PINPOINT INCREMENTAL ANALYTICS – V2.1

Ioannou Katidis, Pavlos
AWS DUE Specialist Solutions Architect

Pinpoint – Incremental Events with Period Based Filtering

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.

Solution Summary

Background

Tracking users' activity is vital for understanding your customers and identifying opportunities. In a data driven world, having actionable data is fundamental but being able to feed these data into your marketing platforms and use them in business logic often comes as a challenge. It is not uncommon for companies in such situations to either resort in manual processes resulting in missed opportunities or try to build something in-house, which won't be scalable. Currently Pinpoint cannot aggregate events neither create segments based on an event's count for a specific period.

Solution

This solution expands Pinpoint's existing segmentation and event based journey capabilities, allowing the creation of business rules on events' count and / or summed metric value per user with the possibility to add date filters. When a rule is met, a user attribute with the event name will be updated to "Ready" or an event with the name trk_event will be fired depending the needs of the Pinpoint user. Business rules will be added through HoneyCode and will require no coding experience.

Use case(s)

User segmentation based on:

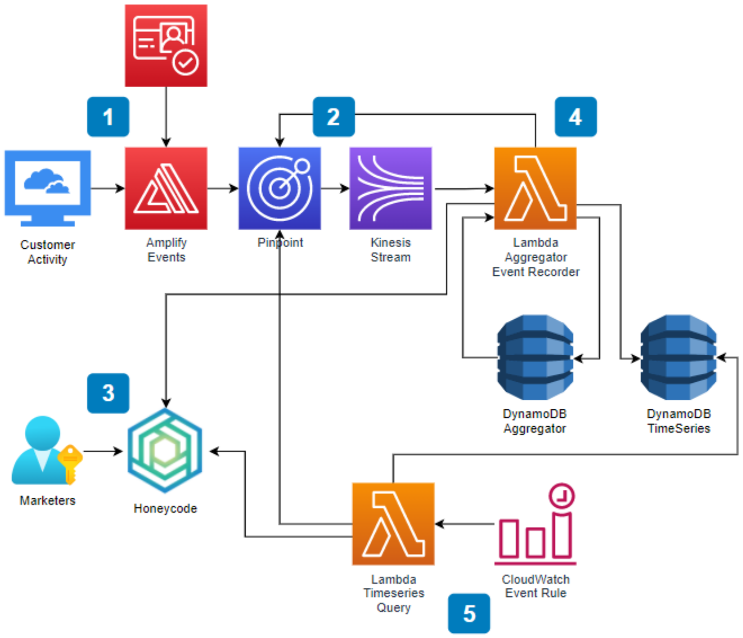
- Aggregated user activity
- Aggregated user activity throughout a specified period

Considerations

1) You will need to install Amplify SDK for sending events to Pinpoint and Cognito for user management
2) Only custom metrics are being processed at the moment
3) HoneyCode is available only in Oregon (us-west-2) region at the moment but the solution will work in any region
4) When you insert a rule that is using the date filters in Honeycode you will need to wait till it performs its first scan based on the CloudWatch Event Rule scheduling settings. By default, the project sets the CloudWatch Event Rule interval to 60 minutes
5) You can only define date periods with start and end date. The date format needs to be strictly YYYY-MM-DD
6) The Lambda used for time series querying is using Pandas library as a Layer (ARN of the layer is public)
7) All metric calculations and user attribute changes are made on a user level and NOT on an endpoint level
8) Build the HoneyCode table as per this guide otherwise the solution won't work as expected. There is a dependency between the Lambda code and the order of the HoneyCode columns
9) Make sure that the values you are inserting in HoneyCode match exactly the values in Pinpoint (case, spaces etc.)
10) When removing an aggregate business rule from HoneyCode then all related records from the DynamoDB aggregate table will be deleted (assessment every 60 minutes via Time Series Lambda CloudWatch Event Rule)
11) Aggregate table records only events that match HoneyCode rules, whereas time series table records all events

Solution Architecture & Business Logic

Solution Architecture



1 Using Amplify SDK and Cognito, you can send custom events to Pinpoint that are enriched with User ID (for logged in users)

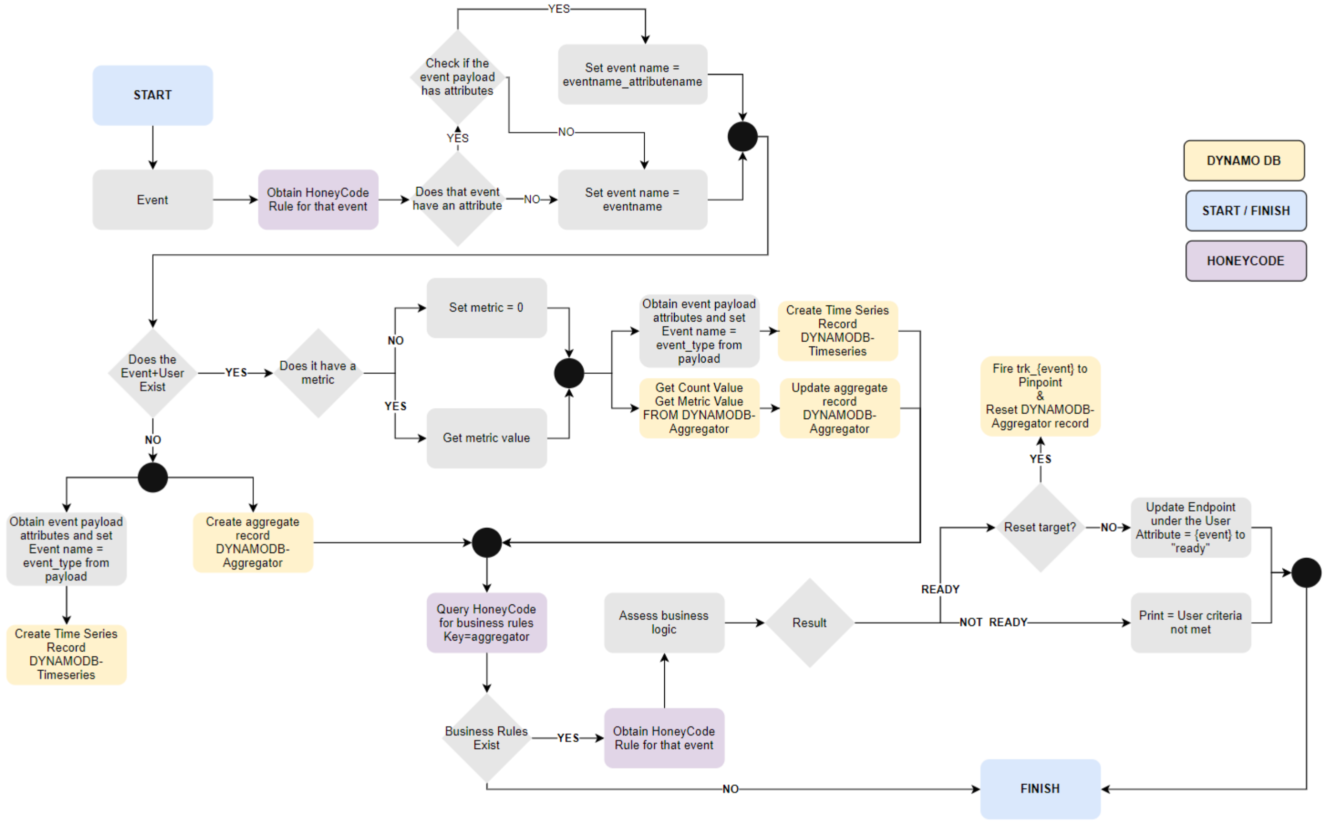
2 An existing Pinpoint features allows to stream all Pinpoint event through Kinesis Stream. This solution can create an existing Pinpoint Kinesis Stream or if there isn't one it will create a new one

3 Honeycode will function as a visual interface for business users, where business users / marketers can input business rules that will define the target value for an event (count or sum) per user and if that target should be reset after it has been reached e.g. if cars pageview > 50
Another type of business utilizes dates, where users can set a date range and a target for that event e.g. all users who clicked more than 3 times on a car between 2021-01-10 and 2021-02-12

4 Lambda Aggregator - Event Recorder will be processing all Pinpoint custom events and based on the business rules set by the users in Honeycode it will perform a put or update function to the two DynamoDB tables. DynamoDB aggregator will aggregate records by user & event while DynamoDB timeseries will be recording each event by date, timestamp and user ID. When criteria are met, a Pinpoint user attribute with the name of the event will be updated from "not-ready" to "ready". This will allow business users to setup event based journeys or dynamic segments

5 Lambda Timeseries Query will be triggered by a CloudWatch Event Rule every 60 minutes (can be changed) and query Honeycode for business rules. Based on the business rules obtained it will then query DynamoDB Timeseries and if any users meet the criteria then a user attribute with the name of the event trk_date_eventname will be updated to "Ready"

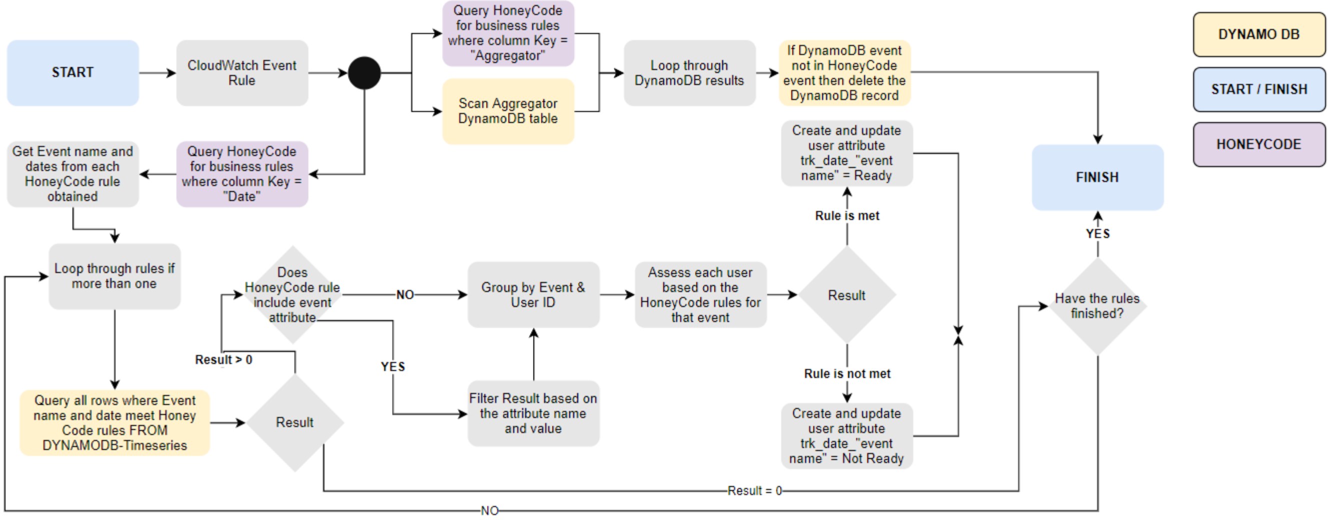
Business logic for Lambda Aggregator



START
Event
Obtain HoneyCode Rule for that event
Does that event have an attribute
NO
YES
Check if the event payload has attributes
YES
NO
Set event name = eventname_attributename
Set event name = eventname
DYNAMO DB
START / FINISH
HONEYCODE
Does the Event+User Exist
YES
Does it have a metric
NO
Set metric = 0
YES
Get metric value
Obtain event payload attributes and set Event name = event_type from payload
Create Time Series Record DYNAMODB-Timeseries
Get Count Value Get Metric Value FROM DYNAMODB-Aggregator
Update aggregate record DYNAMODB-Aggregator
NO
Obtain event payload attributes and set Event name = event_type from payload
Create Time Series Record DYNAMODB-Timeseries
Create aggregate record DYNAMODB-Aggregator
Query HoneyCode for business rules Key=aggregator
Business Rules Exist
YES
Obtain HoneyCode Rule for that event
Assess business logic
Result
READY
NOT READY
Fire trk_{event} to Pinpoint & Reset DYNAMODB-Aggregator record
YES
Reset target?
NO
Update Endpoint under the User Attribute = {event} to "ready"
Print = User criteria not met
NO
FINISH

No production-ready

Business logic for Lambda TimeSeries


START
CloudWatch Event Rule
Query HoneyCode for business rules where column Key = "Aggregator"
Scan Aggregator DynamoDB table
Loop through DynamoDB results
If DynamoDB event not in HoneyCode event then delete the DynamoDB record
DYNAMO DB
START / FINISH
HONEYCODE
Get Event name and dates from each HoneyCode rule obtained
Query HoneyCode for business rules where column Key = "Date"
Create and update user attribute trk_date_"event name" = Ready
FINISH
Rule is met
YES
Loop through rules if more than one
Does HoneyCode rule include event attribute
NO
Group by Event & User ID
Assess each user based on the HoneyCode rules for that event
Result
Have the rules finished?
Result > 0
YES
Rule is not met
Query all rows where Event name and date meet Honey Code rules FROM DYNAMODB-Timeseries
Result
Filter Result based on the attribute name and value
Create and update user attribute trk_date_"event name" = Not Ready
Result = 0
NO

Steps to implement the solution

Step 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

Create a Pinpoint project

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

Create an S3 bucket 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

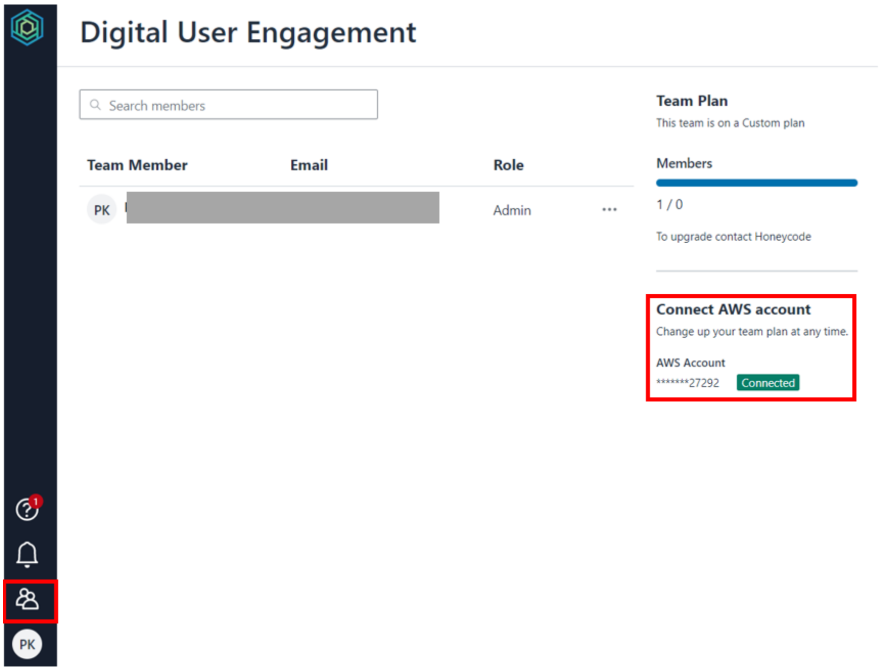
Note: When creating the S3 bucket make sure that it is in the region you want to deploy the rest of the project.

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



Digital User Engagement

Team Member	Email	Role
PK	[redacted]	Admin ···

Team Plan

This team is on a Custom plan

Members

1 / 0

To upgrade contact Honeycode

Connect AWS account

Change up your team plan at any time.

AWS Account

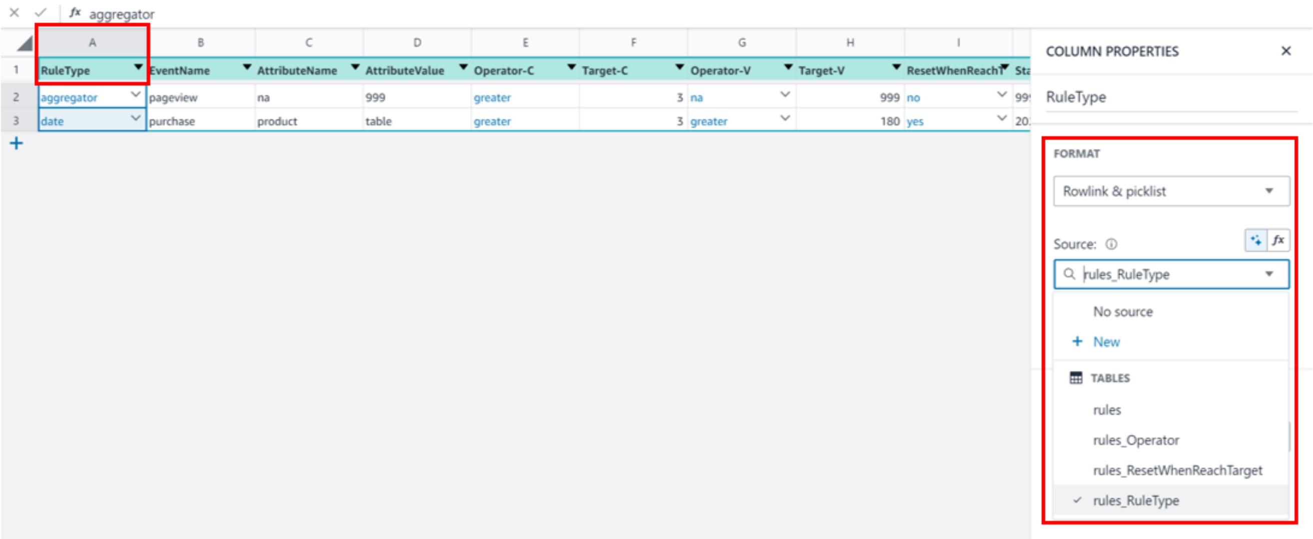
*******27292 Connected

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



fx aggregator

	A	B	C	D	E	F	G	H	I	
1	RuleType	EventName	AttributeName	AttributeValue	Operator-C	Target-C	Operator-V	Target-V	ResetWhenReachT	Sta
2	aggregator	pageview	na	999	greater	3	na	999	no	99
3	date	purchase	product	table	greater	3	greater	180	yes	20

COLUMN PROPERTIES

RuleType

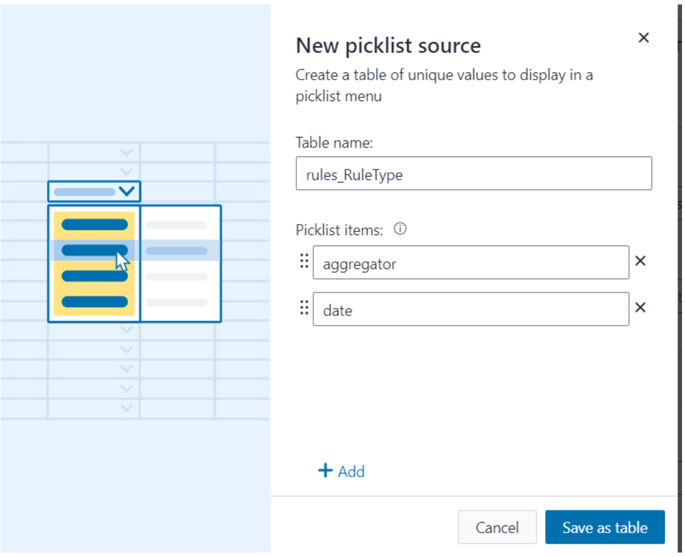
FORMAT

Rowlink & picklist

Source:

rules_RuleType

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”)



New picklist source

Create a table of unique values to display in a picklist menu

Table name: rules_RuleType

Picklist items:

aggregator

date

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.

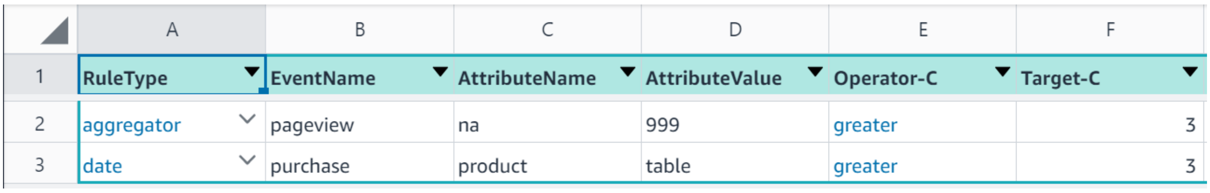
- **EventName:**
  - Format: Auto
  - Description: The event name should be the exact name of the event that you want to track as recorded in Amplify and shown in the Pinpoint platform
- **AttributeName:**
  - Format: Auto
  - Description: Attribute refers to the event attribute (field above). It offers 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

        - 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.

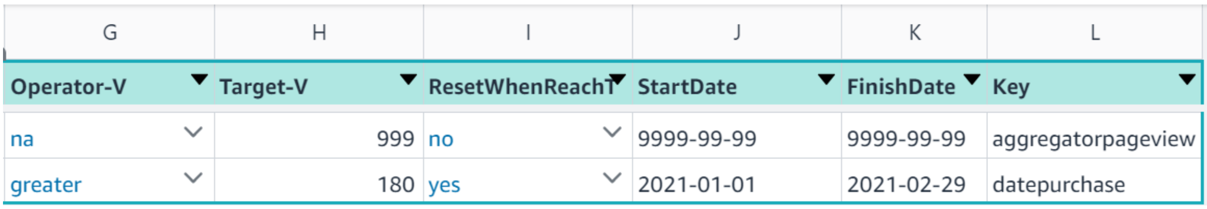
- **StartDate:**
    - Format: Plain text
    - Description: Applicable only for RuleType = Date. Specifies the starting date of the period you want to filter for. The format should be YYYY-MM-DD and if your RuleType = aggregator then type 9999-99-99.
- **FinishDate:**
    - Format: Plain text
    -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



	A	B	C	D	E	F
1	RuleType	EventName	AttributeName	AttributeValue	Operator-C	Target-C
2	aggregator	pageview	na	999	greater	3
3	date	purchase	product	table	greater	3

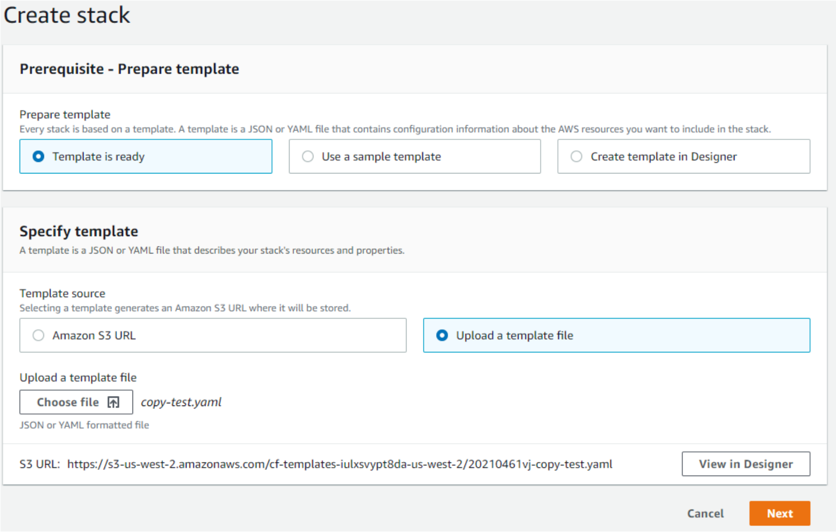


G	H	I	J	K	L
Operator-V	Target-V	ResetWhenReachT	StartDate	FinishDate	Key
na	999	no	9999-99-99	9999-99-99	aggregatorpageview
greater	180	yes	2021-01-01	2021-02-29	datepurchase

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”



Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

● Template is ready ○ Use a sample template ○ Create template in Designer

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

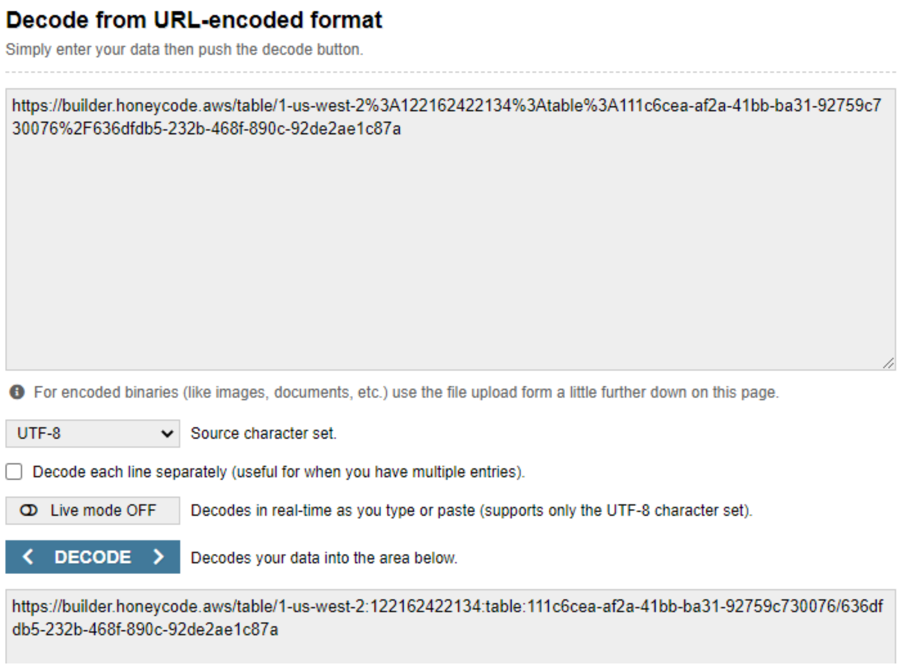
○ Amazon S3 URL ● Upload a template file

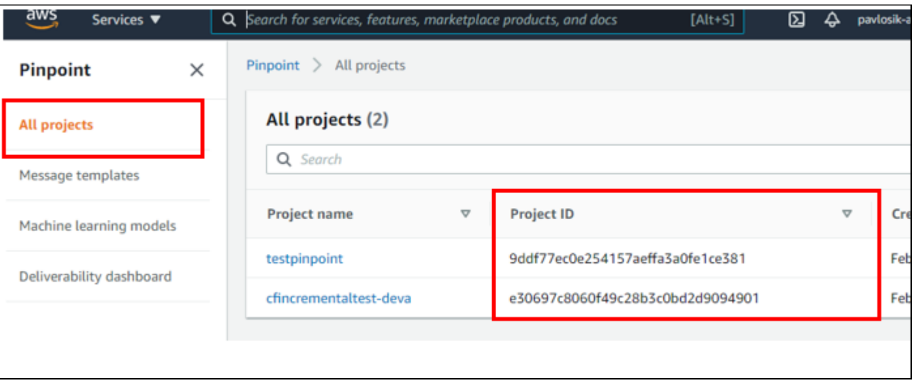
Upload a template file
copy-test.yaml
JSON or YAML formatted file

S3 URL: https://s3-us-west-2.amazonaws.com/cf-templates-iulxsvypt8da-us-west-2/20210461vj-copy-test.yaml

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 & HoneyCodeWorkbookID	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.  **Decode from URL-encoded format** Simply enter your data then push the decode button. https://builder.honeycode.aws/table/1-us-west-2%3A122162422134%3Atable%3A111c6cea-af2a-41bb-ba31-92759c730076%2F636dfdb5-232b-468f-890c-92de2ae1c87a For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page. UTF-8 Source character set. □ Decode each line separately (useful for when you have multiple entries). Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set). DECODE Decodes your data into the area below. https://builder.honeycode.aws/table/1-us-west-2:122162422134:table:111c6cea-af2a-41bb-ba31-92759c730076/636dfdb5-232b-468f-890c-92de2ae1c87a 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.
LambdaCodeBucketName	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"



Project name	Project ID	Cre
testpinpoint	9ddf77ec0e254157aeffa3a0fe1ce381	Feb
cfincrementaltest-deva	e30697c8060f49c28b3c0bd2d9094901	Feb

Once all fields completed, click “Next”.

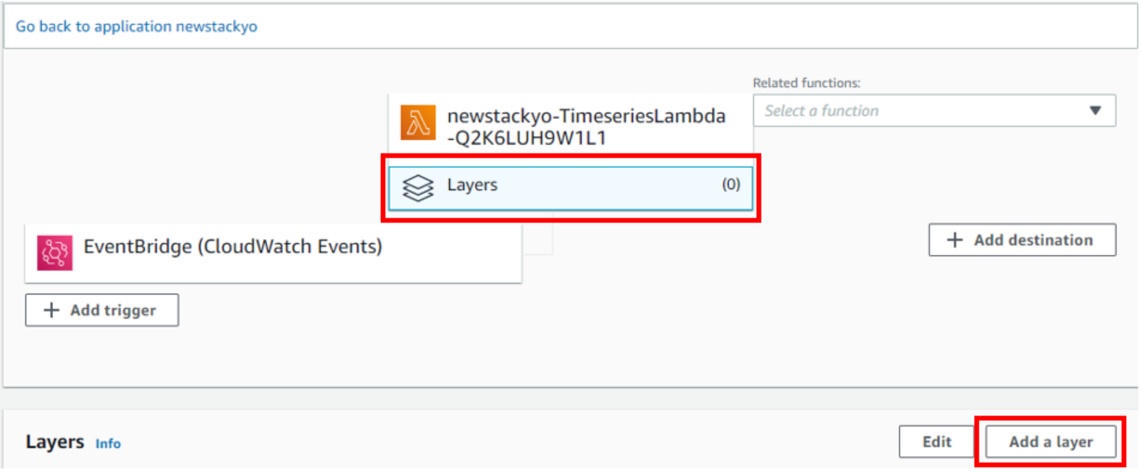
On the “Configure stack options” page, click “Next”

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

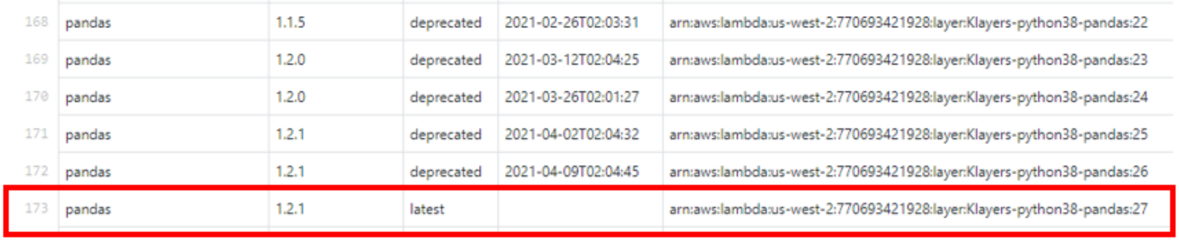
Step 5 – Create Lambda Layer

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”

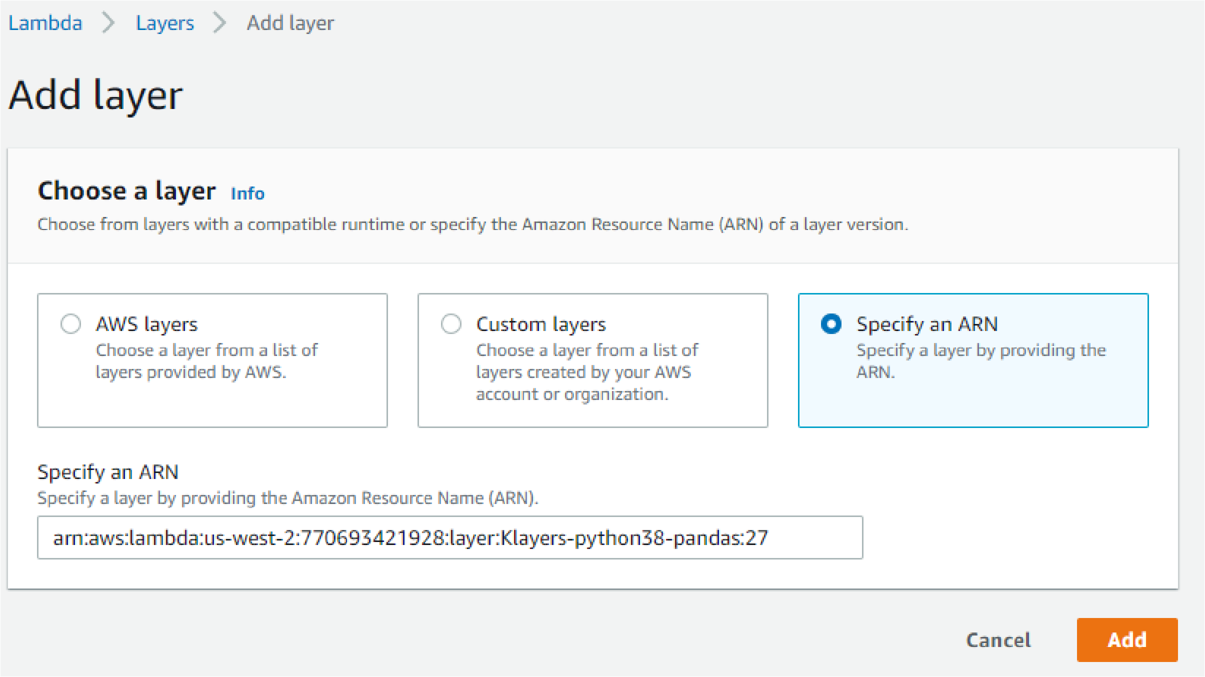

Go back to application newstackyo
Related functions:
Select a function
newstackyo-TimeseriesLambda -Q2K6LUH9W1L1
Layers (0)
EventBridge (CloudWatch Events)
+ Add destination
+ Add trigger
Layers Info
Edit
Add a layer

Depending the Region you have deployed the stack, visit this Github page, 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



pandas	1.1.5	deprecated	2021-02-26T02:03:31	arn:aws:lambda:us-west-2:770693421928:layer:Klayers-python38-pandas:22
pandas	1.2.0	deprecated	2021-03-12T02:04:25	arn:aws:lambda:us-west-2:770693421928:layer:Klayers-python38-pandas:23
pandas	1.2.0	deprecated	2021-03-26T02:01:27	arn:aws:lambda:us-west-2:770693421928:layer:Klayers-python38-pandas:24
pandas	1.2.1	deprecated	2021-04-02T02:04:32	arn:aws:lambda:us-west-2:770693421928:layer:Klayers-python38-pandas:25
pandas	1.2.1	deprecated	2021-04-09T02:04:45	arn:aws:lambda:us-west-2:770693421928:layer:Klayers-python38-pandas:26
pandas	1.2.1	latest		arn:aws:lambda:us-west-2:770693421928:layer:Klayers-python38-pandas:27

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



Add layer

Choose a layer Info

Choose from layers with a compatible runtime or specify the Amazon Resource Name (ARN) of a layer version.

○ AWS layers
Choose a layer from a list of layers provided by AWS.

○ Custom layers
Choose a layer from a list of layers created by your AWS account or organization.

● Specify an ARN
Specify a layer by providing the ARN.

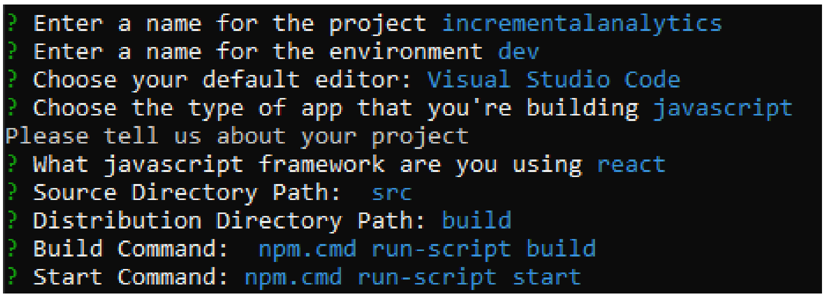
Specify an ARN
Specify a layer by providing the Amazon Resource Name (ARN).

```
arn:aws:lambda:us-west-2:770693421928:layer:Klayers-python38-pandas:27
```

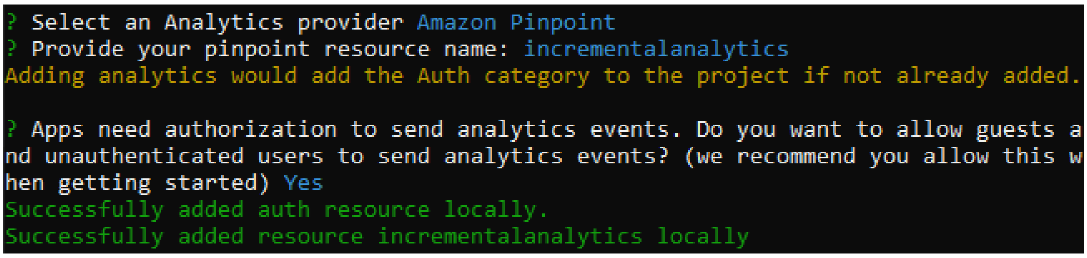
Steps to test the solution

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:

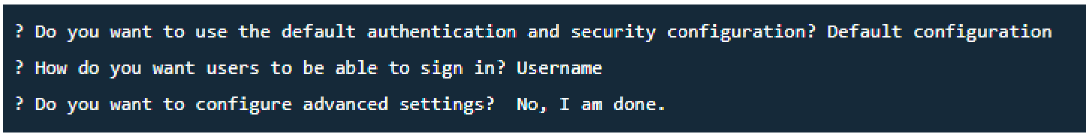
IMPORTANT: Once you complete the steps below, a new Pinpoint Project will be created in the Region of the AWS account you are using. For testing purposes, use the project created from Amplify to deploy and test the solution .

1) Complete the Amplify Tutorial – Prerequisites step in this page
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. 

```
? Enter a name for the project incrementalanalytics
? Enter a name for the environment dev
? Choose your default editor: Visual Studio Code
? Choose the type of app that you're building javascript
Please tell us about your project
? What javascript framework are you using react
? Source Directory Path:  src
? Distribution Directory Path: build
? Build Command:  npm.cmd run-script build
? Start Command: npm.cmd run-script start
```

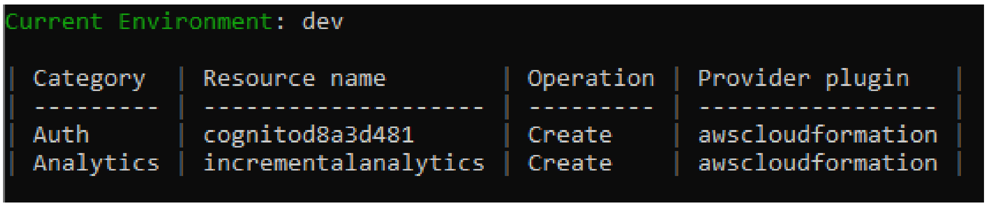
8) To the question "Do you want to use an AWS profile?" answer Yes and select the profile you would like to have this solution implemented
9) Once the above step is completed, type and execute:
   a. Amplify add analytics:
      i. 

```
? Select an Analytics provider Amazon Pinpoint
? Provide your pinpoint resource name: incrementalanalytics
Adding analytics would add the Auth category to the project if not already added.

? Apps need authorization to send analytics events. Do you want to allow guests a
nd unauthenticated users to send analytics events? (we recommend you allow this w
hen getting started) Yes
Successfully added auth resource locally.
Successfully added resource incrementalanalytics locally
```

   b. Amplify add auth:
      i. 

```
? Do you want to use the default authentication and security configuration? Default configuration
? How do you want users to be able to sign in? Username
? Do you want to configure advanced settings?  No, I am done.
```

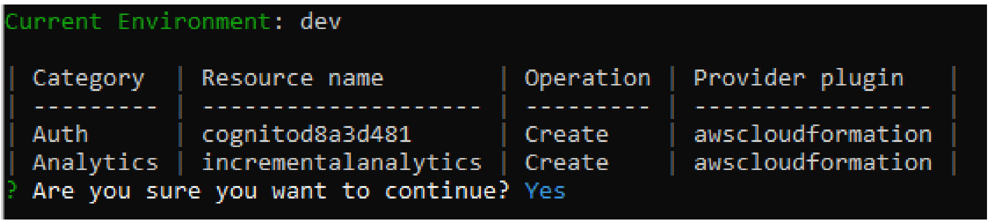
      ii. Note: If you get a message, which says that Auth is already configured for this project, then type and execute: *amplify update auth*
10) Type and execute: *amplify status*. You should see that there are 2 resources where column Operation = Create



```
Current Environment: dev

| Category  | Resource name        | Operation | Provider plugin   |
| --------- | -------------------- | --------- | ----------------- |
| Auth      | cognitod8a3d481      | Create    | awscloudformation |
| Analytics | incrementalanalytics | Create    | awscloudformation |
```

a.

11) Type and execute: *amplify push*. In the question “Are you sure you want to continue?” type Y and press Enter



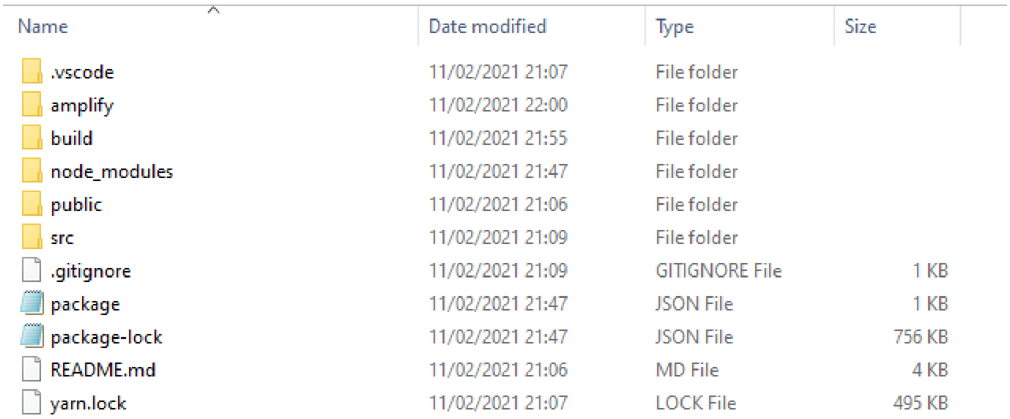
```
Current Environment: dev

| Category  | Resource name        | Operation | Provider plugin   |
| --------- | -------------------- | --------- | ----------------- |
| Auth      | cognitod8a3d481      | Create    | awscloudformation |
| Analytics | incrementalanalytics | Create    | awscloudformation |
? Are you sure you want to continue? Yes
```

a.

12) Type and execute: *npm install aws-amplify @aws-amplify/ui-react*

13) Type and execute: *npm install react-bootstrap bootstrap*

14) Exit the comand prompt or Power Shell and go to the project folder, which should have the below contents



Name	Date modified	Type	Size
.vscode	11/02/2021 21:07	File folder	
amplify	11/02/2021 22:00	File folder	
build	11/02/2021 21:55	File folder	
node_modules	11/02/2021 21:47	File folder	
public	11/02/2021 21:06	File folder	
src	11/02/2021 21:09	File folder	
.gitignore	11/02/2021 21:09	GITIGNORE File	1 KB
package	11/02/2021 21:47	JSON File	1 KB
package-lock	11/02/2021 21:47	JSON File	756 KB
README.md	11/02/2021 21:06	MD File	4 KB
yarn.lock	11/02/2021 21:07	LOCK File	495 KB

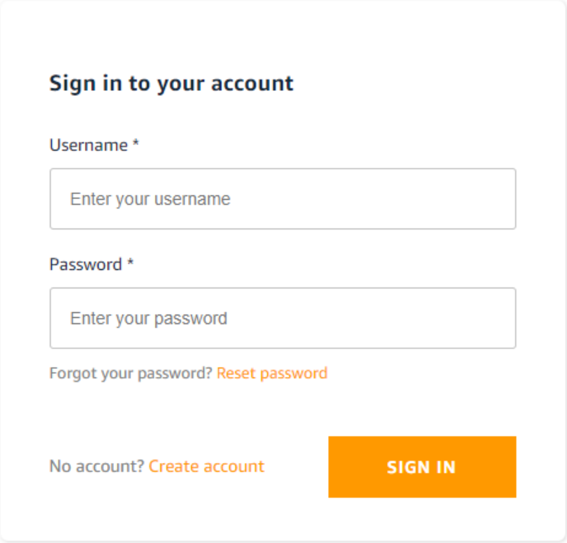
a.

15) Open the “scr” folder, paste & replace the App.js and index.css files with these two

16) Go back and open the “public” folder, paste & replace the index.html file with this one

17) Open the comand prompt or Power Shell, navigate to the app folder “Incremental_Analytics_Pinpoint”, type and execute: npm start

18) The above should open your default browser localhost:3000 and display the below:



Sign in to your account

Username *

Enter your username

Password *

Enter your password

Forgot your password? Reset password

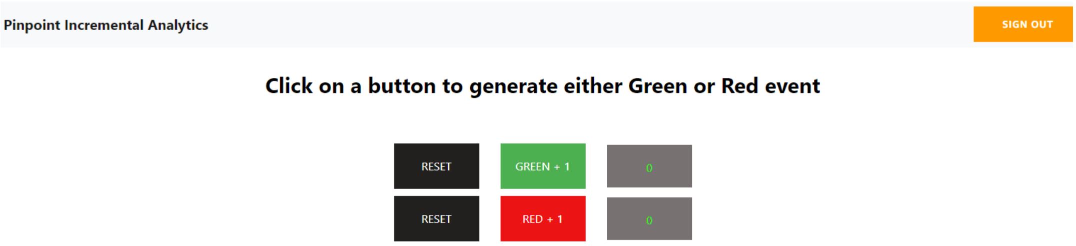
No account? Create account

a.

19) Click on Create account, fill all information required and click on “Create Account”

20) You should receive the activation code on the email you used to sign-up, enter it and click “Activate”

21) Once you are logged in, you should see the screen below:



Pinpoint Incremental Analytics

Click on a button to generate either Green or Red event

a.

22) When clicking either “Green” or “Red” a Pinpoint event with name greenbutton and redbutton respectively will fire

   a. **Note:**

      i. Both events have a metric value of 1

      ii. 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

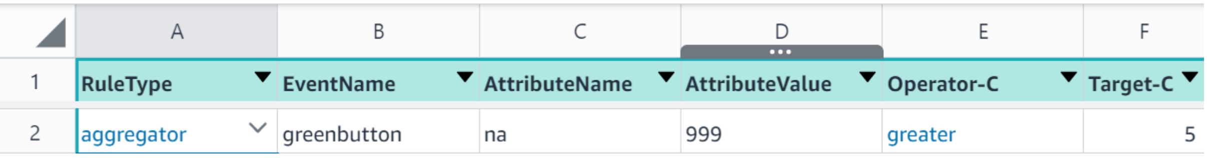
Description: Move customers to a specific segment when the count or metric sum of an event equals or is greater than a specific value

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

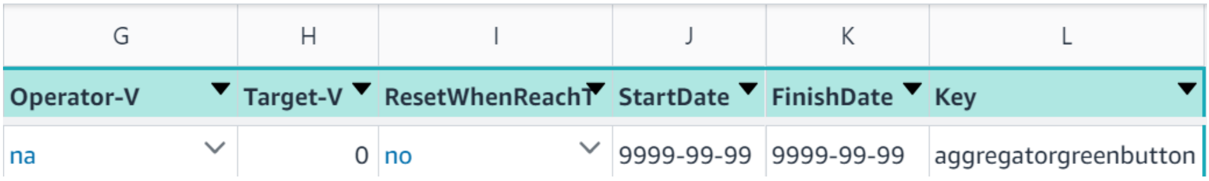
Execution:

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
    a. Select RuleType = aggregator
    b. 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
    c. 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
    d. If you have typed an event attribute then type the attribute value e.g. homepage otherwise type 999
    e. Select Operator-C = greater, Target-C = 5
    f. Select Operator-V = na, Target-V = 999
    g. Select ResetWhenReachTarget = no
    h. For StartDate & FinishDate type 9999-99-99
    i. The column Key will populate automatically

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

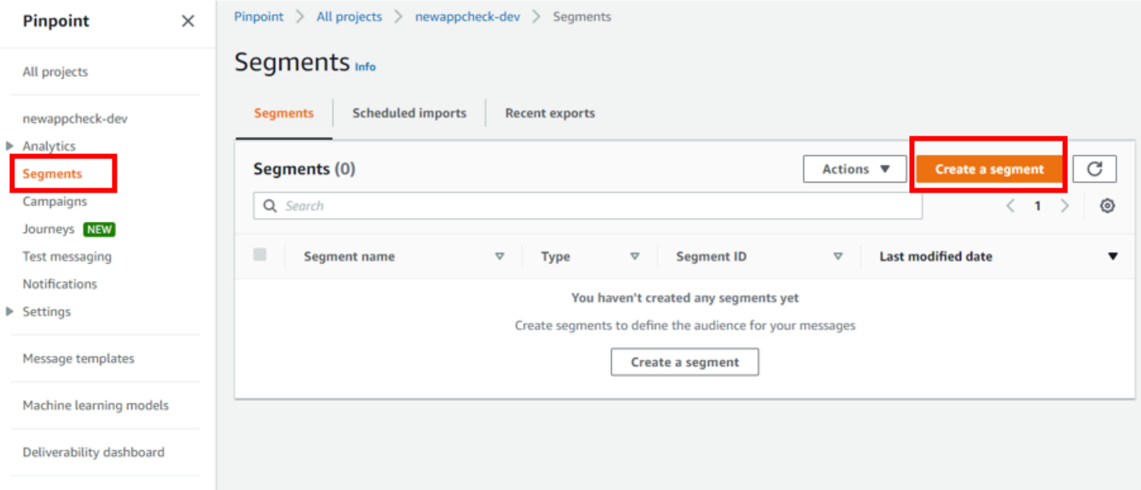


	A	B	C	D	E	F
1	RuleType	EventName	AttributeName	AttributeValue	Operator-C	Target-C
2	aggregator	greenbutton	na	999	greater	5



G	H	I	J	K	L
Operator-V	Target-V	ResetWhenReachT	StartDate	FinishDate	Key
na	0	no	9999-99-99	9999-99-99	aggregatorgreenbutton

3) Go to your Pinpoint project, click on Segments and then Create a segment

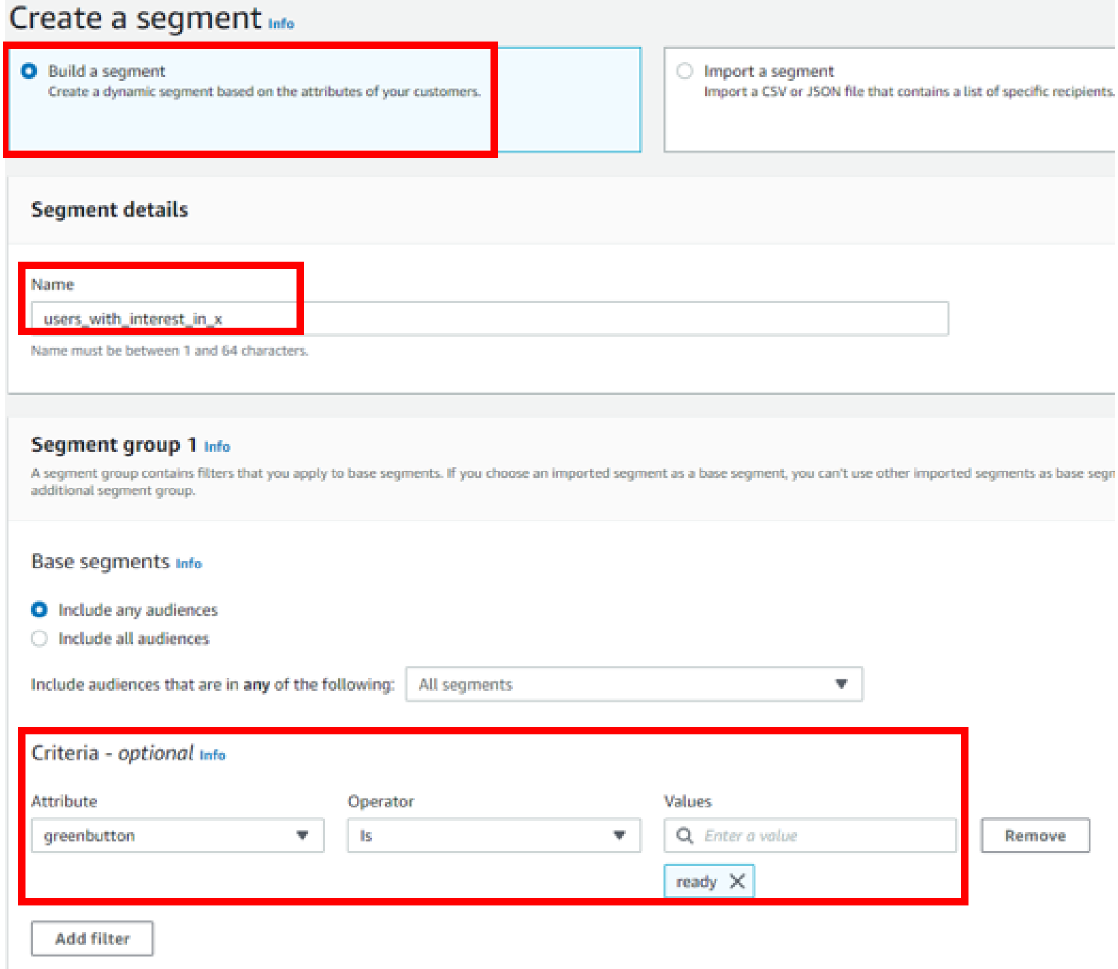


You haven't created any segments yet

Create segments to define the audience for your messages

a.

4) 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.


Create a segment Info

● Build a segment
Create a dynamic segment based on the attributes of your customers.

○ Import a segment
Import a CSV or JSON file that contains a list of specific recipients.

Segment details

Name: users_with_interest_in_x

Name must be between 1 and 64 characters.

Segment group 1 Info

A segment group contains filters that you apply to base segments. If you choose an imported segment as a base segment, you can't use other imported segments as base segm additional segment group.

Base segments Info

● Include any audiences
○ Include all audiences

Include audiences that are in **any** of the following: All segments

Criteria - *optional* Info

Attribute: greenbutton

Operator: Is

Values: ready

a.

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

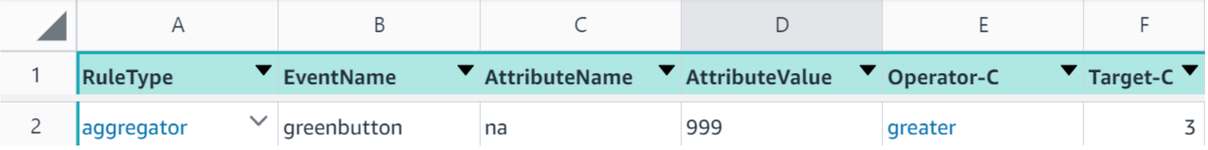
Description: Have the ability to trigger an event when a rule is met (event count or metric sum) and reset the count / sum

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

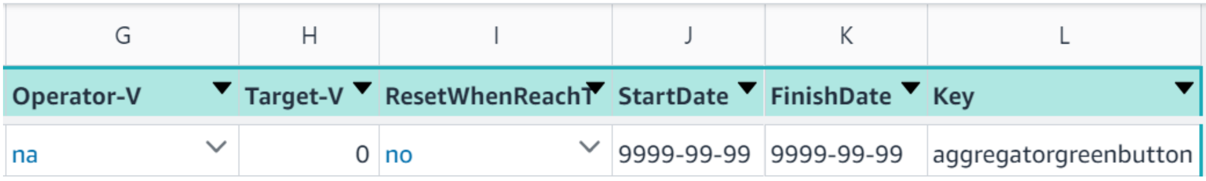
Execution:

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
    a. Select RuleType = aggregator
    b. 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
    c. 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
    d. If you have typed an event attribute then type the attribute value e.g. homepage otherwise type 999
    e. Select Operator-C = greater, Target-C = 3
    f. Select Operator-V = na, Target-V = 999
    g. Select ResetWhenReachTarget = yes
    h. For StartDate & FinishDate type 9999-99-99
    i. The column Key will populate automatically

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

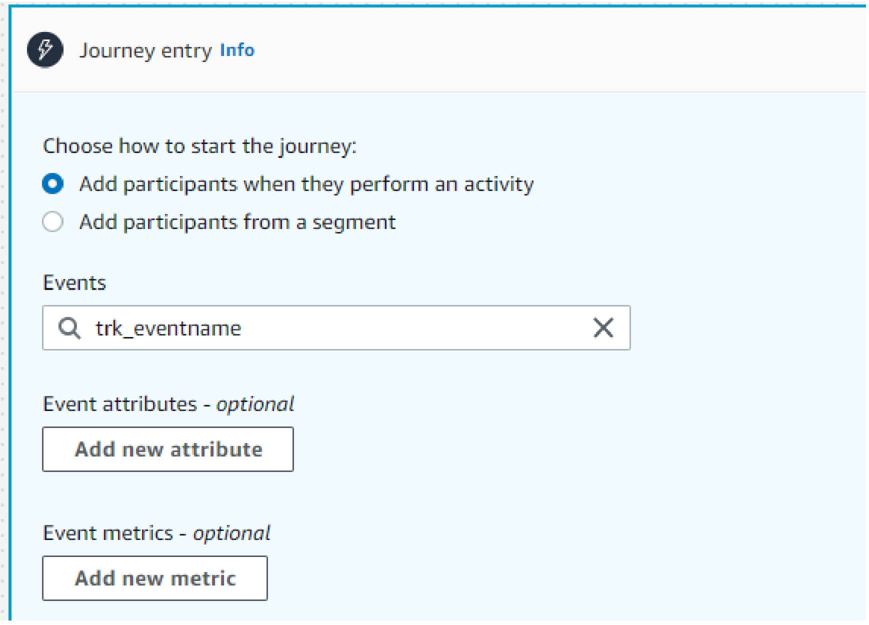


	A	B	C	D	E	F
1	RuleType	EventName	AttributeName	AttributeValue	Operator-C	Target-C
2	aggregator	greenbutton	na	999	greater	3



G	H	I	J	K	L
Operator-V	Target-V	ResetWhenReachT	StartDate	FinishDate	Key
na	0	no	9999-99-99	9999-99-99	aggregatorgreenbutton

3) Go to the Pinpoint project and create a Journey
4) 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



Journey entry Info

Choose how to start the journey:

● Add participants when they perform an activity

○ Add participants from a segment

Events

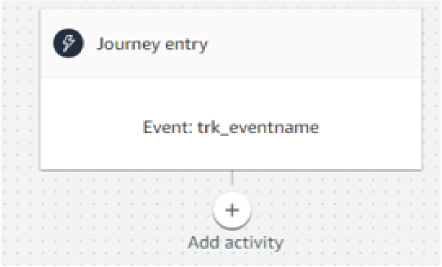
trk_eventname

Event attributes - *optional*

Event metrics - *optional*

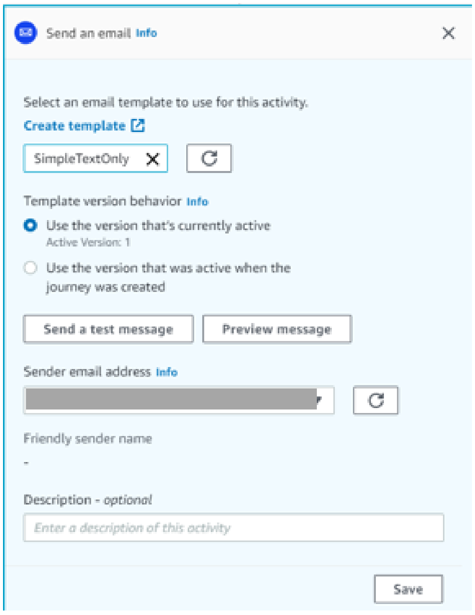
a.

5) Click on Add activity and select Send email. You will need to have Email channel enabled for this project and a ready email template



Journey entry

Event: trk_eventname



Send an email Info

Select an email template to use for this activity.

Create template

SimpleTextOnly

Template version behavior Info

● Use the version that's currently active
Active Version: 1

○ Use the version that was active when the journey was created

Sender email address Info

[redacted]

Friendly sender name

-

Description - *optional*

Enter a description of this activity

a.

6) Click "Save"

7) 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"

8) 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

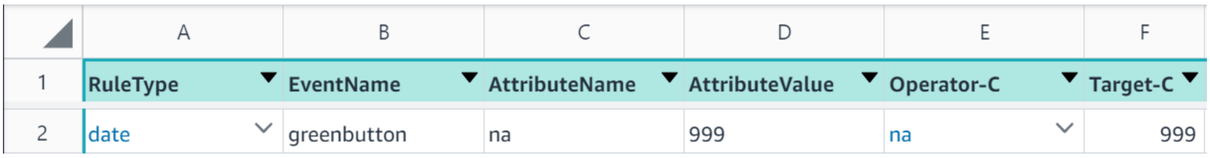
Example: Users who have spent X $500 between 2021-01-01 and 2021-02-01 should move to a dynamic segment “Users with interest in X”

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.

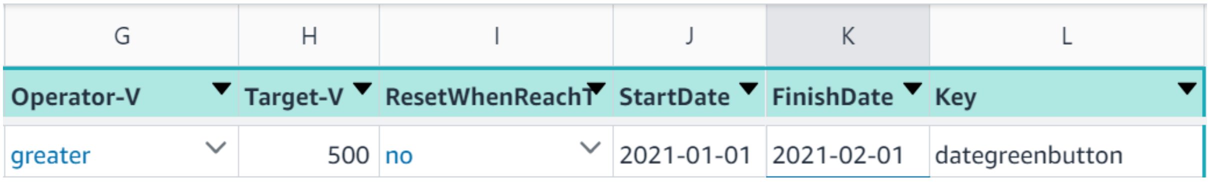
Execution:

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
    a. Select RuleType = date
    b. 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
    c. 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
    d. If you have typed an event attribute then type the attribute value e.g. homepage otherwise type 999
    e. Select Operator-C = na, Target-C = 999
    f. Select Operator-V = greater, Target-V = 500
    g. Select ResetWhenReachTarget = no
    h. StartDate = 2021-01-01
    i. FinishDate = 2021-02-01
    j. The column Key will populate automatically

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

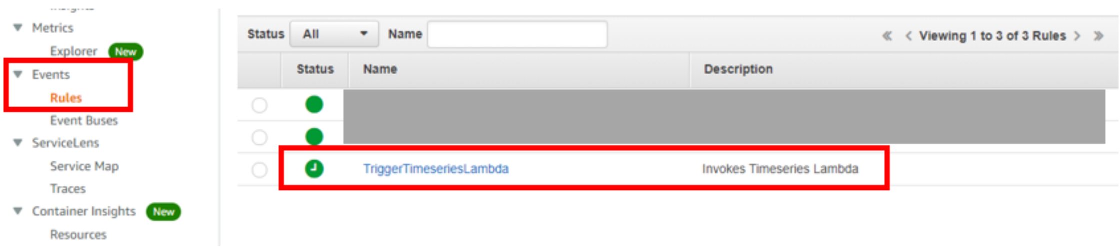


	A	B	C	D	E	F
1	RuleType	EventName	AttributeName	AttributeValue	Operator-C	Target-C
2	date	greenbutton	na	999	na	999



G	H	I	J	K	L
Operator-V	Target-V	ResetWhenReachT	StartDate	FinishDate	Key
greater	500	no	2021-01-01	2021-02-01	dategreenbutton

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



- Metrics
  - Explorer New
- Events
  - Rules
  - Event Buses
- ServiceLens
  - Service Map
  - Traces
- Container Insights New
  - Resources

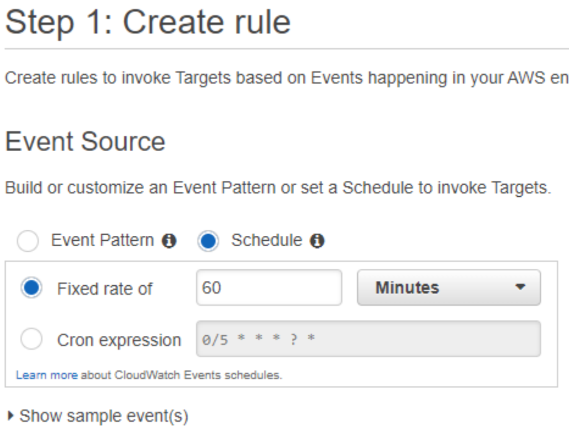
Status All Name

Viewing 1 to 3 of 3 Rules

	Status	Name	Description
○	●	[redacted]	[redacted]
○	●	[redacted]	[redacted]
○	●	TriggerTimeseriesLambda	Invokes Timeseries Lambda

a.

4) 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



Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS envi

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

○ Event Pattern ● Schedule

● Fixed rate of 60 Minutes

○ Cron expression 0/5 * * * ? *

Learn more about CloudWatch Events schedules.

Show sample event(s)

a.