Discovering Hot Topics
Using Machine Learning
Implementation Guide
# Table of Contents

Welcome ................................................................................................................................. 1
Cost ........................................................................................................................................... 3
  Example cost tables ............................................................................................................. 4
Architecture overview ............................................................................................................ 10
  Operational excellence ....................................................................................................... 13
  Security ............................................................................................................................... 13
  Reliability .......................................................................................................................... 13
  Performance efficiency ...................................................................................................... 14
  Cost optimization .............................................................................................................. 14
Pre-built Amazon QuickSight dashboard details ................................................................. 14
Example use cases for Amazon QuickSight .......................................................................... 19
  ................................................................. 19
  ...................................................................................................................................... 20
  ...................................................................................................................................... 21
  ...................................................................................................................................... 22
  ...................................................................................................................................... 23
Customizing S3 ingestion ...................................................................................................... 25
  Microsoft Excel files ......................................................................................................... 13
  JSON files ......................................................................................................................... 13
  Transcribe Call Analytics ................................................................................................. 13
Security .................................................................................................................................. 13
  IAM roles .......................................................................................................................... 29
  Amazon S3 ....................................................................................................................... 29
  Twitter credentials .......................................................................................................... 29
  YouTube credentials ........................................................................................................ 29
Design considerations .......................................................................................................... 30
  Regional deployments ..................................................................................................... 30
AWS CloudFormation template ............................................................................................ 31
Automated deployment .......................................................................................................... 32
  Prerequisites .................................................................................................................... 32
  Launch the stack ............................................................................................................. 32
  Configure Amazon QuickSight ....................................................................................... 37
Resources ............................................................................................................................... 40
Update the stack .................................................................................................................... 41
Retrieve a bearer token for Twitter API authentication ....................................................... 42
Using the Twitter search API ............................................................................................... 43
Twitter API limit throttling .................................................................................................... 44
Retrieve the Amazon QuickSight Principal ARN ................................................................ 45
Retrieve and manage API Key for YouTube API authentication .......................................... 46
Database schema information ............................................................................................... 47
Uninstall the solution ............................................................................................................ 48
  Using the AWS Management Console ............................................................................ 48
  Using AWS Command Line Interface ............................................................................. 48
Troubleshooting ................................................................................................................... 49
  Amazon QuickSight nested stack failures ..................................................................... 49
  Dead Letter Queue for failed ingestion events ............................................................... 49
Operational metrics .............................................................................................................. 50
Source code .......................................................................................................................... 52
Revisions ............................................................................................................................... 53
Contributors ......................................................................................................................... 55
Notices ....................................................................................................................................... 56
AWS glossary ......................................................................................................................... 57
Deploy a solution that helps you identify the most dominant topics associated with your products, policies, events, and brands

Publication date: August 2020 (last update: March 2022)

The Discovering Hot Topics Using Machine Learning solution helps you identify the most dominant topics associated with your products, policies, events, and brands. Implementing this solution helps you react quickly to new growth opportunities, address negative brand associations, and deliver higher levels of customer satisfaction.

The solution automates digital asset ingestion from Twitter, RSS news feeds, YouTube comments, and custom data ingestion to provide near real-time inferences using machine learning (ML) algorithms through Amazon Comprehend, Amazon Translate, and Amazon Rekognition to perform topic modeling, sentiment analysis, entity and key phrase detection, and detect any unsafe images. Custom data ingestion allows ingestion of JSON and XLSX formats uploaded to an Amazon Simple Storage Service (Amazon S3) bucket. With this feature, the 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 Reddit, Twitch, and Discord. This is done by exporting data as JSON or spreadsheet (XLSX) files and uploading them to the bucket created when deploying this solution. The solution then visualizes these large-scale customer analyses using an Amazon QuickSight dashboard. This guide provides step-by-step instructions for deploying this solution including a pre-built dashboard that provides you with the context and insights necessary to identify trends that help or harm your brand.

This solution provides the following key features:

- **Performs topic modeling to detect dominant topics**: identifies the terms that collectively form a topic from within customer feedback
- **Identifies the sentiment of what customers are saying**: uses contextual semantic search to understand the nature of online discussions
- **Determines if images associated with your brand contain unsafe content**: detects unsafe and negative imagery in content
- **Helps you identify insights in near real-time**: uses a visual dashboard to understand context, threats, and opportunities almost instantly

This solution deploys an AWS CloudFormation template that supports Twitter, YouTube comments, RSS feeds, and custom ingestion through an Amazon S3 bucket as data source options for ingestion, but 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 S3 bucket to allow Amazon Transcribe Call Analytics output to be processed for natural language processing (NLP) analysis.

After you deploy the solution, use the included Amazon QuickSight dashboard to visualize the solution’s ML inferences.
Important
For a successful implementation with Twitter as the source, you must create a Twitter account and add an app through the Twitter developer portal and apply for elevated access.

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 in an enterprise by IT infrastructure architects, administrators, and DevOps professionals who have practical experience with the AWS Cloud.
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 — Twitter, RSS feeds, and/or YouTube comments).

When invoking the Search API for Twitter, the search query string and the language filter both contribute to the volume of feeds 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 in a smaller volume of ingested data.

- **Language filter**: The broader list of languages (for example, de, en, es, it, pt, fr, ja, ko, hi, ar, zh-cn, zh-tw) can potentially result in a larger volume of ingested data. Narrow language filters (for example, only en, fr, es) will result in a smaller volume of ingested 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 in 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, and in some cases tweets, which do not have associated languages. When this occurs, 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 tweets and/or RSS feeds.

As of March 2022, the cost for running this solution in the US East (N. Virginia) Region using Twitter, 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.
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 and 4,000 tweets a week**

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Athena</td>
<td>70 queries/week and 100 GB data scanned/query</td>
<td>$40.00/week</td>
</tr>
<tr>
<td>Amazon CloudWatch Event Rule – 1</td>
<td>5,040 events/week (runs every 2 mins)</td>
<td>$0.01/week</td>
</tr>
<tr>
<td>Amazon CloudWatch Event Rule – 2</td>
<td>14 events/week (runs every day)</td>
<td>$0.000084/week</td>
</tr>
<tr>
<td>Amazon Comprehend (Pre-trained, Sync APIs)</td>
<td>3 Sync APIs, 280 characters long, 400,000 (tweets) requests/API + news feeds</td>
<td>$120.00/week</td>
</tr>
<tr>
<td>Amazon Comprehend (Pre-trained, Topic Modeling Job)</td>
<td>7 Twitter jobs/week + 7 news feed jobs with 25 KB data/job</td>
<td>~$14.00/week</td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td>Records inserted with TTL 7-day expiry to keep state for Twitter search and news feeds, on-demand capacity</td>
<td>~$0.16/week</td>
</tr>
<tr>
<td>Amazon Kinesis Data Firehose</td>
<td>13 Firehose, total 3 GB/week</td>
<td>$0.26/week</td>
</tr>
<tr>
<td>Amazon Kinesis Data Streams</td>
<td>4,000 records/week, 80 MB + news feeds</td>
<td>$2.52/week</td>
</tr>
<tr>
<td>Amazon Simple Queue Service (Amazon SQS)</td>
<td>14 queues (regular queues + DLQ) processing news feed information + 4000 tweets every week</td>
<td>$0.67/week</td>
</tr>
<tr>
<td>Amazon Rekognition – Label Detection</td>
<td>4,000 tweets, 60% of tweets have images = 2,400 tweets 2 images a tweet on average = 4,800 images + images associated with news feeds</td>
<td>$24.00/week</td>
</tr>
<tr>
<td>Amazon Rekognition – Text Detection</td>
<td>4,000 tweets, 60% of tweets have images = 2,400 tweets 2 images a tweet on average = 4,800 images + images associated with news feeds</td>
<td>$24.00/week</td>
</tr>
<tr>
<td>Amazon Simple Storage Service (Amazon S3)</td>
<td>5 buckets, 75 GB</td>
<td>~$5.00/week</td>
</tr>
<tr>
<td>Amazon Translate</td>
<td>280 characters/tweet with 4,000 tweets, assumes 50% of 4,000 tweets in non-English language</td>
<td>$70.00/week</td>
</tr>
</tbody>
</table>
Discovering Hot Topics Using Machine Learning Implementation Guide
Example cost tables

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost</th>
</tr>
</thead>
</table>
| AWS Lambda (128 MB)                  | • 128 MB - 15 functions (total of 89,529 invocations/week and duration of 4,739 millisecond/week)  
• 256 MB - 5 functions (total of 31,238 invocations/week and duration of 2,644 millisecond/week)  
• 512 MB - 2 function (total of 84 invocations/week and duration of 717 milliseconds/week) | ~$1.30/week |
| AWS Step Functions                   | 2 workflow definitions ~ 30 states (in all)                                  | $21.00/week |
| AWS Key Management Service           | Using AWS managed keys with DynamoDB, S3 (SSE-S3), Kinesis Data Streams, SQS | $5.00/week |
| **Total**                            |                                                                           | ~$330.00/week |

**Example 2 – Ingesting 4,000 tweets per week**

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Athena</td>
<td>70 queries/week and 100 GB data scanned/query</td>
<td>$40.00/week</td>
</tr>
<tr>
<td>Amazon CloudWatch Events – 1</td>
<td>5,040 events/week (run every 2 mins)</td>
<td>$0.01/week</td>
</tr>
<tr>
<td>Amazon CloudWatch Events – 2</td>
<td>7 events/week (run every 2 hours)</td>
<td>$0.000084/week</td>
</tr>
<tr>
<td>Amazon Comprehend (Pre-trained, Sync APIs)</td>
<td>3 Sync APIs, 280 characters long, 4,000 (tweets) requests/API</td>
<td>$1.20/week</td>
</tr>
<tr>
<td>Amazon Comprehend (Pre-trained, Topic Modeling Job)</td>
<td>7 jobs/week + 25 KB data/job</td>
<td>~$7.00/week</td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td>Records inserted with TTL 7-day expiry, on-demand capacity</td>
<td>~$0.08/week</td>
</tr>
<tr>
<td>Amazon Kinesis Data Firehose</td>
<td>13 Firehose, total 3 GB/week</td>
<td>~$0.26/week</td>
</tr>
<tr>
<td>Amazon Kinesis Data Streams</td>
<td>4,000 records/week, 4 GB</td>
<td>$1.00/week</td>
</tr>
<tr>
<td>Amazon Simple Queue Service (Amazon SQS)</td>
<td>14 queues (regular queues + DLQ) processing + 4,000 tweets every week</td>
<td>$0.67/week</td>
</tr>
</tbody>
</table>
### Example cost tables

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Rekognition – Label Detection</td>
<td>4,000 tweets, assume 60% of tweets have images = 2,400 tweets = 2 images a tweet on average = 4,800 images</td>
<td>$4.80/week</td>
</tr>
<tr>
<td>Amazon Rekognition – Text Detection</td>
<td>4,000 tweets, 60% of tweets have images = 2,400 tweets = 2 images a tweet on average = 4,800 images</td>
<td>$4.80/week</td>
</tr>
<tr>
<td>Amazon Simple Storage Service (Amazon S3)</td>
<td>5 buckets, 75 GB</td>
<td>~$5.00/week</td>
</tr>
<tr>
<td>Amazon Translate</td>
<td>280 characters/tweet with 4,000 tweets, assumes 50% of 4,000 tweets in non-English language = 2,000 tweets = 560,000 characters</td>
<td>$8.40/week</td>
</tr>
</tbody>
</table>
| AWS Lambda (128 MB)                              | • 128 MB - 15 functions (total of 89,529 invocations/week and duration of 4,739 millisecond/week)  
   • 256 MB - 5 functions (total of 31,238 invocations/week and duration of 2,644 millisecond/week)  
   • 512 MB - 2 function (total of 84 invocations/week and duration of 717 milliseconds/week) | ~$1.30/week |
| AWS Step Functions                               | 2 workflow definitions ~ 30 states (in all)                                | $2.00/week |
| AWS Key Management Service                       | Using AWS managed keys with DynamoDB, S3 (SSE-S3), Kinesis Data Streams, SQS | $5.00/week |
| **Total:**                                       |                                                                          | ~$100.00/week |

#### Example 3 – Ingesting 400,000 tweets per week

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Athena</td>
<td>70 queries/week and 100 GB data scanned/query</td>
<td>$40.00/week</td>
</tr>
<tr>
<td>Amazon CloudWatch Events – 1</td>
<td>5,040 events/week (trigger every 2 mins)</td>
<td>$0.01/week</td>
</tr>
<tr>
<td>Amazon CloudWatch Events – 2</td>
<td>7 events/week (runs every day)</td>
<td>$0.000084/week</td>
</tr>
<tr>
<td>Amazon Comprehend (Pre-trained, Sync APIs)</td>
<td>3 Sync APIs, 280 characters long, 400,000 (tweets) requests/API</td>
<td>$150.00/week</td>
</tr>
</tbody>
</table>

---

6
### Example cost tables

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Comprehend (Pre-trained, Topic Modeling Job)</td>
<td>7 jobs/week + 1 MB data/job</td>
<td>~$7.00/week</td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td>Records inserted with TTL 7-day expiry, on-demand capacity</td>
<td>~$0.08/week</td>
</tr>
<tr>
<td>Amazon Kinesis Data Firehose</td>
<td>13 Firehose, total 40 GB/week</td>
<td>~$1.50/week</td>
</tr>
<tr>
<td>Amazon Simple Queue Service</td>
<td>14 queues (regular queues + DLQ) processing news feed information + 4,000 tweets every week</td>
<td>~$0.67/week</td>
</tr>
<tr>
<td>Amazon Kinesis Data Streams</td>
<td>400,000 records/week, 4 GB</td>
<td>$11.12/week</td>
</tr>
<tr>
<td>Amazon Rekognition – Label Detection</td>
<td>400,000 tweets, assumes 60% of tweets have images = 240,000 tweets = 2 images a tweet on average = 480,000 images</td>
<td>$480.00/week</td>
</tr>
<tr>
<td>Amazon Rekognition – Text Detection</td>
<td>400,000 tweets, assumes 60% of tweets have images = 240,000 tweets = 2 images a tweet on average = 480,000 images</td>
<td>$480.00/week</td>
</tr>
<tr>
<td>Amazon Simple Storage Service (Amazon S3)</td>
<td>5 buckets, 750 GB</td>
<td>~$20.00/week</td>
</tr>
<tr>
<td>Amazon Translate</td>
<td>280 characters/tweet with 400,000 tweets, assumes 50% tweets are in a non-English language = 200,000 tweets = 56 M characters</td>
<td>$840.00/week</td>
</tr>
<tr>
<td>AWS Lambda (128 MB)</td>
<td>• 128 MB - 15 functions (total of 89,529 invocations/week and duration of 4739 millisecond/week)</td>
<td>~$100.00/week</td>
</tr>
<tr>
<td></td>
<td>• 256 MB - 5 functions (total of 31,238 invocations/week and duration of 2644 millisecond/week)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 512 MB - 512 MB - 2 function (total of 84 invocations/week and duration of 717 milliseconds/week)</td>
<td></td>
</tr>
<tr>
<td>AWS Step Functions</td>
<td>2 workflow definitions ~ 30 states (in all)</td>
<td>$400.00/week</td>
</tr>
<tr>
<td>Total:</td>
<td></td>
<td>~$2,100.00/week</td>
</tr>
</tbody>
</table>

**Example 4:** Ingesting – 4,000 tweets a week + RSS News Feeds + 4,000 YouTube comments
<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Athena</td>
<td>70 queries/week and 100 GB data scanned/query</td>
<td>$40.00/week</td>
</tr>
<tr>
<td>Amazon CloudWatch Events Rule – 1</td>
<td>5,040 events/week (trigger every 2 mins)</td>
<td>$0.01/week</td>
</tr>
<tr>
<td>Amazon CloudWatch Events Rule– 2</td>
<td>14 events/week (runs every day)</td>
<td>$0.000168/week</td>
</tr>
<tr>
<td>Amazon Comprehend (Pre-trained, Sync APIs)</td>
<td>4 Sync APIs, 280 characters long, 400,000 (tweets) requests/API + news feeds + 4000 YouTube comments (which also requires language detection)</td>
<td>$150.00/week</td>
</tr>
<tr>
<td>Amazon Comprehend (Pre-trained, Topic Modeling Job)</td>
<td>7 Twitter jobs/week + 7 news feed jobs with 25 KB data/job + 7 YouTube comments</td>
<td>~$21.00/week</td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td>Records inserted with TTL 7-day expiry to keep state for Twitter search and news feeds, on-demand capacity</td>
<td>~$0.24/week</td>
</tr>
<tr>
<td>Amazon Kinesis Data Firehose</td>
<td>14 Firehose, total 3 GB/week</td>
<td>$0.26/week</td>
</tr>
<tr>
<td>Amazon Simple Queue Service</td>
<td>14 queues (regular queues + DLQ) processing news feed information + 4,000 tweets every week</td>
<td>$0.67/week</td>
</tr>
<tr>
<td>Amazon Kinesis Data Streams</td>
<td>4,000 records/week, 80 MB + news feeds + YouTube comments</td>
<td>~$3.00/week</td>
</tr>
<tr>
<td>Amazon Rekognition – Label Detection</td>
<td>4,000 tweets, 60% of tweets have images = 2,400 tweets = 2 images a tweet on average = 4,800 images + images associated with news feeds</td>
<td>$24.00/week</td>
</tr>
<tr>
<td>Amazon Rekognition – Text Detection</td>
<td>4,000 tweets, 60% of tweets have images = 2400 tweets = 2 images a tweet on average = 4,800 images + images associated with news feeds</td>
<td>$24.00/week</td>
</tr>
<tr>
<td>Amazon Simple Storage Service (Amazon S3)</td>
<td>5 buckets, 75 GB</td>
<td>~$5.00/week</td>
</tr>
<tr>
<td>Amazon Translate</td>
<td>280 characters/tweet with 4,000 tweets, assumes 50% of 4,000 tweets in non-English language = 2000 tweets = 560,000 characters +</td>
<td>$100.00/week</td>
</tr>
<tr>
<td><strong>AWS service</strong></td>
<td><strong>Dimensions</strong></td>
<td><strong>Cost</strong></td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td><strong>news feed from different</strong></td>
<td><strong>Amazon Web Services (AWS)</strong> service dimensions: <strong>AWS Lambda (128 MB)</strong> - 15 functions (total of 89,529 invocations/week and duration of 4739 millisecond/week)</td>
<td>~$1.30/week</td>
</tr>
<tr>
<td><strong>languages translated to English</strong></td>
<td><strong>AWS Lambda (256 MB)</strong> - 5 functions (total of 31,238 invocations/week and duration of 2644 millisecond/week)</td>
<td></td>
</tr>
<tr>
<td><strong>+ ~2,000 YouTube comments</strong></td>
<td><strong>AWS Lambda (512 MB)</strong> - 512 MB - 2 function (total of 84 invocations/week and duration of 717 milliseconds/week)</td>
<td></td>
</tr>
<tr>
<td><strong>AWS Step Functions</strong></td>
<td><strong>2 workflow definitions ~ 30 states (in all)</strong></td>
<td>$21.00/week</td>
</tr>
<tr>
<td><strong>AWS Key Management Store</strong></td>
<td><strong>Using AWS managed keys with DynamoDB, S3 (SSE-S3), Kinesis Data Streams, SQS</strong></td>
<td>$5.00/week</td>
</tr>
<tr>
<td></td>
<td><strong>Total:</strong></td>
<td>~$400.00/week</td>
</tr>
</tbody>
</table>
Architecture overview

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

Figure 1: 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 Kinesis 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 Twitter, YouTube comments, RSS news feeds, and custom ingestion using an Amazon S3 bucket.
Discovering Hot Topics Using Machine Learning Implementation Guide

Figure 2: Ingesting tweets using the Twitter Search API

Figure 3: Ingesting YouTube comments using the YouTube Data API
Discovering Hot Topics Using Machine Learning Implementation Guide

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. **Inference** – AWS machine learning capabilities through Amazon Translate, Amazon Comprehend, and Amazon Rekognition.
6. **Storage and visualization** – A combination of Amazon Kinesis Data Firehose, Amazon S3 buckets, AWS Glue tables, Amazon Athena, and Amazon QuickSight.

These components are built using the AWS Well-Architected Framework, and the AWS Well-Architected Pillars of Operational Excellence, Security, Reliability, Performance Efficiency, and Cost Optimization—ensuring secure, high-performing, resilient, and efficient infrastructure.

**Operational excellence**

This solution's AWS CloudFormation template, was built with the AWS Cloud Development Kit (CDK). The template was built without hard-coding 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 Kinesis Data Firehose provide observability into the infrastructure.

**Security**

This solution implements encryption-at-rest and encryption-in-transit. For encryption-at-rest, 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.

The ingestion component of this solution requires Twitter credentials. The bearer token required to authenticate with Twitter APIs is read in memory through an AWS Systems Manager Parameter Store.

**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 Detect Personally Identifiable Information (PII) 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 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 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

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.

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.

![Figure 6: Controls on the Amazon QuickSight dashboard](image)

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

1. **Select platform**: Allows selection of multiple platforms and filters the data from the deployed platform. The control displays all source options (newsfeed, Twitter, youtubecomments, and customingestion) irrespective of the ingestion option selected when deploying the solution.
2. **Select topic**: Allows selection of multiple topics as detected by the solution.
3. **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.
4. **Search name**: Provides a mechanism to search by user names (for tweets) or topics for (News feeds). This search is an exact search and is case sensitive.
5. **Search source**: This field provides the user ID on twitter (for tweets) or websites path (for news feeds). This search is an exact search and is case sensitive.
The example QuickSight dashboard in Figure 7 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 Figure 7 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.

You can use the dashboard to filter information and gain insights on sentiment context and customer perception. The visuals in the second row in Figure 7 to explore the most dominant topic ‘000’. The second row contains the following visuals:
5. 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).

6. A donut chart representing the sentiment analysis of the selected topic and phrase. Selecting a specific sentiment (in Figure 1: Negative), 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 table (visual #8).

7. A heat map with details of daily tweet 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 table (visual #8).

8. A tabulated view of the records ingested.

---

Figure 8: Example Amazon QuickSight dashboard for text analyses

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). Figure 8 is an included dashboard for text analysis. It 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 geographical map showing the origin of the tweets (applies only to twitter—not RSS feeds and YouTube comments—and when the tweets contain geo-coordinate information). The size of the circle on the map is a count of tweets originating from that location. After selecting the location on the map, the visuals on the page are filtered to display tweets belonging to the specific geo-coordinates. Tweets that do not have public geo-coordinates are also filtered out.

4. A tabular view of ingested data (visual #4) containing the date (Date), the ID (example: tweet ID), the text, and the tweet text translated to English. Selecting a record (row) in the table opens a new browser window that navigates to its source (this could be twitter.com, the news site, or the YouTube video URL).

This solution also uses Amazon Rekognition to analyze images, detect entities in images, 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.
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:

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 tweets related to a brand. 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.

You can use the geo-coordinate information from tweets analyze tweets originating from a location (when geo-coordinates are publicly available). This analysis worksheet can only be used with social media records that contain geo-coordinates.
Discovering Hot Topics Using Machine Learning Implementation Guide
Pre-built Amazon QuickSight dashboard details

Figure 10: Example Amazon QuickSight dashboard to analyze inferences by geo-coordinates

The Analysis by Geography worksheet tab is only applicable to Twitter (because Twitter is the only source that might have geo-coordinates). Selecting circles on the map filters the visuals on the geo-coordinate tab.

1. A geographical map displaying the origin of the tweets (only when the tweets provide public geo-coordinate information). The size of the circle on the map is a count of tweets originating from that location. After selecting the location on the map, the visuals on the page are filtered to display tweets belonging to the specific geo-coordinates. Tweets that do not provide public geo-coordinates are filtered out.

2. A table displaying the 10 most dominant topics in the data set. Selecting the topic filters the rest of the visuals on the tab to display content associated with that topic.

3. A word cloud of terms found in the most dominant topics.

4. A word cloud of phrases. Selecting the phrase filters the visuals on the tab to display content that is relevant to that phrase.

5. A word cloud of entities. Selecting the entity filters the visuals on the tab to display content that is relevant to that entity.
6. Sentiment trend over the last 1 week.
7. Tabular view of tweets. Selecting a record (row) in the table opens a new browser window that navigates to twitter.com and displays the selected tweet. The table displays the retweet count, quote count, reply count, and favorite count at the time when the tweet was ingested (as returned by the Twitter Search API call).

Example use cases for Amazon QuickSight

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

Select Twitter 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 tweets associated with the dominant topics, and a tabular view of related tweets.

<table>
<thead>
<tr>
<th>Field/Label</th>
<th>View</th>
<th>Field/Value</th>
<th>Field/Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field types</td>
<td>Text Analysis</td>
<td>Image Analysis</td>
<td>Topic Analysis</td>
</tr>
<tr>
<td>Filters</td>
<td>Topic (ML)</td>
<td>Sentiment (ML)</td>
<td>Controls</td>
</tr>
<tr>
<td>List of the most dominant...</td>
<td>Overall sentiment</td>
<td>Sentiment</td>
<td>What is the sentiment trend over the last 7 days?</td>
</tr>
<tr>
<td>Denot... Topic ID...</td>
<td>NEUTRAL</td>
<td>POSITIVE</td>
<td>What is the sentiment trend over the last month?</td>
</tr>
<tr>
<td>Parameters</td>
<td>POSITIVE</td>
<td>NEGATIVE</td>
<td></td>
</tr>
<tr>
<td>Actions</td>
<td>NEGATIVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Themes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 11: 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 tweets associated with that selection.
Example 2: View the tweets with negative sentiments.

With Twitter as the source, from the Topic Analysis tab, select Negative. This refreshes the table of tweets in the third row visual.

Figure 12: Controls to multi-select topics or sentiments to filter visuals on the page
Example use cases for Amazon QuickSight

Figure 13: Filter visuals based on the sentiment

Example 3: Select a phrase in the analysis and view the related tweets.

With Twitter as the source, from the Text Analysis tab, choose a phrase from the Phrases and their associated sentiment horizontal stacked bar chart. This filters the table of tweets that contain the selected phrase and the associated sentiment.
Figure 14: Filter tabular view of tweets with the selected phrase

Example 4: Filter tweets and sentiments for a selected entity.

With Twitter as the source, 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 tweets that contain the selected entity and the associated sentiment.
Figure 15: Filter tabular view of tweets based on the selected entity

Example 5: Filter for tweets and unsafe URLs containing associations with violent or explicit content.

With Twitter as the source, from the Image Analysis tab, select the unsafe moderation label filter the tabular view of tweets. The tabular view provides the date of the tweet, the tweet text, the tweet’s translation, and the image of the URL flagged to have unsafe content.
Discovering Hot Topics Using Machine Learning Implementation Guide
Example use cases for Amazon QuickSight

Figure 16: Filter tabular view of tweets in the second row based on moderation label selection
Discovering Hot Topics Using Machine Learning Implementation Guide

Customizing Amazon S3 ingestion

One of the key features of the solution is the ability to ingest data uploaded to an S3 bucket. The source can be an export of data from internal or external services, or databases in spreadsheets (XLSX) or JSON file formats. Examples that this feature allows analyzing data from include the following:

- 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 irrespective 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.

![Figure 17: customingestion table schema definition in AWS Glue](image)
The solution provides three types of file processor implementations, which can be configured by configuring the environment variables for an AWS Lambda function: Microsoft Excel files, JSON files, and Transcribe Call Analytics.

Microsoft Excel files

Here, each individual record of data in an Excel format file is to be analyzed. Figure 18 shows sample data.

![Sample data in XLSX format for ingestion](image)

Figure 18: Sample data in XLSX format for ingestion

The Excel processor implementation requires the column numbers for certain key elements. To configure the environment variables for the Lambda CustomIngestion function, provide column numbers and remove extra keys. Using the Excel file in Figure 16 as an example, ID becomes column ‘0’, CREATED_DATE becomes ‘1’, TEXT becomes ‘2’, and LANG becomes ‘3’. Delete PROCESSOR_TYPE and LIST_SELECTOR keys, if present. Figure 19 displays the environment variables based on the data from Figure 18.

If the data to be ingested does not have ID or CREATED_DATE, the solution has a capability to generate them as well. To generate ID, set the value as GENERATE and to generate CREATED_DATE set the value as NOW to use system date.

![AWS Lambda environment variables set up for Excel-based data ingestion](image)

Figure 19: AWS Lambda environment variables set up for Excel-based data ingestion
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 list of records:

```json
{
  "list_contents": [
    {
      "content": "Lorem ipsum dolor sit amet, \n      "id": "id1",
      "lang": "en",
      "created_date": "11-19-2021 03:59:07"
    },
    {
      "content": "consectetur adipiscing elit, \n      "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. Hence, in addition to setting ID, CREATED_DATE, LANG, and TEXT, this scenario 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. Following the example, the environment variables would need to be set according to the values in Figure 19.
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`
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 Twitter 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 S3 data storage for the data that is no longer required for the solution.

Note: The S3 buckets are configured with the retention policy set to Retain.

Twitter credentials

If you configure the solution for Twitter ingestion, we recommend that you rotate the credentials for the Twitter bearer token to match your enterprise’s password rotation policy. Twitter supports APIs to retrieve and invalidate tokens. Refer to Retrieve a bearer token for Twitter API authentication (p. 42) section in this guide for more information about how to invalidate, regenerate a fresh bearer token, and update Systems Manager Parameter Store.

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 (p. 46).
Design considerations

Regional deployments

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.

As of February 2022, Discovering Hot Topics Using Machine Learning is supported in the following Regions:

<table>
<thead>
<tr>
<th>Region ID</th>
<th>Region name</th>
</tr>
</thead>
<tbody>
<tr>
<td>us-east-1</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td>us-east-2</td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td>us-west-2</td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td>ap-northeast-1</td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>ap-northeast-2</td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>ap-south-1</td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>ap-southeast-1</td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>ap-southeast-2</td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>eu-central-1</td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td>eu-west-1</td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td>eu-west-2</td>
<td>Europe (London)</td>
</tr>
</tbody>
</table>
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 AWS CloudFormation template, which you can download before deployment.

**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 Kinesis Data Firehose, AWS Step Functions, and AWS Glue tables. You can customize the template to meet your specific needs.
Automated deployment

Before you launch the solution, review the architecture, AWS CloudFormation parameters, AWS Systems Manager parameter configuration, network security, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

**Time to deploy the template:** Approximately 10 mins

Prerequisites

1. For a successful implementation, you must create an account and add an App through the Twitter developer portal. This solution requires a bearer token for Twitter API authorization, which can be found under the **Keys and tokens** tab on the Twitter dashboard inside the App page. The **Name** of the Parameter Store in AWS Systems Manager must match the value of the **CredentialPath** AWS CloudFormation parameter. The bearer token string must be stored in the Parameter Store as a **SecureString Type** in the same account and Region where the solution is deployed. For more information about creating the bearer token and updating the Parameter Store, refer to Retrieve a bearer token for Twitter API authentication (p. 42).

2. 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 S3 bucket and hence charts will display an error.

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 (p. 3)** section in this guide and the pricing webpage for each AWS service.

1. Sign in to the AWS Management Console and select the button below 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.

2. 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.

3. On the Create stack page, verify that the correct template URL is in the Amazon S3 URL text box and choose Next.

4. 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.

5. Under Parameters, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeployTwitter</td>
<td>Yes</td>
<td>Option to select if you would like to ingest tweets. If you select No, the CloudFormation template will not deploy the Twitter ingestion components.</td>
</tr>
</tbody>
</table>
| TwitterSearchQuery     | Amazon OR AWS | The query to search on Twitter. Refer to the Twitter developer guide for details on how to build a query. Examples:
- AWS OR CDK OR "Solution Constructs"

   There is an option to add geo-coordinates to further filter the search based on the location of the tweet. For details, refer to Using the Twitter Search API with geo- coordinates (p. 43).

   **Note**
   This search parameter is updatable and can be modified by updating the CloudFormation template, or by updating the Lambda (ingestion-producer) function environment variable. The change to the Twitter search query would be from this time forth (it is
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TwitterSSMPathForBearerToken</strong></td>
<td>/discovering-hot-topics-using-machine-learning/discovering-hot-topics-using-machine-learning/twitter</td>
<td>The Systems Manager Parameter Store key path to the credentials where the Twitter bearer token is stored as an encrypted string.</td>
</tr>
<tr>
<td><strong>TwitterIngestQueryFrequency</strong></td>
<td>cron(0/10 * * * ? *)</td>
<td>The ingestion schedule as a cron expression supported by a CloudWatch Event Rule that schedules the ingestion of social media feeds.</td>
</tr>
<tr>
<td><strong>SupportedLanguages</strong></td>
<td>en, es</td>
<td>Restricts tweets to the given language, given by an ISO 639-1 code (Example: de, en, es, it, pt, fr, ja, ko, hi, ar, zh-cn, zh-tw). Language detection is best-effort delivery as supported by Twitter's search API. For more details, refer to the Twitter Search API documentation.</td>
</tr>
<tr>
<td><strong>DeployNewsFeeds</strong></td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td><strong>NewsFeedIngestFrequency</strong></td>
<td>cron(0 18 * * ? *)</td>
<td>The ingestion schedule as a cron expression supported by a CloudWatch Event Rule that schedules the ingestion of news feeds.</td>
</tr>
<tr>
<td><strong>NewsSearchQuery</strong></td>
<td>Amazon, AWS</td>
<td>(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.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| NewsFeedIngestConfig   | {"country":"ALL","language":"en","topic":"news"}                | Provide configuration for RSS feeds. This parameter should be configured as a JSON string. Example: {"country":"ALL","language":"ALL","topic":"ALL"}. 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".  
**Note**  
• Not all topics are supported for each RSS provider.  
• Setting the value as ALL is treated as a wild character search. |
| DeployYouTubeCommentsIngestion | 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.                                                          |
| YouTubeSearchQuery     | 'Amazon Web Services| AWS',                                                                 | Optional 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.  
**Note**  
YouTubeSearchQuery or YouTubeChannel is required. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YouTubeChannel</td>
<td>&lt;blank&gt;</td>
<td>Optional YouTube channel ID. This ID is used to retrieve the list of videos and then retrieve comments for each of the videos from the list. <strong>Note</strong> YouTubeSearchQuery or YouTubeChannel is required.</td>
</tr>
<tr>
<td>YouTubeSearchIngestionFreq</td>
<td>cron(0 12 * * ? *)</td>
<td>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.</td>
</tr>
<tr>
<td>YoutubeAPIKey</td>
<td>/discovering-hot-topics-using-machine-learning/youtube/comments</td>
<td>The key name required to retrieve the API key within AWS Systems Manager Parameter Store and access the YouTube APIs.</td>
</tr>
<tr>
<td>DeployCustomIngestion</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>TopicAnalysisFrequency</td>
<td>cron(10 0 * * ? *)</td>
<td>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.</td>
</tr>
<tr>
<td>NumberOfTopics</td>
<td>10</td>
<td>The number of topics you want to detect as part of the topic modeling job between 1-100.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| QuickSightPrincipalArn     | <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.

You can verify that the application is ingesting Twitter information by checking the DynamoDB table status. The table contains entries for every API call that the application makes to Twitter. The **STATUS_COUNT** column in the table also provides information about the number of tweets (records) that were retrieved for each API invocation.

---

**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: SO0122-<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 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, txtinimgentity, 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)**. Figure 21 shows the expected output for normal operation.
Figure 21: 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.
# Additional resources

<table>
<thead>
<tr>
<th>AWS services</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS CloudFormation</td>
<td>Amazon EventBridge</td>
</tr>
<tr>
<td>AWS Lambda</td>
<td>Amazon Translate</td>
</tr>
<tr>
<td>Amazon S3</td>
<td>Amazon Comprehend</td>
</tr>
<tr>
<td>Amazon Kinesis Data Streams</td>
<td>Amazon Rekognition</td>
</tr>
<tr>
<td>Amazon Simple Queue Service</td>
<td>Amazon Athena</td>
</tr>
<tr>
<td>Amazon Kinesis Data Firehose</td>
<td>Amazon CloudWatch</td>
</tr>
<tr>
<td>AWS Step Functions</td>
<td>AWS Systems Manager</td>
</tr>
<tr>
<td>AWS Glue</td>
<td>Amazon Macie</td>
</tr>
<tr>
<td>Amazon QuickSight</td>
<td>AWS Identity and Access Management (IAM)</td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td></td>
</tr>
</tbody>
</table>
Update the stack

To update from v1.4.0 or earlier, 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 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>.

5. From the Amazon Athena console, run MSCK REPAIR TABLE `<table name>` to fix the partitions. For more information refer to How can I resolve the HIVE_METASTORE_ERROR when I query a table in Amazon Athena?
Retrieve a bearer token for Twitter API authentication

This solution uses Twitter's v1.1 standard API. Use the following procedure to create a bearer token for Twitter API access.

1. Create an account on Twitter's Developer Portal, then create a new Application under Project & App. Request for elevated access and provide details about your use case to access the API.
2. After the app is approved by Twitter, you can find the bearer token under the Keys and tokens tab on the Application's dashboard page. If the bearer token doesn't appear on that page, select Generate or Regenerate to generate the bearer token.
3. Copy the token string and store it in the Systems Manager Parameter Store under the key path as configured during CloudFormation deployment.

Use the following procedure to rotate the bearer token.

1. Navigate to the Keys and tokens tab under the Application's dashboard page.
2. Choose Regenerate for the bearer token.
3. Copy the token string and update the Value for the existing Systems Manager Parameter Store key path configured during deployment with Twitter ingestion.

Note
The Value, should be stored as SecureString Type.

For more information about bearer tokens, refer to the Twitter developer guide.
Using the Twitter search API with geo-coordinates

Use the following procedure to set up geo-coordinates with the Twitter search API.

1. Sign in to the AWS Lambda console and select to the AWS Region where the Discovering Hot Topics Using Machine Learning solution was deployed.
2. Search for the producer Lambda function.
3. Select the deployed producer Lambda function.
4. From the Configuration tab, select Environment variables.
5. Add a new environment variable using the LOCATION_GEOCODE key and a value that represents latitude, longitude, and the radius to cover (Format: ‘lat,long,radius’ Example: ‘0,0,5mi’). For more details, refer to the geocode attribute in the Twitter Search API.
Twitter API limit throttling

This solution uses Twitter Search API v1.1. This API endpoint is rate limited by Twitter at 450 requests in a 15 min window. To capture API throttling from Twitter, the solution logs the limit details of these API calls as they are received in the responses from the Lambda log files. The Lambda logs are stored in the CloudWatch Logs corresponding to the `<solution-name>` producer Lambda function. We recommend creating a CloudWatch Alarm using CloudWatch Metric Filters on the following log pattern so that you will be alerted when API calls to Twitter are throttled.

| Throttling limit reached. Hence no more calls to Twitter will be made, until the limit is reset. Limit automatically resets in the next window. |
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.
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 diagram]

Figure 21: Database schema structure
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.

```bash
$ 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 customer master keys topic in the AWS Key Management Service Developer Guide for more details.
Troubleshooting

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.

```bash
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.
Collection of operational metrics

This solution includes an option to send anonymous 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
  - **TwitterSearchQueryComplexity**: The count of phrases used to query the Twitter Search API
  - **TwitterSearchQueryLength**: The string length of the query string used to query the Twitter Search API
  - **TwitterIngestionFreq**: The ingestion frequency configured for the Twitter Search API
  - **TopicJobFreq**: The frequency configured for Topic Modeling jobs
  - **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
  - **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
  - **CustomIngestion**: 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:

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

   to:

   ```json
   "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 (p. 32)** in the Automated Deployment section of this 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.
## Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2020</td>
<td>Initial release</td>
</tr>
<tr>
<td>September 2020</td>
<td>Minor documentation update. Added details to the Overview section and numbered callouts to Figure 1.</td>
</tr>
<tr>
<td>September 2020</td>
<td>Updated documentation to support v1.1.0. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>October 2020</td>
<td>Updated documentation to support v1.2.0. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>November 2020</td>
<td>Updated documentation to support v1.3.0. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>February 2021</td>
<td>Updated documentation to support v1.4.0. Added documentation about using the Twitter Search API with geo-coordinates. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>July 2021</td>
<td>Updated documentation to support v1.5.0. Added documentation about new sources of ingestion (RSS feeds). For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>September 2021</td>
<td>Updated documentation to support v1.6.0. Added documentation about new source of ingestion (YouTube comments). For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>October 2021</td>
<td>Updated documentation to support v1.6.1. This release includes the fix for GitHub issue #42, and library version updates to the AWS CDK and AWS SDK. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>February 2022</td>
<td>Updated documentation to support v1.7.0. This release includes a new feature to ingest data from an S3 bucket, uses the Amazon Kinesis 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 CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>Date</td>
<td>Change</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>March 2022</td>
<td>Updated documentation to support v1.7.1. This release includes updates for dependencies. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
</tbody>
</table>
Contributors

The following individuals contributed to this document:

- Nihit Kasabwala
- Alireza Assadzadeh
- Mukta Dadariya
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 Classless Inter-Domain Routing (CIDR).
AWS glossary

For the latest AWS terminology, see the AWS glossary in the AWS General Reference.