|  |  |
| --- | --- |
| PINPOINT/SES Event Dashboard | AWS DUE Specialist Solutions Architect |

# Pinpoint – Engage Event dashboard

Contents

# Disclaimer

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.

© 2021 Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Solution Summary

## Background

## Currently Amazon Pinpoint and Amazon SES do have a dashboard which provides user engagement data at an aggregated level but not at the individual event or recipient level. This leads to a lot of problem for production support teams to debug individual engage events also Marketers/business users could not gather detail engagement analytics with each and every user. This solution will elaborate on how to create a detail event engagement dashboard for Pinpoint/SES using Amazon Quicksight.

## Solution

This Solution extends [DUE events database](https://aws.amazon.com/solutions/implementations/digital-user-engagement-events-database/) solution allowing all the events queried through Amazon Athena which will be integrated to Amazon Quicksight to display engagement events. This solution can be implemented into three steps in chronological order.

1. Set up event database solution
2. Deploy the Athena View creation CloudFormation template
3. Follow the blog to setup Amazon Quicksight to dashboard engagement event analytics and event dashboard

## Use case(s)

User segmentation based on:

* Deep dive into the event insights. (eg : SMS events, Email events, Campaign events, Journey events)
* Look at the dashboard at individual user level engagement events and metrics.

## Considerations

1. DUE event data base solution mush be setup 1st.
2. Customer should be ready to procure Amazon Quicksight.

# Solution Architecture & Business Logic

## Solution Architecture

## 

# Steps to implement the solution

## Step 1 – Create AWS account & Pinpoint Project

[Create an AWS account](https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/)

[Implement](https://aws.amazon.com/solutions/implementations/digital-user-engagement-events-database/) Event database solution

* Copy and save the 1/DUE database name 2/S3 Bucket name from DUE event database solution, this will be needed as input parameters in step-2

## Step 2 – Create S3 bucket for Lambda code and upload the Zip files

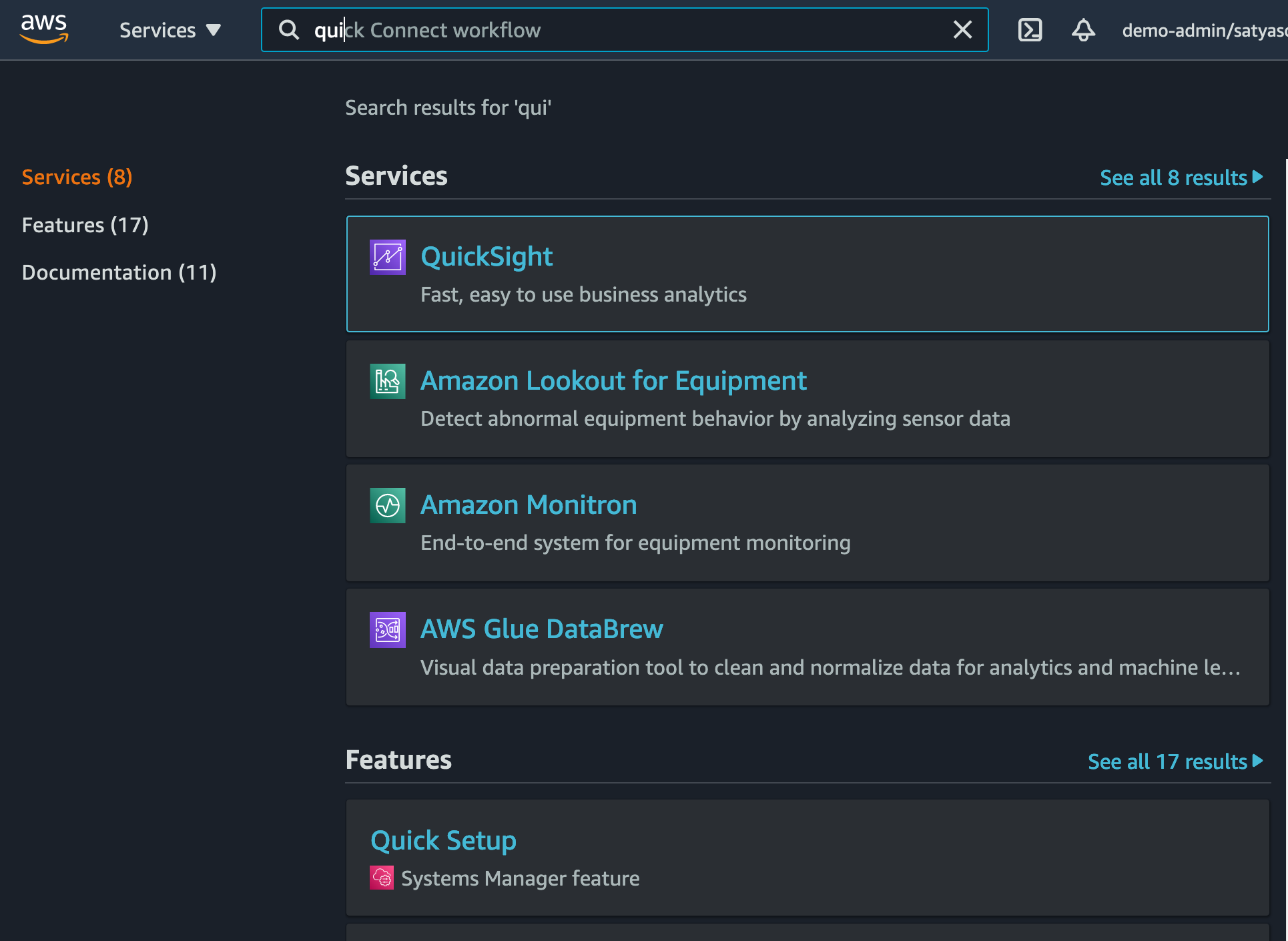
* This step creates several new amazon Athena views that’s to be act as a data source for Amazon Quicksight. Dashboard
* [Create an S3 bucket](https://docs.aws.amazon.com/AmazonS3/latest/userguide/create-bucket-overview.html) in the region that you have your Pinpoint projects and provide it a unique name

Upload the zip file into the root folder: lambda\_view\_creator.zip. Install the cloud formation stack template within the same region as that of your DUE database is. Provide following details as input parameter to cloud formation stack and proceed with the installation.

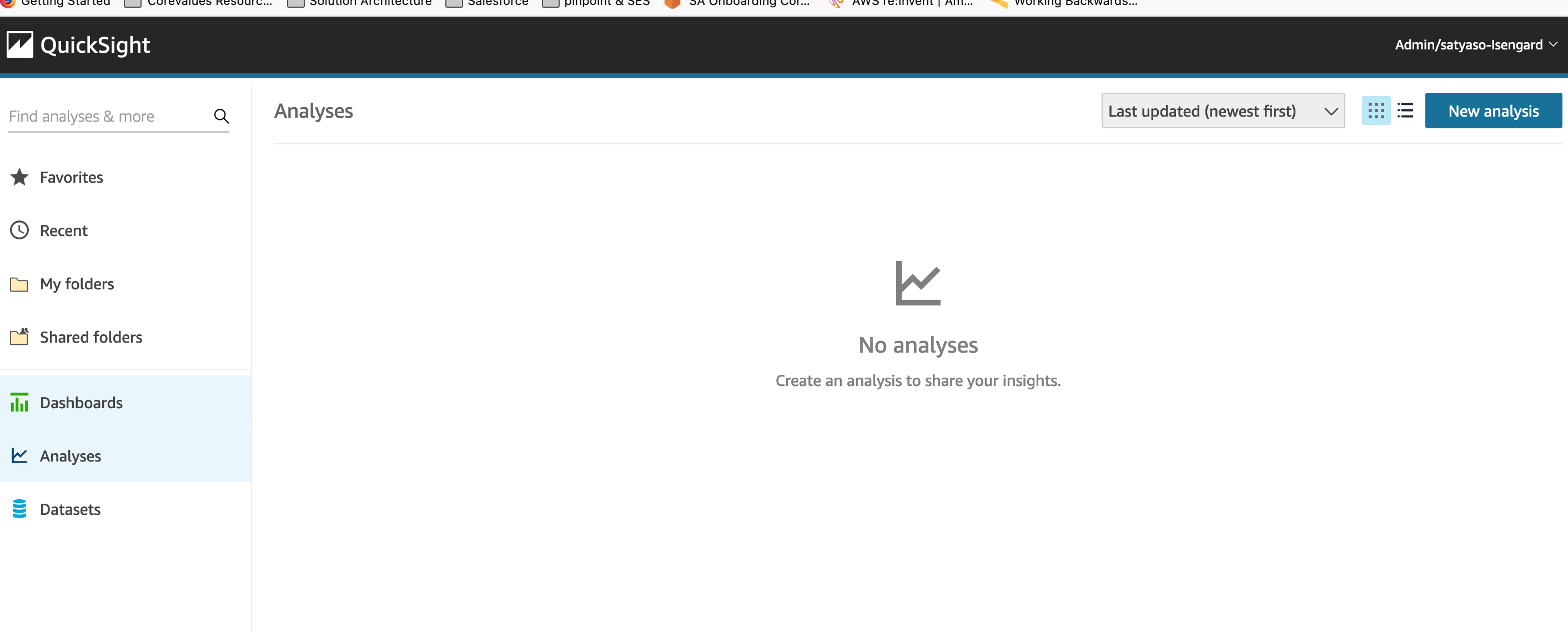
* + EventAthenaDatabaseName - As mentioned in Step1
  + LambdaCodes3bucket – S3 bucket name where you have installed the code zip file.
  + S3DataLogBucket- As mentioned in Step1
* This solution will create additional 5 Athena views which are
  + All\_email\_events
  + All\_SMS\_events
  + All\_custom\_events (Custom events can be Mobile app/WebApp/Push Events)
  + All\_campaign\_events
  + All\_journey\_events

## Step 3 – Create Amazon Quicksight engagement Dashboard

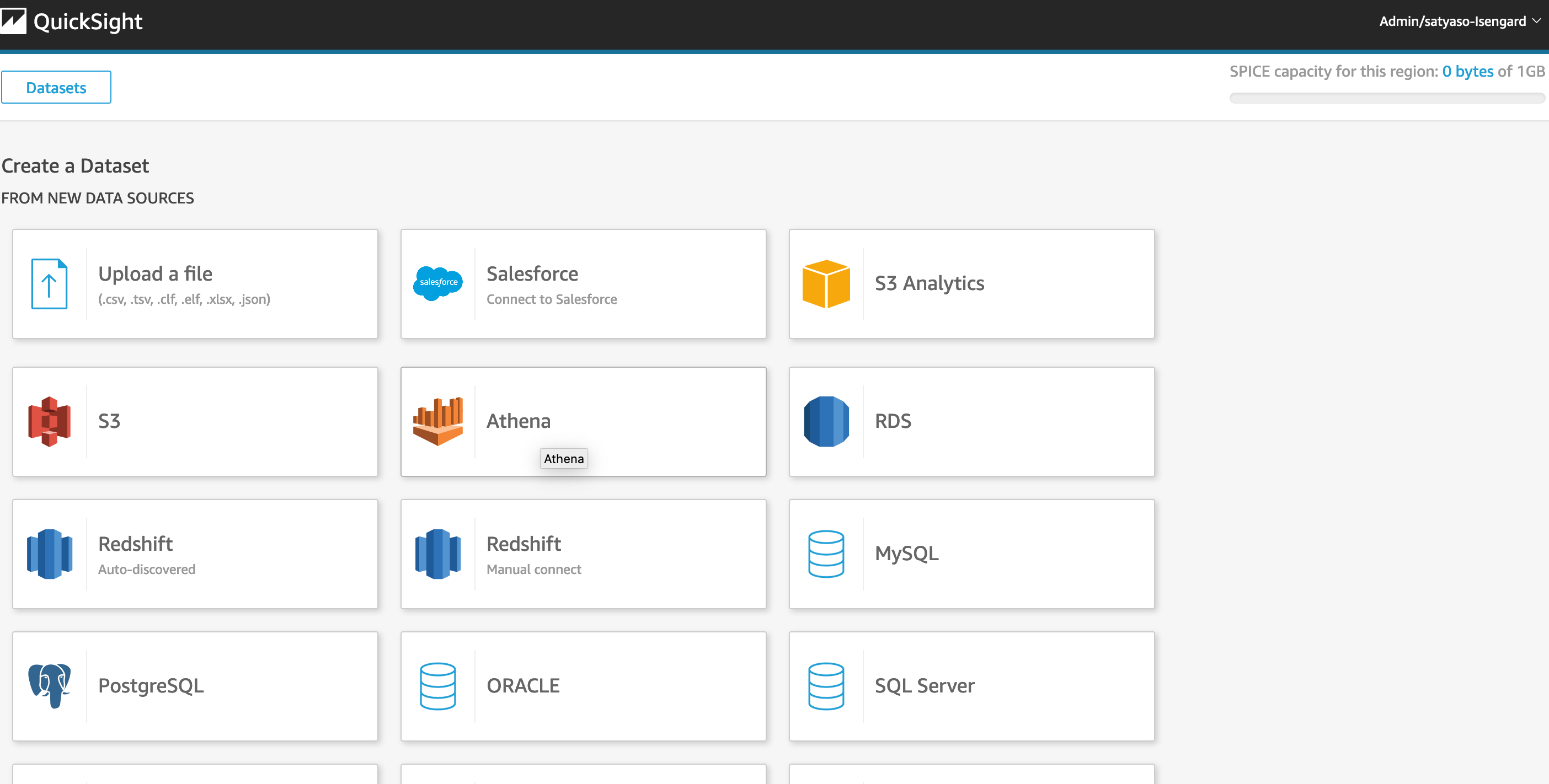
* To Setup Amazon Quicksight please look at the Appendix section 1 (this step is not needed in case you have already setup amazon quick sight)
* Go/search Amazon Quicksight on AWS console.



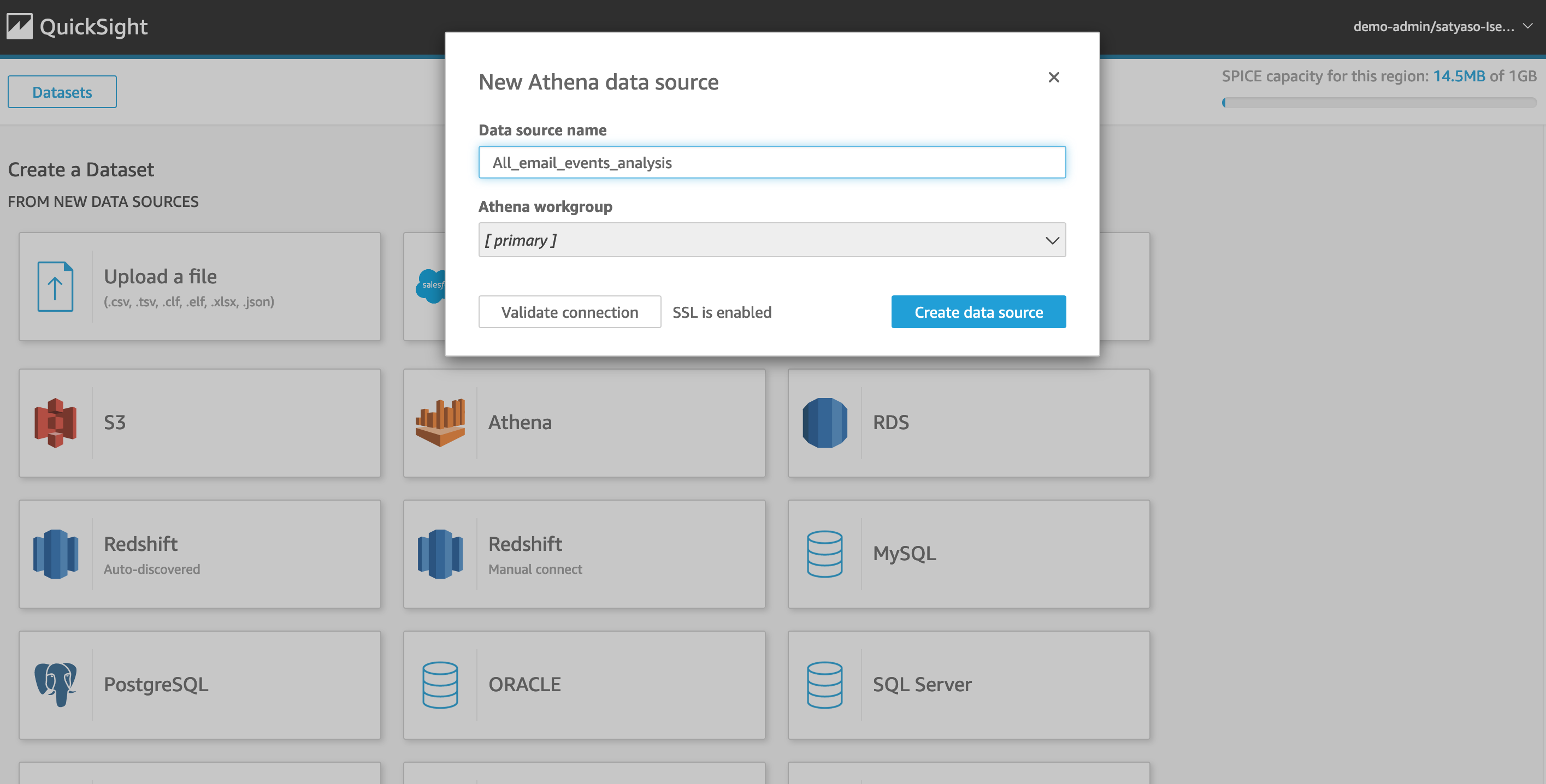
* Create New Analysis and select “New dataset”



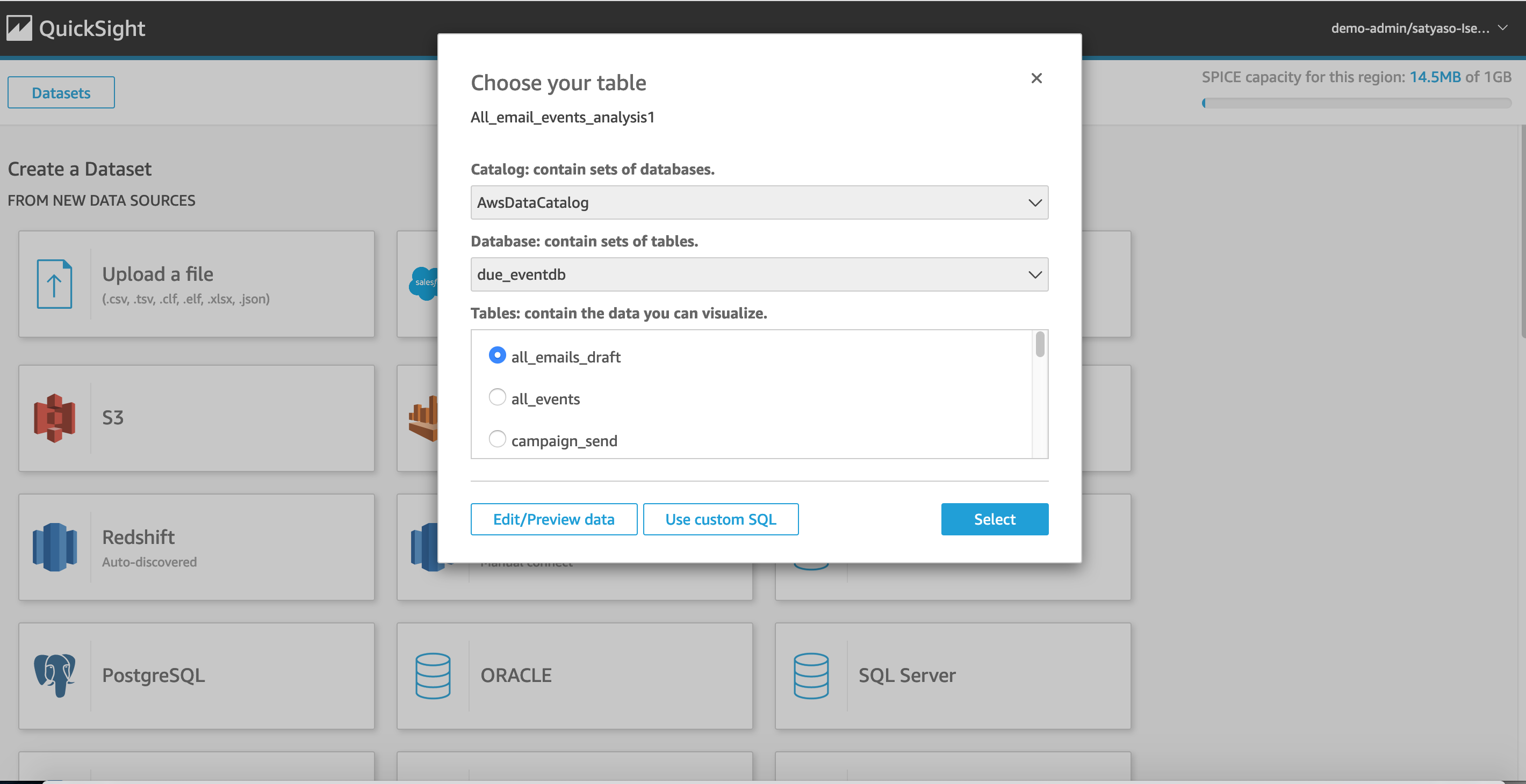
* Select Athena as data source



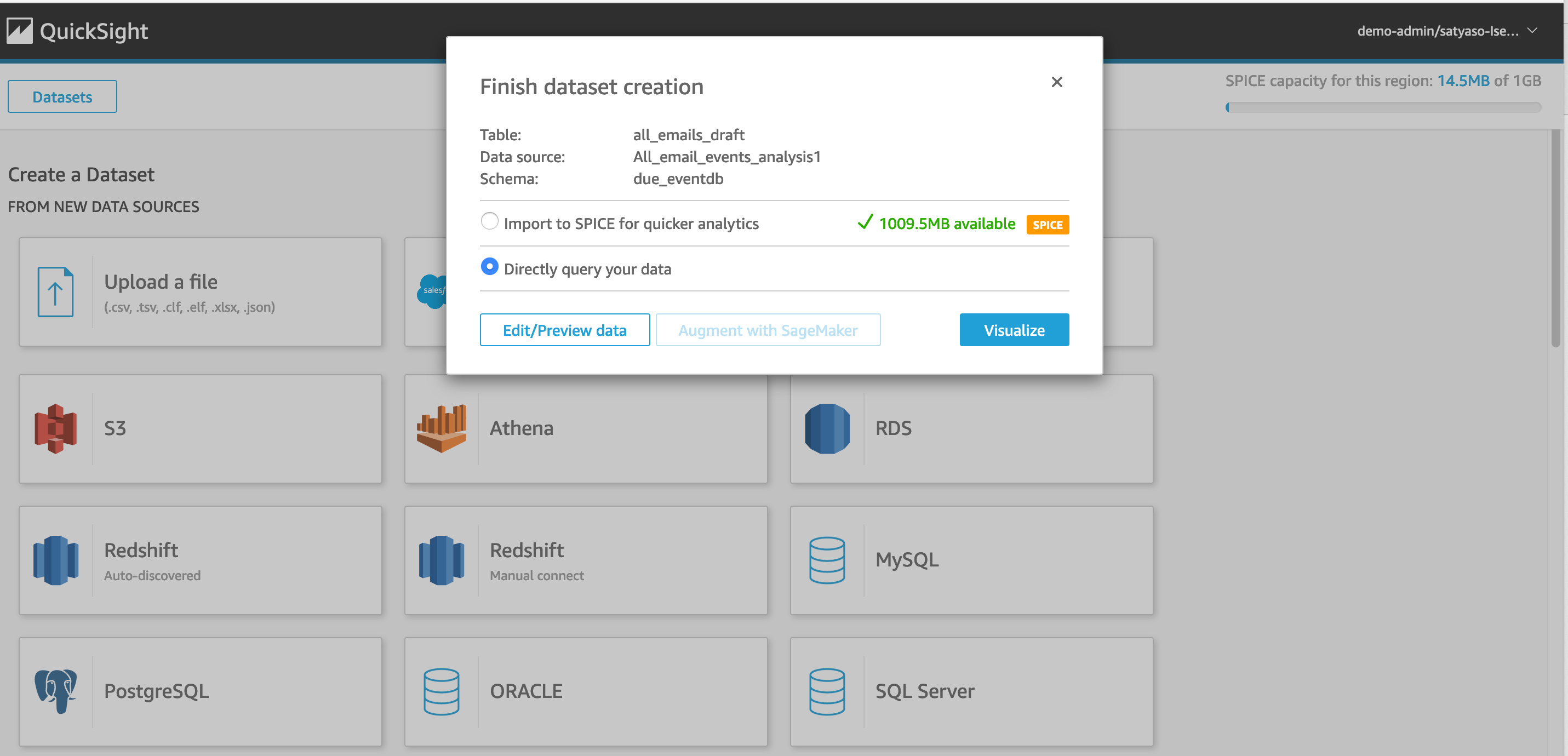
* As a next step, you need to select what all analysis you need for respective events.
* **Email Events** - For all the emails events, name the analysis as “All-emails-event-analysis” (this can be any nomenclature), select Athena workgroup as primary then create data source.



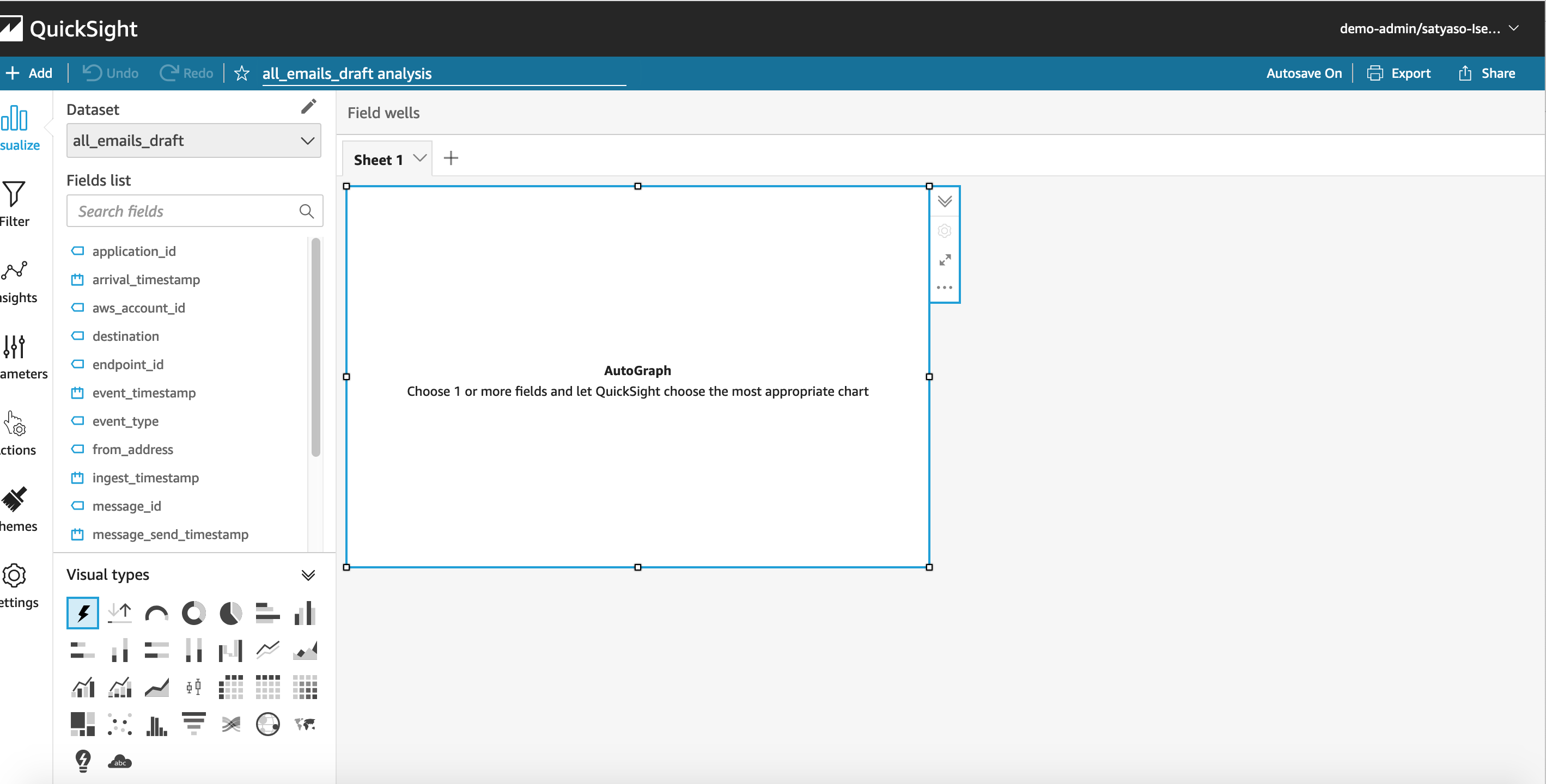
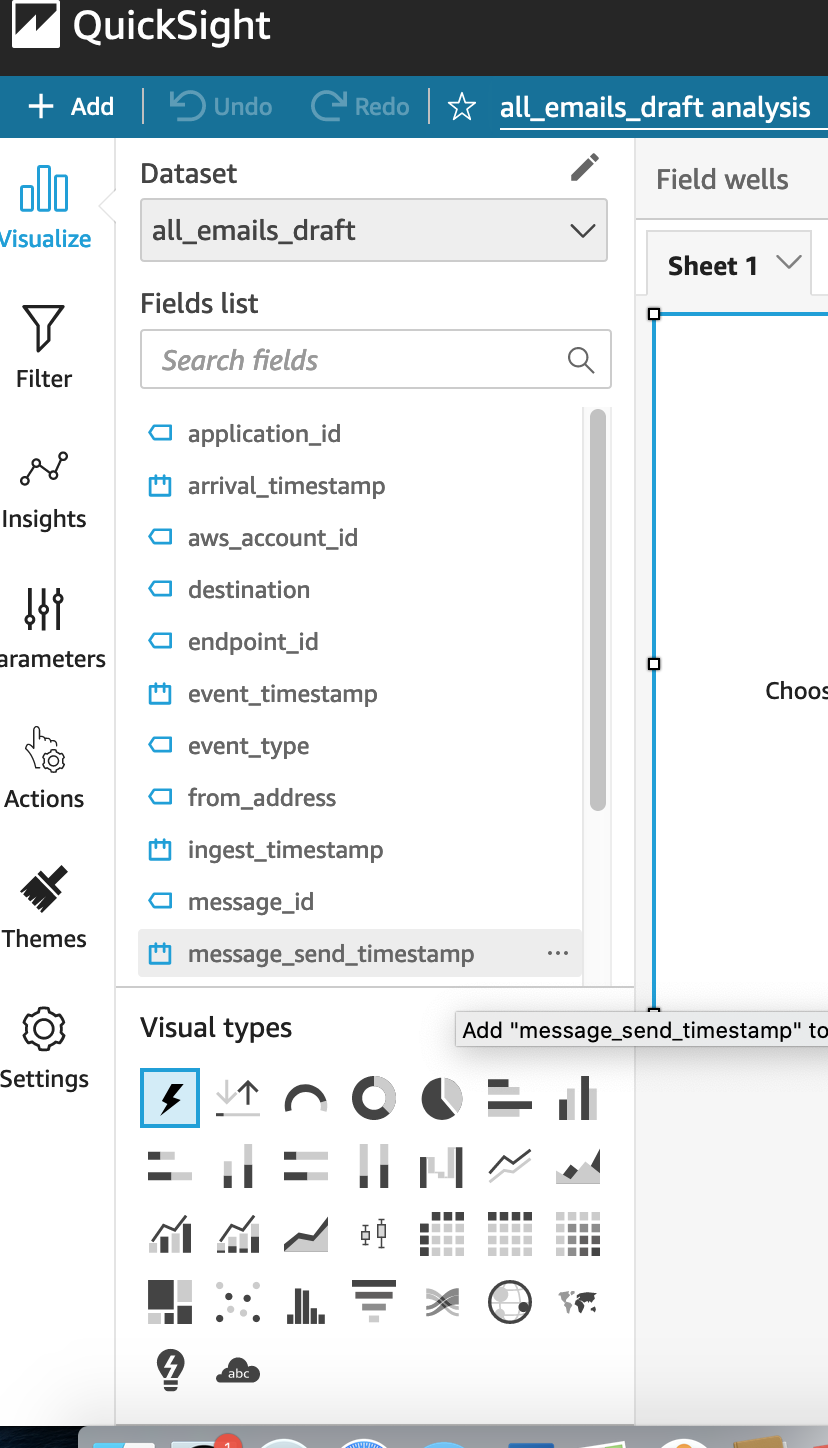
* Once you create the data source Quicksight lists all the views and tables available under the specified database(in our case it is :- due\_eventdb). Select the **all\_email\_events** view as data source.



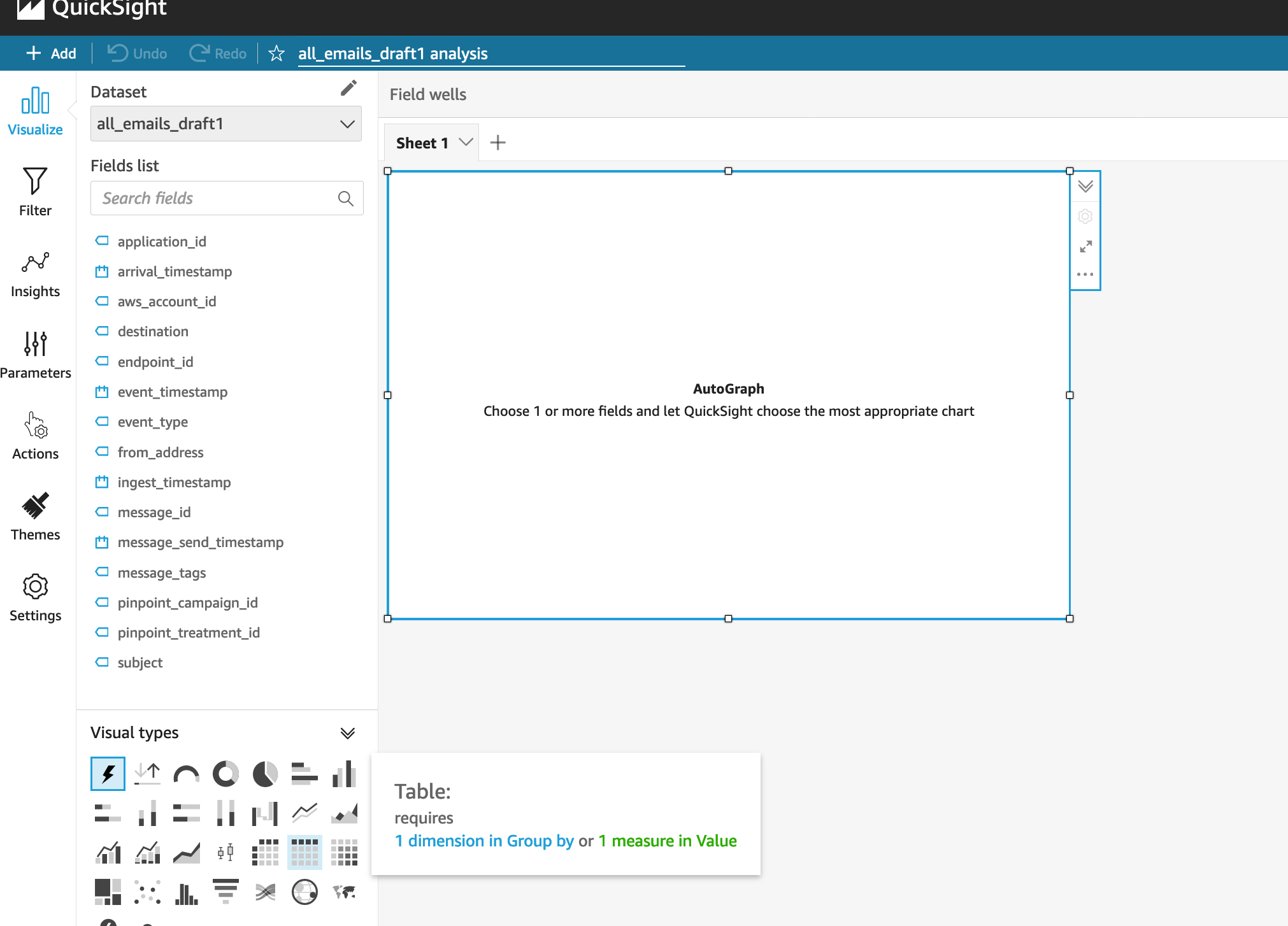
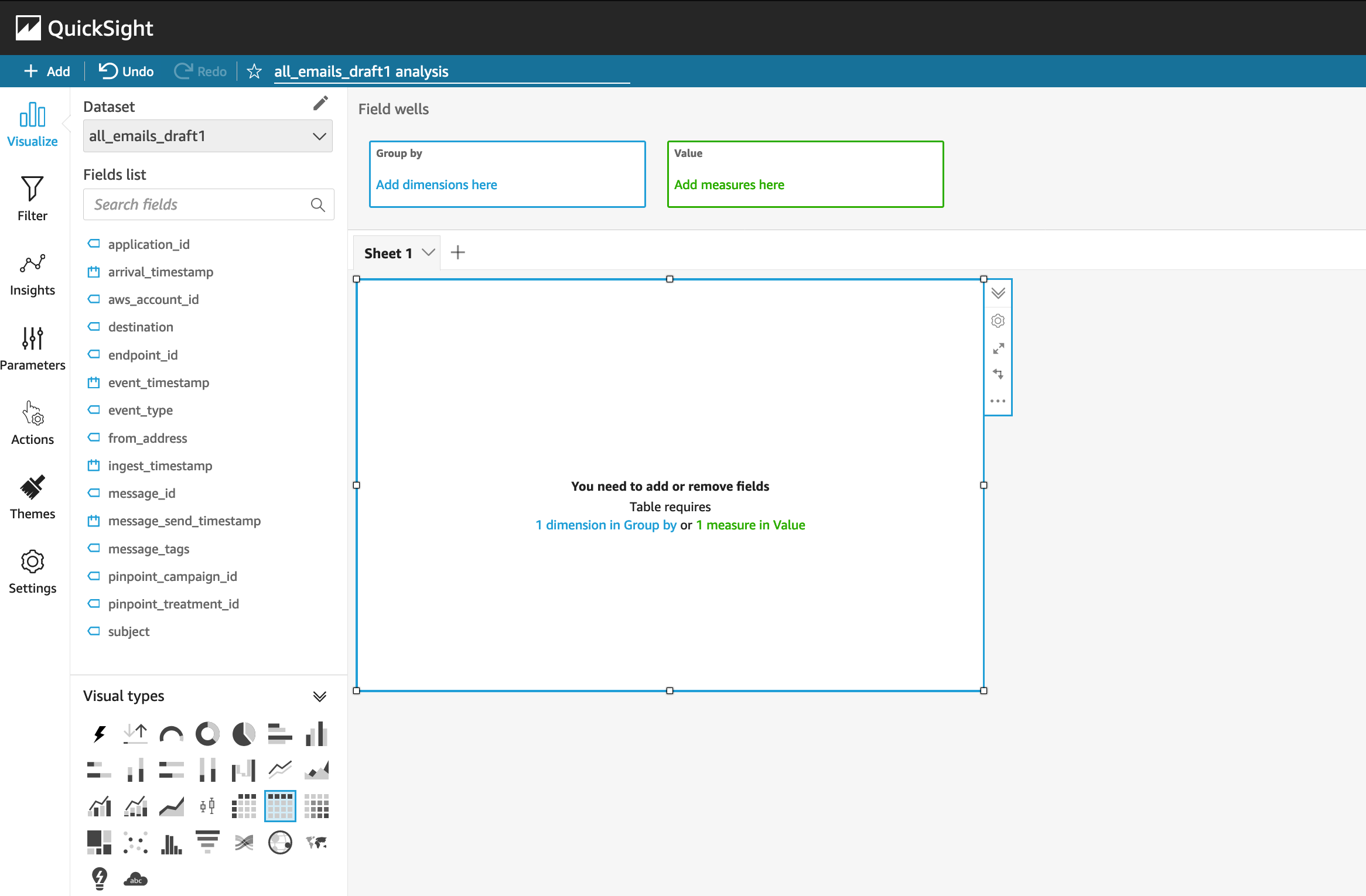
* Select the event data location for analysis. There are mainly two options available which are as follows. Then click on visualize the data.
  + **Import to Spice quicker analysis -** SPICE is the Amazon QuickSight Super-fast, Parallel, In-memory Calculation Engine. It's engineered to rapidly perform advanced calculations and serve data. In Enterprise edition, data stored in SPICE is encrypted at rest. (1 GB of storage is available for free for extra storage customer need to pay extra)
  + **Directly query your data** – This process enables Quicksight to query directly to the Athena or source database (In the current case it is Athena).



* Blank analysis page looks something as follows, please drag and drop what type of graph you need to visualize onto the auto graph pane

* Please note that Amazon QuickSight is a Busines intelligence platform, so customer can the desired graphs for visualizing the individual events.
* As part of this blog, we have displayed how to create some simple analysis graphs to visualize the engagement events.
* As an initial step please Select tabular graph.

* tep 1 – Create AWS account & Pinpoint Project

If you have an AWS account and Pinpoint Project setup already please move to step 2 [Create an AWS account](https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/)

[Create a Pinpoint project](https://docs.aws.amazon.com/pinpoint/latest/userguide/gettingstarted-create-project.html)

## Step 2 – Create S3 bucket for Lambda code and upload the Zip files

[Create an S3 bucket](https://docs.aws.amazon.com/AmazonS3/latest/userguide/create-bucket-overview.html) in the region that you have your Pinpoint projects and provide it a unique name

Upload in the root folder the 2 zip files: lambda\_aggregator.zip and lambda\_timeseries.zip

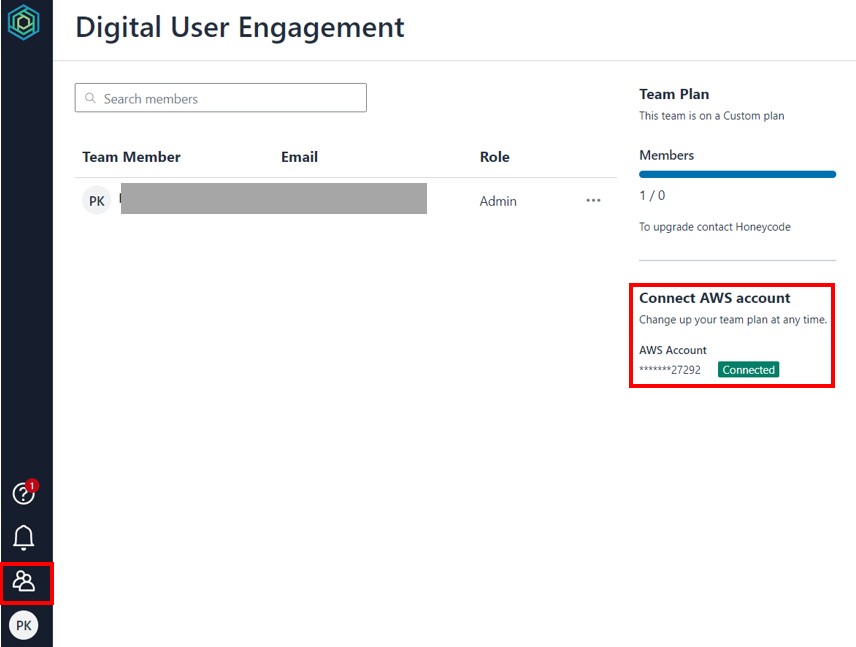
**Note:** When creating the S3 bucket make sure that it is in the region you want to deploy the rest of the project.

## Step 3 – Create HoneyCode account and workbook

On the AWS console, select region US West (Oregon) us-west-2 and click on “Sign up for Honeycode”

**Note:** HoneyCode is a standalone app and an AWS service that charges separately and you will need to link it to your AWS account for the purpose of this solution

When the HoneyCode application loads, click on the 3rd icon from the top to bottom on the navigation bar and connect your AWS account

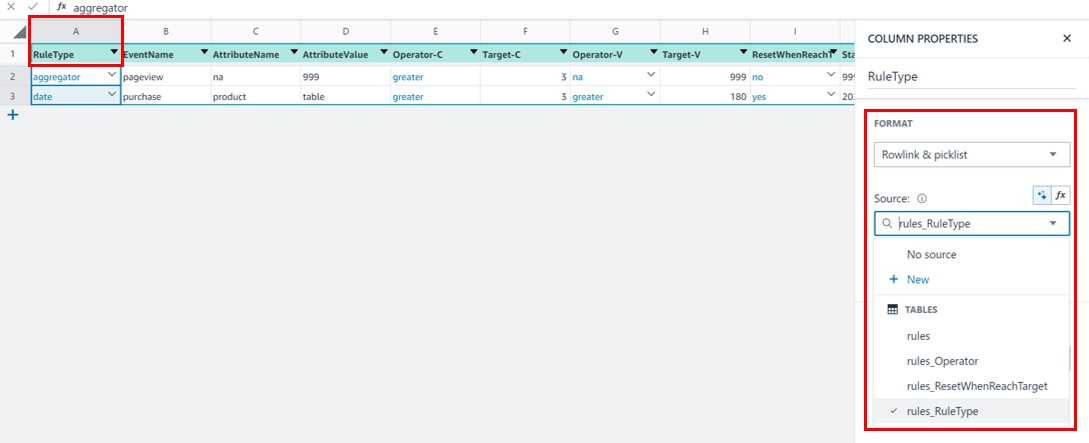


Once you have linked your HoneyCode account with your AWS account, return to the HoneyCode homepage (<https://builder.honeycode.aws/>) and click up right “Create workbook”

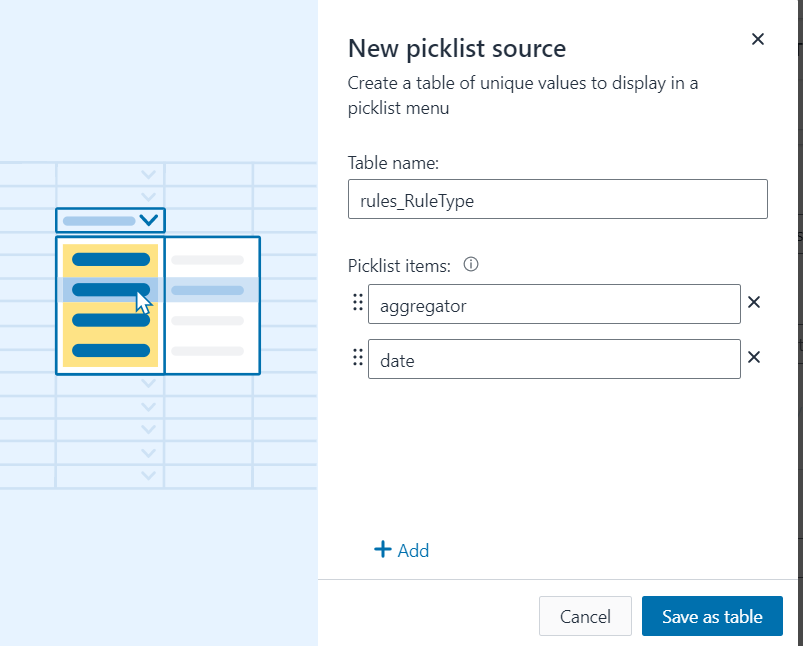
Select “Start from scratch” option

Provide a name to your workbook and select the team you want to link it with

**Note:** For the table we will create, some columns will be using dropdowns with pre-selected values. To create such dropdowns, click right click on the column letter and select “Format…”. A sliding bar from the right side should appear with title “COLUMN PROPERTIES”, select the option “New” under the “Source” list



The latter will create a new table, where for the source value. Provide a table name and the values you would like on the dropdown (example below for the column “RuleType”)



Create the following columns with the exact names and order:

### RuleType:

* + Format: Rowlink & picklist
  + Dropdown values: aggregator, date
  + Description: Rules refer to a set of criteria the business user is defining. These criteria will then be used to qualify customers and allocate them to segments.
    - Aggregator: This rule type does not take into consideration dates and the moment it is implemented, the event of that rule will start being recorded in the Aggregator DynamoDB table
    - Date: This rule is similar with the Aggregator but it looks for customers only for the defined time period the user set for that rule.

### EventName:

* + Format: Auto
  + Description: The event name should be the exact name of the event that you want to track as recorded in Amplify and shown in the Pinpoint platform

### AttributeName:

* + Format: Auto
  + Description: Attribute refers to the event attribute (field above). It offers an extra level of granularity e.g. Event = Purchase & EventAttirubte = Product. If you do not want to use the attribute field, then type “na”. Any deviations on the spelling of the “na” might result to errors. The EventAttribute should be written exactly as recorded in Amplify and shown in the Pinpoint platform.

### AttributeValue:

* + Format: Auto
  + Description: This field is related to the AttributeName field and here you should type the value of the event attribute you want this rule to be on e.g. Event = Purchase & EventAttribute = Product & AttributeName = Chair. If you already have AttributeName = na then on the AttributeValue field type 999. The AttributeValue should be written exactly as recorded in Amplify and shown in the Pinpoint platform.

### Operator-C:

* + Format: Rowlink & picklist
  + Dropdown values: na, equal, greater
  + Description: This field refers to the operator used for the count of that event. E.g. customers who had more than 5 purchases. If you don’t want to use this field then select the “na” option from the dropdown.

### Target-C:

* + Format: Number
  + Decimal Places: 0
  + Description: This field contains the numerical value that you want your customers’ event count to be evaluated against. If in the field Operator-C you have selected “na” then type 999

### Operator-V:

* + Format: Rowlink & picklist
  + Dropdown values: na, equal, greater
  + Description: This field refers to the operator used for the sum of that event’s metric.

E.g. customers whose purchase sum is greater than $500. If you don’t want to use this field then select the “na” option from the dropdown.

### Target-V:

* + Format: Number
  + Decimal Places: 0
  + Description: This field contains the numerical value that you want your customers’ event metric to be evaluated against. If in the field Operator-V you have selected “na” then type 999

### ResetWhenReachTarget:

* + Format: Rowlink & picklist
  + Dropdown values: yes, no
  + Description: This field is applicable only for RuleType = Aggregator

### StartDate:

* Yes: When the rule criteria are met, then both count and metric sum values will be set to 0. Furthermore, when the rules are met an event is fired with the following naming convention trk\_eventname and if the event has an attribute trk\_eventname\_attributevalue. When selecting yes, you can setup a Journey and allow endpoints / users to enter multiple times based on the event you have specified in that rule
* No: When the rule criteria are met, then an attribute with the naming convention trk\_eventname and if the event has an attribute

trk\_eventname\_attributevalue is updated to “ready”. The count and metric sum values do not change once the rule criteria are met.

* Important: The values “yes” and “no” should be all lowercase and with no spaces.
  + Format: Plain text
  + Description: Applicable only for RuleType = Date. Specifies the starting date of the period you want to filter for. The format should be YYYY-MM-DD and if your RuleType = aggregator then type 9999-99-99.

### FinishDate:

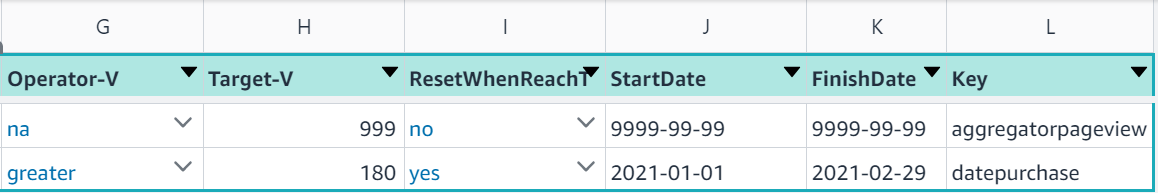
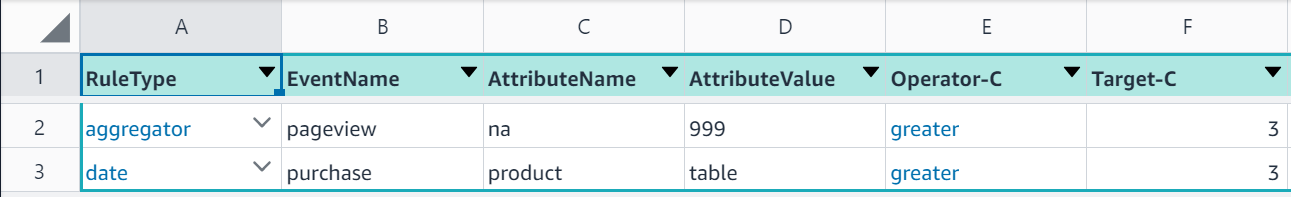
* + Format: Plain text
  + Description: Applicable only for RuleType = Date. Specifies the finishing date of the period you want to filter for. The format should be YYYY-MM-DD and if your RuleType = aggregator then type 9999-99-99.

### Key:

* + Format: Auto
  + Formula: =CONCATENATE([RuleType],[EventName])
  + Description: This field is auto generated based on the formula above and used from Lambda Aggregator & Time series to query the correct rules for each.

Below you can find a screenshot of the table in HoneyCode with two examples and how it should look after you complete the steps above.

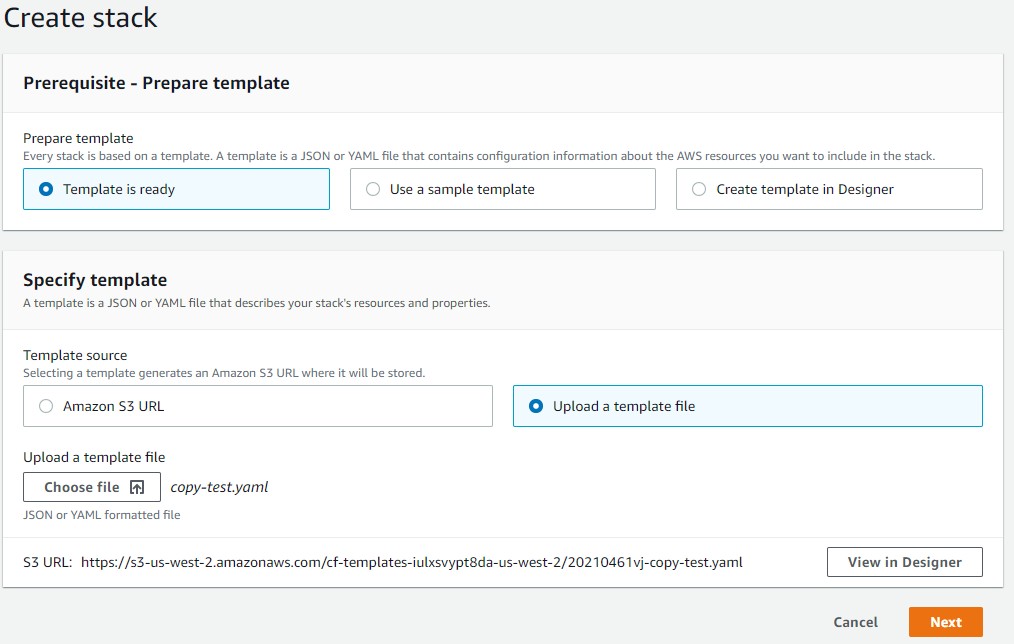
**Note:** The table is presented into two screenshots, first one columns A – F and second one columns G – L. Make sure that all predefined values, formats, and columns are the same.



## Step 4 – Create Cloudformation Stack

Navigate to Cloudfromation page in AWS console, click up right on “Create stack” and select the option “With new resources (standard)”

Leave the “Prerequisite – Prepare template” to “Template is ready” and for the “Specify template” option, select “Upload a template file”. On the same page, click on “Choose file”, browse to find the file “incremental\_analytics\_pinpoint.yaml” file and select it. Once the file is uploaded, click “Next”



See below information for each of the 6 fields under the section “Specify stack details”:

|  |  |
| --- | --- |
| **CloudFormation**  **Fields** | **Values - Description** |
| **Stack name** | Provide a name of your preference for that Cloudformation stack. |
| **EventStreamARN** | If you already have a Pinpoint Kinesis stream setup for the project you want to implement this stack, then copy paste its ARN otherwise leave it empty and a new Pinpoint Event Stream will be created as part of this  Cloudformation stack. |

|  |  |
| --- | --- |
| **HoneyCodeTableID & HoneyCodeWorkbo okID** | Open the HoneyCode workbook and copy the URL, it should look like this “https://builder.honeycode.aws/table/1-us-west- 2%3A122162422134%3Atable%3A111c6cea-af2a-41bb-ba31- 92759c730076%2F636dfdb5-232b-468f-890c-92de2ae1c87a”.  Copy the URL visit <https://www.urldecoder.org/>and paste the URL on the “Decode from URL-encoded format” text box and then click on the “DECODE” CTA. See below the expected result.    Once you complete the decoding, your original URL should look like this:  https://builder.honeycode.aws/table/1-us-west- 2:122162422134:table:111c6cea-af2a-41bb-ba31- 92759c730076/636dfdb5-232b-468f-890c-92de2ae1c87a  The part highlighted in green is the workbook ID and the one highlighted in blue is the table ID.  Unfortunately at the moment there is no easier way to extract both of these values. |
| **LambdaCodeBucket**  **Name** | Type the name of the S3 bucket from step 2 |
| **PinpointProjectId** | Copy paste the Pinpoint Project ID, which you can find on the Pinpoint  console page under “All projects” |

|  |  |
| --- | --- |
|  |  |

Once all fields completed, click “Next”.

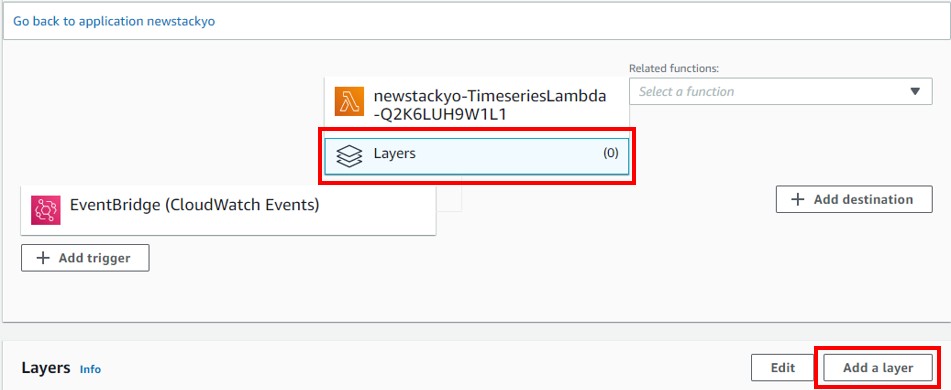
On the “Configure stack options” page, click “Next”

On the “Review [StackName]” page, check the checkbox “I acknowledge that AWS CloudFormation might create IAM resources.” And then click on “Create stack”.

## Step 5 – Create Lambda Layer

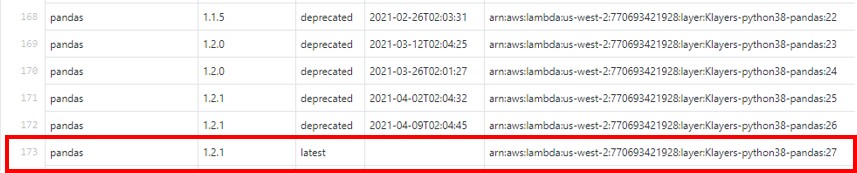
Once step 4 is completed successfully, navigate to Lambda => Functions and click on the function name that starts with “[StackName]-TimeseriesLambda-”

Click on the “Layers” and then click on “Add a layer”

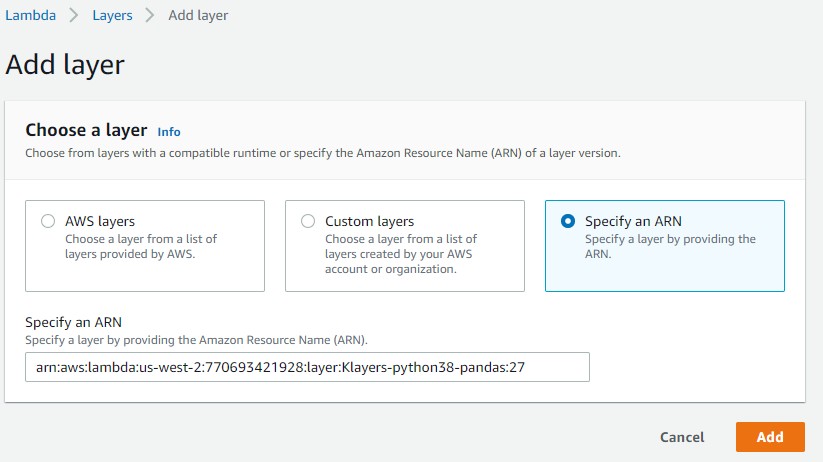


Depending the Region you have deployed the stack, visit this [Github page,](https://github.com/keithrozario/Klayers/tree/master/deployments/python3.8/arns) select the respective

Region from the list and click on it. Scroll down the list, find where the column “Package” = Pandas and select the “Package Version” = Latest



When creating the Lambda Layer, select the option “Specify ARN” paste the ARN obtained from the step above and click “Add”

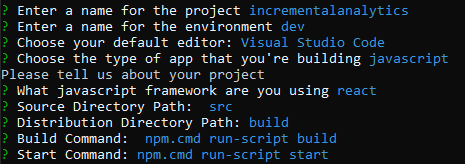


# Steps to test the solution

To test the solution you will require to have a web or native application that is generating Pinpoint events and uses Cognito for user management. If you don’t have the above and you wish to test the functionality, then follow the steps listed below:

**IMPORTANT:** Once you complete the steps below, a new Pinpoint Project will be created in the Region of the AWS account you are using. For testing purposes, use the project created from Amplify to deploy and test the solution .

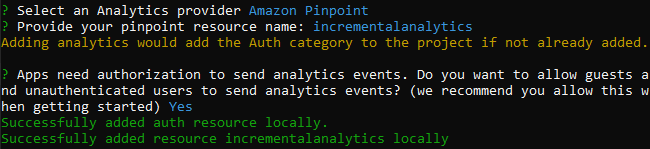
1. Complete the Amplify Tutorial – Prerequisites step in this [page](https://docs.amplify.aws/start/getting-started/installation/q/integration/js)
2. On your desktop create a folder and name it “Incremental\_Analytics\_Pinpoint”
3. Open the command prompt or Powershell if you are in Windows, navigate to the folder “Incremental\_Analytics\_Pinpoint”, type and execute: npm create-react-app incrementalanalytics
4. The above step will create a new React web application with the name of “incrementalanalytics”
5. Browse to the folder “incrementalanalytics”
6. Type and execute “amplify init”
7. It will ask you a set of questions, answer as per the sceenshot below:

a. 

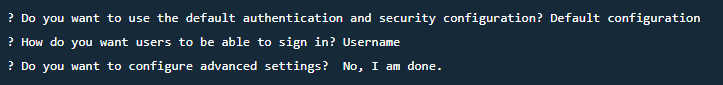
1. To the question “Do you want to use an AWS profile?” answer Yes and select the profile you

would like to have this solution implemented

1. Once the above step is completed, type and execute:
   1. Amplify add analytics:

i. 

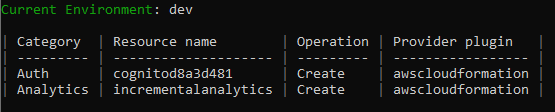
* 1. Amplify add auth:

i. 

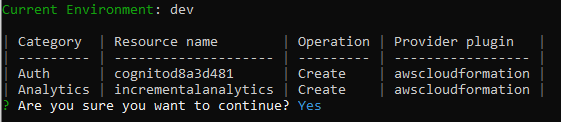
ii. Note: If you get a message, which says that Auth is already configured for

this project, then type and execute: *amplify update auth*

1. Type and execute: *amplify status*. You should see that there are 2 resources where column Operation = Create

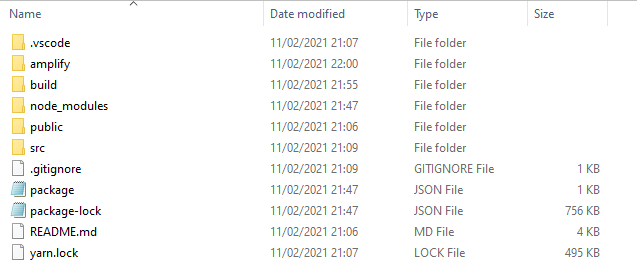
a.

1. Type and execute: *amplify push*. In the question “Are you sure you want to continue?” type

Y and press Enter

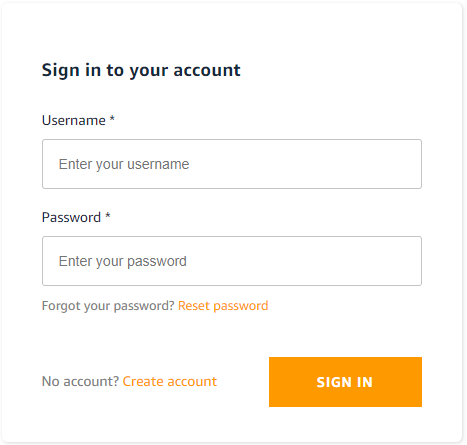
a.

1. Type and execute: *npm install aws-amplify @aws-amplify/ui-react*
2. Type and execute: *npm install react-bootstrap bootstrap*
3. Exit the comand prompt or Power Shell and go to the project folder, which should have the below contents

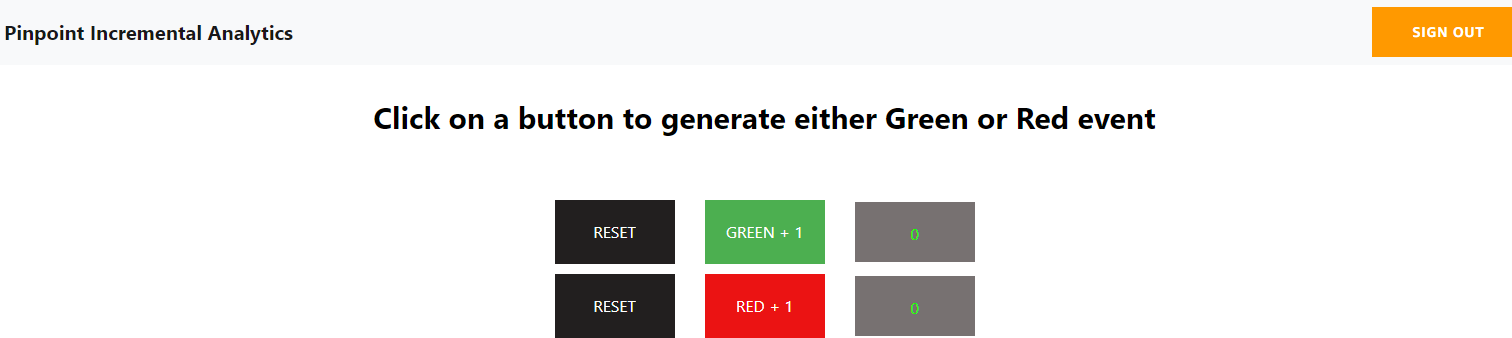


a.

1. Open the “scr” folder, paste & replace the App.js and index.css files with these two
2. Go back and open the “public” folder, paste & replace the index.html file with this one
3. Open the comand prompt or Power Shell, navigate to the app folder “Incremental\_Analytics\_Pinpoint”, type and execute: npm start
4. The above should open your default browser localhost:3000 and display the below:

a. 

1. Click on Create account, fill all information required and click on “Create Account”
2. You should receive the activation code on the email you used to sign-up, enter it and click “Activate”
3. Once you are logged in, you should see the screen below:



a.

1. When clicking either “Green” or “Red” a Pinpoint event with name greenbutton and

redbutton respectively will fire

### Note:

* + 1. Both events have a metric value of 1
    2. Both events have the following series of attributes: pagename:'homepage', Timestamp: timestamp, ChannelType: 'EMAIL', Address: email

# How to use the solution

Case 1 – Move users to segment based on event count / metric sum target **Description:** Move customers to a specific segment when the count or metric sum of an event equals or is greater than a specific value

**Example:** Users who have purchased X product more than 5 times should move to a dynamic segment “Users with interest in X”

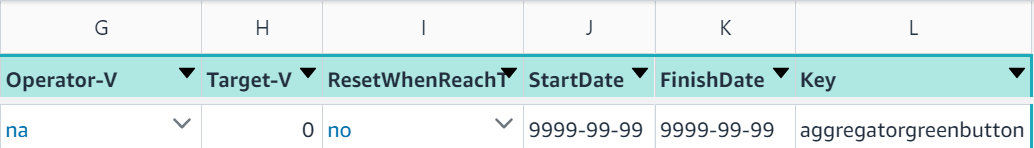
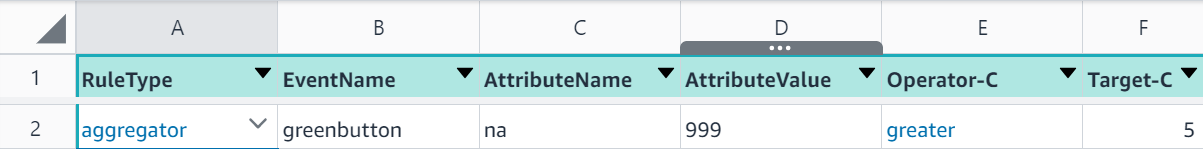
Execution:

1. Get the name of the event you want to include in the rule
2. Go to HoneyCode and add a new row in the “rules’” table
   1. Select RuleType = aggregator
   2. Enter the EventName value as it shows on the Pinpoint console or you can open the App.js from the React App and view the exact values of the event name. For this case if you have implemented the “Steps to test the solution” type greenbutton
   3. If you are planning to use an attribute then type the exact attribute name e.g.

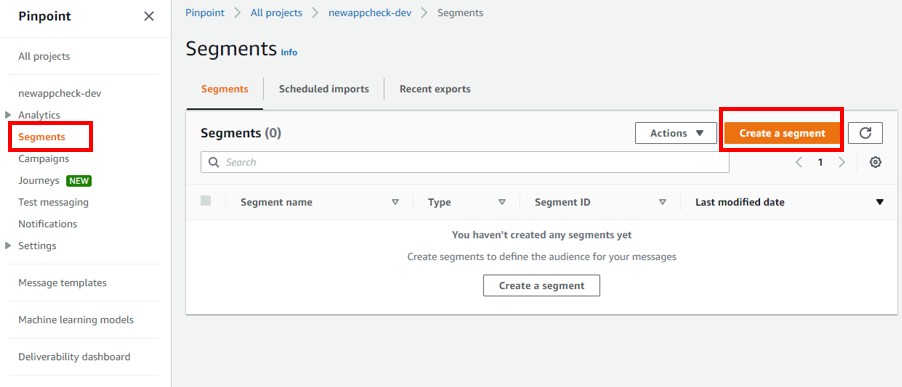
pagename as in Pinpoint otherwise type “na” – make sure you don’t have any spaces and “na” is written all lowercase. For this case type na

* 1. If you have typed an event attribute then type the attribute value e.g. homepage otherwise type 999
  2. Select Operator-C = greater, Target-C = 5
  3. Select Operator-V = na, Target-V = 999
  4. Select ResetWhenReachTarget = no
  5. For StartDate & FinishDate type 9999-99-99
  6. The column Key will populate automatically

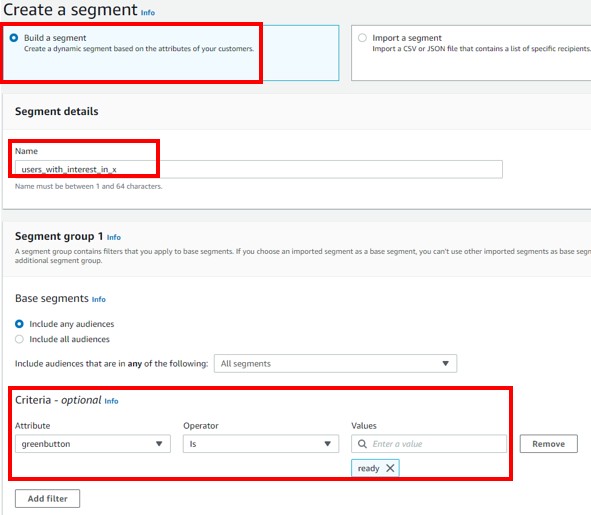
Your HoneyCode table should look like this (table is depicted into two screenshots)



1. Go to your Pinpoint project, click on Segments and then Create a segment

a.

1. In the segment page, select “Build a segment”, give the name “users\_with\_interest\_in\_x”

and select the attribute with name of the event, Operator = is and Values = ready. NOTE: The attribute won’t appear till at least one user meets the criteria set in HoneyCode. For the latter to happen, you might need to wait till at least a user gets qualified.

a.

## Case 2 – Qualify users for a Pinpoint Journey every time an event count / metric sum meets the target

**Description:** Have the ability to trigger an event when a rule is met (event count or metric sum) and reset the count / sum

**Example:** Every time a user completes 3 courses in an e-learning platform, send them an email to congratulate them

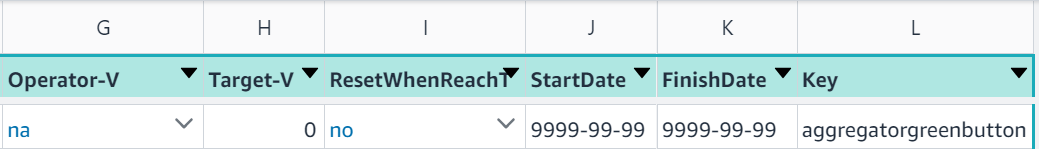
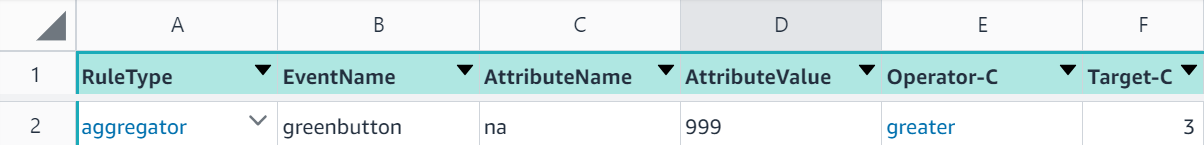
### Execution:

1. Get the name of the event you want to include in the rule
2. Go to HoneyCode and add a new row in the “rules’” table
   1. Select RuleType = aggregator
   2. Enter the EventName value as it shows on the Pinpoint console or you can open the App.js from the React App and view the exact values of the event name. For this case if you have implemented the “Steps to test the solution” type greenbutton
   3. If you are planning to use an attribute then type the exact attribute name e.g.

pagename as in Pinpoint otherwise type “na” – make sure you don’t have any spaces and “na” is written all lowercase. For this case type na

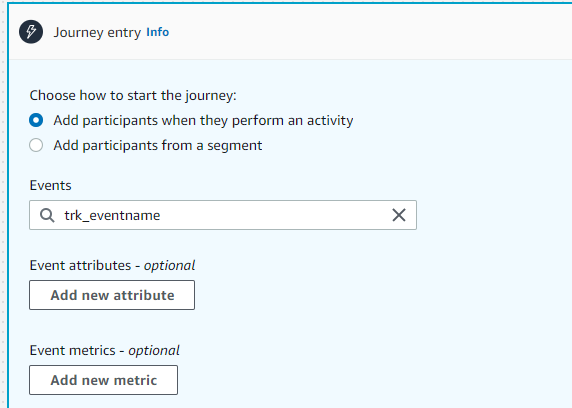
* 1. If you have typed an event attribute then type the attribute value e.g. homepage otherwise type 999
  2. Select Operator-C = greater, Target-C = 3
  3. Select Operator-V = na, Target-V = 999
  4. Select ResetWhenReachTarget = yes
  5. For StartDate & FinishDate type 9999-99-99
  6. The column Key will populate automatically

Your HoneyCode table should look like this (table is depicted into two screenshots)

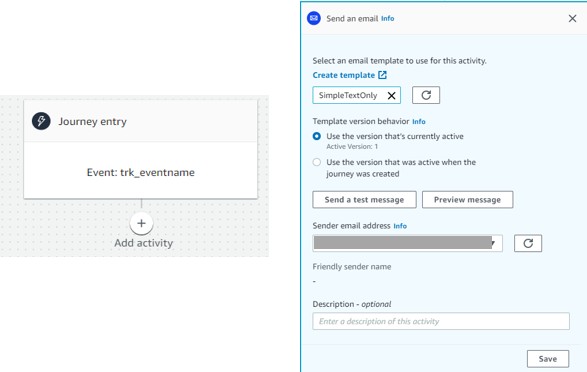


1. Go to the Pinpoint project and create a Journey
2. For the Journey Entry, select the journey to start option “Add participants when they

perform an activity” and for the Events, type the event name but with trk\_ in front of it e.g. trk\_eventname. NOTE: The event might not appear in the list but you can type it

a.

1. Click on Add activity and select Send email. You will need to have Email channel enabled for

this project and a ready email template

a.

1. Click “Save”
2. Click on the “Actions” top right of the screen and select “Settings”. On the overlay that will appear, enter a “Journey title”, Start / End data and click to expand “Advanced settings – optional”
3. In this section you should change the “Journey limits” and “Maximum entries per endpoint” to numbers higher than 0, since you would like users to receive an email every time they complete 3 courses, which will result to multiple Journey entries

## Case 3 – Move users to a segment based on event count / metric sum target for a specific time period

**Description:** Move customers to a specific segment when the count or metric sum of an event equals or is greater than a specific value for a specific time period

**Example:** Users who have spent X $500 between 2021-01-01 and 2021-02-01 should move to a dynamic segment “Users with interest in X”

**Note:** If you would like to see results from this example then you will need to manually insert records in DynamoDB time series table, where the total value of the metric will be higher than 500. If you do the above and then execute the Timeseries Lambda, the attribute trk\_date\_greenbutton will be updated to ready.

### Execution:

1. Get the name of the event you want to include in the rule
2. Go to HoneyCode and add a new row in the “rules’” table
   1. Select RuleType = date
   2. Enter the EventName value as it shows on the Pinpoint console or you can open the App.js from the React App and view the exact values of the event name. For this

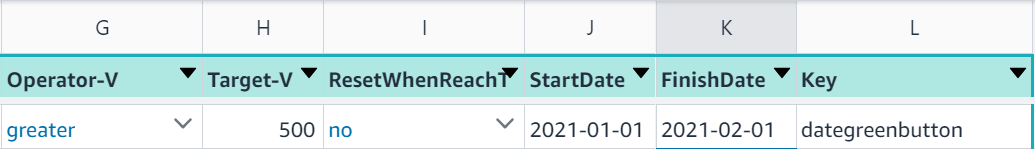
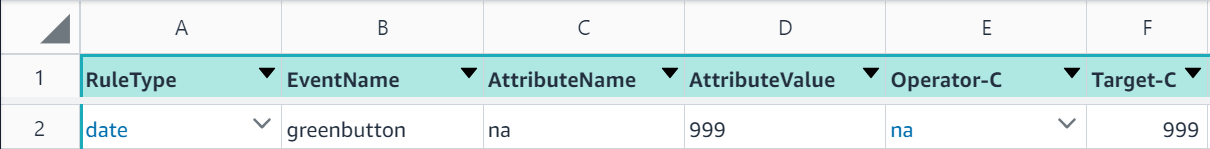
case if you have implemented the “Steps to test the solution” type greenbutton

* 1. If you are planning to use an attribute then type the exact attribute name e.g.

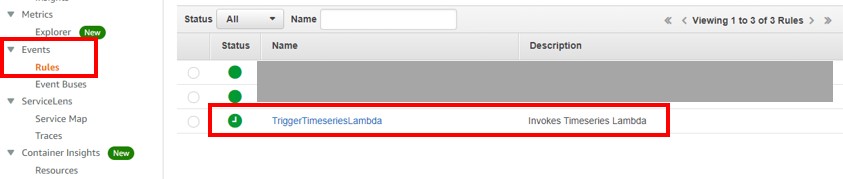
pagename as in Pinpoint otherwise type “na” – make sure you don’t have any spaces and “na” is written all lowercase. For this case type na

* 1. If you have typed an event attribute then type the attribute value e.g. homepage otherwise type 999
  2. Select Operator-C = na, Target-C = 999
  3. Select Operator-V = greater, Target-V = 500
  4. Select ResetWhenReachTarget = no
  5. StartDate = 2021-01-01
  6. FinishDate = 2021-02-01
  7. The column Key will populate automatically

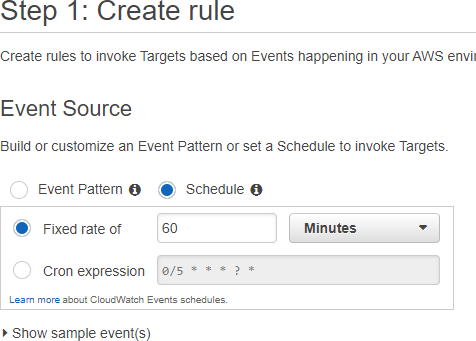
Your HoneyCode table should look like this (table is depicted into two screenshots)



1. All rules where RuleType = date will be evaluated automatically every 60 minutes (default value) based on a CloudWatch Event Rule named “TriggerTimeseriesLambda”

a.

1. If you want to change the interval, click on the rule in the CloudWatch page then from the

upright Actions => Edit and under the “Event Source” section you will see that the option “Schedule” is preselected and the “Fixed rate of” is set to 60 minutes

a.