

Visualizing Call Recordings with Amazon Quicksight

Call recordings have been processed with [Amazon Transcribe](#) and [Amazon Comprehend](#) and the information obtained from them has been dumped into a JSON document.

The current Document structure is as follows:

```
{  
  "ConversationAnalytics": {  
    "ConversationTime": "string",  
    "ProcessTime": "string",  
    "Duration": "float",  
    "LanguageCode": "string",  
    "EntityRecognizerName": "string",  
    "SpeakerLabels": [  
      {  
        "Speaker": "string",  
        "DisplayText": "string"  
      }  
    ],  
    "SentimentTrends": [  
      {  
        "Speaker": "string",  
        "AverageSentiment": "float",  
        "SentimentChange": "float"  
      }  
    ],  
    "CustomEntities": [  
      {  
        "Name": "string",  
        "Count": "integer",  
        "Values": [ "string" ]  
      }  
    ],  
    "SourceInformation": [  
      {  
        "TranscribeJobInfo": {  
          "TranscriptionJobName": "string",  
          "CompletionTime": "string",  
          "VocabularyName": "string",  
          "MediaFormat": "string",  
          "MediaSampleRateHertz": "integer",  
          "MediaFileUri": "string",  
          "MediaOriginalUri": "string",  
          "ChannelIdentification": "boolean",  
          "AverageAccuracy": "float"  
        }  
      }  
    ]  
  }  
}
```

```

        ],
    },
    "SpeechSegments": [
    {
        "SegmentStartTime": "float",
        "SegmentEndTime": "float",
        "SegmentSpeaker": "string",
        "OriginalText": "string",
        "DisplayText": "string",
        "TextEdited": "boolean",
        "SentimentIsPositive": "boolean" __,
        "SentimentIsNegative": "boolean",
        "SentimentScore": "float",
        "BaseSentimentScores": {
            "Positive": "float",
            "Negative": "float",
            "Neutral": "float",
            "Mixed": "float"
        },
        "EntitiesDetected": [
        {
            "Type": "string",
            "Text": "string",
            "BeginOffset": "integer",
            "EndOffset": "integer",
            "Score": "float"
        }
    ],
        "WordConfidence": [
        {
            "Text": "string",
            "Confidence": "float",
            "StartTime": "float",
            "EndTime": "float",
            "Entity": "string"
        }
    ]
}
]
}

```

In order to aggregate this data and consume it we will use [Amazon Athena](#).

The first step is to create a table definition in [AWS Glue](#) Data Catalog, which will allow us to define where the Json files will be stored. At a high level we will define the location of the files and run a Crawler

Adding a Crawler to create a database

On the AWS Glue console, under Data Catalog, click on Crawlers, and then on the “Add Crawler” button.

Crawlers							
Search documents and file names for text							
progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.							
User preferences							
Data catalog	Add crawler	Run crawler	Action	<input type="text"/> Filter by tags and attributes			Showing: 1 - 1 < > Edit Logs
Databases							
Tables							
Connections							
Crawlers							
Classifiers							

2. Enter a name and click Next

Add information about your crawler

Crawler name

► Tags, description, security configuration, and classifiers (optional)

Next

3. Select Data Stores and click Next

Specify crawler source type

Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores.

Crawler source type

Data stores
 Existing catalog tables

Back **Next**

4. For the data store we will choose S3, and we will specify the path where the transcription json documents are stored on the "Include Path" field, then click next

Add a data store

Choose a data store

S3



Connection

Select a connection



Optional include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

[Add connection](#)

Crawl data in

- Specified path in my account
- Specified path in another account

Include path

s3://folder-with-json-docs



All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

▶ [Exclude patterns \(optional\)](#)

5. On add another data store, select No

Add another data store

Yes

No

[Back](#)

[Next](#)

6. For the IAM role we will create a new IAM role, that will include a policy that allows read access to the S3 target we defined before:

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

- Update a policy in an IAM role
- Choose an existing IAM role
- Create an IAM role

IAM role i

AWSGlueServiceRole- TrascriptionsCrawler

To create an IAM role, you must have **CreateRole**, **CreatePolicy**, and **AttachRolePolicy** permissions.

Create an IAM role named "**AWSGlueServiceRole**-rolename" and attach the AWS managed policy, **AWSGlueServiceRole**, plus an inline policy that allows read access to:

- s3:// tests

You can also create an IAM role on the [IAM console](#).

[Back](#)

[Next](#)

7. Define a schedule for your crawler, in this case we will select on demand:

Create a schedule for this crawler

Frequency

Run on demand



[Back](#)

[Next](#)

8. On the "Configure the Crawler's output" screen, click on Add database, and enter a name.

Add database

Database name

- ▶ Description and location (optional)

9. Select the new database you created and click next.

Configure the crawler's output

Database i

Prefix added to tables (optional) i

- ▶ Grouping behavior for S3 data (optional)

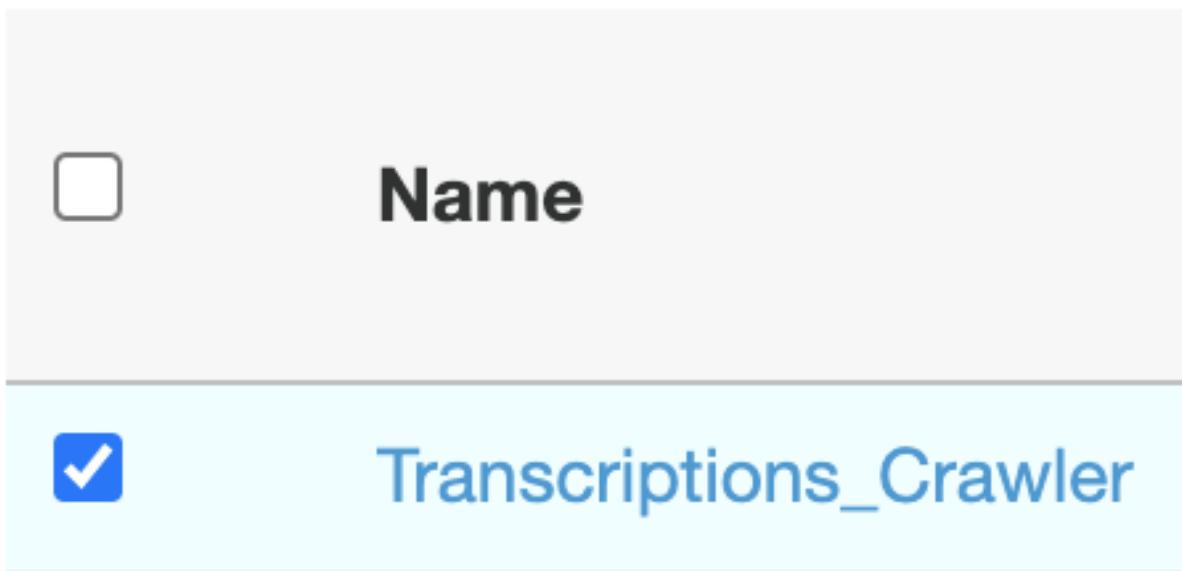
- ▶ Configuration options (optional)

10. Click on Finish.

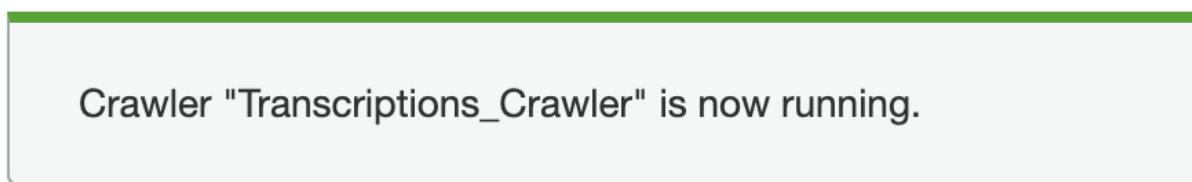
11. On the crawlers list, select your newly created crawler and click on “Run Crawler”

Add crawler

Run crawler



12. Wait for the crawler to finish



13. You can check if a new table has been created based on your transcriptions under Databases -> Tables

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Add tables	Action	Database : transcriptions_database	Filter or search for tables...	Save view	Showing: 1 - 1	<	>	?	
Name	Database	Location	Classification	Last updated	Deprecated				
<input type="checkbox"/> parsedfiles	transcriptions_database	s3://jpb-transcribe-tests/p... json		10 September 2020 2:18 ...					

Creating queries and views with Athena

Transcription data will be made available to Amazon Quicksight through an Amazon Athena query.

We will generate 3 views in order to have different data dimensions available.

On the Amazon Athena query editor, we will select AwsDataCatalog as our Data source and we will choose the database we just created:

The screenshot shows the Amazon Athena Query editor interface. At the top, there are three tabs: "Athena", "Query editor" (which is currently selected), and "Saved queries". Below the tabs, there are two main configuration sections. The first section is labeled "Data source" and contains a dropdown menu set to "AwsDataCatalog". To the right of this section is a blue "Connect data source" button with a refresh icon. The second section is labeled "Database" and contains a dropdown menu set to "transcriptions_database". Both dropdown menus have a downward arrow icon indicating they are expandable.

If you haven't used Amazon Athena before you need to [set up a query result location on S3](#).

1. Create an S3 bucket for storing your results.
2. Click on the “set up a query result location in Amazon S3” link, and enter the name of the bucket you will use to store your results and click save.

Before you run your first query, you need to [set up a query result location in Amazon S3](#). Learn more

Settings

X

Settings apply by default to all new queries. [Learn more](#)

Workgroup: **primary**

Query result location

s3:// -athena-query-results/

i

Example: s3://query-results-bucket/folder/

Encrypt query results

i

Autocomplete

i

Cancel

Save

We are going to generate 3 views that will be used by quick sight.

1st View – Call durations and Languages

```
CREATE OR REPLACE VIEW call_duration_and_language AS
SELECT
  "regexp_extract"("$path", '[^/]+$') "filename"
, "conversationanalytics"."languagecode"
, "conversationanalytics"."speakerlabels"[1].displaytext "speaker_0"
, "conversationanalytics"."duration"
, "conversationanalytics"."conversationtime"
, "substr"("conversationanalytics"."conversationtime", 1, 19) "formatedconversationtime"
FROM
  "Parsedfiles"
```

```
1 CREATE OR REPLACE VIEW call_duration_and_language AS
2 SELECT
3   "regexp_extract"("$path", '[^/]+$') "filename"
4 , "conversationanalytics"."languagecode"
5 , "conversationanalytics"."speakerlabels"[1].displaytext "speaker_0"
6 , "conversationanalytics"."duration"
7 , "conversationanalytics"."conversationtime"
8 , "substr"("conversationanalytics"."conversationtime", 1, 19) "formatedconversationtime"
9 FROM
10  "Parsedfiles"
```

2nd View – Conversation Segments

```
CREATE OR REPLACE VIEW conversation_segments AS
SELECT
  "regexp_extract"("$path", '[^/]+$') "filename"
, "conversationanalytics"."languagecode"
, "conversationanalytics"."speakerlabels"[1].displaytext "speaker_0"
, "conversationanalytics"."conversationtime"
```

```

, "conversationanalytics"."duration"
, "substr("conversationanalytics"."conversationtime", 1, 19) "formatedconversationtime"
, "segment"."segmentstarttime"
, "segment"."segmentendtime"
, "segment"."segmentspeaker"
, "segment"."sentimentispositive"
, "segment"."sentimentisnegative"
, "segment"."sentimentscore"
, "segment"."displaytext"
, "segment"."textedited"
, "segment"."entitiesdetected"
, (CASE WHEN ((conversationanalytics"."speakerlabels"[1]."displaytext" = 'Agent') AND
FROM
(parsedfiles
CROSS JOIN UNNEST("speechsegments") t (segment))

```

```

1 CREATE OR REPLACE VIEW conversation_segments AS
2 SELECT
3   "regexp_extract"("$path", '[^/]+$') "filename"
4   , "conversationanalytics"."languagecode"
5   , "conversationanalytics"."speakerlabels"[1]."displaytext" "speaker_0"
6   , "conversationanalytics"."conversationtime"
7   , "substr("conversationanalytics"."conversationtime", 1, 19) "formatedconversationtime"
8   , "segment"."segmentstarttime"
9   , "segment"."segmentendtime"
10  , "segment"."segmentspeaker"
11  , "segment"."sentimentispositive"
12  , "segment"."sentimentisnegative"
13  , "segment"."sentimentscore"
14  , "segment"."displaytext"
15  , "segment"."textedited"
16  , "segment"."entitiesdetected"
17  , "segment"."entitiesdetected"
18 , (CASE WHEN ((conversationanalytics"."speakerlabels"[1]."displaytext" = 'Agent') AND ("segment"."segmentspeaker" =
19 "spk_0")) THEN 'Agent' WHEN ((conversationanalytics"."speakerlabels"[1]."displaytext" = 'Agent') AND
("segment"."segmentspeaker" = 'spk_1')) THEN 'Caller' END) "Speaker"
20 FROM
21 (parsedfiles
22 CROSS JOIN UNNEST("speechsegments") t (segment))

```

3rd View – Extracted Entities

```

CREATE OR REPLACE VIEW extracted_entities AS

SELECT
"filename"

, "languagecode"

```

```
, "duration"
, "conversationtime"
, "formatedconversationtime"
, "segmentstarttime"
, "segmentendtime"
, "segmentspeaker"
, "speaker"
, "sentimentispositive"
, "sentimentisnegative"
, "sentimentscore"
, "displaytext"
, "textedited"
, "entity"."beginoffset"
, "entity"."endoffset"
, "entity"."score"
, "entity"."text"
, "entity"."type"
FROM
(conversation_segments
CROSS JOIN UNNEST("entitiesdetected") t (entity))
```

```

1 CREATE OR REPLACE VIEW extracted_entities AS
2 SELECT
3   "filename"
4 , "languagecode"
5 , "duration"
6 , "conversationontime"
7 , "formatedconversationtime"
8 , "segmentstarttime"
9 , "segmentendtime"
10 , "segmentspeaker"
11 , "speaker"
12 , "sentimentispositive"
13 , "sentimentisnegative"
14 , "sentimentscore"
15 , "displaytext"
16 , "textedited"
17 , "entity"."beginoffset"
18 , "entity"."endoffset"
19 , "entity"."score"
20 , "entity"."text"
21 , "entity"."type"
22 FROM
23   (conversation_segments
24 CROSS JOIN UNNEST("entitiesdetected") t (entity))
25

```

Creating the Amazon QuickSight Dataset

Now that we have our views created, we will create 3 datasets on Quicksight that will allow us to explore the 3 data dimensions we have available.

If you haven't signed up for Amazon Quicksight, follow the process presented when you visit the console. Also make sure you enable Amazon Athena Permissions (they are available by default).

Also we need to make sure that Amazon Quicksight has access to the S3 bucket where your transcriptions are.

1. Click on Manage QuickSight on the top right corner of your screen:



2. Click on "Security & Permissions":

Manage users

Your subscriptions

SPICE capacity

Account settings

Security & permissions

Mobile settings

3. Click on “Add or remove”:

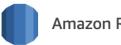
Security & permissions

QuickSight can control access to AWS resources for the entire account in addition to individual users and groups

QuickSight access to AWS services



Amazon Redshift



Amazon RDS



IAM



Amazon S3



Amazon Athena

By configuring access to AWS services, QuickSight can access the data in those services. Access by users and groups can be controlled through the options below.

[Add or remove](#)

4. On “Amazon S3” click on “details” and then click on “Select S3 Buckets”:

Amazon S3

Enables QuickSight to auto-discover your Amazon S3 buckets [Details](#)

Amazon S3

Enables QuickSight to auto-discover your Amazon S3 buckets [Hide](#)

Select S3 buckets 1 buckets selected

5. Select the S3 bucket where your transcriptions are stored and click on “Finish”:

Select Amazon S3 buckets

X

S3 Buckets Linked To QuickSight Account

S3 Buckets You Can Access Across AWS

Select the buckets that you want QuickSight to be able to access.

Selected buckets have read only permissions by default. However, you must give write permissions for Athena Workgroup feature.

Select all

S3 Bucket	Write permission for Athena Workgroup
<input type="checkbox"/>	
<input checked="" type="checkbox"/> -transcriptions	<input type="checkbox"/>
<input type="checkbox"/>	
<input type="checkbox"/>	

Cancel

Finish

After the permissions are granted we can move forward to create the Datasets.

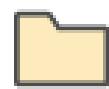
1. On Amazon Quicksight's, on the left menu, click on Datasets



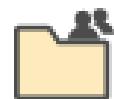
★ Favorites



Recent



My folders



Shared folders



Dashboards



Analyses



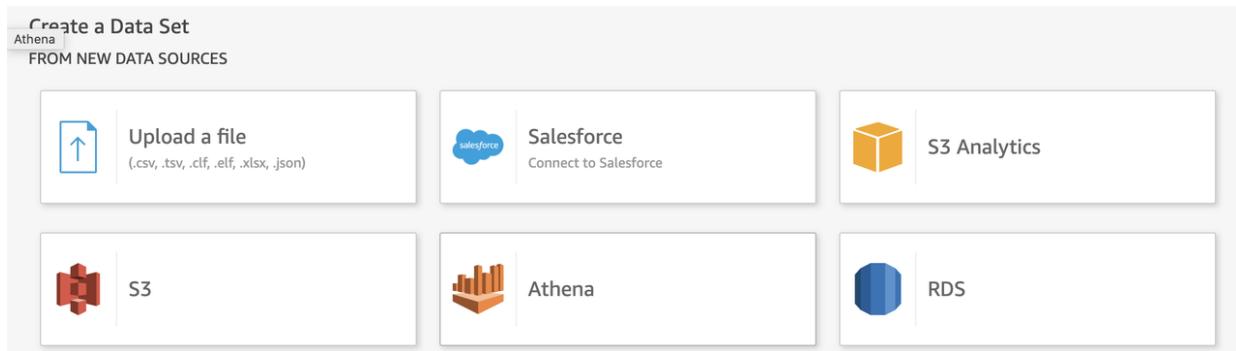
Datasets

2. Click on the “New Dataset” button on the top right

Datasets

New dataset

3. Click on “Athena”



4. We will create one datasource for each view have, we will start with call_duration_and_language:

New Athena data source

X

Data source name

call_duration_and_language

Athena workgroup

[primary]

▼

Validate connection

SSL is enabled

Create data source

5. We will select the “transcriptions_database” and select “call_duration_and_language”, then we will click on “Edit/Preview data”

Choose your table

X

call_duration_and_language

Database: contain sets of tables.

transcriptions_database

▼

Tables: contain the data you can visualize.

- call_duration_and_language
- conversation_segments
- extracted_entities
- parsedfiles

[Edit/Preview data](#)

[Use custom SQL](#)

[Select](#)

6. On the “Edit/Preview data” screen, select [SPICE](#) for query mode.
7. Change the data type for the field “formatedconversationtime” to date and use the format “yyyy-mm-dd HH:mm:ss” and click on Update.

Edit date format

X

Provide the **date format** which represents this field. Formats are case sensitive.

For example, dd/MM/yyyy HH:mm:ss translates to 31/08/2017 23:59:59

Your date format is valid. See below for sample data output.

Source data

2019-08-13 13:51:55

2020-03-09 10:16:53

2020-03-11 09:39:56

Recognized as

2019-01-13T13:51:55.000Z

2020-01-09T10:16:53.000Z

2020-01-11T09:39:56.000Z

CloseValidateUpdate

8. For the field duration, select the data type decimal

Query mode

SPICE Direct query

Fields All fields selected

Add calculated field

Search fields

Calculated fields

No calculated fields.

Select All | None

- filename
- languagecode
- speaker_0

Type

- # Int
- # Decimal
- Date
- Country
- State
- County
- City
- Postcode
- Latitude
- Longitude

Add to a hierarchy

Exclude field

Rename field

Add calculation

Change data type >

9. Click on “Save”



Next we are going to create the data source for the conversation_segments view:

1. We will create a new data source, on Data Sets click on create Data set, select Athena and name it conversation_segments:

New Athena data source

X

Data source name

conversation_segments

Athena workgroup

[primary]

▼

Validate connection

SSL is enabled

Create data source

2. Select under transcriptions_database “conversation_segments” and click on Edit/Preview data:

Choose your table

X

conversation_segments

Database: contain sets of tables.

transcriptions_database

▼

Tables: contain the data you can visualize.

- call_duration_and_language
- conversation_segments
- extracted_entities
- parsedfiles

Edit/Preview data

Use custom SQL

Select

3. Change the data type for the field “formatedconversationtime” to date and use the format “yyyy-mm-dd HH:mm:ss” and click on Update
1009.5MB of remaining

Fields All fields selected

Field	Type
languagecode	Int
speaker_0	Date
conversationtime	Date
duration	Decimal
formatedconversationtime	Date
segmentstarttime	Country
segmentendtime	State
segmentspeaker	County

The screenshot shows a list of fields in a database or data management interface. A context menu is open over the 'duration' field, which is currently set to 'Decimal'. The menu options include: Add to a hierarchy, Exclude field, Rename field, Add calculation, and Change data type >. The 'Change data type >' option is highlighted with a blue background, indicating it is the next step to be taken.

4. Change the data type for the field “duration” to Decimal

Query mode

SPICE Direct query

Fields All fields selected

- `languagename`
- `speaker_0`
- `conversationtime`
- `duration`
- `formatedconversationtime`
- `# segmentstarttime`
- `# segmentendtime`

Data A

Int

Decimal

Date

Country

State

County

City

Postcode

Latitude

Longitude

Add to a hierarchy

Exclude field

Rename field

Add calculation

Change data type >

5. We will create a calculated field called calc_CallSentiment, for this click on Add calculated field.

Query mode

SPICE Direct query

Fields All fields selected



Add calculated field

6. Enter the name and the content of the formula:

```
ifelse(({sentimentispositive} =1),"Positive",({sentimentisnegative}=1),"Negative","Neu
```

The screenshot shows a 'calculated field' editor with the title 'calc_CallSentiment'. The code area contains the following Python-like pseudocode:

```
1 ifelse({{sentimentispositive} =1}, "Positive", {{sentimentisnegative}=1}, "Negative", "Neutral")
```

On the right, there's a 'Fields' section with a search bar labeled 'Search fields'.

7. Click on Save

A confirmation dialog box titled 'conversation_segments' has three buttons: 'Save & visualize', 'Save', and 'Cancel'.

Finally we are going to create the view for the entities.

1. We will create a new data source, on Data Sets click on create Data set, select Athena and name it extracted_entities:

New Athena data source X

Data source name

extracted_entities

Athena workgroup

[primary] ▼

[Validate connection](#)

SSL is enabled

[Create data source](#)

2. Select under transcriptions_database “extracted_entities” and click on Edit/Preview data:

Choose your table

X

extracted_entities

Database: contain sets of tables.

transcriptions_database



Tables: contain the data you can visualize.

- call_duration_and_language
- conversation_segments
- extracted_entities
- parsedfiles

Edit/Preview data

Use custom SQL

Select

3. Change the data type for the field “formatedconversationtime” to date and use the format “yyyy-mm-dd HH:mm:ss” and click on Update

Query mode

SPICE Direct query

Fields All fields selected

Search fields

Calculated fields

calc_CallSentiment

Select All | None

filename

languagancode

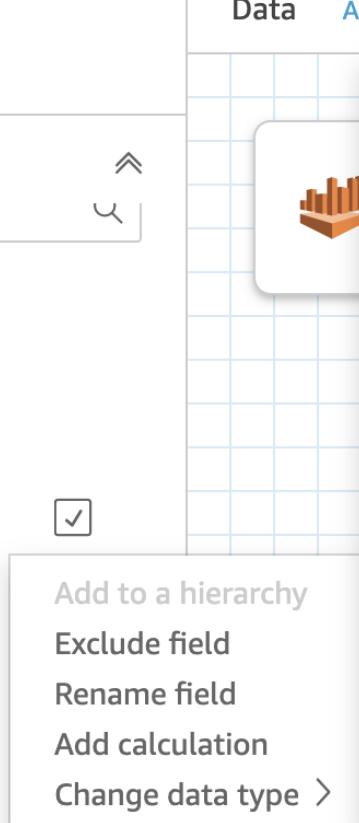
duration

conversationtime

Data Add data

Type

Int
Decimal
Date
Country
State
County
City
Postcode
Latitude
Longitude



A context menu is open over the 'duration' field. It contains the following options: 'Add to a hierarchy' (with a checked checkbox), 'Exclude field', 'Rename field', 'Add calculation', and 'Change data type >'. The 'Change data type' option is highlighted with a blue border.

4. Change the data type for the field "duration" to Decimal

Query mode

SPICE Direct query

Fields All fields selected

Add calculated field

Search fields

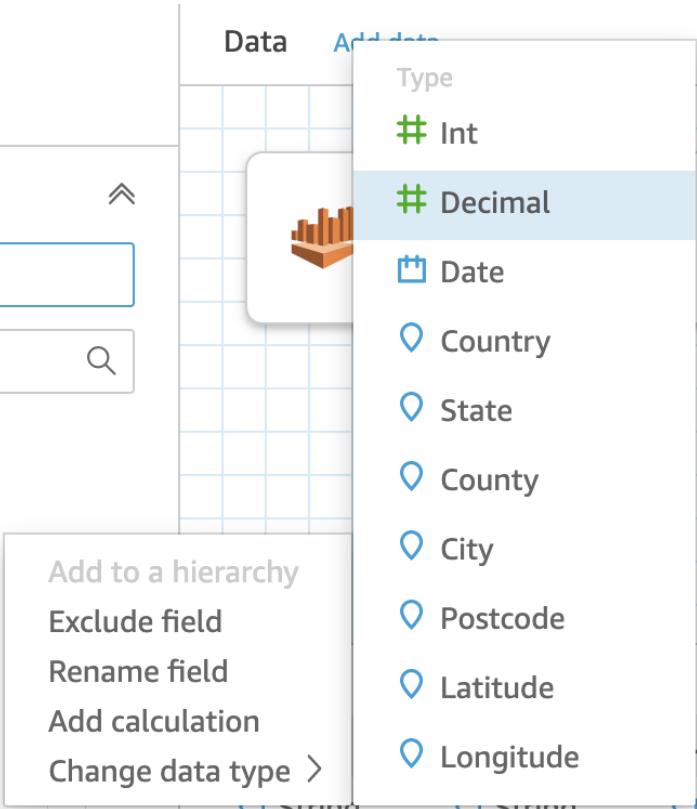
Calculated fields

calc_CallSentiment

Select All | None

filename

languagecode



Type
Int
Decimal
Date
Country
State
County
City
Postcode
Latitude
Longitude

5. We will create a calculated field called calc_CallSentiment, for this click on Add calculated field.

Query mode

SPICE Direct query

Fields All fields selected

Add calculated field

6. Enter the name and the content of the formula:

```
ifelse({sentimentispositive} =1,"Positive",({sentimentisnegative}=1),"Negative","Ne
```

The screenshot shows the AWS Quicksight calculated field editor. At the top, there's a dark header bar with a blue square icon and the text "Add calculated field". Below it, the field name "calc_CallSentiment" is displayed with a save icon. On the right, there are "Cancel" and "Save" buttons. The main area contains a code editor with the following formula:

```
1 ifelse({$sentimentispositive} =1,"Positive",{$sentimentisnegative}=1),"Negative","Neutral")
```

To the right of the code editor is a "Fields" section with a search bar labeled "Search fields".

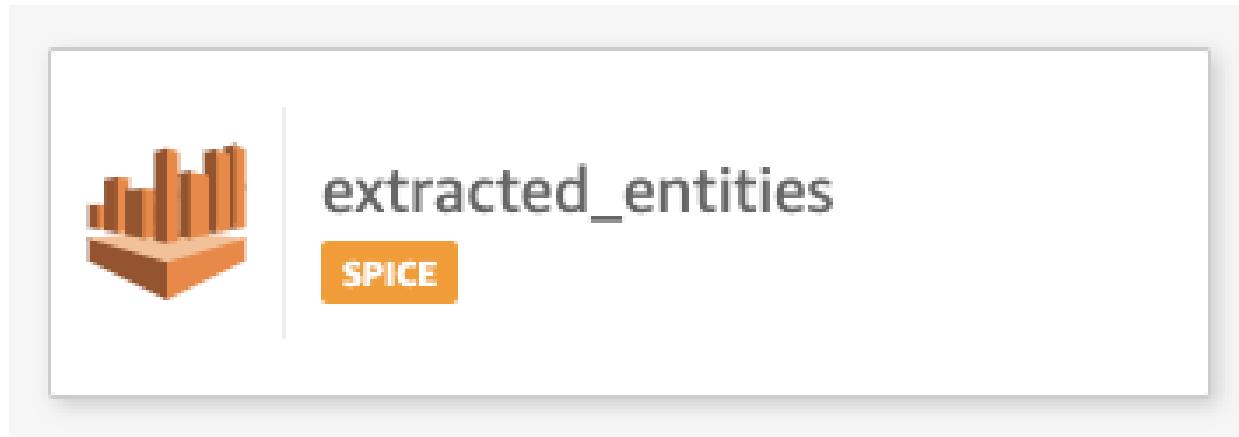
7. Click on Save

Creating sample Dashboards

Now that we have our datasets created in quicksight we can start creating our dashboards.
Click on "New Analysis"



1. Select the "extracted entities" dataset



2. Click on "Create Analysis"



extracted_entities

X

SPICE Data Set 6KB

Import complete:

100% success

8 rows were imported to SPICE

0 rows were skipped

Last refreshed: a day ago

[View history](#)

[Refresh Now](#)

[Schedule refresh](#)

Email owners when a refresh fails

Data source name: extracted_entities

Database name: ATHENA

[Delete data set](#)

[Share](#)

[Edit data set](#)

[Duplicate data set](#)

[Create analysis](#)

3. On the Field list, select the field "Text"

Data set



SPICE

extracted_entities

100%

Fields list



• **Labels**

 **utterance**

 **endoffset**

 **filename**

 **formatedconversation**

 **languagecode**

 **score**

 **segmentendtime**

 **segmentspeaker**

 **segmentstarttime**

 **sentimentisnegative**

 **sentimentispositive**

 **sentimentneutral**



SENTIMENTSCORE



speaker



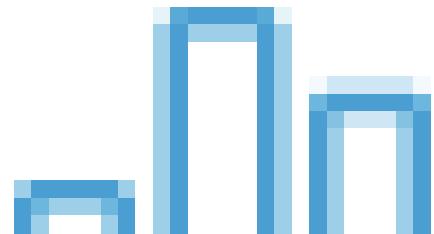
text

4. On the Visual types, select the word cloud

Visual types

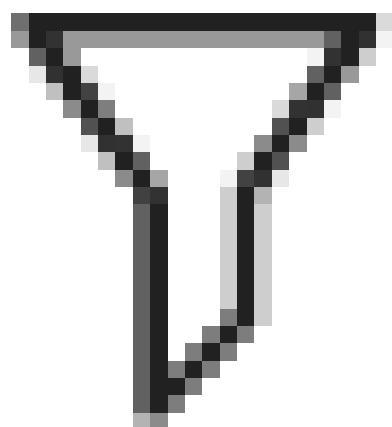


5. On the left toolbar, click on filter

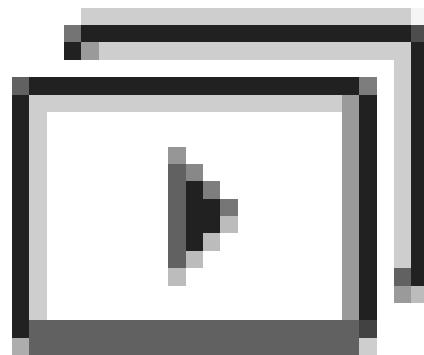


U U U

Visualize



Filter



Story

6. Click on "Create one..."



No filters for the selected visual

[Create one...](#)

7. Click on the field "calc_CallSentiment"

beginoffset

calc_CallSentiment

conversationtime

8. On the filter list select the Sentiment related to that entity and click

Edit filter <

Only this visual ▾

calc_CallSentiment ⚡

Include - all

Filter type

Filter list ▾

Looking for a value? Search or
Refresh this list.

Show selected values

Search values



Select all

Neutral

Positive

OR

Add filter condition

Data sets in this analysis

x

Data set	Status	...
 extracted_entities 	Available	...

[Close](#)

[Add data set](#)

4. Select a dataset, in my case it is "conversation_segments", click on the Select button:

Choose data set to add

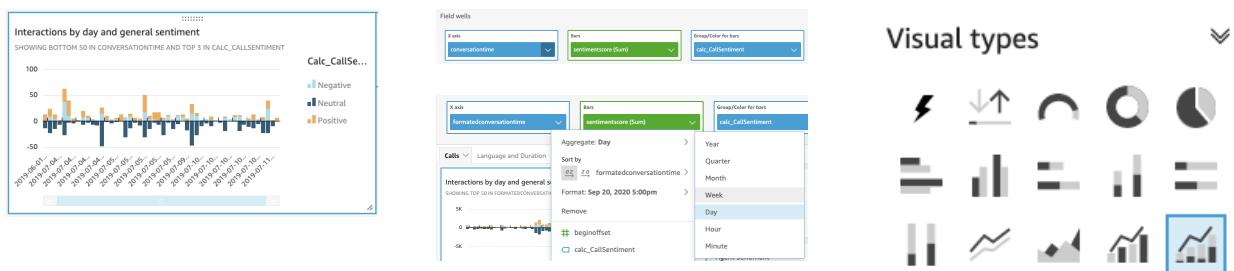
x

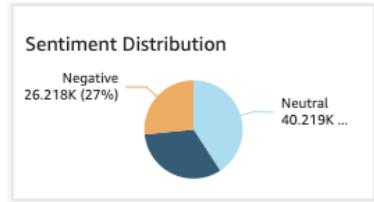
<input checked="" type="radio"/> conversation_segments	 Athena	
<input type="radio"/> call_duration_and_language	 Athena	
<input type="radio"/> Web and Social Media Analytics	 S3	
<input type="radio"/> People Overview	 S3	
<input type="radio"/> Sales Pipeline	 S3	
<input type="radio"/> Business Review	 S3	

[Cancel](#)

[Select](#)

Conversation Segments





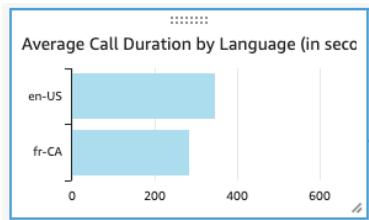
Field wells

Group/Color calc_CallSentiment	Value calc_CallSentiment (Count)
-----------------------------------	-------------------------------------

Visual types



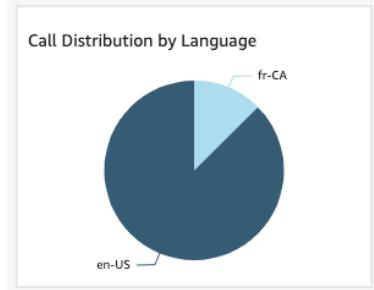
Language and Duration



Field wells

Y axis languagecode	Value duration (Average)
------------------------	-----------------------------

Visual types



Field wells

Y axis languagecode	Value duration (Average)
------------------------	-----------------------------

Visual types

