Amazon DevOps Guru

User Guide

Amazon DevOps Guru: User Guide

Table of Contents

What is Amazon DevOps Guru? 1
How does DevOps Guru work? 1
High level DevOps Guru work flow 1
Detailed DevOps Guru workflow 2
How do I get started? 4
How do I stop incurring DevOps Guru charges? 4
Concepts 4
Anomaly 4
Insight 4
Metrics and operational events 5
Recommendations 5
Setting up 6
Sign up for AWS 6
Determine coverage 6
Identify your notifications topic 7
Estimate your cost 8
Getting started 10
Step 1: Get set up 10
Step 2: Enable DevOps Guru 10
Step 3: Specify AWS CloudFormation stacks for resource coverage 11
Working with insights 12
View insights 12
Understanding insights in the DevOps Guru console 13
Working with AWS CloudFormation stacks 15
Choose stacks to analyze 15
Choose stacks to analyze (console) 15
Choose stacks to analyze (DevOps Guru SDK) 16
Update settings 17
Update your AWS analysis coverage 17
Update your notifications 18
Filter your notifications 19
Example filtered Amazon SNS notification 19
Update Systems Manager integration 20
Best practices 21
Security 22
Data protection 22
Data encryption 23
Traffic privacy 23
Identity and Access Management 23
Audience 24
Authenticating with identities 24
Managing access using policies 26
Policy updates 28
How Amazon DevOps Guru works with IAM 28
Identity-based policies 33
Using service-linked roles 37
DevOps Guru permissions reference 40
Permissions for cross account Amazon SNS topics 43
Permissions for encrypted Amazon SNS topics 43
Troubleshooting 44
Monitoring DevOps Guru 46
Monitoring with CloudWatch 46
Logging DevOps Guru API calls with AWS CloudTrail 48
VPC endpoints (AWS PrivateLink) 50

Considerations for DevOps Guru VPC endpoints 50
Creating an interface VPC endpoint for DevOps Guru 50
Creating a VPC endpoint policy for DevOps Guru 50
Infrastructure security 51
Resilience 51
Quotas 52
Notifications 52
AWS CloudFormation stacks 52
Document history 53
AWS glossary 54

What is Amazon DevOps Guru?

Welcome to the Amazon DevOps Guru user guide.

DevOps Guru is a fully managed operations service that makes it easy for developers and operators to improve the performance and availability of their applications. DevOps Guru lets you offload the administrative tasks associated with identifying operational issues so that you can quickly implement recommendations to improve your application. DevOps Guru creates reactive insights you can use to improve your application now. It also creates proactive insights to help you avoid operational issues that might affect your application in the future.

DevOps Guru applies machine learning to analyze your operational data and application metrics and events to identify behaviors that deviate from normal operating patterns. You are notified when DevOps Guru detects an operational issue or risk. For each issue, DevOps Guru presents intelligent recommendations to address current and predicted future operational issues.

To get started, see How do I get started with DevOps Guru? (p. 4)

How does DevOps Guru work?

The DevOps Guru workflow begins when you configure its coverage and notifications. After you set up DevOps Guru, it starts to analyze your operational data. When it detects anomalous behavior, it creates an insight that contains recommendations and lists of metrics and events that are related to the issue. For each insight, DevOps Guru notifies you. If you enabled AWS Systems Manager OpsCenter, an OpsItem is created so you can use Systems Manager OpsCenter to track and manage addressing your insights. Each insight contains recommendations, metrics, and events related to anomalous behavior. Use information in an insight to help you understand and address the anomalous behavior.

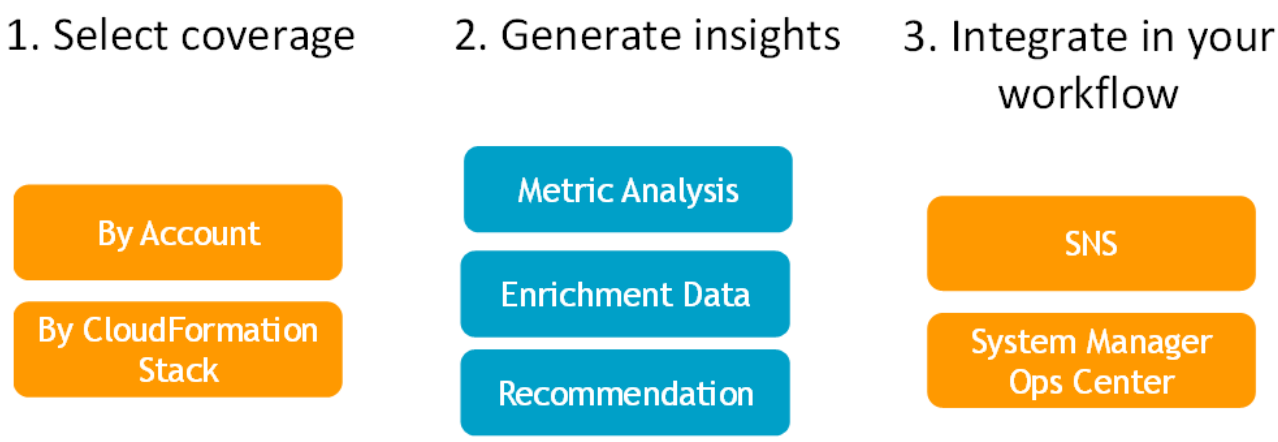
See High level DevOps Guru workflow (p. 1) for more detail about the three high-level workflow steps. See Detailed DevOps Guru workflow (p. 2) to learn about the more detailed DevOps Guru workflow, including how it interacts with other AWS services.

Topics

- High level DevOps Guru workflow (p. 1)
- Detailed DevOps Guru workflow (p. 2)

High level DevOps Guru workflow

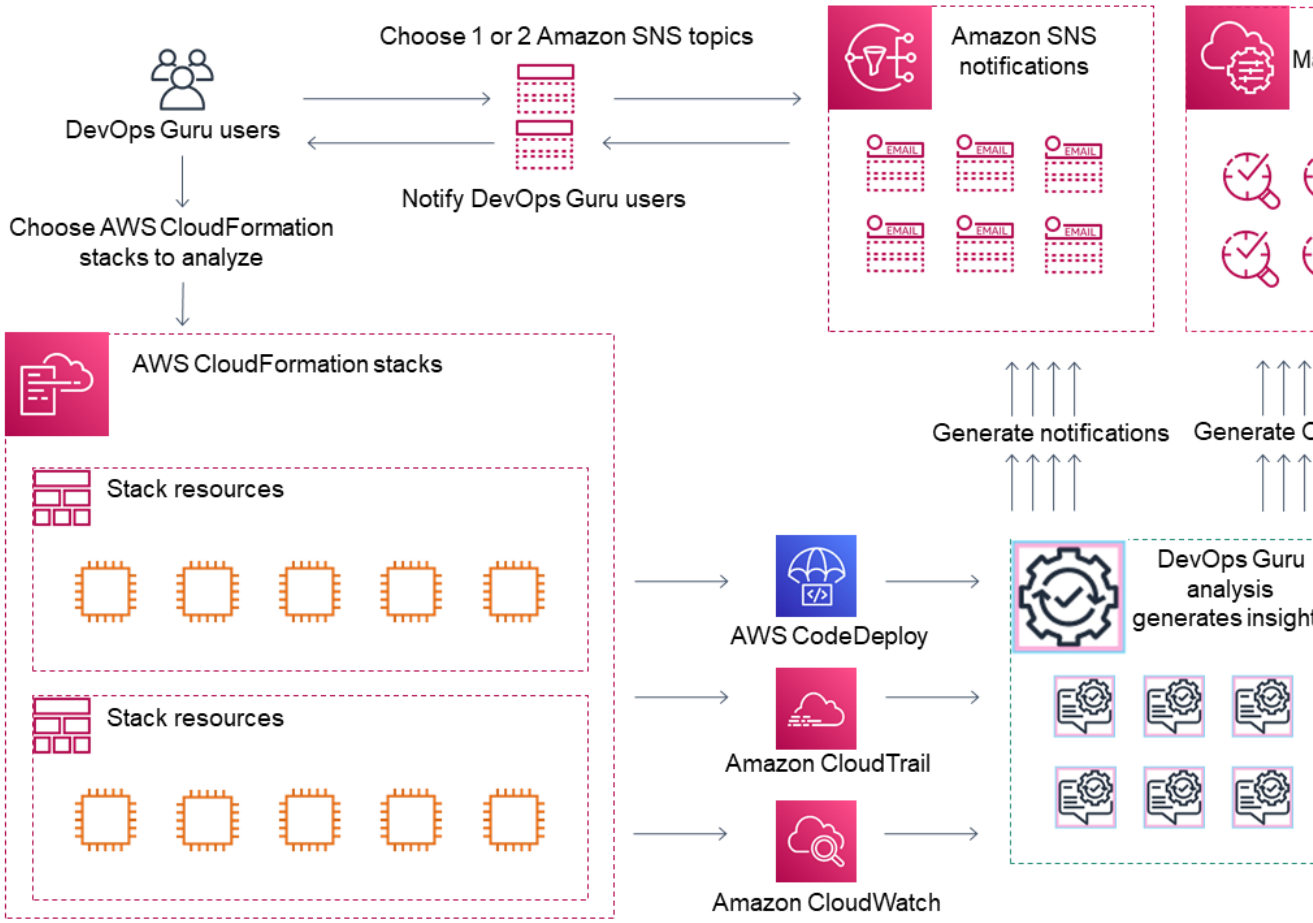
The Amazon DevOps Guru workflow can be broken down into three high level steps. First, you specify DevOps Guru coverage by telling it which AWS resources in your AWS account you want it to analyze. Second, DevOps Guru starts analyzing Amazon CloudWatch metrics, AWS CloudTrail, and other operational data to identify problems that you can fix to improve your operations. Third, DevOps Guru makes sure you know about insights and important information by sending you a notification for important DevOps Guru events. You can also configure DevOps Guru to create an OpsItem in AWS Systems Manager OpsCenter to help you track your insights. The following diagram shows this high-level workflow.



1. In the first step, you choose your coverage by specifying which AWS resources in your AWS account are analyzed. DevOps Guru can cover, or analyze, all the resources in an AWS account, or you can use AWS CloudFormation stacks to specify a subset of the resources in your account to analyze. Make sure that the resources you specify make up your business critical applications, workloads, and micro-services. For more information about the supported services and resources, see Amazon DevOps Guru pricing.
2. In the second step, DevOps Guru analyzes the resources to generate insights. This is an ongoing process. You can view the insights and see the recommendations and related information they contain in the DevOps Guru console. DevOps Guru analyzes the following data to find issues and create insights.
   - Individual Amazon CloudWatch metrics emitted by your AWS resources. When an issue is identified, DevOps Guru collects those metrics together.
   - DevOps Guru pulls enrichment data from AWS CloudTrail management logs to find events that are related to the collected metrics. The events can be resource deployment events and configuration changes.
   - If you use AWS CodeDeploy, DevOps Guru analyzes deployment events to help generate insights. Events for all types of CodeDeploy deployments (on-premises server, Amazon EC2 server, Lambda, or Amazon EC2) are analyzed.
   - When DevOps Guru finds a specific pattern, it generates one or more recommendations to help mitigate or fix the identified issue. The recommendations are collected in one insight. The insight also contains a list of the metrics and events that are related to the issue. You use the insight data to address and understand the identified problem.
3. In the third step, DevOps Guru integrates insight notification into your work flow to help you manage issues and quickly address them.
   - Insights generated in your AWS account are published to the Amazon Simple Notification Service (Amazon SNS) topic chosen during DevOps Guru setup. This is how you are notified as soon as an insight is created. For more information, see Update your notifications in DevOps Guru (p. 18).
   - If you enabled AWS Systems Manager during DevOps Guru setup, each insight creates a corresponding OpsItem to help you track and manage the issues discovered. For more information, see Update AWS Systems Manager integration in DevOps Guru (p. 20).

Detailed DevOps Guru workflow

The DevOps Guru workflow integrates with several AWS services, including Amazon CloudWatch, AWS CloudTrail, Amazon Simple Notification Service, and AWS Systems Manager. The following diagram shows a detailed workflow that includes how it works with other AWS services.



This diagram shows a scenario in which DevOps Guru coverage is specified by the AWS resources that are defined in AWS CloudFormation stacks. If no stacks are chosen, then DevOps Guru coverage analyzes all AWS resources in your account.

1. During setup, you specify one or two Amazon SNS topics that are used to notify you about important DevOps Guru events, such as when an insight is created. Next, you can specify AWS CloudFormation stacks that define the resources you want analyzed. You can also enable Systems Manager to generate an OpsItem for each insight to help you manage your insights.
2. After DevOps Guru is configured, it starts analyzing CloudWatch metrics and events that are emitted from your resources and AWS CloudTrail data related to the CloudWatch metrics. If your operations include CodeDeploy deployments, DevOps Guru also analyzes deployment events.

   DevOps Guru creates insights when it identifies unusual, anomalous behavior in the analyzed data. Each insight contains one or more recommendations, a list of the metrics used to generate the insight, and a list of the events used to generate the insight. Use this information to address the identified problem.
3. After each insight is created, DevOps Guru sends a notification using the Amazon SNS topic or topics specified during DevOps Guru set up. If you enabled DevOps Guru to generate an OpsItem in Systems Manager OpsCenter, then each insight also triggers a new Systems Manager OpsItem. You can use Systems Manager to manage your insight OpsItems.

How do I get started with DevOps Guru?

We recommend that you complete the following steps:

1. **Learn** more about DevOps Guru by reading the information in DevOps Guru concepts (p. 4).
2. **Set up** your AWS account, the AWS CLI, and an IAM user by following the steps in Setting up Amazon DevOps Guru (p. 6).
3. **Use** DevOps Guru, following the instructions in Getting started with DevOps Guru (p. 10).

How do I stop incurring DevOps Guru charges?

To disable Amazon DevOps Guru so that it stops incurring charges from analyzing resources in your AWS account and Region, update your coverage settings so that it doesn't analyze resources. To do this, follow the steps in Update your AWS analysis coverage in DevOps Guru (p. 17) and choose **Don't analyze any resources** in step 4. You must do this for each AWS account and Region where DevOps Guru analyzes resources.

> **Note**
> If you update your coverage to stop analyzing resources, you might continue to incur minor charges if you review existing insights generated by DevOps Guru in the past. These charges are associated with API calls used to retrieve and display insight information. For more information, see Amazon DevOps Guru pricing.

DevOps Guru concepts

The following concepts are important for understanding how Amazon DevOps Guru works.

Topics

- Anomaly (p. 4)
- Insight (p. 4)
- Metrics and operational events (p. 5)
- Recommendations (p. 5)

Anomaly

An anomaly represents one or more related metrics detected by DevOps Guru that are unexpected or unusual. DevOps Guru generates anomalies by using machine learning to analyze metrics and operational data that are related to your AWS resources. You specify the AWS resources that you want analyzed when you set up Amazon DevOps Guru. For more information, see Setting up Amazon DevOps Guru (p. 6).

Insight

An insight is a collection of anomalies that are created during the analysis of the AWS resources you specify when you set up DevOps Guru. Each insight contains observations, recommendations, and analytical data you can use to improve your operational performance. There are two types of insights:

- *Reactive*: A reactive insight identifies anomalous behavior as it occurs. It contains anomalies with recommendations, related metrics, and events to help you understand and address the issues now.

- *Proactive*: A proactive insight lets you know about anomalous behavior before it occurs. It contains anomalies with recommendations to help you address the issues before they are predicted to happen.

Metrics and operational events

The anomalies that make up an insight are generated by analyzing the metrics returned by Amazon CloudWatch and operational events emitted by your AWS resources. You can view the metrics and the operational events that create an insight to help you better understand issues in your application.

Recommendations

Each insight provides recommendations with suggestions to help you improve the performance of your application. The recommendation includes the following:

- A description of the recommendation actions to address the anomalies that comprise the insight.
- A list of the analyzed metrics in which DevOps Guru found anomalous behavior. Each metric includes the name of the AWS CloudFormation stack that generated the resource associated with the metrics, the resource's name, and the name of the AWS service associated with the resource.
- A list of the events that are related to the anomalous metrics associated with the insight. Each related event contains the name of the AWS CloudFormation stack that generated the resource associated with the event, the name of the resource that generated the event, and the name of the AWS service associated with the event.

Setting up Amazon DevOps Guru

Complete the tasks in this section to set up Amazon DevOps Guru for the first time. If you already have an AWS account, know which AWS account or accounts you want to analyze, and have an Amazon Simple Notification Service topic to use for insight notifications, you can skip ahead to Getting started with DevOps Guru (p. 10).

Optionally, you can use Quick Setup, a capability of AWS Systems Manager, to set up DevOps Guru and quickly configure its options. To use Quick Setup in Systems Manager, you must have the following prerequisites in place.

- An organization with AWS Organizations. For more information, see AWS Organizations terminology and concepts in the *AWS Organizations User Guide*.
- Two or more organizational units (OUs).
- One or more target AWS accounts in each OU.
- One administrator account with privileges to manage the target accounts.

To learn how to set up DevOps Guru using Quick Setup, see Congfigure DevOps Guru with Quick Setup in the *AWS Systems Manager User Guide*.

Use the following steps to set up DevOps Guru without Quick Setup.

- Step 1 – Sign up for AWS (p. 6)
- Step 2 – Determine coverage for DevOps Guru (p. 6)
- Step 3 – Identify your Amazon SNS notifications topic (p. 7)

Step 1 – Sign up for AWS

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Step 2 – Determine coverage for DevOps Guru

Think about how you want to configure coverage for Amazon DevOps Guru. Coverage determines the AWS resources that are covered, or analyzed, to detect anomalous behavior. You want DevOps Guru to cover the AWS resources that are created by the AWS services that make up your operational solutions. You have two options.

1. By default, DevOps Guru analyzes all supported AWS resources in your AWS Region and account. If you do not specify AWS CloudFormation stacks that define specific resources to cover, then all resources in your account are covered.

2. You can use AWS CloudFormation stacks to specify which resources are analyzed by DevOps Guru. Think about which resources you need, then create AWS CloudFormation templates that define and generate those resources for you. You specify your stacks when you configure DevOps Guru. You can also update your stacks at any time. For more information, see Working with AWS CloudFormation stacks in DevOps Guru (p. 15).

For more information, see Getting started with DevOps Guru (p. 10). For more information about the supported services and resources, see Amazon DevOps Guru pricing.

Step 3 – Identify your Amazon SNS notifications topic

You use one or two Amazon SNS topics to generate notifications about important DevOps Guru events, such as when an insight is created. This ensures you know about issues that DevOps Guru finds as soon as possible. Have your topics ready when you set up DevOps Guru. When you use the DevOps Guru console to set up DevOps Guru, you specify a notification topic using its name or its Amazon Resource Name (ARN). For more information, see Enable DevOps Guru. You can use the Amazon SNS console to view the name and ARN for each of your topics. If you don't have a topic, you can create one when you enable DevOps Guru using the DevOps Guru console. For more information, see Creating a topic in the *Amazon Simple Notification Service Developer Guide*.

Estimate Amazon DevOps Guru resource analysis costs

You can estimate your monthly cost for Amazon DevOps Guru to analyze your AWS resources. You pay for the number of hours analyzed for each active AWS resource in your specified resource coverage. A resource is active if it produces metrics, events, or logs within an hour.

DevOps Guru scans your selected resources to create a monthly cost estimate. You can view the resources, their hourly billable price, and their estimated monthly charge. The cost estimator assumes as a default that the analyzed active resources are utilized 100 percent of the time. You can change this percentage for each analyzed service based on your estimated usage to create an updated monthly cost estimate. The estimate is for the cost to analyze your resources and does not include costs associated with DevOps Guru API calls.

You can create one cost estimate at a time. The time it takes to generate a cost estimate depends on the number of resources you specify when you create the cost estimate. When you specify a lot of resources, it can take up to four hours to complete. Your actual costs vary and depend on the percentage of time your analyzed active resources are utilized.

> **Note**
> For a cost estimate, you can specify only one AWS CloudFormation stack. For your actual coverage boundary, you can specify up to 1000 stacks.

To create a monthly resource analysis cost estimate

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Choose **Cost estimator** in the navigation pane.
3. If you have not enabled DevOps Guru, you must create an IAM role. In **Create IAM role for DevOps Guru**, choose **Create IAM role**. If you have aleady enabled DevOps Guru, the role has already been created so this option does not appear.
4. Choose the resources you want to use to create your estimate.
   - If you want to estimate the cost for DevOps Guru to analyze the resources defined by one AWS CloudFormation stack, do the following.
     1. Choose **CloudFormation stack**.
     2. Enter the name of an AWS CloudFormation stack in your AWS account in **Enter CloudFormation stack name**. For information about working with and viewing your stacks, see Working with stacks in the *AWS CloudFormation User Guide*.
     3. (Optional) If you use an AWS CloudFormation stack that you are currently not analyzing, choose **Enable resource analysis** to enable DevOps Guru to start analyzing its resources. This option is not available if you have not enabled DevOps Guru or if you are already analyzing the resources in the stack.
   - If you want to estimate the cost for DevOps Guru to analyze the resource in your AWS account and Region, choose **Current AWS account**.
5. Choose **Estimate monthly cost**.
6. (Optional) In the **Active resource utilization %** column, enter an updated percentage value for one or more AWS services. The default *active resource utilization %* is 100%. This means that DevOps Guru generates the estimate for the AWS service by calculating the cost of one hour of analyzing its resources, then extrapolating that over 30 days for a total of 720 hours. If a service is active less than 100% of the time, you can update the percentage based on your estimated usage for a more

accurate estimate. For example, if you update a service's active resource utilization to 75%, the one hour cost of analyzing its resources is extrapolated over (720 x 0.75) hours, or 540 hours.

If your estimate is zero dollars, then the resources you chose likely do not include resources supported by DevOps Guru. For more information about the supported services and resources, see Amazon DevOps Guru pricing.

Getting started with DevOps Guru

In this section, you learn how to get started with Amazon DevOps Guru so it can analyze your application's operational data and metrics to generate insights.

Topics

- Step 1: Get set up (p. 10)
- Step 2: Enable DevOps Guru (p. 10)
- Step 3: Specify AWS CloudFormation stacks for DevOps Guru resource coverage (p. 11)

Step 1: Get set up

Before you get started, you must prepare by running through the steps in Setting up Amazon DevOps Guru (p. 6).

Step 2: Enable DevOps Guru

To configure Amazon DevOps Guru to use for the first time, you must choose which AWS resources in your account and Region is covered, or analyzed, and specify one or two Amazon Simple Notification Service topics that are used to notify you when an insight is created. You can update these settings later as needed.

Enable DevOps Guru

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. In **DevOps Guru analysis coverage**, choose one of the following.

   - **Analyze all AWS resources in the current AWS account**: DevOps Guru analyzes all AWS resources in your account.
   - **Choose AWS resources to analyze later**: DevOps Guru analyzes only AWS resources that are defined in AWS CloudFormation stacks that you specify later.

   DevOps Guru can analyze any resource that is associated with the AWS it supports. For more information about the supported services and resources, see Amazon DevOps Guru pricing.
3. You can add up to two topics. DevOps Guru uses the topic or topics to notify you about important DevOps Guru events, such as the creation of a new insight. If you don't specify a topic now, you can add one later by choosing **Settings** in the navigation pane.

   a. In **Specify an Amazon SNS topic**, choose a topic to use.
   b. To add an Amazon SNS topic, do one of the following.

      - Choose **Chose an existing SNS topic in your AWS account**. Then, from **Choose a topic in your AWS account**, choose the topic you want to use.
      - Choose **Create a new SNS topic**. Then, in **Create a new topic**, enter the name for your new topic.
      - Choose **Use an SNS topic ARN to specify an existing account**. Then, in **Enter an ARN for a topic**, enter the topic ARN. The ARN is the topic's Amazon Resource Name. You can specify

a topic in a different account. If you use a topic in another account, you must add a resource policy to the topic. For more information, see Permissions for cross account Amazon SNS topics (p. 43).

c. Choose **Add SNS topic** if you want to add a second topic.

d. Choose **Save**.

4. Choose **Enable**.

Step 3: Specify AWS CloudFormation stacks for DevOps Guru resource coverage

If you chose to specify AWS resources later than when you enable DevOps Guru, you need to choose which AWS CloudFormation stacks in your AWS account create the resources you want analyzed. An AWS CloudFormation stack is a collection of AWS resources that you manage as a single unit. You can use one or more stacks to include all the resources required to run your operational applications, then specify them so that they are analyzed by DevOps Guru. If you don't specify stacks, then DevOps Guru analyzes all the AWS resources in your account. For more information, see Working with stacks in the *AWS CloudFormation User Guide*, and Step 2 – Determine coverage for DevOps Guru (p. 6). and Working with AWS CloudFormation stacks in DevOps Guru (p. 15)

Note
For more information about supported services and resources, see Amazon DevOps Guru pricing.

Specify AWS CloudFormation stacks for DevOps Guru resource coverage

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Choose **Settings** in the navigation pane.
3. In **DevOps Guru analysis coverage**, choose **Manage**.
4. If you have not enabled any stacks, in **CloudFormation stacks**, choose **Manage analysis coverage**.
5. Select up to 1000 stacks that contain the resources that you want analyzed. You can enter the name of a stack in **Find stacks** to quickly locate a specific stack.
6. Choose **Save**.

Working with insights in DevOps Guru

An *insight* is generated by Amazon DevOps Guru when it detects anomalous behavior in your operational applications. DevOps Guru analyzes the metrics, events, and more in the AWS resources you specified when you set up DevOps Guru. Each insight contains one or more recommendation for you to take to mitigate the issue. It also contains a list of the metrics and a list of the events that were used to identify the unusual behavior.

There are two insight types.

- *Reactive* insights have recommendations you can take to address issues that are happening now.
- *Proactive* insights have recommendations that address issues that DevOps Guru predicts will occur in the future.

Topics

- View DevOps Guru insights (p. 12)
- Understanding insights in the DevOps Guru console (p. 13)

View DevOps Guru insights

You can view your insights using the AWS Management Console.

View your DevOps Guru insights

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Open the navigation pane, then choose **Insights**.
3. On the **Reactive** tab, you can see a list of reactive insights. On the **Proactive** tab, you can see your proactive insights.
4. (Optional) Use one or more of the following filters to find the insights you are looking for.
   - Choose the **Reactive** or **Proactive** tab, depending on the type of insight you are looking for.
   - Choose **Filter insights by status, severity, or resource name**, then choose the **Status**, **Severity**, or **Resource name** option to specify a filter. You can add a status, severity, and resource filter.
   - Choose or specify a time range to filter by insight creation time.
     - **12h** shows insights created in the past 12 hours.
     - **1d** shows insights created in the past day.
     - **1w** shows insights created in the past week.
     - **1M** shows insights created in the past month.
     - **Custom** lets you specify another time range. The maximum time range you can use to filter insights is 180 days.
5. To view details about an insight, choose its name.

Understanding insights in the DevOps Guru console

Use the Amazon DevOps Guru console to view useful information in your insights to help you diagnose and address anomalous behavior. When DevOps Guru analyzes your resources and finds related Amazon CloudWatch metrics, AWS CloudTrail events, and operational data that indicate unusual behavior, it creates an insight that contains recommendations to address the issue and information about the related metrics and events. Use insight data with Best practices in DevOps Guru (p. 21) to address operational problems detected by DevOps Guru.

To view an insight, follow the steps in View insights (p. 12) to find one, then choose its name. The insight page contains the following details.

Insight overview

Use this section to get a high-level overview of the insight. You can see the status of the insight (*Ongoing* or *Closed*), how many AWS CloudFormation stacks are affected, when the insight started, ended, and was last updated, and the related operations item if there is one.

Choose the number of affected stacks to see their names. The anomalous behavior that created the insight occurred in resources created by the affected stacks.

Aggregated metrics

Choose the **Aggregated metrics** tab to view metrics that are related to the insight. In the table, each row represents one metric. You can see which AWS CloudFormation stack created the resource that emitted the metric, the name of the resource, and its type. Not all metrics are associated with an AWS CloudFormation stack or have a name.

When there are multiple resources anomalous at the same time, the timeline view aggregates the resources and presents their anomalous metrics in a single timeline for easy analysis. The red lines on a timeline indicate spans of time when a metric emitted unusual values. To zoom in, use your mouse to choose a specific time range. You can also use the magnifying glass icons to zoom in and out.

Choose a red line in the timeline to view detailed information. In the window that opens, you can:

- Choose **View in CloudWatch** to see how the metric looks in the CloudWatch console. For more information, see Statistics and Dimensions in the *Amazon CloudWatch User Guide*.
- Hover over the graph to view details about the anomalous metric data and when it occurred.
- Choose the box with the downward arrow to download a PNG image of the graph.

Graphed anomalies

Choose the **Graphed anomalies** tab to view detailed graphs for each of the insight's anomalies. One tile appears for each anomaly with details about unusual behavior detected in related metrics. You can investigate and look at an anomaly at the resource level and per statistic. The graphs are grouped by metric name. In each tile, you can choose a specific time range in the timeline to zoom. You can also use the magnifying glass icons to zoom in and out, or choose a predefined duration in hours, days, or weeks (**1H**, **3H**, **12H**, **1D**, **3D**, **1W**, or **2W**).

Choose **View all statistics and dimensions** to see details about the anomaly. In the window that opens, you can:

- Choose **View in CloudWatch** to see how the metric looks in the CloudWatch console.
- Hover over the graph to view details about the anomalous metric data and when it occurred.
- Choose **Statistics** or **Dimension** to customize the graph's display. For more information, see Statistics and Dimensions in the *Amazon CloudWatch User Guide*.

Related events

In **Related events**, view AWS CloudTrail events that are related to your insight. Use these events to help understand, diagnose, and address the underlying cause of the anomalous behavior.

Recommendations

In **Recommendations**, you can view suggestions that might help you resolve the underlying problem. When DevOps Guru detects anomalous behavior, it attempts to create recommendations. An insight might contain one, multiple, or zero recommendations.

Working with AWS CloudFormation stacks in DevOps Guru

You can use AWS CloudFormation stacks to specify which AWS resources you want DevOps Guru to analyze. A stack is a collection of AWS resources that are managed as a single unit. The resources in the stacks you choose make up your DevOps Guru coverage boundary. For each stack you choose, operational data in its supported resources are analyzed for anomalous behavior. Those issues are then grouped into related anomalies to create insights. Each insight includes one or more recommendations to help you address them. The maximum number of stacks you can specify is 1000. For more information, see Working with stacks in the *AWS CloudFormation User Guide* and Update your AWS analysis coverage in DevOps Guru (p. 17).

After you choose a stack, DevOps Guru immediately starts to analyze any resource you add to it. If you remove a resource from a stack, it is no longer analyzed.

> **Note**
> If you choose to have DevOps Guru analyze all supported resources in your account (this means your account is your DevOps Guru coverage boundary), then DevOps Guru analyzes and creates insights for every supported resource in your account. While all your insights appear in the console, only the insights created for a resource that belongs to one of the first 1000 stacks analyzed appear as part of a stack. DevOps Guru can analyze up to 10,000 stacks.

Choose stacks for DevOps Guru to analyze

Specify the resources that you want Amazon DevOps Guru to analyze by choosing the AWS CloudFormation stacks that create them. You can do this using the AWS Management Console or the SDK.

Topics

- Choose stacks for DevOps Guru to analyze (console) (p. 15)
- Choose stacks for DevOps Guru to analyze (DevOps Guru SDK) (p. 16)

Choose stacks for DevOps Guru to analyze (console)

You can add AWS CloudFormation stacks using the console.

To choose the stacks that contain the resources to analyze

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Open the navigation pane, then choose **Settings**.
3. In **DevOps Guru analysis coverage**, choose **Manage**.
4. If you have not enabled any stacks, in **CloudFormation stacks**, choose **Manage analysis coverage**.
5. Select up to 1000 stacks that contain the resources that you want analyzed. You can enter the name of a stack in **Find stacks** to quickly locate a specific stack.
6. Choose **Save**.

Choose stacks for DevOps Guru to analyze (DevOps Guru SDK)

To specify AWS CloudFormation stacks using the Amazon DevOps Guru SDK, use the `UpdateResourceCollection` method. For more information, see UpdateResourceCollection in the *Amazon DevOps Guru API Reference*.

Update DevOps Guru settings

You can update the following Amazon DevOps Guru settings.

- Your DevOps Guru coverage. This determines which resources in your account are analyzed.
- Your notifications. This determines which Amazon Simple Notification Service topics are used to notify you of important DevOps Guru events.
- Your AWS Systems Manager integration. This determines whether an OpsItem is created in Systems Manager OpsCenter for each new insight.

Topics

- Update your AWS analysis coverage in DevOps Guru (p. 17)
- Update your notifications in DevOps Guru (p. 18)
- Filter your DevOps Guru notifications (p. 19)
- Update AWS Systems Manager integration in DevOps Guru (p. 20)

Update your AWS analysis coverage in DevOps Guru

You can update which AWS resources in your account DevOps Guru analyzes. The resources that are analyzed make up your DevOps Guru coverage boundary. You have three boundary coverage options.

- Choose to have DevOps Guru analyze all supported resources in your account.
- Specify specific resources by choosing AWS CloudFormation stacks that define those resources. If you do this, DevOps Guru analyzes every resource specified in the stacks you choose. If a resource in your account is not defined by a stack you choose, it is not analyzed.
- Specify to have no resources analyzed so that you stop incurring charges from resource analyzation.

  Note
  If you update your coverage to stop analyzing resources, you might continue to incur minor charges if you review existing insights generated by DevOps Guru in the past. These charges are associated with API calls used to retrieve and display insight information. For more information, see Amazon DevOps Guru pricing.

DevOps Guru supports all resources that are associated with supported services. For more information about the supported services and resources, see Amazon DevOps Guru pricing.

To manage your DevOps Guru analysis coverage

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Choose **Settings** in the navigation pane.
3. In **DevOps Guru analysis coverage**, choose **Manage**.
4. Choose one of the following coverage options.

   - Choose **Analyze all AWS resources in the current account** if you want DevOps Guru to analyze all supported resources in your AWS account. If you choose this option, your account is your resource analysis coverage boundary.

- Choose **Specify which resources in the current AWS account to analyze using CloudFormation stacks** if you want DevOps Guru to analyze the resources that are in stacks you choose.

  1. If you have not enabled any stacks, in **CloudFormation stacks**, choose **Manage analysis coverage**.
  2. Select up to 1000 stacks that contain the resources that you want analyzed. You can enter the name of a stack in **Find stacks** to quickly locate a specific stack.
  3. Choose **Save**.

  For more information, see Working with AWS CloudFormation stacks in DevOps Guru (p. 15).
- Choose **Don't analyze any resources** if you do not want DevOps Guru to analyze any resources. This option disables DevOps Guru so that you stop incurring charges from resource analyzation.

5. Choose **Save**.

Update your notifications in DevOps Guru

Set up Amazon Simple Notification Service topics that are used to notify you about important Amazon DevOps Guru events. You can choose from a list of topic names that already exist in your AWS account, enter the name for a new topic that DevOps Guru creates in your account, or enter the Amazon Resource Name (ARN) of an existing topic in any AWS account in your Region. If you specify the ARN of a topic that is not in your account, you must grant permission for DevOps Guru to access that topic by adding an IAM policy to it. For more information, see Permissions for cross account Amazon SNS topics (p. 43). You can specify up to two topics.

DevOps Guru sends notifications for the following events.

- A new insight is created.
- A new anomaly is added to an insight.
- The severity of an insight is upgraded from `Low` or `Medium` to `High`.
- The status of an insight changes from ongoing to resolved.
- A recommendation for an insight is identified.

To update your notifications

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Choose **Settings** in the navigation pane.
3. In **SNS notifications**, choose **Set up notifications** if you have not set up any notifications. Otherwise, choose **Edit**.
4. To add an Amazon SNS topic, do one of the following.

   - Choose **Chose an existing SNS topic in your AWS account**. Then, from **Choose a topic in your AWS account**, choose the topic you want to use.
   - Choose **Create a new SNS topic**. Then, in **Create a new topic**, enter the name for your new topic.
   - Choose **Use an SNS topic ARN to specify an existing account**. Then, in **Enter an ARN for a topic**, enter the topic ARN. The ARN is the topic's Amazon Resource Name. You can specify a topic in a different account. If you use a topic in another account, you must add a resource policy to the topic. For more information, see Permissions for cross account Amazon SNS topics (p. 43).
5. Choose **Add SNS topic** if you want to add a second topic.
6. Choose **Save**.
7. To remove an Amazon SNS topic, choose **Remove** next to the topic you want to remove.

Filter your DevOps Guru notifications

You can create an Amazon Simple Notification Service (Amazon SNS) subscription filter policy to reduce the number of notifications you receive from Amazon DevOps Guru.

Topics

- Example filtered Amazon SNS notification for Amazon DevOps Guru (p. 19)

Use a filter policy to specify the types of notifications you receive. You can filter your Amazon SNS messages using the following keywords.

- `NEW_INSIGHT` — Receive a notification when a new insight is created.
- `CLOSED_INSIGHT` — Receive a notification when an existing insight is closed.
- `NEW_RECOMMENDATION` — Receive a notification when a new recommendation is created from an insight.
- `NEW_ASSOCIATION` — Receive a notification when a new anomaly is detected from an insight.
- `CLOSED_ASSOICATION` — Receive a notification when an existing anomaly is closed.
- `SEVERITY_UPGRADED` — Receive a notification when the severity of an insight is upgraded

For information about how to create an Amazon SNS subscription filter policy, see Amazon SNS subscription filter policies in the *Amazon Simple Notification Service Developer Guide*. In your filter policy, you specify one of the keywords with the policy's `MessageType`. For example, the following would appear in a filter that specifies the Amazon SNS topic only deliver notifications when a new anomaly is detected from an insight.

```
{
  "MessageType":["NEW_ ASSOCIATION"]
}
```

Example filtered Amazon SNS notification for Amazon DevOps Guru

The following is an example of an Amazon Simple Notification Service (Amazon SNS) notification from an Amazon SNS topic with a filter policy. Its `MessageType` is set to `NEW_ASSOCIATION`, so it sends notifications only when a new anomaly is detected from an insight.

```
{
  "Type" : "Notification",
  "MessageId" : "9ff514ee-ba4a-515a-9298-4d6887a89c59",
  "TopicArn" : "arn:aws:sns:us-east-1:123456789012:DevOpsGuru-insights-sns",
  "Timestamp" : "2021-08-05T19:27:30.731Z",
  "MessageAttributes" : {
      "MessageType" : {
          "Type":"String",
          "Value":"NEW_ASSOCIATION"
      }
  },
  "Message" : {
      "AccountId": "123456789012",
      "Region": "us-east-1",
      "MessageType": "NEW_ASSOCIATION",
      "InsightId": "ADyf4FvaVNDzu9MA2-IgFDkAAAAAAAAAEGpJd5sjicgauU2wmAlnWUyyI2hiO5it",
      "InsightDescription": "ThrottledRequests",
```

```
      "StartTime": 1628767500000,
      "Anomalies": [
        {
          "Id": "AG2n8ljW74BoI1CHu-m_oAgAAAF7Ohu24N4Yro69ZSdUtn_alzPH7VTpaL30JXiF",
          "StartTime": 1628767500000,
          "SourceDetails": [
            {
              "DataSource": "CW_METRICS",
              "DataIdentifiers": {
                "stat": "Sum",
                "unit": "None",
                "period": "60",
                "ResourceId": "TaskRecords",
                "namespace": "AWS/DynamoDB",
                "name": "ThrottledRequests",
                "ResourceType": "DynamoDB/Table",
                "dimensions": "{\"TableName\":\"TaskRecords\",\"Operation\":\"BatchGetItem
\"}"
              }
            }
          ]
        }
      ],
      "awsInsightSource": "aws.devopsguru"
  }
}
```


Update AWS Systems Manager integration in DevOps Guru

You can enable the creation of an OpsItem for each new insight in AWS Systems Manager OpsCenter. OpsCenter is a centralized system where you can view, investigate, and review operational work items (OpsItems). The OpsItems for your insights can help you manage work that addresses the anomalous behavior that triggered the creation of each insight. For more information, see AWS Systems Manager OpsCenter and Working with OpsItem in the *AWS Systems Manager User Guide*.

> **Note**
> If you change the key or value of the tag field of an OpsItem, then DevOps Guru is not able to update that OpsItem. For example, if you change a tag of an OpsItem from `"aws:RequestTag/DevOps-GuruInsightSsmOpsItemRelated": "true"` to something else, then DevOps Guru cannot update that OpsItem.

To manage your Systems Manager integration

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Choose **Settings** in the navigation pane.
3. In **AWS Systems Manager integration**, select **Enable DevOps Guru to create an AWS OpstItem in OpsCenter for each insight** to have an OpsItem created for each new insight. Deselect it to stop having an OpsItem created for each new insight.

You are charged for OpsItems created in your account. For more information, see AWS Systems Manager pricing.

Best practices in DevOps Guru

The following are some best practices to help you understand, diagnose, and fix anomalous behavior detected by Amazon DevOps Guru. Use best practices with Understanding insights in the DevOps Guru console (p. 13) to address operational problems detected by DevOps Guru.

- In an insight's timeline view, look at the highlighted metrics first. They are often key indicators of the problem.
- Use Amazon CloudWatch to view metrics that occurred immediately before the first highlighted metric in an insight. This can help you pinpoint when and how behavior changed to help you diagnose and fix the problem.
- Multiple dimensions of the same metric can often be anomalous. Look at the dimensions in the graphed view to get a deeper understanding of the problem.
- Look in the events section of an insight for deployment or infrastructure events that happened around the time the insight was created. Knowing which events occurred when an insight's anomalous behavior occurred can help you understand and diagnose the problem.
- Look for tickets in your operational system that happened around the same time as an insight for clues.
- In an insight, read the recommendations and visit the links in recommendations. These often have troubleshooting steps that can help you diagnose and solve problems quickly.
- Don't ignore resolved insights unless you have already solved the problem. Once a day, look at new insights, even if they have been resolved. Try to understand the root cause behind as many of the insights as you can. Look for a pattern that might be the sign of a systemic problem. If a systemic problem is left unresolved, it could cause more serious problems in the future. Fixing transient issues now can help prevent future, more serious, incidents.

Security in Amazon DevOps Guru

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are 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 DevOps Guru, 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 DevOps Guru. The following topics show you how to configure DevOps Guru to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your DevOps Guru resources.

Topics

- Data protection in Amazon DevOps Guru (p. 22)
- Identity and Access Management for Amazon DevOps Guru (p. 23)
- Logging and monitoring DevOps Guru (p. 46)
- DevOps Guru and interface VPC endpoints (AWS PrivateLink) (p. 50)
- Infrastructure security in DevOps Guru (p. 51)
- Resilience in Amazon DevOps Guru (p. 51)

Data protection in Amazon DevOps Guru

The AWS shared responsibility model applies to data protection in Amazon DevOps Guru. 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. This content includes 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 user accounts with 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 recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- 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 personal data that is stored in Amazon S3.

- If you require FIPS 140-2 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) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with DevOps Guru or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form 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.

Data encryption in DevOps Guru

Encryption is an important part of DevOps Guru security. Some encryption, such as for data in transit, is provided by default and does not require you to do anything. Other encryption, such as for data at rest, you can configure when you create your project or build.

- **Encryption of data at-rest**: For all AWS resources analyzed by DevOps Guru, the Amazon CloudWatch metrics and data, resources IDs, and AWS CloudTrail events are stored using Amazon S3, Amazon DynamoDB, and Amazon Kinesis. If AWS CloudFormation stacks are used to define the analyzed resources, then stack data is also collected. DevOps Guru uses the data retention policies of Amazon S3, DynamoDB, and Kinesis. Data stored in Kinesis can be retained for up to one year and depends on the policies set. Data stored in Amazon S3 and DynamoDB is stored for one year.

  Stored data is encrypted using the data-at-rest encryption capabilities of Amazon S3, DynamoDB, and Kinesis.
- **Encryption of data in-transit**: All communication between customers and DevOps Guru and between DevOps Guru and its downstream dependencies is protected using TLS and authenticated using the Signature Version 4 signing process. All DevOps Guru endpoints use certificates managed by AWS Certificate Manager Private Certificate Authority. For more information, see Signature Version 4 signing process and What is ACM PCA.

Traffic privacy

You can improve the security of your resource analysis and insight generation by configuring DevOps Guru to use an interface VPC endpoint. To do this, you do not need an internet gateway, NAT device, or virtual private gateway. It also is not required to configure PrivateLink, though it is recommended. For more information, see DevOps Guru and interface VPC endpoints (AWS PrivateLink) (p. 50). For more information about PrivateLink and VPC endpoints, see AWS PrivateLink and Accessing AWS services through PrivateLink.

Identity and Access Management for Amazon DevOps Guru

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 DevOps Guru resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience (p. 24)

- Authenticating with identities (p. 24)
- Managing access using policies (p. 26)
- DevOps Guru updates to AWS managed policies and service-linked role (p. 28)
- How Amazon DevOps Guru works with IAM (p. 28)
- Identity-based policies for Amazon DevOps Guru (p. 33)
- Using service-linked roles for DevOps Guru (p. 37)
- Amazon DevOps Guru permissions reference (p. 40)
- Permissions for cross account Amazon SNS topics (p. 43)
- Permissions for AWS KMS–encrypted Amazon SNS topics (p. 43)
- Troubleshooting Amazon DevOps Guru identity and access (p. 44)

Audience

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

Service user – If you use the DevOps Guru service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more DevOps Guru 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 DevOps Guru, see Troubleshooting Amazon DevOps Guru identity and access (p. 44).

Service administrator – If you're in charge of DevOps Guru resources at your company, you probably have full access to DevOps Guru. It's your job to determine which DevOps Guru features and resources your employees 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 DevOps Guru, see How Amazon DevOps Guru works with IAM (p. 28).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to DevOps Guru. To view example DevOps Guru identity-based policies that you can use in IAM, see Identity-based policies for Amazon DevOps Guru (p. 33).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see Signing in to the AWS Management Console as an IAM user or root user in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the AWS Management Console, use your password with your root user email address or your IAM user name. You can access AWS programmatically using your root user or IAM users access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 signing process in the *AWS General Reference*.

Regardless of the authentication method that you use, you might also 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 Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you first create an AWS account, you begin with a single 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 do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see Managing access keys for IAM users in the *IAM User Guide*. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

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 When to create an IAM user (instead of a role) 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. You can temporarily assume an IAM role in the AWS Management Console by switching roles. 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 Using IAM roles in the *IAM User Guide*.

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

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the *IAM User Guide*.
- **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 How IAM roles differ from resource-based policies 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.

  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for Amazon DevOps Guru in the *Service Authorization Reference*.
  - **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 Creating 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 IAM 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 Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. You can sign in as the root user or an IAM user, or you can assume an IAM role. When you then make a request, AWS evaluates the related identity-based or resource-based policies. 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**.

Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

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 Creating IAM 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 Choosing 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 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 How SCPs work 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*.

DevOps Guru updates to AWS managed policies and service-linked role

View details about updates to AWS managed policies and service-linked role for DevOps Guru since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the DevOps Guru Amazon DevOps Guru document history (p. 53).

Change	Description	Date
Service-linked role permissions for DevOps Guru (p. 38) – Update to an existing policy.	The `AmazonDevOpsGuruServiceRolePolicy` service-linked role now contains new conditions on the `ssm:CreateOpsItem` and `ssm:AddTagsToResource` actions.	June 14, 2021
AmazonDevOpsGuruReadOnlyAccess (p. 36) – Update to an existing policy	The `AmazonDevOpsGuruReadOnlyAccess` managed policy now allows read-only access to the AWS Identity and Access Management `GetRole` and the DevOps Guru `DescribeFeedback` actions.	June 14, 2021
AmazonDevOpsGuruReadOnlyAccess (p. 36) – Update to an existing policy	The `AmazonDevOpsGuruReadOnlyAccess` managed policy now allows read-only access to the DevOps Guru `GetCostEstimation` and `StartCostEstimation` actions.	April 27, 2021
Service-linked role permissions for DevOps Guru (p. 38) – Update to an existing policy	The `AmazonDevOpsGuruServiceRolePolicy` role now allows access to the AWS Systems Manager `AddTagsToResource` and Amazon EC2 Auto Scaling `DescribeAutoScalingGroups` actions.	April 27, 2021
DevOps Guru started tracking changes	DevOps Guru started tracking changes for its AWS managed policies.	December 10, 2020

How Amazon DevOps Guru works with IAM

Before you use IAM to manage access to DevOps Guru, learn what IAM features are available to use with DevOps Guru.

IAM features you can use with Amazon DevOps Guru

IAM feature	DevOps Guru support
Identity-based policies (p. 29)	Yes
Resource-based policies (p. 29)	No
Policy actions (p. 30)	Yes
Policy resources (p. 30)	No
Policy condition keys (p. 31)	Yes
ACLs (p. 31)	No
ABAC (tags in policies) (p. 32)	No
Temporary credentials (p. 32)	Yes
Principal permissions (p. 32)	Yes
Service roles (p. 33)	No
Service-linked roles (p. 33)	Yes

To get a high-level view of how DevOps Guru 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 DevOps Guru

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 Creating IAM 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 DevOps Guru

To view examples of DevOps Guru identity-based policies, see Identity-based policies for Amazon DevOps Guru (p. 33).

Resource-based policies within DevOps Guru

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 How IAM roles differ from resource-based policies in the *IAM User Guide*.

Policy actions for DevOps Guru

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 DevOps Guru actions, see Actions defined by Amazon DevOps Guru in the *Service Authorization Reference*.

Policy actions in DevOps Guru use the following prefix before the action:

```
awes
```

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

```
"Action": [
      "awes:action1",
      "awes:action2"
         ]
```

To view examples of DevOps Guru identity-based policies, see Identity-based policies for Amazon DevOps Guru (p. 33).

Policy resources for DevOps Guru

Supports policy resources	No

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

To view examples of DevOps Guru identity-based policies, see Identity-based policies for Amazon DevOps Guru (p. 33).

Policy condition keys for DevOps Guru

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

To view examples of DevOps Guru identity-based policies, see Identity-based policies for Amazon DevOps Guru (p. 33).

Access control lists (ACLs) in DevOps Guru

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.

Attribute-based access control (ABAC) with DevOps Guru

Supports ABAC (tags in policies)	No

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.

For more information about ABAC, see What is ABAC? 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 DevOps Guru

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 Switching to a 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 DevOps Guru

Supports principal permissions	Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for Amazon DevOps Guru in the *Service Authorization Reference*.

Service roles for DevOps Guru

Supports service roles	No

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 Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

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

Service-linked roles for DevOps Guru

Supports service-linked roles	Yes

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 IAM 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 policies for Amazon DevOps Guru

By default, IAM users and roles don't have permission to create or modify DevOps Guru resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform actions on the resources that they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating policies on the JSON tab in the *IAM User Guide*.

Topics

- Policy best practices (p. 33)
- Using the DevOps Guru console (p. 34)
- Allow users to view their own permissions (p. 34)
- AWS managed (predefined) policies for DevOps Guru (p. 35)

Policy best practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete DevOps Guru 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 using AWS managed policies** – To start using DevOps Guru quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get started using permissions with AWS managed policies in the *IAM User Guide*.

- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant least privilege in the *IAM User Guide*.
- **Enable MFA for sensitive operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.

Using the DevOps Guru console

To access the Amazon DevOps Guru console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the DevOps Guru 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 (IAM 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 you're trying to perform.

To ensure that users and roles can still use the DevOps Guru console, also attach the DevOps Guru `AmazonDevOpsGuruReadOnlyAccess` or `AmazonDevOpsGuruFullAccess` 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 (predefined) policies for DevOps Guru

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. These AWS-managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see AWS Managed Policies in the *IAM User Guide*.

To create and manage DevOps Guru service roles, you must also attach the AWS-managed policy named `IAMFullAccess`.

You can also create your own custom IAM policies to allow permissions forDevOps Guru actions and resources. You can attach these custom policies to the IAM users or groups that require those permissions.

The following AWS-managed policies, which you can attach to users in your account, are specific to DevOps Guru.

Topics

- AmazonDevOpsGuruFullAccess (p. 35)
- AmazonDevOpsGuruReadOnlyAccess (p. 36)

AmazonDevOpsGuruFullAccess

`AmazonDevOpsGuruFullAccess` – Provides full access to DevOps Guru, including permissions to create Amazon SNS topics, access Amazon CloudWatch metrics, and access AWS CloudFormation stacks. Apply this only to administrative-level users to whom you want to grant full control over DevOps Guru.

The `AmazonDevOpsGuruFullAccess` policy contains the following statement.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AmazonDevOpsGuruFullAccess",
 "Effect": "Allow",
 "Action": [
         "devops-guru:*"
       ],
  "Resource": "*"
    },
    {
      "Sid": "CloudFormationListStacksAccess",
      "Effect": "Allow",
 "Action": [
   "cloudformation:DescribeStacks",
   "cloudformation:ListStacks"
       ],
      "Resource": "*"
   },
   {
```

```
      "Sid": "CloudWatchGetMetricDataAccess",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricData"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SnsListTopicsAccess",
      "Effect": "Allow",
      "Action": [
        "sns:ListTopics"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SnsTopicOperations",
      "Effect": "Allow",
      "Action": [
        "sns:CreateTopic",
        "sns:GetTopicAttributes",
        "sns:SetTopicAttributes",
        "sns:Publish"
    ],
    "Resource": "arn:aws:sns:*:*:DevOps-Guru-*"
    },
    {
      "Sid": "DevOpsGuruSlrCreation",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/devops-guru.amazonaws.com/
AWSServiceRoleForDevOpsGuru",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "devops-guru.amazonaws.com"
        }
      }
    },
    {
      "Sid": "DevOpsGuruSlrDeletion",
      "Effect": "Allow",
      "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
      ],
      "Resource": "arn:aws:iam::*:role/aws-service-role/devops-guru.amazonaws.com/
AWSServiceRoleForDevOpsGuru",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "devops-guru.amazonaws.com"
        }
      }
    }
  ]
}
```

AmazonDevOpsGuruReadOnlyAccess

`AmazonDevOpsGuruReadOnlyAccess` – Grants read-only access to DevOps Guru and related resources in other AWS services. Apply this policy to users to whom you want to grant the ability to view insights, but not to make any updates to DevOps Guru's analysis coverage boundary, Amazon SNS topics, or Systems Manager OpsCenter integration.

The `AmazonDevOpsGuruReadOnlyAccess` policy contains the following statement.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AmazonDevOpsGuruReadOnlyAccess",
      "Effect": "Allow",
      "Action": [
        "devops-guru:DescribeAccountHealth",
        "devops-guru:DescribeAccountOverview",
        "devops-guru:DescribeAnomaly",
        "devops-guru:DescribeFeedback",
        "devops-guru:DescribeInsight",
        "devops-guru:DescribeResourceCollectionHealth",
        "devops-guru:DescribeServiceIntegration",
        "devops-guru:GetCostEstimation",
        "devops-guru:GetResourceCollection",
        "devops-guru:ListAnomaliesForInsight",
        "devops-guru:ListEvents",
        "devops-guru:ListInsights",
        "devops-guru:ListNotificationChannels",
        "devops-guru:ListRecommendations",
        "devops-guru:SearchInsights",
        "devops-guru:StartCostEstimation"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CloudFormationListStacksAccess",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:ListStacks"
      ],
      "Resource": "*"
    },
    {
   "Effect": "Allow",
   "Action": [
   "iam:GetRole"
   ],
   "Resource": "arn:aws:iam::*:role/aws-service-role/devops-guru.amazonaws.com/
AWSServiceRoleForDevOpsGuru"
    },
    {
      "Sid": "CloudWatchGetMetricDataAccess",
      "Effect": "Allow",
      "Action": [
          "cloudwatch:GetMetricData"
        ],
        "Resource": "*"
      }
  ]
}
```

Using service-linked roles for DevOps Guru

Amazon DevOps Guru uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to DevOps Guru. Service-linked roles are predefined by DevOps Guru and include all the permissions that the service requires to call AWS CloudTrail, Amazon CloudWatch, AWS CodeDeploy, and AWS X-Ray, on your behalf.

A service-linked role makes setting up DevOps Guru easier because you don't have to manually add the necessary permissions. DevOps Guru defines the permissions of its service-linked roles, and unless

defined otherwise, only DevOps Guru can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting its related resources. This protects your DevOps Guru resources because you can't inadvertently remove permission to access the resources.

Service-linked role permissions for DevOps Guru

DevOps Guru uses the service-linked role named `AmazonDevOpsGuruServiceRolePolicy`. This is a managed IAM policy with scoped permissions that DevOps Guru needs to run in your account.

The `AmazonDevOpsGuruServiceRolePolicy` service-linked role trusts the following service to assume the role:

- `devops-guru.amazonaws.com`

The role permissions policy allows DevOps Guru to complete the following actions on the specified resources.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "autoscaling:DescribeAutoScalingGroups",
      "cloudtrail:LookupEvents",
      "cloudwatch:GetMetricData",
      "cloudwatch:ListMetrics",
      "cloudwatch:DescribeAnomalyDetectors",
      "cloudwatch:DescribeAlarms",
      "cloudwatch:ListDashboards",
      "cloudwatch:GetDashboard",
      "cloudformation:GetTemplate",
      "cloudformation:ListStacks",
      "cloudformation:ListStackResources",
      "cloudformation:DescribeStacks",
      "cloudformation:ListImports",
      "codedeploy:BatchGetDeployments",
      "codedeploy:GetDeploymentGroup",
      "codedeploy:ListDeployments",
      "config:DescribeConfigurationRecorderStatus",
      "config:GetResourceConfigHistory",
      "events:ListRuleNamesByTarget",
      "xray:GetServiceGraph"
    ],
      "Resource": "*"
  },
  {
    "Sid": "AllowPutTargetsOnASpecificRule",
    "Effect": "Allow",
    "Action": [
      "events:PutTargets",
      "events:PutRule"
    ],
    "Resource": "arn:aws:events:*:*:rule/DevOps-Guru-managed-*"
  },
  {
    "Sid": "AllowCreateOpsItem",
    "Effect": "Allow",
    "Action": [
```

```
      "ssm:CreateOpsItem"
    ],
    "Resource": "*"
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/DevOps-GuruInsightSsmOpsItemRelated": "true"
      }
    }
  },
  {
    "Sid": "AllowAddTagsToOpsItem",
    "Effect": "Allow",
    "Action": [
        "ssm:AddTagsToResource"
    ],
    "Resource": "arn:aws:ssm:*:*:opsitem/*"
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/DevOps-GuruInsightSsmOpsItemRelated": "true"
      }
    }
  },
  {
    "Sid": "AllowAccessOpsItem",
    "Effect": "Allow",
    "Action": [
      "ssm:GetOpsItem",
      "ssm:UpdateOpsItem"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/DevOps-GuruInsightSsmOpsItemRelated": "true"
        }
      }
    }
  ]
}
```

Creating a service-linked role for DevOps Guru

You don't need to manually create a service-linked role. When you create an insight in the AWS Management Console, the AWS CLI, or the AWS API, DevOps Guru creates the service-linked role for you.

> **Important**
> This service-linked role can appear in your account if you completed an action in another service that uses the features supported by this role; for example, it can appear if you added DevOps Guru to a repository from AWS CodeCommit.

Editing a service-linked role for DevOps Guru

DevOps Guru does not allow you to edit the `AmazonDevOpsGuruServiceRolePolicy` service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for DevOps Guru

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must disassociate from all repositories before you can manually delete it.

Note
If the DevOps Guru service is using the role when you try to delete the resources, the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the `AmazonDevOpsGuruServiceRolePolicy` service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Amazon DevOps Guru permissions reference

You can use AWS-wide condition keys in your DevOps Guru policies to express conditions. For a list, see IAM JSON Policy Elements Reference in the *IAM User Guide*.

You specify the actions in the policy's `Action` field. To specify an action, use the `devops-guru:` prefix followed by the API operation name (for example, `devops-guru:SearchInsights` and `devops-guru:ListAnomalies`). To specify multiple actions in a single statement, separate them with commas (for example, `"Action": [ "devops-guru:SearchInsights", "devops-guru:ListAnomalies" ]`).

Using wildcard characters

You specify an Amazon Resource Name (ARN), with or without a wildcard character (*), as the resource value in the policy's `Resource` field. You can use a wildcard to specify multiple actions or resources. For example, `devops-guru:*` specifies all DevOps Guru actions and `devops-guru:List*` specifies all DevOps Guru actions that begin with the word `List`. The following example refers to all insights with a universally unique identifier (UUID) that begins with `12345`.

```
arn:aws:devops-guru:us-east-2:123456789012:insight:12345*
```

You can use the following table as a reference when you are setting up Authenticating with identities (p. 24) and writing permissions policies that you can attach to an IAM identity (identity-based policies).

DevOps Guru API operations and required permissions for actions

`AddNotificationChannel`

Action: `devops-guru:AddNotificationChannel`

Required to add a notification channel from DevOps Guru. A notification channel is used to notify you when DevOps Guru generates an insight that contains information about how to improve your operations.

Resource: `*`

`RemoveNotificationChannel`

`devops-guru:RemoveNotificationChannel`

Required to remove a notification channel from DevOps Guru. A notification channel is used to notify you when DevOps Guru generates an insight that contains information about how to improve your operations.

Resource: `*`

`ListNotificationChannels`

Action: `devops-guru:ListNotificationChannels`

Required to return a list of notification channels configured for DevOps Guru. Each notification channel is used to notify you when DevOps Guru generates an insight that contains information about how to improve your operations. The one notification type supported is Amazon Simple Notification Service.

Resource: `*`

`UpdateResourceCollectionFilter`

Action: `devops-guru:UpdateResourceCollectionFilter`

Required to update the list of AWS CloudFormation stacks that are used to specify which AWS resources in your account are analyzed by DevOps Guru. The analysis generates insights that include recommendations, operational metrics, and operational events that you can use to improve the performance of your operations. This method also creates the IAM roles required for you to use CodeGuru OpsAdvisor.

Resource: `*`

`GetResourceCollectionFilter`

Action: `devops-guru:GetResourceCollectionFilter`

Required to return the list of AWS CloudFormation stacks that are used to specify which AWS resources in your account are analyzed by DevOps Guru. The analysis generates insights that include recommendations, operational metrics, and operational events that you can use to improve the performance of your operations.

Resource: `*`

`ListInsights`

Action: `devops-guru:ListInsights`

Required to return a list of insights in your AWS account. You can specify which insights are returned by their start time, status (`ongoing` or `any`), and type (`reactive` or `predictive`).

Resource: `*`

`DescribeInsight`

Action: `devops-guru:DescribeInsight`

Required to return details about an insight that you specify using its ID.

Resource: `*`

`SearchInsights`

Action: `devops-guru:SearchInsights`

Required to return a list of insights in your AWS account. You can specify which insights are returned by their start time, filters, and type (`reactive` or `predictive`).

Resource: `*`

`ListAnomalies`

Action: `devops-guru:ListAnomalies`

Required to return a list of the anomalies that belong to an insight that you specify using its ID.

Resource: `*`

`DescribeAnomaly`

Action: `devops-guru:DescribeAnomaly`

Required to return details about an anomaly that you specify using its ID.

Resource: `*`

`ListEvents`

Action: `devops-guru:ListEvents`

Required to return a list of the events emitted by the resources that are evaluated by DevOps Guru. You can use filters to specify which events are returned.

Resource: `*`

`ListRecommendations`

Action: `devops-guru:ListRecommendations`

Required to return a list of a specified insight's recommendations. Each recommendation includes a list of metrics and a list of events that are related to the recommendations.

Resource: `*`

`DescribeAccountHealth`

Action: `devops-guru:DescribeAccountHealth`

Required to return the number of open reactive insights, the number of open predictive insights, and the number of metrics analyzed in your AWS account. Use these numbers to gauge the health of operations in your AWS account.

Resource: `*`

`DescribeAccountOverview`

Action: `devops-guru:DescribeAccountOverview`

Required to return the following that happened during a time range: the number of open reactive insights that were created, the number of open predictive insights that were created, and the mean time to recover (MTTR) for all reactive insights that were closed.

Resource: `*`

`DescribeResourceCollectionHealthOverview`

Action: `devops-guru:DescribeResourceCollectionHealthOverview`

Required to return the number of open predictive insights, open reactive insights, and mean time to recover (MTTR) for all insights for each AWS CloudFormation stack specified in DevOps Guru.

Resource: `*`

`DescribeIntegratedService`

Action: `devops-guru:DescribeIntegratedService`

Required to return the integration status of services that can be integrated with DevOps Guru. The one service that can be integrated with DevOps Guru is AWS Systems Manager, which can be used to create an OpsItem for each generated insight.

Resource: `*`

`UpdateIntegratedServiceConfig`

Action: `devops-guru:UpdateIntegratedServiceConfig`

Required to enable or disable integration with a service that can be integrated with DevOps Guru. The one service that can be integrated with DevOps Guru is Systems Manager, which can be used to create an OpsItem for each generated insight.

Resource: *

Permissions for cross account Amazon SNS topics

Use the information in this topic only if you want to configure Amazon DevOps Guru to deliver Amazon SNS topics owned by a different account than yours. DevOps Guru must have permissions to send notifications to an Amazon SNS topic. DevOps Guru adds the required policy on your behalf to send notifications using Amazon SNS topics in your AWS account.

Note
DevOps Guru currently only supports cross-account access in the same Region.

If you want to use an Amazon SNS topic from another account, you must attach the following policy to the existing Amazon SNS topic.

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
     "Action": "sns:Publish",
     "Effect": "Allow",
     "Resource": "arn:aws:sns:region-id:topic-owner-account-id:my-topic-name",
     "Principal": {
       "Service": "region-id.devops-guru.amazonaws.com"
     }
   },
   {
     "Action": "sns:Publish",
     "Effect": "Allow",
     "Resource": "arn:aws:sns:region:topic-owner-account-id:my-topic-name",
     "Principal": {
       "AWS": "arn:aws:iam::topic-sender-account-id:user/devops-guru-user-name"
      }
   }
 ]
}
```

For the `Resource` key, *`topic-owner-account-id`* is the account ID of the topic owner, *`topic-sender-account-id`* is the account ID of the user who set up DevOps Guru, and *`devops-guru-user-name`* is the individual IAM user. You must substitute appropriate values for *`region-id`* (for example, `us-west-2`) and *`my-topic-name`*.

Permissions for AWS KMS–encrypted Amazon SNS topics

The Amazon SNS topic you specify might be encrypted by AWS Key Management Service. To allow DevOps Guru to work with encrypted topics, you must first create a AWS KMS key and then add the following statement to the policy of the KMS key. For more information, see Encrypting messages published to Amazon SNS with AWS KMS, Key identifiers (KeyId) in the *AWS KMS User Guide*, and Data encryption in the *Amazon Simple Notification Service Developer Guide*.

```
{
    "Version": "2012-10-17",
    "Id": "your-kms-key-policy",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "region-id.devops-guru.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
      ]
}
```

Troubleshooting Amazon DevOps Guru identity and access

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

Topics

- I am not authorized to perform an action in DevOps Guru (p. 44)
- I am not authorized to perform iam:PassRole (p. 44)
- I want to view my access keys (p. 45)
- I'm an administrator and want to allow others to access DevOps Guru (p. 45)
- I want to allow people outside of my AWS account to access my DevOps Guru resources (p. 45)

I am not authorized to perform an action in DevOps Guru

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person who provided you with your user name and password.

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 does not have the fictional `awes:GetWidget` permissions.

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

In this case, Mateo asks his administrator to update his policies to allow him to access the *`my-example-widget`* resource using the `awes:GetWidget` action.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to DevOps Guru.

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 DevOps Guru. However, the action requires the service to have permissions 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 asks her administrator to update her policies to allow her to perform the `iam:PassRole` action.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

> **Important**
> Do not provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see Managing access keys in the *IAM User Guide*.

I'm an administrator and want to allow others to access DevOps Guru

To allow others to access DevOps Guru, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in DevOps Guru.

To get started right away, see Creating your first IAM delegated user and group in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my DevOps Guru 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 DevOps Guru supports these features, see How Amazon DevOps Guru works with IAM (p. 28).
- 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 How IAM roles differ from resource-based policies in the *IAM User Guide*.

Logging and monitoring DevOps Guru

Monitoring is an important part of maintaining the reliability, availability, and performance of DevOps Guru and your other AWS solutions. AWS provides the following monitoring tools to watch DevOps Guru, 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 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

- Monitoring DevOps Guru with Amazon CloudWatch (p. 46)
- Logging Amazon DevOps Guru API calls with AWS CloudTrail (p. 48)

Monitoring DevOps Guru with Amazon CloudWatch

You can monitor DevOps Guru using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These statistics are kept for 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing. You can also set alarms that watch for certain thresholds and send notifications or take actions when those thresholds are met. For more information, see the Amazon CloudWatch User Guide.

For DevOps Guru, you can track metrics for insights and metrics for your DevOps Guru usage. You might want to watch for a large number of created `Insights` to help you determine if your operational solutions are experiencing anomalous behavior. Or you might want to watch your DevOps Guru usage to help track your costs.

The DevOps Guru service reports the following metrics in the `AWS/DevOps-Guru` namespace.

Topics

- Insight metrics (p. 46)
- DevOps Guru usage metrics (p. 47)

Insight metrics

You can use CloudWatch to track a metric to show you how many insights are created in your AWS account. You can specify the `Type` dimension to track `proactive` or `reactive` insights. Do not specify a dimension if you want to track all insights.

Metrics

Metric	Description
`Insight`	The number of insights created in an AWS account. Valid dimensions: `Type` Valid statistics: Sample count, Sum Units: Count

The following dimension is supported for the DevOps Guru `Insight` metric.

Dimensions

Dimension	Description
`Type`	This is the type of the insight. Do not specify a dimension for the `Insights` metric if you want to track all insights. Valid values are: `proactive`, `reactive`.

DevOps Guru usage metrics

You can use CloudWatch to track your Amazon DevOps Guru usage.

Metrics

Metric	Description
`CallCount`	The number of calls made by one of the following DevOps Guru methods. • `ListInsights` • `ListAnomaliesForInsight` • `ListRecommendations` • `ListEvents` • `SearchInsights` • `DescribeInsight` • `DescribeAnomaly` Valid dimensions: `Service`, `Class`, `Type`, `Resource` Valid statistics: Sample count, Sum Units: Count

The following dimensions are supported for the DevOps Guru usage metrics.

Dimensions

Dimension	Description
`Service`	This is the name of the AWS service that contains the resource. For example, for DevOps Guru, this value is `DevOps-Guru`.

Dimension	Description
`Class`	This is the class of the resource that is tracked. DevOps Guru uses this dimension with the value `None`.
`Type`	This is type of the resource that is tracked. DevOps Guru uses this dimension with the value `API`.
`Resource`	This is the name of the DevOps Guru operation. Valid values are: `ListInsights`, `ListAnomaliesForInsight`, `ListRecommendations`, `ListEvents`, `SearchInsights`, `DescribeInsight`, `DescribeAnomaly`.

Logging Amazon DevOps Guru API calls with AWS CloudTrail

Amazon DevOps Guru is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in DevOps Guru. CloudTrail captures API calls for DevOps Guru as events. The calls captured include calls from the DevOps Guru console and code calls to the DevOps Guru API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for DevOps Guru. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to DevOps Guru, the IP address from which the request was made, who made the request, when it was made, and additional details.

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

DevOps Guru information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in DevOps Guru, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail Event history.

For an ongoing record of events in your AWS account, including events for DevOps Guru, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, 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 and delivers the log files to the Amazon S3 bucket that you specify. Additionally, 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:

- Overview for creating a trail
- 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

DevOps Guru supports logging all of its actions as events in CloudTrail log files. For more information, see Actions in the *DevOps Guru API Reference*.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) 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.

Understanding DevOps Guru log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. 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 shows a CloudTrail log entry that demonstrates the `UpdateResourceCollection` action.


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AAAAAAAAAEXAMPLE:TestSession",
    "arn": "arn:aws:sts::123456789012:assumed-role/TestRole/TestSession",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::123456789012:role/TestRole",
        "accountId": "123456789012",
        "userName": "sample-user-name"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-12-03T15:29:51Z"
      }
    }
  },
  "eventTime": "2020-12-01T16:14:31Z",
  "eventSource": "devops-guru.amazonaws.com",
  "eventName": "UpdateResourceCollection",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "sample-ip-address",
  "userAgent": "aws-internal/3 aws-sdk-java/1.11.901
 Linux/4.9.217-0.3.ac.206.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.275-b01
 java/1.8.0_275 vendor/Oracle_Corporation",
  "requestParameters": {
    "Action": "REMOVE",
    "ResourceCollection": {
      "CloudFormation": {
        "StackNames": [
          "*"
        ]
      }
    }
  },
  "responseElements": null,
  "requestID": " cb8c167e-EXAMPLE ",
  "eventID": " e3c6f4ce-EXAMPLE ",
  "readOnly": false,
  "eventType": "AwsApiCall",
```

```
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "123456789012"
}
```

DevOps Guru and interface VPC endpoints (AWS PrivateLink)

You can use VPC endpoints when you call Amazon DevOps Guru APIs. When you use VPC endpoints, your API calls are more secure because they are contained within your VPC and do not access the internet. For more information, see Actions in the *Amazon DevOps Guru API Reference*.

You establish a private connection between your VPC and DevOps Guru by creating an *interface VPC endpoint*. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to privately access DevOps Guru APIs without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with DevOps Guru APIs. Traffic between your VPC and DevOps Guru does not leave the Amazon network.

Each interface endpoint is represented by one or more Elastic Network Interfaces in your subnets.

For more information, see Interface VPC endpoints (AWS PrivateLink) in the *Amazon VPC User Guide*.

Considerations for DevOps Guru VPC endpoints

Before you set up an interface VPC endpoint for DevOps Guru, ensure that you review Interface endpoint properties and limitations in the *Amazon VPC User Guide*.

DevOps Guru supports making calls to all of its API actions from your VPC.

Creating an interface VPC endpoint for DevOps Guru

You can create a VPC endpoint for the DevOps Guru service using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Creating an interface endpoint in the *Amazon VPC User Guide*.

Create a VPC endpoint for DevOps Guru using the following service name:

- com.amazonaws.*region*.devops-guru

If you enable private DNS for the endpoint, you can make API requests to DevOps Guru using its default DNS name for the Region, for example, `devops-guru.us-east-1.amazonaws.com`.

For more information, see Accessing a service through an interface endpoint in the *Amazon VPC User Guide*.

Creating a VPC endpoint policy for DevOps Guru

You can attach an endpoint policy to your VPC endpoint that controls access to DevOps Guru. The policy specifies the following information:

- The principal that can perform actions.
- The actions that can be performed.

- The resources on which actions can be performed.

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

Example: VPC endpoint policy for DevOps Guru actions

The following is an example of an endpoint policy for DevOps Guru. When attached to an endpoint, this policy grants access to the listed DevOps Guru actions for all principals on all resources.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "devops-guru:AddNotificationChannel",
            "devops-guru:ListInsights",
            "devops-guru:ListRecommendations"
         ],
         "Resource":"*"
      }
   ]
}
```

Infrastructure security in DevOps Guru

As a managed service, Amazon DevOps Guru 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 DevOps Guru through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend 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.

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.

Resilience in Amazon DevOps Guru

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. DevOps Guru operates in multiple Availability Zones and stores artifact data and metadata in Amazon S3 and Amazon DynamoDB. Your encrypted data is redundantly stored across multiple facilities and multiple devices in each facility, making it highly available and highly durable.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Quotas for Amazon DevOps Guru

The following table lists the current quota in Amazon DevOps Guru. This quota is for each supported AWS Region for each AWS account.

Notifications

Maximum number of Amazon Simple Notification Service topics you can specify at once	2

AWS CloudFormation stacks

Maximum number of AWS CloudFormation stacks you can specify	1000

Amazon DevOps Guru document history

The following table describes the documentation for this release of DevOps Guru.

- **API version: latest**
- **Latest documentation update:** September 10, 2021

update-history-change	update-history-description	update-history-date
General availability release (p. 53)	Amazon DevOps Guru is now generally available (GA).	May 4, 2021
New topic (p. 53)	You can now generate a monthly cost estimate for DevOps Guru to analyze your resources. For more information, see Estimate your Amazon DevOps Guru costs.	April 27, 2021
VPC Endpoint support (p. 53)	You can now use VPC endpoints to improve the security of your resource analysis and insight generation. For more information, see DevOps Guru and interface VPC endpoints (AWS PrivateLink).	April 15, 2021
New topic (p. 53)	A new topic about how to monitor DevOps Guru with Amazon CloudWatch was added. For more information, see Monitoring DevOps Guru with Amazon CloudWatch.	December 11, 2020
Preview release (p. 53)	This is the preview release of the *Amazon DevOps Guru User Guide*.	December 1, 2020

AWS glossary

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