



# Reduce your Splunk spending by 90% with Upsolver and Amazon S3

[Guide](#)

# The challenge

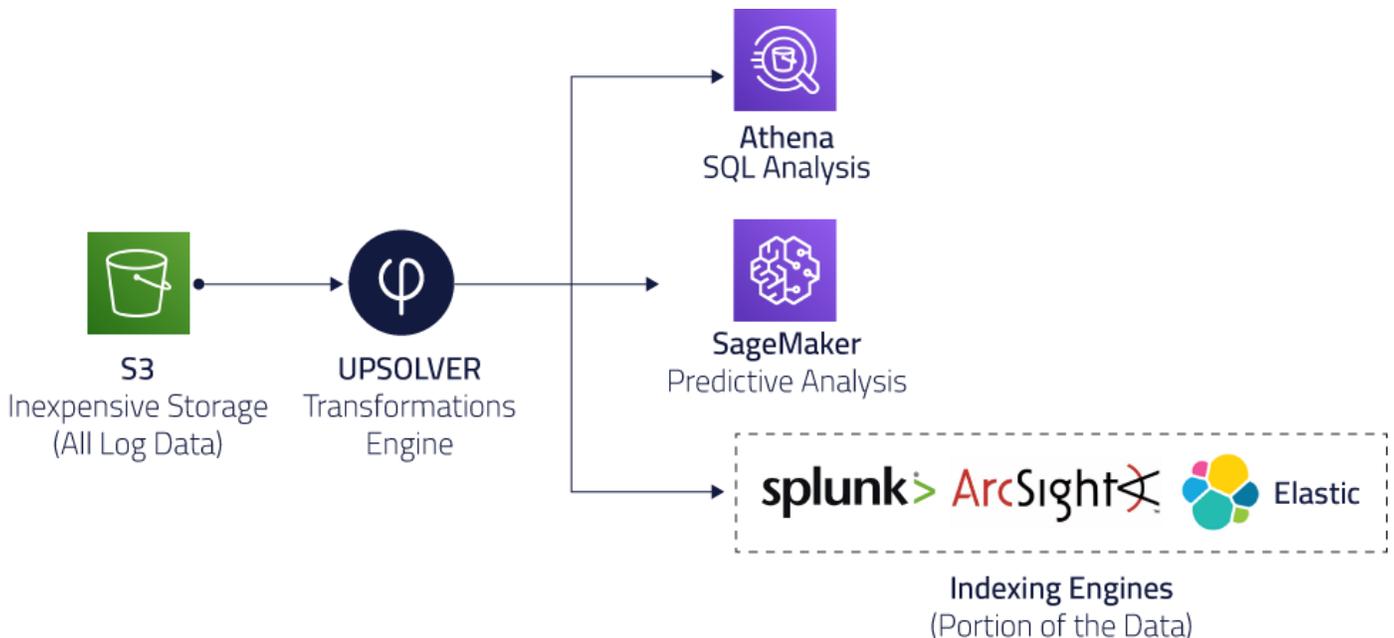
Splunk is an excellent tool for needle in the haystack searches for IT and security teams. Unfortunately, the haystack can be very expensive. Some users index everything into Splunk before realizing the vast majority of data is accessed infrequently and can therefore be stored on cheaper alternatives like AWS S3. The cost of indexing data that's unnecessary for Splunk searches can really add up.

Also, some of the end users prefer a SQL-based approach which can be challenging since Splunk's data structure is not designed for SQL processing. Many Upsolver customers experience the conundrum and we have the solution for it.



# The alternative approach

We have converted the needle in the haystack to needle in the haybucket by only indexing the most relevant data to Splunk. Our customer first filtered and pre-aggregated data with Upsolver only sending useful data to Splunk. The full set of data is routed to S3 by Upsolver for cheaper storage. By storing everything in S3, users now have many options to access the data. Since most data professionals already know SQL, we can easily utilize a SQL engine such as Athena or Redshift and build reports that run directly on data in S3. The architecture also allows flexibility for other tools such as building machine learning models with SageMaker for predictive analytics or loading S3 data to ArcSight or Elasticsearch.



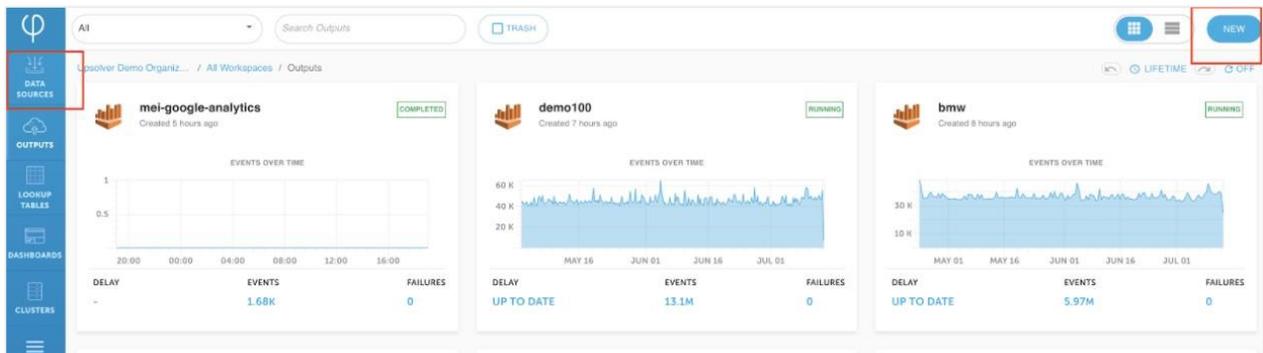
This modernized architecture has 3 main benefits:

- Dramatically reducing the cost of Splunk software.
- SQL access enables organizations to extract more value from log data.
- Uncover data for advanced analytics. Easily retrain and refit machine learning models.

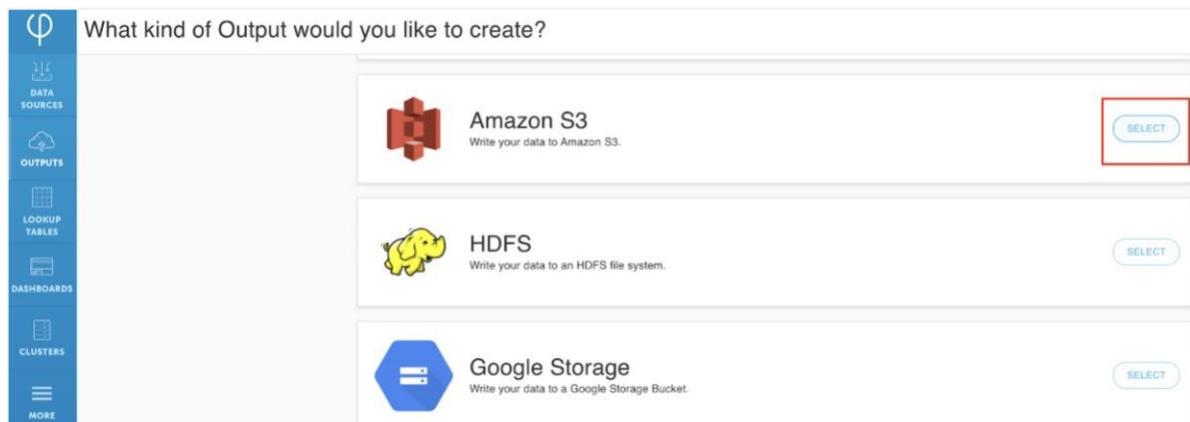
# The Technical Solution

Create an Amazon S3 data output:

1. Make sure you're already [signed up for Upsolver's](#) free trial and [created a data source](#).
2. Create a S3 data output by clicking on **OUTPUTS** on the left hand side and **NEW** on the upper right hand corner.



3. Click on **SELECT** next to Amazon S3



4. Give the data output a **NAME** and define your output format. Fill out your **DATA SOURCES** information. Click on **NEXT** to continue. (If you haven't created a Data Source, follow [this guide](#) to create one) Keep in mind that you can infer data types when you define your DATA SOURCES.) This guide uses [AWS VPC Flow Logs](#).

Create Output to Amazon S3

NAME

splunktest

Tabular  
eg. CSV

Hierarchical  
eg. JSON

DATA SOURCES

bhopp-vpc-flowlogs

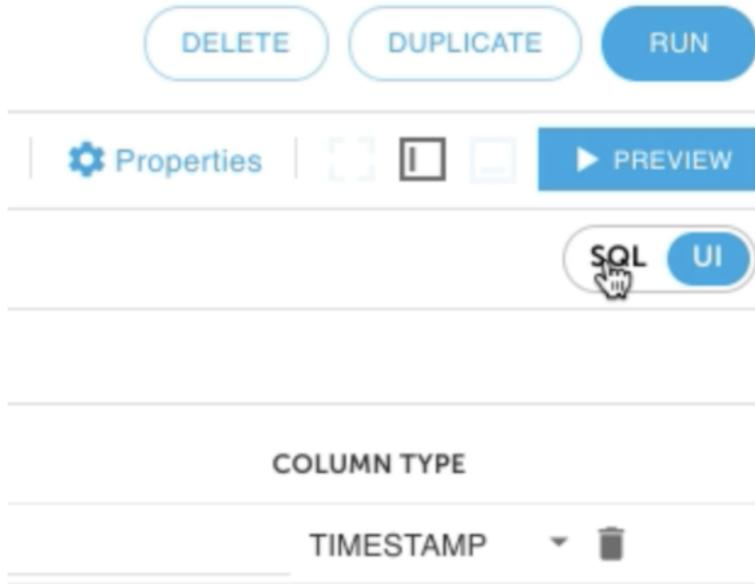
ADD

NEXT

CANCEL

Use the UI or SQL to aggregate data before sending to Splunk:

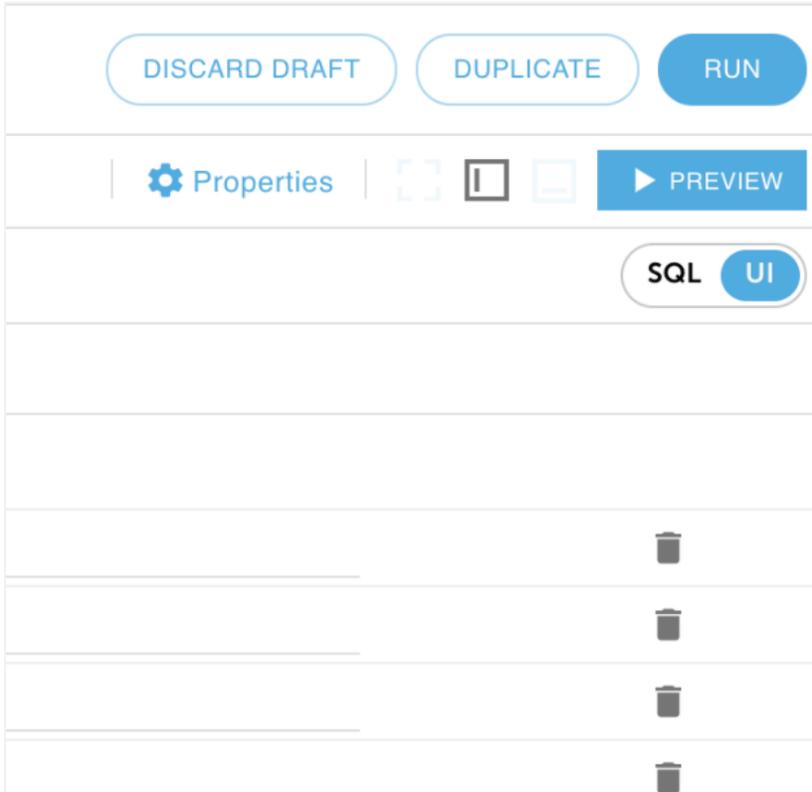
1. Select the SQL window from the upper right hand corner. Keep in mind that everything that you do in the UI will be reflected in SQL and vice versa.



2. The sample SQL aggregates multiple values together for a given period of time. Reducing the amount of data being sent to Splunk.

```
SELECT data."account-id" AS ACCOUNT_ID,  
       data.action AS action,  
       SUM(TO_NUMBER(data.bytes)) AS SUM_BYTES,  
       SUM(TO_NUMBER(data.packets)) AS SUM_PACKETS,  
       COUNT(*) AS "count"  
FROM "bhopp-vpc-flowlogs"  
GROUP BY data."account-id",  
         data.action
```

3. Click on **Properties** on the upper right hand corner.



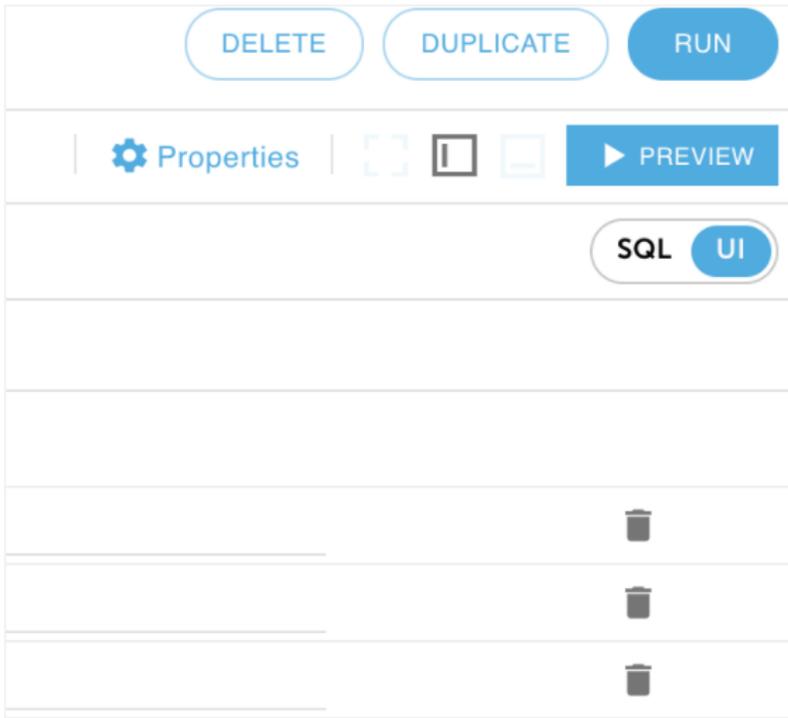
4. Change the **Output Interval** to 10 minutes and click on **UPDATE**.



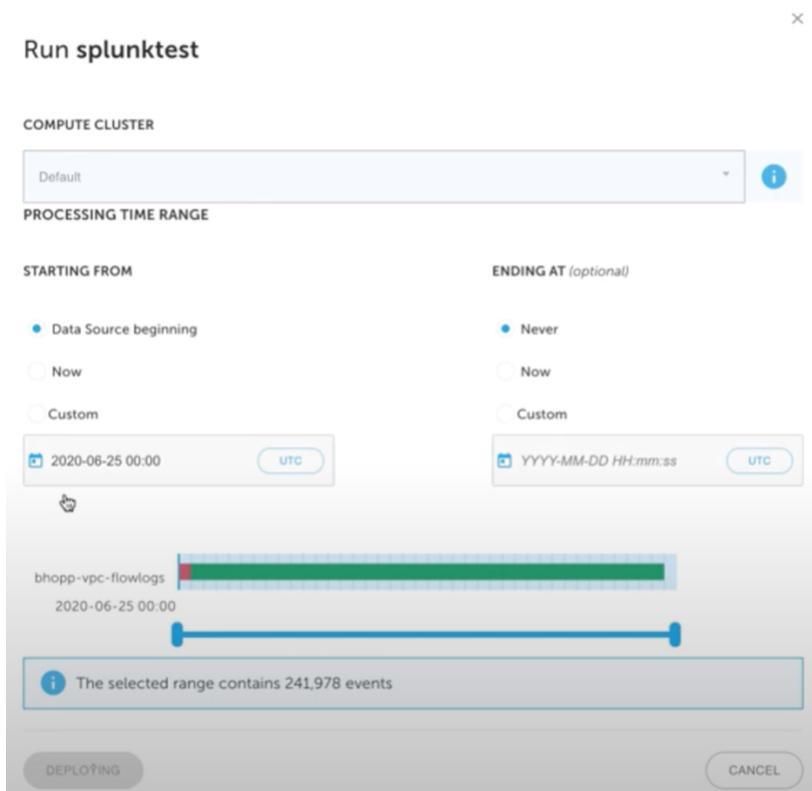
5. Select **OUTPUT FORMAT** the **S3 CONNECTION** that you want the data to be stored in and click on **NEXT**.

The image shows a 'Run Parameters' dialog box with a close button (X) in the top right corner. It contains two dropdown menus: 'OUTPUT FORMAT' with 'JSON' selected, and 'S3 CONNECTION' with 'meiupsolversplunk' selected. At the bottom, there are two buttons: 'NEXT' on the left and 'BACK' on the right. A mouse cursor is visible over the 'NEXT' button.

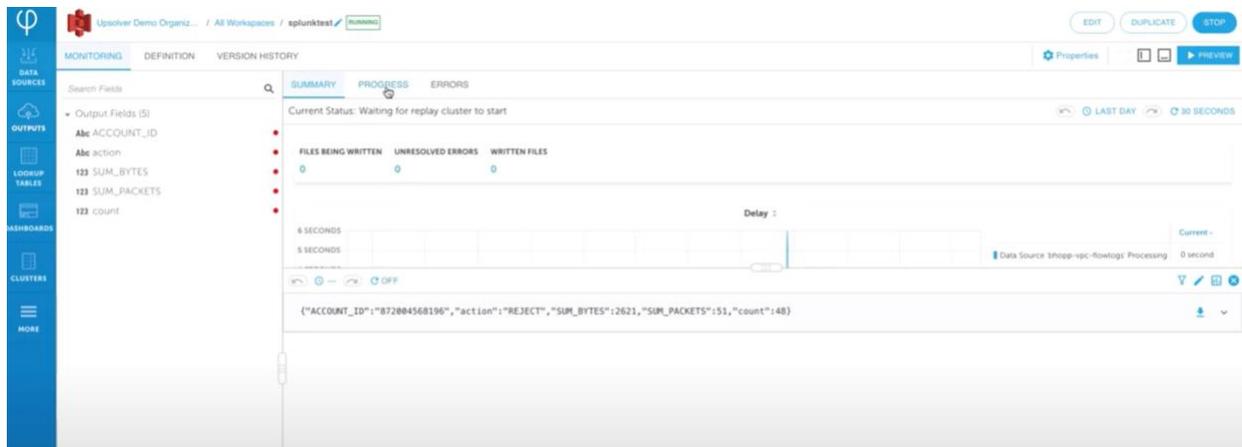
6. Click on **RUN** on the upper right corner.



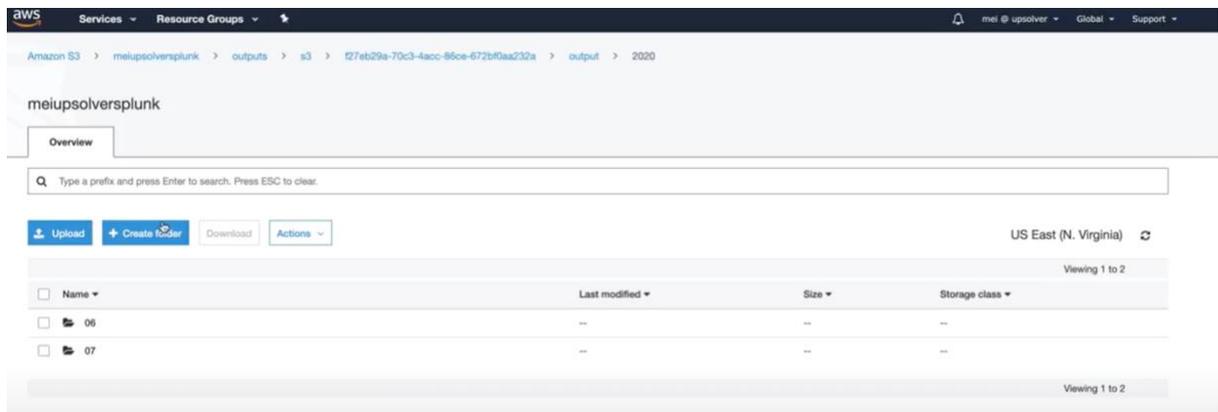
7. Choose **COMPUTE CLUSTER** and the time range of the data that you want to process. Click on **DEPLOY**. For streaming data, leave the **ENDING AT** as **Never**.



8. Your data will start loading to the previously defined S3 bucket.

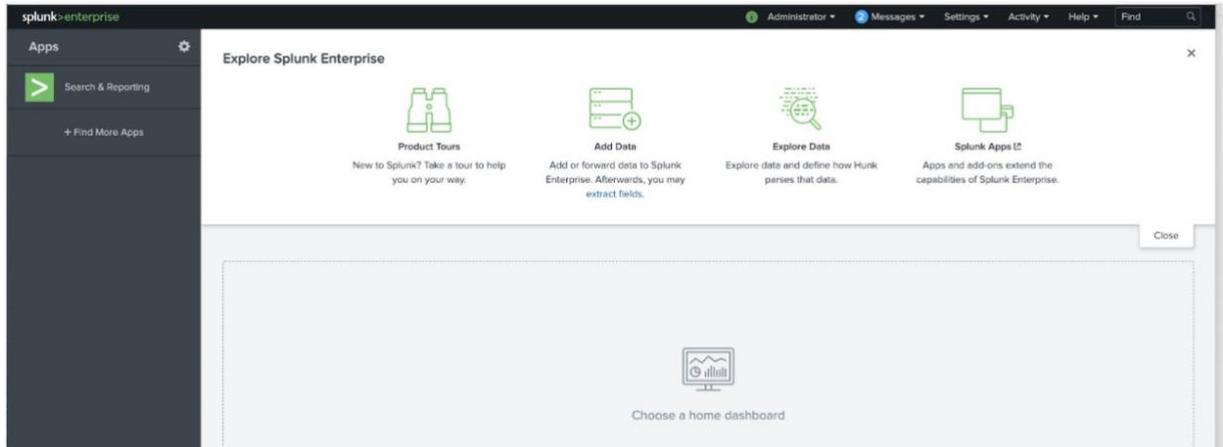


9. Check your S3 bucket to make sure everything is as expected.

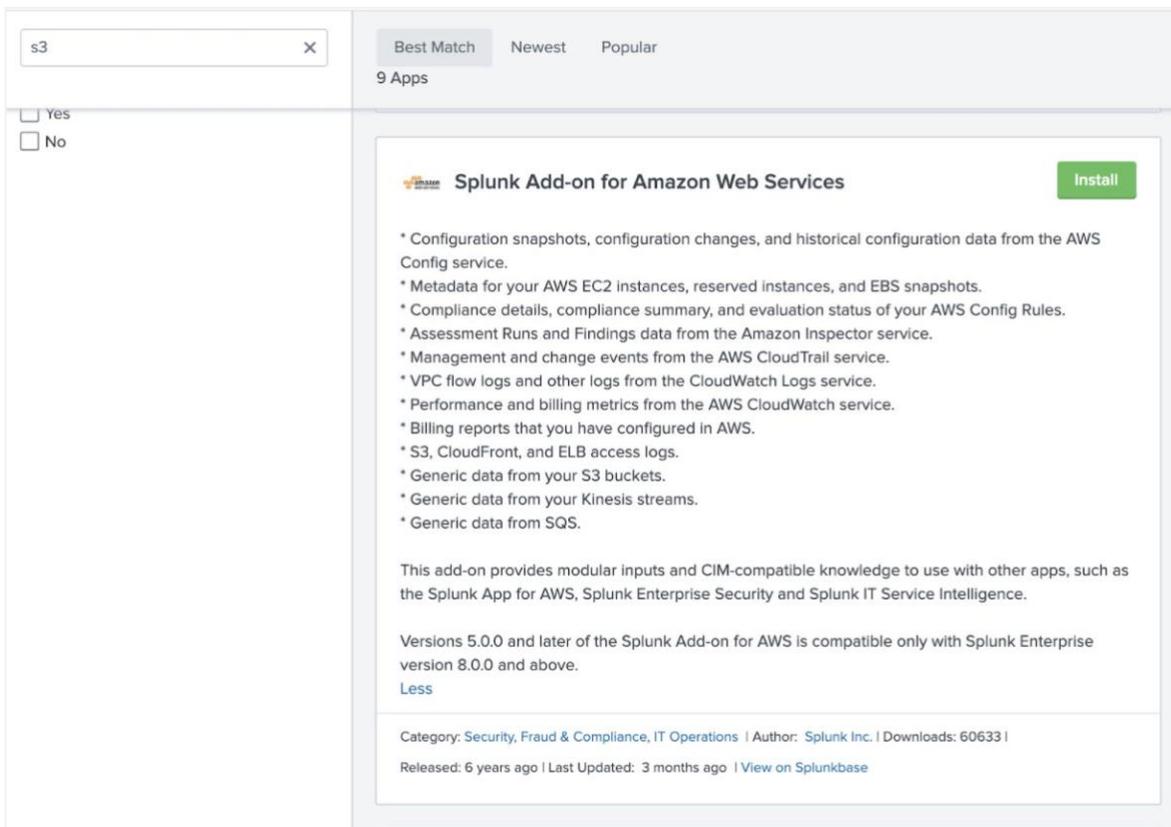


# Configure your Splunk environment:

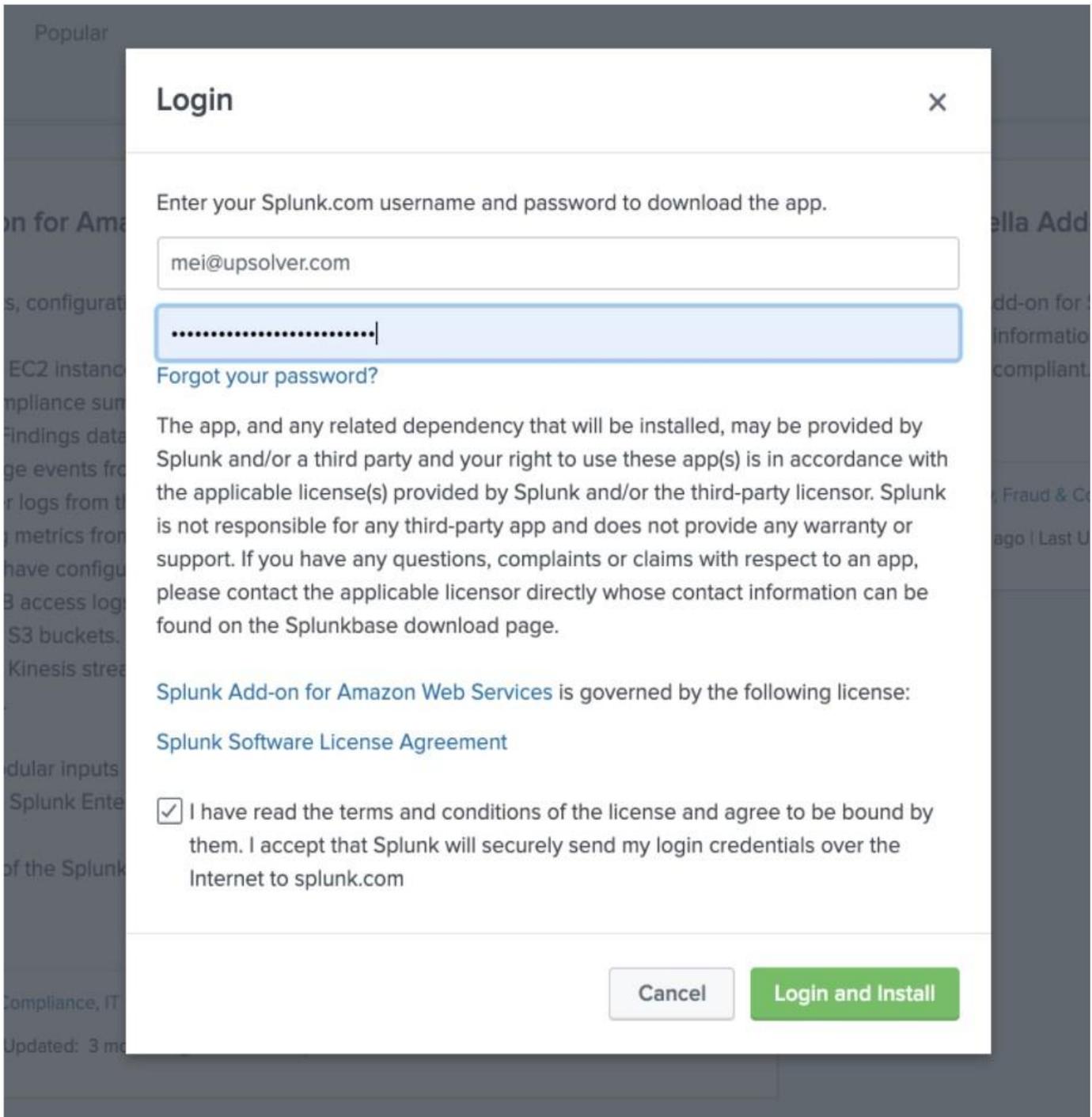
1. Login to your Splunk Enterprise environment and click on **Splunk Apps**.



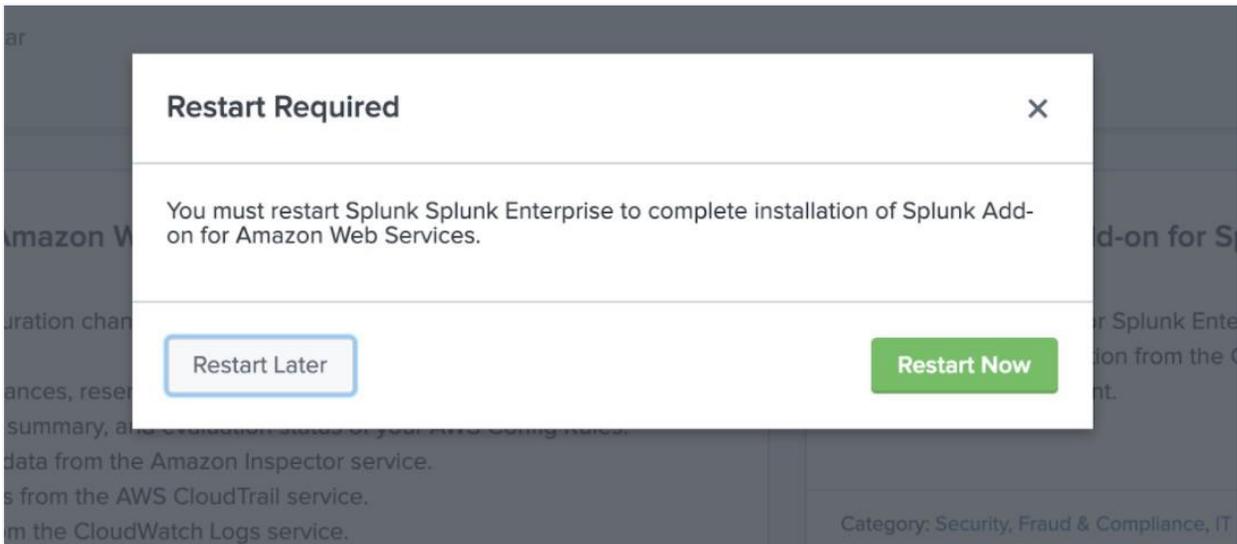
2. Install Splunk **Add-on for Amazon Web Services**.



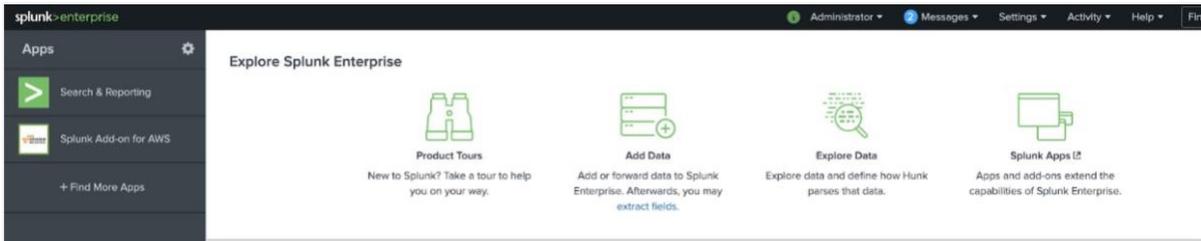
3. Enter your Splunk.com username and password and check the terms and conditions checkbox. Click on **Login and Install**.



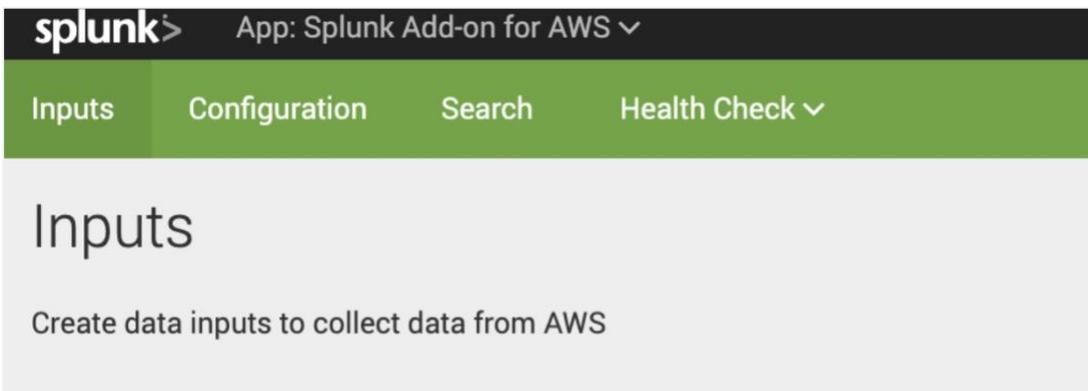
4. Restart Splunk by clicking on **Restart Now**.



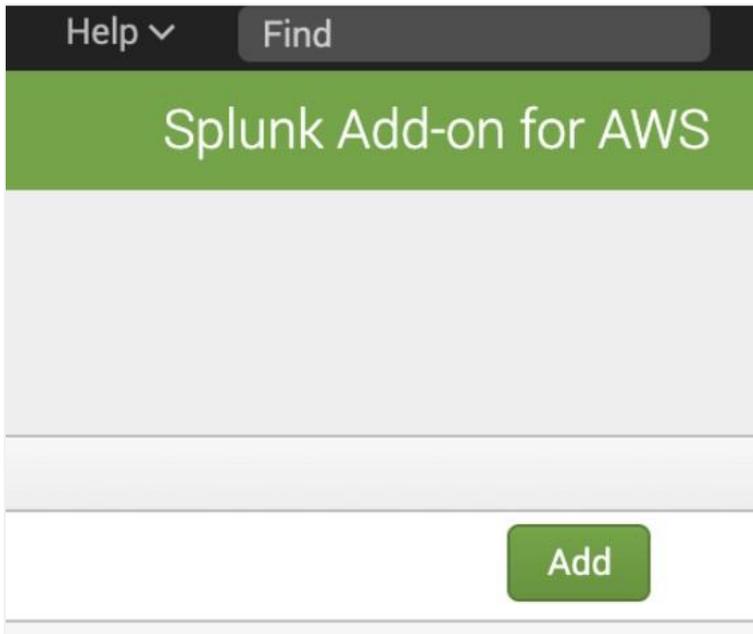
5. After relogging in, select **Splunk Add on for AWS** that you have just installed.



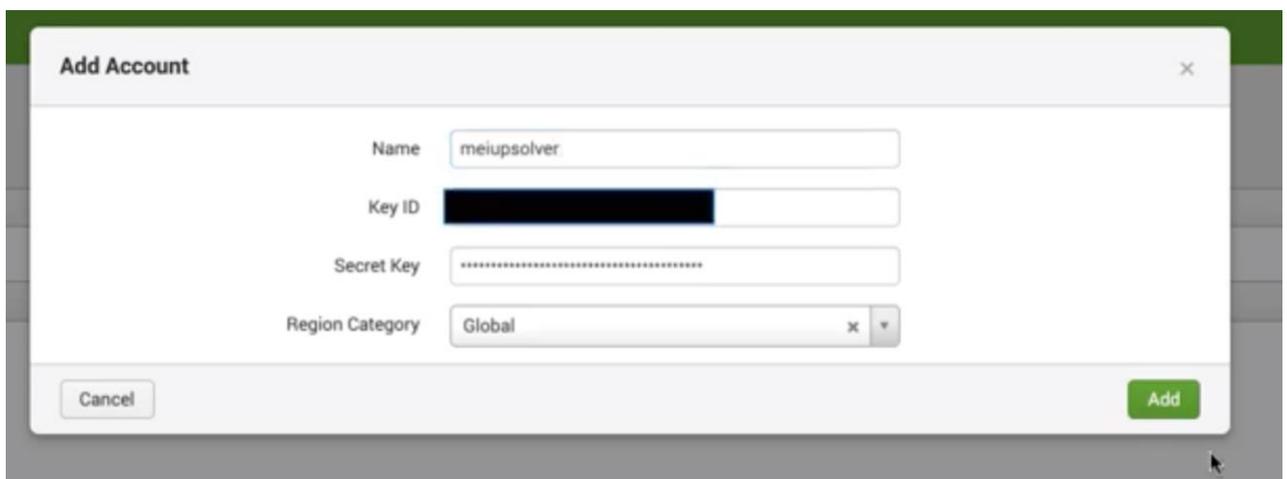
6. Click on Configuration at the top.



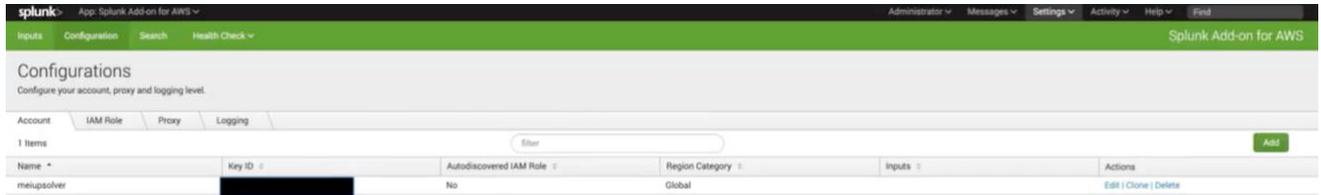
7. Click on **Add** on the right hand of your screen.



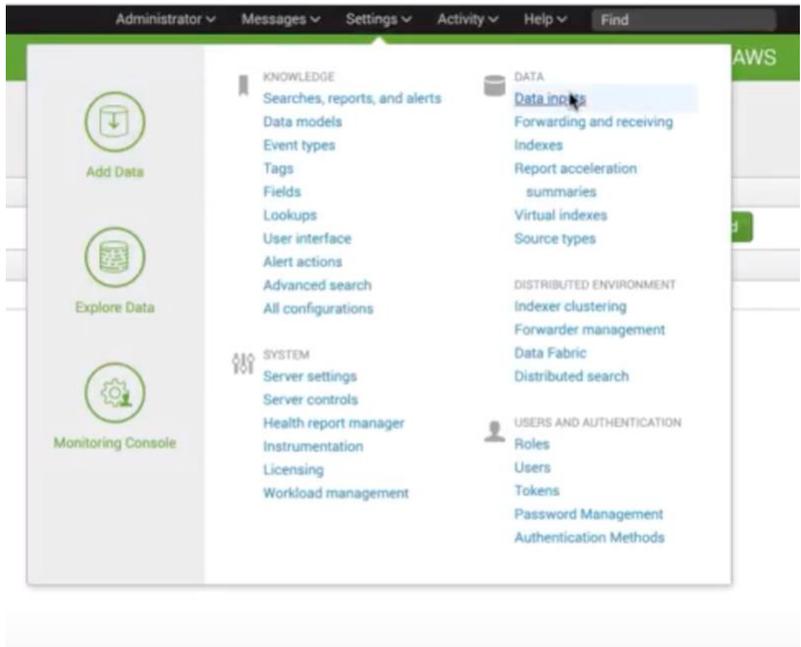
8. Give your account a name. Remember this name because we will use it for Data Inputs later. Enter your AWS credential. And click on **Add**.



9. You should see your account show up under Configurations.



10. Click on **Settings** and **Data Inputs** on the upper right corner.



11. Click on the **AWS S3** input. Most likely It's on the second page.

Type	Inputs	Actions
<b>Files &amp; Directories</b> Index a local file or monitor an entire directory.	9	+ Add new
<b>HTTP Event Collector</b> Receive data over HTTP or HTTPS.	0	+ Add new
<b>TCP</b> Listen on a TCP port for incoming data, e.g. syslog.	0	+ Add new
<b>UDP</b> Listen on a UDP port for incoming data, e.g. syslog.	0	+ Add new
<b>Scripts</b> Run custom scripts to collect or generate more data.	5	+ Add new
<b>AWS S3</b> Collect and index log files stored in AWS S3.	0	+ Add new
<b>AWS SQS-Based S3</b>	0	+ Add new
<b>AWS S3 Incremental Logs</b>	0	+ Add new
<b>Splunk_TA_aws</b> Collect and index AWS SQS messages	0	+ Add new

12. Give the **Data input** a Name. Also fill out your **AWS Account** information. It's the same **Account Name** from step 8 in the previous section. Give it a **Bucket Name**. It has to match the bucket name on your AWS account the output data is being stored. Change the Polling interval to 10. Define **Key prefix** as your S3 folder path.

Files & Directories  
Upload a file, index a local file, or monitor an entire directory.

HTTP Event Collector  
Configure tokens that clients can use to send data over HTTP or HTTPS.

TCP / UDP  
Configure the Splunk platform to listen on a network port.

Scripts  
Get data from any API, service, or database with a script.

AWS Billing  
Collect and index billing report of AWS in CSV format located in AWS S3 bucket.

AWS Billing (Cost And Usage Report)

AWS CloudTrail  
Collect and index log files produced by AWS CloudTrail. CloudTrail logging must be enabled and published to SNS topics and an SQS queue.

AWS CloudWatch Metrics

AWS CloudWatch Logs  
Collect and index events in AWS CloudWatch Logs.

AWS Config  
Collect notifications produced by AWS Config. The feature must be enabled and its SNS topic must be subscribed to an SQS queue.

AWS Config Rules  
Collect and index Config Rules for AWS services.

Collect and index log files stored in AWS S3.

Name \* Unique data input name  
meiupsolversplunk

Secure S3 connection True

S3 host name For example: s3-ap-south-east-1.amazonaws.com  
s3.amazonaws.com

AWS Account \* meiupsolver

Bucket Name \* meiupsolversplunk

Polling interval 10

Key prefix outputs/s3/127eb29a-70c3-4acc-86ce-672bf0aa232a/output

For folder keys -1

Start datetime Only S3 keys which have been modified after this datetime will be considered  
default

End datetime Only S3 keys which have been modified before this datetime will be considered

Max trackable items 100000

Max number of retry attempts to stream incomplete items 3

13. Scroll down and check **More settings**. This will provide you with additional options for settings. **Change Set sourcetype to From list**. From the **Select sourcetype** from list dropdown, select **json\_no\_timestamp**. Click on **Next** on the top.

Select Source  Done  < Back Next >

index for the excluded CloudTrail events

Assume Role

More settings

**Interval**

Interval   
Number of seconds to wait before running the command again, or a valid cron schedule. (leave empty to run this script once)

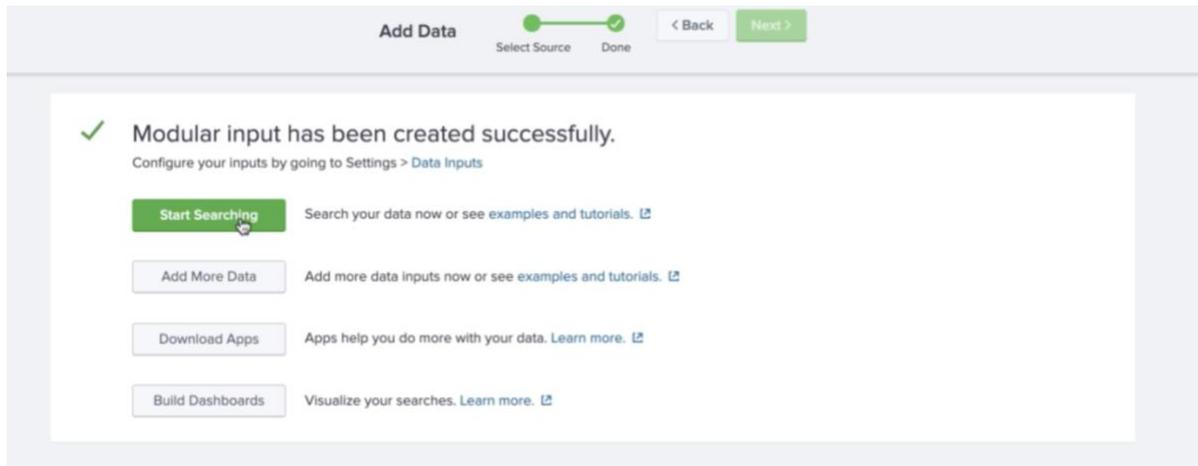
**Source type**  
Set sourcetype field for all events from this source.

Set sourcetype

Select source type from list \*

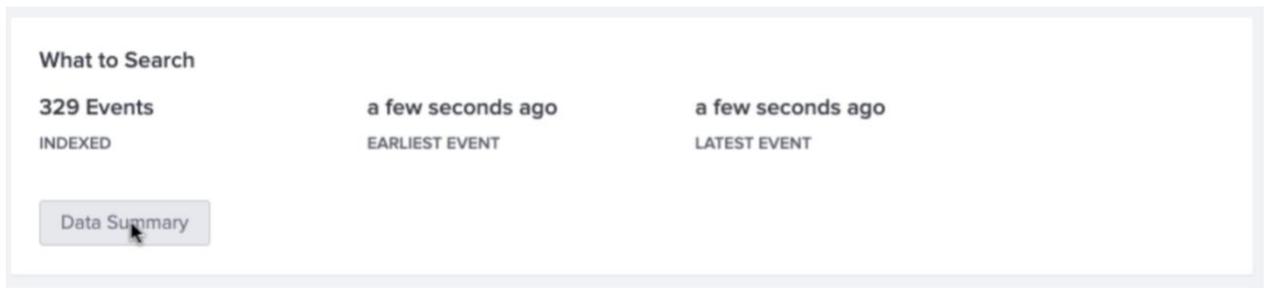
Splunk classifies all common data types automatically, but if you're looking for something specific, you can find more source types in the [SplunkApps apps browser](#) or online at [apps.splunk.com](#).

14. Click on Start searching.



## Verify data in Splunk:

1. Click on **Data Summary** under **What to Search**.



2. Click on **Sourcetype** and **json\_no\_timestamp**.

## Data Summary

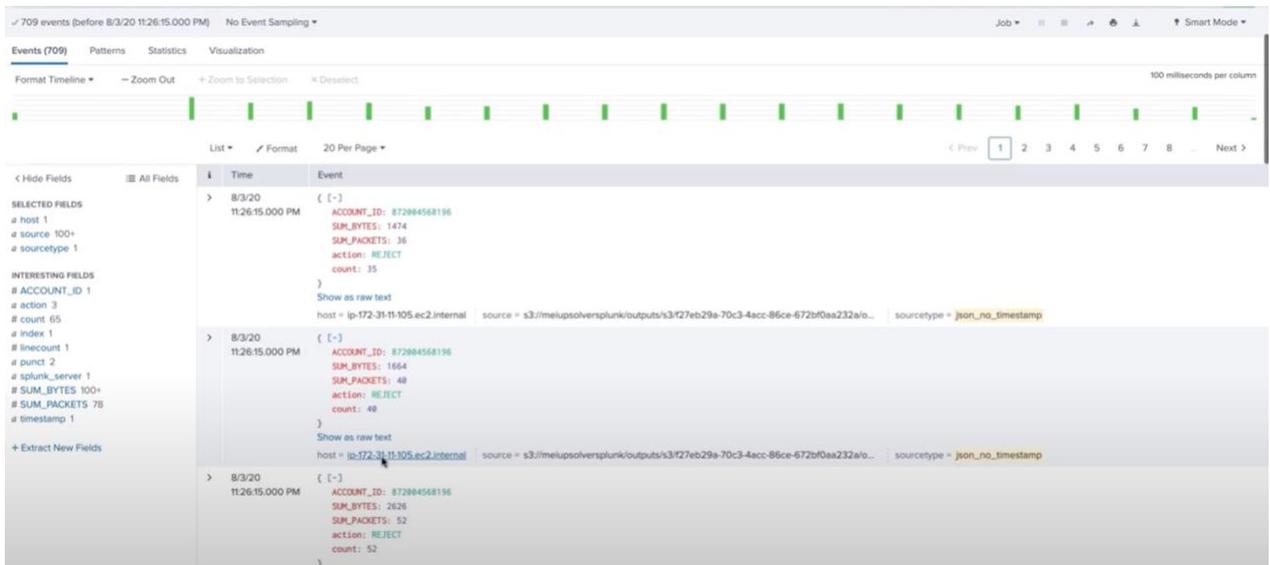


Hosts (1) Sources (556) **Sourcetypes (1)**

filter

Sourcetype		Count	Last Update
<a href="#">json_no_timestamp</a>		92	8/3/20 11:26:11.000 PM

3. Verify your indexed data is the same as the aggregated data from Upsolver.  
Success!



### Success metrics

- Reduced cost by 90%
- 4X increase in scale in 3 months

### Example from a customer

Our customer in the automotive industry was able to save millions from their new Upsolver and S3 to Splunk architecture. They are now focused on building more productive analytics on Athena. We're honored to be able to help them through this journey and we look forward to helping you to reduce your Splunk cost by 90%.