User Guide

Developer Tools console

Developer Tools console: User Guide

Table of Contents

What is the Developer Tools console?

The Developer Tools console is home to a set of services and features that you can use individually or collectively to help you develop software, either individually or as a team. The developer tools can help you securely store, build, test, and deploy your software. Used individually or collectively, these tools provide support for DevOps, continuous integration, and continuous delivery (CI/CD).

The Developer Tools console includes the following services:

- AWS CodeCommit is a fully managed source control service that hosts private Git repositories. You can use repositories to privately store and manage assets (such as documents, source code, and binary files) in the AWS Cloud. Your repositories store your project history from the first commit through the latest changes. You can work collaboratively on code in repositories by commenting on code and creating pull requests to help ensure code quality.

- AWS CodeBuild is a fully managed build service that compiles your source code, runs unit tests, and produces artifacts that are ready to deploy. It provides prepackaged build environments for popular programming languages and build tools such as Apache Maven, Gradle, and more. You can also customize build environments in CodeBuild to use your own build tools.

- AWS CodeDeploy is a fully managed deployment service that automates software deployments to compute services such as Amazon EC2, AWS Lambda, and your on-premises servers. It can help you rapidly release new features, avoid downtime during application deployment, and handle the complexity of updating your applications.

- AWS CodePipeline is a continuous integration and continuous delivery service you can use to model, visualize, and automate the steps required to release your software. You can quickly model and configure the different stages of a software release process. You can build, test, and deploy your code every time there is a code change, based on the release process models you define.

Here's an example of how you can use the services in the Developer Tools console together to help you develop software.

An Example CI/CD Pipeline in CodePipeline
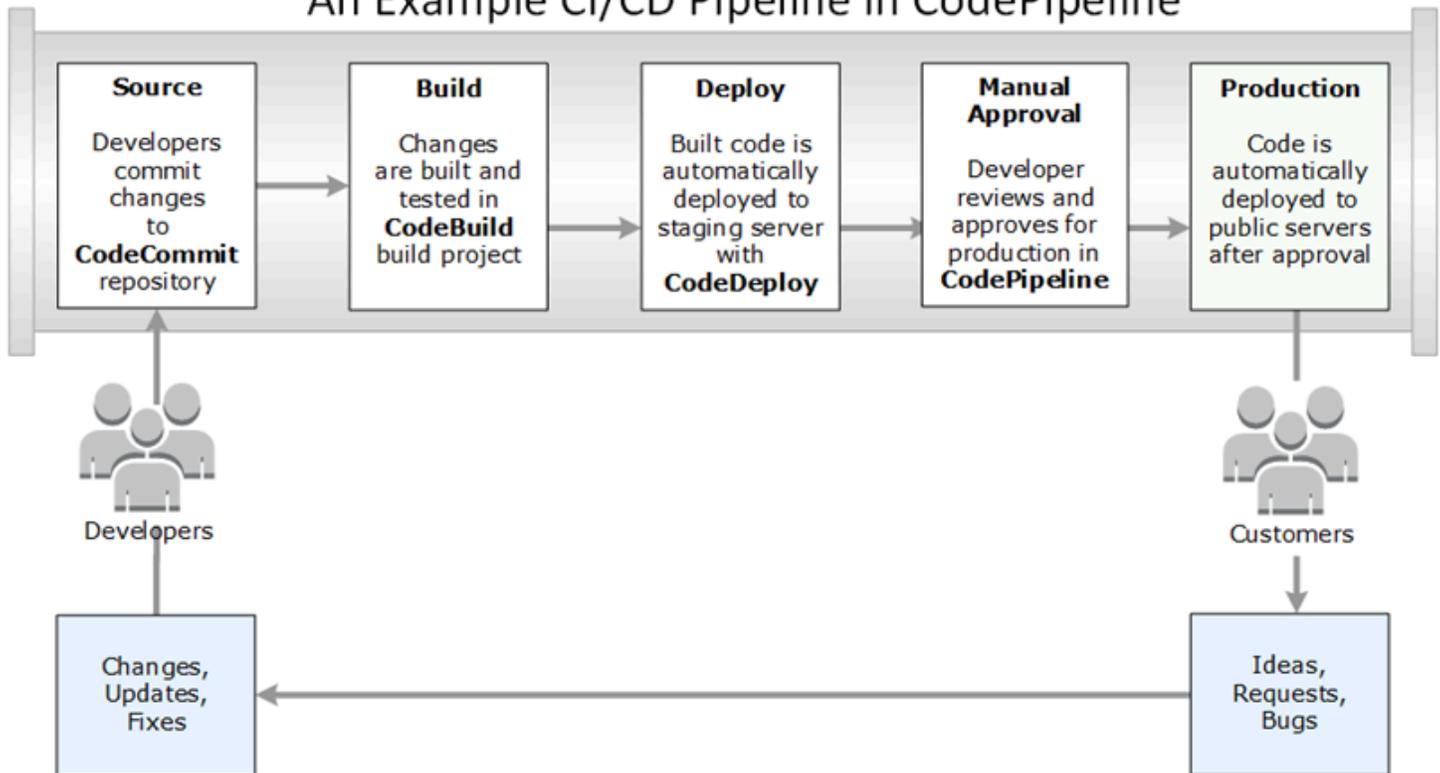
In this example, developers create a repository in CodeCommit and use it to develop and collaborate on their code. They create a build project in CodeBuild to build and test their code, and use CodeDeploy to deploy their code to test and production environments. They want to iterate quickly, so they create a pipeline in CodePipeline to detect the changes in the CodeCommit repository. Those changes are built, tests are run, and successfully built and tested code is deployed to the test server. The team adds test stages to the pipeline to run more tests on the staging server, such as integration or load tests. Upon the successful completion of those tests, a team member reviews the results and if satisfied, manually approves the changes for production. CodePipeline deploys the tested and approved code to production instances.

This is just one simple example of how you can use one or more of the services available in the Developer Tools console to help you develop software. Each of the services can be customized to meet your needs. They offer many integrations with other products and services, both in AWS and with other third-party tools. For more information, see the following topics:

- CodeCommit: Product and service integrations
- CodeBuild: Use CodeBuild with Jenkins
- CodeDeploy: Product and service integrations

- CodePipeline: [Product and service integrations](#)

Are you a first-time user?

If you are a first-time user of one or more of the services available in the Developer Tools console, we recommend that you begin by reading the following topics:

- [Getting started with CodeCommit](#)

- [Getting started with CodeBuild](#), [Concepts](#)

- [Getting started with CodeDeploy](#), [Primary components](#)

- [Getting started with CodePipeline](#), [Concepts](#)

Features of the developer tools console

The Developer Tools console includes the following features:

- The Developer Tools console includes a notifications manager feature that you can use to subscribe to events in AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, and AWS CodePipeline. This feature has its own API, AWS CodeStar Notifications. You can use the notifications feature to quickly notify users about events in the repositories, build projects, deployment applications, and pipelines that are most important to their work. A notifications manager helps make users aware of events that occur on repositories, builds, deployments, or pipelines so that they can quickly take action, such as approving changes or correcting errors. For more information, see [What are notifications?](#)

- The Developer Tools console includes a connections feature that you can use to associate your AWS resources with third-party source code providers. This feature has its own API, AWS CodeConnections. You can use the connections feature to set up an authorized connection with a third-party provider and use the connection resource with other AWS services. For more information, see [What are connections?](#)

What are notifications?

The notifications feature in the Developer Tools console is a notifications manager for subscribing to events in AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy and AWS CodePipeline. It has its own API, AWS CodeStar Notifications. You can use the notifications feature to quickly notify

users about events in the repositories, build projects, deployment applications, and pipelines that are most important to their work. A notifications manager helps make users aware of events that occur on repositories, builds, deployments, or pipelines so that they can quickly take action, such as approving changes or correcting errors.

What can I do with notifications?

You can use the notifications feature to create and manage notification rules to notify users of important changes to their resources, including:

- Build successes and failures in CodeBuild build projects.

- Deployment successes and failures in CodeDeploy applications.

- Creation of and updates in pull requests, including comments on code, in CodeCommit repositories.

- Manual approval statuses and pipeline runs in CodePipeline.

You can set up notifications so that they go to user email addresses that are subscribed to an Amazon SNS topic. You can also integrate this feature with AWS Chatbot and have notifications delivered to Slack channels, Microsoft Teams channel, or Amazon Chime chatrooms.

How do notifications work?

When you configure a notification rule for a supported resource, such as a repository, build project, application, or pipeline, the notifications feature creates an Amazon EventBridge rule that monitors for the events you specify. When an event of that type occurs, the notification rule sends notifications to the Amazon SNS topics specified as targets for that rule. Subscribers to those targets receive notifications about those events.

How do I get started with notifications?

To get started, here are some useful topics to review:

- **Learn** about the concepts for notifications.

- **Set up** the resources you need to start working with notifications.

- **Get started** with your first notification rules and receive your first notifications.

Notification concepts

Setting up and using notifications is easier if you understand the concepts and terms. Here are some concepts to know about as you use notifications.

Topics

- [Notifications](#)
- [Notification rules](#)
- [Events](#)
- [Detail types](#)
- [Targets](#)
- [Notifications and AWS CodeStar Notifications](#)
- [Events for notification rules on repositories](#)
- [Events for notification rules on build projects](#)
- [Events for notification rules on deployment applications](#)
- [Events for notification rules on pipelines](#)

Notifications

A *notification* is a message that contains information about events that occur in the resources you and your developers use. You can set up notifications so that users of a resource, such as a build project, repository, deployment application, or pipeline, receive emails about the event types you specify according to the notification rule you create.

Notifications for AWS CodeCommit can contain user identity information, such as a display name or an email address, through the use of session tags. CodeCommit supports the use of session tags, which are key-value pair attributes that you pass when you assume an IAM role, use temporary credentials, or federate a user in AWS Security Token Service (AWS STS). You can also associate tags with an IAM user. CodeCommit includes the values for `displayName` and `emailAddress` in notification content if those tags are present. For more information, see [Using tags to provide additional identity information in CodeCommit](#).

> ⚠️ **Important**
>
> Notifications include project-specific information such as build status, deployment status, lines of code that have comments, and pipeline approvals. Notification content might

change as new features are added. As a security best practice, you should regularly review the targets of notification rules and the Amazon SNS topic subscribers. For more information, see [Understanding notification contents and security](#).

Notification rules

A *notification rule* is an AWS resource that you create to specify when and where notifications are sent. It defines:

- The conditions under which a notification is created. These conditions are based on events that you choose, which are specific to the resource type. Supported resource types include build projects in AWS CodeBuild, deployment applications in AWS CodeDeploy, pipelines in AWS CodePipeline, and repositories in AWS CodeCommit.
- The targets to which the notification is sent. You can specify up to 10 targets for a notification rule.

Notification rules are scoped to individual build projects, deployment applications, pipelines, and repositories. Notification rules have both user-defined friendly names and Amazon Resource Names (ARNs). Notification rules must be created in the same AWS Region where the resource exists. For example, if your build project is in the US East (Ohio) Region, your notification rule must be created in the US East (Ohio) Region, too.

You can define up to 10 notification rules for a resource.

Events

An *event* is a change of state on a resource that you want to monitor. Each resource has a list of event types you can choose from. When you set up a notification rule on a resource, you specify the events that cause notifications to be sent. For example, if you set up notifications for a repository in CodeCommit, and you select **Created** for both **Pull request** and **Branches and tags**, a notification is sent every time a user in that repository creates a pull request, branch, or Git tag.

Detail types

When you create a notification rule, you can choose the level of detail or *detail type* included in notifications (**Full** or **Basic**). The **Full** setting (the default) includes all information available for the event in the notification, including any enhanced information provided by services for specific events. The **Basic** setting includes only a subset of the available information.

The following table lists the enhanced information available for specific event types and describes the differences between the detail types.

Service	Event	Full includes	Basic does not include
CodeCommit	Comments on commits Comments on pull requests	All event details and the content of the comment, including any replies or comment threads. It also includes the line number and the line of code upon which the comment was made.	The content of the comment. line number, line of code, or any comment threads.
CodeCommit	Pull request created	All event details and the number of files that were added, modified, or deleted in the pull request in relation to the destination branch.	No list of files or details about whether the pull request source branch has added, modified, or deleted files.
CodePipeline	Manual approval needed	All event details and custom data (if configured). The notification also includes a link to the required approval in the pipeline.	No custom data or link.
CodePipeline	Action execution failed Pipeline execution failed	All event details and the content of the error message for the failure.	No error message content.

Service	Event	Full includes	Basic does not include
	Stage execution failed		

Targets

A *target* is a location for receiving notifications from notification rules. The allowed target types are Amazon SNS topics and AWS Chatbot clients configured for Slack or Microsoft Teams channels. Any user subscribed to the target receives notifications about the events that you specify in the notification rule.

If you want to extend the reach of notifications, you can manually configure integration between notifications and AWS Chatbot so that notifications are sent to Amazon Chime chatrooms. You can then choose the Amazon SNS topic that is configured for that AWS Chatbot client as the target for the notification rule. For more information, see To integrate notifications with AWS Chatbot and Amazon Chime.

If you choose to use an AWS Chatbot client as a target, you must first create that client in AWS Chatbot. When you choose an AWS Chatbot client as a target for a notification rule, an Amazon SNS topic is configured for that AWS Chatbot client with all the policies required for notifications to be sent to the Slack or Microsoft Teams channel. You don't have to configure any existing Amazon SNS topics for the AWS Chatbot client.

You can choose to create an Amazon SNS topic as a target as part of creating a notification rule (recommended). You can also choose an existing Amazon SNS topic in the same AWS Region as the notification rule, but you must configure it with the required policy. The Amazon SNS topic that you use for a target must be in your AWS account. It also must be in the same AWS Region as the notification rule and the AWS resource for which the rule was created.

For example, if you create a notification rule for a repository in the US East (Ohio) Region, the Amazon SNS topic must also exist in that Region. If you create an Amazon SNS topic as part of creating a notification rule, the topic is configured with the policy required to allow the publication of events to the topic. This is the best method for working with targets and notification rules. If you choose to use an already-existing topic or create one manually, you must configure it with the required permissions before users receive notifications. For more information, see Configure Amazon SNS topics for notifications.

> ⓘ **Note**
>
> If you want to use an existing Amazon SNS topic instead of creating a new one, in **Targets**, choose its ARN. Make sure the topic has the appropriate access policy, and that the subscriber list contains only those users who are allowed to see information about the resource. If the Amazon SNS topic is a topic that was used for CodeCommit notifications before November 5, 2019, it will contain a policy that allows CodeCommit to publish to it that contains different permissions than those required for AWS CodeStar Notifications. Using these topics is not recommended. If you want to use one created for that experience, you must add the required policy for AWS CodeStar Notifications in addition to the one that already exists. For more information, see [Configure Amazon SNS topics for notifications](#) and [Understanding notification contents and security](#).

Notifications and AWS CodeStar Notifications

While a feature of the Developer Tools console, notifications has its own API, AWS CodeStar Notifications. It also has its own AWS resource type (notification rules), permissions, and events. Events for notification rules are logged in AWS CloudTrail. API actions can be allowed or denied through IAM policies.

Events for notification rules on repositories

Category	Events	Event IDs

Comments	On commits
	
	On pull requests
		`codecommit-repository-comments-on-commits`
		
		`codecommit-repository-comments-on-pull-requests`

Approvals	Status changed
	
	Rule override
		`codecommit-repository-approvals-status-changed`
		
		`codecommit-repository-approvals-rule-override`

Category	Events	Event IDs

Pull request	Created
	
	Source updated
	
	Status changed
	
	Merged
		`codecommit-repository-pull-request-created`
		
		`codecommit-repository-pull-request-source-updated`
		
		`codecommit-repository-pull-request-status-changed`
		
		`codecommit-repository-pull-request-merged`

Branches and tags	Created
	
	Deleted
	
	Updated
		`codecommit-repository-branches-and-tags-created`
		
		`codecommit-repository-branches-and-tags-deleted`
		`codecommit-repository-branches-and-tags-updated`

Events for notification rules on build projects

Category	Events	Event IDs

Build state	Failed
	
	Succeeded
	
	In-progress
	
	Stopped
		`codebuild-project-build-state-failed`
		
		`codebuild-project-build-state-succeeded`

Category	Events	Event IDs

		`codebuild-project-build-state-in-progress`
		
		 `codebuild-project-build-state-stopped`

Build phase	Failure
	
	 Success
		`codebuild-project-build-phase-failure`
		
		 `codebuild-project-build-phase-success`

Events for notification rules on deployment applications

Category	Events	Event IDs

Deployment	Failed
	
	 Succeeded
	
	 Started
		`codedeploy-application-deployment-failed`
		
		 `codedeploy-application-deployment-succeeded`
		 `codedeploy-application-deployment-started`

Events for notification rules on pipelines

Category	Events	Event IDs

Action execution	Succeeded
	
	 Failed
	
	 Canceled
	
	 Started
		`codepipeline-pipeline-action-execution-succeeded`
		
		 `codepipeline-pipeline-action-execution-failed`

Category	Events	Event IDs

		`codepipeline-pipeline-actio`
		`n-execution-canceled`
		
		`codepipeline-pipeline-actio`
		`n-execution-started`

Stage execution	Started
	
	Succeeded
	
	Resumed
	
	Canceled
	
	Failed
		`codepipeline-pipeline-stage`
		`-execution-started`
		
		`codepipeline-pipeline-stage`
		`-execution-succeeded`
		
		`codepipeline-pipeline-stage`
		`-execution-resumed`
		
		`codepipeline-pipeline-stage`
		`-execution-canceled`
		
		`codepipeline-pipeline-stage`
		`-execution-failed`

Pipeline execution	Failed
	
	Canceled
	
	Started
	
	Resumed
	
	Succeeded
	
	Superseded
		`codepipeline-pipeline-pipel`
		`ine-execution-failed`
		
		`codepipeline-pipeline-pipel`
		`ine-execution-canceled`
		
		`codepipeline-pipeline-pipel`
		`ine-execution-started`
		
		`codepipeline-pipeline-pipel`
		`ine-execution-resumed`
		
		`codepipeline-pipeline-pipel`
		`ine-execution-succeeded`
		
		`codepipeline-pipeline-pipel`
		`ine-execution-superseded`

Category	Events	Event IDs

Manual approval	Failed
	
	Needed
	
	Succeeded
		`codepipeline-pipeline-manual-approval-failed`
		
		`codepipeline-pipeline-manual-approval-needed`
		`codepipeline-pipeline-manual-approval-succeeded`

Setting up

If you have a managed policy for AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, or AWS CodePipeline applied to your IAM user or role, you have the permissions required to work with notifications within the limitations of the roles and permissions provided by the policy. For example, users who have the `AWSCodeBuildAdminAccess`, `AWSCodeCommitFullAccess`, `AWSCodeDeployFullAccess`, or `AWSCodePipeline_FullAccess` managed policy applied have full administrative access to notifications.

For more information, including example policies, see Identity-based policies.

If you have one of these policies applied to your IAM user or role, and a build project in CodeBuild, a repository in CodeCommit, a deployment application in CodeDeploy, or a pipeline in CodePipeline, you are ready to create your first notification rule. Continue to Getting started with notifications. If not, see the following topics:

- CodeBuild: Getting started with CodeBuild
- CodeCommit: Getting started with CodeCommit
- CodeDeploy: Tutorials
- CodePipeline: Getting started with CodePipeline

If you want to manage administrative permissions for notifications for IAM users, groups, or roles yourself, follow the procedures in this topic to set up the permissions and resources you need to use the service.

If you want to use previously created Amazon SNS topics for notifications instead of creating topics specifically for notifications, you must configure an Amazon SNS topic to use as the target for a notification rule by applying a policy that allows events to be published to that topic.

> ⓘ **Note**
>
> To perform the following procedures, you must be signed in with an account that has administrative permissions. For more information, see [Creating your first IAM admin user and group](#).

Topics

- [Create and apply a policy for administrative access to notifications](#)
- [Configure Amazon SNS topics for notifications](#)
- [Subscribe users to Amazon SNS topics that are targets](#)

Create and apply a policy for administrative access to notifications

You can administer notifications by signing in with an IAM user or using a role that has permissions to access the service and the services (AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, or AWS CodePipeline) for which you want to create notifications. You can also create your own policies and apply them to users or groups.

The following procedure shows you how to configure an IAM group with permissions for administering notifications and adding IAM users. If you do not want to set up a group, you can apply this policy directly to IAM users or to an IAM role that can be assumed by users. You can also use the managed policies for CodeBuild, CodeCommit, CodeDeploy, or CodePipeline, which include policy-appropriate access to notification features depending on the scope of the policy.

For the policy below, enter a name (for example, `AWSCodeStarNotificationsFullAccess`) and an optional description for this policy. The description helps you remember the purpose of the policy (for example, **This policy provides full access to AWS CodeStar Notifications.**

To use the JSON policy editor to create a policy

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.

2. In the navigation pane on the left, choose **Policies**.

   If this is your first time choosing **Policies**, the **Welcome to Managed Policies** page appears. Choose **Get Started**.

3. At the top of the page, choose **Create policy**.

4. In the **Policy editor** section, choose the **JSON** option.

5. Enter the following JSON policy document:

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "AWSCodeStarNotificationsFullAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:CreateNotificationRule",
            "codestar-notifications:DeleteNotificationRule",
            "codestar-notifications:DescribeNotificationRule",
            "codestar-notifications:ListNotificationRules",
            "codestar-notifications:UpdateNotificationRule",
            "codestar-notifications:Subscribe",
            "codestar-notifications:Unsubscribe",
            "codestar-notifications:DeleteTarget",
            "codestar-notifications:ListTargets",
            "codestar-notifications:ListTagsforResource",
            "codestar-notifications:TagResource",
            "codestar-notifications:UntagResource"
        ],
        "Resource": "*"
      }
    ]
}
```

6. Choose **Next**.

> ⓘ **Note**
>
> You can switch between the **Visual** and **JSON** editor options anytime. However, if you make changes or choose **Next** in the **Visual** editor, IAM might restructure your policy to

optimize it for the visual editor. For more information, see [Policy restructuring](#) in the *IAM User Guide*.

7. On the **Review and create** page, enter a **Policy name** and a **Description** (optional) for the policy that you are creating. Review **Permissions defined in this policy** to see the permissions that are granted by your policy.

8. Choose **Create policy** to save your new policy.

Configure Amazon SNS topics for notifications

The easiest way to set up notifications is to create an Amazon SNS topic when you create a notification rule. You can use an existing Amazon SNS topic if it meets the following requirements:

- It was created in the same AWS Region as the resource (build project, deployment application, repository, or pipeline) for which you want to create notification rules.

- It has not been used for sending notifications for CodeCommit before November 5, 2019. If it has, it will contain policy statements that enabled that functionality. You can choose to use this topic, but you will need to add the additional policy as specified in the procedure. You should not remove the existing policy statement if one or more repositories is still configured for notifications before November 5, 2019.

- It has a policy that allows AWS CodeStar Notifications to publish notifications to the topic.

To configure an Amazon SNS topic to use as a target for AWS CodeStar Notifications notification rules

1. Sign in to the AWS Management Console and open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.

2. In the navigation bar, choose **Topics**, choose the topic you want to configure, and then choose **Edit**.

3. Expand **Access policy**, and then choose **Advanced**.

4. In the JSON editor, add the following statement to the policy. Include the topic ARN, AWS Region, AWS account ID, and topic name.

```
{
    "Sid": "AWSCodeStarNotifications_publish",
    "Effect": "Allow",
```

```
      "Principal": {
        "Service": [
          "codestar-notifications.amazonaws.com"
        ]
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-
  MyTopicForNotificationRules"
      }
```

The policy statement should look like the following.

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "SNS:AddPermission",
        "SNS:RemovePermission",
        "SNS:DeleteTopic",
        "SNS:Subscribe",
        "SNS:ListSubscriptionsByTopic",
        "SNS:Publish"
      ],
      "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-
  MyTopicForNotificationRules",
      "Condition": {
        "StringEquals": {
          "AWS:SourceOwner": "123456789012"
        }
      }
    },
  {
      "Sid": "AWSCodeStarNotifications_publish",
      "Effect": "Allow",
```

```
      "Principal": {
        "Service": [
          "codestar-notifications.amazonaws.com"
        ]
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-
  MyTopicForNotificationRules"
    }
  ]
}
```

5. Choose **Save changes**.

6. If you want to use an AWS KMS-encrypted Amazon SNS topic to send notifications, you must also enable compatibility between the event source (AWS CodeStar Notifications) and the encrypted topic by adding the following statement to the policy of the AWS KMS key. Replace the AWS Region (in this example, us-east-2) with the AWS Region where the key was created.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "codestar-notifications.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "sns.us-east-2.amazonaws.com"
                }
            }
        }
    ]
}
```

For more information, see [Encryption at rest](#) and [Using policy conditions with AWS KMS](#) in the *AWS Key Management Service Developer Guide*.

Subscribe users to Amazon SNS topics that are targets

Before users can receive notifications, they must be subscribed to the Amazon SNS topic that is the target of the notification rule. If users are subscribed by email address, they must confirm their subscription before they receive notifications. To send notifications to users in Slack channels, Microsoft Teams channels, or Amazon Chime chatrooms, see [Configure integration between notifications and AWS Chatbot](#).

To subscribe users to an Amazon SNS topic used for notifications

1. Sign in to the AWS Management Console and open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.

2. In the navigation bar, choose **Topics**, and then choose the topic to which you want to subscribe users.

3. In **Subscriptions**, choose **Create subscription**.

4. In **Protocol**, choose **Email**. In **Endpoint**, enter the email address, and then choose **Create subscription**.

Getting started with notifications

The easiest way to get started with notifications is to set up a notification rule on one of your build projects, deployment applications, pipelines, or repositories.

> ⓘ **Note**
>
> The first time you create a notification rule, a service-linked role is created in your account. For more information, see [Using service-linked roles for AWS CodeStar Notifications](#).

Topics

- [Prerequisites](#)
- [Create a notification rule for a repository](#)

- Create a notification rule for a build project
- Create a notification rule for a deployment application
- Create a notification rule for a pipeline

Prerequisites

Complete the steps in Setting up. You also need a resource for which you create a notification rule.

- Create a build project in CodeBuild or use an existing one.
- Create an application or use an existing deployment application.
- Create a pipeline in CodePipeline or use an existing one.
- Create an AWS CodeCommit repository or use an existing one.

Create a notification rule for a repository

You can create notification rules to send notifications about repository events that are important to you. The following steps show you how to set up a notification rule on a single repository event. These steps are written with the assumption that you have a repository configured in your AWS account.

> ⚠️ **Important**
>
> If you set up notifications in CodeCommit before November 5, 2019, the Amazon SNS topics used for those notifications will contain a policy that allows CodeCommit to publish to it that contains different permissions than those required for AWS CodeStar Notifications. Using these topics is not recommended. If you want to use one created for that experience, you must add the required policy for AWS CodeStar Notifications in addition to the one that already exists. For more information, see Configure Amazon SNS topics for notifications and Understanding notification contents and security.

1. Open the CodeCommit console at https://console.aws.amazon.com/codecommit/.
2. Choose a repository from the list and open it.
3. Choose **Notify**, and then choose **Create notification rule**. You can also choose **Settings**, choose **Notifications**, and then choose **Create notification rule**.
4. In **Notification name**, enter a name for the rule.

5. In **Detail type**, choose **Basic** if you want only the information provided to Amazon EventBridge included in the notification. Choose **Full** if you want to include information provided to Amazon EventBridge and information that might be supplied by the resource service or the notification manager.

    For more information, see [Understanding notification contents and security](#).

6. In **Events that trigger notifications**, under **Branches and tags**, select **Created**.

7. In **Targets**, choose **Create SNS topic**.

> ⓘ **Note**
>
> When you create the topic as part of creating the notification rule, the policy that allows CodeCommit to publish events to the topic is applied for you. Using a topic created for notification rules helps ensure that you subscribe only those users who you want to receive notifications about this repository.

    After the **codestar-notifications-** prefix,enter a name for the topic, and then choose **Submit**.

> ⓘ **Note**
>
> If you want to use an existing Amazon SNS topic instead of creating a new one, in **Targets**, choose its ARN. Make sure the topic has the appropriate access policy, and that the subscriber list contains only those users who are allowed to see information about the resource. If the Amazon SNS topic is a topic that was used for CodeCommit notifications before November 5, 2019, it will contain a policy that allows CodeCommit to publish to it that contains different permissions than those required for AWS CodeStar Notifications. Using these topics is not recommended. If you want to use one created for that experience, you must add the required policy for AWS CodeStar Notifications in addition to the one that already exists. For more information, see [Configure Amazon SNS topics for notifications](#) and [Understanding notification contents and security](#).

8. Choose **Submit**, and then review the notification rule.

9. Subscribe your email address to the Amazon SNS topic you just created. For more information, see [To subscribe users to an Amazon SNS topic used for notifications](#).

10. Navigate to your repository and create a test branch from the default branch.

11. After you create the branch, the notification rule sends a notification to all topic subscribers with information about that event.

Create a notification rule for a build project

You can create notification rules to send notifications about the events on your build project that are important to you. The following steps show you how to set up a notification rule on a single build project event. These steps are written with the assumption that you have a build project configured in your AWS account.

1. Open the CodeBuild console at https://console.aws.amazon.com/codebuild/.

2. Choose a build project from the list and open it.

3. Choose **Notify**, and then choose **Create notification rule**. You can also choose **Settings**, and then choose **Create notification rule**.

4. In **Notification name**, enter a name for the rule.

5.
   In **Detail type**, choose **Basic** if you want only the information provided to Amazon EventBridge included in the notification. Choose **Full** if you want to include information provided to Amazon EventBridge and information that might be supplied by the resource service or the notification manager.

   For more information, see Understanding notification contents and security.

6. In **Events that trigger notifications**, under **Build phase**, select **Success**.

7. In **Targets**, choose **Create SNS topic**.

   > **ⓘ Note**
   >
   > When you create the topic as part of creating the notification rule, the policy that allows CodeBuild to publish events to the topic is applied for you. Using a topic created for notification rules helps ensure that you subscribe only those users you want to receive notifications about this build project.

   After the **codestar-notifications-** prefix, enter a name for the topic, and then choose **Submit**.

> **ⓘ Note**
>
> If you want to use an existing Amazon SNS topic instead of creating a new one, in **Targets**, choose its ARN. Make sure the topic has the appropriate access policy, and that the subscriber list contains only those users who are allowed to see information about the resource. If the Amazon SNS topic is a topic that was used for CodeCommit notifications before November 5, 2019, it will contain a policy that allows CodeCommit to publish to it that contains different permissions than those required for AWS CodeStar Notifications. Using these topics is not recommended. If you want to use one created for that experience, you must add the required policy for AWS CodeStar Notifications in addition to the one that already exists. For more information, see Configure Amazon SNS topics for notifications and Understanding notification contents and security.

8. Choose **Submit**, and then review the notification rule.

9. Subscribe your email address to the Amazon SNS topic you just created. For more information, see To subscribe users to an Amazon SNS topic used for notifications.

10. Navigate to your build project and start a build.

11. After the build phase is successfully completed, the notification rule sends a notification to all topic subscribers with information about that event.

Create a notification rule for a deployment application

You can create notification rules to send notifications about the events on your deployment application that are important to you. The following steps show you how to set up a notification rule on a single build project event. These steps are written with the assumption that you have a deployment application configured in your AWS account.

1. Open the CodeDeploy console at https://console.aws.amazon.com/codedeploy/.

2. Choose an application from the list and open it.

3. Choose **Notify**, and then choose **Create notification rule**. You can also choose **Settings**, and then choose **Create notification rule**.

4. In **Notification name**, enter a name for the rule.

5. In **Detail type**, choose **Basic** if you want only the information provided to Amazon EventBridge included in the notification. Choose **Full** if you want to include information provided to

Amazon EventBridge and information that might be supplied by the resource service or the notification manager.

For more information, see [Understanding notification contents and security](#).

6. In **Events that trigger notifications**, under **Deployment**, select **Succeeded**.

7. In **Targets**, choose **Create SNS topic**.

> **ⓘ Note**
>
> When you create the topic as part of creating the notification rule, the policy that allows CodeDeploy to publish events to the topic is applied for you. Using a topic created for notification rules helps ensure that you subscribe only those users you want to receive notifications about this deployment application.

After the **codestar-notifications-** prefix, enter a name for the topic, and then choose **Submit**.

> **ⓘ Note**
>
> If you want to use an existing Amazon SNS topic instead of creating a new one, in **Targets**, choose its ARN. Make sure the topic has the appropriate access policy, and that the subscriber list contains only those users who are allowed to see information about the resource. If the Amazon SNS topic is a topic that was used for CodeCommit notifications before November 5, 2019, it will contain a policy that allows CodeCommit to publish to it that contains different permissions than those required for AWS CodeStar Notifications. Using these topics is not recommended. If you want to use one created for that experience, you must add the required policy for AWS CodeStar Notifications in addition to the one that already exists. For more information, see [Configure Amazon SNS topics for notifications](#) and [Understanding notification contents and security](#).

8. Choose **Submit**, and then review the notification rule.

9. Subscribe your email address to the Amazon SNS topic you just created. For more information, see [To subscribe users to an Amazon SNS topic used for notifications](#).

10. Navigate to your deployment application and start a deployment.

11. After thedeployment succeeds, the notification rule sends a notification to all topic subscribers with information about the event.

Create a notification rule for a pipeline

You can create notification rules to send notifications about the events on your pipeline that are important to you. The following steps show you how to set up a notification rule on a single pipeline event. These steps are written with the assumption that you have a pipeline configured in your AWS account.

1. Open the CodePipeline console at https://console.aws.amazon.com/codepipeline/.

2. Choose a pipeline from the list and open it.

3. Choose **Notify**, and then choose **Create notification rule**. You can also choose **Settings**, and then choose **Create notification rule**.

4. In **Notification name**, enter a name for the rule.

5. In **Detail type**, choose **Basic** if you want only the information provided to Amazon EventBridge included in the notification. Choose **Full** if you want to include information provided to Amazon EventBridge and information that might be supplied by the resource service or the notification manager.

   For more information, see Understanding notification contents and security.

6. In **Events that trigger notifications**, under **Action execution**, select **Started**.

7. In **Targets**, choose **Create SNS topic**.

   > **ⓘ Note**
   >
   > When you create the topic as part of creating the notification rule, the policy that allows CodePipeline to publish events to the topic is applied for you. Using a topic created for notification rules helps ensure that you subscribe only those users you want to receive notifications about this pipeline.

   After the **codestar-notifications-** prefix,enter a name for the topic, and then choose **Submit**.

> ⓘ **Note**
>
> If you want to use an existing Amazon SNS topic instead of creating a new one, in
> **Targets**, choose its ARN. Make sure the topic has the appropriate access policy, and
> that the subscriber list contains only those users who are allowed to see information
> about the resource. If the Amazon SNS topic is a topic that was used for CodeCommit
> notifications before November 5, 2019, it will contain a policy that allows CodeCommit
> to publish to it that contains different permissions than those required for AWS
> CodeStar Notifications. Using these topics is not recommended. If you want to use
> one created for that experience, you must add the required policy for AWS CodeStar
> Notifications in addition to the one that already exists. For more information, see
> [Configure Amazon SNS topics for notifications](#) and [Understanding notification
> contents and security](#).

8. Choose **Submit**, and then review the notification rule.

9. Subscribe your email address to the Amazon SNS topic you just created. For more information,
   see [To subscribe users to an Amazon SNS topic used for notifications](#).

10. Navigate to your pipeline, and then choose **Release change**.

11. When the action starts, the notification rule sends a notification to all topic subscribers with
    information about the event.

Working with notification rules

A notification rule is where you configure which events you want users to receive notifications
about and specify the targets that receive those notifications. You can send notifications directly
to users through Amazon SNS, or through AWS Chatbot clients configured for Slack or Microsoft
Teams channels. If you want to extend the reach of notifications, you can manually configure
integration between notifications and AWS Chatbot so that notifications are sent to Amazon Chime
chatrooms. For more information, see [Targets](#) and [To integrate notifications with AWS Chatbot and
Amazon Chime](#).

Create notification rule

Notification rules set up a subscription to events that happen with your resources. When these events occur, you will receive notifications sent to the targets you designate. You can manage your notification preferences in Settings. Info

Notification rule settings

Notification name

MyNotificationRuleForPullRequests

Detail type
Choose the level of detail you want in notifications. **Learn more about notifications and security** 🔗

- ● **Full**
  Includes any supplemental information about events provided by the resource or the notifications feature.

- ○ **Basic**
  Includes only information provided in resource events.

Events that trigger notifications

[Select none] [Select all]

Comments	Approvals	Pull request	Branches and tags
☐ On commits	☐ Status changed	☑ Source updated	☐ Created
☑ On pull requests	☑ Rule override	☑ Created	☐ Deleted
		☐ Status changed	☐ Updated
		☑ Merged	

Targets

Choose a target type for the notification rule. SNS topics can be created specifically for use with the notification rule, or existing topics can be modified for use with notifications. AWS Chatbot clients for Slack integration must be created before you can choose them as a target type. Learn more 🔗

You can use the Developer Tools console or the AWS CLI to create and manage notification rules.

Topics

- [Create a notification rule](#)

- [View notification rules](#)

- [Edit a notification rule](#)

- [Enable or disable notifications for a notification rule](#)

- [Delete a notification rule](#)

Create a notification rule

You can use the Developer Tools console or the AWS CLI to create notification rules. You can create an Amazon SNS topic to use as a target for a notification rule as part of creating the rule. If you want to use an AWS Chatbot client as a target, you must create that client before you can create the rule. For more information, see [Configure an AWS Chatbot client for a Slack channel](#).

To create a notification rule (console)

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/notifications.

2. Use the navigation bar to navigate to the resource.

   - For CodeBuild, choose **Build**, choose **Build projects**, and choose a build project.

   - For CodeCommit, choose **Source**, choose **Repositories**, and choose a repository.

   - For CodeDeploy, choose **Applications**, and choose an application.

   - For CodePipeline, choose **Pipeline**, choose **Pipelines**, and choose a pipeline.

3. On the resource page, choose **Notify**, and then choose **Create notification rule**. You can also go to the **Settings** page for the resource, go to **Notifications** or **Notification rules**, and then choose **Create notification rule**.

4. In **Notification name**, enter a name for the rule.

5. In **Detail type**, choose **Basic** if you want only the information provided to Amazon EventBridge included in the notification. Choose **Full** if you want to include information provided to Amazon EventBridge and information that might be supplied by the resource service or the notification manager.

   For more information, see [Understanding notification contents and security](#).

6. In **Events that trigger notifications**, select the events for which you want to send notifications. For event types for a resource, see the following:

   - CodeBuild: [Events for notification rules on build projects](#)

- CodeCommit: [Events for notification rules on repositories](#)

- CodeDeploy: [Events for notification rules on deployment applications](#)

- CodePipeline: [Events for notification rules on pipelines](#)

7.   In **Targets**, do one of the following:

- If you have already configured a resource to use with notifications, in **Choose target type**, choose either **AWS Chatbot (Slack)**, **AWS Chatbot (Microsoft Teams)**, or **SNS topic**. In **Choose target**, choose the name of the client (for a Slack or Microsoft Teams client configured in AWS Chatbot) or the Amazon Resource Name (ARN) of the Amazon SNS topic (for Amazon SNS topics already configured with the policy required for notifications).

- If you have not configured a resource to use with notifications, choose **Create target**, and then choose **SNS topic**. Provide a name for the topic after **codestar-notifications-**, and then choose **Create**.

> ⓘ **Note**
>
> - If you create the Amazon SNS topic as part of creating the notification rule, the policy that allows the notifications feature to publish events to the topic is applied for you. Using a topic created for notification rules helps ensure that you subscribe only those users that you want to receive notifications about this resource.
>
> - You cannot create an AWS Chatbot client as part of creating a notification rule. If you choose AWS Chatbot (Slack) or AWS Chatbot (Microsoft Teams), you will see a button directing you to configure a client in AWS Chatbot. Choosing that option opens the AWS Chatbot console. For more information, see [Configure an AWS Chatbot client for a Slack channel](#).
>
> - If you want to use an existing Amazon SNS topic as a target, you must add the required policy for AWS CodeStar Notifications in addition to any other policies that might exist for that topic. For more information, see [Configure Amazon SNS topics for notifications](#) and [Understanding notification contents and security](#).

8.   Choose **Submit**, and then review the notification rule.

> ⓘ **Note**
>
> Users must subscribe and confirm subscriptions to the Amazon SNS topic you specified
> as the target of the rule before they will receive notifications. For more information,
> see To subscribe users to an Amazon SNS topic used for notifications.

To create a notification rule (AWS CLI)

1. At a terminal or command prompt, run the **create-notification rule** command to generate the
   JSON skeleton.

```
aws codestar-notifications create-notification-rule --generate-cli-skeleton
 > rule.json
```

   You can name the file anything you want. In this example, the file is named *rule.json*.

2. Open the JSON file in a plaintext editor and edit it to include the resource, event types, and
   Amazon SNS target that you want for the rule.

   The following example shows a notification rule named **MyNotificationRule** for a
   repository named *MyDemoRepo* in an AWS account with the ID *123456789012*. Notifications
   with the full detail type are sent to an Amazon SNS topic named *MyNotificationTopic*
   when branches and tags are created.

```
{
    "Name": "MyNotificationRule",
    "EventTypeIds": [
        "codecommit-repository-branches-and-tags-created"
    ],
    "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
    "Targets": [
        {
            "TargetType": "SNS",
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
```

```
    }
```

Save the file.

3. Using the file you just edited, at the terminal or command line, run the **create-notification-rule** command again to create the notification rule.

```
aws codestar-notifications create-notification-rule --cli-input-json
  file://rule.json
```

4. If successful, the command returns the ARN of the notification rule, similar to the following.

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

To list event types for notification rules (AWS CLI)

1. At a terminal or command prompt, run the **list-event-types** command. You can use the `--filters` option to limit the response to a specific resource type or other attribute. For example, the following returns a list of event types for CodeDeploy applications.

```
aws codestar-notifications list-event-types --filters
  Name=SERVICE_NAME,Value=CodeDeploy
```

2. This command produces output similar to the following.

```
{
    "EventTypes": [
        {
            "EventTypeId": "codedeploy-application-deployment-succeeded",
            "ServiceName": "CodeDeploy",
            "EventTypeName": "Deployment: Succeeded",
            "ResourceType": "Application"
        },
        {
            "EventTypeId": "codedeploy-application-deployment-failed",
            "ServiceName": "CodeDeploy",
            "EventTypeName": "Deployment: Failed",
            "ResourceType": "Application"
```

```
        },
        {
            "EventTypeId": "codedeploy-application-deployment-started",
            "ServiceName": "CodeDeploy",
            "EventTypeName": "Deployment: Started",
            "ResourceType": "Application"
        }
    ]
}
```

To add a tag to a notification rule (AWS CLI)

1. At a terminal or command prompt, run the **tag-resource** command. For example, use the
    following command to add a tag key-value pair that has the name *Team* and the value
    Li_Juan.

    ```
    aws codestar-notifications tag-resource --arn arn:aws:codestar-notifications:us-
    east-1:123456789012:notificationrule/fe1efd35-EXAMPLE --tags Team=Li_Juan
    ```

2. This command produces output similar to the following.

    ```
    {
        "Tags": {
            "Team": "Li_Juan"
        }
    }
    ```

View notification rules

You can use the Developer Tools console or the AWS CLI to view all of the notification rules for
all resources in an AWS Region. You can also view the details of each notification rule. Unlike the
process for creating a notification rule, you do not have to go to the resource page for the resource.

To view notification rules (console)

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/
    settings/notifications.

2. In the navigation bar, expand **Settings**, and then choose **Notification rules**.

3. In **Notification rules**, review the list of rules configured for your resources in your AWS account in the AWS Region where you are currently signed in. Use the selector to change the AWS Region.

4. To view the details of a notification rule, choose it from the list, and then choose **View details**. You can also simply choose its name in the list.

To view a list of notification rules (AWS CLI)

1. At a terminal or command prompt, run the **list-notification-rules** command to view all notification rules for the specified AWS Region.

```
aws codestar-notifications list-notification-rules --region us-east-1
```

2. If successful, this command returns the ID and ARN for each notification rule in the AWS Region, similar to the following.

```
{
    "NotificationRules": [
        {
            "Id": "dc82df7a-EXAMPLE",
            "Arn": "arn:aws:codestar-notifications:us-
east-1:123456789012:notificationrule/dc82df7a-EXAMPLE"
        },
        {
            "Id": "8d1f0983-EXAMPLE",
            "Arn": "arn:aws:codestar-notifications:us-
east-1:123456789012:notificationrule/8d1f0983-EXAMPLE"
        }
    ]
}
```

To view details of a notification rule (AWS CLI)

1. At a terminal or command prompt, run the **describe-notification-rule** command, specifying the ARN of the notification rule.

```
aws codestar-notifications describe-notification-rule --arn arn:aws:codestar-
notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE
```

2.   If successful, the command returns output similar to the following.

```json
{
    "LastModifiedTimestamp": 1569199844.857,
    "EventTypes": [
        {
            "ServiceName": "CodeCommit",
            "EventTypeName": "Branches and tags: Created",
            "ResourceType": "Repository",
            "EventTypeId": "codecommit-repository-branches-and-tags-created"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL",
    "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE",
    "Targets": [
        {
            "TargetStatus": "ACTIVE",
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic",
            "TargetType": "SNS"
        }
    ],
    "Name": "MyNotificationRule",
    "CreatedTimestamp": 1569199844.857,
    "CreatedBy": "arn:aws:iam::123456789012:user/Mary_Major"
}
```

To view a list of tags for a notification rule (AWS CLI)

1.   At a terminal or command prompt, run the **list-tags-for-resource** command to view all tags for a specified notification rule ARN.

```
aws codestar-notifications list-tags-for-resource --arn arn:aws:codestar-
notifications:us-east-1:123456789012:notificationrule/fe1efd35-EXAMPLE
```

2.   If successful, this command returns output similar to the following.

```json
{
```

```
        "Tags": {
            "Team": "Li_Juan"
        }
    }
}
```

Edit a notification rule

You can edit a notification rule to change its name, the events for which it sends notifications, the detail type, or the target or targets to which it sends notifications. You can use the Developer Tools console or the AWS CLI to edit a notification rule.

To edit a notification rule (console)

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/notifications.

2. In the navigation bar, expand **Settings**, and then choose **Notification rules**.

3. In **Notification rules**, review the rules configured for resources in your AWS account in the AWS Region where you are currently signed in. Use the selector to change the AWS Region.

4. Choose the rule from the list, and then choose **Edit**. Make your changes, and then choose **Submit**.

To edit a notification rule (AWS CLI)

1. At a terminal or command prompt, run the describe-notification-rule command to view the structure of the notification rule.

2. Run the **update-notification rule** command to generate the JSON skeleton and then save it to a file.

    ```
    aws codestar-notifications update-notification-rule --generate-cli-skeleton
      > update.json
    ```

    You can name the file anything you want. In this example, the file is *update.json*.

3. Open the JSON file in a plaintext editor and make changes to the rule.

    The following example shows a notification rule named **MyNotificationRule** for a repository named *MyDemoRepo* in an AWS account with the ID *123456789012*. Notifications

are sent to an Amazon SNS topic named *MyNotificationTopic* when branches and tags are created. The rule name is changed to *MyNewNotificationRule*.

```
{
    "Name": "MyNewNotificationRule",
    "EventTypeIds": [
        "codecommit-repository-branches-and-tags-created"
    ],
    "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
    "Targets": [
        {
            "TargetType": "SNS",
            "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
}
```

Save the file.

4. Using the file you just edited, at the terminal or command line, run the **update-notification-rule** command again to update the notification rule.

```
aws codestar-notifications update-notification-rule --cli-input-json
  file://update.json
```

5. If successful, the command returns the Amazon Resource Name (ARN) of the notification rule, similar to the following.

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

To remove a tag from a notification rule (AWS CLI)

1. At a terminal or command prompt, run the **untag-resource** command. For example, the following command removes a tag with the name of *Team*.

```
aws codestar-notifications untag-resource --arn arn:aws:codestar-notifications:us-
east-1:123456789012:notificationrule/fe1efd35-EXAMPLE --tag-keys Team
```

2. If successful, this command returns nothing.

See also

- [Add or remove a target for a notification rule](#)

- [Enable or disable notifications for a notification rule](#)

- [Events](#)

Enable or disable notifications for a notification rule

When you create a notification rule, notifications are enabled by default. You do not have to delete the rule to prevent it from sending notifications. You can simply change its notification status.

To change the notification status for a notification rule (console)

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/notifications.

2. In the navigation bar, expand **Settings**, and then choose **Notification rules**.

3. In **Notification rules**, review the rules configured for resources in your AWS account in the AWS Region where you are currently signed in. Use the selector to change the AWS Region.

4. Find the notification rule you want to enable or disable, and choose it to display its details.

5. In **Notification status**, choose the slider to change the status of the rule:

   - **Sending notifications**: This is the default.

   - **Notifications paused**: No notifications are sent to the specified targets.

To change notification status for a notification rule (AWS CLI)

1. Follow the steps in [To edit a notification rule (AWS CLI)](#) to obtain the JSON for the notification rule.

2. Edit the `Status` field to ENABLED (default) or DISABLED (no notifications), and then run the **update-notification-rule** command to change the status.

```
"Status": "ENABLED"
```

Delete a notification rule

There can be only 10 notification rules configured for a resource, so consider deleting rules you no longer need. You can use the Developer Tools console or the AWS CLI to delete a notification rule.

> ⓘ **Note**
>
> You cannot undo the deletion of a notification rule, but you can recreate it. Deleting a notification rule does not delete the target.

To delete a notification rule (console)

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/notifications.

2. In the navigation bar, expand **Settings**, and then choose **Notification rules**.

3. In **Notification rules**, review the rules configured for resources in your AWS account in the AWS Region where you are currently signed in. Use the selector to change the AWS Region.

4. Choose the notification rule, and then choose **Delete**.

5. Type **delete**, and then choose **Delete**.

To delete a notification rule (AWS CLI)

1. At a terminal or command prompt, run the **delete-notification-rule** command, specifying the ARN of the notification rule.

   ```
   aws codestar-notifications delete-notification-rule --arn arn:aws:codestar-
   notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE
   ```

2. If successful, the command returns the ARN of the deleted notification rule, similar to the following.

   ```
   {
   ```

```
        "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
    dc82df7a-EXAMPLE"
    }
```

Working with notification rule targets

A notification rule target is a destination that defines where you want notifications to be sent when a notification rule's event conditions are met. You can choose between Amazon SNS topics and AWS Chatbot clients that are configured for Slack or Microsoft Teams channels. You can create an Amazon SNS topic as a target as part of creating a notification rule (recommended). You can also choose an existing Amazon SNS topic in the same AWS Region as the notification rule, but you must configure it with the required policy. If you choose to use an AWS Chatbot client as a target, you must first create that client in AWS Chatbot.

If you want to extend the reach of notifications, you can manually configure integration between notifications and AWS Chatbot so that notifications are sent to Amazon Chime chatrooms. You can then choose the Amazon SNS topic configured for that AWS Chatbot client as the target for the notification rule. For more information, see To integrate notifications with AWS Chatbot and Amazon Chime.

You can use the Developer Tools console or the AWS CLI to manage notification targets. You can use the console or the AWS CLI to create and configure Amazon SNS topics and AWS Chatbot clients as targets. You can also configure integration between the Amazon SNS topics that you configure as targets and AWS Chatbot. This makes it possible for you to send notifications to Amazon Chime chatrooms. For more information, see Configure integration between notifications and AWS Chatbot.

Topics

- Create or configure a notification rule target
- View notification rule targets
- Add or remove a target for a notification rule
- Delete a notification rule target

Create or configure a notification rule target

Notification rule targets are Amazon SNS topics or AWS Chatbot clients configured for Slack or Microsoft Teams channels.

An AWS Chatbot client must be created before you can select a client as a target. When you choose an AWS Chatbot client as a target for a notification rule, an Amazon SNS topic is configured for that AWS Chatbot client with all the policies required for notifications to be sent to the Slack or Microsoft Teams channel. You don't have to configure any existing Amazon SNS topics for the AWS Chatbot client.

You can create Amazon SNS notification rule targets in the Developer Tools console when you create a notification rule. The policy that allows notifications to be sent to that topic is applied for you. This is the easiest way to create a target for a notification rule. For more information, see Create a notification rule.

If you use an existing Amazon SNS topic, you must configure it with an access policy that allows the resource to send notifications to that topic. For an example, see Configure Amazon SNS topics for notifications.

> **ⓘ Note**
>
> If you want to use an existing Amazon SNS topic instead of creating a new one, in **Targets**, choose its ARN. Make sure the topic has the appropriate access policy, and that the subscriber list contains only those users who are allowed to see information about the resource. If the Amazon SNS topic is a topic that was used for CodeCommit notifications before November 5, 2019, it will contain a policy that allows CodeCommit to publish to it that contains different permissions than those required for AWS CodeStar Notifications. Using these topics is not recommended. If you want to use one created for that experience, you must add the required policy for AWS CodeStar Notifications in addition to the one that already exists. For more information, see Configure Amazon SNS topics for notifications and Understanding notification contents and security.

If you want to extend the reach of notifications, you can manually configure integration between notifications and AWS Chatbot so that notifications are sent to Amazon Chime chatrooms. For more information, see Targets and To integrate notifications with AWS Chatbot and Amazon Chime.

To configure an existing Amazon SNS topic to use as a notification rule target (console)

1. Sign in to the AWS Management Console and open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.

2. In the navigation bar, choose **Topics**. Choose the topic, and then choose **Edit**.

3. Expand **Access policy**, and then choose **Advanced**.

4. In the JSON editor, add the following statement to the policy. Include the topic ARN, AWS Region, AWS account ID, and topic name.

```
{
      "Sid": "AWSCodeStarNotifications_publish",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codestar-notifications.amazonaws.com"
        ]
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-
MyTopicForNotificationRules"
    }
```

The policy statement should look like the following.

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "SNS:AddPermission",
        "SNS:RemovePermission",
        "SNS:DeleteTopic",
        "SNS:Subscribe",
        "SNS:ListSubscriptionsByTopic",
        "SNS:Publish"
      ],
      "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-
MyTopicForNotificationRules",
```

```
        "Condition": {
          "StringEquals": {
            "AWS:SourceOwner": "123456789012"
          }
        }
      },
    {
        "Sid": "AWSCodeStarNotifications_publish",
        "Effect": "Allow",
        "Principal": {
          "Service": [
            "codestar-notifications.amazonaws.com"
          ]
        },
        "Action": "SNS:Publish",
        "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-
    MyTopicForNotificationRules"
      }
    ]
  }
```

5. Choose **Save changes**.

6. In **Subscriptions**, review the list of topic subscribers. Add, edit, or delete subscribers as appropriate for this notification rule target. Make sure that the subscriber list contains only those users who are allowed to see information about the resource. For more information, see Understanding notification contents and security.

To create an AWS Chatbot client with Slack to use as a target

1. Follow the instructions in Setting up AWS Chatbot with Slack in the *AWS Chatbot Administrator Guide*. When you do so, consider the following choices for optimal integration with notifications:

    *   When creating an IAM role, consider choosing a role name that makes it easy to identify the purpose of this role (for example, **AWSCodeStarNotifications-Chatbot-Slack-Role**). This can help you identify the purpose of the role in the future.

    *   In **SNS topics**, you don't have to choose a topic or an AWS Region. When you choose the AWS Chatbot client as a target, an Amazon SNS topic with all the required permissions is created and configured for the AWS Chatbot client as part of the notification rule creation process.

2. Complete the client creation process. This client is then available for you to choose as a target when creating notification rules. For more information, see [Create a notification rule](#).

> **ⓘ Note**
>
> Do not remove the Amazon SNS topic from the AWS Chatbot client after it has been configured for you. Doing so will prevent notifications from being sent to Slack.

To create an AWS Chatbot client with Microsoft Teams to use as a target

1. Follow the instructions in [Setting up AWS Chatbot with Microsoft Teams](#) in the *AWS Chatbot Administrator Guide*. When you do so, consider the following choices for optimal integration with notifications:

   - When creating an IAM role, consider choosing a role name that makes it easy to identify the purpose of this role (for example, **AWSCodeStarNotifications-Chatbot-Microsoft-Teams-Role**). This can help you identify the purpose of the role in the future.

   - In **SNS topics**, you don't have to choose a topic or an AWS Region. When you choose the AWS Chatbot client as a [target](#), an Amazon SNS topic with all the required permissions is created and configured for the AWS Chatbot client as part of the notification rule creation process.

2. Complete the client creation process. This client is then available for you to choose as a target when creating notification rules. For more information, see [Create a notification rule](#).

> **ⓘ Note**
>
> Do not remove the Amazon SNS topic from the AWS Chatbot client after it has been configured for you. Doing so will prevent notifications from being sent to Microsoft Teams.

View notification rule targets

You can use Developer Tools console, not the Amazon SNS console to view all of the notification rule targets for all resources in an AWS Region. You can also view the details of a notification rule target.

To view notification rule targets (console)

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/notifications.

2. In the navigation bar, expand **Settings**, and then choose **Notification rules**.

3. In **Notification rule targets**, review the list of targets used by notification rules in in your AWS account in the AWS Region where you are currently signed in. Use the selector to change the AWS Region. If the target status shows as **Unreachable**, you might need to investigate. For more information, see Troubleshooting.

To view a list of notification rule targets (AWS CLI)

1. At a terminal or command prompt, run the **list-targets** command to view a list of all notification rule targets for the specified AWS Region:

```
aws codestar-notifications list-targets --region us-east-2
```

2. If successful, this command returns the ID and ARN for each notification rule in the AWS Region, similar to the following:

```
{
    "Targets": [
        {
            "TargetAddress": "arn:aws:sns:us-
east-2:123456789012:MySNSTopicForNotificationRules",
            "TargetType": "SNS",
            "TargetStatus": "ACTIVE"
        },
        {
            "TargetAddress": "arn:aws:chatbot::123456789012:chat-configuration/
slack-channel/MySlackChannelClientForMyDevTeam",
            "TargetStatus": "ACTIVE",
            "TargetType": "AWSChatbotSlack"
        },
        {
            "TargetAddress": "arn:aws:sns:us-
east-2:123456789012:MySNSTopicForNotificationsAboutMyDemoRepo",
            "TargetType": "SNS",
            "TargetStatus": "ACTIVE"
        }
```

```
    ]
}
```

Add or remove a target for a notification rule

You can edit a notification rule to change the target or targets to which it sends notifications. You can use the Developer Tools console or or the AWS CLI to change a notification rule's targets.

To change the targets for a notification rule (console)

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/notifications.

2. In the navigation bar, expand **Settings**, and then choose **Notification rules**.

3. In **Notification rules**, review the list of rules configured for your resources in your AWS account in the AWS Region where you are currently signed in. Use the selector to change the AWS Region.

4. Choose the rule, and then choose **Edit**.

5. In **Targets**, do one of the following:

   - To add another target, choose **Add Target**, and then choose the Amazon SNS topic or AWS Chatbot (Slack) or AWS Chatbot (Microsoft Teams) client that you want to add from the list. You can also choose **Create SNS topic** to create a topic and add it as a target. A notification rule can have up to 10 targets.

   - To remove a target, next to the target you want to remove, choose **Remove target**.

6. Choose **Submit**.

To add a target to a notification rule (AWS CLI)

1. At a terminal or command prompt, run the **subscribe** command to add a target. For example, the following command adds an Amazon SNS topic as a target for a notification rule.

   ```
   aws codestar-notifications subscribe --arn arn:aws:codestar-
   notifications:us-east-1:123456789012:notificationrule/dc82df7a-
   EXAMPLE --target TargetType=SNS,TargetAddress=arn:aws:sns:us-
   east-1:123456789012:MyNotificationTopic
   ```

2. If successful, the command returns the ARN of the updated notification rule, similar to the following.

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

To remove a target from a notification rule (AWS CLI)

1. At a terminal or command prompt, run the the **unsubscribe** command to remove a target. For example, the following command removes an Amazon SNS topic as a target for a notification rule.

```
aws codestar-notifications unsubscribe --arn arn:aws:codestar-
notifications:us-east-1:123456789012:notificationrule/dc82df7a-
EXAMPLE --target TargetType=SNS,TargetAddress=arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic
```

2. If successful, the command returns the ARN of the updated notification rule and information about the removed target, similar to the following.

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
    "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"
}
```

See also

- Edit a notification rule
- Enable or disable notifications for a notification rule

Delete a notification rule target

You can delete a target if it is no longer needed. A resource can only have 10 notification rule targets configured for it, so deleting unneeded targets can help create room for other targets you might want to add to that notification rule.

> **ⓘ Note**
>
> Deleting a notification rule target removes the target from all notification rules configured to use it as a target, but it does not delete the target itself.

To delete a notification rule target (console)

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/notifications.

2. In the navigation bar, expand **Settings**, and then choose **Notification rules**.

3. In **Notification rule targets**, review the list of targets configured for your resources in your AWS account in the AWS Region where you are currently signed in. Use the selector to change the AWS Region.

4. Choose the notification rule target, and then choose **Delete**.

5. Type `delete`, and then choose **Delete**.

To delete a notification rule target (AWS CLI)

1. At a terminal or command prompt, run the **delete-target** command, specifying the ARN of the target. For example, the following command deletes a target that uses an Amazon SNS topic.

   ```
   aws codestar-notifications delete-target --target-address arn:aws:sns:us-
   east-1:123456789012:MyNotificationTopic
   ```

2. If successful, the command returns nothing. If unsuccessful, the command returns an error. The most common error is that the topic is the target for one or more notification rules.

   ```
   An error occurred (ValidationException) when calling the DeleteTarget operation:
    Unsubscribe target before deleting.
   ```

   You can use the `--force-unsubscribe-all` parameter to remove the target from all notification rules configured to use it as a target, and then delete the target.

   ```
   aws codestar-notifications delete-target --target-address arn:aws:sns:us-
   east-1:123456789012:MyNotificationTopic --force-unsubscribe-all
   ```

Configure integration between notifications and AWS Chatbot

AWS Chatbot is an AWS service that makes it possible for DevOps and software development teams to use Amazon Chime chat rooms, Slack channels, and Microsoft Team channels to monitor and respond to operational events in the AWS Cloud. You can configure integration between notification rule targets and AWS Chatbot so that notifications about events appear in the Amazon Chime room, Slack channel, or Microsoft Teams channel you choose. For more information, see the [AWS Chatbot documentation](#).

Before you configure integration with AWS Chatbot, you must configure a notification rule and a rule target. For more information, see [Setting up](#) and [Create a notification rule](#). You must also configure a Slack channel, Microsoft Teams channel, or an Amazon Chime chatroom in AWS Chatbot. For more information, see the documentation for these services.

Topics

- [Configure an AWS Chatbot client for a Slack channel](#)
- [Configure an AWS Chatbot client for a Microsoft Teams channel](#)
- [Configure clients for Slack or Amazon Chime manually](#)

Configure an AWS Chatbot client for a Slack channel

You can create notification rules that use an AWS Chatbot client as a target. If you create a client for a Slack channel, you can use this client directly as a target in the workflow for creating a notification rule. This is the easiest way to set up notifications that appear in Slack channels.

To create an AWS Chatbot client with Slack to use as a target

1. Follow the instructions in [Setting up AWS Chatbot with Slack](#) in the *AWS Chatbot Administrator Guide*. When you do so, consider the following choices for optimal integration with notifications:

   - When creating an IAM role, consider choosing a role name that makes it easy to identify the purpose of this role (for example, **AWSCodeStarNotifications-Chatbot-Slack-Role**). This can help you identify the purpose of the role in the future.

   - In **SNS topics**, you don't have to choose a topic or an AWS Region. When you choose the AWS Chatbot client as a [target](#), an Amazon SNS topic with all the required permissions is created and configured for the AWS Chatbot client as part of the notification rule creation process.

2. Complete the client creation process. This client is then available for you to choose as a target when creating notification rules. For more information, see [Create a notification rule](#).

> **ⓘ Note**
>
> Do not remove the Amazon SNS topic from the AWS Chatbot client after it has been configured for you. Doing so will prevent notifications from being sent to Slack.

Configure an AWS Chatbot client for a Microsoft Teams channel

You can create notification rules that use an AWS Chatbot client as a target. If you create a client for a Microsoft Teams channel, you can use this client directly as a target in the workflow for creating a notification rule. This is the easiest way to set up notifications that appear in Microsoft Teams channels.

To create an AWS Chatbot client with Microsoft Teams to use as a target

1. Follow the instructions in [Setting up AWS Chatbot with Microsoft Teams](#) in the *AWS Chatbot Administrator Guide*. When you do so, consider the following choices for optimal integration with notifications:

    - When creating an IAM role, consider choosing a role name that makes it easy to identify the purpose of this role (for example, `AWSCodeStarNotifications-Chatbot-Microsoft-Teams-Role`). This can help you identify the purpose of the role in the future.

    - In **SNS topics**, you don't have to choose a topic or an AWS Region. When you choose the AWS Chatbot client as a [target](#), an Amazon SNS topic with all the required permissions is created and configured for the AWS Chatbot client as part of the notification rule creation process.

2. Complete the client creation process. This client is then available for you to choose as a target when creating notification rules. For more information, see [Create a notification rule](#).

> **ⓘ Note**
>
> Do not remove the Amazon SNS topic from the AWS Chatbot client after it has been configured for you. Doing so will prevent notifications from being sent to Microsoft Teams.

Configure clients for Slack or Amazon Chime manually

You can choose to create the integration between notifications and Slack or Amazon Chime directly. This is the only method available for configuring notifications to Amazon Chime chatrooms. When you configure this integration manually, you create an AWS Chatbot client that uses an Amazon SNS topic that you have previously configured as the target for a notification rule.

To manually integrate notifications with AWS Chatbot and slack

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/ settings/notifications.
2. Choose **Settings**, and then choose **Notification rules**.
3. In **Notification rule targets**, find and copy the target.

> ⓘ **Note**
>
> You can configure more than one notification rule to use the same Amazon SNS topic as its target. This can help you consolidate messaging, but can have unintended consequences if the subscription list is intended for one notification rule or resource.

4. Open the AWS Chatbot console at https://console.aws.amazon.com/chatbot/.
5. Choose **Configure new client**, and then choose **Slack**.
6. Choose **Configure**.
7. Sign in to your Slack workspace.
8. When you are prompted to confirm the choices, choose **Allow**.
9. Choose **Configure new channel**.
10. In **Configuration details**, in **Configuration name**, enter a name for your client. This is the name that will appear in the list of available targets for the **AWS Chatbot (Slack)** target type when you create notification rules.
11. In **Configure Slack Channel**, in **Channel type**, choose **Public** or **Private**, depending on the type of channel with which you want to integrate.

    - In **Public channel**, choose the name of the Slack channel from the list.
    - In **Private channel ID**, enter the channel code or URL.

12. In **IAM permissions**, in **Role**, choose **Create an IAM role using a template**. In **Policy templates**, choose **Notification permissions**. In **Role name**, enter a name for this role (for

example, `AWSCodeStarNotifications-Chatbot-Slack-Role`). In **Policy templates**, choose **Notification permissions**.

13. In **SNS topics**, in **SNS Region**, choose the AWS Region where you created the notification rule target. In **SNS topics**, choose the name of the Amazon SNS topic that you configured as the notification rule target.

> ⓘ **Note**
>
> This step is not necessary if you will create a notification rule using this client as a target.

14. Choose **Configure**.

> ⓘ **Note**
>
> If you configured integration with a private channel, you must invite AWS Chatbot to the channel before you will see notifications in that channel. For more information, see the AWS Chatbot documentation.

15. (Optional) To test the integration, make a change in the resource that matches an event type for a notification rule that is configured to use the Amazon SNS topic as its target. For example, if you have a notification rule configured to send notifications when comments are made on a pull request, comment on a pull request and then watch the Slack channel in the browser to see when the notification appears.

To integrate notifications with AWS Chatbot and Amazon Chime

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/notifications.

2. Choose **Settings**, and then choose **Notification rules**.

3. In **Notification rule targets**, find and copy the target.

> ⓘ **Note**
>
> You can configure more than one notification rule to use the same Amazon SNS topic as its target. This can help you consolidate messaging, but can have unintended consequences if the subscription list is for one notification rule or resource.

4. In Amazon Chime, open the chatroom that you want to configure for integration.

5. Choose the gear icon in the upper-right corner, and then choose **Manage webhooks**.

6. In the **Manage webhooks** dialog box, choose **New**, enter a name for the webhook, and then choose **Create**.

7. Verify that the webhook appears, and then choose **Copy webhook URL**.

8. Open the AWS Chatbot console at https://console.aws.amazon.com/chatbot/.

9. Choose **Configure new client**, and then choose **Amazon Chime**.

10. In **Configuration details**, in **Configuration name**, enter a name for your client.

11. In **Webhook URL**, paste the URL. In **Webhook description**, provide an optional description.

12. In **IAM permissions**, in **Role**, choose **Create an IAM role using a template**. In **Policy templates**, choose **Notification permissions**. In **Role name**, enter a name for this role (for example, `AWSCodeStarNotifications-Chatbot-Chime-Role`).

13. In **SNS topics**, in **SNS Region**, choose the AWS Region where you created the notification rule target. In **SNS topics**, choose the name of the Amazon SNS topic you configured as the notification rule target.

14. Choose **Configure**.

15. (Optional) To test the integration, make a change in the resource that matches an event type for a notification rule that is configured to use the Amazon SNS topic as its target. For example, if you have a notification rule configured to send notifications when comments are made on a pull request, comment on a pull request and then watch the Amazon Chime chatroom to see when the notification appears.

Logging AWS CodeStar Notifications API calls with AWS CloudTrail

AWS CodeStar Notifications is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all API calls for notifications as events. The calls captured include calls from the Developer Tools console and code calls to the AWS CodeStar Notifications API operations. If you create a trail, you can enable continuous delivery of

CloudTrail events to an Amazon S3 bucket, including events for notifications. 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 AWS CodeStar Notifications, the IP address from which the request was made, who made the request, when it was made, and other details.

For more information, see the [AWS CloudTrail User Guide](#).

AWS CodeStar Notifications information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS CodeStar Notifications, 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 AWS CodeStar Notifications, 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](#)

All AWS CodeStar Notifications actions are logged by CloudTrail and are documented in the [*AWS CodeStar Notifications API Reference*](#). For example, calls to the `CreateNotificationRule`, `Subscribe` and `ListEventTypes` actions generate entries in the CloudTrail log files.

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

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

- Whether the request was made with temporary security credentials for a role or federated user.

- Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentity element](#).

Understanding 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 creation of a notification rule, including both the `CreateNotificationRule` and `Subscribe` actions.

> ⓘ **Note**
>
> Some of the events in notification log file entries might come from the service-linked role AWSServiceRoleForCodeStarNotifications.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type":"IAMUser",
        "principalId":"AIDACKCEVSQ6C2EXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/Mary_Major",
        "accountId":"123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName":"Mary_Major"
    },
    "eventTime": "2019-10-07T21:34:41Z",
    "eventSource": "events.amazonaws.com",
    "eventName": "CreateNotificationRule",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "codestar-notifications.amazonaws.com",
    "userAgent": "codestar-notifications.amazonaws.com",
    "requestParameters": {
```

```
        "description": "This rule is used to route CodeBuild, CodeCommit, CodePipeline,
 and other Developer Tools notifications to AWS CodeStar Notifications",
        "name": "awscodestarnotifications-rule",
        "eventPattern": "{\"source\":[\"aws.codebuild\",\"aws.codecommit\",
\"aws.codepipeline\"]}"
    },
    "responseElements": {
        "ruleArn": "arn:aws:events:us-east-1:123456789012:rule/
awscodestarnotifications-rule"
    },
    "requestID": "ff1f309a-EXAMPLE",
    "eventID": "93c82b07-EXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-10-07",
    "recipientAccountId": "123456789012"
}
```

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type":"IAMUser",
        "principalId":"AIDACKCEVSQ6C2EXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/Mary_Major",
        "accountId":"123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName":"Mary_Major"
    },
    "eventTime": "2019-10-07T21:34:41Z",
    "eventSource": "events.amazonaws.com",
    "eventName": "Subscribe",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "codestar-notifications.amazonaws.com",
    "userAgent": "codestar-notifications.amazonaws.com",
    "requestParameters": {
        "targets": [
            {
                "arn": "arn:aws:codestar-notifications:us-east-1:::",
                "id": "codestar-notifications-events-target"
            }
        ],
        "rule": "awscodestarnotifications-rule"
    },
    "responseElements": {
```

```
        "failedEntryCount": 0,
        "failedEntries": []
    },
    "requestID": "9466cbda-EXAMPLE",
    "eventID": "2f79fdad-EXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-10-07",
    "recipientAccountId": "123456789012"
}
```

Troubleshooting

The following information might help you troubleshoot common issues with notifications.

Topics

- I get a permissions error when I try to create a notification rule on a resource
- I cannot view notification rules
- I cannot create notification rules
- I am receiving notifications for a resource I can't access
- I am not receiving Amazon SNS notifications
- I am receiving duplicate notifications about events
- I want to understand why a notification target status shows as unreachable
- I want to increase my quotas for notifications and resources

I get a permissions error when I try to create a notification rule on a resource

Make sure that you have sufficient permissions. For more information, see Identity-based policy examples.

I cannot view notification rules

Problem: When you are in the Developer Tools console and choose **Notifications** under **Settings**, you see a permissions error.

Possible fixes: You might not have the permissions required to view notifications. While most managed policies for AWS Developer Tools services, such as CodeCommit and CodePipeline,

include permissions for notifications, services that do not currently support notifications do not include permissions to view them. Alternatively, you might have a custom policy applied to your IAM user or role that does not allow you to view notifications. For more information, see Identity-based policy examples.

I cannot create notification rules

You might not have the permissions required to create a notification rule. For more information, see Identity-based policy examples.

I am receiving notifications for a resource I can't access

When you create a notification rule and add a target, the notifications feature does not validate whether the recipient has access to the resource. It is possible for you to receive notifications about a resource that you can't access. If you cannot remove yourself, ask to be removed from the subscription list for the target.

I am not receiving Amazon SNS notifications

To troubleshoot problems with the Amazon SNS topic, check the following:

- Make sure that the Amazon SNS topic was created in the same AWS Region as the notification rule.
- Make sure that your email alias is subscribed to the correct topic and that you have confirmed the subscription. For more information, see Subscribing an endpoint to an Amazon SNS topic.
- Verify that the topic policy has been edited to allow AWS CodeStar Notifications to push notifications to that topic. The topic policy should include a statement similar to the following:

```
{
    "Sid": "AWSCodeStarNotifications_publish",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "codestar-notifications.amazonaws.com"
        ]
    },
    "Action": "SNS:Publish",
    "Resource": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopicName",
    "Condition": {
        "StringEquals": {
```

```
                "aws:SourceAccount": "123456789012"
            }
        }
    }
```

For more information, see [Configure Amazon SNS topics for notifications](#).

I am receiving duplicate notifications about events

Here are the most common reasons for receiving multiple notifications:

- Multiple notification rules that include the same event type have been configured for a resource, and you are subscribed to the Amazon SNS topics that are the targets for those rules. To solve this issue, either unsubscribe from one of the topics or edit the notification rules to remove duplication.
- One or more notification rule targets are integrated with AWS Chatbot and you are receiving notifications in your email inbox and a Slack channel, Microsoft Teams channel, or Amazon Chime chatroom. To solve this issue, consider unsubscribing your email address from the Amazon SNS topic that is the target for the rule and use the Slack channel, Microsoft Teams channel, or Amazon Chime chatroom to view notifications.

I want to understand why a notification target status shows as unreachable

Targets have two possible statuses: *Active* and *Unreachable*. **Unreachable** indicates that notifications were sent to a target, and the delivery was not successful. Notifications continue to be sent to that target, and if successful, the status resets to **Active**.

The target for a notification rule might become unavailable for one of the following reasons:

- The resource (Amazon SNS topic or AWS Chatbot client) has been deleted. Choose another target for the notification rule.
- The Amazon SNS topic is encrypted, and either the required policy for encrypted topics is missing, or the AWS KMS key has been deleted. For more information, see [Configure Amazon SNS topics for notifications](#).
- The Amazon SNS topic does not have the required policy for notifications. Notifications cannot be sent to an Amazon SNS topic unless it has the policy. For more information, see [Configure Amazon SNS topics for notifications](#).

- The supporting service for the target (Amazon SNS or AWS Chatbot) might be experiencing issues.

I want to increase my quotas for notifications and resources

Currently, you cannot change any quotas. See Quotas for notifications.

Quotas for notifications

The following table lists the quotas (also referred to as *limits*) for notifications in the Developer Tools console. For information about limits that can be changed, see AWS service quotas.

Resource	Default limit
Maximum number of notification rules in an AWS account	1000
Maximum number of targets for a notification rule	10
Maximum number of notification rules for a resource	10

What are connections?

You can use the *connections* feature in the Developer Tools console to connect AWS resources such as AWS CodePipeline to external code repositories. This feature has its own API, the AWS CodeConnections API reference. Each connection is a resource that you can give to AWS services to connect to a third-party repository, such as Bitbucket. For example, you can add the connection in CodePipeline so that it triggers your pipeline when a code change is made to your third-party code repository. Each connection is named and associated with a unique Amazon Resource Name (ARN) that is used to reference the connection.

> ⚠️ **Important**
>
> The service name AWS CodeStar Connections has been renamed. Resources created with the previous namespace codestar-connections will still be supported.

What can I do with connections?

You can use connections to integrate third-party provider resources with your AWS resources in developer tools, including:

- Connect to a third-party provider, such as Bitbucket, and use the third-party connection as a source integration with your AWS resources, such as CodePipeline.

- Uniformly manage access to your connection across your resources in CodeBuild build projects, CodeDeploy applications, and pipelines in CodePipeline for your third-party provider.

- Use a connection ARN in your stack templates for CodeBuild build projects, CodeDeploy applications, and pipelines in CodePipeline, without the need to reference stored secrets or parameters.

What third-party providers can I create connections for?

Connections can associate your AWS resources with the following third-party repositories:

- Azure DevOps
- Bitbucket Cloud
- GitHub.com
- GitHub Enterprise Cloud

> **ⓘ Note**
>
> Currently, custom domains for GitHub Enterprise Cloud are not supported.

- GitHub Enterprise Server
- GitLab.com

> **⚠ Important**
>
> Connections support for GitLab includes version 15.x and later.

- GitLab self-managed installation (for Enterprise Edition or Community Edition)

For an overview of the connections workflow, see [Workflow to create or update connections](#).

The steps to create connections for a cloud provider type, such as GitHub, are different from the steps for an installed provider type, such as GitHub Enterprise Server. For the high-level steps to create a connection by provider type, see Working with connections.

> **ⓘ Note**
>
> To use connections in the Europe (Milan) AWS Region, you must:
>
> 1. Install a Region-specific app
>
> 2. Enable the Region
>
> This Region-specific app supports connections in the Europe (Milan) Region. It is published on the third-party provider site, and it is separate from the existing app supporting connections for other Regions. By installing this app, you authorize third-party providers to share your data with the service for this Region only, and you can revoke the permissions at any time by uninstalling the app.
> The service will not process or store your data unless you enable the Region. By enabling this Region, you grant our service permissions to process and store your data.
> Even if the Region is not enabled, third-party providers can still share your data with our service if the Region-specific app remains installed, so make sure to uninstall the app once you disable the Region. For more information, see Enabling a Region.

What AWS services integrate with connections?

You can use connections to integrate your third-party repository with other AWS services. To view the service integrations for connections, see Product and service integrations with AWS CodeConnections.

How do connections work?

Before you can create a connection, you must first install, or provide access to, the AWS authentication app on your third-party account. After a connection is installed, it can be updated to use this installation. When you create a connection, you provide access to the AWS resource in your third-party account. This allows the connection to access content, such as source repositories, in the third-party account, on behalf of your AWS resources. You can then share that connection with other AWS services to provide secure OAuth connections between the resources.

Cloud-based connections are configured as follows with differences called out between user accounts or organizations.

- **User accounts:** Each cloud-based third-party user account has a connector app installation. Multiple connections can be associated with the app installation.

- **Organizations:** Each cloud-based third party organization has a connector app installation. For connections in organizations, your connection mapping to each Organization account in the organization is 1:1. Multiple connections cannot be associated with the app installation. For more details about how organizations work with connections, see How connections in AWS CodeConnections work with organizations.

The following diagram shows how cloud-based connections work with user accounts or organizations.
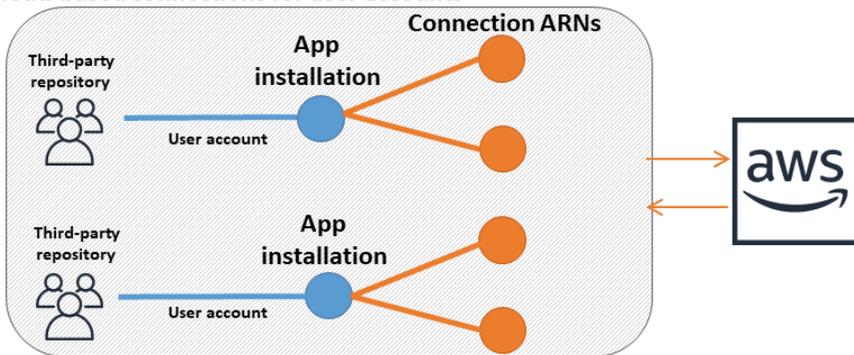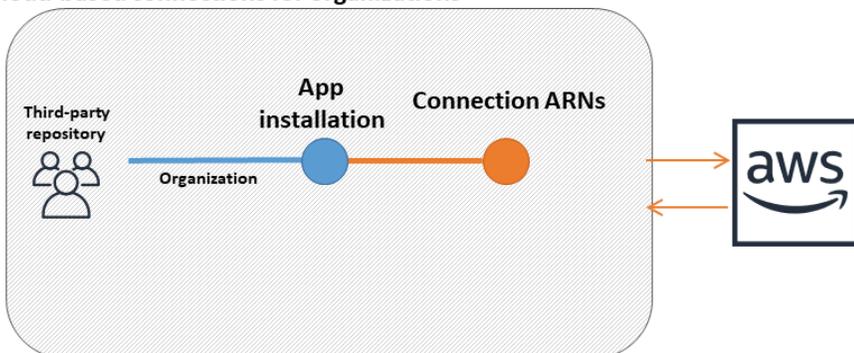
Connections are owned by the AWS account that creates them. Connections are identified by an ARN containing a connection ID. The connection ID is a UUID that cannot be changed or remapped. Deleting and re-establishing a connection results in a new connection ID, and therefore a new connection ARN. This means that connection ARNs are never reused.

A newly created connection is in a `Pending` state. A third-party handshake (OAuth flow) process is required to complete setup of the connection and for it to move from `Pending` to an `Available` state. After this is complete, a connection is `Available` and can be used with AWS services, such as CodePipeline.

If you want to create a connection to an installed provider type (on-prem), such as GitHub Enterprise Server or GitLab self-managed, you use a host resource with your connection.

On-prem connections are configured as follows with differences called out between user accounts or organizations.

- **User accounts:** Each on-prem third-party user account has a connector app installation. Multiple connections for an on-prem provider can be associated with one host.

- **Organizations:** Each on-prem third-party organization has a connector app installation. For on-prem connections in organizations, such as GitHub Organizations for GitHub Enterprise Server, you create a new host for each connection in your organization and be sure to enter the same information in the network fields (VPC, Subnet IDs, and Security Group IDs) for the host. For more details about how organizations work with connections, see How connections in AWS CodeConnections work with organizations.

- **All:** For each on-prem connection, each VPC can only be associated with one host at a time.

In all cases, you will need to provide the URL for your on-prem server. Additionally, if the server is within a private VPC (i.e., not accessible via the internet), you will need to provide VPC information along with optional TLS certificate information. These configurations allow CodeConnections to communicate with the instance and are shared by all connections created for this host. For example, for a single GitHub Enterprise Server instance, you would create a single app represented by a Host. Then, for user account configuration, you could create multiple connections for that host, which correspond to your app installation as shown in the following diagram. Otherwise, for an organization, you create a single app installation and connection for that host.

The following diagram shows how on-prem connections work with user accounts or organizations.

On-prem connections for user accounts
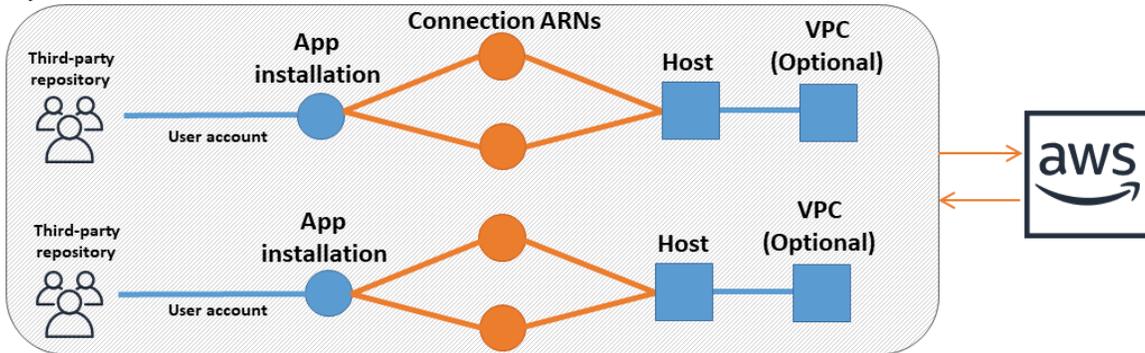
On-prem connections for multiple organizations on a single host
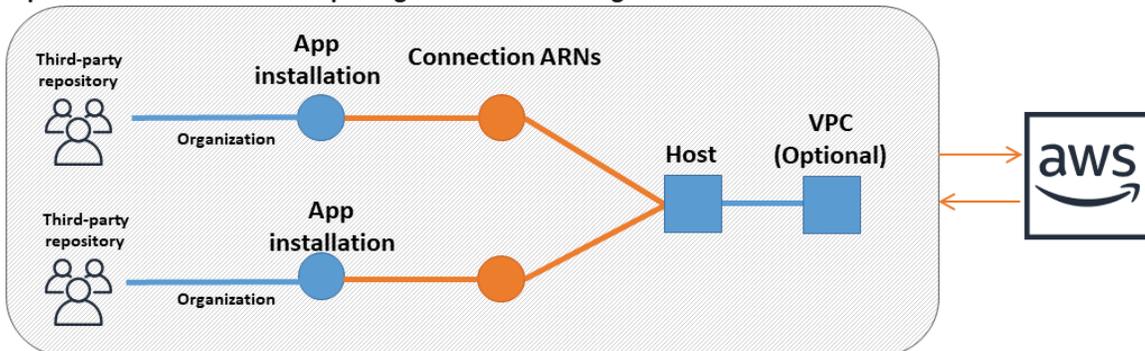
A newly created host is in a `Pending` state. A third-party registration process is required to complete setup of the host and for it to move from `Pending` to an `Available` state. After this is complete, a host is `Available` and can be used for connections to installed provider types.

For an overview of the connections workflow, see Workflow to create or update connections. For an overview of the host creation workflow for installed providers, see Workflow to create or update a host. For the high-level steps to create a connection by provider type, see Working with connections.

How connections in AWS CodeConnections work with organizations

For organizations with a provider, such as GitHub Organizations, you cannot install a GitHub app into multiple GitHub Organizations. A connection has a 1:1 mapping with an organization through the use of the Github connector app. The connector app should be separate for every organization in GitHub or GitHub Enterprise Server and should have a connection associated with it.

For example, to work with multiple organizations on the same GitHub server, you must create separate connections for each organization and install separate GitHub apps for these organizations. The target account on the Github side, however, can be the same.

Workflow to create or update connections

When you create a connection, you also create or use an existing connector app installation for the auth handshake with the third-party provider.

Connections can have the following states:

- `Pending` - A pending connection is a connection that must be completed (moved to `available`) before it can be used.

- `Available` - You can use or pass an `available` connection to other resources and users in your account.

- `Error` - A connection that has an `error` state is retried automatically. It cannot be used until it is `available`.

Workflow: Creating or updating a connection with the CLI, SDK, or AWS CloudFormation

You use the [CreateConnection](#) API to create a connection using the AWS Command Line Interface (AWS CLI), SDK, or CloudFormation. After it is created, the connection is in a `pending` state. You complete the process by using the console **Set up pending connection** option. The console prompts you to create an installation or use an existing installation for the connection. You then use the console to complete the handshake and move the connection to an `available` state by choosing **Complete connection** on the console.

Workflow: Creating or updating a connection with the console

If you are creating a connection to an installed provider type, such as GitHub Enterprise Server, you first create a host. If you are connecting to a cloud provider type, such as Bitbucket, you skip creating the host and continue to creating a connection.

To create or update a connection using the console, you use the CodePipeline edit action page on the console to choose your third-party provider. The console prompts you to create an installation or use an existing installation for the connection, and then use the console to create the connection. The console completes the handshake and moves the connection from `pending` to an `available` state automatically.
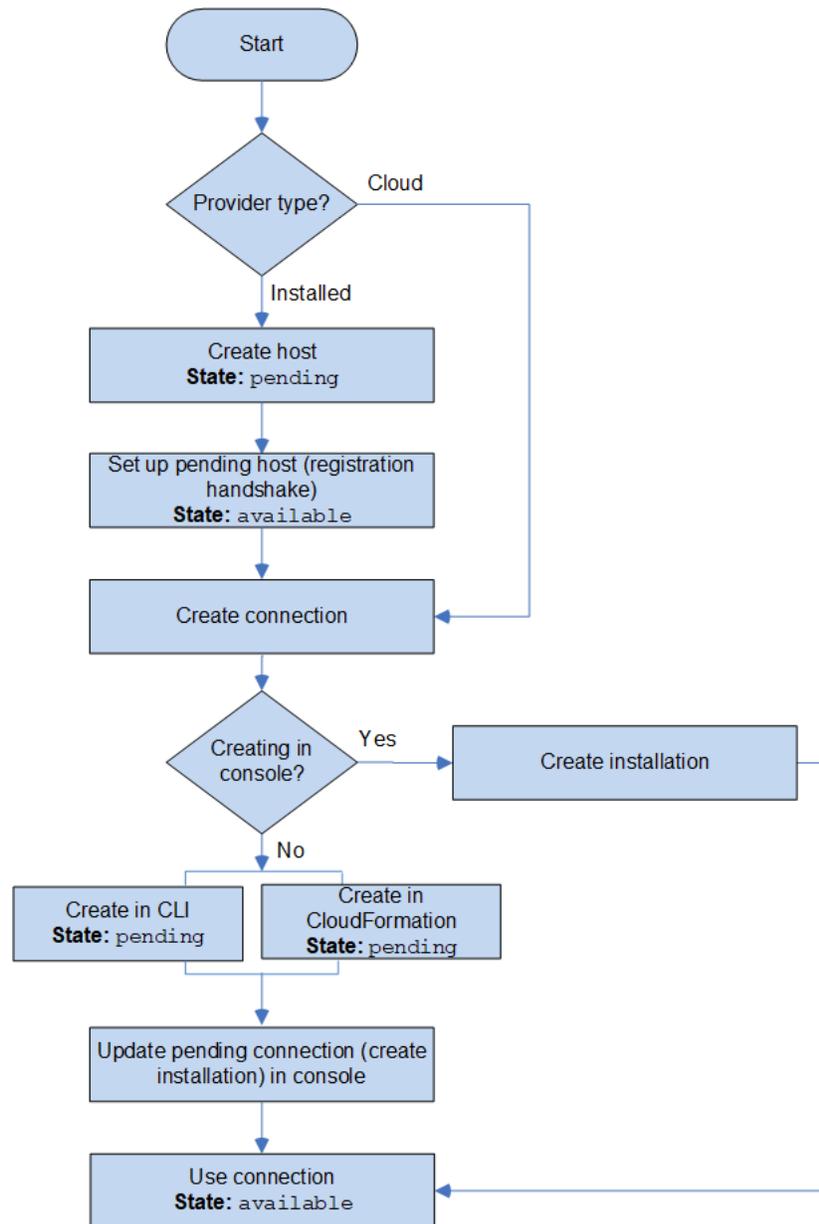
Workflow to create or update a host

When you create a connection for an installed provider (on-prem), you use a host resource.

> **ⓘ Note**
>
> For organizations in GitHub Enterprise Server or GitLab self-managed, you don't pass an available host. You create a new host for each connection in your organization, and you must be sure to enter the same information in the network fields (VPC ID, Subnet IDs, and

Security Group IDs) for the host. For more information, see Connection and host setup for installed providers supporting organizations.

Hosts can have the following states:

- Pending - A `pending` host is a host that has been created and must be set up (moved to `available`) before it can be used.

- Available - You can use or pass an `available` host to your connection.

**Workflow: Creating or updating a host with the CLI, SDK, or AWS CloudFormation**

You use the CreateHost API to create a host using the AWS Command Line Interface (AWS CLI), SDK, or CloudFormation. After it is created, the host is in a `pending` state. You complete the process by using the console **Set up** option in the console.

Workflow: Creating or updating a host with the console

If you are creating a connection to an installed provider type, such as GitHub Enterprise Server or GitLab self-managed, you create a host or use an existing host. If you are connecting to a cloud provider type, such as Bitbucket, you skip creating the host and continue to creating a connection.

Use the console to set up the host and change its status from `pending` to `available`.
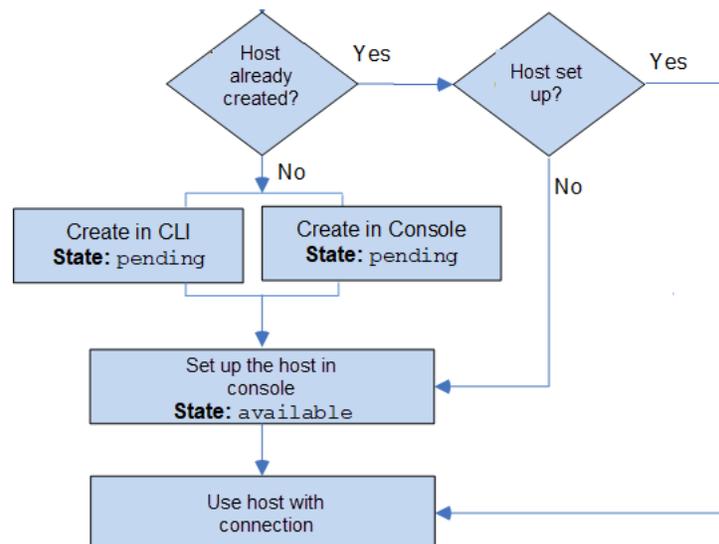
Global resources in AWS CodeConnections

Connections are global resources, meaning that the resource is replicated across all AWS Regions.

Although the connection ARN format reflects the Region name where it was created, the resource is not constrained to any Region. The Region where the connection resource was created is the Region where connection resource data updates are controlled. Examples of API operations that control updates to connection resource data include creating a connection, updating an installation, deleting a connection, or tagging a connection.

Host resources for connections are not globally available resources. You use host resources only in the Region where they were created.

- You only have to create a connection once, and then you can use it in any AWS Region.
- If the Region where the connection was created is having issues, this impacts APIs that control connection resource data, but you can still successfully use the connection in every other Region.
- When you list connection resources in the console or CLI, the list shows all connection resources associated with your account across all Regions.
- When you list host resources in the console or CLI, the list shows host resources associated with your account in the selected Region only.
- When a connection with an associated host resource is listed or viewed with the CLI, the output returns the host ARN regardless of the configured CLI Region.

How do I get started with connections?

To get started, here are some useful topics to review:

- **Learn** about the [concepts](#) for connections.
- **Set up** the [resources you need](#) to start working with connections.
- **Get started** with your [first connections](#) and connect them to a resource.

Connections concepts

Setting up and using the connections feature is easier if you understand the concepts and terms. Here are some concepts to know about as you use connections in the Developer Tools console:

installation

An instance of the AWS app on a third-party account. Installing the AWS CodeStar Connector app allows AWS to access resources within the third-party account. An installation can only be edited on the third-party provider's website.

connection

> An AWS resource used to connect third-party source repositories to other AWS services.

third-party repository

> A repository that is provided by a service or company that is not part of AWS. For example, a Bitbucket repository is a third-party repository.

provider type

> A service or company that provides the third-party source repository you want to connect to. You connect your AWS resources to external *provider types*. A provider type where the source repository is installed on the network and infrastructure is an installed provider type. For example, GitHub Enterprise Server is an installed provider type.

host

> A resource that represents the infrastructure where a third-party provider is installed. Connections use the host to represent the server where your third-party provider is installed, such as GitHub Enterprise Server. You create one host for all connections to that provider type.

> **ⓘ Note**
>
> When you use the console to create a connection to GitHub Enterprise Server, the console creates a host resource for you as part of the process.

AWS CodeConnections supported providers and versions

This chapter provides information about the providers and versions that AWS CodeConnections supports.

Topics

- [Supported provider type for Azure DevOps](#)
- [Supported provider type for Bitbucket](#)
- [Supported provider type for GitHub and GitHub Enterprise Cloud](#)
- [Supported provider type and versions for GitHub Enterprise Server](#)
- [Supported provider type for GitLab.com](#)
- [Supported provider type for GitLab self-managed](#)

Supported provider type for Azure DevOps

You can use the connections app with Azure DevOps.

Installed (hosted) provider types, such as Azure Cloud Hosting, are not supported.

Supported provider type for Bitbucket

You can use the connections app with Atlassian Bitbucket Cloud.

Installed Bitbucket provider types, such as Bitbucket Server, are not supported.

Supported provider type for GitHub and GitHub Enterprise Cloud

You can use the connections app with GitHub and GitHub Enterprise Cloud.

Supported provider type and versions for GitHub Enterprise Server

You can use the connections app with supported versions of GitHub Enterprise Server. For a list of supported versions, see https://enterprise.github.com/releases/.

> ⚠️ **Important**
>
> AWS CodeConnections does not support deprecated GitHub Enterprise Server versions. For example, AWS CodeConnections does not support GitHub Enterprise Server version 2.22.0 due to a known issue in the release. To connect, upgrade to version 2.22.1 or the latest available version.

Supported provider type for GitLab.com

You can use connections with GitLab.com. For more information, see [Create a connection to GitLab](#).

> ⚠️ **Important**
>
> Connections support for GitLab includes version 15.x and later.

Supported provider type for GitLab self-managed

You can use connections with GitLab self-managed installation (for Enterprise Edition or Community Edition). For more information, see [Create a connection to GitLab self-managed](#).

Product and service integrations with AWS CodeConnections

AWS CodeConnections is integrated with a number of AWS services and partner products and services. Use the information in the following sections to help you configure connections to integrate with the products and services you use.

The following related resources can help you as you work with this service.

Topics

- [Amazon CodeGuru Reviewer](#)
- [Amazon Q Developer](#)
- [Amazon SageMaker](#)
- [AWS App Runner](#)
- [AWS CloudFormation](#)
- [AWS CodeBuild](#)
- [AWS CodePipeline](#)
- [Service Catalog](#)
- [AWS Proton](#)

Amazon CodeGuru Reviewer

[CodeGuru Reviewer](#) is a service for monitoring your repository code. You can use connections to associate the third-party repository that has the code you want to review. For a tutorial where you learn how to configure CodeGuru Reviewer to monitor source code in a GitHub repository so that it can create recommendations that improve the code, see [Tutorial: monitor source code in a GitHub repository](#) in the *Amazon CodeGuru Reviewer User Guide*.

Amazon Q Developer

Amazon Q Developer is a generative AI-powered conversational assistant that can help you to understand, build, extend, and operate AWS applications. For more information, see [What is Amazon Q Developer?](#) in the *Amazon Q Developer User Guide*.

Amazon SageMaker

[Amazon SageMaker](#) is a service for building, training, and deploying machine learning language models. For a tutorial where you configure a connection to your GitHub repository, see [SageMaker](#)

MLOps Project Walkthrough Using Third-party Git Repos in the *Amazon SageMaker Developer Guide*.

AWS App Runner

AWS App Runner is a service that provides a fast, simple, and cost-effective way to deploy from source code or a container image directly to a scalable and secure web application in the AWS Cloud. You can deploy application code from your repository with an App Runner automatic integration and delivery pipeline. You can use connections to deploy your source code to an App Runner service from a private GitHub repository. For more information, see Source code repository providers in the *AWS App Runner Developer Guide*.

AWS CloudFormation

AWS CloudFormation is a service that helps you model and set up your AWS resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS. You create a template that describes all the AWS resources that you want (like Amazon EC2 instances or Amazon RDS DB instances), and CloudFormation takes care of provisioning and configuring those resources for you.

You use connections with Git sync in CloudFormation to create a sync configuration that monitors your Git repository. For a tutorial that walks you through using Git sync for stack deployments, see Working with CloudFormation Git sync in the *CloudFormation User Guide*.

For more information about CloudFormation, see Registering your account to publish CloudFormation extensions in the *CloudFormation Command Line Interface User Guide*.

AWS CodeBuild

AWS CodeBuild is a service for building and testing your code. CodeBuild eliminates the need to provision, manage, and scale your own build servers, and it provides prepackaged build environments for popular programming languages and build tools. For more information about using CodeBuild with connections to GitLab, see GitLab connections in the *AWS CodeBuild User Guide*.

AWS CodePipeline

CodePipeline is a continuous delivery service you can use to model, visualize, and automate the steps required to release your software. You can use connections to configure a third-party repository for CodePipeline source actions.

Learn more:

- See the CodePipeline action configuration reference page for the `SourceConnections` action. To view configuration parameters and an example JSON/YAML snippet, see CodeStarSourceConnection in the *AWS CodePipeline User Guide*.

- To view a **Getting started** tutorial that creates a pipeline with a third-party source repository, see Getting started with connections .

Service Catalog

Service Catalog enables organizations to create and manage catalogs of products that are approved for use on AWS.

When you authorize a connection between your AWS account and an external repository provider, such as GitHub, GitHub Enterprise, or Bitbucket, the connection allows you to sync Service Catalog products to template files that are managed through third-party repositories.

For more information, see Syncing Service Catalog products to template files from GitHub, GitHub Enterprise, or Bitbucket in the *Service Catalog User Guide*.

AWS Proton

AWS Proton is a cloud-based service for deploying to cloud infrastructure. You can use connections to create a link to your third-party repositories for the resources in your templates for AWS Proton. For more information, see Create a link to your repository in the *AWS Proton User Guide*.

Setting up connections

Complete the tasks in this section to get set up for creating and using the connections feature in the Developer Tools console.

Topics

- Sign up for AWS

- Create and apply a policy with permissions to create connections

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 or text message 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*.

Create and apply a policy with permissions to create connections

To use the JSON policy editor to create a policy

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.

2. In the navigation pane on the left, choose **Policies**.

If this is your first time choosing **Policies**, the **Welcome to Managed Policies** page appears. Choose **Get Started**.

3. At the top of the page, choose **Create policy**.

4. In the **Policy editor** section, choose the **JSON** option.

5. Enter the following JSON policy document:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codeconnections:CreateConnection",
                "codeconnections:DeleteConnection",
                "codeconnections:GetConnection",
                "codeconnections:ListConnections",
                "codeconnections:GetInstallationUrl",
                "codeconnections:GetIndividualAccessToken",
                "codeconnections:ListInstallationTargets",
                "codeconnections:StartOAuthHandshake",
                "codeconnections:UpdateConnectionInstallation",
                "codeconnections:UseConnection"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

6. Choose **Next**.

> ⓘ **Note**
>
> You can switch between the **Visual** and **JSON** editor options anytime. However, if you make changes or choose **Next** in the **Visual** editor, IAM might restructure your policy to optimize it for the visual editor. For more information, see Policy restructuring in the *IAM User Guide*.

7. On the **Review and create** page, enter a **Policy name** and a **Description** (optional) for the policy that you are creating. Review **Permissions defined in this policy** to see the permissions that are granted by your policy.

8. Choose **Create policy** to save your new policy.

Getting started with connections

The easiest way to get started with connections is to set up a connection that associates your third-party source repository to your AWS resources. If you wanted to connect your pipeline to an AWS source, such as CodeCommit, you would connect to it as a source action. However, if you have an external repository, you have to create a connection to associate your repository with your pipeline. In this tutorial, you set up a connection with your Bitbucket repository and your pipeline.

In this section, you use connections with:

* AWS CodePipeline: In these steps, you create a pipeline with your Bitbucket repository as the pipeline source.
* [Amazon CodeGuru Reviewer](): Next, you associate your Bitbucket repository to your feedback and analysis tools in CodeGuru Reviewer.

Topics

* [Prerequisites]()
* [Step 1: Edit your source file]()
* [Step 2: Create your pipeline]()
* [Step 3: Associate your repository with CodeGuru Reviewer]()

Prerequisites

Before you begin, complete the steps in [Setting up](). You also need a third-party source repository that you want to connect to your AWS services and allow the connection to manage authentication for you. For example, you might want to connect a Bitbucket repository to your AWS services that integrate with source repositories.

* Create a Bitbucket repository with your Bitbucket account.
* Have your Bitbucket credentials ready. When you use the AWS Management Console to set up a connection, you are asked to sign in with your Bitbucket credentials.

Step 1: Edit your source file

When you create your Bitbucket repository, a default `README.md` file is included, which you will edit.

1. Log in to your Bitbucket repository and choose **Source**.

2. Choose the `README.md` file and choose **Edit** at the top of the page. Delete the existing text and add the following text.

    ```
    This is a Bitbucket repository!
    ```

3. Choose **Commit**.

    Make sure the `README.md` file is at the root level of your repository.

Step 2: Create your pipeline

In this section, you create a pipeline with the following actions:

*   A source stage with a connection to your Bitbucket repository and action.

*   A build stage with an AWS CodeBuild build action.

To create a pipeline with the wizard

1. Sign in to the CodePipeline console at https://console.aws.amazon.com/codepipeline/.

2. On the **Welcome** page, **Getting started** page, or **Pipelines** page, choose **Create pipeline**.

3. In **Step 1: Choose pipeline settings**, in **Pipeline name**, enter `MyBitbucketPipeline`.

4. In **Service role**, choose **New service role**.

> **ⓘ Note**
>
> If you choose instead to use your existing CodePipeline service role, make sure that you have added the `codeconnections:UseConnection` IAM permission to your service role policy. For instructions for the CodePipeline service role, see [Add permissions to the the CodePipeline service role](#).

5. Under **Advanced settings**, leave the defaults. In **Artifact store**, choose **Default location** to use the default artifact store, such as the Amazon S3 artifact bucket designated as the default, for your pipeline in the Region you selected for your pipeline.

> **ⓘ Note**
>
> This is not the source bucket for your source code. This is the artifact store for your pipeline. A separate artifact store, such as an S3 bucket, is required for each pipeline.

Choose **Next**.

6. On the **Step 2: Add source stage** page, add a source stage:

   a. In **Source provider**, choose **Bitbucket**.

   b. Under **Connection**, choose **Connect to Bitbucket**.

   c. On the **Connect to Bitbucket** page, in **Connection name**, enter the name for the connection that you want to create. The name helps you identify this connection later.

      Under **Bitbucket apps**, choose **Install a new app**.

   d. On the app installation page, a message shows that the AWS CodeStar app is trying to connect to your Bitbucket account. Choose **Grant access**. After you have authorized the connection, your repositories on Bitbucket are detected, and you can choose to associate one with your AWS resource.

   e. The connection ID for your new installation is displayed. Choose **Complete connection**. You will be returned to the CodePipeline console.

   f. In **Repository name**, choose the name of your Bitbucket repository.

   g. In **Branch name**, choose the branch for your repository.

   h. Make sure the **Start the pipeline on source code change** option is selected.

   i. Under **Output artifact format**, choose one of the following: **CodePipeline default**.

      • Choose **CodePipeline default** to use the default zip format for artifacts in the pipeline.

      • Choose **Full clone** to include Git metadata about the repository for artifacts in the pipeline. This is only supported for CodeBuild actions.

   Choose **Next**.

7. In **Add build stage**, add a build stage:

    a. In **Build provider**, choose **AWS CodeBuild**. Allow **Region** to default to the pipeline Region.

    b. Choose **Create project**.

    c. In **Project name**, enter a name for this build project.

    d. In **Environment image**, choose **Managed image**. For **Operating system**, choose **Ubuntu**.

    e. For **Runtime**, choose **Standard**. For **Image**, choose **aws/codebuild/standard:5.0**.

    f. For **Service role**, choose **New service role**.

    g. Under **Buildspec**, for **Build specifications**, choose **Insert build commands**. Choose **Switch to editor**, and paste the following under **Build commands**:

```
version: 0.2

phases:
  install:
    #If you use the Ubuntu standard image 2.0 or later, you must specify
 runtime-versions.
    #If you specify runtime-versions and use an image other than Ubuntu
 standard image 2.0, the build fails.
    runtime-versions:
      nodejs: 12
      # name: version
    #commands:
      # - command
      # - command
  pre_build:
    commands:
      - ls -lt
      - cat README.md
  # build:
    #commands:
      # - command
      # - command
  #post_build:
    #commands:
      # - command
      # - command
#artifacts:
  #files:
    # - location
    # - location
```

```
   #name: $(date +%Y-%m-%d)
   #discard-paths: yes
   #base-directory: location
 #cache:
   #paths:
     # - paths
```

h. Choose **Continue to CodePipeline**. This returns to the CodePipeline console and creates a CodeBuild project that uses your build commands for configuration. The build project uses a service role to manage AWS service permissions. This step might take a couple of minutes.

i. Choose **Next**.

8. On the **Step 4: Add deploy stage** page, choose **Skip deploy stage**, and then accept the warning message by choosing **Skip** again. Choose **Next**.

9. On **Step 5: Review**, choose **Create pipeline**.

10. When your pipeline is successfully created, a pipeline execution starts.

11. On your successful build stage, choose **Details**.

    Under **Execution details**, view the CodeBuild build output. The commands output the
    README.md file contents as follows:

    > This is a Bitbucket repository!

Step 3: Associate your repository with CodeGuru Reviewer

After you create a connection, you can use that connection for all of your AWS resources in the
same account. For example, you can use the same Bitbucket connection for a CodePipeline source
action in a pipeline and your repository commit analysis in CodeGuru Reviewer.

1. Sign in to the CodeGuru Reviewer console.

2. Under **CodeGuru Reviewer**, choose **Associate repository**.

   The one-page wizard opens.

3. Under **Select source provider**, choose **Bitbucket**.

4. Under **Connect to Bitbucket (with AWS CodeConnections)**, choose the connection you
   created for your pipeline.

5. Under **Repository location**, choose the name of your Bitbucket repository, and choose
   Associate.

   You can continue to set up code reviews. For more information, see Connecting to Bitbucket to
   associate a repository with CodeGuru Reviewer in the *Amazon CodeGuru Reviewer User Guide*.

Working with connections

Connections are configurations that you use to connect AWS resources to external code
repositories. Each connection is a resource that can be given to services such as AWS CodePipeline

to connect to a third-party repository such as Bitbucket. For example, you can add the connection in CodePipeline so that it triggers your pipeline when a code change is made to your third-party code repository. You can also connect your AWS resources to an installed provider type such as GitHub Enterprise Server.

> **ⓘ Note**
>
> For organizations in GitHub or GitHub Enterprise Server, you cannot install a GitHub App into multiple GitHub Organizations. The app to GitHub Organization mapping is a 1:1 mapping. One organization can only have one app at a time; however, you can have multiple connections pointing to the same app. For more detail, see How connections in AWS CodeConnections work with organizations.

If you want to create a connection to an installed provider type, such as GitHub Enterprise Server, the console creates a host for you. A host is a resource that you create to represent the server where your provider is installed. For more information, see Working with hosts.

When you create a connection, you use a wizard in the console to install the connections app with your third-party provider and associate it with a new connection. If you have already installed the app, you can use it.

> **ⓘ Note**
>
> To use connections in the Europe (Milan) AWS Region, you must:
>
> 1. Install a Region-specific app
>
> 2. Enable the Region
>
> This Region-specific app supports connections in the Europe (Milan) Region. It is published on the third-party provider site, and it is separate from the existing app supporting connections for other Regions. By installing this app, you authorize third-party providers to share your data with the service for this Region only, and you can revoke the permissions at any time by uninstalling the app.
> The service will not process or store your data unless you enable the Region. By enabling this Region, you grant our service permissions to process and store your data.

> Even if the Region is not enabled, third-party providers can still share your data with our
> service if the Region-specific app remains installed, so make sure to uninstall the app once
> you disable the Region. For more information, see Enabling a Region.

For more information about connections, see the AWS CodeConnections API reference. For more
information about the CodePipeline source action for Bitbucket, see CodestarConnectionSource in
the *AWS CodePipeline User Guide*.

To create or attach a policy to your AWS Identity and Access Management (IAM) user or role with
the permissions required to use connections, see AWS CodeConnections permissions reference.
Depending on when your CodePipeline service role was created, you might need to update its
permissions to support AWS CodeConnections. For instructions, see Update the service role in the
AWS CodePipeline User Guide.

Topics

- Create a connection
- Create a connection to Azure DevOps
- Create a connection to Bitbucket
- Create a connection to GitHub
- Create a connection to GitHub Enterprise Server
- Create a connection to GitLab
- Create a connection to GitLab self-managed
- Update a pending connection
- List connections
- Delete a connection
- Tag connections resources
- View connection details
- Share connections with AWS accounts

Create a connection

You can create connections to the following third-party provider types:

- To create a connection to Bitbucket, see Create a connection to Bitbucket.

- To create a connection to GitHub or GitHub Enterprise Cloud, see [Create a connection to GitHub](#).

- To create a connection to GitHub Enterprise Server, including creating your host resource, see [Create a connection to GitHub Enterprise Server](#).

- To create a connection to GitLab, see [Create a connection to GitLab](#).

- To create a connection to Azure DevOps, see [Create a connections to Azure DevOps](#).

> ℹ️ **Note**
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

Create a connection to Azure DevOps

You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI) to create a connection to a repository hosted on Azure DevOps.

Before you begin:

- You must have already created an account with Azure DevOps.

- You must have already created a project and Azure repository on the Azure DevOps portal. Your account must have administrator access to the repository.

> ℹ️ **Note**
>
> You can create connections to an Azure DevOps repository. Installed (on a host) Azure provider types, such as Azure Cloud Hosting, are not supported. See [AWS CodeConnections supported providers and versions](#).

> ℹ️ **Note**
>
> Connections only provide access to repositories owned by the account that was used to create the connection.

Topics

- [Create a connection to Azure DevOps (console)](#)

- [Create a connection to Azure DevOps (CLI)](#)

Create a connection to Azure DevOps (console)

You can use the console to create a connection to Azure DevOps.

> ### ⓘ Note
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

Step 1: Create your connection

1. Sign in to the AWS Management Console, and open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Choose **Settings > Connections**, and then choose **Create connection**.

3. To create a connection to an Azure DevOps repository, under **Select a provider**, choose **Azure DevOps**. In **Connection name**, enter the name for the connection that you want to create. Choose **Connect to Azure DevOps**, and proceed to Step 2.
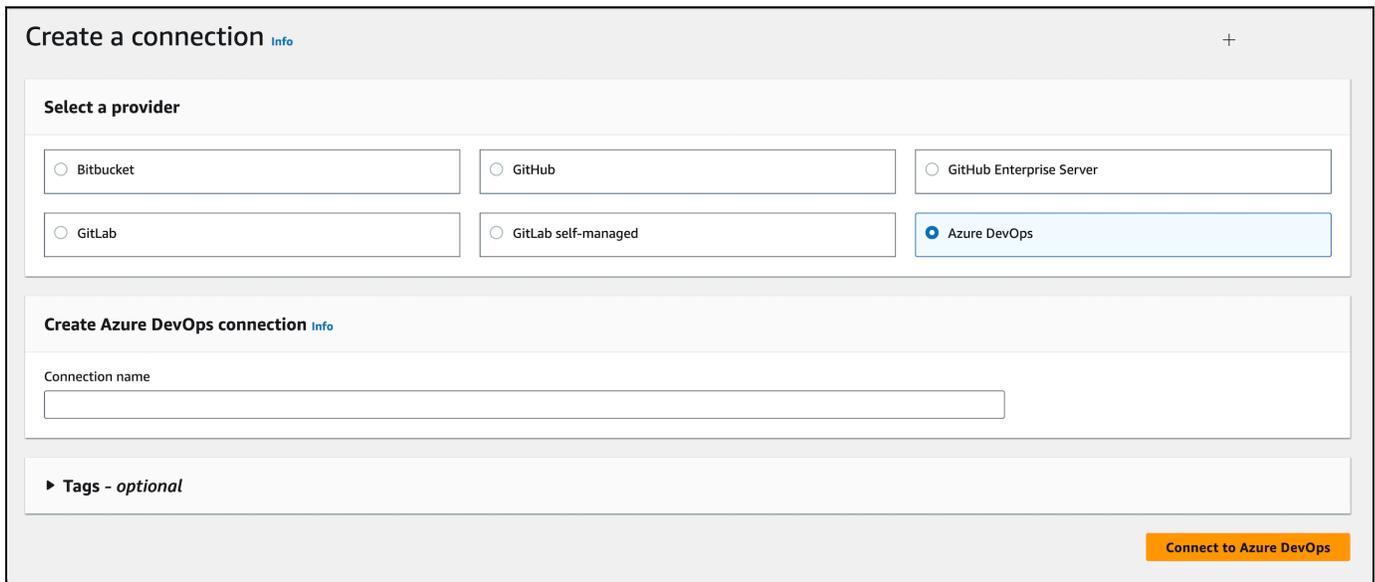
Create a connection Info   +

Select a provider

| ◯ Bitbucket | ◯ GitHub | ◯ GitHub Enterprise Server |
| ◯ GitLab | ◯ GitLab self-managed | ◉ Azure DevOps |

Create Azure DevOps connection Info

Connection name

[   ]

▸ **Tags** – *optional*

                                                                    [Connect to Azure DevOps]

Step 2: Connect to Azure DevOps

1. On the **Connect to Azure DevOps** settings page, your connection name displays.

2. If the login page for Microsoft displays, log in with your credentials and then choose to continue.

    You may need to grant permissions if this is your first time creating a connection to Azure DevOps from AWS Management Console.
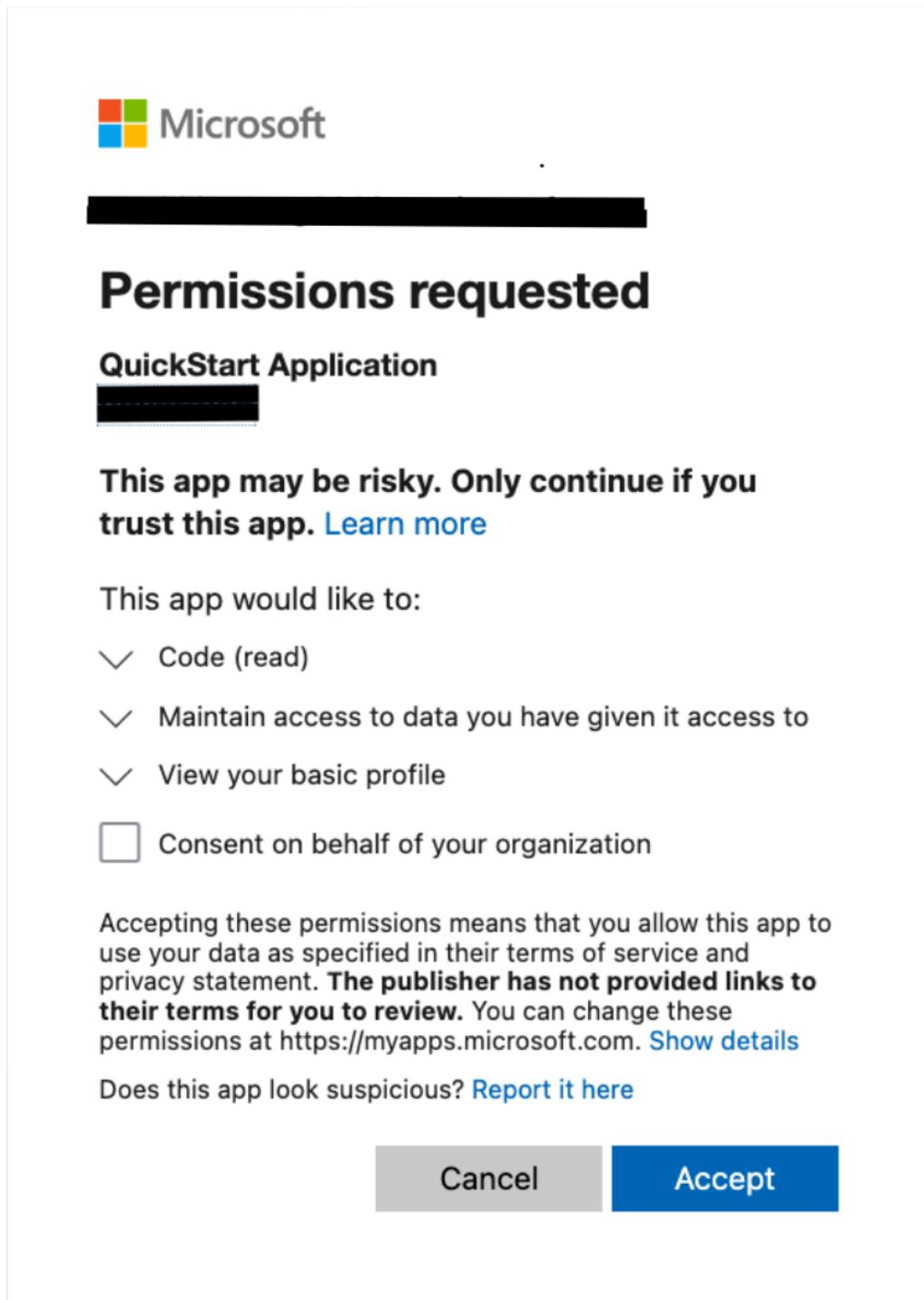
3. Choose **Accept**.

4. On the connection page, the connection ID for your new installation is displayed.

5. Choose **Connect** to establish the connection. The created connection displays in the connections list and is now in available status and ready to use.

Create a connection to Azure DevOps (CLI)

You can use the AWS Command Line Interface (AWS CLI) to create a connection.

To do this, use the **create-connection** command.

> ⚠️ **Important**
>
> A connection created through the AWS CLI or AWS CloudFormation is in PENDING status by default. After you create a connection with the CLI or CloudFormation, use the console to edit the connection to make its status AVAILABLE.

To create a connection to Azure DevOps

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **create-connection** command, specifying the `--provider-type` and `--connection-name` for your connection. In this example, the third-party provider name is AzureDevOps and the specified connection name is MyConnection.

   ```
   aws codeconnections create-connection --provider-type AzureDevOps --connection-name
    MyConnection
   ```

   If successful, this command returns the connection ARN information similar to the following.

   ```
   {
       "ConnectionArn": "arn:aws:codeconnections:us-west-2:account_id:connection/
   aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"
   }
   ```

2. Use the console to complete the connection. For more information, see Update a pending connection.

Create a connection to Bitbucket

You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI) to create a connection to a repository hosted on bitbucket.org.

Before you begin:

- You must have already created an account with Bitbucket.
- You must have already created a code repository on bitbucket.org.

> ⓘ **Note**
>
> You can create connections to a Bitbucket Cloud repository. Installed Bitbucket provider types, such as Bitbucket Server, are not supported. See [AWS CodeConnections supported providers and versions](#).

> ⓘ **Note**
>
> Connections only provide access to repositories owned by the account that was used to create the connection.
> If the application is being installed in a Bitbucket workspace, you need **Administer workspace** permissions. Otherwise, the option to install the app will not display.

Topics

- [Create a connection to Bitbucket (console)](#)
- [Create a connection to Bitbucket (CLI)](#)

Create a connection to Bitbucket (console)

You can use the console to create a connection to Bitbucket.

> ⓘ **Note**
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

Step 1: Create your connection

1. Sign in to the AWS Management Console, and open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.
2. Choose **Settings > Connections**, and then choose **Create connection**.

3. To create a connection to a Bitbucket repository, under **Select a provider**, choose **Bitbucket**. In **Connection name**, enter the name for the connection that you want to create. Choose **Connect to Bitbucket**, and proceed to Step 2.

Step 2: Connect to Bitbucket

1. On the **Connect to Bitbucket** settings page, your connection name displays.

   Under **Bitbucket apps**, choose an app installation or choose **Install a new app** to create one.

   > **ⓘ Note**
   >
   > You only install the app once for each Bitbucket workspace or account. If you have already installed the Bitbucket app, choose it and move to the last step in this section.

2. If the login page for Bitbucket displays, log in with your credentials and then choose to continue.

3. On the app installation page, a message shows that the AWS CodeStar app is trying to connect to your Bitbucket account.

    If you are using a Bitbucket workspace, change the **Authorize for** option to the workspace. Only workspaces where you have administrator access will display.

    Choose **Grant access**.

AWS CodeStar requests access

This app is hosted at https://codestar-connections.webhooks.aws

Read your account information

Read your repositories and their pull requests

Administer your repositories

Read and modify your repositories

Authorize for

Allow AWS CodeStar to do this?

This 3rd party vendor has not provided a privacy policy or terms of use.
Atlassian's Privacy Policy is not applicable to the use of this App.

Grant access   Cancel

4. In **Bitbucket apps**, the connection ID for your new installation is displayed. Choose **Connect**. The created connection displays in the connections list.

Create a connection to Bitbucket (CLI)

You can use the AWS Command Line Interface (AWS CLI) to create a connection.

To do this, use the **create-connection** command.

> ⚠️ **Important**
>
> A connection created through the AWS CLI or AWS CloudFormation is in PENDING status by default. After you create a connection with the CLI or CloudFormation, use the console to edit the connection to make its status AVAILABLE.

To create a connection to Bitbucket

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **create-connection** command, specifying the --provider-type and --connection-name for your connection. In this example, the third-party provider name is Bitbucket and the specified connection name is MyConnection.

   ```
   aws codeconnections create-connection --provider-type Bitbucket --connection-name
    MyConnection
   ```

   If successful, this command returns the connection ARN information similar to the following.

   ```
   {
       "ConnectionArn": "arn:aws:codeconnections:us-west-2:account_id:connection/
   aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"
   }
   ```

2. Use the console to complete the connection. For more information, see Update a pending connection.

Create a connection to GitHub

You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI) to create a connection to GitHub.

Before you begin:

- You must have already created an account with GitHub.

- You must have already created your third-party code repository.

> ### ⓘ Note
>
> To create the connection, you must be the GitHub organization owner. For repositories that are not under an organization, you must be the repository owner.

Topics

- [Create a connection to GitHub (console)](#)

- [Create a connection to GitHub (CLI)](#)

Create a connection to GitHub (console)

You can use the console to create a connection to GitHub.

> ### ⓘ Note
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

1. Sign in to the AWS Management Console, and open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Choose **Settings > Connections**, and then choose **Create connection**.

3. To create a connection to a GitHub or GitHub Enterprise Cloud repository, under **Select a provider**, choose **GitHub**. In **Connection name**, enter the name for the connection that you want to create. Choose **Connect to GitHub**, and proceed to Step 2.

To create a connection to GitHub

1. Under **GitHub connection settings**, your connection name appears in **Connection name**. Choose **Connect to GitHub**. The access request page appears.
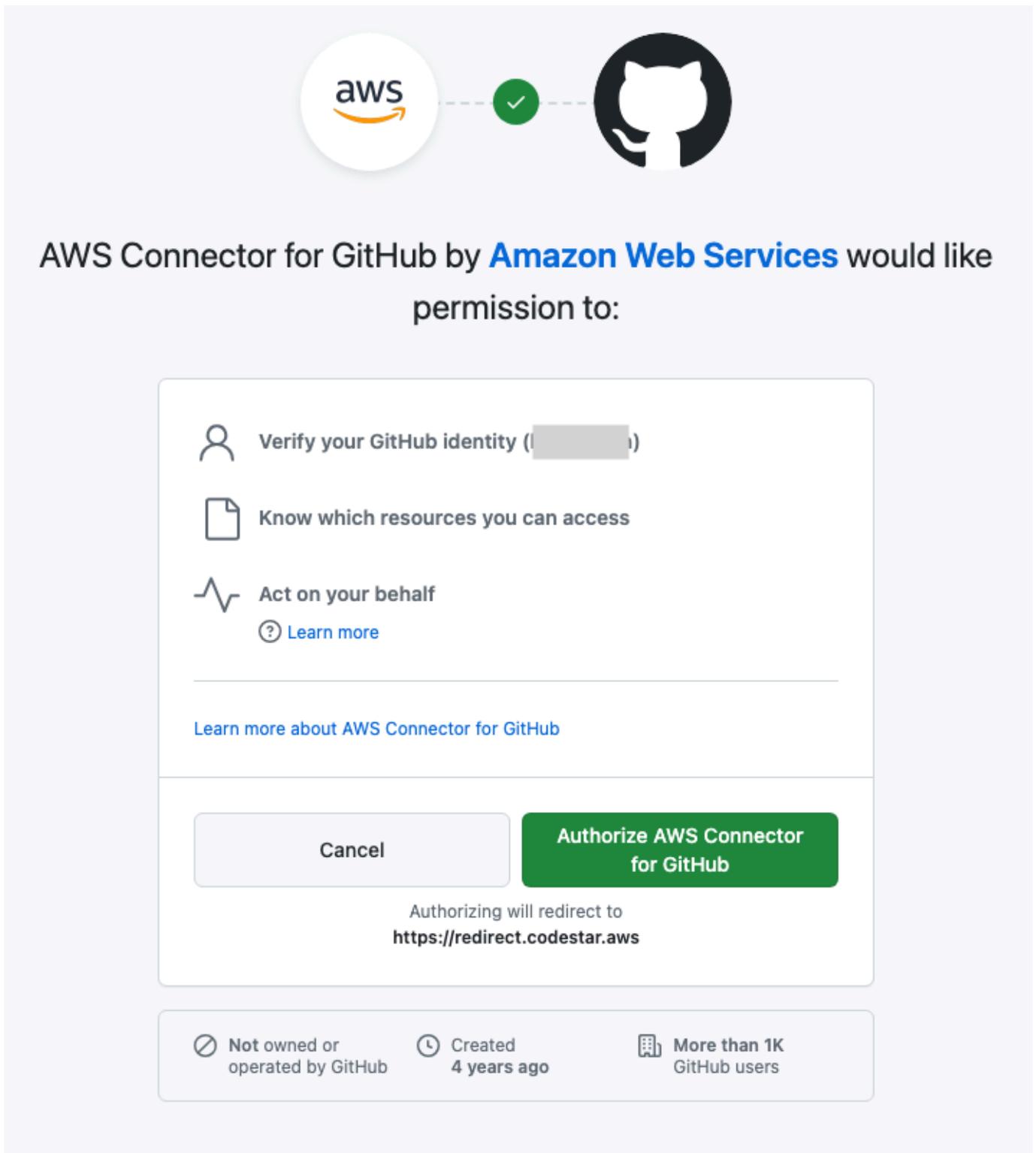
2. Choose **Authorize AWS Connector for GitHub**. The connection page displays and shows the
    GitHub Apps field.

3. Under **GitHub Apps**, choose an app installation or choose **Install a new app** to create one.

> **ⓘ Note**
>
> You install one app for all of your connections to a particular provider. If you have already installed the AWS Connector for GitHub app, choose it and skip this step.

4. On the Install **AWS Connector for GitHub** page, choose the account where you want to install the app.
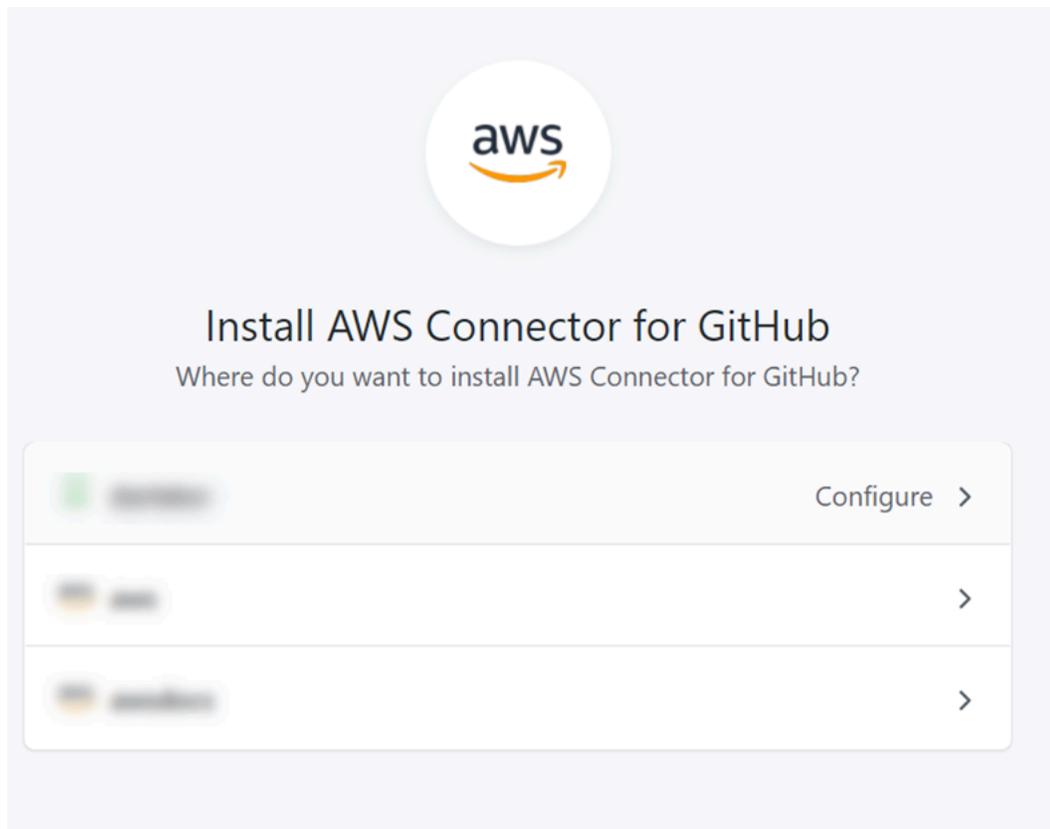
> **ⓘ Note**
>
> You only install the app once for each GitHub account. If you previously installed the app, you can choose **Configure** to proceed to a modification page for your app installation, or you can use the back button to return to the console.

5. On the **Install AWS Connector for GitHub** page, leave the defaults, and choose **Install**.

After this step, an updated permissions page might display in GitHub.

6. If a page displays showing that there are updated permissions for the AWS Connector for GitHub app, choose **Accept new permissions**.
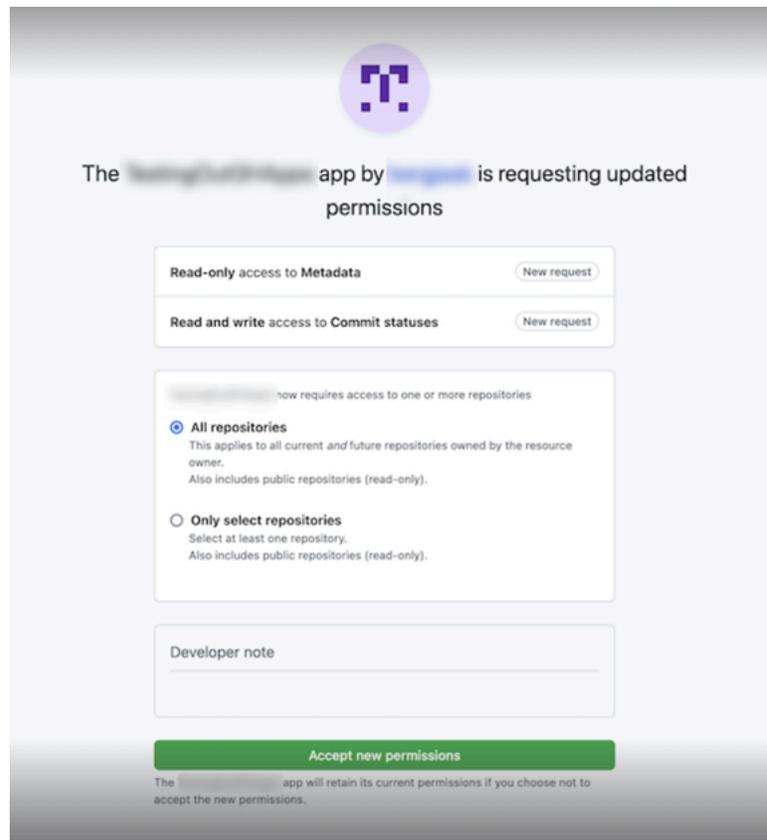
7. You are returned to the **Connect to GitHub** page. The connection ID for your new installation appears in **GitHub Apps**. Choose **Connect**.

View your created connection

*   The created connection displays in the connections list.
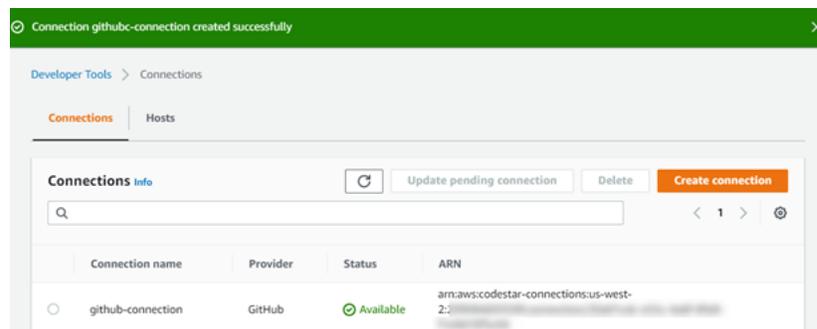
Create a connection to GitHub (CLI)

You can use the AWS Command Line Interface (AWS CLI) to create a connection to GitHub.

To do this, use the **create-connection** command.

> ⚠️ **Important**
>
> A connection created through the AWS CLI or AWS CloudFormation is in PENDING status by default. After you create a connection with the CLI or CloudFormation, use the console to edit the connection to make its status AVAILABLE.

To create a connection to GitHub

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **create-connection** command, specifying the --provider-type and --connection-name for your connection. In this example, the third-party provider name is GitHub and the specified connection name is MyConnection.

   ```
   aws codeconnections create-connection --provider-type GitHub --connection-name
    MyConnection
   ```

   If successful, this command returns the connection ARN information similar to the following.

   ```
   {
       "ConnectionArn": "arn:aws:codeconnections:us-west-2:account_id:connection/
   aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"
   }
   ```

2. Use the console to complete the connection. For more information, see Update a pending connection.

Create a connection to GitHub Enterprise Server

You use connections to associate your AWS resources with a third-party repository. You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI) to create a connection to GitHub Enterprise Server.

Connections only provide access to repositories owned by the GitHub Enterprise Server account that is used during connection creation to authorize installation of the GitHub app.

Before you begin:

- You must already have a GitHub Enterprise Server instance and a repository in it.

- You need to be an administrator of the GitHub Enterprise Server instance in order to create GitHub apps and create a host resource as shown in this section.

> ⚠️ **Important**
>
> When you set up your host for GitHub Enterprise Server, a VPC endpoint for webhooks event data is created for you. If you created your host before November 24, 2020, and you want to use VPC PrivateLink webhook endpoints, you must first delete your host and then create a new host.

> ⓘ **Note**
>
> For organizations in GitHub Enterprise Server or GitLab self-managed, you don't pass an available host. You create a new host for each connection in your organization, and you must be sure to enter the same information in the network fields (VPC ID, Subnet IDs, and Security Group IDs) for the host. For more information, see Connection and host setup for installed providers supporting organizations.

Topics

- Create a connection to GitHub Enterprise Server (console)
- Create a connection to GitHub Enterprise Server (CLI)

Create a connection to GitHub Enterprise Server (console)

To create a GitHub Enterprise Server connection, you provide information for where your GitHub Enterprise Server is installed and authorize the connection creation with your GitHub Enterprise credentials.

> ⓘ **Note**
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

Topics

- [Create your GitHub Enterprise Server connection (console)](#)

Create your GitHub Enterprise Server connection (console)

To create a connection to GitHub Enterprise Server, have your server URL and GitHub Enterprise credentials ready.

To create a host

1. Sign in to the AWS Management Console, and open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. On the **Hosts** tab, choose **Create host**.

3. In **Host name**, enter the name you want to use for your host.

4. In **Select a provider**, choose one of the following:

   - **GitHub Enterprise Server**

   - **GitLab self-managed**

5. In **URL**, enter the endpoint for the infrastructure where your provider is installed.

6. If your server is configured within an Amazon VPC and you want to connect with your VPC, choose **Use a VPC**. Otherwise, choose **No VPC**.

7. If you have launched your instance into an Amazon VPC and you want to connect with your VPC, choose **Use a VPC** and complete the following.

   a. In **VPC ID**, choose your VPC ID. Make sure to choose the VPC for the infrastructure where your instance is installed or a VPC with access to your instance through VPN or Direct Connect.

   b. If you have a private VPC configured, and you have configured your instance to perform TLS validation using a non-public certificate authority, in **TLS certificate**, enter your certificate ID. The TLS Certificate value is the public key of the certificate.

8. Choose **Create host**.

9. After the host details page displays, the host status changes as the host is created.

> ⓘ **Note**
>
> If your host setup includes a VPC configuration, allow several minutes for provisioning
> of host network components.

Wait for your host to reach a **Pending** status, and then complete the setup. For more
information, see Set up a pending host.
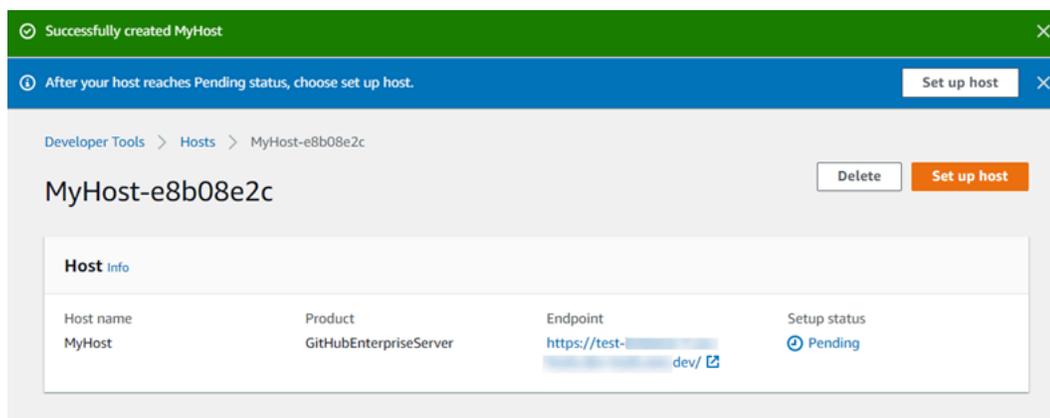
Step 2: Create your connection to GitHub Enterprise Server (console)

1. Sign in to the AWS Management Console and open the Developer Tools console at https://
    console.aws.amazon.com/codesuite/settings/connections.

2. Choose **Settings > Connections**, and then choose **Create connection**.

3. To create a connection to an installed GitHub Enterprise Server repository, choose **GitHub
    Enterprise Server**.

Connect to GitHub Enterprise Server

1. In **Connection name**, enter the name for your connection.

2. In **URL**, enter the endpoint for your server.

    > **ⓘ Note**
    >
    > If the provided URL has already been used to set up a GitHub Enterprise Server for a connection, you will be prompted to choose the host resource ARN that was created previously for that endpoint.

3. (Optional) If you have launched your server into an Amazon VPC and you want to connect with your VPC, choose **Use a VPC** and complete the following.

    > **ⓘ Note**
    >
    > For organizations in GitHub Enterprise Server or GitLab self-managed, you don't pass an available host. You create a new host for each connection in your organization, and you must be sure to enter the same information in the network fields (VPC ID, Subnet

IDs, and Security Group IDs) for the host. For more information, see [Connection and host setup for installed providers supporting organizations](#).

a.   In **VPC ID**, choose your VPC ID. Make sure to choose the VPC for the infrastructure where your GitHub Enterprise Server instance is installed or a VPC with access to your GitHub Enterprise Server instance through VPN or Direct Connect.

b.   Under **Subnet ID**, choose **Add**. In the field, choose the subnet ID you want to use for your host. You can choose up to 10 subnets.

Make sure to choose the subnet for the infrastructure where your GitHub Enterprise Server instance is installed or a subnet with access to your installed GitHub Enterprise Server instance through VPN or Direct Connect.

c.   Under **Security group IDs**, choose **Add**. In the field, choose the security group you want to use for your host. You can choose up to 10 security groups.

Make sure to choose the security group for the infrastructure where your GitHub Enterprise Server instance is installed or a security group with access to your installed GitHub Enterprise Server instance through VPN or Direct Connect.

d.   If you have a private VPC configured, and you have configured your GitHub Enterprise Server instance to perform TLS validation using a non-public certificate authority, in **TLS certificate**, enter your certificate ID. The TLS Certificate value should be the public key of the certificate.

VPC ID

Choose the VPC in which your GitHub Enterprise Server is configured.

| Q vpc-09⬛⬛⬛⬛ | ✕ |

Subnet IDs

Choose the subnet or subnets for the VPC in which your GitHub Enterprise Server is configured.

Subnet ID

| Q subnet-7d⬛⬛⬛⬛ | ✕ | | Remove |

Add

Security group IDs

Choose the security group or groups for the VPC in which your GitHub Enterprise Server is configured.

Security group ID

| Q sg-00⬛⬛⬛⬛⬛⬛ | ✕ | | Remove |

Add

TLS certificate - *optional*

If you have a private certificate authority behind a VPC or you are using a self-signed certificate paste the TLS certificate here.

4. Choose **Connect to GitHub Enterprise Server**. The created connection is shown with a **Pending** status. A host resource is created for the connection with the server information you provided. For the host name, the URL is used.

5. Choose **Update pending connection.**
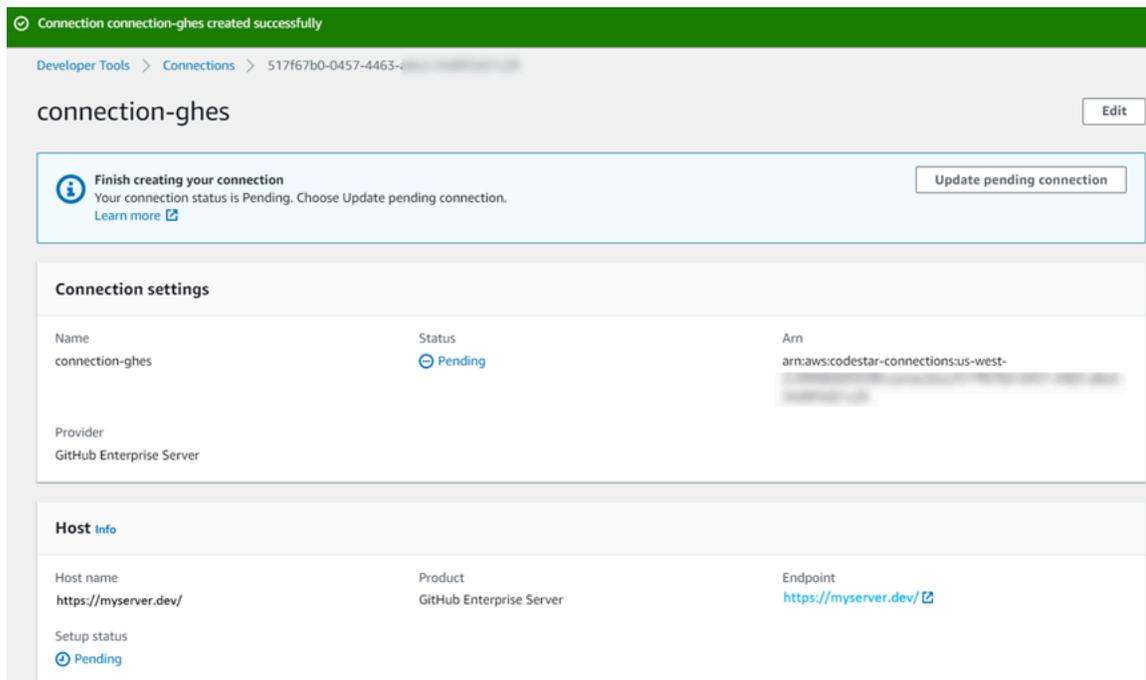
6. If prompted, on the GitHub Enterprise login page, sign in with your GitHub Enterprise credentials.

7. On the **Create GitHub App** page, choose a name for your app.
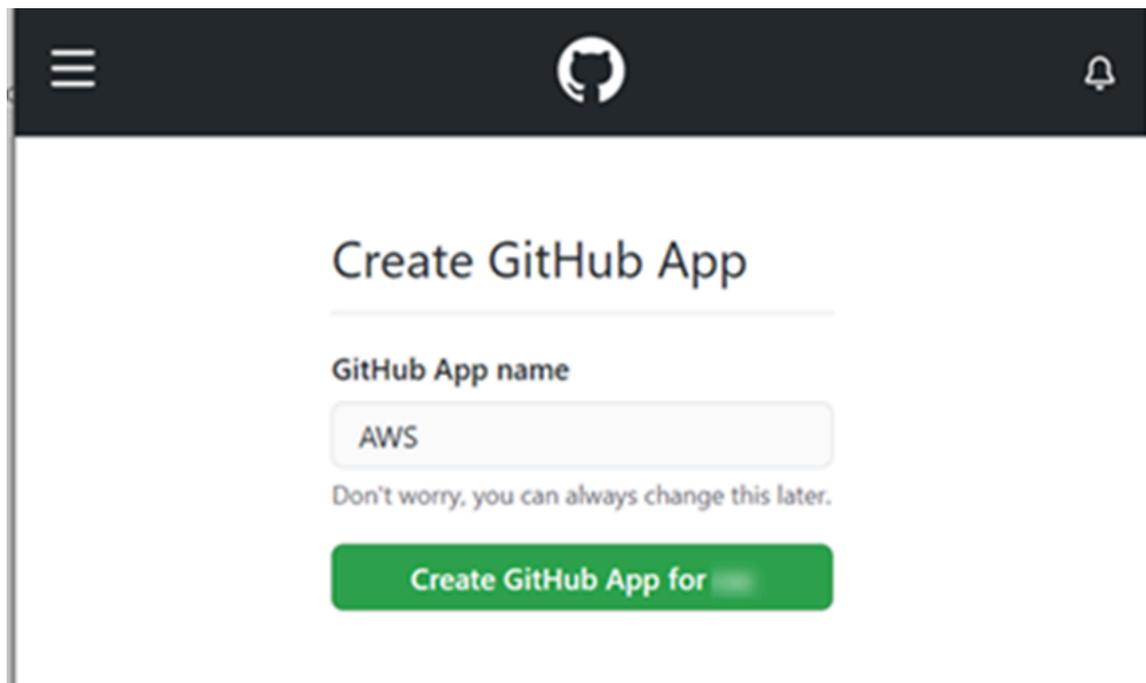
8. On the GitHub authorization page, choose **Authorize <app-name>**.
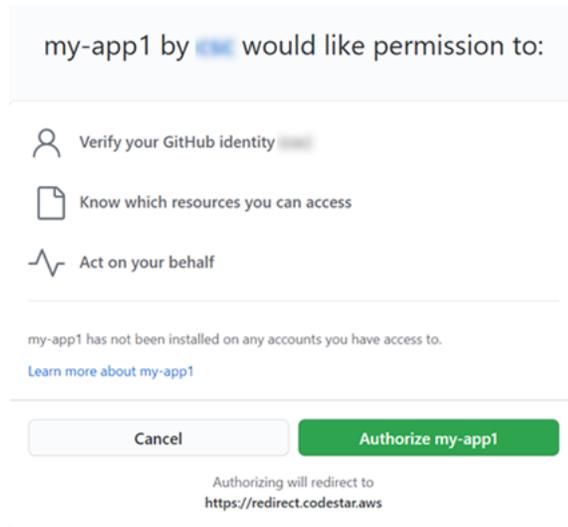
9. On the app installation page, a message shows that the connector app is ready to be installed. If you have multiple organizations, you might be prompted to choose the organization where you want to install the app.

   Choose the repository settings where you want to install the app. Choose **Install**.
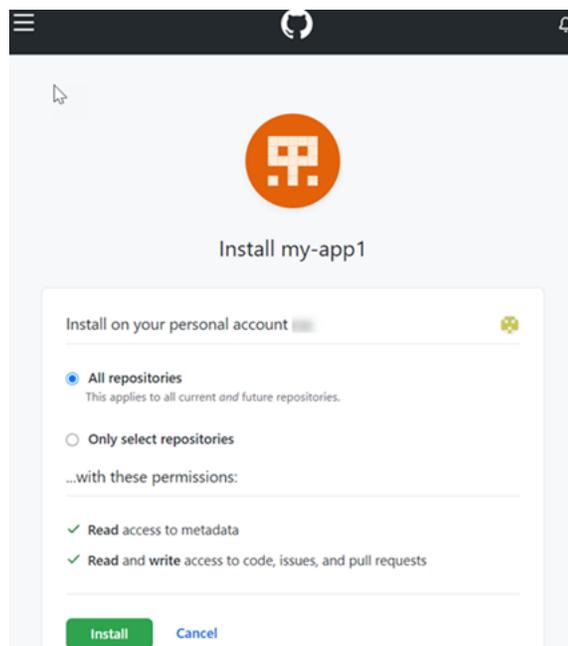
10. The connection page shows the created connection in an **Available** status.

Create a connection to GitHub Enterprise Server (CLI)

You can use the AWS Command Line Interface (AWS CLI) to create a connection.

To do this, use the **create-host** and the **create-connection** commands.

> ⚠️ **Important**
>
> A connection created through the AWS CLI or AWS CloudFormation is in PENDING status by default. After you create a connection with the CLI or CloudFormation, use the console to edit the connection to make its status AVAILABLE.

Step 1: To create a host for GitHub Enterprise Server (CLI)

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **create-host** command, specifying the --name, --provider-type, and --provider-endpoint for your connection. In this example, the third-party provider name is GitHubEnterpriseServer and the endpoint is my-instance.dev.

   ```
   aws codeconnections create-host --name MyHost --provider-type
     GitHubEnterpriseServer --provider-endpoint "https://my-instance.dev"
   ```

   If successful, this command returns the host Amazon Resource Name (ARN) information similar to the following.

   ```
   {
       "HostArn": "arn:aws:codeconnections:us-west-2:account_id:host/My-Host-28aef605"
   }
   ```

   After this step, the host is in PENDING status.

2. Use the console to complete the host setup and move the host to an Available status. For more information, see [Set up a pending host](#).

Step 2: To set up a pending host in the console

1. Sign in to the AWS Management Console and open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Use the console to complete the host setup and move the host to an Available status. See [Set up a pending host](#).

Step 3: To create a connection for GitHub Enterprise Server (CLI)

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **create-connection** command, specifying the `--host-arn` and `--connection-name` for your connection.

   ```
   aws codeconnections create-connection --host-arn arn:aws:codeconnections:us-
   west-2:account_id:host/MyHost-234EXAMPLE --connection-name MyConnection
   ```

   If successful, this command returns the connection ARN information similar to the following.

   ```
   {
       "ConnectionArn": "arn:aws:codeconnections:us-west-2:account_id:connection/
   aEXAMPLE-8aad"
   }
   ```

2. Use the console to set up the pending connection. For more information, see Update a pending connection.

Step 4: To complete a connection for GitHub Enterprise Server in the console

1. Sign in to the AWS Management Console and open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Use the console to set up the pending connection and move the connection to an `Available` status. For more information, see Update a pending connection.

Create a connection to GitLab

You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI) to create a connection to a repository hosted on gitlab.com.

> ⓘ **Note**
>
> By authorizing this connection installation in GitLab, you grant our service permissions to process your data, and you can revoke the permissions at any time by uninstalling the application.

Before you begin:

- You must have already created an account with GitLab.

> **ⓘ Note**
>
> Connections only provide access for the account that was used to create and authorize the connection.

> **ⓘ Note**
>
> You can create connections where you have the **Owner** role in GitLab, and then the connection can be used with the repository with resources such as CodePipeline. For repositories in groups, you do not need to be the group owner.

Topics

- [Create a connection to GitLab (console)](#)
- [Create a connection to GitLab (CLI)](#)

Create a connection to GitLab (console)

You can use the console to create a connection.

> **ⓘ Note**
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

Step 1: Create your connection

1. Sign in to the AWS Management Console, and then open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Choose **Settings**, and then choose **Connections**. Choose **Create connection**.

3. To create a connection to a GitLab repository, under **Select a provider**, choose **GitLab**. In **Connection name**, enter the name for the connection that you want to create. Choose **Connect to GitLab**.

Developer Tools > Connections > Create connection

Create a connection Info

Select a provider

○ Bitbucket   ○ GitHub   ○ GitHub Enterprise Server

● GitLab

Create GitLab connection Info

Connection name

▶ **Tags - *optional***

Connect to GitLab

4. When the sign-in page for GitLab displays, log in with your credentials and then choose **Sign in**.

5. An authorization page displays with a message requesting authorization for the connection to access your GitLab account.

   Choose **Authorize**.

Authorize codestar-connections to use your account?

An application called codestar-connections is requesting access to your GitLab account. This application was created by Amazon AWS. Please note that this application is not provided by GitLab and you should verify its authenticity before allowing access.

This application will be able to:

- **Access the authenticated user's API**
  Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.
- **Read the authenticated user's personal information**
  Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- **Read Api**
  Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- **Allows read-only access to the repository**
  Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- **Allows read-write access to the repository**
  Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

Deny   Authorize

6. The browser returns to the connections console page. Under **Create GitLab connection**, the new connection is shown in **Connection name**.

7. Choose **Connect to GitLab**.

After the connection is created successfully, a success banner displays. The connection details are shown on the **Connection settings** page.

Create a connection to GitLab (CLI)

You can use the AWS Command Line Interface (AWS CLI) to create a connection.

To do this, use the **create-connection** command.

> ⚠️ **Important**
>
> A connection created through the AWS CLI or AWS CloudFormation is in PENDING status by default. After you create a connection with the CLI or CloudFormation, use the console to edit the connection to make its status AVAILABLE.

To create a connection to GitLab

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **create-connection** command, specifying the `--provider-type` and `--connection-name` for your connection. In this example, the third-party provider name is GitLab and the specified connection name is MyConnection.

   ```
   aws codeconnections create-connection --provider-type GitLab --connection-name
    MyConnection
   ```

   If successful, this command returns the connection ARN information similar to the following.

   ```
   {
       "ConnectionArn": "arn:aws:codeconnections:us-west-2:account_id:connection/
   aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"
   }
   ```

2. Use the console to complete the connection. For more information, see Update a pending connection.

Create a connection to GitLab self-managed

You can create connections for GitLab Enterprise Edition or GitLab Community Edition with a self-managed installation.

You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI) to create a connection and host for GitLab self-managed.

> **ⓘ Note**
>
> By authorizing this connection application in GitLab self-managed, you grant our service permissions to process your data, and you can revoke the permissions at any time by uninstalling the application.

Before you create a connection to GitLab self-managed, you must create a host to use for the connection, as detailed in these steps. For an overview of the host creation workflow for installed providers, see Workflow to create or update a host.

You can optionally configure your host with a VPC. For more information about network and VPC configuration for your host resource, see the VPC prerequisites in (Optional) Prerequisites: Network or Amazon VPC configuration for your connection and Troubleshooting VPC configuration for your host.

Before you begin:

- You must have already created an account with GitLab and have GitLab Enterprise Edition or GitLab Community Edition with a self-managed installation. For more information, see https://docs.gitlab.com/ee/subscriptions/self_managed/.

  > **ⓘ Note**
  >
  > Connections only provide access for the account that was used to create and authorize the connection.

> **ⓘ Note**
>
> You can create connections to a repository where you have the **Owner** role in GitLab, and then the connection can be used with with resources such as CodePipeline. For repositories in groups, you do not need to be the group owner.

- You must have already created a GitLab personal access token (PAT) with the following scoped-down permission only: `api`, `admin_mode`. For more information, see https://docs.gitlab.com/ee/user/profile/personal_access_tokens.html. You must be an administrator to create and use the PAT.

> **ⓘ Note**
>
> Your PAT is used to authorize the host and is not otherwise stored or used by connections. To set up a host, you can create a temporary PAT and then after you set up the host, you can delete the PAT.

> **ⓘ Note**
>
> For organizations in GitHub Enterprise Server or GitLab self-managed, you don't pass an available host. You create a new host for each connection in your organization, and you must be sure to enter the same information in the network fields (VPC ID, Subnet IDs, and Security Group IDs) for the host. For more information, see [Connection and host setup for installed providers supporting organizations](#).

Topics

- [Create a connection to GitLab self-managed (console)](#)
- [Create a connection to GitLab self-managed (CLI)](#)

Create a connection to GitLab self-managed (console)

Use these steps to create a host and a connection to GitLab self-managed in the console. For considerations for setting up a host in a VPC, see (Optional) Prerequisites: Network or Amazon VPC configuration for your connection.

> **ⓘ Note**
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

> **ⓘ Note**
>
> You create a host for a single GitLab self-managed installation, and then you can manage one or more GitLab self-managed connections to that host.

Step 1: Create your host

1. Sign in to the AWS Management Console, and then open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. On the **Hosts** tab, choose **Create host**.

3. In **Host name**, enter the name you want to use for your host.

4. In **Select a provider**, choose **GitLab self-managed**.

5. In **URL**, enter the endpoint for the infrastructure where your provider is installed.

6. If your server is configured within an Amazon VPC and you want to connect with your VPC, choose **Use a VPC**. Otherwise, choose **No VPC**.

7. (Optional) If you have launched your host into an Amazon VPC and you want to connect with your VPC, choose **Use a VPC** and complete the following.

    > **ⓘ Note**
    >
    > For organizations in GitHub Enterprise Server or GitLab self-managed, you don't pass an available host. You create a new host for each connection in your organization, and you must be sure to enter the same information in the network fields (VPC ID, Subnet

> IDs, and Security Group IDs) for the host. For more information, see Connection and host setup for installed providers supporting organizations.

    a.   In **VPC ID**, choose your VPC ID. Make sure to choose the VPC for the infrastructure where your host is installed or a VPC with access to your instance through VPN or Direct Connect.

    b.   If you have a private VPC configured, and you have configured your host to perform TLS validation using a non-public certificate authority, in **TLS certificate**, enter your certificate ID. The TLS Certificate value is the public key of the certificate.

8. Choose **Create host**.

9. After the host details page displays, the host status changes as the host is created.

> ⓘ **Note**
>
> If your host setup includes a VPC configuration, allow several minutes for provisioning of host network components.

Wait for your host to reach a **Pending** status, and then complete the setup. For more information, see Set up a pending host.

Step 2: Set up your pending host

1. Choose **Set up host**.

2. A **Set up *host_name*** page displays. In **Provide personal access token**, provide your GitLab PAT with the following scoped-down permissions only: `api` and `admin_mode`.

   > (i) **Note**
   >
   > Only an administrator can create and use the PAT.

3. After your host is successfully registered, the host details page appears and shows that the host status is **Available**.

Step 3: Create your connection

1. Sign in to the AWS Management Console, and then open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Choose **Settings**, and then choose **Connections**. Choose **Create connection**.

3. To create a connection to a GitLab repository, under **Select a provider**, choose **GitLab self-managed**. In **Connection name**, enter the name for the connection that you want to create.

4. In **URL**, enter the endpoint for your server.

5. If you have launched your server into an Amazon VPC and you want to connect with your VPC, choose **Use a VPC** and complete the following.

   a. In **VPC ID**, choose your VPC ID. Make sure to choose the VPC for the infrastructure where your host is installed or a VPC with access to your host through VPN or Direct Connect.

   b. Under **Subnet ID**, choose **Add**. In the field, choose the subnet ID you want to use for your host. You can choose up to 10 subnets.

Make sure to choose the subnet for the infrastructure where your host is installed or a subnet with access to your installed host through VPN or Direct Connect.

c.   Under **Security group IDs**, choose **Add**. In the field, choose the security group you want to use for your host. You can choose up to 10 security groups.

Make sure to choose the security group for the infrastructure where your host is installed or a security group with access to your installed host through VPN or Direct Connect.

d.   If you have a private VPC configured, and you have configured your host to perform TLS validation using a non-public certificate authority, in **TLS certificate**, enter your certificate ID. The TLS Certificate value should be the public key of the certificate.

6.   Choose **Connect to GitLab self-managed**. The created connection is shown with a **Pending** status. A host resource is created for the connection with the server information you provided. For the host name, the URL is used.

7.   Choose **Update pending connection.**

8.   When the sign-in page for GitLab displays, log in with your credentials and then choose **Sign in**.

9.   An authorization page displays with a message requesting authorization for the connection to access your GitLab account.

Choose **Authorize**.

10. The browser returns to the connections console page. Under **Create GitLab connection**, the new connection is shown in **Connection name**.

11. Choose **Connect to GitLab self-managed**.

After the connection is created successfully, a success banner displays. The connection details are shown on the **Connection settings** page.

Create a connection to GitLab self-managed (CLI)

You can use the AWS Command Line Interface (AWS CLI) to create a host and connection for GitLab self-managed.

To do this, use the **create-host** and the **create-connection** commands.

> **⚠ Important**
>
> A connection created through the AWS CLI or AWS CloudFormation is in PENDING status by default. After you create a connection with the CLI or CloudFormation, use the console to edit the connection to make its status AVAILABLE.

Step 1: To create a host for GitLab self-managed (CLI)

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **create-host** command, specifying the --name, --provider-type, and --provider-endpoint for your connection. In this example, the third-party provider name is GitLabSelfManaged and the endpoint is my-instance.dev.

   ```
   aws codeconnections create-host --name MyHost --provider-type GitLabSelfManaged --provider-endpoint "https://my-instance.dev"
   ```

   If successful, this command returns the host Amazon Resource Name (ARN) information similar to the following.

   ```
   {
       "HostArn": "arn:aws:codeconnections:us-west-2:account_id:host/My-Host-28aef605"
   }
   ```

   After this step, the host is in PENDING status.

2. Use the console to complete the host setup and move the host to an Available status in the following step.

Step 2: To set up a pending host in the console

1. Sign in to the AWS Management Console and open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Use the console to complete the host setup and move the host to an Available status. See Set up a pending host.

Step 3: To create a connection for GitLab self-managed (CLI)

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **create-connection** command, specifying the `--host-arn` and `--connection-name` for your connection.

```
aws codeconnections create-connection --host-arn arn:aws:codeconnections:us-
west-2:account_id:host/MyHost-234EXAMPLE --connection-name MyConnection
```

If successful, this command returns the connection ARN information similar to the following.

```
{
    "ConnectionArn": "arn:aws:codeconnections:us-west-2:account_id:connection/
aEXAMPLE-8aad"
}
```

2. Use the console to set up the pending connection in the following step.

Step 4: To complete a connection for GitLab self-managed in the console

1. Sign in to the AWS Management Console and open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Use the console to set up the pending connection and move the connection to an `Available` status. For more information, see Update a pending connection.

Update a pending connection

A connection created through the AWS Command Line Interface (AWS CLI) or AWS CloudFormation is in PENDING status by default. After you create a connection with the AWS CLI or CloudFormation, use the console to update the connection to make its status AVAILABLE.

> ⓘ **Note**
>
> You must use the console to update a pending connection. You cannot update a pending connection using the AWS CLI.

The first time you use the console to add a new connection to a third-party provider, you must complete the OAuth handshake with the third-party provider using the installation associated with your connection.

You can use the Developer Tools console to complete a pending connection.

To complete a connection

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Choose **Settings > Connections**.

   The names of all connections associated with your AWS account are displayed.

3. In **Name**, choose the name of the pending connection you want to update.

   Update a pending connection is enabled when you choose a connection with a **Pending** status.

4. Choose **Update a pending connection**.

5. On the **Connect to Bitbucket** page, in **Connection name**, verify the name of your connection.

   Under **Bitbucket apps**, choose an app installation, or choose **Install a new app** to create one.

6. On the app installation page, a message shows that the AWS CodeStar app is trying to connect to your Bitbucket account. Choose **Grant access**.

7. The connection ID for your new installation is displayed. Choose **Complete connection**.

List connections

You can use the Developer Tools console or the **list-connections** command in the AWS Command Line Interface (AWS CLI) to view a list of connections in your account.

List connections (console)

To list connections

1. Open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Choose **Settings > Connections**.

3. View the name, status, and ARN for your connections.

List connections (CLI)

You can use the AWS CLI to list your connections to third-party code repositories. For a connection associated to a host resource, such as connections to GitHub Enteprise Server, the output additionally returns the host ARN.

To do this, use the **list-connections** command.

To list connections

*   Open a terminal (Linux, macOS, or Unix) or command prompt (Windows), and use the AWS CLI to run the **list-connections** command.

    ```
    aws codeconnections list-connections --provider-type Bitbucket
    --max-results 5 --next-token: next-token
    ```

    This command returns the following output.

    ```
    {
        "Connections": [
            {
                "ConnectionName": "my-connection",
                "ProviderType": "Bitbucket",
                "Status": "PENDING",
                "ARN": "arn:aws:codeconnections:us-west-2:account_id:connection/
    aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
                "OwnerAccountId": "account_id"
            },
            {
                "ConnectionName": "my-other-connection",
                "ProviderType": "Bitbucket",
                "Status": "AVAILABLE",
                "ARN": "arn:aws:codeconnections:us-west-2:account_id:connection/
    aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
                "OwnerAccountId": "account_id"
            },
        ],
        "NextToken": "next-token"
    }
    ```

Delete a connection

You can use the Developer Tools console or the **delete-connection** command in the AWS Command Line Interface (AWS CLI) to delete a connection.

Topics

- [Delete a connection (console)](#)
- [Delete a connection (CLI)](#)

Delete a connection (console)

To delete a connection

1. Open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.
2. Choose **Settings > Connections**.
3. In **Connection name**, choose the name of the connection you want to delete.
4. Choose **Delete**.
5. Enter `delete` in the field to confirm, and then choose **Delete**.

    > ⚠️ **Important**
    >
    > This action cannot be undone.

Delete a connection (CLI)

You can use the AWS Command Line Interface (AWS CLI) to delete a connection.

To do this, use the **delete-connection** command.

> ⚠️ **Important**
>
> After you run the command, the connection is deleted. No confirmation dialog box is displayed. You can create a new connection, but the Amazon Resource Name (ARN) is never reused.

To delete a connection

- Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **delete-connection** command, specifying the ARN of the connection that you want to delete.

  ```
  aws codeconnections delete-connection --connection-arn arn:aws:codeconnections:us-
  west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
  ```

  This command returns nothing.

Tag connections resources

A *tag* is a custom attribute label that you or AWS assigns to an AWS resource. Each AWS tag has two parts:

- A *tag key* (for example, `CostCenter`, `Environment`, or `Project`). 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*.

You can use the console or the CLI to tag resources.

You can tag the following resource types in AWS CodeConnections:

- Connections
- Hosts

These steps assume that you have already installed a recent version of the AWS CLI or updated to the current version. For more information, see Installing the AWS CLI in the *AWS Command Line Interface User Guide*.

In addition to identifying, organizing, and tracking your resource with tags, you can use tags in AWS Identity and Access Management (IAM) policies to help control who can view and interact with your resource. For examples of tag-based access policies, see Using tags to control access to AWS CodeConnections resources.

Topics

- [Tag resources (console)](#)
- [Tag resources (CLI)](#)

Tag resources (console)

You can use the console to add, update, or remove tags on a connections resource.

Topics

- [Add tags to a connections resource (console)](#)
- [View tags for a connections resource (console)](#)
- [Edit tags for a connections resource (console)](#)
- [Remove tags from a connections resource (console)](#)

Add tags to a connections resource (console)

You can use the console to add tags to an existing connection or host.

> **ⓘ Note**
>
> When you create a connection for an installed provider such as GitHub Enterprise Server, and a host resource is also created for you, the tags during creation are added to the connection only. This allows you to tag a host separately if you want to reuse it for a new connection. If you want to add tags to the host, use the steps here.

To add tags for a connection

1. Sign in to the console. From the navigation pane, choose **Settings**.

2. Under **Settings**, choose **Connections**. Choose the **Connections** tab.

3. Choose the connection you want to edit. The connection settings page displays.

4. Under **Connection tags**, choose **Edit**. The **Edit Connection tags** page displays.

5. In the **Key** and **Value** fields, enter a key pair for each set of tags you want to add. (The **Value** field is optional.) For example, in **Key**, enter **Project**. In **Value**, enter **ProjectA**.

6. (Optional) Choose **Add tag** to add more rows and enter more tags.

7. Choose **Submit**. The tags are listed under connection settings.

To add tags for a host

1. Sign in to the console. From the navigation pane, choose **Settings**.

2. Under **Settings**, choose **Connections**. Choose the **Hosts** tab.

3. Choose the host you want to edit. The host settings page displays.

4. Under **Host tags**, choose **Edit**. The **Host tags** page displays.

5. In the **Key** and **Value** fields, enter a key pair for each set of tags you want to add. (The **Value** field is optional.) For example, in **Key**, enter `Project`. In **Value**, enter `ProjectA`.

6. (Optional) Choose **Add tag** to add more rows and enter more tags for a host.

7. Choose **Submit**. The tags are listed under host settings.

View tags for a connections resource (console)

You can use the console to view the tags for existing resources.

To view tags for a connection

1. Sign in to the console. From the navigation pane, choose **Settings**.
2. Under **Settings**, choose **Connections**. Choose the **Connections** tab.
3. Choose the connection you want to view. The connection settings page displays.
4. Under **Connection tags**, view the tags for the connection under the **Key** and **Value** columns.

To view tags for a host

1. Sign in to the console. From the navigation pane, choose **Settings**.
2. Under **Settings**, choose **Connections**. Choose the **Hosts** tab.
3. Choose the host you want to view.
4. Under **Host tags**, view the tags for the host under the **Key** and **Value** columns.

Edit tags for a connections resource (console)

You can use the console to edit tags that have been added to connections resources.

To edit tags for a connection

1. Sign in to the console. From the navigation pane, choose **Settings**.
2. Under **Settings**, choose **Connections**. Choose the **Connections** tab.
3. Choose the connection you want to edit. The connection settings page displays.
4. Under **Connection tags**, choose **Edit**. The **Connection tags** page displays.
5. In the **Key** and **Value** fields, update the values in each field as needed. For example, for the `Project` key, in **Value**, change `ProjectA` to `ProjectB`.
6. Choose **Submit**.

To edit tags for a host

1. Sign in to the console. From the navigation pane, choose **Settings**.
2. Under **Settings**, choose **Connections**. Choose the **Hosts** tab.

3. Choose the host you want to edit. The host settings page displays.

4. Under **Host tags**, choose **Edit**. The **Host tags** page displays.

5. In the **Key** and **Value** fields, update the values in each field as needed. For example, for the `Project` key, in **Value**, change `ProjectA` to `ProjectB`.

6. Choose **Submit**.

Remove tags from a connections resource (console)

You can use the console to remove tags from connections resources. When you remove tags from the associated resource, the tags are deleted.

To remove tags for a connection

1. Sign in to the console. From the navigation pane, choose **Settings**.

2. Under **Settings**, choose **Connections**. Choose the **Connections** tab.

3. Choose the connection you want to edit. The connection settings page displays.

4. Under **Connection tags**, choose **Edit**. The **Connection tags** page displays.

5. Next to the key and value for each tag you want to delete, choose **Remove tag**.

6. Choose **Submit**.

To remove tags for a host

1. Sign in to the console. From the navigation pane, choose **Settings**.

2. Under **Settings**, choose **Connections**. Choose the **Hosts** tab.

3. Choose the host you want to edit. The host settings page displays.

4. Under **Host tags**, choose **Edit**. The **Host tags** page displays.

5. Next to the key and value for each tag you want to delete, choose **Remove tag**.

6. Choose **Submit**.

Tag resources (CLI)

You can use the CLI to view, add, update, or remove tags on a connections resource.

Topics

- [Add tags to a connections resource (CLI)](#)

- View tags for a connections resource (CLI)
- Edit tags for a connections resource (CLI)
- Remove tags from a connections resource (CLI)

Add tags to a connections resource (CLI)

You can use the AWS CLI to tag resources in connections.

At the terminal or command line, run the **tag-resource** command, specifying the Amazon Resource Name (ARN) of the resource where you want to add tags and the key and value of the tag you want to add. You can add more than one tag.

To add tags for a connection

1. Get the ARN for your resource. Use the **list-connections** command shown in List connections to get the connection ARN.

2. In a terminal or at the command line, run the **tag-resource** command.

    For example, use the following command to tag a connection with two tags, a tag key named *Project* with the tag value of *ProjectA*, and a tag key named *ReadOnly* with the tag value of *true*.

    ```
    aws codestar-connections tag-resource --resource-arn arn:aws:codestar-
    connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f --
    tags Key=Project,Value=ProjectA Key=IscontainerBased,Value=true
    ```

    If successful, this command returns nothing.

To add tags for a host

1. Get the ARN for your resource. Use the **list-hosts** command shown in List hosts to get the host ARN.

2. In a terminal or at the command line, run the **tag-resource** command.

    For example, use the following command to tag a host with two tags, a tag key named *Project* with the tag value of *ProjectA*, and a tag key named *IscontainerBased* with the tag value of *true*.

```
aws codestar-connections tag-resource --resource-arn arn:aws:codestar-
connections:us-west-2:account_id:host/My-Host-28aef605 --tags
 Key=Project,Value=ProjectA Key=IscontainerBased,Value=true
```

If successful, this command returns nothing.

View tags for a connections resource (CLI)

You can use the AWS CLI to view the AWS tags for a connections resource. If no tags have been
added, the returned list is empty. Use the **list-tags-for-resource** command to view tags that have
been added to a connection or a host.

To view tags for a connection

1. Get the ARN for your resource. Use the **list-connections** command shown in [List connections](#)
   to get the connection ARN.

2. In a terminal or at the command line, run the **list-tags-for-resource** command. For example,
   use the following command to view a list of tag keys and tag values for a connection.

```
aws codestar-connections list-tags-for-resource --resource-arn arn:aws:codestar-
connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

This command returns the tags associated with the resource. This example shows two key-
value pairs returned for a connection.

```
{
    "Tags": [
        {
            "Key": "Project",
            "Value": "ProjectA"
        },
        {
            "Key": "ReadOnly",
            "Value": "true"
        }
    ]
}
```

To view tags for a host

1. Get the ARN for your resource. Use the **list-hosts** command shown in [List hosts](#) to get the host ARN.

2. In a terminal or at the command line, run the **list-tags-for-resource** command. For example, use the following command to view a list of tag keys and tag values for a host.

```
aws codestar-connections list-tags-for-resource --resource-arn arn:aws:codestar-
connections:us-west-2:account_id:host/My-Host-28aef605
```

This command returns the tags associated with the resource. This example shows two key-value pairs returned for a host.

```
{
    "Tags": [
        {
            "Key": "IscontainerBased",
            "Value": "true"
        },
        {
            "Key": "Project",
            "Value": "ProjectA"
        }
    ]
}
```

Edit tags for a connections resource (CLI)

You can use the AWS CLI to edit a tag for a resource. You can change the value for an existing key or add another key.

At the terminal or command line, run the **tag-resource** command, specifying the ARN of the resource where you want to update a tag and specify the tag key and tag value to update.

When you edit tags, any tag keys not specified will be retained, while anything with the same key but a new value will be updated. New keys that are added with the edit command are added as a new key-value pair.

To edit tags for a connection

1. Get the ARN for your resource. Use the **list-connections** command shown in [List connections](#) to get the connection ARN.

2. In a terminal or at the command line, run the **tag-resource** command.

    In this example, the value for the key `Project` is changed to `ProjectB`.

    ```
    aws codestar-connections tag-resource --resource-arn arn:aws:codestar-
    connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f --
    tags Key=Project,Value=ProjectB
    ```

    If successful, this command returns nothing. To verify the tags associated with the connection, run the **list-tags-for-resource** command.

To edit tags for a host

1. Get the ARN for your resource. Use the **list-hosts** command shown in [List hosts](#) to get the host ARN.

2. In a terminal or at the command line, run the **tag-resource** command.

    In this example, the value for the key `Project` is changed to `ProjectB`.

    ```
    aws codestar-connections tag-resource --resource-arn arn:aws:codestar-
    connections:us-west-2:account_id:host/My-Host-28aef605 --tags
     Key=Project,Value=ProjectB
    ```

    If successful, this command returns nothing. To verify the tags associated with the host, run the **list-tags-for-resource** command.

Remove tags from a connections resource (CLI)

Follow these steps to use the AWS CLI to remove a tag from a resource. When you remove tags from the associated resource, the tags are deleted.

> **ⓘ Note**
>
> If you delete a connection resource, all tag associations are removed from the deleted resource. You do not have to remove tags before you delete a connection resource.

At the terminal or command line, run the **untag-resource** command, specifying the ARN of the resource where you want to remove tags and the tag key of the tag you want to remove. For example, to remove multiple tags on a connection with the tag keys *Project* and *ReadOnly*, use the following command.

```
aws codestar-connections untag-resource --resource-arn arn:aws:codestar-connections:us-
west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f --tag-
keys Project ReadOnly
```

If successful, this command returns nothing. To verify the tags associated with the resource, run the **list-tags-for-resource** command. The output shows that all tags have been removed.

```
{
    "Tags": []
}
```

View connection details

You can use the Developer Tools console or the **get-connection** command in the AWS Command Line Interface (AWS CLI) to view details for a connection. To use the AWS CLI, you must have already installed a recent version of the AWS CLI or updated to the current version. For more information, see Installing the AWS CLI in the *AWS Command Line Interface User Guide*.

To view a connection (console)

1. Open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Choose **Settings > Connections**.

3. Choose the button next to the connection you want to view, and then choose **View details**.

4. The following information appears for your connection:

   - The connection name.

- The provider type for your connection.

- The connection status.

- The connection ARN.

- If the connection was created for an installed provider, such as GitHub Enterprise Server, the host information associated with the connection.

- If the connection was created for an installed provider, such as GitHub Enterprise Server, the endpoint information associated with the host for the connection.

5. If the connection is in **Pending** status, to complete the connection, choose **Update pending connection**. For more information , see [Update a pending connection](#).

To view a connection (CLI)

- At the terminal or command line, run the **get-connection** command. For example, use the following command to view details for a connection with the `arn:aws:codestar-connections:us-west-2:`*`account_id`*`:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f` ARN value.

```
aws codeconnections get-connection --connection-arn arn:aws:codeconnections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

If successful, this command returns the connections details.

Example output for a Bitbucket connection:

```
{
    "Connection": {
        "ConnectionName": "MyConnection",
        "ConnectionArn": "arn:aws:codeconnections:us-west-2:account_id:connection/
cdacd948-EXAMPLE",
        "ProviderType": "Bitbucket",
        "OwnerAccountId": "account_id",
        "ConnectionStatus": "AVAILABLE"
    }
}
```

Example output for a GitHub connection:

```
{
    "Connection": {
        "ConnectionName": "MyGitHubConnection",
        "ConnectionArn": "arn:aws:codeconnections:us-west-2:account_id:connection/
ebcd4a13-EXAMPLE",
        "ProviderType": "GitHub",
        "OwnerAccountId": "account_id",
        "ConnectionStatus": "AVAILABLE"
    }
}
```

Example output for a GitHub Enterprise Server connection:

```
{
    "Connection": {
        "ConnectionName": "MyConnection",
        "ConnectionArn": "arn:aws:codeconnections:us-
west-2:account_id:connection/2d178fb9-EXAMPLE",
        "ProviderType": "GitHubEnterpriseServer",
        "OwnerAccountId": "account_id",
        "ConnectionStatus": "PENDING",
        "HostArn": "arn:aws:ccodeconnections:us-west-2:account_id:host/sdfsdf-
EXAMPLE"
    }
}
```

Share connections with AWS accounts

You can use resource sharing with AWS RAM to share an existing connection with another AWS account or with accounts in your organization. You can use your shared connection with resources in AWS that you manage for third-party source connections, such as in CodePipeline.

> ⚠️ **Important**
>
> Connection sharing is not supported for `codestar-connections` resources. This is only supported for `codeconnections` resources.

Before you begin:

- You must have already created a connection with your AWS account.

- You must have resource sharing enabled.

- You must have the required permissions configured. For more information, see Supported permissions for connection sharing.

> ⓘ **Note**
>
> To share the connection, you must be the organization owner or the repository owner if not under an organization. The account that you are sharing with will also need permissions to the repository.

Topics

- Share a connection (console)

- Share a connection (CLI)

- View shared connections (console)

- View shared connections (CLI)

Share a connection (console)

You can use the console to create shared connection resources.

1. Sign in to the AWS Management Console.

   Choose **Create resource share** on the **Shared by me : Shared resources** page in the AWS RAM console.

2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To create resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia),

   For more information about sharing global resources, see Sharing Regional resources compared to global resources.

3. On the creation page, in **Name**, enter a name for your resource share. Under **Resources**, choose **Code Connections**.

4.   Choose your connection resource and assign the principals with whom you want to share.

5.   Choose **Create**.

Share a connection (CLI)

You can use the AWS Command Line Interface (AWS CLI) to share an existing connection with other accounts and view connections that you own or have had shared with you.

To do this, use the **create-resource-share** and `accept-resource-share-invitation` commands for AWS RAM.

To share a connection

1.   Sign in with the account that will share the connection.

2.   Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **create-resource-share** command, specifying the `--name`, `--resource-arns`, and `--principals` for your connection share. In this example, the name is `my-shared-resource` and the specified connection name is `MyConnection` in the resource ARN. In `principals`, provide the destination account or accounts that you are sharing with.

```
aws ram create-resource-share --name my-shared-resource --resource-
arns connection_ARN --principals destination_account
```

If successful, this command returns the connection ARN information similar to the following.

```
{
    "resourceShare": {
        "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-
share/4476c27d-8feb-4b21-afe9-7de23EXAMPLE",
        "name": "MyNewResourceShare",
        "owningAccountId": "111111111111",
        "allowExternalPrincipals": true,
        "status": "ACTIVE",
        "creationTime": 1634586271.302,
        "lastUpdatedTime": 1634586271.302
    }
}
```

3. Requests to share can be accepted as detailed in the next procedure.

To authenticate and accept the connection share with the destination account

The following procedure is optional for destination accounts that belong to the same organization and have resource sharing enabled in Organizations.

1. Sign in with the destination account that will receive the invitation.

2. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **get-resource-share-invitations** command.

```
aws ram get-resource-share-invitations
```

Capture the resource share invitation ARN for the next step.

3. Run the **accept-resource-share-invitation** command, specifying the `--resource-share-invitation-arn`.

```
aws ram accept-resource-share-invitation --resource-share-invitation-
arn invitation_ARN
```

If successful, this command returns the following output.

```
{
    "resourceShareInvitation": {
```

```
        "resourceShareInvitationArn": "arn:aws:ram:us-west-2:111111111111:resource-
share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE",
        "resourceShareName": "MyResourceShare",
        "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE",
        "senderAccountId": "111111111111",
        "receiverAccountId": "222222222222",
        "invitationTimestamp": "2021-09-22T15:07:35.620000-07:00",
        "status": "ACCEPTED"
    }
}
```

View shared connections (console)

You can use the console to view shared connection resources.

1. Sign in to the AWS Management Console.

    Open the **Shared by me : Shared resources** page in the AWS RAM console.

2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia),

    For more information about sharing global resources, see Sharing Regional resources compared to global resources.

3. For each shared resource, the following information is available:

    - **Resource ID** – The ID of the resource. Choose the ID of a resource to open a new browser tab to view the resource in its native service console.

    - **Resource type** – The type of resource.

    - **Last share date** – The date on which the resource was last shared.

    - **Resource shares** – The number of resource shares that include the resource. To see the list of the resource shares, choose the number.

    - **Principals** – The number of principals who can access the resource. Choose the value to view the principals.

View shared connections (CLI)

You can use the AWS CLI to view connections that you own or have had shared with you.

To do this, use the **get-resource-shares** command.

To view shared connections

- Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **get-resource-shares** command.

  ```
  aws ram get-resource-shares
  ```

  The output returns a list of resource shares for your account.

Working with hosts

To create a connection to an installed provider type, such as GitHub Enterprise Server, you first create a host using the AWS Management Console. A host is a resource that you create to represent the infrastructure where your provider is installed. Then you create a connection using that host. For more information, see Working with connections.

For example, you create a host for your connection so that the third-party app for your provider can be registered to represent your infrastructure. You create one host for a provider type, and then all of your connections to that provider type use that host.

When you use the console to create a connection to an installed provider type, such as GitHub Enterprise Server, the console creates your host resource for you.

Topics

- Create a host
- Set up a pending host
- List hosts
- Edit a host
- Delete a host
- View host details

Create a host

You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI) to create a connection to a third-party code repository that is installed on your infrastructure. For example, you might have GitHub Enterprise Server running as a virtual machine on an Amazon EC2 instance. Before you create a connection to GitHub Enterprise Server, you create a host to use for the connection.

For an overview of the host creation workflow for installed providers, see Workflow to create or update a host.

Before you begin:

- (Optional) If you want to create your host with a VPC, you must have already created a network or virtual private cloud (VPC).

- You must have already created your instance and, if you plan to connect with your VPC, launched your host into your VPC.

> **ⓘ Note**
>
> Each VPC can only be associated with one host at a time.

You can optionally configure your host with a VPC. For more information about network and VPC configuration for your host resource, see the VPC prerequisites in (Optional) Prerequisites: Network or Amazon VPC configuration for your connection and Troubleshooting VPC configuration for your host.

To use the console to create a host and a connection to GitHub Enterprise Server, see Create your GitHub Enterprise Server connection (console). The console creates your host for you.

To use the console to create a host and a connection to GitLab self-managed, see Create a connection to GitLab self-managed. The console creates your host for you.

(Optional) Prerequisites: Network or Amazon VPC configuration for your connection

If your infrastructure is configured with a network connection, you can skip this section.

If your host is only accessible in a VPC, follow these VPC requirements before you continue.

VPC requirements

You can optionally choose to create your host with a VPC. The following are general VPC requirements, depending on the VPC you have set up for your installation.

- You can configure a *public* VPC with public and private subnets. You can use the default VPC for your AWS account if you do not have preferred CIDR blocks or subnets.

- If you have a *private* VPC configured, and you have configured your GitHub Enterprise Server instance to perform TLS validation using a non-public certificate authority, you need to provide the TLS certificate for your host resource.

- When connections creates your host, the VPC endpoint (PrivateLink) for webhooks is created for you. For more information, see AWS CodeConnections and interface VPC endpoints (AWS PrivateLink).

- Security group configuration:

  - The security groups used during host creation need inbound and outbound rules that allow the network interface to connect to your GitHub Enterprise Server instance

  - The security groups attached to your GitHub Enterprise Server instance (not part of the host setup) need inbound and outbound access from the network interfaces created by connections.

- Your VPC subnets must reside in different Availability Zones in your Region. Availability Zones are distinct locations that are isolated from failures in other Availability Zones. Each subnet must reside entirely within one Availability Zone and cannot span zones.

For more information about working with VPCs and subnets, see VPC and Subnet Sizing for IPv4 in the *Amazon VPC User Guide*.

VPC information you provide for host setup

When you create your host resource for your connections in the next step, you need to provide the following:

- **VPC ID:** The ID of the VPC for the server where your GitHub Enterprise Server instance is installed or a VPC which has access to your installed GitHub Enterprise Server instance through VPN or Direct Connect.

- **Subnet ID or IDs:** The ID of the subnet for the server where your GitHub Enterprise Server instance is installed or a subnet with access to your installed GitHub Enterprise Server instance through VPN or Direct Connect.

- **Security group or groups:** The security group for the server where your GitHub Enterprise Server instance is installed or a security group with access to your installed GitHub Enterprise Server instance through VPN or Direct Connect.

- **Endpoint:** Have your server endpoint ready and continue to the next step.

For more information, including troubleshooting VPC or host connections, see Troubleshooting VPC configuration for your host.

Permission requirements

As part of the host creation process, AWS CodeConnections creates network resources on your behalf to facilitate the VPC connectivity. This includes a network interface for AWS CodeConnections to query data from your host, and a VPC endpoint or *PrivateLink* for the host to send event data via webhooks to connections. To be able to create these network resources, make sure that the role used for creating the host has the following permissions:

```
ec2:CreateNetworkInterface
ec2:CreateTags
ec2:DescribeDhcpOptions
ec2:DescribeNetworkInterfaces
ec2:DescribeSubnets
ec2:DeleteNetworkInterface
ec2:DescribeVpcs
ec2:CreateVpcEndpoint
ec2:DeleteVpcEndpoints
ec2:DescribeVpcEndpoints
```

For more information about troubleshooting permissions or host connections in a VPC, see Troubleshooting VPC configuration for your host.

For more information about the webhook VPC endpoint, see AWS CodeConnections and interface VPC endpoints (AWS PrivateLink).

Topics

- Create a host for a connection (console)

- Create a host for a connection (CLI)

Create a host for a connection (console)

For connections for installations, such as with GitHub Enterprise Server or with GitLab self-managed, you use a host to represent the endpoint for the infrastructure where your third-party provider is installed.

> ⓘ **Note**
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

To learn about considerations for setting up a host in a VPC, see Create a connection to GitLab self-managed.

To use the console to create a host and a connection to GitHub Enterprise Server, see Create your GitHub Enterprise Server connection (console). The console creates your host for you.

To use the console to create a host and a connection to GitLab self-managed, see Create a connection to GitLab self-managed. The console creates your host for you.

> ⓘ **Note**
>
> You only create a host once per GitHub Enterprise Server or GitLab self-managed account. All of your connections to a specific GitHub Enterprise Server or GitLab self-managed account will use the same host.

Create a host for a connection (CLI)

You can use the AWS Command Line Interface (AWS CLI) to create a host for installed connections.

> ⓘ **Note**
>
> You only create a host once per GitHub Enterprise Server account. All of your connections to a specific GitHub Enterprise Server account will use the same host.

You use a host to represent the endpoint for the infrastructure where your third-party provider is installed. To create a host with the CLI, you use the **create-host** command. After you finish creating

the host, the host is in **Pending** status. You then *set up* the host to move it to an **Available** status. After the host is available, you complete the steps to create a connection.

> **⚠ Important**
>
> A host created through the AWS CLI is in `Pending` status by default. After you create a host with the CLI, use the console to set up the host to make its status `Available`.

To use the console to create a host and a connection to GitHub Enterprise Server, see Create your GitHub Enterprise Server connection (console). The console creates your host for you.

To use the console to create a host and a connection to GitLab self-managed, see Create a connection to GitLab self-managed. The console creates your host for you.

Set up a pending host

A host created through the AWS Command Line Interface (AWS CLI) or SDK is in `Pending` status by default. After you create a connection with the console, AWS CLI, or the SDK, use the console to set up the host to make its status `Available`.

You must have already created a host. For more information, see Create a host.

To set up a pending host

After your host is created, it is in a **Pending** status. To move the host from **Pending** to **Available**, complete these steps. This process performs a handshake with the third-party provider to register the AWS connection app on the host.

1. After your host reaches **Pending** status on the AWS Developer Tools console, choose **Set up host**.

2. If you are creating a host for GitLab self-managed, a **Set up** page displays. In **Provide personal access token**, provide your GitLab PAT with the followed scoped-down permission only: api.

3. On the third-party installed provider login page, such as the **GitHub Enterprise Server** login page, log in with your account credentials if prompted.

4. On the app install page, in **GitHub App name**, enter a name for the app you want to install for your host. Choose **Create GitHub App**.

5. After your host is successfully registered, the host details page appears and shows that the host status is **Available**.
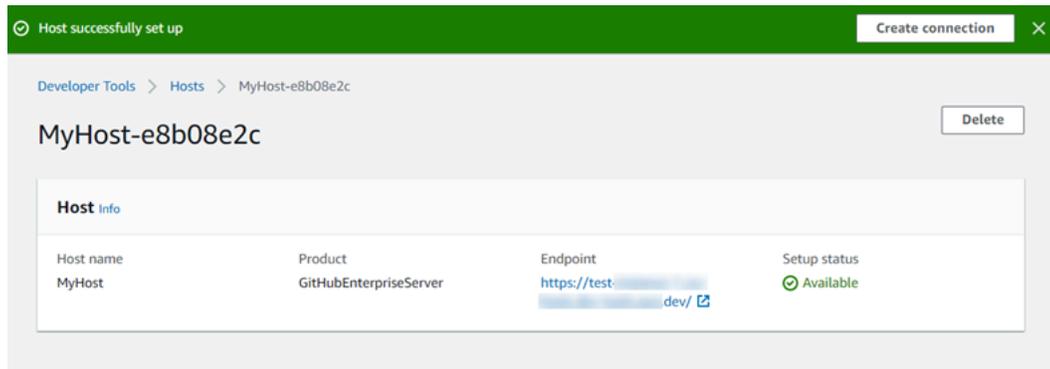
6. You can continue with creating your connection after the host is available. On the success
    banner, choose **Create connection**. Complete the steps in Create a connection.

List hosts

You can use the Developer Tools console or the **list-connections** command in the AWS Command
Line Interface (AWS CLI) to view a list of connections in your account.

List hosts (console)

To list hosts

1. Open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/
    connections.

2. Choose the **Hosts** tab. View the name, status, and ARN for your hosts.

List hosts (CLI)

You can use the AWS CLI to list your hosts for installed third-party provider connections.

To do this, use the **list-hosts** command.

To list hosts

*   Open a terminal (Linux, macOS, or Unix) or command prompt (Windows), and use the AWS CLI
    to run the **list-hosts** command.

    ```
    aws codeconnections list-hosts
    ```

    This command returns the following output.

```
{
    "Hosts": [
        {
            "Name": "My-Host",
            "HostArn": "arn:aws:codeconnections:us-west-2:account_id:host/My-
Host-28aef605",
            "ProviderType": "GitHubEnterpriseServer",
            "ProviderEndpoint": "https://my-instance.test.dev",
            "Status": "AVAILABLE"
        }
    ]
}
```

Edit a host

You can edit host settings for a host in Pending status. You can edit the host name, URL, or VPC configuration.

You cannot use the same URL for more than one host.

> ⓘ **Note**
>
> To learn about considerations for setting up a host in a VPC, see (Optional) Prerequisites: Network or Amazon VPC configuration for your connection.

To edit a host

1. Open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.
2. Choose **Settings > Connections**.
3. Choose the **Hosts** tab.

   The hosts associated with your AWS account and created in the selected AWS Region are displayed.
4. To edit the host name, enter a new value in **Name**.
5. To edit the host endpoint, enter a new value in **URL**.
6. To edit the host VPC configuration, enter new values in **VPC ID**.

7.　Choose **Edit host**.

8.　The updated settings are displayed. Choose **Set up Pending host**.

Delete a host

You can use the Developer Tools console or the **delete-host** command in the AWS Command Line Interface (AWS CLI) to delete a host.

Topics

- Delete a host (console)

- Delete a host (CLI)

Delete a host (console)

To delete a host

1.　Open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2.　Choose the **Hosts** tab. In **Name**, choose the name of the host you want to delete.

3.　Choose **Delete**.

4.　Enter `delete` in the field to confirm, and then choose **Delete**.

> ⚠ **Important**
>
> 　This action cannot be undone.

Delete a host (CLI)

You can use the AWS Command Line Interface (AWS CLI) to delete a host.

To do this, use the **delete-host** command.

> ⚠ **Important**
>
> 　Before you can delete a host, you must delete all connections associated with the host.

After you run the command, the host is deleted. No confirmation dialog box is displayed.

To delete a host

- Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **delete-host** command, specifying the Amazon Resource Name (ARN) of the host that you want to delete.

```
aws codeconnections delete-host --host-arn "arn:aws:codeconnections:us-west-2:account_id:host/My-Host-28aef605"
```

This command returns nothing.

View host details

You can use the Developer Tools console or the **get-host** command in the AWS Command Line Interface (AWS CLI) to view details for a host.

To view host details (console)

1. Sign in to the AWS Management Console and open the Developer Tools console at https://console.aws.amazon.com/codesuite/settings/connections.

2. Choose **Settings > Connections**, and then choose the **Hosts** tab.

3. Choose the button next to the host you want to view, and then choose **View details**.

4. The following information appears for your host:

   - The host name.

   - The provider type for your connection.

   - The endpoint of the infrastructure where your provider is installed.

   - The setup status for your host. A host ready for a connection is in **Available** status. If your host was created but setup was not completed, the host might be in a different status.

     The following statuses are available:

     - PENDING - The host has completed creation and is ready to start the setup by registering the provider app on the host.

- AVAILABLE - The host has completed creation and setup and is available for use with connections.

- ERROR - There was an error during host creation or registration.

- VPC_CONFIG_VPC_INITIALIZING - The VPC configuration for the host is being created.

- VPC_CONFIG_VPC_FAILED_INITIALIZATION - The VPC configuration for the host encountered and error and failed.

- VPC_CONFIG_VPC_AVAILABLE - The VPC configuration for the host has completed setup and is available.

- VPC_CONFIG_VPC_DELETING - The VPC configuration for the host is being deleted.
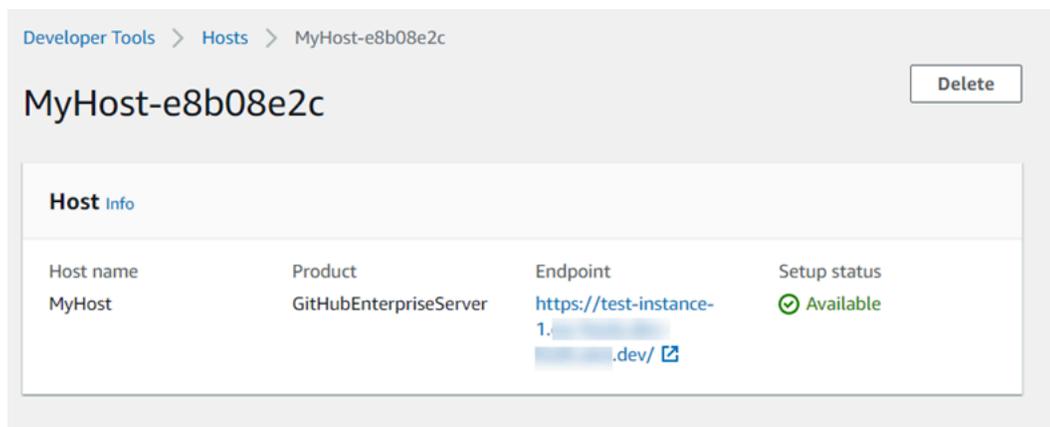
5. To delete the host, choose **Delete**.

6. If the host is in **Pending** status, to complete the setup, choose **Set up host**. For more information , see [Set up a pending host](#).

To view host details (CLI)

- Open a terminal (Linux, macOS, or Unix) or command prompt (Windows), and use the AWS CLI to run the **get-host** command, specifying the Amazon Resource Name (ARN) of the host that you want to view details for.

```
aws codeconnections get-host --host-arn arn:aws:codeconnections:us-
west-2:account_id:host/My-Host-28aef605
```

This command returns the following output.

```
{
```

```
      "Name": "MyHost",
      "Status": "AVAILABLE",
      "ProviderType": "GitHubEnterpriseServer",
      "ProviderEndpoint": "https://test-instance-1.dev/"
}
```

Working with sync configurations for linked repositories

In AWS CodeConnections, you use a connection to associate AWS resources to a third-party repository, such as GitHub, Bitbucket Cloud, GitHub Enterprise Server, and GitLab. Using the CFN_STACK_SYNC sync type, you can create a sync configuration, which allows AWS to sync content from a Git repository to update a specified AWS resource. CloudFormation integrates with connections so that you can use Git sync to manage your template and parameter files in a linked repository that you sync with.

After creating a connection, you can use the connections CLI or the CloudFormation console to create your repository link and sync configuration.

- Repository link: A repository link creates an association between your connection and an external Git repository. The repository link allows Git sync to monitor and sync changes to files in a specified Git repository.

- Sync configuration: Use the sync configuration to sync content from a Git repository to update a specified AWS resource.

For more information, see the *AWS CodeConnections API Reference*.

For a tutorial that walks you through creating a sync configuration for an CloudFormation stack using the CloudFormation console, see Working with CloudFormation Git sync in the *CloudFormation User Guide*.

Topics

- Working with repository links
- Working with sync configurations

Working with repository links

A repository link creates an association between your connection and an external Git repository. The repository link allows Git sync to monitor and sync changes to files in a specified Git repository to an CloudFormation stack.

For more information about repository links, see the [AWS CodeConnections API reference](#).

Topics

- [Create a repository link](#)
- [Update a repository link](#)
- [List repository links](#)
- [Delete a repository link](#)
- [View repository link details](#)

Create a repository link

You can use the **create-repository-link** command in the AWS Command Line Interface (AWS CLI) to create a link between your connection and the external repository to sync to.

Before you can create a repository link, you must have already created your external repository with your third-party provider, such as GitHub.

To create a repository link

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **create-repository-link** command. Specify the ARN of the associated connection, the owner ID, and the repository name.

   ```
   aws codeconnections create-repository-link --connection-arn
    arn:aws:codeconnections:us-east-1:account_id:connection/001f5be2-a661-46a4-
   b96b-4d277cac8b6e --owner-id account_id --repository-name MyRepo
   ```

2. This command returns the following output.

   ```
   {
       "RepositoryLinkInfo": {
           "ConnectionArn": "arn:aws:codeconnections:us-east-1:account_id:connection/
   aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
   ```

```
            "OwnerId": "account_id",
            "ProviderType": "GitHub",
            "RepositoryLinkArn": "arn:aws:codeconnections:us-
    east-1:account_id:repository-link/be8f2017-b016-4a77-87b4-608054f70e77",
            "RepositoryLinkId": "be8f2017-b016-4a77-87b4-608054f70e77",
            "RepositoryName": "MyRepo",
            "Tags": []
        }
    }
```

Update a repository link

You can use the **update-repository-link** command in the AWS Command Line Interface (AWS CLI) to update a specified repository link.

You can update the following information for your repository link:

- `--connection-arn`

- `--owner-id`

- `--repository-name`

You might update a repository link when you want to change the connection that is associated with your repository. To use a different connection, you need to specify the connection ARN. For the steps to view your connection ARN, see [View connection details](#).

To update a repository link

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **update-repository-link** command, specifying the value to update for the repository link. For example, the following command updates the connection associated to the repository link ID. It specifies the new connection ARN with the `--connection` parameter.

   ```
   aws codestar-connections update-repository-link --repository-link-id
    6053346f-8a33-4edb-9397-10394b695173 --connection-arn arn:aws:codestar-
   connections:us-east-1:account_id:connection/aEXAMPLE-f055-4843-adef-4ceaefcb2167
   ```

2. This command returns the following output.

   ```
   {
   ```

```
    "RepositoryLinkInfo": {
        "ConnectionArn": "arn:aws:codestar-connections:us-
east-1:account_id:connection/aEXAMPLE-f055-4843-adef-4ceaefcb2167",
        "OwnerId": "owner_id",
        "ProviderType": "GitHub",
        "RepositoryLinkArn": "arn:aws:codestar-connections:us-
east-1:account_id:repository-link/6053346f-8a33-4edb-9397-10394b695173",
        "RepositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
        "RepositoryName": "MyRepo",
        "Tags": []
    }
}
```

List repository links

You can use the **list-repository-links** command in the AWS Command Line Interface (AWS CLI) to list repository links for your account.

To list repository links

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **list-repository-links** command.

    ```
    aws codeconnections list-repository-links
    ```

2. This command returns the following output.

    ```
    {
        "RepositoryLinks": [
            {
                "ConnectionArn": "arn:aws:codestar-connections:us-
    east-1:account_id:connection/001f5be2-a661-46a4-b96b-4d277cac8b6e",
                "OwnerId": "owner_id",
                "ProviderType": "GitHub",
                "RepositoryLinkArn": "arn:aws:codestar-connections:us-
    east-1:account_id:repository-link/6053346f-8a33-4edb-9397-10394b695173",
                "RepositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
                "RepositoryName": "MyRepo",
                "Tags": []
            }
        ]
    ```

```
}
```

Delete a repository link

You can use the **delete-repository-link** command in the AWS Command Line Interface (AWS CLI) to delete a repository link.

Before you can delete a repository link, you must delete all sync configurations associated with the repository link.

> ⚠️ **Important**
>
> After you run the command, the repository link is deleted. No confirmation dialog box is displayed. You can create a new repository link, but the Amazon Resource Name (ARN) is not reused.

To delete a repository link

- Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **delete-repository-link** command, specifying the ID of the repository link to delete.

  ```
  aws codeconnections delete-repository-link --repository-link-id
    6053346f-8a33-4edb-9397-10394b695173
  ```

  This command returns nothing.

View repository link details

You can use the **get-repository-link** command in the AWS Command Line Interface (AWS CLI) to view details about a repository link.

To view repository link details

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **get-repository-link** command, specifying the repository link ID.

   ```
   aws codestar-connections get-repository-link --repository-link-id
     6053346f-8a33-4edb-9397-10394b695173
   ```

2. This command returns the following output.

```
{
    "RepositoryLinkInfo": {
        "ConnectionArn": "arn:aws:codestar-connections:us-
east-1:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
        "OwnerId": "owner_id",
        "ProviderType": "GitHub",
        "RepositoryLinkArn": "arn:aws:codestar-connections:us-
east-1:account_id:repository-link/be8f2017-b016-4a77-87b4-608054f70e77",
        "RepositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
        "RepositoryName": "MyRepo",
        "Tags": []
    }
}
```

Working with sync configurations

A sync configuration creates an association between a specified repository and connection. Use the sync configuration to sync content from a Git repository to update a specified AWS resource.

For more information about connections, see the [AWS CodeConnections API reference](#).

Topics

- [Create a sync configuration](#)
- [Update a sync configuration](#)
- [List sync configurations](#)
- [Delete a sync configuration](#)
- [View sync configuration details](#)

Create a sync configuration

You can use the **create-repository-link** command in the AWS Command Line Interface (AWS CLI) to create a link between your connection and the external repository to sync to.

Before you can create a sync configuration, you must have already created a repository link between your connection and your third-party repository.

To create a sync configuration

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI
    to run the **create-repository-link** command. Specify the ARN of the associated connection,
    the owner ID, and the repository name. The following command creates a sync configuration
    with a sync type for a resource in CloudFormation. It also specifies the repository branch and
    configuration file in the repository. In this example, the resource is a stack named **mystack**.

    ```
    aws codeconnections create-sync-configuration --branch main --config-file filename
      --repository-link-id be8f2017-b016-4a77-87b4-608054f70e77 --resource-name mystack
      --role-arn arn:aws:iam::account_id:role/myrole --sync-type CFN_STACK_SYNC
    ```

2. This command returns the following output.

    ```
    {
        "SyncConfiguration": {
            "Branch": "main",
            "ConfigFile": "filename",
            "OwnerId": "account_id",
            "ProviderType": "GitHub",
            "RepositoryLinkId": "be8f2017-b016-4a77-87b4-608054f70e77",
            "RepositoryName": "MyRepo",
            "ResourceName": "mystack",
            "RoleArn": "arn:aws:iam::account_id:role/myrole",
            "SyncType": "CFN_STACK_SYNC"
        }
    }
    ```

Update a sync configuration

You can use the **update-sync-configuration** command in the AWS Command Line Interface (AWS
CLI) to update a specified sync configuration.

You can update the following information for your sync configuration:

*   --branch
*   --config-file
*   --repository-link-id
*   --resource-name
*   --role-arn

To update a sync configuration

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **update-sync-configuration** command, specifying the value you want to update, along with the resource name and sync type. For example, the following command updates the branch name associated to the sync configuration with the `--branch` parameter.

```
aws codeconnections update-sync-configuration --sync-type CFN_STACK_SYNC --
resource-name mystack --branch feature-branch
```

2. This command returns the following output.

```
{
    "SyncConfiguration": {
        "Branch": "feature-branch",
        "ConfigFile": "filename.yaml",
        "OwnerId": "owner_id",
        "ProviderType": "GitHub",
        "RepositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
        "RepositoryName": "MyRepo",
        "ResourceName": "mystack",
        "RoleArn": "arn:aws:iam::account_id:role/myrole",
        "SyncType": "CFN_STACK_SYNC"
    }
```

List sync configurations

You can use the **list-sync-configurations** command in the AWS Command Line Interface (AWS CLI) to list repository links for your account.

To list repository links

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **list-sync-configurations** command, specifying the sync type and repository link ID.

```
aws codeconnections list-sync-configurations --repository-link-id
  6053346f-8a33-4edb-9397-10394b695173  --sync-type CFN_STACK_SYNC
```

2. This command returns the following output.

```
{
```

```
    "SyncConfigurations": [
        {
            "Branch": "main",
            "ConfigFile": "filename.yaml",
            "OwnerId": "owner_id",
            "ProviderType": "GitHub",
            "RepositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
            "RepositoryName": "MyRepo",
            "ResourceName": "mystack",
            "RoleArn": "arn:aws:iam::account_id:role/myrole",
            "SyncType": "CFN_STACK_SYNC"
        }
    ]
}
```

Delete a sync configuration

You can use the **delete-sync-configuration** command in the AWS Command Line Interface (AWS CLI) to delete a sync configuration.

> ⚠️ **Important**
>
> After you run the command, the sync configuration is deleted. No confirmation dialog box is displayed. You can create a new sync configuration, but the Amazon Resource Name (ARN) is not reused.

To delete a sync configuration

*   Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **delete-sync-configuration** command, specifying the sync type and resource name for the sync configuration that you want to delete.

    ```
    aws codeconnections delete-sync-configuration --sync-type CFN_STACK_SYNC --
    resource-name mystack
    ```

    This command returns nothing.

View sync configuration details

You can use the **get-sync-configuration** command in the AWS Command Line Interface (AWS CLI) to view details for a sync configuration.

To view details for a sync configuration

1. Open a terminal (Linux, macOS, or Unix) or command prompt (Windows). Use the AWS CLI to run the **get-sync-configuration** command, specifying the repository link ID.

   ```
   aws codeconnections get-sync-configuration --sync-type CFN_STACK_SYNC --resource-
   name mystack
   ```

2. This command returns the following output.

   ```
   {
       "SyncConfiguration": {
           "Branch": "main",
           "ConfigFile": "filename",
           "OwnerId": "owner_id",
           "ProviderType": "GitHub",
           "RepositoryLinkId": "be8f2017-b016-4a77-87b4-608054f70e77",
           "RepositoryName": "MyRepo",
           "ResourceName": "mystack",
           "RoleArn": "arn:aws:iam::account_id:role/myrole",
           "SyncType": "CFN_STACK_SYNC"
       }
   }
   ```

Logging AWS CodeConnections API calls with AWS CloudTrail

AWS CodeConnections is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all API calls for notifications as events. The calls captured include calls from the Developer Tools console and code calls to the AWS CodeConnections API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon Simple Storage Service (Amazon S3) bucket, including events for notifications. 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 AWS

CodeConnections, the IP address from which the request was made, who made the request, when it was made, and other details.

For more information, see the [AWS CloudTrail User Guide](#).

AWS CodeConnections information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS CodeConnections, 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](#) in the *AWS CloudTrail User Guide*.

For an ongoing record of events in your AWS account, including events for AWS CodeConnections, 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 topics in the *AWS CloudTrail User Guide*:

- [Overview for creating a trail](#)
- [CloudTrail supported services and integrations](#)
- [Configuring Amazon SNS notifications for CloudTrail](#)
- [Receiving CloudTrail log files from multiple regions](#)
- [Receiving CloudTrail log files from multiple accounts](#)

All AWS CodeConnections actions are logged by CloudTrail and are documented in the [AWS CodeConnections API reference](#). For example, calls to the `CreateConnection`, `DeleteConnection` and `GetConnection` actions generate entries in the CloudTrail log files.

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 other IAM 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 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.

CreateConnection example

The following example shows a CloudTrail log entry that demonstrates the `CreateConnection` action.

```
{
    "EventId": "b4374fde-c544-4d43-b511-7d899568e55a",
    "EventName": "CreateConnection",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-09T15:13:46-08:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:user/Mary_Major",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-09T23:03:08Z",
```

```
                            "mfaAuthenticated": "false"
                    }
                }
        },
        "eventTime": "2024-01-09T23:13:46Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "CreateConnection",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.13.30 Python/3.11.6 Darwin/23.2.0 exe/x86_64 prompt/off
  command/codeconnections.create-connection",
        "requestParameters": {
            "providerType": "GitHub",
            "connectionName": "my-connection"
        },
        "responseElements": {
            "connectionArn": "arn:aws:codeconnections:us-
  east-1:123456789012:connection/df03df74-8e05-45cf-b420-b39e389dd264"
        },
        "requestID": "57640a88-97b7-481d-9665-cfd79a681379",
        "eventID": "b4374fde-c544-4d43-b511-7d899568e55a",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

CreateHost example

The following example shows a CloudTrail log entry that demonstrates the CreateHost action.

```
{
    "EventId": "af4ce349-9f21-43fb-8003-267fbf9b1a93",
    "EventName": "CreateHost",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-11T12:43:06-08:00",
    "EventSource": "codeconnections.amazonaws.com",
```

```
        "Username": "Mary_Major",
        "Resources": [],
        "CloudTrailEvent": {
            "eventVersion": "1.08",
            "userIdentity": {
                "type": "AssumedRole",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
                "accountId": "123456789012",
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "sessionContext": {
                    "sessionIssuer": {
                        "type": "Role",
                        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
                        "accountId": "123456789012",
                        "userName": "Admin"
                    },
                    "webIdFederationData": {},
                    "attributes": {
                        "creationDate": "2024-01-11T20:09:35Z",
                        "mfaAuthenticated": "false"
                    }
                }
            },
            "eventTime": "2024-01-11T20:43:06Z",
            "eventSource": "codeconnections.amazonaws.com",
            "eventName": "CreateHost",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "52.94.133.137",
            "userAgent": "aws-cli/2.13.30 Python/3.11.6 Darwin/23.2.0 exe/x86_64 prompt/off
 command/codeconnections.create-host",
            "requestParameters": {
                "name": "Demo1",
                "providerType": "GitHubEnterpriseServer",
                "providerEndpoint": "IP"
            },
            "responseElements": {
                "hostArn": "arn:aws:codeconnections:us-east-1:123456789012:host/Demo1-
EXAMPLE"
            },
            "requestID": "974459b3-8a04-4cff-9c8f-0c88647831cc",
            "eventID": "af4ce349-9f21-43fb-8003-267fbf9b1a93",
            "readOnly": false,
```

```
            "eventType": "AwsApiCall",
            "managementEvent": true,
            "recipientAccountId": "123456789012",
            "eventCategory": "Management",
            "tlsDetails": {
                "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
            }
        }
    }
}
```

CreateSyncConfiguration example

The following example shows a CloudTrail log entry that demonstrates the
CreateSyncConfiguration action.

```
{
    "EventId": "be1397e1-eefb-49f0-b4ee-2708c45e94e7",
    "EventName": "CreateSyncConfiguration",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-24T17:38:30+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-24T17:34:55Z",
```

```
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-24T17:38:30Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "CreateSyncConfiguration",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.15.11 Python/3.11.6
 Linux/5.10.205-172.804.amzn2int.x86_64exe/x86_64.amzn.2prompt/offcommand/
codeconnections.create-sync-configuration",
        "requestParameters": {
            "branch": "master",
            "configFile": "filename",
            "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
            "resourceName": "mystack",
            "roleArn": "arn:aws:iam::123456789012:role/my-role",
            "syncType": "CFN_STACK_SYNC"
        },
        "responseElements": {
            "syncConfiguration": {
                "branch": "main",
                "configFile": "filename",
                "ownerId": "owner_ID",
                "providerType": "GitHub",
                "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
                "repositoryName": "MyGitHubRepo",
                "resourceName": "mystack",
                "roleArn": "arn:aws:iam::123456789012:role/my-role",
                "syncType": "CFN_STACK_SYNC"
            }
        },
        "requestID": "bad2f662-3f2a-42c0-b638-6115384896f6",
        "eventID": "be1397e1-eefb-49f0-b4ee-2708c45e94e7",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
```

```
}
```

DeleteConnection example

The following example shows a CloudTrail log entry that demonstrates the `DeleteConnection` action.

```json
{
    "EventId": "672837cd-f977-4fe2-95c7-14280b2af76c",
    "EventName": "DeleteConnection",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-10T13:00:50-08:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::001919387613:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-10T20:41:16Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-10T21:00:50Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "DeleteConnection",
        "awsRegion": "us-east-1",
```

```
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.13.30 Python/3.11.6 Darwin/23.2.0 exe/x86_64 prompt/off
  command/codeconnections.delete-connection",
        "requestParameters": {
            "connectionArn": "arn:aws:codeconnections:us-
  east-1:123456789012:connection/df03df74-8e05-45cf-b420-b39e389dd264"
        },
        "responseElements": null,
        "requestID": "4f26ceab-d665-41df-9e15-5ed0fbb4eca6",
        "eventID": "672837cd-f977-4fe2-95c7-14280b2af76c",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

DeleteHost example

The following example shows a CloudTrail log entry that demonstrates the `DeleteHost` action.

```
{
    "EventId": "6018ba5c-6f24-4a30-b201-16ec19a1687a",
    "EventName": "DeleteHost",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-11T12:56:47-08:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
```

```
                    "sessionIssuer": {
                        "type": "Role",
                        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                        "arn": "arn:aws:iam::123456789012:role/Admin",
                        "accountId": "123456789012",
                        "userName": "Admin"
                    },
                    "webIdFederationData": {},
                    "attributes": {
                        "creationDate": "2024-01-11T20:09:35Z",
                        "mfaAuthenticated": "false"
                    }
                }
            }
        },
        "eventTime": "2024-01-11T20:56:47Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "DeleteHost",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.13.30 Python/3.11.6 Darwin/23.2.0 exe/x86_64 prompt/off
 command/codeconnections.delete-host",
        "requestParameters": {
            "hostArn": "arn:aws:codeconnections:us-east-1:123456789012:host/Demo1-
 EXAMPLE"
        },
        "responseElements": null,
        "requestID": "1b244528-143a-4028-b9a4-9479e342bce5",
        "eventID": "6018ba5c-6f24-4a30-b201-16ec19a1687a",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
 }
```

DeleteSyncConfiguration example

The following example shows a CloudTrail log entry that demonstrates the
DeleteSyncConfiguration action.

```
{
    "EventId": "588660c7-3202-4998-a906-7bb72bcf4438",
    "EventName": "DeleteSyncConfiguration",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-24T17:41:59+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-24T17:34:55Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-24T17:41:59Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "DeleteSyncConfiguration",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "52.94.133.142",
        "userAgent": "aws-
cli/2.15.11Python/3.11.6Linux/5.10.205-172.804.amzn2int.x86_64exe/x86_64.amzn.2prompt/
offcommand/codeconnections.delete-sync-configuration",
        "requestParameters": {
            "syncType": "CFN_STACK_SYNC",
            "resourceName": "mystack"
```

```
        },
        "responseElements": null,
        "requestID": "221e0b1c-a50e-4cf0-ab7d-780154e29c94",
        "eventID": "588660c7-3202-4998-a906-7bb72bcf4438",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

GetConnection example

The following example shows a CloudTrail log entry that demonstrates the `GetConnection`
action.

```
{
    "EventId": "672837cd-f977-4fe2-95c7-14280b2af76c",
    "EventName": "DeleteConnection",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-10T13:00:50-08:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
```

```
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-10T20:41:16Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-10T21:00:50Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "DeleteConnection",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.13.30 Python/3.11.6 Darwin/23.2.0 exe/x86_64 prompt/off
 command/codeconnections.delete-connection",
        "requestParameters": {
            "connectionArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/df03df74-8e05-45cf-b420-b39e389dd264"
        },
        "responseElements": null,
        "requestID": "4f26ceab-d665-41df-9e15-5ed0fbb4eca6",
        "eventID": "672837cd-f977-4fe2-95c7-14280b2af76c",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "001919387613",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

GetHost example

The following example shows a CloudTrail log entry that demonstrates the GetHost action.

```
{
    "EventId": "faa147e7-fe7c-4ab9-a11b-2568a2883c01",
    "EventName": "GetHost",
    "ReadOnly": "true",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "EventTime": "2024-01-11T12:44:34-08:00",
        "EventSource": "codeconnections.amazonaws.com",
        "Username": "Mary_Major",
        "Resources": [],
        "CloudTrailEvent": {
            "eventVersion": "1.08",
            "userIdentity": {
                "type": "AssumedRole",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
                "accountId": "123456789012",
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "sessionContext": {
                    "sessionIssuer": {
                        "type": "Role",
                        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                        "arn": "arn:aws:iam::123456789012:role/Admin",
                        "accountId": "123456789012",
                        "userName": "Admin"
                    },
                    "webIdFederationData": {},
                    "attributes": {
                        "creationDate": "2024-01-11T20:09:35Z",
                        "mfaAuthenticated": "false"
                    }
                }
            },
            "eventTime": "2024-01-11T20:44:34Z",
            "eventSource": "codeconnections.amazonaws.com",
            "eventName": "GetHost",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "52.94.133.137",
            "userAgent": "aws-cli/2.13.30 Python/3.11.6 Darwin/23.2.0 exe/x86_64 prompt/off
 command/codeconnections.get-host",
            "requestParameters": {
                "hostArn": "arn:aws:codeconnections:us-east-1:123456789012:host/Demo1-
EXAMPLE"
            },
            "responseElements": null,
            "requestID": "0ad61bb6-f88f-4f96-92fe-997f017ec2bb",
            "eventID": "faa147e7-fe7c-4ab9-a11b-2568a2883c01",
            "readOnly": true,
            "eventType": "AwsApiCall",
            "managementEvent": true,
```

```
            "recipientAccountId": "123456789012",
            "eventCategory": "Management",
            "tlsDetails": {
                "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
            }
        }
    }
}
```

GetRepositoryLink example

The following example shows a CloudTrail log entry that demonstrates the GetRepositoryLink action.

```
{
    "EventId": "b46acb67-3612-41c7-8987-adb6c9ed4ad4",
    "EventName": "GetRepositoryLink",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-24T02:59:28+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-24T02:58:52Z",
                    "mfaAuthenticated": "false"
                }
```

```
                }
            },
            "eventTime": "2024-01-24T02:59:28Z",
            "eventSource": "codeconnections.amazonaws.com",
            "eventName": "GetRepositoryLink",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "IP",
            "userAgent": "aws-cli/2.15.11
 Python/3.11.6Linux/5.10.205-172.804.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/off
 command/codeconnections.get-repository-link",
            "requestParameters": {
                "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173"
            },
            "responseElements": {
                "repositoryLinkInfo": {
                    "connectionArn": "arn:aws:codeconnections:us-
 east-1:123456789012:connection/7df263cc-f055-4843-adef-4ceaefcb2167",
                    "ownerId": "123456789012",
                    "providerType": "GitHub",
                    "repositoryLinkArn": "arn:aws:codeconnections:us-
 east-1:123456789012:repository-link/6053346f-8a33-4edb-9397-10394b695173",
                    "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
                    "repositoryName": "MyGitHubRepo"
                }
            },
            "requestID": "d46704dd-dbe9-462f-96a6-022a8d319fd1",
            "eventID": "b46acb67-3612-41c7-8987-adb6c9ed4ad4",
            "readOnly": false,
            "eventType": "AwsApiCall",
            "managementEvent": true,
            "recipientAccountId": "123456789012",
            "eventCategory": "Management",
            "tlsDetails": {
                "clientProvidedHostHeader": "api.us-ea-1.codeconnections.aws.dev"
            }
        }
    }
}
```

GetRepositorySyncStatus example

The following example shows a CloudTrail log entry that demonstrates the
GetRepositorySyncStatus action.

```
{
    "EventId": "3e183b74-d8c4-4ad3-9de3-6b5721c522e9",
    "EventName": "GetRepositorySyncStatus",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-25T03:41:44+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-25T02:56:55Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-25T03:41:44Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "GetRepositorySyncStatus",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "52.94.133.138",
        "userAgent": "aws-cli/2.15.11 Python/3.11.6
 Linux/5.10.205-172.807.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/off command/
codeconnections.get-repository-sync-status",
        "errorCode": "ResourceNotFoundException",
        "errorMessage": "Could not find a sync status for repository
 link:6053346f-8a33-4edb-9397-10394b695173",
```

```
            "requestParameters": {
                "branch": "feature-branch",
                "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
                "syncType": "CFN_STACK_SYNC"
            },
            "responseElements": null,
            "requestID": "e0cee3ee-31e8-4ef5-b749-96cdcabbe36f",
            "eventID": "3e183b74-d8c4-4ad3-9de3-6b5721c522e9",
            "readOnly": false,
            "eventType": "AwsApiCall",
            "managementEvent": true,
            "recipientAccountId": "123456789012",
            "eventCategory": "Management",
            "tlsDetails": {
                "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
            }
        }
    }
}
```

GetResourceSyncStatus example

The following example shows a CloudTrail log entry that demonstrates the GetResourceSyncStatus action.

```
{
    "EventId": "9c47054e-f6f6-4345-96d0-9a5af3954a8d",
    "EventName": "GetResourceSyncStatus",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-25T03:44:11+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
```

```
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-25T02:56:55Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-25T03:44:11Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "GetResourceSyncStatus",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.15.11 Python/3.11.6
 Linux/5.10.205-172.807.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/off command/
 codeconnections.get-resource-sync-status",
        "requestParameters": {
            "resourceName": "mystack",
            "syncType": "CFN_STACK_SYNC"
        },
        "responseElements": null,
        "requestID": "e74b5503-d651-4920-9fd2-0f40fb5681e0",
        "eventID": "9c47054e-f6f6-4345-96d0-9a5af3954a8d",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
 }
```

GetSyncBlockerSummary example

The following example shows a CloudTrail log entry that demonstrates the
GetSyncBlockerSummary action.

```
{
    "EventId": "c16699ba-a788-476d-8c6c-47511d76309e",
    "EventName": "GetSyncBlockerSummary",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-25T03:03:02+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-25T02:56:55Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-25T03:03:02Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "GetSyncBlockerSummary",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.15.11 Python/3.11.6
 Linux/5.10.205-172.807.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/off command/
codeconnections.get-sync-blocker-summary",
        "requestParameters": {
            "syncType": "CFN_STACK_SYNC",
            "resourceName": "mystack"
```

```
        },
        "responseElements": {
            "syncBlockerSummary": {
                "resourceName": "mystack",
                "latestBlockers": []
            }
        },
        "requestID": "04240091-eb25-4138-840d-776f8e5375b4",
        "eventID": "c16699ba-a788-476d-8c6c-47511d76309e",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

GetSyncConfiguration example

The following example shows a CloudTrail log entry that demonstrates the [GetSyncConfiguration](#) action.

```
{
    "EventId": "bab9aa16-4553-4206-a1ea-88219233dd25",
    "EventName": "GetSyncConfiguration",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-24T17:40:40+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
```

```
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-24T17:34:55Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-24T17:40:40Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "GetSyncConfiguration",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "52.94.133.142",
        "userAgent": "aws-
cli/2.15.11Python/3.11.6Linux/5.10.205-172.804.amzn2int.x86_64exe/x86_64.amzn.2prompt/
offcommand/codeconnections.get-sync-configuration",
        "requestParameters": {
            "syncType": "CFN_STACK_SYNC",
            "resourceName": "mystack"
        },
        "responseElements": {
            "syncConfiguration": {
                "branch": "main",
                "configFile": "filename",
                "ownerId": "123456789012",
                "providerType": "GitHub",
                "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
                "repositoryName": "MyGitHubRepo",
                "resourceName": "mystack",
                "roleArn": "arn:aws:iam::123456789012:role/my-role",
                "syncType": "CFN_STACK_SYNC"
            }
        },
        "requestID": "0aa8e43a-6e34-4d8f-89fb-5c2d01964b35",
        "eventID": "bab9aa16-4553-4206-a1ea-88219233dd25",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
```

```
            "recipientAccountId": "123456789012",
            "eventCategory": "Management",
            "tlsDetails": {
                "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
            }
        }
    }
}
```

ListConnections example

The following example shows a CloudTrail log entry that demonstrates the [ListConnections](#) action.

```
{
    "EventId": "3f8d80fe-fbe1-4755-903c-4f58fc8262fa",
    "EventName": "ListConnections",
    "ReadOnly": "true",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-08T14:11:23-08:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-08T22:11:02Z",
                    "mfaAuthenticated": "false"
                }
            }
```

```
        },
        "eventTime": "2024-01-08T22:11:23Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "ListConnections",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/1.18.147 Python/2.7.18
 Linux/5.10.201-168.748.amzn2int.x86_64 botocore/1.18.6",
        "requestParameters": {
            "maxResults": 50
        },
        "responseElements": null,
        "requestID": "5d456d59-3e92-44be-b941-a429df59e90b",
        "eventID": "3f8d80fe-fbe1-4755-903c-4f58fc8262fa",
        "readOnly": true,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

ListHosts example

The following example shows a CloudTrail log entry that demonstrates the ListHosts action.

```
{
    "EventId": "f6e9e831-feaf-4ad1-ac47-51681109c401",
    "EventName": "ListHosts",
    "ReadOnly": "true",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-11T13:00:55-08:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
```

```
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-11T20:09:35Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-11T21:00:55Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "ListHosts",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.13.30 Python/3.11.6 Darwin/23.2.0 exe/x86_64 prompt/off
command/codeconnections.list-hosts",
        "requestParameters": {
            "maxResults": 50
        },
        "responseElements": null,
        "requestID": "ea87e2cf-6bf1-4cc7-9666-f3fad85d6d83",
        "eventID": "f6e9e831-feaf-4ad1-ac47-51681109c401",
        "readOnly": true,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

ListRepositoryLinks example

The following example shows a CloudTrail log entry that demonstrates the ListRepositoryLinks action.

```
{
    "EventId": "4f714bbb-0716-4f6e-9868-9b379b30757f",
    "EventName": "ListRepositoryLinks",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-24T01:57:29+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-24T01:43:49Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-24T01:57:29Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "ListRepositoryLinks",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
```

```
            "userAgent": "aws-
cli/2.15.11Python/3.11.6Linux/5.10.205-172.804.amzn2int.x86_64exe/x86_64.amzn.2prompt/
offcommand/codeconnections.list-repository-links",
        "requestParameters": {
            "maxResults": 50
        },
        "responseElements": {
            "repositoryLinks": [
                {
                    "connectionArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/001f5be2-a661-46a4-b96b-4d277cac8b6e",
                    "ownerId": "123456789012",
                    "providerType": "GitHub",
                    "repositoryLinkArn": "arn:aws:codeconnections:us-
east-1:123456789012:repository-link/be8f2017-b016-4a77-87b4-608054f70e77",
                    "repositoryLinkId": "be8f2017-b016-4a77-87b4-608054f70e77",
                    "repositoryName": "MyGitHubRepo"
                },
                {
                    "connectionArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/7df263cc-f055-4843-adef-4ceaefcb2167",
                    "ownerId": "owner",
                    "providerType": "GitHub",
                    "repositoryLinkArn": "arn:aws:codeconnections:us-
east-1:123456789012:repository-link/6053346f-8a33-4edb-9397-10394b695173",
                    "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
                    "repositoryName": "MyGitHubRepo"
                }
            ]
        },
        "requestID": "7c8967a9-ec15-42e9-876b-0ef58681ec55",
        "eventID": "4f714bbb-0716-4f6e-9868-9b379b30757f",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

ListRepositorySyncDefinitions example

The following example shows a CloudTrail log entry that demonstrates the ListRepositorySyncDefinitions action.

```
{
    "EventId": "12e52dbb-b00d-49ad-875a-3efec36e5aa1",
    "EventName": "ListRepositorySyncDefinitions",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-25T16:56:19+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-25T16:43:03Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-25T16:56:19Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "ListRepositorySyncDefinitions",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
```

```
            "userAgent": "aws-cli/2.15.11 Python/3.11.6
 Linux/5.10.205-172.807.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/off command/
 codeconnections.list-repository-sync-definitions",
        "requestParameters": {
            "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
            "syncType": "CFN_STACK_SYNC",
            "maxResults": 50
        },
        "responseElements": {
            "repositorySyncDefinitions": []
        },
        "requestID": "df31d11d-5dc7-459b-9a8f-396b4769cdd9",
        "eventID": "12e52dbb-b00d-49ad-875a-3efec36e5aa1",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

ListSyncConfigurations example

The following example shows a CloudTrail log entry that demonstrates the ListSyncConfigurations action.

```
{
    "EventId": "aa4ae557-ec31-4151-8d21-9e74dd01344c",
    "EventName": "ListSyncConfigurations",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-24T17:42:06+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
```

```
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-24T17:34:55Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-24T17:42:06Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "ListSyncConfigurations",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.15.11 Python/3.11.6
 Linux/5.10.205-172.804.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/offcommand/
codeconnections.list-sync-configurations",
        "requestParameters": {
            "maxResults": 50,
            "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
            "syncType": "CFN_STACK_SYNC"
        },
        "responseElements": {
            "syncConfigurations": [
                {
                    "branch": "feature-branch",
                    "configFile": "filename.yaml",
                    "ownerId": "owner",
                    "providerType": "GitHub",
                    "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
                    "repositoryName": "MyGitHubRepo",
                    "resourceName": "dkstacksync",
                    "roleArn": "arn:aws:iam::123456789012:role/my-role",
                    "syncType": "CFN_STACK_SYNC"
                }
```

```
            ]
        },
        "requestID": "7dd220b5-fc0f-4023-aaa0-9555cfe759df",
        "eventID": "aa4ae557-ec31-4151-8d21-9e74dd01344c",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

ListTagsForResource example

The following example shows a CloudTrail log entry that demonstrates the ListTagsForResource
action.

```
{
    "EventId": "fc501054-d68a-4325-824c-0e34062ef040",
    "EventName": "ListTagsForResource",
    "ReadOnly": "true",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-25T17:16:56+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "dMary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
```

```
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-25T16:43:03Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-25T17:16:56Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "ListTagsForResource",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.15.11 Python/3.11.6
 Linux/5.10.205-172.807.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/off command/
codeconnections.list-tags-for-resource",
        "requestParameters": {
            "resourceArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/9703702f-bebe-41b7-8fc4-8e6d2430a330"
        },
        "responseElements": null,
        "requestID": "994584a3-4807-47f2-bb1b-a64f0af6c250",
        "eventID": "fc501054-d68a-4325-824c-0e34062ef040",
        "readOnly": true,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

TagResource example

The following example shows a CloudTrail log entry that demonstrates the TagResource action.

```
{
    "EventId": "b7fbc943-2dd1-4c5b-a5ad-fc6d60a011f1",
    "EventName": "TagResource",
    "ReadOnly": "false",
```

```
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-11T12:22:11-08:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-11T20:09:35Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-11T20:22:11Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "TagResource",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.13.30 Python/3.11.6 Darwin/23.2.0 exe/x86_64 prompt/off
 command/codeconnections.tag-resource",
        "requestParameters": {
            "resourceArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/8dcf69d1-3316-4392-ae09-71e038adb6ed",
            "tags": [
                {
                    "key": "Demo1",
                    "value": "hhvh1"
                }
            ]
```

```
        },
        "responseElements": null,
        "requestID": "ba382c33-7124-48c8-a23a-25816ce27604",
        "eventID": "b7fbc943-2dd1-4c5b-a5ad-fc6d60a011f1",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

UnTagResource example

The following example shows a CloudTrail log entry that demonstrates the UntagResource action.

```
{
    "EventId": "8a85cdee-2586-4679-be18-eec34204bc7e",
    "EventName": "UntagResource",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-11T12:31:14-08:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
```

```
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-11T20:09:35Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-11T20:31:14Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "UntagResource",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.13.30 Python/3.11.6 Darwin/23.2.0 exe/x86_64 prompt/off
 command/codeconnections.untag-resource",
        "requestParameters": {
            "resourceArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/8dcf69d1-3316-4392-ae09-71e038adb6ed",
            "tagKeys": [
                "Project",
                "ReadOnly"
            ]
        },
        "responseElements": null,
        "requestID": "05ef26a4-8c39-4f72-89bf-0c056c51b8d7",
        "eventID": "8a85cdee-2586-4679-be18-eec34204bc7e",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

UpdateHost example

The following example shows a CloudTrail log entry that demonstrates the UpdateHost action.

```
"Events": [{
        "EventId": "4307cf7d-6d1c-40d9-a659-1bb41b31a2b6",
```

```
            "EventName": "UpdateHost",
            "ReadOnly": "false",
            "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "EventTime": "2024-01-11T12:54:32-08:00",
            "EventSource": "codeconnections.amazonaws.com",
            "Username": "Mary_Major",
            "Resources": [],
            "CloudTrailEvent": "eventVersion": "1.08",
            "userIdentity": {
                "type": "AssumedRole",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
                "accountId": "123456789012",
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "sessionContext": {
                    "sessionIssuer": {
                        "type": "Role",
                        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                        "arn": "arn:aws:iam::123456789012:role/Admin",
                        "accountId": "123456789012",
                        "userName": "Admin"
                    },
                    "webIdFederationData": {},
                    "attributes": {
                        "creationDate": "2024-01-11T20:09:35Z",
                        "mfaAuthenticated": "false"
                    }
                }
            },
            "eventTime": "2024-01-11T20:54:32Z",
            "eventSource": "codeconnections.amazonaws.com",
            "eventName": "UpdateHost",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "IP",
            "userAgent": "aws-cli/2.13.30 Python/3.11.6 Darwin/23.2.0 exe/x86_64 prompt/off
 command/codeconnections.update-host",
            "requestParameters": {
                "hostArn": "arn:aws:codeconnections:us-east-1:123456789012:host/
Demo1-34e70ecb",
                "providerEndpoint": "https://54.218.245.167"
            },
            "responseElements": null,
            "requestID": "b17f46ac-1acb-44ab-a9f5-c35c20233441",
            "eventID": "4307cf7d-6d1c-40d9-a659-1bb41b31a2b6",
```

```
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
```

UpdateRepositoryLink example

The following example shows a CloudTrail log entry that demonstrates the [UpdateRepositoryLink](#) action.

```
{
    "EventId": "be358c9a-5a8f-467e-8585-2860070be4fe",
    "EventName": "UpdateRepositoryLink",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-24T02:03:24+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-24T01:43:49Z",
                    "mfaAuthenticated": "false"
```

```
                }
            }
        },
        "eventTime": "2024-01-24T02:03:24Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "UpdateRepositoryLink",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-
cli/2.15.11Python/3.11.6Linux/5.10.205-172.804.amzn2int.x86_64exe/x86_64.amzn.2prompt/
offcommand/codeconnections.update-repository-link",
        "requestParameters": {
            "connectionArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/7df263cc-f055-4843-adef-4ceaefcb2167",
            "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173"
        },
        "responseElements": {
            "repositoryLinkInfo": {
                "connectionArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/7df263cc-f055-4843-adef-4ceaefcb2167",
                "ownerId": "owner",
                "providerType": "GitHub",
                "repositoryLinkArn": "arn:aws:codeconnections:us-
east-1:123456789012:repository-link/6053346f-8a33-4edb-9397-10394b695173",
                "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
                "repositoryName": "MyGitHubRepo"
            }
        },
        "additionalEventData": {
            "providerAction": "UpdateRepositoryLink"
        },
        "requestID": "e01eee49-9393-4983-89e4-d1b3353a70d9",
        "eventID": "be358c9a-5a8f-467e-8585-2860070be4fe",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

UpdateSyncBlocker example

The following example shows a CloudTrail log entry that demonstrates the [UpdateSyncBlocker](#) action.

```
{
    "EventId": "211d19db-9f71-4d93-bf90-10f9ddefed88",
    "EventName": "UpdateSyncBlocker",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-25T03:01:05+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-25T02:56:55Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-25T03:01:05Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "UpdateSyncBlocker",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
```

```
        "userAgent": "aws-cli/2.15.11 Python/3.11.6
 Linux/5.10.205-172.807.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/off command/
 codeconnections.update-sync-blocker",
        "requestParameters": {
            "id": "ID",
            "syncType": "CFN_STACK_SYNC",
            "resourceName": "mystack",
            "resolvedReason": "Reason"
        },
        "responseElements": null,
        "requestID": "eea03b39-b299-4099-ba55-608480f8d96d",
        "eventID": "211d19db-9f71-4d93-bf90-10f9ddefed88",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
}
```

UpdateSyncConfiguration example

The following example shows a CloudTrail log entry that demonstrates the
UpdateSyncConfiguration action.

```
{
    "EventId": "d961c94f-1881-4fe8-83bf-d04cb9f22577",
    "EventName": "UpdateSyncConfiguration",
    "ReadOnly": "false",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "EventTime": "2024-01-24T17:40:55+00:00",
    "EventSource": "codeconnections.amazonaws.com",
    "Username": "Mary_Major",
    "Resources": [],
    "CloudTrailEvent": {
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Mary_Major",
```

```
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                    "arn": "arn:aws:iam::123456789012:role/Admin",
                    "accountId": "123456789012",
                    "userName": "Admin"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2024-01-24T17:34:55Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2024-01-24T17:40:55Z",
        "eventSource": "codeconnections.amazonaws.com",
        "eventName": "UpdateSyncConfiguration",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "IP",
        "userAgent": "aws-cli/2.15.11
 Python/3.11.6Linux/5.10.205-172.804.amzn2int.x86_64exe/x86_64.amzn.2prompt/offcommand/
codeconnections.update-sync-configuration",
        "requestParameters": {
            "branch": "feature-branch",
            "resourceName": "mystack",
            "syncType": "CFN_STACK_SYNC"
        },
        "responseElements": {
            "syncConfiguration": {
                "branch": "feature-branch",
                "configFile": "filename",
                "ownerId": "owner",
                "providerType": "GitHub",
                "repositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
                "repositoryName": "MyGitHubRepo",
                "resourceName": "mystack",
                "roleArn": "arn:aws:iam::123456789012:role/my-role",
                "syncType": "CFN_STACK_SYNC"
            }
        },
        "requestID": "2ca545ef-4395-4e1f-b14a-2750481161d6",
```

```
        "eventID": "d961c94f-1881-4fe8-83bf-d04cb9f22577",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "123456789012",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "api.us-east-1.codeconnections.aws.dev"
        }
    }
 }
```

AWS CodeConnections and interface VPC endpoints (AWS PrivateLink)

You can establish a private connection between your VPC and AWS CodeConnections by creating an *interface VPC endpoint*. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to privately access AWS CodeConnections 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 AWS CodeConnections APIs, because traffic between your VPC and AWS CodeConnections 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 AWS CodeConnections VPC endpoints

Before you set up an interface VPC endpoint for AWS CodeConnections, ensure that you review Interface endpoints in the *Amazon VPC User Guide*.

AWS CodeConnections supports making calls to all of its API actions from your VPC.

VPC endpoints are supported in all AWS CodeConnections Regions.

VPC endpoint concepts

The following are the key concepts for VPC endpoints:

VPC endpoint

> The entry point in your VPC that enables you to connect privately to a service. The following
> are the different types of VPC endpoints. You create the type of VPC endpoint required by the
> supported service.
>
> - VPC endpoints for AWS CodeConnections actions
> - VPC endpoints for AWS CodeConnections webhooks

AWS PrivateLink

> A technology that provides private connectivity between VPCs and services.

VPC endpoints for AWS CodeConnections actions

You can manage VPC endpoints for the AWS CodeConnections service.

Creating interface VPC endpoints for AWS CodeConnections actions

You can create a VPC endpoint for the AWS CodeConnections 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*.

To start using connections with your VPC, create an interface VPC endpoint for AWS
CodeConnections. When you create a VPC endpoint for AWS CodeConnections, choose **AWS
Services**, and in **Service Name**, choose:

- **com.amazonaws.*region*.codestar-connections.api**: This option creates a VPC endpoint for
  AWS CodeConnections API operations. For example, choose this option if your users use the AWS
  CLI, the AWS CodeConnections API, or the AWS SDKs to interact with AWS CodeConnections for
  operations such as `CreateConnection`, `ListConnections`, and `CreateHost`.

For the **Enable DNS name** option, if you select private DNS for the endpoint, you can make
API requests to AWS CodeConnections using its default DNS name for the Region, for example,
`codestar-connections.us-east-1.amazonaws.com`.

> ⚠️ **Important**
>
> Private DNS is enabled by default for endpoints created for AWS services and AWS
> Marketplace Partner services.

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

Creating a VPC endpoint policy for AWS CodeConnections actions

You can attach an endpoint policy to your VPC endpoint that controls access to AWS CodeConnections. 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*.

> **ⓘ Note**
>
> The com.amazonaws.*region*.codestar-connections.webhooks endpoint does not support policies.

Example: VPC endpoint policy for AWS CodeConnections actions

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

```
{
  "Statement": [
    {
      "Sid": "GetConnectionOnly",
      "Principal": "*",
      "Action": [
        "codestar-connections:GetConnection"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

VPC endpoints for AWS CodeConnections webhooks

AWS CodeConnections creates webhook endpoints for you when you create or delete a host with VPC configuration. The endpoint name is **com.amazonaws.*region*.codestar-connections.webhooks**.

With the VPC endpoint for GitHub webhooks, hosts can send event data via webhooks to your integrated AWS services over the Amazon network.

> ⚠️ **Important**
>
> When you set up your host for GitHub Enterprise Server, AWS CodeConnections creates a VPC endpoint for webhooks event data for you. If you created your host before November 24, 2020, and you want to use VPC PrivateLink webhook endpoints, you must first [delete](#) your host and then [create](#) a new host.

AWS CodeConnections manages the lifecycle of these endpoints. To delete the endpoint, you must delete the corresponding host resource.

How webhook endpoints for AWS CodeConnections hosts are used

The webhook endpoint is where webhooks from third-party repositories are sent for AWS CodeConnections processing. A webhook describes a customer action. When you perform a `git push`, the webhook endpoint receives a webhook from the provider detailing the push. For example, AWS CodeConnections can notify CodePipeline to start your pipeline.

For cloud providers, such as Bitbucket, or GitHub Enterprise Server hosts that do not use a VPC, the webhook VPC endpoint does not apply because the providers are sending webhooks to AWS CodeConnections where the Amazon network is not used.

Troubleshooting connections

The following information might help you troubleshoot common issues with connections to resources in AWS CodeBuild, AWS CodeDeploy, and AWS CodePipeline.

Topics

- [I cannot create connections](#)
- [I get a permissions error when I try to create or complete a connection](#)

- [I get a permissions error when I try to use a connection](#)

- [Connection is not in available state or is no longer pending](#)

- [Add GitClone permissions for connections](#)

- [Host is not in available state](#)

- [Troubleshooting a host with connection errors](#)

- [I'm unable to create a connection for my host](#)

- [Troubleshooting VPC configuration for your host](#)

- [Troubleshooting webhook VPC endpoints (PrivateLink) for GitHub Enterprise Server connections](#)

- [Troubleshooting for a host created before November 24, 2020](#)

- [Unable to create the connection for a GitHub repository](#)

- [Edit your GitHub Enterprise Server connection app permissions](#)

- [Connections error when connecting to GitHub: "A problem occurred, make sure cookies are enabled in your browser" or "An organization owner must install the GitHub app"](#)

- [Connections service prefix in resources might need to be updated for IAM policies](#)

- [Permissions error due to service prefix in resources created using the console](#)

- [Connection and host setup for installed providers supporting organizations](#)

- [I want to increase my limits for connections](#)

I cannot create connections

You might not have permissions to create a connection. For more information, see [Permissions and examples for AWS CodeConnections](#).

I get a permissions error when I try to create or complete a connection

The following error message might be returned when you try to create or view a connection in the CodePipeline console.

User: *username* is not authorized to perform: *permission* on resource: *connection-ARN*

If this message appears, make sure that you have sufficient permissions.

The permissions to create and view connections in the AWS Command Line Interface (AWS CLI) or the AWS Management Console are only part of the permissions that you need to create and complete connections on the console. The permissions required to simply view, edit, or create a connection and then complete the pending connection should be scoped down for users who

only need to perform certain tasks. For more information, see [Permissions and examples for AWS](#) [CodeConnections](#).

I get a permissions error when I try to use a connection

One or both of the following error messages might be returned if you try to use a connection in the CodePipeline console, even though you have the permissions to list, get, and create permissions.

You have failed to authenticate your account.

User: *username* is not authorized to perform: codestar-connections:UseConnection on resource: *connection-ARN*

If this occurs, make sure that you have sufficient permissions.

Make sure you have the permissions to use a connection, including listing the available repositories in the provider location. For more information, see [Permissions and examples for AWS](#) [CodeConnections](#).

Connection is not in available state or is no longer pending

If the console displays a message that a connection is not in an available state, choose **Complete connection**.

If you choose to complete the connection and a message appears that the connection is not in a pending state, you can cancel the request because the connection is already in an available state.

Add GitClone permissions for connections

When you use an AWS CodeStar connection in a source action and a CodeBuild action, there are two ways the input artifact can be passed to the build:

- The default: The source action produces a zip file that contains the code that CodeBuild downloads.
- Git clone: The source code can be directly downloaded to the build environment.

  The Git clone mode allows you to interact with the source code as a working Git repository. To use this mode, you must grant your CodeBuild environment permissions to use the connection.

To add permissions to your CodeBuild service role policy, you create a customer managed policy that you attach to your CodeBuild service role. The following steps create a policy where the

UseConnection permission is specified in the `action` field, and the connection Amazon Resource Name (ARN) is specified in the `Resource` field.

To use the console to add the UseConnection permissions

1. To find the connection ARN for your pipeline, open your pipeline and choose the (**i**) icon on your source action. The Configuration pane opens, and the connection ARN appears next to **ConnectionArn**. You add the connection ARN to your CodeBuild service role policy.

2. To find your CodeBuild service role, open the build project used in your pipeline and navigate to the **Build details** tab.

3. In the Environment section, choose the **Service role** link. This opens the AWS Identity and Access Management (IAM) console, where you can add a new policy that grants access to your connection.

4. In the IAM console, choose **Attach policies**, and then choose **Create policy**.

   Use the following sample policy template. Add your connection ARN in the `Resource` field, as shown in this example.

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "codestar-connections:UseConnection",
               "Resource": "arn:aws:iam::*:role/Service*"
           }
       ]
   }
   ```

   On the **JSON** tab, paste your policy.

5. Choose **Review policy**. Enter a name for the policy (for example, **connection-permissions**), and then choose **Create policy**.

6. Return to the service role **Attach Permissions** page, refresh the policy list, and select the policy you just created. Choose **Attach policies**.

Host is not in available state

If the console displays a message that a host is not in an `Available` state, choose **Set up host**.

The first step for host creation results in the created host now in a `Pending` state. To move the host to an `Available` state, you must choose to set up the host in the console. For more information, see [Set up a pending host](#).

> ℹ️ **Note**
>
> You cannot use the AWS CLI to set up a `Pending` host.

Troubleshooting a host with connection errors

Connections and hosts can move into the error state if the underlying GitHub app is deleted or modified. Hosts and connections in the error state cannot be recovered and the host must be recreated.

- Actions such as changing the app pem key, changing the app name (after initial creation) will cause the host and all associated connections to go into the error state.

If the console or CLI returns a host or a connection related to a host with an `Error` state, you might need to perform the following step:

- Delete and recreate the host resource and then reinstall the host registration app. For more information, see [Create a host](#).

I'm unable to create a connection for my host

To create a connection or host, the following conditions are required.

- Your host must be in the **AVAILABLE** state. For more information, see
- Connections must be created in the same Region as the host.

Troubleshooting VPC configuration for your host

When you create a host resource, you must provide network connection or VPC information for the infrastructure where your GitHub Enterprise Server instance is installed. For troubleshooting your VPC or subnet configuration for your host, use the example VPC information shown here as a reference.

> **ⓘ Note**
>
> Use this section for troubleshooting that is related to your GitHub Enterprise Server host configuration within an Amazon VPC. For troubleshooting that is related to your connection that is configured to use the webhook endpoint for VPC (PrivateLink), see [Troubleshooting webhook VPC endpoints (PrivateLink) for GitHub Enterprise Server connections](#).

For this example, you would use the following process to configure the VPC and server where your GitHub Enterprise Server instance will be installed:

1. Create a VPC. For more information, see https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#Create-VPC.

2. Create a subnet in your VPC. For more information, see https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#AddaSubnet.

3. Launch an instance into your VPC. For more information, see https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#VPC_Launch_Instance.

> **ⓘ Note**
>
> Each VPC can only be associated with one host (GitHub Enterprise Server instance) at a time.

The following image shows an EC2 instance launched using the GitHub Enterprise AMI.

When you use a VPC for a GitHub Enterprise Server connection, you must provide the following for your infrastructure when you set up your host:

- **VPC ID:** The VPC for the server where your GitHub Enterprise Server instance is installed or a VPC which has access to your installed GitHub Enterprise Server instance through VPN or Direct Connect.

- **Subnet ID or IDs:** The subnet for the server where your GitHub Enterprise Server instance is installed or a subnet with access to your installed GitHub Enterprise Server instance through VPN or Direct Connect.

- **Security group or groups:** The security group for the server where your GitHub Enterprise Server instance is installed or a security group with access to your installed GitHub Enterprise Server instance through VPN or Direct Connect.

- **Endpoint:** Have your server endpoint ready and continue to the next step.

For more information about working with VPCs and subnets, see VPC and Subnet Sizing for IPv4 in the *Amazon VPC User Guide*.

Topics

- I'm unable to get a host in pending state
- I'm unable to get a host in available state
- My connection/host was working and has stopped working now

- I'm unable to delete my network interfaces

I'm unable to get a host in pending state

If your host enters the VPC_CONFIG_FAILED_INITIALIZATION state, this is likely because of an issue with the VPC, subnets, or security groups that you have selected for your host.

- The VPC, subnets, and security groups must all belong to the account creating the host.

- The subnets and security groups must belong to the selected VPC.

- Each provided subnet must be in different Availability Zones.

- The user creating the host must have the following IAM permissions:

```
ec2:CreateNetworkInterface
ec2:CreateTags
ec2:DescribeDhcpOptionsec2:DescribeNetworkInterfaces
ec2:DescribeSubnets
ec2:DeleteNetworkInterface
ec2:DescribeVpcs
ec2:CreateVpcEndpoint
ec2:DeleteVpcEndpoints
ec2:DescribeVpcEndpoints
```

I'm unable to get a host in available state

If you are unable to complete the CodeConnections app setup for your host, it may be because of an issue with your VPC configurations or your GitHub Enterprise Server instance.

- If you are not using a public certificate authority, you will need to provide a TLS certificate to your host that is used by your GitHub Enterprise Instance. The TLS Certificate value should be the public key of the certificate.

- You need to be an administrator of the GitHub Enterprise Server instance in order to create GitHub apps.

My connection/host was working and has stopped working now

If a connection/host was working before and is not working now, it could be due to a configuration change in your VPC or the GitHub app has been modified. Check the following:

- The security group attached to the host resource you created for your connection has now changed or no longer has access to the GitHub Enterprise Server. CodeConnections requires a security group which has connectivity to the GitHub Enterprise Server instance.

- DNS Server IP has recently changed. You can verify this by checking the DHCP options attached to the VPC specified in the host resource you created for your connection. Note that if you've recently moved from AmazonProvidedDNS to custom DNS Server or started using a new custom DNS Server, the host/connection would stop working. In order to fix this, delete your existing host and re-create it, which would store the latest DNS settings in our database.

- The network ACLs settings have changed and are no longer allowing HTTP connections to the subnet where your GitHub Enterprise Server infrastructure is located.

- Any configurations of the CodeConnections app on your GitHub Enterprise Server have changed. Modifications to any of the configurations, such as URLs or app secrets, can break the connectivity between your installed GitHub Enterprise Server instance and CodeConnections.

I'm unable to delete my network interfaces

If you are unable to detect your network interfaces, check the following:

- The Network Interfaces created by CodeConnections can only be deleted by deleting the host. They cannot be deleted manually by the user.

- You must have the following permissions:

```
ec2:DescribeNetworkInterfaces
ec2:DeleteNetworkInterface
```

Troubleshooting webhook VPC endpoints (PrivateLink) for GitHub Enterprise Server connections

When you create a host with VPC configuration, the webhook VPC endpoint is created for you.

> **ⓘ Note**
>
> Use this section for troubleshooting that is related to your connection that is configured to use the webhook endpoint for VPC (PrivateLink). For troubleshooting that is related to your GitHub Enterprise Server host configuration within an Amazon VPC, see Troubleshooting VPC configuration for your host.

When you create a connection to an installed provider type, and you have specified that your server is configured within a VPC, then AWS CodeConnections creates your host, and the VPC endpoint (PrivateLink) for webhooks is created for you. This enables the host to send event data via webhooks to your integrated AWS services over the Amazon network. For more information, see AWS CodeConnections and interface VPC endpoints (AWS PrivateLink).

Topics

- I'm unable to delete my webhook VPC endpoints

I'm unable to delete my webhook VPC endpoints

AWS CodeConnections manages the lifecycle of the webhook VPC endpoints for your host. If you want to delete the endpoint, you must do this by deleting the corresponding host resource.

- The webhook VPC endpoints (PrivateLink) created by CodeConnections can only be deleted by deleting the host. They cannot be deleted manually.

- You must have the following permissions:

```
ec2:DescribeNetworkInterfaces
ec2:DeleteNetworkInterface
```

Troubleshooting for a host created before November 24, 2020

As of November 24, 2020, when AWS CodeConnections sets up your host, an additional VPC endpoint (PrivateLink) support is set up for you. For hosts created before this update, use this troubleshooting section.

For more information, see AWS CodeConnections and interface VPC endpoints (AWS PrivateLink).

Topics

- I have a host that was created before November 24, 2020 and I want to use VPC endpoints (PrivateLink) for webhooks
- I'm unable to get a host in available state (VPC error)

I have a host that was created before November 24, 2020 and I want to use VPC endpoints (PrivateLink) for webhooks

When you set up your host for GitHub Enterprise Server, the webhook endpoint is created for you. Connections now use VPC PrivateLink webhook endpoints. If you created your host before November 24, 2020, and you want to use VPC PrivateLink webhook endpoints, you must first delete your host and then create a new host.

I'm unable to get a host in available state (VPC error)

If your host was created before November 24, 2020, and you are unable to complete the CodeConnections app setup for your host, it may be because of an issue with your VPC configurations or your GitHub Enterprise Server instance.

Your VPC will need a NAT Gateway (or outbound internet access) so that your GitHub Enterprise Server instance can send egress network traffic for GitHub webhooks.

Unable to create the connection for a GitHub repository

Problem:

Because a connection to a GitHub repository uses the AWS Connector for GitHub, you need organization owner permissions or admin permissions to the repository to create the connection.

Possible fixes: For information about permission levels for a GitHub repository, see https://docs.github.com/en/free-pro-team@latest/github/setting-up-and-managing-organizations-and-teams/permission-levels-for-an-organization.

Edit your GitHub Enterprise Server connection app permissions

If you installed the app for GitHub Enterprise Server on or before December 23, 2020, you might need to give the app Read-only access to members of the organization. If you are the GitHub app owner, follow these steps to edit the permissions for the app that was installed when your host was created.

> ⓘ **Note**
>
> You must complete these steps on your GitHub Enterprise Server instance, and you must be the GitHub app owner.

1. In GitHub Enterprise Server, from the drop-down option on your profile photo, choose **Settings**.

2. Choose **Developer settings**, and then choose **GitHub Apps**.

3. In the list of apps, choose the name of the app for your connection, and then choose **Permissions and events** in the settings display.

4. Under **Organization permissions**, for **Members**, choose **Read-only** from the **Access** drop-down.
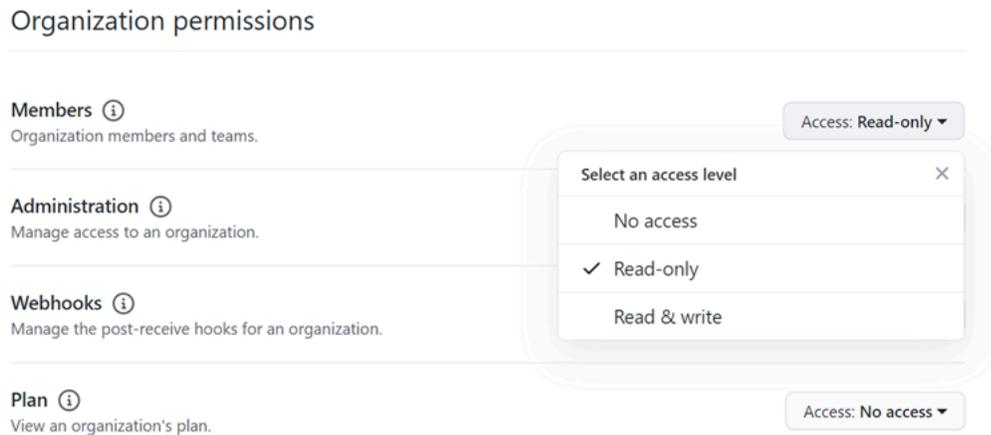
5. In **Add a note to users**, add a description of the reason for the update. Choose **Save changes**.

Connections error when connecting to GitHub: "A problem occurred, make sure cookies are enabled in your browser" or "An organization owner must install the GitHub app"

Problem:

To create the connection for a GitHub repository, you must be the GitHub organization owner. For repositories that are not under an organization, you must be the repository owner. When a connection is created by someone other than the organization owner, a request is created for the organization owner, and one of the following errors display:

A problem occurred, make sure cookies are enabled in your browser

OR

An organization owner must install the GitHub app

Possible fixes: For repositories in a GitHub organization, the organization owner must create the connection to the GitHub repository. For repositories that are not under an organization, you must be the repository owner.

Connections service prefix in resources might need to be updated for IAM policies

On March 29, 2024, the service was renamed from AWS CodeStar Connections to AWS CodeConnections. Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console. The service prefix for resources created using the console is `codeconnections`. New SDK/CLI resources are created with `codeconnections` in the resource ARN. Resources created will automatically have the new service prefix.

The following are the resources that are created in AWS CodeConnections:

* Connections
* Hosts

Problem:

Resources that have been created with `codestar-connections` in the ARN will not automatically be renamed to the new service prefix in the resource ARN. Creating a new resource will create a resource that has the connections service prefix. However, IAM policies with the `codestar-connections` service prefix will not work for resources with the new service prefix.

Possible fixes: To avoid access or permissions issues for the resources, complete the following actions:

* Update IAM policies for the new service prefix. Otherwise, resources renamed or created will not be able to use the IAM policies.
* Update resources for the new service prefix by creating them using the console or CLI/CDK/CFN.

Update the actions, resources, and conditions in the policy as appropriate. In the following example, the `Resource` field has been updated for both service prefixes.

JSON

```
{
```

```
        "Version":"2012-10-17",
        "Statement": {
            "Effect": "Allow",
            "Action": [
                "codeconnections:UseConnection"
            ],
            "Resource": [
                "arn:aws:codestar-connections:*:*:connection/*",
                "arn:aws:codeconnections:*:*:connection/*"
            ]
        }
    }
```

Permissions error due to service prefix in resources created using the console

Currently, connections resources that are created using the console will only have the `codestar-connections` service prefix. For resources created using the console, policy statement actions must include `codestar-connections` as the service prefix.

> **ⓘ Note**
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

Problem:

When creating a connections resource using the console, the `codestar-connections` service prefix must be used in the policy. When using a policy with the `codeconnections` service prefix in the policy, connections resources created using the console receive the following error message:

```
User: user_ARN is not authorized to perform: codestar-connections:action on
 resource: resource_ARN because no identity-based policy allows the codestar-
connections:action action
```

Possible fixes: For resources created using the console, policy statement actions must include `codestar-connections` as the service prefix, as shown in the policy example in Example: A policy for creating AWS CodeConnections with the console.

Connection and host setup for installed providers supporting organizations

For installed providers that support organizations, such as GitHub Organizations, you don't pass an available host. You create a new host for each connection in your organization and be sure to enter the same information in the following network fields:

- **VPC ID**
- **Subnet ID**
- **Security group IDs**

See the related steps to create a [GHES connection](#) or a [GitLab self-managed connection](#).

I want to increase my limits for connections

You can request a limit increase for certain limits in CodeConnections. For more information, see [Quotas for connections](#).

Quotas for connections

The following tables list the quotas (also referred to as *limits*) for connections in the Developer Tools console.

Quotas in this table apply per AWS Region and can be increased. For AWS Region information and quotas that can be changed, see [AWS service quotas](#).

> **ⓘ Note**
>
> You must enable the Europe (Milan) AWS Region before you can use it. For more information, see [Enabling a Region](#).

Resource	Default limit
Maximum number of connections per AWS account	250

Quotas in this table are fixed and cannot be changed.

Resource	Default limit
Maximum characters in connection names	32 characters
Maximum number of hosts per AWS account	50
Maximum number of repository links	100
Maximum number of CloudFormation stack sync configurations	100
Maximum number of sync configurations per repository link	100
Maximum number of sync configurations per branch	50

IP addresses to add to your allow list

If you implement IP filtering, or allowing certain IP addresses on Amazon EC2 instances, add the following IP addresses to your allow list. Doing so enables connections to providers, such as GitHub and Bitbucket.

The following table lists the IP addresses for connections in the Developer Tools console by AWS Region.

> **Note**
>
> For the Europe (Milan) Region, you must enable this Region before you can use it. For more information, see Enabling a Region.

Region	IP addresses
US West (Oregon) (us-west-2)	35.160.210.199, 54.71.206.108, 54.71.36.205
US East (N. Virginia) (us-east-1)	3.216.216.90, 3.216.243.220, 3.217.241.85

Region	IP addresses
Europe (Ireland) (eu-west-1)	34.242.64.82, 52.18.37.201, 54.77.75.62
US East (Ohio) (us-east-2)	18.217.188.190, 18.218.158.91, 18.220.4.80
Asia Pacific (Singapore) (ap-southeast-1)	18.138.171.151, 18.139.22.70, 3.1.157.176
Asia Pacific (Sydney) (ap-southeast-2)	13.236.59.253, 52.64.166.86, 54.206.1.112
Asia Pacific (Tokyo) (ap-northeast-1)	52.196.132.231, 54.95.133.227, 18.181.13.91
Europe (Frankfurt) (eu-central-1)	18.196.145.164, 3.121.252.59, 52.59.104.195
Asia Pacific (Seoul) (ap-northeast-2)	13.125.8.239, 13.209.223.177, 3.37.200.23
Asia Pacific (Mumbai) (ap-south-1)	13.234.199.152, 13.235.29.220, 35.154.230.124
South America (São Paulo) (sa-east-1)	18.229.77.26, 54.233.226.52, 54.233.207.69
Canada (Central) (ca-central-1)	15.222.219.210, 35.182.166.138, 99.79.111.198
Europe (London) (eu-west-2)	3.9.97.205, 35.177.150.185, 35.177.200.225
US West (N. California) (us-west-1)	52.52.16.175, 52.8.63.87
Europe (Paris) (eu-west-3)	35.181.127.138, 35.181.145.22, 35.181.20.200
Europe (Stockholm) (eu-north-1)	13.48.66.148, 13.48.8.79, 13.53.78.182
Europe (Milan) (eu-south-1)	18.102.28.105, 18.102.35.130, 18.102.8.116
AWS GovCloud (US-East)	18.252.168.157, 18.252.207.77, 18.253.185.119

Security for features of the Developer Tools console

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

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

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS CodeStar Notifications and AWS CodeConnections, 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 AWS CodeStar Notifications and AWS CodeConnections. The following topics show you how to configure AWS CodeStar Notifications and AWS CodeConnections to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS CodeStar Notifications and AWS CodeConnections resources.

For more information about security for the services in the Developer Tools console, see the following:

- CodeBuild Security
- CodeCommit Security
- CodeDeploy Security
- CodePipeline Security

Understanding notification contents and security

Notifications provide information about resources to users who are subscribed to the notification rule targets that you configure. This information can include details about your developer tool

resources, including repository contents, build statuses, deployment statuses, and pipeline executions.

For example, you can configure a notification rule for a repository in CodeCommit to include comments on commits or pull requests. If so, the notifications sent in response to that rule might contain the line or lines of code referenced in that comment. Similarly, you can configure a notification rule for a build project in CodeBuild to include successes or failures for build states and phases. Notifications sent in response to that rule will contain that information.

You can configure a notification rule for a pipeline in CodePipeline to include information about manual approvals, and notifications sent in response to that rule might contain the name of the person providing that approval. You can configure a notification rule for an application in CodeDeploy to indicate deployment success, and notifications sent in response to that rule might contain information about the deployment target.

Notifications can include project-specific information such as build statuses, lines of code that have comments, deployment states, and pipeline approvals. So to help ensure the security of your project, make sure that you regularly review both the targets of notification rules and the list of subscribers of the Amazon SNS topics specified as targets. Additionally, the content of notifications sent in response to events might change as additional features are added to the underlying services. This change can happen without notice to already-existing notification rules. Consider reviewing the contents of notification messages periodically to help ensure that you understand what is being sent, as well as to whom it is being sent.

For more information about the event types available for notification rules, see Notification concepts.

You can choose to limit the details included in notifications to only what is included in an event. This is referred to as the **Basic** detail type. These events contain exactly the same information as is sent to Amazon EventBridge and Amazon CloudWatch Events.

Developer Tools console services, such as CodeCommit, might choose to add information about some or all of their event types in notification messages beyond what is available in an event. This supplemental information could be added at any time to enhance current event types or supplement future event types. You can choose to include any supplemental information about the event, if available, in the notification by choosing the **Full** detail type. For more information, see Detail types.

Data protection in AWS CodeStar Notifications and AWS CodeConnections

The AWS shared responsibility model applies to data protection in AWS CodeStar Notifications and AWS CodeConnections. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

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

- Use multi-factor authentication (MFA) with each account.

- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.

- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.

- Use AWS encryption solutions, along with all default security controls within AWS services.

- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.

- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with AWS CodeStar Notifications and AWS CodeConnections 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.

Identity and access management for AWS CodeStar Notifications and AWS CodeConnections

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 AWS CodeStar Notifications and AWS CodeConnections resources. IAM is an AWS service that you can use with no additional charge.

> **ⓘ Note**
>
> Actions for resources that are created under the new service prefix `codeconnections` are available. Creating a resource under the new service prefix will use `codeconnections` in the resource ARN. Actions and resources for the `codestar-connections` service prefix remain available. When specifying a resource in the IAM policy, the service prefix needs to match that of the resource.

Topics

- [Audience](#)
- [Authenticating with identities](#)
- [Managing access using policies](#)
- [How features in the developer tools console work with IAM](#)
- [AWS CodeConnections permissions reference](#)
- [Identity-based policy examples](#)
- [Using tags to control access to AWS CodeConnections resources](#)
- [Using notifications and connections in the console](#)
- [Allow users to view their own permissions](#)
- [Troubleshooting AWS CodeStar Notifications and AWS CodeConnections identity and access](#)
- [Using service-linked roles for AWS CodeStar Notifications](#)
- [Using service-linked roles for AWS CodeConnections](#)
- [AWS managed policies for AWS CodeConnections](#)

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see [Troubleshooting AWS CodeStar Notifications and AWS CodeConnections identity and access](#))
- **Service administrator** - determine user access and submit permission requests (see [How features in the developer tools console work with IAM](#))
- **IAM administrator** - write policies to manage access (see [Identity-based policy examples](#))

Authenticating with identities

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

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see [How to sign in to your AWS account](#) in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see [AWS Signature Version 4 for API requests](#) in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see [Tasks that require root user credentials](#) in the *IAM User Guide*.

IAM users and groups

An [*IAM user*](#) is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see [Require human users to use federation with an identity provider to access AWS using temporary credentials](#) in the *IAM User Guide*.

An [*IAM group*](#) specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see [Use cases for IAM users](#) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by [switching from a user to an IAM role (console)](#) or by calling an AWS CLI or AWS API operation. For more information, see [Methods to assume a role](#) in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see [Cross account resource access in IAM](#) 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 defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see [Overview of JSON policies](#) in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

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

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see [Choose between managed policies and inline policies](#) in the *IAM User Guide*.

How features in the developer tools console work with IAM

Before you use IAM to manage access to features in the Developer Tools console, you should understand which IAM features are available to use with it. To get a high-level view of how

notifications and other AWS services work with IAM, see [AWS services that work with IAM](#) in the *IAM User Guide*.

Topics

- [Identity-based policies in the developer tools console](#)
- [AWS CodeStar Notifications and AWS CodeConnections resource-based policies](#)
- [Authorization based on tags](#)
- [IAM roles](#)

Identity-based policies in the developer tools console

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. AWS CodeStar Notifications and AWS CodeConnections support specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see [IAM JSON policy elements reference](#) in the *IAM User Guide*.

Actions

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

Policy actions for notifications in the Developer Tools console use the following prefixes before the action: `codestar-notifications` and `codeconnections`. For example, to grant someone permission to view all notification rules in their account, you include the `codestar-notifications:ListNotificationRules` action in their policy. Policy statements must include either an `Action` or `NotAction` element. AWS CodeStar Notifications and AWS CodeConnections defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple AWS CodeStar Notifications actions in a single statement, separate them with commas as follows.

```
"Action": [
```

```
      "codestar-notifications:action1",
      "codestar-notifications:action2"
```

To specify multiple AWS CodeConnections actions in a single statement, separate them with commas as follows.

```
"Action": [
      "codeconnections:action1",
      "codeconnections:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word List, include the following action.

```
"Action": "codestar-notifications:List*"
```

AWS CodeStar Notifications API actions include:

- CreateNotificationRule
- DeleteNotificationRule
- DeleteTarget
- DescribeNotificationRule
- ListEventTypes
- ListNotificationRules
- ListTagsForResource
- ListTargets
- Subscribe
- TagResource
- Unsubscribe
- UntagResource
- UpdateNotificationRule

AWS CodeConnections API actions include the following:

- `CreateConnection`

- `DeleteConnection`

- `GetConnection`

- `ListConnections`

- `ListTagsForResource`

- `TagResource`

- `UntagResource`

The following permissions-only actions are required in AWS CodeConnections to complete the auth handshake:

- `GetIndividualAccessToken`

- `GetInstallationUrl`

- `ListInstallationTargets`

- `StartOAuthHandshake`

- `UpdateConnectionInstallation`

The following permissions-only action is required in AWS CodeConnections to use a connection:

- `UseConnection`

The following permissions-only action is required in AWS CodeConnections to pass a connection to a service:

- `PassConnection`

To see a list of AWS CodeStar Notifications and AWS CodeConnections actions, see Actions Defined by AWS CodeStar Notifications and Actions Defined by AWS CodeConnections in the *IAM User Guide*.

Resources

AWS CodeStar Notifications and AWS CodeConnections do not support specifying resource ARNs in a policy.

Condition keys

AWS CodeStar Notifications and AWS CodeConnections define their own sets of condition keys and also support using some global condition keys. To see all AWS global condition keys, see [AWS global condition context keys](#) in the *IAM User Guide*.

All AWS CodeStar Notifications actions support the `codestar-notifications:NotificationsForResource` condition key. For more information, see [Identity-based policy examples](#).

AWS CodeConnections define the following condition keys that can be used in the `Condition` element of an IAM policy. You can use these keys to further refine the conditions under which the policy statement applies. For more information, see [AWS CodeConnections permissions reference](#).

Condition keys	Description
`codeconnections:BranchName`	Filters access by the third-party repository branch name
`codeconnections:FullRepositoryId`	Filters access by the repository that is passed in the request. Applies only to `UseConnection` requests for access to a specific repository
`codeconnections:InstallationId`	Filters access by the third-party ID (such as the Bitbucket app installation ID) that is used to update a connection. Allows you to restrict which third-party app installations can be used to make a connection
`codeconnections:OwnerId`	Filters access by the owner or account ID of the third-party provider
`codeconnections:PassedToService`	Filters access by the service to which the principal is allowed to pass a connection
`codeconnections:ProviderAction`	Filters access by the provider action in a `UseConnection` request such as `ListRepositories`.

Condition keys	Description
codeconnections:ProviderPer missionsRequired	Filters access by the type of third-party provider permissions
codeconnections:ProviderType	Filters access by the type of third-party provider passed in the request
codeconnections:ProviderTyp eFilter	Filters access by the type of third-party provider used to filter results
codeconnections:RepositoryName	Filters access by the third-party repository name

Examples

To view examples of AWS CodeStar Notifications and AWS CodeConnections identity-based policies, see Identity-based policy examples.

AWS CodeStar Notifications and AWS CodeConnections resource-based policies

AWS CodeStar Notifications and AWS CodeConnections do not support resource-based policies.

Authorization based on tags

You can attach tags to AWS CodeStar Notifications and AWS CodeConnections resources or pass tags in a request. To control access based on tags, you provide tag information in the condition element of a policy using the codestar-notifications and codeconnections:ResourceTag/*key-name*, aws:RequestTag/*key-name*, or aws:TagKeys condition keys. For more information about tagging strategies, see Tagging AWS resources. For more information about tagging AWS CodeStar Notifications and AWS CodeConnections resources, see Tag connections resources.

To view example identity-based policies for limiting access to a resource based on the tags on that resource, see Using tags to control access to AWS CodeConnections resources.

IAM roles

An IAM role is an entity within your AWS account that has specific permissions.

Using temporary credentials

You can use temporary credentials to sign in with federation, and assume an IAM role or a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.

AWS CodeStar Notifications and AWS CodeConnections supports the use of temporary credentials.

Service-linked roles

Service-linked roles allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

AWS CodeStar Notifications supports service-linked roles. For details about creating or managing AWS CodeStar Notifications and AWS CodeConnections service-linked roles, see Using service-linked roles for AWS CodeStar Notifications.

CodeConnections does not support service-linked roles.

AWS CodeConnections permissions reference

The following tables list each AWS CodeConnections API operation, the corresponding actions for which you can grant permissions, and the format of the resource ARN to use for granting permissions. The AWS CodeConnections APIs are grouped into tables based on the scope of the actions allowed by that API. Refer to it when writing permissions policies that you can attach to an IAM identity (identity-based policies).

When you create a permissions policy, you specify the actions in the policy's `Action` field. You specify the resource value in the policy's `Resource` field as an ARN, with or without a wildcard character (*).

To express conditions in your connections policies, use the condition keys described here and listed in Condition keys. You can also use AWS-wide condition keys. For a complete list of AWS-wide keys, see Available keys in the *IAM User Guide*.

To specify an action, use the `codeconnections` prefix followed by the API operation name (for example, `codeconnections:ListConnections` or `codeconnections:CreateConnection`.

Using wildcards

To specify multiple actions or resources, use a wildcard character (*) in your ARN. For example, `codeconnections:*` specifies all AWS CodeConnections actions and `codeconnections:Get*` specifies all AWS CodeConnections actions that begin with the word `Get`. The following example grants access to all resources with names that begin with `MyConnection`.

```
arn:aws:codeconnections:us-west-2:account-ID:connection/*
```

You can use wildcards only with the `connection` resources listed in the following table. You can't use wildcards with `region` or `account-id` resources. For more information about wildcards, see [IAM identifiers](#) in *IAM User Guide*.

Topics

- [Permissions for managing connections](#)

- [Permissions for managing hosts](#)

- [Permissions for completing connections](#)

- [Permissions for setting up hosts](#)

- [Passing a connection to a service](#)

- [Using a connection](#)

- [Supported access types for ProviderAction](#)

- [Supported permissions for tagging connection resources](#)

- [Passing a connection to a repository link](#)

- [Supported condition key for repository links](#)

- [Supported permissions for connection sharing](#)

Permissions for managing connections

A role or user designated to use the AWS CLI or SDK to view, create, or delete connections should have permissions limited to the following.

> **ⓘ Note**
>
> You cannot complete or use a connection in the console with only the following permissions. You need to add the permissions in [Permissions for completing connections](#).

```
codeconnections:CreateConnection
codeconnections:DeleteConnection
codeconnections:GetConnection
codeconnections:ListConnections
```

AWS CodeStar Notifications and AWS CodeConnections required permissions for actions for managing connections

CreateConnection

Action(s): `codeconnections:CreateConnection`

Required to use the CLI or console to create a connection.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

DeleteConnection

Action(s): `codeconnections:DeleteConnection`

Required to use the CLI or console to delete a connection.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

GetConnection

Action(s): `codeconnections:GetConnection`

Required to use the CLI or console to view details about a connection.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

ListConnections

Action(s): `codeconnections:ListConnections`

Required to use the CLI or console to list all connections in the account.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

These operations support the following condition keys:

Action	Condition keys
codeconnections:CreateConnection	codeconnections:ProviderType
codeconnections:DeleteConnection	N/A
codeconnections:GetConnection	N/A
codeconnections:ListConnections	codeconnections:ProviderTyp eFilter

Permissions for managing hosts

A role or user designated to use the AWS CLI or SDK to view, create, or delete hosts should have permissions limited to the following.

> ⓘ **Note**
>
> You cannot complete or use a connection in the host with only the following permissions. You need to add the permissions in Permissions for setting up hosts.

```
codeconnections:CreateHost
codeconnections:DeleteHost
codeconnections:GetHost
codeconnections:ListHosts
```

AWS CodeStar Notifications and AWS CodeConnections required permissions for actions for managing hosts

CreateHost

Action(s): codeconnections:CreateHost

Required to use the CLI or console to create a host.

Resource:arn:aws:codeconnections:*region*:*account-id*:host/*host-id*

DeleteHost

Action(s): `codeconnections:DeleteHost`

Required to use the CLI or console to delete a host.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:host/`*`host-id`*

GetHost

Action(s): `codeconnections:GetHost`

Required to use the CLI or console to view details about a host.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:host/`*`host-id`*

ListHosts

Action(s): `codeconnections:ListHosts`

Required to use the CLI or console to list all hosts in the account.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:host/`*`host-id`*

These operations support the following condition keys:

Action	Condition keys
`codeconnections:CreateHost`	`codeconnections:ProviderType` `codeconnections:VpcId`
`codeconnections:DeleteHost`	N/A
`codeconnections:GetHost`	N/A
`codeconnections:ListHosts`	`codeconnections:ProviderTyp eFilter`

For an example policy that uses the **VpcId** condition key, see Example: Limit host VPC permissions using the **VpcId** context key.

Permissions for completing connections

A role or user designated to manage connections in the console should have the permissions required to complete a connection in the console and create an installation, which includes authorizing the handshake to the provider and creating installations for connections to use. Use the following permissions in addition to the permissions above.

The following IAM operations are used by the console when performing a browser-based handshake. The `ListInstallationTargets`, `GetInstallationUrl`, `StartOAuthHandshake`, `UpdateConnectionInstallation`, and `GetIndividualAccessToken` are IAM policy permissions. They are not API actions.

```
codeconnections:GetIndividualAccessToken
codeconnections:GetInstallationUrl
codeconnections:ListInstallationTargets
codeconnections:StartOAuthHandshake
codeconnections:UpdateConnectionInstallation
```

Based on this, the following permissions are needed to use, create, update, or delete a connection in the console.

```
codeconnections:CreateConnection
codeconnections:DeleteConnection
codeconnections:GetConnection
codeconnections:ListConnections
codeconnections:UseConnection
codeconnections:ListInstallationTargets
codeconnections:GetInstallationUrl
codeconnections:StartOAuthHandshake
codeconnections:UpdateConnectionInstallation
codeconnections:GetIndividualAccessToken
```

AWS CodeConnections required permissions for actions for completing connections

GetIndividualAccessToken

> **Action(s):** `codeconnections:GetIndividualAccessToken`
>
> Required to use the console to complete a connection. This is an IAM policy permission only, not an API action.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

GetInstallationUrl

Action(s): `codeconnections:GetInstallationUrl`

Required to use the console to complete a connection. This is an IAM policy permission only, not an API action.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

ListInstallationTargets

Action(s): `codeconnections:ListInstallationTargets`

Required to use the console to complete a connection. This is an IAM policy permission only, not an API action.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

StartOAuthHandshake

Action(s): `codeconnections:StartOAuthHandshake`

Required to use the console to complete a connection. This is an IAM policy permission only, not an API action.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

UpdateConnectionInstallation

Action(s): `codeconnections:UpdateConnectionInstallation`

Required to use the console to complete a connection. This is an IAM policy permission only, not an API action.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

These operations support the following condition keys.

Action	Condition keys
`codeconnections:GetIndividu` `alAccessToken`	`codeconnections:ProviderType`
`codeconnections:GetInstalla` `tionUrl`	`codeconnections:ProviderType`
`codeconnections:ListInstall` `ationTargets`	N/A
`codeconnections:StartOAuthH` `andshake`	`codeconnections:ProviderType`
`codeconnections:UpdateConne` `ctionInstallation`	`codeconnections:InstallationId`

Permissions for setting up hosts

A role or user designated to manage connections in the console should have the permissions required to set up a host in the console, which includes authorizing the handshake to the provider and installing the host app. Use the following permissions in addition to the permissions for hosts above.

The following IAM operations are used by the console when performing a browser-based host registration. `RegisterAppCode` and `StartAppRegistrationHandshake` are IAM policy permissions. They are not API actions.

```
codeconnections:RegisterAppCode
codeconnections:StartAppRegistrationHandshake
```

Based on this, the following permissions are needed to use, create, update, or delete a connection in the console that requires a host (such as installed provider types).

```
codeconnections:CreateConnection
```

```
codeconnections:DeleteConnection
codeconnections:GetConnection
codeconnections:ListConnections
codeconnections:UseConnection
codeconnections:ListInstallationTargets
codeconnections:GetInstallationUrl
codeconnections:StartOAuthHandshake
codeconnections:UpdateConnectionInstallation
codeconnections:GetIndividualAccessToken
codeconnections:RegisterAppCode
codeconnections:StartAppRegistrationHandshake
```

AWS CodeConnections required permissions for actions for completing host setup

RegisterAppCode

Action(s): `codeconnections:RegisterAppCode`

Required to use the console to complete host setup. This is an IAM policy permission only, not an API action.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:host/`*`host-id`*

StartAppRegistrationHandshake

Action(s): `codeconnections:StartAppRegistrationHandshake`

Required to use the console to complete host setup. This is an IAM policy permission only, not an API action.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:host/`*`host-id`*

These operations support the following condition keys.

Passing a connection to a service

When a connection is passed to a service (for example, when a connection ARN is provided in a pipeline definition to create or update a pipeline) the user must have the `codeconnections:PassConnection` permission.

AWS CodeConnections required permissions for passing a connection

PassConnection

Action(s): `codeconnections:PassConnection`

Required to pass a connection to a service.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

This operation also supports the following condition key:

- `codeconnections:PassedToService`

Supported values for condition keys

Key	Valid action providers
`codeconnections:PassedToService`	• `codeguru-reviewer` • `codepipeline.amazonaws.com` • `proton.amazonaws.com`

Using a connection

When a service like CodePipeline uses a connection, the service role must have the `codeconnections:UseConnection` permission for a given connection.

To manage connections in the console, the user policy must have the `codeconnections:UseConnection` permission.

AWS CodeConnections required action for using a connection

UseConnection

Action(s): `codeconnections:UseConnection`

Required to use a connection.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

This operation also supports the following condition keys:

- `codeconnections:BranchName`

- `codeconnections:FullRepositoryId`

- `codeconnections:OwnerId`

- `codeconnections:ProviderAction`

- `codeconnections:ProviderPermissionsRequired`

- `codeconnections:RepositoryName`

Supported values for condition keys

Key	Valid action providers
`codeconnections:FullRepositoryId`	The user name and repository name of a repository, such as `my-owner/my-reposi tory` . Supported only when the connection is being used to access a specific repository.
`codeconnections:ProviderPer missionsRequired`	read_only or read_write
`codeconnections:ProviderAction`	`GetBranch` , `ListRepositories` , `ListOwners` , `ListBranches` , `StartUplo adArchiveToS3` , `GitPush`, `GitPull`, `GetUploadArchiveToS3Status` , `CreatePullRequestDiffCommen t` , `GetPullRequest` , `ListBranc hCommits` , `ListCommitFiles` , `ListPullRequestComments` , `ListPullR equestCommits` . For information, see the next section.

The required condition keys for some functionality might change over time. We recommend that you use `codeconnections:UseConnection` to control access to a connection unless your access control requirements require different permissions.

Supported access types for `ProviderAction`

When a connection is used by an AWS service, it results in API calls being made to your source code provider. For example, a service might list repositories for a Bitbucket connection by calling the `https://api.bitbucket.org/2.0/repositories/`*`username`* API.

The `ProviderAction` condition key allows you to restrict which APIs on a provider can be called. Because the API path might be generated dynamically, and the path varies from provider to provider, the `ProviderAction` value is mapped to an abstract action name rather than the URL of the API. This allows you to write policies that have the same effect regardless of the provider type for the connection.

The following are the access types that are granted for each of the supported `ProviderAction` values. The following are IAM policy permissions. They are not API actions.

AWS CodeConnections supported access types for `ProviderAction`

GetBranch

> **Action(s):** `codeconnections:GetBranch`
>
> Required to access information about a branch, such as the latest commit for that branch.
>
> **Resource:**`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

ListRepositories

> **Action(s):** `codeconnections:ListRepositories`
>
> Required to access a list of public and private repositories, including details about those repositories, that belong to an owner.
>
> **Resource:**`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

ListOwners

> **Action(s):** `codeconnections:ListOwners`
>
> Required to access a list of owners that the connection has access to.
>
> **Resource:**`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

ListBranches

Action(s): `codeconnections:ListBranches`

Required to access the list of branches that exist on a given repository.

Resource: `arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

StartUploadArchiveToS3

Action(s): `codeconnections:StartUploadArchiveToS3`

Required to read source code and upload it to Amazon S3.

Resource: `arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

GitPush

Action(s): `codeconnections:GitPush`

Required to write to a repository using Git.

Resource: `arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

GitPull

Action(s): `codeconnections:GitPull`

Required to read from a repository using Git.

Resource: `arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

GetUploadArchiveToS3Status

Action(s): `codeconnections:GetUploadArchiveToS3Status`

Required to access the status of an upload, including any error messages, started by `StartUploadArchiveToS3`.

Resource: `arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

CreatePullRequestDiffComment

Action(s): `codeconnections:CreatePullRequestDiffComment`

Required to access comments on a pull request.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

GetPullRequest

Action(s): `codeconnections:GetPullRequest`

Required to view pull requests for a repository.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

ListBranchCommits

Action(s): `codeconnections:ListBranchCommits`

Required to view a list of commits for a repository branch.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

ListCommitFiles

Action(s): `codeconnections:ListCommitFiles`

Required to view a list of files for a commit.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

ListPullRequestComments

Action(s): `codeconnections:ListPullRequestComments`

Required to view a list of comments for a pull request.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

ListPullRequestCommits

Action(s): `codeconnections:ListPullRequestCommits`

Required to view a list of commits for a pull request.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

Supported permissions for tagging connection resources

The following IAM operations are used when tagging connection resources.

```
codeconnections:ListTagsForResource
codeconnections:TagResource
codeconnections:UntagResource
```

AWS CodeConnections required actions for tagging connection resources

ListTagsForResource

Action(s): `codeconnections:ListTagsForResource`

Required to view a list of tags associated with the connection resource.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*`,` `arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:host/`*`host-id`*

TagResource

Action(s): `codeconnections:TagResource`

Required to tag a connection resource.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*`,` `arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:host/`*`host-id`*

UntagResource

Action(s): `codeconnections:UntagResource`

Required to remove tags from a connection resource.

Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-`*
`id``,  arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:host/`*`host-id`*

Passing a connection to a repository link

When a repository-link is provided in a sync configuration, the user must have the
`codeconnections:PassRepository` permission for the repository-link ARN/resource.

AWS CodeConnections required permissions for passing a connection

PassRepository

    Action(s): `codeconnections:PassRepository`

    Required to pass a repository-link to a sync configuration.

    Resource:`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:repository-`
    `link/`*`repository-link-id`*

This operation also supports the following condition key:

- `codeconnections:PassedToService`

Supported values for condition keys

Key	Valid action providers
`codeconnections:PassedToService`	- `cloudformation.sync.codecon` `nections.amazonaws.com`

Supported condition key for repository links

Operations for repository links and sync configuration resources are supported by the following
condition key:

- `codeconnections:Branch`

    Filters access by the branch name that is passed in the request.

Supported actions for condition key

Key	Valid values
`codeconnections:Branch`	The following actions are supported for this condition key: • CreateSyncConfiguration • UpdateSyncConfiguration • GetRepositorySyncStatus

Supported permissions for connection sharing

The following IAM operations are used when sharing connections.

```
codeconnections:GetResourcePolicy
```

AWS CodeConnections required actions for sharing connections

GetResourcePolicy

> **Action(s):** `codeconnections:GetResourcePolicy`
>
> Required to access information about the resource policy.
>
> **Resource:**`arn:aws:codeconnections:`*`region`*`:`*`account-id`*`:connection/`*`connection-id`*

For more information on connection sharing, see [Share connections with AWS accounts](#).

Identity-based policy examples

By default, IAM users and roles who have one of the managed policies for AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, or AWS CodePipeline applied have permissions to connections, notifications, and notification rules that align with the intent of those policies. For example, IAM users or roles that have one of the full access policies (**AWSCodeCommitFullAccess**, **AWSCodeBuildAdminAccess**, **AWSCodeDeployFullAccess**, or **AWSCodePipeline_FullAccess**)

applied to them also have full access to notifications and notification rules created for the resources for those services.

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

Permissions and examples for AWS CodeStar Notifications

The following policy statements and examples can help you manage AWS CodeStar Notifications.

Permissions related to notifications in full access managed policies

The **AWSCodeCommitFullAccess**, **AWSCodeBuildAdminAccess**, **AWSCodeDeployFullAccess**, and **AWSCodePipeline_FullAccess** managed policies include the following statements to allow full access to notifications in the Developer Tools console. Users with one of these managed policies applied can also create and manage Amazon SNS topics for notifications, subscribe and unsubscribe users to topics, and list topics to choose as targets for notification rules.

> ### ⓘ Note
>
> In the managed policy, the condition key `codestar-notifications:NotificationsForResource` will have a value specific to the resource type for the service. For example, in the full access policy for CodeCommit, the value is `arn:aws:codecommit:*`.

```
{
    "Sid": "CodeStarNotificationsReadWriteAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications:DescribeNotificationRule",
        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications:DeleteNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe"
    ],
    "Resource": "*",
```

```
        "Condition" : {
            "StringLike" : {"codestar-notifications:NotificationsForResource" :
    "arn:aws:<vendor-code>:*"}
        }
    },
    {
        "Sid": "CodeStarNotificationsListAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:ListNotificationRules",
            "codestar-notifications:ListTargets",
            "codestar-notifications:ListTagsforResource",
            "codestar-notifications:ListEventTypes"
        ],
        "Resource": "*"
    },
    {

        "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
        "Effect": "Allow",
        "Action": [
            "sns:CreateTopic",
            "sns:SetTopicAttributes"
        ],
        "Resource": "arn:aws:sns:*:*:codestar-notifications*"
    },
    {
        "Sid": "SNSTopicListAccess",
        "Effect": "Allow",
        "Action": [
            "sns:ListTopics"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CodeStarNotificationsChatbotAccess",
        "Effect": "Allow",
        "Action": [
            "chatbot:DescribeSlackChannelConfigurations",
            "chatbot:ListMicrosoftTeamsChannelConfigurations"
          ],
      "Resource": "*"
    }
```

Permissions related to notifications in read-only managed policies

The **AWSCodeCommitReadOnlyAccess**, **AWSCodeBuildReadOnlyAccess**, **AWSCodeDeployReadOnlyAccess**, and **AWSCodePipeline_ReadOnlyAccess** managed policies include the following statements to allow read-only access to notifications. For example, they can view notifications for resources in the Developer Tools console, but cannot create, manage, or subscribe to them.

> ⓘ **Note**
>
> In the managed policy, the condition key `codestar-notifications:NotificationsForResource` will have a value specific to the resource type for the service. For example, in the full access policy for CodeCommit, the value is `arn:aws:codecommit:*`.

```
{
    "Sid": "CodeStarNotificationsPowerUserAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:DescribeNotificationRule"
    ],
    "Resource": "*",
    "Condition" : {
        "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:<vendor-code>:*"}
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListEventTypes",
        "codestar-notifications:ListTargets"
    ],
    "Resource": "*"
}
```

Permissions related to notifications in other managed policies

The **AWSCodeCommitPowerUser**, **AWSCodeBuildDeveloperAccess**, and
AWSCodeBuildDeveloperAccess managed policies include the following statements to allow
developers with one of these managed policies applied to create, edit, and subscribe to
notifications. They cannot delete notification rules or manage tags for resources.

> ⓘ **Note**
>
> In the managed policy, the condition key `codestar-`
> `notifications:NotificationsForResource` will have a value specific to the resource
> type for the service. For example, in the full access policy for CodeCommit, the value is
> `arn:aws:codecommit:*`.

```
    {
        "Sid": "CodeStarNotificationsReadWriteAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:CreateNotificationRule",
            "codestar-notifications:DescribeNotificationRule",
            "codestar-notifications:UpdateNotificationRule",
            "codestar-notifications:Subscribe",
            "codestar-notifications:Unsubscribe"
        ],
        "Resource": "*",
        "Condition" : {
            "StringLike" : {"codestar-notifications:NotificationsForResource" :
 "arn:aws:<vendor-code>:*"}
        }
    },
    {
        "Sid": "CodeStarNotificationsListAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:ListNotificationRules",
            "codestar-notifications:ListTargets",
            "codestar-notifications:ListTagsforResource",
            "codestar-notifications:ListEventTypes"
        ],
        "Resource": "*"
    },
```

```
{
    "Sid": "SNSTopicListAccess",
    "Effect": "Allow",
    "Action": [
        "sns:ListTopics"
    ],
    "Resource": "*"
},
{
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
}
```

Example: An administrator-level policy for managing AWS CodeStar Notifications

In this example, you want to grant an IAM user in your AWS account full access to AWS CodeStar Notifications so that the user can review details of notification rules and list notification rules, targets, and event types. You also want to allow the user to add, update, and delete notification rules. This is a full access policy, equivalent to the notification permissions included as part of the **AWSCodeBuildAdminAccess**, **AWSCodeCommitFullAccess**, **AWSCodeDeployFullAccess**, and **AWSCodePipeline_FullAccess** managed policies. Like those managed policies, you should only attach this kind of policy statement to IAM users, groups, or roles that require full administrative access to notifications and notification rules across your AWS account.

> ### ⓘ Note
>
> This policy contains allows `CreateNotificationRule`. Any user with this policy applied to their IAM user or role will be able to create notification rules for any and all resource types supported by AWS CodeStar Notifications in the AWS account, even if that user does not have access to those resources themselves. For example, a user with this policy could create a notification rule for a CodeCommit repository without having permissions to access CodeCommit itself.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
        "Sid": "AWSCodeStarNotificationsFullAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:CreateNotificationRule",
            "codestar-notifications:DeleteNotificationRule",
            "codestar-notifications:DescribeNotificationRule",
            "codestar-notifications:ListNotificationRules",
            "codestar-notifications:UpdateNotificationRule",
            "codestar-notifications:Subscribe",
            "codestar-notifications:Unsubscribe",
            "codestar-notifications:DeleteTarget",
            "codestar-notifications:ListTargets",
            "codestar-notifications:ListTagsforResource",
            "codestar-notifications:TagResource",
            "codestar-notifications:UntagResource"
        ],
        "Resource": "*"
      }
    ]
}
```

Example: A contributor-level policy for using AWS CodeStar Notifications

In this example, you want to grant access to the day-to-day usage of AWS CodeStar
Notifications, such as creating and subscribing to notifications, but not to more destructive
actions, such as deleting notification rules or targets. This is the equivalent to the access
provided in the **AWSCodeBuildDeveloperAccess**, **AWSCodeDeployDeveloperAccess**, and
AWSCodeCommitPowerUser managed policies.

> ⓘ **Note**
>
> This policy contains allows `CreateNotificationRule`. Any user with this policy applied
> to their IAM user or role will be able to create notification rules for any and all resource
> types supported by AWS CodeStar Notifications in the AWS account, even if that user does

not have access to those resources themselves. For example, a user with this policy could create a notification rule for a CodeCommit repository without having permissions to access CodeCommit itself.

```
{
    "Version": "2012-10-17",
    "Sid": "AWSCodeStarNotificationsPowerUserAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:CreateNotificationRule",
            "codestar-notifications:DescribeNotificationRule",
            "codestar-notifications:ListNotificationRules",
            "codestar-notifications:UpdateNotificationRule",
            "codestar-notifications:Subscribe",
            "codestar-notifications:Unsubscribe",
            "codestar-notifications:ListTargets",
            "codestar-notifications:ListTagsforResource"
        ],
        "Resource": "*"
        }
    ]
}
```

Example: A read-only-level policy for using AWS CodeStar Notifications

In this example, you want to grant an IAM user in your account read-only access to the notification rules, targets, and event types in your AWS account. This example shows how you might create a policy that allows viewing these items. This is the equivalent to the permissions included as part of the **AWSCodeBuildReadOnlyAccess**, **AWSCodeCommitReadOnly**, and **AWSCodePipeline_ReadOnlyAccess** managed policies.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "CodeNotificationforReadOnly",
    "Statement": [
        {
            "Sid": "ReadsAccess",
```

```
            "Effect": "Allow",
            "Action": [
                "codestar-notifications:DescribeNotificationRule",
                "codestar-notifications:ListNotificationRules",
                "codestar-notifications:ListTargets",
                "codestar-notifications:ListEventTypes"
            ],
            "Resource": "*"
        }
    ]
}
```

Permissions and examples for AWS CodeConnections

The following policy statements and examples can help you manage AWS CodeConnections.

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

Example: A policy for creating AWS CodeConnections with the CLI and viewing with the console

A role or user designated to use the AWS CLI or SDK to view, create, tag, or delete connections should have permissions limited to the following.

> (i) **Note**
>
> You cannot complete a connection in the console with only the following permissions. You need to add the permissions in the next section.

To use the console to view a list of available connections, view tags, and use a connection, use the following policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
        "Sid": "ConnectionsFullAccess",
        "Effect": "Allow",
```

```
        "Action": [
            "codeconnections:CreateConnection",
            "codeconnections:DeleteConnection",
            "codeconnections:UseConnection",
            "codeconnections:GetConnection",
            "codeconnections:ListConnections",
            "codeconnections:TagResource",
            "codeconnections:ListTagsForResource",
            "codeconnections:UntagResource"
        ],
        "Resource": "*"
    }
  ]
}
```

Example: A policy for creating AWS CodeConnections with the console

A role or user designated to manage connections in the console should have the permissions required to complete a connection in the console and create an installation, which includes authorizing the handshake to the provider and creating installations for connections to use. `UseConnection` should also be added to use the connection in the console. Use the following policy to view, use, create, tag, or delete a connection in the console.

> ⓘ **Note**
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

> ⓘ **Note**
>
> For resources created using the console, policy statement actions must include `codestar-connections` as the service prefix as shown in the following example.

JSON

```
{
    "Version":"2012-10-17",
```

```
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "codestar-connections:CreateConnection",
                    "codestar-connections:DeleteConnection",
                    "codestar-connections:GetConnection",
                    "codestar-connections:ListConnections",
                    "codestar-connections:GetInstallationUrl",
                    "codestar-connections:GetIndividualAccessToken",
                    "codestar-connections:ListInstallationTargets",
                    "codestar-connections:StartOAuthHandshake",
                    "codestar-connections:UpdateConnectionInstallation",
                    "codestar-connections:UseConnection",
                    "codestar-connections:TagResource",
                    "codestar-connections:ListTagsForResource",
                    "codestar-connections:UntagResource"
                ],
                "Resource": [
                    "*"
                ]
            }
        ]
}
```

Example: An administrator-level policy for managing AWS CodeConnections

In this example, you want to grant an IAM user in your AWS account full access to CodeConnections so that the user can add, update, and delete connections. This is a full access policy, equivalent to the **AWSCodePipeline_FullAccess** managed policy. Like that managed policy, you should only attach this kind of policy statement to IAM users, groups, or roles that require full administrative access to connections across your AWS account.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
        "Sid": "ConnectionsFullAccess",
        "Effect": "Allow",
```

```
            "Action": [
                "codeconnections:CreateConnection",
                "codeconnections:DeleteConnection",
                "codeconnections:UseConnection",
                "codeconnections:GetConnection",
                "codeconnections:ListConnections",
                "codeconnections:ListInstallationTargets",
                "codeconnections:GetInstallationUrl",
                "codeconnections:StartOAuthHandshake",
                "codeconnections:UpdateConnectionInstallation",
                "codeconnections:GetIndividualAccessToken",
                "codeconnections:TagResource",
                "codeconnections:ListTagsForResource",
                "codeconnections:UntagResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Example: A contributor-level policy for using AWS CodeConnections

In this example, you want to grant access to the day-to-day usage of CodeConnections, such as creating and viewing details of connections, but not to more destructive actions, such as deleting connections.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCodeConnectionsPowerUserAccess",
            "Effect": "Allow",
            "Action": [
                "codeconnections:CreateConnection",
                "codeconnections:UseConnection",
                "codeconnections:GetConnection",
                "codeconnections:ListConnections",
                "codeconnections:ListInstallationTargets",
                "codeconnections:GetInstallationUrl",
```

```
                "codeconnections:GetIndividualAccessToken",
                "codeconnections:StartOAuthHandshake",
                "codeconnections:UpdateConnectionInstallation",
                "codeconnections:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Example: A read-only-level policy for using AWS CodeConnections

In this example, you want to grant an IAM user in your account read-only access to the connections in your AWS account. This example shows how you might create a policy that allows viewing these items.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "ConnectionsforReadOnly",
    "Statement": [
        {
            "Sid": "ReadsAPIAccess",
            "Effect": "Allow",
            "Action": [
            "codeconnections:GetConnection",
            "codeconnections:ListConnections",
            "codeconnections:ListInstallationTargets",
            "codeconnections:GetInstallationUrl",
            "codeconnections:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Example: Limit host VPC permissions using the VpcId context key

In the following example, the customer can use the **VpcId** context key to limit creation or management of hosts to hosts with specified VPC.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "codeconnections:CreateHost",
                "codeconnections:UpdateHost"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "codeconnections:VpcId": "vpc-EXAMPLE"
                }
            }
        }
    ]
}
```

Using tags to control access to AWS CodeConnections resources

Tags can be attached to the resource or passed in the request to services that support tagging. In AWS CodeConnections, resources can have tags, and some actions can include tags. When you create an IAM policy, you can use tag condition keys to control the following:

- Which users can perform actions on a pipeline resource, based on tags that it already has.

- Which tags can be passed in an action's request.

- Whether specific tag keys can be used in a request.

The following examples demonstrate how to specify tag conditions in policies for AWS CodeConnections users.

Example 1: Allow actions based on tags in the request

The following policy grants users permission to create connections in AWS CodeConnections.

To do that, it allows the `CreateConnection` and `TagResource` actions if the request specifies a tag named `Project` with the value `ProjectA`. (The `aws:RequestTag` condition key is used to control which tags can be passed in an IAM request.) The `aws:TagKeys` condition ensures tag key case sensitivity.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codeconnections:CreateConnection",
        "codeconnections:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Project": "ProjectA"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["Project"]
        }
      }
    }
  ]
}
```

Example 2: Allow actions based on resource tags

The following policy grants users permission to perform actions on, and get information about, resources in AWS CodeConnections.

To do that, it allows specific actions if the pipeline has a tag named `Project` with the value `ProjectA`. (The `aws:RequestTag` condition key is used to control which tags can be passed in an IAM request.) The `aws:TagKeys` condition ensures tag key case sensitivity.

JSON

```json
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codeconnections:CreateConnection",
        "codeconnections:DeleteConnection",
        "codeconnections:ListConnections"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "ProjectA"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["Project"]
        }
      }
    }
  ]
}
```

Using notifications and connections in the console

The notifications experience is built into the CodeBuild, CodeCommit, CodeDeploy, and CodePipeline consoles, as well as in the Developer Tools console in the **Settings** navigation bar itself. To access notifications in the consoles, you must either have one of the managed policies for those services applied, or you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS CodeStar Notifications and AWS CodeConnections resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (IAM users or roles) with that policy. For more information about granting access to AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, and AWS CodePipeline, including access to those consoles, see the following topics:

- CodeBuild: Using identity-based policies for CodeBuild

- CodeCommit: [Using identity-based policies for CodeCommit](#)
- AWS CodeDeploy: [Identity and access management for AWS CodeDeploy](#)
- CodePipeline: [Access control with IAM policies](#)

AWS CodeStar Notifications does not have any AWS managed policies. To provide access to notification functionality, you must either apply one of the managed policies for one of the services listed previously, or you must create policies with the level of permission you want to grant to users or entities, and then attach those policies to the users, groups, or roles that require those permissions. For more information and examples, see the following:

- [Example: An administrator-level policy for managing AWS CodeStar Notifications](#)
- [Example: A contributor-level policy for using AWS CodeStar Notifications](#)
- [Example: A read-only-level policy for using AWS CodeStar Notifications](#).

AWS CodeConnections does not have any AWS managed policies. You use the permissions and combinations of permissions for access, such as the permissions detailed in [Permissions for completing connections](#).

For more information, see the following:

- [Example: An administrator-level policy for managing AWS CodeConnections](#)
- [Example: A contributor-level policy for using AWS CodeConnections](#)
- [Example: A read-only-level policy for using AWS CodeConnections](#)

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

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": "*"
        }
    ]
}
```

Troubleshooting AWS CodeStar Notifications and AWS CodeConnections identity and access

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

Topics

- [I'm an administrator and want to allow others to access notifications](#)
- [I created an Amazon SNS topic and added it as a notification rule target, but I am not receiving emails about events](#)

- I want to allow people outside of my AWS account to access my AWS CodeStar Notifications and AWS CodeConnections resources

I'm an administrator and want to allow others to access notifications

To allow others to access AWS CodeStar Notifications and AWS CodeConnections, you must grant permission to the people or applications that need access. If you are using AWS IAM Identity Center to manage people and applications, you assign permission sets to users or groups to define their level of access. Permission sets automatically create and assign IAM policies to IAM roles that are associated with the person or application. For more information, see Permission sets in the *AWS IAM Identity Center User Guide*.

If you are not using IAM Identity Center, you must create IAM entities (users or roles) for the people or applications that need access. You must then attach a policy to the entity that grants them the correct permissions in AWS CodeStar Notifications and AWS CodeConnections. After the permissions are granted, provide the credentials to the user or application developer. They will use those credentials to access AWS. To learn more about creating IAM users, groups, policies, and permissions, see IAM Identities and Policies and permissions in IAM in the *IAM User Guide*.

For AWS CodeStar Notifications specific information, see Permissions and examples for AWS CodeStar Notifications.

I created an Amazon SNS topic and added it as a notification rule target, but I am not receiving emails about events

In order to receive notifications about events, you must have a valid Amazon SNS topic subscribed as a target for the notification rule, and your email address must be subscribed to the Amazon SNS topic. To troubleshoot problems with the Amazon SNS topic, check the following:

- Make sure that the Amazon SNS topic is in the same AWS Region as the notification rule.
- Check to make sure that your email alias is subscribed to the correct topic, and that you have confirmed the subscription. For more information, see Subscribing an endpoint to an Amazon SNS topic.
- Verify that the topic policy has been modified to allow AWS CodeStar Notifications to push notifications to that topic. The topic policy should include a statement similar to the following:

```
{
    "Sid": "AWSCodeStarNotifications_publish",
```

```
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "codestar-notifications.amazonaws.com"
            ]
        },
        "Action": "SNS:Publish",
        "Resource": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopicName",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "123456789012"
            }
        }
    }
}
```

For more information, see [Setting up](#).

I want to allow people outside of my AWS account to access my AWS CodeStar Notifications and AWS CodeConnections 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 AWS CodeStar Notifications and AWS CodeConnections supports these features, see [How features in the developer tools console work 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*.

Using service-linked roles for AWS CodeStar Notifications

AWS CodeStar Notifications 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 AWS CodeStar Notifications. Service-linked roles are predefined by AWS CodeStar Notifications and include all the permissions that the service requires to call other AWS services on your behalf. This role is created for you the first time you create a notification rule. You don't have to create the role.

A service-linked role makes setting up AWS CodeStar Notifications easier because you don't have to add permissions manually. AWS CodeStar Notifications defines the permissions of its service-linked roles, and unless defined otherwise, only AWS CodeStar Notifications 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.

To delete a service-linked role, you must first delete its related resources. This protects your AWS CodeStar Notifications resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM.

Service-linked role permissions for AWS CodeStar Notifications

AWS CodeStar Notifications uses the AWSServiceRoleForCodeStarNotifications service-linked role to retrieve information about events that occur in your toolchain and send notifications to the targets you specify.

The AWSServiceRoleForCodeStarNotifications service-linked role trusts the following services to assume the role:

- `codestar-notifications.amazonaws.com`

The role permissions policy allows AWS CodeStar Notifications to complete the following actions on the specified resources:

- Action: `PutRule` on `CloudWatch Event rules that are named awscodestar-notifications-*`
- Action: `DescribeRule` on `CloudWatch Event rules that are named awscodestar-notifications-*`

- Action: `PutTargets` on CloudWatch Event rules that are named `awscodestar-notifications-*`

- Action: `CreateTopic` to create Amazon SNS topics for use with AWS CodeStar Notifications with the prefix `CodeStarNotifications-`

- Action: `GetCommentsForPullRequests` on all comments on all pull requests in all CodeCommit repositories in the AWS account

- Action: `GetCommentsForComparedCommit` on all comments on all commits in all CodeCommit repositories in the AWS account

- Action: `GetDifferences` on all commits in all CodeCommit repositories in the AWS account

- Action: `GetCommentsForComparedCommit` on all comments on all commits in all CodeCommit repositories in the AWS account

- Action: `GetDifferences` on all commits in all CodeCommit repositories in the AWS account

- Action: `DescribeSlackChannelConfigurations` on all AWS Chatbot clients in the AWS account

- Action: `UpdateSlackChannelConfiguration` on all AWS Chatbot clients in the AWS account

- Action: `ListActionExecutions` on all actions in all pipelines in the AWS account

- Action: `GetFile` on all files in all CodeCommit repositories in the AWS account unless otherwise tagged

You can see these actions in the policy statement for the AWSServiceRoleForCodeStarNotifications service-linked role.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "events:PutTargets",
                "events:PutRule",
```

```
                "events:DescribeRule"
            ],
            "Resource": "arn:aws:events:*:*:rule/awscodestarnotifications-*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "sns:CreateTopic"
            ],
            "Resource": "arn:aws:sns:*:*:CodeStarNotifications-*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "codecommit:GetCommentsForPullRequest",
                "codecommit:GetCommentsForComparedCommit",
                "codecommit:GetDifferences",
                "chatbot:DescribeSlackChannelConfigurations",
                "chatbot:UpdateSlackChannelConfiguration",
                "codepipeline:ListActionExecutions"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "codecommit:GetFile"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:ResourceTag/ExcludeFileContentFromNotifications": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a service-linked role for AWS CodeStar Notifications

You don't need to manually create a service-linked role. You can use the Developer Tools console or the CreateNotificationRule API from the AWS CLI or SDKs to create a notification rule. You can also directly call the API. No matter which method you use, the service-linked role is created for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. You can use the Developer Tools console or the CreateNotificationRule API from the AWS CLI or SDKs to create a notification rule. You can also directly call the API. No matter which method you use, the service-linked role is created for you.

Editing a service-linked role for AWS CodeStar Notifications

After you create a service-linked role, you cannot change its name because various entities might reference the role. However, you can use IAM to edit the role description. For more information, see Editing a Service-Linked Role in the *IAM User Guide.*

Deleting a service-linked role for AWS CodeStar Notifications

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete the role. That way, you don't have an unused entity that is not actively monitored or maintained. You must clean up the resources for your service-linked role before you can delete it. For AWS CodeStar Notifications, this means deleting all notification rules that use the service role in your AWS account.

> **ⓘ Note**
>
> If the AWS CodeStar Notifications service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete AWS CodeStar Notifications resources used by AWSServiceRoleForCodeStarNotifications

1. Open the AWS Developer Tools console at https://console.aws.amazon.com/codesuite/settings/notifications.

> ⓘ **Note**
>
> Notification rules apply to the AWS Region where they are created. If you have notification rules in more than one AWS Region, use the Region selector to change the AWS Region.

2. Choose all notification rules that appear in the list, and then choose **Delete**.

3. Repeat these steps in all AWS Regions where you created notification rules.

To use IAM to delete the service-linked role

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

Supported regions for AWS CodeStar Notifications service-linked roles

AWS CodeStar Notifications supports using service-linked roles in all of the AWS Regions where the service is available. For more information, see AWS Regions and Endpoints and AWS CodeStar Notifications.

Using service-linked roles for AWS CodeConnections

AWS CodeConnections 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 AWS CodeConnections. Service-linked roles are predefined by AWS CodeConnections and include all the permissions that the service requires to call other AWS services on your behalf. This role is created for you the first time you create a connection. You don't have to create the role.

A service-linked role makes setting up AWS CodeConnections easier because you don't have to add permissions manually. AWS CodeConnections defines the permissions of its service-linked roles, and unless defined otherwise, only AWS CodeConnections 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.

To delete a service-linked role, you must first delete its related resources. This protects your AWS CodeConnections resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see [AWS Services That Work](#) [with IAM](#).

> **ℹ Note**
>
> Actions for resources that are created under the new service prefix `codeconnections` are available. Creating a resource under the new service prefix will use `codeconnections` in the resource ARN. Actions and resources for the `codestar-connections` service prefix remain available. When specifying a resource in the IAM policy, the service prefix needs to match that of the resource.

Service-linked role permissions for AWS CodeConnections

AWS CodeConnections uses the AWSServiceRoleForGitSync service-linked role to use Git sync with connected Git-based repositories.

The AWSServiceRoleForGitSync service-linked role trusts the following services to assume the role:

- `repository.sync.codeconnections.amazonaws.com`

The role permissions policy named AWSGitSyncServiceRolePolicy allows AWS CodeConnections to complete the following actions on the specified resources:

- Action: Grants permissions to allow users to create connections to external Git-based repositories and use Git sync with those repositories.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see [Service-Linked Role Permissions](#) in the *IAM User Guide*.

Creating a service-linked role for AWS CodeConnections

You don't need to manually create a service-linked role. You create the role when you create a resource for your Git-synced project with the CreateRepositoryLink API.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account.

Editing a service-linked role for AWS CodeConnections

After you create a service-linked role, you cannot change its name because various entities might reference the role. However, you can use IAM to edit the role description. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for AWS CodeConnections

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete the role. That way, you don't have an unused entity that is not actively monitored or maintained. You must clean up the resources for your service-linked role before you can delete it. This means deleting all connections that use the service role in your AWS account.

> ⓘ **Note**
>
> If the AWS CodeConnections service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete AWS CodeConnections resources used by AWSServiceRoleForGitSync

1. Open the Developer Tools console, and then choose **Settings**.

2. Choose all connections that appear in the list, and then choose **Delete**.

3. Repeat these steps in all AWS Regions where you created connections.

To use IAM to delete the service-linked role

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

Supported regions for AWS CodeConnections service-linked roles

AWS CodeConnections supports using service-linked roles in all of the AWS Regions where the service is available. For more information, see AWS Regions and Endpoints.

AWS managed policies for AWS CodeConnections

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

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

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

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

> **ⓘ Note**
>
> Actions for resources that are created under the new service prefix `codeconnections` are available. Creating a resource under the new service prefix will use `codeconnections` in the resource ARN. Actions and resources for the `codestar-connections` service prefix remain available. When specifying a resource in the IAM policy, the service prefix needs to match that of the resource.

AWS managed policy: AWSGitSyncServiceRolePolicy

You can't attach AWSGitSyncServiceRolePolicy to your IAM entities. This policy is attached to a service-linked role that allows AWS CodeConnections to perform actions on your behalf. For more information, see Using service-linked roles for AWS CodeConnections.

This policy allows customers to access Git-based repositories for use with connections. Customers will access these resources after using the CreateRepositoryLink API.

Permissions details

This policy includes the following permissions.

- `codeconnections` – Grants permissions to allow users to create connections to external Git-based repositories.

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "AccessGitRepos",
   "Effect": "Allow",
   "Action": [
    "codestar-connections:UseConnection",
    "codeconnections:UseConnection"
   ],
   "Resource": [
    "arn:aws:codestar-connections:*:*:connection/*",
    "arn:aws:codeconnections:*:*:connection/*"
   ],
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  }
 ]
}
```

AWS CodeConnections updates to AWS managed policies

View details about updates to AWS managed policies for AWS CodeConnections since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AWS CodeConnections Document history page.

Change	Description	Date
[AWSGitSyncServiceRolePolicy](#) – Updated policy	AWS CodeStar Connections service name changed to AWS CodeConnections. Updated the policy for resources with ARNs that contain both service prefixes.	April 26, 2024
[AWSGitSyncServiceRolePolicy](#) – New policy	AWS CodeStar Connections added the policy. Grants permissions to allow connections users to use Git sync with connected Git-based repositories.	November 26, 2023
AWS CodeConnections started tracking changes	AWS CodeConnections started tracking changes for its AWS managed policies.	November 26, 2023

Compliance validation for AWS CodeStar Notifications and AWS CodeConnections

For a list of AWS services in scope of specific compliance programs, see [AWS services in scope by compliance program](#). For general information, see [AWS compliance programs](#).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading reports in AWS Artifact](#).

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

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

- [AWS compliance resources](#) – This collection of workbooks and guides might apply to your industry and location.

- [AWS Config](#) – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.

- [AWS Security Hub CSPM](#) – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in AWS CodeStar Notifications and AWS CodeConnections

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

For more information about AWS Regions and Availability Zones, see [AWS global infrastructure](#).

- Notification rules are specific to the AWS Region where they are created. If you have notification rules in more than one AWS Region, use the Region selector to review notification rules in each AWS Region.

- AWS CodeStar Notifications relies on Amazon Simple Notification Service (Amazon SNS) topics as notification rule targets. Information about your Amazon SNS topics and notification rule targets might be stored in an AWS Region different from the Region in which you configured the notification rule.

Infrastructure security in AWS CodeStar Notifications and AWS CodeConnections

As features in a managed service, AWS CodeStar Notifications and AWS CodeConnections are protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of security processes](#) whitepaper.

You use AWS published API calls to access AWS CodeStar Notifications and AWS CodeConnections through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems support these modes.

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

Traffic between AWS CodeConnections resources across Regions

If you use the connections feature to enable connection of your resources, you agree and instruct us to store and process information associated with such connection resources in AWS Regions outside the AWS Regions where you are using the underlying service, solely in connection with, and for the sole purpose of, providing connection to such resources in Regions other than the one where the resource was created.

For more information, see [Global resources in AWS CodeConnections](#).

> **ⓘ Note**
>
> If you use the connections feature to enable connection for your resources in Regions that do not require first being enabled, we will store and process information as detailed in the preceding topics.
> For connections established in Regions that must first be enabled, such as the Europe (Milan) Region, we will only store and process information for that connection in that Region.

Connections rename - Summary of changes

The connections feature in the Developer Tools console allows you to connect your AWS resources to third-party source repositories. On March 29, 2024, AWS CodeStar Connections was renamed to AWS CodeConnections. The following sections describe the different parts of the feature that changed with the rename, and what actions you need to take to ensure that your resources continue to function properly.

Note that this list is not exhaustive. While other parts of the product also changed, these updates are the most relevant.

> ⓘ **Note**
>
> Actions for resources that are created under the new service prefix `codeconnections` are available. Creating a resource under the new service prefix will use `codeconnections` in the resource ARN. Actions and resources for the `codestar-connections` service prefix remain available. When specifying a resource in the IAM policy, the service prefix needs to match that of the resource.

> ⓘ **Note**
>
> Beginning July 1, 2024, the console creates connections with `codeconnections` in the resource ARN. Resources with both service prefixes will continue to display in the console.

Renamed service prefix

Connections APIs use a renamed service prefix: **codeconnections**.

To use the new prefix in CLI commands, download the version 2 of the AWS CLI. The following is an example command with the updated prefix.

```
aws codeconnections delete-connection --connection-arn arn:aws:codeconnections:us-
west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Renamed actions in IAM

The actions in IAM use the new prefix, as shown in the following examples:

```
codeconnections:CreateConnection
codeconnections:DeleteConnection
codeconnections:GetConnection
codeconnections:ListConnections
```

New resource ARN

Connections resources that are created will have a new ARN:

```
arn:aws:codeconnections:us-west-2:account-ID:connection/*
```

Affected service role policies

For the following services, service role policies will use the new prefix in policy statements. You can also update your existing service role policies to use the new permissions, but policies created with the old prefix will continue to be supported.

- The CodePipeline customer-managed service role policy
- The AWS CodeStar service role `AWSCodeStarServiceRole` policy

New CloudFormation resource

To use the CloudFormation resources for connections, a new resource will be available. The existing resource will still be supported.

- The new AWS CloudFormation resource is named AWS::CodeConnections::Connection. See AWS::CodeConnections::Connection in the CloudFormation User Guide.
- The existing AWS::CodeStarConnections::Connection resource will still be supported. See AWS::CodeStarConnections::Connection in the CloudFormation User Guide.

Document history

The following table describes the documentation for this release of the Developer Tools console.

- **AWS CodeStar Notifications API version:** 2019-10-15

- **AWS CodeConnections API version:** 2023-12-01

Change	Description	Date
[New Azure DevOps provider for connections](#)	Support added for configuring connections for AWS resources to interact with Azure DevOps. For more information, see [Create a connection to Azure DevOps](#).	August 5, 2025
[New condition key for host VPC IDs](#)	You can manage host access for GitHub Enterprise Server and GitLab self-managed hosts using the VpcId condition key. The condition key allows you to enforce policies related to creating or updating hosts to use a specified VPC ID. For more information, see the [Connections permission reference](#).	March 13, 2025
[Add support for sharing connections between accounts](#)	You can view and manage connections as resources in AWS Resource Access Manager, and you can share connections between AWS accounts. For more informati	March 6, 2025

on, see [Share connections with AWS accounts](#).

[Update to add and correct information describing how connections work with either user accounts or organizations](#)	The overviews and troubleshooting information was updated to correctly describe how connections work with either user accounts or organizations. See [How do connections work](#), [How connections in AWS CodeConnections work with organizations](#), and [Connection and host setup for installed providers supporting organizations](#).	December 9, 2024
[Update to managed policy for connections service-linked-role](#)	The managed policy for the service-linked role to use Git sync with Git repositories has been updated for resources with both service prefixes. For more information, see [Using service-linked roles for AWS CodeConnections](#) and [Managed policies](#).	April 26, 2024
[AWS CodeStar Connections renamed to AWS CodeConnections](#)	Introducing AWS CodeConnections, which allows you to create and manage connections between AWS resources, such as pipelines in CodePipeline, to third-party Git providers.	March 29, 2024

Connections to GitLab now supported in CodeBuild	Support added in CodeBuild for configuring connections to GitLab. For more informati on, see [Product and service integrations with AWS CodeConnections](#).	March 27, 2024
Support for GitLab self-mana ged	Support added for configuri ng connections and hosts for AWS resources to interact with GitLab self-managed. For more information, see [Workflow to create or update a host](#) and [Create a connectio n to GitLab self-managed](#).	December 28, 2023
New repository links and sync configurations for connectio ns	Added information about configuring repository links and sync configurations. Use the sync configuration to sync content from a Git repositor y to update your CloudForm ation stack resources. For more information, see [Working with repository links](#) and [Working with sync configurations](#).	November 27, 2023
Support for connections service-linked-role	Support added for configuri ng connections to use Git sync with Git repositories. For more information, see [Using service-linked roles for AWS CodeConnections](#) and [Managed policies](#).	November 26, 2023

Support for GitLab groups	Support added for configuri ng connections for AWS resources to interact with GitLab groups. For more information, see Create a connection and Create a connection to GitLab.	September 15, 2023
New GitLab provider type	You can now create connectio ns to GitLab. For more information, see Create a connection and Create a connection to GitLab.	August 10, 2023
New target type for notificat ion rules	You can now choose AWS Chatbot clients configured for Microsoft Teams channels as the target for notification rules. For more information, see Create a notification rule and Working with notification rule targets.	May 17, 2023
Connections are available in the Europe (Milan) Region	Added information for connections in the Europe (Milan) Region. For more information, see Traffic between AWS CodeConne ctions resources across Regions.	May 17, 2023

Added troubleshooting for connections errors with repository permissions	When creating a connection to a repository in a GitHub organization, you must be the GitHub organization owner. For more information, see Connections error when connecting to GitHub.	August 29, 2022
Added information for tagging host resources	You can now tag hosts using the console and the CLI. For more information, see Tag resources in AWS CodeConnections.	April 19, 2021
VPC endpoint support for connections	You can now use VPC endpoints with connections. For more information, see AWS CodeConnections and interface VPC endpoints (AWS PrivateLink).	November 24, 2020
New GitHub and GitHub Enterprise Cloud provider types	You can now create connections to GitHub and GitHub Enterprise Cloud. For more information, see Create a connection and Create a connection to GitHub.	September 30, 2020

[Added the GitHub Enterprise Server provider type and host resources](#)	Information about the host resource for connections has been added to this guide. You can now create connections to GitHub Enterprise Server. For more information, see [Create a connection](#) and [Working with hosts](#). This is the general availability release of the connections feature in the Developer Tools console User Guide.	June 29, 2020
[Added information for using and tagging connections](#)	Information about the connections feature in the console has been added to this guide. You can view concepts, steps for getting started, a permissions reference including example policies, and steps to create, view, and tag connections. For more information, see [What are connections](#), [Connections concepts](#), [Getting started with connections](#), [Create a connection](#), [Tag resources in AWS CodeConnections](#), [Security](#), [Quotas for connections](#), [Troubleshooting](#), and [AWS CodeConnections API calls with AWS CloudTrail](#). To view a list of additional provider actions (permissions only actions), see [Actions for ProviderType](#).	June 28, 2020

New target type for notificat ion rules	You can now choose AWS Chatbot clients configure d for Slack channels as the target for notification rules. For more information, see [Create a notification rule](#) and [Working with notification rule targets](#).	April 2, 2020
[Added notifications about additional AWS CodeCommit events](#)	You can now configure notifications for events related to pull request approvals. For more informati on, see [Events for notificat ion rules on repositories](#) and [Working with pull requests in CodeCommit](#).	February 10, 2020
[Notifications available in two additional AWS regions](#)	The Developer Tools console now supports notifications in Middle East (Bahrain) and Asia Pacific (Hong Kong). For more information, see [AWS CodeStar Notifications](#) in the AWS General Reference.	February 5, 2020
[Added support for encrypted Amazon SNS topics](#)	Guidance has been added for using encrypted Amazon SNS topics as notification targets. For more information, see [Configure Amazon SNS topics for notifications](#) .	February 4, 2020

[Notifications can include session tag information for CodeCommit](#)	Notifications for CodeCommit can now contain user identity information, such as a display name or an email address, through the use of session tags. For more information, see [Concepts](#) and [Using tags to provide identity information in CodeCommit](#).	December 19, 2019
[Initial release](#)	This is the initial release of the Developer Tools console User Guide.	November 5, 2019

AWS Glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS Glossary Reference.*