**Building a Data Flows Between AWS and Salesforce Using Amazon AppFlow**

**Amazon AppFlow** is a fully managed no-code integration service enabling seamless and secure data flow between Amazon Web Services (AWS) and software-as-a-service (SaaS) applications. It allows you to source data from AWS services and **SaaS applications** such as Salesforce, and aggregate them in AWS data lakes and data warehouses to draw unique data-driven insights.

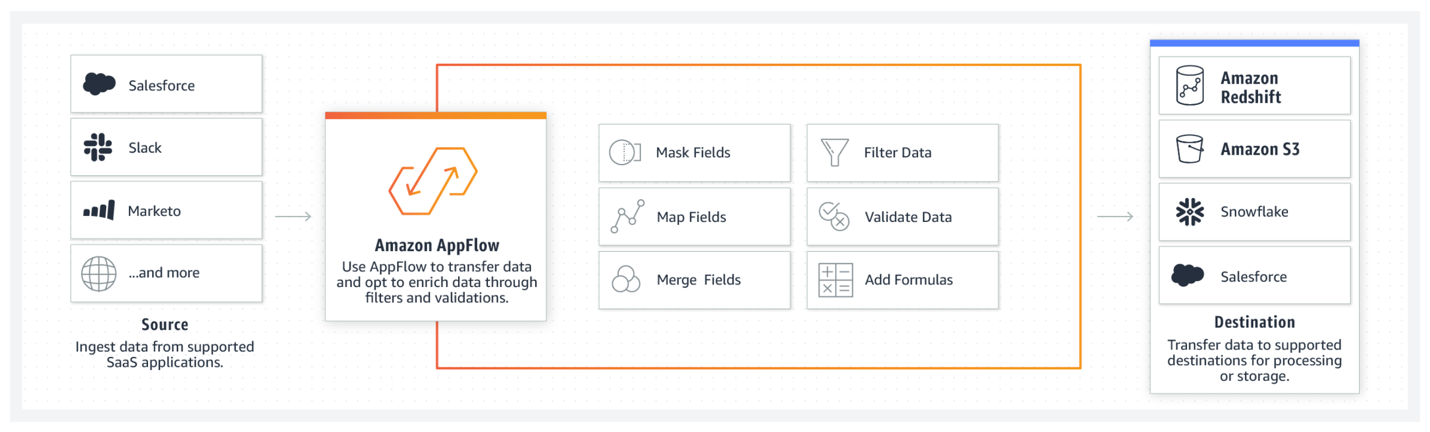
Amazon AppFlow also allows you to create triggers that move data automatically. If you want to copy **Salesforce** data into **Amazon Simple Storage Service** (Amazon S3) for durable and low-cost backup of customer relationship management (CRM) data.

It also helps a sales operations manager who wants to ingest opportunity records from **Salesforce** into **Amazon Redshift** to combine it with data from different sources and update their dashboard.

A Salesforce admin who wants to insert new lead records in Salesforce using campaign data stored in CSV file in Amazon S3 can also benefit from Amazon AppFlow.

With Amazon AppFlow moving data back and forth between Salesforce and AWS is simple and easy process that only takes a few minutes to set up.

**Are You Thinking…How Does Amazon AppFlow Work?**

****

It has built-in connection logic that can pull data from a supported source to a supported destination. The supported source and destination could be a SaaS application or an AWS service such as Amazon S3 or Amazon Redshift.

A flow describes how data is moved, including the source, destination, any information needed to authenticate user access to source or destination, flow triggers (on demand, events, or a schedule), and data processing tasks such as check-pointing, field validation, or masking.

When triggered, Amazon AppFlow executes a flow run, which call APIs to get data from the source, runs data processing tasks, and transfers processed data to the destination.

**Excited to Run Flows**

You can run flows on demand, scheduled, or triggered by an event such as launching a campaign, converting a lead, closing an opportunity, or opening a case. You can also specify filters to narrow the records you transfer, add transformations to fields by specifying the formula, mask sensitive values in fields, and validate records before sending them to the destination.

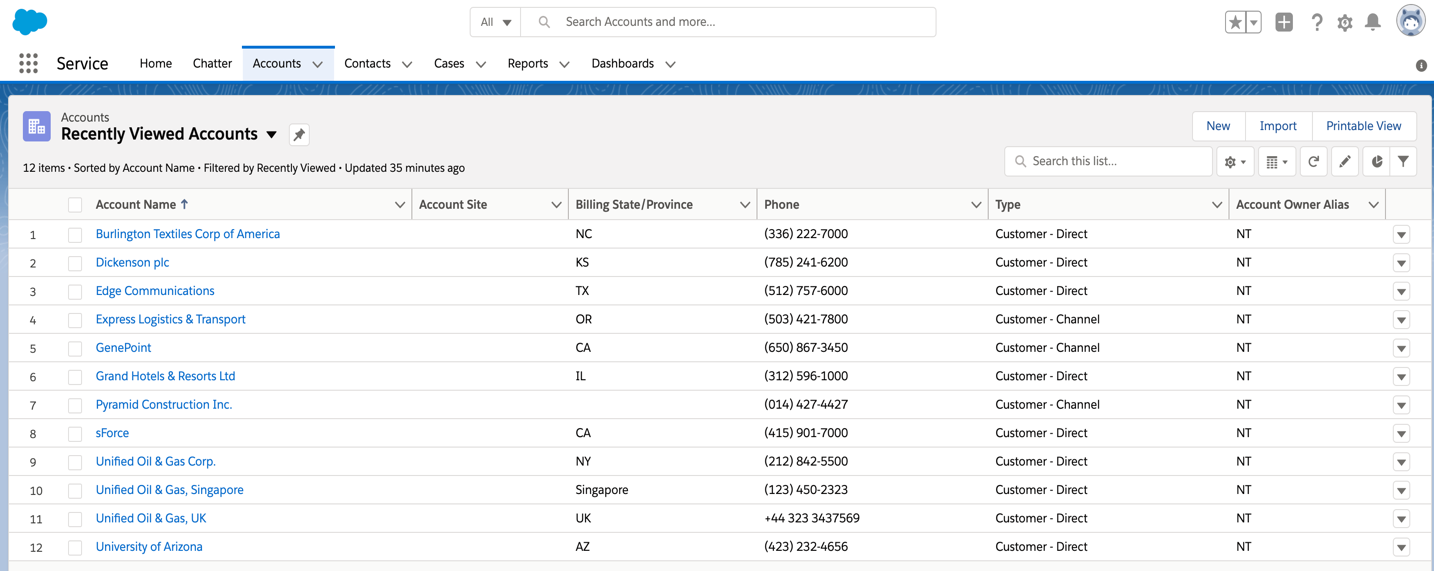
**Set Up Amazon AppFlow for Salesforce Users**

The following procedure sets up Amazon AppFlow for users with certain versions of Salesforce and the proper permissions in their AWS account. For this walkthrough, we will be using Salesforce.com Developer Edition account. You can [sign up for your own](https://developer.salesforce.com/) developer account at Salesforce.

Prerequisites

To access Salesforce from Amazon AppFlow, your Salesforce edition must provide API access. Salesforce Enterprise, Unlimited, Developer, and Performance Edition provides API access whereas Group, Essentials and Professional Edition does not, [find more information](https://help.salesforce.com/articleView?id=000326486&type=1&mode=1) about Salesforce Editions with API access.

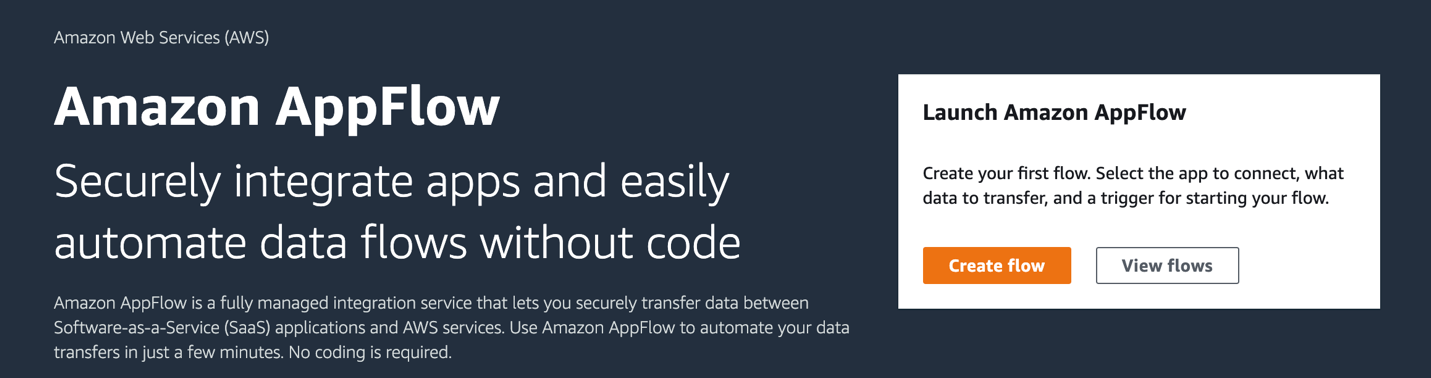
Salesforce Accounts:



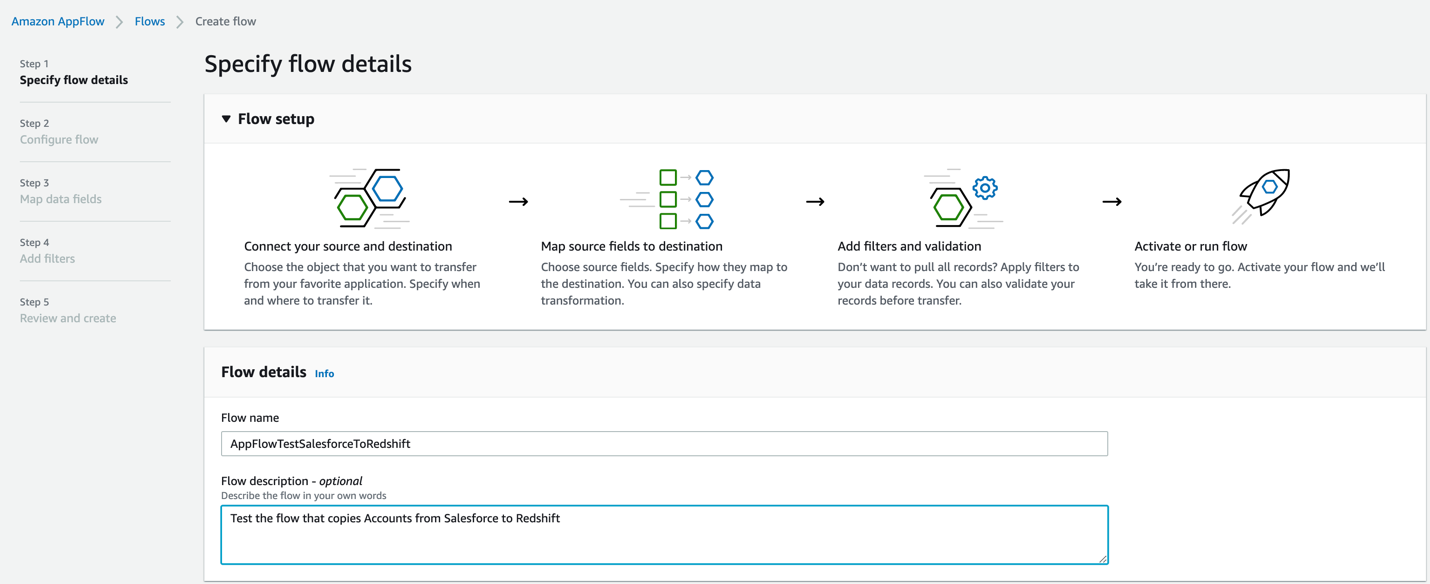
**Let’s get our hands dirty**

**Creating our First Flow**

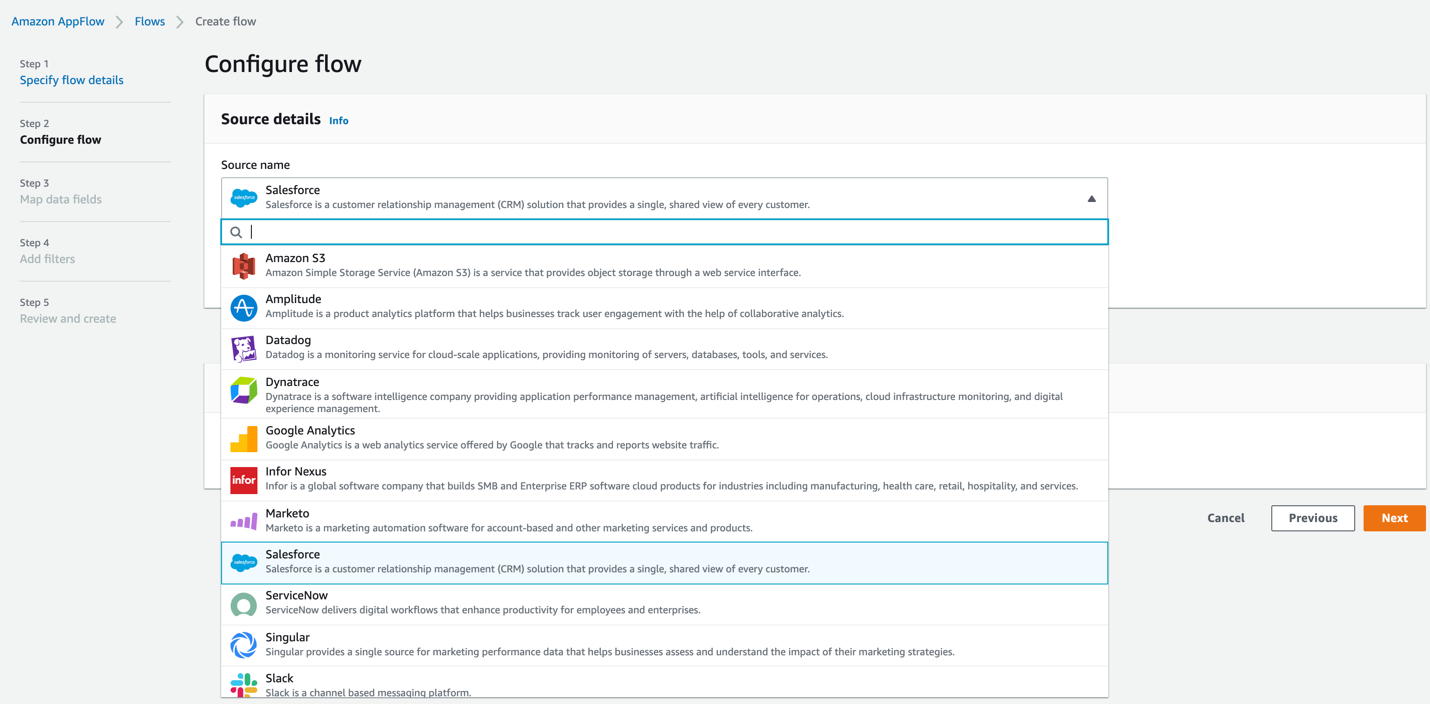
* First, of course you need to sign in to AWS Account and hit the AppFlow Dashboard.



* Click on the bright orange “Create flow” button, it will take you to the first step on creating our first flow. For this flow, I decided the name to be “AppFlowTestSalesforceToRedshift” and move on to the next step without setting any of the optional settings.

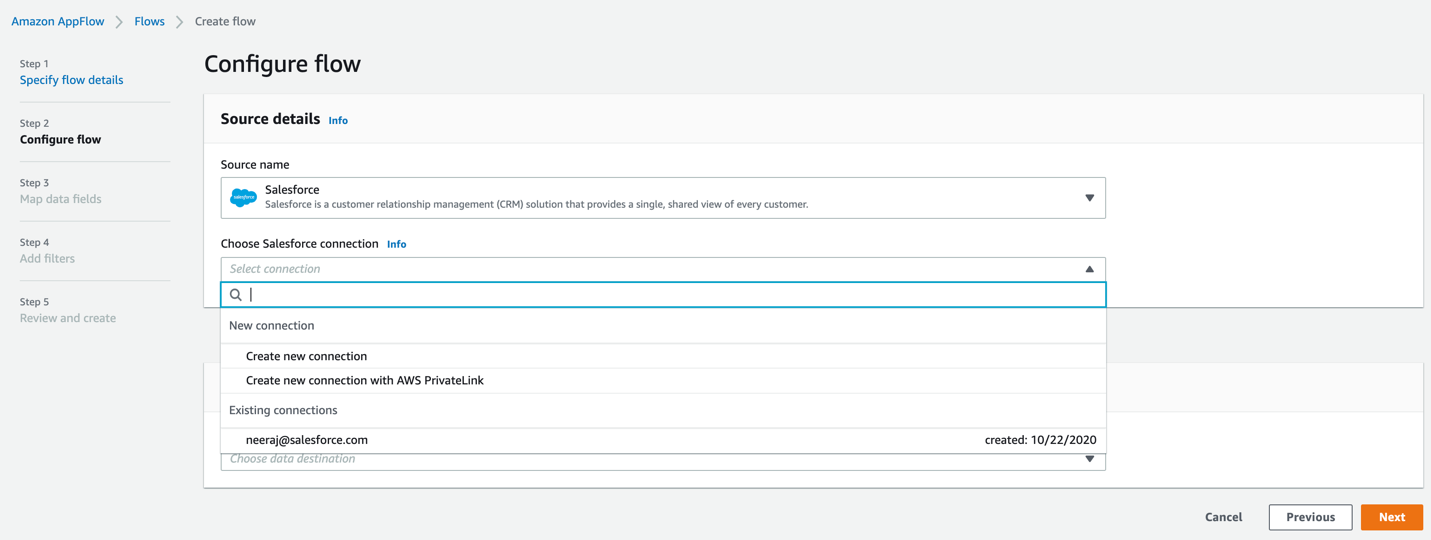


* On the Configure flow screen, under Source name, select Salesforce from the pull-down menu.

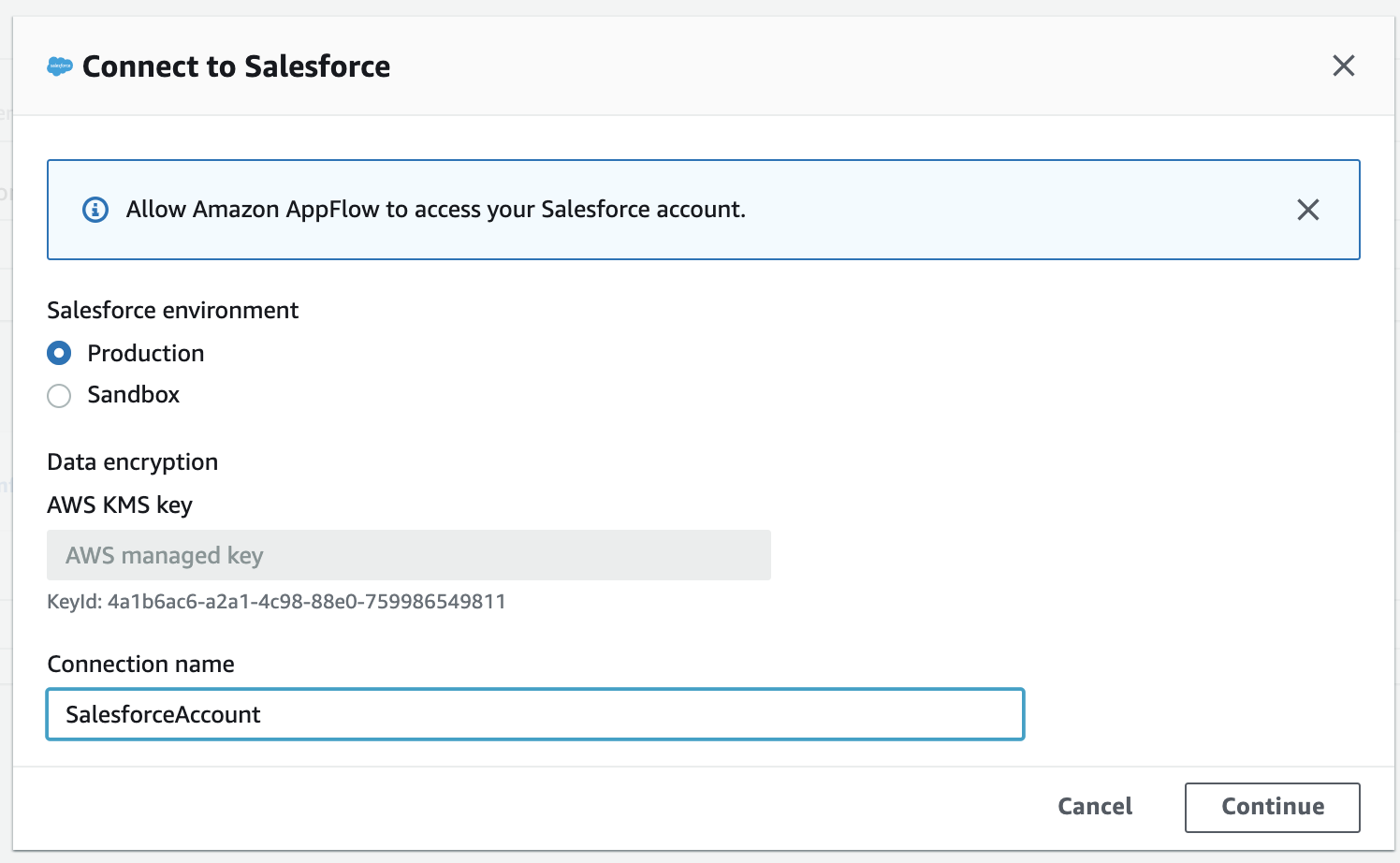


* In the Choose Salesforce Connection field, select Create new connection from the pull-down menu to connect to your Salesforce account.

The new connection is saved for future use and appears under the Existing connections field.

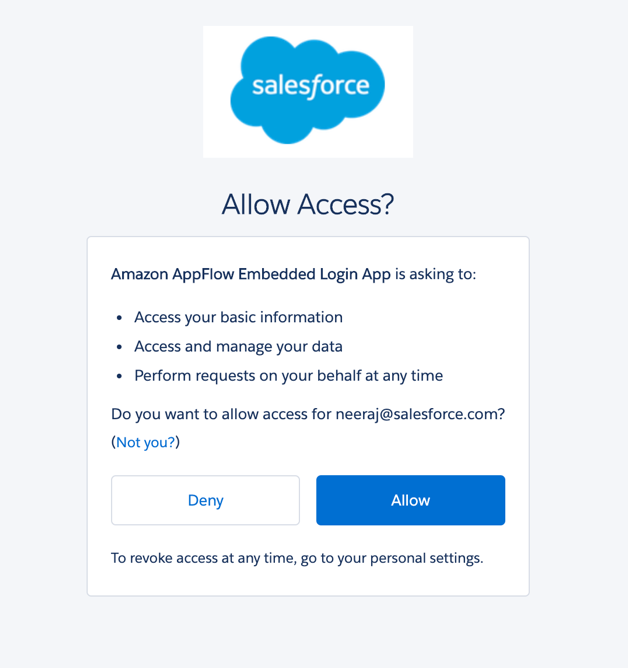


* Once you select “Create new connection”, a new window opens up asking you to choose Production or Sandbox environment.

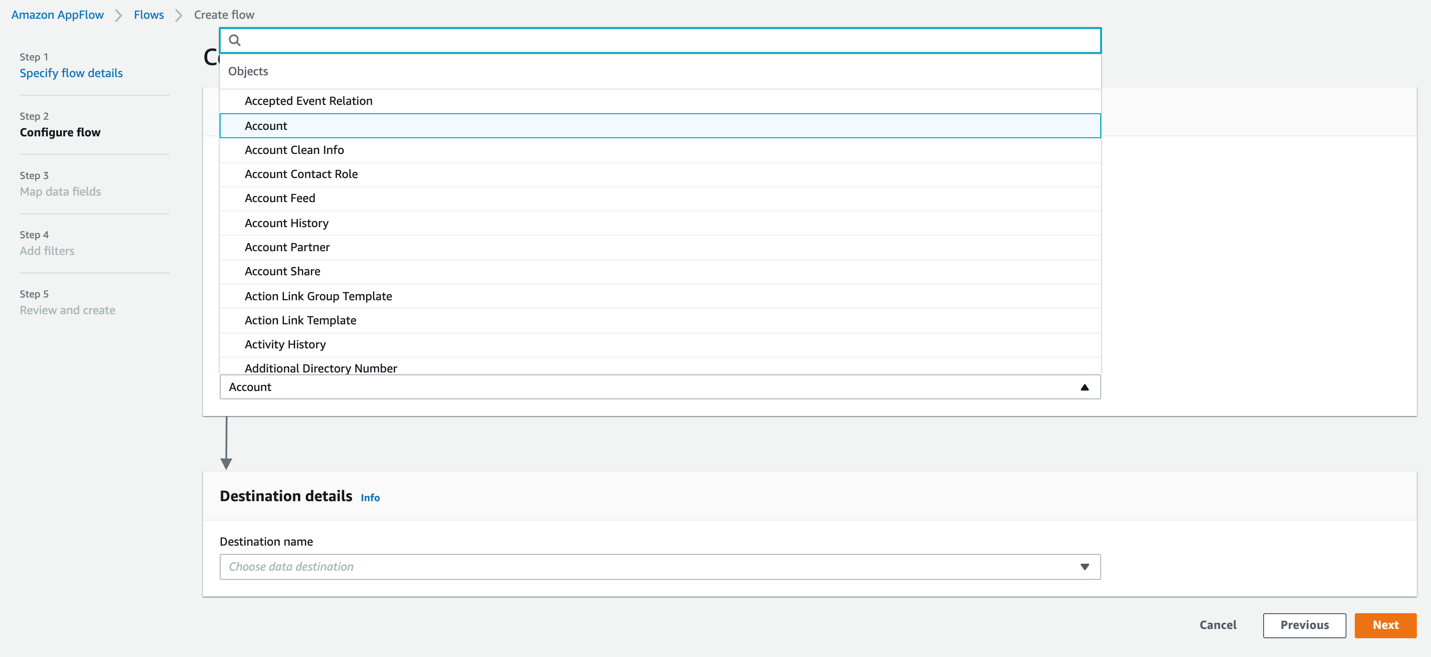


Note that a sandbox account has to be provisioned by a Salesforce Administrator.

* Name the new connection, and select Continue to go to the Salesforce login window.
* Select the username that is displayed, or log in with a different one.
* Select Allow to allow Amazon AppFlow to access your Salesforce data.



Once you select Allow, your Salesforce objects and Salesforce events appear in your Amazon AppFlow account.



Then I picked “Amazon Redshift” as my destination, hold on hold on before you choose Redshift as your destination make sure to follow the below steps as Amazon AppFlow uses the Redshift COPY command to move data into Amazon Redshift using an Amazon S3 buckets. In simple words, the data is stored in a S3 and then ingested into Amazon Redshift.

To set this up, follow the below steps.

1. Create or use and existing Amazon S3 bucket in your AWS account.
2. Create an IAM role for Amazon Redshift that grants AmazonS3ReadOnlyAccess and attach it to your Amazon Redshift Cluster.
3. Create an Amazon Redshift Cluster for storing your data, and make sure to attach the IAM role you created to this cluster.
4. Ensure your Amazon Redshift cluster is publicly accessible from Amazon AppFlow IP address in your region.

Open the Amazon Redshift console, choose the cluster to modify, and then properties.

In the Network and security section, choose the link next to the VPC security groups to open the Amazon EC2 console.

On the Inbound Rules tab, be sure that the port of your Amazon Redshift cluster is allowed.

In the Amazon Redshift console, navigate to Clusters > Properties. In the Network and security section, scroll to the end of the page, and select Yes for Publicly accessible.

1. Create a table in Amazon Redshift. In the left navigation bar, select Editor.

In the Data objects pane, select a schema from the pull-down menu under Select schema.

In the Query editor, create a query for your example. You can create a different query with different fields, if you prefer.

CREATE TABLE accounts (account\_name varchar(200),

account\_site varchar(200),

billing\_state\_province varchar(200),

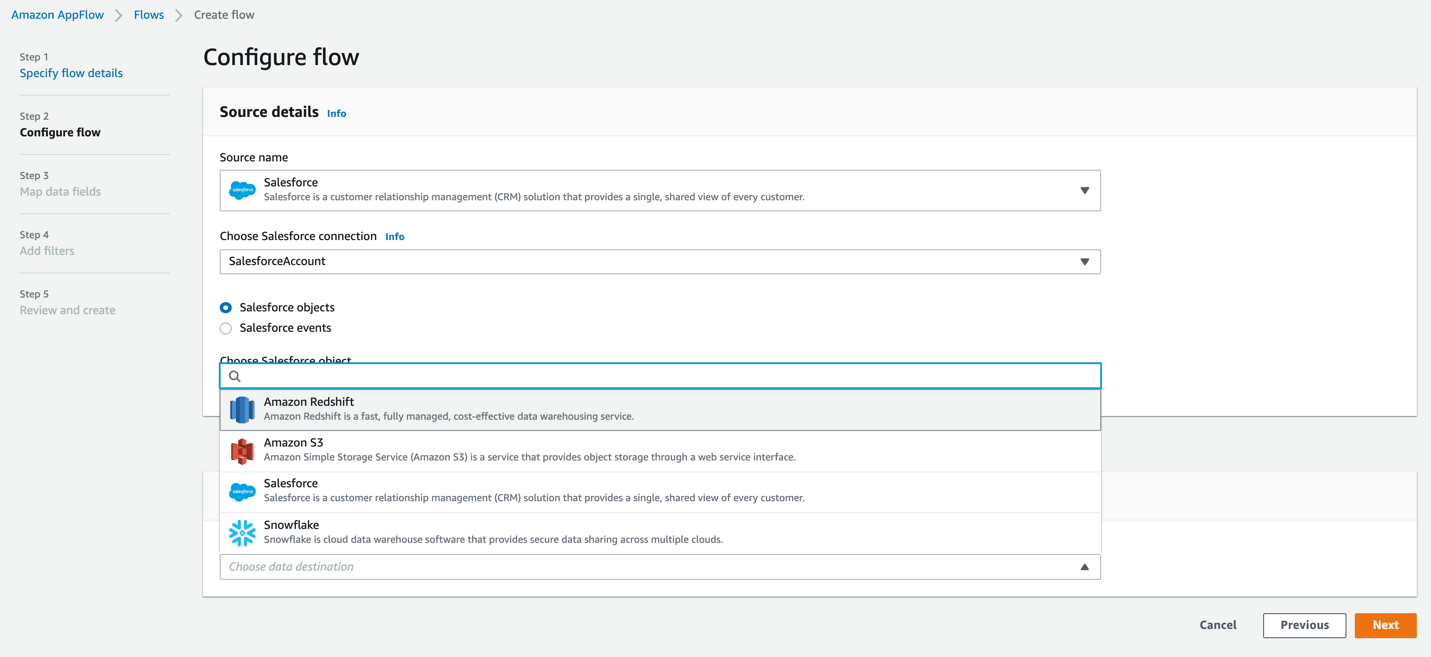
phone varchar(200),

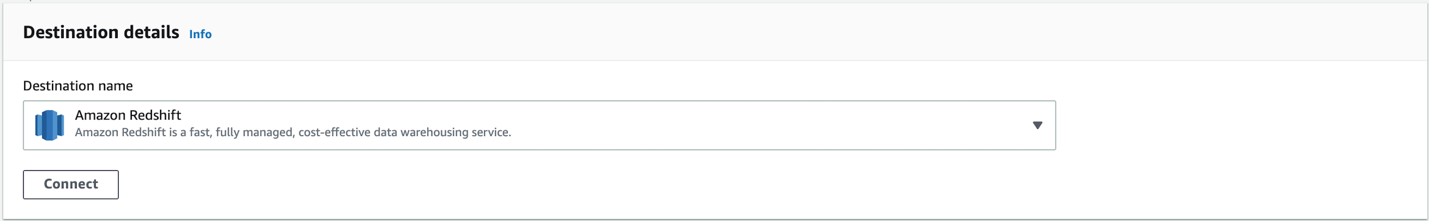
type varchar(200));

Select the Run button to create a table with account\_name, account\_site, billing\_state\_province, phone, type.

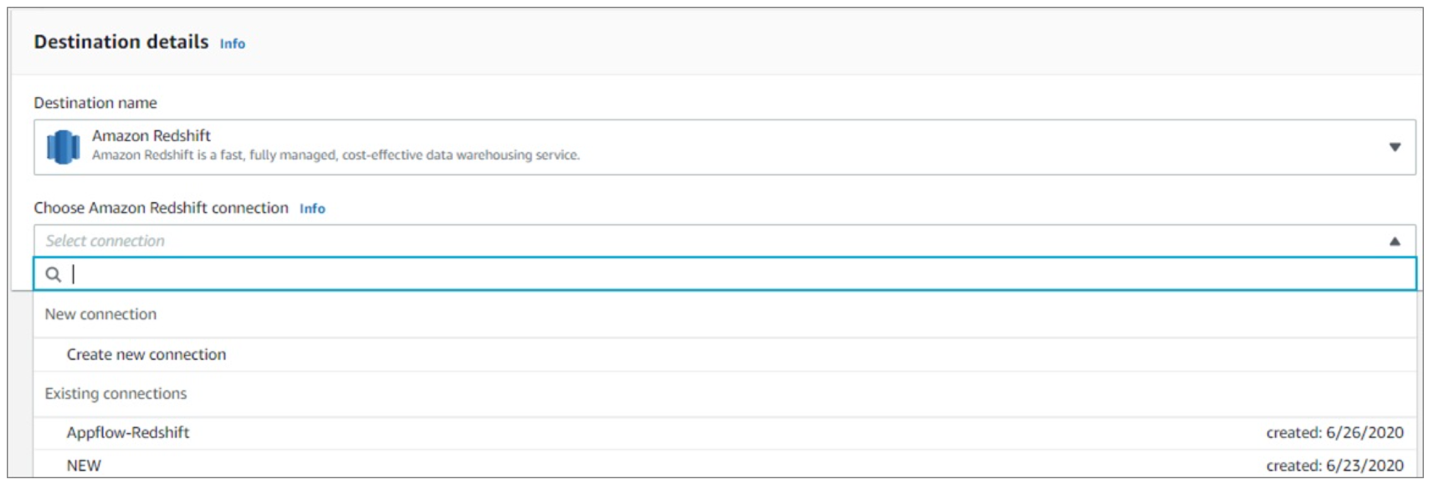
Now, once you have created the Amazon Redshift cluster, created the table and make sure that the Redshift cluster is accessible you can select the destination as Amazon Redshift in Amazon AppFlow “Configure flow” step which you left backwards.

* In the Amazon AppFlow, select Amazon Redshift as the destination in the Destination name. Then select Connect Immediately below it.

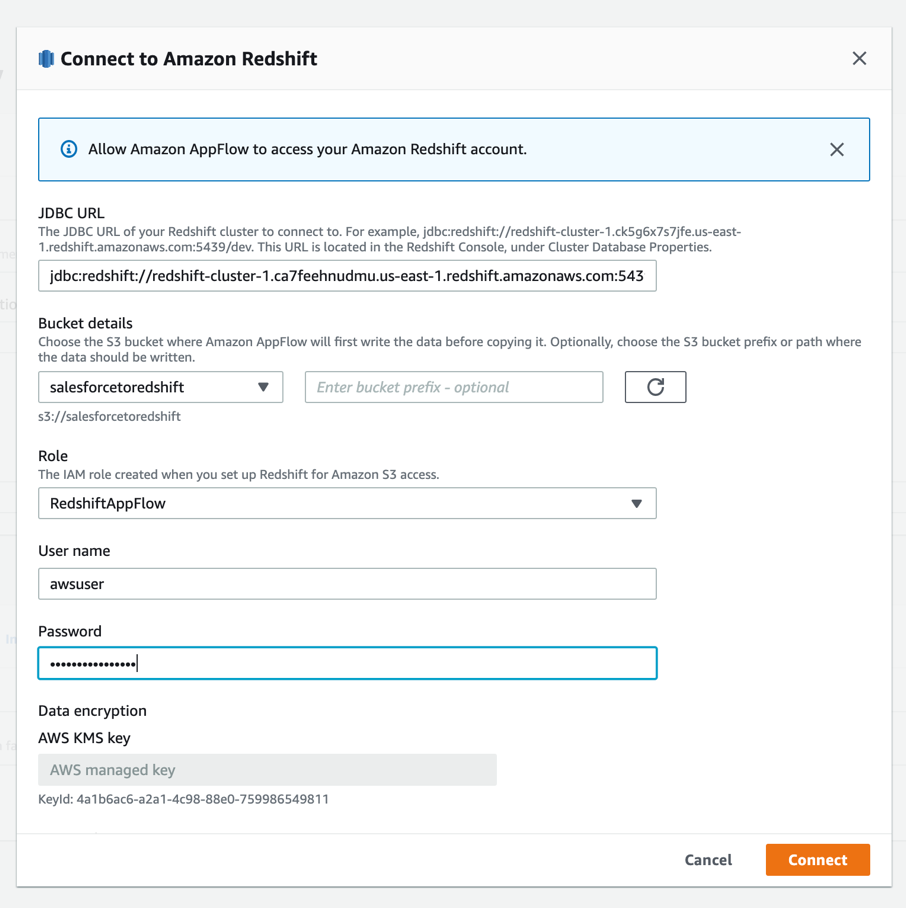




* Under Choose Amazon Redshift connection, select Create new connection from the drop-down menu.

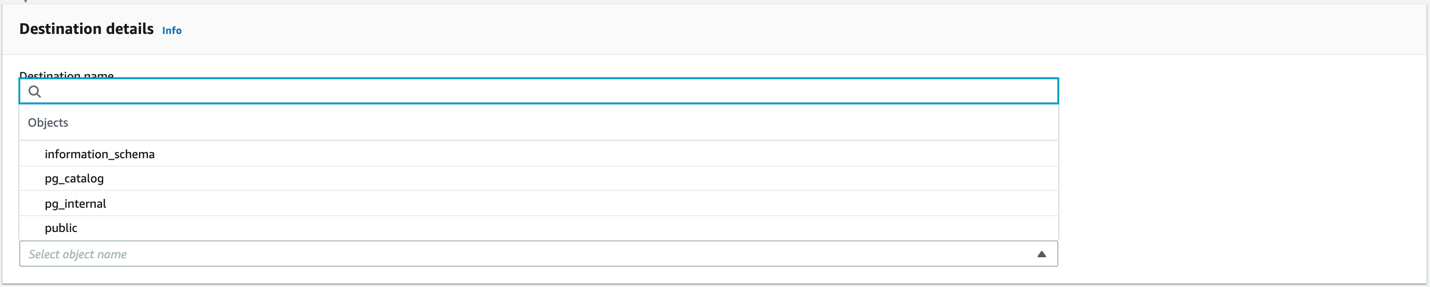


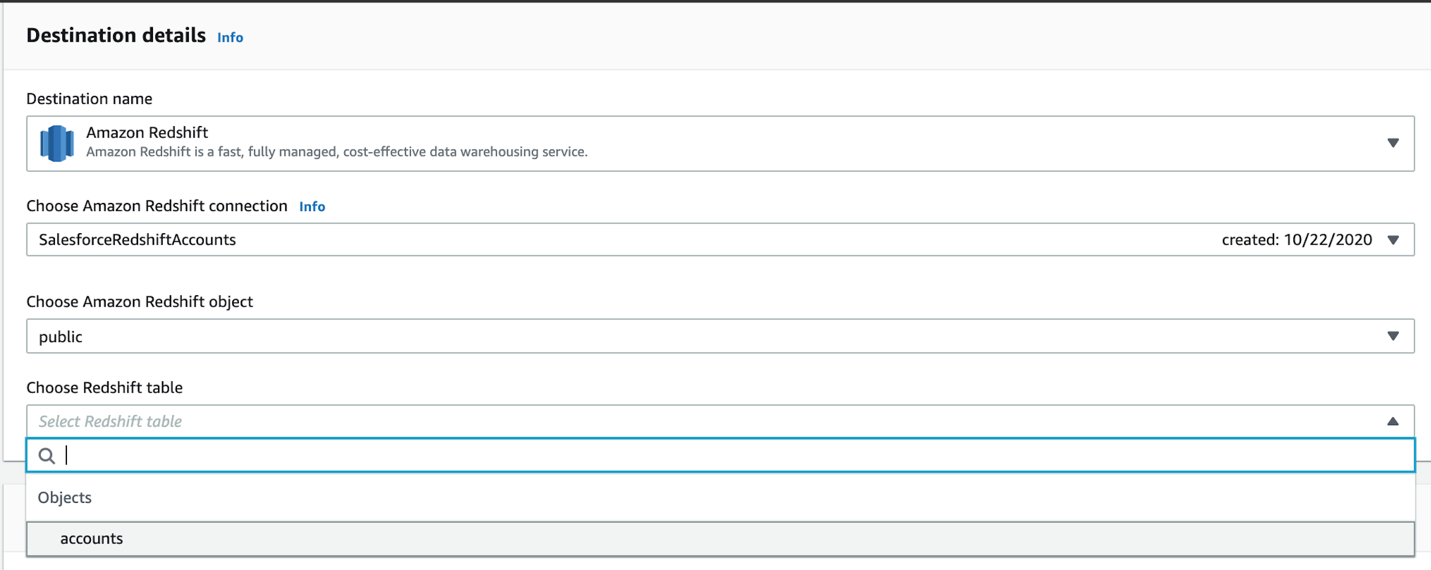
* A Connect to Amazon Redshift window appears.



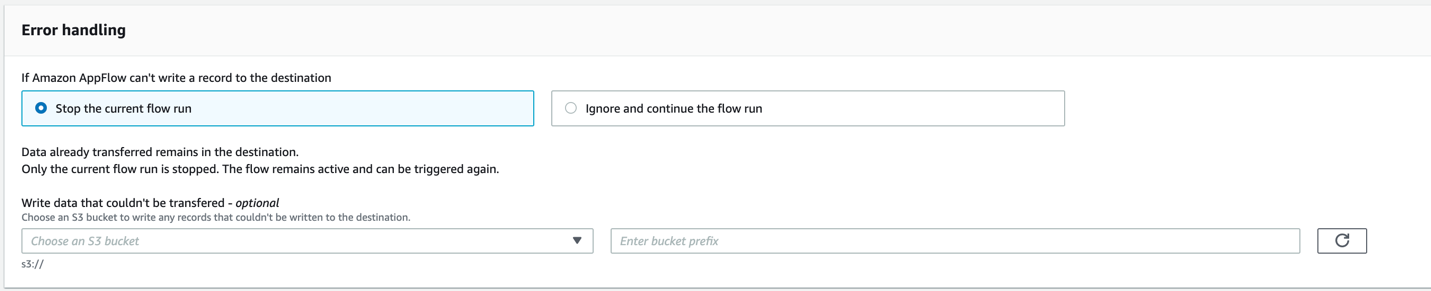
Fill in the following information, and then select Connect. Choose the S3 bucket where Amazon AppFlow will write the data before invoking COPY. This is most likely the S3 bucket you had created earlier, hope you had created one.

* JDBC URL – The JDBC URL of your Amazon Redshift cluster which you would find in the Properties tab of the cluster.
* Bucket details – Choose the S3 bucket and if any prefix (optional) where the data should be written.
* Role – IAM role created in Step 2 above.
* Username – User name to login to your Amazon Redshift account.
* Password – Password to login to your Amazon Redshift account.
* Connection name – Name the new connection.
* In the Destination details window that appears, choose the Amazon redshift object and table which you had created.





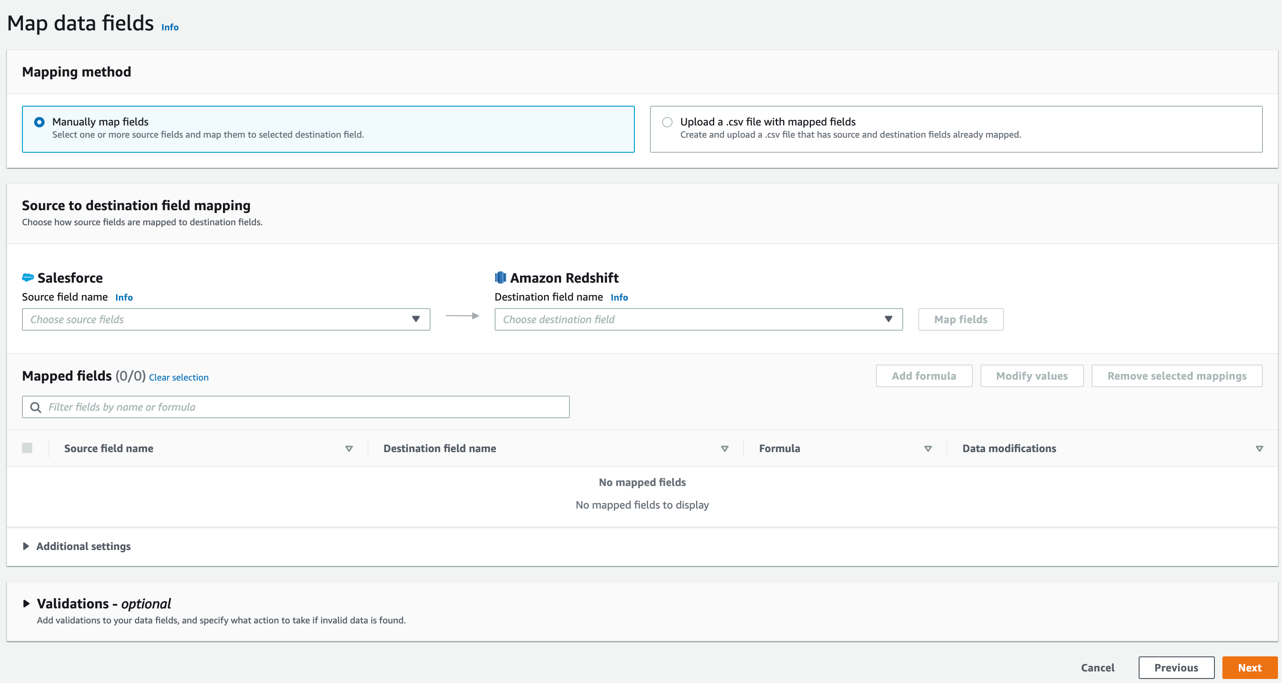
* Under Error handling, select how you want to handle errors.

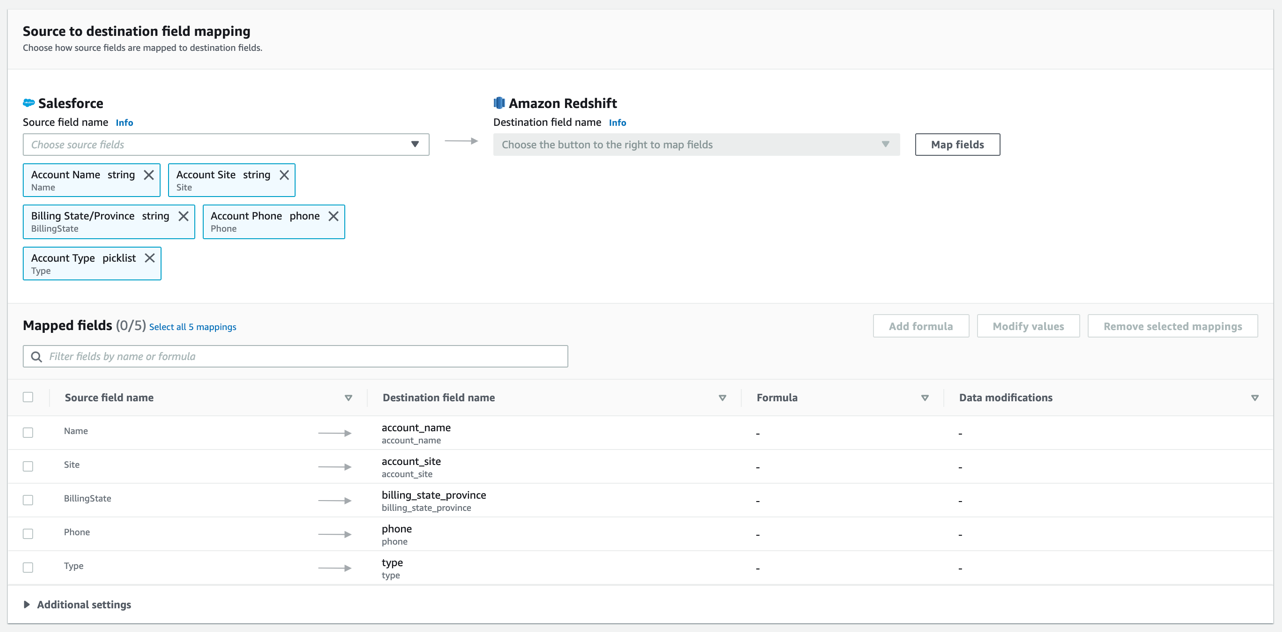


* Under Flow trigger choose how you want to trigger the flow.

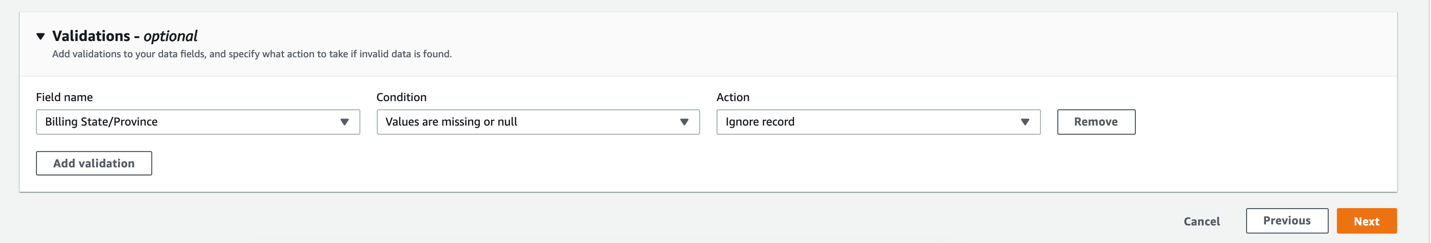


* In the Source to destination field mapping window that appears. use the drop-down menu to map the fields between the Salesforce accounts and your Amazon Redshift table. Then, select Next.



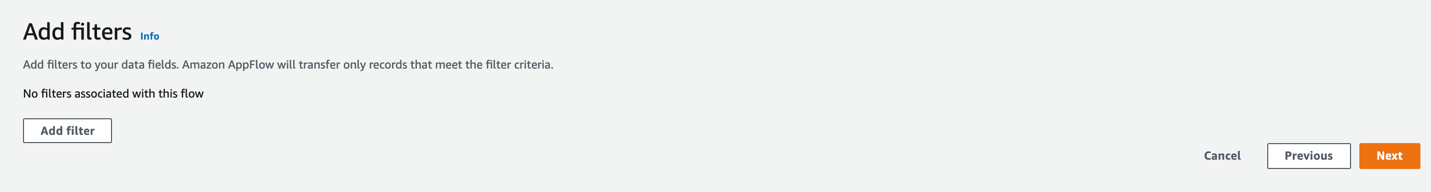


* Add Validations to your data fields, and specify what action to take if invalid data is found.

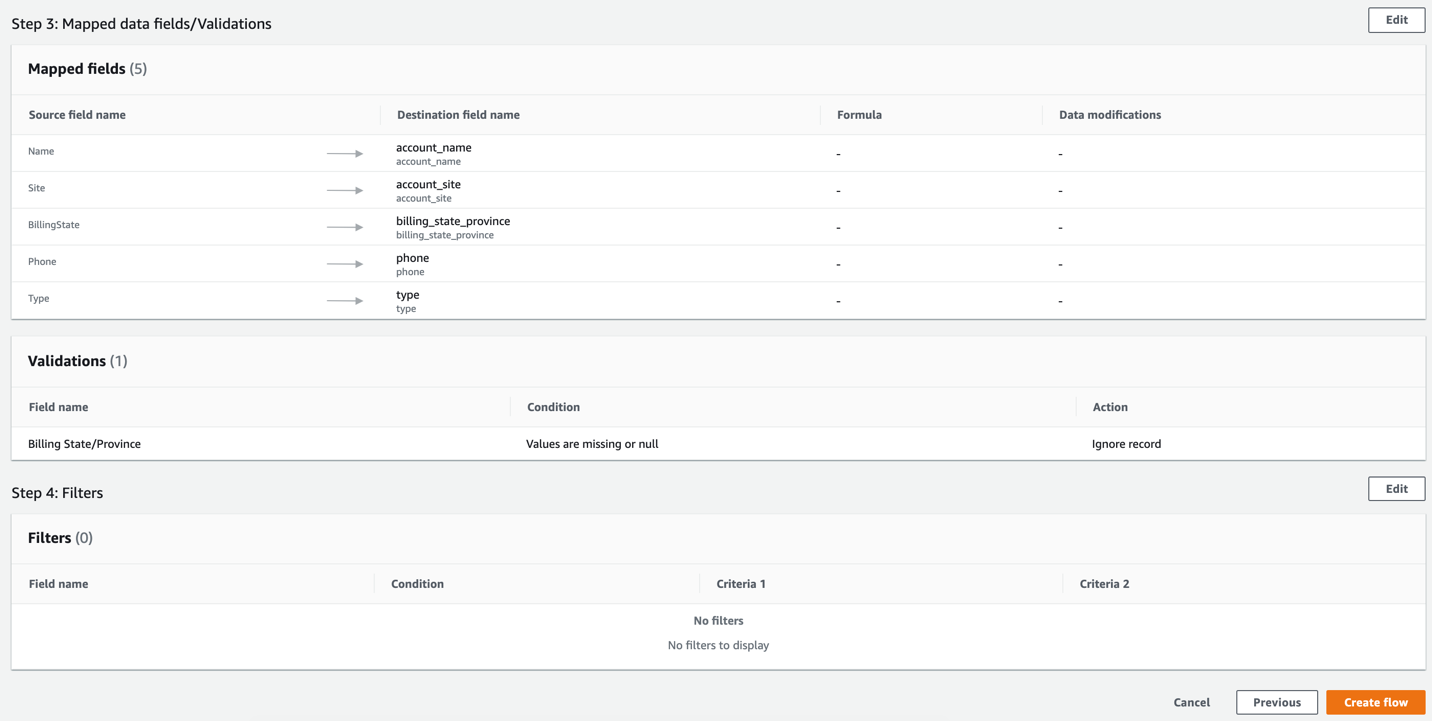


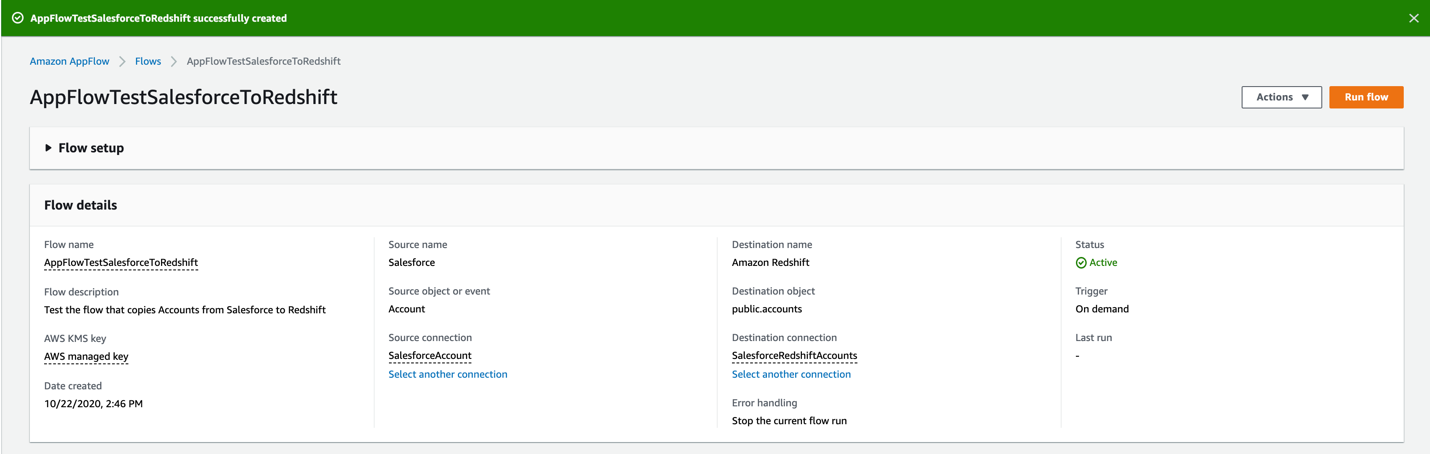
* In the Add filters window, you can add filters which help you pull records of interest. For example, use filters to import only the records that were created during the last month, or records for a particular customer.

When done, select Next.

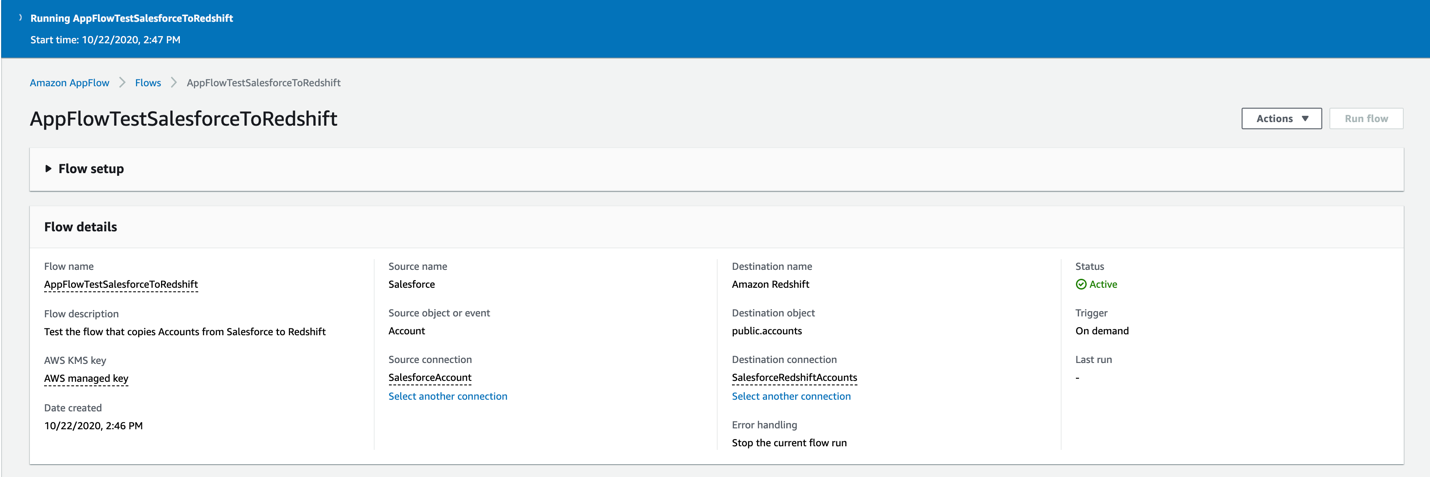


* Under the Review and create window, review the configurations and select Create flow.

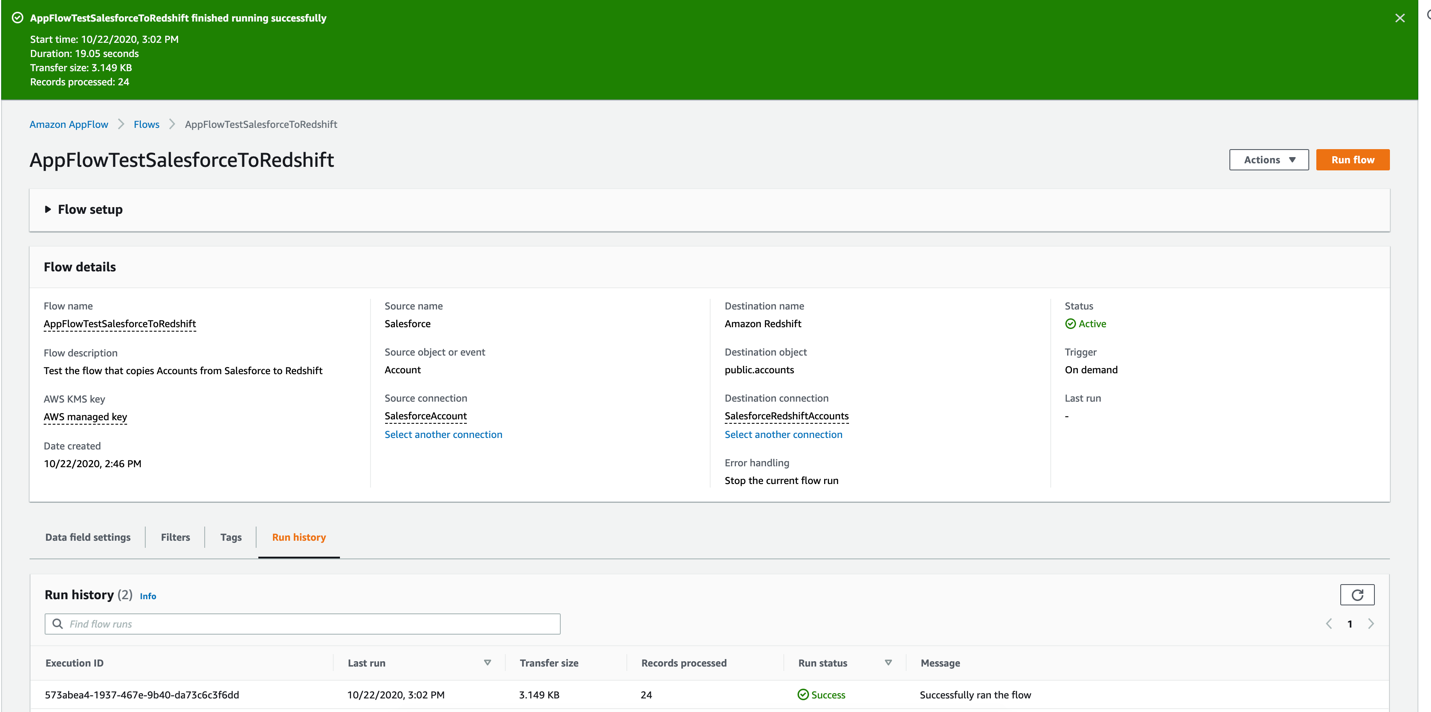




* Under the Flow window, run the flow by selecting Run flow. And that’s it. The flow is ready to be used.

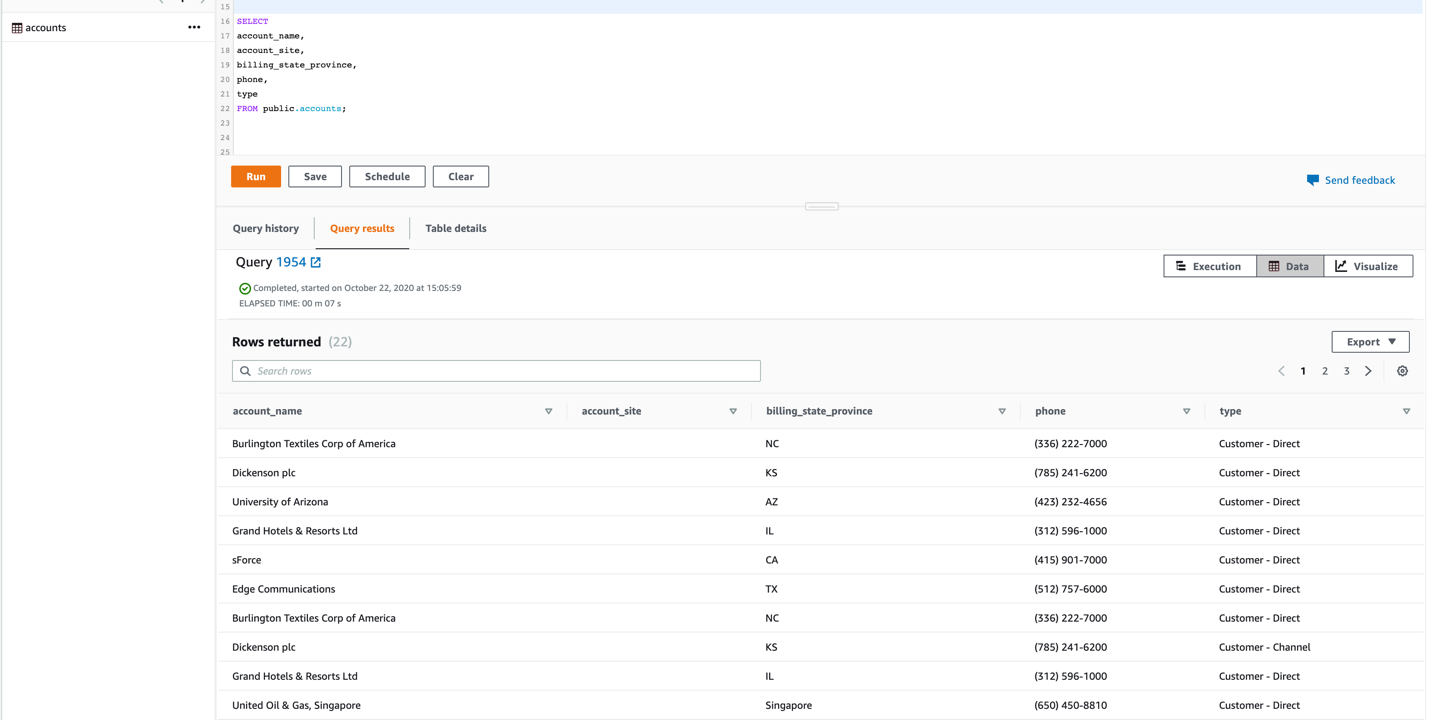


* Select the Run History tab. Under that tab you can review the run status of recent flows.



**Run and view the data**

Right, we are finally done and ready to see our data. It instantly populated the Redshift table.

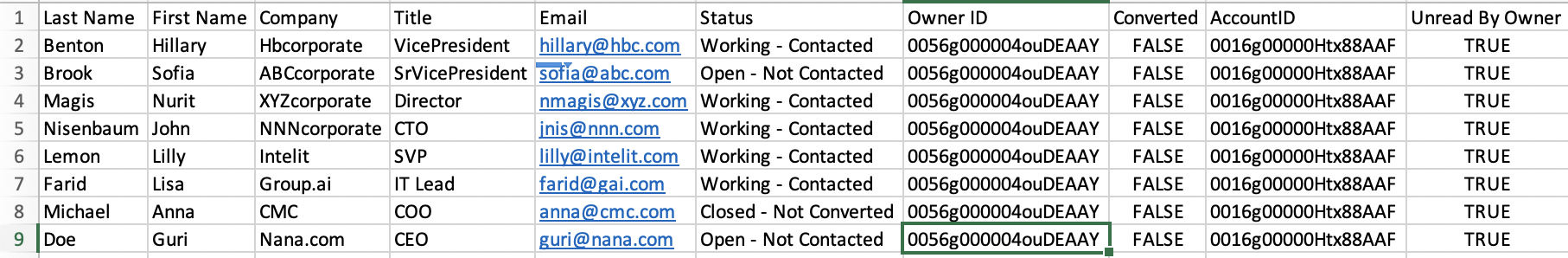


**Let’s Create Another Use Case: Creating New Salesforce Lead Record Using Campaign Data in S3**

In this procedure will create a flow in Amazon AppFlow that creates new Salesforce lead records using campaign data stored in .csv file in Amazon S3.

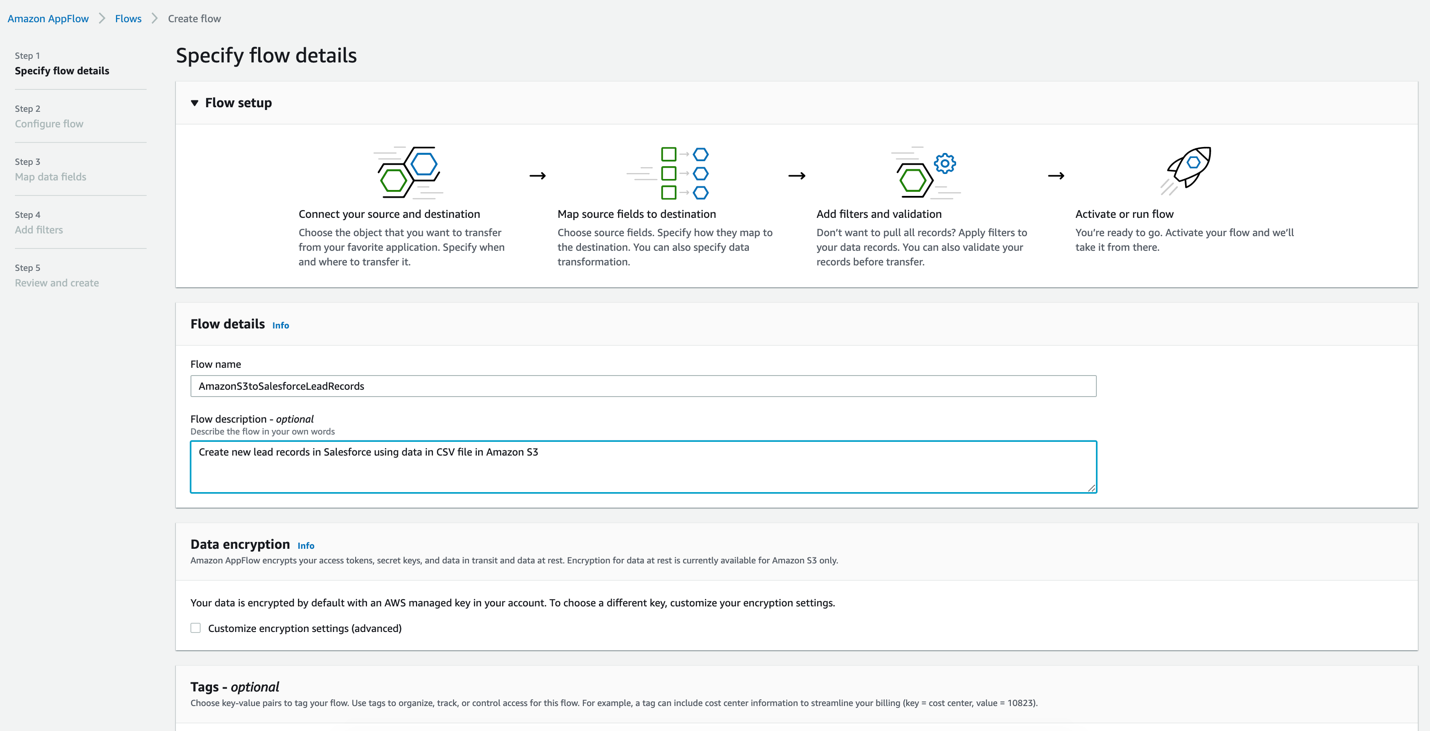
We will create an S3 bucket named appflow-salesforce-test. Under that bucket, will create a folder name s3todemo with a contacts.csv file with lead records from a recent campaign we would like to transfer to Salesforce.

These are the content of contacts.csv file



Let’s create a new flow.

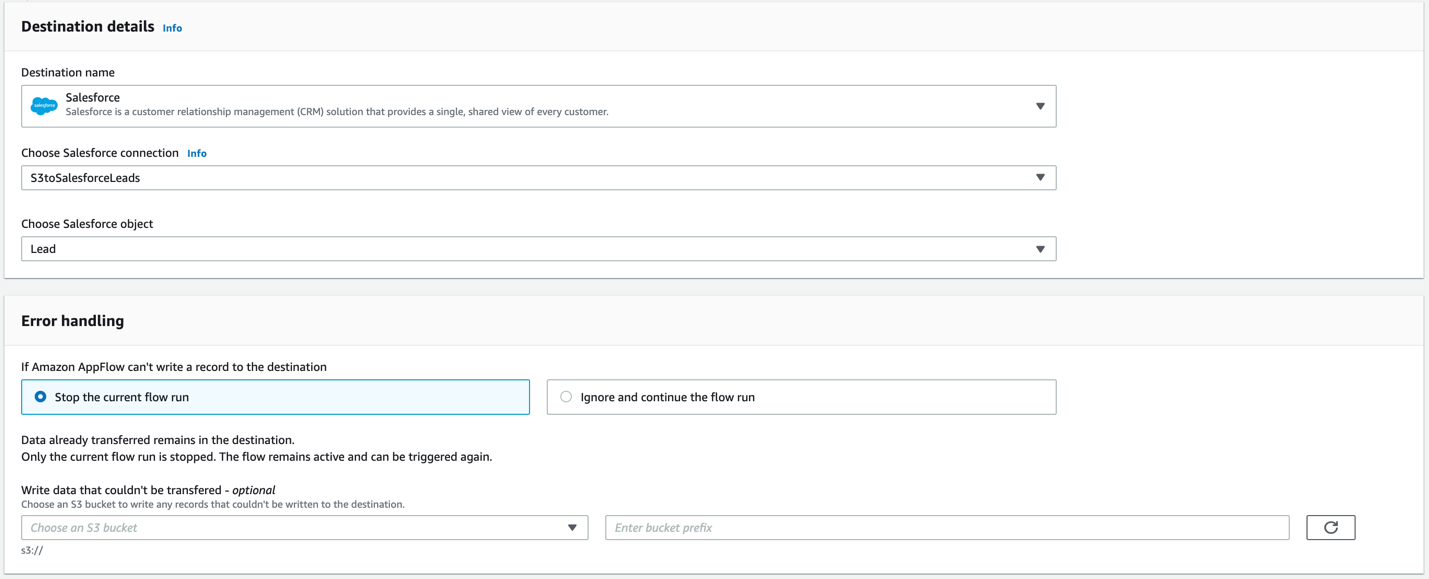
* In Amazon AppFlow, select Create flow, specify the flow details, and then select Next.



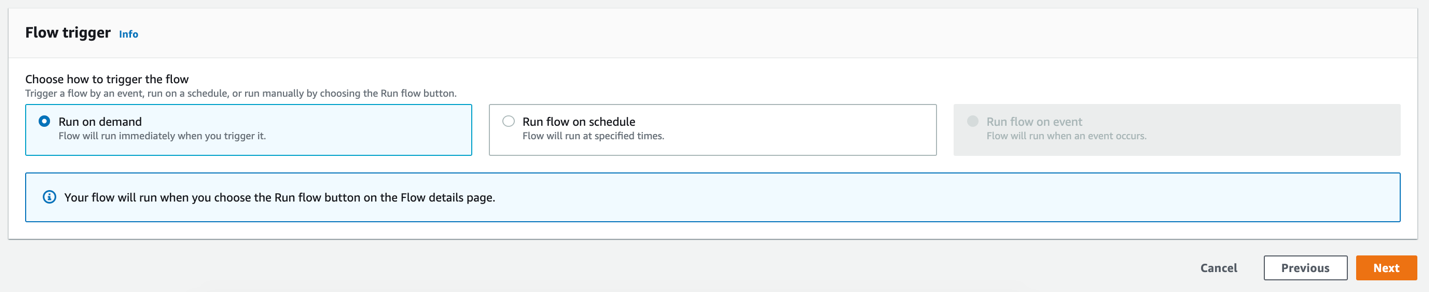
* Go to the Configure flow window, select Amazon S3 as the Source name and, under Bucket details, choose the S3 bucket.
* In the Bucket prefix field, enter the name of the folder that contains the .csv file. If the .csv file is located directly under the Amazon S3 bucket, then leave the Bucket prefix blank.



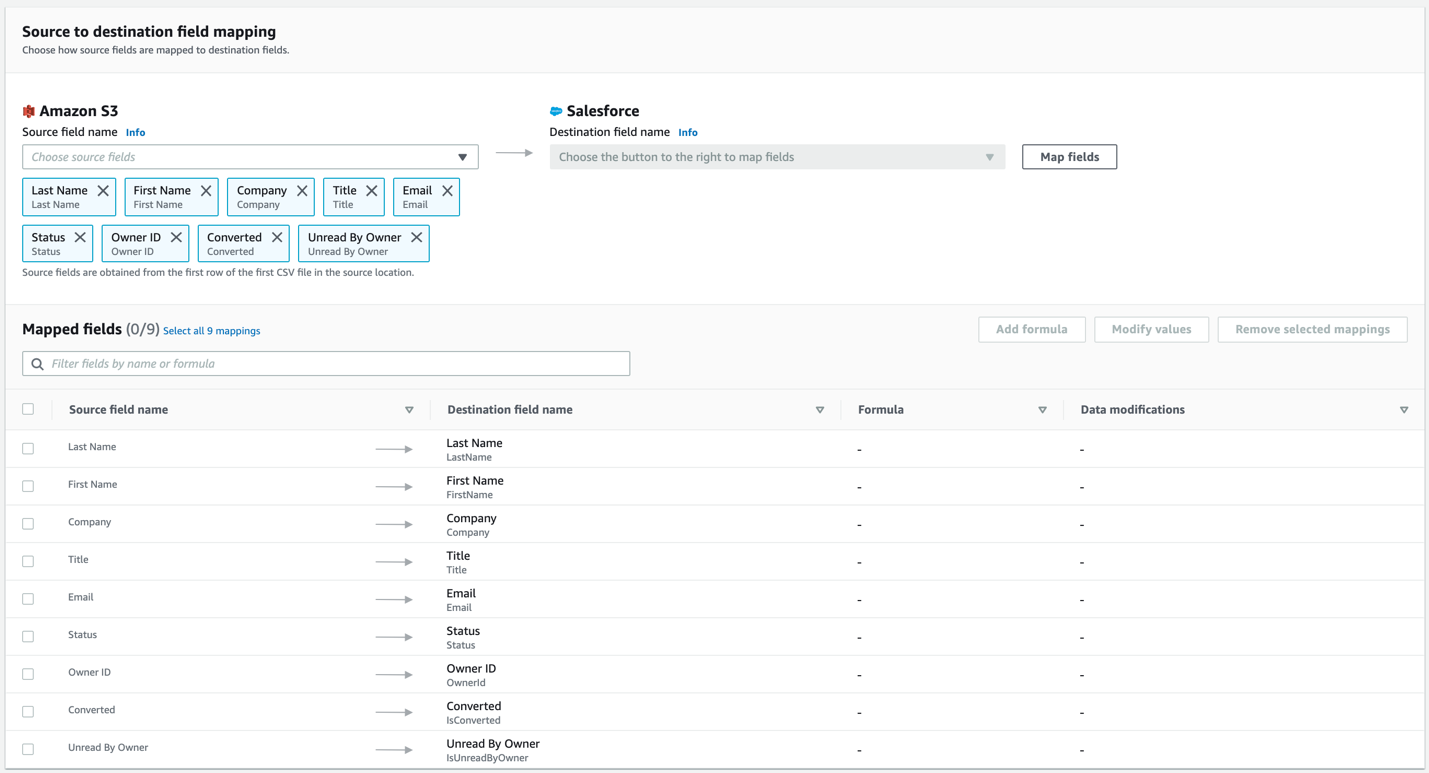
* Go to the Destination details section, and select Salesforce as the destination. Choose your connection or create a new one as shown previously.
* Under Choose Salesforce object, select Leads from the pull-down menu.
* Under Error handling, select how you want to handle errors.



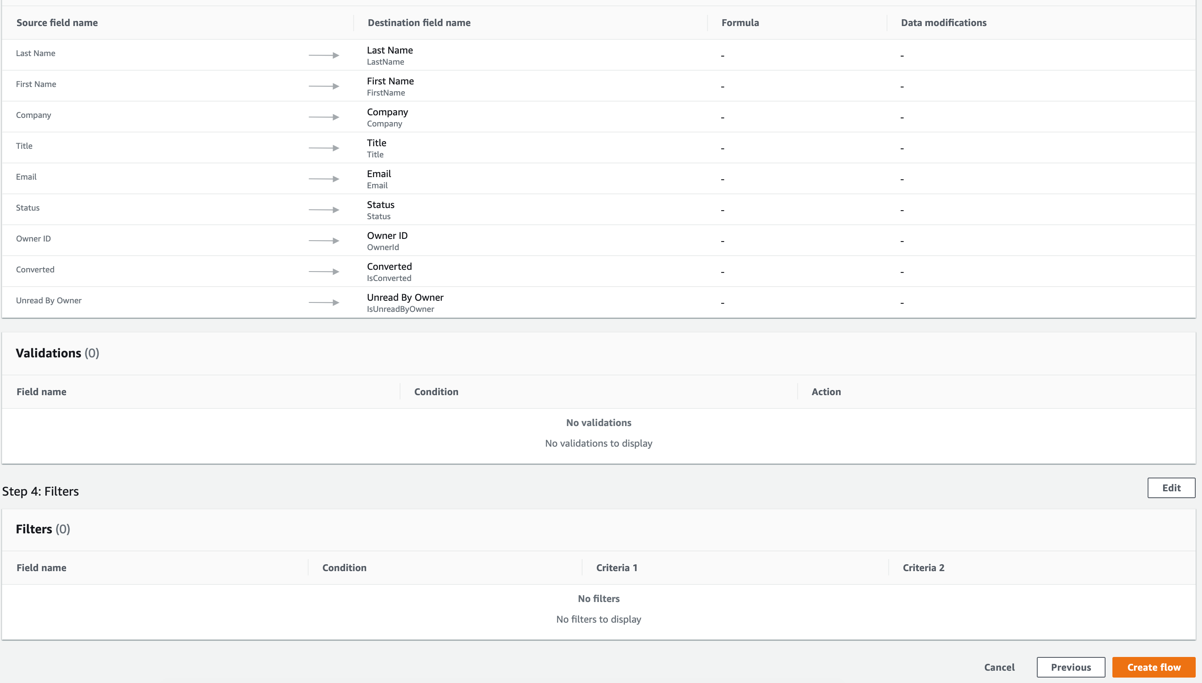
* Under the Flow trigger section, choose how to trigger the flow.



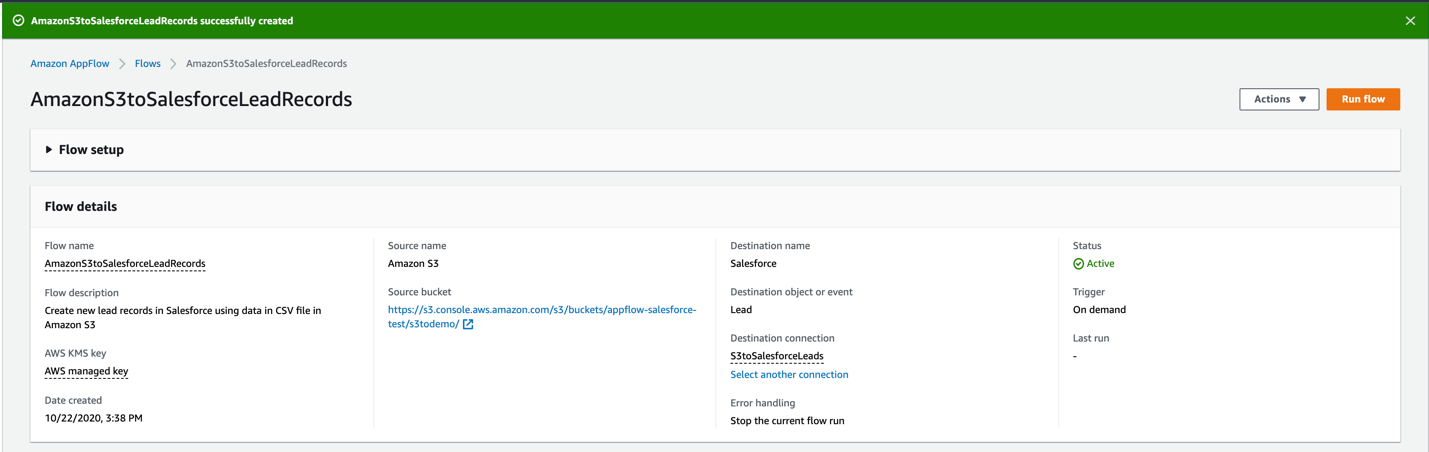
* Under Source to destination field mapping, map the fields in the .csv file to the fields in the Salesforce lead record, and select Next.



* Add filters, if needed.
* In the Review and create window, review your configurations and select Create flow.



* Once you run the flow you will see the records successfully processed with the flow.



* Back in Salesforce, you will see the new leads from the campaign data in the .csv file.

