



Implementation Guide

# Discovering Hot Topics Using Machine Learning



# Discovering Hot Topics Using Machine Learning: Implementation Guide

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

---

# Table of Contents

<b>Solution overview .....</b>	<b>1</b>
Features and benefits .....	3
Use case .....	4
Concepts and definitions .....	4
<b>Architecture overview .....</b>	<b>6</b>
Architecture diagram .....	6
AWS Well-Architected design considerations .....	9
Operational excellence .....	9
Security .....	10
Reliability .....	10
Performance efficiency .....	11
Cost optimization .....	11
Sustainability .....	11
<b>Architecture details .....</b>	<b>12</b>
Amazon QuickSight dashboard .....	12
Pre-built Amazon QuickSight dashboard details .....	12
Example use cases for Amazon QuickSight .....	17
Customizing Amazon S3 ingestion .....	20
Microsoft Excel files .....	22
JSON files .....	23
Transcribe Call Analytics .....	24
AWS services in this solution .....	25
<b>Plan your deployment .....</b>	<b>27</b>
Cost .....	27
Example cost tables .....	28
Security .....	34
IAM roles .....	34
Amazon S3 .....	34
YouTube credentials .....	35
Reddit credentials .....	35
Supported AWS Regions .....	35
Quotas .....	36
Quotas for AWS services in this solution .....	36
AWS CloudFormation quotas .....	36

<b>Deploy the solution</b> .....	<b>37</b>
Deployment process overview .....	37
Prerequisite .....	37
AWS CloudFormation template .....	38
Step 1: Launch the stack .....	38
Step 2: Configure Amazon QuickSight .....	48
Post-deployment configuration .....	51
Update Data Visualization timeframe .....	51
<b>Monitor the solution with Service Catalog AppRegistry</b> .....	<b>52</b>
Activate CloudWatch Application Insights .....	52
Confirm cost tags associated with the solution .....	54
Activate cost allocation tags associated with the solution .....	54
AWS Cost Explorer .....	55
<b>Update the solution</b> .....	<b>56</b>
<b>Troubleshooting</b> .....	<b>57</b>
Amazon QuickSight nested stack failures .....	57
Dead Letter Queue for failed ingestion events .....	58
Contact AWS Support .....	58
Create case .....	58
How can we help? .....	58
Additional information .....	58
Help us resolve your case faster .....	59
Solve now or contact us .....	59
<b>Uninstall the solution</b> .....	<b>60</b>
Using the AWS Management Console .....	60
Using AWS Command Line Interface .....	60
<b>Developer guide</b> .....	<b>62</b>
Source code .....	62
<b>Supplemental topics</b> .....	<b>63</b>
Retrieve the Amazon QuickSight Principal ARN .....	63
Retrieve and manage API Key for YouTube API authentication .....	63
Retrieve and manage API credentials for Reddit API authentication .....	64
Database schema information .....	66
<b>Reference</b> .....	<b>68</b>
Anonymized data collection .....	68
Contributors .....	70

---

<b>Revisions .....</b>	<b>71</b>
<b>Notices .....</b>	<b>76</b>

# Identify the most dominant topics associated with your products, brands, and topics relevant to your business

Publication date: *August 2020* ([last update: July 2024](#))

The Discovering Hot Topics Using Machine Learning solution identifies the most dominant topics associated with your products, policies, events, and brands. This enables you to react quickly to new growth opportunities, address negative brand associations, and deliver a higher level of customer satisfaction for your business. In addition to helping you understand what your customers are saying about your brand, this solution gives you insights into topics that are relevant to your business.

This solution deploys an AWS CloudFormation template to automate data ingestion from these sources:

- RSS news feeds
- YouTube comments tied to videos
- Reddit (comments from subreddits of interest)
- Custom data in JSON or XLSX format

This solution uses pre-trained machine learning (ML) models from Amazon Comprehend, Amazon Translate, and Amazon Rekognition to provide these benefits:

- **Detecting dominant topics using topic modeling**—identifies the terms that collectively form a topic.
- **Identifying the sentiment of what customers are saying**—uses contextual semantic search to understand the nature of online discussions.
- **Determining if images associated with your brand contain unsafe content**—detects unsafe and negative imagery in content.
- **Helping you identify insights in near real-time**—uses a visual dashboard to understand context, threats, and opportunities almost instantly.

The solution can be customized to aggregate other social media platforms and internal enterprise systems. The default CloudFormation deployment sets up custom ingestion configuration

with parameters and an Amazon Simple Storage Service (Amazon S3) bucket to allow [Amazon Transcribe Call Analytics](#) output to be processed for natural language processing (NLP) analysis.

With minimal configuration changes in the custom ingestion functionality, this solution can ingest data from both internal systems and external data sources, such as transcriptions from call center calls, product reviews, movie reviews, and community chat forums including Twitch and Discord. This is done by exporting the custom data in JSON or XLSX format from the respective platforms and then uploading it to an Amazon Simple Storage Service (Amazon S3) bucket that is created when deploying this solution. For more details on how to customize this feature, see [Customizing Amazon S3 ingestion](#).

After you deploy the solution, you can use the included Amazon QuickSight dashboard to visualize the solution's ML inferences.

This implementation guide describes architectural considerations and configuration steps for deploying this solution in the Amazon Web Services (AWS) Cloud. This solution's [AWS CloudFormation](#) template launches and configures the AWS services required to deploy the solution using AWS best practices for security, availability, performance efficiency, and cost optimization.

This solution is intended for deployment by IT Specialists, IT Architects, Administrators and DevOps professionals with experience in the AWS Cloud.

Use this navigation table to quickly find answers to these questions:

If you want to . . .	Read . . .
Know the cost for running this solution.  The estimated cost for running this solution in the US East (N. Virginia) Region is USD \$375.00 per week for AWS resources.	<a href="#">Cost</a>
Understand the security considerations for this solution.	<a href="#">Security</a>
Know how to plan for quotas for this solution.	<a href="#">Quotas</a>
Know which AWS Regions support this solution.	<a href="#">Supported AWS Regions</a>

If you want to . . .	Read . . .
View or download the AWS CloudFormation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution.	<a href="#">AWS CloudFormation template</a>
Access the source code and optionally use the AWS Cloud Development Kit (AWS CDK) to deploy the solution.	<a href="#">GitHub repository</a>

## Features and benefits

The solution provides the following features:

### **Automate data ingestion from internal and external data sources**

The solution can be customized to aggregate other social media platforms and internal enterprise systems. With minimal configuration changes in the custom ingestion functionality, this solution can ingest data from both internal systems and external data sources, such as transcriptions from call center calls, product reviews, movie reviews, and community chat forums (including Twitch and Discord).

### **Detect dominant topics using topic modeling**

The solution uses machine learning models from Amazon Comprehend, Amazon Translate, and Amazon Rekognition to detect dominant topics and identify the terms that collectively form a topic. The solution is also able to use contextual semantic search to identify customer sentiment.

### **Multi-lingual data ingestion**

The solution uses Amazon Translate to ingest data in multiple languages.

### **Determine if images associated with your brand contain unsafe content**

The solution uses machine learning (ML) models to detect unsafe and negative imagery in content.

### **Near real-time insights**



The solution ingests streaming data containing text and images, then analyzes them in near real-time. The solution uses a pre-built Amazon QuickSight dashboard to visualize near real-time insights to understand context, threats, and opportunities almost instantly.

## **Integration with Service Catalog AppRegistry and Application Manager, a capability of AWS Systems Manager**

This solution includes a [Service Catalog AppRegistry](#) resource to register the solution's CloudFormation template and its underlying resources as an application in both Service Catalog AppRegistry and [Application Manager](#). With this integration, centrally manage the solution's resources and enable application search, reporting, and management actions.

## **Use case**

### **Quickly identify new growth opportunities, negative brand associates, and customer needs**

Use the Discovering Hot Topics Using Machine Learning solution to gain insights associated with your products, policies, events, and brands. The solution can be configured to ingest data from social media platforms in near real-time, and from internal data sources such as call center transcripts, reviews, and feedback.

## **Concepts and definitions**

This section describes key concepts and defines terminology specific to this solution:

### **application**

A logical group of AWS resources that you want to operate as a unit.

### **ingestion**

Refers to the processing and conversion of raw data.

### **insight**

An understanding from the processed information.

### **sentiment**

A positive or negative opinion.

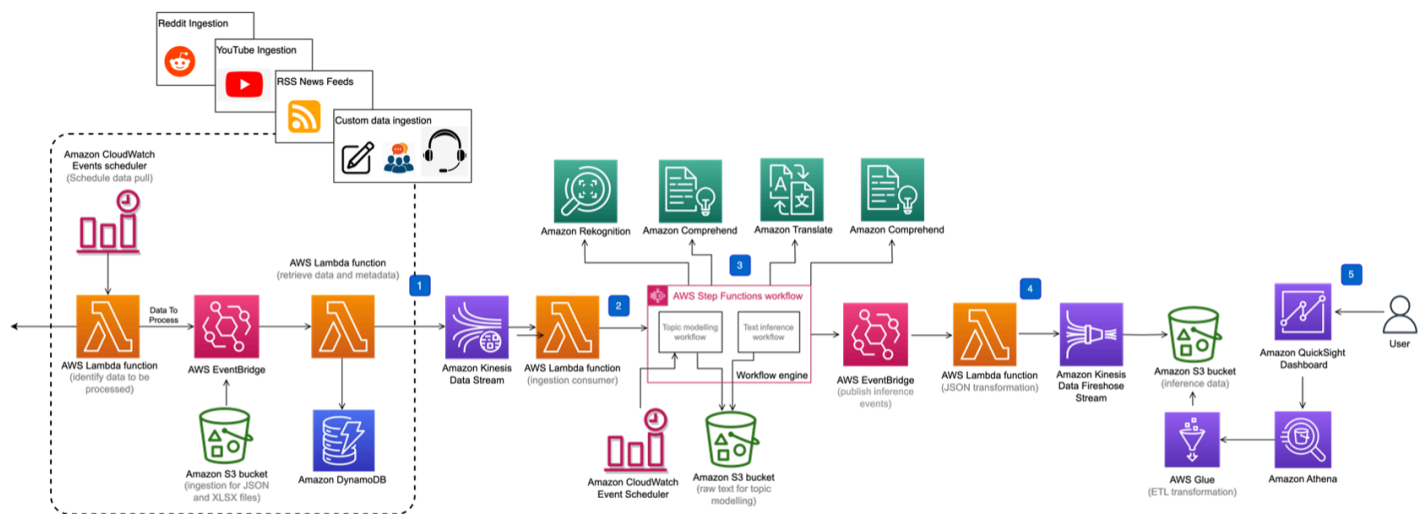
For a general reference of AWS terms, see [AWS Glossary](#).

# Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

## Architecture diagram

Deploying this solution with the default parameters builds the following environment in the AWS Cloud.

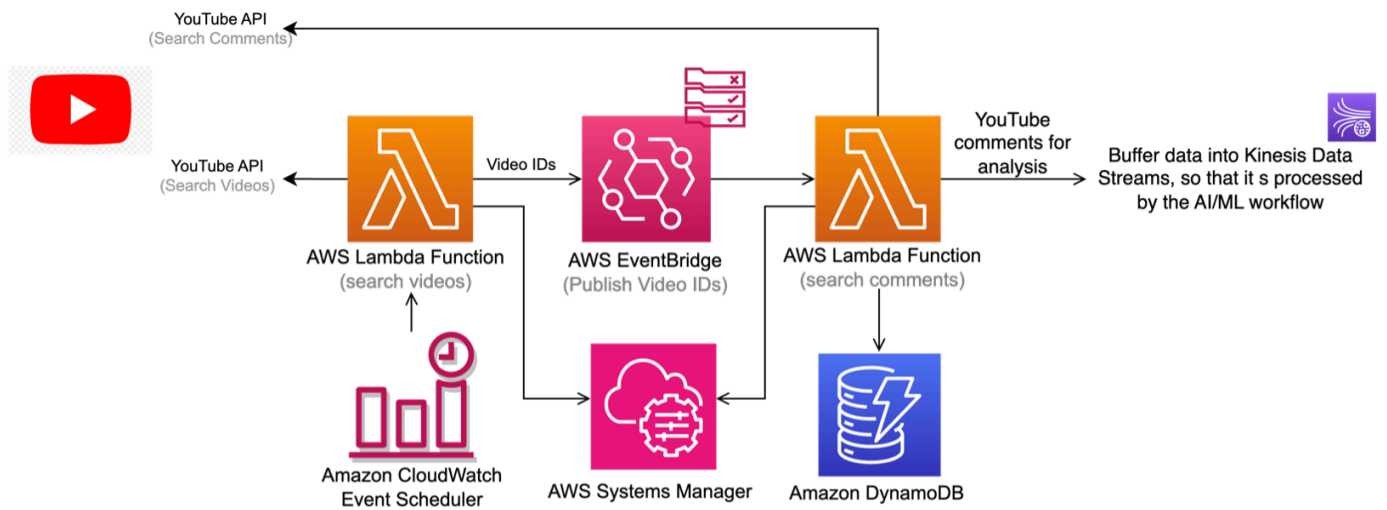


### Discovering Hot Topics Using Machine Learning solution architecture

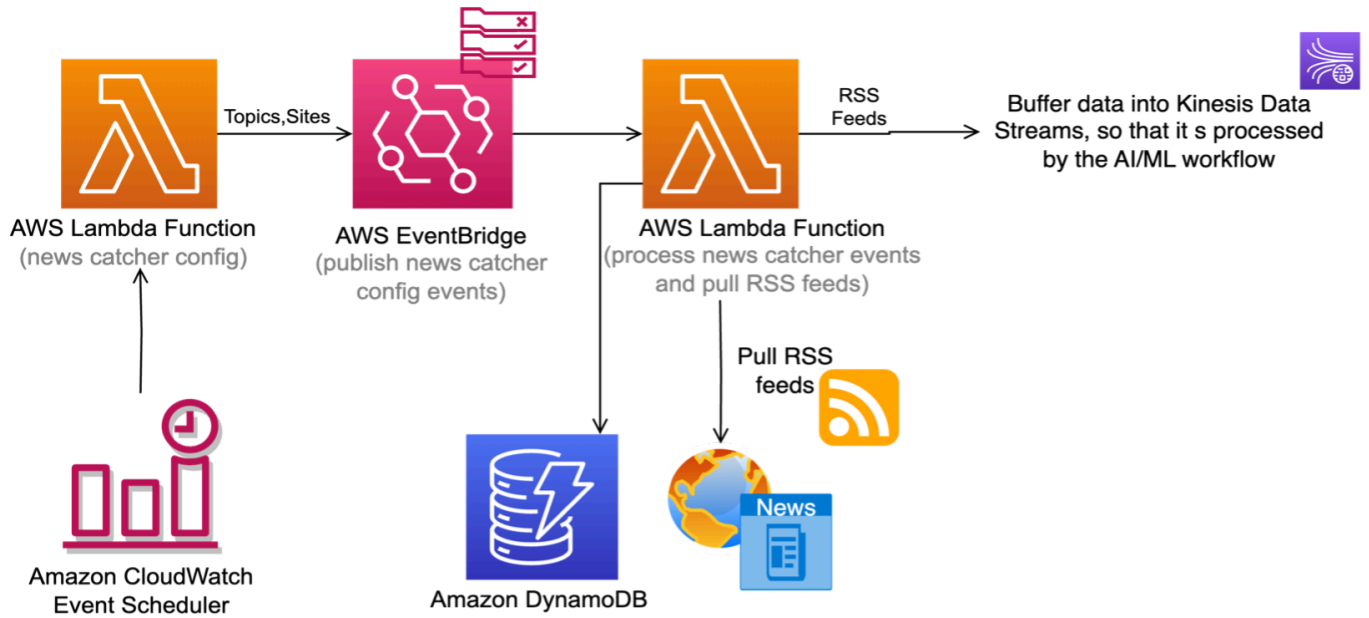
The AWS CloudFormation template automatically deploys [AWS Lambda](#) functions, [Amazon Simple Storage Service](#) (Amazon S3) buckets, [Amazon Kinesis Data Streams](#), [Amazon Simple Queue Service](#) (Amazon SQS) dead-letter queue (DLQ), [Amazon Data Firehose](#), [AWS Step Functions](#) workflows, [AWS Glue](#) tables, and [Amazon QuickSight](#) resources in your account.

The solution architecture includes the following key components and workflows:

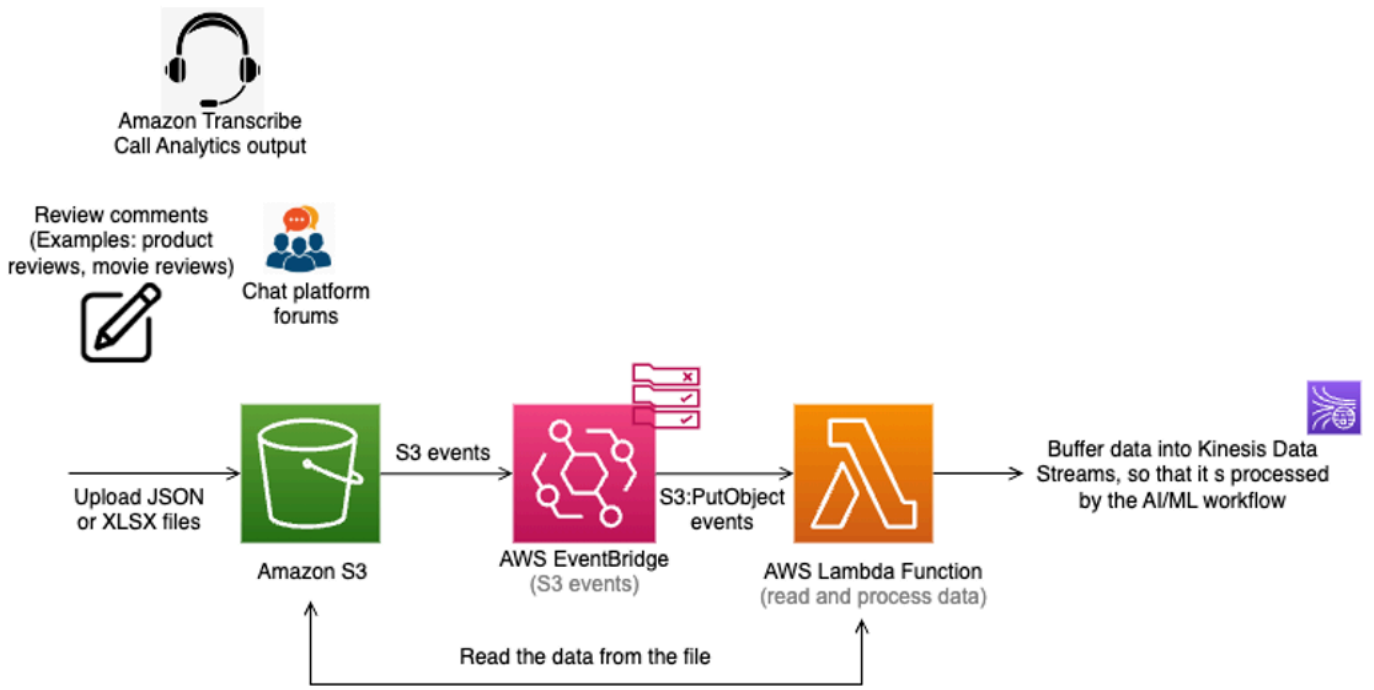
1. **Ingestion** – Social media and RSS feed ingestion and management using Lambda functions, [Amazon DynamoDB](#), and [Amazon EventBridge](#). The following figures are ingestion reference architecture diagrams for YouTube comments, RSS news feeds, and custom ingestion using an Amazon S3 bucket.



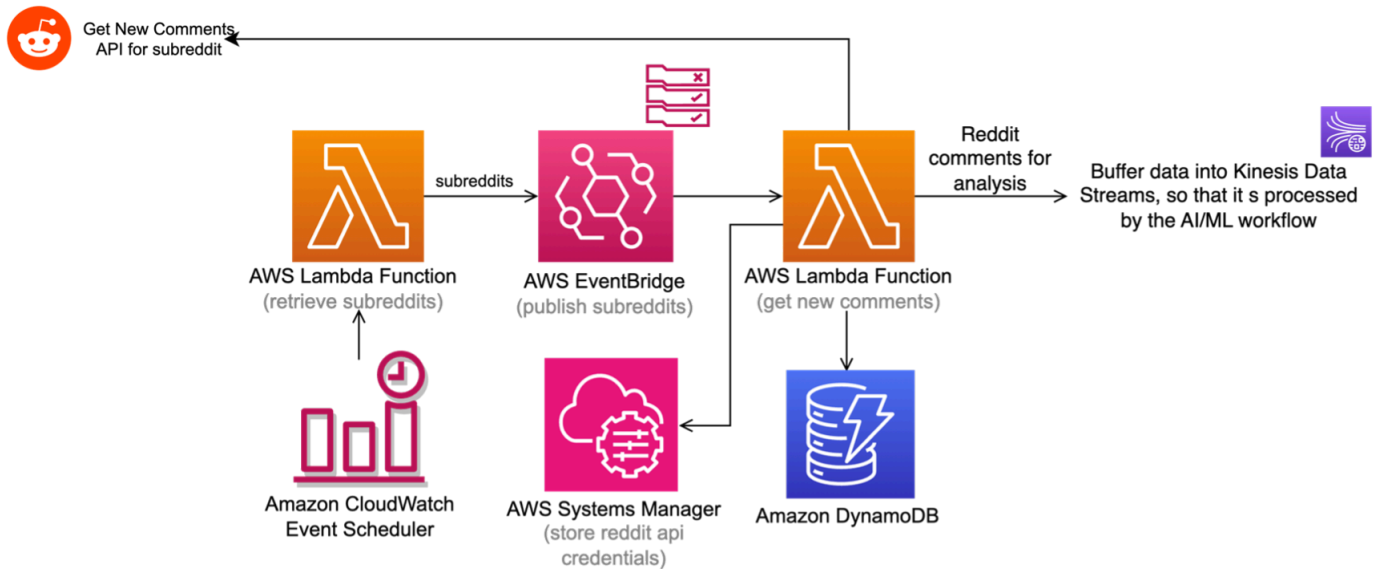
### Ingesting YouTube comments using the YouTube Data API



### Ingesting RSS news feeds



### Ingesting custom data in JSON or XLSX file feeds uploaded to an S3 bucket



### Ingesting comments from Subreddits of interest

The resources for the ingestion adapters are created and deployed based on the choices selected when deploying this solution's CloudFormation template.

2. **Data stream** – The data is buffered through Amazon Kinesis Data Streams to provide resiliency and throttle incoming requests. The Data Streams have a configured DLQ to catch any errors in processing feeds.
3. **Workflow** – Consumer (Lambda function) of the Data Streams initiates a Step Functions workflow that orchestrates Amazon Machine Learning capabilities including: [Amazon Translate](#), [Amazon Comprehend](#), and [Amazon Rekognition](#).
4. **Integration** – The inference data integrates with the storage components through an event-driven architecture using Amazon EventBridge. EventBridge allows further customization to add additional targets by configuring rules.
5. **Storage and visualization** – A combination of Amazon Firehose, Amazon S3 buckets, [AWS Glue](#) tables, [Amazon Athena](#), and [Amazon QuickSight](#).

## AWS Well-Architected design considerations

This solution uses the best practices from the [AWS Well-Architected Framework](#), which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework benefit this solution.

### Operational excellence

This section describes how we architected this solution using the principles and best practices of the [operational excellence pillar](#).

This solution's AWS CloudFormation template was built with the [AWS Cloud Development Kit \(AWS CDK\)](#) (AWS CDK). The template was built without hardcoding resource names or Regions, which ensures that it can be replicated in any Region where the services required by the solution are available.

[Amazon CloudWatch Logs](#) for Lambda functions and monitoring features provided by services such as Step Functions, Kinesis Data Streams, and Firehose provide observability into the infrastructure.

Integrating [AWS Systems Manager](#) Application Manager in this solution helps investigate and remediate issues with AWS resources used by the solution.

## Security

This section describes how we architected this solution using the principles and best practices of the [security pillar](#).

This solution implements encryption-at-rest and encryption-in-transit. For encryption-at-rest, Amazon Amazon S3 buckets and DynamoDB tables have SSE-S3 AWS managed encryption activated. For encryption-in-transit, all endpoints for AWS Cloud services use HTTPS endpoints, and Kinesis Data Streams have AWS managed encryption activated.

### Important

The solution's default implementation ingests various data types, including public information from social media platforms. However, it does not automatically redact personally identifiable information (PII) data. When the solution is extended for use cases that involve processing PII data, we recommend using [Amazon Macie](#) to detect any PII and Amazon Comprehend to redact any personal information before processing it through the solution's workflow. Refer to [Using managed data identifiers in Amazon Macie](#) in the *Amazon Macie User Guide* or [Detecting PII entities](#) in the *Amazon Comprehend Developer Guide*.

Additionally, the interactions between the services within this solution are controlled by [AWS Identity and Access Management](#) (IAM) role policies. The policies are configured on the principle of least privilege access.

## Reliability

This section describes how we architected this solution using the principles and best practices of the [reliability pillar](#).

This solution is based on AWS serverless artificial intelligence (AI), compute, and storage services: Lambda, Amazon Rekognition, Amazon Translate, Amazon Comprehend, and DynamoDB to ensure high availability and reliability. The workflow tasks are backed through an SQS based asynchronous call back [service integration pattern](#) to mitigate throttling errors from burst workloads. The solution also uses Dead Letter Queue (DLQ) as an option to route failed events and allow you to troubleshoot and resolve underlying issues.

## Performance efficiency

This section describes how we architected this solution using the principles and best practices of the [performance efficiency pillar](#).

This solution uses Lambda functions to provide concurrency and scaling, which ensures efficient use of compute resources.

It uses DynamoDB to achieve higher throughput with sub-millisecond latency, and resiliency through automatic scaling and on-demand scaling. Kinesis Data Streams provides data buffering that makes the architecture resilient to data bursts and spikes. Data is stored in columnar format and partition to optimize query performance for reporting.

## Cost optimization

This section describes how we architected this solution using the principles and best practices of the [cost optimization pillar](#).

The choice of serverless components in compute, storage, and AI services ensures that you are only charged for the services you use.

Using the DynamoDB on-demand capacity mode provides customers with the option to better understand their workloads and update their Read Capacity Unit (RCU) and Write Capacity Unit (WCU) based on each individual workload.

A Lambda function is invoked nightly to create partitions for the AWS Glue tables, which eliminates the need for AWS Glue Crawler to scan the entire dataset, and saves on cost.

## Sustainability

This section describes how we architected this solution using the principles and best practices of the [sustainability pillar](#).



# Architecture details

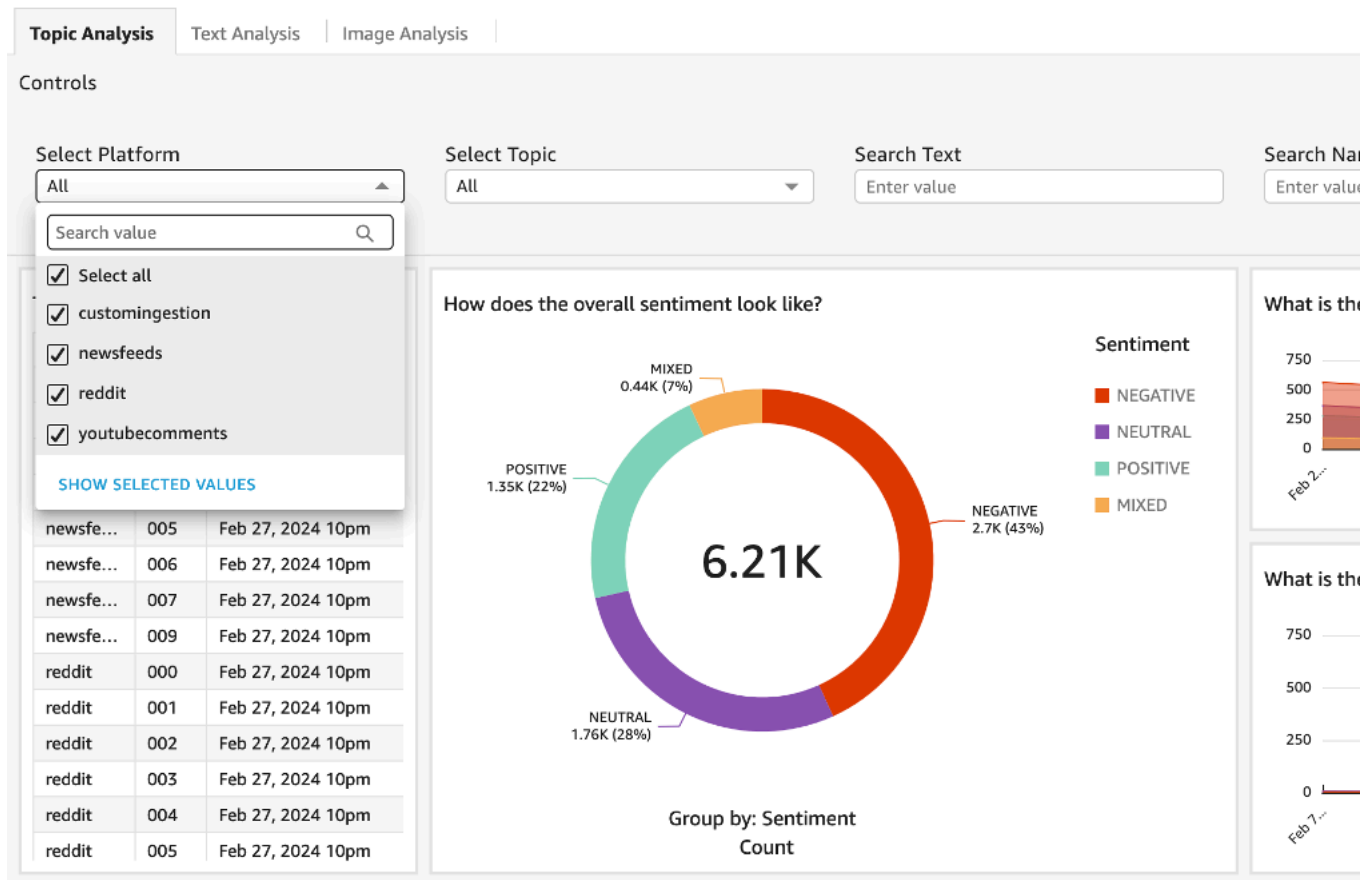
This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

## Amazon QuickSight dashboard

With this solution you can use QuickSight dashboards to visualize and filter data. This section covers some use cases for using them to visualize feeds/comments and sentiments.

### Pre-built Amazon QuickSight dashboard details

The solution uses machine learning algorithms to identify the most dominant topics referenced in ingested text and image data. A list of topics is generated: '000', '001', '002' and so on, where '000' is the most dominant topic within the dataset. Each topic consists of a collection of the relevant phrases within that topic.



### Controls on the Amazon QuickSight dashboard

The following controls allow you to filter the data on the various charts:

- Select platform** – Allows selection of multiple platforms and filters the data from the deployed platform. The control displays all source options (**newsfeeds, youtubecomments, and customingestion**) irrespective of the ingestion option selected when deploying the solution.
- Select topic** – Allows selection of multiple topics as detected by the solution.
- Search text** – Provides a text box to filter the ingested data that contains this key word or phrase. This control is configured to search for data on the translated text field.
- Search name** – Provides a mechanism to search by user id (for Reddit comments) or video title (for YouTube videos). This search is an exact search and is case sensitive.
- Search source** – This field provides the link to the post (for Reddit) or link to the video (for YouTube) or websites path (for news feeds). This search is an exact search and is case sensitive.

Field wells

Topic Analysis | Text Analysis | Image Analysis | Analysis by Geography | +

Controls

Select Platform: All | Select Topic: All | Search Text: Enter value | Search Name: Enter value | Search Source: Enter value

**Topic and job timestamp** 1

Platform	Topic	Job timestamp
newsfe...	000	Jul 14, 2021 12am
newsfe...	001	Jul 14, 2021 12am
newsfe...	002	Jul 14, 2021 12am
newsfe...	003	Jul 14, 2021 12am
newsfe...	005	Jul 14, 2021 12am
newsfe...	006	Jul 14, 2021 12am
newsfe...	007	Jul 14, 2021 12am
newsfe...	008	Jul 14, 2021 12am
newsfe...	009	Jul 14, 2021 12am
twitter	000	Jul 14, 2021 12am
twitter	001	Jul 14, 2021 12am
twitter	002	Jul 14, 2021 12am
twitter	003	Jul 14, 2021 12am

**How does the overall sentiment look like?** 2

Group by: Sentiment  
Tweet Count

**What is the sentiment trend over the last 7 days** 3

**What is the sentiment trend over the last 30 days** 4

**What are the phrases in the selected topic** 5

SHOWING TOP 100 IN TERM

**How does the sentiment look like for the selected topic?** 6

Size: id\_str (Count distinct)  
Group By: sentiment

**How does the heat map of the dominant topics trend over time** 7

Topic job timestamp (HOUR)

**Tabular view of All** 8

Platform	Timestamp	Identifier	Name	Source	Translated text	Text
newsfeeds	Jul 13, 2021	1626134471082#...	tech	https://www.leiphone.com/c...	Author   Yang Li in the past five years, the CTO Wang Xiaobo experienced the process of "digging" and "filling pit", so far let him afterta...	作者   杨丽过去五年, 同程艺龙机票事业...
newsfeeds	Jul 13, 2021	1626134471082#...	tech	https://www.leiphone.com/c...	High availability capability and meet the requirements of security. The purpose is to simplify operational tasks in development and focu...	性高可用的能力, 并且满足安全性的要求
newsfeeds	Jul 13, 2021	1626134598566#...	news	https://www.manager.bg/biz...	Richard Branson's flight to space has shot investors in Virgin Galactic to new heights, Blumerg reported. Shares of the space tourism co...	Полетът на Ричард Брансън до Космос

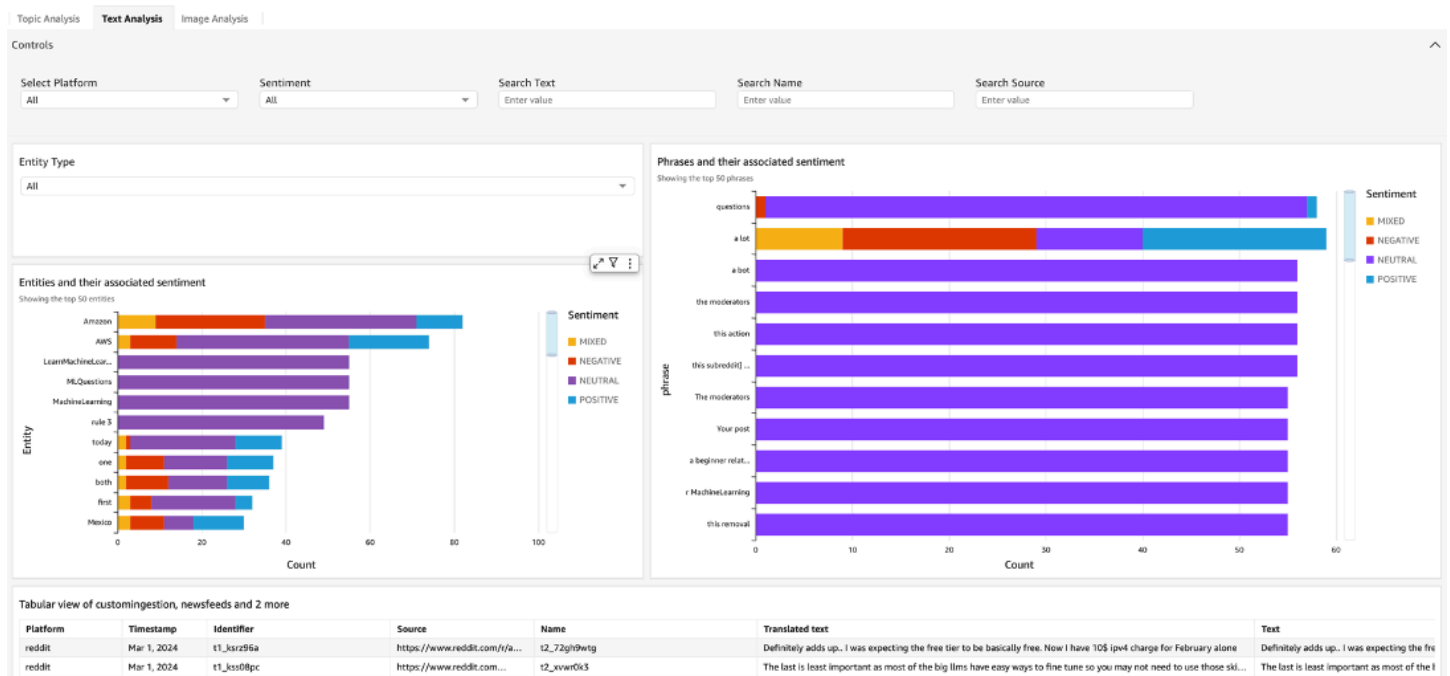
## Example Amazon QuickSight dashboard for aggregating topic analyses

The example QuickSight dashboard in the previous figure is a topic analysis dashboard for aggregating and contextualizing data from an ingestion source. The ingestion source can be selected from the **Source** dropdown in the **Controls** section. All the charts on this analysis worksheet render based on the selected **Source**. The first row in the figure has four visuals:

1. A table displaying the 10 most dominant topics in the dataset. Selecting a specific topic filters the word cloud of phrases (visual #5), the donut chart representing the sentiment for the selected topic and phrase (visual #6), the heat map of topics (visual #7), and a tabular view of ingested data (visual #8) to render information for the selected topic.
2. A donut chart representing overall sentiment analysis of the dominant topics (positive, negative, neutral, or mixed sentiment). Selecting a specific sentiment filters the table with a list of the most dominant topics (visual #1), the word cloud of phrases (visual #5), the donut chart representing the sentiment for the selected topic and phrase (visual #6), the heat map of topics (visual #7), and the table (visual #8) to render information for the selected sentiment.
3. An area line chart plotting a brand's sentiment trend mapped over the last seven days. Selecting a specific plot point for sentiment and date filters a table with the list of the most dominant topics (visual #1), the word cloud of phrases (visual #5), the donut chart representing the sentiment for the selected topic and phrase (visual #6), the heat map of topics (visual #7), and the table (visual #8) to render information for the selected sentiment.
4. An area line chart plotting a brand's sentiment trend mapped over the last 30 days.
5. You can use the dashboard to filter information and gain insights on sentiment context and customer perception. The visuals in the second row in the previous figure explore the most dominant topic '000'. The second row contains the following visuals:
6. A word cloud aggregating all of the detected phrases in the dominant topics. Selecting a specific phrase filters the list of dominant topics (visual #1), the donut chart representing the sentiment for the selected topic and phrase (visual #6), the heat map of topics (visual #7), and the table (visual #8).
7. A donut chart representing the sentiment analysis of the selected topic and phrase. Selecting a specific sentiment (Negative in the previous figure), filters the list of dominant topics with the selected sentiment (visual #1), the word cloud of phrases for the selected sentiment (visual #5), the heat map of topics (visual #7), and the table (visual #8).
8. A heat map with details of daily record counts for each topic. Selecting the heat map cell filters the table with a list of the most dominant topics (visual #1), the word cloud representing all the phrases within the dominant topics (visual #5), the donut chart representing the sentiment

analysis of the selected heat map cell (visual #6), the heat map of topics (visual #7), and the table (visual #8).

### 9. A tabulated view of the records ingested.



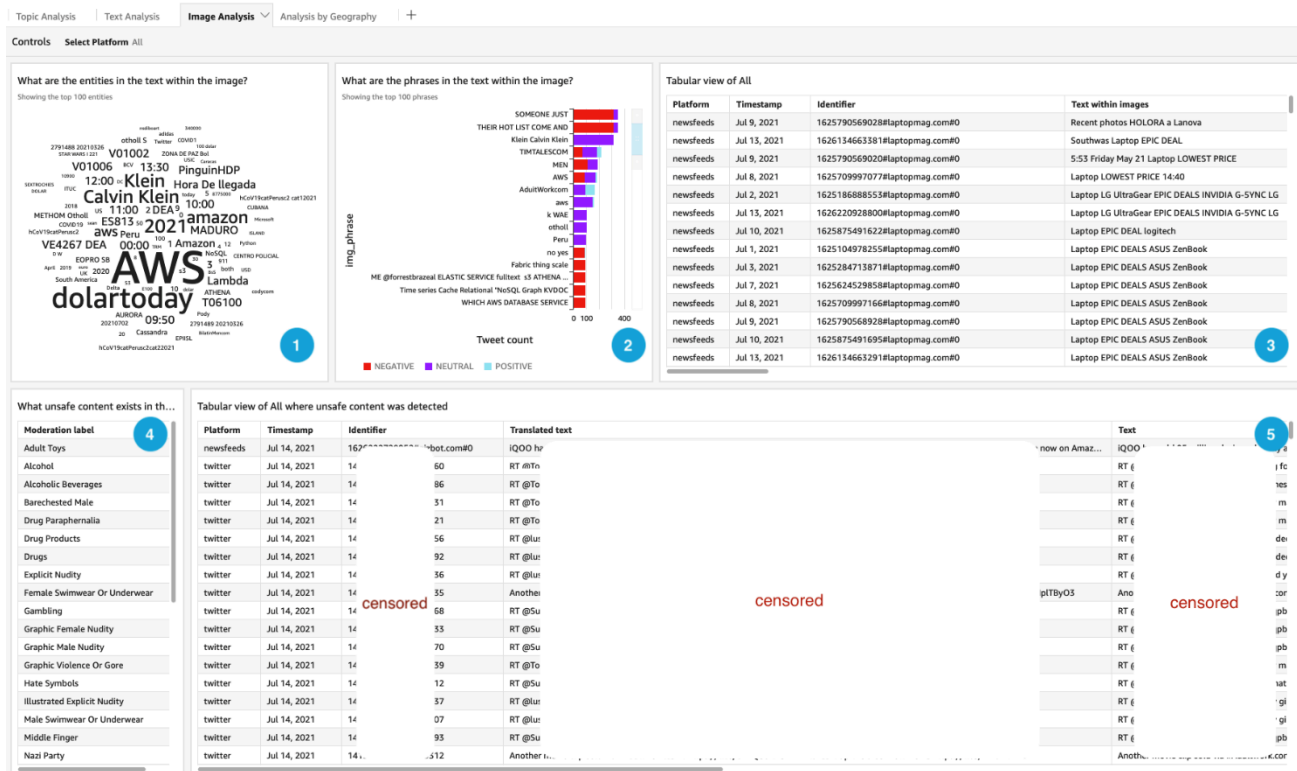
### Example Amazon QuickSight dashboard for text analysis

The ingested records are subjected to entity detection (detection of commercial items, date, event, location, organization, person, title, and quantity) and key phrase detection (detection of descriptive noun phrases). The Amazon QuickSight dashboard for text analysis displays the following elements:

1. A horizontal stacked bar chart displaying the top 50 entities grouped by sentiment (visual #1). You can select a specific entity and the sentiment group filters the tabular view of ingested data (visual #4).
2. A horizontal stacked bar chart displaying the top 50 key phrases grouped by their sentiment (visual #2). Selecting a specific phrase from the sentiment group filters the table (visual #4).
3. A tabular view of ingested data (visual #4) containing the date (Date), the ID (example: comment ID), the text, and the comment text translated to English. Selecting a record (row) in the table opens a new browser window that navigates to its source (this could be the Reddit post, the news site, or the YouTube video URL).

This solution also uses Amazon Rekognition to analyze images, detect entities in images (currently only JPEG images are supported), and extract embedded text from images. The service provides an unsafe image detection feature that creates moderation labels for images containing negative or unsafe content, for example, explicit adult content or content with violent elements. For more information on Amazon Rekognition Detection capabilities and image requirements, see [DetectModerationLabels](#) in the *Amazon Rekognition API Reference*.

The data generated from ingested images is used to generate topic modeling, key phrase, and sentiment analysis inferences. These inferences can be visualized using donut or pie charts, word clouds, or stack charts. This allows you to filter and focus your analysis on the specific context extracted from images. It provides the following visualizations:



### Example Amazon QuickSight dashboard for aggregating image analyses

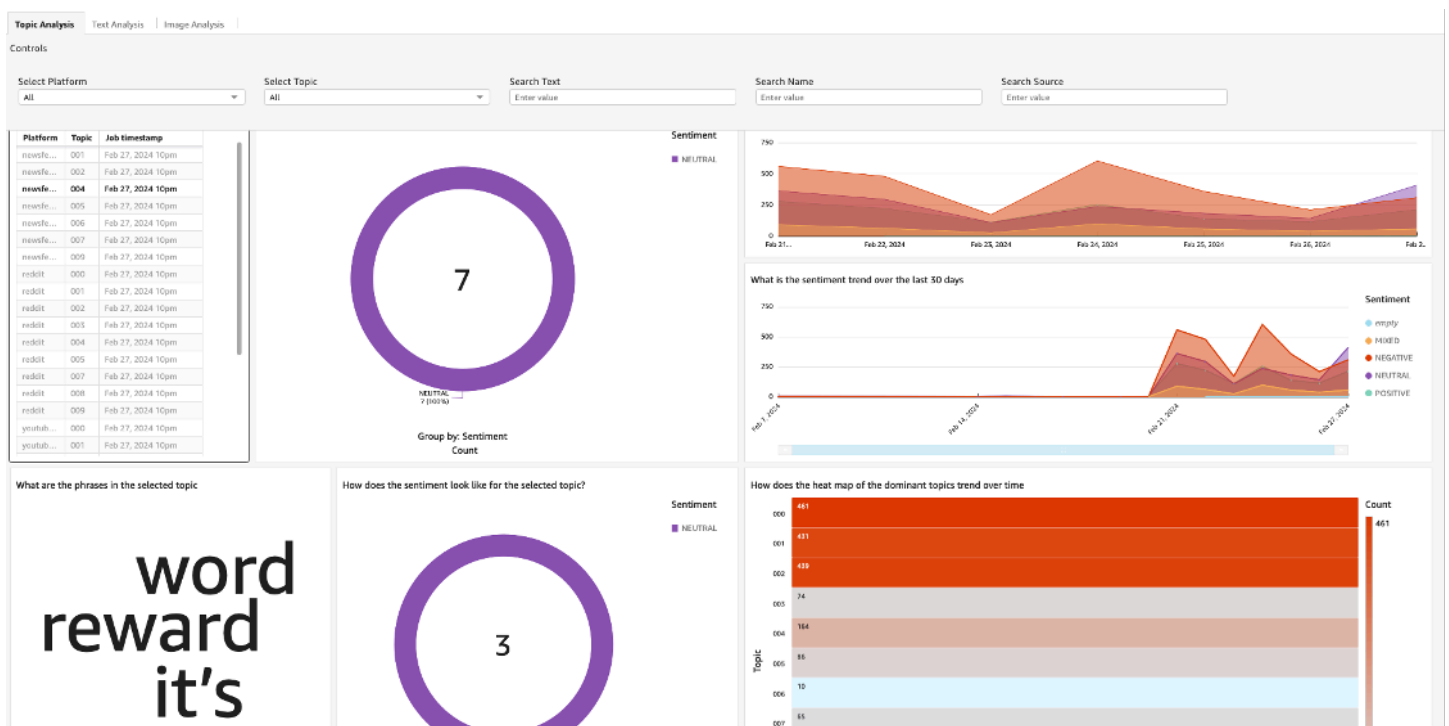
1. A word cloud aggregating the entities existing in the text embedded in the images. Selecting a word or phrase within the word cloud filters the visuals in the first row: the stacked horizontal bar chart of key phrases (visual #2) and the tabular view of ingested data (visual #3) to reflect the selection.
2. A stacked horizontal bar chart displaying key phrases in the text embedded in the images. Selecting a phrase with the sentiment refreshes the visuals in the first row: the word cloud (visual #1) and the table (visual #3).

3. A tabular view of the text embedded within the images and the URLs of the images. Selecting a record (row) in the table opens a new browser window that navigates to its sources.
4. The moderation labels associated with the images in the news feeds. Selecting a specific label filters the table (visual #5) that contains images flagged with that label.
5. A tabular view of records containing the moderation labels detected in the images. Selecting a record (row) in the table opens a new browser window that navigates to its source.

## Example use cases for Amazon QuickSight

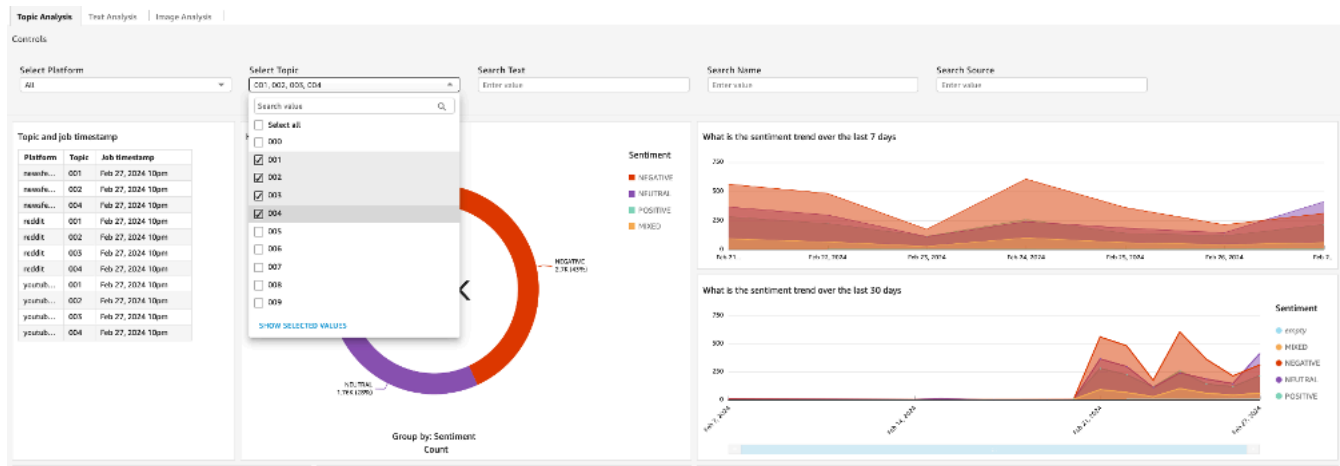
### Example 1: View the sentiments, phrases, and comments for a specific topic.

Select **any/all platforms** as the source, and then select a specific topic (for example, 004) in the table, to view the overall sentiment donut chart for that specific topic sentiment, the phrases in the most dominant topic word cloud, the heat map of feeds/comments associated with the dominant topics, and a tabular view of related comments.



### Visuals in the second and third row render information for the selected Topic '004'

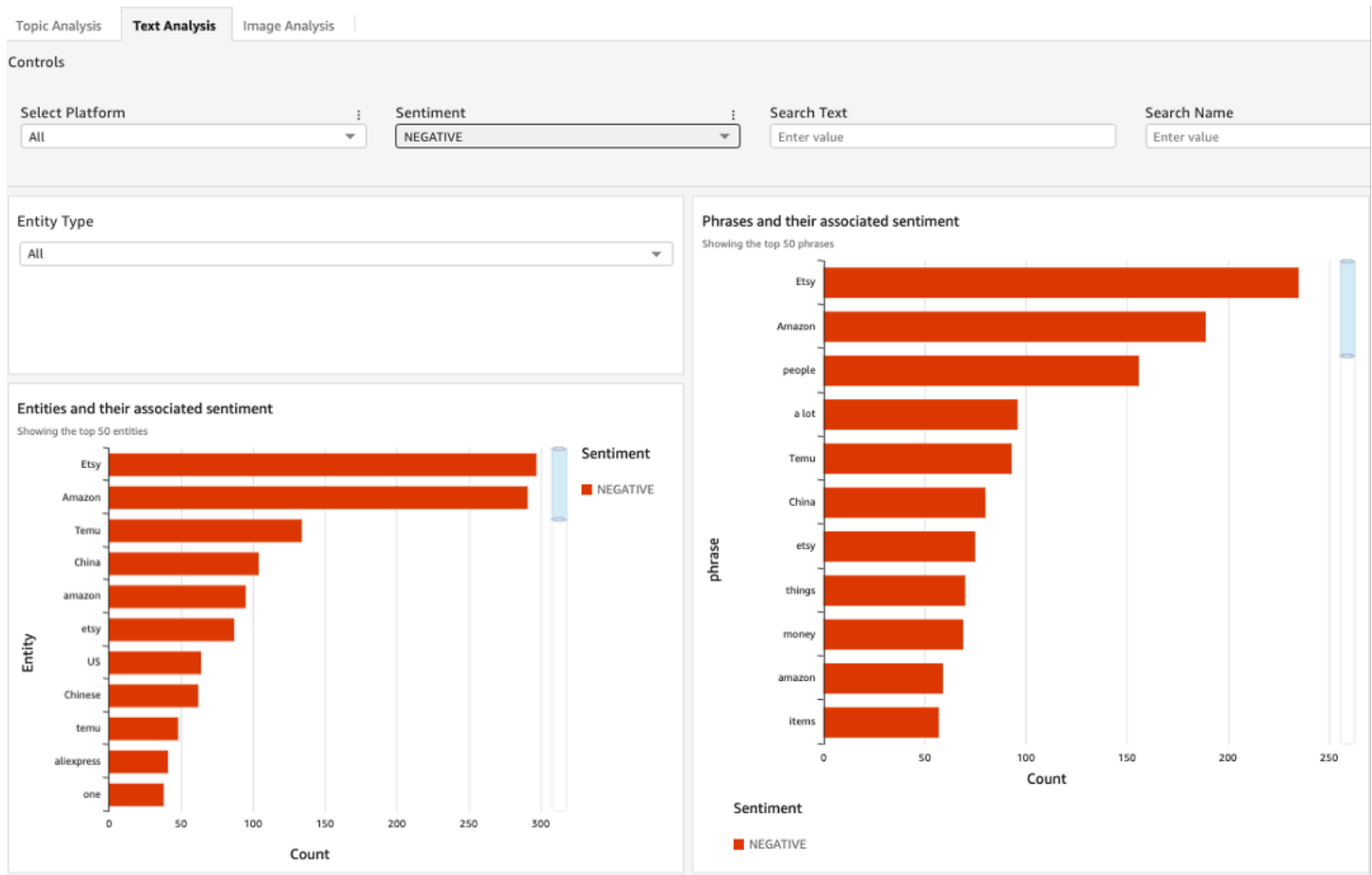
You can also use the **Controls** dropdown to expand the **Topic** and **Sentiment** options. These dropdowns have multi-select activated, thereby allowing you to select more than one topic or sentiment. Select topics that you would like to filter on to view the sentiments, phrases, and feeds/comments associated with that selection.



### Controls to multi-select topics or sentiments to filter visuals on the page

#### Example 2: View the comments/topics with negative sentiments.

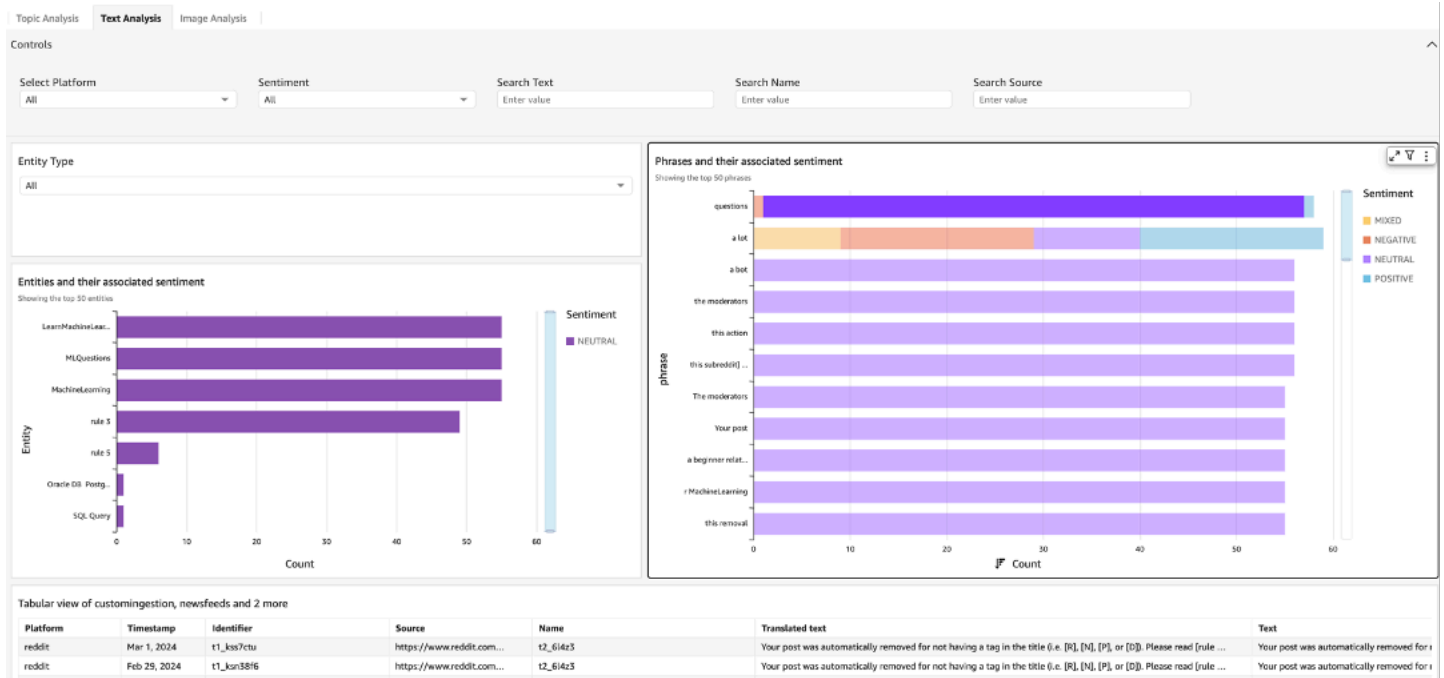
With any platform as the source, from the Text Analysis tab, select Negative. This refreshes the table of data in the third row visual.



### Filter visuals based on the sentiment

### Example 3: Select a phrase in the analysis and view the related comments/topics.

From the **Text Analysis** tab, choose a phrase from the **Phrases and their associated sentiment** horizontal stacked bar chart. This filters the table of comments/topics that contain the selected phrase and the associated sentiment.

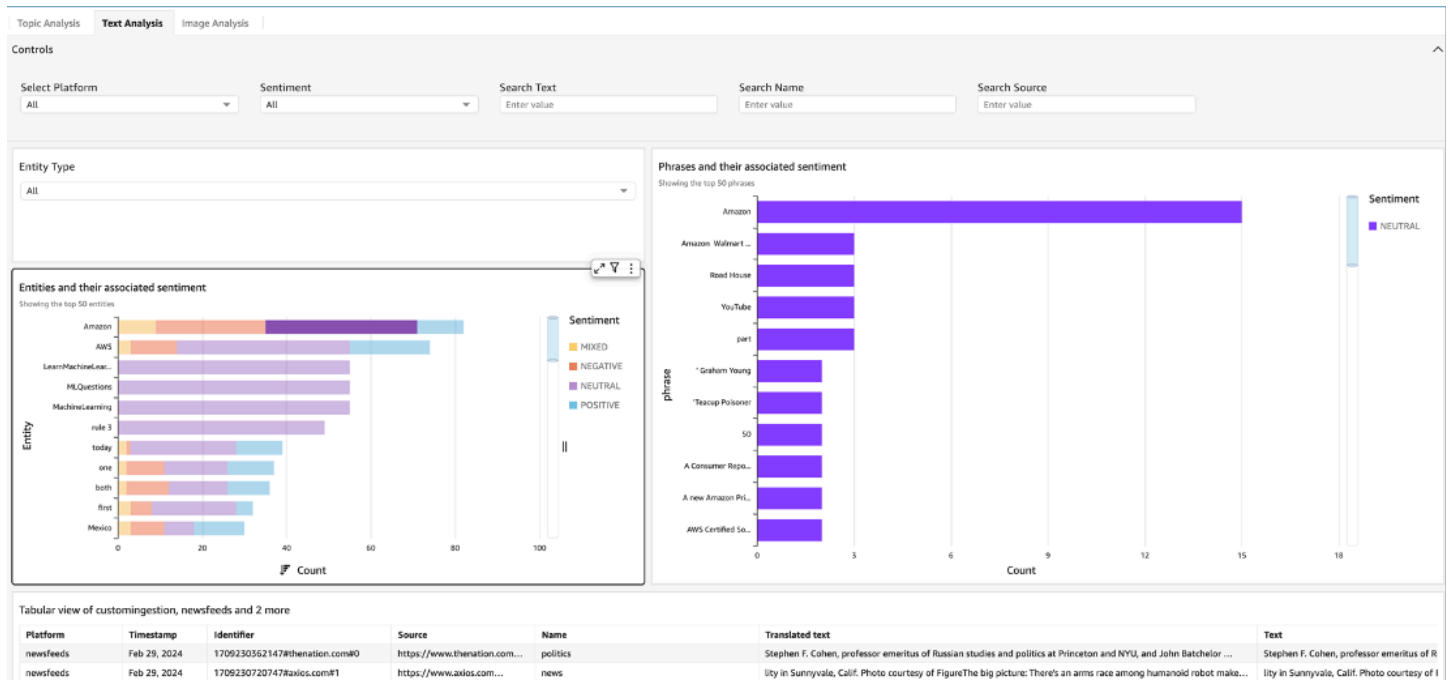


### Filter tabular view with the selected phrase

### Example 4: Filter comments/topics and sentiments for selected entity.

From the **Text Analysis** tab, select an entity from the **Entities** list and its associated sentiment from the horizontal stacked bar chart visual. This filters the table of comments/topics that contain the selected entity and the associated sentiment.





### Filter tabular view based on the selected entity

## Customizing Amazon S3 ingestion

One of the key features of the solution is the ability to ingest data uploaded to an Amazon S3 bucket. The source can be data exported of from internal or external services, or data in XLSX or JSON file format.

Examples of custom data that can be analyzed include:

- Product review system, movie, or content reviews.
- Internal or external chat forums, such as Twitch and Discord.
- Transcriptions from call center calls as generated by Amazon Transcribe Call Analytics.

When the solution is deployed, the default implementation is configured to process transcriptions from Amazon Transcribe Call Analytics.

Key entities that the solution requires to process data regardless of source type include:

- **ID** – A unique identifier for each record. If not known, set to GENERATE and the solution will generate a UUID for each file.

- **CREATED\_DATE** – Date associated with the record, if not known, set it to NOW and the solution will use the system's processing timestamp.
- **LANG** – The language in which text is present. If you do not know, do not set it. The solution will use Amazon Comprehend [Detecting the Dominant Language](#) operation to detect the language before subjecting it for any NLP analysis.
- **TEXT** – The text that should be subjected to NLP processing.

**Note**

The files can have additional columns with data elements, which the solution can store if defined in the schema definition AWS Glue customingestion table. Edit the schema for the customingestion table in the AWS Glue socialmediadb database in the AWS account and Region where the solution is deployed. Add Column Name and Data Type for the additional elements that need to be stored. For more information on working with AWS Glue tables, refer to [Working with Tables on the AWS Glue Console](#) in the *AWS Glue Developer Guide*.

The screenshot shows the AWS Glue console interface for the 'customingestion' table. At the top right, there are buttons for 'Partitions and Indices', 'View partitions', 'Compare versions', and 'Edit schema' (which is circled in red). Below this, the table's metadata is displayed, including its name, description, database, classification, location, connection, and various parameters like 'projection.created\_at.range' and 'projection.enabled'. At the bottom, the 'Schema' section shows a table with the following columns:

	Column name	Data type	Partition key	Comment
1	account_name	string		
2	platform	string		
3	id_str	string		
4	parent_id	string		
5	text	string		
6	lang	string		

### Custom ingestion table schema definition in AWS Glue

The solution provides three types of file processor implementations, which can be configured with the environment variables for an AWS Lambda function:

- Microsoft Excel files



## JSON files

When processing JSON documents, the solution requires keys within the JSON to query the information for analysis (set through environment variables). The solution uses [jmespath](#) to query JSON documents. The information provided through environment variables are jmespath selector expressions, which are processed by the solution.

### Example JSON document containing a list of records

```
{
  "list_contents": [
    {
      "content": "Lorem ipsum dolor sit amet, ",
      "id": "id1",
      "lang": "en",
      "created_date": "11-19-2021 03:59:07"
    },
    {
      "content": "consectetur adipiscing elit, ",
      "id": "id2",
      "lang": "en",
      "created_date": "11-19-2021 03:59:07"
    },
    {
      "content": "sed do eiusmod tempor incididunt ut labore et dolore magna aliqua",
      "id": "id3",
      "lang": "en",
      "created_date": "11-19-2021 03:59:07"
    }
  ]
}
```

In this code sample, the JSON document contains a list under the "list\_contents" key. In addition to setting the **ID**, **CREATED\_DATE**, **LANG**, and **TEXT environment variables**, this example requires setting the **LIST\_SELECTOR** expression as well.

If the JSON document has a list of records, the solution provides a mechanism to specify the key that contains the list using the **LIST\_SELECTOR** environment variable. For this example, the environment variables would need to be set according to the values in the following figure.

**Environment variables (11)**

The environment variables below are encrypted at rest with the default Lambda service key.

Key	Value
ACCOUNT_NAME	call_analytics
AWS_SDK_USER_AGENT	{ "user_agent_extra": "AwsSolution/SO0122, [REDACTED]" }
CREATED_DATE	created_date
ID	id
INTEGRATION_BUS_NAME	InfOutput-AppIntegration
LANG	lang
LIST_SELECTOR	list_contents
NAMESPACE	metadata.call_analytics
PLATFORM	customingestion
STREAM_NAME	[REDACTED]
TEXT	content

**AWS Lambda environment variables set up for JSON-based data ingestion****Example JSON document contains a single record**

```
{
  "content": "Lorem ipsum dolor sit amet, ",
  "id": "id1",
  "lang": "en",
  "created_date": "11-19-2021 03:59:07"
}
```

In this code sample, the JSON file contains a single record. Delete the **LIST\_SELECTOR** environment variable and leave the rest of the variables the same as the JSON document containing multiple records.

**Transcribe Call Analytics**

This is a special case of JSON document processing. In addition to the environment variables defined in Example JSON document containing list of records, set the following two environment variables:

- **SENTIMENT** – set value to sentiment
- **PROCESSOR\_TYPE** – set value to TRANSCRIBE\_CALL\_ANALYTICS

## AWS services in this solution

AWS service	Description
<a href="#">Amazon Comprehend</a>	<b>Core.</b> Derive and understand valuable insights from data sources during the machine learning (ML) workflow.
<a href="#">AWS Lambda</a>	<b>Core.</b> Provides logic for data ingestion and processing from social media and RSS feed.
<a href="#">Amazon QuickSight</a>	<b>Core.</b> Provides a topic analysis dashboard for aggregating and contextualizing data from an ingestion source.
<a href="#">Amazon Rekognition</a>	<b>Core.</b> This solution also uses Amazon Rekognition to analyze images, detect entities in images (currently only JPEG images are supported), and extract embedded text from images.
<a href="#">AWS Step Functions</a>	<b>Core.</b> Provides a workflow that orchestrates Amazon Machine Learning capabilities.
<a href="#">Amazon Translate</a>	<b>Core.</b> Translate data sources from different languages to English for ML processing workflow.
<a href="#">Amazon Athena</a>	<b>Supporting.</b> Works with QuickSight as an analytics tool to query and analyze data.
<a href="#">Amazon DynamoDB</a>	<b>Supporting.</b> Provides higher throughput with sub-millisecond latency and resiliency through automatic scaling and on-demand scaling.

AWS service	Description
<a href="#">Amazon EventBridge</a>	<b>Supporting.</b> Monitors for data added into S3 buckets to create an event-driven application.
<a href="#">AWS Glue</a>	<b>Supporting.</b> Works with QuickSight as a serverless data integration service that makes it easier to discover, prepare, move, and integrate data from multiple sources for analytics, ML, and application development.
<a href="#">AWS Identity and Access Management</a>	<b>Supporting.</b> Manages identity and provides access to different AWS services and resources.
<a href="#">Amazon Data Firehose</a>	<b>Supporting.</b> A fully managed service for delivering real-time streaming data to Amazon S3 buckets.
<a href="#">Amazon Kinesis Data Streams</a>	<b>Supporting.</b> Provides data buffering to Lambda functions that makes the architecture resilient to data bursts and spikes.
<a href="#">Amazon Simple Queue Service</a>	<b>Supporting.</b> Creates queues and DLQ's for processing data information and sources.
<a href="#">Amazon S3</a>	<b>Supporting.</b> Provides storage for raw data for ingestion and topic modeling.
<a href="#">AWS Systems Manager</a>	<b>Supporting.</b> Provides application-level resource monitoring and visualization of resource operations and cost data.
<a href="#">Amazon Macie</a>	<b>Optional.</b> Discovers and protects sensitive data by redaction before processing the data into the solution's workflow.

# Plan your deployment

This section describes the [cost](#), [security](#), [Regions](#), and [quota](#) considerations prior to deploying the solution.

## Cost

You are responsible for the cost of the AWS services used while running this solution, which can vary based on the following factors:

- The volume of data ingested (based on the configuration for the ingestion source — Reddit, RSS feeds, and/or YouTube comments).

When ingesting Reddit comments, the number subreddits of SubRedditsToFollow and ingestion frequency contribute to the volume of data that the solution ingests.

- **SubRedditsToFollow** – Adding more subreddits increases the volume of data.

When ingesting RSS feeds, the number of news sites and the search query string both contribute to the volume of data that the solution ingests.

- **Search query string** – Broader and more generic terms result in a larger volume of ingested data. Specific and precise terms result a smaller volume of ingested data.
- **News feed configuration** – A broad configuration (topics, languages, and countries) increases the volume of ingested data.

When ingesting YouTube comments, the ingestion works in two stages: (1) it retrieves the videos based on search criteria, and (2) it retrieves the comments for each of the videos. Ingestion search can be based on a **search query**, a **channel ID**, or both.

- **Search query** – A generic search query results in a larger list of videos, and a large volume of comments for NLP processing.
- **Channel ID** – The volume of videos in the YouTube channel and the number of comments associated with each of the videos.
- **Ingestion window** – The search filter defaults to filtering out videos published beyond the seven-day window. Increasing the window size can increase the number of videos searched and the volume of comments ingested. This filter can be customized from the Lambda environment variable.



- The number of queries for visualization.
- The number of records that require language detection (RSS feeds, the solution uses Amazon Comprehend to detect the data's source language before processing).
- The number of records that must be translated into English.
- The number of images (media assets) ingested with the RSS feeds.

As of this revision, the cost for running this solution in the US East (N. Virginia) Region using Reddit, RSS news feed, and YouTube comments ingestion, with the default values, and reports queried sporadically, is approximately **\$375 per week**. We recommend creating a [budget](#) through [AWS Cost Explorer](#) to help manage costs. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.

## Example cost tables

The following tables provide an example cost breakdown for deploying this solution with the default parameters in the US East (N. Virginia) Region for one week with different volume scenarios (excludes free tier).

**Example 1: Ingesting – RSS news feeds (~100 items/day) + comments from 2 subreddits (~1000 comments/day) + YouTube comments (~1600 comments/day) + custom ingestion (~600 items/day)**

AWS service	Dimensions	Cost [USD/month]
Amazon Athena	70 queries/week and 50 GB data scanned/query	\$75.00
Amazon CloudWatch Event Rule – 1	5,040 events/week (runs every 2 mins)	\$0.05
Amazon CloudWatch Event Rule – 2	14 events/week (runs every day)	\$0.0004
Amazon Comprehend	~100 news items/day + ~1000 Reddit comments/day + ~1600 YouTube comments/	\$210.00

AWS service	Dimensions	Cost [USD/month]
	day + ~600 custom ingestion items/day	
Amazon DynamoDB	Records inserted with TTL 7-day expiry to keep state for Reddit ingestion, YouTube ingestion and news feeds, on-demand capacity	\$0.50
Amazon Data Firehose	21 Firehose, total 3 GB/week	\$0.50
Amazon Kinesis Data Streams	1 datastream	\$10.00
Amazon Simple Queue Service (Amazon SQS)	15 queues (regular queues + DLQ)	\$2.00
Amazon Rekognition – Label and text detection	20 images/day	\$1.20
Amazon Simple Storage Service (Amazon S3)	5 buckets, 75 GB	\$20.00
Amazon Translate	Assuming 280 characters/item with 4,000 items, assumes 50% of 4,000 items in non-English language = 2,000 items = 560,000 characters/week	\$40.00
AWS Lambda	30 Lambda functions	\$4.00
AWS Step Functions	2 workflow definitions ~ 30 states (in all)	\$30.00

AWS service	Dimensions	Cost [USD/month]
AWS Key Management Service	Using AWS managed keys with DynamoDB, Amazon S3 (SSE-S3), Kinesis Data Streams, SQS	\$50.00
Amazon QuickSight	10 readers reading twice/day	\$60.00
Total:		~\$503.00/month

### Example 2: Ingesting – ~1000 Reddit comments/day

AWS service	Dimensions	Cost [USD/month]
Amazon Athena	70 queries/week and 100 GB data scanned/query	\$75.00
Amazon CloudWatch Event Rule – 1	5040 events/week (runs every 2 mins)	\$0.05
Amazon CloudWatch Event Rule – 2	7 events/week (runs every day)	\$0.0004
Amazon Comprehend	~1000 Reddit comments/day	\$56.00
Amazon DynamoDB	Records inserted with TTL 7-day expiry, on-demand capacity	\$0.20
Amazon Data Firehose	21 Firehose, total 3 GB/week	\$0.20
Amazon Kinesis Data Streams	1 data stream	\$11.00
Amazon Simple Queue Service (Amazon SQS)	15 queues (regular queues + DLQ) processing	\$1.00
Amazon Simple Storage Service (Amazon S3)	5 buckets, 75 GB	\$20.00

AWS service	Dimensions	Cost [USD/month]
Amazon Translate	Assuming 280 character s/item with 4,000 items, assumes 50% of 4,000 items in non-English language = 2,000 items = 560,000 characters/week	\$40.00
AWS Lambda	30 Lambda functions	\$4.00
AWS Step Functions	2 workflow definitions ~ 30 states (in all)	\$9.00
AWS Key Management Service	Using AWS managed keys with DynamoDB, Amazon S3 (SSE-S3), Kinesis Data Streams, SQS	\$30.00
Amazon QuickSight	10 readers reading twice/day	\$60.00
Total:		~\$306.00/month

### Example 3 – Ingesting ~1600 YouTube comments/day

AWS service	Dimensions	Cost [USD/month]
Amazon Athena	70 queries/week and 50 GB data scanned/query	\$75.00
Amazon CloudWatch Events – 1	5,040 events/week (runs every 2 mins)	\$0.05
Amazon CloudWatch Events – 2	14 events/week (runs every day)	\$0.0004
Amazon Comprehend	~1600 YouTube comments/day	\$56.00

AWS service	Dimensions	Cost [USD/month]
Amazon DynamoDB	Records inserted with TTL 7-day expiry to keep state for Reddit ingestion on-demand capacity	\$0.20
Amazon Data Firehose	21 Firehose, total 3 GB/week	\$0.20
Amazon Kinesis Data Streams	1 data stream	\$10.00
Amazon Simple Queue Service (Amazon SQS)	15 queues (regular queues + DLQ)	\$1.00
Amazon Simple Storage Service (Amazon S3)	5 buckets, 75 GB	\$20.00
Amazon Translate	Assuming 280 character s/item with 4,000 items, assumes 50% of 4,000 items in non-English language = 2,000 items = 560,000 characters/week	\$40.00
AWS Lambda (128 MB)	30 Lambda functions	\$4.00
AWS Step Functions	2 workflow definitions ~ 30 states (in all)	\$15.00
AWS Key Management Service	Using AWS managed keys with DynamoDB, Amazon S3 (SSE-S3), Kinesis Data Streams, SQS	\$30.00
Amazon QuickSight	10 readers reading twice/day	\$60.00
Total:		~\$311/month

#### Example 4: Ingesting ~100 news feeds per day

AWS service	Dimensions	Cost [USD/month]
Amazon Athena	70 queries/week and 20 GB data scanned/query	\$30.00
Amazon CloudWatch Event Rule – 1	5,040 events/week (runs every 2 mins)	\$0.04
Amazon CloudWatch Event Rule – 2	14 events/week (runs every day)	\$0.000336
Amazon Comprehend	~100 news feeds/day	\$30.00
Amazon DynamoDB	Records inserted with TTL 7-day expiry to keep state for new feeds on-demand capacity	\$1.20
Amazon Data Firehose	21 Firehose, total 3 GB/week	\$1.00
Amazon Kinesis Data Streams	1 data stream	\$11.00
Amazon Simple Queue Service (Amazon SQS)	14 queues (regular queues + DLQ)	\$0.60
Amazon Rekognition – Label and text detection	20 images/day	\$12.00
Amazon Simple Storage Service (Amazon S3)	5 buckets, 20 GB	\$12.00
Amazon Translate	Assuming 280 characters/item with 4,000 items, assumes 50% of 4,000 items in non-English language = 2,000 items = 560,000 characters/week	\$40.00
AWS Lambda	23 Lambda functions	\$4.00

AWS service	Dimensions	Cost [USD/month]
AWS Step Functions	2 workflow definitions ~ 30 states (in all)	\$2.00
AWS Key Management Service	Using AWS managed keys with DynamoDB, Amazon S3 (SSE-S3), Kinesis Data Streams, SQS	\$20.00
Amazon QuickSight	10 readers reading twice/day	\$60.00
Total:		~\$222.00/month

## Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit the [AWS Cloud Security](#).

### IAM roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users in the AWS Cloud. This solution creates IAM roles that grant the solution's AWS Lambda functions access to create Regional resources.

### Amazon S3

All Amazon S3 buckets are encrypted with SSE-S3 managed encryption. One of the buckets that stores images from news feeds includes a bucket policy that allows Amazon Rekognition to access the images for analysis.

None of the buckets are available publicly.

We recommend that you create lifecycle policies on the buckets based on your use case and your organization's data management policy standards to ensure that you are not paying for Amazon S3 data storage for the data that is no longer required for the solution.

**Note**

The Amazon S3 buckets are configured with the retention policy set to **Retain**.

## YouTube credentials

If you configure the solution for YouTube comment ingestion, we recommend that you rotate the API Key for YouTube Data API v3 to match with your password rotation policy. Google Cloud Platform supports regenerating the key, turning off the key, and removing YouTube Data API access for this key. For more information, refer to [Retrieve and manage API Key for YouTube Data API v3 authentication](#).

## Reddit credentials

If you configure the solution for ingesting subreddit comments, we recommend that you rotate the refreshToken for the Reddit API to match with your password rotation policy. Reddit's platform supports revoking the token and generating a new one. For more information, refer to [Retrieve and manage API credentials for Reddit API authentication](#).

## Supported AWS Regions

This solution uses the Amazon Rekognition, Amazon Translate, and Amazon Comprehend services, which are not currently available in all AWS Regions. You must launch this solution in an AWS Region where these services are available. For the most current availability by Region, refer to the [AWS Regional Services List](#).

Discovering Hot Topics Using Machine Learning is available in the following AWS Regions:

Region ID	Region name
us-east-2	US East (Ohio)
us-east-1	US East (N. Virginia)
us-west-2	US West (Oregon)
ap-south-1	Asia Pacific (Mumbai)



Region ID	Region name
ap-northeast-2	Asia Pacific (Seoul)
ap-southeast-2	Asia Pacific (Singapore)
ap-southeast-2	Asia Pacific (Sydney)
ap-northeast-1	Asia Pacific (Tokyo)
eu-central-1	Europe (Frankfurt)
eu-west-1	Europe (Ireland)
eu-west-2	Europe (London)

## Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

### Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the [services implemented in this solution](#). For more information, see [AWS service quotas](#).

Use the following links to go to the page for that service. To view the service quotas for all AWS services in the documentation without switching pages, view the information in the [Service endpoints and quotas](#) page in the PDF instead.

### AWS CloudFormation quotas

Your AWS account has AWS CloudFormation quotas that you should be aware of when launching the stack in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, see [AWS CloudFormation quotas](#) in the *AWS CloudFormation User's Guide*.

# Deploy the solution

This solution uses [AWS CloudFormation templates and stacks](#) to automate its deployment. The CloudFormation template specifies the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the template.

## Deployment process overview

Before you launch the solution, review the [cost](#), [architecture](#), [security](#), and other considerations discussed in this guide.

**Time to deploy:** Approximately 10 minutes

[Step 1: Launch the stack](#)

[Step 2: Configure Amazon QuickSight \(only if QuickSightPrincipalArn was provided as a CloudFormation parameter\)](#)

### Important

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered through this survey. Data collection is subject to the [AWS Privacy Policy](#).

To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your updated template and deploy the solution. For more information, see the [Anonymized data collection](#) section of this guide.

## Prerequisite

If you plan to use the Amazon QuickSight dashboard feature, you must subscribe to Amazon QuickSight Enterprise in the account where you deploy the solution. Use of Amazon QuickSight to derive insights with this solution is optional. You can use any other analytics tool to query data using Amazon Athena and AWS Glue services.

**Note**

If you deploy the Amazon QuickSight dashboard feature, make sure to also complete Step 2. Without completing Step 2, Amazon QuickSight does not have the required permissions to access the data in the Amazon S3 bucket and hence charts will display an error.

## AWS CloudFormation template

This solution uses AWS CloudFormation to automate the deployment of the Discovering Hot Topics Using Machine Learning solution in the AWS Cloud. It includes the following CloudFormation template, which you can download before deployment.

[View template](#)

**discovering-hot-topics-using-machine-learning.template** - Use this template to launch this solution and all associated components. The default configuration deploys AWS Lambda functions, Amazon DynamoDB, Amazon Kinesis Data Streams, Amazon Data Firehose, AWS Step Functions, and AWS Glue tables. You can customize the template to meet your specific needs.

### Step 1: Launch the stack

This automated AWS CloudFormation template deploys the Discovering Hot Topics Using Machine Learning solution in the AWS Cloud.

**Note**

You are responsible for the cost of the AWS services used while running this solution. For more details, refer to the [Cost](#) section in this guide and the pricing webpage for each AWS service.

1. Sign in to the AWS Management Console and select the button to launch the `discovering-hot-topics-using-machine-learning.template` AWS CloudFormation template.



You can also [download the template](#) as a starting point for your own implementation.

- The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

### Note


This solution uses the Amazon Rekognition, Amazon Translate, and Amazon Comprehend services, which are not currently available in all AWS Regions. You must launch this solution in an AWS Region where these services are available. For the most current availability by Region, refer to the [AWS Regional Services List](#).


- On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
- On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to [IAM and STS Limits](#) in the *AWS Identity and Access Management User Guide*.
- Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
<b>DeployNewsFeeds</b>	Yes	Select Yes if you would like to ingest RSS news feeds. If you select No, the CloudFormation template will not deploy the news feed ingestion components.
<b>NewsFeedIngestFrequency</b>	<code>cron(0 18 * * ? *)</code>	The ingestion schedule as a cron expression supported by a CloudWatch Event Rule

Parameter	Default	Description
		that schedules the ingestion of news feeds.
<b>NewsSearchQuery</b>	Amazon, AWS	(Optional) Comma-separated list of keywords to filter news feeds. Only feeds containing at least one of the keywords from the list will be processed. If no keyword is provided, feeds will not be filtered and all news feeds are processed.


Parameter	Default	Description
<b>NewsFeedIngestConfig</b>	<pre>{"country": "ALL", "language": "en", "topic": "news"}</pre>	<p>Provide configuration for RSS feeds. This parameter should be configured as a JSON string. Example: <code>{"country": "ALL", "language": "ALL", "topic": "ALL"}</code> .</p> <p>For Country and language use ISO code. The list of superset of all supported topics is: "tech", "news", "business", "science", "finance", "food", "politics", "economics", "travel", "entertainment", "music", "sport", "world".</p> <div data-bbox="1081 1150 1507 1562" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p><b>Note</b></p><p>Not all topics are supported for each RSS provider. Setting the value as ALL, is treated as a wild character search.</p></div>

Parameter	Default	Description
<b>DeployYouTubeCommentsIngestion</b>	Yes	Option to select if you would like to ingest YouTube comments. If you select No, the CloudFormation template will not deploy the YouTube ingestion components.
<b>YouTubeSearchQuery</b>	'Amazon Web Services AWS',	<p>Optional</p> <p>Search query for YouTube videos. The search query retrieves the list of videos using YouTube APIs and then retrieves the comments for each of the videos from the list.</p> <div data-bbox="1081 1037 1507 1352" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>YouTubeSearchQuery or YouTubeChannel is required.</p></div>

Parameter	Default	Description
<b>YouTubeChannel</b>	<blank>	<p>Optional</p> <p>YouTube channel ID. This ID is used to retrieve the list of vldeos and then retrieve comments for each of the videos from the list.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>YouTubeSe archQuery or YouTubeChannel is required.</p> </div>
<b>YouTubeSearchIngestionFreq</b>	cron(0 12 * * ? *)	<p>YouTube API invocation schedule as a cron expression supported by a CloudWatch Event Rule that schedules the ingestion of YouTube comments.This parameter is required.</p>
<b>YoutubeAPIKey</b>	/discovering-hot-topics-using-machine-learning/youtube/comments	<p>The key name required to retrieve the API key within AWS Systems Manager Parameter Store and access the YouTube APIs.</p>
<b>DeployRedditIngestion</b>	Yes	<p>Option to select if you would like to ingest comments from subreddits of interest.</p>




Parameter	Default	Description
<b>RedditAPIKey</b>	/discovering-hot-topics-using-machine-learning/reddit/comments .	<p>The key name required to retrieve the JSON string containing clientId, clientSecret, and refreshToken within AWS Systems Manager Parameter Store to access the Reddit API.</p> <div data-bbox="1081 590 1508 1283"><p><b>Note</b></p><p>Here is an example of a JSON string in the parameter store:</p><pre>{"clientId": "clientIdFromReddit", "clientSecret": "clientSecretFromReddit", "refreshToken": "generatedRefreshToken"}</pre></div> <p>To generate the refresh token, use the script or weblink provided at <a href="https://github.com/not-an-aardvark/reddit-oauth-helper">https://github.com/not-an-aardvark/reddit-oauth-helper</a> (Select the <b>Permanent</b> checkbox to generate refreshToken)</p>

Parameter	Default	Description
<b>RedditIngestionFrequency</b>	cron(0/60 * * * ? *)	<p>The frequency at which Reddit ingestion should pull comments from the subreddits of interest.</p> <div data-bbox="1081 445 1507 1142" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Because the API has throttling limits, this schedule configuration only queues the requests. The AWS Lambda implementation has been configured by default to not exceed the hourly limit by inducing sleep after every API call.</p> </div>
<b>SubRedditsToFollow</b>	r/aws,r/MachineLearning	The subreddits of interest to follow as comma separated values, (no space after the comma).

Parameter	Default	Description
<b>DeployCustomIngestion</b>	Yes	Option to select if you would like to ingest data uploaded in an S3 bucket. If you select No, the CloudFormation template will not deploy the custom ingestion components. Selecting Yes creates an S3 bucket where data can be uploaded. The bucket name will be in the output of the nested CloudFormation template that deploys the custom ingestion components.
<b>TopicAnalysisFrequency</b>	<code>cron(10 0 * * ? *)</code>	The schedule at which topic modeling jobs are run. Because the topic modeling jobs take approximately 35 minutes to run, the minute duration between jobs must be one hour. Additionally, because the data is stored in a folder structure that follows Apache Hive naming conventions, we recommend invoking the job a few minutes after the hour.
<b>NumberOfTopics</b>	10	The number of topics you want to detect as part of the topic modeling job between 1-100.

Parameter	Default	Description
<b>QuickSightPrincipalArn</b>	<blank>	Provide the ARN of the Amazon QuickSight user that must have permissions to view and edit the datasets, analysis, and dashboard created by the AWS CloudFormation. For details on how to retrieve the QuickSight User ARN, refer to Retrieve Amazon QuickSight Principal ARN.

 **Note**

If you leave this parameter blank, the solution deploys without the Amazon QuickSight resources that help visualize its various inferences.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a CREATE\_COMPLETE status in approximately 10 minutes.

**Note**

In addition to the primary AWS Lambda functions `<function(s)>`, this solution includes the `solution-helper` Lambda function, that runs only during initial configuration, or when updating or deleting resources.

When you run this solution, you will notice both Lambda functions in the AWS Management Console. Only the `<function>` function is regularly active. However, do not delete the `solution-helper` function, as it is needed to manage associated resources.

## Step 2: Configure Amazon QuickSight (only if QuickSightPrincipalArn was provided as a CloudFormation parameter)

After the stack is successfully deployed, you can retrieve the Amazon QuickSight URLs from the **Outputs** tab.


**Important**

The QuickSight datasets created by this solution use Direct Query to query S3 buckets for data. For better performance, you may use SPICE. Refer to [Using SPICE Data in an Analysis](#) in the *Amazon QuickSight User Guide* for more information about configuring and using SPICE.

Use the following steps to launch the Amazon QuickSight dashboard and view the analysis.

1. Sign in to the AWS Management Console and navigate to Amazon QuickSight.
2. Change the Region in the URL to match the Region where you deployed the solution. For example, if the solution was deployed in the `us-east-1` Region, the QuickSight URL will mirror the following path: `https://us-east-1.quicksight.aws.amazon.com/sn`.
3. From the left navigation menu in QuickSight, choose **Analyses**. The right pane displays the analysis that the solution automatically created: `S00122-<version-solution-name>`. Double-click the analysis title to open it.

Use the following steps to configure Amazon QuickSight permissions and complete the dashboard set up.

 **Note**

Based on the Topic job frequency schedule and the time at which the solution was deployed, it can take up to 24 hours post deployment for the some of the visuals to render. All jobs are scheduled in the UTC time zone.

1. Select your username, then choose **Manage QuickSight**.
2. From the left navigation menu, select **Security & Permissions**.
3. Under **QuickSight access to AWS Services**, choose **Add or Remove**.
4. Select **IAM**, **Amazon S3**, and **Amazon Athena**. If these options are already selected, uncheck and recheck the options.
5. Choose **Amazon S3**, select **Details**, choose **Select S3 buckets**, then choose the S3 bucket name that matches the following pattern: `<stackname>-infoutput-<uuid>`. If the solution has been successfully ingesting feeds for more than 10 minutes after deployment, this S3 bucket will contain some or all of following object keys: entity, keyphrase, moderationlabels, sentiment, topic-mappings, topics, txtinimgency, txtinimgkeyphrase, txtinimgsentiment.

 **Important**

As of February 2021, there is a known issue with the Amazon QuickSight template feature. If your QuickSight template contains visuals with column groups in the field wells, your dashboard may not render correctly. If you receive a No data error, you can use the following procedure to reconfigure your analysis, update the geospatial properties, and generate a successful visual.

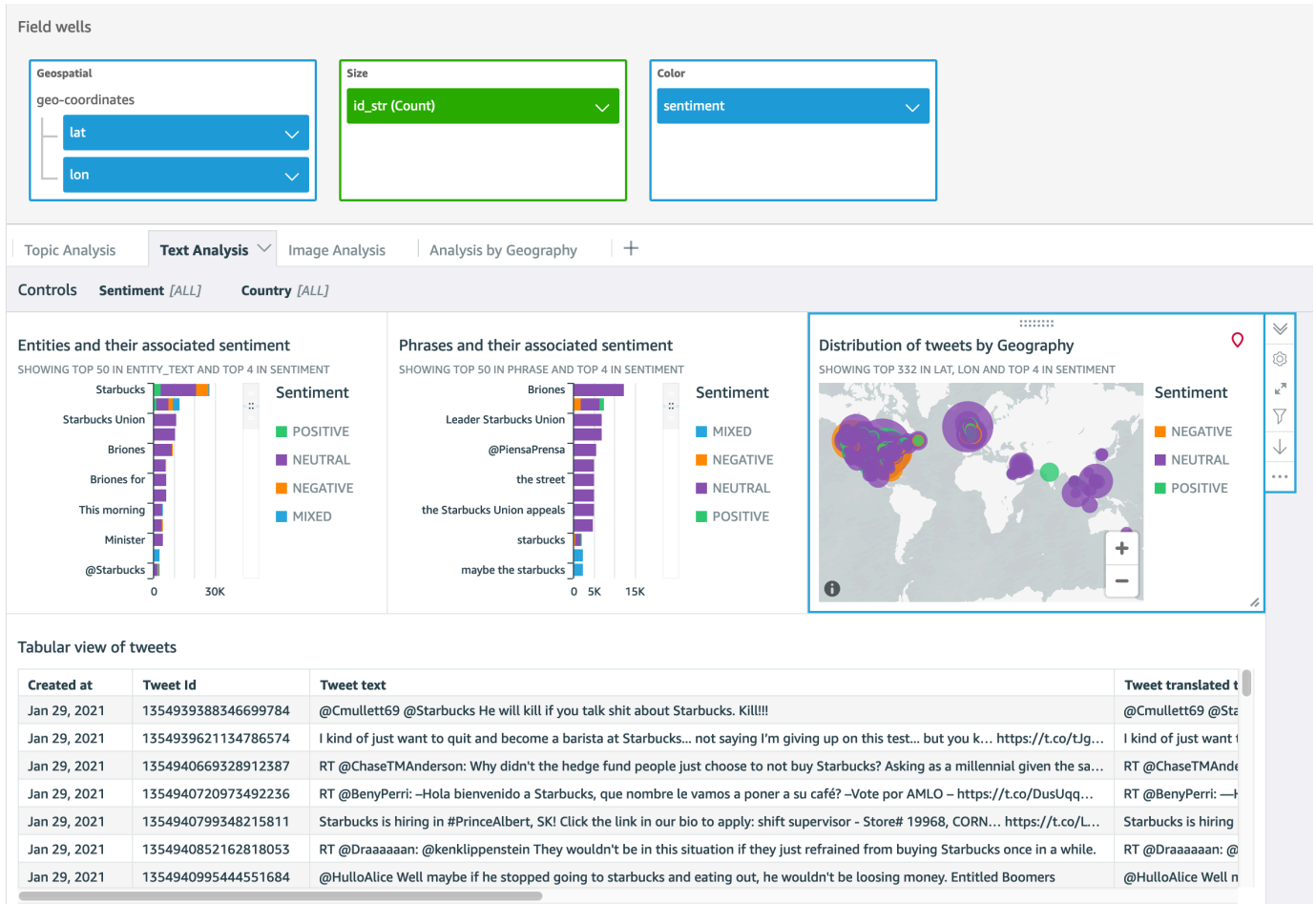
Use the following procedure to correct the **No data** error in Amazon QuickSight.

1. Open either the **Text Analysis** or **Analysis by Geography** tab and select the map visual.

**Note**

You must complete this procedure for *both* tabs to resolve this error.

2. Select **Field wells** to view the Geospatial properties. The expected output for this setting should include both **latitude (lat)** and **longitude (lon)**. The following figure shows the expected output for normal operation.



**Expected behavior for geospatial properties in the Field wells architecture**

3. If only **latitude (lat)** is displayed, you can correct the error by removing the property and manually adding **latitude (lat)** and **longitude (lon)**.
  1. From the **Geospatial** box, select the arrow to the right of **lat**, then choose **Remove**.
  2. Drag the **geo-coordinates** column group (including **lat** and **lon**) from the **Fields List** into the **Field Wells**. For more details, refer to [Using Visual Field Controls](#) in the *Amazon QuickSight User Guide*.

## Post-deployment configuration

This section provides optional recommendations for configuring the solution after deployment.

### Update Data Visualization timeframe

The solution by default displays data with **created\_date** within last 45 days from current date. Follow this procedure to visualize data with **created\_date** older than 45 days in QuickSight.

1. In the [AWS Glue console](#), edit the `projection.created_at.range` property for the table corresponding to your ingestion method. (Find this under the **Advanced properties** section in the **Table** overview).
2. Set the property value to the required date range. (Default value of 45 days corresponds to `'NOW-45DAYS, NOW'`).
3. Go to Amazon QuickSight, edit the dataset to update the date interval within the SQL query.



# Monitor the solution with Service Catalog AppRegistry

This solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both [Service Catalog AppRegistry](#) and [AWS Systems Manager Application Manager](#).

AWS Systems Manager Application Manager gives you an application-level view into this solution and its resources so that you can:

- Monitor its resources, costs for the deployed resources across stacks and AWS accounts, and logs associated with this solution from a central location.
- View operations data for the resources of this solution (such as deployment status, CloudWatch alarms, resource configurations, and operational issues) in the context of an application.

The following figure depicts an example of the application view for the solution stack in Application Manager.

The screenshot displays the AWS Systems Manager Application Manager console. On the left, a sidebar shows a list of components under 'Components (2)', including 'AWS-Systems-Manager-Application-Manager' and 'AWS-Systems-Manager-A'. The main content area is titled 'AWS-Systems-Manager-Application-Manager' and features a 'Start runbook' button. Below the title is the 'Application information' section, which includes fields for 'Application type' (AWS-AppRegistry), 'Name' (AWS-Systems-Manager-Application-Manager), and 'Application monitoring' (Not enabled). A 'View in AppRegistry' link is also present. A navigation bar below this section includes tabs for Overview, Resources, Instances, Compliance, Monitoring, OpsItems, Logs, Runbooks, and Cost. The 'Overview' tab is active, showing 'Insights and Alarms' and 'Cost' sections. The 'Insights and Alarms' section includes a 'View all' button and a description: 'Monitor your application health with Amazon CloudWatch.' The 'Cost' section includes a 'View all' button and a description: 'View resource costs per application using AWS Cost Explorer.' Below the 'Cost' section, there is a 'Cost (USD)' field.

## Solution stack in Application Manager

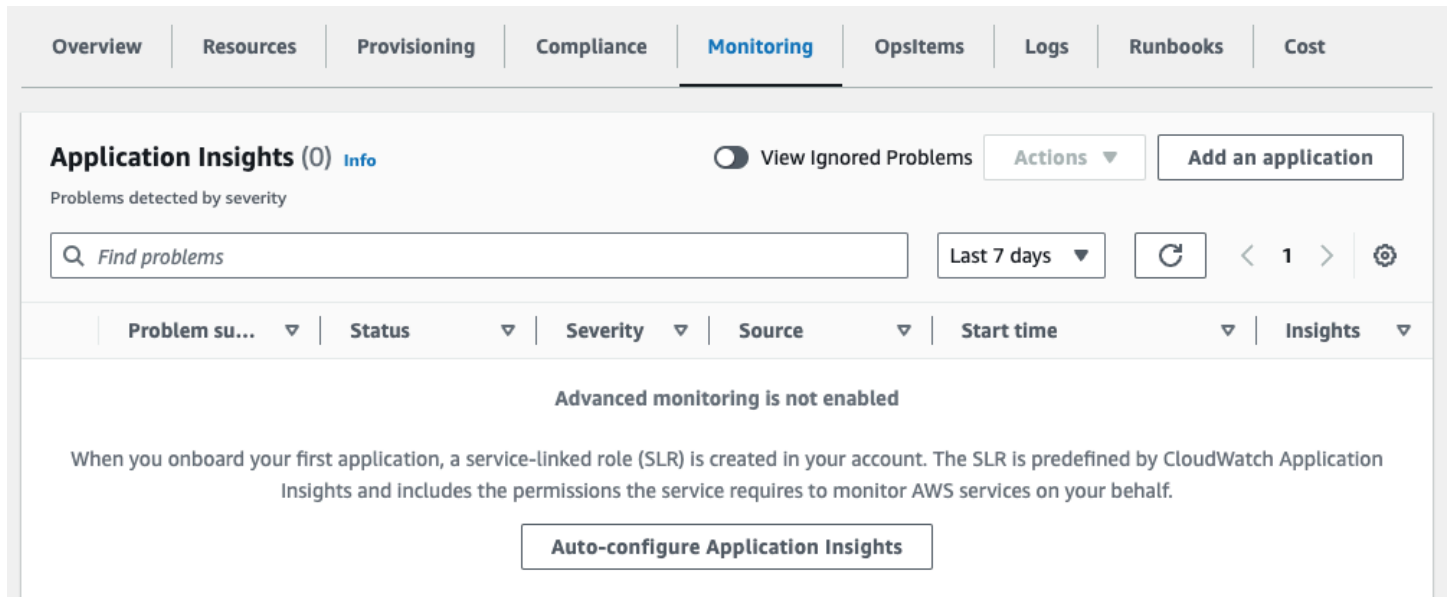
## Activate CloudWatch Application Insights

1. Sign in to the [Systems Manager console](#).

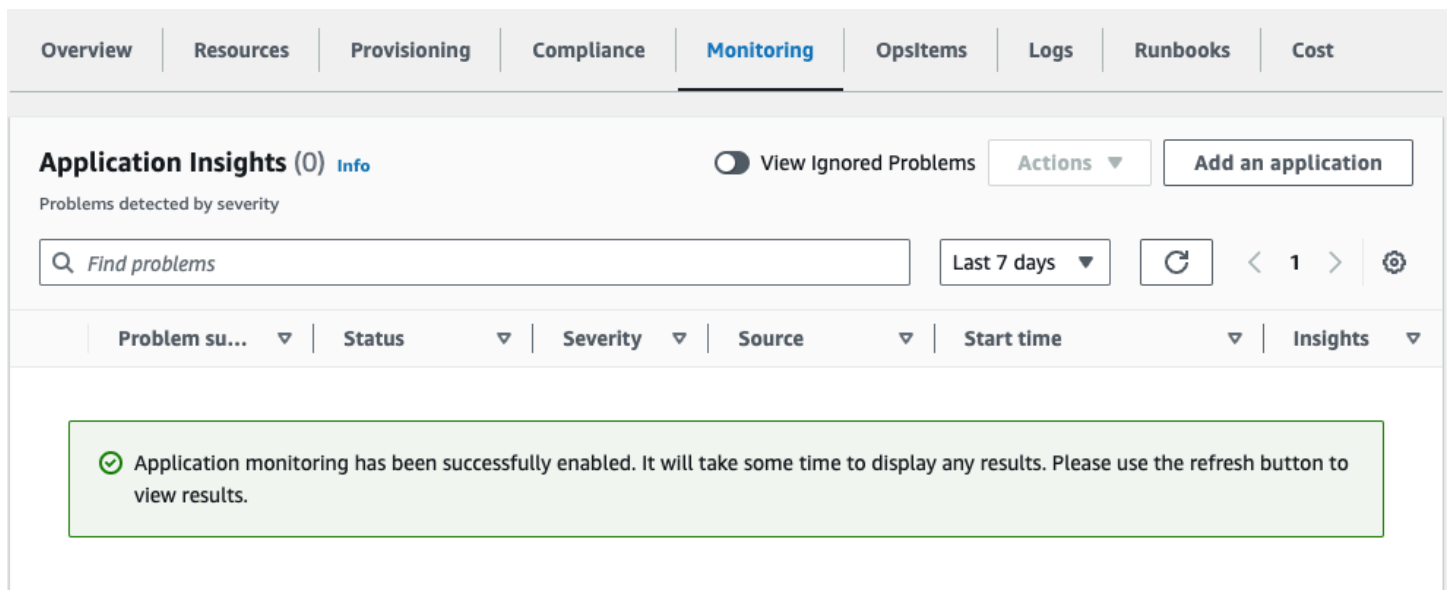
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, search for the application name for this solution and select it.

The application name will have App Registry in the **Application Source** column, and will have a combination of the solution name, Region, account ID, or stack name.

4. In the **Components** tree, choose the application stack you want to activate.
5. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Insights**.



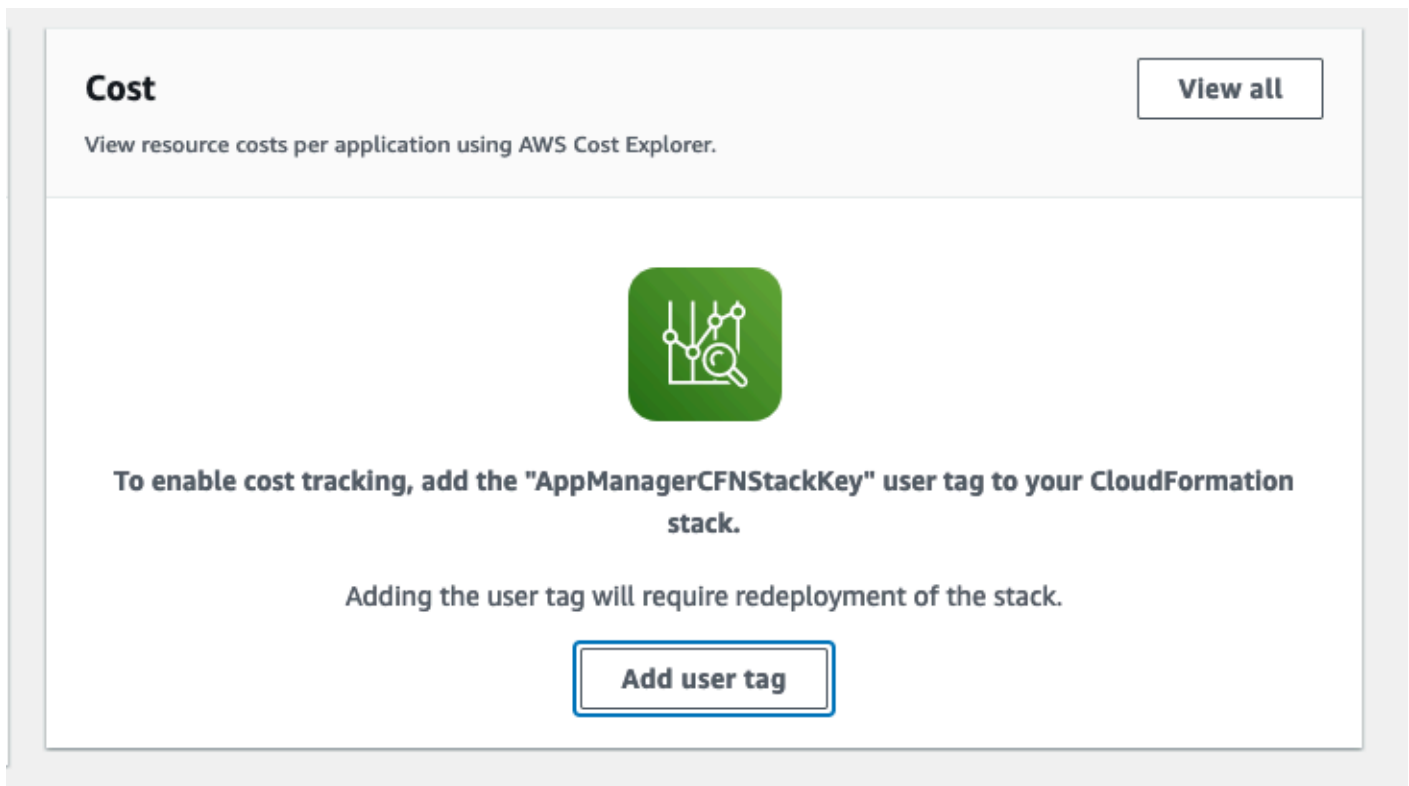
Monitoring for your applications is now activated and the following status box appears:



## Confirm cost tags associated with the solution

After you activate cost allocation tags associated with the solution, you must confirm the cost allocation tags to see the costs for this solution. To confirm cost allocation tags:

1. Sign in to the [Systems Manager console](#).
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose the application name for this solution and select it.
4. In the **Overview** tab, in **Cost**, select **Add user tag**.



5. On the **Add user tag** page, enter `confirm`, then select **Add user tag**.

The activation process can take up to 24 hours to complete and the tag data to appear.

## Activate cost allocation tags associated with the solution

After you confirm the cost tags associated with this solution, you must activate the cost allocation tags to see the costs for this solution. The cost allocation tags can only be activated from the management account for the organization.

To activate cost allocation tags:

1. Sign in to the [AWS Billing and Cost Management and Cost Management console](#).
2. In the navigation pane, select **Cost Allocation Tags**.
3. On the **Cost allocation tags** page, filter for the AppManagerCFNStackKey tag, then select the tag from the results shown.
4. Choose **Activate**.

## AWS Cost Explorer

You can see the overview of the costs associated with the application and application components within the Application Manager console through integration with AWS Cost Explorer. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time.

1. Sign in to the [AWS Cost Management console](#).
2. In the navigation menu, select **Cost Explorer** to view the solution's costs and usage over time.

## Update the solution

To update from a previous version (before version 2.0.0), complete the following steps to update your AWS CloudFormation stack to the current version.

1. Sign in to the [AWS CloudFormation console](#), select your existing CloudFormation stack.

The solution's inferences are stored in the `<old-stackname>-infoutput-<uuid>` bucket. The data in this bucket will be copied to a new location.

2. Delete the existing CloudFormation stack. Do not delete the Amazon S3 buckets. Buckets created by an older version of this solution are configured with Deletion Policy as **Retain** by default. Hence deleting the existing stack will not delete the buckets.
3. After the existing stack has been deleted successfully, deploy the new version of the stack.
4. After the new stack is successfully deployed, using either the AWS CLI or the AWS Management Console, copy the data inside the `<old-stackname>-infoutput-<uuid>` bucket to `<new-stackname>-infoutput-<uuid>`.

# Troubleshooting

If these instructions don't address your issue, see the [Contact AWS Support](#) section for instructions on opening an AWS Support case for this solution.

## Amazon QuickSight nested stack failures

This solution requires that the `aws-quicksight-service-role` IAM Role exists in your account. This role has IAM policies associated with it that allow it write to RDS and Redshift Spectrum. Absence of this IAM Role or its associated policies causes failure to create Amazon QuickSight resources, and cascades as CloudFormation template failures.

To mitigate this error, refer to [How do I troubleshoot AWS resource permission errors in Amazon QuickSight?](#) After following that procedure, use the following script to test data source creation using the AWS CLI.

```
cat tmp/datasource.json
{
  "AwsAccountId": "<account-id>",
  "DataSourceId": "my_test_data_source_id",
  "Name": "my_test_data_source_name",
  "Type": "ATHENA",
  "DataSourceParameters": {
    "AthenaParameters": {
      "WorkGroup": "primary"
    }
  },
  "SslProperties": {
    "DisableSsl": false
  }
}

aws quicksight create-data-source --region <aws-region> \
  --cli-input-json file:///tmp/datasource.json
```

The `<account-id>` should be replaced by the AWS Account ID where you want to deploy the solution, and `<aws-region>` should be replaced with an AWS Region name (example: `us-east-1`).

## Dead-letter-queue for failed ingestion events

The solution deploys a dead-letter-queue (DLQ) for failed ingestion events. You can use the DLQ to troubleshoot any records that failed ingestion in the Amazon Kinesis Data Streams.

### Contact AWS Support

If you have [AWS Developer Support](#), [AWS Business Support](#), or [AWS Enterprise Support](#), you can use the Support Center to get expert assistance with this solution. The following sections provide instructions.

#### Create case

1. Sign in to [Support Center](#).
2. Choose **Create case**.

#### How can we help?

1. Choose **Technical**.
2. For **Service**, select **Solutions**.
3. For **Category**, select **Other Solutions**.
4. For **Severity**, select the option that best matches your use case.
5. When you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions. If you can't resolve your question with these links, choose **Next step: Additional information**.

#### Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail.
3. Choose **Attach files**.
4. Attach the information that AWS Support needs to process the request.

## Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

## Solve now or contact us

1. Review the **Solve now** solutions.
2. If you can't resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.



## Uninstall the solution

You can uninstall the Discovering Hot Topics Using Machine Learning solution from the AWS Management Console or by using the AWS Command Line Interface. You must manually delete the Amazon Simple Storage Service (Amazon S3) buckets and DynamoDB table created by this solution. AWS Solutions Implementations do not automatically delete these resources in case you have stored data to retain.

### Note

The Amazon S3 buckets, and the DynamoDB table, are configured with the retention policy set to **Retain**. You must manually delete them.

## Using the AWS Management Console

1. Sign in to the [AWS CloudFormation console](#).
2. Select this solution's installation stack.
3. Choose **Delete**.

## Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to [What Is the AWS Command Line Interface](#) in the *AWS CLI User Guide*. Optionally, you can use the [AWS CloudShell](#) service to run AWS CLI commands. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

### Note

The Amazon S3 buckets, and the DynamoDB table, are configured with the retention policy set to **Retain**. You must manually delete them.

**⚠ Important**

If you are upgrading this solution from a version released before v1.4.0, you must manually schedule your AWS KMS key for deletion. Refer to the [Deleting AWS KMS keys](#) topic in the *AWS Key Management Service Developer Guide* for more details.

# Developer guide

## Source code

Visit our [GitHub repository](#) to download the templates and scripts for this solution, and to share your customizations with others. If you require an earlier version of the CloudFormation template, you can request from the [GitHub issues page](#). The Discovering Hot Topics Using Machine Learning templates are generated using the [AWS Cloud Development Kit \(AWS CDK\)](#). Refer to the README .md file for more information.

## Supplemental topics

### Retrieve the Amazon QuickSight Principal ARN

To retrieve the Amazon QuickSight User Principal ARN, you must have access to a shell or terminal with the AWS CLI installed. For installation instructions, refer to [What Is the AWS Command Line Interface](#) in the *AWS CLI User Guide*. Optionally, you can use the [AWS CloudShell](#) service to run AWS CLI commands.

Running the following command returns the list of users with their corresponding QuickSight User ARNs.

```
aws quicksight list-users --region <aws-region> --aws-account-id <account-id> --  
namespace <namespace-name>
```

#### Note

The `<namespace-name>` is default, unless explicitly created in Amazon QuickSight.

Choose an **Admin** user, or a user who has permissions to create QuickSight resources in that account and AWS Region.

### Retrieve and manage API Key for YouTube API authentication

You must create a [Google Cloud Platform](#) (GCP) account to access YouTube APIs. After creating a GCP account, you can use the following procedure to retrieve and manage YouTube API.

#### Note

We strongly recommend that you secure your GCP account with Multi-Factor Authentication (MFA) and any other security best practices recommended by GCP.

1. Log in to the GCP console [create a project](#). We recommend creating a unique project for this solution rather than using an existing project, which will allow you to have better control on API Keys and API access.

2. Select your project and select **API and Services** from the left navigation menu.
3. Choose **Enable APIs and Services**. Search for YouTube Data API v3 and select this API option. On the next page, select **Enable** to turn on this API.
4. From the **API and Services** left navigation menu, select **Credentials** and create a new API Key. Restrict the API Key for use with YouTube Data API v3.
5. Copy the new key and store it in AWS Systems Manager Parameter Store under the key path you configured during deployment.

The `Credentials` section provides additional options to regenerate, delete, or revoke access for API keys.

## Retrieve and manage API credentials for Reddit API authentication

The Reddit ingestion uses `clientId`, `clientSecret`, `refreshToken`, and `userAgent`. The `userAgent` is generated dynamically using the deployed stack name. The `clientId`, `clientSecret`, and `refreshToken` should be stored in the AWS Systems Manager Parameter Store as a JSON string. A sample JSON string is as below.

```
{"clientId": "clientIdFromReddit", "clientSecret":  
"clientSecretFromReddit", "refreshToken": "generatedRefreshToken"}
```

Use the Reddit app to retrieve your `clientId` and `clientSecret`. If you don't have them, [sign up for a Reddit app](#).

reddit-client-id-secret

The screenshot shows the GitHub OAuth app management interface for an app named "aggregator". The app is a "web app" with a blue diamond icon containing a question mark. The interface includes the following elements:

- secret**: A redacted field with a red line pointing to the label "clientId".
- clientId**: A redacted field with a red line pointing to the label "clientId".
- clientSecret**: A redacted field with a red line pointing to the label "clientSecret".
- name**: A text input field containing "aggregator".
- description**: A large text area.
- about url**: A text input field.
- redirect uri**: A text input field containing "https://not-an-aardvark.github.io/reddit-oauth-helper/".
- developers**: A list of developers including "api-user" and "(that's you!)", each with a "remove" link.
- add developer:**: A text input field.
- Buttons**: "update app" and "delete app".

## Reddit app to retrieve a client ID and client secret

There are various options to retrieve the `refreshToken`. The recommended options are:

- Use the [reddit-oauth-helper script](#) in the solution's GitHub repository and run it locally. This is the simplest option.

—Or—

- Use the [Reddit OAuth Helper tool](#). Select the **Permanent** checkbox to generate the `refreshToken`.

### Note

The `refreshToken` requires only the **read** scope.

### Important

#### Security considerations

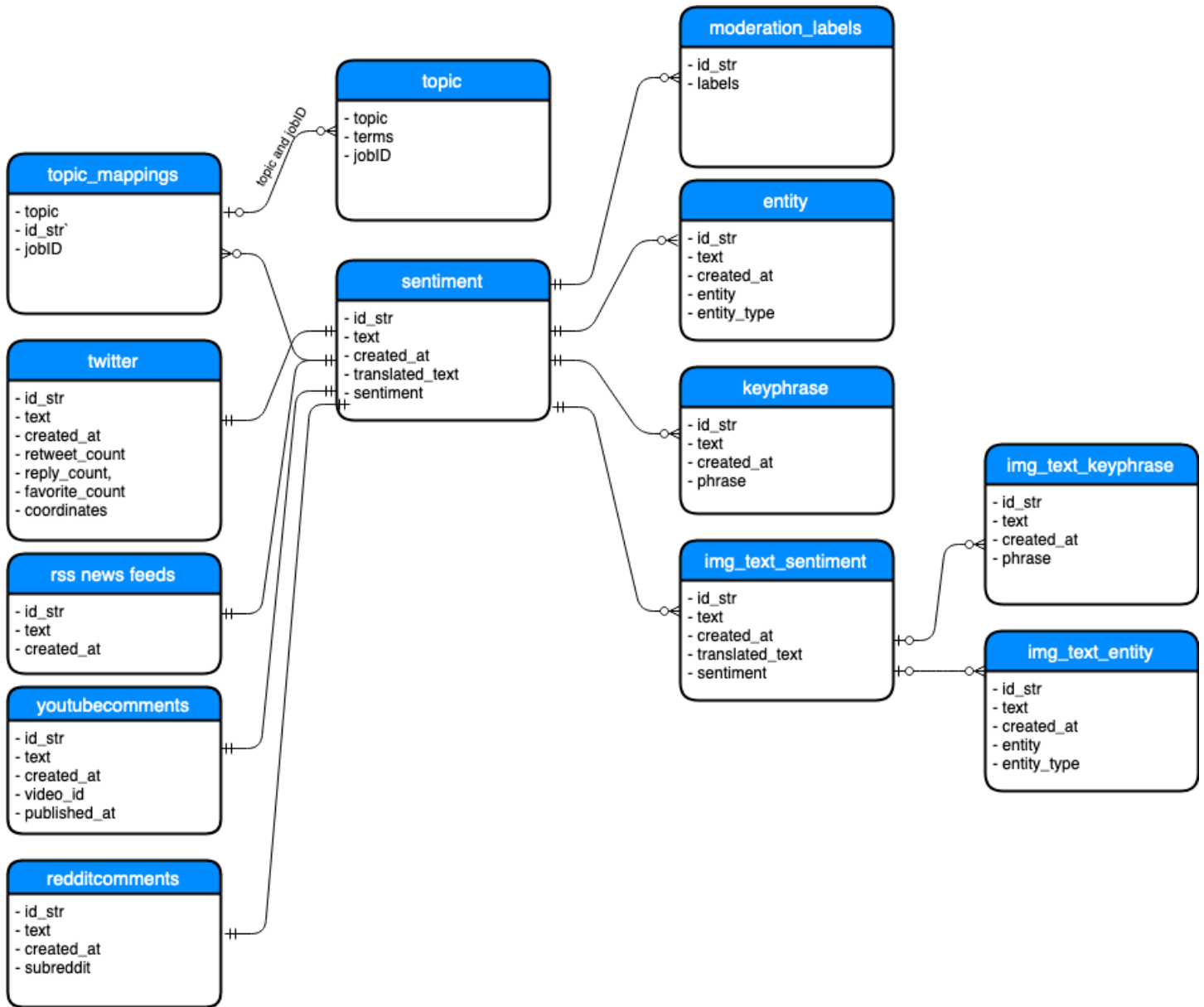
The primary reason for choosing `refreshToken` with `clientId` and `clientSecret` to authenticate with Reddit API is because this option does not require Reddit sign-in credentials.

The generated `refreshToken` can be revoked or the Reddit App deleted so that the solution does not have any access to the Reddit APIs.

We recommended that you at least rotate the `refreshToken` on a regular basis (between every 30-90 days or based on specific Organization's security policy). You must also update the Parameter Store JSON string with every rotation.

## Database schema information

The following diagram displays a high-level schema structure for the various tables created in AWS Glue with their entity relationships. The data model is not normalized and includes redundant attributes for reporting performance.



### Database schema structure



## Reference

This section includes information about an optional feature for collecting unique metrics for this solution, pointers to related resources, and a list of builders who contributed to this solution.

### Anonymized data collection

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When activated, the following information is collected and sent to AWS:

- **Solution ID** – The AWS solution identifier
- **Unique ID (UUID)** – Randomly generated, unique identifier for each deployment of Discovering Hot Topics Using Machine Learning
- **Timestamp** – Data-collection timestamp
- **Data** – Nested structure containing the following information: timestamp
  - **Region** – The AWS Region in which the solution is deployed
  - **RequestType** – Create, Update, or Delete
  - **TopicJobFreq** – The frequency configured for Topic Modeling jobs
  - **NewsFeedsIngestionEnabled** – Yes or No based on the **DeployNewsFeeds** CloudFormation parameter selected when creating or updating a stack
  - **NewsFeedsSearchComplexity** – The count of comma-separated phrases that filter news feed information
  - **NewsFeedsSearchQueryLength** – The string length of the comma-separated phrases that filter news feed information
  - **NewsFeedsIngestionFreq** – The ingestion frequency configured to pull news feeds from websites
  - **YoutubeIngestionEnabled** – Yes or No based on the **DeployYouTubeCommentsIngestion** CloudFormation parameter selected when creating or updating a stack
  - **YouTubeIngestionFreq** – The ingestion frequency configured for YouTube
  - **YouTubeSearchQueryLength** – The length of phrases used to query the YouTube API
  - **YouTubeChannelIDSet** – `True` if the YouTube channel ID is set, or `False` if the YouTube channel ID is not set

- **RedditIngestionEnabled** – Yes or No based on the **DeployRedditIngestion** CloudFormation parameter selected when creating or updating a stack
- **RedditIngestionFreq** – The ingestion frequency configured to pull comments from Reddit
- **RedditIngestionSubredditCount** – Number of subreddits configured for Reddit comment ingestion
- **CustomIngestionEnabled** – Yes or No based on the **DeployCustomIngestion** CloudFormation parameter selected when creating or updating a stack.

AWS owns the data gathered through this survey. Data collection is subject to the [AWS Privacy Policy](#). To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the [AWS CloudFormation template](#) to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

```
"Send" : {  
  "AnonymousUsage" : { "Data" : "Yes" }  
},
```

to:

```
"Send" : {  
  "AnonymousUsage" : { "Data" : "No" }  
},
```

4. Sign in to the [AWS CloudFormation console](#).
5. Select **Create stack**.
6. On the **Create stack** page, **Specify template** section, select **Upload a template file**.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in [Launch the stack](#) in the Automated Deployment section of this guide.

## Contributors

- Nihit Kasabwala
- Tarek Abdunabi
- Mukta Dadariya
- Manish Jangid
- Abhishek Patil

# Revisions

Date	Change
August 2020	Initial release
September 2020	Updated documentation to add details to the Overview section and numbered callouts to the architecture diagram.
September 2020	Updated documentation to support v1.1.0. For more information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
October 2020	Updated documentation to support v1.2.0. For more information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
November 2020	Updated documentation to support v1.3.0. For more information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
February 2021	Updated documentation to support v1.4.0. Added documentation about using the Twitter Search API with geo-coordinates. For more information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
July 2021	Updated documentation to support v1.5.0. Added documentation about new source of ingestion (RSS feeds). For more information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
September 2021	Updated documentation to support v1.6.0. Added documentation about new source of ingestion (YouTube comments). For more

Date	Change
	information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
October 2021	Updated documentation to support v1.6.1. This release includes the fix for <a href="#">GitHub issue #42</a> , and library updates to the AWS CDK and AWS SDK. For more information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
February 2022	Updated documentation to support v1.7.0. This release includes a new feature to ingest data from an Amazon S3 bucket, uses the Amazon Data Firehose dynamic partitioning feature to create partitions in S3, replaces Glue Partitions with Athena partition projections, and library updates to the AWS CDK and AWS SDK. For more information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
May 2022	Updated documentation to support v2.0.0. This release includes a new feature to ingest data from Reddit and library updates to the AWS CDK and AWS SDK. For more information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
November 2022	Updated documentation to support v2.0.1. This release is a fix for Github #69 and library updates to the AWS CDK and AWS SDK. For more information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.

Date	Change
December 2022	Updated documentation to support v2.1.0. This release includes integration of this solution with Service Catalog AppRegistry and AWS Systems Manager Application Manager. For more information, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
December 2022	Updated documentation to support v2.1.1. Removed "AWS" prefix from Service Catalog AppRegistry and Attribute Group Name to correct a common error. For more information about the changes, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
January 2023	Updated documentation to support v2.1.2. This release includes updates to AWS CDK, AWS SDK, Nodejs and Python libraries. For more information about the changes, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
April 2023	Updated documentation to support v2.1.3. Updates for bucket policy on the logging bucket to grant access to the logging service principal (logging.s3.amazonaws.com) for access log delivery.
June 2023	Updated documentation to support v2.1.4. This release includes library version updates and security patches. For more information about the changes, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.

Date	Change
September 2023	Updated documentation to support v2.2.0. Updated operational metrics information. This release includes the following: updates to AWS CDK and AWS SDK, library version updates, security patches, update to Reddit comment ingestion to use Python PRAW library, implementing NewsCatcher locally, disabling Twitter ingestion temporarily, and bug fixes. For more information about the changes, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
October 2023	Updated documentation to support v2.2.1. This release includes a security patch. For more information about the changes, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.
November 2023	Documentation update: Added <a href="#">Confirm cost tags associated with the solution</a> to the Monitoring the solution with Service Catalog AppRegistry section.
February 2024	Updated documentation to the latest template.
March 2024	Updated documentation to support v2.3.0 and removed Twitter references. This release includes security patches, minor bug fixes, and updates to QuickSight template to remove Twitter references. For more information about the changes, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.

Date	Change
July 2024	Updated documentation to support v2.3.1. This release includes library version updates and security patches. For more information about the changes, refer to the <a href="#">CHANGELOG.md</a> file in the GitHub repository.



## Notices

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 AWS current 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. AWS responsibilities and liabilities 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.

Discovering Hot Topics Using Machine Learning is licensed under the terms of the of the Apache License Version 2.0 available at [The Apache Software Foundation](https://www.apache.org/licenses/LICENSE-2.0).