aws

Developer Guide

Amazon Lookout for Metrics

Amazon Lookout for Metrics: Developer Guide

Table of Contents

Amazon Lookout for Metrics is no longer available to new customers. Existing Amazon Lookout for Metrics customers will be able to use the service until September 12, 2025, when we will end support for Amazon Lookout for Metrics. To help transition off of Amazon Lookout for Metrics, please read Transitioning off Amazon Lookout for Metrics.

What is Amazon Lookout for Metrics?

Amazon Lookout for Metrics is a service that finds anomalies in your data, determines their root causes, and enables you to quickly take action. Built from the same technology used by Amazon.com, Amazon Lookout for Metrics reflects 20 years of expertise in anomaly detection and machine learning.

In Lookout for Metrics, you create detectors that monitor data to find anomalies. You configure the detector with a datasource and choose the values that it monitors (the dataset's *measures*). The detector can monitor all values of a measure overall, or use other data to sort measures into groups. For example, you can choose to monitor the availability of an application worldwide, or use a location field in your data as a *dimension* to monitor availability separately in each AWS Region or Availability Zone. Each combination of measure and dimension value is called a *metric*.

To gain an understanding of your data and better find anomalies, a detector uses your data to learn. When you create a detector, you choose an interval from between 5 minutes and 1 day, which determines how often the detector imports data and finds anomalies. Depending on the interval that you choose, a detector spends between a few hours and a few days learning about your data. To speed up the learning process, you can provide *historical data*. You can also use historical data to run a detector in backtest mode to see how well it works on your data.

When Lookout for Metrics finds an anomaly, it gives it a severity score to indicate how unexpected it is, based on the detector's understanding of your data. When multiple metrics are affected by an anomaly, the detector collects them in a group. To send anomaly alerts when high severity anomalies occur, you can add alerts to the detector. Alerts can run a Lambda function or send a notification to an Amazon SNS topic. Anomaly alerts summarize all affected metrics for an anomaly, which reduces the amount of similar alerts that it sends.

Lookout for Metrics can monitor data from many different domains, stored in many different formats. In addition to monitoring operational metrics for emergent issues, you can find anomalies in sales data, marketing data, and customer engagement metrics.

Lookout for Metrics integrates with other services to provide additional datasources and alert channels. Lookout for Metrics supports the following services as datasources.

- ??? – Analyze data retrieved by your queries.
- Amazon S3 – Read metrics data from text objects in a bucket. Supports using historical data for training and testing a detector.

- Amazon CloudWatch – Analyze CloudWatch metrics sent by an AWS service, AWS resource, or an application.
- Amazon Relational Database Service (Amazon RDS) – Analyze items in a relational database. The following database engines are supported:

  - **Amazon Aurora**
  - **MySQL**
  - **PostgreSQL**
  - **MariaDB**
  - **Microsoft SQL Server**
- Amazon Redshift – Analyze items in a Amazon Redshift data warehouse.
- Amazon AppFlow – Analyze data flow from a web service with an Amazon AppFlow data flow. The following services are supported:
  - **Salesforce**
  - **Marketo**
  - **Dynatrace**
  - **Singular**
  - **Zendesk**
  - **Servicenow**
  - **Infor Nexus**
  - **Trendmicro**
  - **Veeva**
  - **Google Analytics**
  - **Amplitude**

To get started with Lookout for Metrics and learn more about the service, continue to Getting started with Lookout for Metrics.

Getting started with Lookout for Metrics

This section shows you how to get started with Amazon Lookout for Metrics by creating a detector and finding anomalies.

Topics

Lookout for Metrics terms and concepts

This section describes key concepts and terminology you need to understand to use Amazon Lookout for Metrics effectively.

Sections

Detector

A *detector* is a Lookout for Metrics resource that monitors a dataset and identifies anomalies. Detectors use machine learning to find patterns in data, and to distinguish between expected variations in data and legitimate anomalies. To improve its performance, a detector learns more about your data over time.

When you create a detector, you configure an *interval* to tell the detector how often to update the dataset and look for anomalies. An interval can range from 5 minutes to 1 day. At the end of each interval, the detector looks for anomalies in the data from the interval.

If all data is not available immediately at the end of each interval, you can configure an *offset* that specifies an amount of time for the detector to wait before it starts processing data. Here are some examples of how the interval and offset values work together.

Offset examples

Offset value	When data is processed	Interval	What data is processed (date range)
0	22 June 2022, 00:00:00	1 day	21 June 2022, 00:00:00 - 22 June 2022, 00:00:00

Offset value	When data is processed	Interval	What data is processed (date range)
1 hour	22 June 2022, 00:01:00	1 day	21 June 2022, 00:00:00 - 22 June 2022, 00:00:00
1 hour	22 June 2022, 00:01:00	10 minutes	21 June 2022 23:50:00 - 22 June 2022, 00:00:00
2 days, 9 hours	22 June 2022, 09:00:00	1 day	19 June 2022, 00:00:00 - 20 June 2022, 00:00:00
2 days, 9 hours	22 June 2022, 09:00:00	8 hours	19 June 2022, 16:00:00 - 20 June 2022, 00:00:00

In addition to a dataset, a detector can have alerts. For more information, see Working with detectors.

Datasource

A datasource is a service or resource that provides time-series data for analysis. A datasource has dated records that each have metrics and (optionally) dimensions. If you make an application that writes records to Amazon S3 as they occur, you can use the Amazon S3 bucket as a datasource.

Lookout for Metrics integrates with other AWS services to provide additional datasources. A detector can access records in a relational database in Amazon Relational Database Service or Amazon Redshift, metrics in Amazon CloudWatch, or data in an Amazon AppFlow flow. With these services, you can collect data from even more services and resources.

Dataset

A *dataset* is a detector's copy of your data. At the end of each interval, the detector copies metrics and dimensions from a datasource into its dataset for analysis. Each row or document in a dataset

is called a *record*. Each labeled value or key-value pair in a record is a *field*. Each record must have a timestamp field and at least one field that the detector monitors, called a *measure*.

A dataset can have continuous data, historical data or both. A detector looks for anomalies in continuous data, and uses historical data for learning. If its dataset has only historical data, a detector runs in *backtest mode*. In backtest mode, a detector uses most of the data for learning and attempts to find anomalies in a smaller subset of recent data.

Datasets have a datasource, a service role, and a mapping that tells the detector which values to monitor. For more information, see Managing a dataset in Amazon S3.

Metrics

After giving your detector a datasource, you choose fields to be the dataset's measures. *Measures* are the numerical fields that the detector monitors. You can also configure up to 10 additional fields as *dimensions*. Dimensions are categorical fields that create subgroups of measures based on their value. Each combination of measure and dimension value is called a *metric*.

For example, you can choose a field named `availability` for a measure. If you don't choose a dimension, the detector monitors availability across all records. If you choose a field named `country` for a dimension, then the detector monitors availability in each country as a separate metric: *availability in Canada*, *availability in Italy*, and so on.

> **Important**
>
> Pricing in Lookout for Metrics is based on the number of metrics that detectors monitor. Each dimension increases the number of metrics by the number of unique values in the dimension. For example, the number of countries that appear in a `country` dimension. Choose dimensions that have a limited number of known values.
> For more information, see Pricing.

For more information, see Setting up a detector.

Alert

To send a notifications or initiate a processing workflow when the detector finds an anomaly, you can create an alert. An alert has a target that can be an Amazon Simple Notification Service (Amazon SNS) topic or an AWS Lambda function. When the detector finds an anomaly with a

severity score over a configurable threshold, the alert sends a record of the anomaly to the target. A severity score is a number between 0 and 100 that indicates how far the metric value is outside of the expected range.

You can process the anomaly record in the programming language of your choice with a Lambda function. With an Amazon SNS topic, you can send the anomaly record to multiple subscribers. A subscriber can use any protocol that Amazon SNS supports, including email addresses, mobile devices, and webhooks. With a webhook, you can send the anomaly record to third-party services, such as Slack.

For more information, see Working with alerts.

Feedback

When the detector finds an anomaly, it creates a report with details about all of the metrics that had unexpected values during the interval in the Lookout for Metrics console. When you view this report , you can provide feedback on the relevance of each metric. The detector uses your feedback to improve its accuracy.

For more information, see Working with anomalies.

Create a datasource in Amazon S3

To get started with Lookout for Metrics, create a dataset that you can look for anomalies in. You can use data that you already have, or generate data with a sample application.

A Lookout for Metrics detector can read data in an Amazon S3 bucket in CSV or JSON lines format. Each record must have a numerical field to monitor (a *measure*) and a field that represents a date or timestamp. Records can also have categorical fields (*dimensions*), additional measures, and fields that are not monitored.

For example, the following example is a CSV file that has a measure field named `Calls`, a timestamp field named `Date`, and text fields that could be used as dimensions.

Example CSV file

```
Date,EventSource,EventName,ReadOnly,AccessKeyId,Username,Calls
2021-03-12 22:00:07,kms.amazonaws.com,GenerateDataKey,true,none,none,1
2021-03-12 22:00:12,kms.amazonaws.com,Decrypt,true,ASIAXMPLV4CQYXGMLRQ,greg,1
2021-03-12 22:00:36,kms.amazonaws.com,Decrypt,true,ASIAXMPLVMP6KEG3BRA,michael,1
```

In the example, there is a header row with names for each field, fields are separated by commas, and values are not enclosed in quotes. The header is optional, but it simplilfies configuration. Values can be enclosed in single or double quotes. In addition to commas, tabs, spaces, and pipes (`|`) can be used as delimeters.

Sections

Generate data with a sample application

The GitHub repository for this guide includes a sample application that you can use to generate a dataset with data from your AWS account. The sample application runs locally or in AWS Lambda. It uses the AWS SDK to read events from AWS CloudTrail, reformats them with Pandas, and stores them in Amazon S3.

Example lambda_function.py – Processing data

```
...
def process_events(end_time, service):
    start_time = end_time - timedelta(minutes=60)
    records = []
    # call CloudTrail
    responses = get_events(service, start_time, end_time)
    records = [{k: event.get(k) for k in fields} for response in responses for event in
 response.get('Events')]
    # format timeseries dataframe
    df = pd.DataFrame(records)
    ...
```

To set up the application, follow the instructions in the README file. To generate a dataset for a backtest, run the backfill script.

```
(data-processor) data-processor$ ./5-backfill.sh
```

In Lambda, the application runs once every hour to generate a timeseries for the previous hour. The backfill script runs the application locally to generate two weeks worth of data and store it in the application's S3 bucket.

Collect existing data

To use existing data to run a backtest, collect it in one or more files and store it in a single folder in an Amazon S3 bucket. The data must contain timestamped entries for at least 285 intervals, where an interval is 5 minutes, 10 minutes, 1 hour, or 1 day long. Each file can have entries for one interval or multiple intervals.

```
s3://lookoutmetrics-dataset-1234567891012/backtest/20210104-20210110.csv
                                                  20210111-20210117.csv
                                                  20210118-20210125.csv
```

To use a detector in continuous mode, you need to store it at a different path for each interval as the intervals occur. For an example of how to do this, run the sample application.

For more information about supported data formats and folder structures, see Managing a dataset in Amazon S3.

Download sample data

Sample data is available in the service's GitHub repository at github.com/aws-samples/amazon-lookout-for-metrics-samples. The samples repository provides an archive of sample data and instructions for using it with a detector.

Set up a detector

After you've prepared data in an Amazon Simple Storage Service (Amazon S3) bucket, create a detector and add a dataset. At the end of each interval, the detector imports data from the bucket (your datasource) into the dataset and analyzes it.

Steps

- Create a detector
- Add a dataset
- Activate the detector
- (Optional) Add an alert
- Review anomalies
- Clean up

Create a detector

Use the Lookout for Metrics console to create the detector that monitors your data.

To create a detector

1. Open the Lookout for Metrics console.
2. Choose **Create detector**.
3. For **Name**, enter `sample-detector`.
4. For **Description**, enter `Getting started detector`.
5. For **Interval**, choose the interval that you used to organize your data.
6. Choose a **Create**.

Add a dataset

Add a dataset by specifying the location of your data in Amazon S3 and the names of fields in your data that are timestamps, measures, and dimensions.

To add a dataset

1. Open the Lookout for Metrics console Detectors page.

2. Choose **sample-detector**.
3. Choose **Add dataset**.
4. For **Name**, enter **`S3-dataset`**.
5. For **Timezone**, choose the timezone that is reflected in your data's timestamps.
6. For **Datasource**, choose Amazon S3.
7. Choose a detector mode. For **Backtest**, you only provide historical data. For **Continuous**, you provide continuous data and can optionally provide historical data to speed up learning.
8. Follow the instructions to specify the data path(s) and then choose **Next**.
9. Choose the field in your data that specifies a timestamp for each record. The detector checks the timestamp on each record, and only processes records that match the interval that it is analyzing.
10. Choose a numerical field in your data to be a **Measure**. The detector aggregates values of the measure field within each interval and monitors the aggregate value. For example, `availability`
11. (Optional) Choose additional fields to be measures and dimensions. A dimension is a field that creates subgroups of measure values that are monitored separately. For example, `city`.

Activate the detector

To import data from the bucket and look for anomalies, activate the detector.

To activate the detector

1. Open the Lookout for Metrics console Detectors page.
2. Choose the detector.
3. Choose **Activate detector**.

When you activate the detector, it uses data from several intervals to learn, before attempting to find anomalies. If no historical data is available, the training process takes approximately one day for a five-minute interval. Training time varies depending on the detector's interval.

(Optional) Add an alert

In this section, you will add an alert to the detector. You will need to create an Amazon SNS topic. You will use its ARN in this step. For more information about creating an Amazon SNS topic, see Tutorial: Creating an Amazon SNS topic.

To add an alert

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Add alert**.
4. Configure the following options.

   - **Name** – `my-alert`
   - **Description** – `Send anomaly alerts to an SNS topic.`
   - **Threshold** – `50`
   - **Channel** – Amazon SNS
   - **Format** – Short Text
   - **Resource** – Your SNS topic
   - **Role** – **Create a role**
5. Choose **Add alert**.

Review anomalies

In this section, you will review the anomalies found by the detector.

To review anomalies

1. Open the Lookout for Metrics console Detectors page.
2. Choose **sample-detector**.
3. Choose **View anomalies**.
4. Choose an anomaly.

Each anomaly can have multiple metrics associated with it. Review each metric to see if it is relevant or not, and use the feedback options to help the detector learn about your data.

Clean up

If you are done using the detector, delete it.

To delete the detector

1. Open the Lookout for Metrics console Detectors page.
2. Choose the detector, **sample-detector**.
3. Choose **Delete**.

The detector's dataset and alerts are deleted automatically.

Pricing

Pricing for Amazon Lookout for Metrics is based on the number of metrics that your detector monitors for anomalies. This number varies based on the number of measures and dimensions that you choose when you configure a dataset. It can also vary from interval to interval, based on the number of different values that appear for each dimension.

For example, if you choose a field named `availability` as a measure and choose no dimensions, Lookout for Metrics monitors values of `availability` for each interval (1 metric). If you choose `availability` as a measure and `state` as a dimension, where `state` has 50 possible values, Lookout for Metrics monitors values of `availability` for each value of `state` (50 metrics).

For another example, if you choose a field named `orders` with a dimension of `state`, the number of metrics can vary from interval to interval. You might have orders from 40 states in 1 interval, and 45 in another. For these intervals, the number of metrics would be 40 and 45, respectively.

For specific prices and pricing examples, see Amazon Lookout for Metrics pricing .

Sections

Charges for metrics

When a detector monitors data, it generates charges for each metric that appears in the dataset during an interval. To determine the number of metrics, the number of measures that you assign when you configure your dataset is multiplied by the number of unique combinations of dimension name and dimension value that appear in live data.

For example, if you choose 2 fields for measures, and 3 dimensions that each have 5 possible values, the maximum number of metrics that can appear in an interval is 250. If not all dimension values appear in every interval, the number of metrics can vary between 6 and 250 for each interval.

Important

The cost of using a detector increases linearly with the number of different values that appear in each interval for the dimensions that you configure for your dataset. For

example, if you use a dimension named `continent` that has 7 possible values, the cost of that dimension is limited by the number of measures times 7. If you use a dimension named `id` that stores a GUID, the cost of the dimension increases with the number of GUIDs that appear in the data.

Charges for other services

When Lookout for Metrics copies data into a dataset from a datasource, data transfer and API usage charges can apply. Several services have a free tier that applies if your overall use of that service is low. Some free tiers are limited to the first year after you create your account. Others apply permanently or under specific conditions. When using Lookout for Metrics, you might incur charges for the following services:

- Amazon Simple Storage Service (Amazon S3) – Charges for storage, API requests, and data transfer. Eligible for free tier under certain conditions.
- Amazon CloudWatch – Charges for CloudWatch metric storage and retrieval (transfer out). Eligible for free tier under certain conditions.
- Amazon Relational Database Service – Charges for database instances and data transfer. Eligible for free tier under certain conditions.
- Amazon Redshift – Charges for data warehouse nodes and data transfer. Eligible for free tier under certain conditions.
- Amazon AppFlow – Charges per flow run (transfer in) and for each GB of data processed.

This list highlights the main billing components of each service. Additional charges apply for use of optional features. For more information, see the pricing page for each service.

Lookout for Metrics quotas

Amazon Lookout for Metrics sets quotas for the amount of data that a detector can use to learn and detect anomalies. There are also quotas for data import intervals, records processing, and Amazon Lookout for Metrics API requests. Additionally, there are data requirements for data retention for re-training, coldstart anomaly detection, backtesting, time series, and record field key value pairs.

Quotas set by other services can impact operation. For information on quotas for other services such as Amazon CloudWatch, see AWS service quotas in the *Amazon Web Services General Reference*.

Topics

Quotas

	Resource	Default quota	Adjustable
Quotas for detectors	Maximum number of detectors	10	Yes
	Maximum number of alerts for a detector	10	Yes
Maximum number of files you can upload per interval	5-minute interval	5 files	Yes
	10-minute interval	5 files	Yes
	1-hour interval	10 files	Yes
	1-day interval	10 files	Yes

	Resource	Default quota	Adjustable
Maximum number of records Lookout for Metrics can process per interval	5-minute interval	15000 records	Yes
	10-minute interval	24000 records	Yes
	1-hour interval	150000 records	Yes
	1-day interval	150000 records	Yes
Maximum number of API requests per second	Each second	10 requests	Yes
	Maximum number of datasets for a detector	1 dataset	Yes
Quotas for datasets	Maximum number of dimensions for a dataset	10 dimensions	No
	Maximum number of measures for a dataset	10 measures	No
Quotas for historical data	Maximum intervals (continuous mode)	2500	No
	Maximum intervals (backtest mode)	3000	No
	Maximum number of files in historical data	3000	No
Maximum number of metrics Lookout for Metrics can process per detector	5-minute interval	5000 metrics	No
	10-minute interval	10000 metrics	No
	1-hour interval	50000 metrics	Yes
	1-day interval	50000 metrics	Yes

To request a quota increase, use the Service Quotas console and follow the steps in the Requesting a quota increase section of the *Service Quotas User Guide*.

Data retention time periods for re-training

The amount of time you should retain your data for re-training depends on the interval of your data collection as follows:

Interval	Maximum time	Average time
5 minutes	9 days	4.5 days
10 minutes	18 days	9 days
1 hour	3.4 months	2 months
1 day	5 years	2 years

Coldstart anomaly detection

For coldstart anomaly detection (no historical data), the amount of time it takes for Lookout for Metrics to detect anomalies depends on the interval of your data collection as follows:

- **5 minute interval** – 25 hours
- **10 minute interval** – 50 hours
- **1 hour interval** – 4 days
- **1 day interval** – 14 days

File size

The amount of data that the detector can process for an interval is limited. You can reduce the size of files by aggregating records or by removing fields that are not used as measures or dimensions.

- **5 minute interval** – 200 MB
- **10 minute interval** – 200 MB
- **1 hour interval** – 200 MB
- **1 day interval** – 200 MB

Backtesting data requirements

The ratio of training and testing data used for backtesting is 70% training and 30% testing. Amazon Lookout for Metrics allows a maximum of 3000 intervals of data, therefore the maximum number of data points you will see for backtesting is 900. The number of days of data Lookout for Metrics considers for backtesting depends on the interval of your data collection as follows:

- **5 minute interval** – 3.125 days, with 12 data points per hour and 288 data points per day.
- **10 minute interval** – 6.25 days, with 6 data points per hour and 144 data points per day.
- **1 hour interval** – 37.5 days, with 1 data point per hour and 24 points per day.
- **1 day interval** – 900 days, with 1 data point per day.

The minimum amount of data required for backtesting depends on the interval of your data collection as follows:

- **5 minute interval** – ~1 day of data.
- **10 minute interval** – 1.9 days of data.
- **1 hour interval** – 11.8 days of data.
- **1 day interval** – 285 days of data.

Record field key-value pair character limits

Character limits for record field key value pairs are as follows:

- Dimensions key – 63 characters
- Dimension value – 40 characters
- Timestamp key – 63 characters
- Measure value – 10^15 with precision of 4 after decimal point

Amazon Lookout for Metrics permissions

You can use AWS Identity and Access Management (IAM) to manage access to the Lookout for Metrics API and resources like detectors and datasets. For users and applications in your account that use Lookout for Metrics, you manage permissions in a permissions policy that you can apply to users, groups, or roles.

Lookout for Metrics uses IAM service roles to access other services on your behalf. You create or choose a service role when you create a dataset that reads data from Amazon S3 or another service. You also pass service roles to Lookout for Metrics when you configure an alert that targets a Lambda function or an Amazon SNS topic. The Lookout for Metrics console can create these roles for you if you have the required IAM permissions.

For more information about IAM, see What is IAM? in the *IAM User Guide*.

Topics

Identity-based IAM policies for Lookout for Metrics

To grant users in your account access to Lookout for Metrics, you use identity-based policies in AWS Identity and Access Management (IAM). Identity-based policies can apply directly to users, or to groups and roles that are associated with a user. You can also grant users in another account permission to assume a role in your account and access your Lookout for Metrics resources.

The following IAM policy allows a user to access all Lookout for Metrics API actions, and to pass service roles to Lookout for Metrics.

Example User policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lookoutmetrics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "lookoutmetrics.amazonaws.com"
        }
      }
    }
  ]
}
```

The preceding policy does not allow a user to create IAM roles. For a user with these permissions to create a dataset or alert, an administrator must create the service role that grants Lookout for Metrics permission to access datasources and alert channels. For more information, see Service roles for Amazon Lookout for Metrics.

In addition to Lookout for Metrics, a user needs permission to view resources in services that they use as a detector's datasource or as alert channels. When you work with a detector in the Lookout for Metrics console, the console uses your permissions to simplify the configuration process.

You can grant full access to each service or limit the scope of permissions by resource name. The following example shows a policy that provides read-only access to a subset of resources in Lookout for Metrics. The `Resources` key for applicable actions limits access to resources whose names start with `intern-`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "readonly-0",
            "Effect": "Allow",
            "Action": [
                "lookoutmetrics:List*",
                "lookoutmetrics:Get*",
                "lookoutmetrics:Describe*"
            ],
            "Resource": [
                "arn:aws:lookoutmetrics:us-east-2:123456789012:MetricSet/intern-*/
intern-*",
                "arn:aws:lookoutmetrics:us-east-2:123456789012:Alert:intern-*",
                "arn:aws:lookoutmetrics:us-east-2:123456789012:AnomalyDetector:intern-
*"
            ]
        },
        {
            "Sid": "readonly-1",
            "Effect": "Allow",
            "Action": [
                "lookoutmetrics:ListAnomalyDetectors",
                "lookoutmetrics:GetSampleData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "readonly-2",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "iam:PassedToService": "lookoutmetrics.amazonaws.com"
                }
            }
        }
    ]
}
```

If you tag your resources, you can also use condition keys to limit access to a resource based on the presence or value of a tag. For permissions purposes, detectors, datasets and alerts are independent resources. If you grant or deny permission to a detector, the permission does not automatically apply to the detector's dataset or alerts.

The resources and conditions supported for use in policies vary among API actions. For more information, see Actions, resources, and condition keys for Amazon Lookout for Metrics in the Service Authorization Reference.

Service roles for Amazon Lookout for Metrics

You can use service roles to grant Amazon Lookout for Metrics permission to access data and send alerts to other services. A service role is an AWS Identity and Access Management (IAM) role that an AWS service can use to access resources from other services in your account. For example, to invoke an AWS Lambda function when an alert is sent, Lookout for Metrics assumes a role that allows it to invoke the function. Lookout for Metrics uses a separate service role for each supported data source and notification target, including Amazon Simple Storage Service (Amazon S3), Lambda, and Amazon Simple Notification Service (Amazon SNS)

When you configure a dataset or alert in the Lookout for Metrics console, the console can create the required role for you if you have the necessary permissions. If you don't have permissions to create roles, or if you use the Lookout for Metrics API to manage resources, you can create the roles manually. Service roles must have the following trust policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lookoutmetrics.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

The trust policy allows Lookout for Metrics to assume the role.

Sections

Sample IAM policies

The GitHub repository for this guide provides sample IAM policies that you can use as reference for developing service roles. You can use a single role that grants permission for both importing data and sending alerts by combining the applicable policies.

Example alert-lambda.json – Send anomaly alerts to a Lambda function

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:${Region}:${Account}:function:${FunctionName}"
            ]
        }
    ]
}
```

Values contained in brackets are placeholders for account-specific information such as resource names and account IDs. For more information, see Sample policies in the GitHub repo.

Sample AWS CloudFormation templates

The GitHub repository for this guide provides sample AWS CloudFormation templates that you can use to automate the creation of service roles. The templates use parameters and naming patterns to apply least-privilege permissions where possible.

The following sample template creates a service role that gives Lookout for Metrics permission to access S3 buckets, SNS topics, and Lambda functions that have names prefixed with `lookoutmetrics-`.

Example servicerole-s3.yml – Amazon S3 and Lambda permissions

```
Parameters:
  bucketName:
    Description: Bucket name
    Type: String
    Default: lookoutmetrics-*
  queueName:
    Description: Queue name
    Type: String
    Default: lookoutmetrics-*
  functionName:
```

```
    Description: Function name
    Type: String
    Default: lookoutmetrics-*
Resources:
  serviceRole:
    Type: AWS::IAM::Role
    Properties:
      Policies:
        - PolicyName: read-s3
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action:
                  - s3:ListBucket
                  - s3:GetBucketAcl
                Resource: !Sub arn:${AWS::Partition}:s3:::${bucketName}
              - Effect: Allow
                Action:
                  - s3:GetObject
                  - s3:GetBucketAcl
                Resource: !Sub arn:aws:s3:::${bucketName}/*
        - PolicyName: invoke-function
          PolicyDocument:
            Version: 2012-10-17
            Statement:
            - Effect: Allow
              Action: lambda:InvokeFunction
              Resource: !Sub arn:${AWS::Partition}:lambda:${AWS::Region}:
${AWS::AccountId}:${functionName}
...
```

For more information, see Sample templates in the GitHub repo.

Working with detectors

In Amazon Lookout for Metrics, a *detector* is a resource that monitors a dataset and identifies *anomalies* (data that falls outside of the expected range). Detectors use machine learning (ML) to find patterns in business data, and to distinguish between expected variations in data and legitimate anomalies. A detector can monitor a dataset that contains metrics data that you manage in Amazon Simple Storage Service (Amazon S3), live data from another service such as Amazon CloudWatch, or events from a database. When new data points fall outside of the expected range, the detector records the anomaly and sends an alert.

A dataset is a collection of timestamped data points that can each have multiple metrics and attributes. You choose one of the metrics to be the *measure*, which is the primary metric that the detector monitors for anomalies.

You can also configure up to 10 additional attributes as *dimensions*. Dimensions are additional information that the detector uses to segment anomalies, filter the dataset, and identify contributing factors.

For example, you can choose a field named `availability` for a measure. If you don't choose a dimension, the detector monitors availability across all records. If you choose a field named `country` for a dimension, then the detector monitors availability in each country as a separate metric: *availability in Canada*, *availability in Italy*, and so on.

Alternately, you can create a filter on `country` that includes only the countries that you are interested in monitoring. In this case, `availability` is monitored only in the countries that are included in the filter.

Detectors primarily work against live data. A detector analyzes new data periodically to find anomalies in measure values. The *interval* at which it analyzes data can be between 5 minutes and 1 day. To allow time for the datasource to collect all data before analysis starts, you also configure a *delay* on the dataset. At the end of an interval, the detector waits for the duration of the delay before analyzing data.

When you create a detector, you can also provide *historical data*. If you provide historical data, the detector uses it to learn patterns and relationships between fields in your data. If not, the detector spends several intervals learning on live data.

Every time it runs, the detector analyzes all of the data generated during the interval, identifies anomalous data points, and assigns a severity score to each. If the severity of an anomaly exceeds a

threshold, the detector sends an *alert*. You can configure alerts to send a notification to an Amazon Simple Notification Service (Amazon SNS) topic, or to invoke an AWS Lambda function. If you get too many or too few results, you can change the threshold that triggers the alert.

Topics

Setting up a detector

An *anomaly detector* is an Amazon Lookout for Metrics resource that monitors a dataset to find anomalies.

By default, detectors use a key managed by Lookout for Metrics. To use a key that you manage in your own account, create a symmetric key in AWS KMS and grant Lookout for Metrics permission to use it.

Topics

Creating a detector

To create a detector, you provide a name, description, and interval. To start the detector and begin finding anomalies, add a dataset and notifications and activate the detector.

To create a detector

1. Open the Lookout for Metrics console Detectors page.
2. Choose **Create detector**.
3. Provide the following information:.

   - **Name** – A name for the detector.
   - **Description** – A description of the detector.
   - **Interval** – The amount of time between attempts to detect anomalies.
4. (Optional) Lookout for Metrics encrypts all imported data with an AWS Key Management Service (AWS KMS) key. To choose an encryption key, select **Customize encryption settings**. You can't change the encryption key later.
5. Choose **Create**.

Detector Statuses

After creating a detector, Lookout for Metrics initializes the detector, uses data for training, and activates the detector so that it can begin identifying anomalies. When you create a detector, Lookout for Metrics displays the following statuses:

1. **Initializing** - Lookout for Metrics begins initializing your anomaly detector based on your datasource, measure, and dimension configurations. It also ingests all historical data that you provided.
2. **Learning** - After initialization, the detector begins learning by ingesting continuous data. Learning time varies depending on the detector's interval setting and whether you provided historical data.
3. **Active** - After the detector has ingested a sufficient amount of data, the status changes to **Active**”and the detector begins identifying anomalies.

Lookout for Metrics usually requires at least 300 cumulative data points before a detector's status changes to **Active**. For example, if you provide 200 data points of historical data, the detector need to ingest 100 data points of continuous data before it is active. For the daily time interval, the detector's status changes to **Active** after 14 days of learning.

If no historical data is available, the learning stage takes the following amount of time:

- **5-minute interval**: 25 hours
- **10-minute interval**: 50 hours
- **1-hour interval**: 12.5 days
- **Daily interval**: 14 days

If you are backtesting, also refer to the backtesting data requirements.

Creating a dataset

A detector imports data from a datasource, transforms it, and stores it in a dataset. The datasource can be an Amazon Simple Storage Service (Amazon S3) bucket that you store data in, a database, or another AWS service that Lookout for Metrics supports.

> **Note**
>
> To communicate with other services, Lookout for Metrics needs permissions from an AWS Identity and Access Management (IAM) service role. When you use the console to configure a dataset, you can create a new service role or choose an existing one. For more information, see Service roles for Amazon Lookout for Metrics.

To create a dataset

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Create dataset**.
4. Provide the following information:

   - **Name** – A name for the dataset.
   - **Description** – A description of the dataset.
   - **Timezone** – The timezone where the data is generated. When a detector analyzes data, it verifies that each data point falls within the current interval.
5. Choose a datasource. The datasource can be an Amazon S3 bucket, an AWS service that Lookout for Metrics supports, or a database.
6. Configure the datasource and then choose **Next**. Options vary depending on the datasource that you choose. Common settings include the following:

   - **Interval** – The amount of time between analysis attempts. Use the same setting as the detector's interval.
   - **Offset** – The minimum number of seconds that the detector waits at the end of each interval before accessing data. Using an offset is supported only for Amazon S3, Amazon Redshift, and Athena datasources.
   - **Permissions** – A service role that gives Lookout for Metrics permission to access either the datasource or a secret that contains credentials for the datasource.
7. Provide the metrics that the detector analyzes:

   - **Measures** – The primary metrics that the detector analyzes to find anomalies.

- **Dimensions** – The secondary metrics, which segment the data by, for example, location or demographic.
- **Timestamp** – The metric that specifies when the data point was created.

8. Choose **Next**.
9. Choose **Save and activate**.

To get started with Amazon S3 as a datasource, see Managing a dataset in Amazon S3. For other datasources, see Using Amazon Lookout for Metrics with other services.

Managing detectors

You manage detectors in the Lookout for Metrics console.

When you create a detector, you configure just a name, description, and interval. Optionally, you can choose an encryption key to use with the dataset's data, instead of the service's default key. After the detector is created, you can add a dataset and activate the detector to start finding anomalies.

Topics

Viewing detector logs

When activation is complete, the detector analyzes data after each interval. If there are any anomalies in the data for an interval, the detector assigns each a severity score. If the score exceeds an alert target's threshold, the detector sends an alert to that target. You can view the results of each analysis in the detector log.

To view the detector log

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Scroll down to the bottom of the page.

Deleting detectors

To stop analyzing data and delete it from Lookout for Metrics, delete the detector. This also deletes the detector's dataset and alerts.

To delete a detector

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.

3. Choose **Delete**.

You can't recover a deleted detector or its subresources. Data stored in Amazon Lookout for Metrics is completely deleted within 90 days of the detector being deleted.

Deactivating detectors

You can deactivate detectors if its status is `LEARNING` or `ACTIVE`. To reconfigure a detector, you must first deactivate the detector.

To deactivate a detector

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Deactivate detector**.

Deactivating a detector does not delete the detector or its subresources. If you are using the Lookout for Metrics SDK, use the DeactivateAnomalyDetector operation.

Reactivating detectors

Failed and inactive detectors can be edited and reactivated. You can reconfigure the detector and dataset before reactivating the detector. If a detector is active, you can deactivate, reconfigure, and then reactivate the detector.

A detector can only be reactivated if its status is `FAILED` or `INACTIVE`. Detectors that are `INITIALIZING`, `LEARNING`, or `ACTIVE` cannot be reactivated.

When reconfiguring the detector, the Detector name (`AnomalyDetectorName`) and Tags fields are not editable. You can change the KMS key (`KmsKeyArn`) if the detector is inactive. If the detector failed, the KMS key cannot be changed.

When reconfiguring the dataset, the Detector Arn (`AnomalyDetectorArn`), Dataset name (`MetricSetName`), and Tags fields are not editable.

To reactivate a detector

1. Open the Lookout for Metrics console Detectors page.
2. Choose the failed or inactive detector.

3. **[Optional]** To reconfigure the detector, choose **Edit** under **Create a detector**.
4. **[Optional]** To reconfigure the dataset, choose **Edit** under **Add a dataset**.
5. Choose **Activate detector**.

If you are using the Lookout for Metrics SDK, use the UpdateAnomalyDetector operation to edit the detector settings and the UpdateMetricSet operation to edit the dataset. Use ActivateAnomalyDetector to reactivate the detector.

Managing a dataset in Amazon S3

You can use Amazon Simple Storage Service (Amazon S3) to store data for an Amazon Lookout for Metrics detector. With Amazon S3, you have complete control over your data's format and content. You can preprocess your data before handing it off to Lookout for Metrics, and aggregate data from multiple sources.

> **Note**
>
> For information about using other AWS services as a datasource, see Using Amazon Lookout for Metrics with other services.

You can provide two types of data to a detector: *continuous data* and *historical data*. A detector monitors continuous data to identify anomalies. You write continuous data to Amazon S3 as it is generated, to a path that represents the current interval. At the end of each interval, the detector reads data from the interval and analyzes it. The following example shows one possible path structure for continuous data with a 5-minute interval.

```
s3://my-lookoutmetrics-dataset-123456789012/
    continuous/20201225/1520/data.jsonl
    continuous/20201225/1525/data.jsonl
    continuous/20201225/1530/data.jsonl
```

In this example, data for each 5-minute interval is stored in a single file named `data.jsonl` at a path that represents the interval. `continuous/20201225/1520/` is a path for data generated in the 5-minute period starting at 3:20 PM on December 25th, 2020. Every 5 minutes, a new path is used.

Historical data is a collection of data stored at a single path in Amazon S3 that represents many previous intervals. You can provide historical data to a detector to train it prior to processing continuous data. Historical data should have metrics from hundreds or thousands of intervals in one or more files. The following example shows historical data in separate files for each month at the path `historical/`.

```
s3://my-lookoutmetrics-dataset-123456789012/
    historical/data-202009.jsonl
    historical/data-202010.jsonl
```

```
    historical/data-202011.jsonl
```

If you provide historical data, but not continuous data, the detector operates in backtest mode. Prior to creating an application or pipeline that generates continuous data, you can run a backtest to see how the detector works with your historical data.

You can store your data in CSV or JSON lines format. Both formats support one record per line of text. With CSV format, each field in a line is plain text separated by a comma or other supported delimiter. A CSV file can have a header row with field names, or define field names in the dataset. With JSON lines format, each line is a JSON object with key-value pairs that define the name and value of each field.

Topics

- Structuring continuous and historical data
- CSV data
- JSON lines
- Providing historical data
- Path template keys

Structuring continuous and historical data

When you configure an Amazon S3 bucket as a datasource, you provide a *path template* that tells the detector where to find the continuous data. Consider the following example path structure.

```
s3://my-lookoutmetrics-dataset-123456789012/
    continuous/20201225/1520/
    continuous/20201225/1525/
    continuous/20201225/1530/
    historical/
```

For historical data for this example, the path is `s3://my-lookoutmetrics-dataset-123456789012/historical`. Lookout for Metrics looks for data files directly under `historical` and ignores subpaths.

For continuous data, the detector needs to know where to look for data for the current interval. The path template for the example structure is `s3://my-lookoutmetrics-`

`dataset-123456789012/continuous/{{yyyyMMdd}}/{{HHmm}}`. The letters in double brackets represent parts of the path that change depending on the date and time. Construct a path template with the following keys.

- `yyyy` – The 4-digit year
- `MM` – The 2- digit month
- `HH` – The 2-digit hour (in 24-hour format)
- `mm` – The 2-digit minute

For a complete list of supported keys, see Path template keys.

Within a path for a single interval, data can be stored in one tor more text files. The detector uses only data with timestamps that fall within the interval for analysis. The detector uses the dataset's timezone to determine if data belongs to the current interval and ignores data that falls outside of the expected range.

CSV data

The following is an example of a correctly formatted CSV input file. Notice that this sample includes headers for each parameter (`target_value` is a metric and `item_id` is a dimension). Headers are optional, but recommended.

```
item_id,timestamp,target_value
item_001,2020-05-07,1591.702780
item_002,2020-05-07,2342.481244
item_003,2020-05-07,1794.275162
item_004,2020-05-07,2716.692446
...
```

The following is an example of a correctly formatted CSV file with headers. Here, there are two measures (`target_1` and `target_2`) and two dimensions (`item_id` and `store_id`).

```
item_id,store_id,timestamp,target_1,target_2
item_001,store_001,2020-04-01 00:00:00,2117.0433697865165,27.521563807224712
item_002,store_002,2020-04-01 00:00:00,2221.312595828157,28.87706374576604
item_002,store_001,2020-04-01 00:00:00,4224.364287792719,54.91673574130534
item_003,store_002,2020-04-01 00:00:00,1420.3210031715096,18.464173041229625
item_001,store_002,2020-04-01 00:00:00,3222.8693491500876,41.89730153895114
```

```
...
```

JSON lines

The following is an example of a correctly formatted JSON lines input file, with `target_value` as a metric and `item_id` as a dimension.

```
{ "item_id":"item_001" , "timestamp" : "2020-05-07", "target_value" : 1591.702780 }
{ "item_id":"item_002" , "timestamp" : "2020-05-07", "target_value" : 2342.481244 }
{ "item_id":"item_003" , "timestamp" : "2020-05-07", "target_value" : 1794.275162 }
{ "item_id":"item_004" , "timestamp" : "2020-05-07", "target_value" : 2716.692446 }
...
```

The following is another example of a correctly formatted JSON lines file. Here, `target1` and `target2` are metrics and `item_id` and `store_id` are dimensions.

```
{"item_id": "item_001", "store_id": "store_001", "timestamp": "2020-04-01 00:00:00",
 "target1": 2117.0433697865165, "target2": 27.521563807224712}
{"item_id": "item_002", "store_id": "store_002", "timestamp": "2020-04-01 00:00:00",
 "target1": 2221.312595828157, "target2": 28.87706374576604}
{"item_id": "item_002", "store_id": "store_001", "timestamp": "2020-04-01 00:00:00",
 "target1": 4224.364287792719, "target2": 54.91673574130534}
{"item_id": "item_003", "store_id": "store_002", "timestamp": "2020-04-01 00:00:00",
 "target1": 1420.3210031715096, "target2": 18.464173041229625}
{"item_id": "item_001", "store_id": "store_002", "timestamp": "2020-04-01 00:00:00",
 "target1": 3222.8693491500876, "target2": 41.89730153895114}
...
```

Providing historical data

Your detector imports continuous data from your Amazon S3 bucket, stores it in its dataset, and uses it for learning. *Learning* is the process of analyzing data over multiple intervals to identify patterns and to distinguish between legitimate anomalies and uncommon but expected variations. A detector can also learn by using *historical data*.

To train a detector before it starts processing continuous data, you can provide historical data that represents up to 2,500 previous intervals. Historical data must fall within a timeframe that varies depending on the dataset's interval.

- 5-minute interval – 3 months

- 10-minute interval – 6 months
- 1-hour interval – 3 years
- 1-day interval – 5 years

If you don't specify a path for historical data when you create the dataset, the detector looks for data from previous intervals in the continuous data path. If available, it uses this data for learning to reduce the amount of time that it takes to start finding anomalies.

Path template keys

The following table lists the supported keys for continuous data path templates. A path template is an Amazon S3 URI that has placeholder keys in double curly brackets, which represent the folder names in the bucket that change for each interval.

Letter	Date or time component	Presentation	Examples
y	Year	Year	1996; 96
Y	Week year	Year	2009; 09
M	Month in year	Month	July;Jul;07
d	Day in month	Number	10
a	AM/PM marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
k (with AM/PM marker)	Hour in AM/PM (0-11)	Number	11
m	Minute in hour	Number	30
s	Second in minute	Number	55

Example path structure – daily interval

For this example, the path template is `s3://my-lookoutmetrics-dataset-123456789012/continuous/{{yyyy}}/{{MM}}/{{dd}}`. The `continuous` folder has a subfolder structure that indicates the year, month, and day of each one day interval.

```
s3://my-lookoutmetrics-dataset-123456789012/
    historical/
      2020Q2.csv
      2020Q3.csv
    continuous/
      2020/12/01/
        20201201-01.csv
      2020/12/02/
        20201202-01.csv
        20201202-02.csv
```

Working with anomalies

An *anomaly* is a data point that the detector doesn't expect based on its understanding of your data. An anomaly isn't necessarily good or bad, it's just unexpected. A detector learns over time to more accurately identify anomalies based on patterns that it finds in your data.

Lookout for Metrics organizes anomalies that affect the same measure during the same interval groups and displays them as a single event. To get details about anomalies, view them in the Amazon Lookout for Metrics console.

To view anomalies

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Anomalies**.
4. Use the options on the page to sort and filter the list of anomalies.
   - **Search bar** – To filter the list of anomalies by title, enter the name of a measure.
   - **Severity threshold** – To filter out lower-severity anomalies but maintain the display order, use the severity threshold slider.
   - **Sort columns** – To sort by title, severity score, or timestamp, choose the name of the relevant column.
5. For details, choose an anomaly from the list.

The detector assigns a severity score between 0 and 100. The severity score indicates how far the data point is outside of the expected range based on the data that the detector has analyzed. The higher the severity score, the more unexpected the anomaly is.

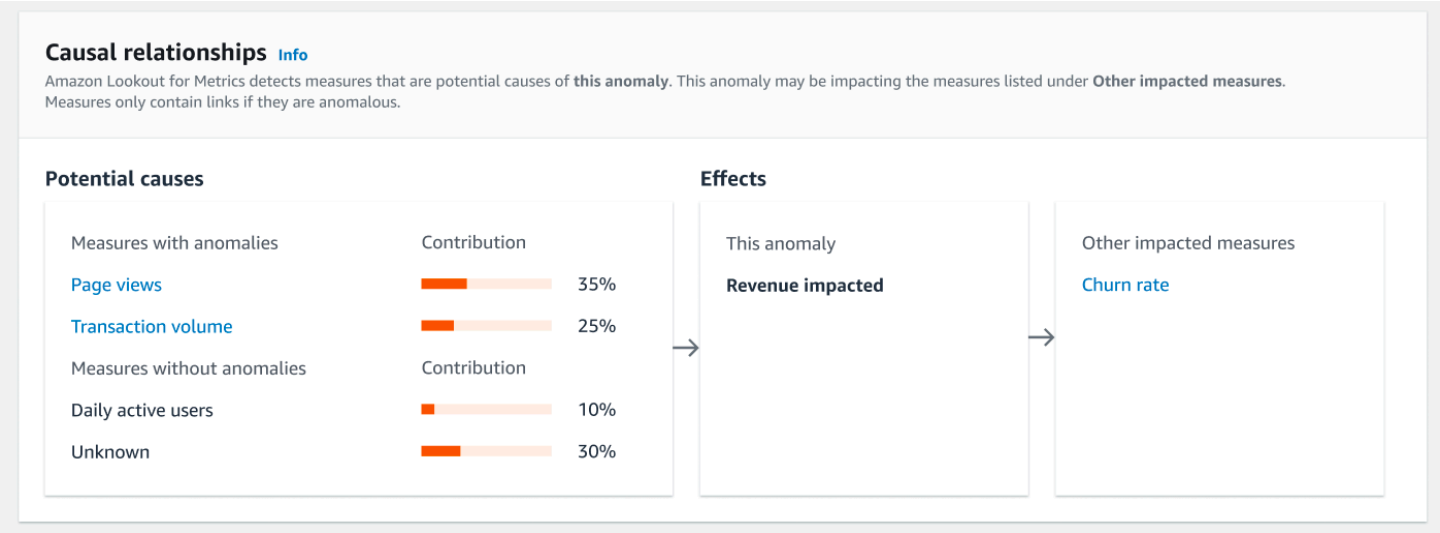
Topics

Causal relationships

The Causal relationships section identifies the measures in your datasets that are potentially causing anomalies at specific time points, and also identifies other measures that may be impacted by the anomaly.

During anomaly detection periods, Lookout for Metrics identifies measures that are likely causing particular anomalies, determines the direction of influence, and quantifies the level of contribution.

Causal relationships is divided into two sections: Potential causes and Effects. Potential causes are measures that are potentially causing the inspected anomaly. The Effects section shows the inspected anomaly and any measures that may be impacted by the anomaly.



Potential causes

All measures listed under Potential causes are contributing to the anomaly. **Measures with anomalies** are anomalous measures, and you can use the links to learn more about those anomalies. **Measures without anomalies** are measures that do not contain anomalies, but still contribute to the anomaly you are inspecting.

You can gauge the impact each measure had in causing the anomaly by comparing the contribution percentages of the listed measures.

The unexplained portion of the anomaly is represented as “Unknown”. This is the contribution of factors that are not captured in the measures.

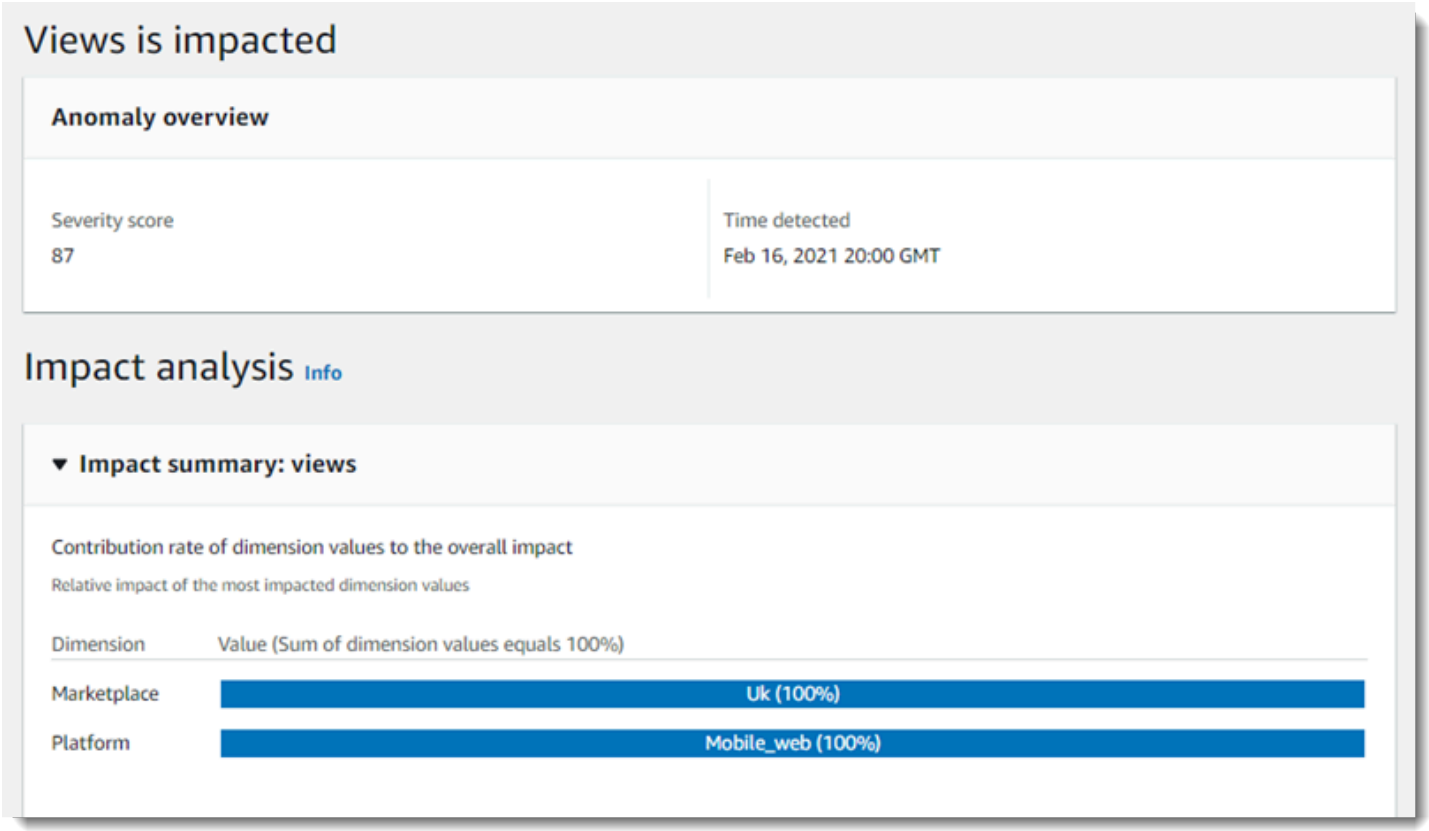
Effects

The Effects section shows the anomaly you are reviewing and any measures that may be impacted by the anomaly.

The anomaly you are reviewing is listed under **This anomaly**, and any measures that may be impacted by this anomaly are listed under **Other impacted measures**. You can use the links to learn more about the anomalies within those measures.

Affected measures

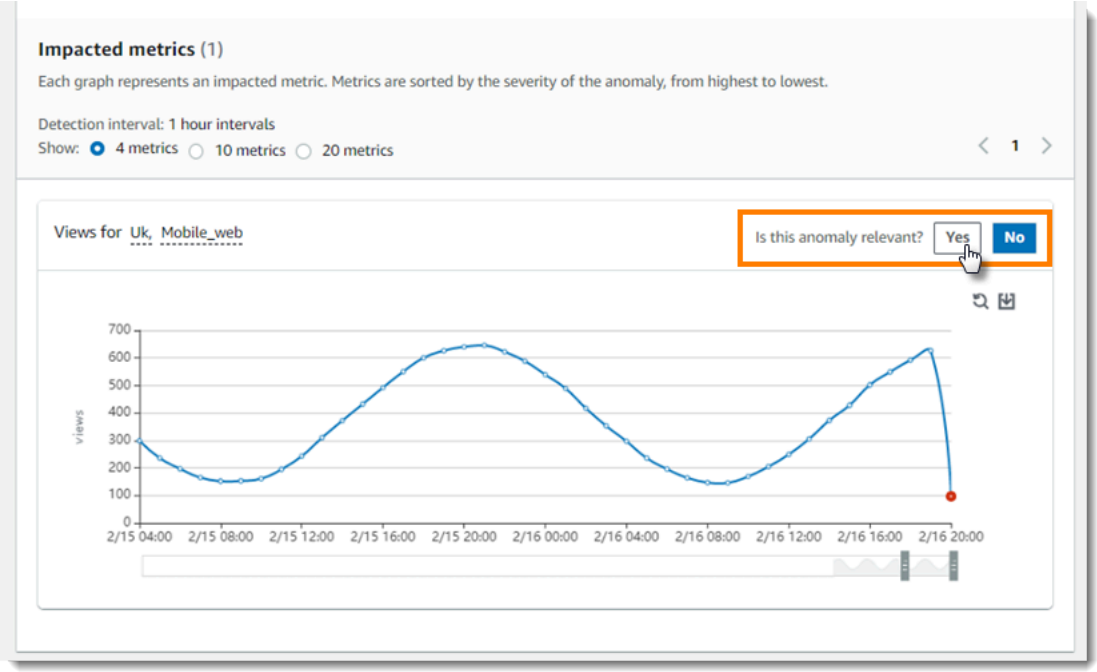
The title of an anomaly indicates the measure that was affected. If your dataset has dimensions, the values of each dimension are also part of the metric. In the following example, the number of `views` recorded was anomalous. The dataset has two dimensions, `marketplace` and `platform`, with values that indicate where the metric was recorded and the kind of device that was in use. The number of views was found to be affected only in the UK marketplace (**Uk**) on mobile platforms (**Mobile_web**). The **100%** for each dimension value indicates that the anomaly was only observed for one value within each dimension.



Below the impact summary, there is a graph of the affected metric over time. In this example, the number of mobile views in the UK dropped sharply between 7 PM and 8 PM. This drop was unexpected considering the marketplace, platform, and time of day.

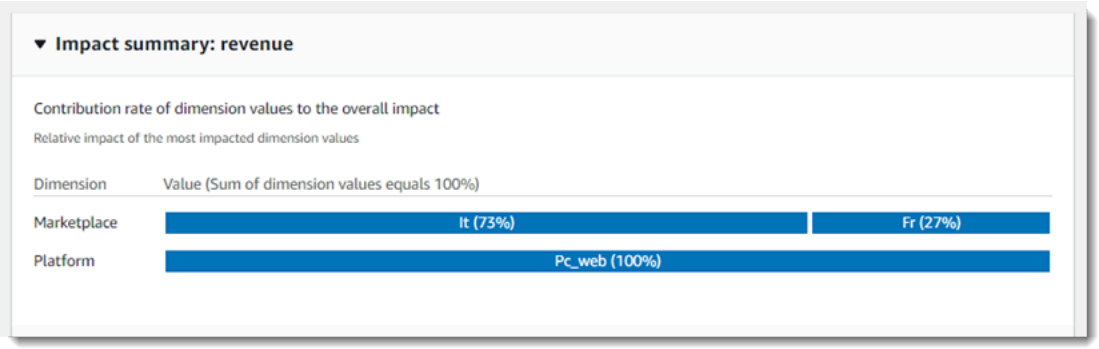
Providing feedback

This drop could indicate an issue in the marketplace and platform, or an external factor that isn't represented in the data. To indicate whether the anomaly is relevant, use the feedback buttons above the graph. When the detector finds similar anomalies later, it will consider the feedback as it determines the severity score.



Contributing metrics

If the detector finds anomalies in multiple metrics for the same measure, it groups them together into a single event. In the following example, metrics for `revenue` on the PC platform are affected on PC in both Italy and France, but to a greater degree in Italy.



In this case, *revenue on PC in Italy* is one metric, and *revenue on PC in France* is a second. If mobile revenue was also affected in both marketplaces, there would be four metrics, and the platform dimension would show the impact of PC vs mobile revenue.

Working with alerts

Creating alerts

Amazon Lookout for Metrics detectors find anomalies in data. When an anomaly is severe, the detector can send details about it to another AWS service or resource. You can configure a detector to run an AWS Lambda function to process anomaly alerts, or send details to an Amazon Simple Notification Service (Amazon SNS) topic. Amazon SNS can then send the information to email subscribers or an HTTP endpoint, among numerous other supported destinations.

A severity threshold determines when the detector sends anomaly alerts. If you get anomaly alerts for anomalies that are not interesting, you can increase the threshold. If you don't get enough alerts, you can lower it.

When you create an alert, you can specify a subset of measures or dimensions that are used to create the alert. That is, you only want to be alerted to anomalies in a certain region or for a specific service, not any anomaly found in the data. Use the **Criteria for sending the alert** area to specify the measures, dimensions, and sensitivity score that define the alert.

For example, imagine that you run an e-commerce business that tracks the measures `revenue` and `orderRate` in the dimensions `marketplace` (country) and `category`. You can configure the alert criteria to notify you only when anomalies are detected for specific combinations of these measures and dimensions. Suppose that you specified the following criteria:

- **Measures** – select both `revenue` and `orderRate`.
- **Dimensions**
  - `marketplace` – select US and CA.
  - `category` – select `Electronics` and `Apparels`.
- **Sensitivity score** – set to 60.

In this case, you'll receive an alert only when all of these conditions are met. That means only when

- **Measure** is `revenue` or `orderRate`,
- **AND** `marketplace` is either US or CA.
- **AND** `category` is either `Electronics` or `Apparels`.
- **AND** the sensitivity score is greater than or equal to 60.

To send anomaly alerts, a detector uses a service role. When you use the console to create alerts, you can create a service role or choose one you already have. If you don't have permission to create roles, ask an administrator to create a service role for Lookout for Metrics.

To create an alert

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Add alert**.
4. Configure the following options.

   - **Alert name** – The name of the alert. Alert names must be unique across all detectors in a Region.
   - **Description** – A description of the alert.
   - **Criteria** – The measures, dimensions, and severity score that trigger an alert.
     - **Measure** – Provide up to 5 measures that you want to receive alerts for when an anomaly is detected.
     - **Dimension** – Provide up to 5 dimensions to monitor for anomalies. You can provide up to 10 values per dimension.
     - **Severity score** – The severity score that the anomaly must reach for the detector to send an anomaly alert.
   - **Channel** – The destination service.
   - **SNS topic** or **Lambda function** – The resource in the target service that receives the anomaly alert.
   - **Role** – A service role that allows the detector to send alerts to the resource.
5. Choose **Add alert**.

When the detector finds an anomaly that meets the filter criteria, it sends an anomaly alert. An anomaly alert is a JSON document with details about the metrics that were affected by the anomaly. The following example shows an anomaly alert where `revenue` is affected in the `electronics` or `apparels` categories in two marketplaces, UK (United Kingdom) or FR (France), and when shipping is either `standard` or `priority`.

Example alert (line endings and indentation added)


```
{
  "anomalyDetectorArn": "arn:aws:lookoutmetrics:us-
west-2:123456789101:AnomalyDetector:testBugBash",
  "anomalyScore": 83.33,
  "consoleUrl": "https://us-west-2.console.aws.amazon.com/
lookoutmetrics/home?region=us-west-2#arn:aws:lookoutmetrics:us-
west-2:123456789101:AnomalyDetector:myDetector/anomalies/anomaly/60359687-afb8-4c46-
af3d-aea03614a53f",
  "anomalyGroupId": "60359687-afb8-4c46-af3d-aea03614a53f",
  "alertName": "myAlertName",
  "alertArn": "arn:aws:lookoutmetrics:us-west-2:123456789101:Alert:myAlertName",
  "alertEventId": "arn:aws:lookoutmetrics:us-
west-2:123456789101:Alert:myAlertName:event/60359687-afb8-4c46-af3d-aea03614a53f",
  "alertDescription": "SNS Alert for anomalies in Revenue",
  "impactedMetric": {
    "metricName": "revenue",
    "dimensionContribution": [
      {
        "dimensionName": "category",
        "dimensionValueContributions": [
          {
            "dimensionValue": "apparels",
            "valueContribution": 68
          },
          {
            "dimensionValue": "electronics",
            "valueContribution": 32
          }
        ]
      },
      {
        "dimensionName": "marketplace",
        "dimensionValueContributions": [
          {
            "dimensionValue": "UK",
            "valueContribution": 68
          },
          {
            "dimensionValue": "FR",
            "valueContribution": 32
          }
```

```
        ]
      },
      {
        "dimensionName": "shipping",
        "dimensionValueContributions": [
          {
            "dimensionValue": "standard",
            "valueContribution": 68
          },
          {
            "dimensionValue": "priority",
            "valueContribution": 32
          }
        ]
      }
    ],
    "relevantTimeSeries": [
      {
        "timeSeriesId":
"978d818762cf3669fajhg8398jk9fd036310832a18bbc4f3f5b17c1f0bf1086683eda16678503580d704f518",
        "dimensions": [
          {
            "dimensionName": "category",
            "dimensionValue": "electronics"
          },
          {
            "dimensionName": "marketplace",
            "dimensionValue": "FR"
          },
          {
            "dimensionName": "shipping",
            "dimensionValue": "priority"
          }
        ],
        "metricValue": 650
      },
      {
        "timeSeriesId":
"f1b5e0a7f64b249jma947nf0wnf7qncsoa48ea08eb0f0afc17a12a11c423944e871659be0748c98b1f039cad5bda0
        "dimensions": [
          {
            "dimensionName": "category",
            "dimensionValue": "apparels"
          },
```

```
          {
            "dimensionName": "marketplace",
            "dimensionValue": "UK"
          },
          {
            "dimensionName": "shipping",
            "dimensionValue": "standard"
          }
        ],
        "metricValue": 1200
      }
    ]
  },
  "timestamp": "2022-01-15T04:25Z[UTC]"
}
```

In the preceding example, the detector identified unusually high revenue values (650 and 1200) in UK and FR for category apparels and electronics. The value for UK, apparels, and standard shipping was farther from the expected value, so its contribution to the anomaly (68%) is greater than the value for FR, electronics and priority shipping. Overall, the anomaly for the two metrics had an severity score of 83.33.

The severity score is a measurement of how unexpected the observed metric values are based on the detector's understanding of your data. It takes into consideration when the anomaly occurred, such as the time of day, and how many metrics were affected.

As the detector learns more about your data, anomalies for similar events might have higher or lower severity scores. You can guide its learning by providing feedback on affected metrics in an anomaly. For more information, see Working with anomalies.

Updating alerts

After an alert is created, you can update the alert if it does not meet your needs. You can change whether AWS Lambda or Amazon Simple Notification Service is used to send updates, the alert description, the alert sensitivity threshold, and the alert criteria.

To update an alert

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Alerts** in the **Detectors** pane.

4. Choose an alert.
5. Configure the following options.

   - **Description** – A description of the alert.
   - **Criteria** – The measures, dimensions, and severity score that trigger an alert.
     - **Measure** – Provide up to 10 measures that you want to receive alerts for when an anomaly is detected.
     - **Dimension** – Provide up to 10 dimensions to monitor for anomalies. You can provide up to 10 values per dimension.
     - **Severity score** – The severity score that the anomaly must reach for the detector to send an anomaly alert.
   - **Channel** – The destination service.
   - **SNS topic** or **Lambda function** – The resource in the target service that receives the anomaly alert.
   - **Role** – A service role that allows the detector to send alerts to the resource.
6. Choose **Save**.

Tagging Lookout for Metrics resources

Organize your Amazon Lookout for Metrics resources by owner, project or department with tags. Tags are key-value pairs that are supported across AWS services. You can use tags to filter resources, create least-privilege permissions policies, and add detail to billing reports. Lookout for Metrics also supports tag-based authorization. With tag-based authorization, you can create permissions policies that limit a user's access to resources with specific tags. For more information about tag-based authorization, see Controlling access to AWS resources using resource tags in the *IAM User Guide*.

You can add tags to detectors, datasets, and alerts when you create them, or you can add tags to existing resources. You can use the Lookout for Metrics console or manage tags with the Lookout for Metrics API. Start by tagging your detectors to organize them into logical groups.

For more information about tags, see Tagging AWS resources in the *Amazon Web Services General Reference*.

To add tags to a detector in the Lookout for Metrics console

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Tags**.
4. Choose **Manage tags**.
5. Enter a key and value. For example, `Department` and `Marketing`.
6. To add additional tags, choose **Add tag**.
7. Choose **Save**.

Tags apply to each detector, dataset, and alert individually. They are not shared or inherited.

Sections

- Using tags (Lookout for Metrics API)
- Tag key and value requirements

Using tags (Lookout for Metrics API)

When you create resources with the CreateAnomalyDetector, CreateMetricSet and CreateAlert operations, you can include tags with the `--tags` option. The following example shows how to apply tags when creating an anomaly detector with the AWS Command Line Interface (AWS CLI).

```
$ aws lookoutmetrics create-anomaly-detector --anomaly-detector-name my-detector \
  --anomaly-detector-config AnomalyDetectorFrequency=PT10M \
  --anomaly-detector-description "10-minute S3 detector" \
  --tags Department=Marketing,CostCenter=1234ABCD
{
    "AnomalyDetectorArn": "arn:aws:lookoutmetrics:us-
east-2:123456789012:AnomalyDetector:my-detector"
}
```

To add tags to an existing resource, use the TagResource operation.

```
$ aws lookoutmetrics tag-resource --resource-arn arn:aws:lookoutmetrics:us-
east-2:123456789012:AnomalyDetector:my-detector \
  --tags Department=Marketing,CostCenter=1234ABCD
```

To remove tags, use the UntagResource operation.

```
$ aws lookoutmetrics untag-resource --resource-arn  arn:aws:lookoutmetrics:us-
east-2:123456789012:AnomalyDetector:my-detector \
  --tag-keys Department
```

To view tags, you can use the following API operations:

- ListTagsForResource – View the tags associated with a resource.

  ```
  $ aws lookoutmetrics list-tags-for-resource --resource-arn arn:aws:lookoutmetrics:us-
  east-2:123456789012:AnomalyDetector:my-detector
  {
      "Tags": {
          "Department": "Marketing",
          "CostCenter": "1234ABCD"
      }
  }
  ```

- ListAnomalyDetectors, ListMetricSets, ListAlerts – Get a list of resources with tag information.

```
$ aws lookoutmetrics list-anomaly-detectors
{
    "AnomalyDetectorSummaryList": [
        {
            "Status": "INACTIVE",
            "AnomalyDetectorName": "my-detector",
            "Tags": {
                "Department": "Marketing",
                "CostCenter": "1234ABCD"
            },
            "LastModificationTime": 1612994728.528,
            "CreationTime": 1612994728.528,
            "AnomalyDetectorArn": "arn:aws:lookoutmetrics:us-
east-2:123456789012:AnomalyDetector:my-detector"
        }
    ]
}
```

Tag key and value requirements

The following requirements apply to tags for Lookout for Metrics resources:

- Maximum number of tags per resource – 50
- Maximum key length – 128 Unicode characters in UTF-8
- Maximum value length – 256 Unicode characters in UTF-8
- Tag keys and values are case sensitive.
- Your tag keys and values can't start with `aws:`. AWS services apply tags that start with `aws:`, and those tags can't be modified. They don't count towards tag limits.
- Tag keys and values can contain the following characters: A-Z, a-z, 0-9, space, and _ . : / = + @ - (hyphen). This is the standard set of characters available across AWS services that support tags. Some services support additional symbols.

Working with filters

When you add a dataset to a detector, you can specify a subset of dimensions that are used to create filters. That way you only monitor for anomalies in a certain region or for a specific service, not any anomaly found in the data. Use the **Filters** area to specify the dimensions and values that define the filter.

For example, imagine that you run an ecommerce business that tracks the measures `revenue` and `orderRate` in the dimensions `marketplace` (country) and `category`. You can create a filter so that only certain countries or categories are monitored. Suppose that you specified the following dimension filtering criteria:

- **Measures** – select both `revenue` and `orderRate`.
- **Dimensions**
  - `marketplace` – select US and CA.
  - `category` – select `Electronics` and `Apparels`.

In this case, you're monitoring only the records that meet all of these conditions. That means only when:

- **Measure** is `revenue` or `orderRate`,
- **AND** `marketplace` is either US or CA.
- **AND** `category` is either `Electronics` or `Apparels`.

Note that when creating a filter, dimensions are joined with the AND operation, and values within a dimension are joined with the OR operation.

Creating filters

When you add a datasource to a detector, you can specify one or more dimensions that are used to filter your data.

Create a filter in the Lookout for Metrics console:

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Add a dataset**.

4. Provide the information to **Choose a datasource** as outlined in Detector Statuses.
5. Choose **Next**.
6. Choose the measures that you want to monitor.
7. In **Dimension filters**, add up to five dimensions for the filter. For each dimension filter, you can include up to 10 values. Remember that dimensions are joined with the AND operation, and values are joined with the OR operation.
8. Choose **Next**.
9. Review your selections, then choose **Save and activate**.

You can create a filter using the `DimensionFilterList` object within the CreateMetricSet operation. For example, consider the following code.

Example

```
"DimensionFilterList": [
  {"Country": [
    { "Value": "US",
      "FilterOperation": "equals" },
    { "Value": "UK",
      "FilterOperation": "equals" }
  ]},
  {"Year": [
    { "Value": "2000",
      "FilterOperation": "equals" }
  ]}
]
```

In this example, you are filtering in all items where the dimension `year` is `2000` and the dimension `country` is either US or UK.

Using Amazon Lookout for Metrics with other services

Amazon Lookout for Metrics works with other AWS services to provide additional options for datasources and notification targets. In addition to reading metrics data from an Amazon Simple Storage Service (Amazon S3) bucket, you can create datasets from databases, observability services, and cloud applications. When a detector finds anomalies, it can invoke a Lambda function or send anomaly alerts to an Amazon SNS topic.

To use a cloud application such as Salesforce as a datasource, you use Amazon AppFlow. Amazon AppFlow is an AWS service that supports additional datasources by providing a *flow* that third party services can write to.

Lookout for Metrics integrates with other services to provide additional datasources and alert channels. Lookout for Metrics supports the following services as datasources.

- ??? – Analyze data retrieved by your queries.
- Amazon S3 – Read metrics data from text objects in a bucket. Supports using historical data for training and testing a detector.
- Amazon CloudWatch – Analyze CloudWatch metrics sent by an AWS service, AWS resource, or an application.
- Amazon Relational Database Service (Amazon RDS) – Analyze items in a relational database. The following database engines are supported:
  - **Amazon Aurora**
  - **MySQL**
  - **PostgreSQL**
  - **MariaDB**
  - **Microsoft SQL Server**
- Amazon Redshift – Analyze items in a Amazon Redshift data warehouse.
- Amazon AppFlow – Analyze data flow from a web service with an Amazon AppFlow data flow. The following services are supported:
  - **Salesforce**
  - **Marketo**
  - **Dynatrace**

- **Singular**
- **Zendesk**
- **Servicenow**
- **Infor Nexus**
- **Trendmicro**
- **Veeva**
- **Google Analytics**
- **Amplitude**

You can configure a detector to send notifications to the following services and resources.

- **AWS Lambda** – Invoke a Lambda function with an event that contains details about an anomaly.
- **Amazon Simple Notification Service (Amazon SNS)** – Send a notification to a topic that email, SMS, or web application subscribers. Lookout for Metrics supports the following application channels with Amazon SNS:

  - **PagerDuty**
  - **Slack**
  - **Datadog**

To interact with other AWS services, Lookout for Metrics uses permissions from an AWS Identity and Access Management (IAM) service role. A service role is an IAM role that you create to give a service permission to access AWS resources in your account. For more information, see Amazon Lookout for Metrics permissions.

Topics

- Using Amazon SNS with Lookout for Metrics
- Using Amazon S3 with Lookout for Metrics
- Working with Amazon Virtual Private Cloud

Using Amazon Athena with Lookout for Metrics

You can use Amazon Athena (Athena) as a datasource for an Amazon Lookout for Metrics detector. With Athena, you can choose columns to monitor (*measures*) and columns that segment measure values (*dimensions*). The detector monitors the values in these columns to find anomalies in your data.

To use an Athena database with Lookout for Metrics, the table must have a timestamp column that is partitioned for queries. This allows Lookout for Metrics to get records for an interval without scanning the entire table.

If you have a table with a large number of partitions that grows over time, consider using AWS Glue partition indexing and filtering. AWS Glue partition indexing allows Athena to optimize partition processing and improve query performance on highly partitioned tables. For more information, see AWS Glue partition indexing and filtering.

Partition projection with Amazon Athena is another option that speeds up query processing of highly partitioned tables. In partition projection, partition values and locations are calculated from configuration rather than read from a repository like the AWS Glue Data Catalog. For more information, see Partition projection with Amazon Athena.

The detector imports data at the end of each interval. You configure an **offset** to allow time after an interval ends for all data to be written. For example, if you choose an offset of 30 seconds, the detector waits 30 seconds after the end of each interval before reading data for that interval.

Before you configure the dataset, you need to know the following information.

- **Database name** – The name of the database where the source Athena table is kept.
- **Data catalog** – The name of the catalog where the source database is kept.
- **Table name** – The name of the table that contains the source data.
- **Workgroup name** – The name of the workgroup where the prepared statement is saved.
- **Amazon S3 results path** – The name of the Amazon S3 bucket where the Athena query results are stored. You can also use the result location that is configured for the workgroup.

To create an Athena dataset

1. Open the Lookout for Metrics console Detectors page.

2. Choose a detector.
3. Choose **Add dataset**.
4. Choose **Amazon Athena**
5. Follow the instructions to create the datasource.

To configure metrics in Lookout for Metrics, you choose columns to be measures and dimensions. Each measure is a column with a numerical value that you want to monitor for anomalies. Each dimension is a column with a string value that segments the measure(s). A metric in Lookout for Metrics is a combination of a measure value and a dimension value, aggregated within an interval. For example, *average availability in Colorado*, or *maximum temperature in furnace 17*.

The detector reads new data from Athena periodically, by getting objects from the folder for the most recent completed interval. If it detects any anomalies in the metrics for the interval, it records an anomaly and sends anomaly alerts, if configured.

When you activate the detector, it uses data from several intervals to learn, before attempting to find anomalies. For a five minute interval, the training process takes approximately one day. Training time varies depending on the detector's interval.

For more information about Athena, see Getting started with Amazon Athena in the Amazon Athena User Guide.

Sections

Configuring permissions

When you add an Athena dataset to your detector, the Lookout for Metrics console creates a service role with permission to read data from the bucket through Athena, and permission to use AWS Key Management Service (AWS KMS) to encrypt and decrypt data.

The console creates a role for the dataset, and a separate role for each alert that you configure. You can create a single role for the detector that gives it all of the permissions that it needs. To

use a custom role, create an IAM role that Lookout for Metrics has permission to assume, and add permission to use Athena and AWS KMS.

If you require federated access to call Amazon Athena, you might need to configure your own IAM roles. For more information, see Using Amazon Athena Federated Query in the Athena User Guide.

Training a detector with an Athena datasource

When using an Athena datasource, the following number of days are used to train a detector:

- **5 minute interval** – 10.42 days
- **10 minute interval** – 20.48 days
- **1 hour interval** – 125 days
- **1 day interval** – 455 days

For example, if a datasource has a 5 minute interval, data from the previous 10.42 days is used to train a detector. If the datasource contains fewer than 10.42 days of data, a detector is trained on all of the available data.

The datasource must meet the minimum data requirements in order to trian a detector.

Running a backtest

In backtest mode, a detector uses historical data to learn and find anomalies. You provide recent data for a large number of intervals at a single path. In backtest mode, Lookout for Metrics splits historical data into two subsets. 70 percent of the data is used to train the detector. The detector then analyzes the other 30 percent to identify anomalies. You can use test mode to validate the detector's results and verify its accuracy.

For backtest mode, you can provide between 285 and 3000 intervals worth of data. The data can be in one file or multiple files in the same folder. This gives the detector at least 200 intervals of data to learn with. The detector always uses older data for learning and newer data for testing.

Using Amazon AppFlow with Lookout for Metrics

You can use Amazon AppFlow to create a datasource for an Amazon Lookout for Metrics detector. With Amazon AppFlow, you can collect data from external cloud applications in a flow. A Lookout for Metrics detector reads data from the flow and monitors it for anomalies.

To use Amazon AppFlow with Lookout for Metrics, you must use an AWS Key Management Service (AWS KMS) encryption key that you create in your account. You can create a symmetric customer-managed key for your flow and detector in the AWS KMS console.

Lookout for Metrics supports the following cloud applications with Amazon AppFlow:

- **Salesforce**
- **Marketo**
- **Dynatrace**
- **Singular**
- **Zendesk**
- **Servicenow**
- **Infor Nexus**
- **Trendmicro**
- **Veeva**
- **Google Analytics**
- **Amplitude**

To create a flow

1. Open the Amazon AppFlow console.
2. Choose **Create flow**.
3. Enter a name for the flow.
4. For **Data encryption** choose **Customize encryption settings**.
5. For **Source details**, choose a datasource.
6. For **Destination details**, choose **Lookout for Metrics**.
7. For **Schedule frequency**, choose a frequency that matches your detector's interval.
8. Follow the instructions to configure the remaining options and then choose **Next**.

9. For **Field mapping**, you can choose **Map all fields directly**, or choose a subset of fields that contains the measures, dimensions, and timestamps that you will use with Lookout for Metrics.
10. Follow the instructions to complete flow setup and then choose **Create flow**.

To create an Amazon AppFlow dataset

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Add dataset**.
4. Choose the cloud application that you chose for your flow.
5. Follow the instructions to create the datasource.

With Amazon AppFlow, you choose fields from the flow's data to be measures and dimensions. The detector monitors the measures for anomalies within values of the dimensions. For example, if you have a measure named `availability` and a dimension named `region`, the detector finds anomalies of `availability` in each `region`.

The detector reads new data from Amazon AppFlow periodically, by accessing the flow. If it detects any anomalies in the metrics for the interval, it records an anomaly and sends anomaly alerts, if configured.

When you activate the detector, it uses data from several intervals to learn, before attempting to find anomalies. For a five minute interval, the training process takes approximately one day. Training time varies depending on the detector's interval.

> **Note**
>
> When you add an Amazon AppFlow dataset to your detector, the Lookout for Metrics console creates a service role with permission to read from the flow.

For more information about Amazon AppFlow, see Getting started with Amazon AppFlow in the Amazon AppFlow User Guide.

Using Amazon CloudWatch with Lookout for Metrics

You can use Amazon CloudWatch metrics as a datasource for an Amazon Lookout for Metrics detector. Most AWS services send metrics to CloudWatch automatically when you use them. You can create a dataset from these metrics, or from metrics that you send to CloudWatch. You can send metrics to CloudWatch from your application code, or from software such as StatsD or collectd.

Topics

Adding a CloudWatch datasource

In CloudWatch, a metric can have a name and value, and optionally can also have a dimension name and dimension value. For example, Amazon EC2 has a metric named `CPUUtilization` with a value that is a number between 0 and 100, and a dimension named `InstanceId` that has the unique ID of an instance.

To create a CloudWatch dataset

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Add dataset**.
4. For **Datasource**, choose **CloudWatch**.
5. Follow the instructions to create the datasource.

To configure metrics in Lookout for Metrics, you choose a CloudWatch namespace and dimension first, and then choose one or more CloudWatch metrics to be measures for the dataset. For CloudWatch metrics that apply to all resources in an AWS Region, or otherwise don't have a dimension in CloudWatch, you set **Dimensions** to **None**.

For example, in AWS Lambda you can monitor concurrency by function, by resource (function version or alias), or across all functions in a Region. If you choose `ConcurrentExecutions` as

a measure and `Function Name` as a dimension, then the detector monitors *concurrency for function-a* and *concurrency for function-b* as two Lookout for Metrics metrics.

The detector reads new data from CloudWatch periodically, by reading the values of metrics that occur in each interval. It aggregates the values of each metric for the interval and looks for anomalies. It records anomalies and sends anomaly alerts, if configured.

When you activate the detector, it uses data from several intervals to learn, before attempting to find anomalies. For a five minute interval, the training process takes approximately one day. Training time varies depending on the detector's interval.

When you add a CloudWatch dataset to your detector, the Lookout for Metrics console creates a service role with permission to read metrics.

Sections

Training a detector with a CloudWatch datasource

When using a CloudWatch datasource, the following number of days are used to train a detector:

- **5 minute interval** - 10.42 days
- **10 minute interval** - 20.48 days
- **1 hour interval** - 125 days
- **1 day interval** - 455 days

For example, if a datasource has a 5 minute interval, data from the previous 10.42 days will be used to train a detector. If the datasource contains fewer than 10.42 days of data, a detector will be trained using the available data.

The datasource must meet the minumum data requirements in order to train a detector.

Running a backtest

In backtest mode, a detector uses historical data to learn and find anomalies. You provide recent data for a large number of intervals at a single path. In backtest mode, Lookout for Metrics splits

historical data into two subsets. 70 percent of the data is used to train the detector. The detector then analyzes the other 30 percent to identify anomalies. You can use test mode to validate the detector's results and verify its accuracy.

When using a CloudWatch datasource, the following number of days are used to train a detector:

- **5 minute interval** - 10.42 days
- **10 minute interval** - 20.48 days
- **1 hour interval** - 125 days
- **1 day interval** - 455 days

For example, if a datasource has a 5 minute interval, data from the previous 10.42 days will be used to train a detector. If the datasource contains fewer than 10.42 days of data, a detector will be trained using the available data. The Cloudwatch metrics must be present in at least 285 intervals worth of data, but not more than 3000 intervals worth of data. This gives the detector at least 200 intervals of data to learn with. The detector always uses older data for learning and newer data for testing.

Using AWS Lambda with Lookout for Metrics

You can use AWS Lambda as a channel for anomaly alerts from an Amazon Lookout for Metrics detector. With a Lambda function, you can process anomaly alerts in your preferred programming language, and use the AWS SDK to interact with other AWS services.

Lambda is a serverless way for you to run code in AWS. Your code only runs and only incurs charges when it is invoked. If you don't already have a Lambda function, see Create a Lambda function with the console in the Lambda Developer Guide to get started.

To create a Lambda alert

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Add alert**.
4. Choose **AWS Lambda**.
5. Choose a function.

> **Note**
>
> When you add a Lambda alert to your detector, the Lookout for Metrics console creates a service role with permission to invoke the function.

When your detector finds an anomaly with a severity score that meets or exceeds the alert's threshold, it invokes your Lambda function with an event that contains details about the anomaly. The Lambda runtime converts this document into an object and passes it to your function's *handler method*. You can use this object to perform additional processing, to record details about the anomaly in a database or storage, or to call another service.

For sample code in all programming languages supported by Lambda, see Lambda sample applications in the AWS Lambda Developer Guide.

Using Amazon RDS with Lookout for Metrics

You can use Amazon Relational Database Service (Amazon RDS) as a datasource for an Amazon Lookout for Metrics detector. With Amazon RDS, you can choose columns to monitor (*measures*) and columns that segment measure values (dimensions). The detector monitors the values in these columns to find anomalies in your data.

Important

Lookout for Metrics can only connect to databases in a subset of Availability Zones in some Regions. The following Availability Zones are supported.

- **US East (N. Virginia)** – `use1-az1`,`use1-az4`, `use1-az6`
- **US West (Oregon)** – `usw2-az1`, `usw2-az2`, `usw2-az3`
- **Asia Pacific (Tokyo)** – `apne1-az1`, `apne1-az2`, `apne1-az4`
- **Other Regions** – All Availability Zones.

Availability Zone names such as `us-west-2a` are aliases for zone IDs that vary by account. To see which names map to which IDs in your account, visit the EC2 dashboard in the AWS Management Console.

The following database engines are supported:

- **Amazon Aurora**
- **MySQL**
- **PostgreSQL**
- **MariaDB**
- **Microsoft SQL Server**

To use an Amazon RDS database with Lookout for Metrics, the table must have a timestamp column that is indexed for queries. This allows Lookout for Metrics to get records for an interval without scanning the entire table. You also need an AWS Secrets Manager secret for

the detector. The secret must have the database password and have a name that starts with `AmazonLookoutMetrics-`.

Before you configure the dataset, you need to know the following information.

- **DB identifier** – The unique identifier of the DB instance or cluster. For example, `mysql-dbi` or `ld1xmplvzghgn47`.
- **Database name** – The software-level database name. For example, `mydb`.
- **Table name** – The name of the table. For example, `events`.
- **Column names** – The names of columns that contain timestamps, measures, and dimensions.
- **Subnets** – The virtual private cloud (VPC) subnets where the detector creates network interfaces to connect to the database. For example, `subnet-0752xmpl92bf2e4b7`.
- **Security group** – A VPC security group that allows traffic to the database. For example, `sg-0f92xmplfbad0bc95`.
- **Secret name** – The name of an AWS Secrets Manager secret that the detector uses to retrieve the database password. For example, `AmazonLookoutMetrics-mysqldbi`.
- **Secret ID** – The ID of the secret, for generating a service role that can access it. For example, `AmazonLookoutMetrics-mysqldbi-Nxmplo`.

To create an Amazon RDS dataset

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Add dataset**.
4. Choose one of the available database engines:

    - **Amazon Aurora**
    - **MySQL**
    - **PostgreSQL**
    - **MariaDB**
    - **Microsoft SQL Server**
5. Follow the instructions to create the datasource.

To configure metrics in Lookout for Metrics, you choose columns to be measures and dimensions. Each measure is a column with a numerical value that you want to monitor for anomalies. Each dimension is a column with a string value that segments the measure(s). A metric in Lookout for Metrics is a combination of a measure value and a dimension value, aggregated within an interval. For example, *average availability in Colorado*, or *maximum temperature in furnace 17*.

The detector reads new data from Amazon RDS periodically, by querying records with timestamps in the most recently completed interval. If it detects any anomalies in the metrics for the interval, it records an anomaly and sends anomaly alerts, if configured.

When you activate the detector, it uses data from several intervals to learn, before attempting to find anomalies. For a five minute interval, the training process takes approximately one day. Training time varies depending on the detector's interval.

> **Note**
>
> When you add an Amazon RDS dataset to your detector, the Lookout for Metrics console creates a service role with permission to use the database secret and monitor Amazon RDS resources. Lookout for Metrics also creates up to two elastic network interfaces, which allow it to connect to your VPC to access your database. When you delete the detector, Lookout for Metrics deletes the network interfaces.

For more information about Amazon RDS, see Getting started with Amazon RDS in the Amazon RDS User Guide.

Sections

Sample IAM policies

The GitHub repository for this guide provides sample IAM policies that you can use as reference for developing service roles. You can use a single role that grants permission for both importing data and sending alerts by combining the applicable policies.

Example datasource-rds.json – Monitor and access an Amazon RDS DB instance

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "rds:DescribeDBInstances"
            ],
            "Resource": [
                "arn:aws:rds:${Region}:${Account}:db:${DatabaseId}"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "rds:DescribeDBSubnetGroups"
            ],
            "Resource": "arn:aws:rds:${Region}:${Account}:subgrp:*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:${Region}:${Account}:secret:${SecretId}"
            ],
            "Effect": "Allow",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "secretsmanager:VersionStage": "AWSCURRENT"
                }
            }
        },
        ...
```

The second sample policy shows how to grant the detector permission to connect to a database across accounts. The account with the database instance (Account B) must be in the same organization and share its subnet with the account that contains the detector (`AccountA`).

Example datasource-rds-xaccount.json – Cross-account access

```
        ...
        {
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:${Region}:${AccountA}:network-interface/*",
                "arn:aws:ec2:${Region}:${AccountA}:security-group/*",
                "arn:aws:ec2:${Region}:${AccountB}:subnet/${SubnetId}"
            ],
            "Effect": "Allow"
        },
        ...
```

For more information, see Working with shared VPCs in the *Amazon VPC User Guide*.

Sample AWS CloudFormation templates

The GitHub repository for this guide provides sample AWS CloudFormation templates that you can use to automate the creation of service roles. The templates use parameters and naming patterns to apply least-privilege permissions where possible.

The sample template creates a service role that gives Lookout for Metrics permission to use secrets prefixed with `AmazonLookout` to get database credentials, monitor Amazon RDS DB instances, and create network interfaces.

Example servicerole-rds.yml – Amazon S3 and Lambda permissions

```
Resources:
  serviceRole:
    Type: AWS::IAM::Role
    Properties:
      Policies:
        - PolicyName: rds-access
          PolicyDocument:
            Version: 2012-10-17
            Statement:
            - Action:
              - rds:DescribeDBInstances
              Resource:
```

```
              - !Sub arn:${AWS::Partition}:rds:${AWS::Region}:${AWS::AccountId}:db:
${databaseId}
              Effect: Allow
            - Action:
              - rds:DescribeDBSubnetGroups
              Resource: !Sub arn:${AWS::Partition}:rds:${AWS::Region}:
${AWS::AccountId}:subgrp:*
              Effect: Allow
            - Action:
              - secretsmanager:GetSecretValue
              Resource:
              - !Sub arn:${AWS::Partition}:secretsmanager:${AWS::Region}:
${AWS::AccountId}:secret:${secretId}
              Effect: Allow
              Condition:
                ForAllValues:StringEquals:
                  secretsmanager:VersionStage: AWSCURRENT
```

For more information, see Sample templates in the GitHub repo.

Using Amazon Redshift with Lookout for Metrics

You can use Amazon Redshift as a datasource for an Amazon Lookout for Metrics detector. With Amazon Redshift, you can choose columns to monitor (*measures*) and columns that segment measure values (dimensions). The detector monitors the values in these columns to find anomalies in your data.

Important

Lookout for Metrics can only connect to databases in a subset of Availability Zones in some Regions. The following Availability Zones are supported.

- **US East (N. Virginia)** – `use1-az1`,`use1-az4`, `use1-az6`
- **US West (Oregon)** – `usw2-az1`, `usw2-az2`, `usw2-az3`
- **Asia Pacific (Tokyo)** – `apne1-az1`, `apne1-az2`, `apne1-az4`
- **Other Regions** – All Availability Zones.

Availability Zone names such as `us-west-2a` are aliases for zone IDs that vary by account. To see which names map to which IDs in your account, visit the EC2 dashboard in the AWS Management Console.

To use an Amazon Redshift data warehouse with Lookout for Metrics, the table must have a timestamp column that is defined as the *sort key*. Amazon Redshift uses the sort key to store data on disk and construct plans that exploit the way that the data is stored. For more information, see Working with sort keys in the Amazon Redshift Database Developer Guide.

You also need an AWS Secrets Manager secret for the detector. The secret must have the database password and have a name that starts with `AmazonLookoutMetrics-`.

The detector imports data at the end of each interval. You configure an **offset** to allow time after an interval ends for all data to be written. For example, if you choose an offset of 30 seconds, the detector waits 30 seconds after the end of each interval before reading data for that interval.

Before you configure the dataset, you need to know the following information.

- **DB identifier** – The unique identifier of the DB instance or cluster. For example, `mysql-dbi` or `ld1xmplvzghgn47`.
- **Database name** – The software-level database name. For example, `mydb`.
- **Table name** – The name of the table. For example, `events`.
- **Column names** – The names of columns that contain timestamps, measures, and dimensions.
- **Subnets** – The virtual private cloud (VPC) subnets where the detector creates network interfaces to connect to the database. For example, `subnet-0752xmpl92bf2e4b7`.
- **Security group** – A VPC security group that allows traffic to the database. For example, `sg-0f92xmplfbad0bc95`.
- **Secret name** – The name of an AWS Secrets Manager secret that the detector uses to retrieve the database password. For example, `AmazonLookoutMetrics-mysqldbi`.
- **Secret ID** – The ID of the secret, for generating a service role that can access it. For example, `AmazonLookoutMetrics-mysqldbi-Nxmplo`.

To create an Amazon Redshift dataset

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Add dataset**.
4. Choose **Amazon Redshift**.
5. Follow the instructions to create the datasource.

To configure metrics in Lookout for Metrics, you choose columns to be measures and dimensions. Each measure is a column with a numerical value that you want to monitor for anomalies. Each dimension is a column with a string value that segments the measure(s). A metric in Lookout for Metrics is a combination of a measure value and a dimension value, aggregated within an interval. For example, *average availability in Colorado*, or *maximum temperature in furnace 17*.

The detector reads new data from Amazon Redshift periodically, by querying records with timestamps in the most recently completed interval. If it detects any anomalies in the metrics for the interval, it records an anomaly and sends anomaly alerts, if configured.

When you activate the detector, it uses data from several intervals to learn, before attempting to find anomalies. For a five minute interval, the training process takes approximately one day. Training time varies depending on the detector's interval.

> **Note**
>
> When you add an Amazon Redshift dataset to your detector, the Lookout for Metrics console creates a service role with permission to use the database secret and monitor Amazon Redshift resources. Lookout for Metrics also creates up to two elastic network interfaces, which allow it to connect to your VPC to access your database. When you delete the detector, Lookout for Metrics deletes the network interfaces.

For more information about Amazon Redshift, see Getting started with Amazon Redshift in the Amazon Redshift Getting Started Guide.

Sections

Sample IAM policies

The GitHub repository for this guide provides sample IAM policies that you can use as reference for developing service roles. You can use a single role that grants permission for both importing data and sending alerts by combining the applicable policies.

Example datasource-redshift.json – Monitor and access an Amazon Redshift cluster

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "redshift:DescribeClusters",
                "redshift:DescribeClusterSubnetGroups"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
```

```
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:${Region}:${Account}:secret:${SecretId}"
            ],
            "Effect": "Allow",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "secretsmanager:VersionStage": "AWSCURRENT"
                }
            }
        },
        ...
```

The second sample policy shows how to grant the detector permission to connect to a cluster across accounts. The account with the cluster (Account B) must be in the same organization and share its subnet with the account that contains the detector (`AccountA`).

Example datasource-redshift-xaccount.json – Cross-account access

```
        ...
        {
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:${Region}:${AccountA}:network-interface/*",
                "arn:aws:ec2:${Region}:${AccountA}:security-group/*",
                "arn:aws:ec2:${Region}:${AccountB}:subnet/${SubnetId}"
            ],
            "Effect": "Allow"
        },
        ...
```

For more information, see Working with shared VPCs in the *Amazon VPC User Guide*.

Using Amazon SNS with Lookout for Metrics

You can use Amazon SNS as a channel for anomaly alerts from a Amazon Lookout for Metrics detector. With Amazon SNS, you can create a topic that has subscribers that use email, SMS, and mobile applications, or applications that use HTTP endpoints, Amazon Simple Queue Service (Amazon SQS) queues, or Lambda functions.

If you don't have an Amazon SNS topic, see Getting started with Amazon SNS in the Amazon Simple Notification Service Developer Guide.

With a Lambda function subscriber, you can send anomaly alerts to a webhook or HTTP endpoint that you create in a third party service. Lookout for Metrics supports the following application channels with Amazon SNS:

- **PagerDuty**
- **Slack**
- **Datadog**

To create an Amazon SNS alert

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Add alert**.
4. Choose Amazon SNS.

> **Note**
>
> When you add an Amazon SNS alert to your detector, the Lookout for Metrics console creates a service role with permission to send notifications to an Amazon SNS topic.

When your detector finds an anomaly with a severity score that meets or exceeds the alert's threshold, it sends a notification to your Amazon SNS topic with an message that contains details about the anomaly.

Sections

Using Webhooks

With an Amazon SNS topic, you can create subscribers that use a webhook in a cloud application. A webhook is an HTTP endpoint that you call to use some functionality of an application. The application can be one that you run in your account, or a third party service.

Amazon SNS has two types of subscriber that you can use with webhooks: Lambda functions and HTTP endpoints. With an HTTP endpoint, Amazon SNS posts a notification document to the endpoint. If the webhook supports Amazon SNS notifications as an input, or doesn't use an input, use an HTTP endpoint subscriber.

If the webhook requires data to be formatted in a specific way, you can use a Lambda function to process anomaly alerts and send a properly formatted document to the webhook. For example, Slack webhooks require a document with a `text` field that contains text to post to the webhook's Slack channel. You can use a Lambda function to get relevant text from the Amazon SNS notification, format it, and send it to the webhook.

Using Datadog

Datadog provides an HTTP endpoint that you can subscribe to an Amazon SNS topic. To send anomaly alerts to Datadog, enable AWS integration in your Datadog account and then create a subscription on your Amazon SNS topic with the following settings:

- **Protocol** – HTTPS
- **Endpoint** –

  `https://app.datadoghq.com/intake/webhook/sns?api_key=`*`YOUR_API_KEY`* (US)

  Or

  `https://app.datadoghq.eu/intake/webhook/sns?api_key=`*`YOUR_API_KEY`* (EU)

For more information, see Amazon SNS at docs.datadoghq.com

Using Slack

With Slack, you can create a webhook that posts a message to a channel. To format the message into a document that Slack can process, you use a Lambda function subscriber with code that reformats the message into a document with a `text` key. You can then send the document to the Slack webhook with an HTTP client.

For instructions on creating the Lambda function, see the following article: How do I use webhooks to publish Amazon SNS messages to Amazon Chime, Slack, or Microsoft Teams?.

For details on creating a webhook in Slack, see Sending messages using Incoming Webhooks at api.slack.com.

Using PagerDuty

With PagerDuty, you can create a webhook that triggers an incident with a service integration and the Events API.

To send anomaly alerts to PagerDuty, create a subscription on your Amazon SNS topic with the following settings:

- **Protocol** – HTTPS
- **Endpoint** – `https://events.pagerduty.com/integration/`*`INTEGRATION_KEY`*`/enqueue`

For details on creating an integration in PagerDuty, see Services and Integrations at support.pagerduty.com.

Using Amazon S3 with Lookout for Metrics

You can use Amazon Simple Storage Service (Amazon S3) as a datasource for an Amazon Lookout for Metrics detector. With Amazon S3, you provide data in a bucket organized into separate folders for each interval. You configure the detector with a pattern that indicates where to find data for each interval, and write data to the path of the current interval as it is generated.

For example, if your detector's interval is 1 day, then you write data to a separate path for each day, in a predictable pattern. For example, the data for January 2nd, 2021 can go in a folder named `2021/01/02`, `2021-01/02`, `20210102`, etc.

Your data can consist of rows of delimited records (CSV format) or JSON objects (JSON lines format). Each line in a file has one record, which has fields for measures, dimensions, and a timestamp. For details on organizing and formatting your data, see Managing a dataset in Amazon S3.

The detector imports data at the end of each interval. You configure an **offset** to allow time after an interval ends for all data to be written. For example, if you choose an offset of 30 seconds, the detector waits 30 seconds after the end of each interval before reading data for that interval.

To create an Amazon S3 dataset

1. Open the Lookout for Metrics console Detectors page.
2. Choose a detector.
3. Choose **Add dataset**.
4. Choose **Amazon S3**.
5. Follow the instructions to create the datasource.

The detector reads new data from Amazon S3 periodically, by getting objects from the folder for the most recent completed interval. If it detects any anomalies in the metrics for the interval, it records an anomaly and sends anomaly alerts, if configured.

When you activate the detector, it uses data from several intervals to learn, before attempting to find anomalies. If no historical data is available, the training process takes approximately one day for a five-minute interval. Training time varies depending on the detector's interval.

For more information about Amazon S3, see Getting started with Amazon Simple Storage Service in the Amazon Simple Storage Service User Guide.

Sections

Configuring permissions

When you add an Amazon S3 dataset to your detector, the Lookout for Metrics console creates a service role with permission to read data from the bucket, and permission to use AWS Key Management Service (AWS KMS) to encrypt and decrypt data.

The console creates a role for the dataset, and a separate role for each alert that you configure. You can create a single role for the detector that gives it all of the permissions that it needs. To use a custom role, create an IAM role that Lookout for Metrics has permission to assume, and add permission to use Amazon S3 and AWS KMS.

Structuring data

To determine the correct pattern for your data, you can enter the URI of any example data file in the bucket. The console analyzes the path and shows one or more patterns that matches. Choose the pattern that matches your folder structure.

For details on organizing your data, see Managing a dataset in Amazon S3.

Timestamps

Entries in your data must have a field with a date or timestamp that indicates which interval they occur in. When you configure your dataset, you specify the format of the timestamp as a pattern with the following keys.

- `yyyy` – Year
- `MM` – Month
- `DD` – Day
- `HH` – Hour (24-hour time)

- hh – Hour (12-hour time)
- a – AM/PM indicator (with 12-hour hh key)
- `mm` – Minutes
- `ss` – Seconds

For daily intervals, you can use a date that indicates only the day. For more granular intervals, the timestamp must be specific enough to distinguish between intervals.

- `yyyy-MM-DD` – `2021-02-28` (daily only)
- `yyyy-MM-DD HH:mm:ss` – `2021-02-28 17:45:32` (any interval)
- `yyyy-MM-dd hh:mm:ss a` – `2021-02-28 05:45:32 pm` (any interval)

You can also format timestamps in the Unix time format. For example, the timestamp for Fri Jan 01 2021 08:00:00 GMT+0000 is `1609488000` in Unix time. The Unix time format is only available to Amazon S3 connectors.

To use the Unix time format, choose **Use Unix time** when adding a dataset. With the SDK, set the value of `ColumnFormat` to `UNIX` within the TimestampColumn data type. If the Unix timestamp is larger than 94694400000, the timestamp is expressed in milliseconds (`UNIX_MS`). Otherwise, the Unix timestamp is expressed in seconds (`UNIX_SECONDS`).

Running a backtest

In backtest mode, a detector uses historical data to learn and find anomalies. You provide recent data for a large number of intervals at a single path. In backtest mode, Lookout for Metrics splits historical data into two subsets. 70 percent of the data is used to train the detector. The detector then analyzes the other 30 percent to identify anomalies. You can use test mode to validate the detector's results and verify its accuracy.

For backtest mode, you can provide between 285 and 3000 intervals worth of data. The data can be in one file or multiple files in the same folder. This gives the detector at least 200 intervals of data to learn with. The detector always uses older data for learning and newer data for testing.

Working with Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) is a service for creating private networks, security rules, and related resources in AWS. You create a virtual private cloud (VPC) to allow resources like EC2 instances and databases to communicate securely over a local network connection.

When you use Amazon Relational Database Service (Amazon RDS) or Amazon Redshift as a datasource, Lookout for Metrics uses Amazon VPC to connect to a database in your account. The detector creates a network interface in your VPC subnet and uses it to connect to the database and run queries at the end of each interval.

If you develop applications that run in a VPC without internet access, you can also create a VPC endpoint. A VPC endpoint lets applications running in a private subnet connect to an AWS service without an internet connection.

Sections

Connecting to a database

When you configure a detector to monitor a database, you choose a VPC subnet and security group. The detector uses these to create a network interface that it can use to connect to the database. The network interface is named `Amazon Lookout for Metrics Network Interface`.

The detector gets permission to manage network interfaces in your VPC from the a service role. When you configure a datasource in the Lookout for Metrics console, the console can create a role for you. To consolidate datasource and alert roles for a detector, or connect to a database in another account, you can create a custom role.

For more information, see the following topics:

- Using Amazon RDS with Lookout for Metrics
- Using Amazon Redshift with Lookout for Metrics

Creating a VPC endpoint

To establish a private connection between your VPC and Amazon Lookout for Metrics, create a *VPC endpoint*. A VPC endpoint is not required to monitor a database for anomalies. You only need to create a VPC endpoint if you run an application that uses the AWS SDK in a private subnet. When the AWS SDK attempts to connect to Lookout for Metrics, the traffic is routed through the VPC endpoint.

Create a VPC endpoint for Lookout for Metrics using the following settings:

- **Service name** – **`com.amazonaws.us-east-2.lookoutmetrics`**
- **Type** – **Interface**

A VPC endpoint uses the service's DNS name to get traffic from AWS SDK clients without any additional configuration. For more information about using VPC endpoints, see Interface VPC endpoints in the *Amazon VPC User Guide*.

Sample AWS CloudFormation templates

The GitHub repository for this guide provides AWS CloudFormation templates that you can use to create resources for use with Lookout for Metrics. The `vpc-network.yml` template creates a VPC with two private subnets, a public subnet, and a VPC endpoint.

You can use the private subnets in the VPC to host a database that is isolated from the internet. Resources in the public subnet can communicate with the database, but the database can't be accessed from the internet.

Example vpc-network.yml – Private subnets

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 172.31.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
      Tags:
        - Key: Name
```

```
        Value: !Ref AWS::StackName
  privateSubnetA:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref vpc
      AvailabilityZone:
        Fn::Select:
         - 0
         - Fn::GetAZs: ""
      CidrBlock: 172.31.3.0/24
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: !Sub  ${AWS::StackName}-subnet-a
  ...
```

The `vpc-network` template shows how to create a VPC endpoint. The VPC endpoint is not required to connect the detector to a database, but can be used to communicate securely with Lookout for Metrics from compute resources in a private subnet.

Example vpc-network.yml – VPC endpoint

```
  lookoutfmEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: !Sub com.amazonaws.${AWS::Region}.lookoutmetrics
      VpcId: !Ref vpc
      VpcEndpointType: Interface
      SecurityGroupIds:
      - !GetAtt vpc.DefaultSecurityGroup
      PrivateDnsEnabled: true
      SubnetIds:
      - !Ref privateSubnetA
      - !Ref privateSubnetB
      PolicyDocument:
        Version: 2012-10-17
        Statement:
        - Effect: Allow
          Principal: "*"
          Action:
            - "lookoutmetrics:*"
          Resource:
            - "*"
```

The `PolicyDocument` is a resource-based permissions policy that defines the API calls that can be made with the endpoint. You can modify the policy to restrict the actions and resources that can be accessed through the endpoint.

For more information, see Controlling access to services with VPC endpoints in the *Amazon VPC User Guide*.

Monitoring Lookout for Metrics

Monitoring is an important part of ensuring the reliability, availability, and performance of Amazon Lookout for Metrics and your other AWS solutions. AWS provides the following monitoring tools to watch Lookout for Metrics, report when something is wrong, and take automatic actions when appropriate:

- *Amazon CloudWatch* monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the Amazon CloudWatch User Guide.
- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and optionally delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Topics

Viewing Amazon Lookout for Metrics API activity in CloudTrail

Amazon Lookout for Metrics is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Lookout for Metrics. CloudTrail captures all API calls for Lookout for Metrics as events. Captured calls include calls from the Lookout for Metrics console and code calls to the Lookout for Metrics API operations.

Using the information collected by CloudTrail, you can determine the request that was made to Lookout for Metrics, the IP address from which the request was made, who made the request, when it was made, and additional details.

All Lookout for Metrics calls are logged by CloudTrail. Log entries contain information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity element.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Topics

Storing Lookout for Metrics information in CloudTrail

AWS CloudTrail is activated on your AWS account when you create it. When activity occurs in Lookout for Metrics, it is automatically recorded in a CloudTrail event. You can view, search, and download recent events in the **Event history** in the CloudTrail console. For more information, see Viewing events with CloudTrail Event history.

For an ongoing record of events in your AWS account, including events for Lookout for Metrics, create a trail. A *trail* enables CloudTrail to send log files to an Amazon S3 bucket. When you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition. It sends the log files to the Amazon S3 bucket that you specify. For more information, see Overview for creating a trail.

You can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple Regions and Receiving CloudTrail log files from multiple accounts

Example: Lookout for Metrics log file entry

AWS CloudTrail log files contain one or more log entries, one entry for every event. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example is a CloudTrail log entry for a `DescribeAnomalyDetector` call. Specific information about the call appears in the `eventName` and `requestParameters` fields. The remaining fields record details about the caller and tracking information such as the request ID, which you can use to find information about the request in places like logs and AWS X-Ray traces.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/fred",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "fred",
        "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-01-09T00:14:34Z"
            }
        }
    },
    "eventTime": "2021-01-09T00:18:12Z",
```

```
    "eventSource": "lookoutmetrics.amazonaws.com",
    "eventName": "DescribeAnomalyDetector",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "205.256.256.182",
    "userAgent": "aws-sdk-java/1.11.930 Linux/4.9.230-0.1.ac.223.84.332.metal1.x86_64
 OpenJDK_64-Bit_Server_VM/25.275-b01 java/1.8.0_275 vendor/Oracle_Corporation",
    "requestParameters": {
        "AnomalyDetectorArn": "arn:aws:lookoutmetrics:us-
east-2:123456789012:AnomalyDetector:my-detector-5m"
    },
    "responseElements": null,
    "requestID": "f587ee3c-xmpl-406b-b573-66100bb14b61",
    "eventID": "f2f879f8-xmpl-4475-9c0c-4291a389e14a",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "123456789012"
}
```

Monitoring Lookout for Metrics with Amazon CloudWatch

When an anomaly detector processes data or sends an alert, Amazon Lookout for Metrics sends metrics to Amazon CloudWatch. You can build graphs and dashboards with these metrics in the CloudWatch console to monitor detector activity, and set alarms that notify you when there are errors in alert delivery or anomaly detection. To filter and sort metrics, use the `AlertArn` and `AnomalyDetectorArn` Amazon CloudWatch dimensions.

To view metrics in the CloudWatch console

1. Open the Amazon CloudWatch console Metrics page (`AWS/LookoutMetrics` namespace).
2. Choose a dimension schema.
   - **AlertArn** – To view metrics for an alert channel.
   - **AnomalyDetectorArn** – To view processing metrics for a detector.
3. Choose metrics to add them to the graph.
4. To choose a different statistic and customize the graph, use the options on the **Graphed metrics** tab. By default, graphs use the `Average` statistic for all metrics.

> **Pricing**
>
> CloudWatch has an Always Free tier. Beyond the free tier threshold, CloudWatch charges for metrics, dashboards, alarms, logs, and insights. For details, see CloudWatch pricing.

For more information about CloudWatch, see the *Amazon CloudWatch User Guide*.

Sections

Using processing metrics

When a detector processes data at the end of each interval, it sends metrics to Amazon CloudWatch. You can use these metrics to monitor detector activity and trigger an alarm if a detector fails to process data.

There are three processing metrics:

- `ExecutionsStarted` – The number of times the detector looked for anomalies. For an active detector, this metric is sent once per interval.
- `ExecutionsSucceeded` – The number of processing attempts that succeeded.
- `ExecutionsFailed` – The number of processing attempts that failed.

View the processing metrics with the `Sum` statistic.

Using alert metrics

When a detector finds an anomaly, it sends anomaly alerts to alert channels that you configure. Each time it sends an anomaly alert, the detector sends a metric to indicate whether the alarm was sent successfully.

There are two alert metrics:

- `Delivered` – An anomaly alert was sent successfully.
- `Undelivered` – An anomaly alert could not be delivered.

View the alert metrics with the `Sum` statistic.

Configuring alarms

To get notifications when a metric exceeds a threshold, create an alarm. For example, you can create an alarm that sends a notification when the sum of the `ExecutionsFailed` metric exceeds 1 for an hour.

To create an alarm

1. Open the Amazon CloudWatch console Alarms page.
2. Choose **Create alarm**.

3. Choose **Select metric** and locate a metric for your detector, such as `ExecutionsFailed` for `my-detector`.
4. Follow the instructions to configure a condition, action, and name for the alarm.

For detailed instructions, see Create a CloudWatch alarm in the *Amazon CloudWatch User Guide*.

Troubleshooting Lookout for Metrics

The following topics provide troubleshooting information for errors and issues that you might encounter when using the Amazon Lookout for Metrics console or API. If you find an issue that is not listed here, use the **Provide feedback** button on this page to report it.

Data validation

Amazon Lookout for Metrics validates your dataset when you configure metric during setup. For an Amazon Simple Storage Service (Amazon S3) datasource, it checks a small sample of files to ensure that the required fields exist and timestamps are in the right format.

Error: *ValidationException : One of specified bucket in historicalDataPathList is not in same region as this AWS service API.*

An anomaly detector can access only data that is stored in the same AWS Region as the detector. If you can't create a detector in the same Region as your data, you can use replication to copy data into the detector's Region. Use the offset setting on the dataset to allow time for replication to complete.

Error: *ValidationException : Timestamp Column values inside file provided in S3 path do not match provided timestamp format.*

This error can occur if you configure the wrong timestamp format, or if the subset of entries used for validation had errors or missing values. Check the format or try again. Note that hh is a 12-hour format and requires a corresponding AM/PM marker. For example, `yyyy-MM-dd hh:mm:ss a`. For 24-hour time, use HH.

ConflictException:*Updating or deleting a resource can cause an inconsistent state.*

Amazon Lookout for Metrics requires that your alerts have a unique name. If you delete or update an alert, you might get this error if you try to reuse the name of the alert that was deleted or updates.

Importing data

At the end of each interval in continuous mode, or at the start of a backtest, a detector imports data from the datasource into the dataset. In continuous mode, the detector skips the failed

interval, records errors in the detector log, and tries again at the end of each interval. In backtest mode, the test fails. In this case, delete the failed detector, resolve the issue, and try again with a new detector.

Error: *Insufficient amount of data was found at the source location. Check service limits and requirements.*

The detector couldn't find usable data for the interval or test period. This error can occur if you choose a field with non-numerical columns as a measure. Measures must have numerical data that can be aggregated (added together or averaged) for each interval.

Error: *No datapoint was found at the source location for time interval from 2021-02-27T01:30:00.000 to 2021-02-27T01:20:00.000.*

This issue can occur if the dataset is configured with the wrong timezone, or if data was written after the interval ended. If data isn't immediately available in the datasource after an interval ends, use the offset setting to make the detector wait before looking for data.

Security in Amazon Lookout for Metrics

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to Amazon Lookout for Metrics, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Lookout for Metrics. The following topics show you how to configure Lookout for Metrics to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Lookout for Metrics resources.

Topics

Data protection in Amazon Lookout for Metrics

The AWS shared responsibility model applies to data protection in Amazon Lookout for Metrics. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on

this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Lookout for Metrics or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Sections

Encryption in transit

Lookout for Metrics API endpoints support secure connections only over HTTPS. When you manage Lookout for Metrics resources with the AWS Management Console, AWS SDK, or the Lookout for Metrics API, all communication is encrypted with Transport Layer Security (TLS).

For a complete list of API endpoints, see AWS Regions and endpoints in the *AWS General Reference.*

Encryption at rest

Lookout for Metrics always encrypts a detector's dataset at rest. Additionally, you can configure Lookout for Metrics to use an encryption key that you create and manage in AWS Key Management Service. These are referred to as *customer managed* AWS KMS key or customer managed keys. If you don't configure a customer managed key, Lookout for Metrics uses a key that the service manages.

Additionally, Lookout for Metrics encrypts your resources' configuration at rest with a key that it manages. This includes the names of fields or columns in your datasource that you use as measures and dimensions. You can't configure Lookout for Metrics to use a CMK for configuration data.

Identity and Access Management for Amazon Lookout for Metrics

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Lookout for Metrics resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How Amazon Lookout for Metrics works with IAM
- Identity-based policy examples for Amazon Lookout for Metrics
- AWS managed policies for Amazon Lookout for Metrics

- Cross-service confused deputy prevention
- Troubleshooting Amazon Lookout for Metrics identity and access

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Lookout for Metrics.

Service user – If you use the Lookout for Metrics service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Lookout for Metrics features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Lookout for Metrics, see Troubleshooting Amazon Lookout for Metrics identity and access.

Service administrator – If you're in charge of Lookout for Metrics resources at your company, you probably have full access to Lookout for Metrics. It's your job to determine which Lookout for Metrics features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Lookout for Metrics, see How Amazon Lookout for Metrics works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Lookout for Metrics. To view example Lookout for Metrics identity-based policies that you can use in IAM, see Identity-based policy examples for Amazon Lookout for Metrics.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and AWS Multi-factor authentication in IAM in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For

information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. To temporarily assume an IAM role in the AWS Management Console, you can switch from a user to an IAM role (console). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Methods to assume a role in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Create a role for a third-party identity provider (federation) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.

- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Use an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that

support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see Service control policies in the *AWS Organizations User Guide*.

- **Resource control policies (RCPs)** – RCPs are JSON policies that you can use to set the maximum available permissions for resources in your accounts without updating the IAM policies attached to each resource that you own. The RCP limits permissions for resources in member accounts and can impact the effective permissions for identities, including the AWS account root user, regardless of whether they belong to your organization. For more information about Organizations and RCPs, including a list of AWS services that support RCPs, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How Amazon Lookout for Metrics works with IAM

Before you use IAM to manage access to Lookout for Metrics, learn what IAM features are available to use with Lookout for Metrics.

IAM features you can use with Amazon Lookout for Metrics

IAM feature	Lookout for Metrics support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes

IAM feature	Lookout for Metrics support
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Principal permissions	Yes
Service roles	Yes
Service-linked roles	No

To get a high-level view of how Lookout for Metrics and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for Lookout for Metrics

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for Lookout for Metrics

To view examples of Lookout for Metrics identity-based policies, see Identity-based policy examples for Amazon Lookout for Metrics.

Resource-based policies within Lookout for Metrics

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for Lookout for Metrics

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Lookout for Metrics actions, see Actions defined by Amazon Lookout for Metrics in the *Service Authorization Reference*.

Policy actions in Lookout for Metrics use the following prefix before the action:

```
lookout for metrics
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "lookout for metrics:action1",
      "lookout for metrics:action2"
         ]
```

To view examples of Lookout for Metrics identity-based policies, see Identity-based policy examples for Amazon Lookout for Metrics.

Policy resources for Lookout for Metrics

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of Lookout for Metrics resource types and their ARNs, see Resources defined by Amazon Lookout for Metrics in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by Amazon Lookout for Metrics.

To view examples of Lookout for Metrics identity-based policies, see Identity-based policy examples for Amazon Lookout for Metrics.

Policy condition keys for Lookout for Metrics

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Lookout for Metrics condition keys, see Condition keys for Amazon Lookout for Metrics in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by Amazon Lookout for Metrics.

To view examples of Lookout for Metrics identity-based policies, see Identity-based policy examples for Amazon Lookout for Metrics.

ACLs in Lookout for Metrics

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with Lookout for Metrics

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with Lookout for Metrics

Supports temporary credentials: Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switch from a user to an IAM role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for Lookout for Metrics

Supports forward access sessions (FAS): Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a

different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for Lookout for Metrics

Supports service roles: Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break Lookout for Metrics functionality. Edit service roles only when Lookout for Metrics provides guidance to do so.

Service-linked roles for Lookout for Metrics

Supports service-linked roles: No

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a `Yes` in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for Amazon Lookout for Metrics

By default, users and roles don't have permission to create or modify Lookout for Metrics resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by Lookout for Metrics, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for Amazon Lookout for Metrics in the *Service Authorization Reference*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Lookout for Metrics resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies

adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.

- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the Lookout for Metrics console

To access the Amazon Lookout for Metrics console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Lookout for Metrics resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the Lookout for Metrics console, also attach the Lookout for Metrics *ConsoleAccess* or *ReadOnly* AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
```

```
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS managed policies for Amazon Lookout for Metrics

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed

policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

Managed policy: AmazonLookoutMetricsReadOnlyAccess

The AmazonLookoutMetricsReadOnlyAccess managed policy grants *read-only* permissions that allow read-only access to all Lookout for Metrics resources. You can attach this policy to your IAM entities.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lookoutmetrics:DescribeMetricSet",
        "lookoutmetrics:ListMetricSets",
        "lookoutmetrics:DescribeAnomalyDetector",
        "lookoutmetrics:ListAnomalyDetectors",
        "lookoutmetrics:DescribeAnomalyDetectionExecutions",
        "lookoutmetrics:DescribeAlert",
        "lookoutmetrics:ListAlerts",
        "lookoutmetrics:ListTagsForResource",
        "lookoutmetrics:ListAnomalyGroupSummaries",
        "lookoutmetrics:ListAnomalyGroupTimeSeries",
        "lookoutmetrics:ListAnomalyGroupRelatedMetrics",
        "lookoutmetrics:GetAnomalyGroup",
        "lookoutmetrics:GetDataQualityMetrics",
        "lookoutmetrics:GetSampleData",
        "lookoutmetrics:GetFeedback"
      ],
      "Resource": "*"
    }
  ]
}
```

Managed policy: AmazonLookoutMetricsFullAccess

The AmazonLookoutMetricsFullAccess managed policy grants full access to Lookout for Metrics resources, and permission to assign a service role to a detector. You can attach this policy to your

IAM entities. A *service role* is an IAM role that grants permissions to an AWS service so it can access resources. Service roles must be created by an administrator

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lookoutmetrics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::*:role/*LookoutMetrics*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "lookoutmetrics.amazonaws.com"
        }
      }
    }
  ]
}
```

Lookout for Metrics updates to AWS managed policies

View details about updates to AWS managed policies for Lookout for Metrics since this service began tracking these changes.

Change	Description	Date
AmazonLookoutMetricsReadOnlyAccess – Update policy	Lookout for Metrics updated the read-only policy to include `ListAnomalyGroupRelatedMetrics` .	November 24, 2021

Change	Description	Date
AmazonLookoutMetricsReadOnlyAccess – New policy	Lookout for Metrics added a new policy to allow read-only access for all Lookout for Metrics resources.	May 06, 2021
AmazonLookoutMetricsFullAccess – New policy	Lookout for Metrics added a new policy to allow full access to all Lookout for Metrics resources.	May 06, 2021
Lookout for Metrics started tracking changes	Lookout for Metrics started tracking changes for its AWS managed policies.	May 06, 2021

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that Amazon Lookout for Metrics gives another service to the resource. If you use both global condition context keys, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must use the same account ID when used in the same policy statement.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. If you don't know the full ARN of the resource or if you are specifying multiple resources, use the `aws:SourceArn`

global context condition key with wildcards (*) for the unknown portions of the ARN. For example, `arn:aws:servicename::123456789012:*`.

The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in Amazon SageMaker AI secret policies to prevent the confused deputy problem.

When you create a dataset in the console, or call the GetSampleData API operation directly, the trust policy can't refer to a specific resource. For existing detectors, you can specify the ARN of the detector.

Sections

Setting up a dataset

When you set up an Amazon S3 dataset, the console uses the service role that you provide to read sample data from the source bucket. For this operation, there isn't a specific resource to use for `SourceArn`. During setup, you can use a trust policy like the following, with a wildcard (*) for the `SourceArn` resource.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {
      "Service": "lookoutmetrics.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:lookoutmetrics:us-east-2:123456789012:*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Replace the region name and account ID with yours. After you've created the detector, you can replace the ARN with that of the detector. If you call the GetSampleData API operation directly, consider creating a separate service role specifically for that task.

Updating the trust policy after setup

After setup, you can update the trust policy to refer to a specific detector ARN.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {
      "Service": "lookoutmetrics.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:lookoutmetrics:us-
east-2:123456789012:AnomalyDetector:my-detector"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Troubleshooting Amazon Lookout for Metrics identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Lookout for Metrics and IAM.

Topics

- I am not authorized to perform an action in Lookout for Metrics
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my Lookout for Metrics resources

I am not authorized to perform an action in Lookout for Metrics

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `lookout for metrics:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: lookout
 for metrics:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `lookout for metrics:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Lookout for Metrics.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Lookout for Metrics. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my Lookout for Metrics resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Lookout for Metrics supports these features, see How Amazon Lookout for Metrics works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Compliance validation for Amazon Lookout for Metrics

Amazon Lookout for Metrics is not in scope of any AWS compliance programs. For a list of AWS services in scope of specific compliance programs, see AWS services in scope by compliance program. For general information, see AWS compliance programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading reports in AWS artifact.

Your compliance responsibility when using Lookout for Metrics is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and compliance quick start guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.

- Architecting for HIPAA security and compliance whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS compliance resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Config – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in Amazon Lookout for Metrics

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS global infrastructure.

Infrastructure security in Amazon Lookout for Metrics

As a managed service, Amazon Lookout for Metrics is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of security processes whitepaper.

You use AWS published API calls to access Lookout for Metrics through the network. Clients must support Transport Layer Security (TLS) 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Releases

The following table shows when features were released for the Lookout for Metrics service and documentation. For more information on a release, see the linked topic.

Change	Description	Date
Alert functionality improvements	You can now create more targeted alerts that focus on specific measures, dimensions, or both. For details, see Working with alerts.	June 14, 2022
Backtest support for Amazon CloudWatch	You can now run a backtest on data stored in Amazon CloudWatch. For details, see Running a backtest when using CloudWatch.	June 3, 2022
Amazon Athena database support	Lookout for Metrics can now connect to Amazon Athena as a datasource. For details, see Using Amazon Athena with Lookout for Metrics.	May 25, 2022
Amazon S3 dataset configuration detection	Lookout for Metrics automatically detects your data's file format settings when you set up a detector with an Amazon S3 dataset. This functionality is built into the management console, which uses the DetectMetricSetConfig API to infer settings from your data.	April 21, 2022
VPC endpoints	You can now create a VPC endpoint to use the Lookout	March 25, 2021

	for Metrics API from within a private VPC subnet without internet access. For details, see Working with Amazon VPC.	
Tags	You can now tag Lookout for Metrics resources to organize them into logical groups and apply least-privilege permissions policies. For details, see Tagging Lookout for Metrics resources.	March 25, 2021
General availability	Lookout for Metrics is now available to all AWS customers.	March 25, 2021
Preview	Preview release of Lookout for Metrics at re:Invent 2020.	December 8, 2020