<td>86</td>
</tr>
<tr>
<td>Related resources</td>
<td>86</td>
</tr>
<tr>
<td>ServiceNow</td>
<td>86</td>
</tr>
<tr>
<td>Requirements</td>
<td>87</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>87</td>
</tr>
<tr>
<td>Notes</td>
<td>88</td>
</tr>
<tr>
<td>Related resources</td>
<td>89</td>
</tr>
<tr>
<td>Singular</td>
<td>89</td>
</tr>
<tr>
<td>Requirements</td>
<td>89</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>89</td>
</tr>
<tr>
<td>Notes</td>
<td>90</td>
</tr>
<tr>
<td>Related resources</td>
<td>90</td>
</tr>
<tr>
<td>Slack</td>
<td>90</td>
</tr>
<tr>
<td>Requirements</td>
<td>91</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>91</td>
</tr>
<tr>
<td>Notes</td>
<td>92</td>
</tr>
<tr>
<td>Related resources</td>
<td>92</td>
</tr>
<tr>
<td>Snowflake</td>
<td>93</td>
</tr>
<tr>
<td>Requirements</td>
<td>93</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>93</td>
</tr>
<tr>
<td>Related resources</td>
<td>95</td>
</tr>
<tr>
<td>Trend Micro</td>
<td>95</td>
</tr>
<tr>
<td>Requirements</td>
<td>95</td>
</tr>
<tr>
<td>Related resources</td>
<td>95</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>95</td>
</tr>
<tr>
<td>Notes</td>
<td>96</td>
</tr>
<tr>
<td>Related resources</td>
<td>96</td>
</tr>
<tr>
<td>Upsolver</td>
<td>96</td>
</tr>
<tr>
<td>Requirements</td>
<td>97</td>
</tr>
<tr>
<td>Setup instructions</td>
<td>97</td>
</tr>
<tr>
<td>Notes</td>
<td>97</td>
</tr>
<tr>
<td>Related resources</td>
<td>98</td>
</tr>
<tr>
<td>Veeva</td>
<td>98</td>
</tr>
<tr>
<td>Requirements</td>
<td>98</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>98</td>
</tr>
<tr>
<td>Extract Veeva VAULT documents with Amazon AppFlow</td>
<td>99</td>
</tr>
<tr>
<td>Notes</td>
<td>101</td>
</tr>
<tr>
<td>Related resources</td>
<td>101</td>
</tr>
<tr>
<td>Zendesk</td>
<td>101</td>
</tr>
<tr>
<td>Requirements</td>
<td>101</td>
</tr>
<tr>
<td>Connection instructions</td>
<td>102</td>
</tr>
<tr>
<td>Notes</td>
<td>103</td>
</tr>
<tr>
<td>Related resources</td>
<td>104</td>
</tr>
<tr>
<td>Zendesk Chat</td>
<td>104</td>
</tr>
<tr>
<td>Zendesk Chat support</td>
<td>104</td>
</tr>
<tr>
<td>Before you begin</td>
<td>105</td>
</tr>
<tr>
<td>Connecting to Zendesk Chat</td>
<td>105</td>
</tr>
<tr>
<td>Transferring data from Zendesk Chat</td>
<td>106</td>
</tr>
<tr>
<td>Supported objects</td>
<td>106</td>
</tr>
<tr>
<td>Supported destinations</td>
<td>107</td>
</tr>
<tr>
<td>Zendesk Sell</td>
<td>107</td>
</tr>
<tr>
<td>Zendesk Sell support</td>
<td>107</td>
</tr>
<tr>
<td>Before you begin</td>
<td>107</td>
</tr>
<tr>
<td>Connecting to Zendesk Sell</td>
<td>108</td>
</tr>
<tr>
<td>Transferring data from Zendesk Sell</td>
<td>109</td>
</tr>
<tr>
<td>Supported objects</td>
<td>109</td>
</tr>
<tr>
<td>Supported destinations</td>
<td>109</td>
</tr>
<tr>
<td>Managing connections</td>
<td>110</td>
</tr>
<tr>
<td>Flows</td>
<td>114</td>
</tr>
<tr>
<td>Creating flows</td>
<td>114</td>
</tr>
<tr>
<td>Create a flow using the AWS console</td>
<td>115</td>
</tr>
<tr>
<td>Create a flow using the AWS CLI</td>
<td>117</td>
</tr>
<tr>
<td>Create a flow using the Amazon AppFlow APIs</td>
<td>119</td>
</tr>
<tr>
<td>Create a flow using CloudFormation resources</td>
<td>121</td>
</tr>
<tr>
<td>Cataloging flow output</td>
<td>125</td>
</tr>
<tr>
<td>Before you begin</td>
<td>126</td>
</tr>
<tr>
<td>Cataloging flow output (Amazon AppFlow console)</td>
<td>126</td>
</tr>
<tr>
<td>Data Catalog table names</td>
<td>127</td>
</tr>
<tr>
<td>Example Data Catalog output from a flow run</td>
<td>128</td>
</tr>
<tr>
<td>Partitioning and aggregating</td>
<td>131</td>
</tr>
<tr>
<td>Partitioning and aggregating flow output (Amazon AppFlow console)</td>
<td>131</td>
</tr>
<tr>
<td>Example file paths for partitioned datasets</td>
<td>133</td>
</tr>
<tr>
<td>Activate a flow</td>
<td>135</td>
</tr>
<tr>
<td>Edit a flow</td>
<td>135</td>
</tr>
<tr>
<td>Delete a flow</td>
<td>136</td>
</tr>
<tr>
<td>Flow triggers</td>
<td>136</td>
</tr>
<tr>
<td>On demand</td>
<td>136</td>
</tr>
<tr>
<td>Event-triggered</td>
<td>136</td>
</tr>
<tr>
<td>Schedule-triggered</td>
<td>136</td>
</tr>
<tr>
<td>Private flows</td>
<td>138</td>
</tr>
<tr>
<td>Flow notifications</td>
<td>138</td>
</tr>
</tbody>
</table>
Audience ................................................................. 145
Flow event detail fields .............................................. 140
Common fields .................................................................. 139
Security ........................................................................ 142

Data protection .............................................................. 142
Encryption at Rest ............................................................ 143
Encryption in Transit ......................................................... 143
Key Management ............................................................... 143
Connection credentials ..................................................... 143
Identity and access management ....................................... 144

Audience ........................................................................ 145
Managing access using policies ........................................ 147
How Amazon AppFlow works with IAM ......................... 149
Managing user permissions .............................................. 154
Identity-based policy examples ........................................ 156
Service role policies ......................................................... 162
Amazon S3 Bucket Policies for Amazon AppFlow .............. 166
AWS managed policies ..................................................... 169
Troubleshooting ............................................................. 173

Compliance validation .................................................... 174
Resilience ..................................................................... 175
Infrastructure security ..................................................... 175
Quotas ........................................................................... 176

CloudTrail logs ............................................................... 178
Amazon AppFlow information in CloudTrail ..................... 178
Understanding Amazon AppFlow log file entries ............... 179
Monitoring with CloudWatch .......................................... 180
Document history ........................................................... 181
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. The following diagram illustrates how it works:

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.
- **Catalog your data for search and discovery** — Catalog the data that you transfer to Amazon S3 in the AWS Glue Data Catalog. When you catalog your data, you make it easier to discover and access with AWS analytics and machine learning services.
- **Organize transferred data into partitions and files** — Use partition and aggregation settings to optimize query performance for applications that access the data that you transfer.
- **Develop custom connectors** — Use the Amazon AppFlow Custom Connector SDKs to build connectors for data sources that aren't already integrated with the service. With custom connectors, you can transfer data between private APIs, on-premise systems, other cloud services, and AWS. The SDKs are available on GitHub:
  - Amazon AppFlow Custom Connector SDK (Python)
  - Amazon AppFlow Custom Connector SDK (Java)

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

Use cases

Following are some example uses cases that illustrate the benefits of using Amazon AppFlow.

Transfer Salesforce opportunities to Amazon Redshift tables
Create a flow triggered on each new record created in Salesforce Cloud that calculates the sales potential and then transfers the modified record to an Amazon Redshift table.

**Analyze Slack conversations**

Create a flow triggered on a schedule that transfers conversation data from a Slack channel to Amazon Redshift, Snowflake, or Amazon S3 for storage and analysis.

**Transfer support tickets from Zendesk for storage and analysis**

Create a manually triggered flow for all tickets with a common case number in Zendesk that transfers ticket data to Amazon Redshift, Snowflake, or Amazon S3 for storage and analysis.

**Transfer aggregate data weekly to S3 at 100GB per flow**

Create a flow triggered on a weekly schedule to transfer Salesforce, Marketo, ServiceNow, and Zendesk data to Amazon S3 in aggregate up to 100GB per flow with low latency.

---

**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 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.
Overview of Amazon AppFlow source and destination connectors

Amazon AppFlow is a bi-directional data transfer service; however, not all source-destination combinations are currently supported.

Following is the compatibility matrix for Amazon AppFlow connectors for the **us-east-1** region.

**Supported Amazon AppFlow Connectors**

<table>
<thead>
<tr>
<th>Source</th>
<th>Marketo</th>
<th>Salesforce</th>
<th>Snowflake</th>
<th>Upsolve</th>
<th>Zendesk</th>
<th>Redshift</th>
<th>S3</th>
<th>EventBridge</th>
<th>Lookout for Metrics</th>
<th>Customer profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplitude</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datadog</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynatrace</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Analytics</td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Infornexus</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marketo</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Pardot</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SAP OData</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salesforce</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ServiceNow</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Singular</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slack</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrendMicro</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Veeva</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zendesk</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S3</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**General information for all applications**

The following general information 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. 23).

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. 135).

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 section provides an introduction to Amazon AppFlow with prerequisites for getting started. The following diagram illustrates how you can use Amazon AppFlow to transfer and enrich data from a data source to a data destination in your flow:

Tasks

- Prerequisites (p. 6)

Prerequisites

Complete the following prerequisites before 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. 23).

- **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.

- **AWS CloudFormation OAuth (Optional)** — 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 (Optional)** — Amazon AppFlow encrypts your data and connection details during transit and at rest. For more information, see Data protection in Amazon AppFlow (p. 142). 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.
Tutorial: Transfer data between applications with Amazon AppFlow

This tutorial explains how to use Amazon AppFlow with Amazon Simple Storage Service (Amazon S3) and Salesforce through the AWS Management Console. Optionally, if you want to use a different supported software as a service (SaaS) application, the tutorial provides general instructions for how to create a flow. A flow uses a connection to transfer data between a source and a destination. When you run a flow, 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.

Objective

In this tutorial, you learn to transfer data between applications. Specifically, you transfer data both from Amazon S3 to Salesforce, and from Salesforce to Amazon S3. First, you synchronize additional account records with the customer relationship management (CRM) data already stored in Salesforce (Flow 1). You can optionally add validations to this flow to only transfer good data. Then, you transfer the account data in Salesforce to Amazon S3 in an event-triggered flow (Flow 2). When Amazon AppFlow detects a change to the target data in the CRM storage service, an event-triggered flow runs. This way, you have access to up-to-date information in Amazon S3, where you can import it into an object for data lake hydration to generate business value.

In this tutorial, you accomplish the following:

- Store a sample data set of accounts in Amazon Simple Storage Service (Amazon S3).
- Flow 1 — Use Amazon AppFlow to transfer data from Amazon S3 to Salesforce.
- Flow 2 — Use Amazon AppFlow to transfer data from Salesforce to Amazon S3.

The following diagram shows the two workflows.
Prerequisites

Before you begin, you need access to an AWS account and an account for a supported application. This tutorial uses Salesforce, but you can follow the steps to create flows with a different application. If you access AWS as an IAM user, your administrator must grant you the permissions required to complete this tutorial.

- **Amazon AppFlow setup** — If you haven’t already done so, complete the Getting started prerequisites (p. 6).
- **AWS Identity and Access Management (IAM) user setup** — You or your administrator must attach the AWS managed policy AmazonAppFlowFullAccess to your IAM group or user. For information on

Estimated cost: Some of the actions in this tutorial may incur minor charges on your AWS account. The provided sample data is 1 KB. Should you choose to use your own data, you might incur greater charges. Reduce charges by completing the tutorial through Step 5: Clean up your resources (p. 21). For information about pricing, see Amazon S3 pricing and Amazon AppFlow pricing.

Topics

- Prerequisites (p. 9)
- Step 1: Upload data to Amazon S3 (p. 10)
- Step 2: Connect Amazon AppFlow to an application (p. 12)
- Step 3: Transfer data from Amazon S3 to a SaaS destination (p. 13)
- Step 4: Transfer data from a SaaS source to Amazon S3 (p. 17)
- Step 5: Clean up your resources (p. 21)
Step 1: Upload data to Amazon S3

Suppose you have data that you want to turn into Salesforce account records. You acquired this data from a web form and used it to generate account records. You can upload this list of additional accounts to Amazon Simple Storage Service (Amazon S3). Amazon AppFlow can transfer the data from Amazon S3 to Salesforce to synchronize your customer relationship management (CRM) data.
To use Amazon S3 as your source for the flow, create a storage container, called a bucket, and populate it with data. Amazon AppFlow can transfer the data within an S3 bucket to any of the supported destinations. In this step, you create an S3 bucket, create a source folder within the S3 bucket, and upload sample data to the source folder.

Topics
- (Optional) Download sample data (p. 11)
- Create an S3 bucket (p. 11)
- Create a folder in an S3 bucket (p. 11)
- Upload data to Amazon S3 (p. 12)
- Additional resources (p. 12)

(Optional) Download sample data

If you have your own data that you want to use for this tutorial, you can skip this step. Also, if you use a SaaS application other than Salesforce, this sample data may not be useful.

The sample data includes nine account records. Download this sample data set.

To get the sample data
1. Download the zip file tutorial-account-data.zip.
2. Extract the zip file. The unzipped file called tutorial-account-data.csv contains the sample data set.

Create an S3 bucket

After you extract your sample data, use the AWS Management Console to create an S3 bucket to store your data. Your S3 bucket must occupy the same AWS Region as the one where you want to use Amazon AppFlow.

To create an S3 bucket
1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Buckets section, choose Create bucket.
3. For Bucket name, enter a descriptive name. The name must be globally unique. For example, enter username-appflow-tutorial.
4. For AWS Region, choose the same Region as your Amazon AppFlow console.
   Warning
   If your S3 bucket isn't in the same AWS Region as your console, your flow can't access it.
5. Keep the other settings at their default values. Choose Create bucket.

Create a folder in an S3 bucket

Now that you have an S3 bucket, use the console to create a folder in the bucket where you want to store the sample data. While a folder isn't essential, it's useful for keeping your files organized.

To create a folder in Amazon S3
1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the **Buckets** section, choose your S3 bucket from the list.
3. Under the **Objects** tab, choose **Create folder**.
4. For the folder name, enter **source**.
5. Choose **Create folder**.

**Upload data to Amazon S3**

Now that you have set up your S3 bucket, upload the data.

**To populate the S3 bucket with data**

1. Open the Amazon S3 console at [https://console.aws.amazon.com/s3/](https://console.aws.amazon.com/s3/).
2. In the **Buckets** section, choose your S3 bucket from the list.
3. Choose the **source** folder. Then, under the **Objects** tab, choose **Upload**.
4. Choose **Add files**, and choose your data set. If you downloaded the sample data set, choose the `tutorial-account-data.csv` file.
5. Choose **Upload**.

You now have an S3 bucket with sample data in the **source** folder.

**Additional resources**

For more information on Amazon S3, see the following resources:

- Amazon S3 in the *Amazon AppFlow User Guide*.
- Amazon S3 in the *Amazon S3 User Guide*.

**Step 2: Connect Amazon AppFlow to an application**

You can securely move your data between supported source and destination applications with a connection in Amazon AppFlow. Connections store the configuration details and credentials necessary to run flows without the need to repeatedly enter information. After you have an established connection with an application, you can use that connection in new or existing flows.

**Topics**

- Prerequisites (p. 12)
- Create a connection between Amazon AppFlow and a SaaS application (p. 13)
- Additional resources (p. 13)

**Prerequisites**

Before you begin, complete the tutorial prerequisites (p. 9).
Create a connection between Amazon AppFlow and a SaaS application

If you want to create and run a flow, you must establish a connection with the software as a service (SaaS). You can create this connection while you create the flow, or you can create the connection separately. Here, you create a connection in Amazon AppFlow before you create the flow.

To create a connection with Salesforce
2. Expand the navigation pane on the left-hand side of the console page and choose Connections.
3. For Connectors, select Salesforce.
4. Choose Create connection.
5. Leave the default selections and enter a Connection name. For example, enter my-salesforce-connection.
6. Choose Continue.
7. If you're not already logged into Salesforce, Amazon AppFlow prompts you to log in.
8. Choose Allow to give Amazon AppFlow access to your Salesforce account.

To create a connection with other applications
• Go to the Supported applications (p. 23) page and select the application that you want to connect with. Follow the instructions for your selected application.

You now have a connection in the Amazon AppFlow console to your SaaS account. If you use the same third-party application in both flows, you only need one connection.

Additional resources
For more information on connections, see the following resources:
• Managing connections in the Amazon AppFlow User Guide.
• Salesforce in the Amazon AppFlow User Guide.

Step 3: Transfer data from Amazon S3 to a SaaS destination

Amazon S3 now hosts your data, but you still need to synchronize all your records in the destination. To transfer data to a supported destination, you must create and run a flow with Amazon AppFlow. In this step, you use the AWS Management Console to send data from Amazon S3 to either Salesforce or another software as a service (SaaS) application.

Topics
• Prerequisites (p. 14)
• Create a flow (p. 14)
• Run a flow (p. 15)
• View transferred data (p. 16)
• (Optional) Edit flow to add validations (p. 16)
Prerequisites

Before you begin, complete Step 1: Upload data to Amazon S3 (p. 10).

Create a flow

The following procedures detail how to create a flow from Amazon S3 to Salesforce, but you can follow the steps with any destination.

To complete Step 1: Specify flow details

1. Open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/. Ensure the AWS Region of your Amazon AppFlow console is the same one as your S3 bucket.
2. Choose Create flow.
3. For Flow name, enter s3-to-SaaS. For example, if your destination is Salesforce, enter s3-to-salesforce.
4. Under Data encryption, you have the option to activate custom encryption settings. By default, Amazon AppFlow encrypts your data with a key in AWS Key Management Service (AWS KMS). AWS creates, uses, and manages this key for you. Amazon AppFlow always encrypts your data during transit and at rest. The default encryption is adequate for this tutorial, so don't select custom encryption settings. For more information, see Data protection in the Amazon AppFlow User Guide.
5. Under Tags, you have the option to add tags to your flow. Tags are key-value pairs that assign metadata to resources that you create. Tags are not necessary for this tutorial. For more information, see Tagging AWS resources in the AWS General Reference.
6. To continue to Step 2: Configure flow, choose Next.

To complete Step 2: Configure flow

1. For Source name, choose Amazon S3.
2. In Bucket details, for Choose an S3 bucket, select your S3 bucket.
3. For Enter bucket prefix, enter source. Bucket prefixes are folders.
4. Ensure Data format preference is CSV format.
5. Configure the Destination details. These details vary based on the destination that you want to transfer data to.
   - If you want to transfer data to Salesforce, do the following:
     a. For Destination name, select Salesforce.
     b. For Choose Salesforce connection, select your connection. For example, select my-salesforce-connection, the connection that you created in the previous step.
        Tip
        If you don't have a connection, you can choose Connect to create one now.
     c. If you want to use the sample data that you downloaded, for Choose Salesforce object, select Account.
   - If you want to transfer data to another supported application besides Salesforce, do the following:
     a. For Destination name, select the destination that you want for your data.
     b. For Choose connection, select the connection that you created, or create one.
     c. Select object and specify the correct object type for your data.
     d. If there are any other destination details, configure the required fields.
6. In the **Error handling** section, you can specify how you want the flow to handle errors and where to put the data that causes errors. For this tutorial, you can leave the settings in this section at their default values.

7. For **Flow trigger**, leave the default selection **Run on demand**. When you select this value, you use a single button in the console to run the flow.

   **Tip**
   You can also run flows on a schedule. Amazon AppFlow bases the time zone for this schedule on your web browser. For more information, see **Schedule-triggered flows** in the *Amazon AppFlow User Guide*.

8. To continue to Step 3: Map data fields, choose **Next**.

**To complete Step 3: Map data fields**

1. Map your data fields. These vary based on the destination for your data transfer.
   - If you're transferring to Salesforce, do the following:
     a. Under **Mapping method**, leave the default selection **Manually map fields**.
     b. Under **Destination record preference**, leave the default selection **Insert new records**.
     c. In the **Source to destination field mapping** section, select the **Choose source fields** dropdown and select **Map all fields directly**.
     
     **Important**
     If you use the sample data, ensure Account Name maps to Account Name, Account Type maps to Account Type, Billing State/Province maps to Billing State/Province, Account Rating maps to Account Rating, and Industry maps to Industry.
     d. Choose **Map selected fields**.
   - If you want to transfer data to another supported application besides Salesforce, do the following:
     a. Select **Mapping method** and specify how you want to map your data. You can choose to map the source fields to the destination fields manually, or else upload a .csv file that includes these mappings.
     b. Map your fields from the source field name to the destination field name.

2. Under **Validations**, specify what happens to invalid data within the flow. For this step, you don't need any validations.

3. To continue to Step 4: Add filters, choose **Next**.

**To complete Step 4: Add filters**

1. Under **Filters**, specify what data the flow transfers. With this setting, you can ensure the flow transfers data only when it meets certain criteria. For this tutorial, you don't need any filters.

2. To continue to Step 5: Review and create, choose **Next**.

**To complete Step 5: Review and create**

- Review the flow settings, and then choose **Create flow**.

**Run a flow**

You now have a run-on-demand flow. When you choose the **Run flow** button in the console, this flow transfers your data.
To run a flow
1. In Flows, select your flow from the list.
2. Choose Run flow.

When the flow successfully runs, a banner appears. If you use the provided data, the banner shows nine processed records.

View transferred data

After your flow runs, you can view the data in the destination.

To view transferred data

• If you use the sample Salesforce account data, navigate to your Salesforce Account tab to view the imported account records. For more information on Salesforce accounts, see Salesforce accounts.

You have now transferred data from Amazon S3 to Salesforce or the SaaS application that you chose. If you used Salesforce and the sample data, you have synchronized and expanded your Salesforce account data.

(Optional) Edit flow to add validations

The flow that you ran transferred all the records in the data set. You can add validations to a flow so that you transfer only valid records. In this procedure, if you use the sample data, you edit your Amazon S3 to Salesforce flow to transfer only account records with ratings.

Before you edit and run the flow again, delete the records that you transferred from the original flow.

To delete account records in Salesforce

• Follow the directions in Mass Delete Records.

For the sample data set, suppose you consider account records valid only if they have an account rating. Two of the account records don't have associated account ratings. You don't want these records to transfer from Amazon S3 so that you only have valid data in Salesforce.

To edit a flow and add validations
2. In Flows, choose your flow.
3. Choose Actions, then choose Edit flow.
5. In Validations, choose Add validation.
6. If you use the sample data, for Field name, select Account rating. For Condition, choose Values missing or null. For Action, choose Ignore record. This configuration will omit the transfer of account records with missing rating values.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Account Name</td>
<td>Account Type</td>
<td>Billing State/Province</td>
<td>Account Rating</td>
</tr>
<tr>
<td>2</td>
<td>Example1</td>
<td>Customer - Direct</td>
<td>Anywhere</td>
<td>Hot</td>
</tr>
<tr>
<td>3</td>
<td>Example2</td>
<td>Customer - Channel</td>
<td>Anywhere</td>
<td>Warm</td>
</tr>
<tr>
<td>4</td>
<td>Example3</td>
<td>Anywhere</td>
<td>Cold</td>
<td>Construction</td>
</tr>
<tr>
<td>5</td>
<td>Example4</td>
<td>Customer - Direct</td>
<td>Anywhere</td>
<td>Consulting</td>
</tr>
<tr>
<td>6</td>
<td>Example5</td>
<td>Customer - Channel</td>
<td>Anywhere</td>
<td>Hot</td>
</tr>
<tr>
<td>7</td>
<td>Example6</td>
<td>Customer - Channel</td>
<td>Anywhere</td>
<td>Warm</td>
</tr>
<tr>
<td>8</td>
<td>Example7</td>
<td>Anywhere</td>
<td>Cold</td>
<td>Energy</td>
</tr>
<tr>
<td>9</td>
<td>Example8</td>
<td>Anywhere</td>
<td>Hospitality</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Example9</td>
<td>Customer - Direct</td>
<td>Anywhere</td>
<td>Hot</td>
</tr>
</tbody>
</table>

7. Choose Save.

To run the edited flow and view transferred data

1. In Flows, select your flow from the list.
2. Choose Run flow. When the flow successfully runs, a banner appears.

3. If you use the sample Salesforce account data, navigate to your Salesforce Account tab to view the imported account records. For more information on Salesforce accounts, see Salesforce Accounts.

If you used the sample data, only seven of the nine records transferred. Example4 and Example8 do not appear because they have no account ratings associated with them.

Step 4: Transfer data from a SaaS source to Amazon S3

Suppose you now want to transfer your data from Salesforce to Amazon S3. With Amazon S3, you can synchronize and replicate customer relationship management (CRM) data into data lakes to analyze or use to drive machine learning. To keep this information up to date, you can create an event-triggered flow from Salesforce to Amazon S3. An event-triggered flow runs when Amazon AppFlow detects a change to the target data in the CRM storage service.

After you create an S3 bucket, you can set up and run a flow with Amazon AppFlow to transfer data from a supported source to the S3 bucket. You can use one S3 bucket as both a source and destination, so you don’t need to create a new S3 bucket if you already created one for this tutorial. In this step, you use the AWS Management Console to create and run a flow from Salesforce or another software as a service (SaaS) application to Amazon S3.

Topics
- Prerequisites (p. 18)
Prerequisites

Before you begin, you need an S3 bucket to receive the data if you don’t already have one. You can use the same S3 bucket as both a source and destination for different flows. This tutorial uses Salesforce for a SaaS account, but you can use another supported source application if you want. Some flow options that this tutorial uses don’t work for a SaaS application other than Salesforce.

- **Amazon S3 setup** — If you don’t already have an S3 bucket, [Create an S3 bucket](p. 11) to prepare Amazon S3 to receive your data.
- **Salesforce setup (Optional)** — If you already have a Salesforce account, or you want to complete this tutorial with a different SaaS application, you can skip this step. Sign up for a free Salesforce developer account [here](#).
- **Transfer data to Salesforce (Optional)** — If you use Salesforce for this tutorial, we recommend that you complete [Step 3: Transfer data from Amazon S3](p. 13) before you continue.

Change data capture in Salesforce

To run an event-triggered flow, Amazon AppFlow needs to receive a notification when a record changes. When you use the change data capture feature in Salesforce, you can generate change event notifications for selected entities. If you don’t have administrator-level credentials, you might not be able to select entities to generate change notifications. However, the free developer account has administrator privileges.

**To enable change data capture**

1. Open Salesforce at [www.salesforce.com](http://www.salesforce.com) and log in to your account.
2. Navigate to the Change Data Capture page.
3. If you use the sample data, select **Account (Account)** to generate change event notifications. Otherwise, select the appropriate entity for your data.

For more information about Salesforce change data capture, see [Change Data Capture](#).

Create a flow

The following procedures detail how to create a flow from Salesforce to Amazon S3, but you can follow the steps with any supported source. Some flow options that this tutorial uses don’t work for a SaaS application other than Salesforce, but alternate steps appear.

**To complete Step 1: Specify flow details**

2. Choose **Create flow**.
3. For **Flow name**, enter **SaaS-to-s3**. For example, if your source is Salesforce, enter **salesforce-to-s3**.
4. Under **Data encryption**, you have the option to activate custom encryption settings. By default, Amazon AppFlow encrypts your data with a key in AWS Key Management Service (AWS KMS). AWS
creates, uses, and manages this key for you. Amazon AppFlow always encrypts your data during transit and at rest. The default encryption is adequate for this tutorial, so don’t select custom encryption settings. For more information, see Data protection in the Amazon AppFlow User Guide.

5. Under Tags, you have the option to add tags to your flow. Tags are key-value pairs that assign metadata to resources that you create. Tags aren’t necessary for this tutorial. For more information, see Tagging AWS resources in the AWS General Reference.

6. To continue to Step 2: Configure flow, choose Next.

To complete Step 2: Configure flow

1. Configure the Source details. These details vary based on the source that you want to transfer data from.
   • If you want to transfer data from Salesforce, do the following:
     a. For Source name, choose Salesforce.
     b. For Choose Salesforce connection, select your connection. For example, select my-salesforce-connection, the connection that you created in a previous step.
        Tip
        If you don’t have a connection, you can choose Connect to create one now.
     c. Select Salesforce events.
     d. If you use the sample data, for Choose Salesforce event, select Account Change Event. Otherwise, select the event that matches your data.
   • If you want to transfer data from another supported application besides Salesforce, do the following:
     a. For Source name, select the source that you want for your data.
     b. For Choose connection, select the connection that you created, or create one.
     c. Select object and specify the correct object type for your data.
     d. If there are any other source details, configure the required fields.

2. For Destination name, choose Amazon S3.

3. In Bucket details, for Choose an S3 bucket, select your S3 bucket. Use the same S3 bucket that contains the source folder from the previous step.

4. For Enter bucket prefix, enter destination. Bucket prefixes are folders.
   Tip
   If you don’t have a folder that matches the name that you entered, the flow automatically creates one when it runs.

5. Configure the Flow trigger. This varies based on the source where you want to transfer data from.
   • If you want to transfer data from Salesforce, leave the default selection Run flow on event.
   • If you want to transfer data from another supported application besides Salesforce, leave the default selection Run on demand. This option allows you to run the flow with the selection of one button in the console.
     Tip
     You can also run flows on a schedule. Amazon AppFlow bases the time zone for this schedule on your web browser. For more information, see Schedule-triggered flows in the Amazon AppFlow User Guide.

6. To continue to Step 3: Map data fields, choose Next.

To complete Step 3: Map data fields

2. In the **Source to destination field mapping** section, select the **Choose source fields** dropdown and select **Map all fields directly**.

3. Under **Validations**, specify what happens to invalid data within the flow. For this step, you don't need any validations.

4. To continue to Step 4: Add filters, choose **Next**.

**To complete Step 4: Add filters**

1. Under **Filters**, specify what data the flow transfers. With this setting, you can ensure the flow transfers data only when it meets certain criteria. For this tutorial, you don't need any filters.

2. To continue to Step 5: Review and create, choose **Next**.

**To complete Step 5: Review and create**

- Review the flow settings, then choose **Create flow**.

**Run a flow**

You now have a flow. The source that you use determines how you run this flow.

**Run an event-triggered flow with Salesforce**

Your event-triggered flow runs when a change occurs to a record that you've set up to generate change event notifications. Here, you change a record within your Salesforce account to activate a flow run.

**To run an event-triggered flow with Salesforce**


2. In **Flows**, select the **salesforce-to-s3** flow.

3. Choose **Activate flow**.

4. Open Salesforce at [www.salesforce.com](http://www.salesforce.com) and log in to your account.

5. Navigate to the page where Salesforce stores your records. For the sample data, this is the **Accounts** page.

6. Edit one of the records. For example, in the sample data, change the **Rating** in **Example3** from **cold** to **hot**.

After about a minute, refresh your flow page in Amazon AppFlow. When the flow successfully runs, a timestamp from the last flow run appears.

**Run an on-demand flow with a supported SaaS source**

Your on-demand flow runs when you choose the **Run flow** button in the console.
To run an on-demand flow

1. In Flows, select your flow from the list.
2. Choose Run flow.

When the flow successfully runs, a banner appears.

View transferred data

The data from your source now resides in your S3 bucket. From the S3 bucket, you can, for example, consume the data from multiple AWS services for analysis. In this step, you download and view the data on your computer.

To retrieve the transferred data

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In Buckets, choose your S3 bucket from the list.
3. In your S3 bucket, choose the destination folder. Then choose the flow folder, for example, salesforce-to-s3.
4. The folder contains one file. Select this file and choose Download.
5. Navigate to the file in your Downloads folder and rename it with a descriptive name.
6. Open the file to view the updated record.

You've now transferred data from Salesforce or the SaaS that you chose to Amazon S3. If you used Salesforce, you set up an event-triggered flow to keep up-to-date with changing data.

Step 5: Clean up your resources

After you've completed the tutorial, it's good practice to clean up any resources that you no longer want to use. This way, your account doesn't incur any further charges.

Topics
- Clean up in Amazon S3 (p. 21)
- Clean up in Amazon AppFlow (p. 22)
- Clean up in Salesforce (p. 22)

Clean up in Amazon S3

Because you used an S3 bucket as both a source and a destination throughout this tutorial, Amazon S3 hosted multiple files. Unless you delete these files, their storage continues to incur charges on your AWS account. Before you delete an S3 bucket, ensure you have saved any important files to another location.

To clean up your S3 bucket

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the **Buckets** section, select your S3 bucket and choose **Empty**. Follow the prompts to delete the contents of the bucket.

3. In the **Buckets** section, select your S3 bucket and choose **Delete**. Follow the prompts to delete the S3 bucket.

   **Warning**
   Because S3 bucket names are globally unique, when you delete your S3 bucket, someone else can use its name. If you want to reserve an S3 bucket name, don't delete the bucket.

Now you have deleted all of the Amazon S3 resources that you created for the tutorial.

For more information on how to empty and delete S3 buckets, see the following resources:

- [Emptying a bucket](https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-deleting-objects.html) in the **Amazon S3 User Guide**.
- [Deleting a bucket](https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-deleting-buckets.html) in the **Amazon S3 User Guide**.

### Clean up in Amazon AppFlow

Amazon AppFlow stores both your connection and flows. To clean up all resources that you created in this tutorial, delete the two flows and your connection to the SaaS application.

#### To clean up your flows

2. In the **Flows** section, select a flow and choose **Delete**. Follow the prompts to delete your flow.
3. Perform the above step for any flows that remain.

#### To clean up your connection

2. In the **Connections** section, under **Connectors**, open the **Choose a connector** box. Select the connector that you used in the tutorial.
3. Select the connection and choose **Delete**.
4. If you used more than one connector, repeat steps 2 and 3 for all connectors.

Now you have deleted all of the resources that you created within Amazon AppFlow for the tutorial.

### Clean up in Salesforce

If you used Salesforce for this tutorial and uploaded the sample data from an S3 bucket to Salesforce, you might want to delete the sample account records.

#### To delete imported records in Salesforce

- Follow the directions in [Mass delete records](https://docs.salesforce.com/itd/exchange-developer/soap/mass-del-of-rec.htm).

After you complete these steps, you have cleaned up all of the resources that you created in this tutorial. Deleted resources no longer incur charges on your AWS account.
Supported source and destination applications

Choose an application in the following list to learn more about its setup requirements.

Topics
- Amazon Connect connector for Amazon AppFlow (p. 23)
- Amazon EventBridge (p. 24)
- Amazon Honeycode (p. 26)
- Amazon Lookout for Metrics (p. 27)
- Amazon Redshift connector for Amazon AppFlow (p. 29)
- Amazon S3 (p. 34)
- Amplitude (p. 36)
- Datadog (p. 38)
- Dynatrace (p. 40)
- Facebook Ads connector for Amazon AppFlow (p. 42)
- Google Ads connector for Amazon AppFlow (p. 45)
- Google Analytics (p. 48)
- Infor Nexus (p. 51)
- Jira Cloud connector for Amazon AppFlow (p. 52)
- Marketo (p. 57)
- Mixpanel connector for Amazon AppFlow (p. 60)
- Salesforce (p. 65)
- Salesforce Marketing Cloud connector for Amazon AppFlow (p. 73)
- Salesforce Pardot (p. 76)
- SAP OData connector for Amazon AppFlow (p. 78)
- ServiceNow (p. 86)
- Singular (p. 89)
- Slack (p. 90)
- Snowflake (p. 93)
- Trend Micro (p. 95)
- Upsolver (p. 96)
- Veeva (p. 98)
- Zendesk (p. 101)
- Zendesk Chat connector for Amazon AppFlow (p. 104)
- Zendesk Sell connector for Amazon AppFlow (p. 107)

Amazon Connect connector for Amazon AppFlow

Amazon Connect is an AWS service that you can use to set up an omnichannel, cloud-based contact center for your customers. Amazon Connect provides the Customer Profiles feature. This feature helps...
you create unified customer profiles. These profiles combine customer information from external applications with contact history from Amazon Connect. For example, you can combine contact information, order history, and interaction history from software as a service (SaaS) applications like Salesforce, Zendesk and other Amazon AppFlow connectors. The contact center agents for your organization can use this consolidated information during customer support interactions.

If you use Amazon Connect, you can also use Amazon AppFlow to transfer data from supported data sources to Customer Profiles.

For more information about Customer Profiles, see Use Amazon Connect Customer Profiles in the Amazon Connect Administrator Guide

Amazon Connect support

Amazon AppFlow supports Amazon Connect as follows.

Supported as a data source?

No. You can't use Amazon AppFlow to transfer data from Amazon Connect.

Supported as a data destination?

Yes. You can use Amazon AppFlow to transfer data to Amazon Connect.

Supported Amazon Connect features

Amazon AppFlow integrates only with the Customer Profiles feature.

Transferring data to Amazon Connect with a flow

To transfer data to Amazon Connect Customer Profiles, you create an Amazon AppFlow flow, and you choose Amazon Connect as the data destination. Then, you use Amazon Connect to set up data mappings in Customer Profiles. These mappings define how data from the data source is mapped to the customer profile.

Before you can use Amazon AppFlow to transfer data to Customer Profiles, you must meet these requirements:

- You have an Amazon Connect instance.
- You have enabled the Customer Profiles feature for your Amazon Connect instance. When you enable Customer Profiles, you create a customer profiles domain, which is the container for your customer data in Amazon Connect.
- You have configured Customer Profiles to encrypt your data under a KMS key.

For more information about creating a flow in Amazon AppFlow and setting up data mappings in Amazon Connect, see Set up integration for external applications using Amazon AppFlow in the Amazon Connect Administrator Guide.

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.
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. Sign in to the AWS Management Console and 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.
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 Amazon AppFlow destination in the AWS US West (Oregon) Region.

Topics

- Connection instructions (p. 26)
- Notes (p. 27)
- Related resources (p. 27)

Connection instructions

To create a flow with Amazon Honeycode as the destination

1. Sign in to the AWS Management Console and 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 (p. 28)
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. Sign in to the AWS Management Console and 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.


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

Amazon Redshift connector for Amazon AppFlow

Amazon Redshift is a data warehouse service in AWS. If you use Amazon Redshift, you can also use Amazon AppFlow to transfer data from supported sources into your Amazon Redshift databases. When you connect Amazon AppFlow to Amazon Redshift with the recommended settings, Amazon AppFlow transfers your data by using the Amazon Redshift Data API.

For more information about Amazon Redshift, see the Amazon Redshift Management Guide.

Amazon Redshift support

Amazon AppFlow supports Amazon Redshift as follows.

Supported as a data source?

No. You can't use Amazon AppFlow to transfer data from Amazon Redshift.
Supported as a data destination?

Yes. You can use Amazon AppFlow to transfer data to Amazon Redshift.

Before you begin

Before you can use Amazon AppFlow to transfer data to Amazon Redshift, you must meet these requirements:

- You have an Amazon Redshift database. If you are new to Amazon Redshift, see the Amazon Redshift Getting Started Guide to learn about basic concepts and tasks. You specify your database in the Amazon Redshift connection settings in Amazon AppFlow.

- **Recommended**: You have an AWS Identity and Access Management (IAM) role that authorizes Amazon AppFlow to access your Amazon Redshift database through the Amazon Redshift Data API. You require this role to configure an Amazon Redshift connection with the recommended settings. For more information, and for the policies that you attach to this role, see Allow Amazon AppFlow to access Amazon Redshift databases with the Data API (p. 163).

- You have an Amazon S3 bucket that Amazon AppFlow can use as an intermediate destination when it transfers data to Amazon Redshift. You specify this bucket in the connection settings. For the steps to create a bucket, see Creating a bucket in the Amazon S3 User Guide.

- You have an IAM role that grants Amazon Redshift read-only access to Amazon S3. You specify this role in the connection settings, and you associate it with your Amazon Redshift cluster. For more information, and for the polices that you attach to this role, see Allow Amazon Redshift to access your Amazon AppFlow data in Amazon S3 (p. 165).

- In IAM, you're authorized with the required pass role permissions below.

Required pass role permissions

Before you can create an Amazon Redshift connection, you must have certain IAM permissions assigned to you as an AWS user. These permissions must allow you pass IAM roles to Amazon AppFlow and Amazon Redshift, as shown by the following example IAM policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": "iam:PassRole",
           "Resource": "arn:aws:iam::account-id:role/appflow-redshift-access-role-name",
           "Condition": {
               "StringEquals": {
                   "iam:PassedToService": ["appflow.amazonaws.com"]
               },
               "StringLike": {
                   "iam:AssociatedResourceARN": ["arn:aws:appflow:region:accountId:connectorprofile/*"]
               }
           }
       }
   ]
}
```
Before you use this example policy, replace the variable elements with the required values:

- **account-id** — Your AWS account ID.
- **appflow-redshift-access-role-name** — The name of the role that authorizes Amazon AppFlow to access your Amazon Redshift database.
- **region** — The code of the AWS Region where you use Amazon AppFlow. For example, the code for the US East (N. Virginia) Region is `us-east-1`. For the AWS Regions that Amazon AppFlow supports, and their codes, see Amazon AppFlow endpoints and quotas in the AWS General Reference.
- **redshift-s3-access-role-name** — The name of the role that grants Amazon Redshift read-only access to Amazon S3.

## Connecting Amazon AppFlow to your Amazon Redshift database

To connect Amazon AppFlow to your Amazon Redshift database, provide the required database details, S3 bucket, and IAM roles. If you haven’t yet created the required resources, see the preceding section, Before you begin (p. 30).

### To create an Amazon Redshift connection

2. In the navigation pane on the left, choose **Connections**.
3. On the **Manage connections** page, for **Connectors**, choose **Amazon Redshift**.
4. Choose **Create connection**.
5. For **Data warehouse type**, choose whether to connect to **Amazon Redshift Serverless** or an **Amazon Redshift cluster**.
6. If you chose to connect to Amazon Redshift Serverless, enter the following information:
   - **Workgroup name** — The name of your Amazon Redshift workgroup.
   - **Database name** — The name of the Amazon Redshift database that stores the data that you transfer with Amazon AppFlow.
• **Bucket details** — The Amazon S3 bucket where Amazon AppFlow writes your data as an intermediate destination. Amazon Redshift gets your data from this bucket.

• **IAM role for Amazon S3 access** — The IAM role that authorizes Amazon Redshift to get and decrypt the data from the S3 bucket.

• **IAM role for Amazon Redshift Data API access** — The IAM role that authorizes Amazon AppFlow to access your Amazon Redshift database through the Data API.

7. If you chose to connect to an Amazon Redshift cluster, do one of the following:

   • **Recommended**: Choose Data API to connect through the Amazon Redshift Data API. This option is recommended because Amazon AppFlow can use the Data API to connect to public and private Amazon Redshift clusters. Enter the following information:

     • **Cluster identifier** — The unique identifier of your Amazon Redshift cluster.

     • **Database name** — The name of the Amazon Redshift database that stores the data that you transfer with Amazon AppFlow.

     • **Bucket details** — The Amazon S3 bucket where Amazon AppFlow writes your data as an intermediate destination. Amazon Redshift gets your data from this bucket.

     • **IAM role for Amazon S3 access** — The IAM role that authorizes Amazon Redshift to get and decrypt the data from the S3 bucket.

     • **IAM role for Amazon Redshift Data API access** — The IAM role that authorizes Amazon AppFlow to access your Amazon Redshift database through the Data API.

     • **Amazon Redshift database user name** — The user name that you use to authenticate with your Amazon Redshift database.

   • **Not recommended**: Choose JDBC URL to connect through a Java Database Connectivity (JDBC) URL. For information about the settings for this option, see the Guidance for connections that use JDBC URLs (p. 32) section that follows.

   **Warning**

   We don't recommend that you choose the JDBC URL option because Amazon AppFlow can't use JDBC URLs to connect to private Amazon Redshift clusters. Amazon AppFlow will discontinue support for JDBC URLs in the near future. We strongly recommend that you configure your connection with the Data API instead.

8. Optionally, under **Data encryption**, choose **Customize encryption settings (advanced)** if you want to encrypt your data with a customer managed key in the AWS Key Management Service (AWS KMS).

   By default, Amazon AppFlow encrypts your data with a KMS key that AWS creates, uses, and manages for you. Choose this option if you want to encrypt your data with your own KMS key instead.

   Amazon AppFlow always encrypts your data during transit and at rest. For more information, see Data protection in Amazon AppFlow (p. 142).

   If you want to use a KMS key from the current AWS account, select this key under **Choose an AWS KMS key**. If you want to use a KMS key from a different AWS account, enter the Amazon Resource Name (ARN) for that key.

9. For **Connection name**, enter a name for your connection.

10. Choose **Connect**.

On the Manage connections page, your new connection appears in the Connections table. When you create a flow that uses Amazon Redshift as the data destination, you can select this connection.

**Guidance for connections that use JDBC URLs**

The following information applies only to Amazon Redshift connections that are configured with JDBC URLs. We don't recommend these types of connections because Amazon AppFlow will discontinue
support for JDBC URLs in the near future. You can refer to this section to manage existing connections that use JDBC URLs. However, for any new Amazon Redshift connections that you create, you should configure them with the Data API instead.

**JDBC requirements**

You must provide Amazon AppFlow with the following:

- 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 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.
- Ensure that your cluster is publicly accessible by going to the AWS Management Console, navigating to the Amazon Redshift console and choose CLUSTERS. Then, select the cluster that you want to modify and choose Actions > Modify Publicly > Enable. Save your changes.
- If you still can't connect to the cluster from the internet or a different network, go to the Amazon Redshift console and select the cluster that you want to modify. Under Properties, choose Network and security settings. Choose the link next to VPC security group to open the Amazon Elastic Compute Cloud (Amazon EC2) console. On the Inbound Rules tab, make sure that your IP address and the port of your Amazon Redshift cluster are allowed. The default port for Amazon Redshift is 5439, but your port might be different.
- Ensure that your Amazon Redshift cluster is accessible from Amazon AppFlow IP address ranges in your Region.

**JDBC settings**

- **JDBC URL** — The JDBC URL of the Amazon Redshift cluster where you want to connect.
- **Bucket details** — The Amazon S3 bucket where Amazon AppFlow writes your data as an intermediate destination. Amazon Redshift gets your data from this bucket.
- **IAM role for Amazon S3 access** — The IAM role that authorizes Amazon Redshift to get and decrypt the data from the S3 bucket.
- **Amazon Redshift database user name** — The user name that you use to authenticate with your Amazon Redshift database.
- **Amazon Redshift database password** — The password you use to authenticate with your Amazon Redshift database.

**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 Management Guide
Transferring data to Amazon Redshift with a flow

To transfer data to Amazon Redshift, create an Amazon AppFlow flow, and choose Amazon Redshift as the data destination. For the steps to create a flow, see Creating flows in Amazon AppFlow (p. 114).

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 (p. 34)
- Connection instructions (p. 34)
- Notes (p. 35)
- Related resources (p. 36)

**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. 176).
- 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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

**Tip**

If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 34) 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.

<table>
<thead>
<tr>
<th>Setting name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AWS Glue Data Catalog settings</strong></td>
<td>Catalog the data that you transfer in the AWS Glue Data Catalog. When you catalog your data, you make it easier to discover and access with AWS analytics and machine learning services. For more information, see Cataloging the data output from an Amazon AppFlow flow (p. 125).</td>
</tr>
<tr>
<td><strong>Data format preference</strong></td>
<td>• You can specify your preferred file format for the input file(s). The following options are currently available: CSV, JSONL.</td>
</tr>
<tr>
<td></td>
<td>• You can specify your preferred file format for the exported records. The following options are currently available: JSONL (default), CSV, or Apache Parquet.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Filename preference</strong></td>
<td>• You can choose to add a timestamp to the filename.</td>
</tr>
<tr>
<td></td>
<td>• Your filename will end with the file creation timestamp in YYYY-MM-DDThh:mm:sss format.</td>
</tr>
<tr>
<td></td>
<td>• The creation date is in UTC time.</td>
</tr>
</tbody>
</table>
## Setting name

<table>
<thead>
<tr>
<th>Setting name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition and aggregation settings</td>
<td>Organize the data that you transfer into partitions and files of a specified size. These settings can help you optimize query performance for applications that access the data. For more information, see Partitioning and aggregating the data output from an Amazon AppFlow flow (p. 131).</td>
</tr>
</tbody>
</table>

## 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

## 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 (p. 36)
- Connection instructions (p. 36)
- Notes (p. 37)
- Related resources (p. 38)

## 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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

Tip
If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 36).

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 (p. 38)
- Connection instructions (p. 38)
- Notes (p. 39)
- Related resources (p. 39)

**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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

**Tip**
If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 38) 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 (p. 40)
- Connection instructions (p. 40)
- Notes (p. 41)
- Related resources (p. 41)

**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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

Tip
If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 40).

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
Facebook Ads connector for Amazon AppFlow

You can use the Facebook Ads connector in Amazon AppFlow to transfer data about the ads that you run with the Facebook Marketing API. The Marketing API is a series of Graph API endpoints that create and manage ads on Facebook and Instagram. After you connect Amazon AppFlow to your Facebook developer account, you can transfer data about your ads, campaigns, budgets, and more.

Topics
- Facebook Ads support (p. 42)
- Before you begin (p. 42)
- Connecting Amazon AppFlow to the Facebook Marketing API (p. 43)
- Transferring data from the Facebook Marketing API with a flow (p. 44)
- Supported objects (p. 44)
- Supported destinations (p. 44)

Facebook Ads support

The following list summarizes how Amazon AppFlow supports the Facebook Marketing API through the Facebook Ads connector.

Supported as a data source?
Yes. You can use Amazon AppFlow to transfer data about your Facebook ads from the Marketing API.

Supported as a data destination?
No. You can't use Amazon AppFlow to transfer data to the Marketing API or your Facebook developer account.

Supported versions

Amazon AppFlow supports the following versions of the Marketing API:
- v14.0
- v13.0
- v12.0

For more information about Marketing API versions, see Changelog in the Meta for Developers documentation.

Before you begin

To use Amazon AppFlow to transfer data from the Marketing API to supported destinations, you'll need to meet these requirements:

- You have a Facebook developer account.
- Your account contains an app with its type set to Business. For information about creating an app, see Create an App in the Meta for Developers App Development documentation.
- Your Facebook developer app includes the Facebook Login product, which you've configured to meet the following additional requirements:
  - Client OAuth login is enabled
  - Web OAuth login is enabled
• One or more OAuth redirect URIs are present for Amazon AppFlow. Each of these URIs has the following form:

https://region.console.aws.amazon.com/appflow/oauth

In this URI, region is the code for the AWS Region where you use Amazon AppFlow to transfer data from the Marketing API. For example, if you use Amazon AppFlow in the US East (N. Virginia) region, the URI is https://us-east-1.console.aws.amazon.com/appflow/oauth.

For the AWS Regions that Amazon AppFlow supports, see Amazon AppFlow endpoints and quotas in the AWS General Reference.

For more information about Facebook Login, see Facebook Login in the Meta For Developers documentation.

• Your Facebook developer app includes the Marketing API product, which you use to manage the ads that Amazon AppFlow transfers data about.

Connecting Amazon AppFlow to the Facebook Marketing API

To connect Amazon AppFlow to data about your Facebook ads, create an Amazon AppFlow connection where you provide details about your Facebook developer app. If you haven't yet configured your app for Amazon AppFlow integration, see Before you begin (p. 42).

To create a Facebook Ads connection

1. Sign in to the AWS Management Console and open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane on the left, choose Connections.
3. On the Manage connections page, for Connectors, choose Facebook Ads.
4. Choose Create connection.
5. In the Connect to Facebook Ads window, enter the following information:
   
   • Custom authorization code URL – Specify the Marketing API version that you use in your Facebook developer app to complete the URL shown in the console:

     https://www.facebook.com/version/dialog/oauth

     For example, if you use v14.0, the URL is https://www.facebook.com/v14.0/dialog/oauth.

     For the Marketing API versions that Amazon AppFlow supports, see Facebook Ads support (p. 42).

   • Client ID – The App ID that's assigned to your Facebook developer app.

   • Client secret – The App secret that's assigned to your Facebook developer app.

   • Facebook Instance URL – Choose https://graph.facebook.com.

   • Facebook API version – Choose the Marketing API version that you use. This version must match the one that you specified for Custom authorization code URL.

   6. Optionally, under Data encryption, choose Customize encryption settings (advanced) if you want to encrypt your data with a customer managed key in the AWS Key Management Service (AWS KMS).

      By default, Amazon AppFlow encrypts your data with a KMS key that AWS creates, uses, and manages for you. Choose this option if you want to encrypt your data with your own KMS key instead.
Amazon AppFlow always encrypts your data during transit and at rest. For more information, see Data protection in Amazon AppFlow (p. 142).

If you want to use a KMS key from the current AWS account, select this key under Choose an AWS KMS key. If you want to use a KMS key from a different AWS account, enter the Amazon Resource Name (ARN) for that key.

7. For Connection name, enter a name for your connection.
8. Choose Connect.

On the Manage connections page, your new connection appears in the Connections table. When you create a flow that uses Facebook Ads as the data source, you can select this connection.

Transferring data from the Facebook Marketing API with a flow

To transfer data about your Facebook ads from the Marketing API, create an Amazon AppFlow flow, and choose Facebook Ads as the data source. For the steps to create a flow, see Creating flows in Amazon AppFlow (p. 114).

When you configure the flow, choose which data object you want to transfer. For most Facebook Ads objects, you must choose two values: one for Choose Facebook Ads object, and another for Choose Facebook Ads subobject. The subobject is an individual instance of the object. For example, if the object that you choose is Campaigns, then the subobject is the specific campaign to transfer data from. For the objects that Amazon AppFlow supports for Facebook Ads, see Supported objects (p. 44).

Also choose the destination where you want to transfer the data object that you selected. For information on how to configure your destination, see Supported destinations (p. 44).

Supported objects

When you create a flow that uses Facebook Ads as the data source, you can transfer any of the following data objects:

- Account
- Campaigns
- Ad Sets
- Campaign Budget
- Ads
- Ad Creatives

For more information about these objects and the data that they contain, see Ad Campaign Structure in the Meta for Developers Marketing API documentation.

Supported destinations

When you create a flow that uses Facebook Ads as the data source, you can set the destination to any of the following connectors:

- Amazon EventBridge (p. 24)
- Amazon Honeycode (p. 26)
Google Ads connector for Amazon AppFlow

Google Ads is a platform that advertisers use to display ads on the web, such as in Google search results, YouTube videos, mobile apps, and on websites. If you are a Google Ads user, you can use Amazon AppFlow to transfer data about your account, ad campaigns, and ad groups to certain AWS services or other supported applications.

Topics
- Google Ads support (p. 45)
- Before you begin (p. 45)
- Connecting Amazon AppFlow to your Google Ads account (p. 46)
- Transferring data from Google Ads with a flow (p. 47)
- Supported objects (p. 47)
- Supported destinations (p. 47)

Google Ads support

Amazon AppFlow supports Google Ads as follows.

Supported as a data source?
Yes. You can use Amazon AppFlow to transfer data from your Google Ads account.

Supported as a data destination?
No. You can't use Amazon AppFlow to transfer data to your Google Ads account.

Before you begin

To use Amazon AppFlow to transfer data from Google Ads to AWS services, you'll need to meet these requirements:

- You have a Google Cloud Platform account and a Google Cloud project.
- Your Google Cloud project includes Google Ads as one of its enabled APIs. For information on how to enable APIs, see Enable and disable APIs in the API Console Help for Google Cloud Platform.
- You have a Google Ads developer token. For information on how to retrieve or create a developer token, see Obtain Your Developer Token in the Google Ads API documentation.
- In your Google Cloud project, you've configured an OAuth consent screen that meets these additional requirements:
• It includes amazon.com as one of its authorized domains.
• It includes Google Ads API as one of its authorized scopes.

For more information about the OAuth consent screen, see Setting up your OAuth consent screen in the Google Cloud Platform Console Help.

• In your Google Cloud project, you’ve configured an OAuth 2.0 client ID that has one or more authorized redirect URIs for Amazon AppFlow. Each of these URIs has the following form:

https://region.consoles.amazon.com/appflow/oauth

In this URI, region is the code for the AWS Region where you use Amazon AppFlow to transfer data from Google Ads. For example, if you use Amazon AppFlow in the US East (N. Virginia) region, the URI is https://us-east-1.consoles.amazon.com/appflow/oauth.

For the AWS Regions that Amazon AppFlow supports, see Amazon AppFlow endpoints and quotas in the AWS General Reference.

For information on how to create an OAuth 2.0 client ID, see Setting up OAuth 2.0 in the Google Cloud Platform Console Help.

Connecting Amazon AppFlow to your Google Ads account

To connect Amazon AppFlow to your Google Ads account, provide details from the Google Cloud project so that Amazon AppFlow can access your Google Ads data. If you haven’t yet configured your Google Cloud project for Amazon AppFlow integration, see Before you begin (p. 45).

To create a Google Ads connection

1. Sign in to the AWS Management Console and open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane on the left, choose Connections.
3. On the Manage connections page, for Connectors, choose Google Ads.
4. Choose Create connection.
5. In the Connect to Google Ads window, enter the following information:
   • Access type – Choose offline.
   • Client ID – The client ID of the OAuth 2.0 client ID in your Google Cloud project.
   • Client secret – The client secret of the OAuth 2.0 client ID in your Google Cloud project.
   • Google Ads developer token – The developer token from your Google Ads account.
   • Google Ads API version – Choose v9.
   • Manager account ID – Optionally, the account ID of a Google Ads manager account that you want to connect with Amazon AppFlow.
6. Optionally, under Data encryption, choose Customize encryption settings (advanced) if you want to encrypt your data with a customer managed key in the AWS Key Management Service (AWS KMS).

   By default, Amazon AppFlow encrypts your data with a KMS key that AWS creates, uses, and manages for you. Choose this option if you want to encrypt your data with your own KMS key instead.

   Amazon AppFlow always encrypts your data during transit and at rest. For more information, see Data protection in Amazon AppFlow (p. 142).
If you want to use a KMS key from the current AWS account, select this key under **Choose an AWS KMS key**. If you want to use a KMS key from a different AWS account, enter the Amazon Resource Name (ARN) for that key.

7. For **Connection name**, enter a name for your connection.
8. Choose **Connect**. A **Sign in with Google** window opens.
9. Choose your Google account, and sign in.
10. On the page titled **amazon.com wants to access your Google Account**, choose Continue.

On the **Manage connections** page, your new connection appears in the **Connections** table. When you create a flow that uses Google Ads as the data source, you can select this connection.

**Transferring data from Google Ads with a flow**

To transfer data from Google Ads, create an Amazon AppFlow flow, and choose Google Ads as the data source. For the steps to create a flow, see **Creating flows in Amazon AppFlow** (p. 114).

When you configure the flow, choose which data object you want to transfer. For the objects that Amazon AppFlow supports for Google Ads, see **Supported objects** (p. 47).

Also choose the destination where you want to transfer the data object that you selected. For more information about how to configure your destination, see **Supported destinations** (p. 47).

**Supported objects**

When you create a flow that uses Google Ads as the data source, you can transfer any of the following data objects to supported destinations:

- Account
- Account Budget
- Campaign
- Campaign Budget
- Ad Group
- Ad Group Ad

**Supported destinations**

When you create a flow that uses Google Ads as the data source, you can set the destination to any of the following connectors:

- Amazon EventBridge (p. 24)
- Amazon Honeycode (p. 26)
- Amazon Lookout for Metrics (p. 27)
- Amazon Redshift (p. 29)
- Amazon S3 (p. 34)
- Marketo (p. 57)
- Salesforce (p. 65)
- SAP OData (p. 78)
- Snowflake (p. 93)
- Upsolver (p. 96)
- Zendesk (p. 101)
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 (p. 48)
- Connection instructions (p. 48)
- Notes (p. 49)
- Related resources (p. 51)

**Requirements**

You must log in to the Google API Console at [https://console.developers.google.com](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/](https://console.aws.amazon.com/).
- 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. Note: Your Google Analytics user account must also be a Google Workspaces user account.

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

**Connection instructions**

**To connect to Google Analytics while creating a flow**

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 *Creating flows in Amazon AppFlow* (p. 114).

**Tip**
If you aren't connected successfully, ensure that you have followed the instructions in the *Requirements* (p. 48) 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:dimensionXX` or `ga:metricXX`, 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 (p. 51)
- Connection instructions (p. 51)
- Notes (p. 52)

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. Sign in to the AWS Management Console and 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.
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 **Creating flows in Amazon AppFlow** (p. 114).

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

**Jira Cloud connector for Amazon AppFlow**

Jira Cloud is a platform developed by Atlassian. The platform includes issue tracking products that help teams plan and track their agile projects. If you’re a Jira Cloud user, your account contains data about
your projects, such as issues, workflows, and events. You can use Amazon AppFlow to transfer your Jira
Cloud data to certain AWS services or other supported applications.

Jira Cloud support

Amazon AppFlow supports Jira Cloud as follows.

Supported as a data source?

Yes. You can use Amazon AppFlow to transfer data from Jira Cloud.

Supported as a data destination?

No. You can't use Amazon AppFlow to transfer data to Jira Cloud.

Supported Jira Cloud products

Amazon AppFlow uses the Jira REST API to transfer data objects from the Jira Software product. It
does not transfer objects that are unique to the other products in Jira Cloud: Jira Work Management
and Jira Service Management.

Amazon AppFlow only connects to Jira Software on Jira Cloud. Amazon AppFlow doesn't connect to
the on-premise Jira Software Data Center product.

Supported Jira API version

Version 2

Before you begin

To use Amazon AppFlow to transfer data from Jira Cloud to supported destinations, you must meet
these requirements:

• You have an Atlassian account where you use the Jira Software product in Jira Cloud.
• In the developer console for your Atlassian account, you’ve created an OAuth 2.0 integration app for
Amazon AppFlow. This app provides the client credentials that Amazon AppFlow uses to access your
data securely when it makes authenticated calls to your account. For more information, see Enabling
OAuth 2.0 (3LO) in the Atlassian Developer documentation.

You must configure your app as follows:
• In the authorization settings, you’ve specified a callback URL for Amazon AppFlow.

These URLs have the following form:

https://region.console.aws.amazon.com/appflow/oauth

In this URL, region is the code for the AWS Region where you use Amazon AppFlow to transfer data
from Jira Cloud. For example, the code for the US East (N. Virginia) Region is us-east-1. For that
Region, the URL is the following:

https://us-east-1.console.aws.amazon.com/appflow/oauth

For the AWS Regions that Amazon AppFlow supports, and their codes, see Amazon AppFlow
endpoints and quotas in the AWS General Reference.
• In the distribution settings, you’ve set the distribution status to Sharing.
• In the permissions settings, you’ve added the Jira API, and you’ve enabled the recommended scopes
below.
In the settings for your app, note the client ID and client secret because you need them to create a connection in Amazon AppFlow.

**Recommended scopes**

Before Amazon AppFlow can securely access your data in Jira Cloud, the permissions settings for your OAuth 2.0 integration app must allow the necessary scopes for the Jira API. We recommend that you enable the scopes below so that Amazon AppFlow can access all supported data objects.

If you want to allow fewer scopes, you can omit any scopes that apply to objects that you don't want to transfer.

You can add scopes to your app by managing permissions in the Atlassian Developer console.

- Under **Jira platform REST API** scopes, we recommend that you add all scopes.
- Under **Granular scopes**, we recommend that you add the following scopes:
  - `read:application-role:jira`
  - `read:audit-log:jira`
  - `read:avatar:jira`
  - `read:field:jira`
  - `read:group:jira`
  - `read:instance-configuration:jira`
  - `read:issue-details:jira`
  - `read:issue-event:jira`
  - `read:issue-link-type:jira`
  - `read:issue-meta:jira`
  - `read:issue-security-level:jira`
  - `read:issue-security-scheme:jira`
  - `read:issue-type-scheme:jira`
  - `read:issue-type-screen-scheme:jira`
  - `read:issue-type:jira`
  - `read:issue.time-tracking:jira`
  - `read:label:jira`
  - `read:notification-scheme:jira`
  - `read:permission:jira`
  - `read:priority:jira`
  - `read:project:jira`
  - `read:project-category:jira`
  - `read:project-role:jira`
  - `read:project-type:jira`
  - `read:project-version:jira`
  - `read:project.component:jira`
  - `read:project.property:jira`
  - `read:resolution:jira`
  - `read:screen:jira`
  - `read:status:jira`
  - `read:user:jira`
  - `read:workflow-scheme:jira`
  - `read:workflow:jira`
Connecting Amazon AppFlow to your Jira Cloud account

To connect Amazon AppFlow to your Jira Cloud account, provide details from your OAuth 2.0 integration app so that Amazon AppFlow can access your data. If you haven't yet configured your Jira Cloud account for Amazon AppFlow integration, see Before you begin (p. 53).

To create a Jira Cloud connection

1. Sign in to the AWS Management Console and open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane on the left, choose Connections.
3. On the Manage connections page, for Connectors, choose Jira Cloud.
4. Choose Create connection.
5. In the Connect to Jira Cloud window, enter the following information:
   - **Client ID** – The client ID from the OAuth 2.0 integration app.
   - **Client secret** – The client secret from the OAuth 2.0 integration app.
   - **Jira Cloud Domain URL** – The URL where you sign in to your Jira Cloud account, for example, https://your-account.atlassian.net.
6. Optionally, under Data encryption, choose Customize encryption settings (advanced) if you want to encrypt your data with a customer managed key in the AWS Key Management Service (AWS KMS).

   By default, Amazon AppFlow encrypts your data with a KMS key that AWS creates, uses, and manages for you. Choose this option if you want to encrypt your data with your own KMS key instead.

   Amazon AppFlow always encrypts your data during transit and at rest. For more information, see Data protection in Amazon AppFlow (p. 142).

   If you want to use a KMS key from the current AWS account, select this key under **Choose an AWS KMS key**. If you want to use a KMS key from a different AWS account, enter the Amazon Resource Name (ARN) for that key.
7. For **Connection name**, enter a name for your connection.
8. Choose Continue. A window appears that asks if you want to allow Amazon AppFlow to access your Atlassian account.
9. Choose Accept.

On the Manage connections page, your new connection appears in the Connections table. When you create a flow that uses Jira Cloud as the data source, you can select this connection.

Transferring data from Jira Cloud with a flow

To transfer data from Jira Cloud, create an Amazon AppFlow flow, and choose Jira Cloud as the data source. For the steps to create a flow, see Creating flows in Amazon AppFlow (p. 114).

When you configure the flow, choose the data object that you want to transfer. For the objects that Amazon AppFlow supports for Jira Cloud, see Supported objects (p. 56).
Also, choose the destination where you want to transfer the data object that you selected. For more information about how to configure your destination, see Supported destinations (p. 57).

## Supported objects

When you create a flow that uses Jira Cloud as the data source, you can transfer any of the following data objects to supported destinations:

<table>
<thead>
<tr>
<th>Object</th>
<th>Jira API endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit Record</td>
<td>/auditing/record</td>
</tr>
<tr>
<td>Groups</td>
<td>/group/bulk</td>
</tr>
<tr>
<td>Issue</td>
<td>/search</td>
</tr>
<tr>
<td>Issue Events</td>
<td>/events</td>
</tr>
<tr>
<td>Issue Fields</td>
<td>/field</td>
</tr>
<tr>
<td>Issue Field Configurations</td>
<td>/fieldconfiguration</td>
</tr>
<tr>
<td>Issue Link Type</td>
<td>/issuelinktype</td>
</tr>
<tr>
<td>Issue Notification Schemes</td>
<td>/notificationscheme</td>
</tr>
<tr>
<td>Issue Priority</td>
<td>/priority</td>
</tr>
<tr>
<td>Issue Resolution</td>
<td>/resolution</td>
</tr>
<tr>
<td>Issue Security Scheme</td>
<td>/issuesecurityschemes</td>
</tr>
<tr>
<td>Issue Type</td>
<td>/issuetype</td>
</tr>
<tr>
<td>Issue Type Scheme</td>
<td>/issuetypescheme</td>
</tr>
<tr>
<td>Issue Type Screen Scheme</td>
<td>/issuetypescreescheme</td>
</tr>
<tr>
<td>Jira Settings</td>
<td>/application-properties</td>
</tr>
<tr>
<td>Jira Settings Advanced</td>
<td>/application-properties/advanced-settings</td>
</tr>
<tr>
<td>Jira Settings Global</td>
<td>/configuration</td>
</tr>
<tr>
<td>Label</td>
<td>/label</td>
</tr>
<tr>
<td>Myself</td>
<td>/myself</td>
</tr>
<tr>
<td>Permission</td>
<td>/mypermissions</td>
</tr>
<tr>
<td>Project</td>
<td>/project/search</td>
</tr>
<tr>
<td>Project Category</td>
<td>/projectCategory</td>
</tr>
</tbody>
</table>
Supported destinations

When you create a flow that uses Jira Cloud as the data source, you can set the destination to any of the following connectors:

- Amazon EventBridge (p. 24)
- Amazon Honeycode (p. 26)
- Amazon Lookout for Metrics (p. 27)
- Amazon Redshift (p. 29)
- Amazon S3 (p. 34)
- Marketo (p. 57)
- Salesforce (p. 65)
- SAP OData (p. 78)
- Snowflake (p. 93)
- Upsolver (p. 96)
- Zendesk (p. 101)

**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 (p. 58)
• Connection instructions (p. 58)
• Notes (p. 59)
• Related resources (p. 60)

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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

Tip
If you aren’t connected successfully, ensure that you have followed the instructions in Requirements (p. 58).

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

Mixpanel connector for Amazon AppFlow

Mixpanel is a service that provides analytics about user engagement in web and mobile applications. If you use Mixpanel, you can also use Amazon AppFlow to transfer your data to certain AWS services or other supported applications.

Topics

- Mixpanel support (p. 60)
- Before you begin (p. 60)
- Connecting Amazon AppFlow to your Mixpanel account (p. 61)
- Transferring data from Mixpanel with a flow (p. 61)
- Supported objects (p. 62)
- Supported destinations (p. 64)

Mixpanel support

Amazon AppFlow supports Mixpanel as follows.

Supported as a data source?

Yes. You can use Amazon AppFlow to transfer data from your Mixpanel account.

Supported as a data destination?

No. You can't use Amazon AppFlow to transfer data to your Mixpanel account.

Before you begin

Before you can use Amazon AppFlow to transfer data from Mixpanel, you need the following:

- A Mixpanel project that contains the data that you want to transfer.
- A service account for your Mixpanel project. In Mixpanel, a service account is a type of user that you authorize to access a project programmatically with the Mixpanel API. Amazon AppFlow needs this account to access your data. For more information, see Service Accounts in the Mixpanel documentation.

When you create a Mixpanel connection in Amazon AppFlow, you provide the following properties from your service account:
Connecting Amazon AppFlow to your Mixpanel account

To connect Amazon AppFlow to your Mixpanel project, provide details about the service account that enables Amazon AppFlow to access your data. To create a service account, see Before you begin (p. 60).

To create a Mixpanel connection

1. Sign in to the AWS Management Console and open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane on the left, choose Connections.
3. On the Manage connections page, for Connectors, choose Mixpanel.
4. Choose Create connection.
5. In the Connect to Mixpanel window, enter the following:
   - User name – The user name of the Mixpanel service account that provides access to your project.
   - Password – The service account secret.
   - MixPanel Instance URL – Choose https://mixpanel.com/api/app/me.
   - MixPanel API version – Choose 2.0.
6. Optionally, under Data encryption, choose Customize encryption settings (advanced) if you want to encrypt your data with a customer managed key in the AWS Key Management Service (AWS KMS).

By default, Amazon AppFlow encrypts your data with a KMS key that AWS creates, uses, and manages for you. Choose this option if you want to encrypt your data with your own KMS key instead.

Amazon AppFlow always encrypts your data during transit and at rest. For more information, see Data protection in Amazon AppFlow (p. 142).

If you want to use a KMS key from the current AWS account, select this key under Choose an AWS KMS key. If you want to use a KMS key from a different AWS account, enter the Amazon Resource Name (ARN) for that key.

7. For Connection name, enter a name for your connection.
8. Choose Connect.

On the Manage connections page, your new connection appears in the Connections table. When you create a flow that uses Mixpanel as the data source, you can select this connection.

Transferring data from Mixpanel with a flow

To transfer data from Mixpanel, create an Amazon AppFlow flow, and choose Mixpanel as the data source. To learn how to create a flow, see Creating flows in Amazon AppFlow (p. 114).

When you configure the flow, choose which data object that you want to transfer. For more information about the objects that Amazon AppFlow supports for Mixpanel, see Supported objects (p. 62).

Required filters for Mixpanel data objects

When you create a flow and use Mixpanel as the data source, most data objects require you to specify one or more filters. Filters are typically optional criteria that you use to transfer data
objects selectively. Specifically for flows that transfer from Mixpanel, you must specify filters to provide Amazon AppFlow with parameter values that it needs to query your data. For the filters that are required for each Mixpanel data object, see the section called “Supported objects” (p. 62).

Also choose the destination where you want to transfer the data object that you selected. For more information on how to configure your destination, see the section called “Supported destinations” (p. 64).

### Supported objects

When you create a flow that uses Mixpanel as the data source, you can transfer any of the data objects shown in the following table. To retrieve each object, Amazon AppFlow sends a query to the URI in the Mixpanel endpoint column. Most data objects support one or more filters that appear under Supported filters. Flows that transfer from Mixpanel require certain filters.

<table>
<thead>
<tr>
<th>Object</th>
<th>Mixpanel endpoint</th>
<th>Supported filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotations</td>
<td>/annotations</td>
<td>• from_date</td>
</tr>
<tr>
<td>Cohorts</td>
<td>/cohorts/list</td>
<td>None</td>
</tr>
<tr>
<td>Engage</td>
<td>/engage</td>
<td>None</td>
</tr>
<tr>
<td>Events</td>
<td>/annotations</td>
<td>• event’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• from_date’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• to_date’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• type’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• unit’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• workspace_id</td>
</tr>
<tr>
<td>Events Names</td>
<td>/events/names</td>
<td>• limit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• type’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• workspace_id</td>
</tr>
<tr>
<td>Events Properties</td>
<td>/events/properties</td>
<td>• event’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• from_date’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• limit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• name’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• to_date’</td>
</tr>
<tr>
<td>Events Properties Top</td>
<td>/events/properties/top</td>
<td>• event’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• limit</td>
</tr>
<tr>
<td>Object</td>
<td>Mixpanel endpoint</td>
<td>Supported filters</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>The following paths are appended to the base URI:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="https://mixpanel.com/api/2.0">https://mixpanel.com/api/2.0</a>.</td>
<td></td>
</tr>
<tr>
<td>Events Properties Values</td>
<td>/events/properties/values</td>
<td>event, limit, name, workspace_id</td>
</tr>
<tr>
<td>Events Top</td>
<td>/events/top</td>
<td>limit, type, workspace_id</td>
</tr>
<tr>
<td>Funnels</td>
<td>/funnels</td>
<td>from_date, funnel_id, interval, length, length_unit, limit, to_date, unit, workspace_id</td>
</tr>
<tr>
<td>Profile Event Activity</td>
<td>/stream/query</td>
<td>distinct_ids, from_date, to_date, workspace_id</td>
</tr>
<tr>
<td>Retention</td>
<td>/retention/addiction</td>
<td>addiction_unit, event, from_date, limit, to_date, unit, workspace_id</td>
</tr>
<tr>
<td>Segmentation</td>
<td>/segmentation</td>
<td>event, from_date, interval, limit, to_date, type, unit, workspace_id</td>
</tr>
<tr>
<td>Object</td>
<td>Mixpanel endpoint</td>
<td>Supported filters</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------</td>
<td>-------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Segmentation Average | /segmentation/average | • event'  
• from_date'  
• on'  
• to_date'  
• unit  
• workspace_id |
| Segmentation Numeric | /segmentation/numeric | • event'  
• from_date'  
• on'  
• to_date'  
• type  
• unit  
• workspace_id |
| Segmentation Sum     | /segmentation/sum  | • event'  
• from_date'  
• on'  
• to_date'  
• unit  
• workspace_id |

* You must specify this filter in your flow definition before Amazon AppFlow can successfully retrieve your data.

**Supported destinations**

When you create a flow that uses Mixpanel as the data source, you can set the destination to any of the following connectors:

- Amazon EventBridge (p. 24)
- Amazon Honeycode (p. 26)
- Amazon Lookout for Metrics (p. 27)
- Amazon Redshift (p. 29)
- Amazon S3 (p. 34)
- Marketo (p. 57)
- Salesforce (p. 65)
- SAP OData (p. 78)
- Snowflake (p. 93)
- Upsolver (p. 96)
- Zendesk (p. 101)
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 (p. 65)
- Connection instructions (p. 65)
- Additional flow settings for Salesforce (p. 69)
- Notes (p. 72)
- Related resources (p. 72)

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, us-west-2, ap-northeast-1, ap-south-1, ap-southeast-2, ca-central-1, and eu-central-1 AWS Regions.

Connection instructions

- **Connect to Salesforce while creating a flow**
- **Use a global connected app with Amazon AppFlow** (p. 66)
- **Create a global connected app in Salesforce** (p. 68)

To connect to Salesforce while creating a flow

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 **Creating flows in Amazon AppFlow** (p. 114).

**Tip**

If you aren’t connected successfully, ensure that you have followed the instructions in the **Requirements** (p. 65) 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. 68).

- To use your own global connected app, you need to pass on the clientId, clientSecret, and Secrets Manager secret ARN to Amazon AppFlow.
- The following example shows a sample Secrets Manager secret with application credentials for Salesforce:

```
{
  "Name": "Salesforce",
  "VersionId": "db85ae8b-0-995-480a-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": {
      "accessToken": "testAccessToken",
      "refreshToken": "testRefreshToken",
      "oauthRequest": {
        "authCode": null,
        "redirectUri": null
      }
    }
  }
}
```

- You must attach a resource policy to the Secrets Manager secret and the KMS key which is used to 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, 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 Amazon AppFlow created resources created on your behalf.

```json
{
"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.

```json
{
"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.
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.

Additional flow settings for Salesforce

When you configure a flow that transfers data to or from Salesforce, the Amazon AppFlow console shows some unique settings that aren't available for flows that don't use Salesforce.

Salesforce API preference

When you use Salesforce as the source or destination, you can configure the Salesforce API preference setting. Use this setting to specify what Salesforce API Amazon AppFlow uses when your flow transfers data to or from Salesforce. Your choice optimizes your flow for small to medium-sized data transfers, large data transfers, or both.

The Amazon AppFlow console provides this setting on the Configure flow page under Source details or Destination details. To view it, expand the Additional settings section.
You can choose one of the following options:

- **Automatic (default)** — For each flow run, Amazon AppFlow selects the API to use based on the number of records that the run transfers. The threshold of records that determines the API varies based on whether Salesforce is the source or the destination, as shown in the following table:

<table>
<thead>
<tr>
<th>Is Salesforce the source or destination?</th>
<th>Number of records transferred</th>
<th>API used to transfer records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Fewer than 1,000,000</td>
<td>Salesforce REST API</td>
</tr>
<tr>
<td></td>
<td>1,000,000 or more</td>
<td>Salesforce Bulk API 2.0</td>
</tr>
<tr>
<td>Destination</td>
<td>Fewer than 1,000</td>
<td>Salesforce REST API</td>
</tr>
<tr>
<td></td>
<td>1,000 or more</td>
<td>Salesforce Bulk API 2.0</td>
</tr>
</tbody>
</table>

**Notes**

- If you choose this option, be aware that each of the potential Salesforce APIs structures data differently. For recurring flows, the data output might vary from one flow run to the next. For example, if a flow runs daily, it might use REST API on one day to transfer 900 records, and it might use Bulk API 2.0 on the next day to transfer 1,100 records. For each of these runs, the respective Salesforce API formats the data differently. Some of the differences include how dates are formatted and how null values are represented.
- Flow runs that use Bulk API 2.0 can’t transfer Salesforce compound fields.

If you choose this option, you optimize flow performance for all data transfer sizes, but the tradeoff is inconsistent formatting in the output.

- **Standard** — Amazon AppFlow uses only Salesforce REST API. This option optimizes your flow for small- to medium-sized data transfers. By choosing this option, you ensure that your flow writes consistent output, but you decrease performance for large data transfers that are better suited for Bulk API 2.0.

  **Note**

  If you choose this option and your flow attempts to transfer a very large set of data, it might fail with a timeout error.

- **Bulk** — Amazon AppFlow uses only Salesforce Bulk API 2.0. This API runs asynchronous data transfers, and it’s optimal for large sets of data. If you choose this option, you ensure that your flow writes consistent output, but you optimize performance only for large data transfers.

  **Note**

  If you choose this option, your flow can’t transfer Salesforce compound fields because Bulk API 2.0 doesn’t support them.
Salesforce destination record preference

When you use Salesforce as a destination, the Amazon AppFlow shows additional settings on the Map data fields page under Destination record preference.

You can choose one of the following options:

**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 with 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 Amazon AppFlow finds a matching record, Amazon AppFlow updates the record in Salesforce. If Amazon AppFlow finds no matching record, Amazon AppFlow inserts the data as a new record. Any errors in performing the operation are handled according to your chosen error handling option. You can specify your error handling preferences on the Configure flow page.

**Delete existing records**

When you choose this setting, Amazon AppFlow deletes Salesforce records that you specify. To specify the records, create a file that contains the IDs that Salesforce assigned to them. Provide that file as the source data for your flow.

For example, the following CSV file lists the IDs of two Salesforce records to delete.

```
salesforce_id
71
```

71
In this example, the IDs appear under the one source field for the file, salesforce_id.

In your flow definition, you must specify the source field that contains the IDs of the objects to delete. You do this when you map data fields. At that point, you map the source field to the corresponding destination field in Salesforce. For example, if you assigned the Salesforce object Opportunity to your flow, then the destination field name is Opportunity ID.

You can provide a source data file that has other fields besides the one with the IDs, but Amazon AppFlow ignores them.

Each flow can delete only one type of object, which is the Salesforce object that you choose when you configure the destination details.

After your flow runs, you can view the records that it deleted in your Salesforce recycle bin. You can recover your files from the recycle bin if needed. However, you must do so before its retention period elapses or before the files are manually purged.

If any errors occur when you run the flow, Amazon AppFlow handles them according to the error handling option that you choose when you configure the flow.

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 55.0 on August 30th, 2022. Flows associated with all Salesforce connections created after this date will use Salesforce API version 55.0. Flows created before this date but after January 19th, 2021, will use Salesforce API version 50.0, while any flows created before January 19th, 2021, will use Salesforce API version 47.0.
- Amazon AppFlow supports Change Data Capture Events and Platform events from Salesforce.

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 Marketing Cloud connector for Amazon AppFlow

Marketing Cloud is a Salesforce platform for digital marketing that helps its customers manage campaigns across multiple channels, including email, mobile, and social. If you use Marketing Cloud, you can also use Amazon AppFlow to transfer your data to certain AWS services or other supported applications.

**Topics**
- Salesforce Marketing Cloud support (p. 73)
- Before you begin (p. 73)
- Connecting Amazon AppFlow to your Salesforce Marketing Cloud account (p. 74)
- Transferring data from Salesforce Marketing Cloud with a flow (p. 75)
- Supported objects (p. 75)
- Supported destinations (p. 76)

**Salesforce Marketing Cloud support**

Amazon AppFlow supports Salesforce Marketing Cloud as follows.

**Supported as a data source?**

Yes. You can use Amazon AppFlow to transfer data from your Marketing Cloud account.

**Supported as a data destination?**

No. You can't use Amazon AppFlow to transfer data to your Marketing Cloud account.

**Before you begin**

Before you can use Amazon AppFlow to transfer data from Marketing Cloud, you need the following:

- A Salesforce Marketing Cloud account that contains the data that you want to transfer. For more information about the Marketing Cloud data objects that Amazon AppFlow supports, see Supported objects (p. 75).
- A Marketing Cloud package so that Amazon AppFlow can access your data. In Marketing Cloud, you create packages to add custom functionality to your account. For the steps to create a package, see Create and Install Packages in the Marketing Cloud documentation.

When you create a package for Amazon AppFlow integration, do the following:

1. Add an API integration component to the package.
2. Set the integration type of the component to server-to-server.
3. Grant read access to every data object that you want to transfer with Amazon AppFlow.

4. After you create the package, note the following properties. You need them to create a connection in Amazon AppFlow:
   - Client ID
   - Client secret
   - Authentication base URI
   - REST base URI or SOAP base URI (You can use either one, and it doesn't matter which one you use)

Connecting Amazon AppFlow to your Salesforce Marketing Cloud account

To connect Amazon AppFlow to your Marketing Cloud account, provide details about the package so that Amazon AppFlow can access your data. To learn how to create a package, see Before you begin (p. 73).

To create a Salesforce Marketing Cloud connection

1. Sign in to the AWS Management Console and open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane on the left, choose Connections.
3. On the Manage connections page, for Connectors, choose Salesforce Marketing Cloud.
4. Choose Create connection.
5. In the Connect to Salesforce Marketing Cloud window, provide the following details:
   - Custom authorization tokens URL – The authentication base URI that's assigned to your Marketing Cloud package. Provide the subdomain to complete the URI shown in the console: https://subdomain.auth.marketingcloudapis.com/v2/token.
   - Client ID – The client ID that is assigned to your Marketing Cloud package.
   - Client secret – The client secret that is assigned to your Marketing Cloud package.
   - Salesforce Marketing Cloud Subdomain Endpoint – The REST base URI or SOAP base URI that is assigned to your Marketing Cloud package. These URIs looks similar to the following examples:
     - https://subdomain.rest.marketingcloudapis.com/
     - https://subdomain.soap.marketingcloudapis.com/
     
     In these examples, subdomain is the same value that you provide for the custom authorization tokens URL.

     You must provide either the REST or SOAP URI, but the one that you use doesn't matter. With either one, Amazon AppFlow connects to your Marketing Cloud package, and it transfers data by using the REST or SOAP endpoint as needed.

     For more information about the authentication, REST, and SOAP URIs for Marketing Cloud packages, see Your Subdomain and Your Tenant's Endpoints in the Marketing Cloud documentation.

6. Optionally, under Data encryption, choose Customize encryption settings (advanced) if you want to encrypt your data with a customer managed key in the AWS Key Management Service (AWS KMS).

     By default, Amazon AppFlow encrypts your data with a KMS key that AWS creates, uses, and manages for you. Choose this option if you want to encrypt your data with your own KMS key instead.

     Amazon AppFlow always encrypts your data during transit and at rest. For more information, see Data protection in Amazon AppFlow (p. 142).
If you want to use a KMS key from the current AWS account, select this key under **Choose an AWS KMS key**. If you want to use a KMS key from a different AWS account, enter the Amazon Resource Name (ARN) for that key.

7. For **Connection name**, enter a name for your connection.

8. Choose **Connect**.

On the **Manage connections** page, your new connection appears in the **Connections** table. When you create a flow that uses Salesforce Marketing Cloud as the data source, you can select this connection.

### Transferring data from Salesforce Marketing Cloud with a flow

To transfer data from Marketing Cloud, create an Amazon AppFlow flow, and choose Salesforce Marketing Cloud as the data source. To learn how to create a flow, see **Creating flows in Amazon AppFlow** (p. 114).

When you configure the flow, choose the data object that you want to transfer. For more information about the objects that Amazon AppFlow supports for Marketing Cloud, see **Supported objects** (p. 75).

Also choose the destination where you want to transfer the data object that you selected. For more information on how to configure your destination, see the section called “Supported destinations” (p. 76).

### Supported objects

When you create a flow that uses Salesforce Marketing Cloud as the data source, you can transfer the following data objects from your Marketing Cloud account:

- Activity
- Bounce Event
- Click Event
- Content Area
- Data Extension
- Email
- Forwarded Email Event
- Forwarded Email OptInEvent
- Link
- Link Send
- List
- List Subscriber
- Not Sent Event
- Open Event
- Send
- Sent Event
- Subscriber
- Survey Event
- Unsub Event
- Audit Events
Supported destinations

When you create a flow that uses Salesforce Marketing Cloud as the data source, you can set the
destination to any of the following connectors:

- Amazon EventBridge (p. 24)
- Amazon Honeycode (p. 26)
- Amazon Lookout for Metrics (p. 27)
- Amazon Redshift (p. 29)
- Amazon S3 (p. 34)
- Marketo (p. 57)
- Salesforce (p. 65)
- SAP OData (p. 78)
- Snowflake (p. 93)
- Upsolver (p. 96)
- Zendesk (p. 101)

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 (p. 76)
- Setup instructions (p. 77)
- Notes (p. 77)
- Related resources (p. 78)

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. 68). 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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

Tip

If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 76) 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 connector for Amazon AppFlow

The Amazon AppFlow SAP OData connector provides the ability to fetch, create, and update records exposed by SAP S/4HANA and SAP on premises systems through OData APIs.

With this connector, you can connect Amazon AppFlow to your OData services, including those that extract data from SAP applications that use the Operational Data Provisioning (ODP) framework. These applications are called ODP providers. For more information about how OData services can extract ODP data in SAP, see ODP-Based Data Extraction via OData in the SAP BW/4HANA documentation.

When you connect Amazon AppFlow to ODP providers, you can create flows that run full data transfers or incremental updates. Incremental updates for ODP data are efficient because they transfer only those records that changed since the prior flow run.

SAP OData support

With the SAP OData connector, Amazon AppFlow supports SAP as follows.

Supported as a data source?

Yes. You can use Amazon AppFlow to transfer data from SAP.

Supported as a data destination?

Amazon AppFlow supports SAP OData as a destination, but not for ODP data. You can use Amazon AppFlow to transfer data to an OData service, but you can't transfer data to an ODP provider.

Before you begin

To use Amazon AppFlow to transfer data from SAP OData to supported destinations, you must meet these 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.
Before you begin

1. **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.

   ![OData V4.0 Service Groups](ServiceGroups.png)

2. 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`.

3. Your SAP OData service must support client side pagination/query options such as `$top` and `$skip`. It must also support system query option `$count`.

4. Amazon 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

5. You must enable secure setup for connecting over HTTPS.

6. 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.

### ODP Requirements

Before you can transfer data from an ODP provider, you need to meet the following requirements:

1. You have an SAP NetWeaver AS ABAP instance.
2. Your SAP NetWeaver instance contains an ODP provider that you want to transfer data from. ODP providers include:
   - SAP DataSources (Transaction code RSO2)
   - SAP Core Data Services ABAP CDS Views
   - SAP BW or SAP BW/4HANA systems (InfoObject, DataStore Object)
   - Real-time replication of Tables and DB-Views from SAP Source System via SAP Landscape Replication Server (SAP SLT)
   - SAP HANA Information Views in SAP ABAP based Sources
3. Your SAP NetWeaver instance has the SAP Gateway Foundation component.
4. You have created an OData service that extracts data from your ODP provider. To create the OData service, you use the SAP Gateway Service Builder. To access your ODP data, Amazon AppFlow calls this service by using the OData API. For more information, see [Generating a Service for Extracting ODP Data via OData](#) in the SAP BW/4HANA documentation.
5. To generate an OData service based on ODP data sources, SAP Gateway Foundation must be installed locally in your ERP/BW stack or in a hub configuration.
   - For your ERP/BW applications, the SAP NetWeaver AS ABAP stack must be at 7.50 SP02 or above.
• For the hub system (SAP Gateway), the SAP NetWeaver AS ABAP of the hub system must be 7.50 SP01 or above for remote hub setup.

Private Connection Requirements

Before you can create a private connection to SAP, you need to meet the following 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.

Connecting Amazon AppFlow to your SAP account

To connect Amazon AppFlow to your SAP account, provide details about your SAP OData service so that Amazon AppFlow can access your data. If you haven't yet configured your SAP OData service for Amazon AppFlow integration, see **Before you begin (p. 78)**.

To create an SAP OData connection

2. In the navigation pane on the left, choose **Connections**.
3. On the **Manage connections** page, for **Connectors**, choose **SAP OData**.
4. Choose **Create connection**.
5. In the **Connect to SAP OData** window, enter the following information:
   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`. Amazon 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-xxxxxxxxxxxxxxxxx`
   g. Select your preferred Authentication Mode.
      • If Basic,
         i. Under **User name**, enter your username.
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. Optionally, under **Data encryption**, choose **Customize encryption settings (advanced)** if you want to encrypt your data with a customer managed key in the AWS Key Management Service (AWS KMS).

By default, Amazon AppFlow encrypts your data with a KMS key that AWS creates, uses, and manages for you. Choose this option if you want to encrypt your data with your own KMS key instead.

Amazon AppFlow always encrypts your data during transit and at rest. For more information, see [Data protection in Amazon AppFlow](p. 142).

If you want to use a KMS key from the current AWS account, select this key under **Choose an AWS KMS key**. If you want to use a KMS key from a different AWS account, enter the Amazon Resource Name (ARN) for that key.

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

j. Choose **Continue**.

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

Key ID: 2816c7bc-1da6-4cb2-bb06-566b8b0d3c01

**Connection name**

Specify a new connection name
On the **Manage connections** page, your new connection appears in the **Connections** table. When you create a flow that uses SAP OData as the data source, you can select this connection.

If you chose to enable PrivateLink, note the following:

- Amazon 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. While the connection status is PENDING, you are unable to transfer SAP OData objects with a flow.
- 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.
- Once the AWS PrivateLink Endpoint connection is established, Amazon AppFlow fetches (only for OAuth) access/refresh tokens using the authCode, makes a test connection call over private network, and finally changes connection status from PENDING to CREATED.
- If for any reason private connection creation fails, connection status would change to FAILED.

### Transferring data from SAP OData with a flow

To transfer data from SAP OData, create an Amazon AppFlow flow, and choose SAP OData as the data source. For the steps to create a flow, see [Creating flows in Amazon AppFlow](p. 114).

When you configure the flow, choose which data object you want to transfer. If the data object originates from an ODP provider, you can configure the flow so that it runs efficient incremental updates that transfer changed records only. For more information, see [Transferring data from ODP providers](p. 114) below.

### Transferring ODP data

When you create a flow that transfers an ODP data object, you can configure the flow to run **incremental** or **full** data transfers.

#### Incremental ODP data transfers

When you create a flow that transfers ODP data incrementally, it does the following:

- It subscribes to the **operational delta queue** of your ODP provider. This queue provides Amazon AppFlow with delta tokens, which indicate changes made to the provider's records in SAP.
- For the initial flow run, it performs a full data transfer. It obtains all available records from your ODP provider, except for any that you omit by adding filters to your flow configuration.
- For subsequent flow runs, it performs incremental data transfers. By using the information provided by the delta tokens, it transfers only those records that changed after the last flow run.

When you create an SAP OData flow in the Amazon AppFlow console, you can configure it to transfer data incrementally in the **Flow trigger** section, where you do the following:

1. Choose **Run flow on schedule**.
2. Use the scheduling fields to specify when the flow begins, how often it repeats, and when it ends.
3. Under **Transfer mode**, choose **Incremental transfer**.

For ODP data objects specifically, the console requires no additional input. This behavior differs from SAP data objects that don't come from an ODP provider. For those objects, you must specify
a source timestamp field that Amazon AppFlow uses to identify new or changed records. For ODP data, no such timestamp is necessary because Amazon AppFlow uses the information that’s provided by the delta token that it receives from the operational delta queue.

Important
When you create an incremental flow for an ODP data object, the flow creates a subscription to the operational delta queue for that object. Although Amazon AppFlow creates these subscriptions, it doesn't administer them on your behalf. Keep the following subscription behaviors in mind to prevent unwanted effects:

- When a flow subscribes to a queue, it also removes all prior subscriptions to that queue. If you previously created any scheduled flows that transfer the same object, delete those flows. They no longer receive delta tokens, and they stop performing incremental data transfers. For any individual ODP object, maintain only one scheduled flow at a time.
- When you delete a flow that subscribes to an operational delta queue, that operation does not delete the subscription itself. You can only delete the subscription by using the SAP system to do so.

Full ODP data transfers

You can create flows that run full data transfers of your ODP data. For these flows, Amazon AppFlow does not create subscriptions to operational delta queues like it does for incremental flows.

When you create an SAP OData flow in the Amazon AppFlow console, you can configure it to run full data transfers in the Flow trigger section, where you do the following:

- Under Choose how to trigger the flow, do either of the following:
  - Choose Run on demand. After you create an on-demand flow, you run it manually by choosing Run flow on its details page in the Amazon AppFlow console.
  - Choose Run flow on schedule and define your schedule:
    a. Use the scheduling fields to specify when the flow begins, how often it repeats, and when it ends.
    b. For Transfer mode, choose Full transfer.

Note
To create a flow that runs full data transfers, the frequency that you choose must be no more frequent than Daily. If it is more frequent, then you won’t be able to choose Full transfer.

Advanced capabilities for the SAP OData connector

For the SAP OData connector, Amazon AppFlow supports a couple unique capabilities that are unavailable with other destination-enabled connectors. With it, you can:

- Capture the SAP success response when you create a new record.
- Create deep entities with the SAP OData deep insert feature. For more information about this feature, see Deep Insert in the SAP Gateway Foundation documentation.

You can use these capabilities individually or in combination. For example, you can capture SAP’s success response when you insert a deep entity.

To enable these capabilities, complete the following steps.
To capture the SAP success response for new records

1. Create an Amazon S3 bucket. The bucket must be in the same AWS Region as the flow that you create for your SAP OData connector. For the steps to create a bucket, see Creating a bucket in the Amazon S3 User Guide.

2. Configure the flow by following the steps in Getting started with Amazon AppFlow (p. 6), but do one additional step:
   - On the Configure flow page, under Response handling, select the bucket that you created. The SAP success response payload is delivered to this bucket when finish creating your flow.

To create SAP deep entities

1. Generate a JSON Lines input file that defines one deep entity per line, as shown by the following example.

   JSON Lines (required format)

   The following input file defines two deep entities in JSON Lines format (also called newline-delimited JSON). In this format, each line is a complete JSON object that defines an individual deep entity.

   Each deep entity can include multiple levels of hierarchical data. This example creates two Sales Orders, and each contains two associated Sales Order Items.

   ```json
   {"SalesOrderType": "OR","SalesOrganization": "1710","DistributionChannel": "10","OrganizationDivision": "00","SoldToParty": "USCU_S13","TransactionCurrency": "USD","PurchaseOrderByCustomer": "TEST-PO2021","to_Item": [{"Material": "MZ-FG-C990","RequestedQuantity": "10","RequestedQuantityUnit": "PC"},{"Material": "MZ-FG-M500","RequestedQuantity": "10","RequestedQuantityUnit": "PC"}]}
   {"SalesOrderType": "OR","SalesOrganization": "1710","DistributionChannel": "10","OrganizationDivision": "00","SoldToParty": "USCU_S13","TransactionCurrency": "USD","PurchaseOrderByCustomer": "TEST-PO2021","to_Item": [{"Material": "MZ-FG-C990","RequestedQuantity": "10","RequestedQuantityUnit": "PC"},{"Material": "MZ-FG-M500","RequestedQuantity": "10","RequestedQuantityUnit": "PC"}]}
   ```

   Formatted JSON (for readability)

   The following example shows one of the deep entities from the JSON Lines input file. This example is formatted for readability so that you can more easily see the nested JSON values.

   ```json
   {
   "SalesOrderType": "OR",
   "SalesOrganization": "1710",
   "DistributionChannel": "10",
   "OrganizationDivision": "00",
   "SoldToParty": "USCU_S13",
   "TransactionCurrency": "USD",
   "PurchaseOrderByCustomer": "TEST-PO2021",
   "to_Item": [
   {
   "Material": "MZ-FG-C990",
   "RequestedQuantity": "10",
   "RequestedQuantityUnit": "PC"
   },
   {
   "Material": "MZ-FG-M500",
   "RequestedQuantity": "10",
   "RequestedQuantityUnit": "PC"
   }
   ]
   ```
Remember that Amazon AppFlow requires JSON Lines format, so this example would be an invalid input file.

2. Create an Amazon S3 bucket. The bucket must be in the same AWS Region as the flow that you create for your SAP OData connector. For the steps to create a bucket, see Creating a bucket in the Amazon S3 User Guide.

3. Upload your deep entities input file to the bucket that you created. For the steps to upload a file, see Uploading objects in the Amazon S3 User Guide.

4. Configure the flow by following the steps in Getting started with Amazon AppFlow (p. 6), but do one alternate step:

   • On the Map data fields page, under Mapping method, choose Passthrough fields without modification.

     **Note**
     When you choose this option, the console disables the options under Source to destination field mapping. With this option, you don't define mappings in the console. Instead, the fields in your input file must match the fields that you use in SAP.

**Notes**

• 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, Amazon 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.

• Amazon 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 (p. 87)
• Connection instructions (p. 87)
• Notes (p. 88)
• Related resources (p. 89)
Requirements

Before you can use Amazon AppFlow to import data from ServiceNow, you need the following:

- A ServiceNow account so that you can provide Amazon AppFlow with your user name, password, and instance name.
- Access to your ServiceNow instance through a role. This can be an admin role or one that allows the read operation for the following:
  - `sys_db_object`
  - `sys_db_object.*`
  - `sys_dictionary`
  - `sys_dictionary.*`
  - `sys_glide_object`
  - Any table that you want to access with Amazon AppFlow. For example, if you want to import data from a table named `incidents`, you need read access to `incidents` and `incidents.*`.

For more information about ServiceNow roles, see Roles in the ServiceNow documentation.

Connection instructions

To connect to ServiceNow while creating a flow

1. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

**Tip**
If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 87) 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 (p. 89)
- Connection instructions (p. 89)
- Notes (p. 90)
- Related resources (p. 90)

**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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

**Tip**
If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 89) 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 (p. 91)
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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

**Tip**
If you aren't connected successfully, ensure that you have followed the instructions in the Requirements (p. 91) 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
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 (p. 93)
- Connection instructions (p. 93)
- Related resources (p. 95)

**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.

```json
{
    "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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).
Tip
If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 93) 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 Trend Micro with Amazon AppFlow.

Note
You can use Trend Micro as a source only.

Topics
• Requirements (p. 95)
• Connection instructions (p. 95)
• Notes (p. 96)
• Related resources (p. 96)

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. Sign in to the AWS Management Console and 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 drop-down 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 Creating flows in Amazon AppFlow (p. 114).

**Tip**
If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 95) 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.
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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

Tip
If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 97).

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 (p. 98)
- Connection instructions (p. 98)
- Extract Veeva VAULT documents with Amazon AppFlow (p. 99)
- Notes (p. 101)
- Related resources (p. 101)

**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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

**Tip**
If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 98) 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. Sign in to the AWS Management Console and 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.

![Configure flow](image)

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 (p. 101)
- Connection instructions (p. 102)
- Notes (p. 103)
- Related resources (p. 104)

**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. Sign in to the AWS Management Console and 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 Creating flows in Amazon AppFlow (p. 114).

**Tip**
If you aren’t connected successfully, ensure that you have followed the instructions in the Requirements (p. 101).

**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:

<table>
<thead>
<tr>
<th>Setting name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert new records</td>
<td>• This is the default data transfer option.</td>
</tr>
<tr>
<td>Setting name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>• When you choose this setting, Amazon AppFlow inserts your source data into the chosen Zendesk object as a new record.</td>
</tr>
<tr>
<td>Update existing records</td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td>Upsert records</td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
</tbody>
</table>

**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

### Zendesk Chat connector for Amazon AppFlow

Zendesk Chat is a live chat service that Zendesk offers as part of its platform. Zendesk Chat helps businesses automate and enhance customer support interactions across web, mobile, and social channels. In a Zendesk Chat account, you store data related to customer conversations. If you use Zendesk Chat, you can also use Amazon AppFlow to transfer this data to certain AWS services or other supported applications.

**Zendesk Chat support**

Amazon AppFlow supports Zendesk Chat as follows.

**Supported as a data source?**

Yes. You can use Amazon AppFlow to transfer data from Zendesk Chat.

**Supported as a data destination?**

No. You can't use Amazon AppFlow to transfer data to Zendesk Chat.
Before you begin

To use Amazon AppFlow to transfer data from Zendesk Chat to supported destinations, you must meet these requirements:

- You have a Zendesk Chat account.
- In the Zendesk Chat account settings, you've registered Amazon AppFlow with an *API client*. The API client provides the client credentials that Amazon AppFlow uses to access your data securely with authenticated calls to your account.
- You've configured your API client with one or more redirect URLs for Amazon AppFlow.

These URLs have the following form:

```
https://region.console.aws.amazon.com/appflow/oauth
```

In this URL, `region` is the code for the AWS Region where you use Amazon AppFlow to transfer data from Zendesk Chat. For example, the code for the US East (N. Virginia) Region is `us-east-1`. For that Region, the URL is the following:

```
https://us-east-1.console.aws.amazon.com/appflow/oauth
```

For the AWS Regions that Amazon AppFlow supports, and their codes, see Amazon AppFlow endpoints and quotas in the *AWS General Reference*.

In the settings for your API client, note the client ID and client secret because you will need them to create a connection in Amazon AppFlow.

Connecting Amazon AppFlow to your Zendesk Chat account

To connect Amazon AppFlow to your Zendesk Chat account, provide your Zendesk subdomain and the client credentials that authorize Amazon AppFlow to access your data. If you haven't yet configured your Zendesk Chat account to integrate with Amazon AppFlow, see Before you begin (p. 105).

**To create a Zendesk Chat connection**

2. In the navigation pane on the left, choose **Connections**.
3. On the Manage connections page, for **Connectors**, choose **Zendesk Chat**.
4. Choose **Create connection**.
5. In the **Connect to Zendesk Chat** window, enter the following information:
   
   - **Custom authorization code URL** – Your Zendesk subdomain. You can find this value in the URL that you visit when you sign in to Zendesk Chat. For example, in the account URL `https://my-account.zendesk.com`, the subdomain is `my-account`.
   - **Client ID** and **Client secret** – The client credentials that Zendesk assigned to your API client.
6. Optionally, under **Data encryption**, choose **Customize encryption settings (advanced)** if you want to encrypt your data with a customer managed key in the AWS Key Management Service (AWS KMS).
By default, Amazon AppFlow encrypts your data with a KMS key that AWS creates, uses, and manages for you. Choose this option if you want to encrypt your data with your own KMS key instead.

Amazon AppFlow always encrypts your data during transit and at rest. For more information, see Data protection in Amazon AppFlow (p. 142).

If you want to use a KMS key from the current AWS account, select this key under Choose an AWS KMS key. If you want to use a KMS key from a different AWS account, enter the Amazon Resource Name (ARN) for that key.

7. For **Connection name**, enter a name for your connection.
8. Choose **Continue**. A window appears that asks if you want to allow Amazon AppFlow to access your Zendesk Chat account.
9. Choose **Allow**.

On the Manage connections page, your new connection appears in the Connections table. When you create a flow that uses Zendesk Chat as the data source, you can select this connection.

### Transferring data from Zendesk Chat with a flow

To transfer data from Zendesk Chat, create an Amazon AppFlow flow, and choose Zendesk Chat as the data source. For the steps to create a flow, see Creating flows in Amazon AppFlow (p. 114).

When you configure the flow, choose the data object that you want to transfer. For the objects that Amazon AppFlow supports for Zendesk Chat, see Supported objects (p. 106).

Also, choose the destination where you want to transfer the data object that you selected. For more information about how to configure your destination, see Supported destinations (p. 107).

### Supported objects

When you create a flow that uses Zendesk Chat as the data source, you can transfer any of the following data objects to supported destinations:

- Chat Offline Message
- Chat Support Chat
- Agent
- Agent Event
- Account
- Department
- Trigger
- Shortcut
- Ban
- Goal
- Skill
- Role
- Route Setting Account
- Route Setting Agent
Supported destinations

When you create a flow that uses Zendesk Chat as the data source, you can set the destination to any of the following connectors:

- Amazon EventBridge (p. 24)
- Amazon Honeycode (p. 26)
- Amazon Lookout for Metrics (p. 27)
- Amazon Redshift (p. 29)
- Amazon S3 (p. 34)
- Marketo (p. 57)
- Salesforce (p. 65)
- SAP OData (p. 78)
- Snowflake (p. 93)
- Upsolver (p. 96)
- Zendesk (p. 101)

Zendesk Sell connector for Amazon AppFlow

Zendesk Sell is a customer relationship management (CRM) service that Zendesk offers as part of its platform. Zendesk Sell automates sales workflows to help its users engage leads and close deals. In a Zendesk Sell account, you store data related to sales opportunities, such as contacts, deals, and leads. If you use Zendesk Sell, you can also use Amazon AppFlow to transfer this data to certain AWS services or other supported applications.

Zendesk Sell support

Amazon AppFlow supports Zendesk Sell as follows.

**Supported as a data source?**

Yes. You can use Amazon AppFlow to transfer data from Zendesk Sell.

**Supported as a data destination?**

No. You can't use Amazon AppFlow to transfer data to Zendesk Sell.

Before you begin

To use Amazon AppFlow to transfer data from Zendesk Sell to supported destinations, you must meet these requirements:

- You have a Zendesk Sell account.
- In the OAuth settings for your Zendesk Sell account, you've registered Amazon AppFlow with a developer app. The developer app provides the client credentials that Amazon AppFlow uses to access your data securely with authenticated calls to the Zendesk Sell API.
- You've configured the developer app with a redirect URL for Amazon AppFlow.

These URLs have the following form:
Connecting Amazon AppFlow to your Zendesk Sell account

To connect Amazon AppFlow to your Zendesk Sell account, provide the client credentials from the developer app that authorizes Amazon AppFlow to access your data. If you haven’t yet configured your Zendesk Sell account to integrate with Amazon AppFlow, see Before you begin (p. 107).

To create a Zendesk Sell connection

1. Sign in to the AWS Management Console and open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane on the left, choose Connections.
3. On the Manage connections page, for Connectors, choose Zendesk Sell.
4. Choose Create connection.
5. In the Connect to Zendesk Sell window, enter values for Client ID and Client secret. Zendesk assigns these client credentials to the developer app in your Zendesk Sell account.
6. Optionally, under Data encryption, choose Customize encryption settings (advanced) if you want to encrypt your data with a customer managed key in the AWS Key Management Service (AWS KMS).

   By default, Amazon AppFlow encrypts your data with a KMS key that AWS creates, uses, and manages for you. Choose this option if you want to encrypt your data with your own KMS key instead.

   Amazon AppFlow always encrypts your data during transit and at rest. For more information, see Data protection in Amazon AppFlow (p. 142).

   If you want to use a KMS key from the current AWS account, select this key under Choose an AWS KMS key. If you want to use a KMS key from a different AWS account, enter the Amazon Resource Name (ARN) for that key.
7. For Connection name, enter a name for your connection.
8. Choose Continue. An Authorize Application window opens. The window prompts you to give Amazon AppFlow read-only access to your data.
9. Choose Authorize.

On the Manage connections page, your new connection appears in the Connections table. When you create a flow that uses Zendesk Sell as the data source, you can select this connection.
Transferring data from Zendesk Sell with a flow

To transfer data from Zendesk Sell, create an Amazon AppFlow flow, and choose Zendesk Sell as the data source. For the steps to create a flow, see Creating flows in Amazon AppFlow (p. 114).

When you configure the flow, choose the data object that you want to transfer. For the objects that Amazon AppFlow supports for Zendesk Sell, see Supported objects (p. 109).

Also, choose the destination where you want to transfer the data object that you selected. For more information about how to configure your destination, see Supported destinations (p. 109).

Supported objects

When you create a flow that uses Zendesk Sell as the data source, you can transfer any of the following data objects to supported destinations:

- Contact
- Deal
- Lead
- Note
- Task

Supported destinations

When you create a flow that uses Zendesk Sell as the data source, you can set the destination to any of the following connectors:

- Amazon EventBridge (p. 24)
- Amazon Honeycode (p. 26)
- Amazon Lookout for Metrics (p. 27)
- Amazon Redshift (p. 29)
- Amazon S3 (p. 34)
- Marketo (p. 57)
- Salesforce (p. 65)
- SAP OData (p. 78)
- Snowflake (p. 93)
- Upsolver (p. 96)
- Zendesk (p. 101)
Managing Amazon AppFlow connections

To enable data flows in Amazon AppFlow, you provide access to your source and destination applications by creating connections. Connections store the configuration details and credentials that Amazon AppFlow requires to transfer data with applications on your behalf. For example, these details include your user names, passwords, secret keys, and API access tokens. After you create a connection, you can assign it to new or existing flows without manually entering the configuration details anew.

Use the following sections to work with your connections by using the Amazon AppFlow console, AWS CLI, or the Amazon AppFlow API. Connections are also called connector profiles in the AWS CLI and Amazon AppFlow API.

Amazon AppFlow console

Complete the following steps to manage your connections by using the Amazon AppFlow console.

1. Sign in to the AWS Management Console and open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.
2. In the navigation pane, select Connections.
3. On the Manage connections page, for Connectors, choose the application that your connection accesses. For example, if your connection enables data to flow to or from Amazon Redshift, you would choose Amazon Redshift.
4. After you choose a connector, you can do any of the following:
   - To create a connection, choose Create connection, and provide the required details. These details vary for each type of connector application. For application-specific instructions, find your application under Supported source and destination applications (p. 23).
   - To view the details for a connection, choose its name in the Connection name column.
   - To edit a connection, select it and choose Edit.
   - To copy a connection, select it and choose Copy to new connection. The console shows a window where you configure a new connection, and it copies the initial settings from the connection that you selected. You can modify these settings before you create the new connection.
   - Amazon AppFlow doesn't copy OAuth credentials, such as client secret and client ID. For connections that require those credentials, you must provide them anew.
   - To delete a connection, select it and choose Delete.

AWS CLI

You can manage your connections in Amazon AppFlow by running commands with the AWS CLI.

To create a connection

- Run the create-connector-profile command. In this command, you provide configuration details and credentials for the --connector-profile-config parameter. The required details vary for each type of connector application.
The following example creates a connection for SAP OData, and it provides the configuration details in a JSON file:

```
$ aws appflow create-connector-profile \
>  --connector-profile-name sap-odata-connection \
>  --connector-type SAPOData \
>  --connection-mode Public \
>  --connector-profile-config file://sap-odata-connector-profile-config.json
```

The `sap-odata-connector-profile-config.json` file contains the following configuration details:

```
{
  "connectorProfileProperties": {
    "SAPOData": {
      "applicationHostUrl": "https://example.connection.url",
      "applicationServicePath": "/sap/opu/odata/example/path;v=2",
      "portNumber": 443,
      "clientNumber": "100",
      "logonLanguage": "EN"
    }
  },
  "connectorProfileCredentials": {
    "SAPOData": {
      "basicAuthCredentials": {
        "username": "username",
        "password": "password"
      }
    }
  }
}
```

The command response provides the Amazon Resource Name (ARN) of the new connection:

```
{
  "connectorProfileArn": "arn:aws:appflow:us-east-1:111122223333:connectorprofile/sap-odata-connection"
}
```

To view the details for all of your connections

- Run the `describe-connector-profiles` command:

```
$ aws appflow describe-connector-profiles
```

The command response is a JSON body with details for each of your connections. The following example response shows the details for an SAP OData connection:

```
{
  "connectorProfileDetails": [
    {
      "connectorProfileName": "sap-odata-connection",
      "connectorProfileType": "SAPOData",
      "connectionMode": "Public",
      "applicationHostUrl": "https://example.connection.url",
      "applicationServicePath": "/sap/opu/odata/example/path;v=2",
      "portNumber": 443,
      "clientNumber": "100",
      "logonLanguage": "EN",
      "basicAuthCredentials": {
        "username": "username",
        "password": "password"
      }
    }
  ]
}
```
To view the details for specific connections

- Run the `describe-connector-profiles` command, and filter the results by using the `--connector-profile-names` or `--connector-type` parameters. The following example gets the details for a single connection:

  ```bash
  $ aws appflow describe-connector-profiles --connector-profile-names sap-odata-connection
  ```

To edit a connection

- Run the `update-connector-profile` command. For this command, you provide the updated configuration details for the `--connector-profile-config` parameter. The following example provides the updated configuration in a JSON file:

  ```bash
  ```

To delete a connection

- Run the `delete-connector-profile` command.

  ```bash
  $ aws appflow delete-connector-profile --connector-profile-name sap-odata-connection
  ```

Amazon AppFlow API

You can manage your connections by using the following actions in the Amazon AppFlow API:

- `CreateConnectorProfile` – Creates a connection.
- `DescribeConnectorProfiles` – Provides details about your connections.
• **UpdateConnectorProfile** – Edits a connection.
• **DeleteConnectorProfile** – Deletes a connection.
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 view flow details**

1. Sign in to the AWS Management Console and 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**

- Creating flows in Amazon AppFlow (p. 114)
- Cataloging the data output from an Amazon AppFlow flow (p. 125)
- Partitioning and aggregating the data output from an Amazon AppFlow flow (p. 131)
- Activate an Amazon AppFlow flow (p. 135)
- Edit an Amazon AppFlow flow (p. 135)
- Delete an Amazon AppFlow flow (p. 136)
- Flow triggers (p. 136)
- Private Amazon AppFlow flows (p. 138)
- Flow notifications (p. 138)

Creating flows in Amazon AppFlow

There are several ways to create flows in Amazon AppFlow. You can use the AWS Management Console, AWS CLI commands, Amazon AppFlow APIs, or AWS CloudFormation.
Create a flow using the AWS console

There are several ways to get started with creating your first flow by using the AWS console user interface, AWS CLI commands, APIs, or by specifying CloudFormation resources. The console enables you to input basic information for your flow and connect as a user of the associated SaaS application.

To create a flow using the console

The following procedure provides the steps to create and configure a flow using the Amazon AppFlow console user interface.

1. Sign in to the AWS Management Console and 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.

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. 23) 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. 23) 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. 23) page.

**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.

To 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.

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. (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.

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.

Create a flow using the AWS CLI

You may also use the CLI to create a connector profile and configure a flow using the AWS CLI commands for create-connector-profile and create-flow. Due to the varying methods of authentication across each target application, the specific information provided for connection creation will vary. Two examples are provided here as a comparison — Salesforce and ServiceNow.

Run the create-connector-profile command to create the connector profile for your flow. The following example creates a new Amazon AppFlow connection to Salesforce. Note that this leverages a Salesforce Connected App, which itself requires several steps to configure across AWS and Salesforce. See Salesforce global connected app for details.

Create Salesforce connection:

```bash
aws appflow create-connector-profile --connector-profile-name MySalesforceConnection
```
Run the **create-connector-profile** command to begin creating your flow. The following example creates a new Amazon AppFlow connection to ServiceNow. Note that, unlike Salesforce, there is no prerequisite configuration for either AWS or ServiceNow.

Create ServiceNow connection:

```
aws appflow create-connector-profile \    
  --connector-profile-name MyServiceNowConnection \    
  --connector-type Servicenow \    
  --connection-mode Public \    
  --connector-profile-config '{ 
    "connectorProfileProperties": { 
      "ServiceNow": { 
        "instanceUrl": "https://<instance-name>.service-now.com"

    
  },
  "connectorProfileCredentials": { 
    "ServiceNow": { 
      "username": "<username-value>", 
      "password": "<password-value>"

    
  }',

```

Run the **create-flow** command to begin creating your flow. The following implements a flow from Salesforce to S3 using a previously created Salesforce connection and S3 bucket, delivering the data in CSV format with all Salesforce source fields mapped directly.

Create Salesforce to S3 flow:

```
aws appflow create-flow \    
  --flow-name MySalesforceToS3Flow \    
  --trigger-config '{ 
    "triggerType": "OnDemand"

  }', 
  --source-flow-config '{ 
    "connectorType": "Salesforce", 
    "connectorProfileName": "MySalesforceConnection", 
    "sourceConnectorProperties": {

```
Create a flow using the Amazon AppFlow APIs

You may also use the APIs to create a connector profile and configure a flow using the CreateConnectorProfile and CreateFlow APIs. Due to the varying methods of authentication across each target application, the specific information provided for connection creation will vary. Two examples are provided below as a comparison — Salesforce and ServiceNow.

Program the CreateConnectorProfile API to create a connector profile associated with your AWS account. 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. The following examples creates a new Amazon AppFlow connection to Salesforce. Note that this leverages a Salesforce Connected App, which itself requires several steps to configure across AWS and Salesforce. See Salesforce global connected app for details.

Create Salesforce connection:

```
POST /create-connector-profile HTTP/1.1
Content-type: application/json
```

Run the start-flow command to start your flow. For on-demand flows, this operation runs the flow immediately. For schedule and event-triggered flows, this operation activates the flow. The following starts the flow MySalesforceToS3Flow which was created in the previous step.

```
aws appflow start-flow --flow-name MySalesforceToS3Flow
```

The describe-flow command is helpful for understanding how previously created flows, including flows created through the Console, are structured.

Describe a flow:

```
aws appflow describe-flow --flow-name MySalesforceToS3Flow
```

Refer to the AWS CLI Command Reference for Amazon AppFlow for additional details about the complete list of commands available for Amazon AppFlow.
The following examples create a new Amazon AppFlow connection to ServiceNow. Note that, unlike Salesforce, there is no pre-requisite configuration for either AWS or ServiceNow.

Create ServiceNow connection

```json
POST /create-connector-profile HTTP/1.1
Content-type: application/json

{
  "connectorProfileName": "MyServiceNowConnection",
  "connectorType": "Servicenow",
  "connectionMode": "Public",
  "connectorProfileConfig": {
    "connectorProfileProperties": {
      "ServiceNow": {
        "instanceUrl": "https://<instance-name>.service-now.com",
        "isSandboxEnvironment": false
      }
    },
    "connectorProfileCredentials": {
      "ServiceNow": {
        "username": "<username-value>",
        "password": "<password-value>"
      }
    }
  }
}
```

The following implements a flow from Salesforce to S3 using a previously created Salesforce connection and S3 bucket, delivering the data in CSV format with all Salesforce source fields mapped directly.

Create Salesforce to S3 flow

```json
POST /create-flow HTTP/1.1
Content-type: application/json

{
  "connectorProfileName": "MySalesforceConnection",
  "connectorType": "Salesforce",
  "connectionMode": "Public",
  "connectorProfileConfig": {
    "connectorProfileProperties": {
      "Salesforce": {
        "instanceUrl": "https://<instance-name>.my.salesforce.com",
        "isSandboxEnvironment": false
      }
    },
    "connectorProfileCredentials": {
      "Salesforce": {
        "accessToken": "<access-token-value>",
        "refreshToken": "<refresh-token-value>",
        "oAuthRequest": {
          "authCode": "<auth-code-value>",
          "redirectUri": "https://login.salesforce.com/"
        },
        "clientCredentialsArn": "<secret-arn-value>"
      }
    }
  }
}
```
Create a flow using CloudFormation resources

You may also use CloudFormation to create a connector profile and configure a flow using the AWS::AppFlow::ConnectorProfile and AWS::AppFlow::Flow resources. The following example creates a new Amazon AppFlow connection to Salesforce. Note that this leverages a Salesforce Connected App, which itself requires several steps to configure across AWS and Salesforce. See Salesforce global connected app for details.

Declare the AWS::AppFlow::ConnectorProfile entity in your CloudFormation template with the following JSON syntax:

```json
{
    "connectorProfileName": "MySalesforceConnection",
    "connectorProfileProperties": {
        "Salesforce": {
            "object": "Account"
        }
    }
}
```

The following starts the flow MySalesforceToS3Flow which was created in the previous step.

Start a flow:

```json
POST /start-flow HTTP/1.1
Content-type: application/json

{
    "flowName": "MySalesforceToS3Flow"
}
```

Refer to the Amazon AppFlow API Reference for details about the complete set of Amazon AppFlow APIs.
Create a flow using CloudFormation resources

```
{
  "AWSTemplateFormatVersion":"2010-09-09",
  "Resources": {
    "MySalesforceConnection": {
      "Type": "AWS::AppFlow::ConnectorProfile",
      "Properties": {
        "ConnectorProfileName": "MySalesforceConnection",
        "ConnectorType": "Salesforce",
        "ConnectionMode": "Public",
        "ConnectorProfileConfig": {
          "ConnectorProfileProperties": {
            "Salesforce": {
              "InstanceUrl": "https://<instance-name>.my.salesforce.com",
              "IsSandboxEnvironment": false
            }
          },
          "ConnectorProfileCredentials": {
            "Salesforce": {
              "AccessToken": "<access-token-value>",
              "RefreshToken": "<refresh-token-value>",
              "ConnectorOAuthRequest": {
                "AuthCode": "<auth-code-value>",
                "RedirectUri": "https://login.salesforce.com/"
              },
              "ClientCredentialsArn": "<secret-arn-value>"
            }
          }
        }
      }
    }
  }
}
```

Following is an example of YAML syntax:

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  MySalesforceConnection:
    Type: AWS::AppFlow::ConnectorProfile
    Properties:
      ConnectorProfileName: MySalesforceConnection
      ConnectorType: Salesforce
      ConnectionMode: Public
      ConnectorProfileConfig:
        ConnectorProfileProperties:
          Salesforce:
            InstanceUrl: https://<instance-name>.my.salesforce.com
            IsSandboxEnvironment: false
        ConnectorProfileCredentials:
          Salesforce:
            AccessToken: <access-token-value>
            RefreshToken: <refresh-token-value>
            ConnectorOAuthRequest:
              AuthCode: <auth-code-value>
              RedirectUri: https://login.salesforce.com/
            ClientCredentialsArn: <secret-arn-value>
```

The following examples creates a new Amazon AppFlow connection to ServiceNow.

Create ServiceNow connection - JSON
Create a flow using CloudFormation resources

```
{
  "AWSTemplateFormatVersion":"2010-09-09",
  "Resources": {
    "MyServiceNowConnection": {
      "Type": "AWS::AppFlow::ConnectorProfile",
      "Properties": {
        "ConnectorProfileName": "MyServiceNowConnection",
        "ConnectorType": "Servicenow",
        "ConnectionMode": "Public",
        "ConnectorProfileConfig": {
          "ConnectorProfileProperties": {
            "ServiceNow": {
              "InstanceUrl": "https://<instance-name>.service-now.com",
            }
          }
        },
        "ConnectorProfileCredentials": {
          "ServiceNow": {
            "Username": "<username-value>",
            "Password": "<password-value>"
          }
        }
      }
    }
  }
}
```

The following is an example of YAML syntax that creates a new Amazon AppFlow connection to ServiceNow.

Create ServiceNow connection - YAML:

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  MyServiceNowConnection:
    Type: AWS::AppFlow::ConnectorProfile
    Properties:
      ConnectorProfileName: MyServiceNowConnection
      ConnectorType: Servicenow
      ConnectionMode: Public
      ConnectorProfileConfig:
        ConnectorProfileProperties:
          ServiceNow:
            InstanceUrl: https://<instance-name>.service-now.com
        ConnectorProfileCredentials:
          ServiceNow:
            Username: <username-value>
            Password: <password-value>
```

The following implements a flow from Salesforce to S3 using a previously created Salesforce connection and S3 bucket, delivering the data in CSV format with all Salesforce source fields mapped directly.

Create Salesforce to S3 flow - JSON:

```
{
  "AWSTemplateFormatVersion":"2010-09-09",
  "Resources": {
    "MySalesforceToS3Flow": {
```
The following implements a flow from Salesforce to S3 using a previously created Salesforce connection and S3 bucket, delivering the data in CSV format with all Salesforce source fields mapped directly.

Create Salesforce to S3 flow - YAML:

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  MySalesforceToS3Flow:
    Type: AWS::AppFlow::Flow
    Properties:
      FlowName: MySalesforceToS3Flow
      TriggerConfig:
        TriggerType: OnDemand
      SourceFlowConfig:
        ConnectorType: Salesforce
        ConnectorProfileName: MySalesforceConnection
        SourceConnectorProperties:
          Salesforce:
            Object: Account
      DestinationFlowConfigList:
        - 
      Tasks:
        - 
```

```
"Type": "AWS::AppFlow::Flow",
"Properties": {
  "FlowName": "MySalesforceToS3Flow",
  "TriggerConfig": {
    "TriggerType": "OnDemand"
  },
  "SourceFlowConfig": {
    "ConnectorType": "Salesforce",
    "ConnectorProfileName": "MySalesforceConnection",
    "SourceConnectorProperties": {
      "Salesforce": {
        "Object": "Account"
      }
    }
  },
  "DestinationFlowConfigList": [{
    "ConnectorType": "S3",
    "DestinationConnectorProperties": {
      "S3": {
        "BucketName": "<s3-bucket-name>",
        "S3OutputFormatConfig": {
          "FileType": "CSV"
        }
      }
    }
  }],
  "Tasks": [
    {
      "TaskType": "Map_all",
      "SourceFields": [],
      "TaskProperties": [{
        "Key": "EXCLUDE_SOURCE_FIELDS_LIST",
        "Value": "[]"
      }],
      "ConnectorOperator": {
        "Salesforce": "NO_OP"
      }
    }
  ]
}
```
Cataloging flow output

- ConnectorType: S3
  DestinationConnectorProperties:
  S3:
    BucketName: <s3-bucket-name>
  S3OutputFormatConfig:
    FileType: CSV
  Tasks:
    - TaskType: Map_all
      SourceFields: []
      TaskProperties:
        - Key: EXCLUDE_SOURCE_FIELDS_LIST
          Value: '[]'
  ConnectorOperator:
    Salesforce: NO_OP

Refer to the AWS CloudFormation User Guide Amazon AppFlow chapter for details about the complete set of resource options for all sources and destinations.

Cataloging the data output from an Amazon AppFlow flow

When you use Amazon AppFlow to transfer data with a flow that meets certain requirements, you get the option to register the data with a data catalog. A data catalog is a metadata repository. The metadata represents aspects of your data, such as the schema, format, and data types. A data catalog provides a unified view of your data, even if the data belongs to multiple datasets that reside in multiple locations. By querying the consolidated metadata in a data catalog, you can more quickly search and discover your data assets.

To catalog your data, you create flows that transfer to Amazon S3, and you configure these flows with the required settings. When the flows run, Amazon AppFlow creates metadata tables in the AWS Glue Data Catalog.

The AWS Glue Data Catalog is a component of the AWS Glue service. You can use the Data Catalog to discover and search your data assets across various locations, including S3 buckets. When you register your data with the Data Catalog, you can more quickly access it with many analytics, AI, and ML services in AWS. These services include AWS Glue, Amazon Athena, Amazon SageMaker Data Wrangler, and more.

It’s convenient to catalog your data with Amazon AppFlow for the following reasons:

- You can transfer and catalog your data in the same operation.
- You don’t have to use crawlers to populate the Data Catalog.

When you run a flow that catalogs your data, Amazon AppFlow does the following in your AWS account:

- Prepares the metadata that represents the data output of the flow.
- Writes the metadata to a Data Catalog table.
- Stores the table in a Data Catalog database.

The Data Catalog table also includes any partition keys that organize your data in Amazon S3. For any flow that transfers data to Amazon S3, you can activate several types of partition keys in the flow settings. For more information, see Partitioning and aggregating the data output from an Amazon AppFlow flow (p. 131).

For more information about the Data Catalog, see AWS Glue Data Catalog in the AWS Glue Developer Guide.
Before you begin

Before you can catalog the data that you transfer with Amazon AppFlow, you must have a user role that you create with the AWS Identity and Access Management (IAM) service. This role grants Amazon AppFlow the permissions it needs to create Data Catalog tables, databases, and partitions.

For an example IAM policy that has the required permissions, see Identity-based policy examples for Amazon AppFlow (p. 156).

Cataloging flow output (Amazon AppFlow console)

Complete the following steps in the Amazon AppFlow console to create a flow that catalogs data.

1. Sign in to the AWS Management Console and open the Amazon AppFlow console at https://console.aws.amazon.com/appflow/.

2. To view the Data Catalog settings, you configure a flow that transfers data to Amazon S3. Do one of the following:
   - If you want to catalog the data from a new flow, choose Create flow and step through the flow creation process.
     When you get to the Configure flow page, under Destination details, set Destination name to Amazon S3.
   - If you want to catalog the data from an existing flow, choose Flows in the navigation pane to view your flows. Then, select the flow and choose Edit.
     On the Edit flow configuration page, under Destination details, ensure that Destination name is set to Amazon S3.

3. To view the Data Catalog settings, expand the AWS Glue Data Catalog settings section, and enable the Create a Data Catalog table checkbox.

4. Configure the following settings:
Data Catalog table names

Amazon AppFlow creates metadata tables in the Data Catalog that have a few different naming formats.

To query the latest data output from your flow, use the table with the name that appears in the following format:

- **prefix_appflow_flow-name_timestamp_latest**

  Amazon AppFlow updates this table continuously with the metadata from the most recent flow run.

  In this name, `timestamp` is the time when Amazon AppFlow created the table. The timestamp is formatted as a Unix epoch. For example, the timestamp for November 14, 2022 at 12:00:00 PM UTC is 1668456000.

To query historical versions of your data output, use the tables with names that appear in the following formats:

- **prefix_appflow_flow-name_schema-version**

  These tables contain metadata for each schema version.

- **prefix_appflow_flow-name_schema-version_execution-id**

  These tables contain metadata from individual flow runs. Amazon AppFlow creates these tables only when you set **Execution ID** as a partition key in the flow settings.

The variable elements in these names are as follows:

- **prefix** — The prefix that you specify in the flow settings.
- **flow-name** — The flow name. Amazon AppFlow modifies this name, if needed, to comply with table naming restrictions in the Data Catalog.
• `schema-version` — The version number of your data schema. Amazon AppFlow assigns this version number and increases it by one when you change any of the following settings for your flow:
  • Field mappings
  • Field data types
  • Partition settings
• `execution-id` — The ID that Amazon AppFlow assigns to a flow run. You can see these IDs in the run history for the flow.

**Example Data Catalog output from a flow run**

The following examples demonstrates how Amazon AppFlow catalogs a dataset by creating metadata tables in the Data Catalog.

**Example dataset**

The following table represents an example dataset of account records from a Salesforce database. The dataset is the source data for a flow that transfers from Salesforce to Amazon S3.

<table>
<thead>
<tr>
<th>Account Name</th>
<th>Account Type</th>
<th>Billing State/Province</th>
<th>Account Rating</th>
<th>Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example1</td>
<td>Customer - Direct</td>
<td>Anywhere</td>
<td>Hot</td>
<td>Apparel</td>
</tr>
<tr>
<td>Example2</td>
<td>Customer - Channel</td>
<td>Anywhere</td>
<td>Warm</td>
<td>Biotechnology</td>
</tr>
<tr>
<td>Example3</td>
<td></td>
<td>Anywhere</td>
<td>Cold</td>
<td>Construction</td>
</tr>
<tr>
<td>Example4</td>
<td>Customer - Direct</td>
<td>Anywhere</td>
<td></td>
<td>Consulting</td>
</tr>
<tr>
<td>Example5</td>
<td>Customer - Channel</td>
<td>Anywhere</td>
<td>Hot</td>
<td>Education</td>
</tr>
<tr>
<td>Example6</td>
<td>Customer - Channel</td>
<td>Anywhere</td>
<td>Warm</td>
<td>Electronics</td>
</tr>
<tr>
<td>Example7</td>
<td></td>
<td>Anywhere</td>
<td>Cold</td>
<td>Energy</td>
</tr>
<tr>
<td>Example8</td>
<td></td>
<td>Anywhere</td>
<td></td>
<td>Hospitality</td>
</tr>
<tr>
<td>Example9</td>
<td>Customer - Direct</td>
<td>Anywhere</td>
<td>Hot</td>
<td>Transportation</td>
</tr>
</tbody>
</table>

**Example flow configuration**

The flow that transfers the example dataset has the following configuration under AWS Glue Data Catalog settings in the console:

• **User role**, is set to `appflow-data-catalog-user-role`, which is an example name for a role that grants the required permissions to Amazon AppFlow.
• **Database** is set to `example-database`. 
• **Table name prefix** is set to `example-prefix`.

---

### Example Data Catalog table

When the flow runs, Amazon AppFlow creates tables in the database named `example-database`. One of these tables is named `example-prefix_appflow_exampleflow_1668036146_latest`, which Amazon AppFlow updates every time the flow runs. You can view the tables that Amazon AppFlow creates for your flows in the Data Catalog console.

The Data Catalog console provides details pages for each table. Each page show the metadata that a table stores, such as the columns and data types in the data schema. For more information, see Working with tables on the AWS Glue console in the *AWS Glue Developer Guide*. 
With the dataset cataloged, it's available to search and discover with data query tools and many AWS services. One way to query the data is to choose the **Table data** link on the database page in the Data Catalog console. That link opens Amazon Athena, which is an AWS service that runs SQL queries to help you analyze data in Amazon S3.

In Amazon Athena, the following SQL query retrieves the data that Amazon AppFlow catalogs in the example table:

```sql
SELECT * FROM "AwsDataCatalog"."example-database"."example-prefix_appflow_exampleflow_1668036146_latest" limit 10;
```

The Amazon Athena console shows the data that the query retrieves.
Partitioning and aggregating the data output from an Amazon AppFlow flow

When you use Amazon AppFlow to transfer data to Amazon S3 with a flow, you get the options to do the following:

- Organize the output data into partitions
- Aggregate the output records into files of a specified size

By using these settings, you can optimize query performance for applications that access the data.

**Partitioning and aggregating flow output (Amazon AppFlow console)**

Complete the following steps to configure the partition and aggregation settings in the Amazon AppFlow console.

2. To view the partition and aggregation settings, you configure a flow that transfers data to Amazon S3. Do one of the following:
   - If you want to configure the output from a new flow, choose **Create flow** and step through the flow creation process.
When you get to the **Configure flow** page, under **Destination details**, set **Destination name** to Amazon S3.

Continue the flow creation process. You configure the partition and aggregation settings when you get to the **Map data fields** page.

- If you want to configure the output from an existing flow, choose **Flows** in the navigation pane to view your flows. Then, select the flow and choose **Edit**.

On the **Edit flow configuration** page, under **Destination details**, ensure that **Destination name** is set to Amazon S3.

To configure the partition and aggregation settings, go to the **Edit data fields** page.

The console shows the settings under **Partition and aggregation settings**.

### Partition and aggregation settings

- **Partition settings**
  - **Date and time** — Represents the dates and times when your flow runs. You choose the precision (yearly, monthly, daily, and so on). The dates and times are shown as Coordinated Universal Time (UTC).

  Each unit of time (such as the year, month, or day) becomes a folder in your output file path. So, when you set the precision to daily, your path has folders for the years, months, and days when your flow runs. Those folders are nested in the path `year/month/day`, as in `2022/11/28`.

  If you choose the **Date and time** key, the **Execution ID** key is required and is selected automatically.

  - **Execution ID** — The ID that Amazon AppFlow assigns to the flow run. Your output file path in Amazon S3 includes a folder for the execution ID.

    If you configured your flow to catalog the output, then the Data Catalog tables also include the execution ID in their names. For more information about cataloging flow output, see **Cataloging the data output from an Amazon AppFlow flow** (p. 125).

  - **Destination fields** — The destination fields that you defined under **Source to destination field mapping** in the flow settings.

    If you choose this option, you can then specify up to 10 fields as partition keys. For each field that you choose, output records that have matching field-value pairs (for example, "BillingState" = "WA") are grouped together in the corresponding Amazon S3 folder.

---

3. For **Partition settings**, choose any of the following partition keys:

   - **Date and time** — Represents the dates and times when your flow runs. You choose the precision (yearly, monthly, daily, and so on). The dates and times are shown as Coordinated Universal Time (UTC).

   Each unit of time (such as the year, month, or day) becomes a folder in your output file path. So, when you set the precision to daily, your path has folders for the years, months, and days when your flow runs. Those folders are nested in the path `year/month/day`, as in `2022/11/28`.

   If you choose the **Date and time** key, the **Execution ID** key is required and is selected automatically.

   - **Execution ID** — The ID that Amazon AppFlow assigns to the flow run. Your output file path in Amazon S3 includes a folder for the execution ID.

     If you configured your flow to catalog the output, then the Data Catalog tables also include the execution ID in their names. For more information about cataloging flow output, see **Cataloging the data output from an Amazon AppFlow flow** (p. 125).

   - **Destination fields** — The destination fields that you defined under **Source to destination field mapping** in the flow settings.

     If you choose this option, you can then specify up to 10 fields as partition keys. For each field that you choose, output records that have matching field-value pairs (for example, "BillingState" = "WA") are grouped together in the corresponding Amazon S3 folder.
In your output file path, the destination field folders are nested in the order that you specify the partition keys. The folders have the path `partition key one=value/partition key two=value/partition key three=value`, and so on.

**Tip**
When you choose your partition keys, consider how they affect query performance for applications that access the data. For example, if you choose a granular partition key, such as `Account ID`, you might create many folders, where each folder contains one or just a few records. In that case, you incur processing delays that offset the benefit of partitioning.

4. For **Aggregation settings**, choose how to aggregate your records into output files in each partition

- **Don’t aggregate** — Don’t aggregate records into files of a specified size. The size of each output file is determined by one of the following:
  - The size of each input file
  - The page size of each API response in the data transfer operation
- **Aggregate all records into one file in each partition** — Write your records to a single file.
- **Aggregate records into multiple files in each partition** — Write your records to multiple files. For each file, Amazon AppFlow tries to achieve the target file size that you specify. The actual file sizes might differ from the target based on the number and size of the records that each file contains.

**Example file paths for partitioned datasets**

The following examples demonstrate how Amazon AppFlow imports source datasets and transfers them into partitioned datasets in Amazon S3.

**Example file paths**

Amazon AppFlow creates file paths in an S3 bucket, like the following example, when it runs a flow that’s configured with partition settings. The partitions in the following paths include schema versions, a date, execution IDs, and the destination fields `Account Rating` and `Industry`.

```
example-flow/
schemaVersion_1/
  520225fa-0ff4-4c95-b5d1-a2a8620812d7/
    Account Rating=Warm/
    Account Rating=null/
    Account Rating=Hot/
    Account Rating=Cold/
schemaVersion_2/
  2022/
    11/
      10/
        267c0ad0-228f-4d25-96fe-0f975005f6c6/
          Industry=Apparel/
          Industry=Biotechnology/
          Industry=Construction/
          Industry=Consulting/
          Industry=Education/
          Industry=Electronics/
```
Example dataset

Amazon AppFlow creates the example file paths when a flow transfers a source dataset that resembles the following example. The dataset contains customer account records from a Salesforce database. Each record has fields called Account Rating and Industry.

<table>
<thead>
<tr>
<th>Account Name</th>
<th>Account Rating</th>
<th>Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example1</td>
<td>Hot</td>
<td>Apparel</td>
</tr>
<tr>
<td>Example2</td>
<td>Warm</td>
<td>Biotechnology</td>
</tr>
<tr>
<td>Example3</td>
<td>Cold</td>
<td>Construction</td>
</tr>
<tr>
<td>Example4</td>
<td></td>
<td>Consulting</td>
</tr>
<tr>
<td>Example5</td>
<td>Hot</td>
<td>Education</td>
</tr>
<tr>
<td>Example6</td>
<td>Warm</td>
<td>Electronics</td>
</tr>
<tr>
<td>Example7</td>
<td>Cold</td>
<td>Energy</td>
</tr>
<tr>
<td>Example8</td>
<td></td>
<td>Hospitality</td>
</tr>
<tr>
<td>Example9</td>
<td>Hot</td>
<td>Transportation</td>
</tr>
</tbody>
</table>

Example flow configurations

The example file paths include two folders for schema version. Amazon AppFlow created the folder schemaVersion_1/ after the flow was initially defined and run. The initial flow configuration set the following partition settings:

- The Execution ID partition key was turned on.
- The Destination fields partition key was turned on, and the field Account Rating was used as a key.

With this configuration, Amazon AppFlow organized the output into datasets that contain records with matching field-value pairs for the Account Rating field. Amazon AppFlow stored each of these datasets in the corresponding folders, such as the folder Account Rating=Warm/.
Amazon AppFlow created the folder `schemaVersion_2/` after the partition settings in the flow were edited and the flow was run again. That revision set the following partition settings:

- The **Date and time** partition key was turned on, and the granularity was set to **Daily**.
- The **Destination fields** partition key was turned on, and the field **Industry** was used as a key.

With this configuration, Amazon AppFlow organized the output into filepaths for the year, month, and day that the flow ran: `2022/11/10`. Within that path, Amazon AppFlow organized the output into datasets that contain records with matching field-value pairs for the **Industry** field. Amazon AppFlow stored each of these datasets in the corresponding folders, such as the folder `Industry=Apparel/`.

### 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**

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**

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. Sign in to the AWS Management Console and 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. 135).

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 $T_0$, $T_1$, $T_2$, and so on, then records that are new or have changed between $T_0$ minus $t$ and $T_1$ minus $t$ will be imported from the source at $T_1$, and those that have changed between $T_1$ minus $t$ and $T_2$ minus $t$ will be imported from the source at $T_2$.

The total offset value can be longer than the schedule interval (for example, $t$ can be longer than $T_1$ minus $T_0$), but it must be less than 10 hours. The default value is 0.

- The flow run at $T_0$ transfers records that changed between $T_0$ minus 30 days and $T_0$ minus $t$ in the source application.
- The flow run at $T_1$ transfers records that changed between $T_0$ minus $t$ and $T_1$ minus $t$ in the source application.
- The flow run at $T_2$ transfers records that changed between $T_1$ minus $t$ and $T_2$ minus $t$ in the source application.
- The flow run at $T_3$ transfers records that changed between $T_2$ minus $t$ and $T_2$ 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.

![Private Flow Diagram]

**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**

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):

   ```json
   {
       "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.

<table>
<thead>
<tr>
<th>Name of the flow event</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppFlow Start Flow Run Report</td>
<td>start-time</td>
<td>The timestamp of the start of the flow run.</td>
</tr>
<tr>
<td>AppFlow Start Flow Run Report, AppFlow End Flow Run Report</td>
<td>incremental-transfer-time-range</td>
<td>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.</td>
</tr>
<tr>
<td>Name of the flow event</td>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AppFlow Event Flow Deactivated</td>
<td>deactivation-time</td>
<td>The time at which the flow was deactivated.</td>
</tr>
<tr>
<td>AppFlow Scheduled Flow Deactivated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AppFlow Event Flow Report</td>
<td>status-report</td>
<td>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.</td>
</tr>
<tr>
<td>AppFlow End Flow Run Report</td>
<td>end-time</td>
<td>The timestamp of the flow run completion.</td>
</tr>
<tr>
<td></td>
<td>num-of-records-processed</td>
<td>The number of records from the source that were processed by Amazon AppFlow.</td>
</tr>
<tr>
<td></td>
<td>num-of-record-failures</td>
<td>The number of records that could not be inserted into the destination.</td>
</tr>
<tr>
<td></td>
<td>data-processed</td>
<td>The volume of data (in bytes) that was processed.</td>
</tr>
<tr>
<td></td>
<td>status</td>
<td>The status that indicates if the flow run failed or was successful.</td>
</tr>
<tr>
<td></td>
<td>error</td>
<td>The reason for flow run failure in the event of a failed flow.</td>
</tr>
</tbody>
</table>
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 (p. 142)
- Identity and access management for Amazon AppFlow (p. 144)
- Compliance validation for Amazon AppFlow (p. 174)
- Resilience in Amazon AppFlow (p. 175)
- Infrastructure security in Amazon AppFlow (p. 175)

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. 138).

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.
### 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**
- Audience (p. 145)
- Authenticating with identities (p. 145)
- Managing access using policies (p. 147)
- How Amazon AppFlow works with IAM (p. 149)
- Managing permissions for Amazon AppFlow users (p. 154)
- Identity-based policy examples for Amazon AppFlow (p. 156)
- Service role policies for Amazon AppFlow (p. 162)
- Amazon S3 Bucket Policies for Amazon AppFlow (p. 166)
• AWS managed policies for Amazon AppFlow (p. 169)
• Troubleshooting Amazon AppFlow identity and access (p. 173)

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. 173).

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 service users 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. 149).

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. 156).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (successor to AWS Single Sign-On) (IAM Identity Center) users, your company’s single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the AWS Sign-In User Guide.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests using your credentials. If you don’t use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signature Version 4 signing process in the AWS General Reference.

Regardless of the authentication method that you use, you might 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 Multi-factor authentication in the AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide and Using multi-factor authentication (MFA) in AWS in the IAM User Guide.

AWS account root user

When you create an AWS account, you begin with one 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. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the AWS General Reference.

**Federated identity**

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A **federated identity** is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center (successor to AWS Single Sign-On). You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide.

**IAM users and groups**

An **IAM user** is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the IAM User Guide.

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:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the IAM User Guide. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about
permissions sets, see Permission sets in the AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide.

- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.

- **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 AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. 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. 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 the 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

<table>
<thead>
<tr>
<th>IAM feature</th>
<th>Amazon AppFlow support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity-based policies (p. 149)</td>
<td>Yes</td>
</tr>
<tr>
<td>Resource-based policies (p. 150)</td>
<td>No</td>
</tr>
<tr>
<td>Policy actions (p. 151)</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy resources (p. 151)</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy condition keys (p. 152)</td>
<td>Partial</td>
</tr>
<tr>
<td>ACLs (p. 153)</td>
<td>No</td>
</tr>
<tr>
<td>ABAC (tags in policies) (p. 153)</td>
<td>Yes</td>
</tr>
<tr>
<td>Temporary credentials (p. 153)</td>
<td>Yes</td>
</tr>
<tr>
<td>Principal permissions (p. 154)</td>
<td>Yes</td>
</tr>
<tr>
<td>Service roles (p. 154)</td>
<td>No</td>
</tr>
<tr>
<td>Service-linked roles (p. 154)</td>
<td>No</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Supports identity-based policies</th>
<th>Yes</th>
</tr>
</thead>
</table>

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.

<table>
<thead>
<tr>
<th>Required AWS KMS permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>kms:ListKeys</td>
<td>Controls permission to view the key ID and Amazon Resource Name (ARN) of all customer master keys (CMKs) in the account.</td>
</tr>
<tr>
<td>kms:DescribeKey</td>
<td>Controls permission to view detailed information about a CMK.</td>
</tr>
<tr>
<td>kms:ListAliases</td>
<td>Controls permission to view the aliases that are defined in the account. Aliases are optional friendly names that you can associate with CMKs.</td>
</tr>
<tr>
<td>kms:CreateGrant</td>
<td>Controls permission to add a grant to a CMK. You can use grants to add permissions without changing the key policy or IAM policy.</td>
</tr>
<tr>
<td>kms:ListGrants</td>
<td>Controls permission to view all grants for a CMK.</td>
</tr>
</tbody>
</table>

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. 156).

Resource-based policies within Amazon AppFlow

<table>
<thead>
<tr>
<th>Supports resource-based policies</th>
<th>No</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>Supports policy actions</th>
<th>Yes</th>
</tr>
</thead>
</table>

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. 156).

### Policy resources for Amazon AppFlow

<table>
<thead>
<tr>
<th>Supports policy resources</th>
<th>Yes</th>
</tr>
</thead>
</table>

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**

<table>
<thead>
<tr>
<th>Supports service-specific policy condition keys</th>
<th>Partial</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>Supports ACLs</th>
<th>No</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>Supports ABAC (tags in policies)</th>
<th>Yes</th>
</tr>
</thead>
</table>

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.

If a service supports all three condition keys for every resource type, then the value is Yes for the service. If a service supports all three condition keys for only some resource types, then the value is Partial.

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

<table>
<thead>
<tr>
<th>Supports temporary credentials</th>
<th>Yes</th>
</tr>
</thead>
</table>

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 (p. 154)
- Granting access for existing users (p. 155)
- Changing access levels (p. 155)
- Creating custom access policies (p. 156)

**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

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. 156).
   - 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

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. 156).
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

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. 156).
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, users and roles don't have permission to create or modify Amazon AppFlow resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (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 for users that require them.

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

For details about actions and resource types defined by Amazon AppFlow, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for Amazon AppFlow in the Service Authorization Reference.

Topics

• Policy best practices (p. 157)
• Example 1: Allow IAM users full administrator access to Amazon AppFlow (p. 157)
Identity-based policy examples

- Example 2: Allow IAM users read-only access to Amazon AppFlow (p. 159)
- Example 3: Grant access to permission-only actions (p. 160)
- Example 4: Allow users to view their own permissions (p. 160)
- Example 5: Allow Amazon AppFlow to access the AWS Glue Data Catalog (p. 161)

Policy best practices

Identity-based policies 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 with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the AWS managed policies that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the IAM User Guide.

- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as least-privilege permissions. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the IAM User Guide.

- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the IAM User Guide.

- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the IAM User Guide.

- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or root users in your account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the IAM User Guide.

For more information about best practices in IAM, see Security best practices in IAM 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-*"
},
{

Identity-based policy examples

```json
"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.

```json
{
  "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",
      }
   ]
}
```

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"
         ]
      }
   ]
}
```
Example 5: Allow Amazon AppFlow to access the AWS Glue Data Catalog

Before you can create a flow that catalogs its output data in the AWS Glue Data Catalog, you must grant Amazon AppFlow the required permissions. Amazon AppFlow requires permissions to create Data Catalog tables, databases, and partitions. To grant those permissions, you create an IAM role that contains the following permissions policy and trust policy. You provide this role to Amazon AppFlow in the settings for your flows.

Example permissions policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "glue:BatchCreatePartition",
        "glue:CreatePartitionIndex",
        "glue:DeleteDatabase",
        "glue:GetTableVersions",
        "glue:GetPartitions",
        "glue:BatchDeletePartition",
        "glue:DeleteTableVersion",
        "glue:UpdateTable",
        "glue:DeleteTable",
        "glue:DeletePartitionIndex",
        "glue:GetTableVersion",
        "glue:CreatePartition",
        "glue:UntagResource",
        "glue:UpdatePartition",
        "glue:TagResource",
        "glue:UpdateDatabase",
        "glue:CreateTable",
        "glue:BatchUpdatePartition",
        "glue:GetTables",
        "glue:BatchGetPartition",
        "glue:GetDatabases",
        "glue:GetPartitionIndexes",
        "glue:GetTable",
        "glue:GetDatabase",
        "glue:GetPartition",
        "glue:CreateDatabase",
        "glue:BatchDeleteTableVersion",
        "glue:BatchDeleteTable",
        "glue:DeletePartition"
      ],
      "Resource": "*"
    }
  ]
}
```

Example trust policy

```
{
  
}
```
Service role policies for Amazon AppFlow

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.

**Warning**
Changing the permissions for a service role might break Amazon AppFlow functionality. Edit service roles only when Amazon AppFlow provides guidance to do so.

Topics
- Allow Amazon AppFlow to access the AWS Glue Data Catalog (p. 162)
- Allow Amazon AppFlow to access Amazon Redshift databases with the Data API (p. 163)
- Allow Amazon Redshift to access your Amazon AppFlow data in Amazon S3 (p. 165)

Allow Amazon AppFlow to access the AWS Glue Data Catalog

Before you can create a flow that catalogs its output data in the AWS Glue Data Catalog, you must grant Amazon AppFlow the required permissions. Amazon AppFlow requires permissions to create Data Catalog tables, databases, and partitions. To grant the required permissions, you provide an IAM role that contains the following permissions policy and trust policy. You provide this role to Amazon AppFlow in the settings for your flows.

**Example permissions policy**

```json
{
  "Version": "2012-10-17",
  "Statement": [
    { 
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [ 
        "glue:BatchCreatePartition",
        "glue:CreatePartitionIndex",
        "glue:DeleteDatabase",
        "glue:GetTableVersions",
        "glue:GetPartitions",
        "glue:BatchDeletePartition",
        "glue:DeleteTableVersion",
        "glue:UpdateTable",
        "glue:DeleteTable",
        "glue:DeletePartitionIndex",
        "glue:GetTableVersion",
        "glue:CreatePartition",
        "glue:UntagResource",
        "glue:UpdatePartition",
```

162
Service role policies

```
"glue:TagResource",
"glue:UpdateDatabase",
"glue:CreateTable",
"glue:BatchUpdatePartition",
"glue:GetTables",
"glue:BatchGetPartition",
"glue:GetDatabases",
"glue:GetPartitionIndexes",
"glue:GetTable",
"glue:GetDatabase",
"glue:GetPartition",
"glue:CreateDatabase",
"glue:BatchDeleteTableVersion",
"glue:BatchDeleteTable",
"glue:DeletePartition"
],
"Resource": "*"
}
```

Example trust policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "appflow.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Allow Amazon AppFlow to access Amazon Redshift databases with the Data API

Before you can create a flow that transfers data to an Amazon Redshift database by using the Amazon Redshift Data API, you must grant Amazon AppFlow the required permissions. Amazon AppFlow requires permissions to do the following with your Amazon Redshift database:

- Gain access through temporary credentials
- Run SQL statements

To grant those permissions, you create an IAM role that contains the permissions policy and trust policy below. You provide this role to Amazon AppFlow in the settings for your Amazon Redshift connections.

Example permissions policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DataAPIPermissions",
      "Effect": "Allow",
      "Action": [
Service role policies

```
{
    "Sid": "GetCredentialsForAPIUser",
    "Effect": "Allow",
    "Action": "redshift:GetClusterCredentials",
    "Resource": [
        "arn:aws:redshift:*:*:dbname:*/*",
        "arn:aws:redshift:*:*:dbuser:*/*"
    ]
},
{
    "Sid": "GetCredentialsForServerless",
    "Effect": "Allow",
    "Action": "redshift-serverless:GetCredentials",
    "Resource": "*",
    "Condition":
        {
            "StringLike":
                {
                    "aws:ResourceTag/RedshiftDataFullAccess": "*"
                }
        }
},
{
    "Sid": "DenyCreateAPIUser",
    "Effect": "Deny",
    "Action": "redshift:CreateClusterUser",
    "Resource": [
        "arn:aws:redshift:*:*:dbuser:*/*"
    ]
},
{
    "Sid": "ServiceLinkedRole",
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "arn:aws:iam::*:role/aws-service-role/redshift-data.amazonaws.com/AWSServiceRoleForRedshift",
    "Condition":
        {
            "StringLike":
                {
                    "iam:AWSServiceName": "redshift-data.amazonaws.com"
                }
        }
}
```

Tag condition for Amazon Redshift Serverless resources

In the example permissions policy, the statement that grants the `redshift-serverless:GetCredentials` action has the following condition block:

```
"Condition":
    {
        "StringLike":
            {
                "aws:ResourceTag/RedshiftDataFullAccess": "*"
            }
    }
```
In IAM policies, condition is an optional element that specifies conditions for when a policy is in effect. With this condition block, the policy allows Amazon AppFlow to get temporary credentials for only those Amazon Redshift Serverless resources that meet the condition. To meet the condition, the resources must be tagged with the key RedshiftDataFullAccess. Therefore, to use this policy, you must apply that tag to the appropriate workgroup.

For more information about tagging resources in Amazon Redshift Serverless, see Tagging resources overview in the Amazon Redshift Management Guide.

Example trust policy

```
{
   "Version": "2012-10-17",
   "Statement": [ 
   {
       "Effect": "Allow",
       "Principal": {
           "Service": "appflow.amazonaws.com"
       },
       "Action": "sts:AssumeRole"
   }
   ]
}
```

For more information about authorizing access to the Data API, see Authorizing access to the Amazon Redshift Data API in the Amazon Redshift Management Guide.

Allow Amazon Redshift to access your Amazon AppFlow data in Amazon S3

When you run a flow that transfers data to an Amazon Redshift database, Amazon AppFlow first stores the data in an S3 bucket that you provide. Then, Amazon Redshift gets the data from the S3 bucket. For the flow to run successfully, you must authorize Amazon Redshift to get and decrypt the data. To grant those permission, you create an IAM role that contains the permissions policies and trust policy below.

You provide the IAM role in the settings when you create an Amazon Redshift connection in Amazon AppFlow.

You must also associate the role with the Amazon Redshift cluster that receives the data that you transfer with Amazon AppFlow. For the steps to associate the role, see Associating IAM roles with clusters in the Amazon Redshift Management Guide.

Example permissions policies

To provide the required permissions to Amazon Redshift, you can attach the following permissions policies to the IAM role:

- The AWS managed policy AmazonS3ReadOnlyAccess. This policy is owned and maintained by AWS. It grants read-only access to Amazon S3. To view the permissions for this policy, see AmazonS3ReadOnlyAccess in the AWS Management Console.
- A policy that permits Amazon Redshift to decrypt the encrypted data that Amazon AppFlow stores in Amazon S3, such as the following example:

```
{
   "Version": "2012-10-17",
   "Statement": [ 
   {
       "Effect": "Allow",
       "Action": "s3:GetObject"
   }]
}
```
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

```json
{}
```
"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:service::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 Redshift. This is required so that you can use Amazon Redshift 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.

• Lambda – Allows principals to list all the functions in customer account. This is required so that you can register new connectors

```json
{
    "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",
```
{
    "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:Equals": {
            "aws:CalledVia": ["appflow.amazonaws.com"]
        }
    }
},
{
    "Sid": "SecretsManagerPutResourcePolicyAccess",
    "Effect": "Allow",
    "Action": ["secretsmanager:PutResourcePolicy"],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:Equals": {
            "aws:CalledVia": ["appflow.amazonaws.com"]
        },
        "StringEqualsIgnoreCase": {
            "secretsmanager:ResourceTag/aws:secretsmanager:owningService": "appflow"
        }
    }
},
{
    "Sid": "LambdaListFunctions",
    "Effect": "Allow",
    "Action": ["lambda:ListFunctions"],
    "Resource": "*"
}
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:DescribeConnector",
        "appflow:DescribeConnectors",
        "appflow:DescribeConnectorProfiles",
        "appflow:DescribeFlows",
        "appflow:DescribeFlowExecution",
        "appflow:DescribeConnectorFields",
        "appflow:ListConnectors",
        "appflow:ListConnectorFields",
        "appflow:ListTagsForResource"
      ],
      "Resource": "*"
    }
  ],
  "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.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmazonAppFlowFullAccess (p. 169) – Update to an existing policy</td>
<td>Amazon AppFlow now allows the <code>lambda:ListFunctions</code> action in the AmazonAppFlowFullAccess policy.</td>
<td>03/01/2022</td>
</tr>
<tr>
<td>AmazonAppFlowReadOnlyAccess (p. 172) – Update to an existing policy</td>
<td>Amazon AppFlow now allows the <code>appflow:DescribeConnector</code> and <code>appflow:ListConnectors</code> actions in the AmazonAppFlowReadOnlyAccess policy.</td>
<td>03/01/2022</td>
</tr>
</tbody>
</table>
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 (p. 173)
- I am not authorized to perform iam:PassRole (p. 173)
- I want to view my access keys (p. 173)
- I'm an administrator and want to allow others to access Amazon AppFlow (p. 174)
- I want to allow people outside of my AWS account to access my Amazon AppFlow resources (p. 174)

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, your policies must be updated 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 that are 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's policies must be updated to allow her to perform the iam:PassRole action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

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, AKIAI0SF0DN7EXAMPLE) and a secret access key (for example, wJalrXUttnFEMI/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. 149).
- 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 an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. 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 on Amazon Web Services** – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.
  
  **Note**: Not all AWS services are HIPAA eligible. For more information, see the [HIPAA Eligible Services Reference](#).
- **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:
  - Events from Salesforce: Amazon AppFlow currently uses a third-party library, which is allocated a fixed buffer size of 10 MB. If a surge of events on a single event channel (such as AccountChangeEvent) exceeds the buffer size, then events might be dropped. You can request a larger buffer by filing a support case in the AWS Management Console. When you create the case, choose the Technical support case type. For Description, provide the ARN of your flow and the buffer size that you request. 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.

```json
{
    "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"
}
```
Monitoring Amazon AppFlow with Amazon CloudWatch

You can monitor your Amazon AppFlow flows by using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These statistics are kept for 15 months, so that you can access historical information and gain a better perspective on how your flows are performing. You can also set alarms that watch for certain thresholds, and send notifications or take actions when those thresholds are met. For more information, see the Amazon CloudWatch User Guide.

The Amazon AppFlow service reports the following metrics in the AWS/AppFlow namespace.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlowExecutionsStarted</td>
<td>The number of flow runs started.</td>
</tr>
<tr>
<td>FlowExecutionsFailed</td>
<td>The number of failed flow runs.</td>
</tr>
<tr>
<td>FlowExecutionsSucceeded</td>
<td>The number of successful flow runs.</td>
</tr>
<tr>
<td>FlowExecutionTime</td>
<td>The interval, in milliseconds, between the time the flow starts and the time it finishes.</td>
</tr>
<tr>
<td>FlowExecutionRecordsProcessed</td>
<td>The number of records that Amazon AppFlow attempted to transfer for the flow run. This metric counts all records that Amazon AppFlow processed internally. The processed records include those that transferred successfully and those that failed to transfer because the flow run failed.</td>
</tr>
</tbody>
</table>

The following dimensions are supported for the Amazon AppFlow metrics.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlowName</td>
<td>The name that you assigned to the flow.</td>
</tr>
</tbody>
</table>
# 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.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Redshift connector update (p. 181)</td>
<td>The Amazon Redshift connector in Amazon AppFlow is updated with new options to connect to your databases. Now you can connect to Amazon Redshift Serverless, and you can connect to public and private Amazon Redshift clusters. For more information, see [Amazon Redshift connector for Amazon AppFlow](p. 29).</td>
<td>November 21, 2022</td>
</tr>
<tr>
<td>CloudWatch metrics (p. 181)</td>
<td>Amazon AppFlow now reports metrics to Amazon CloudWatch. You can monitor these metrics to learn how your flows are performing. For more information, see [Monitoring Amazon AppFlow with Amazon CloudWatch](p. 180).</td>
<td>November 17, 2022</td>
</tr>
</tbody>
</table>
| Cataloging and organizing flow output (p. 181) | You can now use Amazon AppFlow to do the following with any flow that transfers data to Amazon S3:  
  • Catalog the data so that you can discover and access it from AWS analytics and machine learning services. For more information see, [Cataloging the data output from an Amazon AppFlow flow](p. 125).  
  • Organize the data into partitions and files. By organizing flow output, you improve query performance for applications that access the data. For more information see, [Partitioning and aggregating the data output from an Amazon AppFlow flow](p. 131). | November 15, 2022 |
<p>| Salesforce API preference (p. 181) | For flows that transfer data to or from Salesforce, you can now specify which Salesforce API | November 4, 2022 |</p>
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy connection feature (p. 181)</td>
<td>You can now use the Amazon AppFlow console to create a new connection by copying an existing one. For more information, see Managing Amazon AppFlow connections (p. 110).</td>
<td>September 15, 2022</td>
</tr>
<tr>
<td>Record deletion for Salesforce (p. 181)</td>
<td>For flows that transfer data to Salesforce, Amazon AppFlow now provides the option to delete records that you specify in a source data file. For more information, see the Notes (p. 72) section for the Salesforce connector.</td>
<td>September 14, 2022</td>
</tr>
<tr>
<td>New connector for Amazon Connect (p. 181)</td>
<td>This update adds information about the Amazon AppFlow connector for Amazon Connect. You can use Amazon AppFlow to transfer data from supported data sources to Amazon Connect Customer Profiles. For more information, see the section called “Amazon Connect” (p. 23).</td>
<td>September 14, 2022</td>
</tr>
<tr>
<td>New connector for Jira Cloud (p. 181)</td>
<td>Amazon AppFlow now provides a connector that you can use to transfer data from Jira Cloud. For more information, see the section called “Jira Cloud” (p. 52).</td>
<td>August 29, 2022</td>
</tr>
<tr>
<td>New tutorial for data transfers (p. 181)</td>
<td>The Amazon AppFlow User Guide now includes a tutorial that you can use to transfer data from Amazon S3 to Salesforce, and from Salesforce to Amazon S3. For more information, see Tutorial: Transfer data between applications with Amazon AppFlow (p. 8).</td>
<td>August 23, 2022</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>SAP OData connector now supports ODP (p. 181)</td>
<td>With the SAP OData connector, you can now connect Amazon AppFlow to SAP applications that use the Operational Data Provisioning (ODP) framework. When you connect to ODP providers, you can create flows that run full or incremental data transfers. Incremental flows subscribe to delta updates from the operational delta queue of your ODP provider. For more information, see SAP OData connector for Amazon AppFlow (p. 78).</td>
<td>August 11, 2022</td>
</tr>
<tr>
<td>New connector for Zendesk Sell (p. 181)</td>
<td>Amazon AppFlow now provides a connector that you can use to transfer data from Zendesk Sell. For more information, see Zendesk Sell connector for Amazon AppFlow (p. 107).</td>
<td>August 11, 2022</td>
</tr>
<tr>
<td>New connector for Zendesk Chat (p. 181)</td>
<td>Amazon AppFlow now provides a connector that you can use to transfer data from Zendesk Chat. For more information, see Zendesk Chat connector for Amazon AppFlow (p. 104).</td>
<td>August 11, 2022</td>
</tr>
<tr>
<td>New connector for Mixpanel (p. 181)</td>
<td>Amazon AppFlow now provides a connector that you can use to transfer data from Mixpanel. For more information, see Mixpanel connector for Amazon AppFlow (p. 60).</td>
<td>June 16, 2022</td>
</tr>
<tr>
<td>New connector for Google Ads (p. 181)</td>
<td>Amazon AppFlow now provides a connector that you can use to transfer data from Google Ads. For more information, see Google Ads connector for Amazon AppFlow (p. 45).</td>
<td>June 16, 2022</td>
</tr>
<tr>
<td>New connector for Facebook Ads (p. 181)</td>
<td>Amazon AppFlow now provides a connector that you can use to transfer data about your Facebook ads. For more information, see Facebook Ads connector for Amazon AppFlow (p. 42).</td>
<td>June 16, 2022</td>
</tr>
<tr>
<td>Change Description</td>
<td>Details</td>
<td>Date</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>New connector for Salesforce Marketing Cloud (p. 181)</td>
<td>Amazon AppFlow now provides a connector you can use to transfer data from Salesforce Marketing Cloud. For more information, see Salesforce Marketing Cloud connector for Amazon AppFlow (p. 73).</td>
<td>June 9, 2022</td>
</tr>
<tr>
<td>Support for managing connections (p. 181)</td>
<td>This update documents how to manage connections to provide the configuration details and credentials that Amazon AppFlow uses to transfer data with your applications. For more information see Managing Amazon AppFlow connections (p. 110).</td>
<td>March 8, 2022</td>
</tr>
<tr>
<td>Updated IAM policies (p. 181)</td>
<td>This update adds new permissions to the AWS managed policies AmazonAppFlowFullAccess and AmazonAppFlowReadOnlyAccess. For more information, see AWS managed policies for Amazon AppFlow (p. 169).</td>
<td>March 1, 2022</td>
</tr>
<tr>
<td>New documentation (p. 181)</td>
<td>This update adds the following procedures to help you get started with Amazon AppFlow: Create a flow using the AWS console, Create a flow using the AWS CLI, Create a flow using the Amazon AppFlow APIs, and Create a flow using CloudFormation resources. This update also adds a compatibility matrix for Amazon AppFlow connectors under Supported Amazon AppFlow Connectors.</td>
<td>January 31, 2022</td>
</tr>
<tr>
<td>Support for SAP OData as a destination (p. 181)</td>
<td>You can now use SAP OData as a destination. For more information, see SAP OData.</td>
<td>January 26, 2022</td>
</tr>
<tr>
<td>Support for Marketo as a destination (p. 181)</td>
<td>You can now use Marketo as a destination. For more information, see Marketo.</td>
<td>May 25, 2021</td>
</tr>
<tr>
<td>Updated IAM documentation (p. 181)</td>
<td>The Amazon AppFlow User Guide now includes an enhanced IAM documentation chapter, and has started tracking changes for its AWS managed policies.</td>
<td>March 26, 2021</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Support for Zendesk as a destination (p. 181)</td>
<td>You can now use Zendesk as a destination. For more information, see Zendesk.</td>
<td>March 22, 2021</td>
</tr>
<tr>
<td>API support for Amazon Lookout for Metrics (p. 181)</td>
<td>The Amazon AppFlow API Reference now includes the following data type for Amazon Lookout for Metrics: LookoutMetricsDestinationProperties.</td>
<td>February 24, 2021</td>
</tr>
<tr>
<td>API support for Amazon Honeycode (p. 181)</td>
<td>The Amazon AppFlow API Reference now includes the following data types for Amazon Honeycode: HoneycodeConnectorProfileCredentials, HoneycodeConnectorProfileProperties, HoneycodeDestinationProperties, and HoneycodeMetadata.</td>
<td>February 24, 2021</td>
</tr>
<tr>
<td>API support for Amazon Connect Customer Profiles (p. 181)</td>
<td>The Amazon AppFlow API Reference now includes the following data types for Amazon Connect Customer Profiles: CustomerProfilesDestinationProperties and CustomerProfilesMetadata.</td>
<td>February 24, 2021</td>
</tr>
<tr>
<td>Application-specific User Guide pages (p. 181)</td>
<td>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.</td>
<td>January 6, 2021</td>
</tr>
<tr>
<td>Support for Salesforce Pardot as a source (p. 181)</td>
<td>You can now use Salesforce Pardot as a source. For more information, see Salesforce Pardot.</td>
<td>December 18, 2020</td>
</tr>
<tr>
<td>Support for Amazon Lookout for Metrics as a destination (p. 181)</td>
<td>You can now use Amazon Lookout for Metrics as a destination. For more information, see Amazon Lookout for Metrics.</td>
<td>December 8, 2020</td>
</tr>
<tr>
<td>Schedule-triggered flow settings (p. 181)</td>
<td>You can now specify a time offset when configuring incremental data transfer for schedule-triggered flows. For more information, see Incremental transfer.</td>
<td>December 4, 2020</td>
</tr>
<tr>
<td>Support for Amazon Honeycode as a destination (p. 181)</td>
<td>You can now use Amazon Honeycode as a destination. For more information, see Amazon Honeycode.</td>
<td>December 1, 2020</td>
</tr>
<tr>
<td>Feature Description</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Support for Upsolver as a destination (p. 181)</td>
<td>You can now use Upsolver as a destination. For more information, see Upsolver.</td>
<td>November 20, 2020</td>
</tr>
<tr>
<td>Support for Salesforce global connected apps (p. 181)</td>
<td>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.</td>
<td>November 10, 2020</td>
</tr>
<tr>
<td>Support for updating records in Salesforce (p. 181)</td>
<td>You can now update existing records when you use Salesforce as a destination. For more information, see Salesforce, Notes.</td>
<td>October 21, 2020</td>
</tr>
<tr>
<td>Support for Google Analytics custom dimensions and metrics (p. 181)</td>
<td>You can now import custom dimensions and metrics from Google Analytics into Amazon S3. For more information, see Google Analytics, Notes.</td>
<td>October 21, 2020</td>
</tr>
<tr>
<td>Support for upserting and inserting records in Salesforce (p. 181)</td>
<td>You can now insert new records or upsert records when you use Salesforce as a destination. For more information, see Salesforce, Notes.</td>
<td>October 5, 2020</td>
</tr>
<tr>
<td>Schedule-triggered flow settings (p. 181)</td>
<td>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.</td>
<td>October 5, 2020</td>
</tr>
<tr>
<td>AWS CloudFormation support (p. 181)</td>
<td>Amazon AppFlow now supports AWS CloudFormation. For more information, see Related AWS services, AWS CloudFormation.</td>
<td>September 17, 2020</td>
</tr>
<tr>
<td>Support for Amazon EventBridge as a destination (p. 181)</td>
<td>Amazon AppFlow now supports Amazon EventBridge as a flow destination. For more information, see Amazon EventBridge.</td>
<td>August 26, 2020</td>
</tr>
<tr>
<td>Amazon AppFlow API Reference (p. 181)</td>
<td>You can now reference the API operations used with Amazon AppFlow. For more information, see the Amazon AppFlow API Reference.</td>
<td>August 26, 2020</td>
</tr>
<tr>
<td>Support for new data formats (CSV, Parquet) (p. 181)</td>
<td>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.</td>
<td>August 14, 2020</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>Improved filter support (p. 181)</td>
<td>You can now add criteria to your filters and apply multiple filters to a flow. For more information, see Add filters.</td>
<td>August 10, 2020</td>
</tr>
<tr>
<td>Connect over PrivateLink to Salesforce (p. 181)</td>
<td>Amazon AppFlow now supports connections over PrivateLink. For more information, see Private Amazon AppFlow flows.</td>
<td>July 22, 2020</td>
</tr>
<tr>
<td>CloudWatch integration documentation (p. 181)</td>
<td>Amazon AppFlow now supports CloudWatch Event integration. For more information, see Flow notifications.</td>
<td>July 17, 2020</td>
</tr>
<tr>
<td>Additional Amazon S3 destination settings (p. 181)</td>
<td>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.</td>
<td>July 10, 2020</td>
</tr>
<tr>
<td>IAM managed policies (p. 181)</td>
<td>Amazon AppFlow now supports IAM managed policies. For more information, see Identity and access management for Amazon AppFlow.</td>
<td>July 3, 2020</td>
</tr>
<tr>
<td>Google Analytics service quota (p. 181)</td>
<td>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.</td>
<td>June 23, 2020</td>
</tr>
<tr>
<td>Initial release (p. 181)</td>
<td>Initial release of the Amazon AppFlow User Guide.</td>
<td>April 22, 2020</td>
</tr>
</tbody>
</table>