
Real-Time Insights on AWS Account Activity

Implementation Guide



Real-Time Insights on AWS Account Activity: Implementation Guide

Copyright © 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

Home	1
Overview	2
Cost	2
Architecture	2
Components	4
Amazon Kinesis Data Analytics Application	4
Amazon DynamoDB	4
Dashboard	4
Considerations	5
Real-Time Event Monitoring	5
Supported Services	5
Count	5
Regional Deployment	5
Template	6
Deployment	7
Prerequisites	7
Launch the Stack	7
Security	9
AWS CloudTrail	9
Amazon CloudFront	9
Resources	10
Appendix A: Code Components	11
SQL Query	11
JavaScript	11
HTML Element	11
Appendix B: Customizing the Dashboard	13
Step 1. Add the Metric to the Source Schema	13
Step 2. Modify the Application's SQL Code	14
Step 3. Update the JavaScript Code	14
Step 4. Update the Website Assets	15
Appendix C: Operational Metrics	17
Source Code	18
Revisions	19
.....	19

Reference implementation to help customers more easily monitor account activity in real-time and make better-informed decisions

February 2018 (last update (p. 19): May 2021)

This implementation guide discusses architectural considerations and configuration steps for deploying Real-Time Insights on Amazon Web Services (AWS) Account Activity on the AWS Cloud. It includes links to an [AWS CloudFormation](#) template that launches, configures, and runs the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for IT infrastructure architects, administrators, and DevOps professionals who have practical experience architecting on the AWS Cloud.

Overview

Amazon Web Services (AWS) enables customers to achieve significant gains in productivity, innovation, and cost reduction when they move to the AWS Cloud. AWS offers a variety of services and features that allow for flexible control of cloud computing resources and also of the AWS account(s) managing those resources. These options help to ensure proper cost allocation, agility, and security, however customers are sometimes unsure of how to best leverage the elasticity of the AWS Cloud to optimize their costs yet still meet their performance and capacity requirements.

Monitoring AWS account activity can provide valuable insight into who is accessing your resources and how your resources are being used. This insight can help you to make better-informed decisions that increase security and efficiency, facilitate compliance auditing, and optimize costs. Many customers choose to build custom account monitoring solutions using AWS services because these services provide an efficient way to handle a large number of activity events in real-time and flexibility to get specific metrics.

To help customers more easily monitor account activity in real-time, AWS offers the Real-Time Insights on AWS Account Activity solution, a reference implementation that automatically provisions the services necessary to record and visualize resource access and usage metrics for your AWS account(s) in real-time. This solution is designed to provide a framework for visualizing metrics, allowing you to focus on adding new metrics rather than underlying infrastructure operations.

Real-Time Insights on AWS Account Activity uses [AWS CloudTrail](#) to log account activity, [Amazon Kinesis Data Analytics](#) to compute metrics in real-time, and [Amazon DynamoDB](#) to durably store the computed data. The solution features a dashboard that visualizes your account activity metrics in real-time. Metrics are calculated for create, modify, and delete API calls for more than 60 [supported AWS services](#).

Note

The Real-Time Insights on AWS Account Activity solution enables an AWS CloudTrail trail to monitor events that occur in your account in real-time. Some events, however, might take up to 15 minutes to arrive in Amazon Kinesis Data Firehose from CloudTrail.

Cost

You are responsible for the cost of the AWS services used while running this reference deployment. As of the date of publication, the baseline cost for running this solution with default settings in the US East (N. Virginia) Region is approximately **\$100 per month**. Note that the monthly cost will vary depending on your AWS console and API use. Prices are subject to change. For full details, see the pricing webpage for each AWS service you will be using in this solution.

Architecture Overview

Deploying this solution builds the following environment in the AWS Cloud.

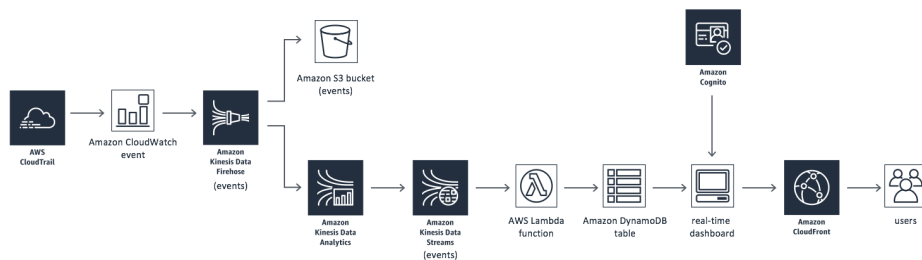


Figure 1: Real-Time Insights on AWS Account Activity architecture

The AWS CloudFormation template deploys an AWS CloudTrail trail, an Amazon CloudWatch event, an Amazon Kinesis Data Firehose delivery stream, Amazon Simple Storage Service (Amazon S3) buckets, a Kinesis data analytics application, a Kinesis data stream, an AWS Lambda function, Amazon DynamoDB tables, an Amazon Cognito user pool, an Amazon CloudFront distribution, and a real-time dashboard.

The AWS CloudTrail trail logs actions taken in your AWS account, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services. When an action is taken, an Amazon CloudWatch event trigger sends data to a Kinesis data delivery stream. The delivery stream archives the events in an Amazon S3 bucket and sends the data to a Kinesis data analytics application for processing. Once the data is processed, it is sent to a Kinesis data stream. A Lambda function (`real-time-insights-account-activity-update-ddb`) reads data from the stream and sends the data in real-time to a DynamoDB table to be stored.

The solution also creates an Amazon Cognito user pool, an Amazon CloudFront distribution, an Amazon S3 bucket, and real-time dashboard to securely read and display the account activity stored in the DynamoDB table.

Solution Components

Amazon Kinesis Data Analytics Application

This solution includes an Amazon Kinesis Data Analytics application with SQL statements that compute metrics for the built-in dashboard. The application reads records from the Amazon Kinesis Data Firehose delivery stream, and runs the SQL queries to emit specific AWS CloudTrail metrics, which are stored in Amazon DynamoDB. For more information, see [Appendix A \(p. 11\)](#).

Amazon DynamoDB

The Real-Time Insights on AWS Account Activity solution creates two Amazon DynamoDB tables: `cloudtrail-log-analytics-metrics` and `cloudtrail-log-ip-metrics`.

The `cloudtrail-log-analytics-metrics` table stores the following information on metrics computed by the Amazon Kinesis Data Analytics application:

- **MetricType:** The name of the computed metric
- **EventTime:** The time the event was generated
- **ConcurrencyToken:** The token used in the event of updates for [optimistic locking](#)
- **Data:** The metric data, in JSON format

The `cloudtrail-log-ip-metrics` table stores the IP address and a count of the number of requests from that IP address for a given hour and minute.

Dashboard

The solution features a simple dashboard that loads data from Amazon DynamoDB into line charts every 10 seconds and bar charts every minute. The dashboard leverages Amazon Cognito for user authentication and is powered by web assets [hosted](#) in an Amazon Simple Storage Service (Amazon S3) bucket. Amazon CloudFront is used to restrict access to the solution's website bucket contents.

The dashboard uses the open-source `chart.js` JavaScript library to draw charts using HTML5. The `dash.html` file contains the HTML elements that render the charts in the dashboard. The `dash.js` file in the `js` folder contains the JavaScript that populates the dashboard with metrics. The Kinesis data application contains the SQL queries that compute metrics. For more information, see [Appendix A \(p. 11\)](#).

After you successfully launch the solution, you will receive an email with instructions for logging into the dashboard.

The dashboard can also be customized to include additional metrics. For more information, see [Appendix B \(p. 13\)](#).

Design Considerations

Real-Time Event Monitoring

The Real-Time Insights on AWS Account Activity solution enables an AWS CloudTrail trail to monitor events that occur in your account in real-time. Some events, however, might take up to 15 minutes to arrive in Amazon Kinesis Data Firehose from CloudTrail.

CloudTrail Supported Services

This solution leverages CloudTrail to record account activity. CloudTrail records account activity and service events from most AWS services. For the list of supported services, see [CloudTrail Supported Services](#) in the CloudTrail User Guide.

CloudTrail Count

You can create up to five CloudTrail trails in a single AWS Region. A trail that applies to all regions will exist in each region and is counted as one trail in each region. Note that this solution will create a trail that applies to all regions. Therefore, this solution's trail will count as one trail in each region in your account.

Regional Deployment

This solution uses the Amazon Kinesis Data Firehose and Amazon Kinesis Data Analytics services, which are currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where these services are available. However, once deployed, the solution monitors AWS Identity and Access Management (IAM) events in all regions. Events will be monitored for all other services in each region the solution is deployed. For the most current service availability by region, see [AWS service offerings by region](#).

AWS CloudFormation Template

This solution uses AWS CloudFormation to automate the deployment of the Real-Time Insights on AWS Account Activity solution. It includes the following AWS CloudFormation template, which you can download before deployment:

[View
Template](#)

real-time-insights-account-activity.template: Use this template to launch the solution and all associated components. The default configuration deploys an AWS CloudTrail trail, an Amazon CloudWatch event, an Amazon Kinesis Data Firehose delivery stream, Amazon Simple Storage Service (Amazon S3) buckets, a Kinesis data analytics application, a Kinesis data stream, an AWS Lambda function, Amazon DynamoDB tables, an Amazon Cognito user pool, an Amazon CloudFront distribution, and a real-time dashboard, but you can also customize the template based on your specific needs.

Automated Deployment

Before you launch the automated deployment, please review the architecture and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy Real-Time Insights on AWS Account Activity into your account.

Time to deploy: Approximately five minutes

Prerequisites

Before you start, verify that you have an AWS account with fewer than five existing AWS CloudTrail trails in a [supported region](#).

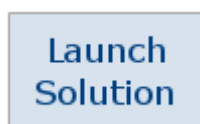
Launch the Stack

This automated AWS CloudFormation template deploys the Real-Time Insights on AWS Account Activity solution. Please make sure that you've verified that you have reviewed the considerations and prerequisites before launching the stack.

Note

You are responsible for the cost of the AWS services used while running this solution. See the [Cost \(p. 2\)](#) section for more details. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Sign in to the AWS Management Console and click the button below to launch the `real-time-insights-account-activity` AWS CloudFormation template.



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

2. The template is launched in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the region selector in the console navigation bar.

Note

This solution uses the Amazon Kinesis Data Firehose and Amazon Kinesis Data Analytics services, which are currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where these services are available. For the most current service availability by region, see [AWS service offerings by region](#).

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template, and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
User Name	<Requires input>	User name to access the real-time dashboard
User Email Address	<Requires input>	Email address of dashboard user. After launch, an email will be sent to this address with dashboard login instructions.
Dashboard Bucket Name	<Requires input>	Specify a name for the new Amazon S3 bucket where the real-time dashboard will be stored. Do not specify an existing bucket.

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

You can view the status of the stack in the AWS CloudFormation console in the Status column. You should see a status of **CREATE_COMPLETE** in approximately five minutes.

The solution sends an email invitation to join the real-time dashboard.

10. In the email, follow the instructions to sign in to the dashboard.

Note

In addition to the primary AWS Lambda function `real-time-insights-account-activity-update-ddb`, this solution includes the `real-time-insights-account-activity-helper` Lambda function, which runs only during initial configuration or when resources are updated or deleted.

After launching this solution, you will see these Lambda functions in the AWS console, but only the `real-time-insights-account-activity-update-ddb` function is regularly active. However, do not delete the `real-time-insights-account-activity-helper` function as it is necessary to manage associated resources.

Security

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

AWS CloudTrail

By default, AWS CloudTrail log files are encrypted using Amazon Simple Storage Service (Amazon S3) Server-Side Encryption (SSE) and placed into your Amazon S3 bucket. You can control access to log files by applying AWS Identity and Access Management (IAM) or Amazon S3 bucket policies. You can add an additional layer of protection against deleted CloudTrail log data by enabling Amazon S3 [Multi Factor Authentication \(MFA\) Delete](#) on the CloudTrail Amazon S3 bucket.

Amazon CloudFront

This solution deploys a static website [hosted](#) in an Amazon S3 bucket. To help reduce latency and improve security, this solution includes an Amazon CloudFront distribution with an origin access identity, which is a special CloudFront user that helps restrict access to the solution's website bucket contents. For more information, see [Restricting Access to Amazon S3 Content by Using an Origin Access Identity](#).

Additional Resources

AWS services

- [Amazon Kinesis Data Firehose](#)
- [Amazon Kinesis Data Analytics](#)
- [AWS CloudTrail](#)
- [AWS Lambda](#)
- [Amazon DynamoDB](#)
- [Amazon Cognito](#)
- [Amazon CloudWatch](#)
- [AWS CloudFormation](#)
- [Amazon CloudFront](#)

Appendix A: Code Components

The Real-Time Insights on AWS Account Activity solution uses three main code components to process and display metrics on the real-time dashboard. The Amazon Kinesis Data Analytics application (`RealTimeInsightsAccountActivityApp`) runs SQL queries against the in-application streams and emits the results. A JavaScript file (`dash.js`) populates the chart with the results of the queries, and an HTML file (`dash.html`) renders the chart on the dashboard in real-time.

The following example shows the SQL, JavaScript, and HTML code for the `CallsPerUniqueIp` metric.

SQL Query

The SQL query calculates the number of calls, in one minute intervals, based on an IP address. The result is stored in an output in-application stream (`DESTINATION_SQL_STREAM`) with the name of the metric (`CallsPerUniqueIp`) and the corresponding values: IP address and count.

```
CREATE OR REPLACE PUMP "PUMP_FOR_CALLS_PER_IP" AS
  INSERT INTO "DESTINATION_SQL_STREAM"
    SELECT eventTimeStamp, 'CallsPerUniqueIp', sip, 'None', 'Sum', callsPerIp FROM (
      SELECT STREAM STEP(cloudtraillogs."eventTimeStamp" BY INTERVAL '1' MINUTE)
        eventTimeStamp, COUNT(*) callsPerIp, "sourceIPAddress" sip
      FROM "SOURCE_SQL_STREAM_001" cloudtraillogs
      GROUP BY "sourceIPAddress", STEP(cloudtraillogs.ROWTIME BY INTERVAL '1' MINUTE),
        STEP(cloudtraillogs."eventTimeStamp" BY INTERVAL '1' MINUTE));
```

JavaScript

The JavaScript populates the chart with the calls per unique IP.

```
var ipParams = retrieveParams("CallsPerUniqueIp", ipQueryTime);
docClient.query(ipParams, function(err, data) {
  if (err) console.log(err);
  else {
    ipQueryTime = updateHorizontalBarChart(data, 5, osChart, ipQueryTime,
splitFunc);
  }
});
```

HTML Element

The HTML element renders the `CallsPerUniqueIp` chart with the results of the SQL query.

```
<div class="row aws-mb-1">
  <div class="col-md-5 col-md-offset-1 col-xs-12">
```

Real-Time Insights on AWS Account
Activity Implementation Guide
HTML Element

```
<div class="x_title">
  <h3>Max calls per IP <small> Over last 24 hours </small></h3>
</div>
<div class="x_content">
  <canvas id="maxIpCanvas"></canvas>
</div>
</div>

<div class="col-md-5 col-xs-12">
  <div class="x_title">
    <h3>Top Calls By IP <small> Over last 1 hour</small></h3>
  </div>
  <div class="x_content">
    <canvas id="osCanvas"></canvas>
  </div>
</div>
</div>
```

Appendix B: Customizing the Dashboard

The Real-Time Insights on AWS Account Activity solution dashboard displays a default set of metrics, but you can customize the dashboard to include any metrics from your AWS CloudTrail logs. Follow the step-by-step instructions in this section to add a metric to the dashboard.

For this exercise, you can add the `awsRegion` metric in the CloudTrail event.

```
{
  "eventVersion": "1.03",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:user/myUserName",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "myUserName"
  },
  "eventTime": "2015-08-26T20:46:31Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "GetBucketVersioning",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "",
  "userAgent": "[]",
  "requestParameters": {
    "bucketName": "myawsbucket"
  },
  "responseElements": null,
  "requestID": "07D681279BD94AED",
  "eventID": "f2b287f3-0df1-4961-a2f4-c4bdfed47657",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Step 1. Add the Metric to the Source Schema

Use this procedure to update the source schema with the new metric. For information on using the Schema Editor, see [Working with the Schema Editor](#) in the Amazon Kinesis Data Analytics Developer Guide.

Note

If the custom metric is already added to the Amazon Kinesis Data Analytics application's input stream, skip to [Step 2 \(p. 14\)](#).

1. Sign in to the AWS Management Console and open the Amazon Kinesis Data Analytics console.
2. Select the **RealTimeInsightsAccountActivityApp** application from the list.
3. Under **Real-Time Analytics**, choose **Go to SQL results**.
4. On the **Source data** tab, choose **Edit schema**.

5. Choose **+ Add column** and enter the following:
 - For **Column name**, enter `awsRegion`.
 - For **Column type**, enter `VARCHAR`.
 - For **Length**, enter `64`.
 - For **Row path**, enter `$.detail.awsRegion`.
6. Choose **Save schema and update stream samples**.
7. To verify that you added the metric correctly, choose **Go to SQL results** and verify that the **Source data** tab shows the new column (`awsRegion`) and an applicable value.

Step 2. Modify the Application's SQL Code

Use this procedure to update the application's code with the new SQL statement. For information on using the SQL Editor, see [Working with the SQL Editor](#) in the Amazon Kinesis Data Analytics Developer Guide.

1. On the Kinesis Data Analytics application's **SQL Editor** page, select the **Real-time analytics** tab.
2. Add the following SQL statement

```
CREATE OR REPLACE PUMP "PUMP_FOR_CALLS_BY_REGION" AS
  INSERT INTO "DESTINATION_SQL_STREAM"
    SELECT eventTimeStamp, 'CallsPerRegion', "awsRegion" , 'None', 'Sum',
    callsPerRegion FROM (
      SELECT STREAM STEP(cloudtraillogs."eventTimeStamp" BY INTERVAL '10' SECOND) as
      eventTimeStamp, COUNT(*) callsPerRegion, "awsRegion"
      FROM "SOURCE_SQL_STREAM_001" cloudtraillogs
      GROUP BY "awsRegion", STEP(cloudtraillogs.ROWTIME BY INTERVAL '10' SECOND),
      STEP(cloudtraillogs."eventTimeStamp" BY INTERVAL '10' SECOND));
```

This SQL statement creates a new metric (`CallsPerRegion`) that stores the number of API calls per AWS Region with the associated region value in the Amazon DynamoDB table.

3. Select **Save and run SQL**.

Step 3. Update the JavaScript Code

The solution creates an Amazon Simple Storage Service (Amazon S3) bucket with a `js` folder that contains a `dash.js` file with the JavaScript code that populates the charts with metrics. To populate the new chart with metrics, download the `dash.js` file and follow the step-by-step instructions to modify the JavaScript.

1. Declare the variables and parameters. For this exercise, add the bold JavaScript to the `dash.js` file.

```
var serviceCallChartData = {'labels': [], 'times': [], 'values': {}}
var serviceCallQueryTime = getTimeSecsAgo(15*60, currentTime);
var serviceCallChart = generateLineChart("callsByServiceCanvas", "No of service
calls");

var regionCallChartData = {'labels': [], 'times': [], 'values': {}}
var regionCallQueryTime = getTimeSecsAgo(15*60, currentTime);
var regionCallChart = generateLineChart("callsByRegionCanvas", "No of region calls");
```

2. Modify the `updateDashboard` function. Add the bold JavaScript to the function.

```
while(isInFastUpdate);
    isInSlowUpdate = true;
    docClient.query(serviceTypeParams, function(err, data) {
        if (err) console.log(err);
        else {
            serviceCallChartData = updateLineChart(data, serviceCallChartData,
            serviceCallChart, splitFunc) ;
        }
    });

var awsRegionParams = retrieveParams("CallsPerRegion", regionCallQueryTime);

    docClient.query(awsRegionParams, function(err, data) {
        if (err) console.log(err);
        else {
            regionCallChartData = updateLineChart(data, regionCallChartData,
            regionCallChart, splitFunc);
        }
    });
```

3. Modify the `fastUpdate` function. Add the following JavaScript to the function.

```
while(isInSlowUpdate);
    isInFastUpdate = true;
    docClient.query(regionParams, function(err, data) {
        if (err) console.log(err);
        else {
            serviceCallQueryTime = fastUpdateLineChart(data, serviceCallChartData,
            serviceCallChart, serviceCallQueryTime, splitFunc) ;
        }
    });

var regionParams = retrieveParams("CallsPerRegion", regionCallQueryTime);

    docClient.query(serviceTypeParams, function(err, data) {
        if (err) console.log(err);
        else {
            regionCallQueryTime = fastUpdateLineChart(data, regionCallChartData,
            regionCallChart, regionCallQueryTime, splitFunc) ;
        }
    });
```

4. Upload the modified `dash.js` file to the solution's Amazon S3 bucket.

Step 4. Update the Website Assets

In the Amazon S3 bucket with the JavaScript, there is a file (`dash.html`) that contains all the HTML elements that render charts on the dashboard. To add a new chart, download the `dash.html` file, modify the HTML, and upload the modified file to the Amazon S3 bucket. You can replace the row of an existing chart with the new row, or add the new row to the end of the file.

For this exercise, add the bold HTML element to the dash.html file.

```
<div class="row aws-mb-1">
  <div class="col-md-5 col-md-offset-1 col-xs-12">

    <div class="x_title">
      <h3>Calls per AWS region <small> Over the last hour </small></h3>
    </div>
    <div class="x_content">
      <canvas id="callsByRegionCanvas"/>
    </div>
  </div>
  <div class="col-xs-5 col-xs-offset-1 col-xs-12">
    <div class="x_title">
      <h3>EC2 Calls <small> over the last hour </small></h3>
    </div>
    <div class="x_content">
      <canvas id="callsByEC2Canvas"/>
    </div>
  </div>
</div>
</div>
```

After you upload the modified dash.html file to the Amazon S3 bucket, open the dashboard in a browser and verify the new chart shows metrics.

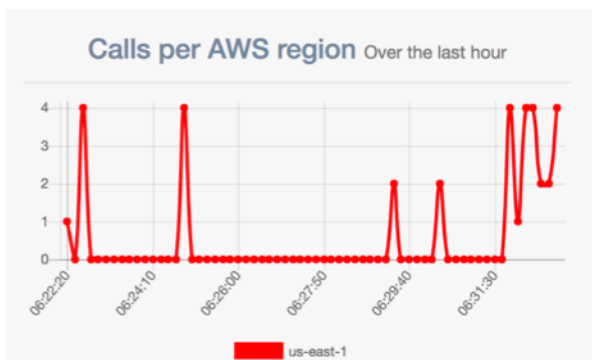


Figure 2. Calls per AWS Region chart

Appendix C: 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 to improve the services and the products that we offer. When enabled, the following information is collected and sent to AWS each time the AWS Lambda function is invoked:

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each solution deployment
- **Timestamp:** Data-collection timestamp
- **All Successful API Calls:** The aggregate number of successful API calls made within the time period
- **Anomaly Score:** The anomaly score reported Amazon Kinesis Data Analytics for the time period
- **Dashboard Views:** The number of times the dashboard is viewed

Note that AWS will own the data gathered via this survey. Data collection will be subject to the [AWS Privacy Policy](#). To opt out of this feature, modify the AWS CloudFormation template mapping section as follows:

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

to

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

Source Code

You can visit our [GitHub repository](#) to download the templates and scripts for this solution, and to share your customizations with others.

Document Revisions

Date	Change	
February 2018	Initial release	
December 2018	Added information about the Amazon CloudFront distribution for the static website hosted in the Amazon S3 bucket.	
August 2019	Upgraded the solution's AWS Lambda functions to the latest Node.js runtime.	
March 2020	Upgraded the solution to the latest Python and Node.js runtime	
May 2021	Release v1.1.2 - For more information about version 1.1.2, refer to the CHANGELOG.md file in the GitHub repository	

Notices

This implementation guide is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Real-Time Insights on AWS Account Activity is licensed under the terms of the Amazon Software License available at <http://aws.amazon.com/asl/>