aws

用户指南

开发工具控制台

开发工具控制台: 用户指南

Table of Contents

什么是开发工具控制台？ 1
您是 新用户吗？ 3
开发工具控制台的功能 3
什么是通知？ 3
可以使用通知完成哪些操作？ 3
通知的工作原理是什么？ 4
如何开始使用通知？ 4
通知概念 4
设置 11
开始使用通知 17
使用通知规则 23
使用通知规则目标 35
配置通知与 AWS Chatbot 之间的集成 43
使用 AWS CloudTrail 记录 AWS CodeStar 通知 API 调用 48
问题排查 51
配额 53
什么是连接？ 54
我可以使用连接执行哪些操作？ 54
我可以为哪些第三方提供商创建连接？ 54
什么与连接 AWS 服务 集成？ 55
连接是如何工作的？ 55
如何开始使用连接？ 59
连接概念 60
AWS CodeStar 连接支持的提供程序和版本 60
产品和服务与 AWS CodeStar Connections 的集成 61
设置连接 64
开始使用连接 67
使用连接 72
使用主机 120
针对已链接存储库使用同步配置 130
使用 CloudTrail 记录 API 调用 139
VPC 端点 (AWS PrivateLink) 141
排除连接故障 144
限额 154

要添加到允许列表的 IP 地址 155
安全性 158
了解通知内容和安全性 158
数据保护 159
Identity and Access Management 160
受众 160
使用身份进行身份验证 161
使用策略管理访问 163
开发工具控制台中的特征如何与 IAM 配合使用 164
AWS CodeConnections 权限参考 169
基于身份的策略示例 184
使用标签控制对 C AWS CodeStar onnections 资源的访问权限 196
使用 控制台 198
允许用户查看他们自己的权限 199
故障排除 200
为 AWS CodeStar 通知使用服务相关角色 202
将服务相关角色用于 AWS CodeConnections 206
AWS 托管式策略 207
合规性验证 209
故障恢复能力 210
基础设施安全性 210
跨区域的 AWS CodeConnections资源之间的流量 211
文档历史记录 212
AWS 术语表 216
..... ccxvii

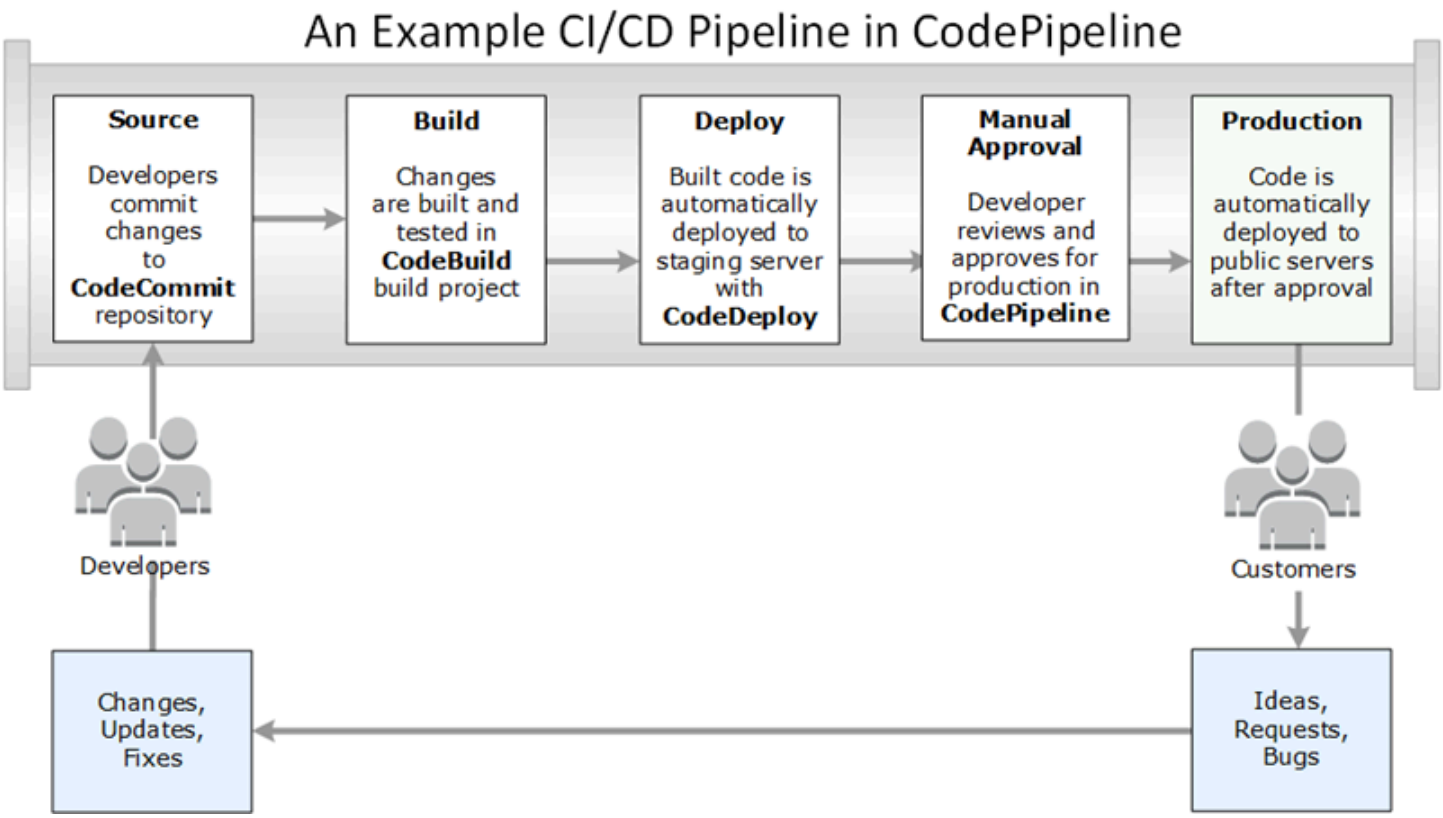
什么是开发工具控制台？

开发工具控制台是一组服务和功能的所在地，您可以单独或共同使用这些服务和功能来帮助您以个人或团队形式开发软件。开发人员工具可帮助您安全地存储、构建、测试和部署软件。这些工具可单独或结合使用，为开发运营、持续集成和持续交付 (CI/CD) 提供支持。

开发工具控制台包括以下服务：

- AWS CodeCommit 是一项完全托管的源代码控制服务，可托管私有 Git 存储库。您可以使用存储库在 AWS Cloud 中私下存储和管理资产（如文档、源代码和二进制文件）。您的存储库将储存从首次提交到最新更改的项目历史记录。您可以通过对代码进行注释并创建拉取请求来帮助确保代码质量，从而协作处理存储库中的代码。
- AWS CodeBuild 是一项完全托管的构建服务，可编译源代码、运行单元测试和生成部署就绪的构件。它提供了适用于常用编程语言的预先打包的构建环境以及 Apache Maven 和 Gradle 等构建工具。您还可以在 CodeBuild 中自定义构建环境以使用自己的构建工具。
- AWS CodeDeploy 是一项完全托管的部署服务，可自动将软件部署到计算服务（如 Amazon EC2、AWS Lambda 和您的本地服务器）。它可帮助您快速推出新功能，避免在应用程序部署过程中出现停机，并简化应用程序的更新工作。
- AWS CodePipeline 是一项持续集成和持续交付服务，可用于建模、可视化和自动执行发布软件所需的步骤。您可以快速对软件发布过程的不同阶段进行建模和配置。根据您定义的发布流程模型，只要代码发生变化，您便能构建、测试和部署您的代码。

以下示例说明了如何将开发工具控制台中的各项服务结合使用来帮助您开发软件。



在此示例中，开发人员在 CodeCommit 中创建一个存储库，并使用该存储库来开发和协作使用其代码。他们在 CodeBuild 中创建一个构建项目来构建和测试其代码，并使用 CodeDeploy 将其代码部署到测试和生产环境中。他们希望快速进行迭代，因此，他们在 CodePipeline 中创建一个管道来检测 CodeCommit 存储库中的更改。构建这些更改并运行了测试，还将成功构建和测试的代码部署到测试服务器。团队将测试阶段添加到管道，以便在临时服务器上运行更多测试，例如集成或负载测试。在成功完成这些测试后，团队成员将检查结果，如果结果令人满意，则手动批准将更改应用于生产。CodePipeline 将经测试和批准的代码部署到生产实例。

这只是一个简单示例，说明如何使用开发工具控制台中提供的一项或多项服务来帮助您开发软件。可以对每项服务进行自定义来满足您的需求。它们提供了与 AWS 中的其他产品和服务以及其他第三方工具的多项集成。有关更多信息，请参阅以下主题：

- CodeCommit：产品和服务集成
- CodeBuild：将 CodeBuild 与 Jenkins 结合使用
- CodeDeploy： 产品和服务集成
- CodePipeline：产品和服务集成

您是 新用户吗？

如果您是开发工具控制台中提供的一项或多项服务的新用户，我们建议您首先阅读以下主题：

- CodeCommit 入门
- CodeBuild 入门、概念
- CodeDeploy 入门、主要组件
- CodePipeline 入门、概念

开发工具控制台的功能

开发工具控制台包括以下功能：

- 开发工具控制台包含通知管理器功能，该功能可用于订阅 AWS CodeBuild、AWS CodeCommit、AWS CodeDeploy 和 AWS CodePipeline 中的事件。此功能具有自己的 API，即 AWS CodeStar 通知。您可以使用通知功能来快速向用户通知有关存储库、构建项目、部署应用程序和管道中对其工作最重要的事件的信息。通知管理器可帮助用户了解存储库、构建、部署或管道上发生的事件，以便他们能够快速采取措施，例如批准更改或更正错误。有关更多信息，请参阅什么是通知？。
- 开发工具控制台包含一个连接功能，您可以使用该功能将 AWS 资源与第三方源代码提供程序关联起来。此功能具有自己的 API，即 AWS CodeStar 连接。您可以使用连接功能设置与第三方提供程序的授权连接，并将连接资源与其他 AWS 服务结合使用。有关更多信息，请参阅什么是连接？。

什么是通知？

开发工具控制台中的通知功能是用于订阅 AWS CodeBuild、AWS CodeCommit、AWS CodeDeploy 和 AWS CodePipeline 中的事件的通知管理器。它具有自己的 API，即 AWS CodeStar 通知。您可以使用通知功能来快速向用户通知有关存储库、构建项目、部署应用程序和管道中对其工作最重要的事件的信息。通知管理器可帮助用户了解存储库、构建、部署或管道上发生的事件，以便他们能够快速采取措施，例如批准更改或更正错误。

可以使用通知完成哪些操作？

您可以使用通知功能来创建和管理通知规则，以向用户告知对其资源进行的重要更改，包括：

- CodeBuild 构建项目中的构建成功与失败。

- CodeDeploy 应用程序中的部署成功和失败。
- 在 CodeCommit 存储库中创建和更新拉取请求，包括对代码的注释。
- CodePipeline 中的手动批准状态和管道运行。

您可以设置通知，以使它们进入订阅了 Amazon SNS 主题的用户电子邮件地址。您还可以将此功能与 AWS Chatbot 集成，并将通知传递到 Slack 通道、Microsoft Teams 通道或 Amazon Chime 聊天室。

通知的工作原理是什么？

当为受支持的资源（例如存储库、构建项目、应用程序或管道）配置通知规则时，通知功能将创建一个 Amazon EventBridge 规则来监控指定的事件。当发生此类事件时，通知规则会将通知发送到指定为该规则的目标的 Amazon SNS 主题。这些目标的订阅者会收到有关这些事件的通知。

如何开始使用通知？

要开始使用通知，请先回顾此处提供的一些实用主题：

- 了解通知的概念。
- 设置开始使用通知所需的资源。
- 开始使用您的第一条通知规则并接收您的第一条通知。

通知概念

如果您了解概念和术语，则可更轻松地设置和使用通知。以下是使用通知时要了解的一些概念。

主题

- 通知
- 通知规则
- 事件
- 详细信息类型
- 目标
- 通知和 AWS CodeStar 通知
- 存储库的通知规则的事件
- 构建项目的通知规则的事件
- 部署应用程序上的通知规则事件

- 管道的通知规则的事件

通知

通知 是一条消息，其中包含有关您和开发人员使用的资源中所发生事件的信息。您可以设置通知，以使资源（例如构建项目、存储库、部署应用程序或管道）的用户根据您创建的通知规则，接收有关您指定的事件类型的电子邮件。

通过使用会话标签，AWS CodeCommit 的通知可以包含用户身份信息，例如显示名称或电子邮件地址。CodeCommit 支持使用会话标签，这些标签是您在代入 IAM 角色、使用临时凭证或在 AWS Security Token Service (AWS STS) 中对用户进行联合身份验证时传递的键值对属性。您还可以将标签与 IAM 用户关联。CodeCommit 在通知内容中包括 `displayName` 和 `emailAddress` 的值（如果存在这些标签）。有关更多信息，请参阅使用标签在 CodeCommit 中提供其他身份信息。

> **Important**
>
> 通知包括特定于项目的信息，例如构建状态、部署状态、具有注释的代码行和管道批准。在添加新功能时，通知内容可能会发生变化。作为安全最佳实践，您应定期查看通知规则和 Amazon SNS 主题订阅者的目标。有关更多信息，请参阅了解通知内容和安全性。

通知规则

通知规则是您创建的 AWS 资源，用于指定发送通知的时间和地点。它定义：

- 创建通知的条件。这些条件基于您选择的事件，特定于资源类型。支持的资源类型包括 AWS CodeBuild 中的构建项目、AWS CodeDeploy 中的部署应用程序、AWS CodePipeline 中的管道和 AWS CodeCommit 中的存储库。
- 将通知发送到的目标。您最多可以为一个通知规则指定 10 个目标。

通知规则的范围仅限于各个构建项目、部署应用程序、管道和存储库。通知规则同时具有用户定义的友好名称和 Amazon 资源名称 (ARN)。必须在资源所在的同一 AWS 区域中创建通知规则。例如，如果您在美国东部（俄亥俄）区域中构建项目，则还必须在美国东部（俄亥俄）区域中创建您的通知规则。

您最多可以为资源定义 10 条通知规则。

事件

事件 是要监视的资源上的状态更改。每个资源都有事件类型的列表，您可以从中进行选择。在资源上设置通知规则时，可以指定导致发送通知的事件。例如，如果您在 CodeCommit 中为存储库设置通知，并为 Pull request (拉取请求) 和 Branches and tags (分支和标签) 选择 Created (已创建)，则每当存储库中的用户创建拉取请求、分支或 Git 标签时，都会发送通知。

详细信息类型

创建通知规则时，您可以选择通知中包含的详细信息级别或详细信息类型 （Full (完整) 或 Basic (基本)）。Full (完整) 设置（默认）在通知中包括可用于事件的所有信息，包括服务针对特定事件提供的任何增强信息。Basic (基本) 设置仅包括可用信息的子集。

下表列出可用于特定事件类型的增强信息，并描述详细信息类型之间的差异。

服务	事件	Full (完整) 包含	Basic (基本) 不包括
CodeCommit	关于提交的注释 关于拉取请求的注释	所有事件详细信息和注释的内容，包括任何回复或注释主题。它还包括注释所依据的行号和代码行。	注释的内容、行号、代码行或任何注释主题。
CodeCommit	已创建拉取请求	拉取请求中与目标分支相关的所有事件详细信息以及添加、修改或删除的文件数。	没有关于拉取请求源分支是否已添加、修改或删除文件的文件列表或详细信息。
CodePipeline	需要手动批准	所有事件详细信息和自定义数据（如果已配置）。通知还包含指向管道中所需批准的链接。	没有自定义数据或链接。
CodePipeline	操作执行失败 管道执行失败 阶段执行失败	故障的所有事件详细信息和错误消息的内容。	没有错误消息内容。

目标

目标 是从通知规则接收通知的位置。允许的目标类型是为 Slack 或 Microsoft Teams 通道配置的 Amazon SNS 主题和 AWS Chatbot 客户端。订阅目标的任何用户都会收到有关您在通知规则中指定的事件的通知。

如果您要扩展通知的范围，可以手动配置通知和 AWS Chatbot 之间的集成，以便将通知发送到 Amazon Chime 聊天室。然后，您可以选择为该 AWS Chatbot 客户端配置的 Amazon SNS 主题作为通知规则的目标。有关更多信息，请参阅将通知与 AWS Chatbot 和 Amazon Chime 集成。

如果选择将 AWS Chatbot 客户端用作目标，则必须首先在 AWS Chatbot 中创建该客户端。当您选择 AWS Chatbot 客户端作为通知规则的目标时，会为该 AWS Chatbot 客户端配置一个 Amazon SNS 主题，其中包含将通知发送到 Slack 或 Microsoft Teams 通道所需的全部策略。您不必为 AWS Chatbot 客户端配置任何现有 Amazon SNS 主题。

您可选择在创建通知规则的过程中，创建 Amazon SNS 主题作为目标（推荐）。您也可以选择与通知规则位于相同 AWS 区域中的现有 Amazon SNS 主题，但必须使用所需的策略对其进行配置。用于目标的 Amazon SNS 主题必须位于您的 AWS 账户中。并且与通知规则和为其创建规则的 AWS 资源位于同一个 AWS 区域中。

例如，如果您在美国东部（俄亥俄）区域中为存储库创建通知规则，则 Amazon SNS 主题也必须存在于该区域中。如果在创建通知规则的过程中创建一个 Amazon SNS 主题，则会使用允许将事件发布到该主题所需的策略来配置该主题。这是使用目标和通知规则的最佳方法。如果您选择使用已存在的主题或手动创建一个主题，则必须在用户接收通知之前使用所需的权限配置该主题。有关更多信息，请参阅配置通知的 Amazon SNS 主题。

> **Note**
>
> 如果要使用现有 Amazon SNS 主题而不是创建新主题，请在 Targets (目标) 中选择其 ARN。请确保主题具有适当的访问策略，并且订阅者列表仅包含允许查看有关资源的信息的用户。如果 Amazon SNS 主题是在 2019 年 11 月 5 日之前用于 CodeCommit 通知的，它将包含允许 CodeCommit 向其发布的策略，该策略包含的权限与 AWS CodeStar 通知所需的权限不同。建议不使用这些主题。如果要使用为该体验创建的策略，则除了已存在的策略之外，还必须添加 AWS CodeStar 通知所需的策略。有关更多信息，请参阅 配置通知的 Amazon SNS 主题 和 了解通知内容和安全性：

通知和 AWS CodeStar 通知

通知是开发工具控制台的一项功能，它具有自己的 API，即 AWS CodeStar 通知。它还具有自己的 AWS 资源类型（通知规则）、权限和事件。通知规则的事件将记录在 AWS CloudTrail 中。可以通过 IAM 策略允许或拒绝 API 操作。

存储库的通知规则的事件

类别	事件	事件 ID
注释	提交时 拉取请求时	codecommit-repository-comments-on-commits codecommit-repository-comments-on-pull-requests
审批	状态已更改 规则覆盖	codecommit-repository-approvals-status-changed codecommit-repository-approvals-rule-override
拉取请求	已创建 源代码已更新 状态已更改 已合并	codecommit-repository-pull-request-created codecommit-repository-pull-request-source-updated codecommit-repository-pull-request-status-changed codecommit-repository-pull-request-merged

类别	事件	事件 ID
分支和标签	已创建 Deleted (已删除) Updated	codecommit-repository-branches-and-tags-created codecommit-repository-branches-and-tags-deleted codecommit-repository-branches-and-tags-updated

构建项目的通知规则的事件

类别	事件	事件 ID
构建状态	Failed Succeeded 进行中 Stopped (已停止)	codebuild-project-build-state-failed codebuild-project-build-state-succeeded codebuild-project-build-state-in-progress codebuild-project-build-state-stopped
构建阶段	失败 成功	codebuild-project-build-phase-failure codebuild-project-build-phase-success

部署应用程序上的通知规则事件

类别	事件	事件 ID
部署	Failed Succeeded Started	codedeploy-application-deployment-failed codedeploy-application-deployment-succeeded codedeploy-application-deployment-started

管道的通知规则的事件

类别	事件	事件 ID
操作执行	Succeeded Failed 已取消 Started	codepipeline-pipeline-action-execution-succeeded codepipeline-pipeline-action-execution-failed codepipeline-pipeline-action-execution-canceled codepipeline-pipeline-action-execution-started
阶段执行	Started Succeeded 已恢复 已取消 已失败	codepipeline-pipeline-stage-execution-started codepipeline-pipeline-stage-execution-succeeded codepipeline-pipeline-stage-execution-resumed

类别	事件	事件 ID
		codepipeline-pipeline-stage-execution-canceled codepipeline-pipeline-stage-execution-failed
管道执行	Failed 已取消 Started 已恢复 Succeeded 已取代	codepipeline-pipeline-pipeline-execution-failed codepipeline-pipeline-pipeline-execution-canceled codepipeline-pipeline-pipeline-execution-started codepipeline-pipeline-pipeline-execution-resumed codepipeline-pipeline-pipeline-execution-succeeded codepipeline-pipeline-pipeline-execution-superseded
手动审批	Failed 需要 Succeeded	codepipeline-pipeline-manual-approval-failed codepipeline-pipeline-manual-approval-needed codepipeline-pipeline-manual-approval-succeeded

设置

如果您已将 AWS CodeBuild、AWS CodeCommit、AWS CodeDeploy 或 AWS CodePipeline 的托管策略应用于 IAM 用户或角色，则您具有在策略提供的角色和权限的限制内使用通知所需的权限。例如，已应用

AWSCodeBuildAdminAccess、AWSCodeCommitFullAccess、AWSCodeDeployFullAccess 或 AWSCodePipeline_FullAccess 托管策略的用户具有对通知的完全管理访问权限。

有关包括示例策略在内的更多信息，请参阅 基于身份的策略。

如果您已将其中一项策略应用于 IAM 用户或角色、CodeBuild 中的构建项目、CodeCommit 中的存储库、CodeDeploy 中的部署应用程序或 CodePipeline 中的管道，则您已准备好创建第一个通知规则。继续浏览 开始使用通知。如果尚未这样做，请参阅以下主题：

- CodeBuild：开始使用 CodeBuild
- CodeCommit：开始使用 CodeCommit
- CodeDeploy：教程
- CodePipeline：开始使用 CodePipeline

如果您要自己管理 IAM 用户、组或角色的通知的管理权限，请按照本主题中的过程进行操作，以设置使用服务所需的权限和资源。

如果要对通知使用以前创建的 Amazon SNS 主题，而不是专门为通知创建主题，则必须通过应用允许将事件发布到 Amazon SNS 主题的策略，来将该主题配置为用作通知规则的目标。

> **Note**
>
> 要执行以下步骤，您必须使用具有管理权限的账户登录。有关更多信息，请参阅创建您的第一个 IAM 管理员用户和组。

主题

- 创建并应用针对通知的管理访问的策略
- 配置通知的 Amazon SNS 主题
- 为用户订阅作为目标的 Amazon SNS 主题

创建并应用针对通知的管理访问的策略

您可以通过使用 IAM 用户，或使用有权访问您要为其创建通知的服务（AWS CodeBuild、AWS CodeCommit、AWS CodeDeploy 或 AWS CodePipeline）的角色进行登录来管理通知。您也可以创建自己的策略并将其应用于用户或组。

以下过程说明了如何配置具有管理通知和添加 IAM 用户的权限的 IAM 组。如果您不想设置组，则可以将此策略直接应用于 IAM 用户或用户可以代入的 IAM 角色。您还可以使用 CodeBuild、CodeCommit、CodeDeploy 或 CodePipeline 的托管策略（其中包括对通知功能的策略访问权），具体取决于策略的范围。

对于下面的策略，输入此策略的名称（例如 `AWSCodeStarNotificationsFullAccess`）和可选描述。该描述可帮助您记住策略的用途（例如 **`This policy provides full access to AWS CodeStar Notifications.`**）

使用 JSON 策略编辑器创建策略

1. 登录AWS Management Console，然后通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在左侧的导航窗格中，选择策略。

   如果这是您首次选择 Policies，则会显示 Welcome to Managed Policies 页面。选择开始使用。
3. 在页面的顶部，选择 Create policy (创建策略)。
4. 在策略编辑器部分，选择 JSON 选项。
5. 输入以下 JSON 策略文档：

```
{
   "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. 选择 Next（下一步）。

> **Note**
>
> 您可以随时在可视化和 JSON 编辑器选项卡之间切换。不过，如果您进行更改或在可视化编辑器中选择下一步，IAM 可能会调整您的策略结构以针对可视化编辑器进行优化。有关更多信息，请参阅《IAM 用户指南》中的调整策略结构。

7. 在 Review and create（查看并创建）页面上，为您要创建的策略输入 Policy name（策略名称）和 Description（描述）（可选）。查看此策略中定义的权限以查看您的策略授予的权限。
8. 选择 Create policy (创建策略) 可保存您的新策略。

配置通知的 Amazon SNS 主题

设置通知最简单的方法是在创建通知规则时创建 Amazon SNS 主题。如果现有 Amazon SNS 主题满足以下要求，则可使用该主题：

- 已在与要为其创建通知规则的资源（构建项目、部署应用程序、存储库或管道）相同的 AWS 区域中创建该主题。
- 在 2019 年 11 月 5 日之前，该主题尚未用于发送 CodeCommit 的通知。如果已经使用该主题，它将包含启用了该功能的策略语句。您可以选择使用此主题，但需要添加在该过程中指定的其他策略。如果在 2019 年 11 月 5 日之前仍然为通知配置了一个或多个存储库，则不应删除现有策略语句。
- 它具有允许 AWS CodeStar 通知将通知发布到该主题的策略。

配置 Amazon SNS 主题以用作 AWS CodeStar 通知规则的目标

1. 访问 https://console.aws.amazon.com/sns/v3/home，登录 AWS Management Console 并打开 Amazon SNS 控制台。
2. 在导航栏中，依次选择 Topics (主题)、要配置的主题和 Edit (编辑)。
3. 展开 Access policy (访问策略)，然后选择 Advanced (高级)。
4. 在 JSON 编辑器中，向策略中添加以下语句。包含主题 ARN、AWS 区域、AWS 账户 ID 和主题名称。

```
 {
      "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"
    }
```

该策略语句应与以下内容类似。

```
{
  "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",
        "SNS:Receive"
      ],
      "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. 选择 Save changes（保存更改）。

6. 如果您要使用 AWS KMS 加密的 Amazon SNS 主题发送通知，则还必须在事件源（AWS CodeStar 通知）与加密主题之间启用兼容性，方法是将以下语句添加到 AWS KMS key 的策略中。将 AWS 区域（本示例中为 us-east-2）替换为在其中创建密钥的 AWS 区域。

```
{
    "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"
                }
            }
        }
    ]
```

```
}
```

有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的静态加密和针对 AWS KMS 使用策略条件。

为用户订阅作为目标的 Amazon SNS 主题

用户必须先订阅作为通知规则的目标的 Amazon SNS 主题，然后才能接收通知。如果用户通过电子邮件地址进行订阅，则必须在收到通知之前确认其订阅。要向 Slack 通道、Microsoft Teams 通道或 Amazon Chime 聊天室中的用户发送通知，请参阅配置通知与 AWS Chatbot 之间的集成。

为用户订阅用于通知的 Amazon SNS 主题

1. 访问 https://console.aws.amazon.com/sns/v3/home，登录 AWS Management Console 并打开 Amazon SNS 控制台。
2. 在导航栏中，选择 Topics (主题)，然后选择要为用户订阅的主题。
3. 在订阅中，选择创建订阅。
4. 在协议中，选择电子邮件。在 Endpoint (终端节点) 中，输入电子邮件地址，然后选择 Create subscription (创建订阅)。

开始使用通知

开始使用通知的最简单方法是在您的构建项目、部署应用程序、管道或存储库之一上设置通知规则。

> **Note**
>
> 首次创建通知规则时，您的账户中会创建服务相关角色。有关更多信息，请参阅为 AWS CodeStar 通知使用服务相关角色。

主题

- 先决条件
- 为存储库创建通知规则
- 为构建项目创建通知规则
- 为部署应用程序创建通知规则

- 为管道创建通知规则

先决条件

完成 设置 中的步骤。您还需要将为其创建通知规则的资源。

- 在 CodeBuild 中创建构建项目或使用现有构建项目。
- 创建应用程序或使用现有部署应用程序。
- 在 CodePipeline 中创建管道或使用现有管道。
- 创建 AWS CodeCommit 存储库或使用现有存储库。

为存储库创建通知规则

您可以创建通知规则，以发送有关对您很重要的存储库事件的通知。以下步骤显示如何在单个存储库事件上设置通知规则。在编写这些步骤时，假设您已在 AWS 账户中配置一个存储库。

> **Important**
>
> 如果您在 2019 年 11 月 5 日之前在 CodeCommit 中设置了通知，则用于这些通知的 Amazon SNS 主题将包含一个策略，该策略允许 CodeCommit 向其发布与 AWS CodeStar 通知所需的权限不同的权限。建议不使用这些主题。如果要使用为该体验创建的策略，则除了已存在的策略之外，还必须添加 AWS CodeStar 通知所需的策略。有关更多信息，请参阅 配置通知的 Amazon SNS 主题 和 了解通知内容和安全性：

1. 打开 CodeCommit 控制台 (https://console.aws.amazon.com/codecommit)。
2. 从列表中选择一个存储库并将其打开。
3. 选择 Notify (通知)，然后选择 Create notification rule (创建通知规则)。您也可以依次选择 Settings (设置)、Notifications (通知) 和 Create notification rule (创建通知规则)。
4. 在 Notification name (通知名称) 中，输入规则的名称。
5. 如果您只想在通知中包含提供给 Amazon EventBridge 的信息，则在 Detail type (详细信息类型) 中，选择 Basic (基本)。如果您希望包含提供给 Amazon EventBridge 的信息以及资源服务或通知管理器可能提供的信息，选择 Full (完整)。

   有关更多信息，请参阅了解通知内容和安全性。

6. 在 Events that trigger notifications (触发通知的事件) 中的 Branches and tags (分支和标签) 下，选择 Created (已创建)。
7. 在 Targets (目标) 中，选择 Create SNS topic (创建 SNS 主题)。

> **Note**
>
> 作为创建通知规则的一部分创建主题时，将为您应用允许 CodeCommit 将事件发布到主题的策略。使用为通知规则创建的主题有助于确保您仅订阅要接收有关此存储库的通知的那些用户。

在 codestar-notifications- 前缀后面，输入主题的名称，然后选择 Submit (提交)。

> **Note**
>
> 如果要使用现有 Amazon SNS 主题而不是创建新主题，请在 Targets (目标) 中选择其 ARN。请确保主题具有适当的访问策略，并且订阅者列表仅包含允许查看有关资源的信息的用户。如果 Amazon SNS 主题是在 2019 年 11 月 5 日之前用于 CodeCommit 通知的，它将包含允许 CodeCommit 向其发布的策略，该策略包含的权限与 AWS CodeStar 通知所需的权限不同。建议不使用这些主题。如果要使用为该体验创建的策略，则除了已存在的策略之外，还必须添加 AWS CodeStar 通知所需的策略。有关更多信息，请参阅 配置通知的 Amazon SNS 主题 和 了解通知内容和安全性：

8. 选择 Submit (提交)，然后查看通知规则。
9. 以您的电子邮件地址订阅您刚创建的 Amazon SNS 主题。有关更多信息，请参阅为用户订阅用于通知的 Amazon SNS 主题。
10. 导航到您的存储库并从原定设置分支创建测试分支。
11. 创建分支后，通知规则会向所有主题订阅者发送通知，其中包含有关事件的信息。

为构建项目创建通知规则

您可以创建通知规则，以发送有关对您非常重要的构建项目事件的通知。以下步骤显示如何在单个构建项目事件上设置通知规则。在编写这些步骤时，假设您已在 AWS 账户中配置一个构建项目。

1. 打开 https://console.aws.amazon.com/codebuild/ 上的 CodeBuild 控制台。
2. 从列表中选择构建项目并将其打开。

3. 选择 Notify (通知)，然后选择 Create notification rule (创建通知规则)。您也可以选择 Settings (设置)，然后选择 Create notification rule (创建通知规则)。
4. 在 Notification name (通知名称) 中，输入规则的名称。
5. 如果您只想在通知中包含提供给 Amazon EventBridge 的信息，则在 Detail type (详细信息类型) 中，选择 Basic (基本)。如果您希望包含提供给 Amazon EventBridge 的信息以及资源服务或通知管理器可能提供的信息，选择 Full (完整)。

   有关更多信息，请参阅[了解通知内容和安全性]。
6. 在 Events that trigger notifications (触发通知的事件) 中的 Build phase (构建阶段) 下，选择 Success (成功)。
7. 在 Targets (目标) 中，选择 Create SNS topic (创建 SNS 主题)。

   > **Note**
   >
   > 作为创建通知规则的一部分创建主题时，将为您应用允许 CodeBuild 将事件发布到主题的策略。使用为通知规则创建的主题有助于确保您仅订阅要接收有关此构建项目的通知的那些用户。

   在 codestar-notifications- 前缀后面，输入主题的名称，然后选择 Submit (提交)。

   > **Note**
   >
   > 如果要使用现有 Amazon SNS 主题而不是创建新主题，请在 Targets (目标) 中选择其 ARN。请确保主题具有适当的访问策略，并且订阅者列表仅包含允许查看有关资源的信息的用户。如果 Amazon SNS 主题是在 2019 年 11 月 5 日之前用于 CodeCommit 通知的，它将包含允许 CodeCommit 向其发布的策略，该策略包含的权限与 AWS CodeStar 通知所需的权限不同。建议不使用这些主题。如果要使用为该体验创建的策略，则除了已存在的策略之外，还必须添加 AWS CodeStar 通知所需的策略。有关更多信息，请参阅 [配置通知的 Amazon SNS 主题] 和 [了解通知内容和安全性]：
8. 选择 Submit (提交)，然后查看通知规则。
9. 以您的电子邮件地址订阅您刚创建的 Amazon SNS 主题。有关更多信息，请参阅[为用户订阅用于通知的 Amazon SNS 主题]。
10. 导航到您的构建项目并开始构建。
11. 在成功完成构建阶段之后，通知规则将向所有主题订阅者发送通知，其中包含有关该事件的信息。

为部署应用程序创建通知规则

您可以创建通知规则，以发送有关对您非常重要的部署应用程序事件的通知。以下步骤显示如何在单个构建项目事件上设置通知规则。在编写这些步骤时，假设您已在 AWS 账户中配置一个部署应用程序。

1. 从 https://console.aws.amazon.com/codedeploy/ 打开 CodeDeploy 控制台。
2. 从列表中选择一个应用程序并将其打开。
3. 选择 Notify (通知)，然后选择 Create notification rule (创建通知规则)。您也可以选择 Settings (设置)，然后选择 Create notification rule (创建通知规则)。
4. 在 Notification name (通知名称) 中，输入规则的名称。
5. 如果您只想在通知中包含提供给 Amazon EventBridge 的信息，则在 Detail type (详细信息类型) 中，选择 Basic (基本)。如果您希望包含提供给 Amazon EventBridge 的信息以及资源服务或通知管理器可能提供的信息，选择 Full (完整)。

   有关更多信息，请参阅了解通知内容和安全性。

6. 在 Events that trigger notifications (触发通知的事件) 中的 Deployment (部署) 下，选择 Succeeded (已成功)。
7. 在 Targets (目标) 中，选择 Create SNS topic (创建 SNS 主题)。

   > **Note**
   >
   > 作为创建通知规则的一部分创建主题时，将为您应用允许 CodeDeploy 将事件发布到主题的策略。使用为通知规则创建的主题有助于确保您仅订阅要接收有关此部署应用程序的通知的那些用户。

   在 codestar-notifications- 前缀后面，输入主题的名称，然后选择 Submit (提交)。

   > **Note**
   >
   > 如果要使用现有 Amazon SNS 主题而不是创建新主题，请在 Targets (目标) 中选择其 ARN。请确保主题具有适当的访问策略，并且订阅者列表仅包含允许查看有关资源的信息的用户。如果 Amazon SNS 主题是在 2019 年 11 月 5 日之前用于 CodeCommit 通知的，它将包含允许 CodeCommit 向其发布的策略，该策略包含的权限与 AWS CodeStar 通知所需的权限不同。建议不使用这些主题。如果要使用为该体验创建的策略，则除了已存在

> 的策略之外，还必须添加 AWS CodeStar 通知所需的策略。有关更多信息，请参阅 配置通知的 Amazon SNS 主题 和 了解通知内容和安全性：

8. 选择 Submit (提交)，然后查看通知规则。
9. 以您的电子邮件地址订阅您刚创建的 Amazon SNS 主题。有关更多信息，请参阅为用户订阅用于通知的 Amazon SNS 主题。
10. 导航到部署应用程序并启动部署。
11. 部署成功后，通知规则会向所有主题订阅者发送通知，其中包含有关事件的信息。

为管道创建通知规则

您可以创建通知规则，以发送有关对您非常重要的管道的事件的通知。以下步骤显示如何在单个管道事件上设置通知规则。在编写这些步骤时，假设您已在 AWS 账户中配置一个管道。

1. 在 https://console.aws.amazon.com/codepipeline/ 打开 CodePipeline 控制台。
2. 从列表中选择一个管道并将其打开。
3. 选择 Notify (通知)，然后选择 Create notification rule (创建通知规则)。您也可以选择 Settings (设置)，然后选择 Create notification rule (创建通知规则)。
4. 在 Notification name (通知名称) 中，输入规则的名称。
5. 如果您只想在通知中包含提供给 Amazon EventBridge 的信息，则在 Detail type (详细信息类型) 中，选择 Basic (基本)。如果您希望包含提供给 Amazon EventBridge 的信息以及资源服务或通知管理器可能提供的信息，选择 Full (完整)。

   有关更多信息，请参阅了解通知内容和安全性。
6. 在 Events that trigger notifications (触发通知的事件) 中的 Action execution (操作执行) 下，选择 Started (已开始)。
7. 在 Targets (目标) 中，选择 Create SNS topic (创建 SNS 主题)。

> **Note**
>
> 作为创建通知规则的一部分创建主题时，将为您应用允许 CodePipeline 将事件发布到主题的策略。使用为通知规则创建的主题有助于确保您仅订阅要接收有关此管道的通知的那些用户。

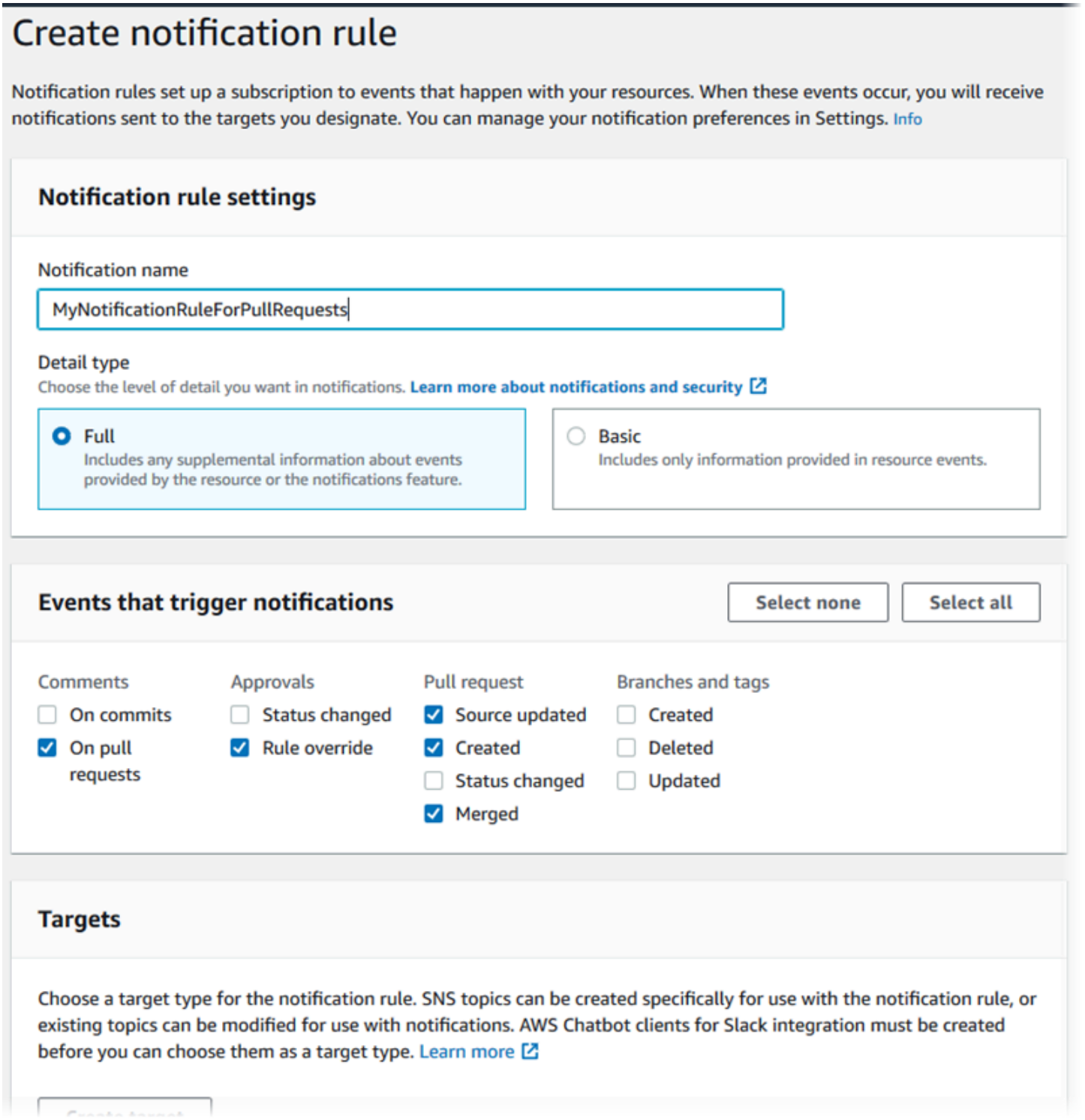
在 codestar-notifications- 前缀后面，输入主题的名称，然后选择 Submit (提交)。

> **Note**
>
> 如果要使用现有 Amazon SNS 主题而不是创建新主题，请在 Targets (目标) 中选择其 ARN。请确保主题具有适当的访问策略，并且订阅者列表仅包含允许查看有关资源的信息的用户。如果 Amazon SNS 主题是在 2019 年 11 月 5 日之前用于 CodeCommit 通知的，它将包含允许 CodeCommit 向其发布的策略，该策略包含的权限与 AWS CodeStar 通知所需的权限不同。建议不使用这些主题。如果要使用为该体验创建的策略，则除了已存在的策略之外，还必须添加 AWS CodeStar 通知所需的策略。有关更多信息，请参阅 配置通知的 Amazon SNS 主题 和 了解通知内容和安全性：

8. 选择 Submit (提交)，然后查看通知规则。
9. 以您的电子邮件地址订阅您刚创建的 Amazon SNS 主题。有关更多信息，请参阅为用户订阅用于通知的 Amazon SNS 主题。
10. 导航到您的管道，然后选择发布更改。
11. 在操作开始时，通知规则会向所有主题订阅者发送通知，其中包含有关事件的信息。

使用通知规则

通知规则可让您配置希望用户接收有关哪些事件的通知，并指定将接收这些通知的目标。您可以通过 Amazon SNS，或者通过为 Slack 或 Microsoft Teams 通道配置的 AWS Chatbot 客户端直接向用户发送通知。如果您要扩展通知的范围，可以手动配置通知和 AWS Chatbot 之间的集成，以便将通知发送到 Amazon Chime 聊天室。有关更多信息，请参阅 目标 和 将通知与 AWS Chatbot 和 Amazon Chime 集成：



您可以使用开发工具控制台或 AWS CLI 创建和管理通知规则。

主题

- 创建通知规则

- 查看通知规则
- 编辑通知规则
- 为通知规则启用或禁用通知
- 删除通知规则

创建通知规则

您可以使用开发工具控制台或 AWS CLI 创建通知规则。您可以在创建规则的过程中，创建一个 Amazon SNS 主题来用作通知规则的目标。如果要将 AWS Chatbot 客户端用作目标，您必须先创建该客户端，然后才能创建规则。有关更多信息，请参阅为 Slack 通道配置 AWS Chatbot 客户端。

创建通知规则（控制台）

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。
2. 使用导航栏导航到该资源。
   - 对于 CodeBuild，选择 Build (构建)，选择 Build projects (构建项目)，然后选择一个构建项目。
   - 对于 CodeCommit，选择 Source (源)，选择 Repositories (存储库)，并选择一个存储库。
   - 对于 CodeDeploy，请选择 Applications (应用程序)，然后选择一个应用程序。
   - 对于 CodePipeline，请选择 Pipeline (管道)，再选择 Pipelines (管道)，然后选择一个管道。
3. 在资源页面上，选择 Notify (通知)，然后选择 Create notification rule (创建通知规则)。您也可以转到资源的 Settings (设置) 页面，转到 Notifications (通知) 或 Notification rules (通知规则)，然后选择 Create notification rule (创建通知规则)。
4. 在 Notification name (通知名称) 中，输入规则的名称。
5. 如果您只想在通知中包含提供给 Amazon EventBridge 的信息，则在 Detail type (详细信息类型) 中，选择 Basic (基本)。如果您希望包含提供给 Amazon EventBridge 的信息以及资源服务或通知管理器可能提供的信息，选择 Full (完整)。

   有关更多信息，请参阅了解通知内容和安全性。
6. 在 Events that trigger notifications (触发通知的事件) 中，选择要为其发送通知的事件。有关资源的事件类型，请参阅以下内容：
   - CodeBuild：构建项目的通知规则的事件
   - CodeCommit：存储库的通知规则的事件
   - CodeDeploy：部署应用程序上的通知规则事件

- CodePipeline：管道的通知规则的事件

7. 在目标中，执行下列操作之一：

   - 如果您已将资源配置为与通知一起使用，请在选择目标类型中，选择 AWS Chatbot (Slack)、AWS Chatbot (Microsoft Teams) 或 SNS 主题。在 选择目标中，选择客户端的名称（对于在 AWS Chatbot 中配置的 Slack 或 Microsoft Teams 客户端）或 Amazon SNS 主题的 Amazon 资源名称（ARN）（对于已使用通知所需策略配置的 Amazon SNS 主题）。
   - 如果您尚未将资源配置为与通知一起使用，请选择 Create target (创建目标)，然后选择 SNS topic (SNS 主题)。在 codestar-notifications- 之后提供主题的名称，然后选择 Create (创建)。

   > **Note**
   >
   > - 如果您在创建通知规则的过程中创建 Amazon SNS 主题，则为您应用允许通知功能将事件发布到主题的策略。使用为通知规则创建的主题有助于确保您仅订阅要接收有关此资源的通知的那些用户。
   > - 您不能在创建通知规则的过程中创建 AWS Chatbot 客户端。如果您选择 AWS Chatbot (Slack) 或 AWS Chatbot (Microsoft Teams)，则将看到一个按钮，指示您在 AWS Chatbot 中配置客户端。选择该选项将打开 AWS Chatbot 控制台。有关更多信息，请参阅为 Slack 通道配置 AWS Chatbot 客户端。
   > - 如果要使用现有 Amazon SNS 主题作为目标，则在该主题可能存在的任何其他策略之外，您还必须为 AWS CodeStar 通知添加所需的策略。有关更多信息，请参阅 配置通知的 Amazon SNS 主题 和 了解通知内容和安全性：

8. 选择 Submit (提交)，然后查看通知规则。

   > **Note**
   >
   > 用户必须订阅并确认订阅您指定为规则目标的 Amazon SNS 主题，然后才能收到通知。有关更多信息，请参阅为用户订阅用于通知的 Amazon SNS 主题。

创建通知规则 (AWS CLI)

1. 在终端或命令提示符处，运行 create-notification rule 命令以生成 JSON 骨架。

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

您可以将此文件命名为所需的任意名称。在本示例中，文件命名为 *rule.json*。

2. 在纯文本编辑器中打开 JSON 文件，然后对其进行编辑，以包括该规则所需的资源、事件类型和 Amazon SNS 目标。

   以下示例显示了一个名为 **MyNotificationRule** 的通知规则，应用于 ID *123456789012* 的 AWS 账户名为 *MyDemoRepo* 的存储库。在创建分支和标签时，具有完整详情类型的通知将发送到名为 *MyNotificationTopic* 的 Amazon SNS 主题。

   ```
   {
       "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"
   }
   ```

   保存该文件。

3. 通过使用您刚编辑的文件，在终端或命令行上，再次运行 create-notification-rule 命令以创建通知规则。

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

4. 如果成功，该命令将返回通知规则的 ARN，类似于以下内容。

   ```
   {
   ```

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

列出通知规则的事件类型 (AWS CLI)

1. 在终端或命令提示符处，运行 list-event-types 命令。您可以使用 `--filters` 选项将响应限制为特定资源类型或其他属性。例如，以下内容返回 CodeDeploy 应用程序的事件类型列表。

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

2. 此命令生成类似于下述信息的输出。

```
{
    "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"
        }
    ]
}
```

向通知规则添加标签 (AWS CLI)

1. 在终端或命令提示符处，运行 tag-resource 命令。例如，使用以下命令添加名称为 *Team* 且值为 *Li_Juan* 的标签键值对。

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

2. 此命令生成类似于下述信息的输出。

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

查看通知规则

您可以使用开发工具控制台或 AWS CLI 查看 AWS 区域中所有资源的所有通知规则。您还可以查看每个通知规则的详细信息。与创建通知规则的过程不同，您不必转到资源的资源页面。

查看通知规则（控制台）

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。
2. 在导航栏中，展开设置，然后选择通知规则。
3. 在 Notification rules（通知规则）中，查看为您当前登录的 AWS 区域的 AWS 账户中的资源配置的规则列表。使用选择器更改 AWS 区域。
4. 要查看一个通知规则的详细信息，请从列表中选择该规则，然后选择 View details (查看详细信息)。您也可以仅在列表中选择其名称。

查看通知规则列表 (AWS CLI)

1. 在终端或命令提示符处，运行 list-notification-rules 命令可查看指定的 AWS 区域的所有通知规则。

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

2. 如果成功，此命令将为 AWS 区域中的每个通知规则返回 ID 和 ARN，类似于以下内容。

```
{
    "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"
        }
    ]
}
```

查看通知规则的详细信息 (AWS CLI)

1. 在终端或命令提示符处，运行 describe-notification-rule 命令，并指定通知规则的 ARN。

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

2. 如果成功，该命令将返回类似于以下内容的输出。

```
{
    "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"
}
```

查看通知规则的标签列表 (AWS CLI)

1. 在终端或命令提示符处，运行 list-tags-for-resource 命令可查看指定通知规则 ARN 的所有标签。

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

2. 如果成功，该命令返回类似以下内容的输出。

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

编辑通知规则

您可以通过编辑通知规则来更改其名称、其发送通知的事件、详细信息类型或其发送通知的一个或多个目标。您可以使用开发工具控制台或 AWS CLI 编辑通知规则。

编辑通知规则（控制台）

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。
2. 在导航栏中，展开设置，然后选择通知规则。
3. 在 Notification rules（通知规则）中，查看为您当前登录的 AWS 区域的 AWS 账户中的资源配置的规则。使用选择器更改 AWS 区域。

4. 从列表中选择规则，然后选择 Edit (编辑)。进行更改，然后选择 Submit (提交)。

编辑通知规则 (AWS CLI)

1. 在终端或命令提示符处，运行 describe-notification-rule 命令可查看通知规则的结构。
2. 运行 update-notification rule 命令可生成 JSON 骨架并将其保存到文件中。

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

您可以将此文件命名为所需的任意名称。在本示例中，文件为 *update.json*。

3. 在纯文本编辑器中打开 JSON 文件并更改规则。

以下示例显示了一个名为 **MyNotificationRule** 的通知规则，应用于 ID *123456789012* 的 AWS 账户名为 *MyDemoRepo* 的存储库。在创建分支和标记时，具有完整详情类型的通知将发送到名为 *MyNotificationTopic* 的 Amazon SNS 主题。规则名称将更改为 *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"
}
```

保存该文件。

4. 通过使用您刚编辑的文件，在终端或命令行上，再次运行 update-notification-rule 命令以更新通知规则。

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

5. 如果成功，该命令将返回通知规则的 Amazon 资源名称 (ARN)，类似于以下内容。

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

从通知规则中删除标签 (AWS CLI)

1. 在终端或命令提示符处，运行 untag-resource 命令。例如，以下命令删除名为 *Team* 的标签。

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

2. 如果成功，该命令不返回任何内容。

另请参阅

- 添加或删除通知规则的目标
- 为通知规则启用或禁用通知
- 事件

为通知规则启用或禁用通知

创建通知规则时，默认情况下会启用通知。您无需删除规则即可阻止其发送通知。您只需更改其通知状态。

更改通知规则的通知状态（控制台）

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。
2. 在导航栏中，展开设置，然后选择通知规则。
3. 在 Notification rules（通知规则）中，查看为您当前登录的 AWS 区域的 AWS 账户中的资源配置的规则。使用选择器更改 AWS 区域。

4. 找到要启用或禁用的通知规则，然后选择它以显示其详细信息。
5. 在 Notification status (通知状态) 中，选择滑块可更改规则的状态：
   - Sending notifications (发送通知)：这是默认值。
   - Notifications paused (通知已暂停)：不向指定目标发送任何通知。

更改通知规则的通知状态 (AWS CLI)

1. 执行 编辑通知规则 (AWS CLI) 中的步骤可获取通知规则的 JSON。
2. 将 `Status` 字段编辑为 `ENABLED`（默认值）或 `DISABLED`（无通知），然后运行 update-notification-rule 命令来更改状态。

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

删除通知规则

一个资源只能配置 10 个通知规则，因此，请考虑删除不再需要的规则。您可以使用开发工具控制台或 AWS CLI 删除通知规则。

> **Note**
> 您无法撤消删除通知规则的操作，但可以重新创建该规则。删除通知规则不会删除目标。

删除通知规则（控制台）

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。
2. 在导航栏中，展开设置，然后选择通知规则。
3. 在 Notification rules（通知规则）中，查看为您当前登录的 AWS 区域的 AWS 账户中的资源配置的规则。使用选择器更改 AWS 区域。
4. 选择通知规则，然后选择 Delete (删除)。
5. 键入 **`delete`**，然后选择删除。

删除通知规则 (AWS CLI)

1. 在终端或命令提示符处，运行 delete-notification-rule 命令，并指定通知规则的 ARN。

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

2. 如果成功，该命令将返回删除的通知规则的 ARN，类似于以下内容。

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

使用通知规则目标

通知规则目标是一个目标，用于定义在满足通知规则的事件条件时希望将通知发送到的位置。您可以在为 Slack 或 Microsoft Teams 通道配置的 Amazon SNS 主题和 AWS Chatbot 客户端之间进行选择。您可在创建通知规则的过程中，创建 Amazon SNS 主题作为目标（推荐）。您也可以选择与通知规则位于相同 AWS 区域中的现有 Amazon SNS 主题，但必须使用所需的策略对其进行配置。如果选择将 AWS Chatbot 客户端用作目标，则必须首先在 AWS Chatbot 中创建该客户端。

如果您要扩展通知的范围，可以手动配置通知和 AWS Chatbot 之间的集成，以便将通知发送到 Amazon Chime 聊天室。然后，您可以选择为该 AWS Chatbot 客户端配置的 Amazon SNS 主题作为通知规则的目标。有关更多信息，请参阅将通知与 AWS Chatbot 和 Amazon Chime 集成。

您可以使用开发工具控制台或 AWS CLI 管理通知规则。您可以使用控制台或 AWS CLI 创建 Amazon SNS 主题和 AWS Chatbot 客户端，并将其配置为目标。您还可以配置 Amazon SNS 主题（您将其配置为目标）与 AWS Chatbot 之间的集成。这使您可以将通知发送到 Amazon Chime 聊天室。有关更多信息，请参阅配置通知与 AWS Chatbot 之间的集成。

主题

- 创建或配置通知规则目标
- 查看通知规则目标
- 添加或删除通知规则的目标
- 删除通知规则目标

创建或配置通知规则目标

通知规则目标是为 Slack 或 Microsoft Teams 通道配置的 Amazon SNS 主题或 AWS Chatbot 客户端。

必须先创建 AWS Chatbot 客户端，然后才能选择客户端作为目标。当您选择 AWS Chatbot 客户端作为通知规则的目标时，会为该 AWS Chatbot 客户端配置一个 Amazon SNS 主题，其中包含将通知发送到 Slack 或 Microsoft Teams 通道所需的全部策略。您不必为 AWS Chatbot 客户端配置任何现有 Amazon SNS 主题。

在创建通知规则时，您可以在开发工具控制台中创建 Amazon SNS 通知规则目标。将为您应用允许将通知发送到该主题的策略。这是为通知规则创建目标的最简单方法。有关更多信息，请参阅创建通知规则。

如果您使用现有 Amazon SNS 主题，则必须使用允许资源向该主题发送通知的访问策略来配置该主题。有关示例，请参阅 配置通知的 Amazon SNS 主题。

> **Note**
>
> 如果要使用现有 Amazon SNS 主题而不是创建新主题，请在 Targets (目标) 中选择其 ARN。请确保主题具有适当的访问策略，并且订阅者列表仅包含允许查看有关资源的信息的用户。如果 Amazon SNS 主题是在 2019 年 11 月 5 日之前用于 CodeCommit 通知的，它将包含允许 CodeCommit 向其发布的策略，该策略包含的权限与 AWS CodeStar 通知所需的权限不同。建议不使用这些主题。如果要使用为该体验创建的策略，则除了已存在的策略之外，还必须添加 AWS CodeStar 通知所需的策略。有关更多信息，请参阅 配置通知的 Amazon SNS 主题 和 了解通知内容和安全性：

如果您要扩展通知的范围，可以手动配置通知和 AWS Chatbot 之间的集成，以便将通知发送到 Amazon Chime 聊天室。有关更多信息，请参阅 目标 和 将通知与 AWS Chatbot 和 Amazon Chime 集成：

配置现有 Amazon SNS 主题以用作通知规则目标（控制台）

1. 访问 https://console.aws.amazon.com/sns/v3/home，登录 AWS Management Console 并打开 Amazon SNS 控制台。
2. 在导航栏中，选择主题。选择主题，然后选择 Edit (编辑)。
3. 展开 Access policy (访问策略)，然后选择 Advanced (高级)。

4. 在 JSON 编辑器中，向策略中添加以下语句。包含主题 ARN、AWS 区域、AWS 账户 ID 和主题名称。

```
 {
      "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"
    }
```

该策略语句应与以下内容类似。

```
{
  "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",
        "SNS:Receive"
      ],
      "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. 选择 Save changes（保存更改）。
6. 在 Subscriptions (订阅) 中，查看主题订阅者的列表。添加、编辑或删除适用于此通知规则目标的订阅者。确保订阅者列表仅包含那些被允许查看有关资源的信息的用户。有关更多信息，请参阅了解通知内容和安全性。

创建与 Slack 结合使用的 AWS Chatbot 客户端作为目标

1. 按照《AWS Chatbot 管理员指南》中的设置 AWS Chatbot 与 Slack 结合使用中的说明进行操作。执行此操作时，请考虑以下选项以实现与通知的最佳集成：
   - 创建 IAM 角色时，请考虑选择一个角色名称，以便轻松识别此角色的用途（例如 **AWSCodeStarNotifications-Chatbot-Slack-Role**）。这可以帮助您确定未来角色的用途。
   - 在 SNS topics (SNS 主题) 中，您无需选择主题或 AWS 区域。选择 AWS Chatbot 客户端作为目标时，在通知规则创建过程中，将为 AWS Chatbot 客户端创建并配置具有所有必需权限的 Amazon SNS 主题。
2. 完成客户端创建过程。然后，您可以在创建通知规则时选择此客户端作为目标。有关更多信息，请参阅创建通知规则。

> **Note**
>
> 在为您配置 Amazon SNS 主题后，请勿将其从 AWS Chatbot 客户端中删除。这样做将阻止向 Slack 发送通知。

创建与 Microsoft Teams 结合使用的 AWS Chatbot 客户端作为目标

1. 按照《AWS Chatbot 管理员指南》中的设置 AWS Chatbot 与 Microsoft Teams 结合使用中的说明进行操作。执行此操作时，请考虑以下选项以实现与通知的最佳集成：
   - 创建 IAM 角色时，请考虑选择一个角色名称，以便轻松识别此角色的用途（例如 **`AWSCodeStarNotifications-Chatbot-Microsoft-Teams-Role`**）。这可以帮助您确定未来角色的用途。
   - 在 SNS topics (SNS 主题) 中，您无需选择主题或 AWS 区域。选择 AWS Chatbot 客户端作为目标时，在通知规则创建过程中，将为 AWS Chatbot 客户端创建并配置具有所有必需权限的 Amazon SNS 主题。
2. 完成客户端创建过程。然后，您可以在创建通知规则时选择此客户端作为目标。有关更多信息，请参阅创建通知规则。

> **Note**
>
> 在为您配置 Amazon SNS 主题后，请勿将其从 AWS Chatbot 客户端中删除。这样做将阻止向 Microsoft Teams 发送通知。

查看通知规则目标

您可以使用开发工具控制台而不是 Amazon SNS 控制台查看 AWS 区域中所有资源的所有通知规则目标。您也可以查看通知规则目标的详细信息。

查看通知规则目标（控制台）

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。
2. 在导航栏中，展开设置，然后选择通知规则。

3. 在 Notification rule targets（通知规则目标）中，查看您当前登录的 AWS 区域的 AWS 账户中的通知规则所使用的目标列表。使用选择器更改 AWS 区域。如果目标状态显示为 Unreachable (无法访问)，您可能需要进行调查。有关更多信息，请参阅问题排查 。

查看通知规则目标的列表 (AWS CLI)

1. 在终端或命令提示符处，运行 list-targets 命令可查看指定的 AWS 区域的所有通知规则目标的列表：

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

2. 如果成功，此命令将为 AWS 区域中的每个通知规则返回 ID 和 ARN，类似于以下内容：

```
{
    "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"
        }
    ]
}
```

添加或删除通知规则的目标

您可以编辑通知规则以更改其发送通知的一个或多个目标。您可以使用开发工具控制台或 AWS CLI 更改通知规则目标。

更改通知规则的目标（控制台）

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。
2. 在导航栏中，展开设置，然后选择通知规则。
3. 在 Notification rules（通知规则）中，查看为您当前登录的 AWS 区域的 AWS 账户中的资源配置的规则列表。使用选择器更改 AWS 区域。
4. 选择规则，然后选择 Edit (编辑)。
5. 在目标中，执行下列操作之一：
   - 要添加另一个目标，请选择添加目标，然后从列表中选择要添加的 Amazon SNS 主题或 AWS Chatbot (Slack) 或 AWS Chatbot (Microsoft Teams) 客户端。您还可以选择 Create SNS topic (创建 SNS 主题) 来创建一个主题并将该主题添加为目标。一个通知规则最多可具有 10 个目标。
   - 要删除目标，请选择要删除的目标旁边的 Remove target (删除目标)。
6. 选择 Submit（提交）。

向通知规则添加目标 (AWS CLI)

1. 在终端或命令提示符处，运行 subscribe 命令以添加目标。例如，以下命令添加 Amazon SNS 主题作为通知规则的目标。

```
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. 如果成功，该命令将返回更新的通知规则的 ARN，类似于以下内容。

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

从通知规则中删除目标 (AWS CLI)

1. 在终端或命令提示符处，运行 unsubscribe 命令以删除目标。例如，以下命令删除作为通知规则目标的 Amazon SNS 主题。

```
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. 如果成功，该命令将返回更新的通知规则的 ARN 以及有关删除的目标的信息，类似于以下内容。

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

另请参阅

- 编辑通知规则
- 为通知规则启用或禁用通知

删除通知规则目标

如果不再需要某个目标，您可以删除该目标。一个资源只能配置 10 个通知规则目标，因此删除不需要的目标可以帮助为您可能想要添加到该通知规则的其他目标创造空间。

> **Note**
>
> 删除通知规则目标会从所有配置为将其用作目标的通知规则中删除该目标，但不会删除目标本身。

删除通知规则目标（控制台）

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。
2. 在导航栏中，展开设置，然后选择通知规则。

3. 在 Notification rule targets（通知规则目标）中，查看为您当前登录的 AWS 区域的 AWS 账户中的资源配置的目标列表。使用选择器更改 AWS 区域。
4. 选择该通知规则目标，然后选择 Delete (删除)。
5. 键入 **delete**，然后选择删除。

删除通知规则目标 (AWS CLI)

1. 在终端或命令提示符处，运行 delete-target 命令，并指定目标的 ARN。例如，以下命令删除使用 Amazon SNS 主题的目标。

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

2. 如果成功，该命令不返回任何内容。如果不成功，该命令会返回错误。最常见的错误是该主题是一个或多个通知规则的目标。

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

您可以使用 `--force-unsubscribe-all` 参数从配置为使用它作为目标的所有通知规则中移除此目标，然后删除目标。

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

配置通知与 AWS Chatbot 之间的集成

AWS Chatbot 是一项 AWS 服务，该服务使 DevOps 和软件开发团队能够使用 Amazon Chime 聊天室、Slack 通道和 Microsoft Teams 通道来监控和响应 AWS Cloud 中的操作事件。您可以配置通知规则目标与 AWS Chatbot 之间的集成，以便有关事件的通知显示在您选择的 Amazon Chime 聊天室、Slack 通道和 Microsoft Teams 通道中。有关更多信息，请参阅 AWS Chatbot 文档。

在配置与 AWS Chatbot 的集成之前，您必须配置通知规则和规则目标。有关更多信息，请参阅 设置 和 创建通知规则：您还必须在 AWS Chatbot 中配置 Slack 通道、Microsoft Teams 通道或 Amazon Chime 聊天室。有关更多信息，请参阅这些服务的文档。

主题

- 为 Slack 通道配置 AWS Chatbot 客户端
- 为 Microsoft Teams 通道配置 AWS Chatbot 客户端
- 为 Slack 或 Amazon Chime 手动配置客户端

为 Slack 通道配置 AWS Chatbot 客户端

您可以创建使用 AWS Chatbot 客户端作为目标的通知规则。如果您为 Slack 通道创建客户端，则可以在工作流中直接使用此客户端作为创建通知规则的目标。这是设置在 Slack 通道中显示的通知的最简单方法。

创建与 Slack 结合使用的 AWS Chatbot 客户端作为目标

1. 按照《AWS Chatbot 管理员指南》中的设置 AWS Chatbot 与 Slack 结合使用中的说明进行操作。执行此操作时，请考虑以下选项以实现与通知的最佳集成：
   - 创建 IAM 角色时，请考虑选择一个角色名称，以便轻松识别此角色的用途（例如 **`AWSCodeStarNotifications-Chatbot-Slack-Role`**）。这可以帮助您确定未来角色的用途。
   - 在 SNS topics (SNS 主题) 中，您无需选择主题或 AWS 区域。选择 AWS Chatbot 客户端作为目标时，在通知规则创建过程中，将为 AWS Chatbot 客户端创建并配置具有所有必需权限的 Amazon SNS 主题。
2. 完成客户端创建过程。然后，您可以在创建通知规则时选择此客户端作为目标。有关更多信息，请参阅创建通知规则。

> **Note**
>
> 在为您配置 Amazon SNS 主题后，请勿将其从 AWS Chatbot 客户端中删除。这样做将阻止向 Slack 发送通知。

为 Microsoft Teams 通道配置 AWS Chatbot 客户端

您可以创建使用 AWS Chatbot 客户端作为目标的通知规则。如果您为 Microsoft Teams 通道创建客户端，则可以在工作流中直接使用此客户端作为创建通知规则的目标。这是设置在 Microsoft Teams 通道中显示的通知的最简单方法。

创建与 Microsoft Teams 结合使用的 AWS Chatbot 客户端作为目标

1. 按照《AWS Chatbot 管理员指南》中的设置 AWS Chatbot 与 Microsoft Teams 结合使用中的说明进行操作。执行此操作时，请考虑以下选项以实现与通知的最佳集成：

   - 创建 IAM 角色时，请考虑选择一个角色名称，以便轻松识别此角色的用途（例如 **`AWSCodeStarNotifications-Chatbot-Microsoft-Teams-Role`**）。这可以帮助您确定未来角色的用途。
   - 在 SNS topics (SNS 主题) 中，您无需选择主题或 AWS 区域。选择 AWS Chatbot 客户端作为目标时，在通知规则创建过程中，将为 AWS Chatbot 客户端创建并配置具有所有必需权限的 Amazon SNS 主题。

2. 完成客户端创建过程。然后，您可以在创建通知规则时选择此客户端作为目标。有关更多信息，请参阅创建通知规则。

> **Note**
>
> 在为您配置 Amazon SNS 主题后，请勿将其从 AWS Chatbot 客户端中删除。这样做将阻止向 Microsoft Teams 发送通知。

为 Slack 或 Amazon Chime 手动配置客户端

您可以选择在通知和 Slack 或 Amazon Chime 之间直接创建集成。这是可用于配置 Amazon Chime 聊天室通知的唯一方法。手动配置此集成时，您将创建一个 AWS Chatbot 客户端，该客户端使用之前配置作为通知规则目标的 Amazon SNS 主题。

手动将通知与 AWS Chatbot 和 Slack 集成

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。
2. 选择 Settings (设置)，然后选择 Notification rules (通知规则)。
3. 在 Notification rule targets (通知规则目标) 中，查找并复制目标。

> **Note**
>
> 您可以配置多个通知规则，以将同一 Amazon SNS 主题用作其目标。这可以帮助您整合消息传递，但如果订阅列表旨在用于一个通知规则或资源，则可能产生意想不到的后果。

4. 打开 https://console.aws.amazon.com/chatbot/ 上的 AWS Chatbot 控制台。
5. 选择 Configure new client (配置新客户端)，然后选择 Slack。
6. 选择 Configure（配置）。
7. 登录您的 Slack 工作区。
8. 当系统提示您确认选择时，请选择 Allow (允许)。
9. 选择 Configure new channel (配置新通道)。
10. 在 Configuration details (配置详细信息) 的 Configuration name (配置名称) 中，输入客户端的名称。这是在您创建通知规则时，将显示在对 AWS Chatbot (Slack) 目标类型可用的目标列表中的名称。
11. 在 Configure Slack Channel（配置 Slack 通道）的 Channel type（通道类型）中，根据要集成的通道类型选择 Public（公有）或 Private（私有）。
    - 在 Public channel (公有通道) 中，从列表中选择 Slack 通道的名称。
    - 在 Private channel ID (私有通道 ID) 中，输入通道代码或 URL。
12. 在 IAM permissions (IAM 权限) 的 Role (角色) 中，选择 Create an IAM role using a template (使用模板创建 IAM 角色)。在 Policy templates (策略模板) 中，选择 Notification permissions (通知权限)。在 Role name (角色名称) 中，输入此角色的名称（例如 **`AWSCodeStarNotifications-Chatbot-Slack-Role`**）。在 Policy templates (策略模板) 中，选择 Notification permissions (通知权限)。
13. 在 SNS topics（SNS 主题）的 SNS Region（SNS 区域）中，选择您在其中创建通知规则目标的 AWS 区域。在 SNS topics (SNS 主题) 中，选择您配置为通知规则目标的 Amazon SNS 主题的名称。

> **Note**
>
> 如果您将创建使用此客户端作为目标的通知规则，则无需执行此步骤。

14. 选择 Configure（配置）。

> **Note**
>
> 如果您配置了与私有通道的集成，则必须先将 AWS Chatbot 邀请到通道，然后才能在该通道中看到通知。有关更多信息，请参阅 AWS Chatbot 文档。

15. （可选）要测试集成，请在与通知规则（配置为使用 Amazon SNS 主题作为其目标）的事件类型匹配的资源中进行更改。例如，如果将通知规则配置为在对拉取请求进行注释时发送通知，请对拉取请求进行注释，然后在浏览器中观看 Slack 通道以查看通知何时出现。

将通知与 AWS Chatbot 和 Amazon Chime 集成

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。
2. 选择 Settings (设置)，然后选择 Notification rules (通知规则)。
3. 在 Notification rule targets (通知规则目标) 中，查找并复制目标。

> **Note**
>
> 您可以配置多个通知规则，以将同一 Amazon SNS 主题用作其目标。这可以帮助您整合消息传递，但如果订阅列表旨在用于一个通知规则或资源，则可能产生意想不到的后果。

4. 在 Amazon Chime 中，打开要配置以进行集成的聊天室。
5. 选择右上角的齿轮图标，然后选择 Manage webhooks (管理 Webhook)。
6. 在 Manage webhooks (管理 Webhook) 对话框中，选择 New (新建)，输入 Webhook 的名称，然后选择 Create (创建)。
7. 确认 Webhook 已出现，然后选择 Copy webhook URL (复制 Webhook URL)。
8. 打开 https://console.aws.amazon.com/chatbot/ 上的 AWS Chatbot 控制台。
9. 选择 Configure new client (配置新客户端)，然后选择 Amazon Chime。
10. 在 Configuration details (配置详细信息) 的 Configuration name (配置名称) 中，输入客户端的名称。
11. 在 Webhook URL 中，粘贴该 URL。在 Webhook description (Webhook 描述) 中，提供可选的描述。
12. 在 IAM permissions (IAM 权限) 的 Role (角色) 中，选择 Create an IAM role using a template (使用模板创建 IAM 角色)。在 Policy templates (策略模板) 中，选择 Notification permissions (通知权限)。在 Role name (角色名称) 中，输入此角色的名称（例如 **`AWSCodeStarNotifications-Chatbot-Chime-Role`**）。
13. 在 SNS topics（SNS 主题）的 SNS Region（SNS 区域）中，选择您在其中创建通知规则目标的 AWS 区域。在 SNS topics (SNS 主题) 中，选择您配置为通知规则目标的 Amazon SNS 主题的名称。
14. 选择 Configure（配置）。

15. （可选）要测试集成，请在与通知规则（配置为使用 Amazon SNS 主题作为其目标）的事件类型匹配的资源中进行更改。例如，如果将通知规则配置为在对拉取请求进行注释时发送通知，请对拉取请求进行注释，然后观看 Amazon Chime 聊天室以查看通知何时出现。

使用 AWS CloudTrail 记录 AWS CodeStar 通知 API 调用

AWS CodeStar 通知与 AWS CloudTrail 集成，后者是一项服务，提供由用户、角色或 AWS 服务所采取操作的记录。CloudTrail 将所有通知的 API 调用作为事件记录。记录的调用包含来自开发工具控制台的调用和对 AWS CodeStar 通知 API 操作的代码调用。如果您创建跟踪记录，则可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶（包括通知的事件）。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的 Event history（事件历史记录）中查看最新事件。通过使用 CloudTrail 收集的信息，您可以确定向 AWS CodeStar 通知发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

有关更多信息，请参阅 AWS CloudTrail 用户指南。

CloudTrail 中的 AWS CodeStar 通知信息

在您创建 AWS 账户 时，将在该账户上启用 CloudTrail。当 AWS CodeStar 通知中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在 Event history (事件历史记录) 中。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅使用 CloudTrail 事件历史记录查看事件。

要持续记录 AWS 账户中的事件（包括 AWS CodeStar 通知的事件），请创建跟踪记录。通过跟踪记录，CloudTrail 可将日志文件传送至 Simple Storage Service（Amazon S3）存储桶。预设情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有AWS 区域。此跟踪记录在 AWS 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Simple Storage Service（Amazon S3）桶。此外，您可以配置其他 AWS 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- 创建跟踪概览
- CloudTrail 支持的服务和集成
- 为 CloudTrail 配置 Amazon SNS 通知
- 从多个区域接收 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件

CloudTrail 记录所有 AWS CodeStar 通知操作，*AWS CodeStar ## API ##*中介绍了这些操作。例如，对 `CreateNotificationRule`、`Subscribe` 和 `ListEventTypes` 操作的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 AWS Identity and Access Management（IAM）用户凭证发出的。
- 请求是使用角色还是联合身份用户的临时安全凭证发出的。
- 请求是否由其它 AWS 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

以下示例显示一个 CloudTrail 日志条目，该条目演示如何创建通知规则，包括 `CreateNotificationRule` 和 `Subscribe` 操作。

> **Note**
>
> 通知日志文件条目中的一些事件可能来自服务相关角色 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"
}
```

问题排查

以下信息可帮助您解决有关通知的常见问题。

主题

- 当我尝试在资源上创建通知规则时出现权限错误
- 我无法查看通知规则
- 我无法创建通知规则
- 我收到我无法访问的资源的通知
- 我未收到 Amazon SNS 通知
- 我接收重复的事件通知
- 我想了解为什么通知目标状态显示为“Unreachable (无法到达)”
- 我想增大通知和资源的配额

当我尝试在资源上创建通知规则时出现权限错误

请确保您有足够的权限。有关更多信息，请参阅 基于身份的策略示例。

我无法查看通知规则

问题：当您在开发工具控制台中选择 Settings (设置) 下的 Notifications (通知)时，您会看到一个权限错误。

可能的修复措施：您可能没有查看通知所需的权限。虽然 AWS 开发工具服务的大多数托管式策略（如 CodeCommit 和 CodePipeline）都包含通知权限，但当前不支持通知的服务不包括查看通知的权限。或者，您可能会将自定义策略应用于您的 IAM 用户或角色，该策略不允许您查看通知。有关更多信息，请参阅 基于身份的策略示例。

我无法创建通知规则

您可能没有创建通知规则所需的权限。有关更多信息，请参阅 基于身份的策略示例。

我收到我无法访问的资源的通知

在创建通知规则并添加目标时，通知功能不会验证收件人是否有权访问资源。您可能会收到有关您无法访问的资源的通知。如果您无法删除自己，则要求从目标的订阅列表中予以删除。

我未收到 Amazon SNS 通知

要排查 Amazon SNS 主题的问题，请检查以下内容：

- 确保已在通知规则所在的 AWS 区域中创建 Amazon SNS 主题。
- 确保您的电子邮件别名已订阅到正确的主题，并且您已确认订阅。有关更多信息，请参阅将终端节点订阅到 Amazon SNS 主题。
- 验证是否已编辑主题策略以允许 AWS CodeStar 通知将通知推送到该主题。该主题策略应包含类似于以下内容的声明：

```
{
    "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"
        }
    }
}
```

有关更多信息，请参阅配置通知的 Amazon SNS 主题。

我接收重复的事件通知

以下是收到多个通知的最常见原因：

- 已为资源配置多个包含同一事件类型的通知规则，并且已为您订阅作为这些规则的目标的 Amazon SNS 主题。要解决此问题，请取消订阅其中一项主题，或者编辑通知规则以删除重复项。
- 一个或多个通知规则目标与 AWS Chatbot 集成，并且您正在您的电子邮件收件箱以及 Slack 通道、Microsoft Teams 通道或 Amazon Chime 聊天室中接收通知。要解决此问题，请考虑从作为该规则的目标的 Amazon SNS 主题中取消订阅电子邮件地址，并使用 Slack 通道、Microsoft Teams 通道或 Amazon Chime 聊天室查看通知。

我想了解为什么通知目标状态显示为“Unreachable (无法到达)”

目标具有两种可能的状态：Active (活动) 状态和 Unreachable (无法访问)。Unreachable (无法访问) 表示已向目标发送通知，但传递不成功。通知将继续发送到该目标，如果成功，状态将重置为 Active (活动)。

由于以下原因之一，通知规则的目标可能变得不可用：

- 已删除资源（Amazon SNS 主题或 AWS Chatbot 客户端）。为通知规则选择另一个目标。
- Amazon SNS 主题已加密，并且缺少加密主题所需的策略，或者 AWS KMS 密钥已删除。有关更多信息，请参阅配置通知的 Amazon SNS 主题。
- Amazon SNS 主题没有通知所需的策略。除非 Amazon SNS 主题具有策略，否则无法向其发送通知。有关更多信息，请参阅配置通知的 Amazon SNS 主题。
- 目标（Amazon SNS 或 AWS Chatbot）的支持服务可能遇到问题。

我想增大通知和资源的配额

目前，您无法更改任何配额。请参阅通知的配额。

通知的配额

下表列出了开发工具控制台中通知的配额（也称为限制)。有关可以更改的限制的信息，请参阅 AWS 服务限额。

资源	默认限制
AWS 账户中的最大通知规则数	1000
通知规则的最大目标数	10
资源的最大通知规则数	10

什么是连接？

您可以使用开发者工具控制台中的连接功能将诸如外部代码存储库之类 AWS CodePipeline 的 AWS 资源连接起来。此功能有自己的 API，即AWS CodeStar 连接 API 参考。每个连接都是您可以提供给 AWS 服务的资源，用于连接到第三方存储库，例如 BitBucket。例如，您可以在中添加连接，CodePipeline 以便在对第三方代码存储库进行代码更改时，它会触发您的管道。每个连接都被命名并与用于引用连接的唯一 Amazon Resource Name (ARN) 相关联。

我可以使用连接执行哪些操作？

您可以使用连接将第三方提供商资源与开发人员工具中的 AWS 资源集成，包括：

- 连接到第三方提供商（例如 Bitbucket），并使用第三方连接作为与您的 AWS 资源的源集成，例如。 CodePipeline
- 在第三方提供商的 CodeBuild 构建项目、 CodeDeploy 应用程序和管道中，统一管理跨资源 CodePipeline 对连接的访问权限。
- 在堆栈模板中使用连接 ARN 来 CodeBuild 构建项目、 CodeDeploy应用程序和管道 CodePipeline，无需引用存储的机密或参数。

我可以为哪些第三方提供商创建连接？

Connections 可以将您的 AWS 资源与以下第三方存储库相关联：

- Bitbucket Cloud
- GitHub
- GitHub 企业云
- GitHub 企业服务器

- GitLab
- GitLab 自行管理安装（适用于企业版或社区版）

有关连接工作流的概述，请参阅创建或更新连接的工作流。

为云提供商类型（例如）创建连接的步骤与为 GitHub已安装的提供商类型（例如 En GitHub terprise Server）创建连接的步骤不同。有关按提供程序类型创建连接的概要步骤，请参阅使用连接。

> **Note**
>
> 要在欧洲（米兰）使用连接 AWS 区域，您必须：
>
> 1. 安装区域特定的应用程序
> 2. 启用该区域
>
> 这一特定于区域的应用程序支持欧洲地区（米兰）区域中的连接。该应用程序在第三方提供商网站上发布，与支持其他区域的连接的现有应用程序是分开的。安装此应用程序，即表示您授权第三方提供商仅与该区域的服务共享您的数据，并且您可以随时通过卸载该应用程序来撤消权限。
> 除非您启用区域，否则该服务不会处理或存储您的数据。启用此区域，即表示您授予我们的服务处理和存储您的数据的权限。
> 即使未启用该区域，如果区域特定的应用程序仍保持安装状态，第三方提供商也仍可以与我们的服务共享您的数据，因此，请务必在禁用该区域后立即卸载该应用程序。有关更多信息，请参阅启用区域。

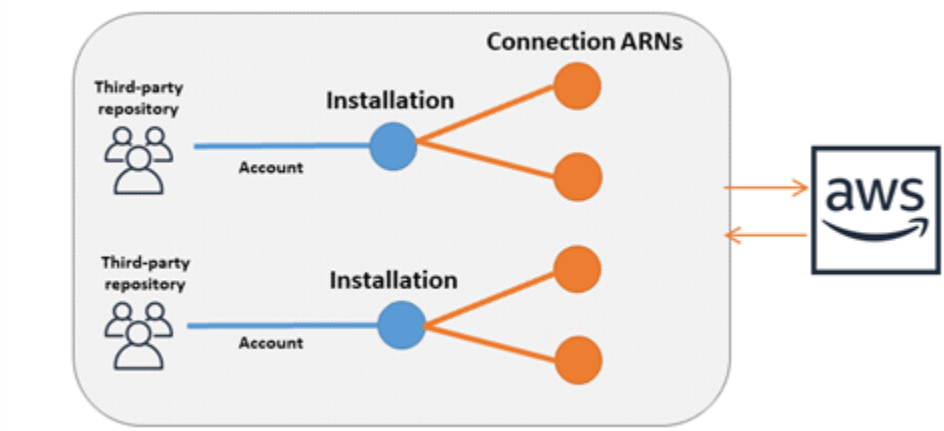
什么与连接 AWS 服务 集成？

您可以使用连接将您的第三方存储库与其他 AWS 服务集成。要查看连接的服务集成，请参阅产品和服务与 AWS CodeStar Connections 的集成。

连接是如何工作的？

在创建连接之前，您必须首先在您的第三方账户上安装 AWS 身份验证应用程序或提供对该应用程序的访问。安装连接后，可以更新它以使用此安装。创建连接时，您可以提供对第三方账户中 AWS 资源的访问。这允许连接代表您的 AWS 资源访问第三方账户中的内容，例如源存储库。然后，您可以与其他人共享该连接， AWS 服务 以便在资源之间提供安全的 OAuth 连接。

如果要创建与已安装的提供程序类型（例如 En GitHub terprise Server）的连接，则首先要使用创建主机资源 AWS Management Console。



连接归创建 AWS 账户 它们的人所有。连接由包含连接 ID 的 ARN 标识。连接 ID 是无法更改或重新映射的 UUID。删除和重新建立连接会生成一个新的连接 ID，因此会产生一个新的连接 ARN。这意味着连接 ARN 绝不会重复使用。

新创建的连接处于 Pending 状态。需要第三方握手（OAuth 流）流程才能完成连接的设置，并将其从 Pending 变为 Available 状态。完成后，可以将连接与 AWS 服务一起使用，例如 CodePipeline。Available

新创建的主机处于 Pending 状态。需要第三方注册流程才能完成主机的设置，并将其从 Pending 变为 Available 状态。完成此操作后，主机将为 Available，可用于连接到已安装的提供程序类型。

有关连接工作流的概述，请参阅 创建或更新连接的工作流。有关已安装提供程序的主机创建工作流程的概述，请参阅创建或更新主机的工作流程。有关按提供程序类型创建连接的概要步骤，请参阅使用连接。

《AWS CodeStar 连接》中的全球资源

连接是全局资源，这意味着资源将在所有 AWS 区域间复制。

虽然连接 ARN 格式反映了其创建位置的区域名称，但资源不受限于任何区域。创建连接资源的区域是控制连接资源数据更新的区域。控制连接资源数据更新的 API 操作示例包括创建连接、更新安装、删除连接或标记连接。

用于连接的主机资源不是全局可用的资源。您只在创建主机资源的区域中使用主机资源。

- 您只需创建一次连接，然后您就可以在任何 AWS 区域区域中使用它。

- 如果创建连接的区域存在问题，这会影响控制连接资源数据的 API，但您仍然可以在其他每个区域中成功使用该连接。
- 当您在控制台或 CLI 中列出连接资源时，列表会显示所有区域中与您的账户关联的所有连接资源。
- 当您在控制台或 CLI 中列出主机资源时，列表仅显示选定区域中与您的账户关联的主机资源。
- 当使用 CLI 列出或查看与关联主机资源的连接时，输出将返回主机 ARN，而不考虑配置的 CLI 区域如何。

创建或更新主机的工作流程

为已安装的提供程序创建连接时，首先要创建一个主机。

主机可具有以下状态：

- `Pending` - `pending` 主机是指已创建的主机，必须先对其进行设置（移至 `available`），然后才能使用。
- `Available` - 您可以使用 `available` 主机或将其传递给您的连接。

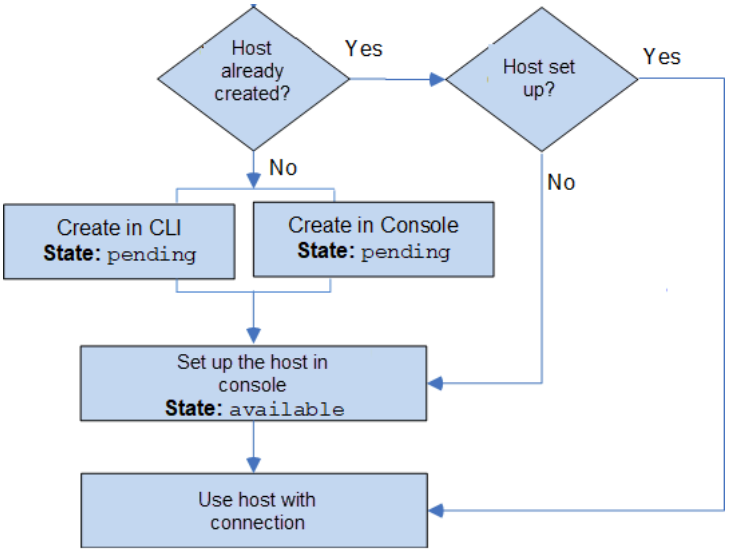
工作流程：使用 CLI、SDK 或 AWS CloudFormation创建或更新主机

您可以使用 CreateHostAPI 使用 AWS Command Line Interface (AWS CLI)、SDK 或创建主机 AWS CloudFormation。创建后，主机处于 `pending` 状态。您可以使用控制台中的设置选项完成该过程。

工作流程：使用控制台创建或更新主机

如果要创建与已安装的提供商类型（例如 GitHub企业服务器或 GitLab 自行管理）的连接，则需要先创建主机。如果要连接到云提供程序类型（如 Bitbucket），则跳过创建主机并继续创建连接。

使用控制台设置主机并将其状态从 `pending` 更改为 `available`。



创建或更新连接的工作流

创建连接时，您还可以创建或使用现有安装来与第三方提供程序进行身份验证握手。

连接可能为以下状态之一：

- `Pending` - `pending` 连接是必须先完成（移动到 `available`）然后才能使用的连接。
- `Available` - 您可以使用 `available` 连接或将其传递到账户中的其他资源和用户。
- `Error` - 具有 `error` 状态的连接会自动重试。在变为 `available` 前此连接无法使用。

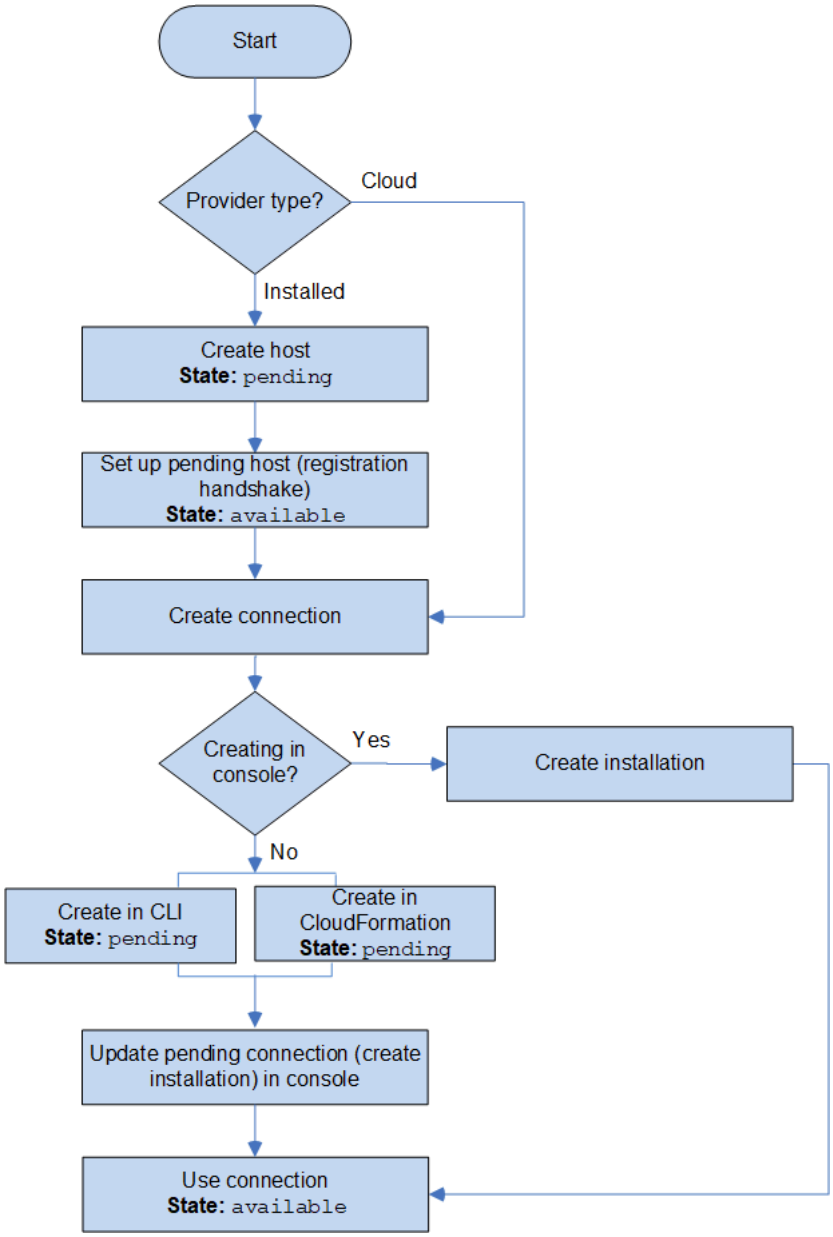
工作流程：创建或更新连接，使用 CLI、SDK 或 AWS CloudFormation

您可以使用 CreateConnectionAPI 通过 AWS Command Line Interface (AWS CLI)、SDK 或创建连接 AWS CloudFormation。创建后，连接处于 `pending` 状态。您可以使用控制台的 Set up pending connection (设置待处理的连接) 选项完成该过程。控制台会提示您创建安装或使用现有安装进行连接。然后，您可以使用控制台完成握手，并选择控制台上的 Complete connection (完成连接) 将连接变为 `available` 状态。

工作流：使用控制台创建或更新连接

如果要创建与已安装的提供程序类型（例如 GitHub企业服务器）的连接，则需要先创建主机。如果要连接到云提供程序类型（如 Bitbucket），则跳过创建主机并继续创建连接。

要使用控制台创建或更新连接，您可以使用控制台上的 CodePipeline 编辑操作页面来选择您的第三方提供商。控制台会提示您创建安装或使用现有安装进行连接，然后使用控制台创建连接。控制台完成握手并自动将连接从 `pending` 设置为 `available` 状态。



如何开始使用连接？

要开始使用通知，请先回顾此处提供的一些实用主题：

- 学习关于连接的概念。
- 设置开始使用连接所需的资源。
- 开始尝试使用您的第一批连接，并将其连接到资源。

连接概念

如果您了解概念和术语，则可更轻松地设置和使用连接特征。以下是使用开发工具控制台中的连接时要了解的一些概念：

安装

第三方账户中一个 AWS 应用程序的实例。安装 AWS CodeStar 连接器应用程序使 AWS 可以访问第三方账户中的资源。只能在第三方提供商的网站上编辑安装。

连接

用于将第三方源存储库连接到其他 AWS 服务的 AWS 资源。

第三方存储库

由不属于 AWS 的服务或公司提供的存储库。例如，BitBucket 存储库是第三方存储库。

提供商类型

提供要连接到的第三方源存储库的服务或公司。您可以连接 AWS 资源到外部提供商类型。源存储库安装在网络和基础设施上的提供商类型是已安装的提供程序类型。例如，GitHub Enterprise Server 是已安装的提供程序类型。

host

表示安装第三方提供程序的基础设施的资源。连接使用主机来表示安装第三方提供程序的服务器，例如 GitHub Enterprise Server。您可以为到该提供程序类型的所有连接创建一个主机。

> **Note**
>
> 使用控制台创建到 GitHub Enterprise Server 的连接时，作为流程的一部分，控制台会为您创建主机资源。

AWS CodeStar 连接支持的提供程序和版本

本章提供有关 C AWS CodeStar onnections 支持的提供程序和版本的信息。

主题

- Bitbucket 支持的提供程序类型
- GitHub 和 GitHub 企业云支持的提供商类型

- GitHub 企业服务器支持的提供程序类型和版本
- 支持的提供商类型 GitLab
- GitLab 自我管理支持的提供商类型

Bitbucket 支持的提供程序类型

你可以在 Atlassian Bitbucket Cloud 上使用该 AWS CodeStar 应用程序。

不支持已安装的 Bitbucket 提供程序类型（如 Bitbucket 服务器）。

GitHub 和 GitHub 企业云支持的提供商类型

您可以将 GitHub 应用程序 AWS 连接器 GitHub 与 GitHub 企业云一起使用。

GitHub 企业服务器支持的提供程序类型和版本

您可以将该 AWS CodeStar 应用程序与支持的 GitHub 企业服务器版本一起使用。有关受支持的版本的列表，请参阅 https://enterprise.github.com/releases/。

> **Important**
>
> AWS CodeStar Connections 不支持已弃用的 GitHub 企业服务器版本。例如，由于版本中存在已知问题，Conn AWS CodeStar ections 不支持 GitHub 企业服务器版本 2.22.0。要执行连接，请升级到版本 2.22.1 或最新的可用版本。

支持的提供商类型 GitLab

您可以将连接与配合使用 GitLab。有关更多信息，请参阅 创建与的连接 GitLab。

GitLab 自我管理支持的提供商类型

您可以将连接与 GitLab 自行管理安装（适用于企业版或社区版）配合使用。有关更多信息，请参阅 创建与 GitLab自我管理的连接。

产品和服务与 AWS CodeStar Connections 的集成

AWS CodeStar Connections 与许多 AWS 服务以及合作伙伴的产品和服务集成。使用以下部分中的信息来帮助您将连接配置为与您使用的产品和服务集成。

下列相关资源在您使用此服务的过程中会有所帮助。

主题

- Amazon CodeGuru Reviewer
- Amazon CodeWhisperer
- Amazon SageMaker
- AWS App Runner
- AWS CloudFormation
- AWS CodePipeline
- AWS CodeStar
- 服务目录
- AWS Proton

Amazon CodeGuru Reviewer

CodeGuru Reviewer 是一项用于监控存储库代码的服务。您可以使用连接来关联包含您要查看的代码的第三方存储库。有关学习如何配置 CodeGuru Reviewer 以监控 GitHub 存储库中的源代码，以便创建可改进代码的建议的教程，请参阅 Amazon CodeGuru Reviewer 用户指南 中的教程：监控 GitHub 存储库中的源代码。

Amazon CodeWhisperer

Amazon CodeWhisperer 是一项用于审查存储库代码的服务。CodeWhisperer 实时审查您的代码并为您提供代码建议。有关在 CodeWhisperer 中配置旨在使用连接来访问数据来源的自定义项的步骤，请参阅《Amazon CodeWhisperer 用户指南》中的创建自定义项。

Amazon SageMaker

Amazon SageMaker 是一项用于构建、训练和部署机器学习语言模型的服务。有关配置与 GitHub 存储库的连接的教程，请参阅《Amazon SageMaker 开发人员指南》中的使用第三方 Git 存储库进行 SageMaker MLOps 项目演练。

AWS App Runner

AWS App Runner 服务提供了一种快速、简单且经济高效的方式，从源代码或容器镜像直接部署到 AWS Cloud 中可扩展且安全的 Web 应用程序。您可以使用 App Runner 自动集成和交付管道从存储库

部署应用程序代码。您可以使用连接将源代码从私有 GitHub 存储库部署到 App Runner 服务。有关更多信息，请参阅 AWS App Runner 开发人员指南 中的源代码存储库提供商。

AWS CloudFormation

AWS CloudFormation 是一项服务，可帮助您对 AWS 资源进行建模和设置，以便能花较少的时间管理这些资源，而将更多的时间花在运行于 AWS 中的应用程序上。您创建一个描述您所需的所有 AWS 资源（如 Amazon EC2 实例或 Amazon RDS 数据库实例）的模板，并且 CloudFormation 将负责为您预置和配置这些资源。有关更多信息，请参阅《CloudFormation 命令行界面用户指南》中的注册账户以发布 CloudFormation 扩展。

AWS CodePipeline

CodePipeline 是一种持续交付服务，可用于对发布软件所需的步骤进行建模、可视化和自动化。您可以使用连接为 CodePipeline 源代码操作配置第三方存储库。

了解更多：

- 请参阅 `CodeStarSourceConnection` 操作的 CodePipeline 操作配置参考页面。要查看配置参数和 JSON/YAML 代码段示例，请参阅 AWS CodePipeline 用户指南 中的 CodeStarSourceConnection。
- 要查看使用第三方源存储库创建管道的入门 教程，请参阅开始使用连接。

AWS CodeStar

AWS CodeStar 是一项基于云的服务，用于在 AWS 上创建、管理和使用软件开发项目。您可以使用 AWS CodeStar 项目在 AWS 上快速开发、构建和部署应用程序。您可以使用连接来为您的 AWS CodeStar 项目中的管道配置第三方存储库。有关创建连接到 GitHub 存储库的 AWS CodeStar 项目的教程，请参阅《AWS CodeStar 用户指南》中的创建指向存储库的链接。

服务目录

利用 Service Catalog，组织可以创建和管理已批准在 AWS 上使用的产品的目录。

当您向 AWS 账户与外部存储库提供商（例如 GitHub、GitHub Enterprise 或 BitBucket）之间的连接授权时，该连接允许您将 Service Catalog 产品同步到通过第三方存储库管理的模板文件。

有关更多信息，请参阅《Service Catalog 用户指南》中的将 Service Catalog 产品同步到来自 GitHub、GitHub Enterprise 或 Bitbucket 的模板文件。

AWS Proton

AWS Proton 是一项基于云的服务，用于部署到云基础架构。您可以使用连接为 AWS Proton 模板中的资源创建指向第三方存储库的链接。有关更多信息，请参阅 AWS Proton 用户指南 中的创建指向存储库的链接。

设置连接

完成本部分中的任务，以便准备在开发工具控制台中创建和使用连接功能。

主题

- 注册 AWS
- 创建并应用具有创建连接的权限的策略

注册 AWS

注册 AWS 账户

如果您还没有 AWS 账户，请完成以下步骤来创建一个。

注册 AWS 账户

1. 打开 https://portal.aws.amazon.com/billing/signup。
2. 按照屏幕上的说明进行操作。

   在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

   当您注册 AWS 账户时，系统将会创建一个 AWS 账户根用户。根用户有权访问该账户中的所有 AWS 服务 和资源。作为安全最佳实践，请为管理用户分配管理访问权限，并且只使用根用户执行需要根用户访问权限的任务。

注册过程完成后，AWS 会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 https://aws.amazon.com/ 并选择我的账户来查看当前的账户活动并管理您的账户。

创建管理用户

注册 AWS 账户后，保护您的 AWS 账户根用户，启用 AWS IAM Identity Center，并创建一个管理用户，以避免使用根用户执行日常任务。

保护您的 AWS 账户根用户

1. 选择根用户并输入您的 AWS 账户电子邮件地址，以账户拥有者身份登录 AWS Management Console。在下一页上，输入您的密码。

   要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的以根用户身份登录。
2. 对您的根用户启用多重身份验证（MFA）。

   有关说明，请参阅《IAM 用户指南》中的为 AWS 账户 根用户启用虚拟 MFA 设备（控制台）。

创建管理用户

1. 启用 IAM Identity Center

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的启用 AWS IAM Identity Center。
2. 在 IAM Identity Center 中，为管理用户授予管理访问权限。

   有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《AWS IAM Identity Center 用户指南》中的使用默认 IAM Identity Center 目录 配置用户访问权限。

作为管理用户登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。

  要获取使用 IAM Identity Center 用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的登录 AWS 访问门户。

创建并应用具有创建连接的权限的策略

使用 JSON 策略编辑器创建策略

1. 登录AWS Management Console，然后通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在左侧的导航窗格中，选择策略。

   如果这是您首次选择策略，则会显示欢迎访问托管式策略页面。选择开始使用。
3. 在页面的顶部，选择创建策略。

4. 在策略编辑器部分，选择 JSON 选项。
5. 输入以下 JSON 策略文档：

```
{
   "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"
           ],
           "Resource": [
               "*"
           ]
       }
   ]
}
```

6. 选择下一步。

> **Note**
>
> 您可以随时在可视化和 JSON 编辑器选项卡之间切换。不过，如果您进行更改或在可视化编辑器中选择下一步，IAM 可能会调整策略结构以针对可视化编辑器进行优化。有关更多信息，请参阅《IAM 用户指南》中的调整策略结构。

7. 在查看并创建页面上，为您要创建的策略输入策略名称和描述（可选）。查看此策略中定义的权限以查看策略授予的权限。
8. 选择创建策略可保存您的新策略。

开始使用连接

开始使用连接的最简单方法是建立将第三方源存储库与 AWS 资源相关联的连接。如果您希望将管道连接到 AWS 源（如 CodeCommit），则可以作为源操作执行连接。但是，如果您有外部存储库，则必须创建连接以将存储库与管道相关联。在本教程中，您将设置与 Bitbucket 存储库和管道的连接。

在本部分中，您将使用以下连接：

- AWS CodePipeline：在这些步骤中，您可以使用 Bitbucket 存储库创建管道作为管道源。
- Amazon CodeGuru Reviewer：接下来，您将 Bitbucket 存储库与 CodeGuru Reviewer 中的反馈和分析工具相关联。

主题

- 先决条件
- 步骤 1：编辑源文件
- 步骤 2：创建管道
- 步骤 3：将您的存储库与 CodeGuru Reviewer 关联

先决条件

在开始之前，请完成 设置 中的步骤。您还需要一个您希望连接到 AWS 服务的第三方源存储库，并允许连接为您管理身份验证。例如，您可能希望将 Bitbucket 存储库连接到与源存储库集成的 AWS 服务。

- 使用您的 Bitbucket 账户创建一个 Bitbucket 存储库。
- 准备好您的 Bitbucket 凭证。当您使用 AWS Management Console 设置连接时，系统会要求您使用 Bitbucket 凭证登录。

步骤 1：编辑源文件

当您创建您的 Bitbucket 存储库时，包含默认的 `README.md` 文件，您要对其进行编辑。

1. 登录到您的 Bitbucket 存储库，然后选择 Source (源)。
2. 选择 `README.md` 文件，然后选择页面顶部的 Edit (编辑)。删除现有文本并添加以下文本。

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

3. 选择 Commit (提交)。

   确保 `README.md` 文件位于存储库的根级别。

步骤 2：创建管道

在此部分中，您将使用以下操作创建管道：

- 与您的 Bitbucket 存储库和操作连接的源阶段。
- AWS CodeBuild 构建操作的构建阶段。

使用向导创建管道

1. 在 https://console.aws.amazon.com/codepipeline/ 登录 CodePipeline 控制台。
2. 在 Welcome (欢迎) 页面、Getting started (入门) 页面或 Pipelines (管道) 页面上，选择 Create pipeline (创建管道)。
3. 在 Step 1: Choose pipeline settings (步骤 1: 选择管道设置) 的管道名称中，输入 **`MyBitbucketPipeline`**。
4. 在 Service role (服务角色) 中，选择 New service role (新建服务角色)。

   > **Note**
   >
   > 如果您选择改为使用现有的 CodePipeline 服务角色，请确保您已将 `codestar-connections:UseConnection` IAM 权限添加到您的服务角色策略。有关 CodePipeline 服务角色的说明，请参阅为 CodePipeline 服务角色添加权限。

5. 在高级设置下，保留原定设置值。在 Artifact store (构件存储) 中，选择 Default location (默认位置) 以将默认构件存储（如指定为默认值的 Amazon S3 项目存储桶）用于为管道选择的区域中的管道。

   > **Note**
   >
   > 这不是源代码的源存储桶。这是管道的项目存储。每个管道都需要一个单独的构件存储，例如 S3 存储桶。

   选择 Next（下一步）。

6. 在 Step 2: Add source stage (步骤 2：添加源阶段) 页面上，添加源阶段：

   a. 对于 Source provider (源提供商)，选择 Bitbucket。

   b. 在 Connection (连接) 下，选择 Connect to Bitbucket (连接到 Bitbucket)。

   c. 在 Connect to Bitbucket (连接到 Bitbucket) 页面上的 Connection name (连接名称) 中，输入要创建的连接的名称。该名称可帮助您稍后识别此连接。

      在 Bitbucket apps (Bitbucket 应用程序) 下，选择Install a new app (安装新应用程序)。

   d. 在应用程序安装页面上，一条消息显示 AWS CodeStar 应用程序正在尝试连接到您的 Bitbucket 账户。选择授予访问权限。授权连接后，检测到您在 Bitbucket 上的存储库，您可以选择将其与 AWS 资源关联。

   e. 将显示新安装的连接 ID。选择完成连接。您将返回到 CodePipeline 控制台。

   f. 在 Repository name (存储库名称) 中，选择 Bitbucket 存储库的名称。

   g. 在 Branch name (分支名称) 中，选择存储库的分支。

   h. 确保选择了在源代码更改时启动管道选项。

   i. 在输出构件格式下，选择以下选项之一：CodePipeline 原定设置。

      - 选择 CodePipeline 原定设置，以便对管道中的构件使用原定设置 zip 格式。
      - 选择完全克隆，以包含有关管道中构件的存储库的 Git 元数据。只有 CodeBuild 操作支持此功能。

   选择 Next（下一步）。

7. 在 Add build stage (添加构建阶段)，添加一个构建阶段：

   a. 在 Build provider (构建提供商) 中，选择 AWS CodeBuild。允许 Region (区域) 默认为管道区域。

   b. 请选择 Create project（创建项目）。

   c. 在 Project name (项目名称)，输入此构建项目的名称。

   d. 在 Environment image (环境映像) 中，选择 Managed image (托管映像)。对于 Operating system (操作系统)，选择 Ubuntu。

   e. 对于 Runtime (运行时)，选择 Standard (标准)。对于映像，选择 aws/codebuild/standard:5.0。

   f. 对于 Service role (服务角色)，选择 New service role (新建服务角色)。

g. 在 Buildspec (构建规范) 下，为 Build specifications (构建规范) 选择 Insert build commands (插入构建命令)。选择 Switch to editor (切换到编辑器)，然后将以下内容粘贴到 Build commands (生成命令)：

```
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
```

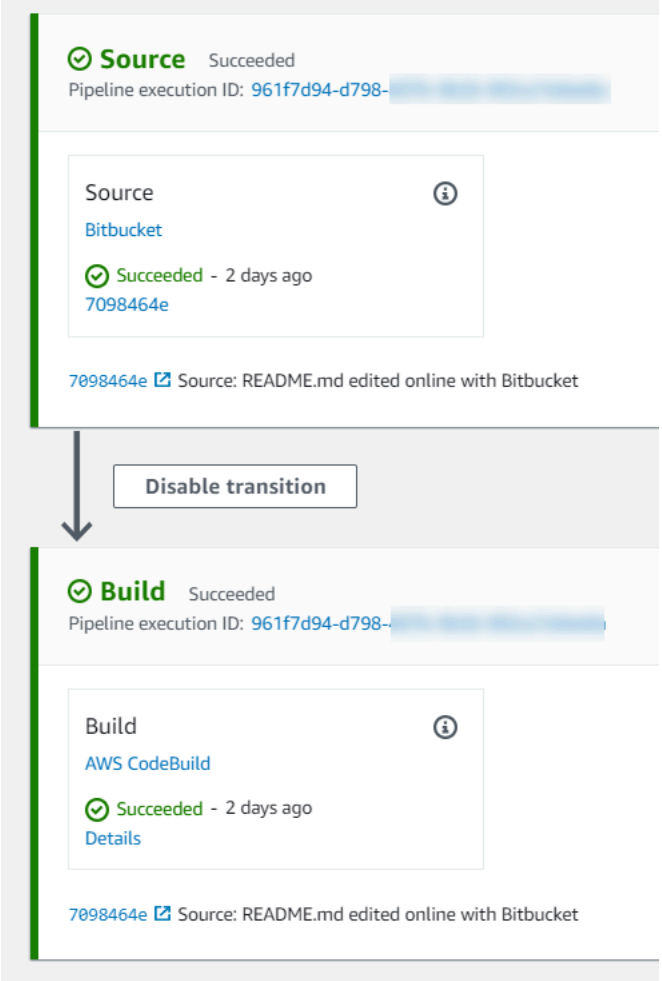
h. 选择 Continue to CodePipeline (前往 CodePipeline)。这将返回到 CodePipeline 控制台并创建一个 CodeBuild 项目，该项目使用您的构建命令用于配置。该构建项目使用服务角色来管理 AWS 服务权限。此步骤可能需要几分钟时间。

i. 选择 Next（下一步）。

8. 在 Step 4: Add deploy stage (步骤 4：添加部署阶段) 页面上，选择 Skip deploy stage (跳过部署阶段)，并通过再次选择 Skip (跳过) 接受警告消息。选择 Next（下一步）。

9. 在 Step 5: Review (步骤 5: 审核) 中，选择 Create pipeline (创建管道)。

10. 成功创建管道后，将开始管道执行。



11. 在成功构建阶段，选择 Details (详细信息)。

在 Execution details (执行详细信息) 下，查看 CodeBuild 构建输出。这些命令将输出 `README.md` 文件内容，如下所示：

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

```
[Container] 2020/06/05 19:14:51 Running command cat README.md
This is a Bitbucket repository!
[Container] 2020/06/05 19:14:51 Phase complete: PRE_BUILD State: SUCCEEDED
[Container] 2020/06/05 19:14:51 Phase context status code:  Message:
[Container] 2020/06/05 19:14:51 Entering phase BUILD
[Container] 2020/06/05 19:14:51 Phase complete: BUILD State: SUCCEEDED
[Container] 2020/06/05 19:14:51 Phase context status code:  Message:
[Container] 2020/06/05 19:14:51 Entering phase POST_BUILD
[Container] 2020/06/05 19:14:51 Phase complete: POST_BUILD State: SUCCEEDED
[Container] 2020/06/05 19:14:51 Phase context status code:  Message:
```

步骤 3：将您的存储库与 CodeGuru Reviewer 关联

创建连接后，您可以将该连接用于同一账户的所有 AWS 资源。例如，您可以对管道中的 CodePipeline 源操作和 CodeGuru Reviewer 中的存储库提交分析使用相同的 Bitbucket 连接。

1. 登录 CodeGuru Reviewer 控制台。
2. 在 CodeGuru Reviewer 中，选择 Associate repository (关联存储库)。

   此时将打开一页式向导。
3. 在 Select source provider (选择源提供程序) 下，选择 Bitbucket。
4. 在连接到 Bitbucket（使用 AWS CodeStar 连接）中，选择您为管道创建的连接。
5. 在 Repository location (存储库位置) 下，选择 Bitbucket 存储库的名称，然后选择 Associate (关联)。

   您可以继续设置代码审阅。有关更多信息，请参阅 Amazon CodeGuru Reviewer 用户指南 中的连接到 Bitbucket 以将存储库与 CodeGuru Reviewer 关联。

使用连接

连接是您用于将 AWS 资源连接到外部代码存储库的配置。每个连接都是可以提供给服务的资源，例如用于 AWS CodePipeline 连接到第三方存储库（例如 Bitbucket）。例如，您可以在中添加连接，CodePipeline 以便在对第三方代码存储库进行代码更改时，它会触发您的管道。您也可以将 AWS 资源连接到已安装的提供程序类型，例如 GitHub 企业服务器。

如果要创建与已安装的提供程序类型（例如 En GitHub terprise Server）的连接，则控制台会为您创建主机。主机是您创建的资源，用于表示安装提供程序的服务器。有关更多信息，请参阅 使用主机。

创建连接时，您可以使用控制台中的向导将 AWS CodeStar 应用程序与第三方提供商一起安装，并将其与新连接相关联。如果您已经安装了该 AWS CodeStar 应用程序，则可以使用它。

> **Note**
>
> 要在欧洲（米兰）使用连接 AWS 区域，您必须：
>
> 1. 安装区域特定的应用程序
> 2. 启用该区域
>
> 这一特定于区域的应用程序支持欧洲地区（米兰）区域中的连接。该应用程序在第三方提供商网站上发布，与支持其他区域的连接的现有应用程序是分开的。安装此应用程序，即表示您授权第三方提供商仅与该区域的服务共享您的数据，并且您可以随时通过卸载该应用程序来撤消权限。
> 除非您启用区域，否则该服务不会处理或存储您的数据。启用此区域，即表示您授予我们的服务处理和存储您的数据的权限。
> 即使未启用该区域，如果区域特定的应用程序仍保持安装状态，第三方提供商也仍可以与我们的服务共享您的数据，因此，请务必在禁用该区域后立即卸载该应用程序。有关更多信息，请参阅启用区域。

有关连接的更多信息，请参阅AWS CodeStar 连接 API 参考。有关 Bitbucket CodePipeline 源操作的更多信息，请参阅AWS CodePipeline 用户指南CodestarConnectionSource中的。

要为你的 AWS Identity and Access Management (IAM) 用户或角色创建或关联具有使用 AWS CodeStar 连接所需权限的策略，请参阅AWS CodeConnections 权限参考。根据您的 CodePipeline 服务角色的创建时间，您可能需要更新其权限以支持 AWS CodeStar 连接。有关说明，请参阅AWS CodePipeline 用户指南中的更新服务角色。

主题

- 创建连接
- 创建到 Bitbucket 的连接
- 创建与的连接 GitHub
- 创建与 GitHub 企业服务器的连接
- 创建与的连接 GitLab
- 创建与 GitLab自我管理的连接
- 更新挂起的连接
- 列出连接

- 删除连接
- 为连接资源添加标签
- 查看连接详细信息

创建连接

您可以创建与以下第三方提供程序类型的连接：

- 要创建到 Bitbucket 的连接，请参阅创建到 Bitbucket 的连接。
- 要创建与我们的 GitHub 企业云 GitHub 的连接，请参阅创建与的连接 GitHub。
- 要创建与 GitHub 企业服务器的连接，包括创建主机资源，请参阅创建与 GitHub 企业服务器的连接。
- 要创建与的连接 GitLab，请参阅创建与的连接 GitLab。

创建到 Bitbucket 的连接

您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 来创建与 bitbucket.org 上托管的存储库的连接。

开始前的准备工作：

- 您必须已使用 Bitbucket 创建有账户。
- 您必须已在 bitbucket.org 上创建了一个代码存储库。

> **Note**
>
> 您可以创建到 Bitbucket Cloud 存储库的连接。不支持已安装的 Bitbucket 提供程序类型（如 Bitbucket 服务器）。请参阅 AWS CodeStar 连接支持的提供程序和版本。

> **Note**
>
> 连接只能访问用于创建连接的账户所拥有的存储库。
> 如果要将应用程序安装在 Bitbucket 工作区中，则需要 Administer workspace（管理工作区）权限。否则，安装应用程序的选项将不会显示。

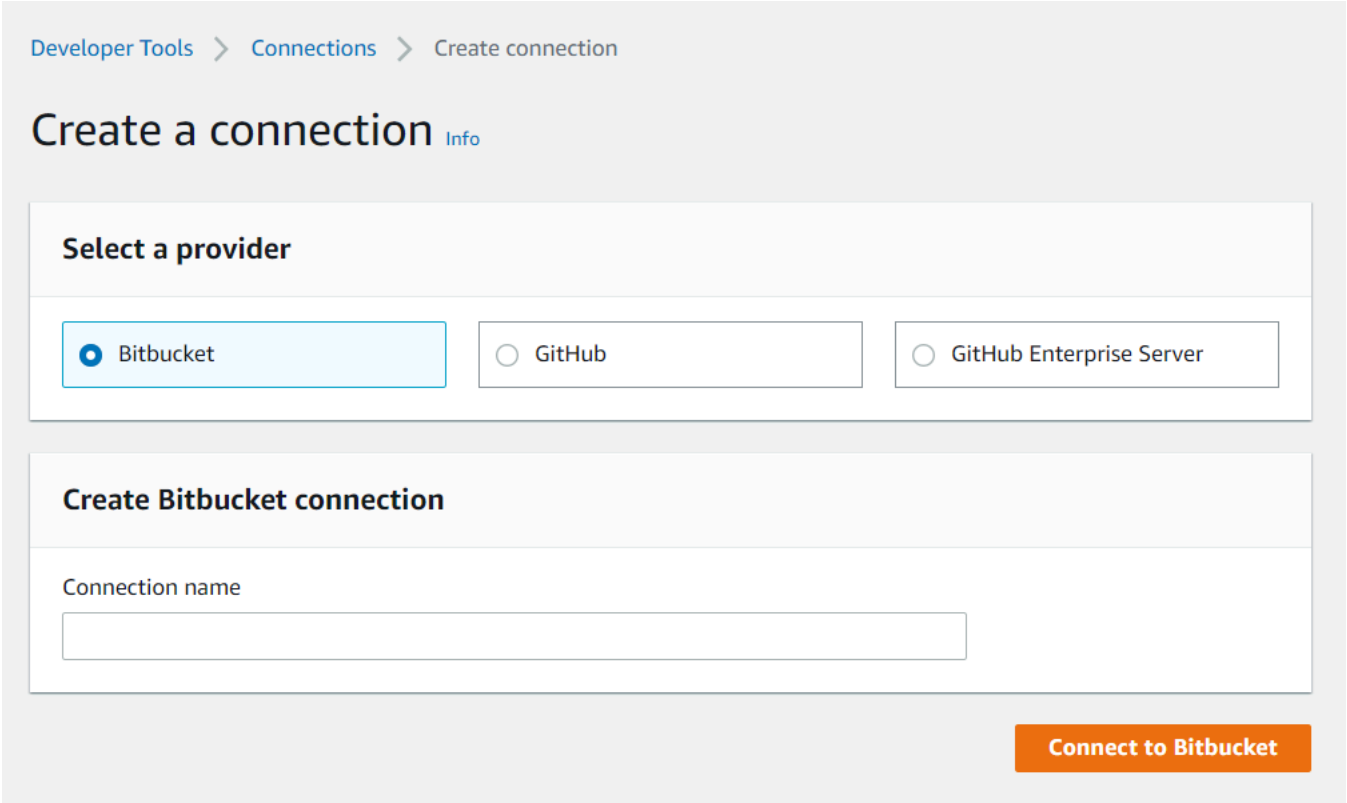
主题

- 创建到 Bitbucket 的连接（控制台）
- 创建到 Bitbucket 的连接 (CLI)

创建到 Bitbucket 的连接（控制台）

步骤 1：创建连接

1. 登录并打开 AWS 开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。 AWS Management Console
2. 选择设置 > 连接，然后选择创建连接。
3. 要创建到 Bitbucket 存储库的连接，请在 Select a provider (选择提供商) 下，选择 Bitbucket。在 Connection name (连接名称) 中，输入要创建的连接的名称。选择 Connect to Bitbucket (连接到 Bitbucket)，然后继续执行步骤 2。

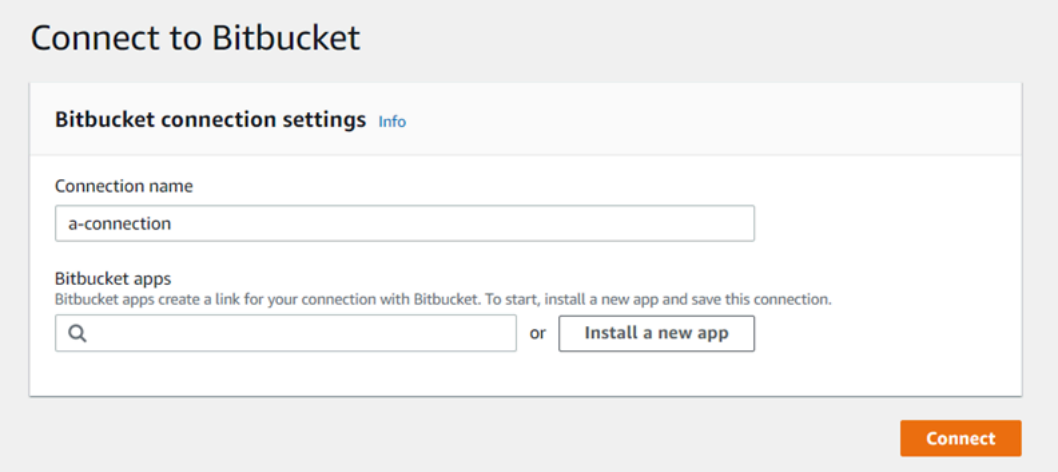


步骤 2：连接到 Bitbucket

1. 在 Connect to Bitbucket (连接到 Bitbucket) 设置页面上，将显示您的连接名称。

在 Bitbucket 应用程序下，选择一个应用程序安装，或者选择安装新应用程序来创建一个应用程序安装。

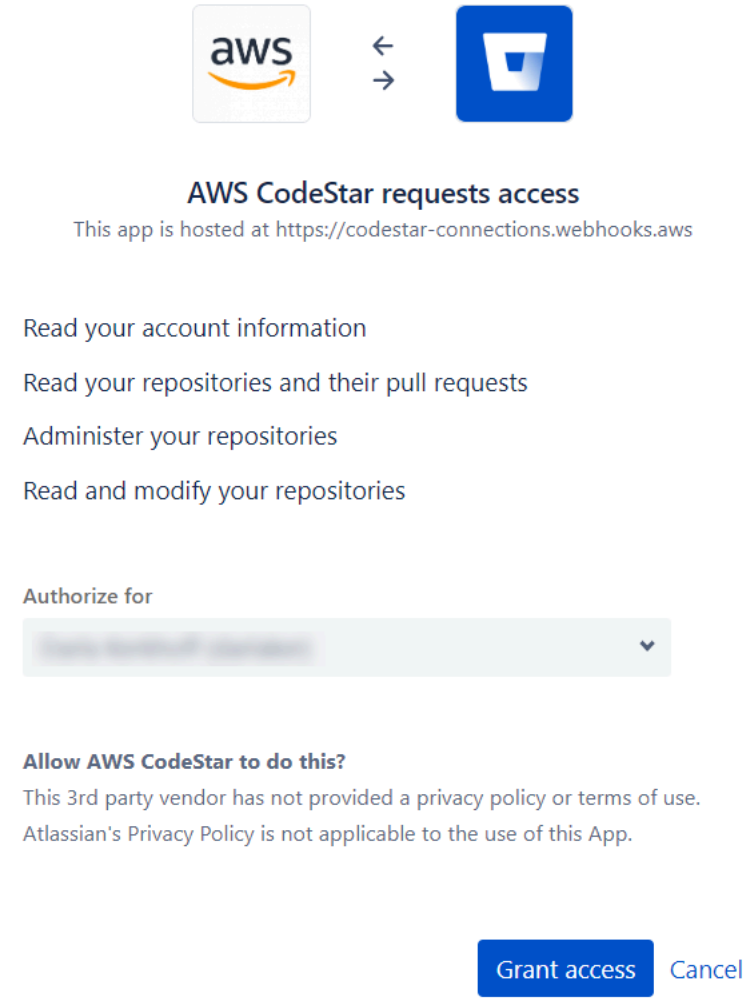
> **Note**
>
> 您只为每个 Bitbucket 工作区或账户安装一次应用程序。如果您已经安装了 Bitbucket 应用程序，请选择它，然后移至本部分内容中的最后一步。



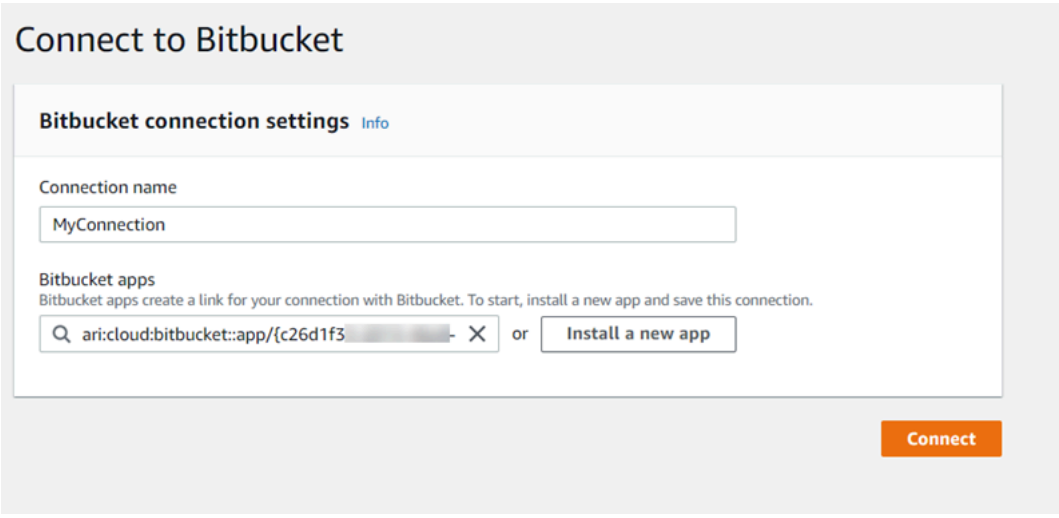
2. 如果显示 Bitbucket 的登录页面，请使用您的凭证登录，然后选择继续。
3. 在应用程序安装页面上，一条消息显示该 AWS CodeStar 应用程序正在尝试连接到您的 Bitbucket 帐户。

   如果您使用的是 Bitbucket 工作区，请更改工作区的 Authorize for（授权）选项。只有您拥有管理员访问权限的工作区才会显示。

   选择授予访问权限。



4. 在 Bitbucket 应用程序中，将显示新安装的连接 ID。选择连接。创建的连接将显示在连接列表中。



创建到 Bitbucket 的连接 (CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 来创建连接。

为此，请使用 create-connection 命令。

> **Important**
>
> 默认情况下，通过 AWS CLI 或创建的连接 AWS CloudFormation 处于PENDING状态。使用 CLI 或创建连接后 AWS CloudFormation，使用控制台编辑连接以使其处于状态AVAILABLE。

要创建到 Bitbucket 的连接

1. 打开终端（Linux、macOS 或 Unix）或命令提示符 (Windows)。 AWS CLI 使用运行create-connection命令，--connection-name为您的连接指定--provider-type和。在此示例中，第三方提供方名称为 Bitbucket，指定的连接名称为 MyConnection。

   ```
   aws codestar-connections create-connection --provider-type Bitbucket --connection-
   name MyConnection
   ```

   如果成功，该命令将返回类似以下内容的连接 ARN 信息。

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

2. 使用控制台完成连接。有关更多信息，请参阅 更新挂起的连接。

创建与的连接 GitHub

您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 来创建与的连接 GitHub。

开始前的准备工作：

- 您必须已经使用创建了账户 GitHub。
- 您必须已创建第三方代码存储库。

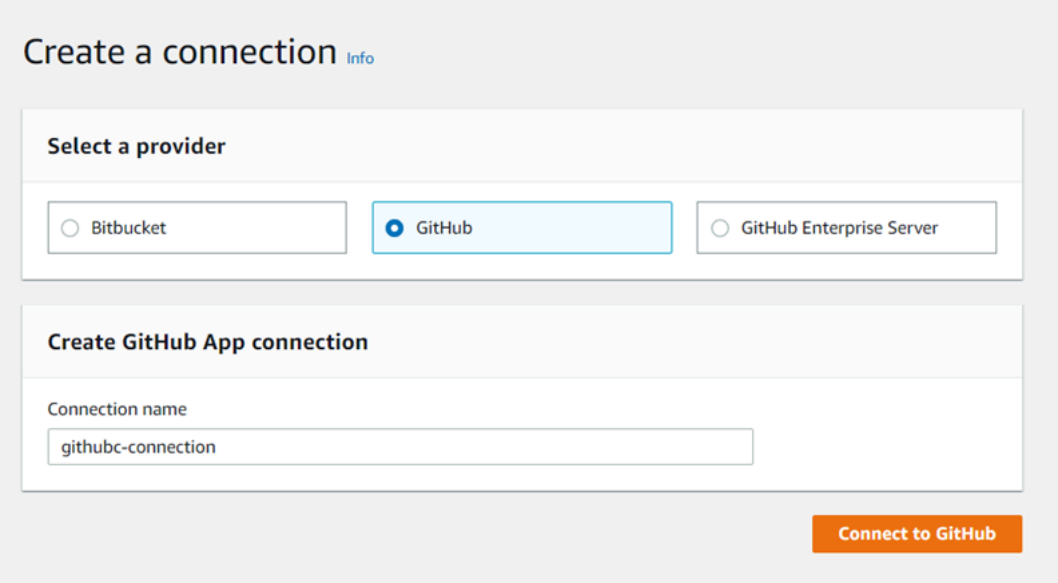
> **Note**
>
> 要创建连接，您必须是 GitHub 组织所有者。对于不属于组织的存储库，您必须是存储库拥有者。

主题

- 创建与 GitHub（控制台）的连接
- 创建与 GitHub (CLI) 的连接

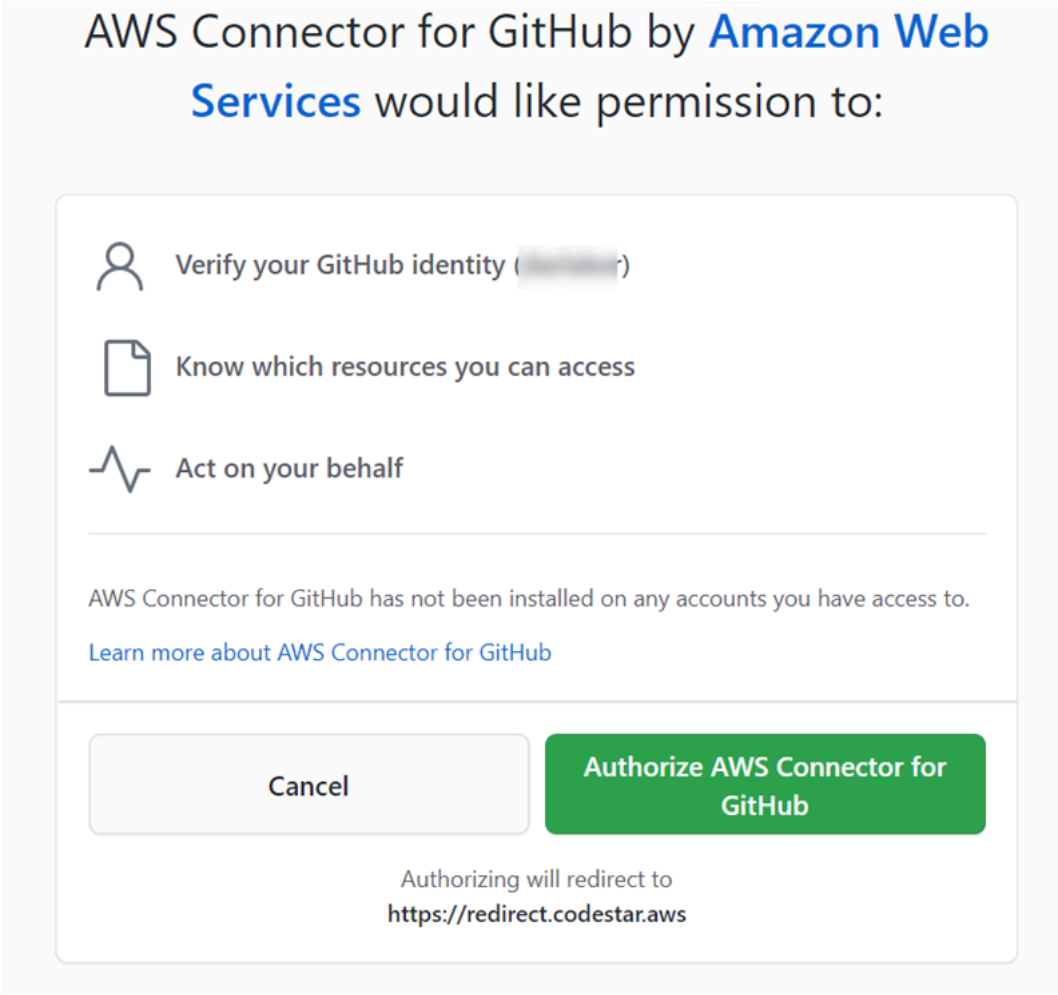
创建与 GitHub（控制台）的连接

1. 登录并打开开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。 AWS Management Console
2. 选择设置 > 连接，然后选择创建连接。
3. 要创建与 GitHub 或 GitHub 企业云存储库的连接，请在选择提供商下选择GitHub。在连接名称中，输入要创建的连接的名称。选择 Connect t o GitHub，然后继续执行步骤 2。

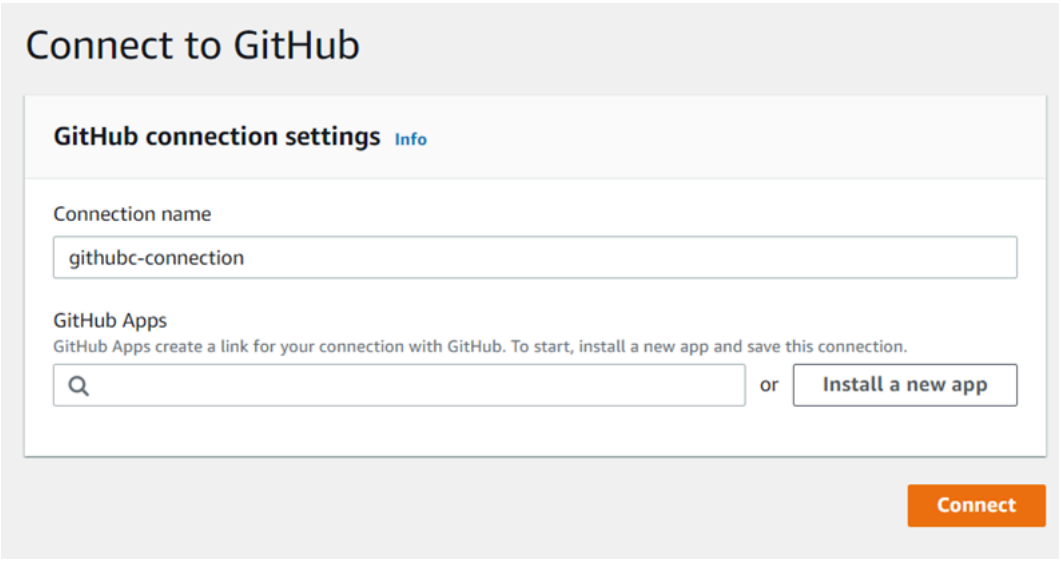


要创建与的连接 GitHub

1. 在 “GitHub 连接设置” 下，您的连接名称显示在 “连接名称” 中。选择 Connect to GitHub。此时将显示访问请求页面。



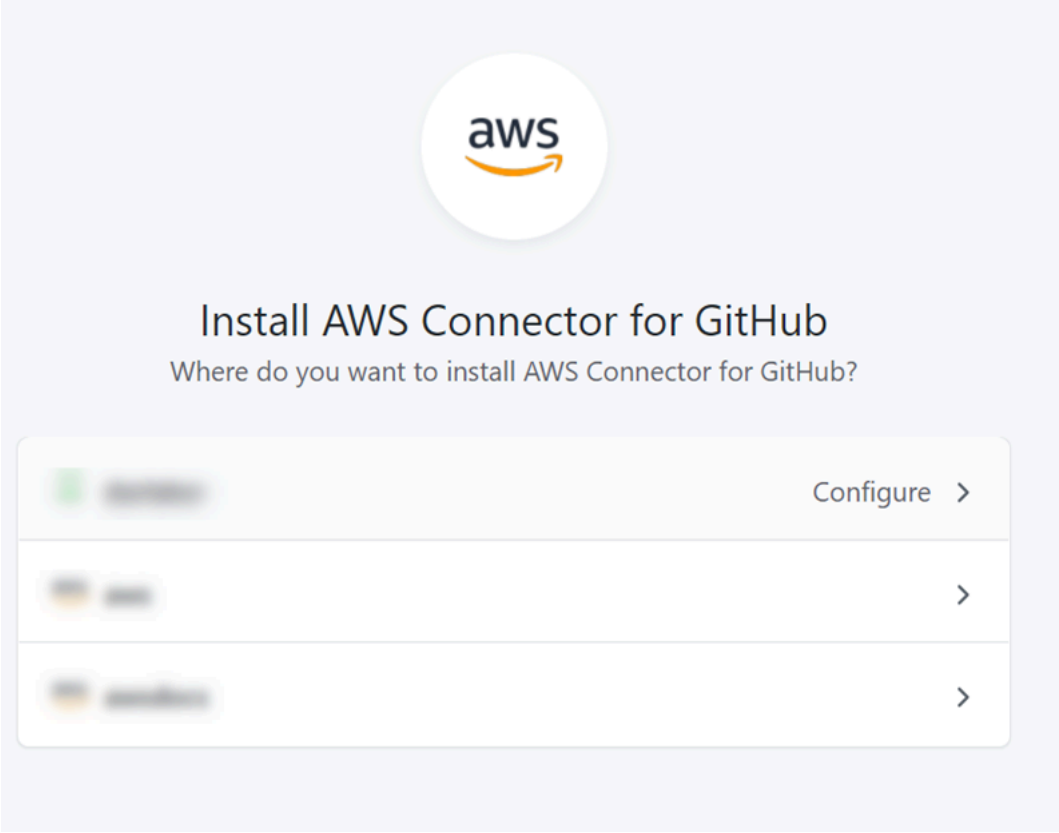
2. 选择 “为 AWS 连接器授权” GitHub。连接页面将显示并显示 “GitHub 应用程序” 字段。



3. 在 “GitHub 应用程序” 下，选择应用程序安装或选择 “安装新应用程序” 来创建一个。

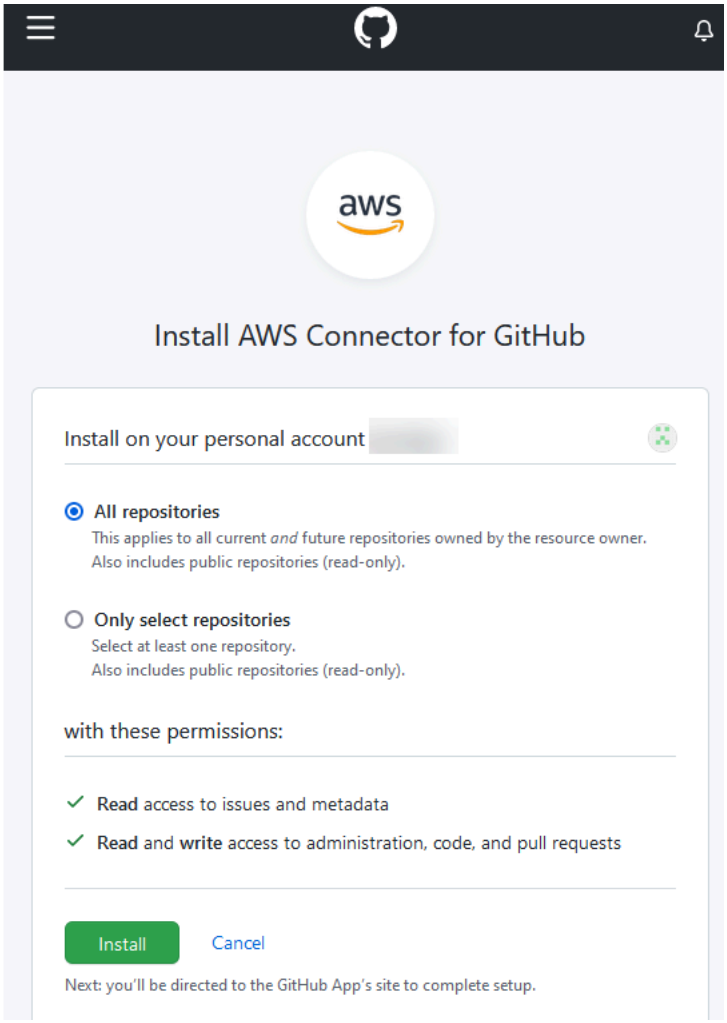
> **Note**
>
> 您可以为与特定提供程序的所有连接安装一个应用程序。如果您已经安装了 GitHub 应用程序 AWS 连接器，请选择它并跳过此步骤。

4. 在 “安装AWS 连接器 GitHub” 页面上，选择要在其中安装应用程序的帐户。



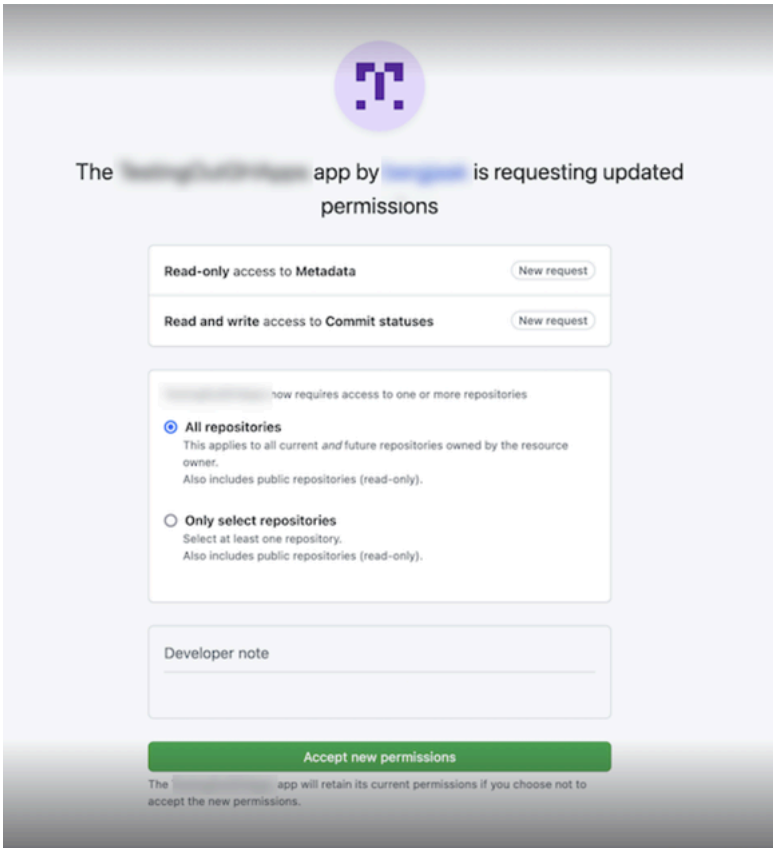
> **Note**
>
> 您只需为每个 GitHub 账户安装一次该应用程序。如果您之前已安装了应用程序，则可以选择配置，继续进入应用程序安装的修改页面，也可以使用后退按钮返回到控制台。

5. 在 “安装 AWS 连接器 GitHub” 页面上，保留默认值，然后选择 “安装”。



完成此步骤后，中可能会显示更新的权限页面 GitHub。

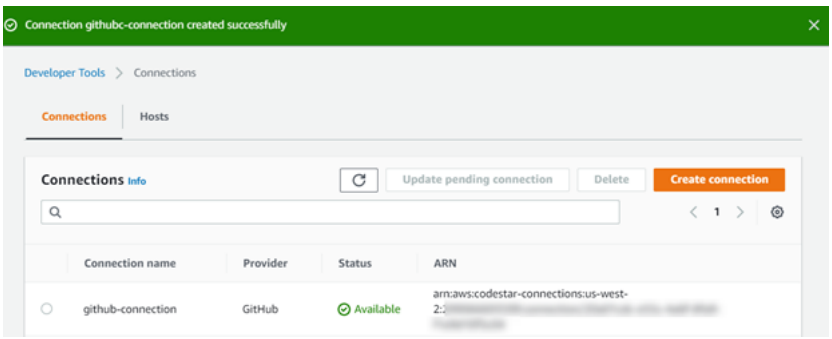
6. 如果显示的页面显示 GitHub 应用程序的 Conn AWS ector 权限已更新，请选择 “接受新权限”。



7. 您将返回到 “Connect to GitHub” 页面。新安装的连接 ID 会显示在 “GitHub应用程序” 中。选择连接。

查看您创建的连接

- 创建的连接将显示在连接列表中。



创建与 GitHub (CLI) 的连接

您可以使用 AWS Command Line Interface (AWS CLI) 创建与的连接 GitHub。

为此，请使用 create-connection 命令。

> **Important**
>
> 默认情况下，通过 AWS CLI 或创建的连接 AWS CloudFormation 处于PENDING状态。使用 CLI 或创建连接后 AWS CloudFormation，使用控制台编辑连接以使其处于状态AVAILABLE。

要创建与的连接 GitHub

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行create-connection命令，--connection-name为您的连接指定--provider-type和。在此示例中，第三方提供方名称为 GitHub，指定的连接名称为 MyConnection。

   ```
   aws codestar-connections create-connection --provider-type GitHub --connection-name
    MyConnection
   ```

   如果成功，该命令将返回类似以下内容的连接 ARN 信息。

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

2. 使用控制台完成连接。有关更多信息，请参阅 更新挂起的连接。

创建与 GitHub 企业服务器的连接

您可以使用连接将您的 AWS 资源与第三方存储库关联。您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 来创建与 GitHub 企业服务器的连接。

连接仅提供对 GitHub 企业服务器帐户拥有的存储库的访问权限，该帐户在连接创建期间用于授权安装 GitHub应用程序。

开始前的准备工作：

- 您必须已经有一个 GitHub 企业服务器实例和一个存储库。
- 您需要是 GitHub 企业服务器实例的管理员才能创建 GitHub 应用程序和创建主机资源，如本节所示。

> **Important**
>
> 在为 GitHub 企业服务器设置主机时，会为您创建一个 Webhook 事件数据的 VPC 终端节点。如果您在 2020 年 11 月 24 日之前创建了主机，并且想要使用 VPC PrivateLink Webhook 终端节点，则必须先删除主机，然后再创建新主机。

主题

- 创建与 GitHub企业服务器（控制台）的连接
- 创建与 GitHub企业服务器 (CLI) 的连接

创建与 GitHub企业服务器（控制台）的连接

要创建 GitHub 企业服务器连接，您需要提供 GitHub 企业服务器安装位置的信息，并使用您的 GitHub 企业凭据授权创建连接。

主题

- 创建您的 GitHub企业服务器连接（控制台）

创建您的 GitHub企业服务器连接（控制台）

要创建与 GitHub 企业服务器的连接，请准备好服务器 URL 和 GitHub 企业凭据。

要创建主机

1. 登录并打开 AWS 开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。 AWS Management Console
2. 在 Host (主机) 选项卡上，选择 Create host (创建主机)。
3. 在 Host name (产品名称) 中，输入您要用于主机的名称。
4. 在选择提供程序中，选择以下选项之一：
   - GitHub 企业服务器
   - GitLab 自我管理
5. 在 URL 中，输入安装提供程序的基础设施的终端节点。
6. 如果您的服务器是在 Amazon VPC 中配置的，并且您想要连接 VPC，请选择 Use a VPC (使用 VPC)。否则，请选择 No VPC (无 VPC)。

7. 如果您已将实例启动到 Amazon VPC 中，并且想要连接 VPC，请选择 Use a VPC (使用 VPC) 并完成以下操作。

   a. 在 VPC ID 中，选择您的 VPC ID。确保为基础设施选择安装实例的 VPC，或选择通过 VPN 或 Direct Connect 访问实例的 VPC。

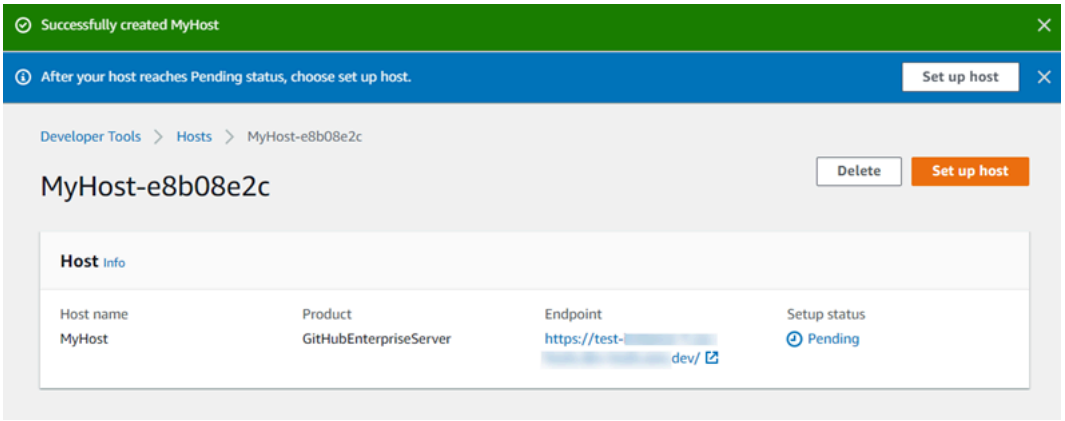
   b. 如果您配置了私有 VPC，并且您已将实例配置为使用非公有证书颁发机构执行 TLS 验证，请在 TLS 证书中输入证书 ID。TLS 证书值为证书的公有密钥。

8. 选择 Create host (创建主机)。

9. 显示主机详细信息页面后，主机状态随着主机的创建而发生变化。

> **Note**
>
> 如果您的主机设置包括 VPC 配置，请等待几分钟时间来预置主机网络组件。

等待您的主机进入 Pending (待处理) 状态，然后完成安装。有关更多信息，请参阅 设置待处理的主机。



步骤 2：创建与 GitHub 企业服务器（控制台）的连接

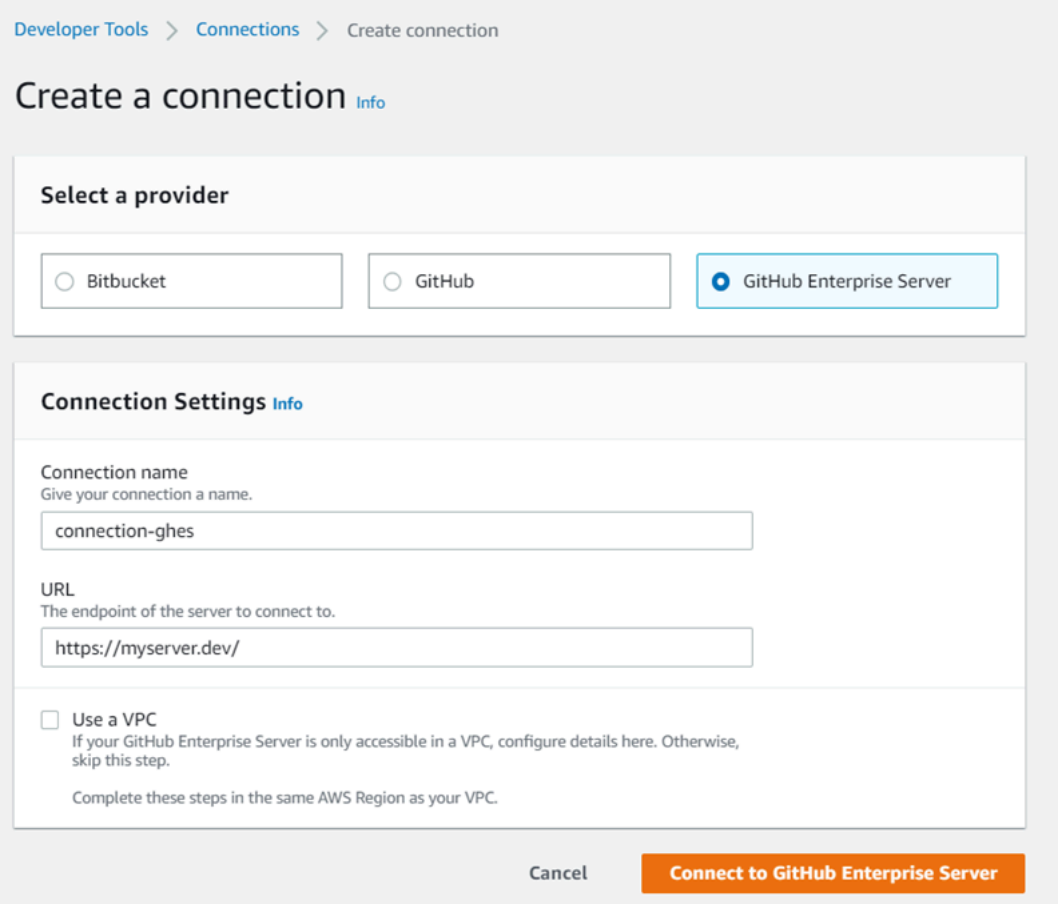
1. 登录 AWS Management Console 并打开开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。

2. 选择设置 > 连接，然后选择创建连接。

3. 要创建与已安装的 GitHub 企业服务器存储库的连接，请选择GitHub 企业服务器。

Connect 连接到 GitHub 企业服务器

1. 在 Connection name (连接名称)，输入您的连接的名称。



2. 在 URL 中，输入服务器的终端节点。

> **Note**
>
> 如果所提供的 URL 已用于为连接设置 GitHub企业服务器，则系统将提示您选择之前为该终端节点创建的主机资源 ARN。

3. （可选）如果您已将服务器启动到 Amazon VPC 中，并且想要连接 VPC，请选择使用 VPC 并完成以下操作。

   a. 在 VPC ID 中，选择您的 VPC ID。请务必为安装 GitHub 企业服务器实例的基础设施选择 VPC，或者选择可通过 VPN 或 Direct Connect 访问 GitHub企业服务器实例的 VPC。

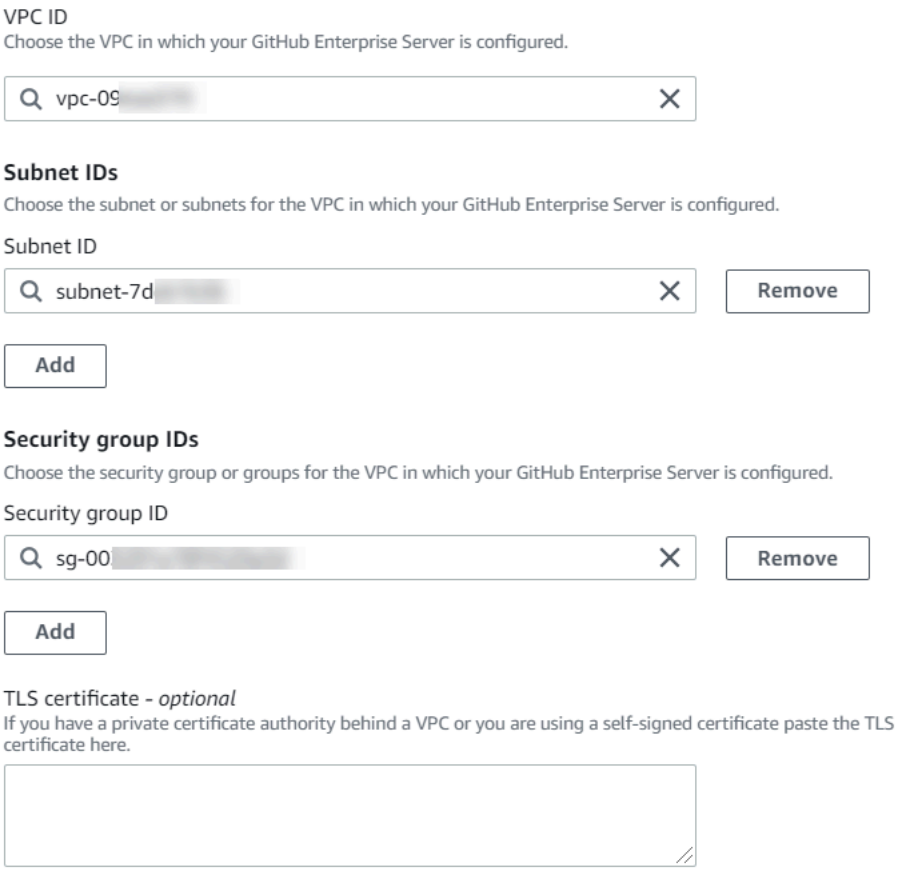
   b. 在 Subnet ID (子网 ID) 下，选择 Add (添加)。在字段中，选择要用于主机的子网 ID。您可以选择最多 10 个子网。

请务必为安装 GitHub 企业服务器实例的基础架构选择子网，或者选择可以通过 VPN 或 Direct Connect 访问已安装的 GitHub 企业服务器实例的子网。

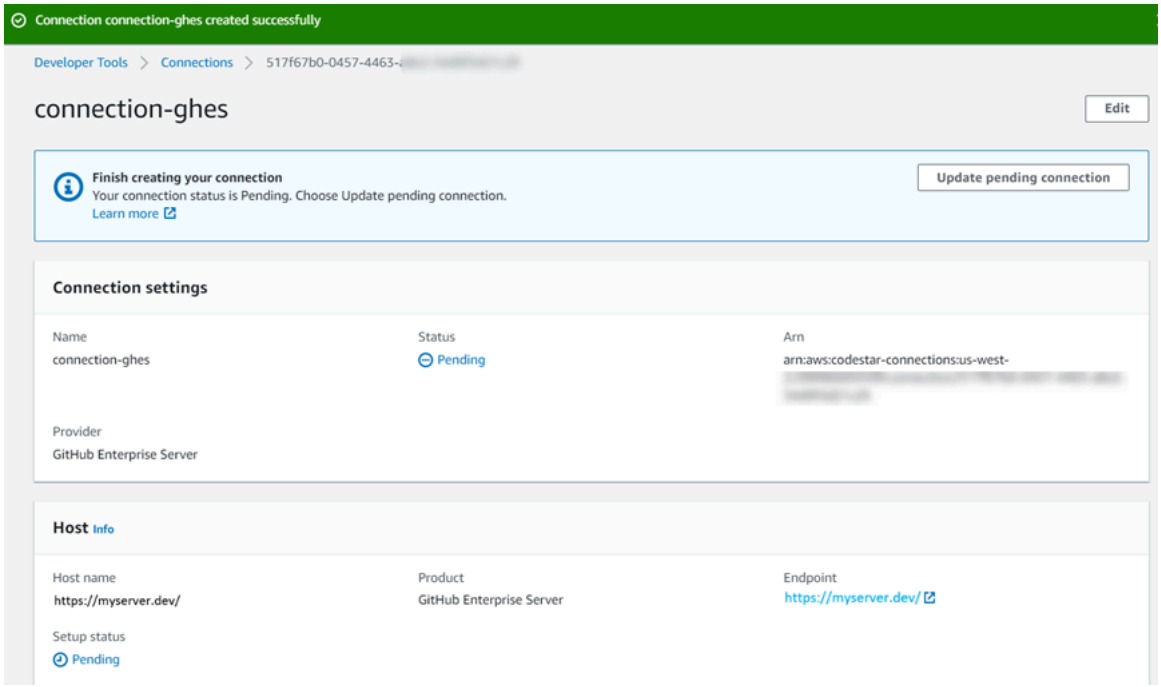
c. 在 Security group IDs (安全组 ID) 下，选择 Add (添加)。在字段中，选择要用于主机的安全组。您最多可以选择 10 个安全组。

请务必为安装 GitHub 企业服务器实例的基础架构选择安全组，或者选择可以通过 VPN 或 Direct Connect 访问已安装的 GitHub 企业服务器实例的安全组。

d. 如果您配置了私有 VPC，并且已将 GitHub 企业服务器实例配置为使用非公共证书颁发机构执行 TLS 验证，请在 TLS 证书中输入您的证书 ID。TLS 证书值应该是证书的公有密钥。

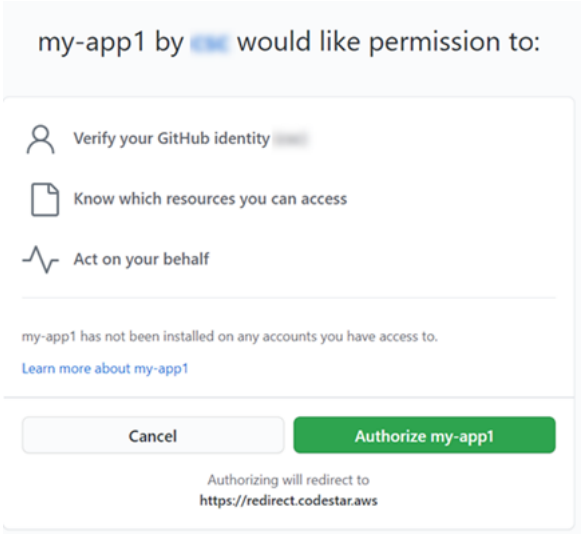


4. 选择 “Connect 到 GitHub 企业服务器”。创建的连接显示为待处理状态。将使用您提供的服务器信息为连接创建一个主机资源。对于主机名，将使用 URL。
5. 选择更新待处理的连接。



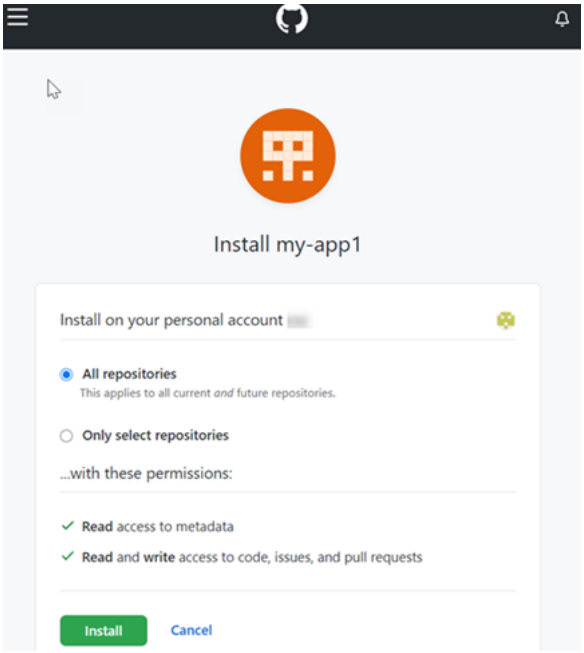
6. 如果出现提示，请在 GitHub 企业登录页面上使用您的 GitHub企业凭据登录。
7. 在 “创建 GitHub 应用程序” 页面上，为您的应用程序选择一个名称。

8. 在 GitHub 授权页面上，选择授权<app-name>。



9. 在应用程序安装页面上，一条消息显示 Connecto AWS CodeStar r 应用程序已准备就绪，可以安装了。如果您有多个组织，系统可能会提示您选择要安装该应用程序的组织。

   选择要安装应用程序的存储库设置。选择安装。



10. 连接页面显示已创建的连接处于 Available (可用) 状态。

创建与 GitHub企业服务器 (CLI) 的连接

您可以使用 AWS Command Line Interface (AWS CLI) 来创建连接。

为此，请使用 create-host 和 create-connection 命令。

> **Important**
>
> 默认情况下，通过 AWS CLI 或创建的连接 AWS CloudFormation 处于PENDING状态。使用 CLI 或创建连接后 AWS CloudFormation，使用控制台编辑连接以使其处于状态AVAILABLE。

步骤 1：为 GitHub 企业服务器创建主机 (CLI)

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行create-host命令，--provider-endpoint为您的连接指定--name--provider-type、和。在此示例中，第三方提供程序名称为 GitHubEnterpriseServer，终端节点为 my-instance.dev。

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

   如果成功，该命令将返回类似以下内容的主机 Amazon 资源名称（ARN）信息。

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

   完成此步骤后，主机处于 PENDING 状态。
2. 使用控制台完成主机设置并将主机变为 Available 状态。有关更多信息，请参阅 设置待处理的主机。

步骤 2：在控制台中设置待处理的主机

1. 登录 AWS Management Console 并打开开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。
2. 使用控制台完成主机设置并将主机变为 Available 状态。请参阅 设置待处理的主机。

步骤 3：为 GitHub 企业服务器创建连接 (CLI)

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行create-connection命令，--connection-name为您的连接指定--host-arn和。

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

如果成功，该命令将返回类似以下内容的连接 ARN 信息。

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

2. 使用控制台设置待处理的连接。有关更多信息，请参阅 更新挂起的连接。

步骤 4：在控制台中完成与 GitHub企业服务器的连接

1. 登录 AWS Management Console 并打开开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。
2. 使用控制台设置待处理的连接，并将连接移到 `Available` 状态。有关更多信息，请参阅 更新挂起的连接。

创建与的连接 GitLab

您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 来创建与 gitlab.com 上托管的存储库的连接。

> **Note**
>
> 在中授权安装此连接 GitLab，即表示您授予我们的服务权限以处理您的数据，并且您可以随时通过卸载应用程序来撤消这些权限。

开始前的准备工作：

- 您必须已经使用创建了账户 GitLab。

> **Note**
>
> 连接只为用于创建并授权连接的账户提供访问权限。

> **Note**
>
> 您可以在其中拥有所有者角色的连接中创建连接 GitLab，然后可以将该连接与包含诸如之类的资源的存储库一起使用 CodePipeline。对于群组中的仓库，您无需成为群组拥有者。

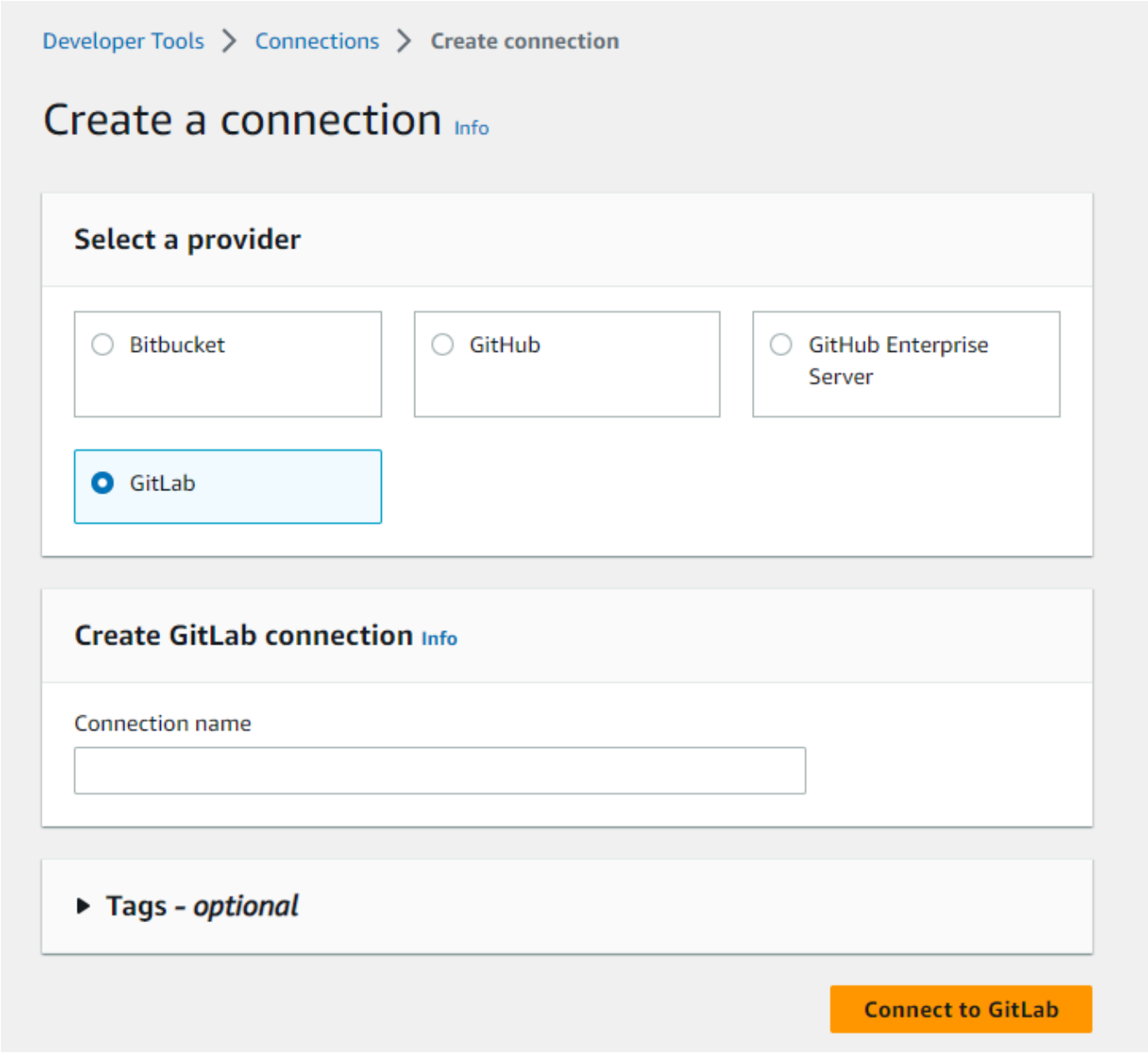
主题

- 创建与 GitLab（控制台）的连接
- 创建与 GitLab (CLI) 的连接

创建与 GitLab（控制台）的连接

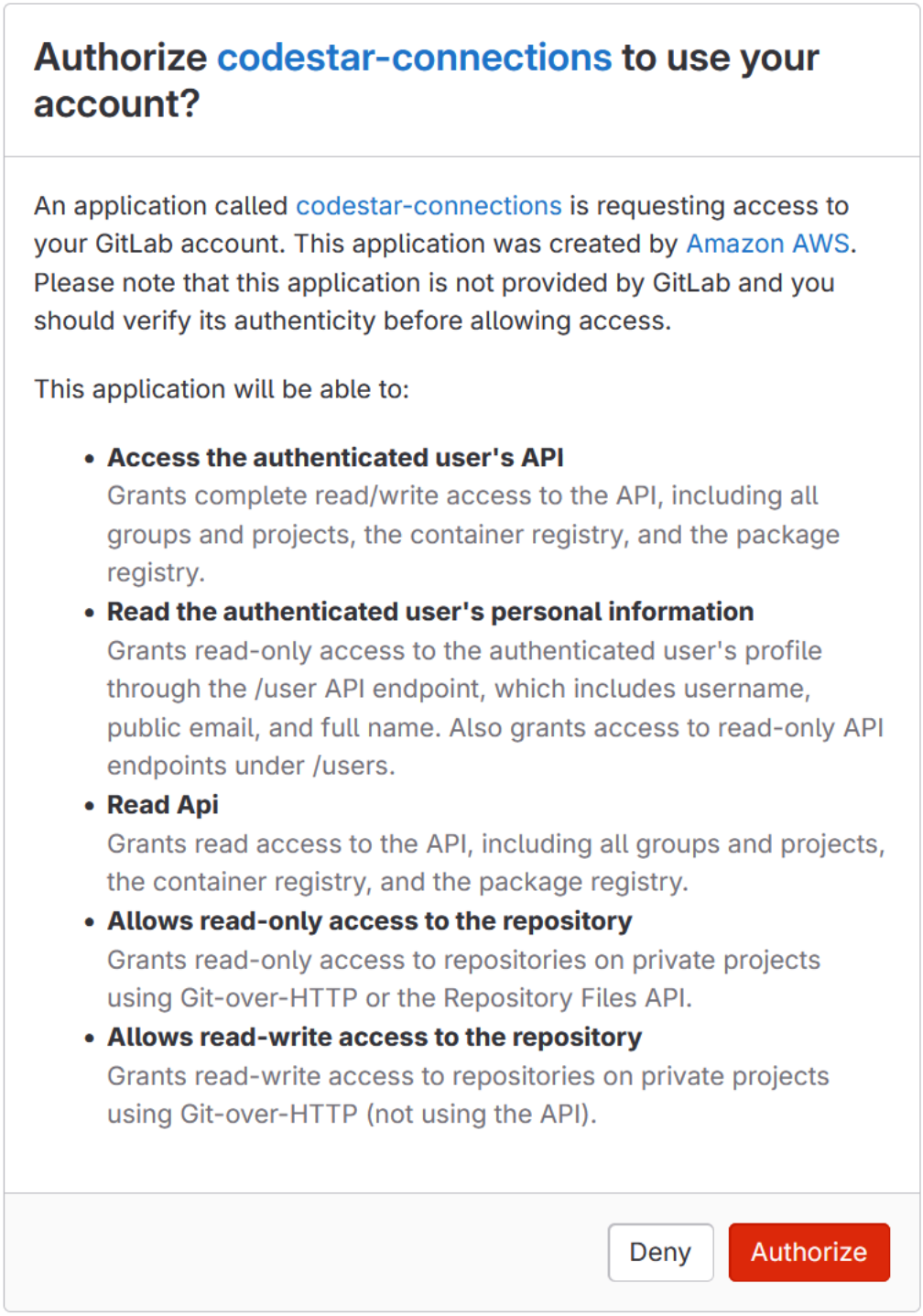
步骤 1：创建连接

1. 登录 AWS Management Console，然后打开 AWS 开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。
2. 选择设置，然后选择连接选项卡。选择创建连接。
3. 要创建与 GitLab 存储库的连接，请在选择提供商下选择GitLab。在连接名称中，输入要创建的连接的名称。选择 Connect t GitLab o。



4. GitLab 显示的登录页面时，使用您的凭据登录，然后选择登录。
5. 将显示一个授权页面，其中包含一条消息，请求授权连接访问您的 GitLab 账户。

   选择授权。



6. 浏览器返回到连接控制台页面。在 “创建 GitLab 连接” 下，新连接显示在 “连接名称” 中。
7. 选择 Connect t GitLab o。

成功创建连接后，将显示成功横幅。连接详细信息显示在连接设置页面上。

创建与 GitLab (CLI) 的连接

您可以使用 AWS Command Line Interface (AWS CLI) 来创建连接。

为此，请使用 create-connection 命令。

> **Important**
>
> 默认情况下，通过 AWS CLI 或创建的连接 AWS CloudFormation 处于PENDING状态。使用 CLI 或创建连接后 AWS CloudFormation，使用控制台编辑连接以使其处于状态AVAILABLE。

要创建与的连接 GitLab

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行create-connection命令，--connection-name为您的连接指定--provider-type和。在此示例中，第三方提供方名称为 GitLab，指定的连接名称为 MyConnection。

   ```
   aws codestar-connections create-connection --provider-type GitLab --connection-name
    MyConnection
   ```

   如果成功，该命令将返回类似以下内容的连接 ARN 信息。

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

2. 使用控制台完成连接。有关更多信息，请参阅 更新挂起的连接。

创建与 GitLab自我管理的连接

您可以使用自行管理安装为 GitLab 企业版或 GitLab 社区版创建连接。

您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 创建用于 GitLab 自我管理的连接和主机。

> **Note**
>
> 通过在 GitLab 自我管理模式下授权此连接应用程序，即表示您授予我们的服务权限以处理您的数据，并且您可以随时通过卸载该应用程序来撤消这些权限。

在创建 GitLab 自我管理连接之前，必须创建用于连接的主机，详见以下步骤。有关已安装提供程序的主机创建工作流程的概述，请参阅创建或更新主机的工作流程。

（可选）您可以为主机配置 VPC。有关主机资源的网络和 VPC 配置的更多信息，请参阅（可选）先决条件：您的连接的网络或 Amazon VPC 配置和对主机的 VPC 配置进行故障排查中的 VPC 先决条件。

开始前的准备工作：

- 您必须已经使用自己管理安装的 GitLab 企业版或 GitLab 社区版创建了帐户， GitLab 并且安装了企业版或社区版。有关更多信息，请参阅 https://docs.gitlab.com/ee/subscriptions/self_managed/。

  > **Note**
  >
  > 连接只为用于创建并授权连接的账户提供访问权限。

  > **Note**
  >
  > 您可以创建与拥有所有者角色的存储库的连接 GitLab，然后该连接可以与资源一起使用，例如 CodePipeline。对于群组中的仓库，您无需成为群组拥有者。

- 您必须已经创建了仅具有以下范围缩小权限的 GitLab 个人访问令牌 (PAT)：api。有关更多信息，请参阅 https://docs.gitlab.com/ee/user/profile/personal_access_tokens.html。只能使用管理员使用的 PAT。

  > **Note**
  >
  > 您的 PAT 用于对主机进行授权，不会以其它方式存储或由连接使用。要设置主机，您可以创建一个临时 PAT，然后在设置主机之后，可以删除 PAT。

主题

- 创建与 GitLab 自我管理（控制台）的连接
- 创建与 GitLab自我管理 (CLI) 的连接

创建与 GitLab 自我管理（控制台）的连接

使用这些步骤在控制台中创建主机和 GitLab 自我管理连接。有关在 VPC 中设置主机的注意事项，请参阅（可选）先决条件：您的连接的网络或 Amazon VPC 配置。

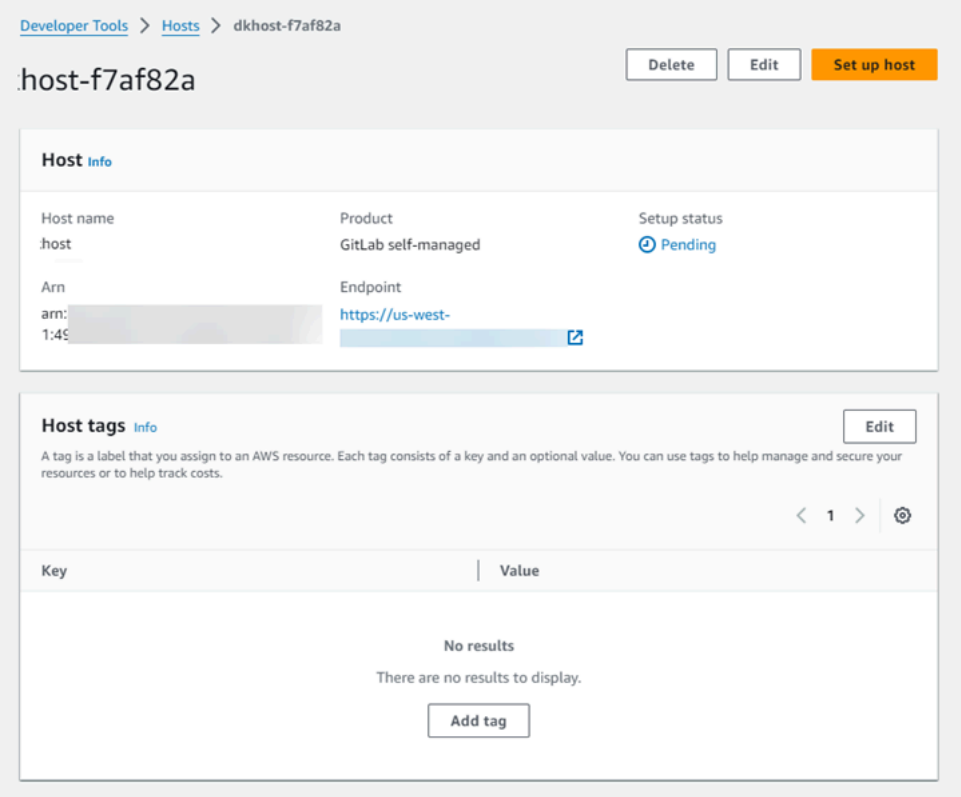
> **Note**
>
> 您可以为单个 GitLab 自管理安装创建主机，然后可以管理与该主机的一个或多个 GitLab 自管理连接。

步骤 1：创建主机

1. 登录 AWS Management Console，然后打开 AWS 开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。
2. 在 Host (主机) 选项卡上，选择 Create host (创建主机)。
3. 在 Host name (产品名称) 中，输入您要用于主机的名称。
4. 在选择提供商中，选择GitLab自我管理。
5. 在 URL 中，输入安装提供程序的基础设施的终端节点。
6. 如果您的服务器是在 Amazon VPC 中配置的，并且您想要连接 VPC，请选择 Use a VPC (使用 VPC)。否则，请选择 No VPC (无 VPC)。
7. （可选）如果您已将主机启动到 Amazon VPC 中，并且想要连接 VPC，请选择使用 VPC 并完成以下操作。
   a. 在 VPC ID 中，选择您的 VPC ID。确保为基础设施选择安装主机的 VPC，或选择通过 VPN 或 Direct Connect 访问实例的 VPC。
   b. 如果您配置了私有 VPC，并且您已将主机配置为使用非公有证书颁发机构执行 TLS 验证，请在 TLS 证书中输入证书 ID。TLS 证书值为证书的公有密钥。
8. 选择 Create host (创建主机)。
9. 显示主机详细信息页面后，主机状态随着主机的创建而发生变化。

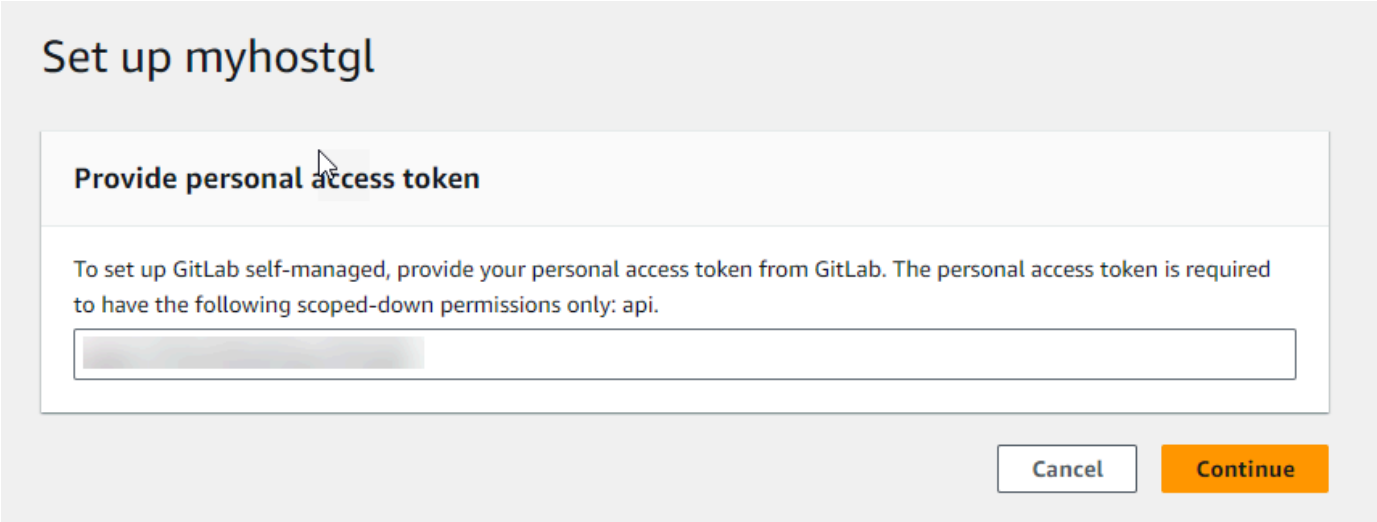
> **Note**
>
> 如果您的主机设置包括 VPC 配置，请等待几分钟时间来预置主机网络组件。

等待您的主机进入 Pending (待处理) 状态，然后完成安装。有关更多信息，请参阅 设置待处理的主机。

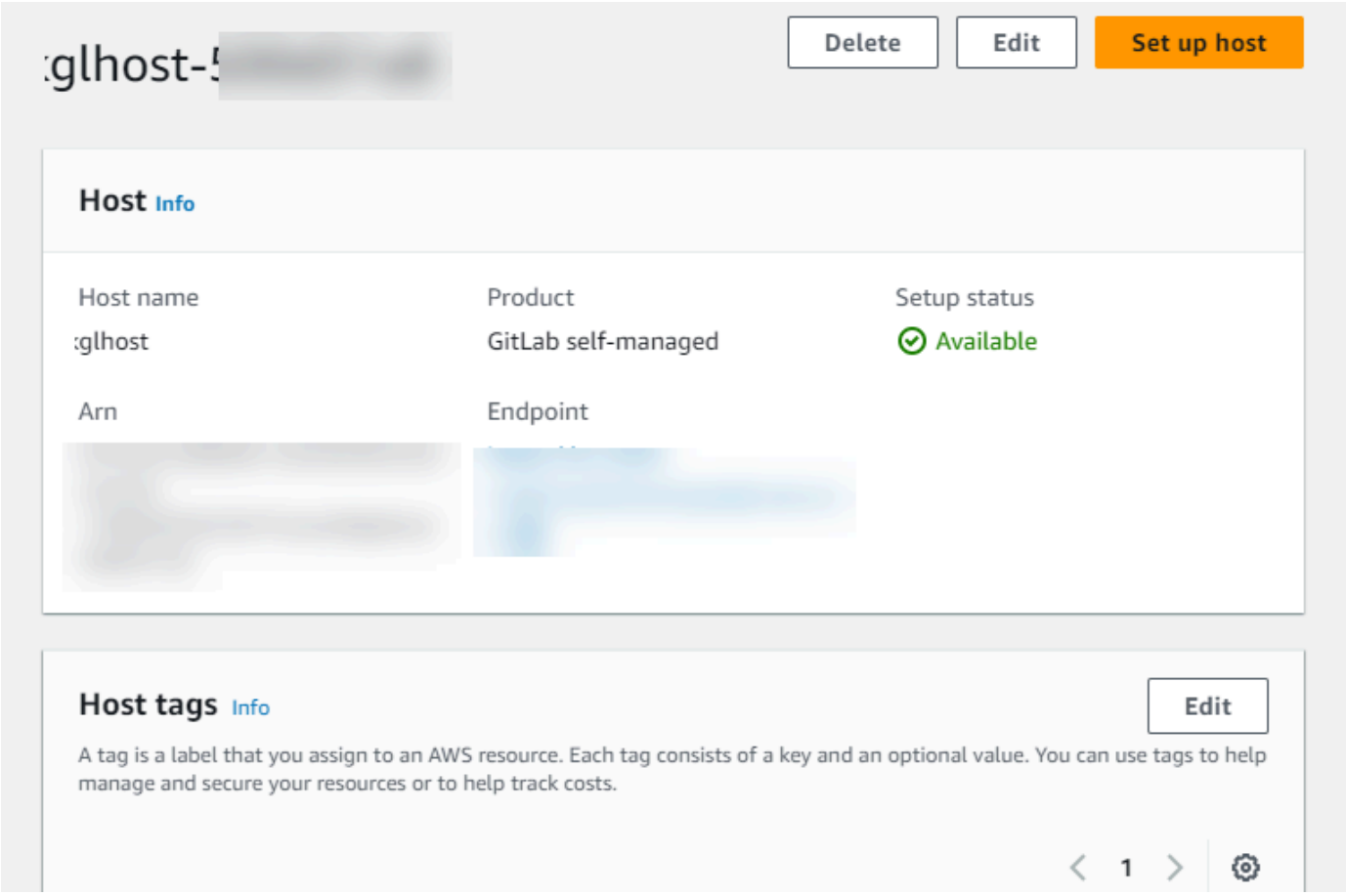


步骤 2：设置待处理的主机

1. 选择设置主机。
2. 将显示设置 ***host_name*** 页面。在提供个人访问令牌中，仅向你的 GitLab PAT 提供以下限定范围的权限：api。

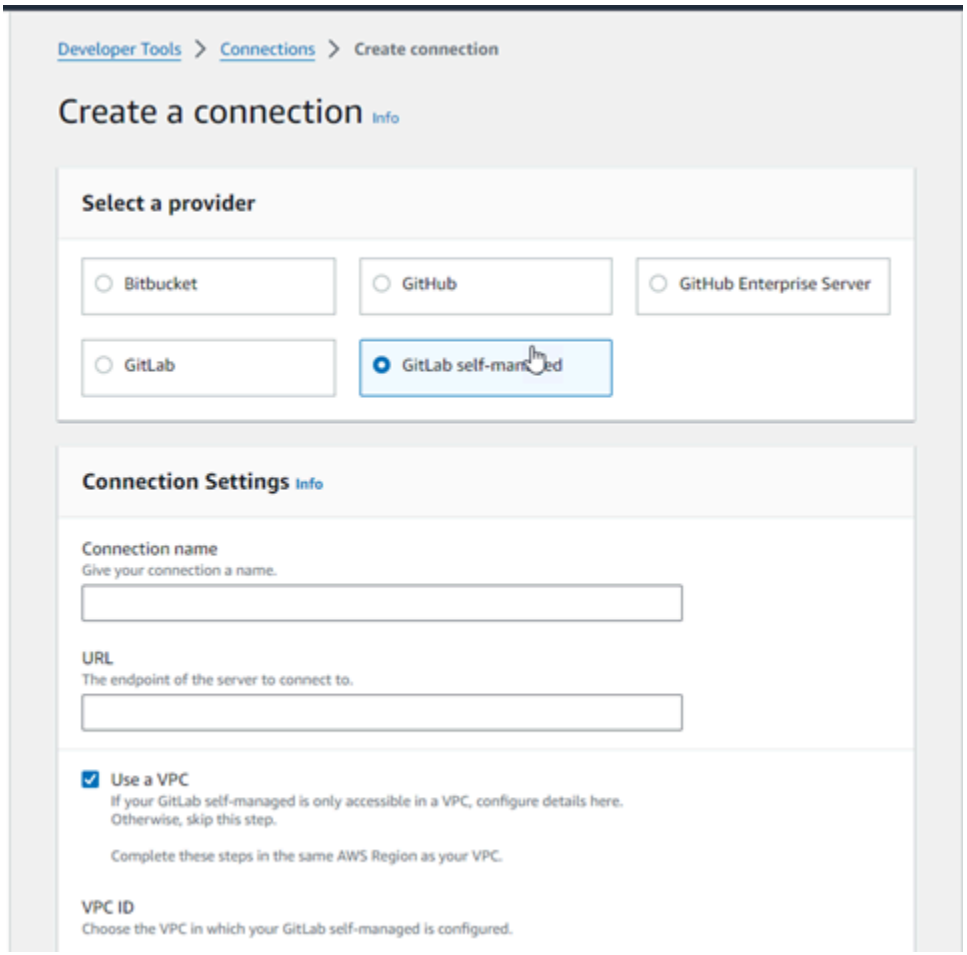


3. 成功注册主机后，将显示主机详细信息页面，并显示主机状态为 Available (可用)。



步骤 3：创建连接

1. 登录 AWS Management Console，然后打开 AWS 开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。
2. 选择设置，然后选择连接选项卡。选择创建连接。
3. 要创建与 GitLab 存储库的连接，请在 “选择提供商” 下，选择 “GitLab 自我管理”。在连接名称中，输入要创建的连接的名称。



4. 在 URL 中，输入服务器的终端节点。
5. 如果您已将服务器启动到 Amazon VPC 中，并且想要连接 VPC，请选择 Use a VPC (使用 VPC) 并完成以下操作。

   a. 在 VPC ID 中，选择您的 VPC ID。确保为基础设施选择安装主机的 VPC，或选择通过 VPN 或 Direct Connect 访问主机的 VPC。

   b. 在 Subnet ID (子网 ID) 下，选择 Add (添加)。在字段中，选择要用于主机的子网 ID。您可以选择最多 10 个子网。

确保为基础设施选择安装主机的子网，或选择通过 VPN 或 Direct Connect 访问已安装主机的子网。

c. 在 Security group IDs (安全组 ID) 下，选择 Add (添加)。在字段中，选择要用于主机的安全组。您最多可以选择 10 个安全组。

确保为基础设施选择安装主机的安全组，或选择通过 VPN 或 Direct Connect 访问已安装主机的安全组。

d. 如果您配置了私有 VPC，并且您已将主机配置为使用非公有证书颁发机构执行 TLS 验证，请在 TLS 证书中输入证书 ID。TLS 证书值应该是证书的公有密钥。

6. 选择 “Connect 以进行 GitLab 自我管理”。创建的连接显示为待处理状态。将使用您提供的服务器信息为连接创建一个主机资源。对于主机名，将使用 URL。
7. 选择更新待处理的连接。
8. GitLab 显示的登录页面时，使用您的凭据登录，然后选择登录。
9. 将显示一个授权页面，其中包含一条消息，请求授权连接访问您的 GitLab 账户。

选择授权。

10. 浏览器返回到连接控制台页面。在 “创建 GitLab 连接” 下，新连接显示在 “连接名称” 中。
11. 选择 “Connect 以进行 GitLab 自我管理”。

成功创建连接后，将显示成功横幅。连接详细信息显示在连接设置页面上。

创建与 GitLab自我管理 (CLI) 的连接

您可以使用 AWS Command Line Interface (AWS CLI) 创建用于 GitLab自行管理的主机和连接。

为此，请使用 create-host 和 create-connection 命令。

> **Important**
>
> 默认情况下，通过 AWS CLI 或创建的连接 AWS CloudFormation 处于`PENDING`状态。使用 CLI 或创建连接后 AWS CloudFormation，使用控制台编辑连接以使其处于状态`AVAILABLE`。

步骤 1：创建用于 GitLab 自行管理的主机 (CLI)

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行create-host命令，--provider-endpoint为您的连接指定--name--provider-type、和。在此示例中，第三方提供程序名称为 GitLabSelfManaged，终端节点为 my-instance.dev。

```
aws codestar-connections create-host --name MyHost --provider-type
 GitLabSelfManaged --provider-endpoint "https://my-instance.dev"
```

如果成功，该命令将返回类似以下内容的主机 Amazon 资源名称（ARN）信息。

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

完成此步骤后，主机处于 PENDING 状态。

2. 在以下步骤中，使用控制台完成主机设置并将主机移到 Available 状态。

步骤 2：在控制台中设置待处理的主机

1. 登录 AWS Management Console 并打开开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。
2. 使用控制台完成主机设置并将主机变为 Available 状态。请参阅 设置待处理的主机。

步骤 3：为 GitLab 自行管理 (CLI) 创建连接

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行create-connection命令，--connection-name为您的连接指定--host-arn和。

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

如果成功，该命令将返回类似以下内容的连接 ARN 信息。

```
{
```

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

2. 在以下步骤中，使用控制台设置待处理的连接。

步骤 4：在控制台中完成 GitLab 自我管理的连接

1. 登录 AWS Management Console 并打开开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。
2. 使用控制台设置待处理的连接，并将连接移到 Available 状态。有关更多信息，请参阅 更新挂起的连接。

更新挂起的连接

默认情况下，通过 AWS Command Line Interface (AWS CLI) 或创建的连接 AWS CloudFormation 处于PENDING状态。使用 AWS CLI 或创建连接后 AWS CloudFormation，使用控制台更新连接以使其处于状态AVAILABLE。

> **Note**
>
> 您必须使用控制台更新待处理的连接。您无法使用 AWS CLI更新待处理的连接。

首次使用控制台向第三方提供方添加新连接时，您必须使用与您的连接关联的安装完成与第三方提供方的 OAuth 握手。

您可以使用开发人员工具控制台完成挂起的连接。

完成连接

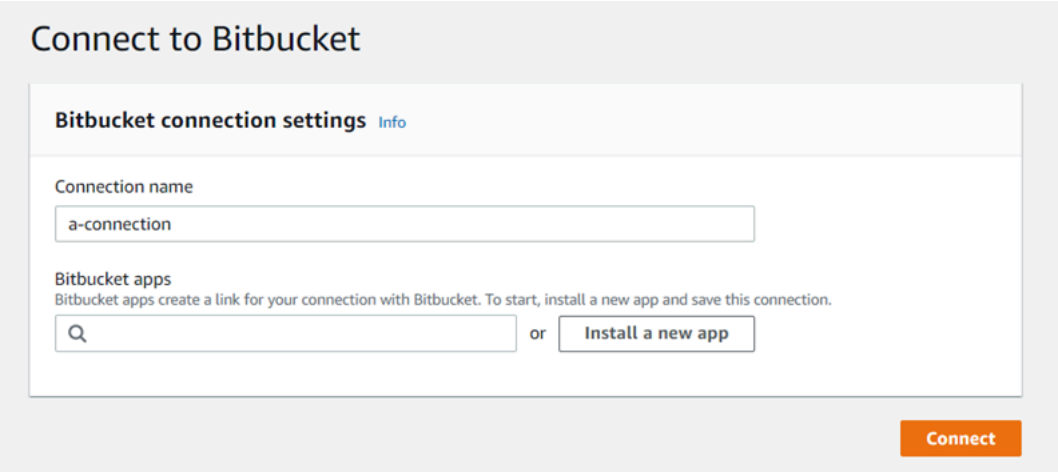
1. 打开 AWS 开发者工具控制台，网址为https://console.aws.amazon.com/codesuite/settings/connections。
2. 选择设置 > 连接。

   将显示与您的 AWS 账户关联的所有连接的名称。
3. 在名称中，选择要更新的挂起连接的名称。

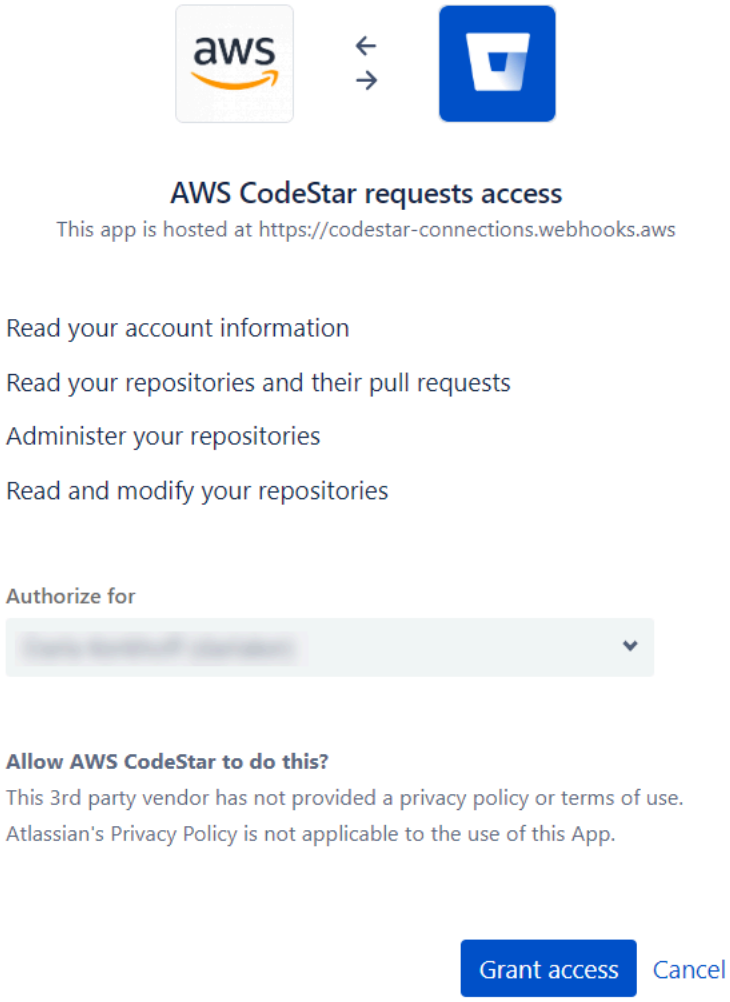
当您选择具有 Pending (待处理) 状态的连接时，Update a pending connection (更新待处理的连接) 将变为可用。

4. 选择 Update a pending connection (更新待处理的连接)。
5. 在 Connect to Bitbucket (连接到 Bitbucket) 页面上的 Connection name (连接名称) 中，验证连接的名称。

   在 Bitbucket apps (Bitbucket 应用程序) 下，选择一个应用程序安装或选择 Install a new app (安装新应用程序) 来创建一个应用程序安装。



6. 在应用程序安装页面上，一条消息显示该 AWS CodeStar 应用程序正在尝试连接到您的 Bitbucket 帐户。选择授予访问权限。



7. 将显示新安装的连接 ID。选择完成连接。

列出连接

您可以使用开发人员工具控制台或 AWS Command Line Interface (AWS CLI) 中的 list-connections 命令,来查看您账户中的连接列表。

列出连接（控制台）

列出连接

1. 在 https://console.aws.amazon.com/codesuite/settings/connections 打开开发工具控制台。
2. 选择 Settings > Connections (设置 > 连接)。
3. 查看连接的名称、状态和 ARN。

列出连接 (CLI)

您可以使用列出您与第三方代码存储库的连接。 AWS CLI 对于与主机资源关联的连接（例如与 En GitHub terprise Server 的连接），输出还会返回主机 ARN。

为此，请使用 list-connections 命令。

列出连接

- 打开终端（Linux、macOS 或 Unix）或命令提示符 (Windows)，然后使用运行命令。 AWS CLI list-connections

```
aws codestar-connections list-connections --provider-type Bitbucket
--max-results 5 --next-token: next-token
```

此命令将返回以下输出。

```
{
     "Connections": [
         {
             "ConnectionName": "my-connection",
             "ProviderType": "Bitbucket",
             "Status": "PENDING",
             "ARN": "arn:aws:codestar-connections: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:codestar-connections:us-west-2:account_id:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
             "OwnerAccountId": "account_id"
          },
      ],
     "NextToken": "next-token"
}
```

删除连接

您可以使用开发工具控制台或 AWS Command Line Interface (AWS CLI) 中的 delete-connection 命令来删除连接。

主题

- 删除连接（控制台）
- 删除连接 (CLI)

删除连接（控制台）

删除连接

1. 在 https://console.aws.amazon.com/codesuite/settings/connections 打开开发工具控制台。
2. 选择 Settings > Connections (设置 > 连接)。
3. 在连接名称中，选择要删除的连接的名称。
4. 选择 Delete。
5. 在字段中输入 **`delete`** 以确认，然后选择删除。

> **Important**
>
> 并且无法撤消。

删除连接 (CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 删除连接。

为此，请使用 delete-connection 命令。

> **Important**
>
> 运行该命令后，将删除连接。不显示任何确认对话框。您可以创建新连接，但不能重复使用 Amazon 资源名称 (ARN)。

删除连接

- 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。使用运行delete-connection命令，指定要删除的连接的 ARN。 AWS CLI

```
aws codestar-connections delete-connection --connection-arn arn:aws:codestar-
connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

该命令不返回任何内容。

为连接资源添加标签

标签是您或 AWS 分配给 AWS 资源的自定义属性标签。每个 AWS 标签分为两部分：

- 标签键（例如，CostCenter、Environment 或 Project）。标签键区分大小写。
- 一个称为标签值的可选字段（例如，111122223333、Production 或团队名称）。省略标签值与使用空字符串效果相同。与标签键一样，标签值区分大小写。

这些被统称为键/值对。

您可以使用控制台或 CLI 来标记资源。

您可以在 CodeConnections中标记以下资源类型：

- 连接
- 主机

这些步骤假设您已经安装了的最新版本 AWS CLI 或已更新到当前版本。有关更多信息，请参阅 AWS Command Line Interface 用户指南 中的安装 AWS CLI。

除了使用标签识别、组织和跟踪您的资源外，您还可以在 AWS Identity and Access Management (IAM) 策略中使用标签来帮助控制谁可以查看您的资源并与之交互。有关基于标签的访问策略示例，请参阅使用标签控制对 C AWS CodeStar onnections 资源的访问权限。

主题

- 标记资源（控制台）
- 标签资源 (CLI)

标记资源（控制台）

您可以使用控制台添加、更新或删除连接资源上的标签。

主题

- 为连接资源添加标签（控制台）
- 查看连接资源的标签（控制台）
- 编辑连接资源的标签（控制台）
- 从连接资源中删除标签（控制台）

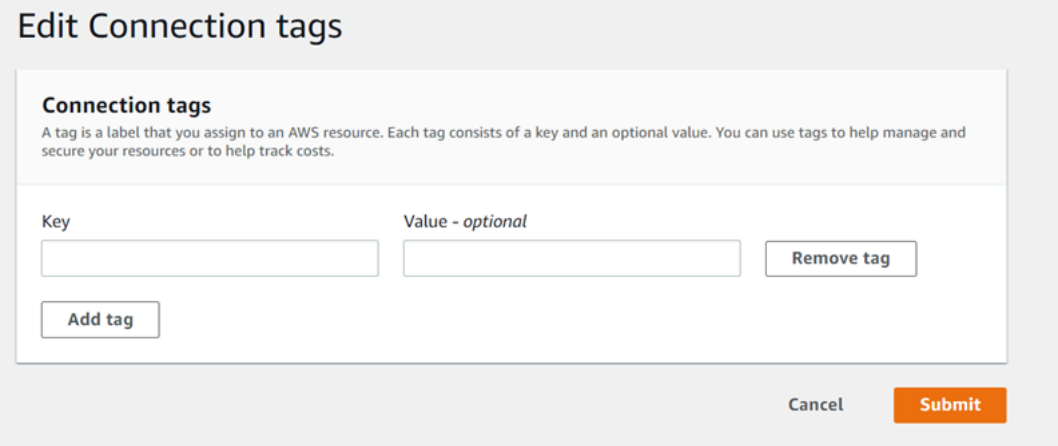
为连接资源添加标签（控制台）

您可以使用控制台向现有连接或主机添加标签。

> **Note**
>
> 当您为已安装的提供商（例如 En GitHub terprise Server）创建连接并且还为您创建主机资源时，创建期间的标签只会添加到连接中。如果要将主机重用于新连接，则可以单独标记主机。如果您要将标签添加到主机，请使用此处的步骤。

为连接添加标签

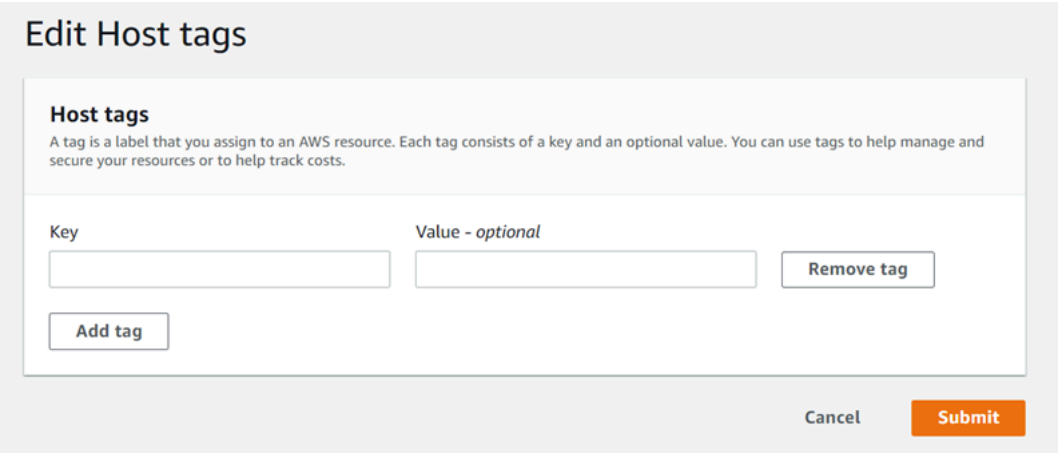
1. 登录到控制台。从导航窗格中，选择设置。
2. 在 Settings (设置) 下，选择 Connections (连接)。选择连接选项卡。
3. 选择要编辑的连接。此时将显示连接设置页面。
4. 在 Connection tags (连接标记) 下，选择 Edit (编辑)。Edit Connection tags (编辑连接标签) 页面随即显示。
5. 在键和值字段中，输入您要添加的每组标签的键/值对。（值字段为可选项。）例如，在键中，输入 **`Project`**。在值中，输入 **`ProjectA`**。



6. （可选）选择添加标签以添加多行并输入多个标签。
7. 选择 Submit (提交)。标签在连接设置下列出。

为主机添加标签

1. 登录到控制台。从导航窗格中，选择设置。
2. 在 Settings (设置) 下，选择 Connections (连接)。选择 Host (主机) 选项卡。
3. 选择要编辑的主机。此时将显示主机设置页面。
4. 在 Host tags (主机标签) 下，选择 Edit (编辑)。Host tags (主机标签) 页面随即显示。
5. 在键和值字段中，输入您要添加的每组标签的键/值对。（值字段为可选项。）例如，在键中，输入 **`Project`**。在值中，输入 **`ProjectA`**。



6. （可选）选择 Add tag (添加标签) 以添加多行并为主机输入多个标签。
7. 选择提交。标签在主机设置下列出。

查看连接资源的标签（控制台）

您可以使用控制台查看现有资源的标签。

要查看连接的标签

1. 登录到控制台。从导航窗格中，选择设置。
2. 在 Settings (设置) 下，选择 Connections (连接)。选择连接选项卡。
3. 选择要查看的连接。此时将显示连接设置页面。
4. 在 Connection tags (连接标签) 下，在 Key (键) 和 Value (值) 列下查看连接的标签。

查看主机的标签

1. 登录到控制台。从导航窗格中，选择设置。
2. 在 Settings (设置) 下，选择 Connections (连接)。选择 Host (主机) 选项卡。
3. 选择要查看的主机。
4. 在 Host tags (主机标签) 下，在 Key (键) 和 Value (值) 列下查看主机的标签。

编辑连接资源的标签（控制台）

您可以使用控制台来编辑已添加到连接资源的标签。

编辑连接的标签

1. 登录到控制台。从导航窗格中，选择设置。
2. 在 Settings (设置) 下，选择 Connections (连接)。选择连接选项卡。
3. 选择要编辑的连接。此时将显示连接设置页面。
4. 在 Connection tags (连接标记) 下，选择 Edit (编辑)。Connection tags (连接标记) 页面随即显示。
5. 在键和值字段中，根据需要更新每个字段的值。例如，对于 **`Project`** 键，在值中，将 **`ProjectA`** 更改为 **`ProjectB`**。
6. 选择 Submit (提交)。

编辑主机的标签

1. 登录到控制台。从导航窗格中，选择设置。

2. 在 Settings (设置) 下，选择 Connections (连接)。选择 Host (主机) 选项卡。
3. 选择要编辑的主机。此时将显示主机设置页面。
4. 在 Host tags (主机标签) 下，选择 Edit (编辑)。Host tags (主机标签) 页面随即显示。
5. 在键和值字段中，根据需要更新每个字段的值。例如，对于 **`Project`** 键，在值中，将 **`ProjectA`** 更改为 **`ProjectB`**。
6. 选择 Submit (提交)。

从连接资源中删除标签（控制台）

您可以使用控制台从连接资源中删除标签。当您移除关联资源的标签时，对应标签会被删除。

删除连接的标签

1. 登录到控制台。从导航窗格中，选择设置。
2. 在 Settings (设置) 下，选择 Connections (连接)。选择连接选项卡。
3. 选择要编辑的连接。此时将显示连接设置页面。
4. 在 Connection tags (连接标记) 下，选择 Edit (编辑)。Connection tags (连接标记) 页面随即显示。
5. 接下来，对于您要删除的每个标签的键和值，选择删除标签。
6. 选择提交。

删除主机的标签

1. 登录到控制台。从导航窗格中，选择设置。
2. 在 Settings (设置) 下，选择 Connections (连接)。选择 Host (主机) 选项卡。
3. 选择要编辑的主机。此时将显示主机设置页面。
4. 在 Host tags (主机标签) 下，选择 Edit (编辑)。Host tags (主机标签) 页面随即显示。
5. 接下来，对于您要删除的每个标签的键和值，选择删除标签。
6. 选择提交。

标签资源 (CLI)

您可以使用 CLI 在连接资源上查看、添加、更新或删除标签。

主题

- 向连接资源添加标签 (CLI)
- 查看连接资源的标签 (CLI)
- 编辑连接资源的标签 (CLI)
- 从连接资源中删除标签 (CLI)

向连接资源添加标签 (CLI)

您可以使用 AWS CLI 来标记连接中的资源。

在终端或命令行运行 tag-resource 命令，指定要为其添加标签的资源的 Amazon 资源名称 (ARN)，以及要添加的标签的键和值。您可以添加多个标签。

为连接添加标签

1. 为您的资源获取 ARN。使用显示在 列出连接 的 list-connections 命令来获取连接 ARN。
2. 在终端或命令行中，运行 tag-resource 命令。

   ########################## Proj ec t ########## ProjectA############ ReadOnlytrue#

   ```
   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
   ```

   如果成功，该命令不返回任何内容。

为主机添加标签

1. 为您的资源获取 ARN。使用显示在 列出主机 的 list-hosts 命令来获取主机 ARN。
2. 在终端或命令行中，运行 tag-resource 命令。

   ########################## Proj ec t ########## ProjectA############ IscontainerBasedtrue#

   ```
   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
   ```

如果成功，该命令不返回任何内容。

查看连接资源的标签 (CLI)

您可以使用 AWS CLI 来查看连接资源的 AWS 标签。如果尚未添加标签，则返回的列表为空。使用 list-tags-for-resource 命令查看已添加到连接或主机的标签。

查看连接的标签

1. 为您的资源获取 ARN。使用显示在 列出连接 的 list-connections 命令来获取连接 ARN。
2. 在终端或命令行中，运行 list-tags-for-resource 命令。例如，使用以下命令可查看连接的标签键和标签值列表。

```
aws codestar-connections list-tags-for-resource --resource-arn arn:aws:codestar-
connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

此命令返回与资源关联的标签。此示例显示了为连接返回的两个“键-值”对。

```
{
    "Tags": [
        {
            "Key": "Project",
            "Value": "ProjectA"
        },
        {
            "Key": "ReadOnly",
            "Value": "true"
        }
    ]
}
```

查看主机的标签

1. 为您的资源获取 ARN。使用显示在 列出主机 的 list-hosts 命令来获取主机 ARN。
2. 在终端或命令行中，运行 list-tags-for-resource 命令。例如，使用以下命令可查看主机的标签键和标签值列表。

```
aws codestar-connections list-tags-for-resource --resource-arn arn:aws:codestar-
connections:us-west-2:account_id:host/My-Host-28aef605
```

此命令返回与资源关联的标签。此示例显示了为主机返回的两个“键-值”对。

```
{
    "Tags": [
        {
            "Key": "IscontainerBased",
            "Value": "true"
        },
        {
            "Key": "Project",
            "Value": "ProjectA"
        }
    ]
}
```

编辑连接资源的标签 (CLI)

您可以使用 AWS CLI 来编辑资源的标签。您可以更改现有键的值或添加另一个键。

在终端或命令行运行 tag-resource 命令，指定要为其更新标签的资源的 ARN，并指定要更新的标签键和标签值。

编辑标签时，任何未指定的标签键都将被保留，而具有相同键和新值的任何标签键都将被更新。通过编辑命令添加的新键将作为新的键-值对添加。

编辑连接的标签

1. 为您的资源获取 ARN。使用显示在 列出连接 的 list-connections 命令来获取连接 ARN。
2. 在终端或命令行中，运行 tag-resource 命令。

   在此示例中，键 `Project` 的值将更改为 `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
```

如果成功，该命令不返回任何内容。要验证与连接关联的标签，请运行 list-tags-for-resource 命令。

编辑主机的标签

1. 为您的资源获取 ARN。使用显示在 列出主机 的 list-hosts 命令来获取主机 ARN。
2. 在终端或命令行中，运行 tag-resource 命令。

   在此示例中，键 `Project` 的值将更改为 `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
   ```

   如果成功，该命令不返回任何内容。要验证与主机关联的标签，请运行 list-tags-for-resource 命令。

从连接资源中删除标签 (CLI)

按照以下步骤使用 AWS CLI 从资源中移除标签。当您移除关联资源的标签时，对应标签会被删除。

> **Note**
>
> 如果删除连接资源，则会从删除的资源中删除所有标签关联。在删除连接资源之前，无需删除标签。

在终端或命令行运行 untag-resource 命令，指定要从中删除标签的资源的 ARN 以及要删除的标签的标签键。例如，要在使用标签键为 P *ro* ject and 的连接上删除多个标签 *ReadOnly*，请使用以下命令。

```
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
```

如果成功，该命令不返回任何内容。要验证与资源关联的标签，请运行 list-tags-for-resource 命令。输出将显示所有已删除的标签。

```
{
```

```
    "Tags": []
}
```

查看连接详细信息

您可以使用开发工具控制台或 AWS Command Line Interface (AWS CLI) 中的 get-connection 命令来查看连接的详细信息。要使用 AWS CLI，您必须已经安装了的最新版本 AWS CLI 或已更新到当前版本。有关更多信息，请参阅 AWS Command Line Interface 用户指南 中的安装 AWS CLI。

要查看连接（控制台）

1. 在 https://console.aws.amazon.com/codesuite/settings/connections 打开开发工具控制台。
2. 选择 Settings > Connections (设置 > 连接)。
3. 选择要查看的连接旁边的按钮，然后选择 View details (查看详细信息)。
4. 此时将显示以下连接信息：
   - 连接名称。
   - 连接的提供程序类型。
   - 连接状态。
   - 连接 ARN。
   - 如果连接是为已安装的提供商（例如 En GitHub terprise Server）创建的，则为与该连接关联的主机信息。
   - 如果连接是为已安装的提供商（例如 En GitHub terprise Server）创建的，则需要与该连接的主机关联的端点信息。
5. 如果连接位于 Pending (待处理) 状态，要完成连接，请选择 Update pending connection (更新待处理的连接)。有关更多信息，请参阅更新待处理的连接。

要查看连接 (CLI)

- 在终端或命令行中，运行 get-connection 命令。例如，使用以下命令可查看 ARN 值为 `arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f` 的连接的详细信息。

```
aws codestar-connections get-connection --connection-arn arn:aws:codestar-
connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

如果成功，则此命令会返回连接详细信息。

Bitbucket 连接的输出示例：

```
{
    "Connection": {
        "ConnectionName": "MyConnection",
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:account_id:connection/cdacd948-EXAMPLE",
        "ProviderType": "Bitbucket",
        "OwnerAccountId": "account_id",
        "ConnectionStatus": "AVAILABLE"
    }
}
```

GitHub 连接的输出示例：

```
{
    "Connection": {
        "ConnectionName": "MyGitHubConnection",
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:account_id:connection/ebcd4a13-EXAMPLE",
        "ProviderType": "GitHub",
        "OwnerAccountId": "account_id",
        "ConnectionStatus": "AVAILABLE"
    }
}
```

GitHub 企业服务器连接的输出示例：

```
{
    "Connection": {
        "ConnectionName": "MyConnection",
        "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:account_id:connection/2d178fb9-EXAMPLE",
        "ProviderType": "GitHubEnterpriseServer",
        "OwnerAccountId": "account_id",
        "ConnectionStatus": "PENDING",
        "HostArn": "arn:aws:codestar-connections:us-west-2:account_id:host/sdfsdf-
EXAMPLE"
    }
```

```
}
```

使用主机

要创建到已安装的提供程序类型（如 GitHub Enterprise Server）的连接，请首先使用 AWS Management Console 创建主机。主机是您创建的资源，用于表示安装提供程序的基础设施。然后使用该主机创建连接。有关更多信息，请参阅使用连接。

例如，您为您的连接创建一个主机，以便为您的提供商注册第三方应用程序以表示您的基础设施。为提供程序类型创建一个主机，然后与该提供程序类型的所有连接都使用该主机。

当您使用控制台创建到已安装的提供程序类型（如 GitHub Enterprise Server）的连接时，控制台会为您创建主机资源。

主题

- 创建主机
- 设置待处理的主机
- 列出主机
- 编辑主机
- 删除主机
- 查看主机详细信息

创建主机

您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI)，以创建与安装在基础设施上的第三方代码存储库的连接。例如，您可能让 GitHub Enterprise Server 作为虚拟机在 Amazon EC2 实例上运行。在创建到 GitHub Enterprise Server 的连接之前，您需要创建一个用于连接的主机。

有关已安装提供程序的主机创建工作流程的概述，请参阅创建或更新主机的工作流程。

开始前的准备工作：

- （可选）如果您想创建带有 VPC 的主机，您必须已经创建了网络或虚拟私有云（VPC）。
- 您必须已创建实例，并且如果您计划连接 VPC，则将主机启动到您的 VPC 中。

> **Note**
>
> 每个 VPC 一次只能关联一个主机。

（可选）您可以为主机配置 VPC。有关主机资源的网络和 VPC 配置的更多信息，请参阅（可选）先决条件：您的连接的网络或 Amazon VPC 配置和对主机的 VPC 配置进行故障排查中的 VPC 先决条件。

要使用控制台创建主机和到 GitHub Enterprise Server 的连接，请参阅创建您的 GitHub企业服务器连接（控制台）。控制台将为您创建一个主机。

要使用控制台创建主机和到 GitLab 自托管的连接，请参阅创建与 GitLab自我管理的连接。控制台将为您创建一个主机。

（可选）先决条件：您的连接的网络或 Amazon VPC 配置

如果为您的基础设施配置了网络连接，则可以跳过本节内容。

如果您的主机只能在 VPC 中访问，请在继续操作之前遵循以下 VPC 要求。

VPC 要求

您可以选择性地创建带有 VPC 的主机。以下是一般 VPC 要求，具体取决于您为安装设置的 VPC。

- 您可以配置带有公有和私有子网的公有 VPC。如果您没有首选的 CIDR 块或子网，您可以为 AWS 账户使用默认 VPC。
- 如果您已配置有私有 VPC，并且您已将 GitHub Enterprise Server 实例配置为使用非公有证书颁发机构执行 TLS 验证，则需要为您的主机资源提供 TLS 证书。
- 当 AWS CodeStar 连接创建您的主机时，系统会为您创建 Webhook 的 VPC 端点（PrivateLink）。有关更多信息，请参阅AWS CodeStar 连接和接口 VPC 端点（AWS PrivateLink）。
- 安全组配置
  - 创建主机期间使用的安全组需要允许网络接口连接到 GitHub Enterprise Server 实例的入站和出站规则
  - 附加到 GitHub Enterprise Server 实例（不是主机设置的一部分）的安全组需要通过连接创建的网络接口进行入站和出站访问。
- 您的 VPC 子网必须位于您所在区域的不同可用区中。可用区是可以隔离其他可用区的故障的不同位置。每个子网都必须完全位于一个可用区之内，不能跨越多个可用区。

有关使用 VPC 和子网的更多信息，请参阅 Amazon VPC 用户指南中的 为 IPv4 进行 VPC 和子网定型。

您为主机设置提供的 VPC 信息

在下一步中为连接创建主机资源时，需要提供以下内容：

- VPC ID：安装 GitHub Enterprise Server 实例的服务器的 VPC 的 ID，或通过 VPN 或 Direct Connect 访问已安装的 GitHub Enterprise Server 实例的 VPC 的 ID。
- 子网 ID：安装 GitHub Enterprise Server 实例的服务器的子网的 ID，或通过 VPN 或 Direct Connect 访问已安装的 GitHub Enterprise Server 实例的子网的 ID。
- 安全组：安装 GitHub Enterprise Server 实例的服务器的安全组，或通过 VPN 或 Direct Connect 访问已安装的 GitHub Enterprise Server 实例的安全组。
- 终端节点：准备好您的服务器终端节点，并继续下一步。

有关更多信息，包括 VPC 或主机连接的故障排查，请参阅 对主机的 VPC 配置进行故障排查。

权限要求

作为主机创建过程的一部分，AWS CodeStar 连接会代表您创建网络资源，以促进 VPC 连接。这包括一个网络接口（供 AWS CodeStar 连接从您的主机查询数据），以及一个 VPC 端点或 PrivateLink（供主机通过 Webhook 将事件数据发送到 AWS CodeStar 连接）。为了能够创建这些网络资源，请确保用于创建主机的 IAM 用户具有以下权限：

```
ec2:CreateNetworkInterface
ec2:CreateTags
ec2:DescribeDhcpOptions
ec2:DescribeNetworkInterfaces
ec2:DescribeSubnets
ec2:DeleteNetworkInterface
ec2:DescribeVpcs
ec2:CreateVpcEndpoint
ec2:DeleteVpcEndpoints
ec2:DescribeVpcEndpoints
```

有关 VPC 中权限或主机连接故障排除的更多信息，请参阅对主机的 VPC 配置进行故障排查。

有关 Webhook VPC 终端节点的更多信息，请参阅 AWS CodeStar 连接和接口 VPC 端点（AWS PrivateLink）。

主题

- 创建连接的主机（控制台）
- 创建用于连接的主机（CLI）

创建连接的主机（控制台）

对于用于安装的连接，例如与 GitHub Enterprise Server 或与 GitLab 自托管的连接，您可以使用主机来表示安装第三方提供程序的基础设施的端点。

要了解有关在 VPC 中设置主机的注意事项，请参阅 创建与 GitLab自我管理的连接。

要使用控制台创建主机和到 GitHub Enterprise Server 的连接，请参阅创建您的 GitHub企业服务器连接（控制台）。控制台将为您创建一个主机。

要使用控制台创建主机和到 GitLab 自托管的连接，请参阅创建与 GitLab自我管理的连接。控制台将为您创建一个主机。

> **Note**
>
> 您只为每个 GitHub Enterprise Server 或 GitLab 自托管账户创建一次主机。您与特定 GitHub Enterprise Server 或 GitLab 自托管账户的所有连接都将使用同一主机。

创建用于连接的主机（CLI）

您可以使用 AWS Command Line Interface (AWS CLI) 为已安装的连接创建主机。

> **Note**
>
> 您只需为每个 GitHub Enterprise Server 账户创建一次主机。您与特定 GitHub Enterprise Server 帐户的所有连接都将使用同一主机。

您可以使用主机来表示安装第三方提供程序的基础设施的终端节点。要使用 CLI 创建主机，请使用 create-host 命令。完成主机创建后，主机处于 Pending (待处理) 状态。然后您设置主机将其变为 Available (可用) 状态。主机可用后，完成创建连接的步骤。

> **Important**
>
> 默认情况下，通过 AWS CLI 创建的连接处于 `Pending` 状态。使用 CLI 创建主机后，可使用控制台设置主机使其状态为 `Available`。

要使用控制台创建主机和到 GitHub Enterprise Server 的连接，请参阅创建您的 GitHub企业服务器连接（控制台）。控制台将为您创建一个主机。

要使用控制台创建主机和到 GitLab 自托管的连接，请参阅创建与 GitLab自我管理的连接。控制台将为您创建一个主机。

设置待处理的主机

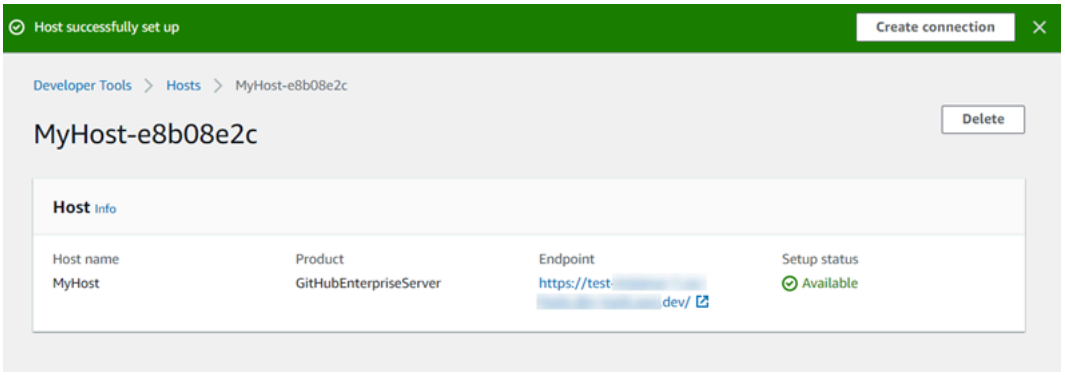
默认情况下，通过 AWS Command Line Interface (AWS CLI) 或 SDK 创建的主机处于 `Pending` 状态。使用控制台、AWS CLI 或 SDK 创建一个连接后，可使用控制台设置主机以使其状态为 `Available`。

您必须已创建一个主机。有关更多信息，请参阅 Create a host (创建主机)。

要设置待处理的主机

创建主机后，它处于 Pending (待处理) 状态。要将主机从 Pending (待处理) 变为 Available (可用) 状态，请完成以下步骤。此过程将执行与第三方提供程序的握手，以便在主机上注册 AWS 连接应用程序。

1. 当 AWS 开发工具控制台上您的主机变为 Pending（待处理）状态时，选择 Set up host（设置主机）。
2. 如果您正在为 GitLab 自托管创建主机，则会显示设置页面。在提供个人访问令牌中，仅向您的 GitLab PAT 提供以下范围缩小权限：api。
3. 在第三方安装的提供程序登录页面上，例如 GitHub Enterprise Server 登录页面上，如果出现提示，请使用您的账户凭证登录。
4. 在应用程序安装页面上，在 GitHub App name (GitHub 应用程序名称) 中，输入要为主机安装的应用程序的名称。选择 Create GitHub App (创建 GitHub 应用程序)。
5. 成功注册主机后，将显示主机详细信息页面，并显示主机状态为 Available (可用)。



6. 在主机可用后，您可以继续创建连接。在成功横幅上，选择 Create connection (创建连接)。完成 Create a connection (创建连接) 中的步骤。

列出主机

您可以使用开发工具控制台或 AWS Command Line Interface (AWS CLI) 中的 list-connections 命令来查看您账户中的连接列表。

列出主机（控制台）

要列出主机

1. 在 https://console.aws.amazon.com/codesuite/settings/connections 打开开发工具控制台。
2. 选择 Host (主机) 选项卡。查看主机的名称、状态和 ARN。

列出主机 (CLI)

您可以使用 AWS CLI 以列出已安装的第三方提供程序连接的主机。

为此，请使用 list-hosts 命令。

要列出主机

- 打开终端（Linux、macOS 或 Unix）或命令提示符 (Windows)，并使用 AWS CLI 运行 list-hosts 命令。

```
aws codestar-connections list-hosts
```

此命令将返回以下输出。

```
{
    "Hosts": [
        {
            "Name": "My-Host",
            "HostArn": "arn:aws:codestar-connections:us-west-2:account_id:host/My-
Host-28aef605",
            "ProviderType": "GitHubEnterpriseServer",
            "ProviderEndpoint": "https://my-instance.test.dev",
            "Status": "AVAILABLE"
        }
    ]
}
```

编辑主机

您可以编辑处于 Pending 状态主机的主机设置。您可以编辑主机名、URL 或 VPC 配置。

您不能将同一个 URL 用于多个主机。

> **Note**
>
> 要了解有关在 VPC 中设置主机的注意事项，请参阅 （可选）先决条件：您的连接的网络或 Amazon VPC 配置。

要编辑主机

1. 在 https://console.aws.amazon.com/codesuite/settings/connections 打开开发工具控制台。
2. 选择 Settings > Connections (设置 > 连接)。
3. 选择 Host (主机) 选项卡。

   此时会显示与您的 AWS 账户相关，并在选定的 AWS 区域内创建的主机。
4. 要编辑主机名，则在 Name (名称) 中输入新值。
5. 要编辑主机终端节点，则在 URL 中输入新值。
6. 要编辑主机 VPC 配置，请在 VPC ID 中输入新值。
7. 选择 Edit host (编辑主机)。
8. 将显示更新后的设置。选择 Set up Pending host (设置待处理主机)。

删除主机

您可以使用开发工具控制台或 AWS Command Line Interface (AWS CLI) 中的 delete-host 命令来删除主机。

主题

- 删除主机（控制台）
- 删除主机 (CLI)

删除主机（控制台）

要删除主机

1. 在 https://console.aws.amazon.com/codesuite/settings/connections 打开开发工具控制台。
2. 选择 Host (主机) 选项卡。在 Name (名称) 中，选择您要删除的主机的名称。
3. 选择 Delete。
4. 在字段中输入 **`delete`** 以确认，然后选择 Delete (删除)。

> **Important**
> 此操作无法撤消。

删除主机 (CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 删除主机。

为此，请使用 delete-host 命令。

> **Important**
> 必须先删除与主机关联的所有连接，然后才能删除主机。
> 运行该命令后，将删除主机。不显示任何确认对话框。

要删除主机

- 打开终端（Linux、macOS 或 Unix）或命令提示符 (Windows)。使用 AWS CLI 运行 delete-host 命令，指定要删除主机的 Amazon Resource Name (ARN)。

  ```
  aws codestar-connections delete-host --host-arn "arn:aws:codestar-connections:us-west-2:account_id:host/My-Host-28aef605"
  ```

  该命令不返回任何内容。

查看主机详细信息

您可以使用开发工具控制台或 AWS Command Line Interface (AWS CLI) 中的 get-host 命令来查看主机的详细信息。

要查看主机详细信息（控制台）

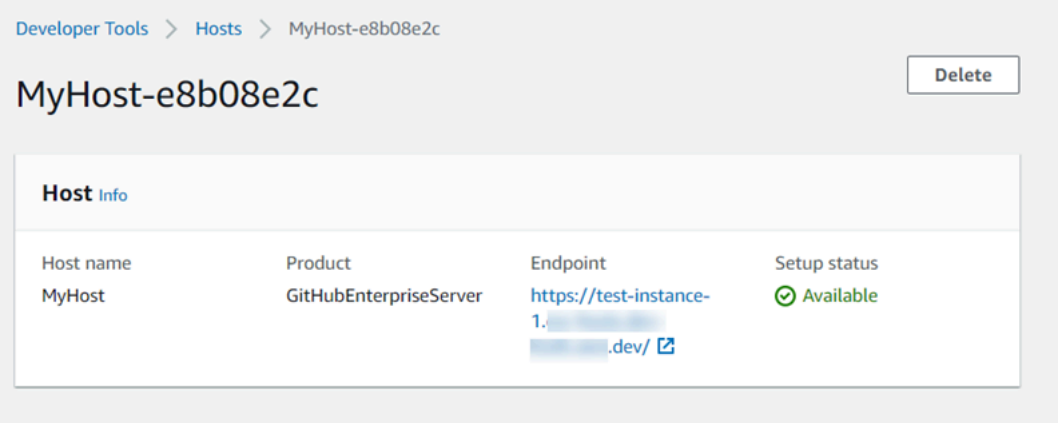
1. 登录到 AWS Management Console 并打开 https://console.aws.amazon.com/codesuite/settings/connections 上的开发工具控制台。
2. 选择 Settings > Connections (设置 > 连接)，然后选择 Hosts (主机) 选项卡。
3. 选择要查看的主机旁边的按钮，然后选择 View details (查看详细信息)。
4. 此时将显示以下主机信息：

   - 主机名。
   - 连接的提供程序类型。
   - 安装提供程序的基础设施的终端节点。
   - 主机的设置状态。准备好连接的主机处于 Available (可用) 状态。如果您的主机已创建但设置尚未完成，则主机可能处于不同的状态。

     以下是可能的状态：

     - PENDING (待处理) - 主机已完成创建，并准备好通过在主机上注册提供程序应用程序开始设置。
     - AVAILABLE (可用) - 主机已完成创建和设置，可用于连接。
     - ERROR (错误) - 主机创建或注册过程中出错。
     - VPC_CONFIG_VPC_INITIALIZING - 正在创建主机的 VPC 配置。
     - VPC_CONFIG_VPC_FAILED_INITIALIZATION - 主机的 VPC 配置遇到错误并失败。

- VPC_CONFIG_VPC_AVAILABLE - 主机的 VPC 配置已完成设置并且可用。
- VPC_CONFIG_VPC_DELETING - 正在删除主机的 VPC 配置。



5. 要删除该主机，请选择 Delete (删除)。
6. 如果主机位于 Pending (待处理) 状态，要完成设置，请选择 Set up host (设置主机)。有关更多信息，请参阅设置待处理的主机。

要查看主机详细信息 (CLI)

- 打开终端 (Linux、macOS 或 Unix) 或命令提示符 (Windows)，并使用 AWS CLI 运行 get-host 命令，指定要查看其详细信息的主机的 Amazon Resource Name (ARN)。

```
aws codestar-connections get-host --host-arn arn:aws:codestar-connections:us-
west-2:account_id:host/My-Host-28aef605
```

此命令将返回以下输出。

```
{
    "Name": "MyHost",
    "Status": "AVAILABLE",
    "ProviderType": "GitHubEnterpriseServer",
    "ProviderEndpoint": "https://test-instance-1.dev/"
}
```

针对已链接存储库使用同步配置

在 C AWS CodeStar onnections 中，您可以使用连接将 AWS 资源关联到第三方存储库，例如 Bitbucket Cloud GitHub、E GitHub nterprise Server 和 GitLab。使用CFN_STACK_SYNC同步类型，您可以创建同步配置，该配置 AWS 允许同步 Git 存储库中的内容以更新指定 AWS 资源。 AWS CloudFormation 与连接集成，因此您可以使用 Git sync 在与之同步的链接存储库中管理模板和参数文件。

创建连接后，您可以使用连接 CLI 或 AWS CloudFormation 控制台来创建存储库链接和同步配置。

- 存储库链接：存储库链接在您的连接和外部 Git 存储库之间创建关联。存储库链接使 Git 同步功能可监控和同步指定 Git 存储库中的文件更改。
- 同步配置：使用同步配置同步 Git 存储库中的内容以更新指定 AWS 资源。

有关更多信息，请参阅《AWS CodeStar 连接 API 参考》。

有关如何使用 AWS CloudFormation 控制台为 AWS CloudFormation 堆栈创建同步配置的教程，请参阅《CloudFormation 用户指南》中的 “使用 AWS CloudFormation Git sync”。

主题

- 使用存储库链接
- 使用同步配置

使用存储库链接

存储库链接会在您的连接和外部 Git 存储库之间创建关联。仓库链接允许 Git sync 监控指定 Git 仓库中文件的更改并将其同步到 AWS CloudFormation 堆栈。

有关存储库链接的更多信息，请参阅 C AWS CodeStar onnections API 参考。

主题

- 创建存储库链接
- 更新存储库链接
- 列出存储库链接
- 删除存储库链接
- 查看存储库链接详细信息

创建存储库链接

您可以使用 AWS Command Line Interface (AWS CLI) 中的create-repository-link命令在连接和要同步到的外部存储库之间创建链接。

在创建存储库链接之前，您必须已经使用第三方提供商创建了外部存储库，例如 GitHub。

创建存储库链接

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行create-repository-link命令。指定关联连接的 ARN、所有者 ID 和存储库名称。

```
aws codestar-connections create-repository-link --connection-arn  arn:aws:codestar-connections:us-east-1:account_id:connection/001f5be2-a661-46a4-b96b-4d277cac8b6e --owner-id account_id --repository-name MyRepo
```

2. 此命令将返回以下输出。

```
{
    "RepositoryLinkInfo": {
        "ConnectionArn": "arn:aws:codestar-connections:us-east-1:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
        "OwnerId": "account_id",
        "ProviderType": "GitHub",
        "RepositoryLinkArn": "arn:aws:codestar-connections:us-east-1:account_id:repository-link/be8f2017-b016-4a77-87b4-608054f70e77",
        "RepositoryLinkId": "be8f2017-b016-4a77-87b4-608054f70e77",
        "RepositoryName": "MyRepo",
        "Tags": []
    }
}
```

更新存储库链接

您可以使用 AWS Command Line Interface (AWS CLI) 中的update-repository-link命令来更新指定的存储库链接。

您可以更新存储库链接的以下信息：

- `--connection-arn`
- `--owner-id`

- `--repository-name`

当您想要更改与存储库关联的连接时，可以更新存储库链接。要使用其他连接，您需要指定连接 ARN。有关查看连接 ARN 的步骤，请参阅查看连接详细信息。

更新存储库链接

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。使用运行update-repository-link命令，为存储库链接指定要更新的值。 AWS CLI 例如，以下命令将更新与存储库链接 ID 关联的连接。它使用 `--connection` 参数指定新的连接 ARN。

```
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. 此命令将返回以下输出。

```
{
    "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": []
    }
}
```

列出存储库链接

您可以使用 AWS Command Line Interface (AWS CLI) 中的list-repository-links命令列出您账户的存储库链接。

列出存储库链接

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行list-repository-links 命令。

```
aws codestar-connections list-repository-links
```

2. 此命令将返回以下输出。

```
{
    "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": []
        }
    ]
}
```

删除存储库链接

您可以使用 AWS Command Line Interface (AWS CLI) 中的delete-repository-link命令来删除存储库链接。

必须先删除与存储库链接关联的所有同步配置，然后才能删除该存储库链接。

> **Important**
>
> 运行该命令后，将删除存储库链接。不显示任何确认对话框。您可以创建新的存储库链接，但无法重复使用 Amazon 资源名称 (ARN)。

删除存储库链接

- 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。使用运行delete-repository-link命令，指定 AWS CLI 要删除的存储库链接的 ID。

```
aws codestar-connections delete-repository-link --repository-link-id
 6053346f-8a33-4edb-9397-10394b695173
```

该命令不返回任何内容。

查看存储库链接详细信息

您可以使用 AWS Command Line Interface (AWS CLI) 中的get-repository-link命令来查看有关存储库链接的详细信息。

查看存储库链接详细信息

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行get-repository-link命令，指定存储库链接 ID。

```
aws codestar-connections get-repository-link --repository-link-id
 6053346f-8a33-4edb-9397-10394b695173
```

2. 此命令将返回以下输出。

```
{
    "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": []
    }
}
```

使用同步配置

同步配置在指定的存储库和连接之间创建关联。使用同步配置可同步 Git 存储库中的内容以更新指定 AWS 资源。

有关连接的更多信息，请参阅AWS CodeStar 连接 API 参考。

主题

- 创建同步配置
- 更新同步配置
- 列出同步配置
- 删除同步配置
- 查看同步配置详细信息

创建同步配置

您可以使用 AWS Command Line Interface (AWS CLI) 中的create-repository-link命令在连接和要同步到的外部存储库之间创建链接。

在创建同步配置之前，您必须已经在连接和第三方存储库之间创建了存储库链接。

创建同步配置

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行create-repository-link命令。指定关联连接的 ARN、所有者 ID 和存储库名称。以下命令将创建一个同步配置，其同步类型适用于 AWS CloudFormation中的资源。它还指定了存储库分支和存储库中的配置文件。在此示例中，资源是一个名为 **mystack** 的堆栈。

```
aws codestar-connections 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. 此命令将返回以下输出。

```
{
    "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"
```

```
    }
```

更新同步配置

您可以在 AWS Command Line Interface (AWS CLI) 中使用 update-sync-configuration 命令更新指定同步配置。

您可以更新同步配置的以下信息：

- `--branch`
- `--config-file`
- `--repository-link-id`
- `--resource-name`
- `--role-arn`

更新同步配置

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。使用运行update-sync-configuration命令，指定要更新的值以及资源名称和同步类型。 AWS CLI 例如，以下命令使用 `--branch` 参数更新与同步配置关联的分支名称。

```
aws codestar-connections update-sync-configuration --sync-type CFN_STACK_SYNC --
resource-name mystack --branch feature-branch
```

2. 此命令将返回以下输出。

```
{
    "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"
    }
```

列出同步配置

您可以在 AWS Command Line Interface (AWS CLI) 中使用 list-sync-configurations 命令列出您账户中的存储库链接。

列出存储库链接

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。使用运行list-sync-configurations命令，指定同步类型和存储库链接 ID。 AWS CLI

```
aws codestar-connections list-sync-configurations --repository-link-id
 6053346f-8a33-4edb-9397-10394b695173  --sync-type CFN_STACK_SYNC
```

2. 此命令将返回以下输出。

```
{
    "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"
        }
    ]
}
```

删除同步配置

您可以在 AWS Command Line Interface (AWS CLI) 中使用 delete-sync-configuration 命令删除同步配置。

> **Important**
>
> 运行该命令后，将删除同步配置。不显示任何确认对话框。您可以创建新的同步配置，但无法重复使用 Amazon 资源名称 (ARN)。

删除同步配置

- 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。使用运行delete-sync-configuration命令，为 AWS CLI 要删除的同步配置指定同步类型和资源名称。

```
aws codestar-connections delete-sync-configuration --sync-type CFN_STACK_SYNC --
resource-name mystack
```

该命令不返回任何内容。

查看同步配置详细信息

您可以使用 AWS Command Line Interface (AWS CLI) 中的get-sync-configuration命令来查看同步配置的详细信息。

查看同步配置详细信息

1. 打开终端（Linux、macOS 或 Unix）或命令提示符（Windows）。 AWS CLI 使用运行get-sync-configuration命令，指定存储库链接 ID。

```
aws codestar-connections get-sync-configuration --sync-type CFN_STACK_SYNC --
resource-name mystack
```

2. 此命令将返回以下输出。

```
{
    "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"
    }
}
```

使用 AWS CodeConnections 记录 AWS CloudTrail API 调用

AWS CodeConnections与 AWS CloudTrail 集成，后者是记录由用户、角色或 AWS 服务所执行的操作的服务。CloudTrail 将所有通知的 API 调用作为事件记录。捕获的调用包含来自开发人员工具控制台的调用和对 AWS CodeConnections API 操作的代码调用。

如果您创建跟踪记录，则可以使 CloudTrail 事件持续传送到 Amazon Simple Storage Service (Amazon S3) 存储桶，包括通知事件。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的 Event history (事件历史记录) 中查看最新事件。使用 CloudTrail 收集的信息，您可以确定向 AWS CodeConnections 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其它详细信息。

有关更多信息，请参阅 AWS CloudTrail 用户指南。

CloudTrail 中的 AWS CodeConnections 信息

在您创建 AWS 账户时，将在该账户上启用 CloudTrail。当 AWS CodeConnections 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在 Event history (事件历史记录) 中。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅 AWS CloudTrail 用户指南中的使用 CloudTrail 事件历史记录查看事件。

要持续记录 AWS 账户中的事件（包括 AWS CodeConnections 的事件），请创建跟踪。通过跟踪记录，CloudTrail 可将日志文件传送至 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有AWS 区域。此跟踪记录在 AWS 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Simple Storage Service（Amazon S3）桶。此外，您可以配置其他AWS服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。

有关更多信息，请参阅 AWS CloudTrail 用户指南 中的以下主题：

- 创建跟踪概览
- CloudTrail 支持的服务和集成
- 为 CloudTrail 配置 Amazon SNS 通知
- 接收来自多个区域的 CloudTrail 日志文件
- 从多个账户接收 CloudTrail 日志文件

CloudTrail 记录所有 AWS CodeConnections操作，AWS CodeConnections API 参考中介绍了这些操作。例如，对 `CreateConnection`、`DeleteConnection` 和 `GetConnection` 操作的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 使用根用户凭证还是其他 IAM 凭证发出请求。
- 请求是使用角色还是联合身份用户的临时安全凭证发出的。
- 请求是否由其它 AWS 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解日志文件条目

跟踪记录是一种配置，可用于将事件作为日志文件传送到您指定的 Simple Storage Service（Amazon S3）存储桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 `CreateConnection` 操作。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type":"IAMUser",
        "principalId":"AIDACKCEVSQ6C2EXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/Mary_Major",
        "accountId":"123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName":"Mary_Major"
    },
    "eventTime": "2020-04-21T01:09:48Z",
    "eventSource": "codestar-connections.amazonaws.com",
    "eventName": "CreateConnection",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "IP",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/80.0.3987.163 Safari/537.36",
    "requestParameters": {
        "providerType": "Bitbucket",
        "connectionName": "my-connection"
    },
    "responseElements": {
        "connectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/7EXAMPLE-5da1-4867-960c-4918175ea3ce"
```

```
    },
    "requestID": "ac1fbc15-a84f-4568-9f90-f05f1a57749c",
    "eventID": "7548f5b0-7ecf-430f-84bf-72e364644359",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

AWS CodeStar 连接和接口 VPC 端点（AWS PrivateLink）

您可以通过创建接口 VPC 端点在 VPC 和 AWS CodeStar 连接之间建立私有连接。接口端点由 AWS PrivateLink 提供支持，该技术支持您通过私密方式访问 AWS CodeStar 连接 API，而无需互联网网关、NAT 设备、VPN 连接或 AWS Direct Connect 连接。VPC 中的实例不需要公有 IP 地址即可与 AWS CodeStar 连接 API 进行通信，因为 VPC 与 AWS CodeStar 连接之间的流量不会离开 Amazon 网络。

每个接口端点均由子网中的一个或多个弹性网络接口表示。

有关更多信息，请参阅 Amazon VPC 用户指南 中的接口 VPC 端点 (AWS PrivateLink)。

AWS CodeStar 连接 VPC 端点的注意事项

请务必先查看《Amazon VPC 用户指南》中的接口端点，然后再为 AWS CodeStar 连接设置接口 VPC 端点。

AWS CodeStar 连接支持从 VPC 调用它的所有 API 操作。

所有 AWS CodeStar 连接区域都支持 VPC 端点。

VPC 终端节点概念

以下是 VPC 终端节点的主要概念：

VPC 终端节点

VPC 中可让您私下连接到服务的入口点。以下是不同类型的 VPC 终端节点。您可以创建受支持的服务所需要的 VPC 终端节点类型。

- AWS CodeStar 连接操作的 VPC 端点
- AWS CodeStar 连接 Webhook 的 VPC 端点

AWS PrivateLink

一种在 VPC 与服务之间提供私有连接的技术。

AWS CodeStar 连接操作的 VPC 端点

您可以管理 AWS CodeStar 连接服务的 VPC 端点。

创建 AWS CodeStar 连接操作的接口 VPC 端点

您可以使用 Amazon VPC 控制台或 AWS Command Line Interface（AWS CLI）为 AWS CodeStar 连接服务创建 VPC 端点。有关更多信息，请参阅 Amazon VPC 用户指南.中的创建接口端点

要开始将连接与 VPC 结合使用，请为 AWS CodeStar 连接创建接口 VPC 端点。当您为 AWS CodeStar 连接创建 VPC 端点时，请选择 AWS Services（AWS 服务），并在 Service Name（服务名称）中选择：

- com.amazonaws.***region***.codestar-connections.api：此选项为 AWS CodeStar 连接 API 操作创建 VPC 端点。例如，如果您的用户使用 AWS CLI、AWS CodeStar 连接 API 或 AWS SDK 与 AWS CodeStar 连接交互以执行诸如 `CreateConnection`、`ListConnections` 和 `CreateHost` 等操作，则选择此选项。

对于 Enable DNS name（启用 DNS 名称）选项，如果为端点启用私有 DNS，则可以将其原定设置 DNS 名称用于区域以向 AWS CodeStar 连接发送 API 请求，例如 `codestar-connections.us-east-1.amazonaws.com`。

> **Important**
>
> 原定设置情况下，对于为 AWS 服务和 AWS Marketplace Partner 服务创建的端点会启用私有 DNS。

有关更多信息，请参阅 Amazon VPC 用户指南中的通过接口端点访问服务。

创建 AWS CodeStar 连接操作的 VPC 端点策略

您可以为 VPC 端点附加端点策略，以控制对 AWS CodeStar 连接的访问。该策略指定以下信息：

- 可执行操作的委托人。
- 可执行的操作。
- 可对其执行操作的资源。

有关更多信息，请参阅 Amazon VPC 用户指南中的使用 VPC 终端节点控制对服务的访问。

> **Note**
>
> com.amazonaws.*region*.codestar-connections.webhooks 终端节点不支持策略。

示例：AWS CodeStar 连接操作的 VPC 端点策略

以下是适用于 AWS CodeStar 连接的端点策略示例。当附加到端点时，此策略会向所有资源上的所有主体授予对列出的 AWS CodeStar 连接操作的访问权限。

```
{
  "Statement": [
    {
      "Sid": "GetConnectionOnly",
      "Principal": "*",
      "Action": [
        "codestar-connections:GetConnection"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

AWS CodeStar 连接 Webhook 的 VPC 端点

当您创建或删除具有 VPC 配置的主机时，AWS CodeStar 连接会为您创建 Webhook 端点。终端节点名为 com.amazonaws.***region***.codestar-connections.webhooks。

使用适用于 GitHub Webhook 的 VPC 端点，主机可以通过 Webhook 将事件数据发送到 Amazon 网络上的集成 AWS 服务。

> **Important**
>
> 当您为 GitHub Enterprise Server 设置主机时，AWS CodeStar 连接会为您创建一个 VPC 端点，用于 Webhook 事件数据。如果您在 2020 年 11 月 24 日之前创建了主机，并且希望使用 VPC PrivateLink Webhook 终端节点，则必须先删除您的主机，然后创建一个新的主机。

AWS CodeStar 连接管理这些端点的生命周期。要删除终端节点，您必须删除相应的主机资源。

如何使用 AWS CodeStar 连接主机的 Webhook 端点

Webhook 端点是发送来自第三方存储库的 Webhook 以进行 AWS CodeStar 连接处理的地方。Webhook 描述了客户操作。当您执行 `git push`，Webhook 终端节点会从提供程序接收一个详细说明推送的 Webhook。例如，AWS CodeStar 连接可以通知 CodePipeline 启动您的管道。

对于不使用 VPC 的云提供程序（如 Bitbucket 或 GitHub Enterprise Server 主机），Webhook VPC 端点不适用，因为提供程序正在向未使用 Amazon 网络的 AWS CodeStar 连接发送 Webhook。

排除连接故障

以下信息可帮助您排查有关 AWS CodeBuild、AWS CodeDeploy 和 AWS CodePipeline 中资源连接的常见问题。

主题

- 我无法创建连接
- 当我尝试创建或完成连接时出现权限错误
- 当我尝试使用连接时出现权限错误
- Connection is not in available state or is no longer pending (确保您具有使用连接的权限，包括在提供程序位置列出可用存储库。连接不处于可用状态或不再处于待处理状态)
- 添加连接的 GitClone 权限
- 主机不处于可用状态
- 故障排查有连接错误的主机
- 我无法为主机创建连接
- 对主机的 VPC 配置进行故障排查
- GitHub Enterprise Server 连接的 Webhook VPC 终端节点 (PrivateLink) 故障排查
- 对 2020 年 11 月 24 日之前创建的主机进行故障排查
- 无法为 GitHub 存储库创建连接
- 编辑您的 GitHub Enterprise Server 连接应用程序权限
- 连接到 GitHub 时出现连接错误：“出现问题，请确保在浏览器中启用 Cookie” 或“组织拥有者必须安装 GitHub 应用程序”
- 我想提高连接的限制

我无法创建连接

您可能无权创建连接。有关更多信息，请参阅的权限和示例 AWS CodeConnections。

当我尝试创建或完成连接时出现权限错误

当您尝试在 CodePipeline 控制台中创建或查看连接时，可能会返回以下错误消息。

User: *username* is not authorized to perform: *permission* on resource: *connection-ARN*

If this message appears, make sure that you have sufficient permissions. (用户：username 无权对资源：connection-ARN 执行操作：permission。如果出现此消息，请确保您有足够的权限。)

在 AWS Command Line Interface (AWS CLI) 或 AWS Management Console 中创建和查看连接的权限只是您是在控制台上创建和完成连接所需之权限的一部分。只执行查看、编辑或创建连接，然后完成待处理连接所需的权限，应该分配给仅需执行某些任务的用户。有关更多信息，请参阅的权限和示例 AWS CodeConnections。

当我尝试使用连接时出现权限错误

如果您尝试在 CodePipeline 控制台中使用连接，即使您有列出、获取和创建权限，也可能会返回以下一个或两个错误消息。

您无法对您的账户进行身份验证。

User: *username* is not authorized to perform: codestar-connections:UseConnection on resource: *connection-ARN*

If this occurs, make sure that you have sufficient permissions. (您帐户的身份验证失败。用户：username 无权对资源：connection-ARN 执行：codestar-connections:UseConnection。如出现此消息，请确保您有足够的权限。)

Make sure you have the permissions to use a connection, including listing the available repositories in the provider location。有关更多信息，请参阅的权限和示例 AWS CodeConnections。

Connection is not in available state or is no longer pending (确保您具有使用连接的权限，包括在提供程序位置列出可用存储库。连接不处于可用状态或不再处于待处理状态)

如果控制台显示连接不处于可用状态的消息，请选择 Complete connection (完成连接)。

如果选择完成连接，并显示一条消息，指出连接未处于待处理状态，则可以取消请求，因为连接已处于可用状态。

添加连接的 GitClone 权限

当您在源操作和 CodeBuild 操作中使用 AWS CodeStar 连接时，可以通过两种方式将输入构件传递到生成包：

- 默认方式：源操作生成一个 zip 文件，其中包含 CodeBuild 下载的代码。
- Git 克隆：源代码可以直接下载到构建环境中。

  Git 克隆模式允许您将源代码作为工作 Git 存储库进行交互。要使用此模式，您必须授予 CodeBuild 环境使用连接的权限。

要向 CodeBuild 服务角色策略添加权限，请创建附加到 CodeBuild 服务角色的客户管理策略。以下步骤会创建一个策略，其中 `UseConnection` 权限在 `action` 字段中指定，而 Amazon Resource Name (ARN) 在 `Resource` 字段中指定。

使用控制台添加 UseConnection 权限

1. 要查找管道的连接 ARN，请打开管道，然后在源操作中选择 (i) 图标。此时将打开 Configuration (配置) 窗格，连接 ARN 显示在 ConnectionArn 旁边。您将连接 ARN 添加到您的 CodeBuild 服务角色策略。
2. 要查找 CodeBuild 服务角色，请打开管道中使用的构建项目，然后导航到 Build details (构建详细信息) 选项卡。
3. 在 Environment (环境) 部分，选择 Service role (服务角色) 链接。此时将打开 AWS Identity and Access Management (IAM) 控制台，您可以在其中添加新策略，以授予对连接的访问权限。
4. 在 IAM 控制台中，选择 Attach policies (附加策略)，然后选择 Create policy (创建策略)。

   使用以下示例策略模板。将您的连接 ARN 添加到 `Resource` 字段，如此示例中所示。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "codestar-connections:UseConnection",
            "Resource": "insert connection ARN here"
        }
```

```
    ]
}
```

在存储库的 JSON 选项卡上，粘贴您的策略。

5. 选择查看策略。为策略输入名称（例如 **connection-permissions**），然后选择 Create policy (创建策略)。
6. 返回到服务角色 Attach Permissions (附加权限) 页面上，刷新策略列表，然后选择您刚才创建的策略。选择附加策略。

主机不处于可用状态

如果控制台显示一条消息，指出主机不处于 `Available` 状态，则选择 Set up host (设置主机)。

创建主机的第一步将导致创建的主机现在处于 `Pending` 状态。将主机变为 `Available` 状态时，必须选择在控制台中设置主机。有关更多信息，请参阅设置待处理的主机。

> **Note**
>
> 您无法使用 AWS CLI 设置 `Pending` 主机。

故障排查有连接错误的主机

如果底层 GitHub 应用程序被删除或修改，连接和主机可能变为错误状态。无法恢复处于错误状态的主机和连接，必须重新创建主机。

- 更改应用程序 pem 密钥、更改应用名称（初始创建后）等操作将导致主机和所有相关连接变为错误状态。

如果控制台或 CLI 返回 `Error` 状态的主机或与此类主机相关的连接，您可能需要执行以下步骤：

- 删除并重新创建主机资源，然后重新安装主机注册应用程序。有关更多信息，请参阅创建主机。

我无法为主机创建连接

要创建连接或主机，必须满足以下条件。

- 您的主机必须处于 AVAILABLE (可用) 状态。有关更多信息，请参阅
- 连接必须在与主机相同的区域中创建。

对主机的 VPC 配置进行故障排查

创建主机资源时，必须为安装 GitHub Enterprise Server 实例的基础设施提供网络连接或 VPC 信息。要对主机的 VPC 或子网配置进行故障排查，请使用此处显示的示例 VPC 信息作为参考。

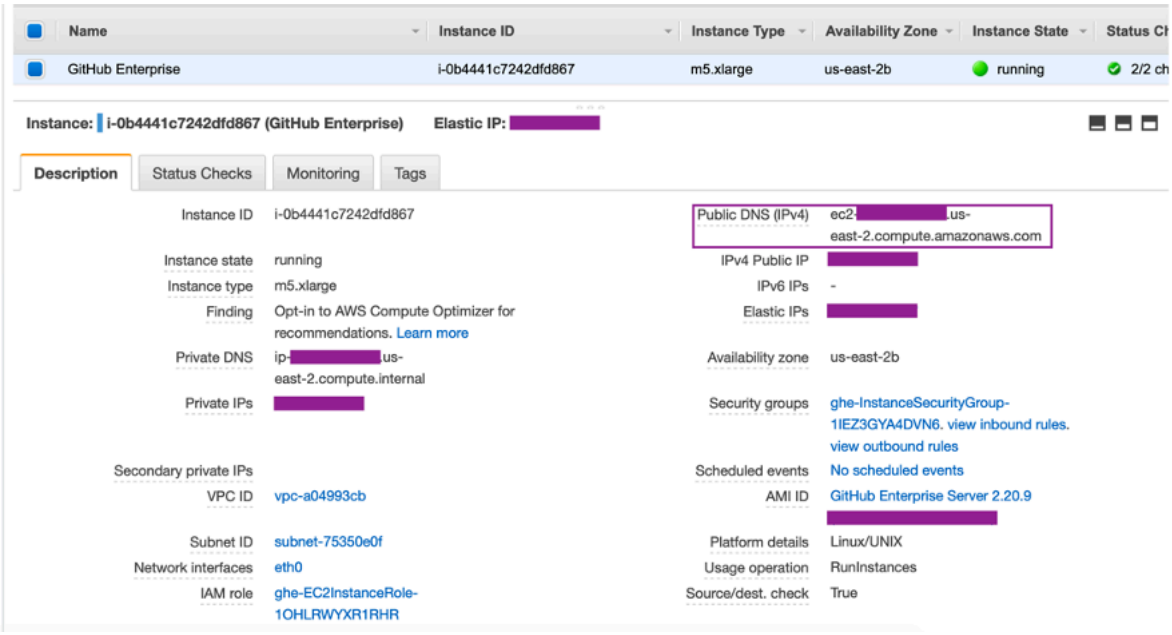
> **Note**
>
> 使用此部分进行与 Amazon VPC 中 GitHub Enterprise Server 主机配置相关的故障排查。有关为使用 VPC 的 Webhook 终端节点 (PrivateLink) 而配置的连接的故障排查，请参阅 GitHub Enterprise Server 连接的 Webhook VPC 终端节点 (PrivateLink) 故障排查。

在此示例中，您将使用以下过程配置要安装 GitHub Enterprise Server 实例的 VPC 和服务器：

1. 创建 VPC。有关更多信息，请参阅https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#Create-VPC。
2. 在您的 VPC 中创建子网。有关更多信息，请参阅https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#AddaSubnet。
3. 将实例启动到 VPC 中。有关更多信息，请参阅https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#VPC_Launch_Instance。

> **Note**
>
> 每个 VPC 一次只能与一个主机（GitHub Enterprise Server 实例）关联。

下图显示了使用 GitHub Enterprise AMI 启动的 EC2 实例。



当您将 VPC 用于 GitHub Enterprise Server 连接时，您必须在设置主机时为您的基础设施提供以下内容：

- VPC ID：安装 GitHub Enterprise Server 实例的服务器的 VPC，或通过 VPN 或 Direct Connect 访问已安装的 GitHub Enterprise Server 实例的 VPC。
- 子网 ID：安装 GitHub Enterprise Server 实例的服务器的子网，或通过 VPN 或 Direct Connect 访问已安装的 GitHub Enterprise Server 实例的子网。
- 安全组：安装 GitHub Enterprise Server 实例的服务器的安全组，或通过 VPN 或 Direct Connect 访问已安装的 GitHub Enterprise Server 实例的安全组。
- 终端节点：准备好您的服务器终端节点，并继续下一步。

有关使用 VPC 和子网的更多信息，请参阅 Amazon VPC 用户指南中的 为 IPv4 进行 VPC 和子网定型。

主题

- 我无法获取处于待处理状态的主机
- 我无法获取处于可用状态的主机
- 我的连接/主机刚才正常现在却停止工作
- 我无法删除我的网络接口

我无法获取处于待处理状态的主机

如果您的主机进入 VPC_CONFIG_FAILED_INITIALIZATION 状态，这很可能是因为您为主机选择的 VPC、子网或安全组存在问题。

- VPC、子网和安全组必须全部属于创建主机的账户。
- 子网和安全组必须属于所选 VPC。
- 每个提供的子网必须位于不同的可用区。
- 创建主机的用户必须具有以下 IAM 权限：

```
ec2:CreateNetworkInterface
ec2:CreateTags
ec2:DescribeDhcpOptionsec2:DescribeNetworkInterfaces
ec2:DescribeSubnets
ec2:DeleteNetworkInterface
ec2:DescribeVpcs
ec2:CreateVpcEndpoint
ec2:DeleteVpcEndpoints
ec2:DescribeVpcEndpoints
```

我无法获取处于可用状态的主机

如果您无法为您的主机完成 AWS CodeStar 连接应用程序设置，可能是由于您的 VPC 配置或 GitHub Enterprise Server 实例存在问题。

- 如果您不使用公有证书颁发机构，则需要向您的主机提供由 GitHub Enterprise 实例使用的 TLS 证书。TLS 证书值应该是证书的公有密钥。
- 您需要成为 GitHub Enterprise Server 实例的管理员才能创建 GitHub 应用程序。

我的连接/主机刚才正常现在却停止工作

如果连接/主机之前正常工作，但现在不工作，则可能是由于 VPC 中的配置更改或 GitHub 应用程序已被修改。请检查以下事项：

- 连接到您为连接创建的主机资源的安全组现在已更改或不再具有对 GitHub Enterprise Server 的访问权限。AWSCodeStar 连接需要具有与 GitHub Enterprise Server 实例连接的安全组。
- DNS 服务器 IP 最近发生了更改。您可以通过检查连接到您为连接创建的主机资源中指定的 VPC 的 DHCP 选项来验证这一点。请注意，如果您最近从 AmazonProvidedDNS 迁移到自定义 DNS 服务

器或开始使用新的自定义 DNS 服务器，则主机/连接将停止工作。为了解决这个问题，请删除您现有的主机并重新创建它，这将在我们的数据库中存储最新的 DNS 设置。

- 网络 ACL 设置已更改，不再允许 HTTP 连接到 GitHub Enterprise Server 基础设施所在的子网。
- GitHub Enterprise Server 上的 AWS CodeStar 连接应用程序有配置发生更改。对任何配置（例如 URL 或应用程序密钥）的修改都可能会中断已安装的 GitHub Enterprise Server 实例与 AWS CodeStar 连接之间的连接。

我无法删除我的网络接口

如果无法检测到网络接口，请确认以下内容：

- 由 AWS CodeStar 连接创建的网络接口只能通过删除主机来删除。用户不能手动删除它们。
- 您必须拥有以下权限：

```
ec2:DescribeNetworkInterfaces
ec2:DeleteNetworkInterface
```

GitHub Enterprise Server 连接的 Webhook VPC 终端节点 (PrivateLink) 故障排查

当您创建具有 VPC 配置的主机时，会为您创建 Webhook VPC 终端节点。

> **Note**
>
> 有关为使用 VPC 的 Webhook 终端节点 (PrivateLink) 而配置的连接的故障排查，请使用此部分内容。有关 Amazon VPC 中与 GitHub Enterprise Server 主机配置相关的故障排查，请参阅对主机的 VPC 配置进行故障排查。

当您创建到已安装的提供程序类型的连接，并已指定您的服务器在 VPC 中配置时，AWS CodeStar 连接会创建您的主机，并为您创建适用于 Webhook 的 VPC 端点（PrivateLink）。这使主机能够通过 Webhook 将事件数据发送到 Amazon 网络上集成的 AWS 服务。有关更多信息，请参阅AWS CodeStar 连接和接口 VPC 端点（AWS PrivateLink）。

主题

- 我无法删除我的 Webhook VPC 终端节点

我无法删除我的 Webhook VPC 终端节点

AWS CodeStar 连接为您的主机管理 Webhook VPC 端点的生命周期。要删除终端节点，您必须删除相应的主机资源。

- 由 AWS CodeStar 连接创建的 Webhook VPC 端点（PrivateLink）只能通过删除主机来删除。无法手动删除它们。
- 您必须拥有以下权限：

```
ec2:DescribeNetworkInterfaces
ec2:DeleteNetworkInterface
```

对 2020 年 11 月 24 日之前创建的主机进行故障排查

自 2020 年 11 月 24 日起，当 AWS CodeStar 连接设置您的主机时，将为您设置额外的 VPC 端点（PrivateLink）支持。对于在此更新之前创建的主机，请使用此故障排查内容。

有关更多信息，请参阅AWS CodeStar 连接和接口 VPC 端点（AWS PrivateLink）。

主题

- 我有一个在 2020 年 11 月 24 日之前创建的主机，我想将 VPC 终端节点 (PrivateLink) 用于 Webhook
- 我无法获取处于可用状态的主机（VPC 错误）

我有一个在 2020 年 11 月 24 日之前创建的主机，我想将 VPC 终端节点 (PrivateLink) 用于 Webhook

当您为 GitHub Enterprise Server 设置主机时，会为您创建 Webhook 终端节点。连接现在使用 VPC PrivateLink Webhook 终端节点。如果您在 2020 年 11 月 24 日之前创建了主机，并且希望使用 VPC PrivateLink Webhook 终端节点，则必须先删除您的主机，然后创建一个新的主机。

我无法获取处于可用状态的主机（VPC 错误）

如果您的主机是在 2020 年 11 月 24 日之前创建的，而且您无法为您的主机完成 AWS CodeStar 连接应用程序设置，则可能是由于您的 VPC 配置或 GitHub Enterprise Server 实例存在问题。

您的 VPC 将需要 NAT 网关（或出站互联网访问），以便您的 GitHub Enterprise Server 实例可以为 GitHub Webhook 发送出口网络流量。

无法为 GitHub 存储库创建连接

问题：

由于到 GitHub 存储库的连接使用 AWS Connector for GitHub，因此您需要组织所有者权限或存储库的管理员权限才能创建连接。

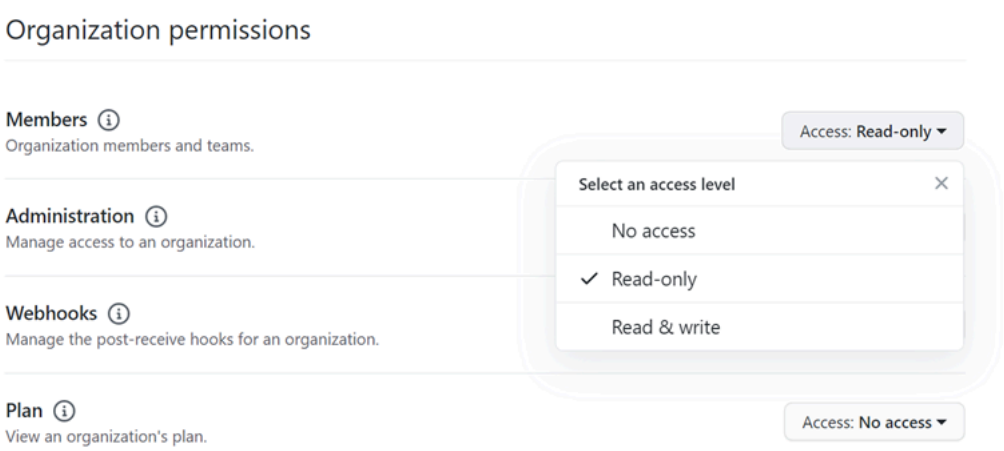
可能的修复措施：有关 GitHub 存储库的权限级别的信息，请参阅 https://docs.github.com/en/free-pro-team@latest/github/setting-up-and-managing-organizations-and-teams/permission-levels-for-an-organization。

编辑您的 GitHub Enterprise Server 连接应用程序权限

如果您在 2020 年 12 月 23 日或之前安装了适用于 GitHub Enterprise Server 的应用程序，则可能需要向组织成员授予该应用程序的只读访问权限。如果您是 GitHub 应用程序所有者，请按照以下步骤编辑创建主机时安装的应用程序的权限。

> **Note**
>
> 您必须在 GitHub Enterprise Server 实例上完成这些步骤，并且您必须是 GitHub 应用程序所有者。

1. 在 GitHub Enterprise Server 中，从您的资料照片上的下拉列表选项中选择 Settings (设置)。
2. 选择 Developer settings (开发人员设置)，然后选择 GitHub Apps (GitHub 应用程序)。
3. 在应用程序列表中，为您的连接选择应用程序的名称，然后在设置显示中选择 Permissions and events (权限和事件)。
4. 在 Organization permissions（组织权限）下，对于 Members（成员），从 Access（访问权限）下拉列表中选择 Read-only（只读）。



5. 在 Add a note to users (向用户添加注释) 下，添加更新原因的说明。选择保存更改。

连接到 GitHub 时出现连接错误：“出现问题，请确保在浏览器中启用 Cookie” 或“组织拥有者必须安装 GitHub 应用程序”

问题：

要为 GitHub 存储库创建连接，您必须是 GitHub 组织的拥有者。对于不属于组织的存储库，您必须是存储库拥有者。当连接由非组织拥有者的其他用户创建时，将针对组织拥有者创建请求，并显示以下错误之一：

出现问题，请确保在浏览器中启用 Cookie

或

组织拥有者必须安装 GitHub 应用程序

可能的修复：对于 GitHub 组织中的存储库，组织拥有者必须创建与 GitHub 存储库的连接。对于不属于组织的存储库，您必须是存储库拥有者。

我想提高连接的限制

您可以请求提高 AWS CodeStar 连接中的某些限制。有关更多信息，请参阅连接的配额。

连接的配额

下表列出了开发工具控制台中连接的配额（也称为限制）。

此表中的限额按 AWS 区域应用，并且可以提高。要请求提高限制，请使用支持中心控制台。有关 AWS 区域和可以更改的限额的信息，请参阅 AWS service quotas。

> **Note**
>
> 对于欧洲地区（米兰）AWS 区域，必须先启用此区域，然后才能使用它。有关更多信息，请参阅启用区域。

资源	默认限制
每个 AWS 账户的最大连接数	250

此表中的配额是固定的，不能更改。

资源	默认限制
连接名称中的最大字符数	32 个字符
每个 AWS 账户的最大主机数量	50
最大存储库链接数	100
AWS CloudFormation 堆栈同步配置的最大数量	100
每个存储库链接的最大同步配置数	100
每个分支的最大同步配置数	50

要添加到允许列表的 IP 地址

如果您实施 IP 筛选或允许在 Amazon EC2 实例上使用某些 IP 地址，请将以下 IP 地址添加到您的允许列表中。这样做可以连接到提供商，例如 GitHub 和 Bitbucket。

下表按 AWS 区域列出了开发人员工具控制台中连接的 IP 地址。

> **Note**
>
> 对于欧洲地区（米兰）区域，必须先启用此区域，然后才能使用它。有关更多信息，请参阅启用区域。

区域	IP 地址
美国西部（俄勒冈州）(us-west-2)	35.160.210.199、54.71.206.108、54.71.36.205
美国东部（弗吉尼亚州北部）(us-east-1)	3.216.216.90、3.216.243.220、3.217.241.85
欧洲地区（爱尔兰）(eu-west-1)	34.242.64.82、52.18.37.201、54.77.75.62
美国东部（俄亥俄州）(us-east-2)	18.217.188.190、18.218.158.91、18.220.4.80
亚太地区（新加坡）(ap-southeast-1)	18.138.171.151、18.139.22.70、3.1.157.176
亚太地区（悉尼）(ap-southeast-2)	13.236.59.253、52.64.166.86、54.206.1.112
亚太地区（东京）(ap-northeast-1)	52.196.132.231、54.95.133.227、18.181.13.91
欧洲地区（法兰克福）(eu-central-1)	18.196.145.164、3.121.252.59、52.59.104.195
亚太地区（首尔）(ap-northeast-2)	13.125.8.239、13.209.223.177、3.37.200.23
亚太地区（孟买）(ap-south-1)	13.234.199.152、13.235.29.220、35.154.230.124
南美洲（圣保罗）（sa-east-1）	18.229.77.26、54.233.226.52、54.233.207.69
加拿大（中部）(ca-central-1)	15.222.219.210、35.182.166.138、99.79.111.198
欧洲（伦敦）(eu-west-2)	3.9.97.205、35.177.150.185、35.177.200.225
美国西部（北加利福尼亚）(us-west-1)	52.52.16.175、52.8.63.87
欧洲地区（巴黎）（eu-west-3）	35.181.127.138、35.181.145.22、35.181.20.200

区域	IP 地址
欧洲地区（斯德哥尔摩）(eu-north-1)	13.48.66.148、13.48.8.79、13.53.78.182
欧洲（米兰）(eu-south-1)	18.102.28.105、18.102.35.130、18.102.8.116

开发工具控制台功能的安全性

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的 安全性和云中 的安全性：

- 云安全 — AWS 负责保护在 AWS 云中运行 AWS 服务的基础架构。 AWS 还为您提供可以安全使用的服务。作为 AWS 合规性计划的一部分，第三方审核人员将定期测试和验证安全性的有效性。要了解适用于 AWS CodeStar 通知和 AWS CodeStar 连接的合规性计划，请参阅按合规性计划划分的范围内的AWS 服务。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档可帮助您了解在使用 AWS CodeStar 通知和 AWS CodeStar 连接时如何应用分担责任模型。以下主题向您介绍如何配置 AWS CodeStar 通知和 AWS CodeStar 连接以满足您的安全和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 AWS CodeStar 通知和 AWS CodeStar 连接资源。

有关开发工具控制台中服务的安全性的更多信息，请参阅以下内容：

- CodeBuild 安全
- CodeCommit 安全
- CodeDeploy 安全
- CodePipeline 安全

了解通知内容和安全性

通知向订阅您配置的通知规则目标的用户提供有关资源的信息。此等信息可以包括有关开发工具资源的详细信息，包括存储库内容、构建状态、部署状态和管道执行。

例如，您可以为中的仓库配置通知规则，使其包含 CodeCommit 对提交或拉取请求的评论。如果是这样，响应该规则而发送的通知可能包含该注释中引用的一行或多行代码。同样，您可以在中为构建项目配置通知规则， CodeBuild 以包括生成状态和阶段的成功或失败。响应该规则发送的通知将包含该信息。

您可以为中的管道配置通知规则， CodePipeline 使其包含有关手动批准的信息，而为响应该规则而发送的通知可能包含提供该批准的人员的姓名。您可以在中为应用程序配置通知规则 CodeDeploy 以指示部署成功，而为响应该规则而发送的通知可能包含有关部署目标的信息。

通知可以包括特定于项目的信息，例如构建状态、具有注释的代码行、部署状态和管道批准。因此为了帮助确保项目的安全性，请确保定期检查通知规则的目标以及指定为目标的 Amazon SNS 主题的订阅者列表。此外，随着将附加功能添加到基础服务中，响应事件而发送的通知的内容可能会更改。发生此等更改时不通知已存在的通知规则。考虑定期查看通知消息的内容，以帮助确保您了解发送的内容以及发送给谁。

有关可用于通知规则的事件类型的更多信息，请参阅 通知概念。

您可以选择将通知中包含的详细信息限制为仅包含在事件中的内容。这被称为 Basic (基本) 详细信息类型。这些事件包含的信息与发送给亚马逊 EventBridge 和亚马逊 CloudWatch 活动的信息完全相同。

开发者工具控制台服务（例如 CodeCommit）可能会选择在通知消息中添加有关其部分或全部事件类型的信息，而不是活动中提供的信息。此补充信息可随时添加，以增强当前的活动类型或补充未来的活动类型。您可以通过选择 Full (完整) 详细信息类型，在通知中包含有关事件的所有补充信息。有关更多信息，请参阅 详细信息类型。

AWS CodeStar 通知和 AWS CodeStar 连接中的数据保护

AWS 责任共担模式适用于 AWS CodeStar 通知和 AWS CodeStar 连接中的数据保护。如该模式中所述，AWS 负责保护运行所有 AWS Cloud 的全球基础设施。您负责维护对托管在此基础设施上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS 安全性博客 上的 AWS 责任共担模式和 GDPR 博客文章。

出于数据保护目的，建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置单个用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 使用 SSL/TLS 与 AWS 资源进行通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用 AWS CloudTrail 设置 API 和用户活动日记账记录。
- 使用 AWS 加密解决方案以及 AWS 服务 中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。

- 如果在通过命令行界面或 API 访问 AWS 时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准 (FIPS) 第 140-2 版》。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如 Name（名称）字段）。这包括使用控制台、API、AWS CLI 或 AWS SDK 处理 AWS CodeStar 通知和 AWS CodeStar 连接或其他 AWS 服务时。您在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日记账。如果您向外部服务器提供 URL，我们强烈建议您不要在 URL 中包含凭证信息来验证您对该服务器的请求。

AWS CodeStar 通知和 AWS CodeStar 连接的身份和访问管理

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（有权限）使用 AWS CodeStar 通知和 AWS CodeStar 连接资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

主题

- 受众
- 使用身份进行身份验证
- 使用策略管理访问
- 开发工具控制台中的特征如何与 IAM 配合使用
- AWS CodeConnections 权限参考
- 基于身份的策略示例
- 使用标签控制对 C AWS CodeStar onnections 资源的访问权限
- 使用控制台中的通知和连接
- 允许用户查看他们自己的权限
- 排除 AWS CodeStar 通知和 AWS CodeStar 连接身份和访问权限故障
- 为 AWS CodeStar 通知使用服务相关角色
- 将服务相关角色用于 AWS CodeConnections
- 适用于 AWS CodeConnections 的 AWS 托管策略

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您在 AWS CodeStar 通知和 AWS CodeStar 连接中所做的工作。

服务用户-如果您使用 AWS CodeStar 通知和 AWS CodeStar 连接服务完成工作，则您的管理员会为您提供所需的凭证和权限。当你使用更多的 AWS CodeStar 通知和 AWS CodeStar 连接功能来完成工作时，你可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 “ AWS CodeStar 通知和 AWS CodeStar 连接” 中的某项功能，请参阅排除 AWS CodeStar 通知和 AWS CodeStar 连接身份和访问权限故障。

服务管理员-如果您负责公司的 AWS CodeStar 通知和 AWS CodeStar 连接资源，则可能拥有对 AWS CodeStar 通知和 AWS CodeStar 连接的完全访问权限。您的工作是确定您的服务用户应访问哪些 AWS CodeStar 通知和 AWS CodeStar 连接功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要详细了解贵公司如何使用带 AWS CodeStar 通知和 AWS CodeStar 连接的 IAM，请参阅开发工具控制台中的特征如何与 IAM 配合使用。

IAM 管理员 — 如果您是 IAM 管理员，则可能需要详细了解如何编写策略来管理对 AWS CodeStar 通知和 AWS CodeStar 连接的访问权限。要查看您可以在 IAM 中使用的基于身份的 AWS CodeStar 通知和 AWS CodeStar 连接策略示例，请参阅。 基于身份的策略示例

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。 AWS IAM Identity Center （IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的如何登录到您 AWS 账户的。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM 用户指南中的签署 AWS API 请求。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如， AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的 多重身份验证 和《IAM 用户指南》中的 在 AWS中使用多重身份验证（MFA）。

AWS 账户根用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务 和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关需要您以根用户身份登录的任务的完整列表，请参阅 IAM 用户指南 中的 需要根用户凭证的任务。

IAM 用户和群组

I AM 用户是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，我们建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的对于需要长期凭证的使用场景定期轮换访问密钥。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅 IAM 用户指南中的何时创建 IAM 用户（而不是角色）。

IAM 角色

I AM 角色是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定人员不关联。您可以 AWS Management Console 通过切换角色在中临时担任 IAM 角色。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的 使用 IAM 角色。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的 为第三方身份提供商创建角色。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的权限集。
- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附

加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同。

- 跨服务访问 — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，此操作然后在不同服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。
  - 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。
  - 服务相关角色-服务相关角色是一种链接到的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 A@@ mazon EC2 上运行的应用程序 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的 使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限。

要了解是使用 IAM 角色还是 IAM 用户，请参阅 IAM 用户指南中的何时创建 IAM 角色（而不是用户）。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS ，当与身份或资源关联时，它会定义其权限。 AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅《IAM 用户指南》中的 JSON 策略概览。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM policy。管理员随后可以向角色添加 IAM policy，用户可以代入角色。

IAM policy 定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅 IAM 用户指南中的创建 IAM policy。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管策略和内联策略之间进行选择，请参阅 IAM 用户指南中的在托管策略与内联策略之间进行选择。

开发工具控制台中的特征如何与 IAM 配合使用

在使用 IAM 管理对开发工具控制台中特征的访问之前，您应了解哪些 IAM 特征可与其结合使用。要全面了解通知和其他 AWS 服务如何与 IAM 配合使用，请参阅 IAM 用户指南中的与 IAM 配合使用的 AWS 服务。

主题

- 开发工具控制台中基于身份的策略
- AWS CodeStar 基于资源的通知和 AWS CodeStar 连接策略
- 基于标签的授权
- IAM 角色

开发工具控制台中基于身份的策略

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。AWS CodeStar 通知和 AWS CodeStar 连接支持特定的操作、资源和条件键。要了解在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素参考。

操作

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么 条件 下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

开发工具控制台中的通知的策略操作在操作前使用以下前缀：codestar-notifications and codestar-connections。例如，要授予某人查看其账户中所有通知规则的权限，请将 codestar-notifications:ListNotificationRules 操作包括在其策略中。策略声明必须包含Action或NotAction元素。 AWS CodeStar 通知和 AWS CodeStar 连接定义了自己的一组操作，这些操作描述了您可以使用此服务执行的任务。

要在单个语句中指定多个 AWS CodeStar 通知操作，请用逗号分隔它们，如下所示。

```
"Action": [
      "codestar-notifications:action1",
      "codestar-notifications:action2"
```

要在单个语句中指定多个 AWS CodeConnections 操作，请用逗号分隔它们，如下所示。

```
"Action": [
      "codestar-connections:action1",
      "codestar-connections:action2"
```

您也可以使用通配符 (*) 指定多个操作。例如，要指定以单词 List 开头的所有操作，请包括以下操作。

```
"Action": "codestar-notifications:List*"
```

AWS CodeStar 通知 API 操作包括：

- CreateNotificationRule
- DeleteNotificationRule
- DeleteTarget

- DescribeNotificationRule
- ListEventTypes
- ListNotificationRules
- ListTagsForResource
- ListTargets
- Subscribe
- TagResource
- Unsubscribe
- UntagResource
- UpdateNotificationRule

AWS CodeConnections API 操作包括以下内容：

- CreateConnection
- DeleteConnection
- GetConnection
- ListConnections
- ListTagsForResource
- TagResource
- UntagResource

要完成身份验证握手，需要执行以下仅限权限的操作： AWS CodeConnections

- GetIndividualAccessToken
- GetInstallationUrl
- ListInstallationTargets
- StartOAuthHandshake
- UpdateConnectionInstallation

要使用连接，需要执行以下仅限权限 AWS CodeConnections 的操作：

- UseConnection

要将连接传递 AWS CodeConnections 到服务，需要执行以下仅限权限的操作：

- PassConnection

要查看 AWS CodeStar 通知和 AWS CodeStar 连接操作列表，请参阅 IAM 用户指南中的AWS CodeStar 通知定义的操作和 AWS CodeStar 连接定义的操作。

资源

AWS CodeStar 通知和 AWS CodeStar 连接不支持在策略中指定资源 ARN。

条件键

AWS CodeStar 通知和 AWS CodeStar 连接定义自己的条件键集，还支持使用一些全局条件键。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

所有 AWS CodeStar 通知操作都支持codestar-notifications:NotificationsForResource条件键。有关更多信息，请参阅 基于身份的策略示例。

AWS CodeConnections 定义以下可在 IAM 策略Condition元素中使用的条件键。您可以使用这些键进一步细化应用策略语句的条件。有关更多信息，请参阅 AWS CodeConnections 权限参考。

条件键	描述
codestar-connections:BranchName	按第三方存储库的分支名称筛选访问权限。
codestar-connections:FullRepositoryId	按请求中传递的存储库来筛选访问权限。仅适用于访问特定存储库的 UseConnection 请求
codestar-connections:InstallationId	按用于更新连接的第三方 ID（如 Bitbucket 应用程序安装 ID）来筛选访问权限。允许您限制哪些第三方应用程序安装可用于建立连接
codestar-connections:OwnerId	按所有者或第三方提供程序的帐户 ID 来筛选访问权限
codestar-connections:PassedToService	按允许委托人向其传递连接的服务来筛选访问权限

条件键	描述
codestar-connections:ProviderAction	按 UseConnection 请求中的提供程序操作（如 ListRepositories ）来筛选访问权限。
codestar-connections:ProviderPermissionsRequired	按第三方提供程序权限的类型来筛选访问权限
codestar-connections:ProviderType	按请求中传递的第三方提供程序的类型来筛选访问权限
codestar-connections:ProviderTypeFilter	按用于筛选结果的第三方提供程序的类型来筛选访问权限
codestar-connections:RepositoryName	按第三方存储库的名称筛选访问权限

示例

要查看基于 AWS CodeStar 通知和 AWS CodeStar 连接身份的策略示例，请参阅。 基于身份的策略示例

AWS CodeStar 基于资源的通知和 AWS CodeStar 连接策略

AWS CodeStar 通知和 AWS CodeStar 连接不支持基于资源的策略。

基于标签的授权

您可以将标签附加到 AWS CodeStar 通知和 AWS CodeStar 连接资源或在请求中传递标签。要基于标签控制访问，您需要使用 codestar-notifications and codestar-connections:ResourceTag/*key-name*、aws:RequestTag/*key-name* 或 aws:TagKeys 条件键在策略的条件元素中提供标签信息。有关标记策略的更多信息，请参阅标记资源。 AWS有关为 AWS CodeStar 通知和 AWS CodeStar 连接资源添加标签的更多信息，请参阅为连接资源添加标签。

要查看基于身份的策略（用于根据资源上的标签来限制对该资源的访问）的示例，请参阅 使用标签控制对 C AWS CodeStar onnections 资源的访问权限。

IAM 角色

I AM 角色是您的 AWS 账户中具有特定权限的实体。

使用临时凭证

您可以使用临时凭证进行联合身份登录，代入 IAM 角色或跨账户角色。您可以通过调用AssumeRole或之类的 AWS STS API 操作来获取临时安全证书GetFederationToken。

AWS CodeStar 通知和 AWS CodeStar 连接支持使用临时证书。

服务相关角色

服务相关角色允许 AWS 服务访问其他服务中的资源以代表您完成操作。服务相关角色显示在 IAM 账户中，并归该服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

AWS CodeStar 通知支持服务相关角色。有关创建或管理 AWS CodeStar 通知和 AWS CodeStar 连接服务相关角色的详细信息，请参阅为 AWS CodeStar 通知使用服务相关角色。

AWS CodeStar 连接不支持服务相关角色。

AWS CodeConnections 权限参考

下表列出了每个 AWS CodeConnections API 操作、您可以为其授予权限的相应操作以及用于授予权限的资源 ARN 的格式。根据该 AWS CodeConnections API 允许的操作范围将这些 API 分为多个表。在编写可附加到 IAM 身份的权限策略（基于身份的策略）时，可参考此表。

在创建权限策略时，可以在策略的 `Action` 字段中指定操作。在策略的 `Resource` 字段中以 ARN 的形式指定资源值，可以使用或不使用通配符 (*)。

要在连接策略中表达条件，可以使用此处描述的和 条件键 中列出的条件键。您也可以使用 AWS-wide 条件键。有关 AWS范围密钥的完整列表，请参阅 IAM 用户指南中的可用密钥。

要指定操作，请在 API 操作名称之前使用 `codestar-connections:` 前缀 (例如，`codestar-connections:ListConnections` 或 `codestar-connections:CreateConnection`)。

使用通配符

要指定多个操作或资源，可以在 ARN 中使用通配符 (*)。例如，`codestar-connections:*`指定所有 AWS CodeConnections 动作并`codestar-connections:Get*`指定以单词开头的所有 AWS CodeConnections 动作`Get`。以下示例授予对以 `MyConnection` 名称开头的所有资源的访问权限。

```
arn:aws:codestar-connections:us-west-2:account-ID:connection/*
```

只能对下表中列出的*##*资源使用通配符，不能对 *region* 或 *account-id* 资源使用通配符。有关通配符的更多信息，请参阅 IAM 用户指南中的 IAM 标识符。

主题

- 用于管理连接的权限
- 用于管理主机的权限
- 用于完成连接的权限
- 设置主机的权限
- 将连接传递到服务
- 使用连接
- 对于 ProviderAction 支持的访问类型
- 标记连接资源支持的权限
- 将连接传递到存储库链接
- 存储库链接支持的条件键

用于管理连接的权限

被指定使用或 SDK 查看、创建 AWS CLI 或删除连接的角色或用户的权限应限制为以下内容。

> **Note**
>
> 您无法在仅具有以下权限的情况时，在控制台中完成或使用连接。您需要添加 用于完成连接的权限 中的权限。

```
codestar-connections:CreateConnection
codestar-connections:DeleteConnection
codestar-connections:GetConnection
codestar-connections:ListConnections
```

AWS CodeStar 通知和 AWS CodeStar 连接需要权限才能执行管理连接的操作

CreateConnection

操作： `codestar-connections:CreateConnection`

使用 CLI 或控制台创建连接所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

DeleteConnection

操作： codestar-connections:DeleteConnection

使用 CLI 或控制台删除连接所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

GetConnection

操作： codestar-connections:GetConnection

使用 CLI 或控制台查看连接详细信息所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

ListConnections

操作： codestar-connections:ListConnections

使用 CLI 或控制台列出账户中的所有连接所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

这些操作支持以下条件键：

操作	条件键
codestar-connections:CreateConnection	codestar-connections:ProviderType
codestar-connections:DeleteConnection	不适用

操作	条件键
codestar-connections:GetConnection	不适用
codestar-connections:ListConnections	codestar-connections:ProviderTypeFilter

用于管理主机的权限

被指定使用或 SDK 查看、创建 AWS CLI 或删除主机的角色或用户的权限应限制为以下内容。

> **Note**
>
> 您无法在仅具有以下权限的情况时，在主机中完成或使用连接。您需要添加 设置主机的权限 中的权限。

```
codestar-connections:CreateHost
codestar-connections:DeleteHost
codestar-connections:GetHost
codestar-connections:ListHosts
```

AWS CodeStar 管理主机的操作需要通知和 AWS CodeStar 连接权限

CreateHost

操作： codestar-connections:CreateHost

使用 CLI 或控制台创建主机所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:host/*host-id*

DeleteHost

操作： codestar-connections:DeleteHost

使用 CLI 或控制台删除主机所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:host/*host-id*

GetHost

操作： codestar-connections:GetHost

使用 CLI 或控制台查看主机详细信息所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:host/*host-id*

ListHosts

操作： codestar-connections:ListHosts

使用 CLI 或控制台列出账户中的所有主机所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:host/*host-id*

这些操作支持以下条件键：

操作	条件键
codestar-connections:CreateHost	codestar-connections:ProviderType
codestar-connections:DeleteHost	不适用
codestar-connections:GetHost	不适用
codestar-connections:ListHosts	codestar-connections:ProviderTypeFilter

用于完成连接的权限

指定用于在控制台中管理连接的角色或用户，应具有在控制台中完成连接和创建安装所需的权限，这包括授权与提供程序握手和为要使用的连接创建安装。除了上述权限之外，还可以使用以下权限。

控制台在执行基于浏览器的握手时使用以下 IAM 操作。ListInstallationTargets、GetInstallationUrl、StartOAuthHandshake、UpdateConnec和 GetIndividualAccessToken 是 IAM 策略权限。它们不是 API 操作。

```
codestar-connections:GetIndividualAccessToken
codestar-connections:GetInstallationUrl
codestar-connections:ListInstallationTargets
codestar-connections:StartOAuthHandshake
codestar-connections:UpdateConnectionInstallation
```

基于这一点，在控制台中使用、创建、更新或删除连接需要以下权限。

```
codestar-connections:CreateConnection
codestar-connections:DeleteConnection
codestar-connections:GetConnection
codestar-connections:ListConnections
codestar-connections:UseConnection
codestar-connections:ListInstallationTargets
codestar-connections:GetInstallationUrl
codestar-connections:StartOAuthHandshake
codestar-connections:UpdateConnectionInstallation
codestar-connections:GetIndividualAccessToken
```

AWS CodeConnections 完成连接所需的操作权限

GetIndividualAccessToken

操作： `codestar-connections:GetIndividualAccessToken`

使用控制台完成连接所必需。这只是一种 IAM 策略权限，不是 API 操作。

资源： `arn:aws:codestar-connections:region:account-id:connection/connection-id`

GetInstallationUrl

操作： `codestar-connections:GetInstallationUrl`

使用控制台完成连接所必需。这只是一种 IAM 策略权限，不是 API 操作。

资源： `arn:aws:codestar-connections:region:account-id:connection/connection-id`

ListInstallationTargets

操作： `codestar-connections:ListInstallationTargets`

使用控制台完成连接所必需。这只是一种 IAM 策略权限，不是 API 操作。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

StarTo AuthHandshake

操作： codestar-connections:StartOAuthHandshake

使用控制台完成连接所必需。这只是一种 IAM 策略权限，不是 API 操作。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

UpdateConnectionInstallation

操作： codestar-connections:UpdateConnectionInstallation

使用控制台完成连接所必需。这只是一种 IAM 策略权限，不是 API 操作。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

这些操作支持以下条件键。

操作	条件键
codestar-connections:GetIndividualAccessToken	codestar-connections:ProviderType
codestar-connections:GetInstallationUrl	codestar-connections:ProviderType
codestar-connections:ListInstallationTargets	不适用
codestar-connections:StartOAuthHandshake	codestar-connections:ProviderType
codestar-connections:UpdateConnectionInstallation	codestar-connections:InstallationId

设置主机的权限

指定在控制台中管理连接的角色或用户应具有在控制台中设置主机所需的权限，包括向提供程序授权握手和安装主机应用程序。除了上述主机权限之外，还可以使用以下权限。

控制台在执行基于浏览器的主机注册时使用以下 IAM 操作。RegisterAppCode 和 StartAppRegistrationHandshake 是 IAM 策略权限。它们不是 API 操作。

```
codestar-connections:RegisterAppCode
codestar-connections:StartAppRegistrationHandshake
```

基于这一点，在控制台中使用、创建、更新或删除需要主机的连接需要以下权限（例如已安装的提供程序类型）。

```
codestar-connections:CreateConnection
codestar-connections:DeleteConnection
codestar-connections:GetConnection
codestar-connections:ListConnections
codestar-connections:UseConnection
codestar-connections:ListInstallationTargets
codestar-connections:GetInstallationUrl
codestar-connections:StartOAuthHandshake
codestar-connections:UpdateConnectionInstallation
codestar-connections:GetIndividualAccessToken
codestar-connections:RegisterAppCode
codestar-connections:StartAppRegistrationHandshake
```

AWS CodeConnections 完成主机设置所需的操作权限

RegisterAppCode

操作： `codestar-connections:RegisterAppCode`

使用控制台来完成主机设置所必需。这只是一种 IAM 策略权限，不是 API 操作。

资源： `arn:aws:codestar-connections:`*`region`*`:`*`account-id`*`:host/`*`host-id`*

StartAppRegistrationHandshake

操作： `codestar-connections:StartAppRegistrationHandshake`

使用控制台来完成主机设置所必需。这只是一种 IAM 策略权限，不是 API 操作。

资源： arn:aws:codestar-connections:*region*:*account-id*:host/*host-id*

这些操作支持以下条件键。

将连接传递到服务

将连接传递到服务时（例如，在管道定义中提供连接 ARN 以创建或更新管道时），用户必须具有 codestar-connections:PassConnection 权限。

AWS CodeConnections 传递连接所需的权限

PassConnection

操作： codestar-connections:PassConnection

将连接传递到服务所需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

此操作还支持以下条件键：

- codestar-connections:PassedToService

条件键的受支持值

键	有效操作提供方
codestar-connections:PassedToService	• codeguru-reviewer • codepipeline.amazonaws.com • proton.amazonaws.com

使用连接

当类似的服务 CodePipeline 使用连接时，该服务角色必须拥有给定连接的codestar-connections:UseConnection权限。

要在控制台中管理连接，用户策略必须具有 codestar-connections:UseConnection 权限。

AWS CodeConnections 使用连接所需的操作

UseConnection

操作： codestar-connections:UseConnection

使用连接所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

此操作还支持以下条件键：

- codestar-connections:BranchName
- codestar-connections:FullRepositoryId
- codestar-connections:OwnerId
- codestar-connections:ProviderAction
- codestar-connections:ProviderPermissionsRequired
- codestar-connections:RepositoryName

条件键的受支持值

键	有效操作提供方
codestar-connections:FullRepositoryId	存储库的用户名和存储库名称，例如 my-owner/my-repository 。仅当连接用于访问特定存储库时才支持。
codestar-connections:ProviderPermissionsRequired	read_only 或 read_write
codestar-connections:ProviderAction	GetBranch ，ListRepositories ，ListOwners ，ListBranches ，StartUploadArchiveToS3 ，GitPush, GitPull, GetUploadArchiveToS3Status ，CreatePullRequestDiffComment ，GetPullRequest ，ListBranchCommits ，ListCommitFiles ，

键	有效操作提供方
	ListPullRequestComments ，ListPullRequestCommits 。 有关信息，请参阅下一部分。

某个功能所需的条件键可能会随着时间的推移而更改。建议使用 codestar-connections:UseConnection 来控制对连接的访问，除非您的访问控制需求不同的权限。

对于 **ProviderAction** 支持的访问类型

当 AWS 服务使用连接时，会导致向您的源代码提供商发出 API 调用。例如，服务可能会通过调用 https://api.bitbucket.org/2.0/repositories/*username* API 来列出 Bitbucket 连接的存储库。

ProviderAction 条件键允许您限制可以调用提供程序上的哪些 API。由于 API 路径可能是动态生成的，并且路径因提供方而异，因此 ProviderAction 值映射到抽象操作名称而不是 API 的 URL。这允许您编写具有相同效果的策略，无论连接的提供程序类型如何。

以下是授予每个受支持的 ProviderAction 值的访问类型。以下是 IAM 策略权限。它们不是 API 操作。

AWS CodeConnections 支持的访问类型 **ProviderAction**

GetBranch

操作： codestar-connections:GetBranch

访问有关分支的信息所必需，例如该分支的最新提交。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

ListRepositories

操作： codestar-connections:ListRepositories

访问属于所有者的公有存储库和私有存储库列表（包括有关这些存储库的详细信息）时所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

ListOwners

操作： codestar-connections:ListOwners

访问连接有权访问的拥有者列表所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

ListBranches

操作： codestar-connections:ListBranches

访问给定存储库上存在的分支列表所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

StartUploadArchiveToS3

操作： codestar-connections:StartUploadArchiveToS3

读取源代码并将其上传到 Amazon S3 所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

GitPush

操作： codestar-connections:GitPush

使用 Git 写入存储库所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

GitPull

操作： codestar-connections:GitPull

使用 Git 从存储库读取所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

GetUploadArchiveToS3Status

操作： codestar-connections:GetUploadArchiveToS3Status

访问 StartUploadArchiveToS3 启动的上传的状态所必需，包括任何错误消息。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

CreatePullRequestDiffComment

操作： codestar-connections:CreatePullRequestDiffComment

访问拉取请求的注释所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

GetPullRequest

操作： codestar-connections:GetPullRequest

查看存储库的拉取请求所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

ListBranchCommits

操作： codestar-connections:ListBranchCommits

查看存储库分支的提交的列表所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

ListCommitFiles

操作： codestar-connections:ListCommitFiles

查看要提交文件的列表所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

ListPullRequestComments

操作： codestar-connections:ListPullRequestComments

查看拉取请求的注释的列表所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

ListPullRequestCommits

操作： codestar-connections:ListPullRequestCommits

查看拉取请求的提交的列表所必需。

资源： arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

标记连接资源支持的权限

标记连接资源时使用以下 IAM 操作。

```
codestar-connections:ListTagsForResource
codestar-connections:TagResource
codestar-connections:UntagResource
```

AWS CodeConnections 标记连接资源所需的操作

ListTagsForResource

操作： codestar-connections:ListTagsForResource

查看与连接资源关联的标签列表所必需。

资源：arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*， arn:aws:codestar-connections:*region*:*account-id*:host/*host-id*

TagResource

操作： codestar-connections:TagResource

标记连接资源所必需。

资源：arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*， arn:aws:codestar-connections:*region*:*account-id*:host/*host-id*

UntagResource

操作： codestar-connections:UntagResource

从连接资源移除标签时所必需。

资源：arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*， arn:aws:codestar-connections:*region*:*account-id*:host/*host-id*

将连接传递到存储库链接

在同步配置中提供存储库链接时，用户必须拥有存储库链接 ARN/resource 的 codestar-connections:PassRepository 权限。

AWS CodeConnections 传递连接所需的权限

PassRepository

操作： codestar-connections:PassRepository

需要将存储库链接传递到同步配置。

资源： arn:aws:codestar-connections:*region*:*account-id*:repository-link/*repository-link-id*

此操作还支持以下条件键：

- codestar-connections:PassedToService

条件键的受支持值

键	有效操作提供方
codestar-connections:PassedToService	• cloudformation.sync.codeconnections.amazonaws.com

存储库链接支持的条件键

以下条件键支持对存储库链接和同步配置资源的操作：

- codestar-connections:Branch

  按请求中传递的分支名称来筛选访问权限。

条件键支持的操作

键	有效值
codestar-connections:Branch	此条件键支持以下操作： • CreateSyncConfiguration • UpdateSyncConfiguration • GetRepositorySyncStatus

基于身份的策略示例

默认情况下，拥有、 AWS CodeCommit AWS CodeBuild AWS CodeDeploy、或 AWS CodePipeline 应用了其中一项托管策略的 IAM 用户和角色有权访问与这些策略意图一致的连接、通知和通知规则。例如，应用了其中一项完全访问策略（AWSCodeCommitFullAccess、AWSCodeBuildAdminAccessAWSCodeDeployFullAccess、或 AWSCodePipeline_FullAccess）的 IAM 用户或角色也可以完全访问为这些服务的资源创建的通知和通知规则。

其他 IAM 用户和角色无权创建或修改 AWS CodeStar 通知和 AWS CodeStar 连接资源。他们也无法使用 AWS Management Console AWS CLI、或 AWS API 执行任务。IAM 管理员必须创建适当 IAM 策略，为用户和角色授予权限，以便对他们所需的指定资源执行 API 操作。然后，管理员必须将这些策略附加到需要这些权限的 IAM 用户或组。

AWS CodeStar 通知的权限和示例

以下政策声明和示例可以帮助您管理 AWS CodeStar 通知。

完全访问托管策略中的通知的相关权限

AWSCodeCommitFullAccess、AWSCodeBuildAdminAccessAWSCodeDeployFullAccess、和AWSCodePipeline_FullAccess托管策略包括以下语句，允许在开发者工具控制台中完全访问通知。已应用其中一项托管策略的用户还可以创建和管理通知的 Amazon SNS 主题、为用户订阅和取消订阅主题以及列出要选择作为通知规则目标的主题。

> **Note**
>
> 在托管策略中，条件键 `codestar-notifications:NotificationsForResource` 将具有特定于服务的资源类型的值。例如，在的完全访问策略中 CodeCommit，值为`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": "*"
    }
```

只读托管策略中的通知的相关权限

AWSCodeCommitReadOnlyAccess、AWSCodeBuildReadOnlyAccessAWSCodeDeployReadOnlyAccess、和AWSCodePipeline_ReadOnlyAccess托管策略包括以下语句，允许对通知进行只读访问。例如，它们可以在 开发工具控制台中查看资源的通知，但无法创建、管理或订阅这些通知。

> **Note**
>
> 在托管策略中，条件键 `codestar-notifications:NotificationsForResource` 将具有特定于服务的资源类型的值。例如，在的完全访问策略中 CodeCommit，值为`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": "*"
    }
```

其他托管策略中的通知的相关权限

AWSCodeCommitPowerUserAWSCodeBuildDeveloperAccess、
和AWSCodeBuildDeveloperAccess托管策略包括以下声明，允许应用其中一个托管策略的开发者创建、编辑和订阅通知。他们无法删除通知规则或管理资源的标签。

> **Note**
>
> 在托管策略中，条件键 `codestar-notifications:NotificationsForResource` 将具有特定于服务的资源类型的值。例如，在的完全访问策略中 CodeCommit，值为`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": "*"
    }
```

示例：用于管理通知的管理员级别策略 AWS CodeStar

在此示例中，您想向 AWS 账户中的 IAM 用户授予对 AWS CodeStar 通知的完全访问权限，以便该用户可以查看通知规则的详细信息并列出通知规则、目标和事件类型。您还想要允许该用户添加、更新和删除通知规则。这是一个完全访问策略，等同于、AWSCodeBuildAdminAccessAWSCodeCommitFullAccessAWSCodeDeployFullAccess、和AWSCodePipeline_FullAccess托管策略中包含的通知权限。与这些托管策略一样，您只应将此类政策声明附加到需要对整个 AWS 账户中的通知和通知规则具有完全管理权限的 IAM 用户、群组或角色。

> **Note**
>
> 此策略包含允许 `CreateNotificationRule`。将此策略应用于其 IAM 用户或角色的任何用户都可以为 AWS 账户中通知支持的任何和所有资源类型创建 AWS CodeStar 通知规则，即使该用户自己无权访问这些资源也是如此。例如，拥有此策略的用户可以在没有访问权限的情况下为 CodeCommit 仓库创建通知 CodeCommit规则。

```
{
    "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": "*"
      }
   ]
}
```

示例：用于使用通知的贡献者级别策略 AWS CodeStar

在此示例中，您希望授予 day-to-day 使用 AWS CodeStar 通知的权限，例如创建和订阅通知，但不允许访问更具破坏性的操作，例如删除通知规则或目标。这等同于AWSCodeBuildDeveloperAccessAWSCodeDeployDeveloperAccess、和AWSCodeCommitPowerUser托管策略中提供的访问权限。

> **Note**
>
> 此策略包含允许 `CreateNotificationRule`。将此策略应用于其 IAM 用户或角色的任何用户都可以为 AWS 账户中通知支持的任何和所有资源类型创建 AWS CodeStar 通知规则，即使该用户自己无权访问这些资源也是如此。例如，拥有此策略的用户可以在没有访问权限的情况下为 CodeCommit 仓库创建通知 CodeCommit规则。

```
{
    "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": "*"
        }
    ]
}
```

示例：使用 AWS CodeStar 通知的 read-only-level 策略

在此示例中，您要向您账户中的 IAM 用户授予对 AWS 账户中的通知规则、目标和事件类型的只读访问权限。该示例说明了如何创建策略以允许查看这些项。这等同于AWSCodeBuildReadOnlyAccessAWSCodeCommitReadOnly、和AWSCodePipeline_ReadOnlyAccess托管策略中包含的权限。

```
{
    "Version": "2012-10-17",
    "Id": "CodeNotification__ReadOnly",
    "Statement": [
        {
            "Sid": "Reads_API_Access",
            "Effect": "Allow",
```

```
            "Action": [
                "CodeNotification:DescribeNotificationRule",
                "CodeNotification:ListNotificationRules",
                "CodeNotification:ListTargets",
                "CodeNotification:ListEventTypes"
            ],
            "Resource": "*"
        }
    ]
}
```

的权限和示例 AWS CodeConnections

以下策略语句和示例可帮助您管理 AWS CodeConnections。

要了解如何使用这些示例 JSON 策略文档创建 IAM 基于身份的策略，请参阅 IAM 用户指南 中的在 JSON 选项卡上创建策略。

示例：使用 CLI 创建和 AWS CodeConnections 使用控制台查看的策略

被指定使用 AWS CLI 或 SDK 查看、创建、标记或删除连接的角色或用户的权限应限制为以下内容。

> **Note**
>
> 您无法在仅具有以下权限的情况时，在控制台中完成连接。您需要在下一部分中添加权限。

要使用控制台查看可用连接的列表、查看标签以及使用连接，请使用以下策略。

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "ConnectionsFullAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-connections:CreateConnection",
            "codestar-connections:DeleteConnection",
            "codestar-connections:UseConnection",
            "codestar-connections:GetConnection",
            "codestar-connections:ListConnections",
```

```
            "codestar-connections:TagResource",
            "codestar-connections:ListTagsForResource",
            "codestar-connections:UntagResource"
        ],
        "Resource": "*"
     }
   ]
}
```

示例：使用控制台创建 AWS CodeConnections 的策略

指定用于在控制台中管理连接的角色或用户，应具有在控制台中完成连接和创建安装所需的权限，这包括授权与提供程序握手和为要使用的连接创建安装。在控制台中使用连接还需要添加 UseConnection。使用以下策略在控制台中查看、使用、创建、标记或删除连接。

```
{
    "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": [
                "*"
            ]
        }
    ]
}
```

示例：管理员级别的管理策略 AWS CodeConnections

在此示例中，您想向 AWS 账户中的 IAM 用户授予完全访问权限， CodeConnections 以便该用户可以添加、更新和删除连接。这是完全访问策略，等同于AWSCodePipeline_FullAccess托管策略。与该托管策略一样，您只应将此类策略声明附加到需要对 AWS 账户连接具有完全管理权限的 IAM 用户、群组或角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "ConnectionsFullAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-connections:CreateConnection",
            "codestar-connections:DeleteConnection",
            "codestar-connections:UseConnection",
            "codestar-connections:GetConnection",
            "codestar-connections:ListConnections",
            "codestar-connections:ListInstallationTargets",
            "codestar-connections:GetInstallationUrl",
            "codestar-connections:StartOAuthHandshake",
            "codestar-connections:UpdateConnectionInstallation",
            "codestar-connections:GetIndividualAccessToken",
            "codestar-connections:TagResource",
            "codestar-connections:ListTagsForResource",
            "codestar-connections:UntagResource"
        ],
        "Resource": "*"
      }
   ]
}
```

示例：用于使用的贡献者级别策略 AWS CodeConnections

在此示例中，您希望授予 day-to-day 使用权限 CodeConnections，例如创建和查看连接的详细信息，但不允许授予更具破坏性的操作（例如删除连接）的访问权限。

```
{
    "Version": "2012-10-17",
    "Sid": "AWSCodeStarConnectionsPowerUserAccess",
        "Effect": "Allow",
```

```
        "Action": [
            "codestar-connections:CreateConnection",
            "codestar-connections:UseConnection",
            "codestar-connections:GetConnection",
            "codestar-connections:ListConnections",
            "codestar-connections:ListInstallationTargets",
            "codestar-connections:GetInstallationUrl",
            "codestar-connections:GetIndividualAccessToken",
            "codestar-connections:StartOAuthHandshake",
            "codestar-connections:UpdateConnectionInstallation",
            "codestar-connections:ListTagsForResource"
        ],
        "Resource": "*"
        }
    ]
}
```

示例：使用 read-only-level 策略 AWS CodeConnections

在此示例中，您想向账户中的 IAM 用户授予对您账户中连接的只读访问权限。 AWS 该示例说明了如何创建策略以允许查看这些项。

```
{
    "Version": "2012-10-17",
    "Id": "Connections__ReadOnly",
    "Statement": [
        {
            "Sid": "Reads_API_Access",
            "Effect": "Allow",
            "Action": [
            "codestar-connections:GetConnection",
            "codestar-connections:ListConnections",
            "codestar-connections:ListInstallationTargets",
            "codestar-connections:GetInstallationUrl",
            "codestar-connections:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

示例：用于 AWS CodeConnections 指定存储库的范围缩小策略

在以下示例中，客户希望 CodeBuild 服务角色访问指定的 Bitbucket 存储库。关于 CodeBuild 服务角色的政策：

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "codestar-connections:UseConnection"
        ],
        "Resource": "arn:aws:codestar-connections:us-
west-2:connection:3dee99b9-172f-4ebe-a257-722365a39557",
        "Condition": {"ForAllValues:StringEquals": {"codestar-
connections:FullRepositoryId": "myrepoowner/myreponame"}}
    }
}
```

示例：使用与之连接的策略 CodePipeline

在以下示例中，管理员希望用户使用与的连接 CodePipeline。附加至用户的策略：

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "codestar-connections:PassConnection"
        ],
        "Resource": "arn:aws:codestar-connections:us-west-2:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
        "Condition": {"ForAllValues:StringEquals": {"codestar-
connections:PassedToService": "codepipeline.amazonaws.com"}}
    }
}
```

示例：使用 CodeBuild 服务角色进行 Bitbucket 读取操作 AWS CodeConnections

在以下示例中，无论存储库如何，客户都希望 CodeBuild 服务角色对 Bitbucket 执行读取操作。关于 CodeBuild 服务角色的政策：

```
{
```

```
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "codestar-connections:UseConnection"
        ],
        "Resource": "arn:aws:codestar-connections:us-west-2:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
        "Condition": {"ForAllValues:StringEquals": {"codestar-
connections:ProviderPermissionsRequired": "read_only"}}
    }
}
```

示例：限制 CodeBuild 服务角色使用执行操作 AWS CodeConnections

在以下示例中，客户希望阻止 CodeBuild 服务角色执行类似的操作`CreateRepository`。关于 CodeBuild 服务角色的政策：

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "codestar-connections:UseConnection"
        ],
        "Resource": "arn:aws:codestar-connections:us-west-2:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
        "Condition": {"ForAllValues:StringNotEquals": {"codestar-
connections:ProviderPermissionsRequired": "CreateRepository"}}
    }
}
```

使用标签控制对 C AWS CodeStar onnections 资源的访问权限

标签可以附加到资源，也可以从请求传入支持标签的服务。在中 CodeConnections，资源可以有标签，有些操作可以包含标签。在创建 IAM 策略时，您可以使用标签条件键来控制以下：

- 基于资源已有的标签，哪些用户可以对管道资源执行操作。
- 哪些标签可以在操作的请求中传递。
- 是否特定标签键可在请求中使用。

以下示例演示了如何为 CodeConnections 用户指定策略中的标签条件。

Example 1：基于请求中的标签允许操作

以下策略授予用户在 CodeConnections中创建连接的权限。

为此，如果请求指定一个名为 Project 的带有值为 ProjectA 的标签，则它允许 CreateConnection 和 TagResource 操作。（aws:RequestTag 条件键用于控制可以通过 IAM 请求传递哪些标签。）aws:TagKeys 条件确保标签键区分大小写。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codestar-connections:CreateConnection",
        "codestar-connections:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Project": "ProjectA"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["Project"]
        }
      }
    }
  ]
}
```

Example 2：基于资源标签允许操作

以下策略授予用户对 CodeConnections中的资源执行操作以及获取其相关信息的权限。

为此，如果该管道具有名为 Project、值为 ProjectA 的标签，则它允许特定操作。（aws:RequestTag 条件键用于控制可以通过 IAM 请求传递哪些标签。）aws:TagKeys 条件确保标签键区分大小写。

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

```
    {
      "Effect": "Allow",
      "Action": [
        "codestar-connections:CreateConnection",
        "codestar-connections:DeleteConnection",
        "codestar-connections:ListConnections"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "ProjectA"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["Project"]
        }
      }
    }
  ]
}
```

使用控制台中的通知和连接

通知体验内置于 CodeBuild、 CodeCommit CodeDeploy、和 CodePipeline控制台中，以及开发者工具控制台的 “设置” 导航栏本身。要访问控制台中的通知，您必须为这些服务应用其中一项托管策略，或者必须具有一组最低权限。这些权限必须允许您列出和查看有关您 AWS 账户中 AWS CodeStar 通知和 AWS CodeStar 连接资源的详细信息。如果您创建的基于身份的策略比所需的最低权限更严格，则无法为具有该策略的实体（IAM 用户或角色）正常运行控制台。有关授予 AWS CodeBuild、 AWS CodeCommit AWS CodeDeploy AWS CodePipeline、和访问权限（包括对这些控制台的访问权限）的更多信息，请参阅以下主题：

- CodeBuild: 将基于身份的策略用于 CodeBuild
- CodeCommit: 将基于身份的策略用于 CodeCommit
- AWS CodeDeploy: 身份和访问管理 AWS CodeDeploy
- CodePipeline: 使用 IAM 策略进行访问控制

AWS CodeStar 通知没有任何 AWS 托管策略。为了提供对通知功能的访问权限，您必须为前面列出的其中一项服务应用某种托管策略，或者您必须创建具有要授予用户或实体的权限级别的策略，然后将这些策略附加到需要这些权限的用户、组或角色。有关更多信息和示例，请参阅以下：

- 示例：用于管理通知的管理员级别策略 AWS CodeStar

- 示例：用于使用通知的贡献者级别策略 AWS CodeStar
- 示例：使用 AWS CodeStar 通知的 read-only-level 策略.

AWS CodeStar 连接没有任何 AWS 托管策略。您可以使用权限和访问权限组合，例如，用于完成连接的权限中详细叙述的权限。

有关更多信息，请参阅下列内容：

- 示例：管理员级别的管理策略 AWS CodeConnections
- 示例：用于使用的贡献者级别策略 AWS CodeConnections
- 示例：使用 read-only-level 策略 AWS CodeConnections

您无需为仅调用 AWS CLI 或 AWS API 的用户授予控制台权限。相反，只允许访问与您尝试执行的 API 操作相匹配的操作。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
    "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": "*"
        }
    ]
}
```

排除 AWS CodeStar 通知和 AWS CodeStar 连接身份和访问权限故障

使用以下信息可帮助您诊断和修复在使用通知和 IAM 时可能遇到的常见问题。

主题

- 我是管理员并希望允许其他人访问通知
- 我创建了一个 Amazon SNS 主题并将其添加为通知规则目标，但是我没有收到有关事件的电子邮件
- 我想允许 AWS 账户以外的用户访问我的 AWS CodeStar 通知和 AWS CodeStar 连接资源

我是管理员并希望允许其他人访问通知

要允许其他人访问 AWS CodeStar 通知和 AWS CodeStar 连接，您必须为需要访问的人员或应用程序创建 IAM 实体（用户或角色）。它们将使用该实体的凭证访问 AWS。然后，您必须将策略附加到授予他们在 AWS CodeStar 通知和 AWS CodeStar 连接中的正确权限的实体。

要立即开始使用，请参阅《IAM 用户指南》中的创建您的第一个 IAM 委派用户和组。

有关 AWS CodeStar 通知的特定信息，请参阅AWS CodeStar 通知的权限和示例。

我创建了一个 Amazon SNS 主题并将其添加为通知规则目标，但是我没有收到有关事件的电子邮件

为了接收有关事件的通知，您必须订阅一个有效的 Amazon SNS 主题作为通知规则的目标，并且您的电子邮件地址也必须订阅 Amazon SNS 主题。要排查 Amazon SNS 主题的问题，请检查以下内容：

- 确保 Amazon SNS 主题与通知规则位于同一 AWS 区域。

- 检查以确保您的电子邮件别名已订阅到正确的主题，并且您已确认订阅。有关更多信息，请参阅将终端节点订阅到 Amazon SNS 主题。
- 确认已修改主题策略以允许 AWS CodeStar 通知向该主题推送通知。该主题策略应包含类似于以下内容的声明：

```
{
    "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"
        }
    }
}
```

有关更多信息，请参阅 设置。

我想允许 AWS 账户以外的用户访问我的 AWS CodeStar 通知和 AWS CodeStar 连接资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表（ACL）的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解 AWS CodeStar 通知和 AWS CodeStar 连接是否支持这些功能，请参阅开发工具控制台中的特征如何与 IAM 配合使用。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户 ，请参阅 IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问权限。
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限。 AWS 账户

- 要了解如何通过联合身份验证提供访问权限，请参阅 IAM 用户指南中的为经过外部身份验证的用户（联合身份验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同。

为 AWS CodeStar 通知使用服务相关角色

AWS CodeStar 通知使用 AWS Identity and Access Management (IAM) 服务相关资源。服务相关角色是一种独特类型的 IAM 角色，它与 AWS CodeStar 连接直接相关。服务相关角色由 AWS CodeStar 通知预定义，并包含该服务代表您调用其他 AWS 服务所需的一切权限。您首次创建通知规则时为您创建此角色。您不必创建角色。

服务相关角色可让您更轻松地设置 AWS CodeStar 通知，因为您不必手动添加权限。AWSCodeStar 通知定义其服务相关角色的权限，除非另外定义，否则只有 AWS CodeStar 通知可以代入该角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其它 IAM 实体的权限策略。

要删除服务相关角色，您必须先删除其相关资源。这将保护您的 AWS CodeStar 通知资源，因为您不会无意中删除对资源的访问权限。

有关支持服务相关角色的其他服务的信息，请参阅与 IAM 配合使用的AWS服务。

AWS CodeStar 通知的服务相关角色权限

AWS CodeStar 通知使用 AWSServiceRoleForCodeStarNotifications 服务相关角色检索有关工具链中发生的事件的信息，并将通知发送到您指定的目标。

AWSServiceRoleForCodeStarNotifications 服务相关角色信任以下服务以担任该角色：

- `codestar-notifications.amazonaws.com`

角色权限策略允许 AWS CodeStar 通知对指定资源完成以下操作：

- 操作：`CloudWatch Event rules that are named awscodestar-notifications-*` 上的 `PutRule`
- 操作：`DescribeRule` 上的 `CloudWatch Event rules that are named awscodestar-notifications-*`
- 操作：`PutTargets` 上的 `CloudWatch Event rules that are named awscodestar-notifications-*`

- 操作：CreateTopic 到 create Amazon SNS topics for use with AWS CodeStar Notifications with the prefix CodeStarNotifications-
- 操作：all comments on all pull requests in all CodeCommit repositories in the AWS account 上的 GetCommentsForPullRequests
- 操作：GetCommentsForComparedCommit 上的 all comments on all commits in all CodeCommit repositories in the AWS account
- 操作：GetDifferences 上的 all commits in all CodeCommit repositories in the AWS account
- 操作：GetCommentsForComparedCommit 上的 all comments on all commits in all CodeCommit repositories in the AWS account
- 操作：GetDifferences 上的 all commits in all CodeCommit repositories in the AWS account
- 操作：DescribeSlackChannelConfigurations 上的 all AWS Chatbot clients in the AWS account
- 操作：UpdateSlackChannelConfiguration 上的 all AWS Chatbot clients in the AWS account
- 操作：ListActionExecutions 上的 all actions in all pipelines in the AWS account
- 操作：GetFile 上的 all files in all CodeCommit repositories in the AWS account unless otherwise tagged

您可以在 AWSServiceRoleForCodeStarNotifications 服务相关角色的策略语句中看到以下操作。

```
{
    "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"
        }
    ]
}
```

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅 IAM 用户指南中的服务相关角色权限。

为 AWS CodeStar 通知创建服务相关角色

无需手动创建服务相关角色。您可以使用开发工具控制台或开发工具包中的 CreateNotificationRule API 创建通知规则。您也可以直接调用 API。无论您使用哪种方法，都会为您创建服务相关角色。

如果删除此服务相关角色，然后需要再次创建，可以使用相同流程在账户中重新创建此角色。您可以使用开发工具控制台或开发工具包中的 CreateNotificationRule API 创建通知规则。您也可以直接调用 API。无论您使用哪种方法，都会为您创建服务相关角色。

为 AWS CodeStar 通知编辑服务相关角色

创建服务相关角色后，您将无法更改其名称，因为可能有多种实体引用该角色。但是，您可以使用 IAM 编辑角色描述。有关更多信息，请参见 IAM 用户指南中的编辑服务相关角色。

为 AWS CodeStar 通知删除服务相关角色

如果您不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样您就没有未被主动监控或维护的未使用实体。您必须先清除服务相关角色的资源，然后才能将其删除。对于 AWS CodeStar 通知，这意味着删除所有使用您 AWS 账户中的服务角色的通知规则。

> **Note**
>
> 如果在您试图删除资源时 AWS CodeStar 通知服务正在使用该角色，则删除操作可能会失败。如果发生这种情况，请等待几分钟后重试。

要删除 AWSServiceRoleForCodeStarNotifications 使用的 AWS CodeStar 通知资源

1. 打开 https://console.aws.amazon.com/codesuite/settings/notifications 上的 AWS 开发工具控制台。

   > **Note**
   >
   > 通知规则将应用于创建它们的 AWS 区域。如果您在多个 AWS 区域中有通知规则，请使用区域选择器更改 AWS 区域。

2. 选择列表中出现的所有通知规则，然后选择删除。
3. 在创建通知规则的所有 AWS 区域中重复这些步骤。

要使用 IAM 删除服务相关角色

使用 IAM 控制台、AWS CLI 或 AWS Identity and Access Management API 删除 AWSServiceRoleForCodeStarNotifications 服务相关角色。有关更多信息，请参阅 IAM 用户指南的删除服务相关角色。

AWS CodeStar 通知服务相关角色的受支持区域

AWS CodeStar 通知支持在服务可用的所有 AWS 区域中使用服务相关角色。有关更多信息，请参阅 AWS 区域和端点和 AWS CodeStar 通知。

将服务相关角色用于 AWS CodeConnections

AWS CodeConnections 使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与 AWS CodeConnections 直接相关。服务相关角色由 AWS CodeConnections 预定义，并包含该服务代表您调用其他 AWS 服务所需的一切权限。首次创建连接时会为您创建此角色。您不必创建角色。

服务相关角色可让您更轻松地设置 AWS CodeConnections，因为您无需手动添加权限。AWS CodeConnections 定义其服务相关角色的权限，除非另行定义，否则仅 AWS CodeConnections 可以代入其角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其它 IAM 实体的权限策略。

要删除服务相关角色，您必须先删除其相关资源。这将保护您的 AWS CodeConnections 资源，因为您不会无意中删除对资源的访问权限。

有关支持服务相关角色的其他服务的信息，请参阅与 IAM 配合使用的AWS服务。

AWS CodeConnections 的服务相关角色权限

AWS CodeConnections 使用 AWSServiceRoleForGitSync 服务相关角色，来通过 Git 同步功能与连接的基于 Git 的存储库进行同步。

AWSServiceRoleForGitSync 服务相关角色信任以下服务代入该角色：

- `repository.sync.codeconnections.amazonaws.com`

名为 AWSGitSyncServiceRolePolicy 的角色权限策略允许 AWS CodeConnections 对指定资源完成以下操作：

- 操作：授予权限，允许用户创建与基于 Git 的外部存储库的连接，并使用 Git 同步功能与这些存储库同步。

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色权限。

为 AWS CodeConnections 创建服务相关角色

无需手动创建服务相关角色。使用 CreateRepositoryLink API 为 Git 同步项目创建资源时，即会创建该角色。

如果删除此服务相关角色，然后需要再次创建，可以使用相同流程在账户中重新创建此角色。

为 AWS CodeConnections 编辑服务相关角色

创建服务相关角色后，您将无法更改其名称，因为可能有多种实体引用该角色。但是，您可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除 AWS CodeConnections 的服务相关角色

如果您不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样您就没有未被主动监控或维护的未使用实体。您必须先清除服务相关角色的资源，然后才能将其删除。这意味着删除所有使用您 AWS 账户中的服务角色的连接。

> **Note**
>
> 如果在您试图删除资源时 AWS CodeConnections 服务正在使用该角色，则删除操作可能会失败。如果发生这种情况，请等待几分钟后重试。

删除 AWSServiceRoleForGitSync 使用的 AWS CodeConnections 资源

1. 打开开发人员工具控制台，然后选择设置。
2. 选择列表中出现的所有连接，然后选择删除。
3. 在创建连接的所有 AWS 区域中重复这些步骤。

要使用 IAM 删除服务相关角色

使用 IAM 控制台、AWS CLI 或 AWS Identity and Access Management API 删除 AWSServiceRoleForGitSync 服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的删除服务相关角色。

AWS CodeConnections 服务相关角色的受支持区域

AWS CodeConnections 支持在服务可用的所有 AWS 区域中使用服务相关角色。有关更多信息，请参阅 AWS 区域和端点。

适用于 AWS CodeConnections 的 AWS 托管策略

AWS 托管策略是由 AWS 创建和管理的独立策略。AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住，AWS 托管策略可能不会为您的特定使用案例授予最低权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于您的使用场景的客户托管型策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新在 AWS 托管策略中定义的权限，则更新会影响该策略所附加到的所有主体身份（用户、组和角色）。当新的 AWS 服务 启动或新的 API 操作可用于现有服务时，AWS 最有可能更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管策略。

AWS 托管策略：AWSGitSyncServiceRolePolicy

您不能将 AWSGitSyncServiceRolePolicy 附加到 IAM 实体。此附加到服务相关角色的策略允许 AWS CodeConnections 代表您执行操作。有关更多信息，请参阅将服务相关角色用于 AWS CodeConnections。

此策略允许客户访问基于 Git 的存储库以使用连接。客户将在使用 CreateRepositoryLink API 后访问这些资源。

权限详细信息

此策略包含以下权限。

- `codestar-connections` – 授予权限以允许用户创建与基于 Git 的外部存储库的连接。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AccessGitRepos",
   "Effect": "Allow",
   "Action": [
    "codestar-connections:UseConnection"
   ],
```

```
    "Resource": "arn:aws:codestar-connections:*:*:connection/*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     }
    }
   }
  ]
}
```

对 AWS 托管策略的 AWS CodeConnections 更新

查看有关 AWS CodeConnections 的 AWS 托管策略更新的详细信息（从该服务开始跟踪这些更改开始）。有关此页面更改的自动提示，请订阅 AWS CodeConnections 文档历史记录页面上的 RSS 源。

更改	描述	日期
AWSGitSyncServiceRolePolicy – 新策略	AWS CodeConnections 新增的策略。 授予权限以允许 AWS CodeConnections 用户使用 Git 同步功能与连接的基于 Git 的存储库进行同步。	2023 年 11 月 26 日
AWS CodeConnections 开启了跟踪更改	AWS CodeConnections 为其 AWS 托管策略开启了跟踪更改。	2023 年 11 月 26 日

AWS CodeStar 通知和 AWS CodeStar 连接的合规性验证

AWS CodeStar 通知和 AWS CodeStar 连接不在任何 AWS 合规计划的范围内。

有关特定合规计划范围内的 AWS 服务列表，请参阅按合规计划划分的范围内的AWS 服务。有关一般信息，请参阅 AWS 合规性计划。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅在 Artifact 中 AWS 下载报告。

您在使用 AWS CodeStar 通知和 AWS CodeStar 连接时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 AWS 提供了以下资源来帮助实现合规性：

- 安全与合规性快速入门指南 — 这些部署指南讨论了架构注意事项，并提供了在上部署以安全性和合规性为重点的基准环境的步骤。 AWS
- AWS 合规资源-此工作簿和指南集合可能适用于您的行业和所在地区。
- AWS Config— 该 AWS 服务评估您的资源配置在多大程度上符合内部实践、行业指导方针和法规。
- AWS Security Hub— 此 AWS 服务可全面了解您的安全状态 AWS ，帮助您检查是否符合安全行业标准和最佳实践。

AWS CodeStar 通知和 AWS CodeStar 连接中的故障恢复能力

AWS全球基础设施围绕AWS区域和可用区构建。AWS区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅 AWS 全球基础设施。

- 通知规则特定于创建它们的 AWS 区域。如果您在一个以上的 AWS 区域中有通知规则，则必须使用区域选择器来查看每个 AWS 区域中的通知规则。
- AWS CodeStar 通知依赖于 Amazon Simple Notification Service (Amazon SNS) 主题作为通知规则目标。这样，有关您的 Amazon SNS 主题和通知规则目标的信息可能会存储在您配置通知规则的区域之外的 AWS 区域中。

AWS CodeStar 通知和 AWS CodeStar 连接中的基础设施安全性

作为托管式服务中的功能，AWS CodeStar 通知和 AWS CodeStar 连接由 Amazon Web Services：安全流程概览白皮书中所述的 AWS 全球网络安全程序提供保护。

您使用 AWS 发布的 API 调用通过网络访问 AWS CodeStar 通知和 AWS CodeStar 连接。客户端必须支持传输层安全性 (TLS) 1.0 或更高版本。客户端还必须支持具有完全向前保密 (PFS) 的密码套件，例如 Ephemeral Diffie-Hellman (DHE) 或 Elliptic Curve Ephemeral Diffie-Hellman (ECDHE)。大多数现代系统支持这些模式。

必须使用访问密钥 ID 以及与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 AWS Security Token Service (AWS STS) 生成临时安全凭证来对请求进行签名。

跨区域的 AWS CodeConnections资源之间的流量

如果您使用连接功能启用资源的连接，则表示您同意并指示我们将与此类连接资源相关的信息存储在您使用底层服务的 AWS 区域以外的 AWS 区域中并予以处理，这些信息仅用于以下目的：提供与创建资源的区域以外的区域中的此类资源的连接。

有关更多信息，请参阅《 AWS CodeStar 连接》中的全球资源。

> **Note**
>
> 如果您使用连接功能为不要求先启用的区域中的资源启用连接，我们将存储和处理前述主题中详细介绍的信息。
> 对于在必须先启用的区域 [例如欧洲地区（米兰）区域] 中建立的连接，我们只会在该区域中存储和处理该连接的信息。

文档历史记录

下表介绍了此开发工具控制台版本的文档。

- AWS CodeStar 通知 API 版本：2019-10-15
- AWS CodeStar 通知 API 版本：2019-12-01

变更	说明	日期
支持 GitLab 自托管	增加了对配置用于 AWS 资源的连接和主机以与 GitLab 自托管进行交互的支持。有关更多信息，请参阅创建或更新主机的工作流程和创建到 GitLab 自托管的连接。	2023 年 12 月 28 日
新的存储库链接和针对连接的同步配置	添加了有关配置存储库链接和同步配置的信息。使用同步配置来同步 Git 存储库中的内容以更新 AWS CloudFormation 堆栈资源。有关更多信息，请参阅使用存储库链接和使用同步配置。	2023 年 11 月 27 日
对 Connections 服务相关角色的支持	增加了对配置连接以使用 Git 同步功能与 Git 存储库同步的支持。有关更多信息，请参阅使用 AWS CodeStar Connections 的服务相关角色和托管策略。	2023 年 11 月 26 日
支持 GitLab 群组	增加了对配置 AWS 资源连接以与 GitLab 群组进行交互的支持。有关更多信息，请参阅创建连接和创建到 GitLab 的连接。	2023 年 9 月 15 日

新的 GitLab 提供商类型	您现在可以创建到 GitLab 的连接。有关更多信息，请参阅创建连接和创建到 GitLab 的连接。	2023 年 8 月 10 日
通知规则的新目标类型	您现在可以选择为 Microsoft Teams 通道配置的 AWS Chatbot 客户端作为通知规则的目标。有关更多信息，请参阅创建通知规则和使用通知规则目标。	2023 年 5 月 17 日
连接在欧洲地区（米兰）区域可用	添加了有关欧洲地区（米兰）区域的连接的信息。有关更多信息，请参阅跨区域的 AWS CodeStar 连接资源之间的流量。	2023 年 5 月 17 日
对于使用存储库权限的连接错误添加了故障排除	在 GitHub 组织中创建与存储库的连接时，您必须是 GitHub 组织拥有者。有关更多信息，请参阅连接到 GitHub 时出现连接错误。	2022 年 8 月 29 日
添加了用于标记主机资源的信息	现在可以使用控制台和 CLI 标记主机。有关更多信息，请参阅标记 AWS CodeStar 连接中的资源。	2021 年 4 月 19 日
VPC 端点支持连接	现在可以配合连接使用 VPC 端点。有关更多信息，请参阅 AWS CodeStar 连接和接口 VPC 端点（AWS PrivateLink）。	2020 年 11 月 24 日

新的 GitHub 和 GitHub Enterprise Cloud 提供商类型	您现在可以创建到 GitHub 和 GitHub Enterprise Cloud 的连接。有关更多信息，请参阅创建连接和创建到 GitHub 的连接。	2020 年 9 月 30 日
添加了 GitHub Enterprise Server 提供程序类型和主机资源	有关连接的主机资源的信息已添加到本指南中。现在可以创建到 GitHub Enterprise Server 的连接。有关更多信息，请参阅创建连接和使用主机。这是开发工具控制台用户指南中连接功能的一般可用版。	2020 年 6 月 29 日
添加了有关使用和标记连接的信息	有关控制台中的连接功能的信息已添加到本指南中。您可以查看概念、入门步骤、包括示例策略在内的权限参考，以及创建、查看和标记连接的步骤。有关更多信息，请参阅什么是连接、连接概念、开始使用连接、创建连接、标记 AWS CodeStar 连接中的资源、安全、连接的配额、故障排除和使用 AWS CloudTrail 进行 AWS CodeStar 连接 API 调用。要查看其他提供程序操作（仅限权限操作）的列表，请参阅提供程序类型的操作。	2020 年 6 月 28 日
通知规则的新目标类型	您现在可以选择为 Slack 通道配置的 AWS Chatbot 客户端作为通知规则的目标。有关更多信息，请参阅创建通知规则和使用通知规则目标。	2020 年 4 月 2 日

增加了有关其他 AWS CodeCommit 事件的通知	现在，您可以为与拉取请求审批相关的事件配置通知。有关更多信息，请参阅存储库上的通知规则的事件和处理 CodeCommit 中的拉取请求。	2020 年 2 月 10 日
在额外的两个 AWS 区域中提供了通知	发人员工具控制台现在支持中东（巴林）和亚太地区（香港）的通知。有关更多信息，请参阅《AWS 一般参考》中的 AWS CodeStar 通知。	2020 年 2 月 5 日
增加了对加密的 Amazon SNS 主题的支持	增加了有关使用加密 Amazon SNS 主题作为通知目标的指南。有关更多信息，请参阅为通知配置 Amazon SNS 主题。	2020 年 2 月 4 日
通知可以包含 CodeCommit 的会话标签信息	通过使用会话标签，CodeCommit 的通知现在可以包含用户身份信息，例如显示名称或电子邮件地址。有关更多信息，请参阅概念和使用标签在 CodeCommit 中提供身份信息。	2019 年 12 月 19 日
初始版本	这是开发工具控制台用户指南的初始版本。	2019 年 11 月 5 日

AWS 术语表

有关最新的 AWS 术语，请参阅《AWS 词汇表参考》中的 AWS 词汇表。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。