splunk>

Levelling up your AWS Connect with

Overview

In our new world of working from anywhere having a reliable contact center is more critical than ever. AWS Connect offers customers the ability to spin up a scalable omnichannel customer service experience in just a few clicks.

But getting easy to digest visibility into how your omnichannel platform is performing isn't easy. Splunk can help AWS connect customers get a full and easy to digest view into the performance and customer experience of your AWS connect platform. This includes logs, metrics, sentiment analysis and more.

This tech brief will take you through everything you need to know about configuring your AWS Connect platform to talk to a Splunk Cloud Instance.

Note: This tech paper has been written with one purpose in mind. To allow the reader to configure their AWS Connect instance to ingest data into Spunk Cloud without the need to open the inevitable one hundred tabs usually required when doing anything like this.

Every deployment and configuration of AWS Connect and Splunk is different. This paper will try as best as it can to point out best practice vs what is right for your organisation.

AWS Connect Splunk Ingest Points

AWS Connect has several ingest points which will configured to send data into Splunk. These are outlined below
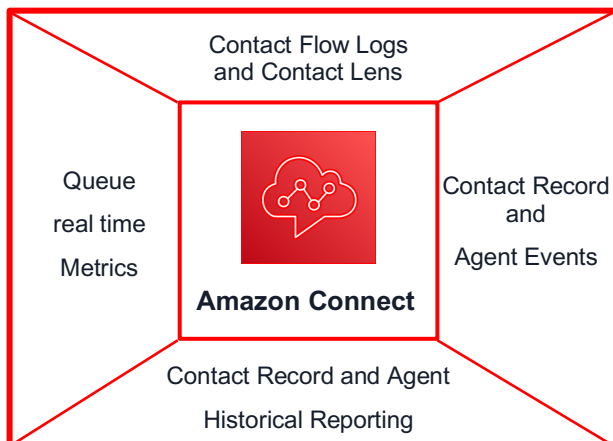
Queue Realtime Metrics: Connect API metrics captures both historic and real time queue metric data from the application's API endpoints

Contact Flow Logs: Contact flows are used within the AWS connect product to allow an administrator to define a customer flow experience from start to finish. When an AWS connect instance is created an AWS CloudWatch log group is also created.

Agent Events: Near real time streams report on agent activity within your AWS connect instance. These include:

- Agent Login / Logout
- Agent connects with a contact
- Agent status change, such as ability to handle contacts or on a break/ training

Contact Record (CTR) Events: Near real-time streams of contact (voice calls, chat, and task) events (for example, call is queued) in your Amazon Connect contact center. These events include:

- Initiated or transferred a voice call, chat or task
- The date or time the customer endpoint connected to AWS Connect
- A voice call, chat, or task is queued to be assigned to an agent.
- A voice call, chat, or task is connected to an agent
- A voice call, chat, or task is disconnected

Contact Record (CTR) and Agent Historical Reporting: Used to capture data about the past, historical reports are stored based on a configured report and schedule. The reports are stored within a S3 bucket for future retrieval.

Contact Lens for Amazon Connect

Contact Lens for Amazon Connect helps you follow the sentiment and trends of customer conversations in real time to identify crucial company and product feedback. You can also track the agent compliance of customer conversations in your contact center to ensure standard greetings and sign-offs are used, help train agents, and replicate successful interactions.

Splunk Required Apps and Add-Ons

Throughout this guide we will use several Apps and Add-Ons available free on the splunkbase
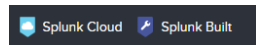
> **FUN FACT:** Do you know the difference between a Splunk Add-On and a Splunk App (or trusted app, TA)?
>
> The easiest way to think about is it a Splunk App will show you something, eg dashboards or a visualization. An Add-on will enhance your Splunk functionality.

Required Add-Ons

Splunk Add-On for Amazon Kinesis Firehose

The Splunk Add-on for Amazon Kinesis Firehose allows Splunk software to receive (*push*ed) data from AWS Kinesis Firehose. Splunk has pre-built integration with AWS Kinesis Firehose as a recognized 'Destination' endpoint. This add-on is build, maintained and supported by Splunk.

Splunk Add-On for Amazon Web Services (AWS)

The Splunk Add-on for AWS allows Splunk software to collect a whole range of data from AWS by *pulling* data via authenticated API calls. This add-on is build, maintained and supported by Splunk.

Required Apps

Splunk App for Amazon Connect

The Splunk App for Amazon Connect is the main app which has pre-configured dashboards and visuals for gaining insights into the performance of your AWS connect platform. The App is developed by Splunk and aimed at being am excellent starting point for AWS Connect customers. This app does not carry standard Splunk support.

Splunk Timeline – Custom Visualization

The Splunk Timeline -Custom Visualization App allows Splunk users to gain better visualization in dashboards and reports. This is required by the Splunk App for Amazon Connect to build our custom visualizations build within the dashoards. Built and Supported by Splunk.

Event Timeline Viz

The Splunk Event Timeline Viz is similar to the Splunk Timeline – Custom Visulization App above allowing Splunk users to get better event timeline visualizations. This is again required by the Splunk App for Amazon Connect in some of the custom dashboard visualizations. This app is not Supported by Splunk

Assumptions and Starting point

As cool as it would be this guide cannot cover everything. Below is a list of some required items.

It is also assumed you have the relevant access to AWS Cloud Instance where you Amazon Connnect is running as well as your Splunk Cloud Instance.

Required

- Splunk Cloud
- AWS Connect deployed and configured appropriately within your AWS tennancy

High Level Architecture

Below are the main functional interactions points between AWS and Splunk including data flow direction (pull vs push).
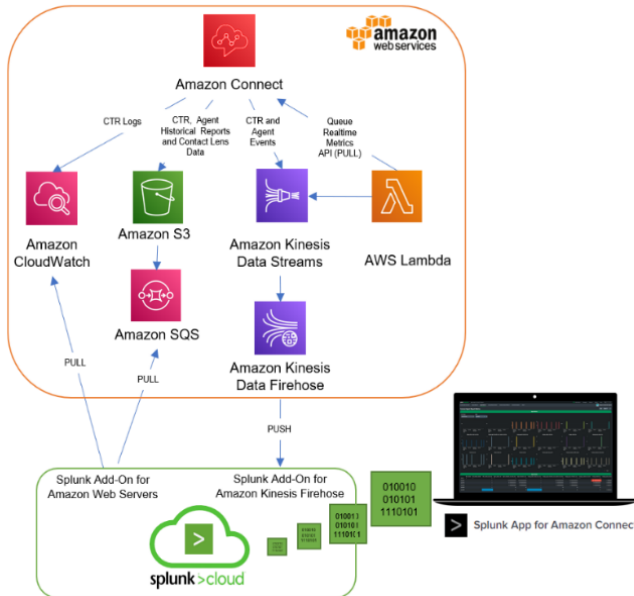
Installing the Splunk Apps / Add-Ons

Below are the steps required to install the Splunk App for Amazon Connect. This technical brief will not go through every app and add-on but the steps are primarily the same.

> **RESTARTS:** Some Add-Ons require a restart of the Splunk Cloud platform. If prompted to do so you can restart either at the very end of installing all the required apps/add-ons of restart when prompted.
>
> **IMPORANT:** Before restarting you should follow your companies change control policies as restarting your Splunk Cloud instance may cause outages or disconnections of the Splunk platform. More information can be found here.

From your Splunk Cloud instance **login** as a user that **has permissions to install applications**.

There are a number of ways you can install applications but for this guide we will **click on the Splunk cloud logo** on the top left corner.

From there you should see **+Find more Apps**. Click it and on this page search for **Connect**, this will return within the results **Splunk App for Amazon Connect** as shown.
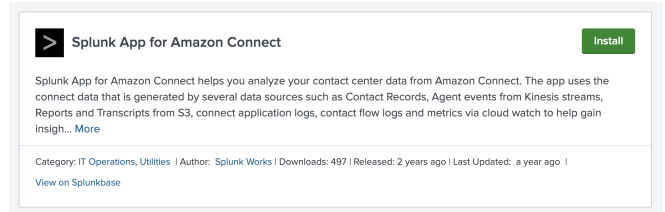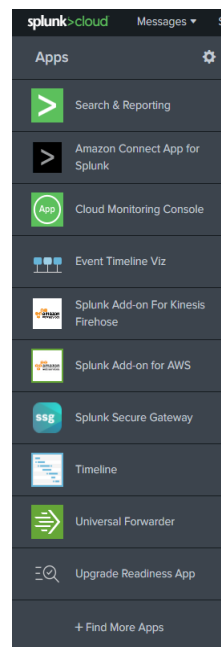

Click **Install**

Enter your Splunk username and password and tick the box agreeing to the terms and conditions.

Once completed it will appear on your list installed applications. Repeat this process until all of the required apps / add-ons are installed on your Splunk Cloud instance.

Splunk Add-On for Amazon Kinesis Firehose ✓

Splunk Add-On for Amazon Web Services (AWS) ✓

Splunk App for Amazon Connect ✓

*Splunk Timeline – Custom Visualization* ✓

Event Timeline Viz ✓

Once completed your applications list should look something like the picture shown here.

Creating an AWS Connect Splunk Index

It is recommended that you create a specific Splunk Index for storing your AWS connect ingested data.

Within your Splunk Cloud console click the **Settings** followed by **Indexes** (under DATA section)
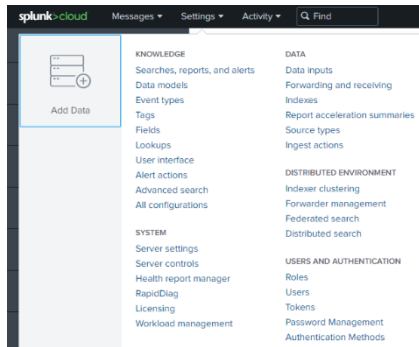
From the Index screen, click **New Index**.

Choose an appropriate **name** for your index.

Select **Events** as your Index type.

Max raw data size can be set to **0GB** (unlimited) and searchable time (days) to your preferred timeframe. **We are going to use 365 days**

> **Note:** *Max raw data size* and *Searchable time (days)* fields as shown in example may vary depending on your organizations policies and procedures in regards to creating Splunk indexes. Adjust as required.

For **Splunk Cloud Victoria** customers you will see an additional options for Dynamic Data Storage (for this example we will select **No Additional Storage**)

Splunk Classic View


Splunk Victoria

Once completed click **Save** to create the new index.

Create a Splunk HEC token

A Splunk HEC (HTTP Event Collector) service is used to collect data from many different sources. In this instance it will be the primary ingestion service for our AWS firehose streaming services.

A Splunk Cloud instance can have many HEC endpoints. We will create a new one specifically for you're the AWS connect data.

Create a HEC endpoint by clicking on **Settings** followed by **Data Inputs** (under the DATA section)
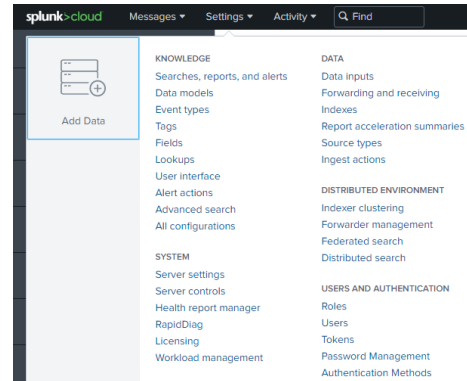
Click **+Add new** on the HTTP Event Collector

Enter a **Name** and optional description for your HEC endpoint. For AWS Connect we recommend leaving **Source name override empty** and **HEC acknowledgment** options **ticked.**

Click Next once done.

> **NOTE:** HEC acknowledgement allows the AWS Kinesis firehose to know that the data sent into Splunk has been successfully sent, without this the retry feature of firehose will be invoked. There is also a different between acknowledgement of data successfully sent vs indexed. For more information on this please see the documentation linked here.

On the **Input Settings** page click the **Select** option under Source type

Select **aws:connect:firehose** as the source type to use for this HEC token

Select the index you created earlier as your **allowed** and **default index's**.

Example below

Click **Review** to review your Setup. Click **Submit** to create your token when ready.

Copy your HEC token down somewhere for future use. You can find it again later by going back to your Settings, Data Inputs, HEC section.

AWS Connect IAM Policies

The following AWS IAM policies need to be created in order for the Splunk Add-On for AWS to be authenticated to pull data out of AWS. From our earlier diagram this is relating to the following ingests paths.
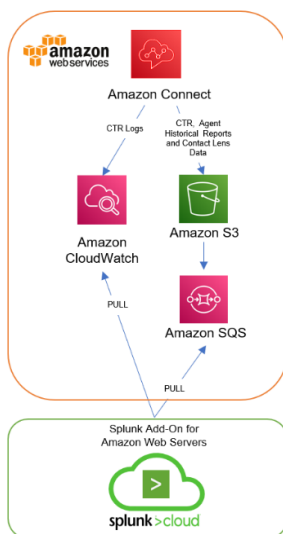
Create required policies by searching and clicking on the **IAM** section within your **AWS console**.

Select **Policies** and click **Create Policy**

Select the **JSON** tab and past the following policy JSON code into the editor. For more information see

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid":
"MinPermissionsRequiredToAllowUseOfSQSBasedS3Actions",
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:SendMessage",
                "sqs:DeleteMessage",
                "sqs:GetQueueAttributes",
                "sqs:ListQueues",
                "s3:GetObject",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

Click Next: Tags and add any tags required by your organisation

Click **Next: Review** and enter in a **Name**, **Description** (optional) and finally **Create policy**

This policy is specifically for viewing and modifying SQS related activities and the ability to get an object from S3.

Create another policy for reading in Amazon CloudWatch data.

Repeat steps above but this time insert the following JSON code. For more information see

```
{
    "Statement": [
        {
            "Action": [
                "cloudwatch:List*",
                "cloudwatch:Get*",
                "autoscaling:Describe*",
                "ec2:Describe*",
                "s3:List*",
                "sqs:List*",
                "sns:List*",
                "lambda:List*",
                "elasticloadbalancing:Describe*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ],
    "Version": "2012-10-17"
}
```

NOTE: Policies listed here are a guide and should be checked by your internal AWS or Security team in order to make sure they comply with your AWS security policies.

Creating an AWS IAM User

An AWS IAM programmatic user is required for Splunk to authenticate into AWS to pull back data.

This is used by the Splunk Add-on for AWS as shown in the diagram earlier.

Login to your AWS console and enter **the IAM section** of your AWS tenancy.

In IAM section click **Users** followed by **Add User**.

Enter in a **name** for your AWS Splunk user and select the **Access key – Programmatic access** option.

Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name*   aws-splunk-user

Add another user

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. Learn more

Select AWS credential type*   ☑ Access key - Programmatic access
Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.

☐ Password - AWS Management Console access
Enables a password that allows users to sign-in to the AWS Management Console.

Click **Next: Permissions** and select the box saying **Attached existing policies directly**.

Select the **policies** you created earlier followed by **Next: Tags**

Add any tags required by your organisation and click **Next: Review** when ready.
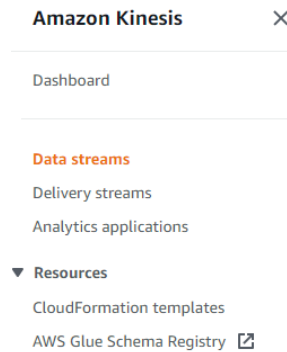
Review your user details. Example shown below. Click **Create user** when ready

Once user is created either **copy** your **Access key ID** and **Secret access key** or click the **Download .csv button** to download a copy of the new use credentials.

AWS Streams and Firehose Connections

AWS Connect supports AWS Data streams as a method of data streaming. In order to stream data into Splunk we need to first create an AWS Kinesis Data Stream and the relevant Kinesis Data Firehose connection to Splunk.

From your AWS Console search for **Kinesis Data Streams** and select it from the search list.

Amazon Kinesis   ✕

Dashboard

Data streams

Delivery streams

Analytics applications

▼ Resources

CloudFormation templates

AWS Glue Schema Registry ⬏

Select **Data streams** from the left-hand side and then click **Create data stream**

Enter a **Data Stream Name** and the click **Create data stream** leaving capacity as **On-Demand**.

Next Click **Delivery Streams** from the left-hand side followed by **Create delivery stream button**.

From the Create a delivery stream wizard choose **Amazon Kinesis Data Streams** as the **Source** followed by **Splunk** as the **Destination**. This will then expand out the rest of the options as per picture below

Amazon Kinesis > Delivery streams > Create delivery stream

Create a delivery stream Info

▶ Amazon Kinesis Data Firehose: How it works

Choose source and destination
Specify the source and the destination for your delivery stream. You cannot change the source and destination of your delivery stream once it has been created.

Source Info
Amazon Kinesis Data Streams ▼

Destination Info
Splunk ▼

Tags
No tags were added.

Browse and **Select** your **Kinesis data stream** you created earlier

Enter in a **Delivery stream name**

Delivery stream name

Delivery stream name
KDS-Splunk-Data-Stream

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

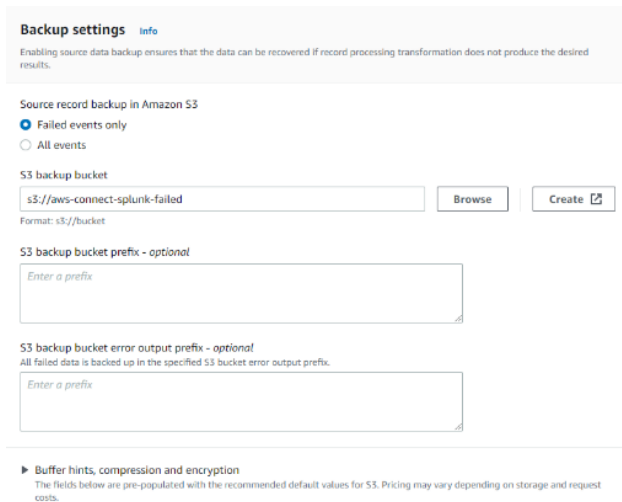Leave the **transform records** section as **Disabled.**

Enter in your **Splunk cluster endpoint URL** into the Splunk cluster endpoint section.

Leave the Splunk endpoint type as **Raw endpoint**.

Enter in the **HEC authentication token** you copied earlier into the Authentication token box

Leave the **timeout settings** as the **default**

For **Backup settings** select Failed events only and you can then select an existing bucket or choose to create a new S3 bucket for any failed messages

Backup settings Info

Enabling source data backup ensures that the data can be recovered if record processing transformation does not produce the desired results.

Source record backup in Amazon S3
◉ Failed events only
○ All events

S3 backup bucket
| s3://aws-connect-splunk-failed | Browse | Create ⧉ |
Format: s3://bucket

S3 backup bucket prefix - *optional*

Enter a prefix

S3 backup bucket error output prefix - *optional*
All failed data is backed up in the specified S3 bucket error output prefix.

Enter a prefix

▶ Buffer hints, compression and encryption
The fields below are pre-populated with the recommended default values for S3. Pricing may vary depending on storage and request costs.

Click **Create Delivery Stream** when ready
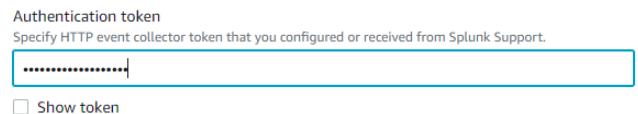
AWS Connect real time metrics API

The AWS Connect has a real time metrics API which can be used to pull data from AWS Connect and send to a kinesis streams and then through to Splunk. It's a combination of both pull and push.

To do this we need to first configure an AWS lambda function to run and pull the data from the API and send it to a chosen kinesis stream (one we created earlier). More details here
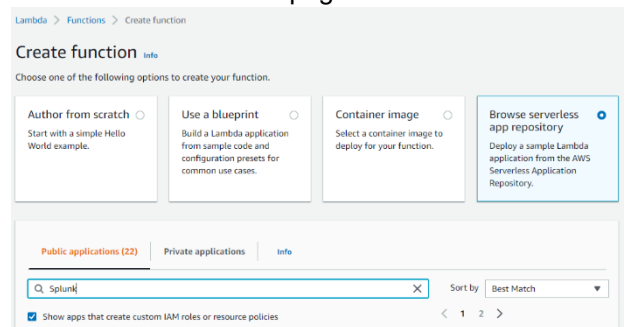
The lambda function is build and maintained by Splunk and available via the AWS lambda console.

To set this up, search and click on **lambda** from the **AWS console.**

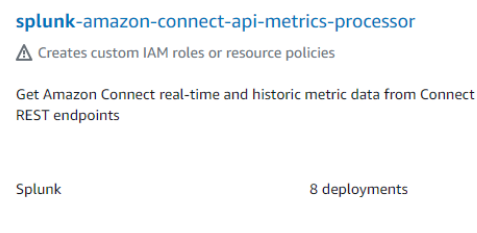From the AWS lambda **select Functions** from the left-hand menu followed by **Create function**

Authentication token
Specify HTTP event collector token that you configured or received from Splunk Support.

| •••••••••••••••••• |

☐ Show token

On the create function page select **Browse**

Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

| Author from scratch ○ | Use a blueprint ○ | Container image ○ | Browse serverless app repository ◉ |
| Start with a simple Hello World example. | Build a Lambda application from sample code and configuration presets for common use cases. | Select a container image to deploy for your function. | Deploy a sample Lambda application from the AWS Serverless Application Repository. |

Public applications (22) Private applications Info

| 🔍 Splunk ✕ | Sort by Best Match ▼ |
☑ Show apps that create custom IAM roles or resource policies   ‹ 1 2 ›

serverless app repository and enter in **Splunk** in the **search box**. Also tick the box for **Show apps that create custom IAM role.** See example below

Find an application called **splunk-amazon-connect-api-metrics-processor.** It maybe on the second page.

splunk-amazon-connect-api-metrics-processor

⚠ Creates custom IAM roles or resource policies

Get Amazon Connect real-time and historic metric data from Connect REST endpoints

Splunk   8 deployments

Select this and you will be taken to the next screen to fill in the application details.

Enter an **Application Name.** This can be anything you will use to uniquely identify your application.

Fill in the **Connect Instance ID**. This can be found by clicking on your AWS Connect instance and

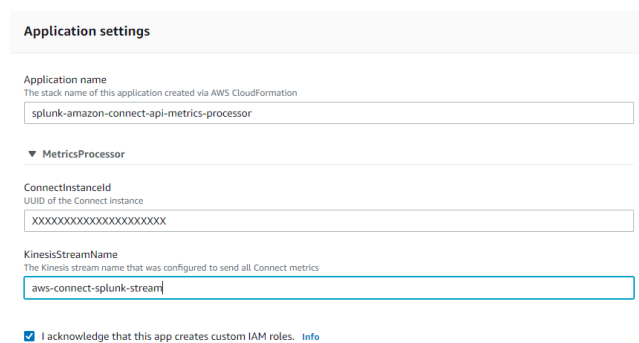grabbing the last remaining characters from your AWS Connect ARN.

Connect Instance ID Example: arn:aws:connect:<REGION>:<AWSACCOUNTID>:instance/**<CONNECT INSTANCE ID>**

Enter the name of the **Kinesis Data Stream** you created earlier (The data stream, not the firehose stream). See example below.

> **NOTE:** Before saving check that you do not have spaces at the start or end of your kinesis data stream when you enter it into Amazon Lambda as described above, this is tricky to trouble shoot later.

Tick the box for **acknowledging** the IAM roles.

Click **Deploy** once ready and once deployed it will be in your AWS Lambda application list.

Configure a SQS Queue for Splunk

An SQS queue is used to monitor the S3 buckets for when reports or outputs from AWS connect get transferred into a S3 bucket.

Search and click on the **Simple Queue Service** in the **AWS Console**

Click on **Create Queue**

Name the queue and set the **visibility timeout** settings to **5 minutes**

Under **Access Policy** select the **Advanced** radio button and **copy the existing block of JSON code** into a separate text editor.

Replace that JSON code in AWS console with this code below modifying the sections in **<xxx>**

```
{
  "Version": "2012-10-17",
  "Id": "SQS-policy",
  "Statement": [
    {
      "Sid": "s3-principal",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
```

```
      "Resource": "<the ARN of the SQS queue>",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<your AWS account number>"
        },
        "ArnLike": {
          "aws:SourceArn": "<the ARN of the Amazon Connect S3 bucket>"
        }
      }
    }
  ]
}
```

> **NOTE:** SQS queue ARN and Account ID will be in the code copied earlier. The AWS Connect S3 bucket ARN can be found by going to the AWS S3 bucket service and selecting your S3 bucket corresponding to your AWS Connect instance. From here the ARN will be under properties.

Leave the rest of the defaults and click **Create Queue** when ready.

Repeat the exact steps you just did for a **second queue**. But this time we want to append the word **_DEADLETTER at the end** of the queue name.

Once completed you should have **two** SQS queues like the example shown below.

Now we need to **click on the first** (non DEADLETTER one) and **select the Dead-letter queue tab**.

Click **edit** and **scroll down to the Dead-letter queue** section **expanding it**.

Select the **Enable** option and **choose your dead-letter queue** you created earlier.

Leave retries as is.

Click **Save** once done.

Configure S3 Bucket to trigger SQS

From the AWS console, **search and click on S3**

Select your **AWS Connect S3 bucket** followed by the **Properties tab.**

Scroll down to the section **Event Notifications** and click the **Create event notification**.

In the **General Configuration choose a name for the event.**

Under **Prefix**, set this to the following

connect/<CALL CENTER NAME>/Reports/

Set the **Suffix** to be **.csv** as this is the format the reports are written in.

Under Event Types select the **All object creation events** tick box on the left hand side as shown.

Leave all other tick boxes empty.

Under Destination select **SQS queue** and then select **Choose from your SQS queues**.

Select the queue you created earlier from the drop-down list. Example below

Click **Save changes** once done.

Repeat the exact steps above **for Creating an event notification** but this time create one for the **AWS contact lens information**.

The only difference is the prefix should be **Analysis/Voice/** and the Suffix should be **.json** all other steps including SQS queue should be the same.

Once completed you should see **two event notifications** in your list.

Configure Amazon Connect to Stream Data

Configure Amazon connect to send agent and CTR data to the AWS data streams.

Search and **select the Amazon connect** from the AWS console. Select your **Amazon connect call center**.

On the left hand side panel select **Data Streaming**

Tick the **Enable data streaming** option and **select the Kinesis stream radio button**

Under both **Contact Trace Records** and **Agent events select the kinesis stream you created earlier** from the drop down list. Example below

Click **Save** when done

Enable Contact Flow Logs

Search and **select the Amazon connect** from the AWS console. Select your **Amazon connect call center**.

On the left hand panel, **select Contact flows**

Under Contact flow logs, **tick the Enable Contact flow logs** if not already ticked.
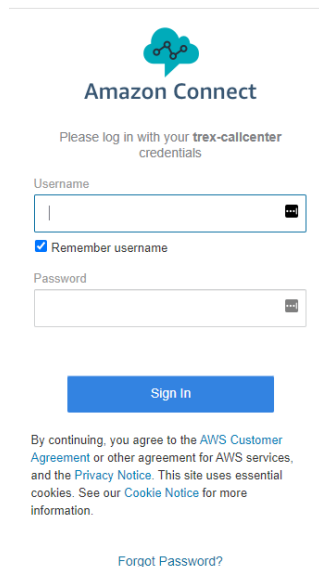
Click **Save** to update the details.

Configure Amazon Connect Contact Lens

AWS Connect contact lens needs to be enables within an Amazon Connect contact flow.

Search and select for **Amazon Connect** from your AWS console.

Click on your call center from your list.

From the main overview screen **select the Access url** to open a new tab and **login to your Amazon Connect Call Center**

From your Amazon Connect Dashboard on **the left hand side panel** select **Routing** then click the **contact flows menu option**.

Select the contact flow you wish to enable contact lens on.

As part of your contact flow you should have a **Set recording and analytics behavior contact flow box**.

Click this and a **side panel should appear**

From here, **tick the Enable Contact Lens for speech analytics** followed by the **Read-time and post-call analytics**

Contact Lens speech analytics

☑ Enable Contact Lens for speech analytics.

Contact Lens provides ML-based speech analytics such as call transcript and sentiment to identify customer issues and trends. Use post-call analytics to get speech analysis for transcripts, sentiment, and other data, after customer interaction. Use real-time analytics to get access to transcripts and sentiment analysis during the customer interaction. Learn more

○ Post-call analytics

Recommended for best transcription accuracy.

◉ Real-time and post-call analytics

Enables real-time alerts and speech analytics on live calls.
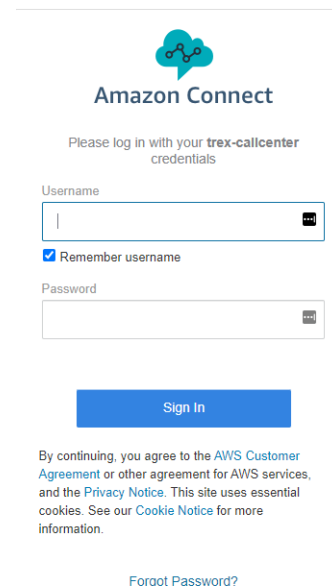
Click **Save** when done

> **NOTE:** If you do not have the Set recording and analytics behavior contact flow box then you will need to add one to get call recordings and contact lens working. Your Amazon Connect team or partner will help you to implement this option appropriately for your call center.

## Scheduled Amazon Connect Reports

Amazon Connect can run and schedule regular reports in .csv format. These will then be ingested into Splunk via the Splunk Add-On for AWS.

To create and schedule a report login to your Amazon Connect instance by **searching and selecting for Amazon Connect** from your AWS console.
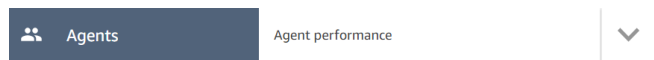
From the Amazon Connect overview screen **select** the **Amazon Connect instance url** and **login to the Amazon Connect Instance**

From here select the **Analytics option** from the **left-hand menu** followed by the **Historical metrics** option
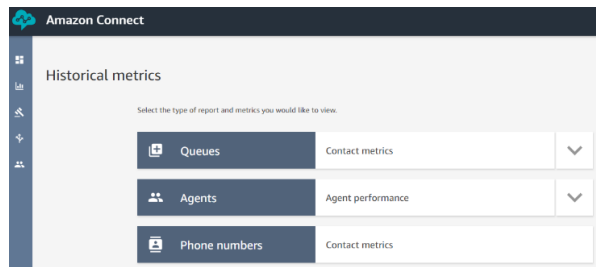
First create an Agent performance report by selecting the **Agent performance box**

From the **Historical metrics: Agents** section **click** the **small grey cog** on the right hand side.

Under the **Interval and Time range** click the Time range drop down and select **Today (since 12am)**

While in the same section **select the drop down** for **Time Zone** and **select the time zone** format for the report.

> **NOTE:** The time zone selected in this section is used to align the time zone to when the day starts, eg 12am (midnight) in a time zone of your choice. Usually, the time zone you are operating your call center in.

Select the **Metrics Tab** and then proceed to **tick** *EVERY* **tick box** making sure you **scroll down** as well. This will collect all metrics.

Click **Apply** to build the report. Once the report has been run we now need to create a schedule for the same report to continue to be run.

Select the **drop arrow next to the Save button** and click the **Schedule** option
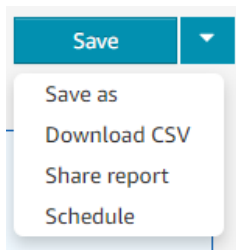

Name the report example as below **historic_agent_performance_all_metrics-<XX>**

The <XX> can be replaced with something specific for your organisation but the **prefix section must remain the same**.

This ensures that the Splunk App for Amazon Connect reporting dashboard works correctly out of the box. Example below
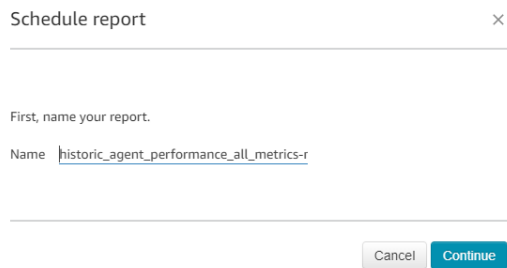
Click **Continue** once named

Click **Continue** again to accept the reporting note.

Select the report to be **generated every hour** for the **previous 1 hour**. See example
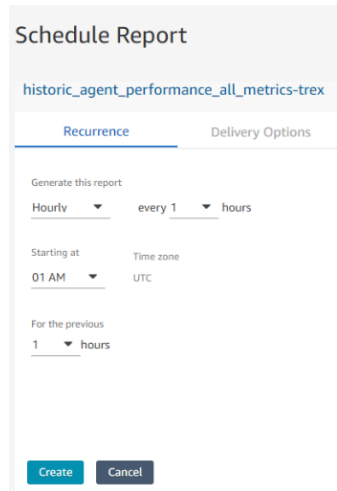
Click **Create** once done.

Repeat the steps above for creating and scheduling report but this time **select Queues** instead of Agents instead.
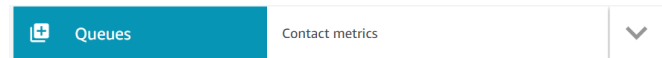
Make sure you name the report **historic_queue_contact_all_metrics-<XX>**

Once completed, select **Analytics**, **Saved Reports** option from the left-hand side
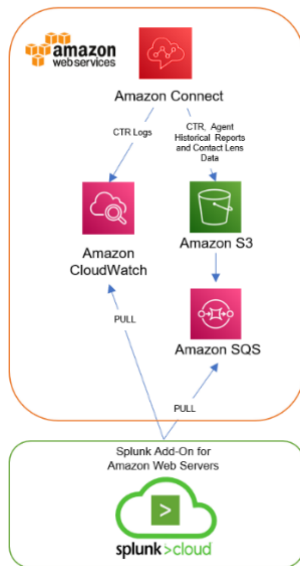
Then select **Historical Reports**. You should see your two reports created. Example below
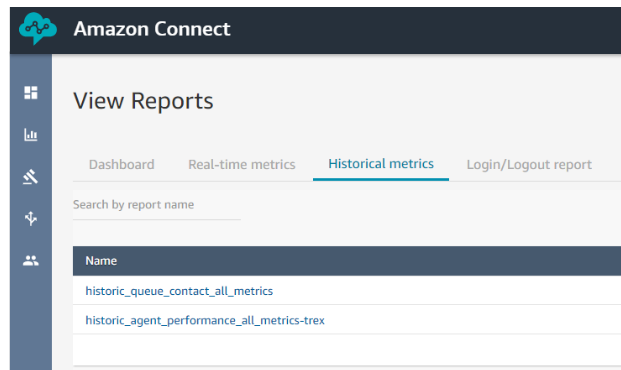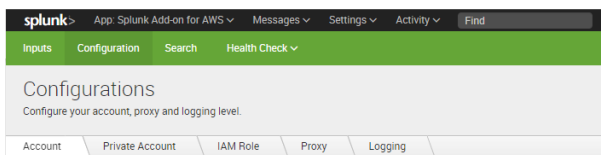
Configure Splunk Add-on for AWS Inputs

Next we need setup the PULL sections from the Splunk Add-on for AWS to ingest our reports as depicted in the diagram below

Login to Splunk Cloud and from the **Apps drop down** list **select Splunk Add-on for AWS**

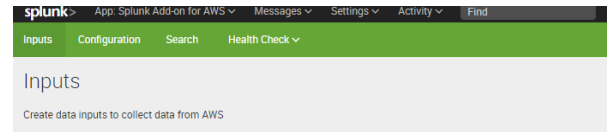Select the **Configuration tab** from the top section.

From the right hand side, click the **Add button**.

Enter a **Name** (describing the user account) and then copy in your **Key ID** and **Secret key** from you excel sheet you downloaded when creating the AWS IAM user earlier in this guide.

Select your **Region** and click **add**

Your username should now appear in the list under configurations.

Next **select** the **Inputs Tab** on the top left corner.

From the **Create New Input drop down** on the right-hand side select **Custom Data Type, SQS based S3** from the list.

Name the AWS input something meaningful, eg Historic_Agent_Queue_Reports

Select from the **drop down list** the **AWS Account** you configured earlier.

Select your AWS region where your AWS Connect instance is running. This will then populate the list of SQS Queue's.

Select the SQS queue you created earlier

Modify the Source Type name to be **aws:connect:s3:reports**

Set **Index** to be the **index you created earlier**

leave remaining defaults and click **Submit** once completed

The input will now be configured.

Repeat the steps above creating **another Input** of the same type **custom SQS base S3** but this time for our contact lens files and using the Source Type of **aws:contactlens:s3**

Lastly, configure a final input for the Amazon Connect CloudWatch logs by selecting the **Create New Input button**, **Customer Data Type**, **CloudWatch Logs**

Similar to before **enter in a name**, **your AWS account**, **region where your Amazon Connect instance is running**

For Log Group enter in the following details

/aws/connect/<*CALL CENTER NAME*>

For Source Type enter in **aws:connect:cloudwatchlogs**

Set **Index** to be the **index you created earlier**

Click Save to **save** the input

Configure CloudWatch Metrics (custom)

In order to extract the Amazon Connect CloudWatch metrics we need to configure a fourth Input in the Splunk Add-on for AWS app.

From the Splunk Add-on for AWS under **Inputs**, select **Create new Input** then **CloudWatch.**
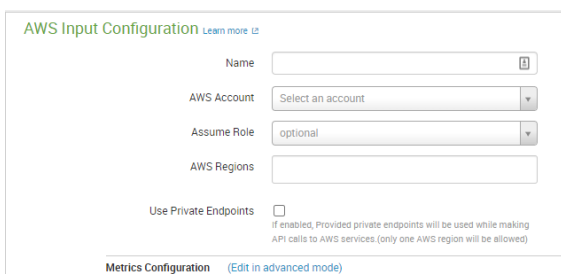
Enter a **name**, eg Connect Instance Metrics and same as before select your **AWS Account**.

Select your **AWS Regions** where your Amazon Connect instance is running.

Choose a **sourcetype** of **aws:connect:cloudwatch:metrics** and **select** the **index your created earlier for Amazon connect**

> **NOTE:** Under advanced settings you can adjust the polling period. This is the same for all the Splunk Inputs. This is how often Splunk will wait till it attempts to make another pull of the data to get the latest. This figure will vary depending on how often the data on the AWS side is also updated. Adjust as required.

Lastly, we need to configure customer metrics configuration. **Click** the **(Edit in Advanced mode)** section next to Metrics Configuration as shown below

First thing is to **remove all** the **default Namespace** sections on the left by **clicking** the **'X'** on all of them.

Click **+ Add Namespace** and enter a name of **AWS/Connect**

On the right-hand side **click + Add Another** which will bring up boxes for Dimension Value, Metrics and Metrics Statistics, **repeat this twice** more to get **three sets of empty boxes up**

In the first box under **Dimension Value?**

Enter in Box1:

`[{"Participant":[".*"],"Type of Connection":[".*"],"Instance ID":[".*"],"Stream Type":[".*"]}]`

Once you click away from the box it should **auto populate the Dimension section** to the left of it.

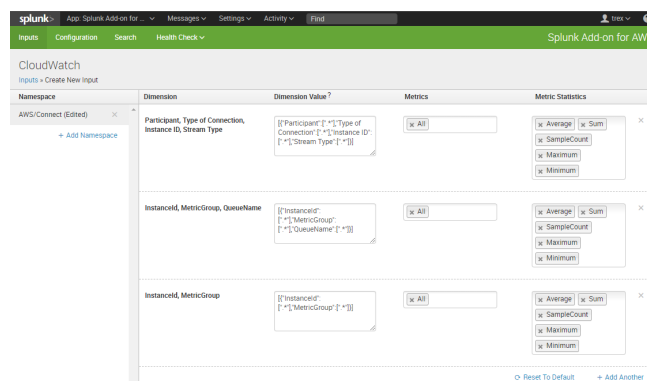**Repeat the same** for the two bits of code below in the **second and third box**.

Box 2:

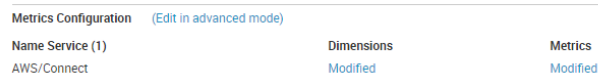`[{"InstanceId":[".*"],"MetricGroup":[".*"],"QueueName":[".*"]}]`

Box 3:

`[{"InstanceId":[".*"],"MetricGroup":[".*"]}]`

Next **click inside** the **Metrics box** and select **All** for each of the **three boxes**

In the **Metrics Statistics click** multiple times **selecting all** the options for **Average, SampleCount, Sum, Maximum and Minimum** for **all three boxes**. Final example shown below.
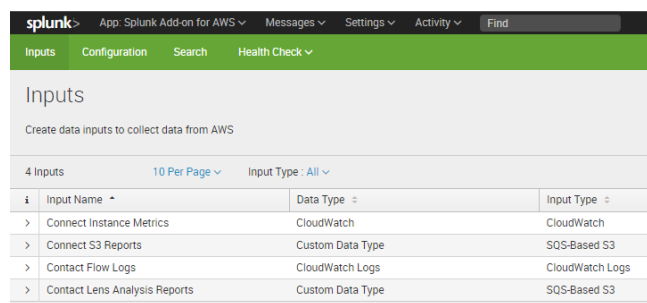

Click **Ok** to save and you should now see it represented on the input page
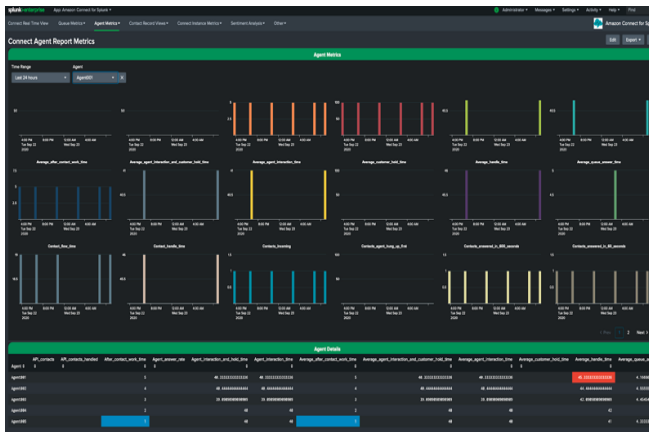

Once completed click **save** to save this input.

Once completed you should now have **four Input types** listed as shown below

CONGRATULATIONS! If you have done all of this correctly then you should now start seeing Amazon Connect data flowing into your **Amazon Connect App for Splunk** dashboards.


Troubleshooting

Below is some basic go look here statements around not seeing data being ingested into Splunk from your Amazon Connect Instance or particular dashboards which are not correctly loading.

Where to begin?

If you are not seeing a particular set of data you are expecting in a dashboard a good place to start is to check the search results manually. To do this **put your mouse over the dashboard** and **click on the Open in search** option as shown below
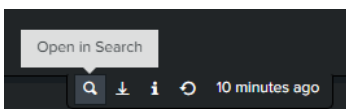
This will show you what the search is doing.

You can then start to modify, eg adjust time range, cutting the complexity back to just source type or modify some parameters.

If nothing is coming up in the results of that particular source type then it is most likely something wrong with the configuration.

More Specific Issues with…

Queue or Agent Metrics – Real Time

Requires: Amazon Data Streams/Firehose, Splunk Add-on for Firehose, HEC Token and Lambda setup

Queue or Agent Metrics – Historical

Requires: SQS, Amazon Connect Reports, Splunk Add-On for AWS, SQS for S3 Input in Add-on and IAM Account

Contact Records – Real Time

Requires: Amazon Data Streams/Firehose, Splunk Add-on for Firehose, HEC Token and Lambda setup

Connect Instance Metrics – CloudWatch Metrics

Requires: Splunk Add-On for AWS, CloudWatch Input in Add-on and IAM Account, Enabling Contact Flow Logs

Sentiment Analysis AKA Contact Lens

Requires: SQS, Splunk Add-On for AWS, SQS for S3 Input in Add-on, Configuring Contact Lens and IAM Account

Supported vs Non-Support Apps/Add-ons

Above are the apps and add-ons published by Splunk Inc. that are supported and maintained by Splunk. Splunk will provide customers with active support subscriptions an initial response and acknowledgement to any support request for these apps or add-ons in accordance with Splunk Support terms. Splunk will also ensure compatibility of Splunk-supported apps and add-ons with future releases of applicable Software. Splunk ensures this compatibility for any Splunk-supported apps or add-ons installed in Splunk Cloud Platform before commencing Splunk Cloud Platform upgrades.

Splunk does not provide support or maintenance for apps or add-ons published by any party other than Splunk Inc., including third-party developers.

More details here
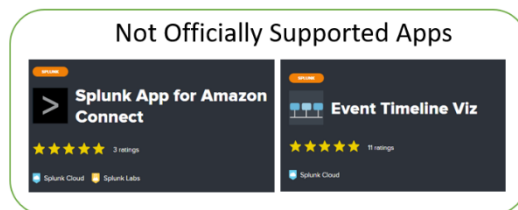
Although built by Splunk, these Apps do not carry the same support arrangement as the ones above.

An example of this is the **Splunk App for Amazon Connect.** In this case the App is designed to provide the user a collection of pre-configured dashboards and content designed to greatly reduce the initial heavy lifting in gaining visibility from your Amazon Connect Instance. Generally every user of

this application will have different use cases, SLAs and metrics required with a no one size fits all.

Final notes about this guide

We hope you got some value out of this guide. This guide was a joint project by Splunk and AWS to help our users gain more value out of the platforms faster.

If you have any feedback or comments please reach out to authors as your opinion and feedback counts.

Thanks for taking the time to go through this guide.

Meet the authors

Travis Kane

Cloud Technical Strategist at Splunk

Aurelien Plancque

Partner Solutions Architect at Amazon Web Services (AWS)