aws

User Guide

Amazon DevOps Guru

Amazon DevOps Guru: User Guide

Table of Contents

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?

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, log groups, 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, log groups, 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 for more detail about the three high-level workflow steps. See Detailed DevOps Guru workflow to learn about the more detailed DevOps Guru workflow, including how it interacts with other AWS services.

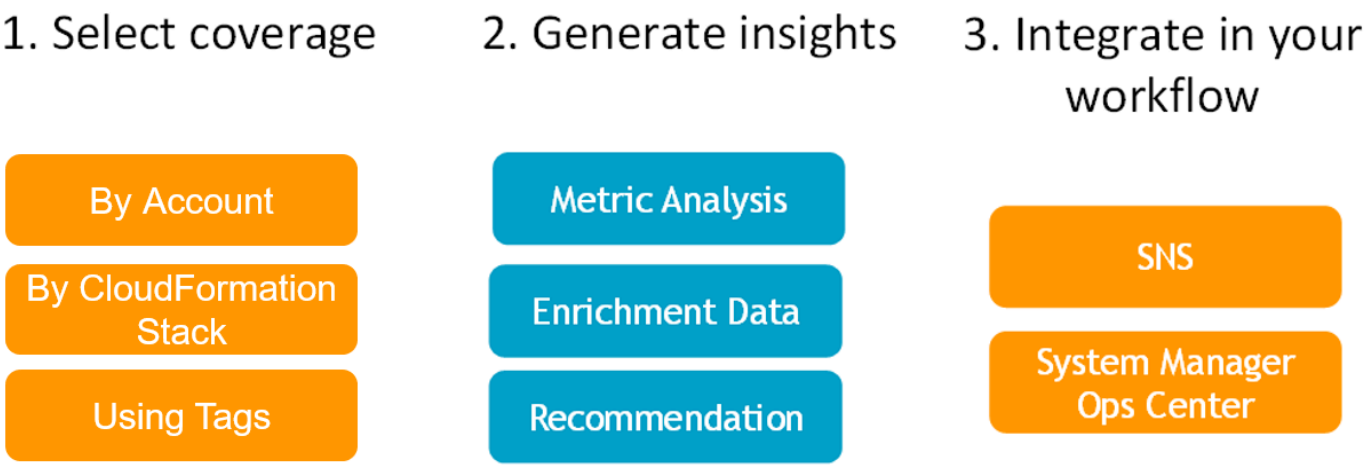
Topics

High level DevOps Guru workflow

The Amazon DevOps Guru workflow can be broken down into three high level steps.

1. Specify DevOps Guru coverage by telling it which AWS resources in your AWS account you want it to analyze.
2. DevOps Guru starts analyzing Amazon CloudWatch metrics, AWS CloudTrail, and other operational data to identify problems that you can fix to improve your operations.
3. DevOps Guru makes sure that you know about insights and important information by sending you a notification for each important DevOps Guru event.

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.



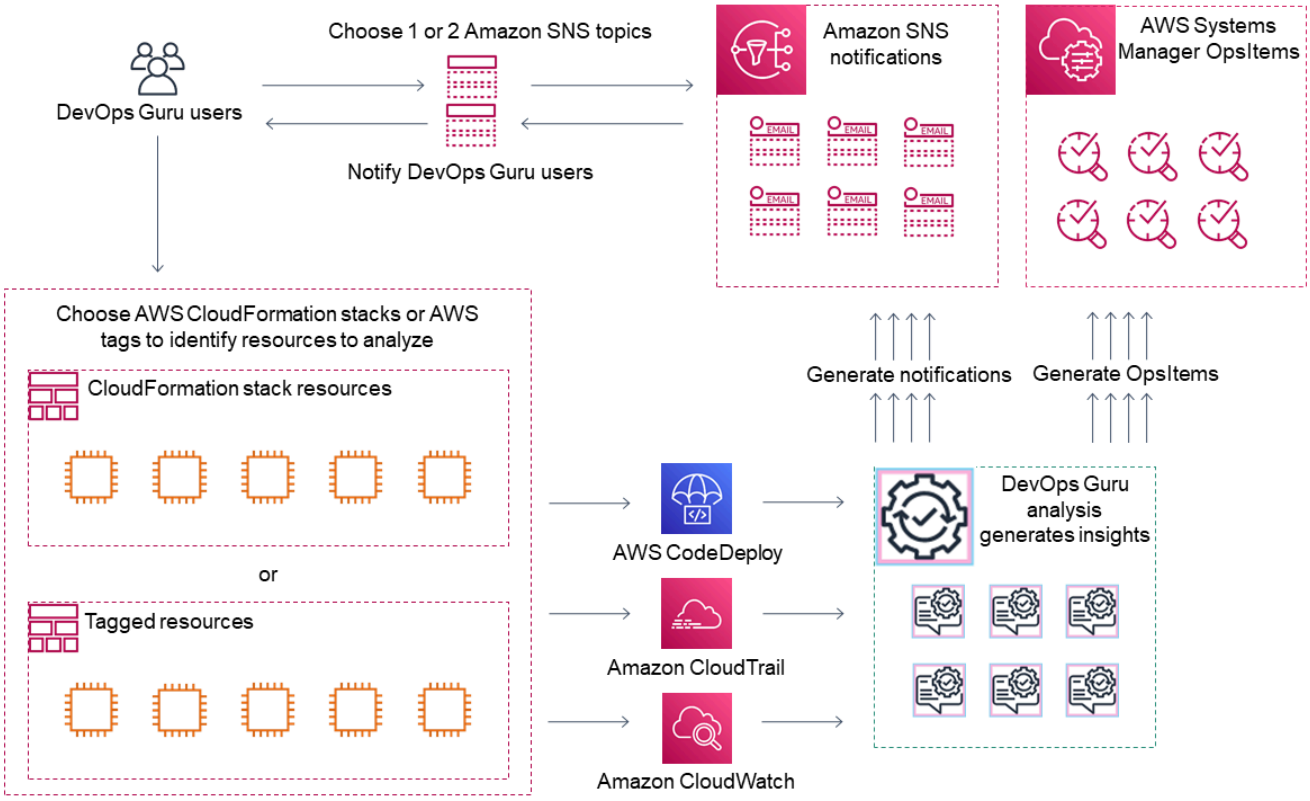
1. In the first step, you choose your coverage by specifying which AWS resources in your AWS account are analyzed. DevOps Guru can cover, or analyze, all the resources in an AWS account, or you can use AWS CloudFormation stacks or AWS tags to specify a subset of the resources in your account to analyze. 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.
- Log anomalies from Amazon CloudWatch log groups. If you enable log anomaly detection, DevOps Guru displays related log anomalies when an issue occurs.
- 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 workflow 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 Updating your notifications in DevOps Guru.
   - 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 Updating AWS Systems Manager integration in DevOps Guru.

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 or using AWS tags. If no stacks or tags are chosen, then DevOps Guru coverage analyzes all AWS resources in your account. For more information, see Defining applications using AWS resources and Determine coverage for DevOps Guru.

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, log groups, 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, a list of related log groups, 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.
2. **Set up** your AWS account, the AWS CLI, and an administrative user by following the steps in Setting up Amazon DevOps Guru.
3. **Use** DevOps Guru, following the instructions in Getting started with DevOps Guru.

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 Updating your AWS analysis coverage in DevOps Guru and choose **None** 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
- Insight

- Metrics and operational events
- Log groups and log anomalies
- Recommendations

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.

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.

Log groups and log anomalies

When you enable log anomaly detection, relevant log groups are displayed on DevOps Guru insight pages in the DevOps Guru console. A log group lets you know about critical diagnostic information about how a resource is performing and being accessed.

A log anomaly represents a cluster of similar anomalous log events found within a log group. Examples of anomalous log events that may be displayed in DevOps Guru include keyword anomalies, format anomalies, HTTP code anomalies, and more.

You can use log anomalies to diagnose the root cause of an operational issue. DevOps Guru also references log lines in insight recommendations to provide more context for recommended solutions.

> **Note**
>
> DevOps Guru works with Amazon CloudWatch to enable log anomaly detection. When you enable log anomaly detection, DevOps Guru adds tags to your CloudWatch log groups. When you turn off log anomaly detection, DevOps Guru removes tags from your CloudWatch log groups.
> In addition, administrators should ensure that only users with permissions to view CloudWatch logs have permissions to view anomalous CloudWatch logs. We recommend that you use IAM policies to allow or deny access to the `ListAnomalousLogs` operation. For more information, see Identity and Access Management for DevOps Guru.

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.

- A list of log groups that are related to the anomalous behavior associated with the insight. Each log group contains a sample log message, information about the kinds of log anomalies reported, the times the log anomalies occurred, and a link to view the log lines on CloudWatch.

DevOps Guru coverage

DevOps Guru addresses and creates insights for a number of different AWS services. For each service that DevOps Guru creates insights for, DevOps Guru displays a variety of analyzed metrics and generated insights.

Example use case for reactive insights:

Service Name	Use Case	Examples	Metrics
AWS Lambda	Detect latency or duration anomalies for Lambda functions caused by various root causes like cold starts, increased requests, downstrea m throttling, or code deployments. Recommend ways to quickly mitigate.	Code deployment: Amazon API Gateway latency is affected by an increase in Lambda latency after a recent Lambda code deployment. Downstream throttlin g: the operator reduced capacity on read units for DynamoDB, causing increased retries. This results in throttlin g. Cold start: the Lambda function is under-provisioned, so Lambda takes longer when requests are made.	Duration Throttles

Example use case for proactive insights:

Service Name	Use Case	Metrics
Amazon DynamoDB	**The DynamoDB table read consumed capacity is at risk of reaching table limit.** Recommended action: if you are using provisioned capacity mode, use auto scaling to actively manage throughpu t capacity for tables or purchase reserved capacity in advance for tables. Switch to on-demand capacity mode to pay per read request, paying only for what is used. Detec tion time: 6 days	ConsumedReadCapacityUnits

Service coverage list

For some services, DevOps Guru creates reactive insights. 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.

For some services, DevOps Guru creates proactive insights. 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.

DevOps Guru creates reactive insights for services such as the following:

- Amazon API Gateway
- Amazon CloudFront
- Amazon DynamoDB
- Amazon EC2

> **Note**
>
> DevOps Guru monitoring is at an Auto Scaling group level, and not at a single instance level.

- Amazon ECS
- Amazon EKS
- AWS Elastic Beanstalk
- Elastic Load Balancing
- Amazon Kinesis
- AWS Lambda
- Amazon OpenSearch Service
- Amazon RDS
- Amazon Redshift
- Amazon Route 53
- Amazon S3
- Amazon SageMaker
- AWS Step Functions
- Amazon SNS
- Amazon SQS
- Amazon SWF
- Amazon VPC

DevOps Guru creates proactive insights for services such as the following:

- Amazon DynamoDB
- Amazon Kinesis
- AWS Lambda
- Amazon RDS
- Amazon SQS

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.

Optionally, you can use Quick Setup, a capability of AWS Systems Manager, to set up DevOps Guru and quickly configure its options. You can use Quick Setup to set up DevOps Guru for a standalone account or an organization. To use Quick Setup in Systems Manager to set up DevOps Guru for an organization, 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 Configure 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
- Step 2 – Determine coverage for DevOps Guru
- Step 3 – Identify your Amazon SNS notifications topic

Step 1 – Sign up for AWS

Sign up for an AWS account

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.

   When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the *IAM User Guide*.

Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

Sign in as the user with administrative access

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

  For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the *AWS Sign-In User Guide*.

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.
2. Assign users to a group, and then assign single sign-on access to the group.

   For instructions, see Add groups in the *AWS IAM Identity Center User Guide*.

Step 2 – Determine coverage for DevOps Guru

Your boundary coverage determines the AWS resources that are analyzed by Amazon DevOps Guru for anomalous behavior. We recommend that you group your resources into your operational applications. All the resources in your resource boundary should comprise one or more of your applications. If you have one operational solution, then your coverage boundary should include all of its resources. If you have multiple applications, choose the resources that make up each solution and group them together using AWS CloudFormation stacks or AWS tags. All of the combined resources you specify, whether they define one or more applications, are analyzed by DevOps Guru and make up its coverage boundary.

Use one of the following methods to specify the resources in your operational solutions.

- Choose to have your AWS Region and account define your coverage boundary. With this option, DevOps Guru analyzes all resources in your account and Region. This is a good option to choose if you use your account for only one application.
- Use AWS CloudFormation stacks to define the resources in your operational application. AWS CloudFormation templates define and generate your resources for you. Specify the stacks that create your application resources when you configure DevOps Guru. You can update your stacks at any time. All of the resources in the stacks that you choose define your boundary coverage. For more information, see Using AWS CloudFormation stacks to identify resources in your DevOps Guru applications.
- Use AWS tags to specify AWS resources in your applications. DevOps Guru analyzes only the resources that contain the tags you choose. Those resources make up your boundary.

  An AWS tag consists of a tag *key* and a tag *value*. You can specify one tag *key* and you can specify one or more *values* with that *key*. Use one *value* for all the resources in one of your applications. If you have multiple applications, then use a tag with the same *key* for all of them, and group the resources into your applications using the tags' *values*. All of the resources with the tags that you choose make up the coverage boundary for DevOps Guru. For more information, see Using tags to identify resources in your DevOps Guru applications.

If your boundary coverage includes resources that make up more than one application, you can use tags to filter your insights by to view them by one application at a time. For more information, see Step 4 in Viewing DevOps Guru insights.

For more information, see Defining applications using AWS resources. 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*.

Permissions added to your Amazon SNS topic

An Amazon SNS topic is a resource that contains an AWS Identity and Access Management (IAM) resource policy. When you specify a topic here, DevOps Guru appends the following permissions to its resource policy.

```
{
    "Sid": "DevOpsGuru-added-SNS-topic-permissions",
    "Effect": "Allow",
    "Principal": {
        "Service": "region-id.devops-guru.amazonaws.com"
    },
    "Action": "sns:Publish",
    "Resource": "arn:aws:sns:region-id:topic-owner-account-id:my-topic-name",
    "Condition" : {
      "StringEquals" : {
        "AWS:SourceArn": "arn:aws:devops-guru:region-id:topic-owner-account-
id:channel/devops-guru-channel-id",
        "AWS:SourceAccount": "topic-owner-account-id"
    }
  }
}
```

These permissions are required for DevOps Guru to publish notifications using a topic. If you prefer to not have these permissions on the topic, you can safely remove them and the topic will continue to work as it did before you chose it. However, if these appended permissions are removed, DevOps Guru cannot use the topic to generate notifications.

Estimating 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 few resources, it can take 1 to 2 hours to complete. When you specify a lot of resources, it can take up to 4 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 the **Create IAM role for DevOps Guru** popup window that appears, choose **Agree to create IAM role**. This allows DevOps Guru to create an IAM service-linked role for you when you choose to begin the cost estimate analysis or begin using DevOps Guru. That way, DevOps Guru has the permissions needed to create the cost estimate. If you have already enabled DevOps Guru, the role has already been created and 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 in the current Region**.
  2. In **Choose a CloudFormation stack**, choose the name of an AWS CloudFormation stack in your AWS account. You can also enter the name of a stack to find it quickly. 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 resources with a tag, do the following.
  1. Choose **Tags on AWS resources in the current Region**
  2. In **Tag key** choose your tag's *key*
  3. In **Tag value** choose **(all values)** or choose one *value*.
- If you want to estimate the cost for DevOps Guru to analyze the resource in your AWS account and Region, choose **AWS account in the current Region**.

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

Before you get started, prepare by running through the steps in Setting up Amazon DevOps Guru.

Step 2: Enable DevOps Guru

To configure Amazon DevOps Guru to use for the first time, you must choose how you want to set up DevOps Guru. You can either monitor applications across your organization or monitor applications in your current account.

You can either monitor your applications across your organization or enable DevOps Guru for exclusively the current account. The following procedures outline different ways to set up DevOps Guru based on your needs.

Monitor accounts across your organization

If you choose to monitor applications across your organization, log into your organization management account. You can optionally set up an organization member account as a **delegated administrator**. You can only have one delegated administrator at a time and can modify the administrator settings later. Both the management account and the delegated administrator account that you set up have access to all insights across all accounts in your organization.

You can either add cross account support for your organization using the Console, or you can do so by using the AWS CLI.

Onboard with the DevOps Guru Console

You can use the Console to add support for accounts across your organization.

Use the Console to enable DevOps Guru to view aggregated insights

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Choose **Monitor applications across your organizations** as the setup type.
3. Choose which account you'd like to use as your delegated administrator. Then, choose **Register delegated administrator**. This provides access to a consolidated view for any account that has DevOps Guru enabled. The delegated administrator has a consolidated view of all DevOps Guru insights and metrics across your organization. You can enable other accounts with SSM quick setup or AWS CloudFormation stack sets. To learn more about quick setup, see Configure DevOps Guru with Quick Setup. To learn more about setting up with stack sets, see Working with stacks in the *AWS CloudFormation User Guide*, and Step 2 – Determine coverage for DevOps Guru. and Using AWS CloudFormation stacks to identify resources in your DevOps Guru applications.

Onboard with the AWS CLI

You can use the AWS CLI to enable DevOps Guru to view aggregated insights. Run the following commands.

```
aws iam create-service-linked-role --aws-service-name devops-guru.amazonaws.com --
description "My service-linked role to support DevOps Guru"

aws organizations enable-aws-service-access --service-principal devops-
guru.amazonaws.com

aws organizations register-delegated-administrator --account-id >ACCOUNT_ID< --service-
principal devops-guru.amazonaws.com
```

The following table describes the commands.

Command	Description
`create-service-linked-role`	Gives DevOps Guru permission to gather information about your organization. Don't proceed if this step is not successful.
	Onboards your organization to DevOps Guru.

Command	Description
`enable-aws-service-access`	
`register-delegated-administrator`	Gives access to the member account to view insights.

Monitor your current account

If you choose to monitor applications in your current AWS account, choose which AWS resources in your account and Region are 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 to monitor applications in your current AWS account

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Choose **Monitor applications in the current AWS account** as the setup type.
3. 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**: You choose your analysis boundary later. For more information, see Determine coverage for DevOps Guru and Updating your AWS analysis coverage in DevOps Guru.

   DevOps Guru can analyze any resource that is associated with the AWS account it supports. For more information about the supported services and resources, see Amazon DevOps Guru pricing.
4. 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 **Generate a new SNS topic using email**. Then, from **Specify the email address**, enter the email address you want to receive notifications. To enter in additional email addresses, choose **Add new email**.
- Choose **Use an existing SNS topic**. Then, from **Choose a topic in your AWS account**, choose the topic you want to use.
- Choose **Use an existing SNS topic ARN to specify an existing topic from another 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 Amazon SNS topics.

5. Choose **Enable**.

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.

Step 3: Specify your DevOps Guru resource coverage

If you chose to specify AWS resources later when you enabled DevOps Guru, you need to choose the AWS CloudFormation stacks in your AWS account that 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 Determine coverage for DevOps Guru. and Using AWS CloudFormation stacks to identify resources in your DevOps Guru applications.

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

Specify DevOps Guru resource coverage

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

   - Choose **All account resources** if you want DevOps Guru to analyze all supported resources in your AWS account and Region. If you choose this option, your AWS account is your resource analysis coverage boundary. All resources in each stack in your account are grouped into their own application. Any remaining resources that are not in a stack are grouped into their own application.
   - Choose **CloudFormation stacks** if you want DevOps Guru to analyze the resources that are in stacks you choose, then choose one of the following options.
     - **All resources** – All resources that are in stacks in your account are analyzed. Resources in each stack are grouped into their own application. Any resources in your account that are not in a stack are not analyzed.
     - **Select stacks** – Select the stacks that you want DevOps Guru to analyze. The resources in each stack you select are grouped into their own application. You can enter the name of a stack in **Find stacks** to quickly locate a specific stack. You can select up to 1,000 stacks.

     For more information, see Using AWS CloudFormation stacks to identify resources in your DevOps Guru applications.
   - Choose **Tags** if you want DevOps Guru to analyze all resources that contain the tags you choose. Choose a *key*, then choose one of the following options.
     - **All account resources** –Analyze all AWS resources in the current Region and account. Resources with the selected tag key are grouped by tag value, if any exist. Resources without this tag key are grouped and analyzed separately.
     - **Choose specific tag values** – All resources that contain a tag with the *key* you chose are analyzed. DevOps Guru groups your resources into applications by your tag's *values*.

     The tag's *key* must begin with the prefix `devops-guru-`. This prefix isn't case-sensitive. For example, a valid *key* is `DevOps-Guru-Production-Applications`. For more information, see Using tags to identify resources in your DevOps Guru applications.
   - Choose **None** 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**.

Enabling AWS services for DevOps Guru analysis

Amazon DevOps Guru can analyze the performance of any AWS resource that it supports. When it finds anomalous behavior, it generates an insight with details about the behavior and how to address it. For more information about the supported services and resources, see Amazon DevOps Guru pricing.

DevOps Guru uses Amazon CloudWatch metrics, AWS CloudTrail events, and more to help analyze resources. Most of the resources it supports generate the metrics required for DevOps Guru analysis automatically. However, a few AWS services require extra action to generate the required metrics. For some services, enabling these metrics provides additional analysis to existing DevOps Guru coverage. For others, analysis is not possible until you enable these metrics. For more information, see Determine coverage for DevOps Guru and Updating your AWS analysis coverage in DevOps Guru.

Services that require action for DevOps Guru analysis

- Amazon Elastic Container Service – To generate additional metrics that improve DevOps Guru coverage of its resources, follow the steps in Setting up container insights on Amazon ECS. Doing this might incur Amazon CloudWatch charges.
- Amazon Elastic Kubernetes Service – To generate metrics for DevOps Guru to analyze, follow the steps in Setting up container insights on Amazon EKS and Kubernetes. DevOps Guru doesn't analyze any Amazon EKS resources until generation of these metrics is set up. Doing this might incur Amazon CloudWatch charges.
- Amazon Simple Storage Service – To generate metrics for DevOps Guru to analyze, you must enable request metrics. Follow the steps in Creating a CloudWatch metrics configuration for all the objects in your bucket. DevOps Guru doesn’t analyze any Amazon S3 resources until generation of these metrics is set up. Doing this might incur CloudWatch and Amazon S3 charges.

For more information, see Amazon CloudWatch pricing.

Working with insights in DevOps Guru

Amazon DevOps Guru generates an *insight* 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 recommendations for you to take to mitigate the issue. It also contains a list of the metrics, a list of log groups, 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

Viewing 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 a list of 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 for which you are looking.

- Choose **Filter insights**, then choose an option to specify a filter. You can add a combination of status, severity, resource, and tag filters. Use an AWS tag filter to view insights generated by only resources with specific tags. To learn more, see Using tags to identify resources in your DevOps Guru applications.

> **Note**
>
> DevOps Guru can analyze the following resources, but can't filter their insights using tags.
>
> - Amazon API Gateway paths and routes
> - Amazon DynamoDB streams
> - Amazon EC2 Auto Scaling group instances
> - AWS Elastic Beanstalk environments
> - Amazon Redshift nodes

- 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 to address operational problems detected by DevOps Guru.

To view an insight, follow the steps in Viewing insights 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.

If an insight is grouped at the *stack level*, then you can 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. If an insight is grouped at the *account level*, then the number is zero or does not appear.

For more information, see Understanding how anomalous behaviors are grouped into insights.

Insight name

The name of an insight depends on whether it is grouped at the *stack level* or the *account level*.

- *Stack level* insight names include the name of the stack that contains the resource with its anomalous behavior.
- *Account level* insight names do not include a stack name.

For more information, see Understanding how anomalous behaviors are grouped into insights.

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*.

Log groups

When you enable log anomaly detection, DevOps Guru tags your CloudWatch log groups so you can view log groups related to your insights. In the **Log groups** section on the insight details page, each row in the table represents one log group and lists the related resource.

When there are multiple anomalous log groups at the same time, the timeline view aggregates them and presents them in a single timeline for easy analysis. The purple lines on a timeline indicate spans of time when a log group experienced log anomalies.

Choose a purple line in the timeline to view a sample of log anomaly information such as keyword exceptions and numerical deviations. Choose **View log group details** to view log anomalies. In the window that opens, you can:

- View a graph of log anomalies and relevant events.
- Hover over the graph to view details about the anomalous log data and when it occurred.
- View log anomalies in detail with sample messages, ocurrence frequency, related recommendations, and time of occurrence.

- Click on **View details in CloudWatch** to view log lines from a log anomaly.

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.

Understanding how anomalous behaviors are grouped into insights

An insight is grouped at the *stack level* or the *account level*. If an insight is generated for a resource that is in an AWS CloudFormation stack, then it is a *stack level* insight. Otherwise, it is an *account level* insight.

How a stack is grouped can depend on how you configured your resource analysis coverage in Amazon DevOps Guru.

If your coverage is defined by AWS CloudFormation stacks

All resources contained in the stacks you choose are analyzed, and all detected insights are grouped at the *stack level*.

If your coverage is your current AWS account and Region

All resources in your account and Region are analyzed, and there are three possible grouping scenarios for detected insights.

- An insight generated from a resource that is not part of a stack is grouped at the *account level*.
- An insight generated from a resource that is in one of the first 10,000 analyzed stacks is grouped at the *stack level*.
- An insight generated from a resource that is not in one of the first 10,000 analyzed stacks is grouped at the *account level*. For example, an insight generated for a resource in the 10,001st analyzed stack is grouped at the *account level*.

For more information, see Determine coverage for DevOps Guru.

Understanding insight severities

An insight can have one of three severities, *high*, *medium*, or *low*. An insight is created by Amazon DevOps Guru after it detects related anomalies and assigns each anomaly a severity. DevOps Guru assigns an anomaly a severity of *high*, *medium*, or *low* using domain knowledge and years of collective experience. An insight's severity is determined by the most severe anomaly that contributed to creating the insight.

- If the severity of all the anomalies that generated the insight is *low*, then the insight's severity is *low*.
- If the highest severity of all the anomalies that generated the insight is *medium*, then the insight's severity is *medium*. The severity of some of the anomalies that generated the insight might be *low*.
- If the highest severity of all the anomalies that generated the insight is *high*, then the insight's severity is *high*. The severity of some of the anomalies that generated the insight might be *low* or *medium*.

Monitoring databases using DevOps Guru

DevOps Guru provides significant value for operating databases on AWS. By leveraging its machine learning algorithms, DevOps Guru can help optimize database performance, improve reliability, and reduce operational overhead. This section of the user guide provides a high-level overview of these database capabilities, including specific DevOps Guru use cases for different AWS database services.

DevOps Guru can provide insights for relational databases such as Amazon RDS and Amazon Redshift. It can also provide insights for non-relational or NoSQL databases such as Amazon DynamoDB and Amazon ElastiCache.

Topics

Monitoring relational databases using DevOps Guru

DevOps Guru pulls from two primary data sources to look for insights and anomalies in relational databases. For Amazon RDS and Amazon Redshift, CloudWatch vended metrics are analyzed for all instance types. For Amazon RDS, Performance Insights data is also ingested for the following engine types: RDS for PostgreSQL, Aurora PostgreSQL, and Aurora MySQL.

Monitoring database operations in Amazon RDS

This section includes specific information about use cases and metrics monitored in DevOps Guru for RDS, including data from CloudWatch vended metrics and Performance Insights. For more information about DevOps Guru for RDS, including key concepts, configurations, and benefits, see the section called “Working with anomalies in DevOps Guru for RDS”.

Monitoring RDS using data from CloudWatch vended metrics

DevOps Guru is capable of monitoring every type of RDS instance by ingesting default CloudWatch metrics, such as CPU utilization and read and write operation latency. Because these metrics are vended by default, when you monitor your RDS instances with DevOps Guru, no further configuration is required to gain insights. DevOps Guru automatically establishes a baseline

for these metrics based on historical patterns and compares them to real-time data to detect anomalies and potential issues in your database.

The following table shows a list of potential reactive insights for Amazon RDS from CloudWatch vended metrics.

AWS resource monitored by DevOps Guru	Scenario that DevOps Guru identifies	CloudWatch metrics monitored
Amazon RDS (all instance types)	CPU or memory reaching limits	DBLoad, DBLoadCPU
RDS for PostgreSQL	High replication slot lag	OldestReplicationSlotLag

Additional CloudWatch vended metrics from Amazon RDS instances that DevOps Guru monitors:

- CPUUtilization
- DatabaseConnections
- DiskQueueDepth
- FailedSQLServerAgentJobsCount
- ReadLatency
- ReadThroughput
- ReplicaLag
- WriteLatency

Monitoring RDS using data from Performance Insights

For certain types of Amazon RDS instances, such as Aurora PostgreSQL, Aurora MySQL, and RDS for PostgreSQL, you unlock more capability from DevOps Guru monitoring by ensuring that Performance Insights is enabled on those instances.

DevOps Guru provides reactive insights for a variety of situations, including the following scenarios:

Scenario that DevOps Guru identifies to generate a reactive insight
Locking contention issue

Scenario that DevOps Guru identifies to generate a reactive insight
Missing index
Misconfiguration of application pool
Suboptimal JDBC defaults

DevOps Guru provides proactive insights for a variety of situations, including the following scenarios:

AWS resource monitored by DevOps Guru	Scenario that DevOps Guru identifies to generate a proactive insight
Aurora MySQL	InnoDB history list growing too large, which can lead to degraded performance such as lengthy database shutdown time
Aurora MySQL	An increase in temporary tables created on disk that can impact database performance
RDS for PostgreSQL, Aurora PostgreSQL	A connection that has been idle in transacti on for too long, potential impact of holding locks, blocking other queries, and preventing vacuum (including autovacuum) from cleaning up dead rows

Monitoring database operations in Amazon Redshift

DevOps Guru is capable of monitoring your Amazon Redshift resources by ingesting default CloudWatch metrics, including CPU utilization and the percentage of disk space used. Because these metrics are vended by default, no further configuration is required for DevOps Guru to automatically monitor your Amazon Redshift resources. DevOps Guru establishes a baseline for these metrics based on historical patterns and compares them to real-time data to detect anomalies.

Scenario that DevOps Guru identifies	CloudWatch metrics monitored
Detect high CPU utilization of an Amazon Redshift instance caused by factors such as cluster workload, skewed and unsorted data, or leader node tasks	CPUUtilization
Detect when an Amazon Redshift instance is running out of disk space due to issues with query processing, distribution and sort key, maintenance operations, or tombstone blocks	PercentageDiskSpaceUsed

Additional CloudWatch vended metrics from Amazon Redshift instances that DevOps Guru monitors:

- DatabaseConnections
- HealthStatus
- MaintenanceMode
- NumExceededSchemaQuotas
- PercentageQuotaUsed
- QueryDuration
- QueryRuntimeBreakdown
- ReadIOPS
- ReadLatency
- WLMQueueLength
- WLMQueueWaitTime
- WLMQueryDuration
- WriteLatency

Working with anomalies in DevOps Guru for RDS

DevOps Guru detects, analyzes, and provides recommendations for supported AWS resources, including Amazon RDS engines. For Amazon Aurora and RDS for PostgreSQL database instances

with Performance Insights turned on, DevOps Guru for RDS provides detailed, database-specific analyses of performance issues and recommends corrective actions.

Topics

Overview of DevOps Guru for RDS

Following, you can find a summary of the key benefits and features of DevOps Guru for RDS. For background on insights and anomalies, see DevOps Guru concepts.

Topics

Benefits of DevOps Guru for RDS

If you're responsible for an Amazon RDS database, you might not know that an event or regression that is affecting that database is occurring. When you learn about the issue, you might not know why it's occurring or what to do about it. Rather than turning to a database administrator (DBA) for help or relying on third-party tools, you can follow recommendations from DevOps Guru for RDS.

You gain the following advantages from the detailed analysis of DevOps Guru for RDS:

Fast diagnosis

DevOps Guru for RDS continuously monitors and analyzes database telemetry. Performance Insights, Enhanced Monitoring, and Amazon CloudWatch collect telemetry data for your database instances. DevOps Guru for RDS uses statistical and machine learning techniques to mine this data and detect anomalies. To learn more about telemetry data for Amazon Aurora databases, see Monitoring DB load with Performance Insights on Amazon Aurora and

Monitoring the OS by using Enhanced Monitoring in the *Amazon Aurora User Guide*. To learn more about telemetry data for other Amazon RDS databases, see Monitoring DB load with Performance Insights on Amazon Relational Database Service and Monitoring OS metrics with Enhanced Monitoring in the *Amazon RDS User Guide*.

Fast resolution

Each anomaly identifies the performance issue and suggests avenues of investigation or corrective actions. For example, DevOps Guru for RDS might recommend that you investigate specific wait events. Or it might recommend that you tune your application pool settings to limit the number of database connections. Based on these recommendations, you can resolve performance issues more quickly than by troubleshooting manually.

Proactive insights

DevOps Guru for RDS uses metrics from your resources to detect potentially problematic behavior before it becomes a bigger problem. For example, it can detect when sessions connected to the database are not performing active work and might be keeping database resources blocked. DevOps Guru then provides recommendations to help you address issues before they become bigger problems.

Deep knowledge of Amazon engineers and machine learning

To detect performance issues and help you resolve bottlenecks, DevOps Guru for RDS relies on machine learning (ML) and advanced statistical analysis. Amazon database engineers contributed to the development of the DevOps Guru for RDS findings, which encapsulate many years of managing hundreds of thousands of databases. By drawing on this collective knowledge, DevOps Guru for RDS can teach you best practices.

Key concepts for database performance tuning

DevOps Guru for RDS assumes that you're familiar with a few key performance concepts. To learn more about these concepts, see Overview of Performance Insights in the *Amazon Aurora User Guide* or Overview of Performance Insights in the *Amazon RDS User Guide*.

Topics

- Metrics
- Problem detection
- DB load
- Wait events

Metrics

A metric represents a time-ordered set of data points. Think of a metric as a variable to monitor, and the data points as representing the values of that variable over time. Amazon RDS provides metrics in real time for the database and for the operating system (OS) that your DB instance runs on. You can view all the system metrics and process information for your Amazon RDS DB instances on the Amazon RDS console. DevOps Guru for RDS monitors and provides insights for some of these metrics. For more information, see Monitoring metrics in an Amazon Aurora cluster or Monitoring metrics in an Amazon Relational Database Service instance.

Problem detection

DevOps Guru for RDS employs database and operating system (OS) metrics to detect critical database performance issues, whether those issues are impending or ongoing. There are 2 primary ways DevOps Guru for RDS problem detection works:

- Using thresholds
- Using anomalies

Detecting problems with thresholds

Thresholds are the bounding values against which the monitored metrics are evaluated. You can think of a threshold as a horizontal line on a metric chart that separates normal behavior from potentially problematic behavior. DevOps Guru for RDS monitors specific metrics and creates thresholds by analyzing what levels are considered potentially problematic for a specified resource. DevOps Guru for RDS then creates insights in the DevOps Guru console when new metric values cross a specified threshold over a given period of time on a consistent basis. The insights contain recommendations to prevent future database performance impact.

For example, DevOps Guru for RDS might monitor the number of temporary tables using disk over a period of 15 minutes and create an insight when the rate of temporary tables using disk per second is abnormally high. Increased levels of on-disk temporary table usage might impact the database performance. By exposing this situation before it becomes critical, DevOps Guru for RDS helps you take corrective actions to prevent problems.

Detecting problems with anomalies

While thresholds provide a simple and effective way to detect database problems, in some situations they are not sufficient. Consider a case where metric values are spiking and crossing into

potentially problematic behavior on a regular basis because of a known process, such as a daily reporting job. Since such spikes are expected, creating insights and notifications for each of them would be counterproductive and would likely lead to alert fatigue.

However, it is still necessary to detect spikes that are highly unusual, since metrics that are much higher than the rest or last much longer could represent real database performance issues. To address this concern, DevOps Guru for RDS monitors certain metrics to detect when a metric's behavior becomes highly unusual or anomalous. DevOps Guru then reports these anomalies in insights.

For example, DevOps Guru for RDS might create an insight when DB load is not only high, but also significantly deviates from its usual behavior, which indicates a major unexpected slowdown of database operations. By recognizing only the anomalous DB load spikes, DevOps Guru for RDS lets you focus on the issues that are truly important.

DB load

The key concept for database tuning is the *database load (DB load)* metric. The DB load represents how busy your database is at any given time. An increase in DB load means an increase in database activity.

A *database session* represents an application's dialogue with a relational database. An *active session* is a session that is in the process of running a database request. A session is active when it's either running on CPU or waiting for a resource to become available so that it can proceed. For example, an active session might wait for a page to be read into memory, and then consume CPU while it reads data from the page.

The `DBLoad` metric in Performance Insights is measured in *average active sessions (AAS)*. To calculate AAS, Performance Insights samples the number of active sessions every second. For a specific time period, the AAS is the total number of active sessions divided by the total number of samples. An AAS value of 2 means that, on average, 2 sessions were active in requests at any given time.

An analogy for DB load is activity in a warehouse. Suppose that the warehouse employs 100 workers. If 1 order comes in, 1 worker fulfills the order while the other workers are idle. If 100 or more orders come in, all 100 workers fulfill orders simultaneously. If you periodically sample how many workers are active over a given time period, you can calculate the average number of active workers. The calculation shows that, on average, *N* workers are busy fulfilling orders at any given time. If the average was 50 workers yesterday and 75 workers today, the activity level in the warehouse increased. In the same way, DB load increases as session activity increases.

To learn more, see Database load in the *Amazon Aurora User Guide* or Database load in the *Amazon RDS User Guide*.

Wait events

A *wait event* is a type of database instrumentation that tells you which resource a database session is waiting for so it can proceed. When Performance Insights counts active sessions to calculate database load, it also records the wait events that are causing the active sessions to wait. This technique allows Performance Insights to show you which wait events are contributing to DB load.

Every active session is either running on the CPU or waiting. For example, sessions consume CPU when they search memory, perform a calculation, or run procedural code. When sessions aren't consuming CPU, they might be waiting for a data file to be read or a log to be written to. The more time that a session waits for resources, the less time it runs on the CPU.

When you tune a database, you often try to find the resources that sessions are waiting for. For example, two or three wait events might account for 90% of DB load. This measure means that, on average, active sessions are spending most of their time waiting for a small number of resources. If you can find out the cause of these waits, you can try to remedy the problem.

Consider the analogy of a warehouse worker. An order comes in for a book. The worker might be delayed in fulfilling the order. For example, a different worker might be currently restocking the shelves, or a trolley might not be available. Or the system used to enter the order status might be slow. The longer the worker waits, the longer the order takes to fulfill. Waiting is a natural part of the warehouse workflow, but if wait time become excessive, productivity decreases. In the same way, repeated or lengthy session waits can degrade database performance.

For more information about wait events in Amazon Aurora, see Tuning with wait events for Aurora PostgreSQL and Tuning with wait events for Aurora MySQL in the *Amazon Aurora User Guide*.

For more information about wait events in other Amazon RDS databases, see Tuning with wait events for RDS for PostgreSQL in the *Amazon RDS User Guide*.

Key concepts for DevOps Guru for RDS

An *insight* is generated by DevOps Guru when it detects anomalous or problematic behavior in your operational applications. An insight contains anomalies for one or more resources. An *anomaly* represents one or more related metrics detected by DevOps Guru that are unexpected or unusual.

An insight has a severity of *high*, *medium*, or *low*. The insight severity is determined by the most severe anomaly that contributed to creating the insight. For example, if the insight **AWS-**

ECS_MemoryUtilization_and_others includes one anomaly with low severity and another with high severity, the overall severity of the insight is high.

If Amazon RDS DB instances have Performance Insights turned on, DevOps Guru for RDS provides detailed analysis and recommendations in the anomalies for these instances. To identify an anomaly, DevOps Guru for RDS develops a baseline for database metric values. DevOps Guru for RDS then compares current metric values to the historical baseline.

Topics

Proactive insights

A proactive insight lets you know about problematic behavior before it occurs. It contains anomalies with recommendations and related metrics to help you address the issues before they become bigger problems.

Each proactive insight page provides details about one anomaly.

Reactive insights

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.

Causal anomalies

A *causal anomaly* is a top-level anomaly within a reactive insight. It is shown as the **Primary metric** on the anomaly details page in the DevOps Guru console.**Database load (DB load)** is the causal anomaly for DevOps Guru for RDS. For example, the insight **AWS-ECS_MemoryUtilization_and_others** could have several metric anomalies, one of which is **Database load (DB load)** for the resource **AWS/RDS**.

Within an insight, the anomaly **Database load (DB load)** can occur for multiple Amazon RDS DB instances. The severity of the anomaly might be different for each DB instance. For example, the severity for one DB instance might be high while the severity for the others is low. The console defaults to the anomaly with the highest severity.

Contextual anomalies

A *contextual anomaly* is a finding within **Database load (DB load)** that is related to a reactive insight. It is displayed in the **Related metrics** section of the anomaly details page in the DevOps Guru console. Each contextual anomaly describes a specific Amazon RDS performance issue that requires investigation. For example, a causal anomaly can include the following contextual anomalies:

- **CPU capacity exceeded** – The CPU run queue or CPU utilization are above normal.
- **Database memory low** – Processes don't have enough memory.
- **Database connections spiked** – The number of database connections is above normal.

Recommendations

Each insight has at least one suggested action. The following examples are recommendations generated by DevOps Guru for RDS:

- Tune SQL IDs *`list_of_IDs`* to reduce CPU usage, or upgrade the instance type to increase CPU capacity.
- Review the associated spike of current database connections. Consider tuning the application pool settings to avoid frequent dynamic allocation of new database connections.
- Look for SQL statements that perform excessive memory operations, such as in-memory sorting or large joins.
- Investigate the heavy I/O usage for the following SQL IDs: *`list_of_IDs`*.
- Check for statements that create large amounts of temporary data, for example those that perform large sorts or use large temporary tables.
- Check applications to see what is causing the increase in database workload.
- Consider enabling the MySQL Performance Schema.
- Check for long-running transactions and end them with a commit or rollback.
- Configure the idle_in_transaction_session_timeout parameter to end any session that has been in the 'idle in transaction' state for longer than the specified time.

How DevOps Guru for RDS works

DevOps Guru for RDS collects metric data, analyzes it, and then publishes anomalies in the dashboard.

Topics

Data collection and analysis

DevOps Guru for RDS collects data about your Amazon RDS databases from Amazon RDS Performance Insights. This feature monitors Amazon RDS DB instances, collects metrics, and makes it possible for you to explore the metrics in a chart. The most important performance metric is DBLoad. DevOps Guru for RDS consumes Performance Insights metrics and analyzes them to detect anomalies. For more information about Performance Insights, see Monitoring DB load with Performance Insights on Amazon Aurora in the *Amazon Aurora User Guide* or Monitoring DB load with Performance Insights on Amazon RDS in the *Amazon RDS User Guide*.

DevOps Guru for RDS uses machine learning and advanced statistical analysis to analyze the data that it collects from Performance Insights. If DevOps Guru for RDS finds performance issues, it proceeds to the next step.

Anomaly publication

A database performance issue such as high DB load can degrade the quality of service for your database. When DevOps Guru detects an issue in an RDS database, it publishes an insight in the dashboard. The insight contains an anomaly for the resource **AWS/RDS**.

If Performance Insights is turned on for your instances, the anomaly contains a detailed analysis of the problem. DevOps Guru for RDS also recommends that you perform an investigation or specific corrective action. For example, the recommendation might be to investigate a specific high-load SQL statement, consider increasing CPU capacity, or to close idle-in-transaction sessions.

Supported database engines

DevOps Guru for RDS is supported for the following database engines:

Amazon Aurora with MySQL compatibility

> To learn more about this engine, see Working with Amazon Aurora MySQL in the *Amazon Aurora User Guide*.

Amazon Aurora with PostgreSQL compatibility

To learn more about this engine, see Working with Amazon Aurora PostgreSQL in the *Amazon Aurora User Guide*.

Amazon RDS for PostgreSQL compatibility

To learn more about this engine, see Amazon RDS for PostgreSQL in the *Amazon RDS User Guide*.

DevOps Guru reports anomalies and gives basic analysis for other database engines. DevOps Guru for RDS gives detailed analysis and recommendations only for Amazon Aurora and RDS for PostgreSQL instances.

Enabling DevOps Guru for RDS

When you enable DevOps Guru for RDS, you enable DevOps Guru to analyze anomalies in resources such as DB instances. Amazon RDS makes it easy to discover and enable recommended functionality for an RDS DB instance or DB cluster. To achieve this, RDS makes API calls to other services, such as Amazon EC2, DevOps Guru, and IAM. When the RDS console makes these API calls, AWS CloudTrail logs them for visibility.

To allow DevOps Guru to publish insights for an Amazon RDS database, complete the tasks in the following sections.

Topics

- Turning on Performance Insights for your Amazon RDS DB instances
- Configuring access policies for DevOps Guru for RDS
- Adding Amazon RDS DB instances to your DevOps Guru coverage

Turning on Performance Insights for your Amazon RDS DB instances

For DevOps Guru for RDS to analyze anomalies on a DB instance, make sure that Performance Insights is turned on. If Performance Insights isn't turned on for a DB instance, DevOps Guru for RDS notifies you in the following places:

Dashboard

If you view insights by resource type, the **RDS** tile alerts you that Performance Insights isn't turned on. Choose the link to turn on Performance Insights in the Amazon RDS console.

Insights

In the **Recommendations** section at the bottom of the page, choose **Enable Amazon RDS Performance Insights**.

Settings

In the **Service: Amazon RDS** section, choose the link to turn on Performance Insights in the Amazon RDS console.

For more information, see Turning Performance Insights on and off in the *Amazon Aurora User Guide*, or Turning Performance Insights on and off in the *Amazon RDS User Guide*.

Configuring access policies for DevOps Guru for RDS

For a user to access DevOps Guru for RDS, they must have permissions from either of the following policies:

- The AWS managed policy `AmazonRDSFullAccess`
- A customer managed policy that allows the following actions:
  - `pi:GetResourceMetrics`
  - `pi:DescribeDimensionKeys`
  - `pi:GetDimensionKeyDetails`

For more information, see Configuring access policies for Performance Insights in the *Amazon Aurora User Guide* or Configuring access policies for Performance Insights in the *Amazon RDS User Guide*.

Adding Amazon RDS DB instances to your DevOps Guru coverage

You can configure DevOps Guru to monitor your Amazon RDS databases either in the DevOps Guru console or the Amazon RDS console.

In the DevOps Guru console, you have the following options:

- Turn on DevOps Guru at the account level. This is the default. When you choose this option, DevOps Guru analyzes all supported AWS resources in your AWS Region and AWS account, including Amazon RDS databases.
- Specify AWS CloudFormation stacks for DevOps Guru for RDS.

For more information, see Using AWS CloudFormation stacks to identify resources in your DevOps Guru applications.

- Tag your Amazon RDS resources.

  A *tag* is a custom attribute label that you assign to an AWS resource. Use tags to identify the AWS resources that make up your application. You can then filter your insights by tag to view only those created by your application. To view only insights generated by the Amazon RDS resources in your application, add a value such as `Devops-guru-rds` to your Amazon RDS resource tags. For more information, see Using tags to identify resources in your DevOps Guru applications.

  > **Note**
  >
  > When you tag Amazon RDS resources, you must tag the database instance and not the cluster.

To enable DevOps Guru monitoring from the Amazon RDS console, see Turning on DevOps Guru in the RDS console. Note that to enable DevOps Guru from the Amazon RDS console you must use tags. For more information about tags, see the section called "Using tags to identify resources in your applications".

Analyzing anomalies in Amazon RDS

When DevOps Guru for RDS publishes a performance anomaly in the dashboard, you typically perform the following steps:

1. View the insight in the DevOps Guru dashboard. DevOps Guru for RDS reports both reactive and proactive insights.

   For more information, see Viewing insights.
2. View anomalies for **AWS/RDS** resources.

   For more information, see Viewing reactive anomalies and Viewing proactive anomalies.
3. Respond to DevOps Guru for RDS recommendations.

   For more information, see Responding to recommendations.

4. Monitor the health of your DB instances to make sure that resolved performance problems don't recur.

   For more information, see Monitoring metrics in an Amazon Aurora DB cluster in the *Amazon Aurora User Guide* and Monitoring metrics in an Amazon RDS instance in the *Amazon RDS User Guide*.

Viewing insights

Access the **Insights** page in the DevOps Guru console to find reactive and proactive insights. From there, you can choose an insight from the list to view a detailed page of metrics, recommendations, and more information about the insight.

To view an insight

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Open the navigation pane, and then choose **Insights**.
3. Choose the **Reactive** tab to view reactive insights, or choose **Proactive** to view proactive insights.
4. Choose the name of an insight, prioritizing by status and severity.

   The detailed insight page appears.

Viewing reactive anomalies

Within an insight, you can view anomalies for Amazon RDS resources. On a reactive insight page, in the **Aggregated metrics** section, you can view a list of anomalies with corresponding timelines. There are also sections that display information about log groups and events related to the anomalies. Causal anomalies in a reactive insight each have a corresponding page with details about the anomaly.

Viewing the detailed analysis of an RDS reactive anomaly

In this stage, drill down in the anomaly to get the detailed analysis and recommendations for your Amazon RDS DB instances.

The detailed analysis is only available for Amazon RDS DB instances that have Performance Insights turned on.

To drill down to the anomaly details page

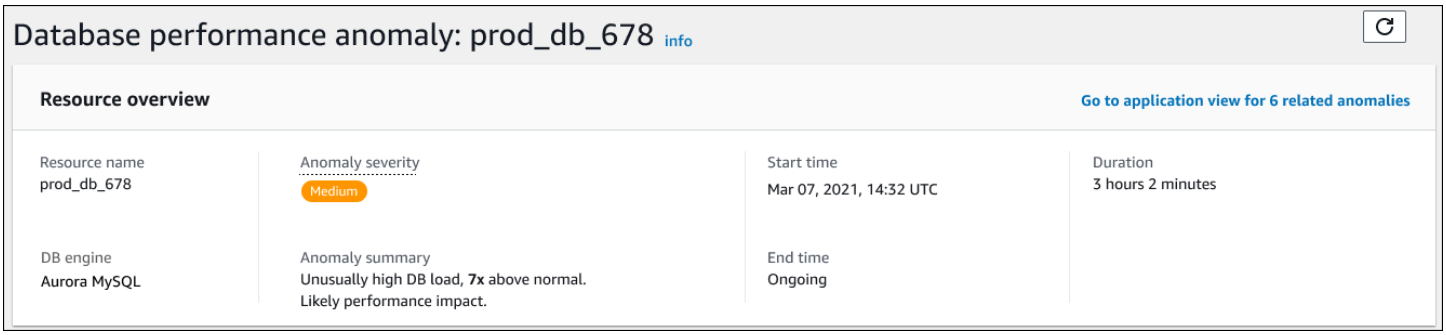
1. On the insight page, find an aggregated metric with the resource type **AWS/RDS**.
2. Choose **View details**.

   The anomaly details page appears. The title begins with **Database performance anomaly** and names the resource show. The console defaults to the anomaly with the highest severity, regardless of when the anomaly occurred.
3. (Optional) If multiple resources are affected, choose a different resource from the list at the top of the page.

Following, you can find descriptions for the components of the details page.

Resource overview

The top section of the details page is **Resource overview**. This section summarizes the performance anomaly experienced by your Amazon RDS DB instance.



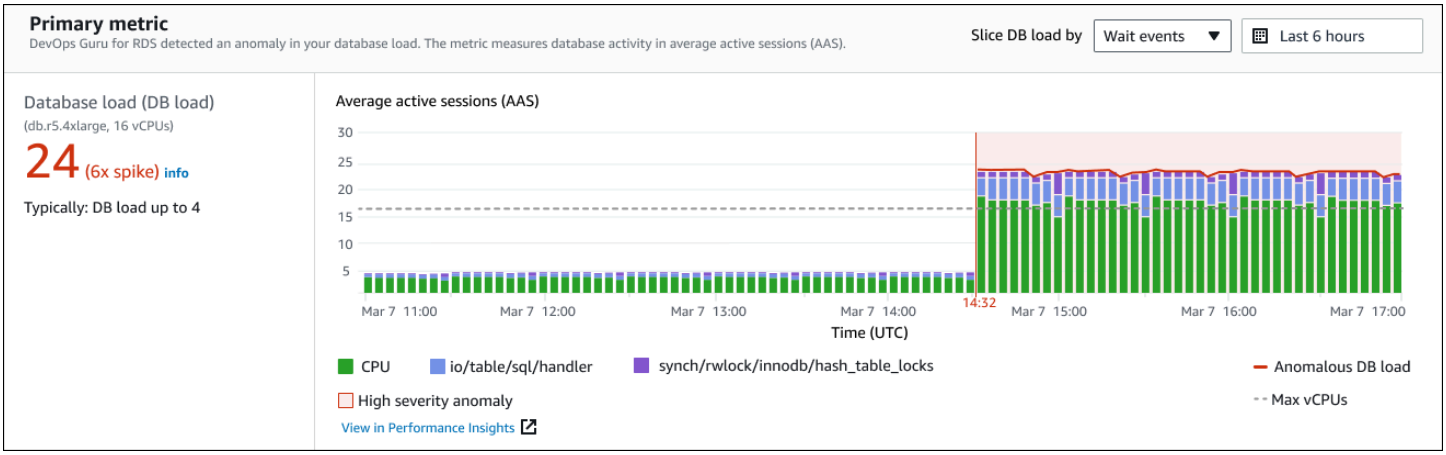
This section has the following fields:

- **Resource name** – The name of the DB instance that is experiencing the anomaly. In this example, the resource is named **prod_db_678**.
- **DB engine** – The name of the DB instance that experiencing the anomaly. In this example, the engine is **Aurora MySQL**.
- **Anomaly severity** – The measure of the negative impact of the anomaly on your instance. Possible severities are **High**, **Medium**, and **Low**.
- **Anomaly summary** – A brief summary of the issue. A typical summary is **Unusually high DB load**.
- **Start time** and **End time** – The time when the anomaly began and ended. If the end time is **Ongoing**, the anomaly is still occurring.

- **Duration** – The duration of the anomalous behavior. In this example, the anomaly is ongoing and has been occurring for 3 hours and 2 minutes.

Primary metric

The **Primary metric** section summarizes the casual anomaly, which is the top-level anomaly within the insight. You can think of the causal anomaly as the general problem experienced by your DB instance.



The left panel provides more details about the issue. In this example, the summary includes the following information:

- **Database load (DB load)** – A categorization of the anomaly as a database load issue. The corresponding metric in Performance Insights is DBLoad. This metric is also published to Amazon CloudWatch.
- **db.r5.4xlarge** – The DB instance class. The number of vCPUs, which is 16 in this example, corresponds to the dotted line in the **Average active sessions (AAS)** chart.
- **24 (6x spike)** – The DB load, measured in average active sessions (AAS) during the time interval reported in the insight. Thus, at any given time during the period of the anomaly, an average of 24 sessions were active on the database. The DB load is 6 times the normal DB load for this instance.
- **Typically: DB load up to 4** – The baseline of DB load, measured in AAS, during a typical workload. The value 4 means that, during normal operations, an average of 4 or fewer sessions are active on the database at any given time.

By default, the load chart is sliced by wait events. This means that for each bar in the chart, the largest colored area represents the wait event that is contributing most to total DB load. The chart

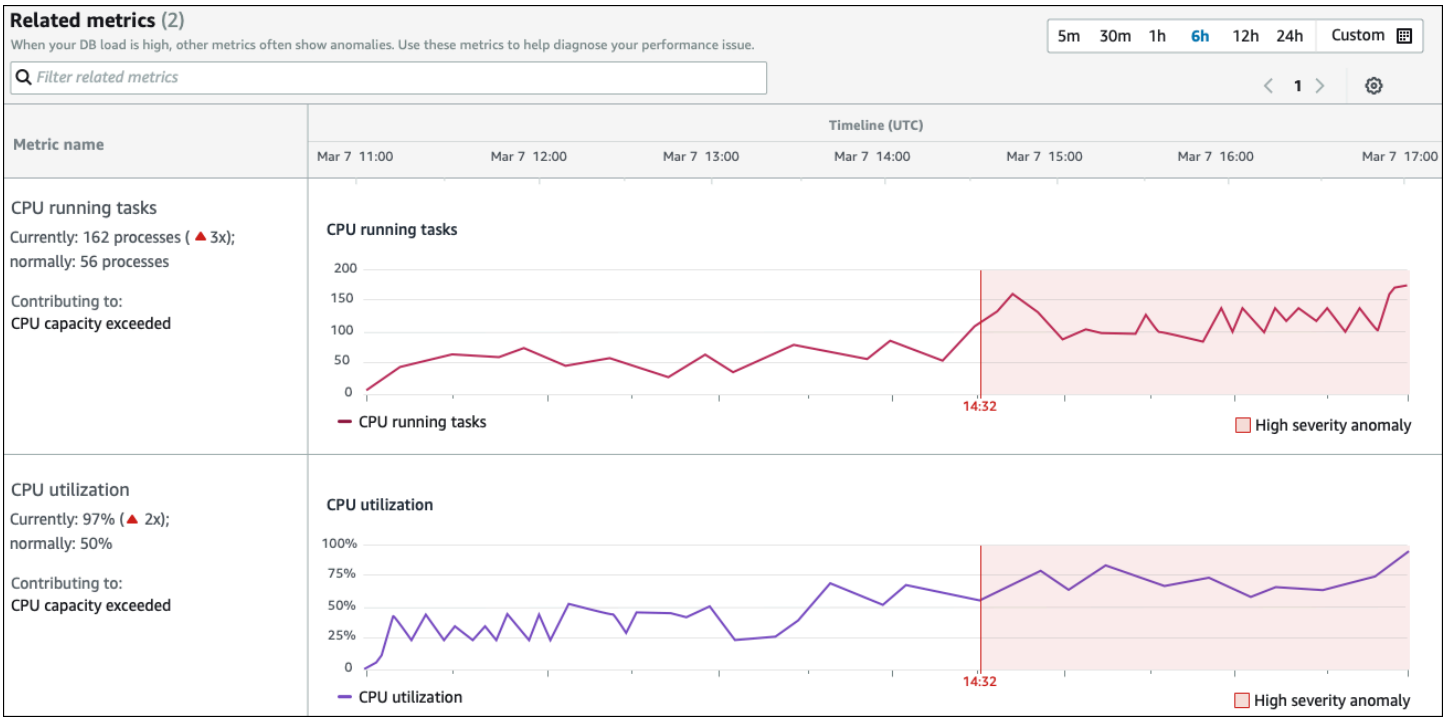
shows the time (in red) when the issue began. Focus your attention on the wait events that take up the most space in the bar:

- CPU
- `IO:wait/io/sql/table/handler`

The preceding wait events appear more than normal for this Aurora MySQL database. To learn how to tune performance using wait events in Amazon Aurora, see Tuning with wait events for Aurora MySQL and Tuning with wait events for Aurora PostgreSQL in the *Amazon Aurora User Guide*. To learn how to tune performance using wait events in RDS for PostgreSQL, see Tuning with wait events for RDS for PostgreSQL in the *Amazon RDS User Guide*.

Related metrics

The **Related metrics** section lists the contextual anomalies, which are specific findings within the causal anomaly. These findings give additional information about the performance issues.



The **Related metrics** table has two columns: **Metrics name** and **Timeline (UTC)**. Every row in the table corresponds to a specific metric.

The first column of every row has the following pieces of information:

- ***Name*** – The name of the metric. The first row identifies the metric as **CPU running tasks**.

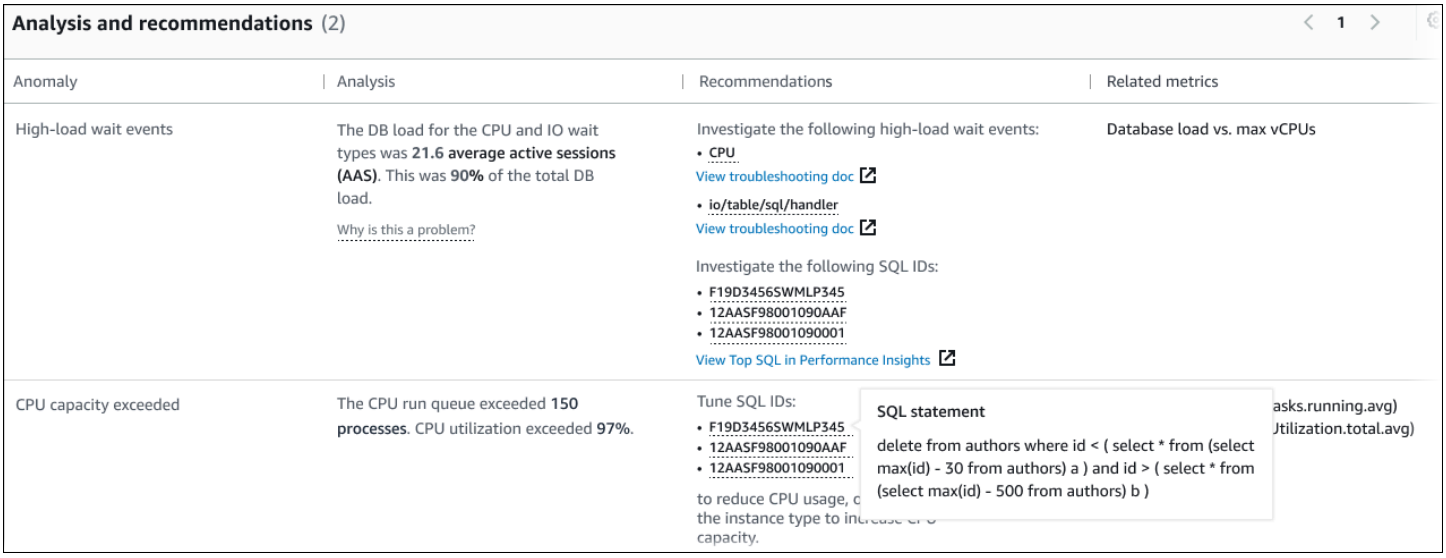
- **Currently** – The current value of the metric. In the first row, the current value is **162 processes (3x)**.
- **Normally** – The baseline of this metric for this database when it is functioning normally. DevOps Guru for RDS calculates the baseline as the 95th percentile value over 1 week of history. The first row indicates that 56 processes are typically running on the CPU.
- **Contributing to** – The finding associated with this metric. In the first row, the **CPU running tasks** metric is associated with the **CPU capacity exceeded** anomaly.

The **Timeline** column shows a line chart for the metric. The shaded area shows the time interval when DevOps Guru for RDS designated the finding as high severity.

Analysis and recommendations

Whereas the causal anomaly describes the overall issue, a contextual anomaly describes a specific finding that requires investigation. Each finding corresponds to a set of related metrics.

In the following example of an **Analysis and recommendations** section, the high DB load anomaly has two findings.



The table has the following columns:

- **Anomaly** – A general description of this contextual anomaly. In this example, the first anomaly is high-load wait events, and the second is CPU capacity exceeded.
- **Analysis** – A detailed explanation of the anomaly.

In the first anomaly, three wait types contribute to 90% of DB load. In the second anomaly, the CPU run queue exceeded 150, which means that at any given time, more than 150 sessions were waiting for CPU time. CPU utilization was over 97%, which means that for the duration of the issue, the CPU was busy 97% of the time. Thus, the CPU was almost continually occupied while an average of 150 sessions waited to run on the CPU.

- **Recommendations** – The suggested user response to the anomaly.

  In the first anomaly, DevOps Guru for RDS recommends that you investigate the wait events `cpu` and `io/table/sql/handler`. To learn how to tune your database performance based on these events, see cpu and io/table/sql/handler in the *Amazon Aurora User Guide*.

  In the second anomaly, DevOps Guru for RDS recommends that you reduce CPU consumption by tuning three SQL statements. You can hover over the links to see the SQL text.

- **Related metrics** – Metrics that give you specific measurements for the anomaly. For more information about these metrics, see Metrics reference for Amazon Aurora in the *Amazon Aurora User Guide* or Metrics reference for Amazon RDS in the *Amazon RDS User Guide*.

  In the first anomaly, DevOps Guru for RDS recommends that compare DB load to the maximum CPU for your instance. In the second anomaly, the recommendation is to look at CPU run queue, CPU utilization, and SQL execution rate.

Viewing proactive anomalies

Within insights, you can view anomalies for Amazon RDS resources. Each proactive insight provides details about one proactive anomaly. On a proactive insight page, you can view an insight overview, detailed metrics about the anomaly, and recommendations to prevent future issues. To view a proactive anomaly, go to the proactive insight page.

Insight overview

The **Insight overview** section provides details about why the insight was created. It displays the severity of the insight as well as a description of the anomaly and a timeframe for when the anomaly occurred. It also lists the number of affected services and applications detected by DevOps Guru.

Metrics

The **Metrics** section provides graphs of the anomaly. Each graph displays a threshold determined by the resource's baseline behavior, as well as data of the metric reported from the time of the anomaly.

Recommendations for aggregated resources

This section suggests actions that you can take to mitigate the reported issues before they become a bigger problem. Actions that you can take are presented in the **Recommended custom change** column. The rationale behind the recommendations is presented in the **Why is DevOps Guru recommending this?** column. For more information about how to respond to recommendations, see the section called “Responding to recommendations”.

Responding to recommendations

Recommendations are the most important part of the insight. In this stage of the analysis, you act to resolve the performance issue. Typically, you take the following steps:

1. Decide whether the reported performance issue indicates a real problem.

   In some cases, an issue might be expected and benign. For example, if you subject a test database to an extreme DB load, DevOps Guru for RDS reports the load as a performance anomaly. However, you don't need to remedy this anomaly because it's an expected result of your testing.

   If you determine that the issue needs a response, go to the next step.
2. Decide whether to implement the recommendation.

   In the table of recommendations, a column shows the recommended actions. For reactive insights, this is the **What we recommend** column on a reactive anomaly detail page. For proactive insights, this is the **Recommended custom change** column on a proactive insight page.

   DevOps Guru for RDS offers a list of recommendations that cover several potential problematic scenarios. After reviewing this list, determine which recommendation is more relevant to your current situation and consider applying it. If a recommendation works for your situation, go to the next step. If not, skip the remaining step and troubleshoot the issue using manual techniques.
3. Perform the recommended actions.

DevOps Guru for RDS recommends that you do either of the following:

- Perform a specific corrective action.

  For example, DevOps Guru for RDS might recommend that you upgrade CPU capacity, tune application pool settings, or enable the Performance Schema.

- Investigate the cause of the issue.

  Typically, DevOps Guru for RDS recommends that you investigate specific SQL statements or wait events. For example, a recommendation might be to investigate the wait event `io/table/sql/handler`. Look up the listed wait event in Tuning with wait events for Aurora PostgreSQL or Tuning with wait events for Aurora MySQL in the *Amazon Aurora User Guide*, or in Tuning with wait events for RDS for PostgreSQL in the *Amazon RDS User Guide*. Then perform the recommended actions.

  > **Important**
  >
  > We recommend that you test any changes on a test instance before modifying a production instance. In this way, you understand the impact of the change.

Monitoring non-relational databases using DevOps Guru

DevOps Guru is capable of generating insights for your non-relational or NoSQL databases that help you keep your resources configured according to best practices. For example, DevOps Guru can help you stay on top of capacity planning by forecasting future needs based on existing traffic. DevOps Guru can identify if you are utilizing less resources than you configured and provide recommendations to improve application availability based on your historic usage. This can help you reduce unnecessary cost.

Beyond capacity planning, DevOps Guru detects and helps you troubleshoot operational issues such as throttling, transaction conflicts, conditional check failures, and areas for improvement in SDK parameters. Databases are typically connected with multiple services and resources, and DevOps Guru can correlate your application structure for analysis using groups based on tagging or AWS CloudFormation aggregation. Anomalies can involve multiple resources that are all affected by the same solution. DevOps Guru is capable of correlating across different resource metrics, configurations, logs, and events. For example, DevOps Guru can analyze and relate data from a Lambda function that might be reading or writing data from a Amazon DynamoDB table. In this

way, DevOps Guru monitors multiple related resources to detect anomalies and provide useful insights for your database solutions.

Monitoring database operations in Amazon DynamoDB

The table below shows example scenarios and insights that DevOps Guru monitors for Amazon DynamoDB.

Amazon DynamoDB use case	Examples	Metrics
Detect when a large percentage of AccountProvisionedReadCapacityUtilization and AccountProvisionedWriteCapacityUtilization are being used, due to a large number of read and write requests.	Amazon DynamoDB table consumption capacities for read or write requests is reaching table-level limits.	AccountProvisionedReadCapacityUtilization, AccountProvisionedWriteCapacityUtilization
Detect conditional check failures in Amazon DynamoDB requests caused by a provided condition expression not matching what is expected in the database.	Conditional check failures are caused by bad data in your table, a strict condition expression, or race conditions.	ConditionalCheckFailedRequests

Monitoring database operations in Amazon ElastiCache

The table below shows example scenarios and insights that DevOps Guru monitors for Amazon ElastiCache.

Scenario that DevOps Guru identifies	CloudWatch metrics monitored
Detect when an Amazon ElastiCache cluster is reaching its compute limit for Redis or Memcached due to changing demands on your clusters.	CPUUtilization, EngineCPUUtilization, Evictions

Integrating with CodeGuru Profiler

This section provides an overview of how Amazon DevOps Guru integrates with Amazon CodeGuru Profiler. You can view recommendations from CodeGuru Profiler as insights in the DevOps Guru console.

Amazon DevOps Guru integrates with Amazon CodeGuru Profiler with an EventBridge managed rule. CodeGuru Profiler sends events to EventBridge. The managed rule routes events that are sent with the default event bus. Each inbound event from CodeGuru Profiler is a proactive anomaly report. For more information, see Working with EventBridge with CodeGuru Profiler.

DevOps Guru supports inbound events with EventBridge. An event indicates a change in a recommendation that DevOps Guru identified. CodeGuru Profiler sends a heartbeat event every 24 hours to show the continuity of the event. Events carry CodeGuru Profiler recommendation information as well as metadata for your compute resources. For information on an event lifecycle, see Amazon EventBridge Events.

When you set up DevOps Guru, DevOps Guru creates the EventBridge Managed Rule in your account that routes events from another service. This rule routes to DevOps Guru. Notifications are sent when there is an inbound event.

An event bus receives events from a source such as DevOps Guru and routes them to rules associated with that event bus. For more information on event buses, see Event buses.

For information on some of the parameters, see Amazon EventBridge events.

To receive CodeGuru Profiler insights in DevOps Guru, you must have the following.

- CodeGuru Profiler must be enabled. For information on enabling CodeGuru Profiler, see Setting up CodeGuru Profiler.
- DevOps Guru must be enabled. For information on enabling DevOps Guru, see Enable DevOps Guru.
- The same resources must be monitored in the same Region in both CodeGuru Profiler and DevOps Guru.

Defining applications using AWS resources

Amazon DevOps Guru groups the resources that are in the coverage boundary that specifies which resources it analyzes for operational insights. The resources are grouped by resources in AWS CloudFormation stacks or by resources with tags. You choose the stacks or tags when you set up DevOps Guru. You can also update the stacks or tags later. We recommend that you think of your resource groups as applications. For example, you might have all resources that you use for a monitoring application defined in one stack. Or you might add the same tag to all the resources that you use in a database application. the boundary that defines which resources DevOps Guru analyzes. All the resources in the collection are inside this boundary. Any resources in your account that are not in your resource collection are outside the boundary and are not analyzed. For more information about the supported services and resources, see Amazon DevOps Guru pricing.

You can define your coverage boundary that contains the resources in your applications three ways.

- Specify that all supported AWS resources in your AWS account and Region. This makes your account and Region your resource boundary. With this option, DevOps Guru analyzes every supported resource in your account and Region. All resources that are in one stack are grouped into an application. Any resources that are not in a stack are grouped into their own application.
- Use AWS CloudFormation stacks to specify the resources in your applications. A stack contains resources that are generated using AWS CloudFormation. In DevOps Guru, you choose stacks in your account. The resources you in each stack you choose are grouped into an application. All resources in the stacks are analyzed by DevOps Guru for insights.
- Use AWS tags to specify the resources in your applications. An AWS tag contains a *key* and a *value*. In DevOps Guru, choose one tag *key* and optionally choose one or more *values* that are paired with that *key*. You can use the *values* to group your resources into applications.

For more information, see Updating your AWS analysis coverage in DevOps Guru.

Topics

Using tags to identify resources in your DevOps Guru applications

You can use tags to identify the AWS resources that Amazon DevOps Guru analyzes and to specify which resources are grouped for monitoring with the selected tag key and tag values. You can edit these configurations when you set up DevOps Guru or when you choose **Edit analyzed resources** from the **Analyzed resources** page. After you select **Tags**, you choose a specific tag key that begins with 'devops-guru-'. To analyze all resources in the account and use tag values to group the resources, select **All Account Resources**. To use tag values to specify the resources for DevOps Guru to analyze, select **Choose specific tag values**.

> **Note**
>
> When **All Account Resources** is selected and no tag value exists, resources without the tag key are grouped and analyzed separately.

You use a tag's *key* to identify the resources, then use *values* with that *key* to group resources into your applications. For example, you can tag your resources with the *key* `devops-guru-applications`, then use that *key* with a different *value* for each of your applications. You might use the tag *key-value* pairs `devops-guru-applications/database`, `devops-guru-applications/cicd`, and `devops-guru-applications/monitoring` to identify three applications in your account. Each application is made up of related resources that contain the same tag *key-value* pair. You add tags to your resources using the AWS service to which they belong. For more information, see Adding AWS tags to AWS resources.

After you add a tag to the resources in your application, you can filter your insights by the tags on resources that generated them. For more information about how to filter your insights using a tag, see Viewing DevOps Guru insights.

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

Topics

- What is an AWS tag?
- Defining a DevOps Guru application using a tag
- Using tags with DevOps Guru

- Adding AWS tags to AWS resources

What is an AWS tag?

Tags help you identify and organize your AWS resources. Many AWS services support tagging, so you can assign the same tag to resources from different services to indicate that the resources are related. For example, you can assign the same tag to an Amazon DynamoDB table resource that you assign to an AWS Lambda function. For more information about using tags, see the Tagging best practices whitepaper.

Each AWS tag has two parts.

- A tag *key* (for example, `CostCenter`, `Environment`, `Project`, or `Secret`). Tag *keys* are case-sensitive.
- An optional field known as a tag *value* (for example, `111122223333`, `Production`, or a team name). Omitting the tag *value* is the same as using an empty string. Like tag *keys*, tag *values* are case-sensitive.

Together these are known as *key*-*value* pairs.

Defining a DevOps Guru application using a tag

To define your Amazon DevOps Guru application using a tag, add that tag to the AWS resources in your account that make up your application. Your tag contains a *key* and a *value*. We recommend that you add a tag to each of your AWS resources analyzed by DevOps Guru that has the same *key*. Use a different *value* in the tag to group resources into your applications. For example, you might assign tags with the *key* `devops-guru-analysis-boundary` to all the AWS resources in your coverage boundary. Use different *values* with that *key* to identify applications in your account. You might use the *values* `containers`, `database`, and `monitoring` for three applications. For more information, see Updating your AWS analysis coverage in DevOps Guru.

If you use AWS tags to specify which resources to analyze, you can use tags with only one *key*. You can pair your tags' *key* paired with any *value*. Use the *value* to group the resources that contain your *key* into your operational applications.

Important

The string used for a *key* in a tag that you use to define your resource coverage must begin with the prefix `Devops-guru-`. The tag *key* might be `DevOps-Guru-deployment-application` or `devops-guru-rds-application`. When you create a *key*, the case of characters in the *key* can be whatever you choose. After you create a *key*, it is case-sensitive. For example, DevOps Guru works with a *key* named `devops-guru-rds` and a *key* named `DevOps-Guru-RDS`, and these act as two different *keys*. Possible *key*/*value* pairs in your application might be `Devops-Guru-production-application/RDS` or `Devops-Guru-production-application/containers`.

Using tags with DevOps Guru

Specify the AWS tags that identify the AWS resources that you want Amazon DevOps Guru to analyze, or specify tag values that identify which resources will be grouped. These resources are your resource coverage boundary. You can choose one *key* and zero or more *values*.

To choose your tags

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Open the navigation pane, then expand **Settings**.
3. In **Analyzed resources**, choose **Edit**.
4. Choose **Tags** if you want DevOps Guru to analyze all resources that contain the tags you choose. Choose a *key*, then choose one of the following options.

   - **All account resources** – Analyze all AWS resources in the current Region and account. Resources with the selected tag key are grouped by tag value, if any exist. Resources without this tag key are grouped and analyzed separately.
   - **Choose specific tag values** – All resources that contain a tag with the *key* you chose are analyzed. DevOps Guru groups your resources into applications by your tag's *values*.

   The tag's *key* must begin with the prefix `devops-guru-`. This prefix isn't case-sensitive. For example, a valid *key* is `DevOps-Guru-Production-Applications`.
5. Choose **Save**.

Adding AWS tags to AWS resources

When you specify the AWS tags that identify the AWS resources that you want DevOps Guru to analyze, choose tags that have resources associated with them. You can add tags to your resources using the AWS service to which each resource belongs, or using the AWS Tag Editor.

- To manage tags using your resources' service, use the console, AWS Command Line Interface, or SDK of the service to which a resource belongs. For example, you can tag an Amazon Kinesis stream resource or an Amazon CloudFront distribution resource. These are two examples of services with resources that can be tagged. Most resources that DevOps Guru can analyze support tags. For more information, see Tagging your streams in the *Amazon Kinesis Developer Guide* and Tagging a distribution in the *Amazon CloudFront Developer Guide*. To learn how to add tags to other types of resources, see the user guide or developer guide for the AWS service to which they belong.

  > **Note**
  >
  > When you tag Amazon RDS resources, you must tag the database instance and not the cluster.

- You can use the AWS Tag Editor to manage tags by resources in your Region and by resources in specific AWS services. For more information, see Tag editor in the *AWS Resource Group and Tags User Guide*.

When you add a tag to a resource, you can add the *key* only, or the *key* and a *value*. For example, you can create a tag with the *key* `devops-guru-` for all the resources that are part of your DevOps application. You can also add a tag with the *key* `devops-guru-` and the *value* RDS, then add that *key-value* pair to only the Amazon RDS resources in your application. This is useful if you want to view insights in the console that are generated from only the Amazon RDS resources in your application.

Using AWS CloudFormation stacks to identify resources in your DevOps Guru applications

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 Updating your AWS analysis coverage in DevOps Guru.

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.

If you choose to have DevOps Guru analyze all supported resources in your account (this means your AWS account and Region is your DevOps Guru coverage boundary), then DevOps Guru analyzes and creates insights for every supported resource in your account, including those in stacks. An insight created from anomalies in a resource that is not in a stack is grouped at the *account level*. If an insight is created from anomalies in a resource that is in a stack,then it is grouped at the *stack level*. For more information, see Understanding how anomalous behaviors are grouped into insights.

Choosing 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

- Choosing stacks for DevOps Guru to analyze (console)
- Choosing stacks for DevOps Guru to analyze (DevOps Guru SDK)

Choosing 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. Choose **CloudFormation stacks** if you want DevOps Guru to analyze the resources that are in stacks you choose, then choose one of the following options.

- **All resources** – All resources that are in stacks in your account are analyzed. Resources in each stack are grouped into their own application. Any resources in your account that are not in a stack are not analyzed.
- **Select stacks** – Select the stacks that you want DevOps Guru to analyze. The resources in each stack you select are grouped into their own application. You can enter the name of a stack in **Find stacks** to quickly locate a specific stack. You can select up to 1,000 stacks.

5. Choose **Save**.

Choosing 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*.

Working with Amazon EventBridge

Amazon DevOps Guru integrates with Amazon EventBridge to notify you of certain events relating to insights and corresponding insight updates. Events from AWS services are delivered to EventBridge in near real time. You can write simple rules to indicate which events are of interest to you and what automated actions to take when an event matches a rule. The actions that can be automatically initiated include the following examples:

- Invoking an AWS Lambda function
- Invoking an Amazon Elastic Compute Cloud run command
- Relaying the event to Amazon Kinesis Data Streams
- Activating a Step Functions state machine
- Notifying an Amazon SNS or an Amazon SQS

You can select any of the following predefined patterns to filter events or create a custom pattern rule to initiate actions in a supported AWS resources.

- DevOps Guru New Insight Open
- DevOps Guru New Anomaly Association
- DevOps Guru Insight Severity Upgraded
- DevOps Guru New Recommendation Created
- DevOps Guru Insight Closed

Events for DevOps Guru

The following are example events from DevOps Guru. Events are emitted on a best-effort basis. To learn more about event patterns, see Getting started with Amazon EventBridge or Amazon EventBridge event patterns.

DevOpsGuru New Insight Open Event

When DevOps Guru opens a new insight, it sends the following event.

```
{
    "version" : "0",
```

```
    "id" : "08108845-ef90-00b8-1ad6-2ee5570ac6c4",
    "detail-type" : "DevOps Guru New Insight Open",
    "source" : "aws.devops-guru",
    "account" : "123456789012",
    "time" : "2021-11-01T17:06:10Z",
    "region" : "us-east-1",
    "resources" : [ ],
    "detail" : {
      "insightSeverity" : "high",
      "insightDescription" : "ApiGateway 5XXError Anomalous In Stack TestStack",
      "insightType" : "REACTIVE",
      "anomalies" : [
        {
          "startTime" : "1635786000000",
          "id" : "AL41JDFFQPYlZlXD8cpREkAAAAF83HGGgC9TmTr9lbfJ7sCiISlWMeFCbHY_XXXX",
          "sourceDetails" : [
            {
              "dataSource" : "CW_METRICS",
              "dataIdentifiers" : {
                "period" : "60",
                "stat" : "Average",
                "unit" : "None",
                "name" : "5XXError",
                "namespace" : "AWS/ApiGateway",
                "dimensions" : [
                  {
                    "name" : "ApiName",
                    "value" : "Test API Service"
                  },
                  {
                    "name" : "Stage",
                    "value" : "prod"
                  }
                ]
              }
            }
          ]
        }
      ],
      "accountId" : "123456789012",
      "messageType" : "NEW_INSIGHT",
      "insightUrl" : "https://us-east-1.console.aws.amazon.com/devops-guru/#/insight/
reactive/AIYH6JxdbgkcG0xJmypiL4MAAAAAAAAAL0SLEjkxiNProXWcsTJbLU07EZ7XXXX",
      "startTime" : "1635786120000",
```

```
      "insightId" : "AIYH6JxdbgkcG0xJmypiL4MAAAAAAAAAL0SLEjkxiNProXWcsTJbLU07EZ7XXXX",
      "region" : "us-east-1"
    }
  },
```

Custom sample event pattern for high severity new Insight

Rules use event patterns to select events and route them to targets. The following is a sample DevOps Guru event pattern.

```
{
  "source": [
    "aws.devops-guru"
  ],
  "detail-type": [
    "DevOps Guru New Insight Open"
  ],
  "detail": {
    "insightSeverity": [
         "high"
         ]
  }
}
```

Updating 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.
- Features for enhanced insights. This includes log anomaly detection, encryption, and your AWS Systems Manager integration settings. This determines whether DevOps Guru displays log data, whether you use additional security keys, and whether an OpsItem is created in Systems Manager OpsCenter for each new insight.

Topics

Updating your management account settings

You can configure DevOps Guru for accounts in your organization. If you haven't registered a delegated administrator, you can do so by choosing **Register delegated administrator**. For more information on registering a delegated administrator, see Enable DevOps Guru.

Updating your AWS analysis coverage in DevOps Guru

You can update which AWS resources in your account DevOps Guru analyzes. To do this, navigate to the **Analyzed resources** page in the console and then choose **Edit**. For more information, see *Viewing analyzed resources*.

Updating 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 Amazon SNS topics. You can specify up to two topics.

DevOps Guru sends notifications for the following updates:

- 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.

DevOps Guru also sends notifications if a selected AWS CloudFormation stack or tag key is invalid when you are attempting to add resources to your DevOps Guru account.

You can choose to receive Amazon SNS notifications for all kinds of updates to an issue or to receive Amazon SNS notifications only when the issue is opened, closed, or has a change in severity. By default, you receive notifications for all updates.

To update your notifications, first navigate to the notifications page and then choose whether to add, remove, or update configurations for Amazon SNS notification topics.

Topics

Navigate to notification settings in the DevOps Guru console

To update notifications, you must first navigate to the notification settings section.

To navigate to the notification settings section

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Choose **Settings** in the navigation pane.

   The Settings page includes the **Notifications** section, with information about configured Amazon SNS topics.

Adding Amazon SNS notification topics in the DevOps Guru console

To add an Amazon SNS notification topic in the DevOps Guru console

1. the section called “Navigate to notification settings in the DevOps Guru console”.
2. Choose **Add notification**.
3. To add an Amazon SNS topic, do one of the following.

   - Choose **Generate a new SNS topic using email**. Then, from **Specify the email address**, enter the email address you want to receive notifications. To enter in additional email addresses, choose **Add new email**.
   - Choose **Use an existing SNS topic**. Then, from **Choose a topic in your AWS account**, choose the topic you want to use.
   - Choose **Use an existing SNS topic ARN to specify an existing topic from another 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 Amazon SNS topics.
4. Choose **Save**.

Removing Amazon SNS notification topics in the DevOps Guru console

To remove Amazon SNS topics in the DevOps Guru console

1. the section called “Navigate to notification settings in the DevOps Guru console”.

2. Choose **Select existing topic**.
3. From the drop-down menu, select the topic you want to remove.
4. Choose **Remove**.
5. Choose **Save**.

Updating Amazon SNS notification configurations

There are two types of notification configurations for Amazon SNS notification topics in DevOps Guru. You can choose to receive notifications of all severity levels or only notifications with **High** and **Medium** severity levels. You can also choose to receive notifications for all kinds of updates or only some kinds of updates.

When you choose to receive Amazon SNS notifications for all kinds of updates to the issue, DevOps Guru sends notifications for the following updates:

- 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.

By default, you receive only **High** and **Medium** severity level notifications, and you receive notifications for all kinds of updates.

To update notification configurations for Amazon SNS notification topics

1. the section called “Navigate to notification settings in the DevOps Guru console”.
2. Choose **Select existing topic**.
3. From the drop-down menu, select the topic you want to make updates to.
4. Choose **All severity levels** to receive notifications with High, Medium, and Low severity levels, or choose **Only High and Medium** to receive notifications with High and Medium severity levels.
5. Choose **Notify me on all updates to the insight**, or choose **Notify me when an insight is opened or closed, or the severity level changes from Low or Medium to High**.

6. Choose **Save**.

Permissions added to your Amazon SNS topic

An Amazon SNS topic is a resource that contains an AWS Identity and Access Management (IAM) resource policy. When you specify a topic here, DevOps Guru appends the following permissions to its resource policy.

```
{
    "Sid": "DevOpsGuru-added-SNS-topic-permissions",
    "Effect": "Allow",
    "Principal": {
        "Service": "region-id.devops-guru.amazonaws.com"
    },
    "Action": "sns:Publish",
    "Resource": "arn:aws:sns:region-id:topic-owner-account-id:my-topic-name",
    "Condition" : {
      "StringEquals" : {
        "AWS:SourceArn": "arn:aws:devops-guru:region-id:topic-owner-account-
id:channel/devops-guru-channel-id",
        "AWS:SourceAccount": "topic-owner-account-id"
    }
  }
}
```

These permissions are required for DevOps Guru to publish notifications using a topic. If you prefer to not have these permissions on the topic, you can safely remove them and the topic will continue to work as it did before you chose it. However, if these appended permissions are removed, DevOps Guru cannot use the topic to generate notifications.

Filtering your DevOps Guru notifications

You can filter your DevOps Guru notifications by the section called “Updating Amazon SNS notification configurations” or by using a Amazon SNS subscription filter policy.

Topics

Filtering notifications with a Amazon SNS subscription filter policy

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.

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_ASSOCIATION` — 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.

```
{
      "accountId": "123456789012",
      "region": "us-east-1",
      "messageType": "NEW_ASSOCIATION",
      "insightId": "ADyf4FvaVNDzu9MA2-IgFDkAAAAAAAAAEGpJd5sjicgauU2wmAlnWUyyI2hiO5it",
```

```
      "insightName": "Repeated Insight: Anomalous increase in Lambda
 ApigwLambdaDdbStack-22-Function duration due to increased number of invocations",
      "insightUrl": "https://us-east-1.console.aws.amazon.com/devops-guru/insight/
reactive/ADyf4FvaVNDzu9MA2-IgFDkAAAAAAAAAEGpJd5sjicgauU2wmAlnWUyyI2hiO5it",
      "insightType": "REACTIVE",
      "insightDescription": "At March 29, 2023 22:02 GMT, Lambda function
 ApigwLambdaDdbStack-22-Function had\n an increased duration anomaly possibly caused by
 the Lambda function invocation increase. DevOps Guru has detected this is a repeated
 insight. DevOps Guru treats repeated insights as 'Low Severity'.",
      "startTime": 1628767500000,
      "startTimeISO": "2023-03-29T22:00:00Z",
      "anomalies": [
        {
          "id": "AG2n8ljW74BoI1CHu-m_oAgAAAF7Ohu24N4Yro69ZSdUtn_alzPH7VTpaL30JXiF",
          "startTime": 1628767500000,
          "startTimeISO": "2023-03-29T22:00:00Z",
          "openTime": 1680127740000,
          "openTimeISO": "2023-03-29T22:09:00Z",
          "sourceDetails": [
            {
              "dataSource": "CW_METRICS",
              "dataIdentifiers": {
                "namespace": "AWS/SQS",
                "name": "ApproximateAgeOfOldestMessage",
                "stat": "Maximum",
                "unit": "None",
                "period": "60",
                "dimensions": "{\"QueueName\":\"FindingNotificationsDLQ\"}"
              }
            }
          ],
          "associatedResourceArns":[
           "arn:aws:sns:us-east-1:123456789012:DevOpsGuru-insights-sns"
          ]
        }
      ],
      "resourceCollection":{
      "cloudFormation":{
         "stackNames":[
            "CapstoneNotificationPublisherEcsApplicationInfrastructure"
          ]
        }
      }
```

```
}
```

Updating 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.

Updating log anomaly detection in DevOps Guru

To manage your log anomaly detection settings

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Choose **Settings** in the navigation pane.

3. In **Log anomaly detection**, select **Enable log anomaly detection by granting DevOps Guru permissions to display log data associated with an insight.** to have DevOps Guru display log data related to insights.

Updating encryption settings in DevOps Guru

You can update encryption settings to use AWS owned keys or AWS KMS customer managed keys. When switching to a new customer managed AWS KMS key from an existing customer managed AWS KMS key, DevOps Guru automatically starts encrypting newly ingested metadata using the new key. The historical data will remain encrypted with the previous configured customer managed AWS KMS key.

> **Note**
>
> If you revoke the grant, or disable or delete the previous AWS KMS key, DevOps Guru won't be able to access any of the data encrypted by this key and you might see the `AccessDeniedException` when performing a read operation.

To manage your encryption settings

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Choose **Settings** in the navigation pane.
3. In the **Encryption** section, choose **Edit encryption**.
4. Select the encrpytion type you would like to use to protect your data. You can use a default AWS owned key, choose an existing customer managed key, or create a new customer managed AWS KMS key.
5. Choose **Save**.

Encryption is an important part of DevOps Guru security. For more information, see the section called “Data protection”.

Viewing notifications

There are different types of notifications in DevOps Guru.

Topics

The sections on this page show examples of each type of notification.

New insight

Notifications for new insights contain the following information:


```
{
   "accountId":"123456789101",
   "region":"eu-west-1",
   "messageType":"NEW_INSIGHT",
   "insightId":"a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
   "insightName": "Repeated Insight: ApiGateway 5XXError Anomalous In Application
 CanaryCommonResources-123456789101-LogAnomaly-4",
   "insightUrl":"https://eu-west-1.console.aws.amazon.com/devops-guru/insight/reactive/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
   "insightType":"REACTIVE",
   "insightDescription":"DevOps Guru has detected this is a repeated insight. DevOps
 Guru treats repeated insights as 'Low Severity'.",
   "insightSeverity":"medium",
   "startTime": 1680148920000,
   "startTimeISO": "2023-03-30T04:02:00Z",
   "anomalies":[
      {
         "id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
         "startTime": 1680148800000,
         "startTimeISO": "2023-03-30T04:00:00Z",
```

```
         "openTime": 1680148920000,
         "openTimeISO": "2023-03-30T04:02:00Z",
         "sourceDetails":[
            {
               "dataSource":"CW_METRICS",
               "dataIdentifiers":{
                  "name":"ApproximateAgeOfOldestMessage",
                  "namespace":"AWS/SQS",
                  "period":"60",
                  "stat":"Maximum",
                  "unit":"None",
                  "dimensions":"{\"QueueName\":\"SampleQueue\"}"
               }
            }
         ],
         "associatedResourceArns":[
            "arn:aws:sqs:eu-west-1:123456789101:SampleQueue"
         ]
      }
   ],
   "resourceCollection":{
        "cloudFormation":{
            "stackNames":[
                "SampleApplication"
            ]
        },
   }
}
```

Closed insight

Notifications for closed insights contain the following information:

```
{
"accountId":"123456789101",
   "region":"us-east-1",
   "messageType":"CLOSED_INSIGHT",
   "insightId":"a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
   "insightName": "DynamoDB table writes are under utilized in mock-stack",
   "insightUrl":"https://us-east-1.console.aws.amazon.com/devops-guru/insight/
proactive/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
   "insightType":"PROACTIVE",
   "insightDescription":"DynamoDB table writes are under utilized",
```

```
    "insightSeverity":"medium",
    "startTime": 1670612400000,
    "startTimeISO": "2022-12-09T19:00:00Z",
    "endTime": 1679994000000,
    "endTimeISO": "2023-03-28T09:00:00Z",
    "anomalies":[
       {
          "id":"a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
          "startTime": 1665428400000,
          "startTimeISO": "2022-10-10T19:00:00Z",
          "endTime": 1679986800000,
          "endTimeISO": "2023-03-28T07:00:00Z",
          "openTime": 1670612400000,
          "openTimeISO": "2022-12-09T19:00:00Z",
          "closeTime": 1679994000000,
          "closeTimeISO": "2023-03-28T09:00:00Z",
          "description":"Empty receives while messages are available",
          "anomalyResources":[
             {
                "type":"AWS::SQS::Queue",
                "name":"SampleQueue"
             }
          ],
          "sourceDetails":[
             {
                "dataSource":"CW_METRICS",
                "dataIdentifiers":{
                "name":"NumberOfEmptyReceives",
                   "namespace":"AWS/SQS",
                   "period":"60",
                   "stat":"Sum",
                   "unit":"COUNT",
                   "dimensions":"{\"QueueName\":\"SampleQueue\"}"
                }
             }
           ],
         "associatedResourceArn": [
             "arn:aws:sqs:us-east-1:123456789101:SampleQueue"
         ]
       }
    ],
    "resourceCollection":{
         "cloudFormation":{
             "stackNames":[
```

```
                "SampleApplication"
            ]
        }
    }
}
```

New association

Notifications for new associations contain the following information:

```
{
"accountId":"123456789101",
   "region":"eu-west-1",
   "messageType":"NEW_ASSOCIATION",
   "insightId":"a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
   "insightName": "Repeated Insight: Anomalous increase in Lambda
 ApigwLambdaDdbStack-22-GetOneFunction duration due to increased number of
 invocations",
   "insightUrl":"https://eu-west-1.console.aws.amazon.com/devops-guru/insight/reactive/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
   "insightType":"REACTIVE",
   "insightDescription":"At March 29, 2023 22:02 GMT, Lambda function
 ApigwLambdaDdbStack-22-GetOneFunction had\nan increased duration anomaly possibly
 caused by the Lambda function invocation increase. DevOps Guru has detected this is a
 repeated insight. DevOps Guru treats repeated insights as 'Low Severity'.",
   "insightSeverity":"medium",
   "startTime": 1680127200000,
   "startTimeISO": "2023-03-29T22:00:00Z",
   "anomalies":[
      {
         "id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
         "startTime":1672945500000,
         "startTimeISO": "2023-03-29T22:00:00Z",
         "openTime": 1680127740000,
         "openTimeISO": "2023-03-29T22:09:00Z",
         "sourceDetails":[
            {
               "dataSource":"CW_METRICS",
               "dataIdentifiers":{
               "namespace":"AWS/SQS",
                  "name":"ApproximateAgeOfOldestMessage",
                  "stat":"Maximum",
                  "unit":"None",
```

```
                  "period":"60",
                  "dimensions":"{\"QueueName\":\"SampleQueue\"}"
               }
            }
         ],
         "associatedResourceArns":[
            "arn:aws:sqs:eu-west-1:123456789101:SampleQueue"
         ]
      }
   ],
   "resourceCollection":{
        "cloudFormation":{
            "stackNames":[
                "SampleApplication"
            ]
        }
   }
}
```

New recommendation

Notifications for new recommendations contain the following information:

```
{
   "accountId":"123456789101",
   "region":"us-east-1",
   "messageType":"NEW_RECOMMENDATION",
   "insightId":"a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
   "insightName": "Recreation of AWS SDK Service Clients",
   "insightUrl":"https://us-east-1.console.aws.amazon.com/devops-guru/insight/
proactive/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
   "insightType":"PROACTIVE",
   "insightDescription": "Usually for a given service you can create one [AWS SDK
 service client](https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/creating-
clients.html) and reuse that client across your entire service.\n\nWhen instead you
 create a new AWS SDK service client for each call (e.g. for DynamoDB) it\u0027s
 generally a waste of CPU time.",
   "insightSeverity":"medium",
   "startTime": 1680125893576,
   "startTimeISO": "2023-03-29T21:38:13.576Z",
   "recommendations":[
      {
         "name":"Tune Availability Zones of your Lambda Function",
```

```
        "description":"Based on your configurations, we recommend that you set
 SampleFunction to be deployed in at least 3 Availability Zones to maintain Multi
 Availability Zone Redundancy.",
        "reason":"Lambda Function SampleFunction is currently only deployed to 2
 unique Availability zones in a region with 7 total Availability zones.",
        "link":"https://docs.aws.amazon.com/lambda/latest/dg/configuration-vpc.html",
        "relatedAnomalies":[
           {
              "sourceDetails":{
                   "cloudWatchMetrics":null
              },
              "resources":[
                 {
                    "name":"SampleFunction",
                    "type":"AWS::Lambda::Function"
                 }
              ],
              "associatedResourceArns": [
                 "arn:aws:lambda:arn:123456789101:SampleFunction"
              ]
           }
        ]
     }
  ],
  "resourceCollection": {
       "cloudFormation": {
       "stackNames":[
           "SampleApplication"
        ]
     }
  }
}
```


Severity upgraded

Notifications for severity upgrades contain the following information:


```
{
"accountId":"123456789101",
   "region":"eu-west-1",
   "messageType":"SEVERITY_UPGRADED",
   "insightId":"a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
```

```
   "insightName": "Repeated Insight: ApiGateway 5XXError Anomalous In Application
 CanaryCommonResources-123456789101-LogAnomaly-11",
   "insightUrl":"https://eu-west-1.console.aws.amazon.com/devops-guru/insight/reactive/
a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
   "insightType":"REACTIVE",
   "insightDescription": "DevOps Guru has detected this is a repeated insight. DevOps
 Guru will treat future occurrences of this insight as 'Low Severity' for the next 7
 days.",
   "insightSeverity":"high",
   "startTime": 1680127320000,
   "startTimeISO": "2023-03-29T22:02:00Z",
   "resourceCollection":{
        "cloudFormation":{
            "stackNames":[
                "SampleApplication"
            ]
        }
   }
}
```

Resource validation failure

You can use AWS CloudFormation stacks and AWS tags to filter and identify the AWS resources that you want DevOps Guru to analyze. When you choose an invalid stack or tag for DevOps Guru to identify resources with, DevOps Guru creates a `SELECTED_RESOURCE_FILTER_VALIDATION_FAILURE` notification. This can happen when the tag or stack name that you specify does not have resources associated with it. To get the most out of DevOps Guru filtering methods, choose stacks and tags that have resources associated with them.

```
{
    "accountId":"123456789101",
    "region":"eu-west-1",
    "messageType":"SELECTED_RESOURCE_FILTER_VALIDATION_FAILURE",
    "ResourceFilterType": "Tags",
    "InvalidResourceNames": [
         "Devops-Guru-tag-key-tag-value"
    ],
    "awsInsightSource": "aws.devopsguru"
}
```

Viewing resources analyzed by DevOps Guru

DevOps Guru provides a list of resource names and their application boundaries under analysis using the `ListMonitoredResources` action. This information is collected from Amazon CloudWatch, AWS CloudTrail, and other AWS services using the DevOps Guru service linked role.

Note that even if a user does not have explicit permission to access the APIs for another service such as AWS Lambda or Amazon RDS, DevOps Guru still provides a list of resources from that service as long as the `ListMonitoredResources` action is allowed.

Topics

Updating 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. When you specify your boundary, your resources are grouped in applications. You have four boundary coverage options.

- Choose to have DevOps Guru analyze all supported resources in your account. All resources in your account that are in a stack are grouped into an application. If you have multiple stacks in your account, then the resources in each stack make up their own application. If any resources in your account are not in a stack, they are grouped into their own application.
- Specify 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. For more information, see Working with stacks in the *AWS CloudFormation User Guide* and Determine coverage for DevOps Guru.
- Specify resources by using AWS tags. DevOps Guru either analyzes all the resources in your account and Region or all the resources that contain the tag key that you choose. Resources are grouped based on selected tag values. For more information, see Using tags to identify resources in your DevOps Guru applications.
- 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. Expand **Analyzed resources** in the navigation pane.
3. Choose **Edit**.
4. Choose one of the following coverage options.

   - Choose **All account resources** if you want DevOps Guru to analyze all supported resources in your AWS account and Region. If you choose this option, your AWS account is your resource analysis coverage boundary. All resources in each stack in your account are grouped into their own application. Any remaining resources that are not in a stack are grouped into their own application.
   - Choose **CloudFormation stacks** if you want DevOps Guru to analyze the resources that are in stacks you choose, then choose one of the following options.
     - **All resources** – All resources that are in stacks in your account are analyzed. Resources in each stack are grouped into their own application. Any resources in your account that are not in a stack are not analyzed.
     - **Select stacks** – Select the stacks that you want DevOps Guru to analyze. The resources in each stack you select are grouped into their own application. You can enter the name of a stack in **Find stacks** to quickly locate a specific stack. You can select up to 1,000 stacks.

     For more information, see Using AWS CloudFormation stacks to identify resources in your DevOps Guru applications.
   - Choose **Tags** if you want DevOps Guru to analyze all resources that contain the tags you choose. Choose a *key*, then choose one of the following options.

- **All account resources** –Analyze all AWS resources in the current Region and account. Resources with the selected tag key are grouped by tag value, if any exist. Resources without this tag key are grouped and analyzed separately.
- **Choose specific tag values** – All resources that contain a tag with the *key* you chose are analyzed. DevOps Guru groups your resources into applications by your tag's *values*.

  The tag's *key* must begin with the prefix `devops-guru-`. This prefix isn't case-sensitive. For example, a valid *key* is `DevOps-Guru-Production-Applications`. For more information, see Using tags to identify resources in your DevOps Guru applications.

- Choose **None** 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**.

Removing analyzed resource view for users

Even if a user does not have explicit permission to access the APIs for another service such as Lambda or Amazon RDS, DevOps Guru still provides a list of resources from that service as long as the `ListMonitoredResources` action is allowed. To change this behavior, you can update your AWS IAM policy to deny this action.

```
{
            "Sid": "DenyListMonitoredResources",
            "Effect": "Deny",
            "Action": [
                "devops-guru:ListMonitoredResources"
            ]
}
```

Best practices in DevOps Guru

The following best practices can help you understand, diagnose, and fix anomalous behavior detected by Amazon DevOps Guru. Use best practices with Understanding insights in the DevOps Guru console 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 to pinpoint when and how behavior changed. This can help you diagnose and fix the problem.
- For Amazon RDS resources, look at Performance Insights metrics. By correlating counter metrics with database load, you can get detailed information about performance issues. For more information, see Analyzing performance anomalies with DevOps Gurufor Amazon RDS.
- 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

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

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

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 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 text 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 text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

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 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 Private Certificate Authority. For more information, see Signature Version 4 signing process and What is ACM PCA.

- **Encryption of data at-rest**: For all AWS resources analyzed by DevOps Guru, the Amazon CloudWatch metrics and data, resource 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.

  Customer managed keys: DevOps Guru supports encrypting customer content and sensitive metadata such as log anomalies generated from CloudWatch Logs with customer managed keys. This feature provides you the option of adding a self-managed security layer to help you meet the compliance and regulatory requirements of your organization. For information on enabling customer managed keys in your DevOps Guru settings, see the section called “Updating encryption”.

  Because you have full control of this layer of encryption, you can perform such tasks as:

  - Establishing and maintaining key policies
  - Establishing and maintaining IAM policies and grants
  - Enabling and disabling key policies
  - Rotating key cryptographic material
  - Adding tags
  - Creating key aliases
  - Scheduling keys for deletion

  For more information, see Customer managed keys in the AWS Key Management Service Developer Guide.

  > **Note**
  >
  > DevOps Guru automatically enables encryption at rest using AWS owned keys to protect sensitive metadata at no charge. However, AWS KMS charges apply for using a customer

managed key. For more information about pricing, see the AWS Key Management Service pricing.

How DevOps Guru uses grants in AWS KMS

DevOps Guru requires a grant to use your customer managed key.

When you choose to enable encryption with a customer managed key, DevOps Guru creates a grant on your behalf by sending a CreateGrant request to AWS KMS. Grants in AWS KMS are used to give DevOps Guru access to a AWS KMS key in a customer account.

DevOps Guru requires the grant to use your customer managed key for the following internal operations:

- Send DescribeKey requests to AWS KMS to verify that the symmetric customer managed KMS key ID entered when creating a tracker or geofence collection is valid.
- Send GenerateDataKey requests to AWS KMS to generate data keys encrypted by your customer managed key.
- Send Decrypt requests to AWS KMS to decrypt the encrypted data keys so that they can be used to encrypt your data.

You can revoke access to the grant, or remove the service's access to the customer managed key at any time. If you do, DevOps Guru won't be able to access any of the data encrypted by the customer managed key, which affects operations that are dependent on that data. For example, if you attempt to get encrypted log anomaly information that DevOps Guru can't access, then the operation would return an AccessDeniedException error.

Monitoring your encryption keys in DevOps Guru

When you use an AWS KMS customer managed key with your DevOps Guru resources, you can use AWS CloudTrail or CloudWatch Logs to track requests that DevOps Guru sends to AWS KMS.

Create a customer managed key

You can create a symmetric customer managed key by using the AWS Management Console or the AWS KMS APIs.

To create a symmetric customer managed key, see Creating symmetric encryption KMS keys.

Key policy

Key policies control access to your customer managed key. Every customer managed key must have exactly one key policy, which contains statements that determine who can use the key and how they can use it. When you create your customer managed key, you can specify a key policy. For more information, see Authentication and access control for AWS KMS in the AWS Key Management Service Developer Guide.

To use your customer managed key with your DevOps Guru resources, the following API operations must be permitted in the key policy:

- `kms:CreateGrant` – Adds a grant to a customer managed key. Grants control access to a specified AWS KMS key, which allows access to grant operations DevOps Guru requires. For more information about using grants, see the AWS Key Management Service Developer Guide.

This allows DevOps Guru to do the following:

- Call GenerateDataKey to generate an encrypted data key and store it, because the data key isn't immediately used to encrypt.
- Call Decrypt to use the stored encrypted data key to access encrypted data.
- Set up a retiring principal to allow the service to RetireGrant.
- Use kms:DescribeKey to provide the customer managed key details to allow DevOps Guru to validate the key.

The following statement includes policy statement examples you can add for DevOps Guru:

```
"Statement" : [
  {
    "Sid" : "Allow access to principals authorized to use DevOps Guru",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "*"
    },
    "Action" : [
      "kms:DescribeKey",
      "kms:CreateGrant"
    ],
    "Resource" : "*",
    "Condition" : {
```

```
        "StringEquals" : {
          "kms:ViaService" : "devops-guru.Region.amazonaws.com",
          "kms:CallerAccount" : "111122223333"
        }
    },
    {
      "Sid": "Allow access for key administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
       },
      "Action" : [
        "kms:*"
       ],
      "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
    },
    {
      "Sid" : "Allow read-only access to key metadata to the account",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:root"
      },
      "Action" : [
        "kms:Describe*",
        "kms:Get*",
        "kms:List*"
      ],
      "Resource" : "*"
    }
  ]
```

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). 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

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.

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 service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page

to understand the basic concepts of IAM. To learn more about how your company can use IAM with DevOps Guru, see How Amazon DevOps Guru works with IAM.

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

Authenticating with identities

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

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

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

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

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

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account.

We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

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

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

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

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

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

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see 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:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

- **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see 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 AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see 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 AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

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

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

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A

user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see 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 an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see 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.

Change	Description	Date
AmazonDevOpsGuruConsoleFullAccess – Update to an existing policy.	The `AmazonDevOpsGuruFullAccess` managed policy now supports Amazon SNS subscriptions.	August 9, 2023
AmazonDevOpsGuruReadOnlyAccess – Update to an existing policy	The `AmazonDevOpsGuruReadOnlyAccess` managed policy now supports read-only access to Amazon SNS subscription lists.	August 9, 2023
AmazonDevOpsGuruServiceRolePolicy – Update to an existing policy.	The `AWSServiceRoleForDevOpsGuru` service-linked role now supports access to API Gateway GET actions on REST APIs.	January 11, 2023
AmazonDevOpsGuruServiceRolePolicy – Update to an existing policy.	The `AWSServiceRoleForDevOpsGuru` service-linked role now supports several Amazon Simple Storage Service and Service Quotas actions.	October 19, 2022
AmazonDevOpsGuruFullAccess – Update to an existing policy	The `AmazonDevOpsGuruFullAccess` managed policy now supports access to the CloudWatch `FilterLogEvents` action.	August 30, 2022
AmazonDevOpsGuruConsoleFullAccess – Update to an existing policy	The `AmazonDevOpsGuruConsoleFullAccess` managed policy now supports	August 30, 2022

Change	Description	Date
	access to the CloudWatch `FilterLogEvents` action.	
AmazonDevOpsGuruReadOnlyAccess – Update to an existing policy	The `AmazonDevOpsGuruReadOnlyAccess` managed policy now supports read-only access to the CloudWatch `FilterLogEvents` action.	August 30, 2022
AmazonDevOpsGuruServiceRolePolicy – Update to an existing policy.	The `AWSServiceRoleForDevOpsGuru` service-linked role now supports the CloudWatch logs actions `FilterLogEvents`, `DescribeLogGroups`, and `DescribeLogStreams`.	July 12, 2022
Identity-based policies for DevOps Guru – New managed policy.	The `AmazonDevOpsGuruConsoleFullAccess` policy has been added.	December 16, 2021
AmazonDevOpsGuruServiceRolePolicy – Update to an existing policy.	The `AWSServiceRoleForDevOpsGuru` service-linked role now supports Performance Insights `DescribeMetricsKeys`, and Amazon RDS `DescribeDBInstances` actions.	December 1, 2021

Change	Description	Date
AmazonDevOpsGuruReadOnlyAccess – Update to an existing policy	The `AmazonDevOpsGuruReadOnlyAccess` managed policy now supports read-only access to Amazon RDS `DescribeDBInstances` actions.	December 1, 2021
AmazonDevOpsGuruFullAccess – Update to an existing policy	The `AmazonDevOpsGuruFullAccess` managed policy now supports access to Amazon RDS `DescribeDBInstances` actions.	December 1, 2021
Identity-based policies for Amazon DevOps Guru – New policy added.	The `AWSServiceRoleForDevOpsGuru` service-linked role now supports access to Amazon RDS `DescribeDBInstances` and Performance Insights `GetResourceMetrics` actions. The `AmazonDevOpsGuruOrganizationsAccess` managed policy provides access to DevOps Guru within an organization.	November 16, 2021
AmazonDevOpsGuruServiceRolePolicy – Update to an existing policy.	The `AWSServiceRoleForDevOpsGuru` service-linked role now supports AWS Organizations.	November 4, 2021

Change	Description	Date
AmazonDevOpsGuruServiceRolePolicy – Update to an existing policy.	The `AWSServiceRoleForDevOpsGuru` service-linked role now contains new conditions on the `ssm:CreateOpsItem` and `ssm:AddTagsToResource` actions.	October 11, 2021
Service-linked role permissions for DevOps Guru – Update to an existing policy.	The `AWSServiceRoleForDevOpsGuru` service-linked role now contains new conditions on the `ssm:CreateOpsItem` and `ssm:AddTagsToResource` actions.	June 14, 2021
AmazonDevOpsGuruReadOnlyAccess – 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 – 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

Change	Description	Date
AmazonDevOpsGuruServiceRolePolicy – Update to an existing policy.	The `AWSServiceRoleForDevOpsGuru` 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	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys	Yes
ACLs	No
ABAC (tags in policies)	No

IAM feature	DevOps Guru support
Temporary credentials	Yes
Principal permissions	Yes
Service roles	No
Service-linked roles	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.

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 Cross account resource access in IAM 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:

```
aws
```

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

```
"Action": [
      "aws:action1",
```

```
      "aws:action2"
         ]
```

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

Policy resources for DevOps Guru

Supports policy resources: Yes

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

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

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

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

To see a list of 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.

Policy condition keys for DevOps Guru

Supports service-specific policy condition keys: Yes

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

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use

condition operators, such as equals or less than, to match the condition in the policy with values in the request.

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

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

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

To see a list of 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.

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.

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

For more information about ABAC, see 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 forward access sessions (FAS): Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to

complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for 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 AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

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

Identity-based policies for Amazon DevOps Guru

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

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

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

Topics

Policy best practices

Identity-based policies 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 with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and

functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.

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

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

Using the 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 (users or roles) with that policy.

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

To ensure that users and roles can still use the 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 for DevOps Guru actions and resources. You can attach these custom policies to the 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
- AmazonDevOpsGuruConsoleFullAccess
- AmazonDevOpsGuruReadOnlyAccess
- AmazonDevOpsGuruOrganizationsAccess

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": "DevOpsGuruFullAccess",
            "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",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SnsTopicOperations",
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:GetTopicAttributes",
                "sns:SetTopicAttributes",
                "sns:Subscribe",
                "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"
        },
```

```
        {
            "Sid": "RDSDescribeDBInstancesAccess",
            "Effect": "Allow",
            "Action": [
                "rds:DescribeDBInstances"
            ],
            "Resource": "*"
        },
        {
          "Sid": "CloudWatchLogsFilterLogEventsAccess",
          "Effect": "Allow",
          "Action": [
              "logs:FilterLogEvents"
          ],
          "Resource": "arn:aws:logs:*:*:log-group:*",
          "Condition": {
              "StringEquals": {
                  "aws:ResourceTag/DevOps-Guru-Analysis": "true"
              }
          }
        }
    ]
}
```

AmazonDevOpsGuruConsoleFullAccess

`AmazonDevOpsGuruConsoleFullAccess` – Provides full access to DevOps Guru, including permissions to create Amazon SNS topics, access Amazon CloudWatch metrics, and access AWS CloudFormation stacks. This policy has additional performance insights permissions so you can view detailed analysis related to anomalous Amazon RDS Aurora DB instances in the console. Apply this only to administrative-level users to whom you want to grant full control over DevOps Guru.

The `AmazonDevOpsGuruConsoleFullAccess` policy contains the following statement.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DevOpsGuruFullAccess",
            "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",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SnsTopicOperations",
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:GetTopicAttributes",
                "sns:SetTopicAttributes",
                "sns:Subscribe",
                "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"
        },
        {
            "Sid": "RDSDescribeDBInstancesAccess",
            "Effect": "Allow",
            "Action": [
                "rds:DescribeDBInstances"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PerformanceInsightsMetricsDataAccess",
            "Effect": "Allow",
            "Action": [
                "pi:GetResourceMetrics",
                "pi:DescribeDimensionKeys"
            ],
            "Resource": "*"
        },
        {
          "Sid": "CloudWatchLogsFilterLogEventsAccess",
          "Effect": "Allow",
          "Action": [
              "logs:FilterLogEvents"
          ],
          "Resource": "arn:aws:logs:*:*:log-group:*",
          "Condition": {
              "StringEquals": {
```

```
                    "aws:ResourceTag/DevOps-Guru-Analysis": "true"
                }
            }
        }
    ]
}
```

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": "DevOpsGuruReadOnlyAccess",
            "Effect": "Allow",
            "Action": [
                "devops-guru:DescribeAccountHealth",
                "devops-guru:DescribeAccountOverview",
                "devops-guru:DescribeAnomaly",
                "devops-guru:DescribeEventSourcesConfig",
                "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:ListAnomalousLogGroups",
                "devops-guru:ListMonitoredResources",
                "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": "*"
        },
        {
            "Sid": "RDSDescribeDBInstancesAccess",
            "Effect": "Allow",
            "Action": [
                "rds:DescribeDBInstances"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SnsListTopicsAccess",
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": "*"
```

```
        },
        {
          "Sid": "CloudWatchLogsFilterLogEventsAccess",
          "Effect": "Allow",
          "Action": [
              "logs:FilterLogEvents"
          ],
          "Resource": "arn:aws:logs:*:*:log-group:*",
          "Condition": {
              "StringEquals": {
                  "aws:ResourceTag/DevOps-Guru-Analysis": "true"
              }
          }
        }
    ]
}
```

AmazonDevOpsGuruOrganizationsAccess

`AmazonDevOpsGuruOrganizationsAccess` – Provides Organizations administrators access to the DevOps Guru multi-account view within an organization. Apply this policy to your organization's administrator-level users for whom you want to grant full access to DevOps Guru within an organization. You can apply this policy in your organization's management account and delegated administrator account for DevOps Guru. You can apply `AmazonDevOpsGuruReadOnlyAccess` or `AmazonDevOpsGuruFullAccess` in addition to this policy to provide read-only or full access to DevOps Guru.

The `AmazonDevOpsGuruOrganizationsAccess` policy contains the following statement.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AmazonDevOpsGuruOrganizationsAccess",
   "Effect": "Allow",
   "Action": [
    "devops-guru:DescribeOrganizationHealth",
    "devops-guru:DescribeOrganizationResourceCollectionHealth",
    "devops-guru:DescribeOrganizationOverview",
    "devops-guru:ListOrganizationInsights",
    "devops-guru:SearchOrganizationInsights"
   ],
```

```
    "Resource": "*"
   },
   {
    "Sid": "OrganizationsDataAccess",
    "Effect": "Allow",
    "Action": [
     "organizations:DescribeAccount",
     "organizations:DescribeOrganization",
     "organizations:ListAWSServiceAccessForOrganization",
     "organizations:ListAccounts",
     "organizations:ListChildren",
     "organizations:ListOrganizationalUnitsForParent",
     "organizations:ListRoots"
    ],
    "Resource": "arn:aws:organizations::*:"
   },
   {
    "Sid": "OrganizationsAdminDataAccess",
    "Effect": "Allow",
    "Action": [
     "organizations:DeregisterDelegatedAdministrator",
     "organizations:RegisterDelegatedAdministrator",
     "organizations:ListDelegatedAdministrators",
     "organizations:EnableAWSServiceAccess",
     "organizations:DisableAWSServiceAccess"
    ],
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "organizations:ServicePrincipal": [
       "devops-guru.amazonaws.com"
      ]
     }
    }
   }
  ]
}
```

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, AWS X-Ray, and AWS Organizations 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 `AWSServiceRoleForDevOpsGuru`. This is an AWS managed policy with scoped permissions that DevOps Guru needs to run in your account.

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

- `devops-guru.amazonaws.com`

The role permissions policy, `AmazonDevOpsGuruServiceRolePolicy` 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",
    "organizations:ListRoots",
    "organizations:ListChildren",
    "organizations:ListDelegatedAdministrators",
    "pi:GetResourceMetrics",
    "tag:GetResources",
    "lambda:GetFunction",
    "lambda:GetFunctionConcurrency",
    "lambda:GetAccountSettings",
    "lambda:ListProvisionedConcurrencyConfigs",
    "lambda:ListAliases",
    "lambda:ListEventSourceMappings",
    "lambda:GetPolicy",
    "ec2:DescribeSubnets",
    "application-autoscaling:DescribeScalableTargets",
    "application-autoscaling:DescribeScalingPolicies",
    "sqs:GetQueueAttributes",
    "kinesis:DescribeStream",
    "kinesis:DescribeLimits",
    "dynamodb:DescribeTable",
    "dynamodb:DescribeLimits",
    "dynamodb:DescribeContinuousBackups",
    "dynamodb:DescribeStream",
    "dynamodb:ListStreams",
    "elasticloadbalancing:DescribeLoadBalancers",
    "elasticloadbalancing:DescribeLoadBalancerAttributes",
    "rds:DescribeDBInstances",
    "rds:DescribeDBClusters",
    "rds:DescribeOptionGroups",
    "rds:DescribeDBClusterParameters",
    "rds:DescribeDBInstanceAutomatedBackups",
    "rds:DescribeAccountAttributes",
    "logs:DescribeLogGroups",
    "logs:DescribeLogStreams",
    "s3:GetBucketNotification",
    "s3:GetBucketPolicy",
```

```
    "s3:GetBucketPublicAccessBlock",
    "s3:GetBucketTagging",
    "s3:GetBucketWebsite",
    "s3:GetIntelligentTieringConfiguration",
    "s3:GetLifecycleConfiguration",
    "s3:GetReplicationConfiguration",
    "s3:ListAllMyBuckets",
    "s3:ListStorageLensConfigurations",
    "servicequotas:GetServiceQuota",
    "servicequotas:ListRequestedServiceQuotaChangeHistory",
    "servicequotas:ListServiceQuotas"
   ],
   "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": "*"
  },
  {
   "Sid": "AllowAddTagsToOpsItem",
   "Effect": "Allow",
   "Action": [
    "ssm:AddTagsToResource"
   ],
   "Resource": "arn:aws:ssm:*:*:opsitem/*"
  },
  {
   "Sid": "AllowAccessOpsItem",
   "Effect": "Allow",
   "Action": [
    "ssm:GetOpsItem",
```

```
    "ssm:UpdateOpsItem"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/DevOps-GuruInsightSsmOpsItemRelated": "true"
    }
   }
  },
  {
   "Sid": "AllowCreateManagedRule",
   "Effect": "Allow",
   "Action": "events:PutRule",
   "Resource": "arn:aws:events:*:*:rule/DevOpsGuruManagedRule*"
  },
  {
   "Sid": "AllowAccessManagedRule",
   "Effect": "Allow",
   "Action": [
    "events:DescribeRule",
    "events:ListTargetsByRule"
   ],
   "Resource": "arn:aws:events:*:*:rule/DevOpsGuruManagedRule*"
  },
  {
   "Sid": "AllowOtherOperationsOnManagedRule",
   "Effect": "Allow",
   "Action": [
    "events:DeleteRule",
    "events:EnableRule",
    "events:DisableRule",
    "events:PutTargets",
    "events:RemoveTargets"
   ],
   "Resource": "arn:aws:events:*:*:rule/DevOpsGuruManagedRule*",
   "Condition": {
    "StringEquals": {
     "events:ManagedBy": "devops-guru.amazonaws.com"
    }
   }
  },
  {
   "Sid": "AllowTagBasedFilterLogEvents",
   "Effect": "Allow",
```

```
    "Action": [
     "logs:FilterLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/DevOps-Guru-Analysis": "true"
     }
    }
   },
   {
    "Sid": "AllowAPIGatewayGetIntegrations",
    "Effect": "Allow",
    "Action": "apigateway:GET",
    "Resource": [
     "arn:aws:apigateway:*::/restapis/??????????",
     "arn:aws:apigateway:*::/restapis/*/resources",
     "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integration"
    ]
   }
  ]
}
```

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 `AWSServiceRoleForDevOpsGuru` 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 `AWSServiceRoleForDevOpsGuru` 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 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 Amazon SNS topics

Use the information in this topic only if you want to configure Amazon DevOps Guru to deliver notifications to Amazon SNS topics owned by another AWS account.

For DevOps Guru to deliver notifications to an Amazon SNS topic owned by a different account, you must attach a policy to the Amazon SNS topic that grants DevOps Guru permissions to send

notifications to it. If you configure DevOps Guru to deliver notifications to Amazon SNS topics owned by the same account you use for DevOps Guru, then DevOps Guru adds a policy to the topics for you.

After you attach a policy to configure permissions for an Amazon SNS topic in another account, you can add the Amazon SNS topic in DevOps Guru. You can also update your Amazon SNS policy with a notification channel to make it more secure.

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

Topics

Configuring permissions for an Amazon SNS topic in another account

Adding permissions as an IAM role

To use an Amazon SNS topic from another account after logging in with an IAM role, you must attach a policy to the Amazon SNS topic you want to use. To attach a policy to an Amazon SNS topic from another account while using an IAM role, you need to have the following permissions for that account resource as part of your IAM role:

- sns:CreateTopic
- sns:GetTopicAttributes
- sns:SetTopicAttributes
- sns:Publish

Attach the following policy to the Amazon SNS topic you want to use. 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-role`* is the IAM role of the individual user involved. You must substitute appropriate values for *`region-id`* (for example, `us-west-2`), and *`my-topic-name`*.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "EnableDevOpsGuruServicePrincipal",
            "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"
            },
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "topic-sender-account-id"
                }
            }
        },
        {
            "Sid": "EnableAccountPrincipal",
            "Action": "sns:Publish",
            "Effect": "Allow",
            "Resource": "arn:aws:sns:region-id:topic-owner-account-id:my-topic-name",
            "Principal": {
                "AWS": ["arn:aws:iam::topic-sender-account-id:role/devops-guru-role"]
            }
        }
    ]
}
```

Adding permissions as an IAM user

To use an Amazon SNS topic from another account as an IAM user, attach the following policy to the Amazon SNS topic you want to use. 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 involved. You must substitute appropriate values for *`region-id`* (for example, `us-west-2`) and *`my-topic-name`*.

> **Note**
>
> Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. For more

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

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "EnableDevOpsGuruServicePrincipal",
            "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"
            },
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "topic-sender-account-id"
                }
            }
        },
        {
            "Sid": "EnableAccountPrincipal",
            "Action": "sns:Publish",
            "Effect": "Allow",
            "Resource": "arn:aws:sns:region-id:topic-owner-account-id:my-topic-name",
            "Principal": {
                "AWS": ["arn:aws:iam::topic-sender-account-id:user/devops-guru-user-
name"]
            }
        }
    ]
}
```

Adding an Amazon SNS topic from another account

After you configure permissions for an Amazon SNS topic in another account, you can add that Amazon SNS topic to your DevOps Guru notification settings. You can add the Amazon SNS topic using the AWS CLI or the DevOps Guru console.

- When you use the console, you must select the option **Use an SNS topic ARN to specify an existing topic** in order to use a topic from another account.
- When you use the AWS CLI operation add-notification-channel, you must specify the `TopicArn` within the `NotificationChannelConfig` object.

Add an Amazon SNS topic from another account using the console

1. Open the Amazon DevOps Guru console at https://console.aws.amazon.com/devops-guru/.
2. Open the navigation pane, and then choose **Settings**.
3. Go to the **Notifications** section and choose **Edit**.
4. Choose **Add SNS topic**.
5. Choose **Use an SNS topic ARN to specify an existing topic**.
6. Enter the ARN of the Amazon SNS topic you want to use. You should have already configured permissions for this topic by attaching a policy to it.
7. (Optional) Choose **Notification configuration** to edit notification frequency settings.
8. Choose **Save**.

After you add the Amazon SNS topic to your notification settings, DevOps Guru uses that topic to notify you of important events, such as when a new insight is created.

Updating your Amazon SNS policy with a notification channel (recommended)

After you add a topic, we recommend that you make your policy more secure by specifying permissions for only the DevOps Guru notification channel that contains your topic.

Update your Amazon SNS topic policy with a notification channel (recommended)

1. Run the `list-notification-channels` DevOps Guru AWS CLI command in your account that you want to send notifications from.

   ```
   aws devops-guru list-notification-channels
   ```

2. In the `list-notification-channels` response, make a note of the channel ID that contains your Amazon SNS topic's ARN. The channel ID is a guid.

For example, in the following response, the channel ID for the topic with the ARN `arn:aws:sns:region-id:111122223333:topic-name` is `e89be5f7-989d-4c4c-b1fe-e7145037e531`

```
{
  "Channels": [
    {
      "Id": "e89be5f7-989d-4c4c-b1fe-e7145037e531",
      "Config": {
      "Sns": {
          "TopicArn": "arn:aws:sns:region-id:111122223333:topic-name"
        },
      "Filters": {
          "MessageTypes": ["CLOSED_INSIGHT", "NEW_INSIGHT", "SEVERITY_UPGRADED"],
          "Severities": ["HIGH", "MEDIUM"]
        }
      }
    }
  ]
}
```

3. Go to the policy that you created in another account using the topic owner ID in the section called “Configuring permissions for an Amazon SNS topic in another account”. In the `Condition` statement of the policy, add the line that specifies the `SourceArn`. The ARN contains your Region ID (for example, `us-east-1`), the AWS account number of the topic's sender, and the channel ID you made a note of.

   Your updated `Condition` statement looks like the following.

```
"Condition" : {
  "StringEquals" : {
    "AWS:SourceArn": "arn:aws:devops-guru:us-
east-1:111122223333:channel/e89be5f7-989d-4c4c-b1fe-e7145037e531",
    "AWS:SourceAccount": "111122223333"
   }
 }
```

If `AddNotificationChannel` is unable to add your SNS Topic, check that your IAM policy has the following permissions.

```
{
     "Version": "2012-10-17",
     "Statement": [{
           "Sid": "DevOpsGuruTopicPermissions",
           "Effect": "Allow",
           "Action": [
                 "sns:CreateTopic",
                 "sns:GetTopicAttributes",
                 "sns:SetTopicAttributes",
                 "sns:Publish"
           ],
           "Resource": "arn:aws:sns:region-id:account-id: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": "*"
    }
  ]
}
```

> **Note**
>
> DevOps Guru currently supports encrypted topics for use within a single account. Using an encrypted topic across multiple accounts is not supported at this time.

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

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance.

The following example error occurs when the user `mateojackson` tries to use the console to view details about a fictional *`my-example-widget`* resource but does not have the fictional `aws:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 aws: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 `aws:`*`GetWidget`* action.

I want to give users programmatic access

Users need programmatic access if they want to interact with AWS outside of the AWS Management Console. The way to grant programmatic access depends on the type of user that's accessing AWS.

To grant users programmatic access, choose one of the following options.

Which user needs programmatic access?	To	By
Workforce identity (Users managed in IAM Identity Center)	Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions for the interface that you want to use. • For the AWS CLI, see Configuring the AWS CLI to use AWS IAM Identity Center in the *AWS Command Line Interface User Guide*. • For AWS SDKs, tools, and AWS APIs, see IAM Identity Center authentication in the *AWS SDKs and Tools Reference Guide*.
IAM	Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions in Using temporary credentials with AWS resources in the *IAM User Guide*.
IAM	(Not recommended) Use long-term credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions for the interface that you want to use. • For the AWS CLI, see Authenticating using IAM user credentials in the *AWS Command Line Interface User Guide*. • For AWS SDKs and tools, see Authenticate using long-term credentials in

Which user needs programmatic access?	To	By
		the *AWS SDKs and Tools Reference Guide*. • For AWS APIs, see Managing access keys for IAM users in the *IAM User Guide*.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to 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 that are granted by a service role. Mary does not have permissions to pass the role to the service.

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

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

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

I want to allow people outside of my AWS account to access my 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.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

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
- Logging Amazon DevOps Guru API calls with AWS CloudTrail

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

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`

Metric	Description
	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` .
`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 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 AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access DevOps Guru through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

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

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 and limits 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

DevOps Guru resource monitoring limits

Resource description	Limit	Can be increased
Default limit for monitoring Amazon Simple Queue Service (Amazon SQS) queues	100*	Yes**

*For new DevOps Guru accounts created on or after June 29, 2023, and for existing accounts that were active as of the same date and have less than 100 Amazon SQS queues.

**To request a change in this limit, contact AWS Support at https://aws.amazon.com/contact-us. You can request an Amazon SQS queue monitoring limit of 100, 500, 1,000, 5,000, or 10,000.

DevOps Guru quotas for creating, deploying, and managing an API

The following fixed quotas apply to creating, deploying, and managing an API in DevOps Guru, using the AWS CLI, the API Gateway console, or the API Gateway REST API and its SDKs.

For a list of all DevOps Guru APIs, see Amazon DevOps Guru Actions.

Default quota	Can be increased	
20 requests every 1 second per account	Yes	

Amazon DevOps Guru document history

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

- **API version: latest**
- **Latest documentation update:** August 9, 2023

Change	Description	Date
Managed policy updates	Amazon SNS subscriptions and subscription list access have been added to the `AmazonDevOpsGuruConsoleFullAccess` policy. Subscription list access has also been added to the `AmazonDevOpsGuruReadOnlyAccess` policy. For more information, see Identity-based policies for Amazon DevOps Guru.	August 9, 2023
Customer managed encryption keys	DevOps Guru now supports encryption with customer managed keys using AWS KMS. For more information, see Data protection in DevOps Guru.	July 5, 2023
DevOps Guru for RDS supports RDS PostgreSQL	DevOps Guru for RDS can detect performance bottlenecks and other insights in PostgreSQL databases. For more information, see Benefits of DevOps Guru for RDS.	March 30, 2023

DevOps Guru for RDS supports proactive insights	DevOps Guru for RDS publishes proactive insights with recommendations to help you address issues in your Aurora databases before they become bigger problems. For more informati on, see Working with anomalies in DevOps Guru for RDS.	February 28, 2023
Analyzed resources page	A new page in the DevOps Guru console lists resources in your account that are analyzed by DevOps Guru. For more information, see Viewing resources analyzed by DevOps Guru.	October 20, 2022
New notification configura tion settings	You can now choose whether to receive all notifications or to only receive notificat ions for certain severities and events. For more informati on, see Updating Amazon Amazon SNS notification configurations.	September 30, 2022

Log anomaly analysis addition to managed policies	AWS managed policies for DevOps Guru ahave been updated in the IAM console to support access to the CloudWatch action `FilterLogEvents` . For more information, see DevOps Guru updates to AWS managed policies and service-linked role.	August 30, 2022
Log anomaly analysis added	You can view detailed information about log groups related to insights in the DevOps Guru console. There is also an expanded service-linked role available to d escribe CloudWatch logs and streams. For more informati on, see Understanding insights in the DevOps Guru console and DevOps Guru updates to AWS managed policies and service-linked role.	July 12, 2022
CodeGuru Profiler Integration	DevOps Guru now integrate s with Amazon CodeGuru Profiler with an EventBridge managed rule. Each inbound event from CodeGuru Profiler is a proactive anomaly report. For more information, see Integrating with CodeGuru Profiler.	March 7, 2022

Service-linked role and managed policy updates	Expanded policies available in the IAM console. The changes allow DevOps Guru to support enhanced integration with Amazon Relational Database Service (Amazon RDS). For more information, see Using service-linked roles and AWS managed (predefined) policies for DevOps Guru.	December 21, 2021
New managed policy added	The `AmazonDevOpsGuruConsoleFullAccess` policy has been added. For more information, see Identity-based policies for Amazon DevOps Guru.	December 6, 2021
Support to define your application with AWS tags	You can now use AWS tags to identify the resources you want DevOps Guru to analyze, identify the resources in your applications, and filter insights in the console. For more information, see Use tags to identify resources in your applications.	December 1, 2021

Service-linked role and managed policy updates	Expanded policies available in the IAM console. The changes allow DevOps Guru to support enhanced integration with Amazon Relational Database Service (Amazon RDS). For more information, see Using service-linked roles and AWS managed (predefined) policies for DevOps Guru.	December 1, 2021
Amazon RDS support	DevOps Guru now provides comprehensive analysis and insights for Amazon Relational Database Service (Amazon RDS) resources in your application. For more information, see Working with anomalies in DevOps Guru for Amazon RDS.	December 1, 2021
Amazon EventBridge integration	DevOps Guru now integrates with EventBridge to notify you of certain events relating to your DevOps Guru insights. For more information, see Working with EventBridge.	November 18, 2021
AWS managed policy added	New AWS managed policy added. The `AmazonDevOpsGuruOrganizationsAccess` policy provides access to DevOps Guru within an organization. For more information, see identity-based policies.	November 16, 2021

Service-linked role policy update	Expanded policy available in the IAM console. The change allows DevOps Guru to s upport the multi account view. For more informati on, see Using service-linked roles.	November 4, 2021
Cross account support	You can now view insights and metrics across multiple accounts in your organiza tion. For more informati on, see What is Amazon DevOps Guru.	November 4, 2021
General availability release	Amazon DevOps Guru is now generally available (GA).	May 4, 2021
New topic	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	You can now use VPC endpoints to improve the security of your resource anal ysis and insight generation. For more information, see DevOps Guru and interface VPC endpoints (AWS PrivateLi nk).	April 15, 2021

New topic	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	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 Glossary Reference*.