

Salesforce and Amazon Web Services Integrations

November 2020



Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

© 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Contents

Abstract	4
Introduction	5
Integration layers.....	6
Connectivity	7
SSL Connection	7
AWS Site-to-Site-VPN.....	7
AWS Direct Connect	7
AWS PrivateLink.....	7
VPC Peering	8
Salesforce Private Connect	8
Related information about connectivity components.....	8
Use cases.....	9
Integration use cases.....	9
Use case #1: Drive new value by integrating and extending new capabilities, data exchange to developers using Amazon AppFlow, Heroku, MuleSoft from customer data.....	9
Use case #2: Secure data exchange, data pipeline with visualization and near real- time analytics.....	16
Use case #3: Event-based actions by Salesforce, improving decision-making with sentiment analysis that uses machine learning (ML)	20
Application use cases	23
Use case #4: Salesforce Service Cloud customers adding contact center capabilities for voice, chat, transcription, analytics, and AI/ML.....	23
Use case #5: Integrating Alexa for Business to add voice and connected voice devices while using Salesforce	25
Conclusion and next steps	28

Abstract

Integrations between Salesforce software as a service (SaaS) solutions and AWS extend each platform with new capabilities that enable organizations to strengthen relationships with their customers, solve business challenges, and reduce churn.

The long-standing partnership between Salesforce and AWS delivers integrations that can be implemented at low cost, that reduce total cost of ownership, and that use applications already integrated into the Salesforce platform. These integrations between Salesforce and AWS solutions enable better control of business-critical processes. They capture the right data and send it to the right tools, making it a valuable source for the insights and predictive analytics required by a modern data-driven and customer-centric business.

By implementing these integrations, organizations expand Salesforce and AWS services, while still leveraging each platform independently. They can use data pipelines, automation, and tightly integrated services to meet security, connectivity, and ever-changing business requirements, as well as the demands of their customers.

Throughout this white paper, there are introductions to necessary integration components like networking and integration services such as Amazon AppFlow, Mulesoft, and Heroku, along with explanations of how and when to use them. This paper provides a set of table stakes in the form of five use cases that enable business and IT to understand how they can expand, create, and blend integrations together for beginner or expert users of Salesforce and AWS.

More importantly, as systems, business processes, and platforms become interdependent, architecture, tradeoffs, and technology choices are critical to the success of customers. This paper, therefore, also provides best practices, recommends architectures, and discusses what to consider when choosing components for building secure, scalable integrations.

Introduction

As organizations begin the journey of implementing integrations between SaaS applications and cloud providers, they discover that the process requires vast amounts of time for development, high licensing costs, maintenance and upgrades. Another challenge is that data lives in different places, such as on-premises databases, SaaS applications, or the cloud. It becomes increasingly difficult for organizations to pull and push data from these sources to build a complete 360-degree view of their customers and gain valuable insights. Organizations are looking to Salesforce, the #1 CRM, and AWS, the highest adopted cloud platform, for solutions.

Salesforce and AWS offer services with integrations based on security, simplicity, and scale. Recent Salesforce and AWS partnerships and innovations in artificial intelligence (AI), voice, productivity, and cloud training have enabled customers of both industry-leading platforms to explore new ways to extract incremental business value from their investments at a lower cost. The global strategic partnership between Salesforce and AWS deepens the interoperability between these platforms and enables organizations to accelerate digital transformation. Salesforce and AWS customers can add new capabilities in each platform by using no-code, low-code, high-code, or out-of-the-box integration services, applications, or both, which can significantly shorten development and upgrade times while keeping costs low.

This whitepaper serves as a guide for individuals seeking guidance and recommendations on how to implement a secure, simple, and scalable integration between Salesforce and AWS. Five integration use cases are used to show the scope of the innovations provided by the global Salesforce and AWS partnership.

Integration layers

Multiple options are available to those organizations that want to integrate their Salesforce and AWS environments (Fig. 1). There are three layers of integration:

- Connectivity
- Integration
- Application

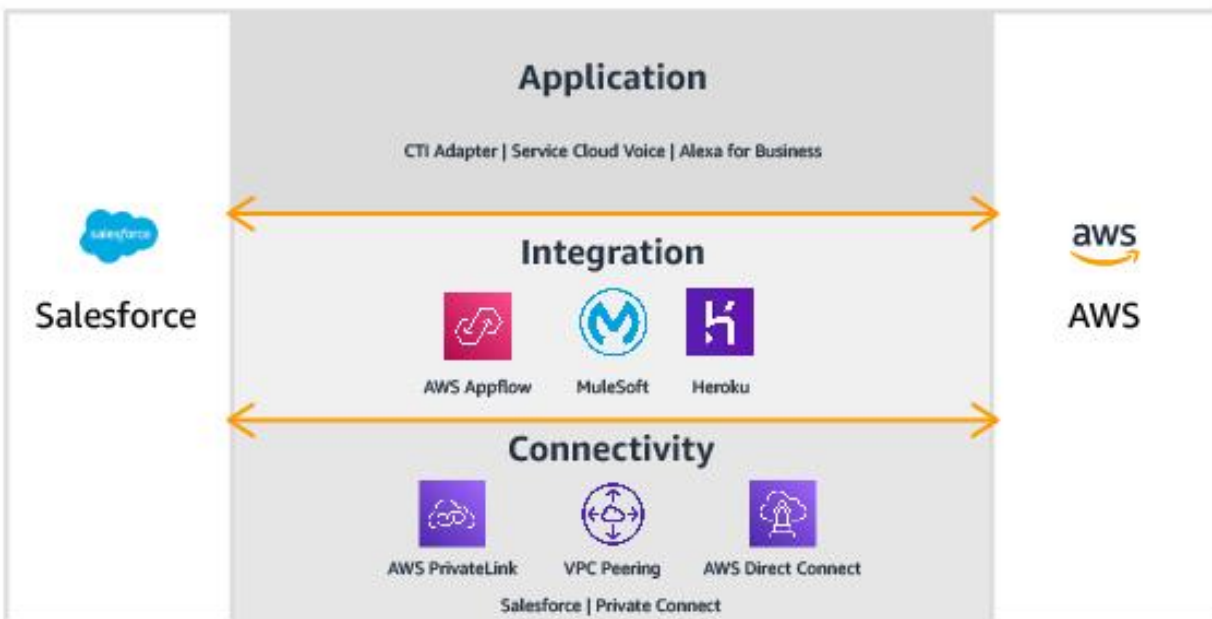


Fig. 1

As part of the integration strategy, both the **Salesforce and AWS** environments can serve as data sources and destinations, depending on the business requirements and use case.

Connectivity is the layer where network connectivity is established between the Salesforce environment, AWS, and an on-premises data center. **Integration** is the layer where services such as Heroku, MuleSoft, and Amazon AppFlow, can be used to develop integration components and APIs to securely transfer data between the Salesforce and AWS environments. **Application** refers to the layer where applications such as Amazon Connect CTI Adapter, Salesforce Service Cloud Voice, and Alexa for Business are integrated. .

This paper walks through five integration use cases in the integration and application layers. However, before looking at those use cases, we explore the connectivity layer in technical depth to show what networking and connections are available and how they work.

Connectivity

This section covers the approaches by which Salesforce and AWS can connect to each other. By understanding these components, customers can better decide which meet the connection and networking requirements for their integration use cases:

- SSL connection
- AWS Site-to-Site-VPN
- AWS Direct Connect
- AWS PrivateLink
- VPC Peering
- Salesforce Private Connect

SSL Connection

A common network connection that customers use to connect to Salesforce is an **SSL connection**. This is an application-level network connection at the Salesforce application layer and traverses the public internet via Transport Layer Security (TLS) encryption without any additional configuration. The connection is used when a Salesforce application needs to communicate to an on-premises data center or an application in AWS.

AWS Site-to-Site-VPN

Customers seeking a network-level connection use **AWS Site-to-Site VPN** to establish a secure and private session with IP Security (IPSec) and TLS tunnels. This type of connection allows for a network connection over two VPN tunnels that span multiple Availability Zones with the AWS global network.

AWS Direct Connect

AWS Direct Connect (DX) serves as a highly available, durable, private network connection. This allows data to traverse over a private AWS network backbone with dedicated 1G or 10G bandwidth. A typical [AWS DX](#) connection is from a data center to AWS, enabling the movement of data from on-premises to AWS for processing.

AWS PrivateLink

AWS PrivateLink is an AWS network service that provides a secure, private connection to a Salesforce-managed Virtual Private Cloud (VPC) inside an AWS region that has direct connectivity to a Salesforce organization. [AWS PrivateLink](#) also provides private connections between VPCs, AWS services, and Salesforce applications, running securely on the Amazon network. Heroku Postgres via PrivateLink and Salesforce Private Connect are integration application connectivity components that use AWS PrivateLink for secure, private connectivity between cloud-native platforms.

VPC Peering

VPC peering connection is a networking connection that enables traffic to be routed between two VPCs using private IPv4 addresses or IPv6 addresses. [VPC peering](#) works best for connecting across Amazon VPCs. An example is MuleSoft Anypoint Platform (a SaaS API development platform) connected to AWS.

Salesforce Private Connect

Salesforce, in partnership with AWS, has launched [Salesforce Private Connect](#), a service that is backed by AWS PrivateLink. This service enables private API callouts from Salesforce to a service running in AWS to send or retrieve data. Salesforce Private Connect also [enables calls to Salesforce APIs from within Amazon VPCs](#) privately (A2S or inbound connection).

[Salesforce Private Connect](#) integrates with and is supported by Amazon AppFlow, an AWS integration service. Customers can use the AppFlow private data transfer option to ensure that data is not exposed to the public Internet during transfers between AWS and Salesforce, improving security and minimizing risks of Internet-based attack vectors. AppFlow makes it easy to configure private data transfers with Salesforce in just a few clicks. It automatically sets up PrivateLink endpoints and manages the lifecycle of these endpoints without requiring customers to set up or monitor any network infrastructure.

Related information about connectivity components

To learn about AWS VPC and VPN connectivity, see:

- [AWS VPC Connectivity](#)
- [AWS VPN Connectivity](#)

Use cases

The following five use cases focus on Salesforce and AWS integrations used to solve customer challenges. These challenges range from data transfer to data enrichment and data visualization. The customers are using AppFlow, Heroku, or MuleSoft for integration, as well as Salesforce applications, add-ons (out of the box), deeply integrated capabilities (e.g. Amazon Connect, Alexa for Business). These use cases fall into the other two integration layers: integration and application, but all can also use the connectivity methods described previously.

Integration use cases

Use case #1: Drive new value by integrating and extending new capabilities, data exchange to developers using Amazon AppFlow, Heroku, MuleSoft from customer data

Integration services such as Amazon Appflow, Mulesoft, and Heroku enable customers to integrate data, extend new capabilities, and extend data to drive new value. Data extracted from multiple Salesforce orgs or AWS is enriched with reduced code complexity and low configuration setup. The data is analyzed and visualized using software like Tableau or Amazon Quicksight. Developers bring applications to market quicker, providing greater utility and performance. The integration between both platforms enables a more complete view of the customer. These services can be used in all the use cases mentioned in this paper. Which integration service to use depends on your preferred approach and existing implementations.

Amazon AppFlow is a simplified, no-code tool for data integration. Development teams that want to connect and orchestrate flows between AWS services and Salesforce applications without writing code or customizing and maintaining pre-built connectors should consider AppFlow. SaaS application administrators and business analysts can also use AppFlow to implement integrations by themselves to innovate at a fast pace and satisfy their specific business requirements.

Mulesoft and Heroku provide integration with more traditional coding interfaces and are well-suited for organizations that prefer to use customizable pre-built connectors or that need to integrate high-code applications with Salesforce and AWS data. Salesforce and AWS developers get access to data that Salesforce and AWS capture, also benefitting from bidirectional data transfer from either platform and the fact that no special coding or complicated tooling is necessary.

Amazon AppFlow

Amazon AppFlow, an integration service from AWS, enables the transfer of data between Salesforce and AWS services like [Amazon Simple Storage Service](#) (Amazon S3) and [Amazon Redshift](#)—in a matter of clicks. [Amazon AppFlow](#) can be used to create new records in any Salesforce org using data

that is stored in an Amazon S3 bucket. Amazon AppFlow can be set up to run on a preferred schedule or when a platform event or change data capture occurs. Data can be transformed or processed as part of the transfer.

The setup of these processing tasks is simple and can even be done without coding. Amazon AppFlow users do not have to spend significant amounts of time on the undifferentiated heavy lifting required by extensive coding efforts and data updates and can instead focus on rapid product and application delivery.

Amazon AppFlow is based on a source and destination model configured through flows. There is up front verification in the Salesforce org. The Salesforce account can be enabled for API access and configured to allow connected apps to be installed. This is typically the default setting. The name of the Amazon AppFlow connected app is "AppFlow Embedded Login App." The "AppFlow Embedded Login App" has the refresh token policy in Salesforce set as "Refresh token is valid until revoked." To verify, users can navigate to Setup→Apps→Connected Apps→Connected Apps OAuth Usage→Manage App Policies in their Salesforce accounts and use Edit Policies to correct it if required. If the Salesforce app enforces IP restrictions, the "[AppFlow IP address](#)" has the range of IP addresses to whitelist.

An Amazon AppFlow flow transfers data between Salesforce and AWS. An example of a flow setup is Salesforce as the source and Amazon S3 as the destination. The flow is configured to extract data from Salesforce and puts a JSON data file into Amazon S3 using a Salesforce account login connection. Another flow setup is with Amazon S3 as the source and a Salesforce org as the destination to update records via a CSV file uploaded to the source Amazon S3 bucket. Amazon AppFlow also supports Parquet files.

An on-demand option enables each flow to run as soon as the flow configuration is saved. Another option is to set up the flow for execution on a schedule. The final step to configuring a flow is to map fields between the source and destination. AppFlow enables either mapping the fields manually or uploading CSV files with mapped fields and selecting the fields of the Salesforce object being transferred. There is also the option to map all fields and another to set a filter on the create date field and select only those records created before or after a specific date.

For example, a maintenance company owner wants to create an application that will extract maintenance customer information from Salesforce Service Cloud. He wants to provide a mobile dashboard so that customers can see appointment status and account info while allowing for interactive changes/updates to appointments. The customer information includes:

- Scheduled maintenance dates
- Equipment needing maintenance
- Past maintenance performed
- City
- State
- Phone number

- Customer name
- Account number.

The owner does not have a big development team or Salesforce or ETL experts on staff and decides to implement Amazon AppFlow.

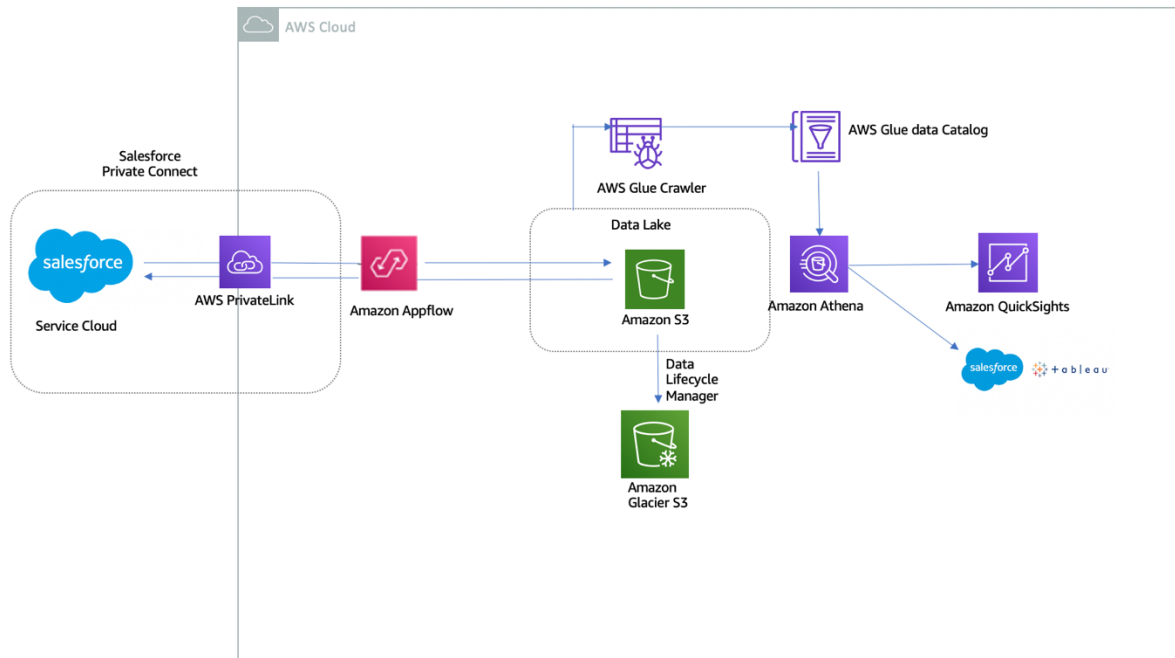


Fig. 2: Diagram of Amazon AppFlow and Salesforce Private Connect

Support for Salesforce is already built into Amazon AppFlow. The owner of a batched (scheduled) or event-driven (triggered by change to a data object) action can extract data and put it into S3. This will depend on workflow and end customer requirements. If it is a monthly extraction, a batched approach would work well here. If a dashboard needs to be updated with the latest data point, such as scheduled maintenance dates, a change will trigger an event invoking the event-driven extraction.

Once data is in Amazon S3, the owner can use [AWS Glue](#)-managed services to transform data and store it in a metadata datastore—the AWS data catalog. Using ANSI-standard SQL, [Amazon Athena](#) queries customer data, creating external data tables and views. The data and views can be imported into [Amazon QuickSight](#) to produce a data-rich dashboard that maintenance customers access through their mobile device or PC.

Alternatively, using [Amazon Athena](#) Tableau connectors, Tableau can be used to create visualized dashboards for internal and external customers. As for the connectivity between AWS and Salesforce, it is an SSL connection. [Salesforce Private Connect serves as a cloud-native secure connection](#) between Salesforce and AWS, so that network traffic is not exposed over the public internet. Salesforce continues to manage end-to-end connections and access controls, further supporting

reduced configuration setup, so developers can focus on delivering data-rich applications and native dashboards and metrics.

More information about integration with AppFlow

For information about other services related to this use case, see:

- [Amazon S3 Glacier](#)
- [AWS Glue Crawlers](#)
- [AWS Glue Data Catalog](#)
- [Amazon Data Lifecycle Manager](#)
- [Tableau Connectors](#)
- [Dataflows Between AWS and Salesforce](#)

Heroku

Salesforce and AWS customers have a myriad of developer pipelines that they rely on every day to develop and release new product features. It is difficult to change, upgrade, or implement an existing pipeline without causing disruption or encountering a steep learning curve. There are developer shops that use Heroku to build and deploy their applications. Heroku provides an application deployment and a coupled Postgres backend, all connected through a platform add-on, [Heroku Connect](#), that enables teams accustomed to Heroku to connect their applications to Salesforce and AWS.

When Heroku Connect is used, applications that are either developed in Salesforce or Heroku share data synchronously between both platforms. Not having to worry about how the data communication flow is happening between Heroku and Salesforce is a time-saver for developers and users. Instead, they can control data synchronization through the Heroku application or Salesforce. This is only half of the Salesforce and Heroku integration.

The other half is connecting [Heroku Postgres](#) to AWS. This connection is done with [Heroku Postgres via PrivateLink](#) and a secure connection between an AWS VPC and a Heroku Postgres database running in a [Private Space](#) or a [Shield Private Space](#). The entire process can take as little as 20 minutes to provision.

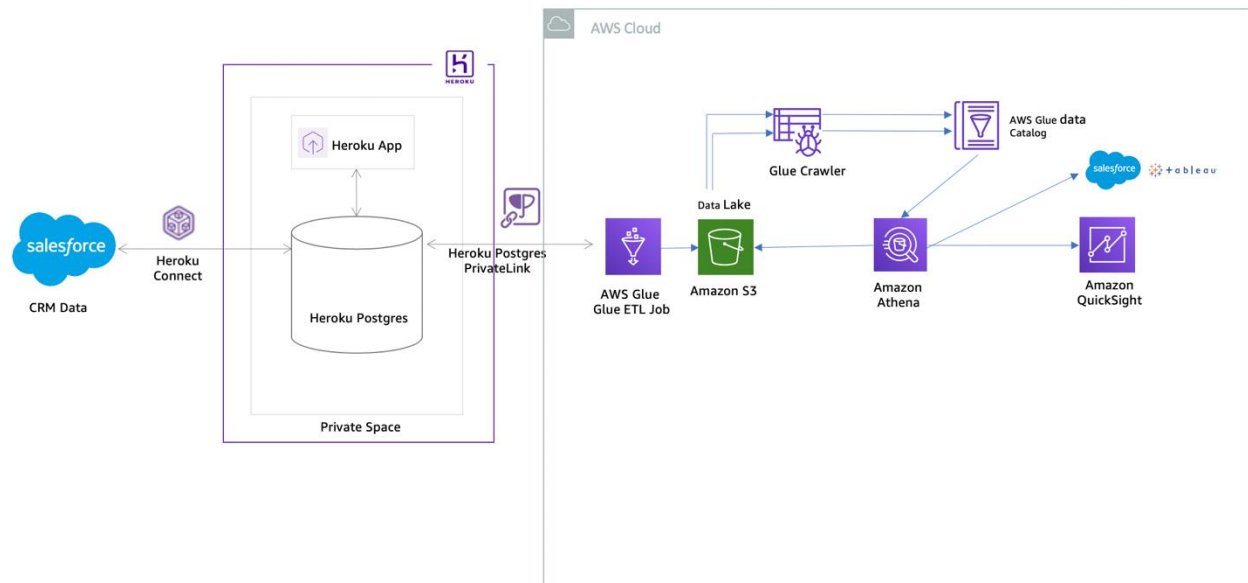


Fig. 3: Diagram of Heroku and Salesforce Heroku Connect/Heroku Postgres via PrivateLink

When can Heroku Connect and Heroku Postgres via PrivateLink be used for AWS and Salesforce integrations? A good example is when part of the development pipeline is on-premises and the goal is to go all-in on the cloud. It is easy to spin up the pipeline in AWS and migrate slowly or all at once. Data is shared from Salesforce to Heroku to AWS securely over AWS PrivateLink. The development of richer datasets is enabled by combining Salesforce, Heroku, and AWS or updating each of these platforms, together or separately.

Dev teams can accelerate rich data features for Heroku-native applications and those in Salesforce through data visualization and queries done with AWS managed services (Amazon Athena, [Amazon QuickSight](#)) and Tableau, which also enables them to build highly personalized apps and experiences.

Once the data is in Amazon S3 (data lake), it can be further processed by machine learning algorithms and models. The results produced can be ported back into Heroku or Salesforce. Each of these platforms scales easily, keeping pace with the need to deliver new applications to market much quicker, thereby meeting customer needs and enhancements.

Additional Heroku technical articles

For more details, articles, and documentation about the elements involved in Salesforce and AWS integration using Heroku, see:

- [Heroku Application and Heroku Connect](#)
- [Heroku Connect Configuration](#)
- [Heroku using AWS S3 to Store Static Assets and File Uploads](#)
- [Heroku Private Spaces](#)
- [Heroku Private Space](#)
- [Heroku Shield Private Space](#)

MuleSoft

As mobile and API-based applications increase in popularity and usage, developers are rapidly integrating their enterprise applications with Salesforce using MuleSoft Anypoint Platform, which offers prebuilt, reusable connectors to data and systems. However, their applications are often missing enriched data features, and delivery lead times are longer, tinged by the constant reminder of a complete move to the cloud. By integrating both AWS, Salesforce, and their applications with MuleSoft, developers can begin to solve these challenges. But how do they connect to MuleSoft? How is integration done between AWS and MuleSoft? MuleSoft offers several options for connecting to AWS.

For example, there is a scenario where numerous teams are using MuleSoft on-premises and are interested in migrating to the cloud. Typically, they already have an AWS footprint and are also using the designer in [MuleSoft Anypoint Platform](#) for API development and [MuleSoft CloudHub \(Runtime Fabric\)](#), both offered as PaaS and SaaS. MuleSoft Anypoint Platform serves as the connection hub to Salesforce, using basic username and password authentication and a Salesforce security token over SSL.

Alternatively, two recommended connectivity options between AWS and MuleSoft are AWS VPC Peering and DirectConnect. If the team is already in AWS, they can use AWS VPC peering to connect MuleSoft's VPC to the AWS VPC. In the case where data or a workload must remain in an on-premises data center, AWS DirectConnect allows a connection over a private network from the data center to AWS and MuleSoft.

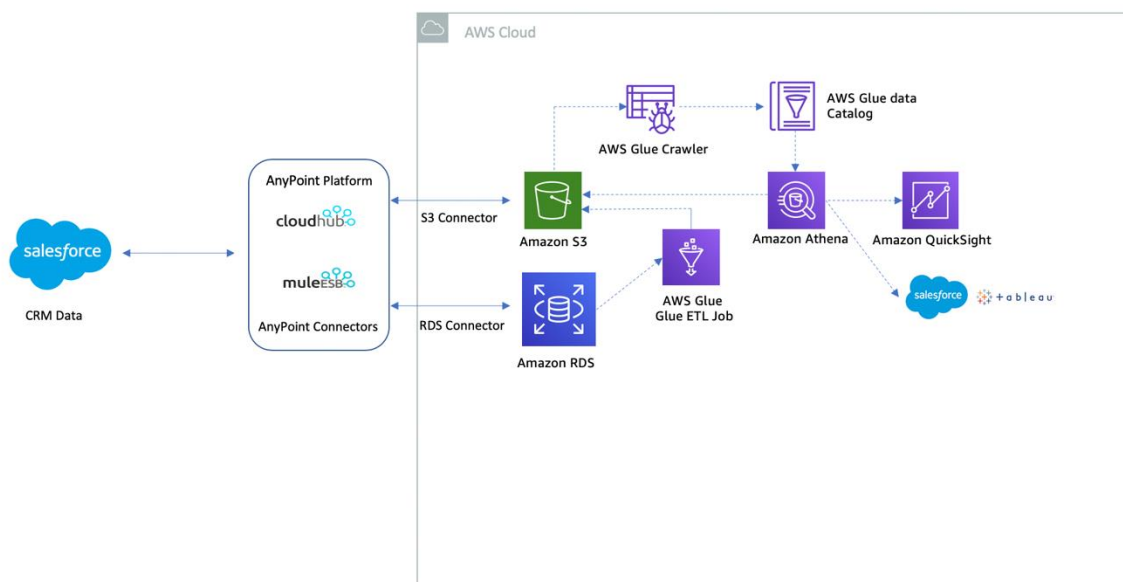


Fig. 4: Diagram of MuleSoft Salesforce and AWS

With MuleSoft Anypoint platform, developers can transfer data bidirectionally through platform workflows and connectors. MuleSoft provides the [Anypoint Exchange](#), cataloging numerous connectors, triggers, and templates based on an API framework ([RAML](#)). These are used as components coded into MuleSoft Anypoint workflows to update, load, or extract database data or records or archive data. Primarily, MuleSoft Anypoint platform functions as both an API IDE and workflow engine, not storing any data locally and acting as a passthrough.

The integration between AWS and MuleSoft is half of the total integration. The other half includes Salesforce. By using MuleSoft for the integration between Salesforce and AWS, developers can enhance their application data elements with richer dashboards and analysis processed with specific Salesforce, Amazon S3, and [Amazon Relational Database Service](#) (RDS) connectors that are configured in a MuleSoft Anypoint workflow. Data from these different platforms can be easily exchanged, transformed, and stored. Developers are free to be more creative with their applications, meeting the ever-changing demands of customers. The integration flexibility of these platforms allows for easy migration, upgrades, and continued innovations, [such as serverless](#).

Related MuleSoft Documentation

A wealth of MuleSoft documentation is available that goes into more detail about using MuleSoft in the Amazon and Salesforce integration layer:

- [MuleSoft to Amazon Connectivity](#)
- [MuleSoft Enterprise, CloudHub Connect with Salesforce Example](#)
- [MuleSoft Enterprise, Cloudhub Salesforce to Database Example](#)
- [MuleSoft Amazon S3 Connector](#)
- [MuleSoft Amazon S3 Connector Examples](#)
- [MuleSoft Amazon RDS Connector](#)
- [Mule SDK](#)

Use case #2: Secure data exchange, data pipeline with visualization and near real-time analytics

Data transfer from Salesforce to AWS is key when adopting different platforms for integration. Two types of data are transferred: data that is unstructured from resources like application logs and SaaS applications or structured data from transaction applications and relational databases. As a result, there is a need to maintain a single source of truth and establish a data pipeline—from ingestion to transformation and analytics—for extracting value from data quickly.

As customers begin to build Salesforce and AWS integrations with AppFlow, MuleSoft, or Heroku and use these integrations to move data between more and more applications, the efficiency of data pipelines is a critical consideration in their overall integration strategy and development. Data pipelines include data transformation, enrichment, filtering, grouping, aggregating, and running algorithms against that data. As data pipelines evolve based on velocity, volume, and variety, they are taking on two workload profiles, big data, and ETL.

Different mechanisms are used to transfer data, including streaming and batch ETL. ETL stands for “extract, transform, load.” It is the process of moving data from a source, such as an application, to a destination, commonly a data lake. “Extract” refers to pulling data out of a source; “transform” is about modifying the data so that it can be loaded into the destination; “load” is about inserting the data into a destination. Streaming ETL data pipelines may occur alongside batch ETL pipelines, depending on the use case.

Streaming and Batch ETL Options

In both streaming and batch references, Amazon S3 serves as the single source of truth for data or a data lake. [Amazon S3](#) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Amazon S3 provides easy-to-use management, organizing data and configuring finely tuned access controls to meet specific business, organizational, and compliance requirements. Raw and processed data can both be stored in S3, with the option to move data to another tier storage, such as Amazon S3 Standard-Infrequent Access (S3 Standard-IA) or to archive to [Amazon S3 Glacier](#) using data lake data lifecycle policies for compliance or regulatory requirements.

Alternatively, it is possible to update [Amazon Redshift](#) or [Kinesis Data Streams](#) by streaming data directly. The data will update tables, customer information, or downstream applications responsible for starting services or issuing a payment notice to customers.

An example of batch is a Service Cloud account record that must be transferred to S3. One option is to use a partner third-party connection tool, Progress DataDirect, with [AWS Glue](#) services (Glue ETL

Job). Progress provides a Salesforce JDBC driver that uses the AWS Glue ETL job service to batch extract Salesforce data objects such as CSV, Parquet, or ORC file formats and puts them into an Amazon S3 bucket. Because it uses AWS Glue ETL managed services, Progress is a less costly alternative to commercial ETL offerings, also offering easy setup, low code, and integration with both Salesforce and AWS platforms.

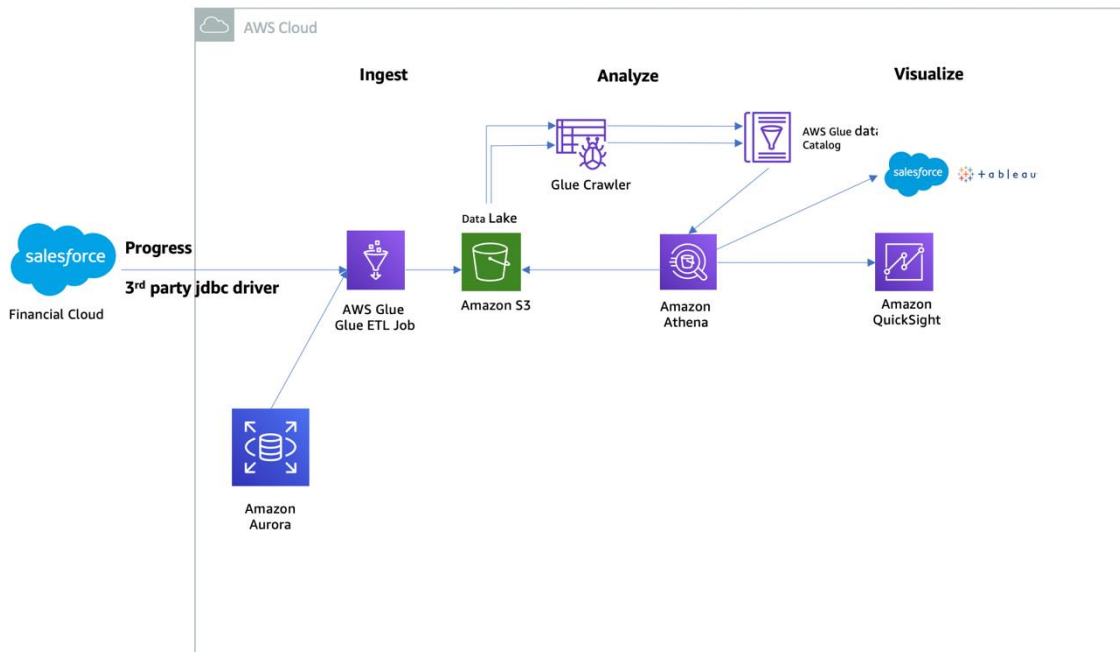


Fig. 5: Diagram of batch ETL from multiple data sources, data pipeline

An example of streaming ETL is a business finance manager who is looking for a payment on her outstanding accounts receivable dashboard ([Tableau](#)) that is updated when a change occurs in Salesforce. She called a customer and asked them to make a payment to their account's receivables. The streaming ETL and data pipeline enable her to see the payment status update when the company made a payment right after she called.

Streaming ETL may be referred to as real-time ETL. The data sources feed data to a stream processing service like Amazon Kinesis Data Streams and AWS Glue job streams. Amazon Kinesis Data Streams coupled with AWS Glue job streams serve as the backbone to streaming ETL applications and processes. The streaming ETL application extracts data from the source, or the source may publish data directly to the streaming ETL application. When a streaming ETL process completes, it passes data to a destination Amazon S3 bucket or Amazon Redshift. In addition, it can concurrently deliver data to other applications and repositories.

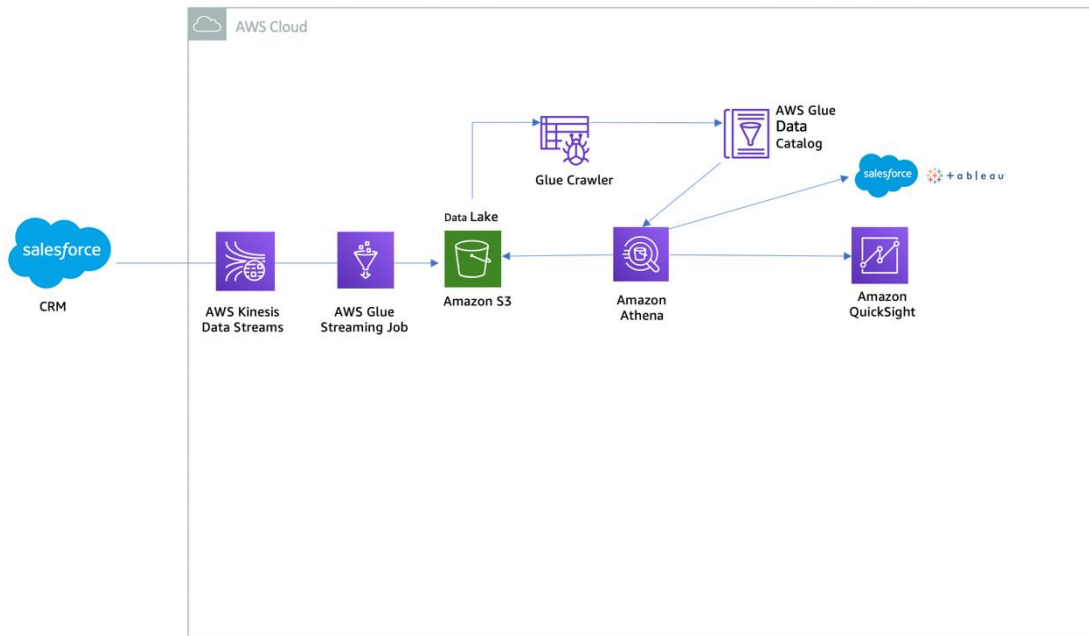


Fig. 6: Diagram of streaming ETL, data pipeline, analytics, and visualization

Options for Processing Data through a Data Pipeline

Processing data ad hoc and on-the-fly is complex because it is an always-on system that needs to be managed, patched, scaled, and maintained. Extending AWS Glue jobs to run continuously and consume data from streaming platforms such as Amazon Kinesis Data Streams and Apache Kafka, including the fully managed Amazon Managed Streaming for Apache Kafka (Amazon MSK), makes this processing easier and more cost effective.

A common practice is to write ETL jobs to extract data from different data sources in Python or Scala. Alternatively, AWS Glue services support custom Python scripts and code for data transfer and processing. This option requires the execution of these scripts and code to transfer data and the inclusion of custom logic for authentication and authorization. Continual development time, configuration management, and tweaks as workloads change are required.

There are different options available for extracting data and processing it through a data pipeline. Which option is best depends on workload, staff, expertise, budget, and requirements.

Several recommendations and best practices are to reduce complexity by minimizing writing application code for data transfer. AWS managed services can offset the extensive code, complex data pipelines, and high costs of this process by implementing pay-as-you-go models for data pipeline components. A third-party JDBC driver (Progress) that utilizes AWS Glue ETL services can minimize the time and effort required by data teams and upkeep of specific ETL code.

Another recommendation is to use SQL for data transformation and an analysis engine. Several data sources have features built into SQL, like maps, row-to-column, and analytical functions. Several business challenges can be expressed in SQL. The use of Amazon Athena allows standardization on SQL while providing fast analysis and query capabilities to data pipelines.

A final recommendation is to set up a secure communication with Salesforce Private Connect and AWS encryption for data in transit and at rest. Salesforce Private Connect is a good choice for exchanging data securely and privately. For moving data to Amazon S3, objects are encrypted on the server side with either Amazon S3-managed keys (SSE-S3) or customer master keys (CMKs) stored in AWS Key Management Service (KMS).

Ideally, data analytics can run across Amazon S3 objects with query-in-place services. Amazon Athena can query S3 data with standard SQL. When AWS Glue Crawler is used to scan an S3 bucket, a data schema is created that is then stored as metadata in an AWS Glue Data Catalog. Every time the AWS Crawler is run, if the data has changed in the S3 bucket on read, a new schema version is captured in the AWS Glue Data Catalog, preserving any data changes and versioning it. Amazon Athena queries and views can be imported into SPICE, an in-memory optimized calculation engine for [Amazon QuickSight](#), designed specifically for fast, ad hoc data visualization. Amazon Athena connectors are available for Tableau to connect directly to Athena data sources or materialized views.

Those customers with existing ETL solutions might be using products like Dell Boomi, Informatica, and Talend. An alternative to consider is to implement the ETL and data pipelines described in this section.

Related data exchange documentation and articles

For more technical details and related information about the options in this use case, see:

- [AWS Glue Crawlers](#)
- [AWS Glue Data Catalog](#)
- [AWS SPICE - importing data](#)
- [Accessing Data using JDBC on AWS Glue](#)
- [AWS Secrets Manager](#)

Use case #3: Event-based actions by Salesforce, improving decision-making with sentiment analysis that uses machine learning (ML)

Now that moving data between the Salesforce and AWS environment has been covered, the next item to consider is how to drive business value by infusing AWS Machine Learning throughout the business. This use case shows how [Amazon Comprehend](#), a natural language service that uses machine learning to find insights, can be used to analyze content of a customer support ticket and capture the customer sentiment from it. The goal is to analyze and record the sentiment in the customer record in real time as the ticket is updated but then analyze the overall customer sentiment content derived from analyzing all the customer tickets once a day (batch).

Amazon Comprehend identifies the language of the text, extracts key phrases and events, determines the measure of positivity or negativity in the text, analyzes text using tokenization and parts of speech, and automatically organizes a collection of text files by topic. With AutoML capabilities, it is possible to build a custom set of entities or text classification models that are unique to the requirements of an organization. A fully managed service, [Amazon Elasticsearch Service](#), which offers APIs, Kibana, and SQL querying, enables the search, analysis, and visualization of log data while using existing tools and code.

An alternative is using an open-source software library for advanced natural language processing (NLP), such as spaCy. Both alternatives offer tokenization for text and speech analysis coupled with machine learning models for classification. However, an open-source NLP library can increase developer time and costs and create code built only for one type of workload, which requires constant additional coding and maintenance. Since it is open source, bug fixes and the latest security patches may not be readily available. By contrast, Amazon Comprehend is a fully managed service and requires minimal NLP expertise. With spaCy, there may be API incompatibilities and issues with interoperating with Python, Java, and other API programming languages. By comparison, AWS APIs easily interoperate with other AWS services and familiar SDC.

For this use case, there are two options for invoking a data pipeline for sentiment analysis—using the platform events and event bus built into Salesforce to avoid hitting API limits or using a serverless event bus from Amazon, enabled by Amazon Appflow.

Salesforce Event Bus Option

This option uses the platform events and event bus built into Salesforce in an event-driven invocation of a data pipeline. Event frameworks enable the capture and creation of events from applications more easily that can then be processed to provide data in real time downstream for transformation and analysis.

For example, a user enters an opportunity record in the Salesforce platform. The user attaches a PDF or image to that record with a comment, and a platform event is sent via Amazon Simple Notification

Service (SNS). AWS is listening for events by polling the event bus. [AWS Lambda](#), which can run code without provisioning or managing servers, receives that event and archives the attachment in Amazon S3 and the opportunity record. AWS Lambda posts the comment to an API that sends it to Amazon Comprehend for analysis, which derives the sentiment. AWS Lambda then takes that output and sends it to Amazon Elasticsearch for indexing, summary, and aggregation.

The data then goes back into Salesforce by putting the link to the object in the Salesforce opportunity record. The opportunity record is updated with the sentiment and the Amazon S3 archive location. [Amazon Athena](#) is used to query and validate both events and the archived objects in [Amazon S3](#).

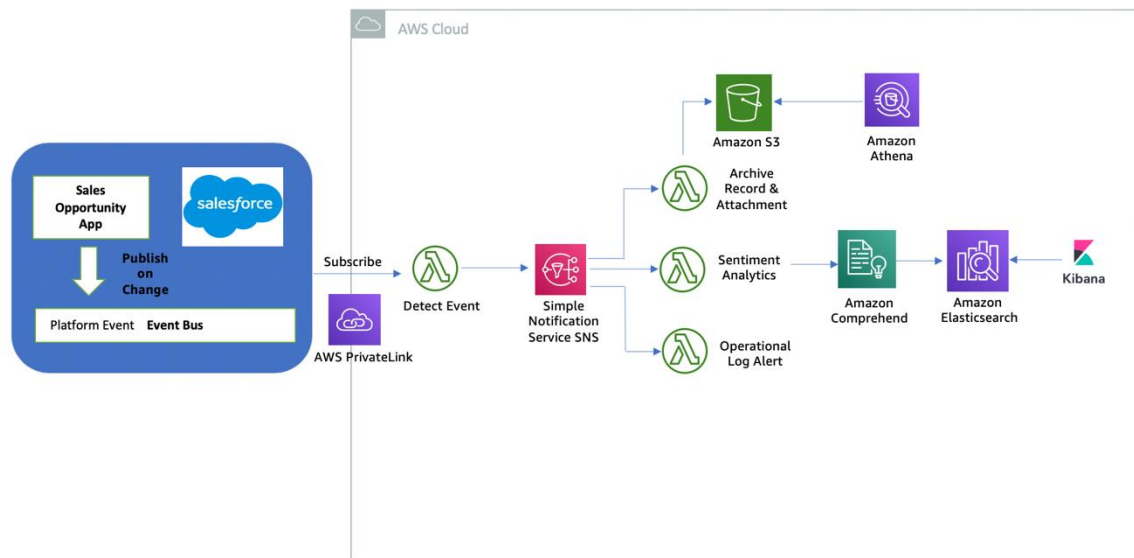


Fig. 7: Diagram of Event Based trigger, real-time sentiment analysis

This integration option can be tricky and complex. It involves developer time and specific Apex coding, Salesforce Lightning configurations, and platform events and event bus logs in Salesforce. The fact that Lambda polls for the event bus event log adds to the complexity and break points for automation based on code dependencies.

Serverless Event Bus: Amazon EventBridge

A recommended alternative is [Amazon EventBridge](#), a serverless event bus that makes it easy for organizations to connect applications using data from their own applications, integrated SaaS applications, and AWS services, by routing data to targets like AWS Lambda. Amazon EventBridge delivers a stream of real-time data from event sources and is supported as a destination by Amazon AppFlow. As the applications become more interconnected through events, more effort is required to find events and understand their structure so that code can be written to react to those events. The Amazon EventBridge schema registry stores event structure—or schema—in a shared central location

and maps those schemas to code for Java, Python, and Typescript so that it is easy to use events as objects in code.

In this use case, a real-time event-based trigger from Salesforce on any org initiates further downstream data analysis. By using AI to receive real-time feedback and sentiment from customers, customer service can be improved through valuable data insights. Teams can evaluate marketing documents or sales campaigns launched in Salesforce and determine the best approach. They spend less time and money trying to figure out which content or campaigns yield the highest customer success, satisfaction, and loyalty.

Related event bus and integration information

For articles about components in this use case, see:

- [Salesforce Event Bus](#)
- [Amazon SNS](#)
- [Amazon AppFlow and EventBridge](#)

Application use cases

Use case #4: Salesforce Service Cloud customers adding contact center capabilities for voice, chat, transcription, analytics, and AI/ML

Salesforce Service Cloud customers have two options for integrating Amazon Connect into their agent and supervisor experience:

1. Amazon Connect [CTI Adapter](#)
2. [Salesforce Service Cloud Voice](#)

CTI Adapter

In a time of uncertainty, businesses are pivoting so they can continue delivering products and services to their customers. These customers include first responders, restaurants, and health product distributors. By integrating [Amazon Connect with Salesforce](#), customer business units can quickly set up contact centers and realize the benefits with minimal development and expensive components to implement.

The technology responsible for integrating Amazon Connect and Salesforce is the [Amazon Connect CTI Adapter](#), which consists of a managed Salesforce package and an AWS serverless application deployed to an organization's AWS environment. The managed package provides the core CTI integration. The serverless repository enriches the integration with additional tools for accessing and analyzing data from Amazon Connect and Salesforce.

An example of Amazon Connect and Salesforce integration using the CTI Adapter, which can be accessed from Salesforce AppExchange for free, is a software add-on developed by a partner. This solution enables the quick and seamless deployment of an integrated contact center in the Salesforce platform and reduces the undifferentiated heavy lifting of a complicated setup.

In just a few clicks, a Salesforce environment provides an intelligent and responsive contact center that maximizes Salesforce integration with Amazon Connect. Amazon Connect includes voice, chat, analytics, chatbots, artificial intelligence, and machine learning capabilities to route calls appropriately and resolve issues quicker. The result is expedient problem resolution, fewer dropped or missed calls, greater operational efficiency, and increased customer satisfaction. Amazon Connect delivers voice capabilities with contact flows that control call flow actions, invoking [Amazon Lex Chatbots](#) based on [Amazon Polly](#), a service that turns text into lifelike speech. Amazon Polly enables developers to create applications that talk and to build entirely new categories of speech-enabled products. The Amazon Polly Text-to-Speech (TTS) service uses advanced deep learning technology to synthesize natural sounding human speech, and Amazon Polly Brand Voice can create a custom voice for an organization.

Amazon Lex is a service for building conversational interfaces into any application using voice and text. It provides the advanced deep learning functionality for automatic speech recognition (ASR), which converts speech to text, and natural language understanding (NLU), which recognizes the intent of the text. Lex powers Amazon Alexa enabling so that building sophisticated, natural language, and conversational bots (“chatbots”) is easy.

[Contact Lens for Amazon Connect](#) is a set of machine learning (ML) capabilities integrated into Amazon Connect. These analytics provide insights into the actual conversations between agents and customers, such as whether agents are communicating effectively, whether there are any interesting trends in customer sentiment, and whether agents are complying with regulatory requirements.

Another real-life use case of Amazon Connect and Salesforce integration involves first responders with PPE delivery delays. The fire chief calls the PPE supplier about the delays. Call volumes happen to be high at the time of day he is calling. However, the fire chief advances in the queue because of the call flow configuration that prioritizes first responders.

The fire chief routes through a series of prompts and interactive chatbots that gather his order information, noting the quantity and urgency. An agent working in Salesforce then receives the call to triage the delivery delay. The chief voices his concern about the current supplies of PPE running low at his firehouse. The agent says she will check into earlier delivery dates. Analytics pick up the chief's sentiment, which triggers an inventory search in Salesforce Platform, which locates the amount of PPE needed. Before the end of the day, the agent calls the chief back with confirmation of the delivery date. This is just one example of a no-code, fully supported integration between Amazon Connect and the Salesforce platform.

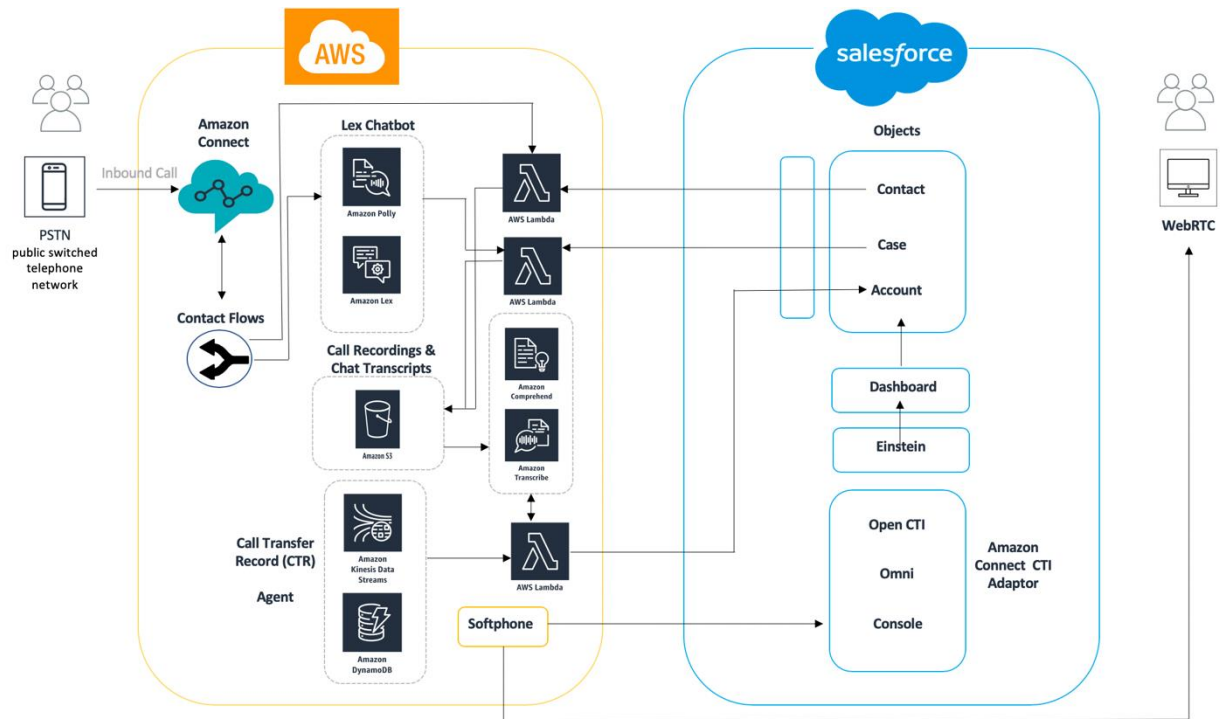


Fig. 8: Diagram of Amazon Connect and Salesforce platform integration

Service Cloud Voice

Customers can add contact center capability natively to the Service Cloud console through [Service Cloud Voice](#), which removes the burden of configuration, maintenance, and supportability associated with the CTI Adapter. [Service Cloud Voice](#) is available for purchase exclusively from Salesforce through an add-on SKU to Service Cloud. It allows teams to use the solution right out the box, realizing benefits immediately. This fully integrated contact center solution is the result of Salesforce addressing customer-identified gaps in the CRM landscape and strategically collaborating with AWS to deliver a deeply integrated solution based on CTI adapter code nested in a Visualforce framework.

Service Cloud Voice unifies phone, digital channels, and CRM data in real-time in one centralized console. This “contact center of the future” is intelligent, unified, and fast. Companies can integrate telephony and route calls on a single platform: Agents can work from a single, powerful console, eliminating the need to toggle back and forth from multiple screens and find manual workarounds. With call transcriptions surfacing in real-time directly in the Service Cloud console, Einstein AI can provide recommended solutions and next best actions for the agent in the background.

Related Amazon Connect and Salesforce integration technical information

For documentation, videos, tutorials, and guides related to Amazon Connect and Salesforce integration, see:

- [Salesforce Service Cloud Voice](#)

- [Amazon Connect Documentation](#)
- [Amazon Connect Salesforce Integration Guide](#)
- [Amazon Connect CTI Adapter for Salesforce Lightning Setup and Installation Guide](#)
- [Amazon Connect Training Videos](#)
- [Build an Amazon Connect Integration](#)
- [Build an Amazon Connect SSO Integration with Salesforce](#)

Use case #5: Integrating Alexa for Business to add voice and connected voice devices while using Salesforce

In this integration use case, Alexa for Business is integrated with Salesforce through a voice application (also known as an Alexa Skill) to enhance business workflows and collaboration. This type of integration enables businesses to use voice-controlled devices to work interactively with their Salesforce platforms.

For example, a sales team is attending their quarterly sales review meeting. All members work from home and have an Alexa Echo device. They are heavy Salesforce users. Instead of logging in to the Salesforce platform to get information, they decide to launch an Alexa Skill from an Alexa Echo. They ask for the latest top five opportunities and for the most current wins and losses report. The Alexa Skill retrieves this information from Salesforce and reads it to the sales team. Before finishing the meeting, the sales manager asks Alexa to schedule the next sales meeting in the corporate calendar.

Alexa for Business enables developers to create and manage private Alexa Skills like the ones in the example that can be accessed exclusively by company employees. Alexa Skills can be coded by individual developers (using NodeJS, JavaScript, or Python code hosted in AWS). This code connects to Salesforce Flows via API. To simplify the creation of an Alexa Skill, there is an option to use pre-built no-code packages from the AWS Partner Network (APN), like VoiceWorx.ai.

The Amazon Skill back-end modules run in the background to authenticate and route requests to Salesforce and Flows for processing via API. The Salesforce objects can be accessed directly via REST APIs or the salesforce SDK. Salesforce Apex flows can be accessed programmatically via web services hosted on the Salesforce platform. The flow action can access data in Salesforce, either Sales Cloud or Service Cloud, and retrieve information from individuals and customers.

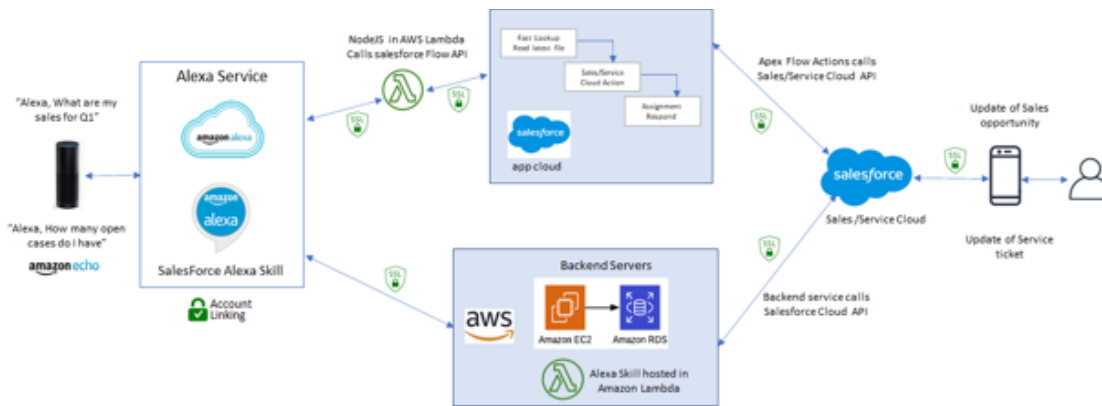


Fig. 9: Diagram of Salesforce Apex flows

Alexa for Business and Salesforce integrations use Alexa Skills and Echo devices to boost voice experiences that enable front-office and customer service teams to access sales data and marketing campaign reports, register new opportunities seamlessly, create new cases in Service Cloud, or access insights and analytics from Einstein on the fly.

Related Alexa for Business and Salesforce integration technical information

For quick starts and tutorials related to this use case, see:

- [Amazon Connect Voicemail Quick start](#)
- [Amazon Connect Data Streaming Quick start](#)
- [Build an Amazon Connect Integration](#)
- [Build an Amazon Connect SSO Integration with Salesforce](#)
- [Innovate with Alexa and Amazon Web Services](#)
- [Alexa-Salesforce Trailhead](#)

Conclusion and next steps

This whitepaper outlined how organizations, users, and customers can take advantage of the integration components, applications, and core services available from the AWS and Salesforce platforms. It demonstrated how the strategic partnership between these two industry leaders enables their customers to overcome the critical business and integration challenges of digital transformation and innovation by accelerating SaaS and cloud adoption.

Secure, bi-directional data exchange, analysis, and visualization are at the core of these integrations. A common theme for the five integration use cases in this paper were the solutions available at the integration layer and a data pipeline that stitches data lakes, data warehouses, and other data storage together. These data pipelines also meet customer requirements for AI and machine learning and real-time analytics.

In addition, integration services like Heroku and MuleSoft deliver applications and APIs to market quicker with data rich features and visualization. Amazon AppFlow and Salesforce Private Connect, two services aimed squarely at a more cloud-native Salesforce and AWS integration, expand the use of Salesforce applications and add-ons with fully integrated services.

Customers, independent of the environments they are operating in, have no-code, low-code, and out-of-the box solutions to try out and modify to fit their security, workload, or data requirements. Customers are now equipped with the knowledge, architectures, references, and integrations to build a more cost effective, scalable solution. There is now greater depth and breadth of what can be achieved with Salesforce and AWS core services and platforms.

To get started on your AWS and Salesforce integration journey, explore the resources provided at the end of each use case.

For more general partnership information, visit:

- [AWS and Salesforce Partners](#)
- [Best Practices for Integrating and Securing Salesforce Data with AWS](#)
- [AWS Well-Architected Framework](#)